

MODELO ACADÉMICO DE LA FORMACIÓN LABORAL BÁSICA EN DESARROLLO DE SOFTWARE

MÓDULO No. IV

APLICACIONES

MÓVILES

6^{to} Semestre

DIDÁCTICA PARA EL ESTUDIANTE

Nombre _____

Plantel _____ Grupo _____ Turno _____

2025 – 2026B

COLEGIO DE BACHILLERES DE TABASCO

MTRO. FERNANDO YRYS HERNÁNDEZ

Director General

MTRA. ANA FANNY HERNÁNDEZ MONTERO

Directora Académica

LIC. CELIS DEL CARMEN RANERO LARA

Subdirectora de Planeación Académica

MTRA. NATIVIDAD VERTIZ HERNÁNDEZ

Jefa de Departamento de Capacitación para el Trabajo

FORMACIÓN LABORAL BÁSICA EN **DESARROLLO DE SOFTWARE** MÓDULO IV. APLICACIONES MÓVILES

SUBMÓDULO I. PROGRAMACIÓN MÓVIL Y JUEGOS SUBMÓDULO II. DISEÑO Y PROGRAMACIÓN EN ROBÓTICA

En la realización del presente material participaron:

COORDINADOR: JORGE ARMANDO FLORES PALACIOS PLANTEL 6

ASESOR: ÁNGEL ALBERTO TORRES RAMÍREZ PLANTEL 6

EDICIÓN. 2025-2026B.

Revisado por.

LIC. MARÍA MARGARITA GARCÍA VIDAL

Este material fue elaborado bajo la coordinación y supervisión del Departamento de Capacitación para el Trabajo de la Dirección Académica del Colegio de Bachilleres de Tabasco y el contenido del mismo es gratuito, sin fines de lucro y con la responsabilidad de su buen uso para fines educativos

www.cobatab.edu.mx





“Educación que genera cambio”

CONTENIDO

MENSAJE DEL DIRECTOR GENERAL	I
PRESENTACIÓN	II
PROGRAMA DE ESTUDIOS DE LA FORMACIÓN LABORAL BÁSICA EN DESARROLLO DE SOFTWARE	III
FUNDAMENTACIÓN	IV
JUSTIFICACIÓN DE LA FORMACIÓN LABORAL BÁSICA EN DESARROLLO DE SOFTWARE	IX
ENFOQUE DE LA FORMACIÓN LABORAL BÁSICA.	XII
UBICACIÓN DE LA FORMACIÓN LABORAL	XVII
MAPA DE LA CAPACITACIÓN DESARROLLO DE SOFTWARE	XVIII
COMPETENCIAS LABORALES BÁSICAS	XIX
PROCESO DE EVALUACIÓN BAJO EL ENFOQUE EN COMPETENCIAS	XXII
EVALUACIÓN POR COMPETENCIAS	XXIV
INSTRUMENTOS DE EVALUACIÓN	XXIV
ROL DEL ESTUDIANTADO	XXVI
SIMBOLOGÍA.....	XXVII
TEMARIO	XXIX
SUBMÓDULO I. PROGRAMACIÓN MÓVIL Y JUEGOS	1
DOSIFICACIÓN PROGRAMÁTICA	2
CRITERIOS DE EVALUACIÓN	3
SITUACIÓN DIDÁCTICA.....	4
EVALUACIÓN DIAGNÓSTICA	5
LECTURA NO 1. INTRODUCCIÓN A LA PROGRAMACIÓN MÓVIL Y JUEGOS	8
PROGRAMACIÓN MÓVIL.....	9
PROGRAMACIÓN DE JUEGOS.....	11
ACTIVIDAD 1. GAMIFICACIÓN DE PROGRAMACIÓN	17
LECTURA NO 2. SOFTWARE DE DESARROLLO DE APLICACIONES MÓVILES	21
PROGRAMACIÓN POR BLOQUES.....	25
INSTALACIÓN Y USO BÁSICO DE UN PWA	27
ACTIVIDAD 2. IDENTIFICA TU ENTORNO DESARROLLADOR	31

“Educación que genera cambio”

LECTURA NO 3. DISEÑA UNA APP MÓVIL.....	33
DESARROLLO DE APP MÓVILES	36
ACTIVIDAD 3. FASES DE UNA APP	42
PRÁCTICA 1. PROPONIENDO MI APP.....	45
PRÁCTICA GUIADA “APLÍCATE PARA DIBUJAR”	47
LECTURA NO 4. PROGRAMACIÓN DE JUEGOS	59
ACTIVIDAD 4. TIMELINE: EVOLUCIÓN DE LOS VIDEOJUEGOS	63
DISEÑO DE VIDEOJUEGOS	65
ACTIVIDAD 5. CREACIÓN DE UN VIDEOJUEGO	71
DESARROLLO DE VIDEOJUEGOS PARA WEB	73
FUNCIONALIDAD DE VIDEOJUEGOS EN EL MÓVIL	76
ACTIVIDAD 6. DIFERENCIAS Y SIMILITUDES	79
SOFTWARE DE DESARROLLO PARA VIDEOJUEGOS	81
LECTURA NO 5. FUNCIONALIDADES: PROGRAMACIÓN EN SCRATCH	86
PRÁCTICA GUIADA “MOSTRAR Y ESCONDER”	90
PRÁCTICA GUIADA “DISFRACES Y SONIDOS”	92
PRÁCTICA GUIADA “MOVER OBJETO”	94
PRÁCTICA GUIADA “SUMAR PUNTOS”	96
PRÁCTICA 2. VAMO A JUGAR	98
INNOVANDO CON CREATIVIDAD	100
SUBMÓDULO II. DISEÑO Y PROGRAMACIÓN EN ROBÓTICA.....	103
DOSIFICACIÓN PROGRAMÁTICA	104
CRITERIOS DE EVALUACIÓN	105
SITUACIÓN DIDÁCTICA.....	106
EVALUACIÓN DIAGNÓSTICA	107
LECTURA NO 1. ORIGEN DE LA ROBÓTICA.....	109
ACTIVIDAD 1. PON A PRUEBA TUS CONOCIMIENTOS	115
LECTURA NO 2. ELEMENTOS EN LA ROBÓTICA	118



“Educación que genera cambio”

ROBÓTICA: TIPOS Y FASES	123
ACTIVIDAD 2. QUE NO SE TE PASE UN PASO	132
LECTURA NO 3. LA MECÁNICA DEL ROBOT	134
ACTIVIDAD 3. CUESTIONÁNDOME	139
ACTIVIDAD GUIADA: JUGANDO AL MECÁNICO	141
LECTURA NO 4. ENERGÍA Y FUERZA DE ROBOT	142
ACTIVIDAD 4. ANATOMÍA DEL ROBOT	148
ACTIVIDAD GUIADA: ARTICULANDO MI BRAZO	150
LECTURA NO 5. PROGRAMACIÓN DE ROBOTS	151
AUTODESK TINKERCAD	157
INSTALACIÓN Y USO BÁSICO DE TINKERCAD.....	160
PRÁCTICA 1. HOLA MUNDO EN ARDUINO.....	163
PRÁCTICA 2. ¿QUÉ ME HACE FALTA?.....	168
PRÁCTICA 3. SEMÁFORO	171
PRÁCTICA 4. CABLEADO DE MI DRON	179
AUTOMATIZANDO MI CREATIVIDAD.....	182
ANEXOS	186



GUÍA DE ESTUDIANTES

Estimada comunidad estudiantil del Colegio de Bachilleres de Tabasco: quiero hacer énfasis en nuestra Institución, la cual tiene 5 décadas forjando el aprendizaje de cada uno de los jóvenes, hoy muchos de los que han pasado por nuestras aulas de clases son profesionistas exitosos.

Dentro del marco del 50 aniversario COBATAB, me permito como Director General enviarles un cordial saludo y dirigirme a ustedes a través de esta guía de estudio diseñada para acompañarlos como una herramienta en su formación académica y personal.

En esta etapa de su vida, el aprendizaje autónomo es clave para descubrir sus talentos, fortalecer sus habilidades y construir un futuro con propósito. Cada actividad, cada lectura y cada reto que encuentren en esta guía de estudio está pensado para impulsar su Desarrollo integral.

Les invito a asumir con responsabilidad su proceso de aprendizaje, a colaborar con sus compañeros, docentes y a mantener siempre una actitud de respeto y apertura. En el Colegio de Bachilleres de Tabasco promovemos la equidad y creemos firmemente en el potencial de cada uno de ustedes.

Esta guía de estudio busca ser clara y accesible, con objetivos definidos y contenidos relevantes para su contexto. Les animo a aprovecharla al máximo, a preguntar, reflexionar y aplicar lo aprendido en su vida diaria.

Confío en su capacidad para superar desafíos, alcanzar metas y contribuir positivamente a su comunidad. Cuentan con el apoyo de sus docentes y de todos los que integramos esta institución.

Con entusiasmo y compromiso, sigamos construyendo juntos una educación de calidad, que genera cambio.

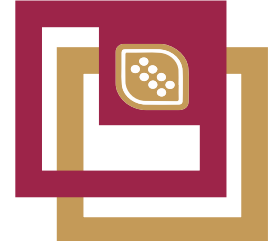
Con aprecio y confianza,

Fernando Yrys Hernández

Director General

Colegio de Bachilleres de Tabasco





“Educación que genera cambio”

Presentación

Desde su creación, el Colegio de Bachilleres de Tabasco tiene como objetivo impartir educación de nivel medio superior en la modalidad de bachillerato general, operando desde el año 2009 un plan de estudios bajo un enfoque por competencias a través del Marco Curricular Común definiendo así los rasgos del perfil del egresado de la EMS en donde las competencias genéricas en su conjunto delinean el perfil deseable del joven bachiller.

En este sentido, los planteamientos del Marco Curricular Común de la Educación Media Superior (MCCEMS)¹ buscan una formación integral en el estudiantado mediante el desarrollo de la capacidad creadora, productiva, libre y digna del ser humano, con amor al país, a su cultura e historia. Por ello, el Bachillerato General plantea las diversas Unidades de Aprendizaje Curricular (UAC) y Formaciones para el Trabajo para que con sus estudiantes egresados y egresadas contribuya al logro de su objetivo específico, el cual radica en la “conformación de una ciudadanía reflexiva, con capacidad de formular y asumir responsabilidades de manera comunitaria, interactuar en contextos plurales y propositivos, trazarse metas y aprender de manera continua y colaborativa”.

El Colegio de Bachilleres de Tabasco a través de su Competente de Formación Laboral Básica ofrece en ocho de sus planteles la capacitación de **Desarrollo de Software** con la cual se busca desarrollar en el estudiantado competencias laborales básicas, que les permitan aplicar en forma integrada los conocimientos, destrezas, habilidades, actitudes y valores con responsabilidad y autonomía para desenvolverse en contextos específicos del desarrollo personal, académico, social y profesional en situaciones de la vida común, de estudio o trabajo a lo largo de la vida.

En ese contexto, conforme a las necesidades sociales, económicas y demográficas en relación y congruencia con los objetivos y líneas de acción del programa institucional para el Colegio de Bachilleres de Tabasco, se presenta la formación laboral básica en **Desarrollo de Software**, específica del bachillerato general, con objetivos delimitados acorde a las características del subsistema y la necesidad del conglomerado social en el cual será aplicado, situación que deviene

¹ El cual puede ser consultado a través del siguiente enlace:

<https://educacionmediasuperior.sep.gob.mx/work/models/sems/Resource/13516/1/image s/Documento%20base%20MCCEMS.pdf>



"Educación que genera cambio"

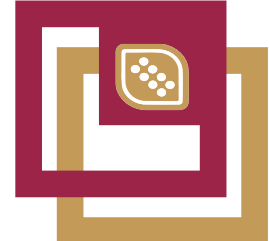
de tendencias académicas que obligan a este subsistema a tener en consideración las áreas de conocimiento y perfil profesional y su estrecha relación con la demanda ocupacional y la oferta laboral, esto con plena convicción de que los estudiantes del Colegio de Bachilleres de Tabasco concluyan satisfactoriamente con sus estudios de educación media superior e ingresen a sus estudios de nivel superior con herramientas y conocimientos que son de interés general conforme al área geográfica y entorno social en el que se desarrollen; por lo que el presente documento se encuentra conformado en apartados mediante los cuales describe la justificación y los elementos claves para su implementación en el aula.

Programa de Estudios de la Formación Laboral Básica en Desarrollo de Software

Semestre	Tercero, Cuarto, Quinto y Sexto	
Créditos	56 totales / 14 por semestre	
Componente	De formación laboral	
Nivel de formación laboral	Básica	
Tiempo asignado	Mediación docente	Estudio independiente
	448 h. totales 112 h. por semestre	112 h. totales 28 h. por semestre
Sector productivo	Servicios profesionales, Científicos y Técnicos. ²	



² Acorde al RENECE – Registro Nacional de Estándares de Competencia por Sector Productivo. Disponible en <https://conocer.gob.mx/renec-registro-nacional-de-estandares-de-competencia-porsector-productivo/>



Fundamentación

La Dirección General del Colegio de Bachilleres de Tabasco en coordinación con la Dirección General del Bachillerato (DGB), acorde a la Nueva Escuela Mexicana (NEM) y al Marco Curricular Común de la Educación Media Superior (MCCEMS), y en su responsabilidad de enfrentar tanto los nuevos retos como los objetivos que de estos se desprenden, actualiza el presente Programa de estudios, el cual responde a una visión de educación integral, pertinente, de calidad y excelencia.

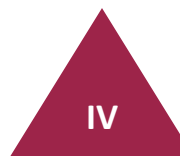
Dicho programa forma parte del Componente de Formación Laboral Básica del Bachillerato General, el cual busca ser un espacio vinculado con el sector productivo, permitiendo al estudiantado no solo cumplir con su trayecto educativo, sino construir su proyecto de vida, con mayores posibilidades de inserción en el mercado de trabajo.

Esto atendiendo al mandato constitucional que en materia educativa, con base en la Reforma Constitucional a los artículos 3°, 31° y 73° de la Constitución Política de los Estados Unidos Mexicanos y de la emisión de la Legislación Secundaria, publicadas el 30 de septiembre de 2019 en el Diario Oficial de la Federación, determinan la reorientación del Sistema Educativo Nacional para “garantizar el derecho a la educación con un enfoque de derechos humanos y de igualdad sustantiva, para incidir en la cultura educativa mediante la corresponsabilidad y el impulso de transformaciones sociales dentro de la escuela y en la comunidad”.

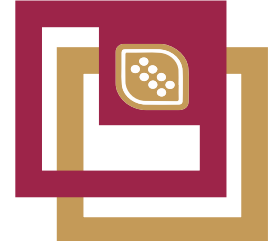
Bajo este contexto, es que el Componente de Formación Laboral Básica, adquiere mayor relevancia, pues tendrá como ejes rectores:

- **Enfoque en competencias**, que busca desarrollar las capacidades y habilidades necesarias para el desempeño laboral.
- **Enfoque humanista**, que valora y respeta la diversidad y la dignidad del estudiantado, su potencial creativo, su participación, su bienestar integral y su compromiso social.

Estos enfoques se orientan a promover una educación inclusiva, equitativa y de calidad, que favorezca el aprendizaje permanente y el máximo logro de los aprendizajes.



“Educación que genera cambio”



Así pues, el Componente de Formación Laboral Básica, busca desarrollar en el estudiantado competencias laborales básicas, que les permitan aplicar en forma integrada los conocimientos, destrezas, habilidades, actitudes y valores con responsabilidad y autonomía para desenvolverse en contextos específicos del desarrollo personal, académico, social y profesional en situaciones de la vida común, de estudio o trabajo a lo largo de la vida.³

Para lograr dicho propósito, lo que a continuación se presenta es una serie de competencias laborales básicas las cuales permitirán al personal docente el abordaje de aprendizajes con la amplitud y profundidad acorde a los diversos contextos.

Es decir, a partir de estas competencias, se diseñarán estrategias de enseñanza-aprendizaje que permita a las y los estudiantes ser capaces de conducir su vida hacia su futuro con bienestar y satisfacción, con sentido de pertenencia social, conscientes de los problemas sociales, económicos y políticos que aquejan al país, pero también de su entorno inmediato, dispuestos a participar de manera responsable y decidida en los procesos de democracia participativa y a comprometerse en las soluciones de las problemáticas que los aquejan y que tengan la capacidad de aprender a aprender en el trayecto de su vida.⁴

Para lo cual, es de primordial importancia visualizar que debe existir una articulación contextualizada del Componente Laboral Básico, con el Currículo Fundamental y el Ampliado, para garantizar así la transferencia de los conocimientos, experiencias, habilidades, capacidades, actitudes y valores.

Es decir, esta articulación permitirá:

- La transversalidad del conocimiento adquirido en el Currículo Fundamental con las competencias laborales básicas requeridas en el mercado laboral.
- Fomentar el aprendizaje en contextos diversos, utilizando métodos y estrategias de aprendizaje.
- Centrar las necesidades del mercado laboral.

³ Subsecretaría de Educación Media Superior. (2023b). El currículum laboral en la educación media superior. SEP.

⁴ Subsecretaría de Educación Media Superior. (2023b). El currículum laboral en la educación media superior. SEP.





“Educación que genera cambio”

- Fomentar la interacción entre la vida educativa y la comunidad, a fin de propiciar la adquisición de conocimientos y competencias, de acuerdo con el desarrollo biopsicosociocultural del estudiantado.
- Aprovechar, mediante el Programa Aula, Escuela, Comunidad (PAEC), todas las oportunidades para poner en práctica lo que se ha aprendido; su aplicación no se limitará solo a los recursos sociocognitivos y a las áreas de conocimiento, sino que abarcará los aspectos funcionales (competencias laborales) y recursos socioemocionales.
- Mejorar la relación entre la escuela y los sectores productivos.⁵

El componente de Formación Laboral Básico aporta al estudiantado elementos que le permiten iniciarse en diversos aspectos del sector productivo, fomentando una actitud positiva hacia el trabajo y en su caso, su integración al mismo. Los módulos que conforman este programa son el resultado del trabajo colegiado con personal docente que imparte esta capacitación en los diferentes subsistemas coordinados por esta Dirección General, quienes brindan su experiencia y conocimientos buscando responder a los diferentes contextos existentes en el país, así como a la formación de una ciudadanía socialmente útil, para que el estudiantado cuente con la opción de iniciar una ruta laboral que le promueva una proyección hacia las diferentes modalidades laborales.

Aunado a ello, en virtud de que la Educación Media Superior debe favorecer la convivencia, el respeto a los derechos humanos y la responsabilidad social, el cuidado de las personas, el entendimiento del entorno, la protección del medio ambiente, la puesta en práctica de habilidades productivas para el desarrollo integral de los seres humanos, la actualización del presente programa de estudios, incluye temas transversales que según Figueroa de Katra (2005), enriquecen la labor formativa de manera tal que conectan y articulan los saberes de los distintos sectores de aprendizaje que dotan de sentido a los conocimientos disciplinares, con los temas y contextos sociales, culturales y éticos presentes en su entorno; buscan mirar toda la experiencia escolar como una oportunidad para que los aprendizajes integren sus dimensiones cognitivas y formativas, favoreciendo de esta forma una educación incluyente y con equidad.

VI

⁵ Subsecretaría de Educación Media Superior. (2023b). El currículum laboral en la educación media superior. SEP.



“Educación que genera cambio”

De igual forma, con base en el fortalecimiento de la educación para la vida, se abordan dentro de este programa de estudios los temas transversales, mismos que se clasifican a través de ejes temáticos de los campos Social, Ambiental, Salud y Habilidad Lectora como el componente básico, con la particularidad de que se complementan con características propias de la formación para el trabajo. Dichos temas no son únicos ni pretenden imitar el quehacer educativo en el aula, ya que es necesario tomar en consideración temas propios de cada comunidad, por lo que el personal docente podrá considerar ya sea uno o varios, en función del contexto escolar y de su pertinencia en cada submódulo:

- Eje transversal Emprendimiento: se sugiere retomar temas referentes a la detección de oportunidades y puesta en práctica de acciones que contribuyen a la demostración de actitudes tales como iniciativa, liderazgo, trabajo colaborativo, visión, innovación y creatividad promoviendo la responsabilidad social.
- Eje transversal Vinculación Laboral: se recomienda abordar temas referentes a la realización de acciones que permiten al estudiantado identificar los sitios de inserción laboral o autoempleo.
- Eje transversal Iniciar, Continuar y Concluir sus estudios de nivel superior: se recomienda abordar temas referentes a los mecanismos que permiten al estudiantado reflexionar sobre la importancia de darle continuidad a sus estudios superiores.

Asimismo, otro aspecto importante que promueve el programa de estudios es la interdisciplinariedad entre asignaturas del mismo semestre, en donde diferentes disciplinas se conjuntan para trabajar de forma colaborativa para la obtención de resultados en los aprendizajes esperados de manera integral permitiendo al estudiantado confrontarse a situaciones cotidianas aplicando dichos saberes de forma vinculada.

Por otro lado, en cada submódulo se observa la relación de las competencias genéricas y profesionales básicas, los conocimientos, las habilidades y actitudes que darán como resultado los aprendizajes esperados, permitiendo llevar de la mano al personal docente con el objetivo de generar un desarrollo progresivo no sólo de los conocimientos, sino también de aspectos actitudinales.



“Educación que genera cambio”

En este sentido, el rol docente dentro del proceso de enseñanza – aprendizaje, tiene un papel fundamental, como lo establece el Acuerdo Secretarial 4474, ya que el profesorado que imparte el componente de formación profesional, es quien facilita el proceso educativo al diseñar actividades significativas que promuevan el desarrollo de las competencias (conocimientos, habilidades y actitudes); propicia un ambiente de aprendizaje que favorece el conocimiento social, la colaboración, la toma responsable de decisiones y al perseverancia a través del desarrollo de habilidades socioemocionales del estudiantado, tales como la confianza, seguridad, autoestima, entre otras, propone estrategias disciplinares y transversales en donde el objetivo no es la formación de técnicos en diferentes actividades productivas, sino la promoción de las diferentes competencias profesionales básicas que permitan a la población estudiantil del Bachillerato General tener alternativas para iniciar una ruta a su integración laboral, favoreciendo el uso de herramientas tecnológicas de la información y la comunicación; así como el diseño de instrumentos de evaluación que atiendan al enfoque por competencias.

Es por ello que la Dirección General del Bachillerato a través del Trabajo colegiado busca promover una mejor formación docente a partir de la creación de redes de gestión escolar, analizar los indicadores del logro académico del estudiantado, generar técnicas exitosas de trabajo en el aula, compartir experiencias de manera asertiva, exponer problemáticas comunes que presenta el estudiantado respetando la diversidad de opiniones y mejorar la práctica pedagógica, donde es responsabilidad del profesorado realizar secuencias didácticas innovadoras a partir del análisis de los programas de estudio, promoviendo el desarrollo de habilidades socioemocionales y el abordaje de temas transversales de manera interdisciplinar; rediseñar las estrategias de evaluación y generar materiales didácticos.

Finalmente, este programa de estudios brinda herramientas disciplinares y pedagógicas al personal docente, quienes deberán, a través de los elementos antes mencionados, potenciar el papel de los educandos como gestores autónomos de su propio aprendizaje, promoviendo la participación creativa de las nuevas generaciones en la economía, en el ámbito laboral, la sociedad y la cultura, reforzar el proceso de formación de la personalidad, construir un espacio valioso para la adopción de valores y el desarrollo de actitudes positivas para la vida.





“Educación que genera cambio”

Justificación de la Formación Laboral Básica en Desarrollo de Software

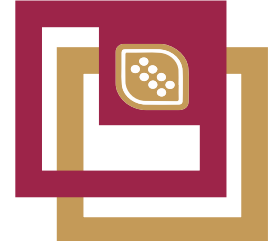
La Formación Laboral Básica de Desarrollo de Software se ubica dentro de los Recursos Sociocognitivos, dentro de las disciplinas de Comunicación y Matemáticas, las cuales en el MCCEMS tienen en el proceso de aprendizaje, la función de ampliar, potenciar y consolidar el conocimiento de la experiencia; permitiendo aprovechar y ampliar los conocimientos de las Áreas de Conocimiento, asimismo, los Recursos Sociocognitivos contribuyen a desarrollar capacidades, destrezas, habilidades, actitudes y valores en las y los estudiantes, brindando la posibilidad de construir la propia experiencia, para que sepan qué hacer con el conocimiento que tienen, sepan actuar, entendiendo lo que hacen, comprendiendo cómo participar y colaborar, asumiendo la responsabilidad de las acciones realizadas, sus implicaciones y consecuencias, y transformando los contextos locales y comunitarios en pro del bien común.

Con base en ello, a lo largo del Componente de Formación Laboral Básico, se brindará una formación de carácter genérico y transversal que le permita al estudiantado incorporarse al sector productivo con actividades contextualizadas, de autonomía y responsabilidad mínima, fomentando una actitud positiva hacia el trabajo y en su caso, su integración al mismo.

De manera específica, a través de la Formación Laboral Básica en Desarrollo de Software, tiene como eje acercar al estudiantado al desarrollo científico - tecnológico en el sector de los servicios profesionales, científicos y técnicos, como en las Tecnologías de la Información. Evaluando estrategias de solución a problemáticas en diversas áreas, utilizando software de programación de alto nivel, clasificando y diseñando el software mediante diversos lenguajes de programación, para mejorar su entorno, demostrando conciencia social y responsabilidad ante dichas problemáticas.

Con base en lo anterior, esta Formación Laboral Básica tiene el propósito general de: Evaluar estrategias para el desarrollo de software, de manera creativa e innovadora; identificando, clasificando y diseñando software mediante diversos lenguajes de programación, para mejorar su entorno, demostrando conciencia social y responsabilidad ante las problemáticas de su contexto, fomentando el trabajo colaborativo y propositivo.





“Educación que genera cambio”

El uso de los lenguajes de programación en esta capacitación demuestra el manejo en forma avanzada de dichos lenguajes para la solución de diversas problemáticas de la vida cotidiana y del contexto, con un desarrollo metodológico, creativo y comunicativo en pro del beneficio personal y social.

De esta manera, las personas egresadas de esta Formación Laboral Básica en Desarrollo de Software se pueden integrar a la vida laboral tanto en instituciones públicas como privadas en los siguientes perfiles ocupacionales: en educación, ciberseguridad, visualización y gestión de datos, comercio electrónico, aplicaciones móviles, robótica y en el desarrollo web.

Los módulos que conforman este programa son el resultado del trabajo colegiado con personal docente que imparte esta capacitación en los diferentes subsistemas coordinados por esta Dirección General, quienes brindan su experiencia y conocimientos buscando responder a los diferentes contextos existentes en el país, así como a la formación de una ciudadanía socialmente útil, para que el estudiantado cuente con la opción de iniciar una ruta laboral que le promueva una proyección hacia las diferentes modalidades laborales.

La capacitación de Desarrollo de software busca que el estudiantado desarrolle las competencias profesionales en el desarrollo de sistemas y aplicaciones para Internet y celular, en donde también se desarrollan las competencias genéricas, la interdisciplinaridad y los ejes transversales de vinculación laboral, emprendimiento y la continuación de sus estudios a nivel superior.

La Capacitación de Desarrollo de Software en la formación para el trabajo del estudiantado está basada en las Normas Técnicas de Competencia Laboral (NTCL) del Consejo Nacional de Normalización y Certificación de Competencias Laborales (CONOCER), son una necesidad para cumplir con las exigencias del mundo actual y de los sectores productivos, porque hoy en día se exige tener trabajadores calificados, capaces de desarrollar en todo momento las áreas de la organización en la cual están inmersos, promoviendo los productos o servicios en el entorno nacional o internacional, proporcionando las herramientas y técnicas que son básicas para los egresados del nivel medio superior, que les va a permitir vencer todas las fronteras e incorporarse al mundo globalizado por medio de la programación, así como de las Tecnologías de la Información y de la Comunicación (TIC'S) y de la utilización de las Tecnologías del Aprendizaje y del Conocimiento (TAC'S).





“Educación que genera cambio”

Además, para consolidar la formación profesional y el compromiso con el medio ambiente, enfrentando los desafíos del mundo actual en materia de producción y consumo responsable; se consideran las metas de los 17 objetivos de la Agenda 2030 de acuerdo con la situación contextualizada de cada comunidad, para hacer más y mejores cosas con menos recursos, incrementando las ganancias netas de bienestar de las actividades económicas mediante la reducción de la utilización de los recursos, la degradación y la contaminación durante todo el ciclo de vida, logrando al mismo tiempo una mejor calidad de vida.⁶

De igual manera, con el desarrollo de las Tecnologías diversas y el desarrollo de habilidades como el pensamiento crítico, la toma de decisiones, la resolución de problemas, la comunicación efectiva y el trabajo colaborativo. La inteligencia artificial (IA) es la base a partir de la cual se imitan los procesos de inteligencia humana mediante la creación y la aplicación de algoritmos creados en un entorno dinámico de computación. Con esto se pretende que los ordenadores piensen y actúen como los humanos. Tomando tres componentes fundamentales:

- Sistemas computacionales.
- Datos y gestión de estos.
- Algoritmos para tomar decisiones (IA avanzados, código).⁷

Con todo esto, la Formación Laboral Básica de Desarrollo de Software pretende atender las demandas de desarrollo en materia de cómputo y servicios a través de tecnologías aplicadas, así como el desarrollo científico y tecnológico para el progreso humano social y ambiental productivo.

De esta manera, los estudiantes de Desarrollo de Software pueden proporcionar soluciones de software: aplicaciones, programas y sistemas que permitan atender las necesidades sociales y ambientales para bien de todos. Consolidando las competencias laborales en torno a la situación del contexto de los estudiantes, así como del país.

El Desarrollo de Software considera la metodología STEAM, donde los estudiantes de la capacitación pueden desarrollar proyectos que integran las disciplinas de ciencia, tecnología, ingeniería, arte y matemáticas (STEAM). Con la particularidad de la interdisciplinariedad vinculando los saberes y conocimientos en la vinculación con las instituciones de formación profesional universitarias.

⁶ La Agenda 2030 y los Objetivos de Desarrollo Sostenible. Una oportunidad para América Latina y el Caribe.

⁷ Inteligencia Artificial aplicada a la solución de problemas nacionales. Consuelo Doddoli, Ciencia UNAM-DGDC.





Enfoque de la Formación Laboral Básica.

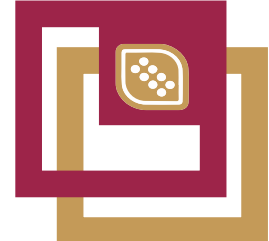
La Formación Laboral Básica de Desarrollo de Software se ubica dentro de los Recursos Sociocognitivos, dentro de las disciplinas de Comunicación y Matemáticas, las cuales en el MCCEMS tienen en el proceso de aprendizaje, la función de ampliar, potenciar y consolidar el conocimiento de la experiencia; permitiendo aprovechar y ampliar los conocimientos de las Áreas de Conocimiento, asimismo, los Recursos Sociocognitivos contribuyen a desarrollar capacidades, destrezas, habilidades, actitudes y valores en las y los estudiantes, brindando la posibilidad de construir la propia experiencia, para que sepan qué hacer con el conocimiento que tienen, sepan actuar, entendiendo lo que hacen, comprendiendo cómo participar y colaborar, asumiendo la responsabilidad de las acciones realizadas, sus implicaciones y consecuencias, y transformando los contextos locales y comunitarios en pro del bien común.

Con base en ello, a lo largo del Componente de Formación Laboral Básico, se brindará una formación de carácter genérico y transversal que le permita al estudiantado incorporarse al sector productivo con actividades contextualizadas, de autonomía y responsabilidad mínima, fomentando una actitud positiva hacia el trabajo y en su caso, su integración al mismo.

Con base en ello, a lo largo del Componente de Formación Laboral Básico, se brindará una formación de carácter genérico y transversal que le permita al estudiantado incorporarse al sector productivo con actividades contextualizadas, de autonomía y responsabilidad mínima, fomentando una actitud positiva hacia el trabajo y en su caso, su integración al mismo.

De manera específica, a través de la Formación Laboral Básica en Desarrollo de Software, tiene como eje acercar al estudiantado al desarrollo científico - tecnológico en el sector de los servicios profesionales, científicos y técnicos, como en las Tecnologías de la Información. Evaluando estrategias de solución a problemáticas en diversas áreas, utilizando software de programación de alto nivel, clasificando y diseñando el software mediante diversos lenguajes de programación, para mejorar su entorno, demostrando conciencia social y responsabilidad ante dichas problemáticas.





“Educación que genera cambio”

Con base en lo anterior, esta Formación Laboral Básica tiene el **propósito general** de: Evaluar estrategias para el desarrollo de software, de manera creativa e innovadora; identificando, clasificando y diseñando software mediante diversos lenguajes de programación, para mejorar su entorno, demostrando consciencia social y responsabilidad ante las problemáticas de su contexto, fomentando el trabajo colaborativo y propositivo.

El uso de los lenguajes de programación en esta capacitación demuestra el manejo en forma avanzada de dichos lenguajes para la solución de diversas problemáticas de la vida cotidiana y del contexto, con un desarrollo metodológico, creativo y comunicativo en pro del beneficio personal y social.

De esta manera, las personas egresadas de esta Formación Laboral Básica en Desarrollo de Software se pueden integrar a la vida laboral tanto en instituciones públicas como privadas en los siguientes perfiles ocupacionales: en educación, ciberseguridad, visualización y gestión de datos, comercio electrónico, aplicaciones móviles, robótica y en el desarrollo web.

Los módulos que conforman este programa son el resultado del trabajo colegiado con personal docente que imparte esta capacitación en los diferentes subsistemas coordinados por esta Dirección General, quienes brindan su experiencia y conocimientos buscando responder a los diferentes contextos existentes en el país, así como a la formación de una ciudadanía socialmente útil, para que el estudiantado cuente con la opción de iniciar una ruta laboral que le promueva una proyección hacia las diferentes modalidades laborales.

La capacitación de Desarrollo de software busca que el estudiantado desarrolle las competencias profesionales en el desarrollo de sistemas y aplicaciones para Internet y celular, en donde también se desarrollan las competencias genéricas, la interdisciplinaridad y los ejes transversales de vinculación laboral, emprendimiento y la continuación de sus estudios a nivel superior.

La Capacitación de Desarrollo de Software en la formación para el trabajo del estudiantado está basada en las Normas Técnicas de Competencia Laboral (NTCL) del Consejo Nacional de Normalización y Certificación de Competencias Laborales (CONOCER), son una necesidad para cumplir con las exigencias del mundo actual y de los sectores productivos, porque hoy en día se exige tener trabajadores calificados, capaces de desarrollar en todo momento las áreas de la organización en la cual están inmersos, promoviendo los productos o servicios en el entorno





“Educación que genera cambio”

nacional o internacional, proporcionando las herramientas y técnicas que son básicas para los egresados del nivel medio superior, que les va a permitir vencer todas las fronteras e incorporarse al mundo globalizado por medio de la programación, así como de las Tecnologías de la Información y de la Comunicación (TIC’S) y de la utilización de las Tecnologías del Aprendizaje y del Conocimiento (TAC’S).

Además, para consolidar la formación profesional y el compromiso con el medio ambiente, enfrentando los desafíos del mundo actual en materia de producción y consumo responsable; se consideran las metas de los 17 objetivos de la Agenda 2030 de acuerdo con la situación contextualizada de cada comunidad, para hacer más y mejores cosas con menos recursos, incrementando las ganancias netas de bienestar de las actividades económicas mediante la reducción de la utilización de los recursos, la degradación y la contaminación durante todo el ciclo de vida, logrando al mismo tiempo una mejor calidad de vida.⁸

De igual manera, con el desarrollo de las Tecnologías diversas y el desarrollo de habilidades como el pensamiento crítico, la toma de decisiones, la resolución de problemas, la comunicación efectiva y el trabajo colaborativo. La inteligencia artificial (IA) es la base a partir de la cual se imitan los procesos de inteligencia humana mediante la creación y la aplicación de algoritmos creados en un entorno dinámico de computación. Con esto se pretende que los ordenadores piensen y actúen como los humanos. Tomando tres componentes fundamentales:

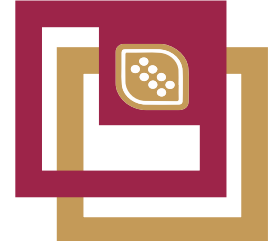
- Sistemas computacionales.
- Datos y gestión de estos.
- Algoritmos para tomar decisiones (IA avanzados, código).⁹

Con todo esto, la Formación Laboral Básica de Desarrollo de Software pretende atender las demandas de desarrollo en materia de cómputo y servicios a través de tecnologías aplicadas, así como el desarrollo científico y tecnológico para el progreso humano social y ambiental productivo.

⁸ La Agenda 2030 y los Objetivos de Desarrollo Sostenible. Una oportunidad para América Latina y el Caribe.

⁹ Inteligencia Artificial aplicada a la solución de problemas nacionales. Consuelo Doddoli, Ciencia UNAM-DGDC.





“Educación que genera cambio”

De esta manera, los estudiantes de Desarrollo de Software pueden proporcionar soluciones de software: aplicaciones, programas y sistemas que permitan atender las necesidades sociales y ambientales para bien de todos. Consolidando las competencias laborales en torno a la situación del contexto de los estudiantes, así como del país.

El Desarrollo de Software considera la metodología STEAM, donde los estudiantes de la capacitación pueden desarrollar proyectos que integran las disciplinas de ciencia, tecnología, ingeniería, arte y matemáticas (STEAM). Con la particularidad de la interdisciplinariedad vinculando los saberes y conocimientos en la vinculación con las instituciones de formación profesional universitarias.

El contenido de la capacitación de Desarrollo de software se divide en cuatro módulos, impartidos a partir del tercer semestre con una carga de 7 horas semanales, cada módulo se integra por dos submódulos en los que se busca desarrollar en el estudiantado la creación de programas con características avanzadas, utilizando C++, Java, Visual .NET, así como la creación de aplicaciones móviles y juegos, además de diseño y programación en robótica, esto con el fin de desarrollar software con bases de datos y creación de páginas web comunicando ideas e información en el entorno laboral y escolar.

XV

El Módulo I. Programación básica, el estudiantado revisará sistemas de información mediante software de programación de alto nivel.

El Módulo II. Base de datos y desarrollo web, en este apartado se analizará la creación de bases de datos y sitios web mediante el software de programación adecuado.

El Módulo III. Aplicaciones para web, el estudiantado utilizará software de aplicación y elementos multimedia para crear sitios web.

El Módulo IV. Aplicaciones móviles, en este apartado se construirá soluciones para aplicaciones móviles -y juegos, así como propuestas para robótica básica respetando su arquitectura y anatomía. Todas las competencias mencionadas hacen posible en el egresado tener los conocimientos, técnicas, métodos y lenguajes necesarios en el desarrollo de software para ingresar a estudios superiores y desempeñarse de forma eficiente, además de desarrollar las habilidades y actitudes necesarias para la realización de una actividad productiva socialmente útil como auxiliar en áreas de desarrollo de software en diferentes instituciones públicas o privadas.





La Capacitación de Desarrollo de Software en la formación laboral básica del estudiantado está basada en las Normas Técnicas de Competencia Laboral (NTCL) del Consejo Nacional de Normalización y Certificación de Competencias Laborales (CONOCER), son una necesidad para cumplir con las exigencias del mundo actual y de los sectores productivos, porque hoy en día se exige tener trabajadores calificados, capaces de desarrollar en todo momento las áreas de la organización en la cual están inmersos, promoviendo los productos o servicios en el entorno nacional o internacional, proporcionando las herramientas y técnicas que son básicas para los egresados del nivel medio superior, que les va a permitir vencer todas las fronteras e incorporarse al mundo globalizado por medio de la programación, así como de las Tecnologías de la información y de la comunicación (TIC'S) y de la utilización de las Tecnologías del aprendizaje y del conocimiento (TAC'S).

Así mismo se señalan algunas normas de competencia laboral vigentes, avaladas por el Consejo Nacional de Normalización y Certificación de Competencias Laborales, que podrán servir de guía para el desarrollo de los módulos:

NMX-J-515-ANCE-2021, Equipos De Control Y Distribución-Requisitos Generales De Seguridad.

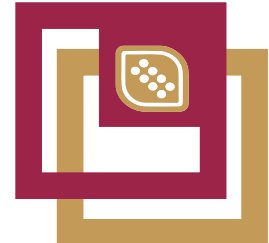
NOM-003-SCFI-2014 “Productos Eléctricos – especificaciones de seguridad”.

UINF0947.01 Operar las herramientas de cómputo en ambiente de red. **EC0835** Ejecución de software con codificación de comandos y datos orientada a objetos. **EC0160** Desarrollo de código de software.





TOTAL DE HORAS CON DOCENTE SEMANA MES: 187
TOTAL DE HORAS SEMANA MES: 233.75
TOTAL DE HORAS: 3740
TOTAL DE CRÉDITOS: 374



Mapa de la Capacitación Desarrollo de Software

Módulo II: Base de Datos y Desarrollo Web

Módulo I: Programación Básica

Submódulo 1	Programación en C++
	48 horas de mediación docente 12 horas de estudio independiente 6 créditos
Submódulo 2	Programación en Java
	64 horas de mediación docente 16 horas de estudio independiente 8 créditos

Nota: Elaboración propia.

Submódulo 1	Desarrollo de Aplicaciones de Escritorio con Base de Datos
	48 horas de mediación docente 12 horas de estudio independiente 6 créditos
Submódulo 2	Programación Web
	64 horas de mediación docente 16 horas de estudio independiente 8 créditos

Nota: Elaboración propia.

Módulo III: Aplicaciones Web

Submódulo 1	Aplicaciones para Web
	48 horas de mediación docente 12 horas de estudio independiente 6 créditos
Submódulo 2	Herramientas de Diseño para Web
	64 horas de mediación docente 16 horas de estudio independiente 8 créditos

Nota: Elaboración propia.

Módulo IV: Aplicaciones Móviles

Submódulo 1	Programación Móvil y Juegos
	48 horas de mediación docente 12 horas de estudio independiente 6 créditos
Submódulo 2	Diseño y Programación en Robótica
	64 horas de mediación docente 16 horas de estudio independiente 8 créditos

XVIII





Competencias Laborales Básicas

Hacen referencia a la capacidad para aplicar conocimientos, destrezas, habilidades, actitudes y valores en el desarrollo personal, académico, social y profesional en situaciones de la vida común, de estudio o trabajo para que el estudiantado desarrolle la formación fundamental o laboral básica, que les permite desempeñar **funciones laborales de nivel dos** de competencia, aplicando soluciones a problemas simples en contextos conocidos y específicos.

Tienen validez oficial dentro del Sistema Educativo Nacional (SEN), lo cual se expresa con la emisión del documento que acredita su formación.

Estas competencias se caracterizan por:

- **Pertinencia:** Atiende a las necesidades del sector productivo y son valoradas.
- **Relevancia:** Favorece la empleabilidad y emprendimientos productivos, sin disparidades de género, étnicas o exclusión de grupos vulnerables.
- **Coherencia:** Acorde al tipo educativo de media superior.
- **Suficiencia:** Abarca un proceso completo e independiente a las demás funciones que desempeña quien egresa de la formación para el trabajo, técnica o tecnológica en el sector productivo.
- **Autonomía:** Faculta para el análisis y toma de decisiones.
- **Responsabilidad:** Capacidad para asumir compromisos orientados al logro de objetivos y metas laborales.
- **Variedad:** Abarca la ejecución de actividades rutinarias – no rutinarias, predecibles – impredecibles – contextos diversos.
- **Complejidad:** Moviliza recursos cognitivos, procedimentales y actitudinales en diferentes niveles para la ejecución de actividades y funciones.





De igual forma es importante señalar, que, para el caso de la Formación Laboral Básica, como se señaló anteriormente, se logrará en el estudiantado el nivel de competencia dos, el cual se caracteriza por:

- ✓ Realización de actividades programadas
- ✓ Aplicación de habilidades cognitivas y de comunicación para recibir, transmitir y recordar información
- ✓ Utiliza técnicas, materiales, herramientas y equipamiento que no requieren un nivel de especialización para realizar actividades en contextos conocidos, además del uso de tecnologías de la información y comunicación básicas, actuando con ética, con un enfoque de sostenibilidad y responsabilidad sobre su entorno.¹⁰

1. Descubre sistemas de información mediante la codificación y compilación de instrucciones pertinentes utilizando software de programación, de forma creativa y responsable, en diversos ámbitos.	CLBDS1.
2. Predice sistemas de información a través de sus fundamentos, utilizando software de programación que cumpla con los requerimientos de funcionalidad y rendimiento en diferentes contextos, siendo creativo y reflexionando sobre las consecuencias que se deriven de su toma de decisiones.	CLBDS2.
3. Estructura bases de datos relacionadas con situaciones sociales de su comunidad utilizando la programación y mostrando un comportamiento metódico, además de organizado, con la finalidad de promover un pensamiento crítico y analítico.	CLBDS3.
4. Construye sitios web creativos y funcionales mediante software de diseño web, para transmitir información a gran escala de forma responsable y empática en diversos contextos.	CLBDS4.
5. Prueba la funcionalidad de los sitios web, acorde a las necesidades y requerimientos de los usuarios, para el manejo de datos a gran escala, actuando con eficiencia y reflexionando sobre diferentes posturas de conducirse en el contexto.	CLBDS5.

¹⁰ Subsecretaría de Educación Media Superior. (2023b). El currículum laboral en la educación media superior. SEP.





6. Explica la creación de sitios web creativos a través de la utilización de software de aplicación y elementos multimedia para la optimización de páginas web, favoreciendo la solución a diversas situaciones de su entorno, actuando de manera consciente y previniendo riesgos.

CLBDS6.

7. Recomienda soluciones mediante software de aplicación, para aplicaciones móviles y juegos, relacionándose con sus semejantes de forma colaborativa, mostrando disposición al trabajo metódico y organizado en diferentes contextos.

CLBDS7.

8. Selecciona propuestas de robótica básica, a través de sus fundamentos, arquitectura y anatomía, en cualquier ámbito, afrontando retos y asumiendo la frustración como parte del proceso.

CLBDS8.

Así mismo se señalan algunas normas de competencia laboral vigentes, avaladas por el Consejo Nacional de Normalización y Certificación de Competencias Laborales, que podrán servir de guía para el desarrollo de los módulos:

NMX-J-515-ANCE-2021, Equipos De Control Y Distribución-Requisitos Generales De Seguridad.

NOM-003-SCFI-2014 “Productos Eléctricos – especificaciones de seguridad”.





Proceso de Evaluación bajo el Enfoque en Competencias

La evaluación por competencias es un proceso que permitirá mediante la obtención de evidencias conocer el dominio de conocimientos, habilidades y actitudes socioafectivas desarrolladas por el estudiantado.

Dicho proceso deberá ser formativo e integral, es decir, permitirá visualizar no solo el saber, saber hacer y saber ser, sino también el bagaje histórico y cultural lo cual permitirá al estudiantado la comprensión de la realidad social y laboral de los sectores y de la comunidad para a partir de ello lograr su intervención y aporte.

Para ello lo que a continuación se presentan son los principios que orientarán el proceso de evaluación:

1. **Validez:** debe existir correlación entre los resultados de la evaluación y los resultados esperados en situaciones laborales reales.
2. **Confiabilidad:** producir resultados consistentes al evaluar en momentos diferentes y en diversos contextos.
3. **Accesibilidad:** facilitar el acceso a cualquier persona que pueda ser capaz de demostrar el desarrollo de la competencia.
4. **Comunicación:** dar a conocer previamente las condiciones en que se va a evaluar, y posteriormente, los resultados mediante la retroalimentación.
5. **Equidad:** evitar cualquier práctica discriminatoria, es decir, el estudiantado será evaluado bajo los mismos criterios e indicadores.
6. **Flexibilidad:** adaptarse a diferentes modalidades y opciones de formación, así como a las características y necesidades del estudiantado.¹¹

Así pues para poder llevar a cabo lo aquí planteado, se propone la utilización de una amplia gama de instrumentos que permitan visualizar tanto el proceso como el resultado final del aprendizaje

¹¹ Subsecretaría de Educación Media Superior. (2023). El currículum laboral en la educación media superior. SEP.

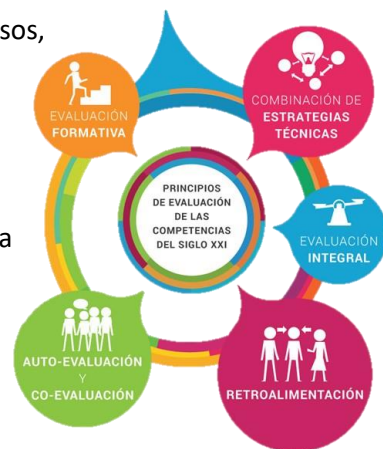




del estudiantado, entre los que se encuentran: rúbricas, pruebas de ejecución, portafolios de evidencias, diario de campo o bitácora, organizadores gráficos, ensayos, resolución de ejercicios y problemas, exámenes o pruebas tipo saber, exposición, método de casos, proyectos y debates o discusiones dirigidas, entre otros.

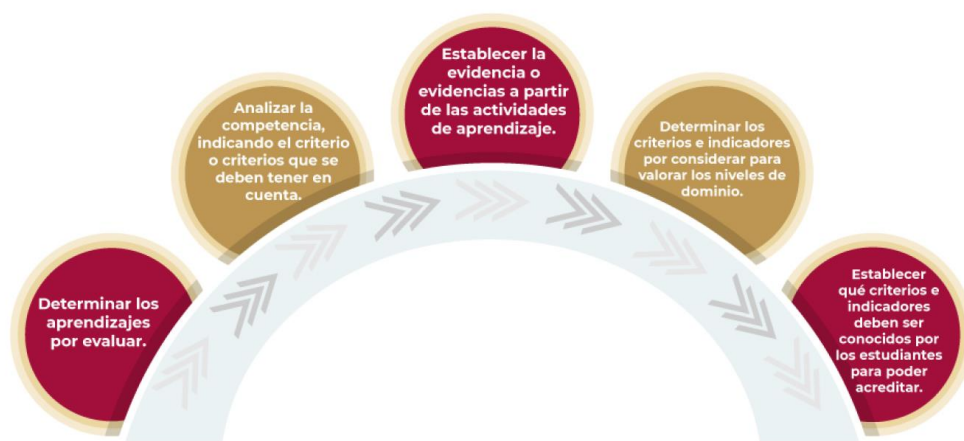
De igual forma, es necesario que la evaluación contemple:

- Autoevaluación: cuando el estudiantado valora su desarrollo y la forma en que aprendió.
- Coevaluación: a través de la retroalimentación entre pares, fomentando la cooperación, la colaboración, la empatía y la crítica constructiva.
- Heteroevaluación: emitida por el personal docente en función de los conocimientos, habilidades, actitudes y valores ponderados en los instrumentos de evaluación.



Finalmente, respecto a los pasos para evaluar las competencias laborales básicas, se presenta a continuación una propuesta elaborada por la Subsecretaría de Educación Media Superior (2023b), la cual ilustra la ruta a seguir y constituye una referencia para el personal docente.

Pasos para evaluar competencias laborales



Fuente: Subsecretaría de Educación Media Superior, 2023b.



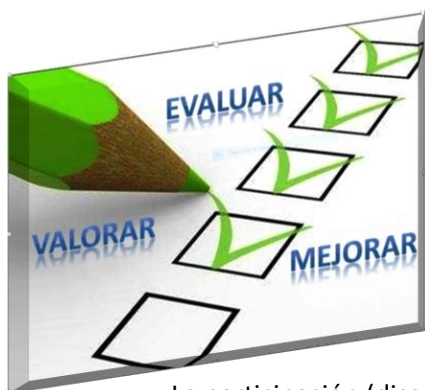


Evaluación por Competencias

Para evaluar por competencias, se debe favorecer el proceso de formación a través de:

- **La Evaluación Diagnóstica:** se realiza antes de algún proceso educativo (curso, secuencia o segmento de enseñanza) para estimar los conocimientos previos del estudiantado, identificar sus capacidades cognitivas con relación al programa de estudios y apoya al personal docente en la toma de decisiones para el trabajo en el aula.
- **La Evaluación Formativa:** se lleva a cabo durante el proceso educativo y permite precisar los avances logrados en el desarrollo de competencias por cada estudiante y advierte las dificultades que encuentra durante el aprendizaje. Tiene por objeto mejorar, corregir o reajustar su avance y se fundamenta, en parte, en la autoevaluación. Implica una reflexión y un diálogo con el estudiantado acerca de los resultados obtenidos y los procesos de aprendizaje y enseñanza que le llevaron a ello; permite estimar la eficacia de las experiencias de aprendizaje para mejorarlas y favorece su autonomía.
- **La Evaluación Sumativa:** se realiza al final de un proceso o ciclo educativo considerando el conjunto de diversas evidencias que surgen de los aprendizajes logrados.

Instrumentos de Evaluación



Con el fin de que el estudiantado muestre el saber hacer que subyace en una competencia, los Sugerencia de Evidencia de Evaluación permiten establecer una estrategia de evaluación, por lo tanto, contienen elementos observables que deben ser considerados en la evaluación tales como:

La participación (discurso y comunicación, compromiso, empeño e iniciativa, cooperación).

- Las actividades generativas (trabajo de campo, proyectos, solución de casos y problemas, composición de textos, arte y dramatizaciones).





- Las actividades de análisis (comprensión e integración de conceptos como interpretación, síntesis y clasificación, toma de decisiones, juicio y evaluación, creación e invención y pensamiento crítico e indagación).

Para ello se consideran instrumentos que pueden agruparse principalmente en (Díaz-Barriga, 2014):

- **Rúbricas:** Son guías que describen las características específicas de lo que se pretende evaluar (productos, tareas, proyectos, exposiciones, entre otras) precisando los niveles de rendimiento que permiten evidenciar los aprendizajes logrados de cada estudiante, valorar su ejecución y facilitar la retroalimentación.
- **Portafolios:** permiten mostrar el crecimiento gradual y los aprendizajes logrados con relación al programa de estudios, centrándose en la calidad o nivel de competencia alcanzado y no en una mera colección al azar de trabajos sin relación. Estos establecen criterios y estándares para elaborar diversos instrumentos para la evaluación del aprendizaje ponderando aspectos cualitativos de lo cuantitativo.

Los trabajos que se pueden integrar en un portafolio y que pueden ser evaluados a través de rúbricas son: ensayos, videos, series de problemas resueltos, trabajos artísticos, trabajos colectivos, comentarios a lecturas realizadas, autorreflexiones, reportes de laboratorio, hojas de trabajo, guiones, entre otros, los cuales deben responder a una lógica de planeación o proyecto.

Con base en lo anterior, los programas de estudio de la Dirección General del Bachillerato al incluir elementos que enriquecen la labor formativa tales como la transversalidad, las habilidades socioemocionales y la interdisciplinariedad trabajadas de manera colegiada y permanentemente en el aula, consideran a la evaluación formativa como eje central al promover una reflexión sobre el progreso del desarrollo de competencias del alumnado. Para ello, es necesario que el personal docente brinde un acompañamiento continuo con el propósito de mejorar, corregir o reajustar el logro del desempeño del bachiller sin esperar la conclusión del semestre para presentar una evaluación final.





Rol del Estudiantado

El rol del estudiantado en el proceso educativo no se limita simplemente a recibir información y repetirla, sino que debe ser un agente activo en la construcción de su propio conocimiento y de su identidad. En este sentido, no sólo se trata de aprender a leer y escribir; implica aprender a narrar y comprender su propia vida, tanto como autor o autora de su historia personal, como testigo de su contexto social y cultural. Este proceso es fundamental para que el estudiantado se convierta en un sujeto consciente y crítico de su realidad.

La educación es un motor de transformación social, pero también puede perpetuar las desigualdades existentes al tratar a todos y todas por igual sin considerar la diversidad inherente al estudiantado. La educación debe empoderarles, dándoles las condiciones necesarias para reconocer y cuestionar las desigualdades que les rodean.

Si las y los estudiantes son insertados en una educación que no considera su clase, sexo, género, etnia, lengua, cultura, capacidad, condición migratoria, religión o cualquier otro aspecto de su identidad, es muy probable que se apropien de la idea de que “la escuela no es para ellos y ellas”, ya que se enfrentarían constantemente a comentarios o actitudes que les califican de incapaces, ignorantes, indolentes o inútiles terminando por creerlo y asumirlo como verdad. Esta autodesvalorización es una barrera significativa para su desarrollo ya que puede llevar a creer que el conocimiento y la sabiduría pertenecen únicamente a las y los “profesionales” y no reconocen el valor de su propio conocimiento y experiencia.

El rol de las y los estudiantes, entonces, debe ser el de un sujeto activo que desafía y transforma estas narrativas opresivas que fomentan las desigualdades. Debe aprender a valorar su propia voz y experiencia, y a reconocer su capacidad para conocer y transformar su realidad. La educación debe ser un proceso liberador que les permita verse a sí mismos o mismas como agentes de transformación social, capaces de escribir su propia historia y de participar activamente en la construcción de una sociedad más justa y humana.



SIMBOLOGÍA

Icono	Descripción
	Lectura: Indica que se realizará la lectura que contiene información de los conocimientos básicos del Programa de Estudio.
	Material audio visual: Representa recursos adicionales al contenido de los conocimientos básicos del Programa de Estudio a través de un video mostrado por el docente, indicado por QR.
	Actividad: Es momento de realizar una actividad teórica – escrita que contribuye a evidenciar el propósito del aprendizaje esperado de los conocimientos básicos del Programa de Estudio.
	Práctica: Hagamos la práctica guiada, que contribuye a la aplicación de los saberes obtenidos con relación al conocimiento básico del Programa de Estudio, se acompaña con el instrumento de evaluación correspondiente.
	Docente explica: Se sugiere que esta sección el docente profundice los conocimientos para adecuarlos a su contexto.
	Actividad SIGA: Indica el producto que se contempla para la plataforma SIGA.



Icono	Descripción
	Evaluación Diagnóstica: Presenta el momento para llevar a cabo la evaluación diagnóstica del submódulo.
	Instrumento de evaluación: Documento en el que tanto el estudiante como el docente observan los criterios con los que se evaluará el producto a realizar.
	Encuadre: Presenta cada aspecto y el porcentaje de calificación con el que se evaluará el submódulo.
	Dosificación: Muestra las fechas en el que los conocimientos básicos del Programa de Estudio se van a desarrollar.
	PAEC: Proyecto Escolar Comunitario
	Cuadernillo: Estudio Independiente

Temario

Submódulo I. Programación Móvil y Juegos

Introducción a la programación móvil y juegos.

- Programación móvil.
- Lenguajes de programación para el diseño de videojuegos.

Software para el desarrollo para aplicaciones móviles.

Diseño y desarrollo de aplicaciones móviles.

Software para el desarrollo de videojuegos.

Diseño y desarrollo de videojuegos en:

- ❖ Navegadores de Internet.
- ❖ Teléfonos celulares.



Submódulo II. Diseño y Programación en Robótica

Introducción a la robótica y la programación de robots.

Robótica.

- ❖ Definición.
- ❖ Tipos.
- ❖ Fases.

Diseño de robot.

- ❖ Mecánica.
- ❖ Eléctrica
- ❖ Electrónica.

Software para programación de robot.

Diseño y desarrollo de robot.



Submódulo I. Programación Móvil y Juegos



Propósito del Módulo

Evaluar soluciones con aplicaciones móviles y juegos, mediante software de aplicación; para solucionar problemas de su entorno, escuchando y respetando diferentes puntos de vista, con el fin de promover el bien común y estructurar propuestas de robótica básica, a través de su arquitectura y anatomía para desarrollarlos en su contexto, afrontando las consecuencias de sus actos.

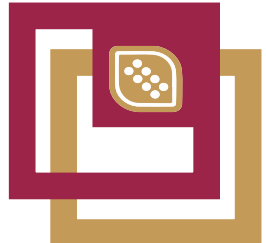
Sugerencias de Evidencia de Evaluación

- ❖ Decide el software comprobando su funcionalidad en Internet y dispositivos móviles, para el desarrollo de aplicaciones móviles, relacionándose con sus semejantes, para trabajar en forma organizada y respetando los diferentes puntos de vista en diferentes ámbitos.
- ❖ Recomienda el software, comprobando su funcionalidad en Internet y teléfonos celulares, para el desarrollo de juegos, siendo innovador y expresando sus ideas, afrontando los resultados de su toma de decisiones para el bien común.

Competencia Laboral Básica

CLBDS7. Reconoce soluciones mediante software de aplicación, para aplicaciones móviles y juegos, relacionándose con sus semejantes de forma colaborativa, mostrando disposición al trabajo metódico y organizado en diferentes contextos.





DOSIFICACIÓN PROGRAMÁTICA

DESARROLLO DE SOFTWARE

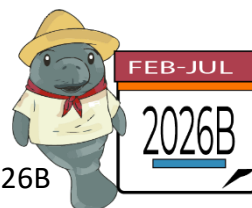
Formación Laboral Básica: Desarrollo de Software

Módulo IV: Aplicaciones Móviles

Submódulo I: Programación móvil y juegos: C6MJ

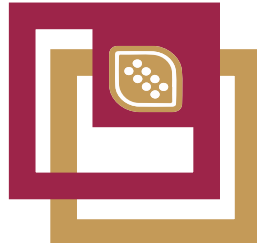
Semestre: 6to.

Turno: Matutino/Vespertino **Periodo:** 2025 – 2026B



Submódulo	Momento	Tiempo (minutos)	Conocimientos	Semana	Fecha inicio	Observaciones
SUBMÓDULO I. PROGRAMACIÓN MÓVIL Y JUEGOS	Apertura	50 min	Bienvenida.	1	03 – 06 de febrero de 2026	Inicio del semestre 03 de febrero
		50 min	Encuadre.			
		100 min	Evaluación diagnóstica.			
		150 min	Introducción a la programación móvil y juegos.			
	Desarrollo	350 min	Software de desarrollo de aplicaciones móviles.	2	09 – 13 de febrero de 2026	
		350 min	Diseño y desarrollo de aplicaciones móviles.			
		100 min	Diseño y desarrollo de aplicaciones móviles.			
		250 min	Instalación de software de desarrollo para videojuegos.			





Submódulo	Momento	Tiempo (minutos)	Conocimientos	Semana	Fecha inicio	Observaciones
SUBMÓDULO I. PROGRAMACIÓN MÓVIL Y JUEGOS	Desarrollo	350 min	Diseño y desarrollo de videojuegos.	5	02 – 06 de marzo de 2026	
		350 min	Funcionalidad de aplicaciones móviles y juegos en: • Navegadores de Internet. • Teléfonos celulares	6	09 – 13 de marzo de 2026	
	Cierre	350 min	Funcionalidad de aplicaciones móviles y juegos en: • Navegadores de Internet. • Teléfonos celulares. Evaluación Situación Didáctica	7	17 – 20 de marzo de 2026	 Suspensión oficial 16 de marzo.



Criterios de Evaluación

Didáctica:

Innovando con Creatividad



30%

Actividades:

- Actividad 1. Gamificación de programación.
- Actividad 2. Identifica tu entorno desarrollador.
- Actividad 3. Fases de una app.
- Actividad 4. Timeline evolución de los videojuegos.
- Actividad 5. Creación de un videojuego.
- Actividad 6. Diferencias y similitudes.

5%
5%
5%
5%
5%
5%

30%

¡Un paso maj
campeones!

Prácticas:

- Práctica 1. Proponiendo mi App.
- Práctica 2. Vamo a juga.

10%
10%

20%

Examen:

20%





Situación Didáctica

INNOVANDO CON CREATIVIDAD

Propósito:	Desarrollar una aplicación móvil con temática sostenible en equipos de 5 integrantes donde presenta un catálogo de diversas plantas para el cuidado, uso, protección de éstas o hábitos, actividades de prevención, reducción, reciclado y reutilización de los desechos en el hogar utilizando el software de programación propuesto, para socializarlo en plenaria.
Contexto:	Las afectaciones climáticas en nuestro planeta han ido incrementando, sufriendo los estragos como las inundaciones y altas temperaturas en nuestro estado. Una situación alarmante que se ha percatado son los desechos electrónicos, ocasionando un mayor índice de contaminación. Por ello el Colegio de Bachilleres de Tabasco, toma su participación en la Agenda 2030 al realizar acciones por el clima, solicitando a los estudiantes de 6to semestre de la capacitación de desarrollo de software que elaboren juegos con mensajes ambientales para que las generaciones de 12 a 19 años tengan información de los riesgos que implica el no tener una cultura de reciclaje o los beneficios generados a la sociedad y el ambiente al crear hábitos de prevención, reducción, reciclado, reutilización, así como un catálogo de diversas plantas para el cuidado, uso y protección de las mismas.
Conflicto cognitivo	Menciona el software que consideras en la programación móvil. ¿Cuáles son las fases de desarrollo para aplicaciones móviles? Menciona el software que consideren en el diseño y desarrollo de juegos. ¿Los juegos solo sirven para divertirse? ¿Cuál es la utilidad de crear aplicaciones móviles?





Evaluación Diagnóstica



Instrucciones: Lee cuidadosamente los siguientes cuestionamientos y selecciona en base a tus conocimientos la respuesta que consideres correcta.

- ¿Es el arte de conjugar la lógica con la creatividad que es proceso por el cual se limpia, codifica, traza y protege el código fuente de programas computacionales, en otras palabras, es indicarle a la computadora lo que tiene que hacer?

A) Programación	C) Informática
B) Internet	D) Diseño Digital
- ¿Se puede definir como un aparato de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales?

A) Control Remonto	C) Consolas de videojuegos
B) Dispositivo Móvil	D) Monitores
- Consiste en desarrollar aplicaciones para aparatos como celulares, smartphones, PDA's, PocketPC's y dispositivos con recursos limitados.

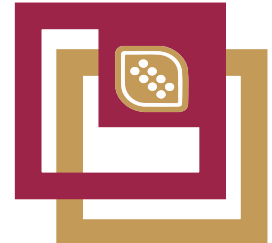
A) Programación Web	C) Programación Móvil
B) Programación de BD	D) Programación Aplicaciones Web
- ¿Es una actividad multidisciplinaria, que involucra profesionales de la programación, diseño gráfico, animación, sonido, música y actuación?

A) Programación Web	C) Programación Móvil
B) Programación de BD	D) Programación de Videojuegos
- Es una de las 3 categorías de los desarrolladores de videojuegos.

A) Desarrolladores de BD	C) Desarrolladores Externos
B) Desarrolladores Web	D) Desarrolladores Internos
- ¿Este lenguaje de programación es uno de los más utilizados en el sector por profesionales. Es un lenguaje popular en los títulos AAA, se utiliza en videojuegos para PlayStation y Xbox, y en juegos independientes?

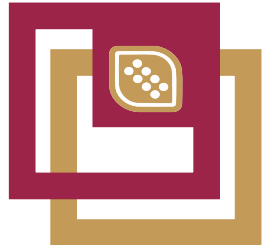
A) Objective-C	C) C++
B) Python	D) Lua





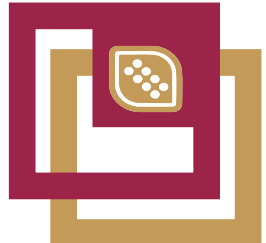
7. Es uno de los lenguajes más utilizados en el desarrollo de **videojuegos web** y de navegador.
A) Lua
B) JavaScript
C) Objective-C
D) Scratch
8. Es un lenguaje visual de programación centrado en fomentar la creatividad y el pensamiento lógico.
A) Scratch
B) Objective-C
C) JavaScript
D) Java
9. ¿Qué significa PWA?
A) Aplicación Web Progresiva
B) Programación Web Analítica
C) Programación de Base de Datos
D) Aplicación Móvil y Juegos
10. Implica encastrar piezas pre diseñadas (como en un rompecabezas) de forma tal que se genere una lista de pasos o acciones a seguir para solucionar un problema planteado o creado por el mismo usuario..
A) Programación Web
B) Programación por Bloques
C) Programación Funcional
D) Programación de Base de Datos
11. Es una Aplicación Web Progresiva.
A) Eclipse
B) Visual Studio Code
C) Mit App Inventor
D) Laravel
12. Es la versión inicial de la idea de un producto o servicio, nos permite probar, evaluar y validar si la idea que tenemos en mente cumple los objetivos en el diseño de una app móvil.
A) Administrador de archivos
B) Programa de aplicación
C) Wireframing
D) Prototipado
13. Es la parte de desarrollo y escritura de código de una app, lo que no se ve. Para darle forma, y que realice en un primer momento una o varias de las acciones requeridas, es necesario saber programar.
A) Prototipado
B) Wireframing
C) Frontend
D) Backend
14. Es lo que se ve en diseño de aplicaciones, es decir, lo que se ve de la aplicación: sus pantallas.
A) Prototipado
B) Wireframing
C) Frontend
D) Backend





15. El resultado de esta etapa es una idea de aplicación, que tiene en cuenta las necesidades y problemas de los usuarios. La idea responde a una investigación preliminar y a la posterior comprobación de la viabilidad del concepto.
A) Desarrollo
B) Conceptualización
C) Publicación
D) Diseño
16. Este proceso se realiza normalmente sobre redes TCP/IP. Esto nos permite jugar contra otros jugadores situados en nuestra propia red local o en Internet.
A) Comunicación
B) Sonido
C) Visualización
D) Entrada
17. En esta etapa es necesario definir los aspectos fundamentales que conformarán el videojuego.
A) Concepción
B) Diseño
C) Planeación
D) Producción
18. En esta fase se detallan todos los elementos que compondrán el juego, dando una idea clara a todos los miembros del grupo desarrollador acerca de como son.
A) Concepción
B) Diseño
C) Planificación
D) Producción
19. Aquí se llevan a cabo todas las tareas especificadas en la fase de planificación, teniendo como guía fundamental el documento de diseño. Esto incluye entre otras cosas la codificación del programa, la creación de sprites, tiles y modelos 3D, la grabación de sonidos, voces y música, la creación de herramientas para acelerar el proceso de desarrollo, entre otras.
A) Concepción
B) Diseño
C) Producción
D) Planeación
20. Con este software se pueden crear historias interactivas, juegos, animaciones, música, videojuegos y producciones artísticas.
A) Laragon
B) Dev C++
C) Mit App Inventor
D) Scratch





Lectura No 1. Introducción a la Programación Móvil y Juegos

Instrucciones: Realiza la Lectura 1. “Introducción a la Programación Móvil” subrayando las ideas principales, y al finalizar realiza la Actividad 1 – “Gamificación de Programación”



LECTURA 1. “INTRODUCCIÓN A LA PROGRAMACIÓN MÓVIL Y JUEGOS”



PROGRAMACIÓN



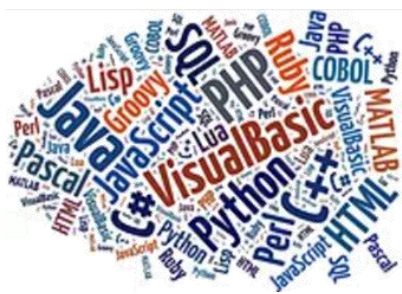
En el mundo de la computación, la programación es el arte de conjugar la lógica con la creatividad que es proceso por el cual se limpia, codifica, traza y protege el código fuente de programas computacionales, en otras palabras, es indicarle a la computadora lo que tiene que hacer.

Detrás de todos los programas informáticos que conocemos y usamos de manera cotidiana para facilitarnos diversas actividades de nuestro día con día, existe todo un proceso para poderlos crear. Este proceso es conocido como programación, conozcamos un poco más sobre lo que conlleva este proceso.



Por medio de la programación se establecen los pasos a seguir para la creación del código fuente de los diversos programas informáticos.

Este código le indicara al programa informático que tiene que hacer y cómo realizarlo.



Medita en esto

DSW_SUB1_V1

<https://youtu.be/MEuZICEDLJo>



PROGRAMACIÓN MÓVIL

Permite poder simplificar muchas actividades de la vida cotidiana, agilizando la comunicación y el acceso a información. Con el auge del internet y las aplicaciones sobre la web son elementos que han influido en el desarrollo de la programación móvil.



La evolución de los dispositivos móviles es uno de los aspectos en los cuales ha influido el avance tecnológico de aplicaciones en la web, dando con ello una evolución en los diversos enfoques de funcionalidad de los dispositivos móviles, los cuales han permitido poder evolucionar en todos nuestros ámbitos, como son: productividad laboral, personal, económico, entre otros.

Las personas a través de dispositivos móviles como tabletas, celulares, etc., logran resolver tareas cotidianas de manera más rápida y eficiente; es por lo que cada vez más necesario implementar aplicaciones que puedan ser utilizadas a través de los mismos como **programas transaccionales**, **programas educativos** como **lectura** de libros en dispositivos móviles, uso de diccionarios, ubicación geográfica de lugares de interés, videojuegos, entre otros.



Un dispositivo móvil se puede definir como un aparato de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales. De acuerdo con esta definición existen multitud de dispositivos móviles, desde los reproductores de audio portátiles hasta los navegadores GPS, pasando por los teléfonos móviles, los PDAs o los Tablet PCs.



¿Qué es la programación móvil?

Programación Móvil es desarrollar aplicaciones para aparatos como celulares, smartphones, PDA's, PocketPC's y dispositivos con recursos limitados. Hay que ser claros con la diferencia entre Programación Móvil y Programación Embebida, que son cosas parecidas (recursos limitados) pero distintas en esencia. Sistemas Operativos para móviles están Symbian, Palm OS y Windows Mobile; iOS de Apple, Blackberry OS de RIM, Android de Google, entre otras.

El **desarrollo móvil** es la actividad encaminada a la creación de aplicaciones o programas para dispositivos como los Smartphones y las Tablets, entre otros. Esta actividad es llevada a cabo por programadores y diseñadores, quienes, valiéndose de las herramientas necesarias, como lenguajes de programación, APIs y SDKs, realizan aplicaciones para una plataforma móvil o para múltiples de ellas. Existen varias plataformas móviles; Bada de Samsung, Symbian principalmente de Nokia, Windows Phone de Microsoft, iOS de Apple, Blackberry OS de RIM, Android de Google, entre otras.



El **OS Symbian** es una plataforma desarrollada originalmente por un conjunto de empresas como Samsung, Ericsson, Nokia, Siemens, etc. Actualmente, es mantenida principalmente por Nokia. Utiliza una IDE conocido como QT y un SDK para el lenguaje de programación C++, el cual es el lenguaje nativo de esta plataforma.

Windows Phone es una plataforma desarrollada por Microsoft y es una versión mejorada del antiguo SO Windows Mobile, utilizado en terminales y dispositivos como PDAs entre otros. Este sistema operativo móvil ofrece como herramientas para su desarrollo a la plataforma de desarrollo .NET, sus lenguajes principales como C# y Visual BASIC., y el entorno Visual Studio.



La plataforma **iOS** es desarrollada por Apple exclusivamente para sus dispositivos móviles, como lo son el iPhone, iPod, iPad y Apple TV. Goza de una gran popularidad y es rival directo de Android, otra plataforma



bastante popular. El desarrollo para esta plataforma se lleva a cabo de manera nativa mediante el IDE Xcode, el cual incluye todas las herramientas necesarias, como el SDK y emuladores, para los dispositivos Apple. Utiliza el lenguaje de programación Objective C, es cual es una adaptación del clásico lenguaje C. Como desventaja, se puede mencionar que el desarrollo nativo solo puede realizarse bajo el Sistema Operativo OSX, lo cual supone una gran limitación pues es también necesario contar con una computadora Mac.

La plataforma **Blackberry** es desarrollada por la empresa canadiense RIM (Research In Motion). Es una plataforma sumamente popular entre los usuarios de dispositivos móviles, y sus dispositivos han sido reconocidos desde su origen por contar con un teclado QWERTY y estar enfocados al uso de oficina. En cuanto al desarrollo para esta plataforma, se ofrecen una variedad de herramientas y lenguajes. Es posible desarrollar, como en casi todas las plataformas, haciendo uso de tecnologías web; se puede usar Java de manera nativa, o adaptar aplicaciones para Android de tal forma que se puedan ejecutar en el Blackberry OS.

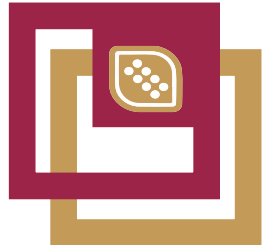


PROGRAMACIÓN DE JUEGOS

En la programación de aplicaciones de videojuegos, debemos considerar que los procesos involucran desde la creación del concepto inicial hasta el videojuego en su versión final. Es una actividad multidisciplinaria, que involucra profesionales de la programación, diseño gráfico, animación, sonido, música y actuación.

Se tiene registro que los primeros videojuegos, que involucran estos procesos, surgieron en la década de 1960, sin embargo, requerían grandes aparatos tecnológicos, siendo poco accesibles al público. Para la década de 1970, el crecimiento comercial de las consolas domésticas potenció esta tecnología.





Debido a las limitadas capacidades y altos costos de las computadoras de aquella época, un programador se encargaba del desarrollo de todo el videojuego; hoy, gracias a la tecnología actual y las altas expectativas del consumidor, hacen que sea muy difícil que un único programador produzca un videojuego, es por eso que se invierten fuertes sumas de dinero en el proyecto además de un personal de 2 hasta equipos enteros divididos que se encargan de un acción en concreto, como el diseño de personajes, sonido, ambientación, entre otros.



Hoy en día existen guías o patrones que puede seguir el programador y desarrollar el videojuego hasta su etapa final, generalmente hace es el siguiente proceso:

- ↳ Concepción de la idea del videojuego
- ↳ Diseño
- ↳ Planificación
- ↳ Pre Producción
- ↳ Producción
- ↳ Pruebas
- ↳ Mantenimiento

🔍

Considéralo

DSW_SUB1_V2

<https://www.youtube.com/watch?v=DFau2pCHrnU>



Los desarrolladores de videojuegos se clasifican en 3 categorías: desarrolladores de terceros (desarrolladores de software libre), desarrolladores internos (forman parte de una empresa que es dueña de una consola) y los desarrolladores independientes (quienes hacen sus juegos por su cuenta y lo lanzan de la misma manera).

La industria del entretenimiento demanda especialistas en estas disciplinas para satisfacer las expectativas de los gamers del mundo. Los programadores se encargan de hacer posibles las interacciones del personaje con su entorno a partir de las órdenes del jugador y traducir estas órdenes en acciones que se muestren en la pantalla. Para ello, primero el programador prepara unos prototipos para ver cómo se va ejecutar un juego determinado.



Lenguajes de programación para el diseño de videojuegos

Existen miles de lenguajes de programación. Para elegir uno, tenemos que saber en qué plataforma se jugará el videojuego, por ejemplo, si será para PlayStation o para IOS o para XBOX. También hay que tener en cuenta qué software vamos a utilizar para programarlo. Entre los lenguajes de programación más utilizados está el llamado lenguaje C, para juegos más sencillos; o versiones algo más avanzadas como el C# (C Sharp) o el C++ (C Plus Plus).



Para ser programador de videojuegos hay que tener conocimientos de programación, pero también es muy importante que el programador tenga

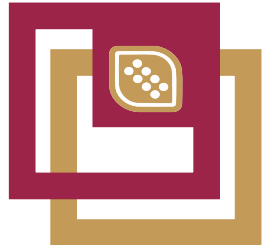
“conocimientos en todas las ramas del desarrollo de videojuegos, aunque no sean especialistas en el tema”, aconseja Sergio Martínez de CEV Barcelona.

El lenguaje con el que están programados los videojuegos es un elemento fundamental para garantizar una experiencia de juego fluida. Parece que cuando jugamos a un videojuego solo nos fijamos en la parte visual, en su estética, porque es lo más espectacular, pero detrás de esa estética hay un arduo trabajo de programación que permite su funcionamiento.

¿Qué lenguaje de programación usar?

Dependiendo del tipo de videojuego que quieras desarrollar, existen diferentes lenguajes de programación. En función del grado de interactividad y el dispositivo que se vaya a utilizar para jugar y la rapidez de respuesta que queramos conseguir, deberemos elegir uno u otro. Además, podemos emplear varios lenguajes si el grado de complejidad del diseño es alto, para que todas las capas del juego funcionen correctamente. Te explicamos algunos de los lenguajes de programación de videojuegos más usados, para diferentes niveles y usos.





A continuación, algunos de los mejores lenguajes de programación para que desarrolles videojuegos interesantes y de la máxima calidad.

C++ Este lenguaje de programación es uno de los más utilizados en el sector por profesionales. Es un lenguaje popular en los títulos AAA, se utiliza en videojuegos para PlayStation y Xbox, y en juegos independientes. Se trata del lenguaje más compatible con la mayoría de los motores de juego y tiene un tiempo de ejecución bastante rápido. Por otro lado, permite a los desarrolladores tener un control amplio sobre el hardware, la gestión de la memoria y los gráficos, y, aunque al principio puede resultar complejo de utilizar, una vez te haces a él, podrás manejar cualquier otro lenguaje.



C# es un lenguaje de programación muy popular, sobre todo en entornos Windows. Es un poco menos flexible y compatible que C++, pero algunos motores como Unity permiten programar con él y no está limitado a un determinado sistema operativo o plataforma; se pueden crear juegos para iOS, Android, Windows Play Station y Xbox. Es un lenguaje más fácil de aprender que el C++ y será una buena opción si estás empezando o si quieres que lo disfruten más personas, con independencia del dispositivo.



Java se trata de un lenguaje frecuentemente utilizado y presenta muchas similitudes con C++. Su principal característica es la **versatilidad**, ya que se puede utilizar en todas las plataformas, dispone de gran cantidad de frameworks para el desarrollo 3D, ofrece **módulos de código abierto** y su modelo se puede actualizar constantemente. ¿El problema? Que se ejecuta dentro de su máquina virtual, y esto supone una pérdida de rendimiento.



JavaScript es uno de los lenguajes más utilizados en el desarrollo de **videojuegos web** y de navegador. La mayoría de los motores de videojuegos son compatibles con JavaScript, y cuenta con múltiples **frameworks** para 3D y una gran **variedad de bibliotecas**. Además, algunos motores de videojuegos como Unity lo utilizan, por lo que podremos usarlo para crear todo tipo de scripts dentro del juego.





Python a pesar de no ser un lenguaje de programación exclusivo para la creación de videojuegos, es un lenguaje muy flexible y potente para esto. Su ejecución es mucho más simple que la de otros lenguajes (permite plasmar ideas complejas con pocas líneas de código), y su framework Pygame permite a los desarrolladores crear prototipos de sus videojuegos de manera rápida y sencilla, y funciona prácticamente en todas las plataformas y sistemas operativos.



Lua finalmente, es un lenguaje de programación sencillo, rápido y fácil de aprender. Compatible con lenguajes más complejos y de rápida ejecución, también se usa para aplicaciones web y procesamiento de imágenes. Este lenguaje es especialmente útil para proyectos independientes y programadores que estén empezando en la profesión.



Scratch es un lenguaje visual de programación centrado en fomentar la creatividad y el pensamiento lógico. Aunque cualquier edad es buena para aprenderlo, los niños/as son destinatarios ideales para Scratch. Enseñándoles a usar Scratch estamos ayudándoles a afrontar y resolver situaciones y problemas de todo tipo de una manera lógica y estructurada.

JS

☕

?

Python

C#

🔍

Medita en esto

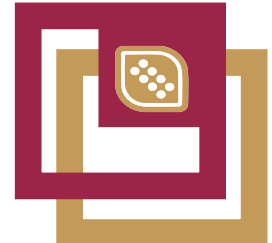
DSW_SUB1_V3

**EL RINCON
DEL INGE**
By Alex

<https://youtu.be/udrcro45Jo4>

PRESENTACIÓN ELECTRÓNICA

Apertura _ Introducción a la programación móvil y juegos



BIBLIOGRAFÍA

Alvarado, I. (2020, 11 abril). *Principales características de los tipos de aplicaciones móviles*. Cero Ideas. <https://ceroideas.es/principales-caracteristicas-de-los-tipos-de-aplicaciones-moviles/>

Capdevila, A. G. (2021, 27 diciembre). *Los 6 mejores lenguajes de programación para videojuegos*. Profile Software Services. <https://profile.es/blog/lenguajes-programacion-videojuegos/>

https://dmd.unadmexico.mx/contenidos/DCEIT/BLOQUE2/DS/08/DPMO/U0/descargables/DPMO_Informacion_general.pdf

<http://www.itcg.edu.mx/pdf/ISIC/7/Programacion%20Movil.pdf>

Programación Móvil. (2014, 29 noviembre). Blogspot.com. <http://programacionmovilufps.blogspot.com/2014/11/que-es-la-programacion-la-programacion.html>



¿Qué es la programación y el desarrollo de videojuegos? – Instituto Artek. (2021, 4 febrero). ¿Qué es la programación y el desarrollo de videojuegos? https://artek.edu.mx/noticia_eventos/que-es-la-programacion-y-el-desarrollo-de-videojuegos/



ACTIVIDAD 1. GAMIFICACIÓN DE PROGRAMACIÓN

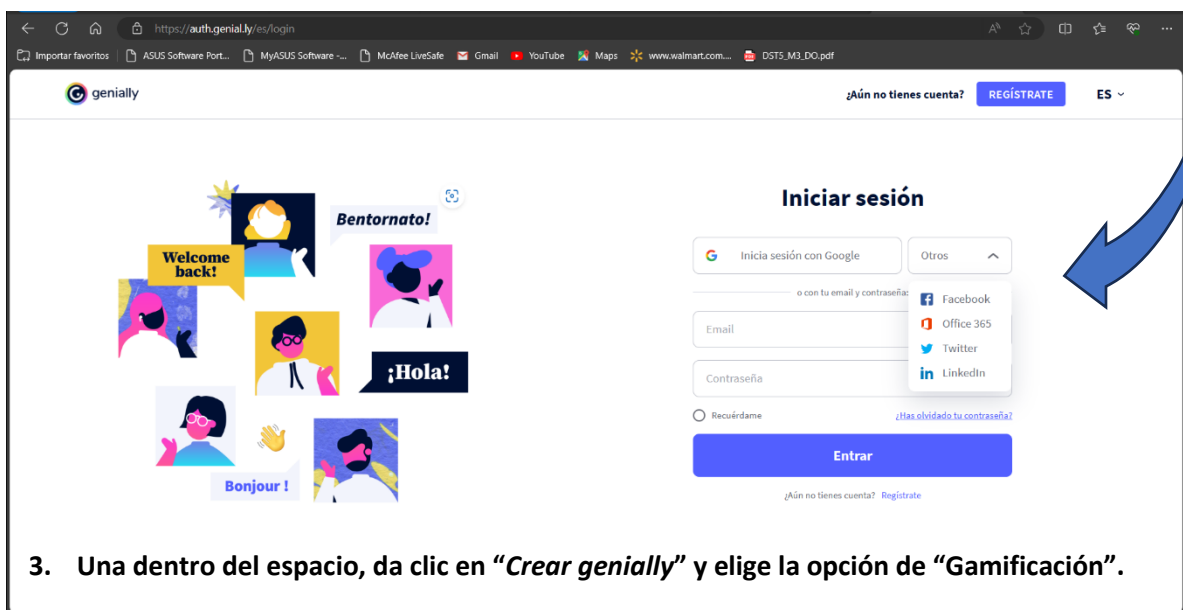
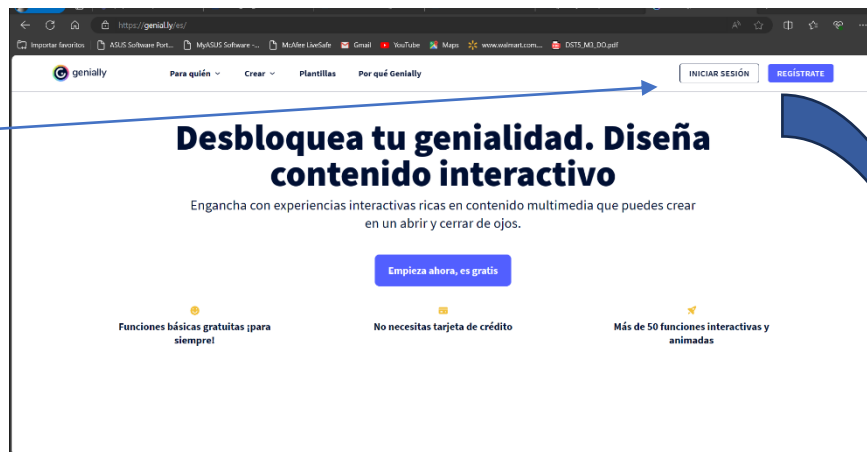


Instrucciones: Después de realizar la Lectura No 1. Construye un cuestionario a través de Genially de 10 preguntas de acuerdo con lo estudiado en la lectura y compártelo con tus compañeros, o responde las preguntas en el espacio en línea, al finalizar comenta tus respuestas con tus compañeros de clase.



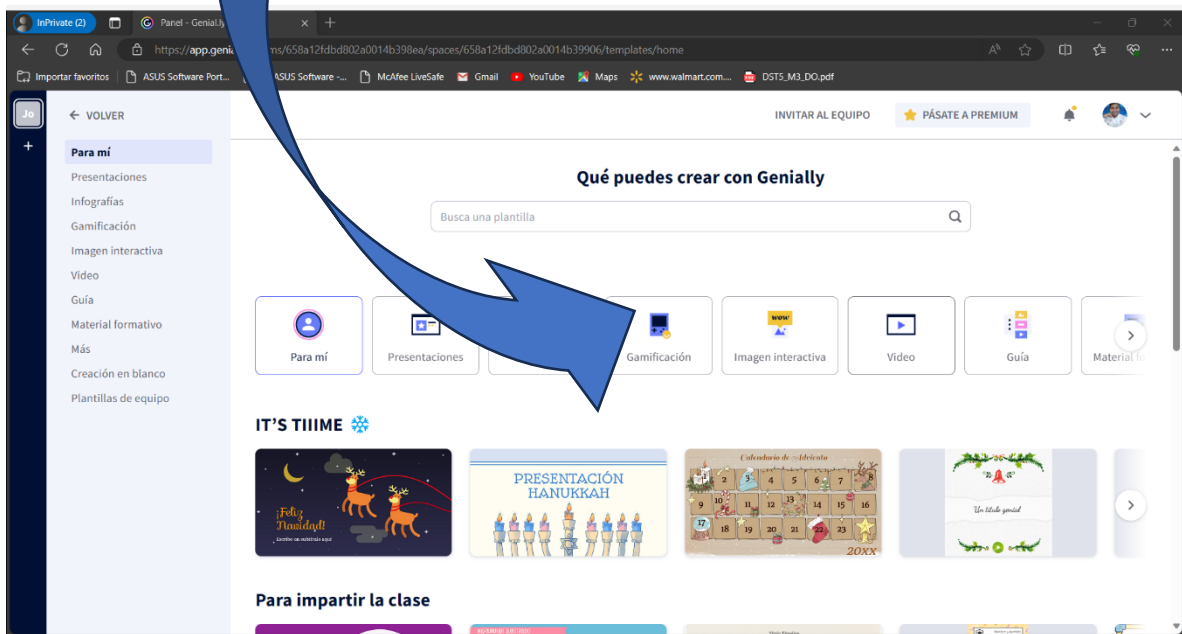
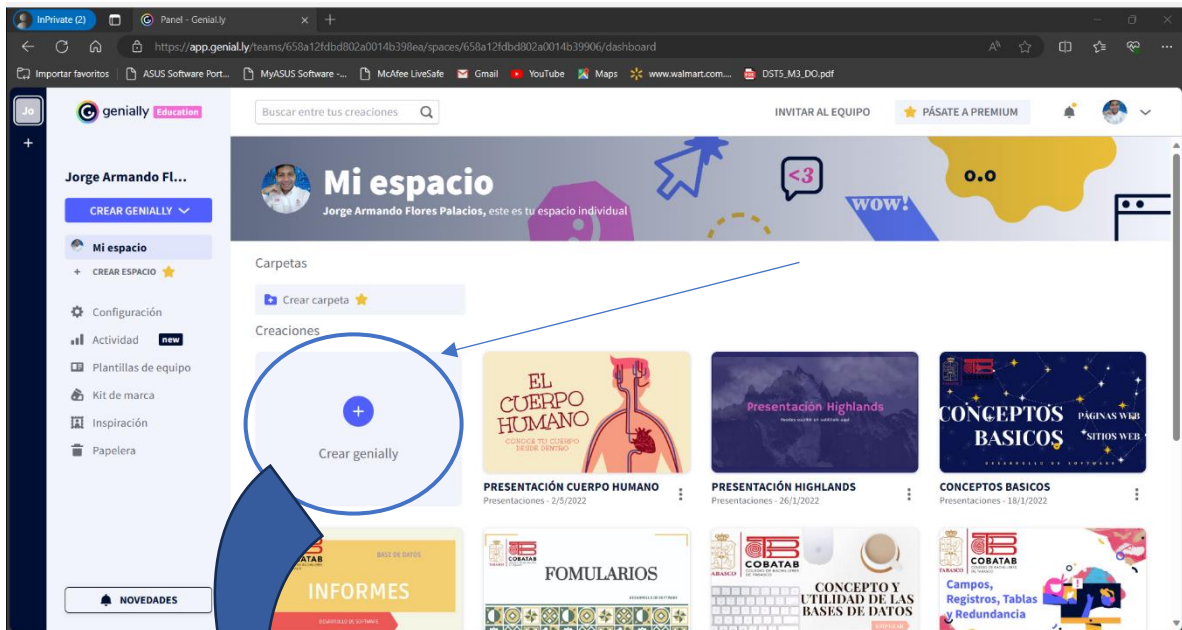
1. Ingresa al espacio de Genially: <https://auth.genially.ly/es/>

2. Inicia sesión en Genially con tu cuenta institucional (Microsoft) o con tu cuenta personal (Gmail, Microsoft).

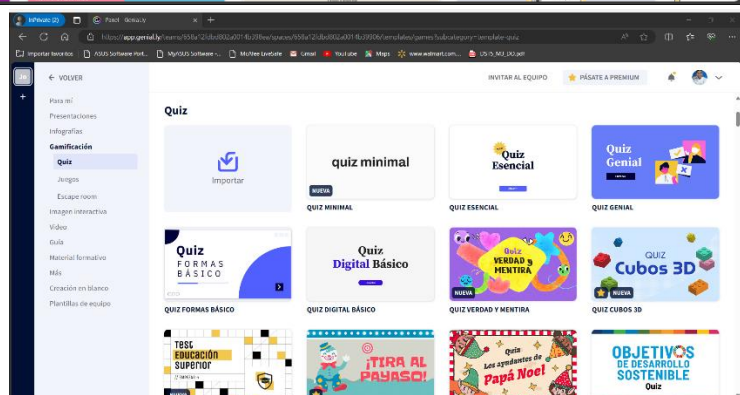


3. Una dentro del espacio, da clic en “Crear genially” y elige la opción de “Gamificación”.





- 4. Elige una de las plantillas y construye el quiz en su forma de juego:**



5. Recuerda filtrar las plantillas gratuitas de las plantillas Premium.
6. Considera 5 preguntas relacionadas a la programación móvil con sus respectivas respuestas.
7. Considera 5 preguntas relacionadas a la programación de juegos con sus respectivas respuestas.
8. Al terminarlo, genera el enlace para compartirlo con tus compañeros.
9. No olvides colocar tus datos.



🔍

Medita en esto
DSW_SUB1_V3

genially

Cómo crear una gamificación

https://youtu.be/dKqeaEsXwB8?si=HhYu3-Ghone_rtqt

Ejemplo de gamificación:

https://es.educaplay.com/recursos-educativos/11183117-introduccion_a_la_programacion_movil.html





INSTRUMENTO DE EVALUACIÓN
LISTA DE COTEJO 1
Actividad 1: "Gamificación de Programación"



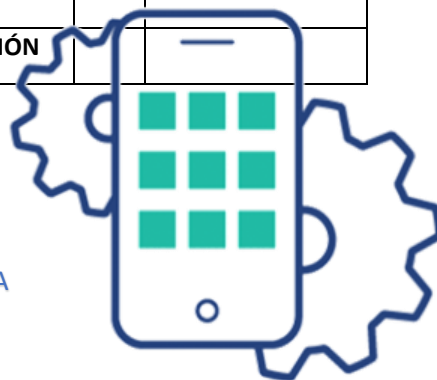
DATOS GENERALES

Nombre(s) del alumno(s)			Matrícula(s)			
Producto: Quiz gamificado			Fecha			
Submódulo 1: Programación Móvil y Juegos			Periodo: 2025 – 2026B			
Nombre del docente			Firma del docente			
CRITERIO	INDICADORES	VALOR OBTENIDO		PONDERACIÓN	CALIF	OBSERVACIONES Y/O SUGERENCIAS DE MEJORA
		SÍ	NO			
1	Las preguntas están redactadas de forma clara y entendible.			1		
2	Presenta 5 reactivos relacionados a la programación móvil.			2		
3	Presenta 5 reactivos relacionados a la programación de juegos.			2		
4	Emplea un diseño atractivo, utilizando correctamente las herramientas del software.			1		
5	No presenta faltas de ortografía.			1		
6	El quiz indica cuando una pregunta respondida es correcta o incorrecta.			2		
7	Entrega el enlace de la gamificación en tiempo y forma.			1		
CALIFICACIÓN						

PRESENTACIÓN ELECTRÓNICA



M4_SUB1_NATIVO_PWA



Lectura No 2. Software de Desarrollo de Aplicaciones Móviles

Instrucciones: Realiza la Lectura 2. “Software de Desarrollo de Aplicaciones Móviles” subrayando las ideas principales, prepárate para instalar e inspeccionar el software de desarrollo y al finalizar realiza la Actividad 2 – “Identifica tu entorno desarrollador”

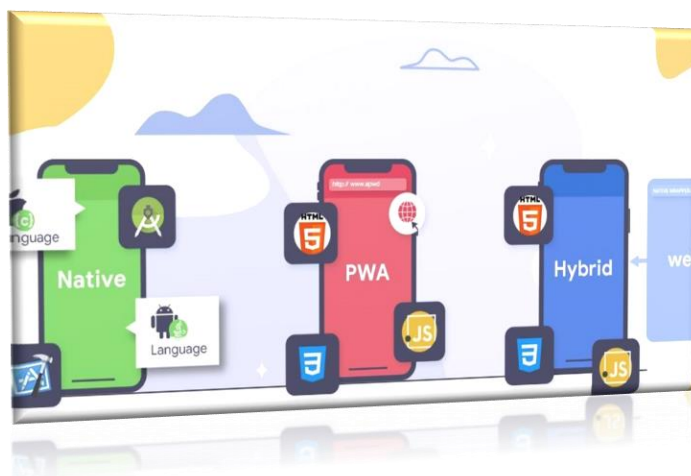


LECTURA 2. “SOFTWARE DE DESARROLLO DE APLICACIONES MÓVILES”



Conocer los lenguajes de programación para el desarrollo de aplicaciones móviles es primordial para un programador novato. Aunque actualmente existen diferentes paradigmas de programación que permiten desarrollar el software de manera más específica y simple; no perdemos la base de la programación que es **la lógica**, además de comprender la sintaxis, la funcionalidad, el diseño, menciona González (2021).

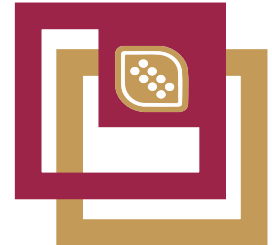
Uno de los mercados más rentables para los programadores son las tiendas de apps para móviles. Ya sea a la hora de buscar trabajo en otras empresas, o para convertirnos en programadores freelance, conocer los lenguajes de programación para móviles nos abrirá muchas puertas y, además, nos brindará una gran oportunidad de conseguir dinero en un mercado tan grande como es el de **Android** o **iOS**.



Hay muchas formas de **programar aplicaciones móviles**. Podemos utilizar los lenguajes de programación nativos para crear todo tipo de apps, recurrir a los lenguajes web (ya que las PWA – Progressive Web App están cada vez más de moda), usar programas que nos permitan crear apps sin tener que saber de

programación, e incluso podemos usar motores 3D (como Unity) para crear juegos. Todo depende del tipo de app que vayamos a crear y para quién estará dirigida.





Con todo esto, es preciso concluir que cualquier usuario con conocimientos básicos/medios en informática o cómputo puede adentrarse en el mundo de la **programación**. En gran medida esto se logra gracias a los entornos de desarrollo de lo que podemos echar mano en estos momentos. Estos intentan facilitarnos las cosas en la medida de lo posible a la hora de escribir código, ya sea para dispositivos móviles o para aplicaciones más de equipos de sobremesa.

¿Qué necesito para crear aplicaciones de iOS y Android?

A continuación, veamos los entornos y lenguajes más utilizados para desarrollar aplicaciones móviles. Claro está que, para lograr objetivos más avanzados o profesionales, tendremos que ir ganando experiencia, algo que no se logra más que con el **tiempo** y el **esfuerzo**.

Al contrario de lo que pueda parecer, cualquiera puede crear sus propios programas sin problemas para estas plataformas. No es necesario tener un hardware especial para crear aplicaciones, y tampoco pagas costosas licencias. Si vamos a usar **IDEs** de programación gratuitos para crear los programas, lo único que tendremos que pagar son las cuotas de las tiendas de aplicaciones. En Google es un pago único de 20-25 euros, mientras que en Apple son 100 euros al año, vendas o no.



Podemos crear apps para Android desde cualquier PC, desde cualquier sistema operativo. El propio IDE de Google (Android Studio) es capaz de compilar los programas independientemente de qué sistema usemos. Además, no necesitamos un móvil físico ya que podemos emular Android desde el IDE y cargar los programas. En el caso de iOS, la cosa cambia. Apple solo permite crear y compilar programas desde su sistema operativo. Por tanto, necesitaremos tener un Mac para poder programar para iOS. Y, además, un iPhone o iPad para probar los programas que creemos.



Lenguajes de Programación para Android



Android es el sistema operativo para móviles más utilizado que podemos encontrar. Este sistema operativo se basa en la máquina virtual de Java, y, por lo tanto, este es el lenguaje de programación principal para crear apps de todo tipo.

Java es uno de los lenguajes de programación más populares. Al ser un lenguaje multiplataforma, es muy sencillo crear aplicaciones para una determinada plataforma y, con unos pocos cambios (o con ninguno), hacerlo funcionar en otros sistemas operativos. El problema de este lenguaje es que funciona a través de la máquina virtual de Java, lo que a menudo se traduce en problemas de rendimiento.

Además de Java, Android tiene otros lenguajes de programación muy populares. **Kotlin**, por ejemplo, es un lenguaje de programación inspirado en Java creado por Google para incitar a los usuarios a crear apps escritas en este lenguaje. Las ventajas de este es que es más sencillo que Java y ofrece un mejor rendimiento. Por desgracia, no es tan popular como esperaba Google.



Por supuesto, también podemos usar los lenguajes de programación web para crear todo tipo de aplicaciones para móviles. Lenguajes como **HTML5** o **CSS3** nos ayudarán a dar forma a nuestras aplicaciones. Estos lenguajes son los utilizados para crear las PWA, apps que están ganando una gran popularidad últimamente y que tendrán gran importancia en el futuro.



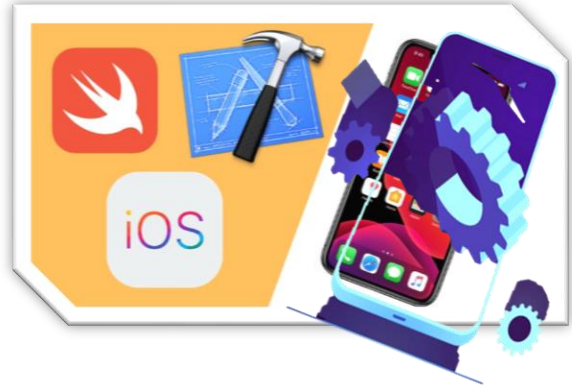
Sin duda, el IDE de programación para Android más popular que podemos encontrar es **Android Studio**. Este es un IDE creado a partir de Eclipse (uno de los IDE para Java más populares) con el cual podremos sacarle el máximo partido a nuestro código. Contiene ayudas de sintaxis, un creador de interfaz e incluso un emulador de Android para probar nuestras apps y depurarlas antes de publicarlas.



Lenguajes de

Programación para iOS

Si lo que queremos es programar apps para iOS, el sistema operativo de Apple para los iPhone, la cosa cambia un poco. El lenguaje de programación más utilizado para crear apps para este sistema operativo es **Objective-C**, creado como un superconjunto de C al que se le han añadido objetos.



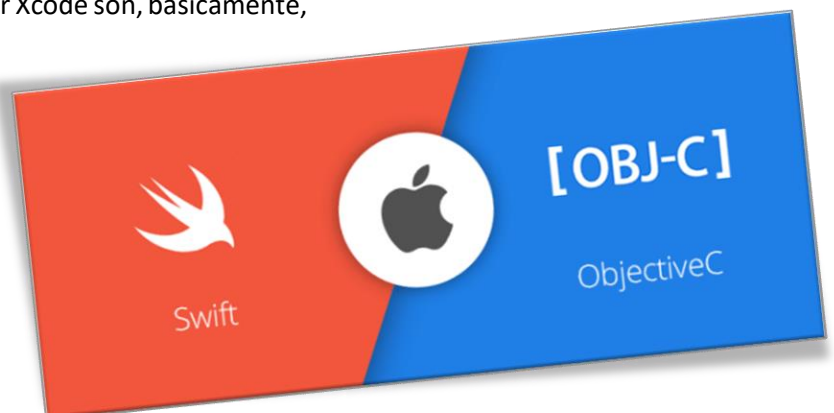
Aunque funcional, y muy popular, Objective-C es un lenguaje de programación bastante antiguo y complicado para aquellos que nunca han programado. Por ello, Apple creó otro lenguaje de programación universal para todos sus sistemas y dispositivos: Swift.

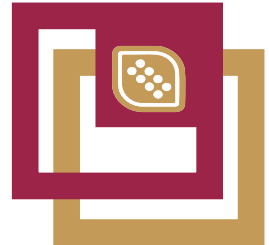
Swift es un lenguaje de programación igual de completo que Objective-C pero más sencillo de aprender. Cada vez más aplicaciones se basan en este lenguaje de programación ya que, entre otras cosas, permite crear fácilmente todo tipo de apps (da igual su finalidad), y ofrece un rendimiento bastante superior al de aplicaciones creadas en Objective-C (concretamente 2.6 veces superior) y Python (8.4 veces superior).



Igual que ocurre en Android, hay muchos programas que nos ayudan a programar y dar forma a las apps para iOS que estemos creando. Sin embargo, la mayoría de ellas deben pasar a través de **XCode**, el compilador de Apple para las apps de iOS, el cual solo está disponible para macOS.

Los requisitos mínimos para poder utilizar Xcode son, básicamente, cualquier MacBook, Mac Pro o iMac con 2 GB de memoria RAM. Si lo ejecutamos en una máquina virtual con VMware o VirtualBox, entonces necesitaremos una CPU bastante potente, y una configuración similar a la de estos equipos.

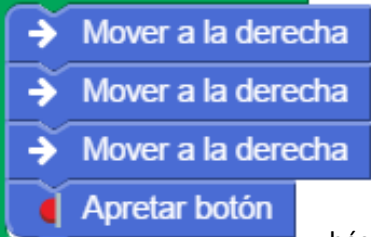




Programación por bloques

La programación es el proceso mediante el cual se analizan problemas, se diseñan soluciones y se codifican instrucciones en un lenguaje que la computadora o el dispositivo móvil puede comprender y ejecutar. En pocas palabras, un programa es un conjunto ordenado de instrucciones que permite a una aplicación o sistema realizar una tarea específica.

Al empezar a ejecutar



Aprender a programar fortalece el pensamiento lógico, analítico y crítico, impulsa la creatividad y favorece la resolución de problemas, ya que enseña a dividir situaciones complejas en partes más sencillas. Por esta razón, hoy en día la programación es una habilidad esencial que se promueve desde la educación básica hasta niveles superiores.

La programación por bloques para aplicaciones móviles consiste en construir programas mediante bloques gráficos prediseñados, los cuales se encastran entre sí, como si fueran piezas de un rompecabezas. Cada bloque representa una acción, una condición o un evento, lo que permite crear aplicaciones sin necesidad de escribir código tradicional.



Este tipo de programación es visual, intuitiva y accesible, por lo que permite a los estudiantes diseñar, probar, corregir errores y mejorar sus propios proyectos desde edades tempranas. Además, facilita la comprensión de conceptos fundamentales como eventos, ciclos, condiciones, variables y funciones, que también se utilizan en lenguajes de programación profesionales.

Gracias a la programación por bloques aplicada

a apps móviles, es posible crear de forma sencilla

juegos, animaciones, aplicaciones educativas,

calculadoras, controles con sensores, historias interactivas y muchas

herramientas más. Todo esto se logra de manera creativa, organizada y divertida.

Control/Eventos

Bucle

Condicional



Primitiva

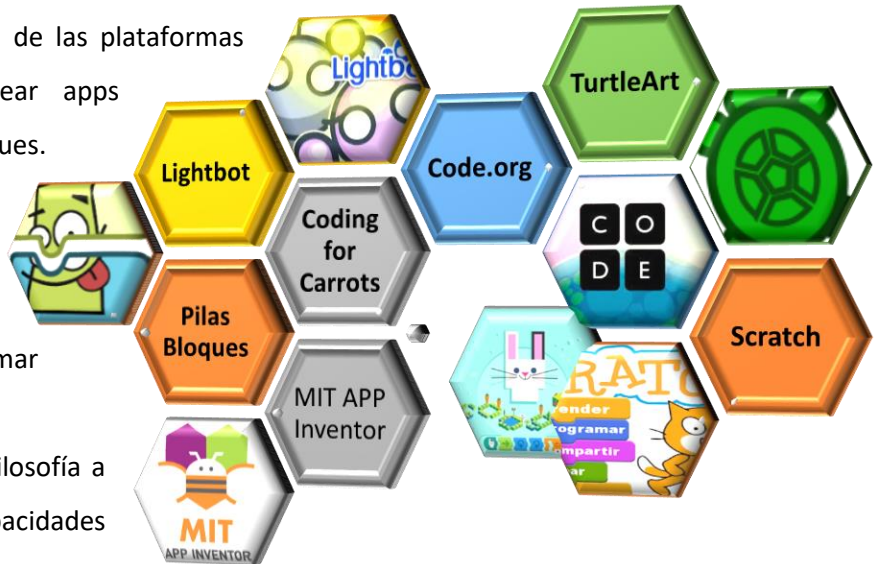
Variable

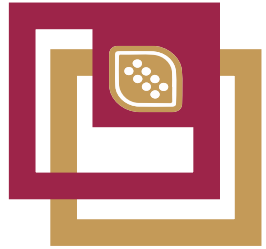
Primitiva



Ejemplos de herramientas de programación en bloques

- MIT App Inventor — Una de las plataformas más conocidas para crear apps Android mediante bloques. Con un navegador y un dispositivo Android (o emulador) puedes diseñar la interfaz, programar la lógica, y generar tu app.
- Thunkable — Similar en filosofía a App Inventor, pero con capacidades más modernas y soporte para iOS y Android. Es ideal si quieres hacer apps móviles más pulidas sin programar en texto.
- Kodular — Otra plataforma “drag-and-drop” para crear apps Android sin necesidad de código. Es útil si buscas más flexibilidad o funciones adicionales desde bloques.
- AppyBuilder — Basada en los mismos principios de App Inventor, permite crear apps Android desde el navegador, sin conocimientos previos de programación.
- Blockly — Aunque no es una plataforma de apps completa por sí sola, es una librería de bloques (desarrollada por Google) que puede usarse como base o “motor” visual para crear entornos de programación por bloques, lo que lo hace útil para quienes desean desarrollar algo más personalizado.





Instalación y uso Básico de un PWA



Ahora que ya tienes la diferencia entre un IDE, como aplicaciones nativas y un PWA como desarrollo en web. Es momento de seguir las siguientes instrucciones para trabajar con la herramienta adecuada en la programación móvil.

En la computadora o en el teléfono, vamos a utilizar la Aplicación Web Progresiva (PWA)

MIT APPINVENTOR



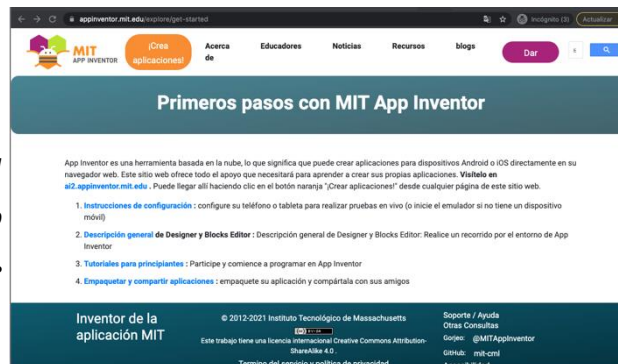
Para ello, debes abrir tu navegador preferido; de preferencia utiliza Google Chrome para hacer uso de las extensiones y datos que Google tiene con tu cuenta.



Escribe la URL: <https://appinventor.mit.edu/>



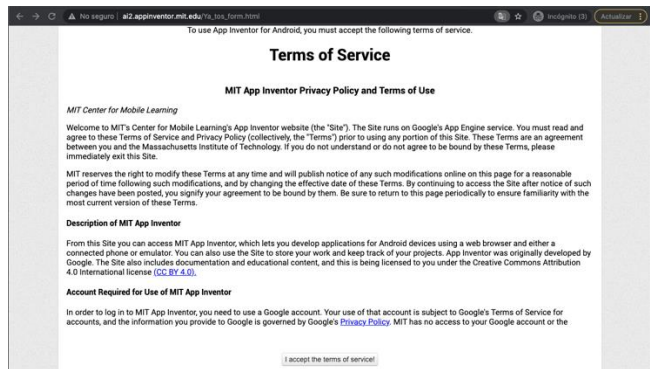
Después de ingresar
Puede aparecer en inglés,
pero puedes usar la
herramienta para traducir o
elegir en la lista desplegable
del idioma.

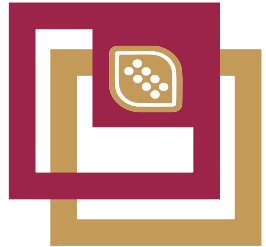


Seguidamente, elige o escribe tu cuenta
de correo electrónico de GMAIL

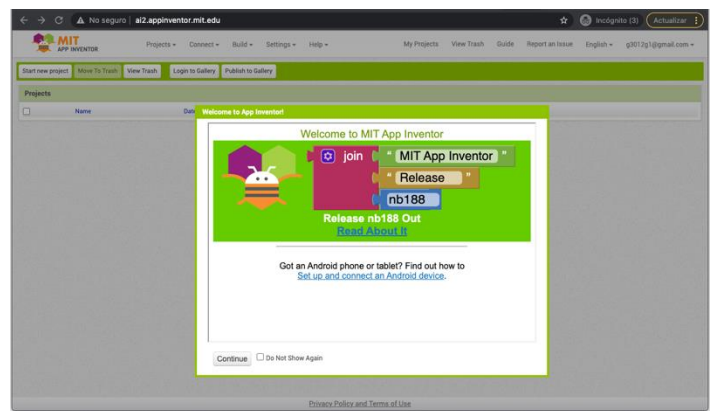


Una vez hecho login, debes leer y aceptar los términos y condiciones para poder utilizar la PWA y sincronizar tus dispositivos.

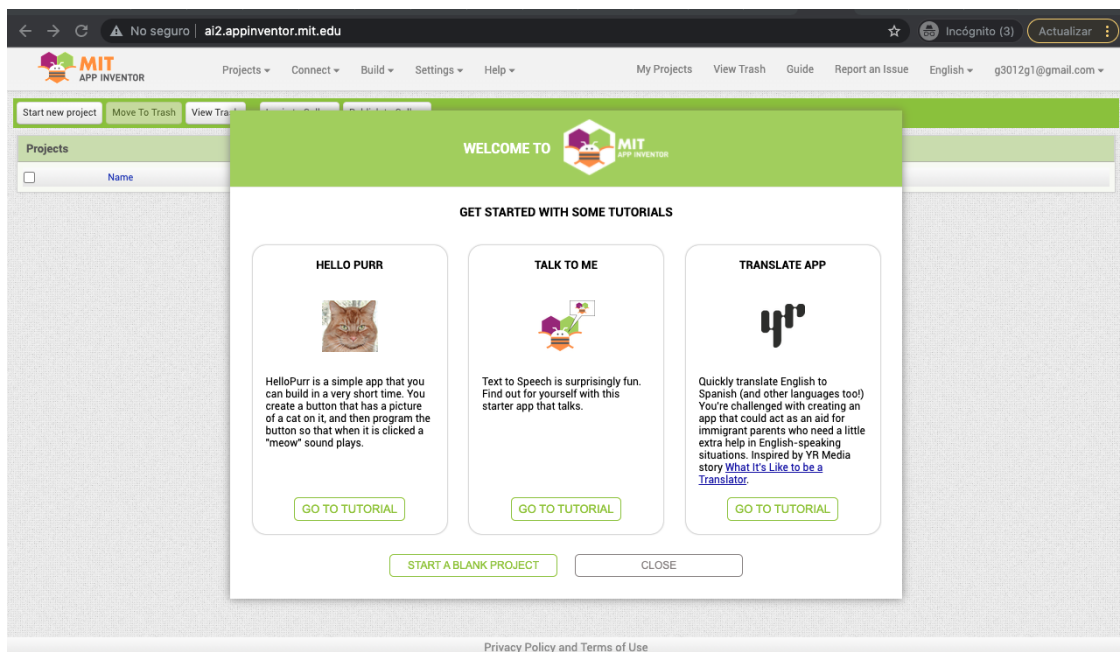




Recuerda que en caso de que te aparezca en inglés, puedes usar la herramienta del traductor:



Una vez aceptados los términos y condiciones, podrás ver la siguiente pantalla. ➡



Ahora puedes elegir entre los tutoriales de las aplicaciones de ejemplo o iniciar un PROYECTO en blanco. Los tutoriales corresponden a:

1. Ver el tutorial de una aplicación móvil que tiene una imagen en un botón y hace vibrar el teléfono además de escuchar un audio.
2. Ver el tutorial de una aplicación móvil que tiene un cuadro de texto de bienvenida y reproduce el texto en un audio.
3. Ver el tutorial de una aplicación móvil que traduce de inglés a español.



« En el móvil »



ANDROID

Debes buscar en **PlayStore Google Play**

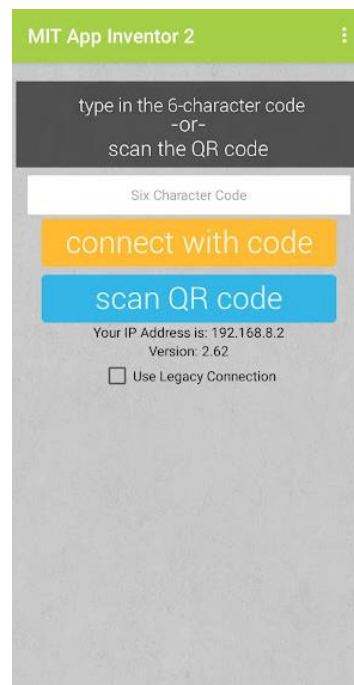
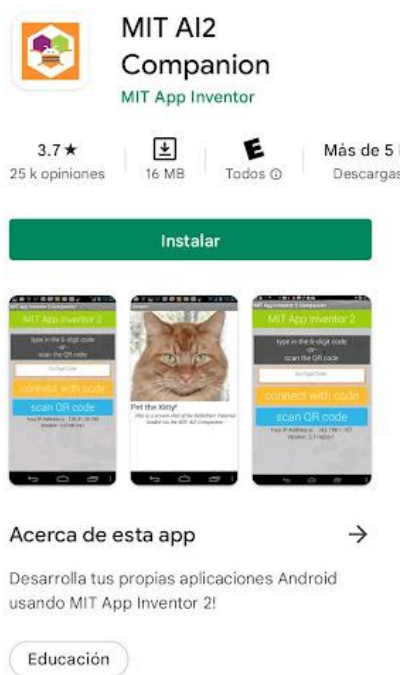


MIT AI2 Companion

O puedes buscarla como MIT App Inventor



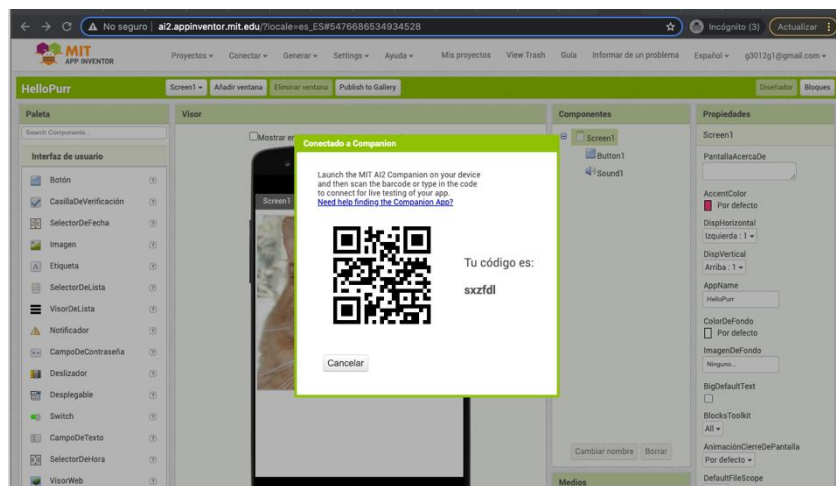
MIT AI2 Companion no es una aplicación independiente. Está destinado a ser utilizado con el sistema MIT App Inventor, una herramienta de creación de aplicaciones basada en la web que es de uso gratuito.

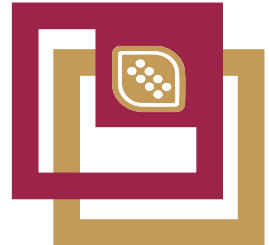


La aplicación sirve para poder visualizar las acciones, multimedia e interacciones que el programador integra en el proyecto de la aplicación móvil.



Para hacer la conexión, primero se debe tener un proyecto en desarrollo en el APPINVENTOR; para luego usar el botón desplegable **CONECTAR**, que te permitirá un código escrito o a través del escaneo del QR:





BIBLIOGRAFÍA

Claudia, A., & Claudia, A. (2018). *Educación Primaria: 6 Herramientas de Programación por bloques*. Educación Primaria. <https://blogedprimaria.blogspot.com/2018/01/programacion-por-bloques.html>

Download Android Studio and SDK tools |. (2021). Android Developers. https://developer.android.com/studio?hl=es-419&gclid=CjwKCAiAxJSPBhAoEiwAeO_fP-32akTVVM6HTS3SXEhS18GkXmpCddIq6kyYS9KV49ErhAwkGkHvPBoCPPAQAvD_BwE

González, J. D. M. (2021, 26 agosto). *Errores al Programar. Errores de programación*. Tipos de errores al programar y su respectiva solución. <https://www.programarya.com/Cursos/Fundamentacion/Errores>

Lenguaje de programación Objective-C 【características actualizadas】. (2019, 9 octubre). Lenguajes de programación. <https://lenguajesdeprogramacion.net/objective-c/>

Primeros pasos con App Inventor 2. (2016, 2 febrero). Código 21. <https://codigo21.educacion.navarra.es/autoaprendizaje/primeros-pasos-con-app-inventor-2/>

Velasco, R. (2021, 10 agosto). *¿Quieres hacer apps para móviles?* Estos son los lenguajes que debes conocer. SoftZone. <https://www.softzone.es/programas/lenguajes/programar-apps-moviles/>



ACTIVIDAD 2. IDENTIFICA TU ENTORNO DESARROLLADOR



Instrucciones: Después de ingresar e instalar el software sugerido en la Lectura No 2. Debes identificar los elementos de la ventana o capturar las pantallas de lo que lograste realizar al ingresar con tu cuenta en el MIT APPINVENTOR:



Cuenta de usuario

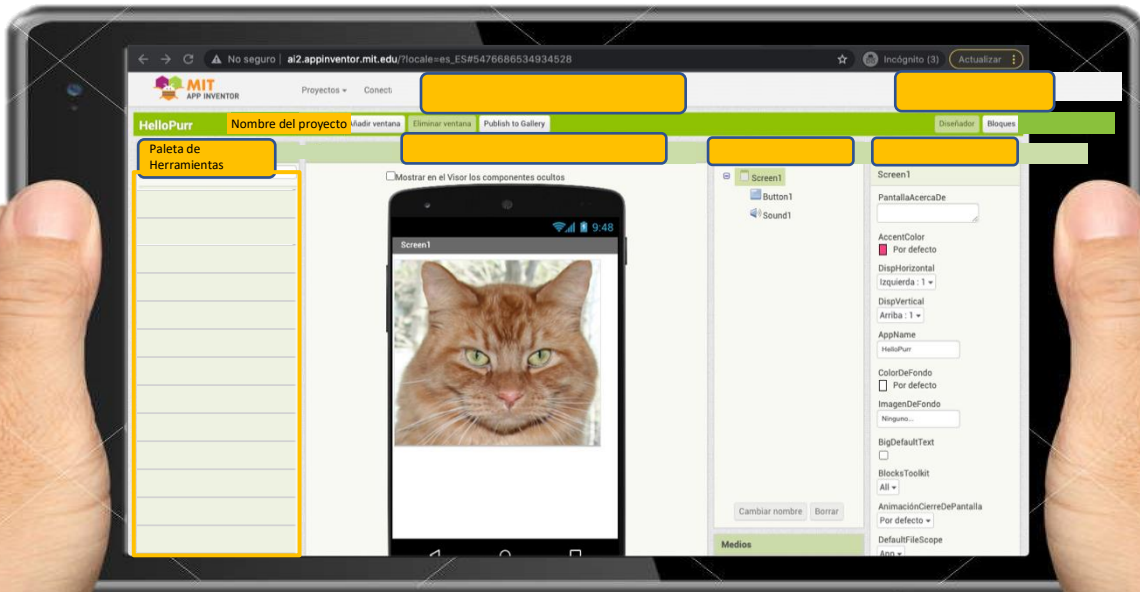
Componentes

Propiedades

Área de Menú

Paleta de Herramientas

Visor



Después de identificar los 6 elementos en el entorno, debes colocar el nombre del grupo de las herramientas que ubicamos en la paleta de herramientas.

No olvides colocar tus datos de identificación.

CONCLUSIÓN:

Menciona qué es una aplicación web progresiva.

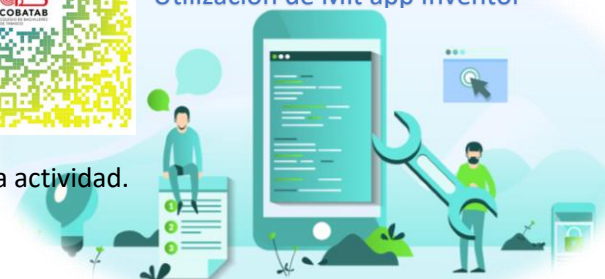
En qué consiste la programación por bloques.

Describe las dificultades presentadas en el desarrollo de esta actividad.



PRESENTACIÓN ELECTRÓNICA

Utilización de Mit app Inventor





INSTRUMENTO DE EVALUACIÓN
LISTA DE COTEJO 2
Actividad 2: "Identifica Tu Entorno Desarrollador"



DATOS GENERALES

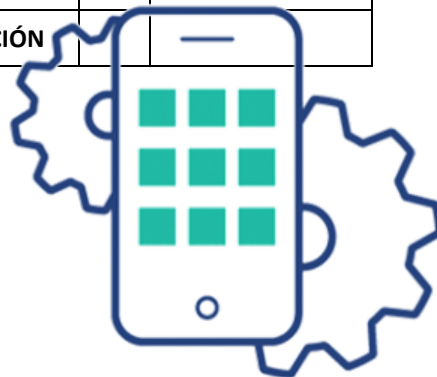
Nombre(s) del alumno(s)	Matrícula(s)
Producto: Interfaz del entorno	Fecha
Submódulo 1: Programación Móvil y Juegos	Periodo: 2025 – 2026B
Nombre del docente	Firma del docente

CRITERIO	INDICADORES	VALOR OBTENIDO		PONDERACIÓN	CALIF	OBSERVACIONES Y/O SUGERENCIAS DE MEJORA
		SÍ	NO			
1	Presenta sus datos para identificarse.			1		
2	Identifica los 6 elementos de la interfaz.			2		
3	Presenta los 12 bloques de la paleta de herramientas.			2		
4	Presenta la ilustración de la interfaz.			1		
5	No presenta faltas ortográficas.			1		
6	Elabora una conclusión personal y entendible.			2		
7	Entrega en tiempo y forma.			1		
CALIFICACIÓN						



PRESENTACIÓN ELECTRÓNICA

Programación por bloques



Lectura No 3. Diseña una App Móvil

Instrucciones: Realiza el análisis de la siguiente información, posteriormente, desarrolla la Actividad 3 Fases de una app.



LECTURA 3. "DISEÑA UNA APP MÓVIL"

PRESENTACIÓN ELECTRÓNICA



Diseño de aplicaciones: así se crea una app desde cero

Contextualización problema_Diseño



Los dispositivos móviles actualmente se han vuelto una herramienta necesaria para satisfacer ciertas necesidades. No solo por la comodidad que proporcionan, sino también porque desde ellos se pueden utilizar todo tipo de apps destinadas a diversas tareas, como pueden ser actividades laborales, de ocio, o de realización de compras.

Para ello, empresas e instituciones de todo tipo, e incluso desarrolladores independientes, han creado sus apps tras exhaustivos procesos de diseño y su posterior desarrollo. Tomando en cuenta todas las fases necesarias para la creación de una app desde cero.



Lo primero, es tener una buena idea.

Antes de iniciar el diseño de pantallas o la escritura de código, hay que tener una buena idea de que tipo de app crear y que beneficios podrá tener. Generalmente, para saber si una idea es buena o viable debe cumplir un objetivo principal: solucionar uno o varios problemas de los usuarios de forma sencilla.

¿Cómo se desarrolla este proceso?



En caso de que no tengas una idea, puedes encontrar primero un problema que pueda solucionarse con una app para después diseñar cómo resolverlo con una aplicación.



Una vez que hayas identificado el tipo de aplicación que deseas diseñar, el otro aspecto que igual se debe considerar es el presupuesto con el que vas a contar para desarrollarla. Porque de este depende que puedas adquirir de todas las funciones que deseas, o solo de algunas.

Diseño de aplicaciones: primeros esbozos y explicación de la idea

Cuando tengas definida la idea y el presupuesto para solventar el desarrollo de la aplicación, debes considerar elegir un equipo de desarrollo, con la finalidad de agilizar el proceso de la creación de la app.



Cuando tengas elegido el equipo de desarrollo se debe explicar cuál es el objetivo de la aplicación a desarrollar, las funciones que quieres que tenga. Los integrantes del equipo analizarán los datos, verán si es viable desarrollar todo lo propuesto, y comenzarán a diseñar las primeras pantallas y bocetos de la aplicación; como la interfaz que debe contener las principales funciones, estructura y acceso al menú u opciones para interactuar en la aplicación.

Diseño de aplicaciones: prototipado

El prototipado es una versión inicial de la idea de un producto o servicio, nos permite probar, evaluar y validar si efectivamente la idea que tenemos en mente cumple los objetivos.



Wireframing



Es una de las fases iniciales en la creación de un producto digital y consiste en el diseño de la estructura que tendrá cada una de las pantallas o interfaces de ese producto. Es decir, es el boceto de cómo serán cada una de las pantallas de una web o app. Pero uno sin colores, logos o tipografías. Porque lo que importa en el Wireframing no es el aspecto visual, sino cuál es el orden, qué tipo de

contenido irá en cada bloque y a dónde lleva cada botón.

Por lo que en este punto toca desarrollar el boceto y sus partes. Para ello, es habitual utilizar distintas herramientas y utilidades de creación de prototipos, que serán de gran ayuda para el diseño de nuestra app. Con ellas puedes colocar datos y gráficos en el punto en el que quieres que aparezcan, insertar botones, y ver cómo responderían al navegar por ellos.

Desarrollo del backend y frontend

Backend: Es la parte de desarrollo y escritura de código de una app, lo que no se ve. Para darle forma, y que realice en un primer momento una o varias de las acciones requeridas, es necesario saber programar.

Conforme vaya avanzando el diseño y el

desarrollo de la app, es conveniente que haya personas que prueben el funcionamiento de las novedades.



Frontend: lo que se ve en diseño de aplicaciones, es decir, lo que se ve de la aplicación: sus pantallas. Hay que definir qué aparecerá exactamente en cada una, y después desarrollarla. Básicamente, el trabajo de desarrollo del frontend es crear las versiones digitales y en alta calidad de las distintas pantallas diseñadas durante el proceso de Wireframing.



Desarrollo de App Móviles

Según lo definido por MicroServices, “el desarrollo de aplicaciones móviles son los procedimientos y procesos establecidos que intervienen cuando se crea software para pequeños dispositivos informáticos inalámbricos, como tabletas y teléfonos inteligentes. Al igual que el desarrollo de aplicaciones web, los procesos de desarrollo de aplicaciones móviles tienen sus raíces en el desarrollo de software tradicional.”

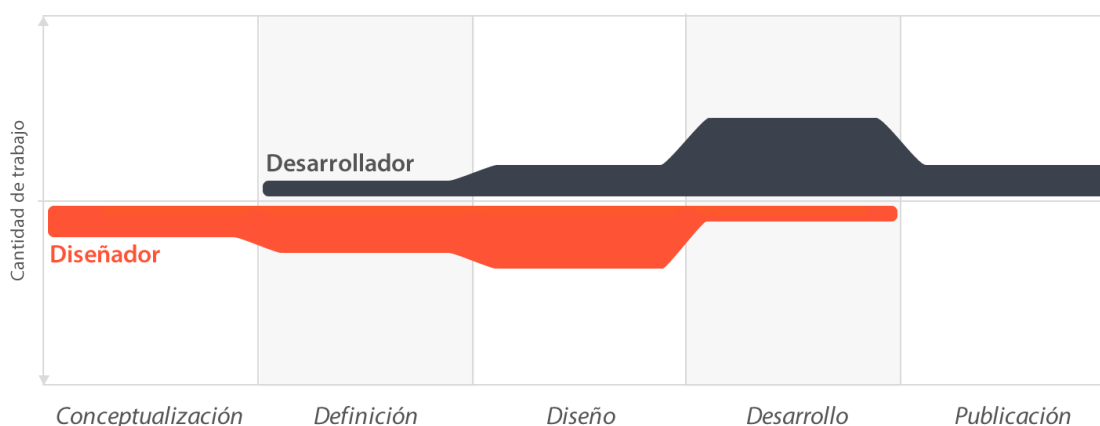


El proceso de diseño y desarrollo de apps

El proceso de diseño y desarrollo de apps abarca desde la concepción de la idea hasta el análisis posterior a su publicación en las tiendas. Durante las diferentes etapas, diseñadores y desarrolladores trabajan —la mayor parte del tiempo— de manera simultánea y coordinada.

El proceso de diseño abarca diferentes etapas donde diseñador y desarrollador trabajan simultáneamente, con más o menos carga de trabajo dependiendo del momento.

Para crear una aplicación deben considerarse las siguientes fases:





Fase del Proceso	Rol Dominante	Descripción de la Carga de Trabajo
Conceptualización	Diseñador (Carga Máxima)	Es la etapa inicial donde el diseñador lidera. Se define el core del producto, el público objetivo y se crean los primeros wireframes y flujos de usuario. La carga del desarrollador es mínima o nula.
Definición	Diseñador (Carga Media) y Desarrollador (Carga Baja)	El diseñador refina la experiencia de usuario (UX). El desarrollador comienza la planificación técnica, definiendo la arquitectura del sistema y evaluando la viabilidad de las características propuestas por el diseño.
Diseño	Diseñador (Carga Media)	El diseñador finaliza la Interfaz de Usuario (UI), creando mockups de alta fidelidad, la guía de estilos y el design system. El desarrollador incrementa su carga preparándose para la codificación.
Desarrollo	Desarrollador (Carga Máxima)	Esta es la fase central del desarrollador. El diseñador reduce su carga, enfocándose en entregar activos, resolver dudas del desarrollador e implementar correcciones de diseño (QA). El desarrollador se encarga de codificar el frontend, backend y realizar las pruebas unitarias.
Publicación	Desarrollador (Carga Media)	El desarrollador se encarga de la compilación final, la configuración de metadatos y la subida a las tiendas de aplicaciones (App Store o Google Play).



1. CONCEPTUALIZACIÓN

El resultado de esta etapa es una idea de aplicación, que tiene en cuenta las necesidades y problemas de los usuarios. La idea responde a una investigación preliminar y a la posterior comprobación de la viabilidad del concepto.

- Ideación
- Investigación
- Formalización de la idea



2. DEFINICIÓN

En esta fase del proceso se describe con detalle a los usuarios para quienes se diseñará la aplicación, usando metodologías como «Personas» y «Viaje del usuario». También aquí se sientan las bases de la funcionalidad, lo cual determinará el alcance del proyecto y la complejidad de diseño y desarrollo de la app.

- Definición de usuarios
- Definición funcional



3. DISEÑO

En la etapa de diseño se llevan a un plano tangible los conceptos y definiciones anteriores, primero en forma de wireframes, que permiten crear los primeros prototipos para ser probados con usuarios, y posteriormente, en un diseño visual acabado que será provisto al desarrollador, en forma de archivos separados y pantallas modelo, para la programación del código.



- Wireframes
- Prototipos
- Test con usuarios
- Diseño visual



4. DESARROLLO

El programador se encarga de dar vida a los diseños y crear la estructura sobre la cual se apoyará el funcionamiento de la aplicación. Una vez que existe la versión inicial, dedica gran parte del tiempo a corregir errores funcionales para asegurar el correcto desempeño de la app y la prepara para su aprobación en las tiendas.



- Programación del código
- Corrección de bugs



CODING

5. PUBLICACIÓN

La aplicación es finalmente puesta a disposición de los usuarios en las tiendas digitales. Luego de este paso trascendental se realiza un seguimiento a través de analíticas, estadísticas y comentarios de usuarios, para evaluar el comportamiento y desempeño de la app, corregir errores, realizar mejoras y actualizarla en futuras versiones.



La app ya sería funcional, pero hay que comprobar que todos sus elementos funcionan correctamente. En esta fase debe anotarse todo lo que no funciona o puede mejorar, y mejorarlo en el código de la app.

TIPOS DE APLICACIONES MÓVILES

La popularidad exponencial de los teléfonos inteligentes y tabletas ha llevado al aumento de la creación de software en línea con el desarrollo de aplicaciones móviles. Los dos sistemas operativos líderes, iOS y Android, han marcado el ritmo en la estandarización de los diferentes tipos de desarrollo de aplicaciones móviles para programadores.

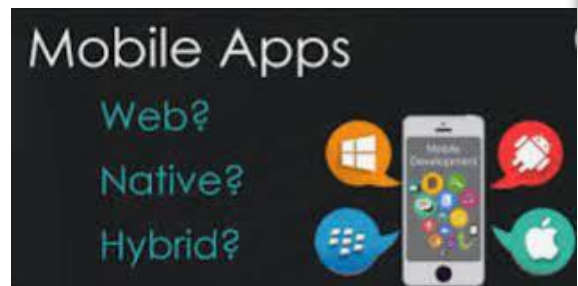
Los tipos de aplicaciones móviles incluyen:

📌 Aplicaciones Nativas

Las aplicaciones nativas están diseñadas para plataformas de dispositivos específicos, ya sea Android o iOS. Se descargan o instalan a través de una tienda de aplicaciones y se accede a través de un ícono en el dispositivo. Las aplicaciones nativas están diseñadas para aprovechar al máximo las características del dispositivo como el GPS, la cámara y las listas de contactos, entre otros. Cuando usted piensa en una aplicación móvil, la primera que le viene a la mente es probablemente una aplicación nativa. Ejemplos:



- Aplicaciones de redes sociales: Facebook, Twitter, Pinterest
- Juegos: Pokémon Go, Candy Crush
- Programas de navegación: Waze



📌 Aplicaciones web o HTML5

Basándose en las tecnologías web universales y estandarizadas, como HTML5, JavaScript y CSS, las aplicaciones web se implementan como un sitio web que simplemente se ve y se siente como aplicaciones nativas. Funcionan y se ejecutan en un navegador escrito típicamente en HTML5. Estas aplicaciones sólo aprovechan las funciones de GPS y cámara de un dispositivo.

- Ejemplos: AliExpress, OLX y Twitter Lite.



📌 Aplicaciones híbridas

Las aplicaciones híbridas de JavaScript, HTML y CSS son una combinación de aplicaciones nativas y web. Se obtienen de una tienda de aplicaciones y aprovechan las características del dispositivo, como una aplicación nativa. Al igual que la aplicación web, se accede desde un navegador y se basa en HTML.

En pocas palabras, las aplicaciones híbridas se instalan como una aplicación nativa, pero operan como una aplicación web.

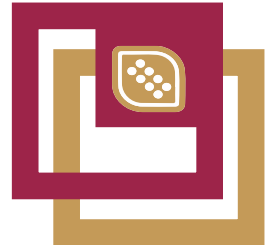
Ejemplos:

- Streaming de motores de búsqueda como JustWatch
- Redes sociales como Instagram y Untapped



	Aplicaciones Nativas	Aplicaciones Web	Aplicaciones Híbridas
Lenguajes de Programación	C++ C# Java Python	HTML CSS JavaScript PHP	Una combinación de lenguajes
IDE / ENTORNO DE DESARROLLO	Objective C Swift Eclipse Kotlin	El uso de componentes UX creados con un lenguaje propio nos ata a un framework concreto, React Native y Nativescript utilizan componentes de UX nativos para intentar mejorar el rendimiento.	FRAMEWORKS Ionic React Native Laragon / Laravel Angular





BIBLIOGRAFÍA

Batanero, A. S. M. Y. (2021, 29 julio). *VERSUS: Apps híbridas VS Apps Nativas*. Paradigma Digital.
<https://www.paradigmadigital.com/dev/versus-apps-hibridas-vs-apps-nativas/>

Feliciano, C. (2021, 16 febrero). *¿Qué es el desarrollo de aplicaciones móviles?* INVID.
<https://invidgroup.com/es/que-es-el-desarrollo-de-aplicaciones-moviles/>

García, M. (2021, 30 agosto). *Diseño de aplicaciones: así se crea una app desde cero*. ESDESIGN.
<https://www.esdesignbarcelona.com/actualidad/disenio-web/disenio-de-aplicaciones-asi-se-crea-una-app-desde-cero>

K. (2020, 24 octubre). *¿Cuál es el proceso de diseño y desarrollo de apps?* Kubo - Agencia digital de desarrollo de aplicaciones móviles, Colombia. <https://kubo.co/2018/11/19/desarrollo-de-apps-moviles/>



P. (2020b, marzo 6). *Apps híbridas, 7 frameworks para crearlas*. Presupuesto APP.
<https://presupuestoapp.com/descubre-7-frameworks-para-crear-apps-hibridas/>

Unir, V. (2021, 7 enero). *Qué es wireframing y qué características tienen estos prototipos*. UNIR.
<https://www.unir.net/ingenieria/revista/que-es-wireframing/>



Actividad 3. Fases de una App



Instrucciones: Realizar una imagen interactiva o infografía, donde se muestres las fases de desarrollo de una aplicación que lograste comprender de la lectura 3. Puedes hacer uso de las Aplicaciones Nativas, Web o Híbridas propuestas:



HAIKU DECK





Instrucciones: Genially como aplicación web te permitirá

Desarrollar imágenes interactivas sin mucho esfuerzo.

Para realizar la actividad, se requiere registrarse en línea en la aplicación Genially, puedes dar de alta a tu cuenta de Google para que sea más rápido el acceso.

Dar clic en este link <https://genial.ly/es/>



Manos a la obra



Cómo crear una imagen interactiva | Tutorial Genialmente



https://www.youtube.com/watch?v=_CMPVKlw9PU&t=4s



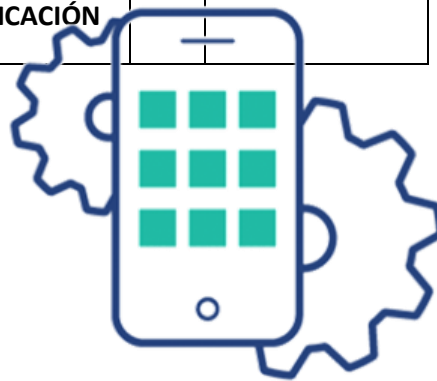
INSTRUMENTO DE EVALUACIÓN
LISTA DE COTEJO 3
Actividad 3: Fases de una App



DATOS GENERALES

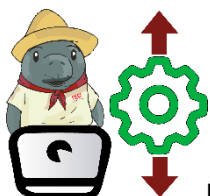
Nombre(s) del alumno(s)	Matrícula(s)
Producto: Imagen interactiva o Infografía	Fecha
Submódulo 1: Aplicación Móvil y Juegos	Periodo: 2025 – 2026B
Nombre del docente	Firma del docente

CRITERIO	INDICADORES	VALOR OBTENIDO		PONDERACIÓN	CALIF	OBSERVACIONES Y/O SUGERENCIAS DE MEJORA
		SÍ	NO			
1	Presenta las 5 fases del desarrollado de una app.			2		
2	Presenta ilustraciones acordes a la información que proporciona.			2		
3	Se aprecia el conocimiento y dominio del tema.			2		
4	Muestra dominio en el uso de las tecnologías de la Información y de la comunión.			2		
5	No presenta faltas ortográficas.			1		
6	Entrega en tiempo y forma.			1		
	CALIFICACIÓN					



Práctica 1. Proponiendo Mi APP

Instrucciones: Organízate con tus compañeros para integrar un equipo de 5 integrantes, es momento de aplicar lo aprendido y desarrollar una propuesta de aplicación móvil.



PRÁCTICA 1. "PROPONIENDO MI APP"

De manera escrita o digital, deben desarrollar una propuesta de aplicación móvil, que solvete una necesidad, describiendo las características y que beneficios proporciona.

Considera el siguiente recurso audiovisual como apoyo para desarrollar su propuesta de la aplicación móvil, además de la lectura 3; no olviden presentar sus datos completos.



Manos a la obra



<https://youtu.be/-vkgGJvIG80>

TEAM
LEWAI



DESARROLLO DE SOFTWARE



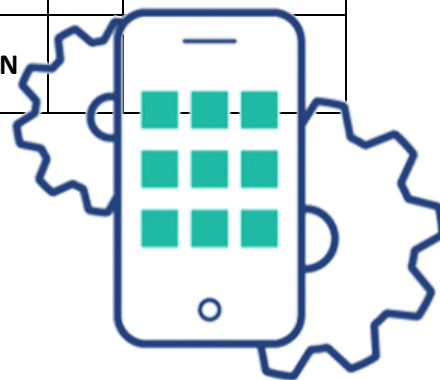
INSTRUMENTO DE EVALUACIÓN
LISTA DE COTEJO 4
Práctica 1: Proponiendo mi App



DATOS GENERALES

Nombre(s) del alumno(s)	Matrícula(s)
Producto: Documento digital o escrito	Fecha
Submódulo 1: Aplicación Móvil y Juegos	Periodo: 2025 – 2026B
Nombre del docente	Firma del docente

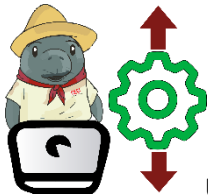
CRITERIO	INDICADORES	VALOR OBTENIDO		PONDERACIÓN	CALIF	OBSERVACIONES Y/O SUGERENCIAS DE MEJORA
		SÍ	NO			
1	La app propuesta una conceptualización correcta.			2		
2	Expresan correctamente la definición de la app propuesta.			2		
3	El diseño propuesto, coincide con la conceptualización.			2		
4	Se aprecia el conocimiento y dominio del tema.			2		
5	Muestra dominio en el uso de las tecnologías de la Información y de la comunión.			1		
6	Entrega en tiempo y forma.			1		
	CALIFICACIÓN					





Práctica Guiada “Aplicate para dibujar”

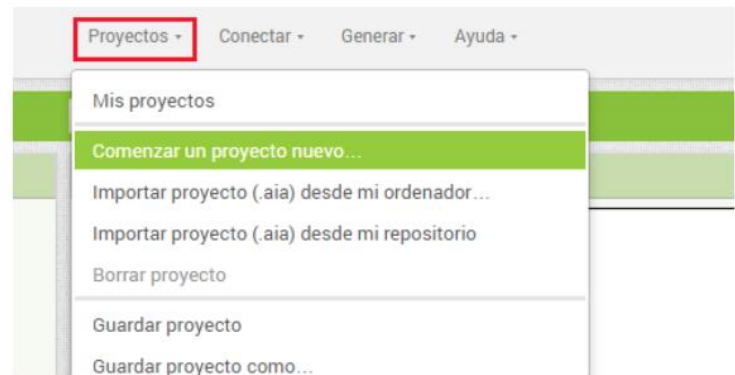
Instrucciones: Prepara tu equipo de cómputo con MIT APP INVENTOR y sigue las indicaciones para crear una aplicación móvil que permite dibujar y guardar el dibujo.



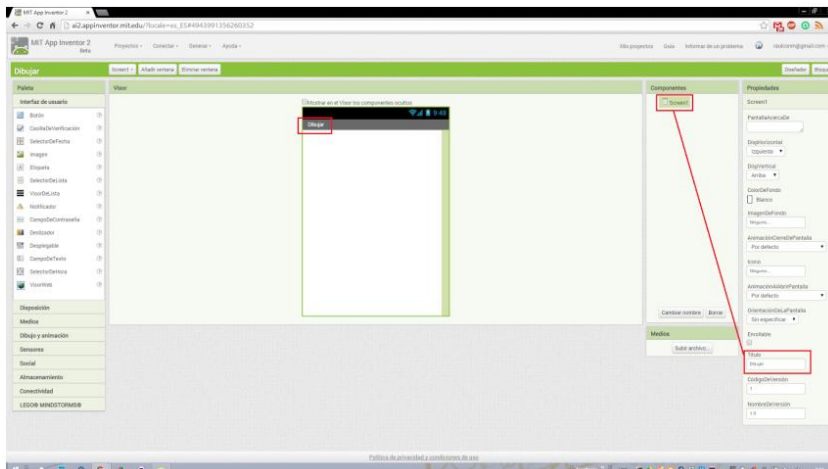
PRÁCTICA GUIADA. “APLÍCATE PARA DIBUJAR”

Una vez que hemos visto el manejo más básico de las

pantallas, el Diseñador, y el Editor de Bloques de App Inventor, ya estamos listos para hacer algo más avanzado. Comenzaremos haciendo clic en el desplegable **Proyectos**, y eligiendo la opción **Comenzar un proyecto nuevo...**



Una vez dentro, cambiaremos el nombre de la pantalla Screen1 por Dibujar (en la propiedad Título del componente Screen1).



ATENCIÓN

Es importante siempre utilizar nombres descriptivos para los objetos que vamos creando. Al principio, cuando nuestras

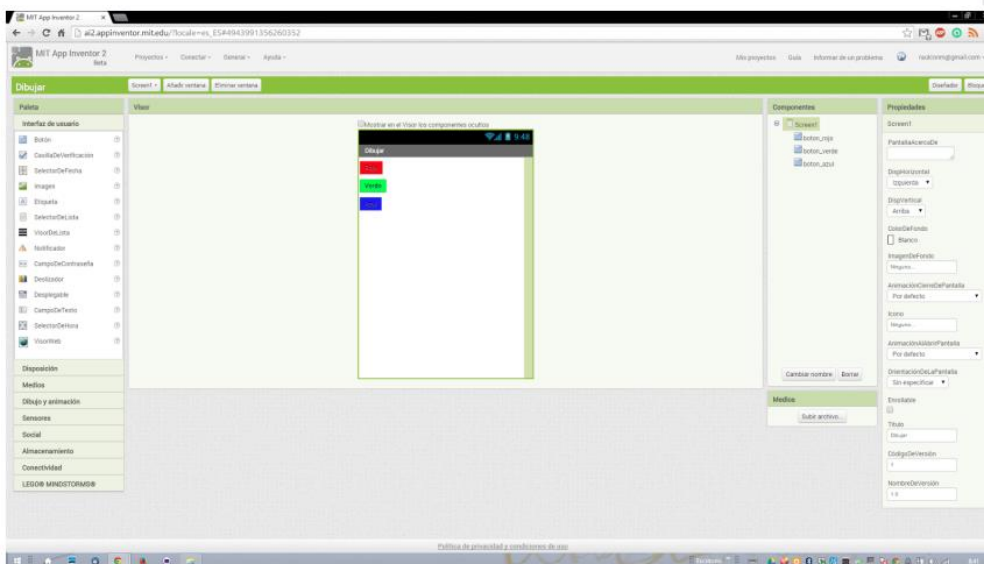
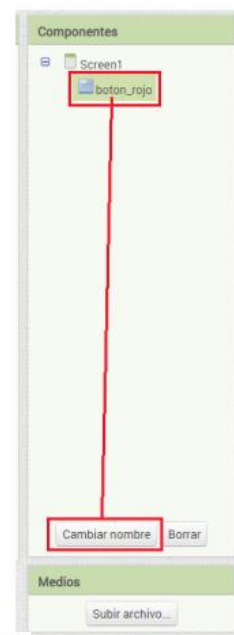
aplicaciones son pequeñas, es fácil recordar cada objeto, pero a medida que los programas van haciéndose más complejos es fundamental saber para qué sirve cada objeto, cada variable, y sólo podremos hacerlo si le hemos dado un nombre que describe qué es o para qué sirve.





Ahora crearemos por nuestra cuenta un botón, y cambiaremos su propiedad **Texto** por “Rojo”, y finalmente pondremos en rojo su propiedad **ColorDeFondo**. Adelante, solamente hay que jugar con las del botón. Cambiaremos el nombre del botón **Botón1** por **boton_rojo** en el cuadro de componentes del Diseñador. En lugar de usar un espacio utilizaremos un signo de subrayado, y no emplearemos tildes, porque el sistema entiende que es un carácter especial no válido (esto aún está en inglés 😊).

Crearemos después otro botón verde, y finalmente otro azul, y haremos lo mismo que hemos hecho con el rojo, cambiaremos su nombre, el color de fondo, y el texto que aparece en el botón.



Como queremos que los botones queden en una misma línea horizontal, vamos a recolocarlos en la pantalla añadiendo un componente **DisposiciónHorizontal**. Lo arrastramos desde el cajón hasta el Visor. Este objeto aparecerá en el Visor como un cuadro vacío.

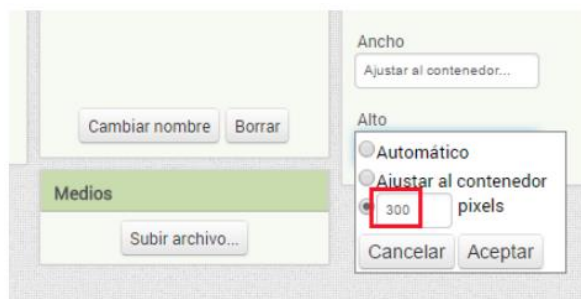
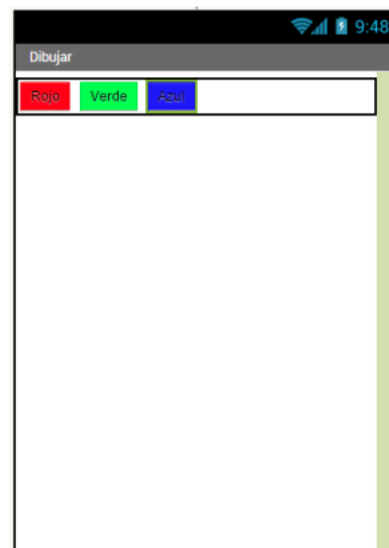
Para que se ajuste al ancho de la pantalla del visor ¿qué haremos? En su propiedad **Ancho** elegiremos la opción **Ajustar al contenedor...**





Ahora, en el Visor, arrastraremos los tres botones dentro de **DisposiciónHorizontal1**. Si no nos gusta este nombre podemos cambiarlo por **MarcoHorizontal**, por ejemplo. Para que la altura del marco se ajuste al tamaño de los botones podemos decir que su **Altura** sea automática. Así siempre se adaptará al tamaño del componente más alto de todos los que contiene.

Ahora añadimos el lienzo, arrastrando al Visor el componente **Lienzo**, que se encuentra dentro del cajón (grupo) **Dibujo y animación** de la Paleta. Lo colocamos justo debajo, fuera del **MarcoHorizontal**. Lo más cómodo es definir que su anchura (propiedad **Ancho**) sea automática, para que se extienda hasta los bordes izquierdo y derecho del Visor. En cuanto a su altura, mejor experimentar con diferentes tamaños hasta que ocupe el espacio que queremos.



El tamaño se especifica en *pixels*, es decir, en puntos de la pantalla. Cada línea de la pantalla tiene un número de pixels. El número de ellos que haya, en líneas y columnas, define la resolución de la pantalla. Podemos probar con 300 pixels, y modificarlo más tarde si vemos que no es el mejor tamaño.



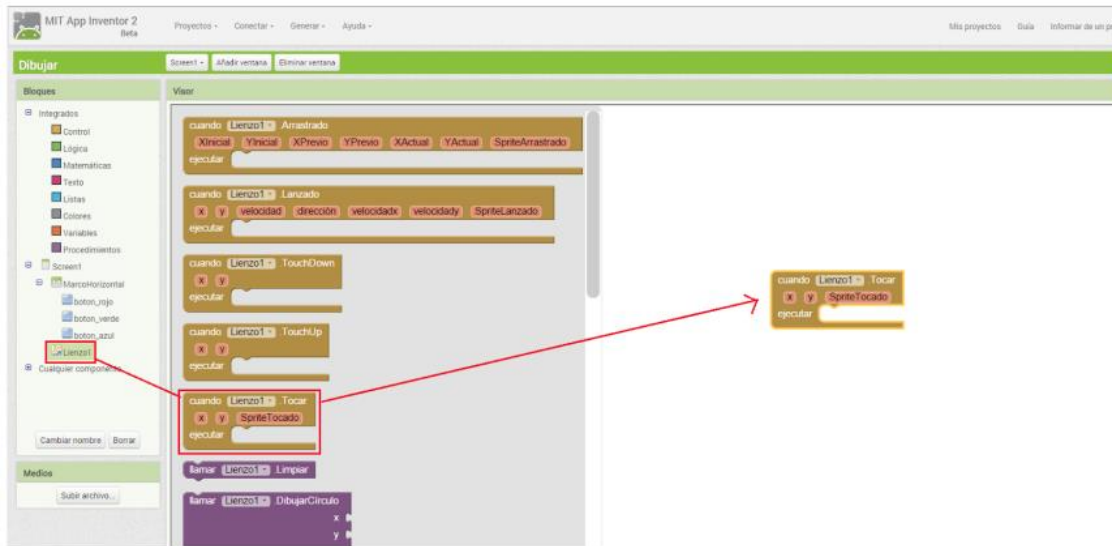
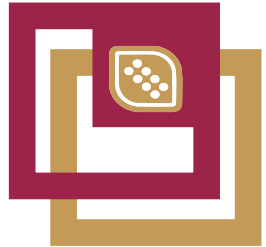
Ahora vamos a definir el comportamiento del programa, a decir cómo tiene que funcionar. Vamos entonces al editor de bloques.



MIT
APP INVENTOR

Arrastraremos el bloque **cuando.Lienzo1.Tocar** desde el cajón del objeto **Lienzo1** hasta el editor. Esto indica que cada vez que toquemos el lienzo con el dedo tendrá que ocurrir lo que digamos dentro de este bloque mostaza.





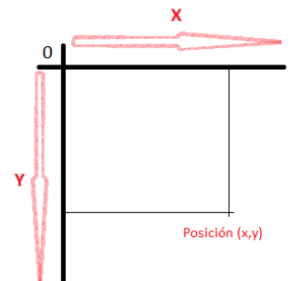
Ahora arrastraremos el bloque morado (de acción) **llamar.Lienzo1.DibujarCírculo** hasta encajarlo dentro del bloque mostaza. Veremos que para funcionar necesita que le asociemos a la derecha tres bloques de información adicional (x, y, r).

Coordenadas x e y

El pensador René Descartes inventó hace tiempo un sistema para definir la posición de un objeto sobre un plano. En su honor, este sistema se conoce como *Coordenadas Cartesianas*. Para conocer más sobre este tema visitemos la página http://es.wikipedia.org/wiki/Coordenadas_cartesianas

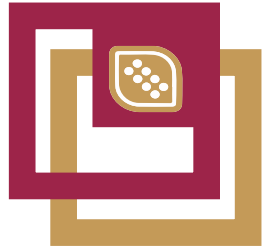


Nosotros vamos a utilizar el sistema de Descartes para referirnos a un punto cualquiera dentro de nuestro **Lienzo1**. Siempre que toquemos con el dedo dentro del lienzo, el bloque mostaza guardará cuál es el valor de la *coordenada x* y la *coordenada y* del punto que estamos tocando, porque seguro que lo necesitaremos. App Inventor utiliza el sistema cartesiano para determinar la posición de un punto determinado dentro de un lienzo. La x aumenta desde la izquierda a la derecha, y la y lo hace desde arriba hacia abajo.

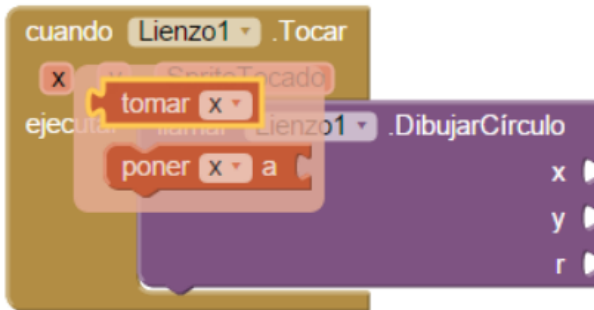


Precisamente para saber dónde dibujar el círculo, el bloque morado deberá saber que **x** e **y** tiene el punto de la pantalla que estamos tocando. Lo tomaremos del bloque mostaza.



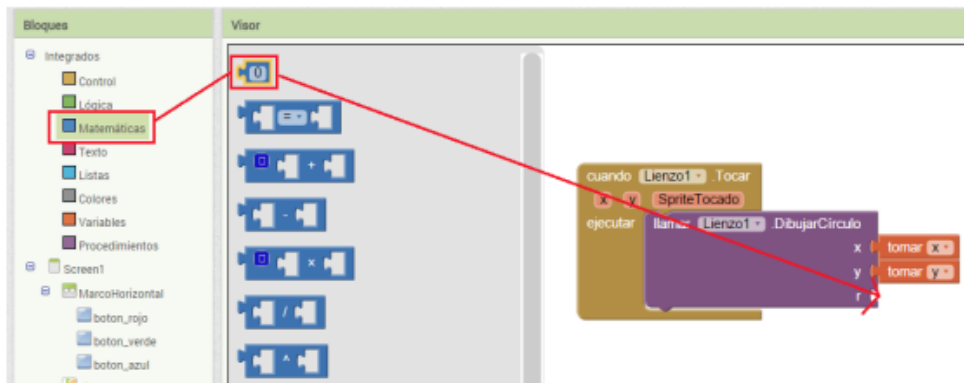


Al dejar quieto el puntero del ratón sobre el icono de la **x** del bloque mostaza nos saldrá una ventanita en la que aparecerá un pequeño bloque **tomarx** de color naranja. Tenemos que arrastrarlo hasta encajarlo con el hueco superior del bloque morado **llamar.Lienzo1.DibujarCírculo**.



Haremos lo mismo con la **y**. Esto hará que el círculo se dibuje sobre la **x** y la **y** que estamos tocando, y no en cualquier otro punto de la pantalla.

Nos falta definir la **r**. ¿Qué es? Indica el tamaño del círculo que vamos a dibujar, su radio (de ahí la **r**). Para definir el tamaño, el radio, abrir el cajón (Bloque) **Matemáticas** de la zona de bloques y arrastrar el bloque azul que indica el valor 0 (cero) hasta el hueco **r** de nuestro bloque morado.



Queremos que los círculos se vean bien, así que le asignaremos a **r** el valor 10. El radio del círculo será 10.

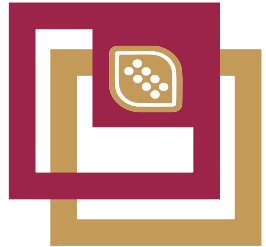


Ahora podemos tocar dentro del lienzo, en la pantalla del móvil o el emulador, para ver cómo se dibujan pequeños círculos. Pero para que los círculos sean rojos, verdes o azules en lugar de negros, hay que decirle al móvil que lo haga creando nuevos bloques. Una pista: hay que utilizar un bloque mostaza del objeto **botón_rojo**. Ya hemos usado este tipo de bloque mostaza en el programa del gato.

Utilizaremos el bloque **cuando.boton_rojo.Clic**. Lo arrastramos hasta el editor de bloques.

¿Qué tiene que ocurrir cuando hagamos clic sobre el botón rojo? Tiene que cambiarse a rojo el color del pincel que usamos en el lienzo. ¿Dónde tendremos que definir eso? Una pista: el color del pincel es una característica del **Lienzo1**.

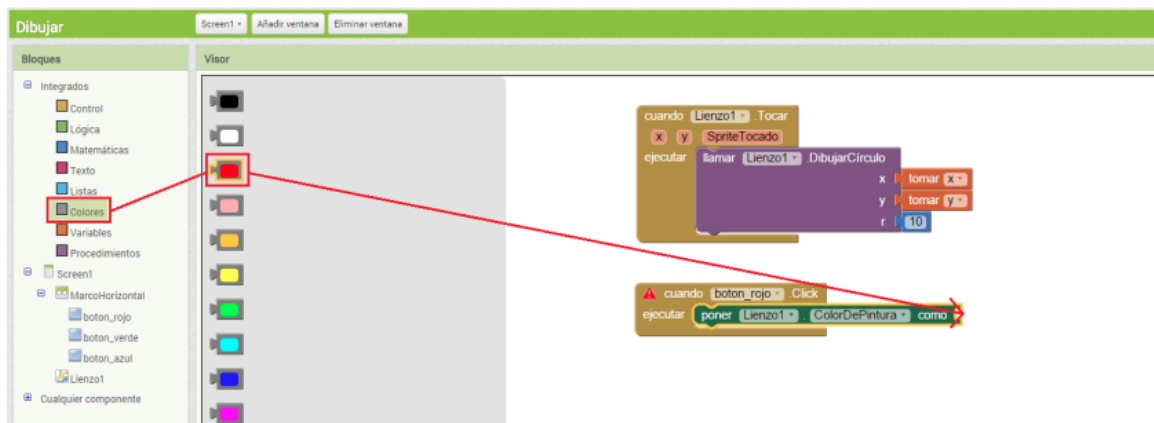




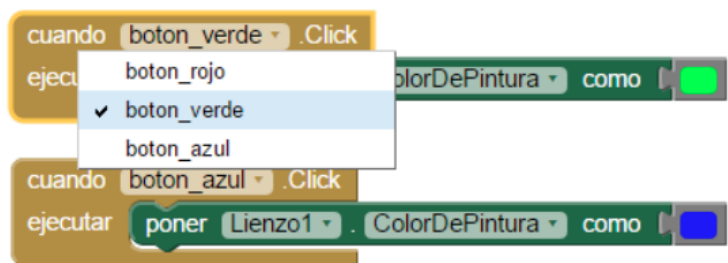
Así que tendremos que buscar la manera de hacerlo usando algún bloque del cajón del objeto **Lienzo**. Otra pista: es un bloque de color verde oscuro.

Es el bloque **poner.Lienzo1.ColorDePintura.como**. Lo arrastraremos hasta encajarlo dentro del bloque mostaza **cuando.boton_rojo.Clic**.

Falta indicar que queremos que sea el color rojo. Para ello escogeremos el pequeño bloque que identifica a este color de entre los colores que hay en el cajón **Colores** de la Paleta de App Inventor.

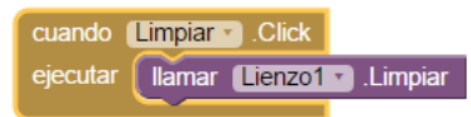


Haremos lo mismo con los botones para color verde y color azul. Lo mejor es hacerlo duplicando el bloque mostaza que hemos creado para el color rojo. Una vez hecho, en el nuevo bloque mostaza, podemos desplegar la lista del nombre de los botones y elegir el del color verde. Entonces sólo tendremos que cambiar el bloque rojo por el verde, y listo. Lo mismo con el botón para el azul.



Cuando probemos el resultado se nos “ensuciará” el lienzo enseguida, así que hay que poner un botón para dejar el lienzo en blanco de nuevo. ¿Cómo se hará? Varias pistas:

- ❖ Crear **DisposiciónHorizontal1** debajo del lienzo.
- ❖ Meter dentro un nuevo botón y llamarlo **Limpiar**.
- ❖ Cambiar el texto del botón por “Limpiar”.
- ❖ Poner los bloques para que al hacer clic sobre **Limpiar** se limpie el lienzo.





En realidad, sería más interesante si en lugar de hacer circulitos, pudieran dibujarse líneas al arrastrar el dedo por la pantalla. **¿Qué bloque mostaza del objeto Lienzo habría que usar para que se dibujara según fuéramos arrastrando el dedo por el lienzo?** Una pista: deberemos utilizar un bloque naranja del componente **Lienzo1**.

ATENCIÓN

En programación no hay una única manera de hacer bien las cosas, es decir, podemos conseguir el mismo resultado utilizando bloques diferentes. El objetivo crear el programa de la forma más simple y más eficiente, para que nuestro código sea más “elegante”.

Podemos usar estos dos tipos de bloques, por ejemplo, para dibujar un círculo por cada punto donde vayamos arrastrando el dedo. **XActual** e **YActual** se refieren precisamente al punto representado por las coordenadas x e y actuales. Veremos cómo funciona esto al arrastrar el dedo despacio por el lienzo. Si deslizamos el dedo demasiado rápido se verán los puntos separados.



Más apropiado puede ser entonces utilizar este otro bloque morado, que se basa en dibujar líneas, y no círculos. Para dibujar una línea hace falta especificar el punto inicial de la línea y el punto final. Por ello para usar este bloque hay que indicarle dos coordenadas x y dos y. El primer (x1, y1) define el punto inicial de la línea, y el segundo par (x2, y2) define el punto final de la misma. Al hacer clic sobre el lienzo, sin levantar el dedo, el programa guardará en su memoria el valor de la x e y del punto que hemos tocado (determinado por XPrevio e YPrevio). Según vayamos deslizando el dedo el programa irá detectando el cambio de posición y dibujará una línea desde ese punto guardado hasta el punto de que estamos tocando actualmente (determinado por XActual e YActual). Esto se repetirá todo el tiempo, muy rápidamente, mientras sigamos arrastrando el dedo por la pantalla. Se dibujarán por lo tanto muchas líneas muy cortas, una detrás de la otra, dando la sensación de ser una única línea continua.





Como hemos decidido utilizar el bloque

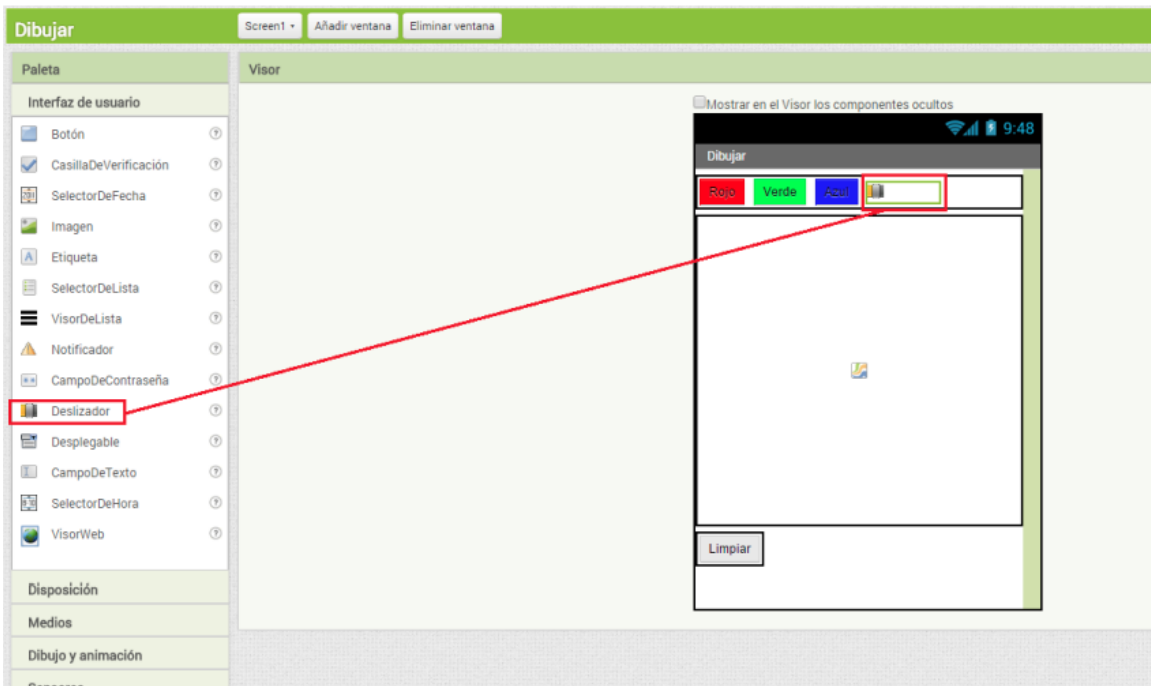
cuando.Lienzo1.Arraigado

podemos eliminar el bloque naranja **cuando.Lienzo1.Tocar** que había utilizado

anteriormente, y que dibujaba círculos, a no ser que queramos mantener ambos botones, para que nuestra aplicación haga dos cosas distintas dependiendo de la acción que tomemos: si sólo tocamos el lienzo se dibujará un círculo, y si tocamos el lienzo y deslizamos el dedo se dibujará una línea.

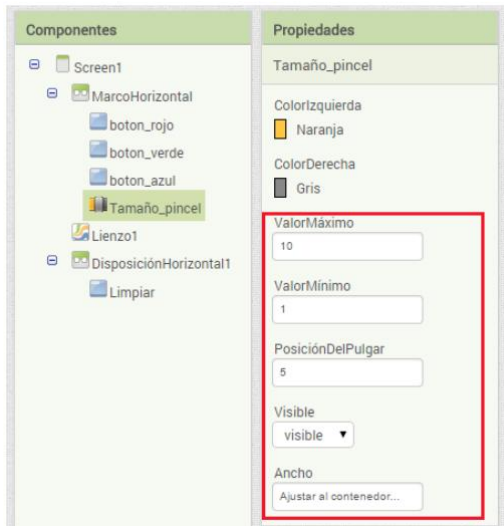
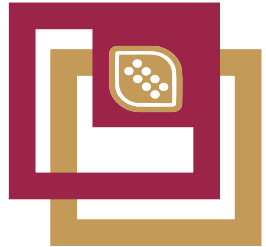
Cambiar el tamaño del pincel con un Deslizador

Un objeto **Deslizador** es perfecto para modificar el valor numérico de un parámetro sin tener que escribirlo. Para incluir un deslizador en nuestro interfaz de usuario tenemos que arrastrarlo desde la Paleta. En nuestro caso lo llamaremos **Tamaño_pincel**.



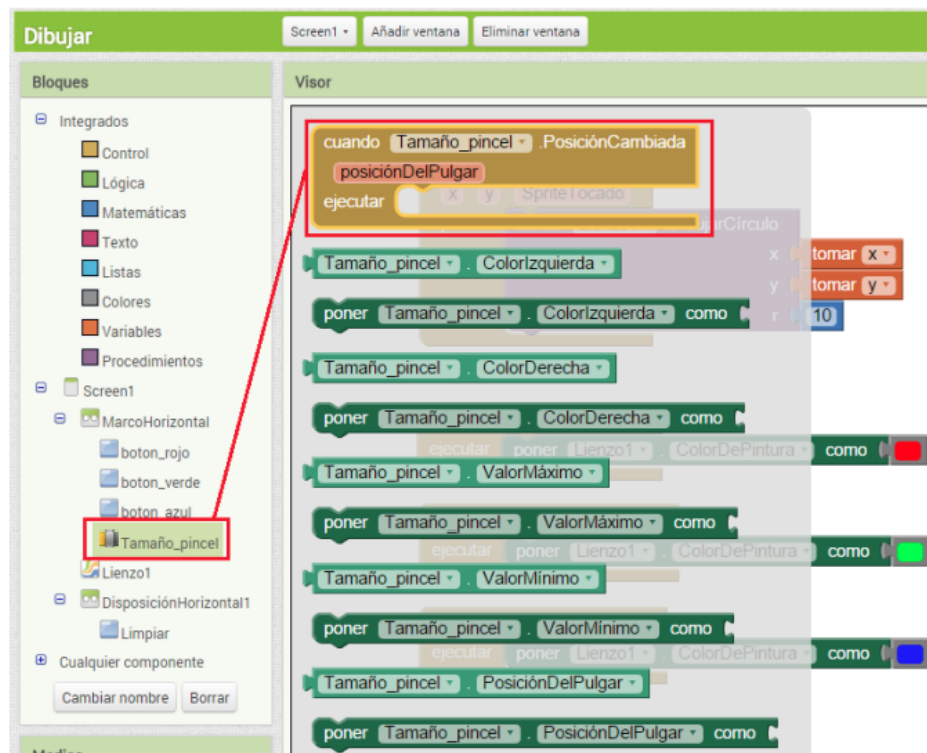
Tenemos que definir el valor mínimo y máximo que podrá tomar el deslizador, y el valor que tendrá inicialmente (**PosiciónDelPulgar**). También es conveniente especificar la anchura del **Slider** para que se ajuste al espacio restante (**Ajustar al contenedor**).



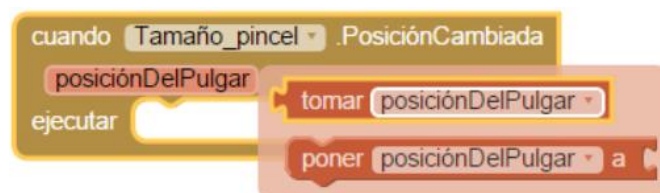


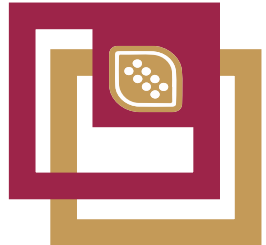
Tenemos que indicarle ahora a nuestro programa que modifique el ancho de la línea de nuestro dibujo según variemos la posición de **Tamaño_pincel**. Para ello, utilizaremos el bloque mostaza **cuando.Tamaño_pincel.PosiciónCambiada**, que se ejecuta cada vez que se cambia la posición del selector dentro del deslizador.

Este bloque incluye una variable, **posiciónDelPulgar**, que contiene el valor actual de la posición, y que puede utilizarse para nuestro propósito.

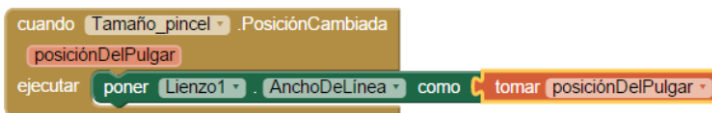


Si dejamos momentáneamente el puntero del ratón sobre la casilla **posiciónDelPulgar** del bloque mostaza, se mostrará un desplegable que nos permitirá elegir el bloque que define el valor de esa variable (**tomar**), o definir un nuevo valor para ella (**poner**).



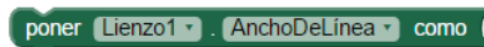


Elegiremos en este caso el bloque **tomar**, y lo combinaremos con el bloque verde del objeto **Lienzo** que define el ancho de línea de dibujo, para indicar al programa que cambie el ancho de la línea cada vez que arrastremos el puntero del deslizador **Tamaño_pincel** a una nueva posición.

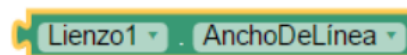


Igual que ocurre con los bloques de color naranja **tomar** y **poner** que aparecen dentro de algunos bloques mostaza, y que sirven para poner o tomar un valor (en este caso para **posiciónDelPulgar**), en muchos componentes existen bloques de color verde, que nos permiten tomar o poner valores a sus propiedades.

Por ejemplo, el siguiente bloque nos permite **poner**, asignar, definir el valor del ancho de línea del componente **Lienzo1**. La propiedad **AnchoDeLínea** de **Lienzo1** tendrá entonces el valor del bloque que encajemos a la derecha del bloque verde oscuro.



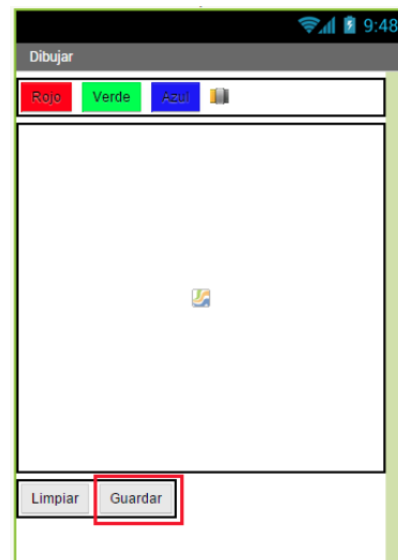
Y este otro sirve para **tomar** el valor actual del ancho de línea del componente **Lienzo1**. Es decir, cuando encajemos este bloque verde claro a la derecha de un bloque verde oscuro, éste tomará el valor de la propiedad **AnchoDeLínea** de **Lienzo1**.

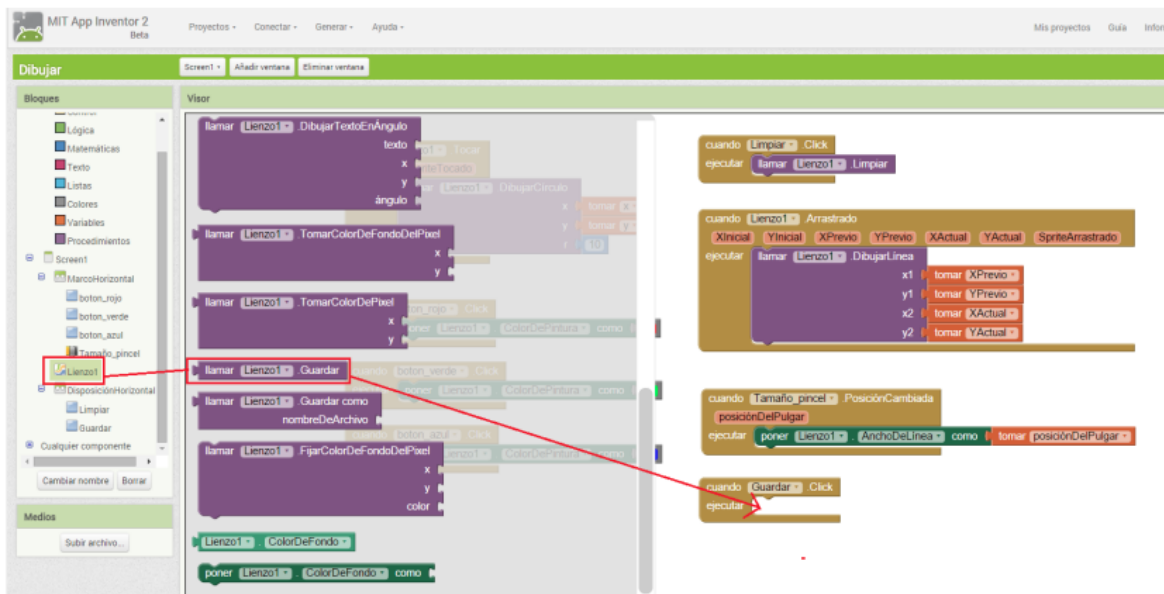
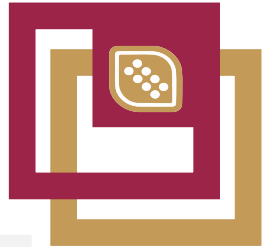


Guardar un archivo con el dibujo que hemos hecho

Cuando el usuario termine de hacer un dibujo interesante, seguramente querrá conservarlo en la memoria permanente del teléfono. Para ofrecerle esa posibilidad, tendremos que indicarle a nuestro programa cómo almacenar en la memoria el contenido actual del lienzo. En primer lugar, hay que incluir en el interfaz un botón **Guardar**, que al ser pulsado ejecutará el código correspondiente.

El bloque de código que usaremos, incluido dentro del cajón Lienzo, será **llamar.Lienzo1.Guardar**.

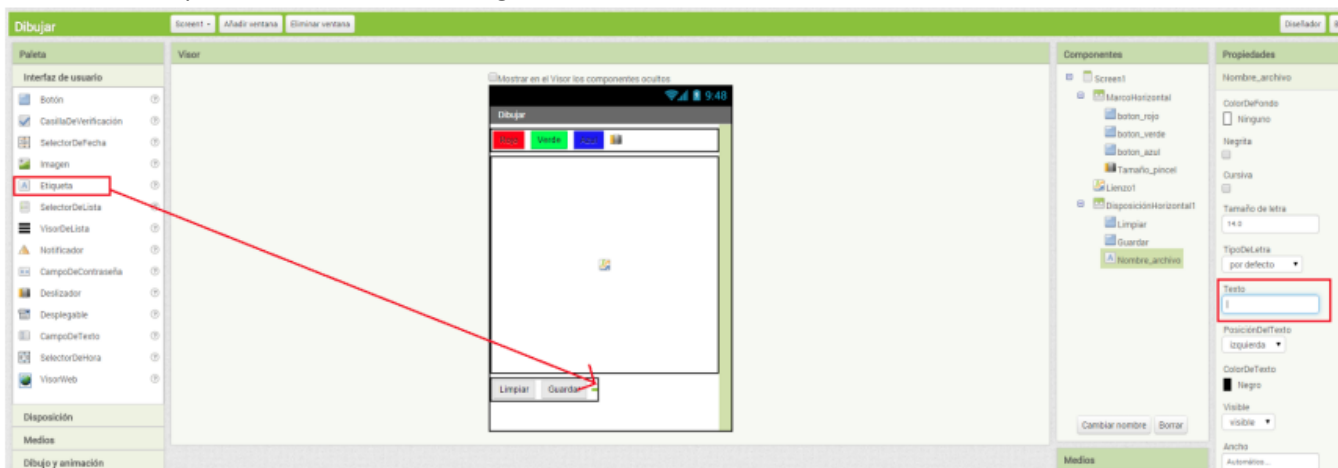




Este bloque es diferente a los vistos hasta ahora, porque no puede ser encajado directamente debajo de otro bloque, o dentro de un bloque color mostaza. El bloque tiene que ser encajado a otro por su izquierda. Esto se debe a que es un bloque que devuelve un valor, como acabamos de ver al final del apartado anterior, con el bloque de color verde claro **Lienzo1.AnchoDeLínea**. En este caso el bloque violeta devuelve un texto, que contiene el nombre del archivo donde se ha almacenado el dibujo dentro de la memoria del dispositivo.



Es conveniente colocar un componente Etiqueta, que llamaremos **Nombre_Archivo**, a la derecha del botón **Guardar**, y que tomará el valor de texto devuelto por el procedimiento **Lienzo1.Guardar**. Definiremos que su propiedad **Texto** esté inicialmente en blanco. El componente será difícil de ver en el dispositivo mientras no le asignemos un nuevo valor.



Veamos cuidadosamente el nuevo componente y los bloques utilizados, hasta interiorizar mentalmente su funcionamiento.



Cada vez que se pulse el botón **Guardar**, el dibujo quedará almacenado con un nombre que se asignará automáticamente, en función del día y la hora. Este nombre aparecerá en el texto de la etiqueta **Nombre_archivo** cada vez que hagamos clic en el botón **Guardar**.

Ideas para mejorar la aplicación

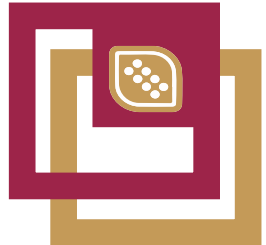
Podemos mejorar esta aplicación hasta donde queramos, usando la imaginación, e investigando cómo podemos utilizar los recursos en App Inventor para incluir en nuestra aplicación todo lo que vayamos inventando.

Por ejemplo...

- ❖ Poner un sello con tu nombre cuando pulses un botón.
- ❖ Tomar una foto existente en la memoria del dispositivo y usarla como fondo del lienzo, etc.

Recuerda conectar la Aplicación Web Progresiva con la aplicación móvil utilizando el código en el menú **CONECTAR**.





Lectura No 4. Programación de Juegos

Instrucciones: Realiza el análisis de la siguiente información, posteriormente, realiza la Actividad 4.

Timeline: Evolución de los Videojuegos.

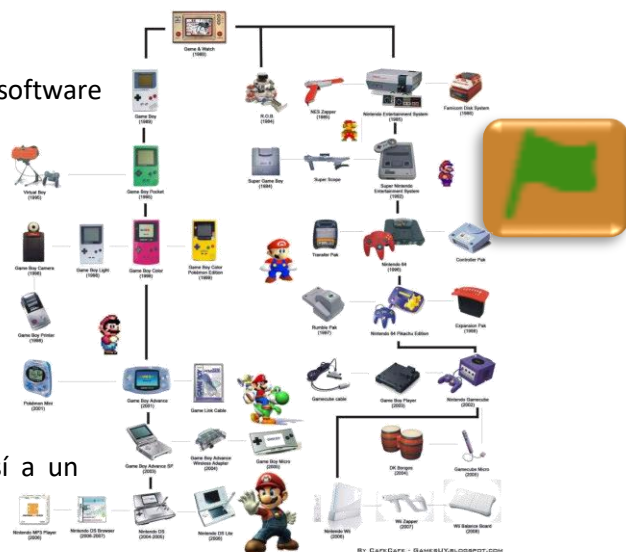


LECTURA 4. "PROGRAMACIÓN DE JUEGOS"



El desarrollo de videojuegos comenzó en 1947, cuando la idea de un videojuego fue concebida y patentada por Thomas T. Goldsmith Jr y Estle Ray Mann, los cuales llenaron una aplicación de patente en Estados Unidos el 25 de enero de 1947. El éxito de los videojuegos se ha extendido hasta hoy día y posee un futuro prometedor para beneficio de la Industria de los videojuegos y de los usuarios finales.

Un videojuego (del inglés video game) o juego de video es un software informático creado para el entretenimiento en general y basado en la interacción entre una o varias personas y un aparato electrónico que ejecuta dicho videojuego; este dispositivo electrónico puede ser una computadora, un sistema arcade, una videoconsola, un dispositivo handheld o un teléfono móvil, los cuales son conocidos como "plataformas". Aunque, usualmente el término "video" en la palabra "videojuego" se refiere en sí a un visualizador de gráficos rasterizados, hoy en día se utiliza para mencionar cualquier tipo de visualizador.



Cada vez que juegas a tu videojuego preferido, dentro del ordenador están ocurriendo muchas cosas. Se hace patente que un factor importante en un juego es que se mueva con soltura y a buena velocidad. Todos los elementos del juego parecen funcionar de forma independiente, como con vida propia y a la vez. Sin embargo, esto es sólo una apariencia, ya que dentro del programa se van sucediendo las diferentes fases de ejecución de forma secuencial y ordenada.



La alta calidad técnica de los videojuegos actuales ha creado un mercado cada vez más exigente a nivel estético, funcional y narrativo.

La Experiencia de juego

Como se mencionó en lecturas anteriores la programación es un elemento clave para que podamos utilizar aplicaciones y

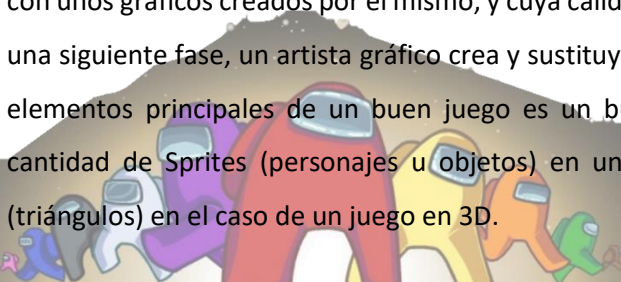
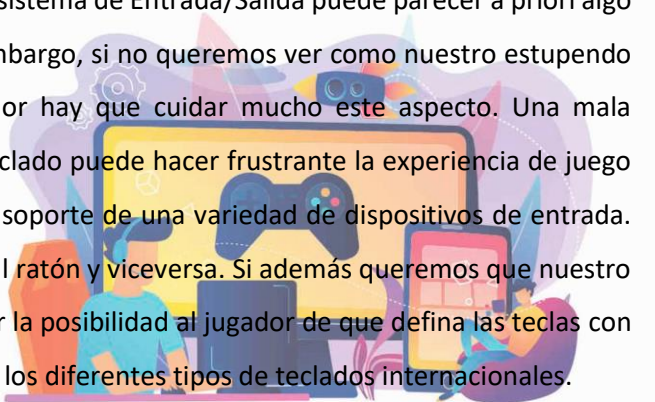
programas sencillos o sofisticados tanto para entretenimiento como para desarrollo. A continuación, te daremos una visión general y sin entrar en detalles en la implementación de la anatomía de un videojuego. Vamos a ver qué partes lo componen y como se relacionan:

Entrada.

Un videojuego necesita comunicarse con el jugador a través de un dispositivo de entrada, como un Joystick o un teclado. La programación del sistema de Entrada/Salida puede parecer a priori algo fácil de implementar, y de hecho lo es. Sin embargo, si no queremos ver como nuestro estupendo juego se hace poco “jugable” para el jugador hay que cuidar mucho este aspecto. Una mala respuesta del juego ante una pulsación del teclado puede hacer frustrante la experiencia de juego para el jugador. Otro factor importante es el soporte de una variedad de dispositivos de entrada. Algunos jugadores preferirán usar el teclado al ratón y viceversa. Si además queremos que nuestro juego tenga un toque profesional, hay que dar la posibilidad al jugador de que defina las teclas con las que quiere jugar, además de dar soporte a los diferentes tipos de teclados internacionales.

Visualización.

La misión de esta capa es la de convertir el estado interno del juego en una salida gráfica. Ya sea en 2D o 3D, la calidad gráfica es un factor importantísimo. Normalmente, el programador trabaja con unos gráficos creados por él mismo, y cuya calidad dependerá de los dotes artísticos de éste. En una siguiente fase, un artista gráfico crea y sustituye estas imágenes por las definitivas. Uno de los elementos principales de un buen juego es un buen motor gráfico, capaz de mover una gran cantidad de Sprites (personajes u objetos) en un juego 2D o una gran cantidad de polígonos (triángulos) en el caso de un juego en 3D.



En el diseño del motor gráfico, siempre hay que llegar a un compromiso entre la compatibilidad y el rendimiento. Hay tarjetas gráficas en el mercado con unas capacidades casi sobrenaturales, pero que no son estándar, sin embargo, tampoco podemos dar la espalda a los jugadores “profesionales” que equipan sus ordenadores con la última aceleradora 3D del mercado.



Sonido.

El sonido es un aspecto al que, a veces, no se le da la importancia que merece. Tan importante como unos buenos gráficos es un buen sonido. Tanto el sonido como la banda sonora de un juego son

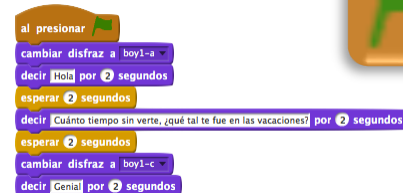


capaces de transmitir una gran cantidad de sensaciones al jugador, y eso es, al fin y al cabo, lo que se busca. Tanto como la calidad, la buena sincronización con lo que va sucediendo en pantalla es de suma importancia.

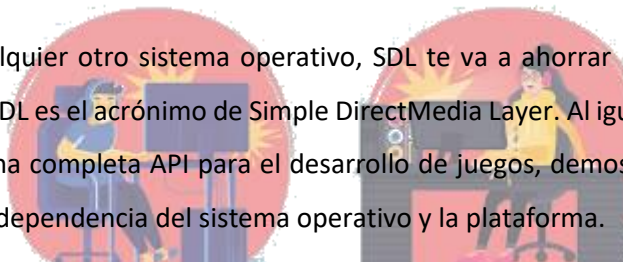


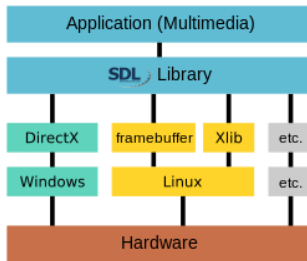
Comunicaciones.

Cada vez más, los juegos en red van ganando en adeptos. La comunicación se realiza normalmente sobre redes TCP/IP. Esto nos permite jugar contra otros jugadores situados en nuestra propia red local o en Internet. La comunicación entre máquinas se lleva a cabo mediante la programación de sockets. Afortunadamente SDL nos provee de herramientas muy potentes para este tipo de comunicación, aunque esto es algo que queda fuera del alcance de este libro.

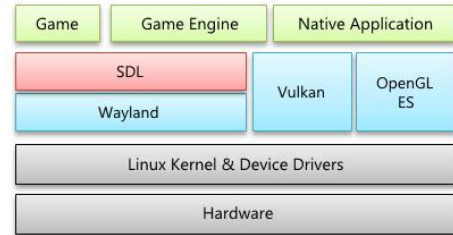


Tanto si utilizas Windows, Linux o cualquier otro sistema operativo, SDL te va a ahorrar un montón de trabajo y de dolores de cabeza. SDL es el acrónimo de Simple DirectMedia Layer. Al igual que DirectX en Windows, SDL nos ofrece una completa API para el desarrollo de juegos, demos y todo tipo de aplicaciones multimedia con independencia del sistema operativo y la plataforma.

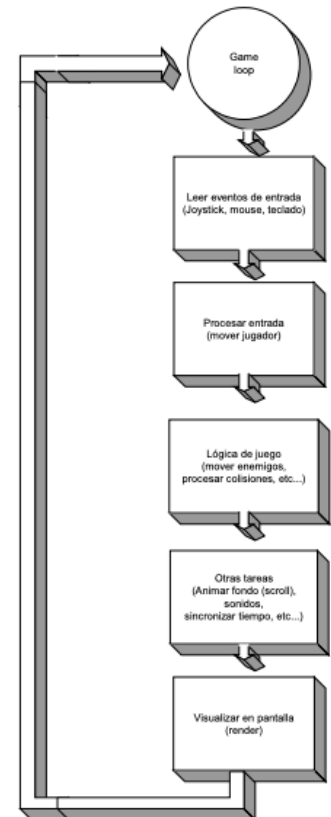
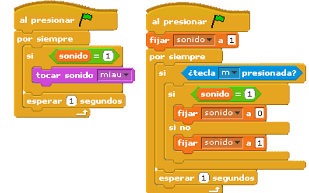




Actualmente SDL soporta Linux, Windows, BeOS, MacOS, MacOS X, FreeBSD, OpenBSD, BSD/OS, Solaris, y IRIX. Aunque tiene portes no oficiales a Windows CE, AmigaOS, Atari, QNX, NetBSD, AIX, OSF/Tru64, y SymbianOS. SDL está compuesto por subsistemas separados que nos ofrecen soporte a diferentes partes del hardware.



El “game loop” o bucle de juego es el encargado de “dirigir” en cada momento que tarea se está realizando. En la figura podemos ver un ejemplo de game loop. Es sólo un ejemplo, y aunque más o menos todos son similares, no tienen por qué tener exactamente la misma estructura. Analicemos el ejemplo. Lo primero que hacemos es leer los dispositivos de entrada para ver si el jugador ha realizado alguna acción. Si hubo alguna acción por parte del jugador, el siguiente paso es procesarla, esto es, actualizar su posición, disparar, etc... dependiendo de que acción sea. En el siguiente paso realizamos la lógica de juego, es decir, todo aquello que forma parte de la acción del juego y que no queda bajo control del jugador, por ejemplo, el movimiento de los enemigos, cálculo de trayectoria de sus disparos, comprobamos colisiones entre la nave enemiga y la del jugador, etc... Fuera de la lógica del juego quedan otras tareas que realizamos en la siguiente fase, como son actualizar el scroll de fondo (si lo hubiera), activar sonidos (si fuera necesario), realizar trabajos de sincronización, etc. Ya por último nos resta volcar todo a la pantalla y mostrar el siguiente frame. Esta fase es llamada “fase de render”.



Normalmente, el game loop tendrá un aspecto similar a lo siguiente:

```
int done = 0;
while (!done) {
    // Leer entrada
    // Procesar entrada (si se pulsó ESC, done = 1)
    // Lógica de juego
    // Otras tareas
    // Mostrar frame
}
```



Actividad 4. Timeline: Evolución de los Videojuegos



Instrucciones: Realiza una investigación para desarrollar una línea de tiempo, donde muestres la evolución de los videojuegos, considerando el software de programación, el hardware utilizado, las tecnologías utilizadas; considera lo que lograste comprender de la lectura 4.



Considera los siguientes criterios para tu línea de tiempo:





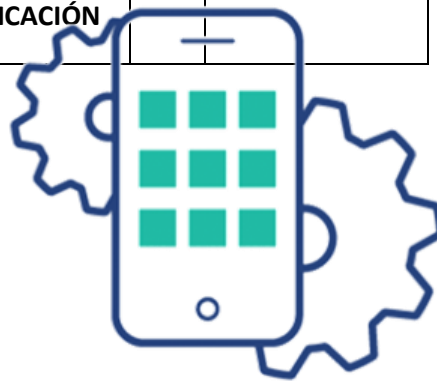
INSTRUMENTO DE EVALUACIÓN
LISTA DE COTEJO 5
Actividad 4: Timeline: Evolución de los Videojuegos



DATOS GENERALES

Nombre(s) del alumno(s)	Matrícula(s)
Producto: Línea de tiempo	Fecha
Submódulo 1: Aplicación Móvil y Juegos	Periodo: 2025 – 2026B
Nombre del docente	Firma del docente

CRITERIO	INDICADORES	VALOR OBTENIDO		PONDERACIÓN	CALIF	OBSERVACIONES Y/O SUGERENCIAS DE MEJORA
		SÍ	NO			
1	Demuestra organización en los elementos presentados.			2		
2	Presenta ilustraciones acordes a la información que proporciona.			2		
3	Se aprecia el conocimiento y dominio del tema.			2		
4	Muestra dominio en el uso de las tecnologías de la Información y de la comunión.			2		
5	No presenta faltas ortográficas.			1		
6	Entrega en tiempo y forma.			1		
	CALIFICACIÓN					



Diseño de Videojuegos



Muchas veces nos hemos preguntado como desarrollaremos un videojuego, o que pasos debemos seguir para desarrollarlo, esta propuesta pretende de una forma muy flexible explicarte algunas de las partes más importantes a tener en cuenta en el desarrollo de un videojuego.



El desarrollo de un videojuego generalmente sigue el siguiente proceso:

Primero es fundamental tener una concepción de la idea del juego después deberán realizarse los siguientes procesos.



El proceso es similar a la creación de software en general, aunque difiere en la gran cantidad de aportes creativos (música, historia, diseño de personajes, niveles, etc) necesarios. El desarrollo también varía en función de la plataforma objetivo (PC, móviles, consolas), el género (estrategia en tiempo real, rpg, aventura gráfica, plataformas, etc) y la forma de visualización (2d, 2.5d y 3d).

CONCEPCIÓN

En esta etapa es necesario definir los aspectos fundamentales que conformarán el videojuego, entre los que se encuentran:

- 🎮 **Género:** Dentro de que género o géneros se va a desarrollar el juego. De no corresponder a un género conocido, se deben especificar las características
- 🎮 **Game play:** Lo que generará diversión a la hora de jugarlo.
- 🎮 **Story board:** Algunas ideas sueltas acerca de cómo debe lucir el juego en cuanto a personajes, ambientación, música, etc.



DISEÑO

En esta fase se detallan todos los elementos que compondrán el juego, dando una idea clara a todos los miembros del grupo desarrollador acerca de cómo son. Entre estos elementos tenemos:

Historia

Forma en que se desenvolverán los personajes del juego y la historia del mundo representado avanza. No todos los juegos tienen historia.



Arte Conceptual



entre las
Algunas veces los
producción, pero lo usual es

Se establece el aspecto general del juego. En esta etapa un grupo de artistas se encargan de visualizar o conceptualizar personajes, escenarios, criaturas, objetos, etc. Estos artistas se basan en las ideas originales de los creadores y luego entregan una serie de propuestas impresas o digitales de cómo lucirá el juego.

Posteriormente el director de arte se encargará de escoger de opciones aquellas que se apeguen más a la idea original. artistas conceptuales permanecen durante todo el proceso de que sólo participen en las primeras etapas del proceso.

Sonido

Detallada descripción de todos los elementos sonoros que el juego necesita para su realización. Voces, sonidos ambientales, efectos sonoros y música.



Mecánica de juego

Es la especificación del funcionamiento general del juego. Es dependiente del género y señala la forma en que los diferentes entes virtuales interactuarán dentro del juego, es decir, las reglas que rigen éste.

Diseño de programación

Describe la manera en que el videojuego será implementado en una máquina real (un PC, consola, teléfono móvil, etc) mediante un cierto lenguaje de programación y siguiendo una determinada metodología. Generalmente en esta fase se generan diagramas de UML que describen el funcionamiento estático y dinámico, la interacción con los usuarios y los diferentes estados que atravesará el videojuego como software.

De toda la fase de diseño es necesario generar un documento llamado Documento de Diseño,

que contiene todas las especificaciones de arte, mecánicas y programación.

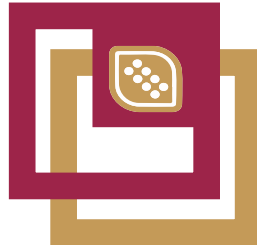


PLANIFICACIÓN

En esta fase se identifican las tareas necesarias para la ejecución del videojuego y se reparten entre los distintos componentes del equipo desarrollador.

También se fijan plazos para la ejecución de dichas tareas y reuniones clave, con la ayuda de herramientas de diagramación de actividades como GANTT y PERT.



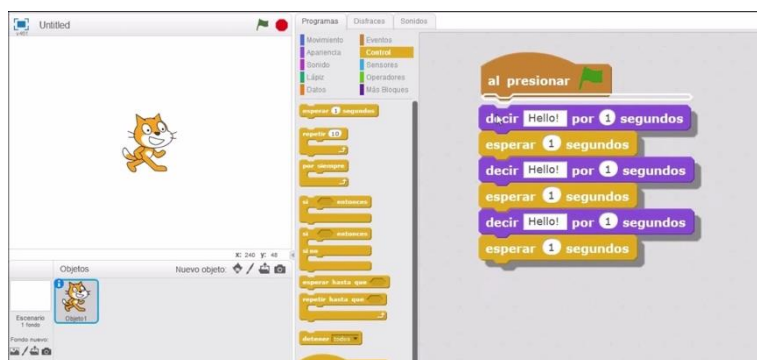


PRODUCCIÓN

Aquí se llevan a cabo todas las tareas especificadas en la fase de planificación, teniendo como guía fundamental el documento de diseño. Esto incluye entre otras cosas la codificación del programa, la creación de sprites, tiles y modelos 3D, la grabación de sonidos, voces y música, la creación de herramientas para acelerar el proceso de desarrollo, entre otras.

Programación

La mayoría de los juegos se programan utilizando el lenguaje C++ dado que es un lenguaje de alto nivel que permite un rápido acceso a los componentes de hardware de una computadora o consola de juegos que lo hace más accesible.



Ilustración

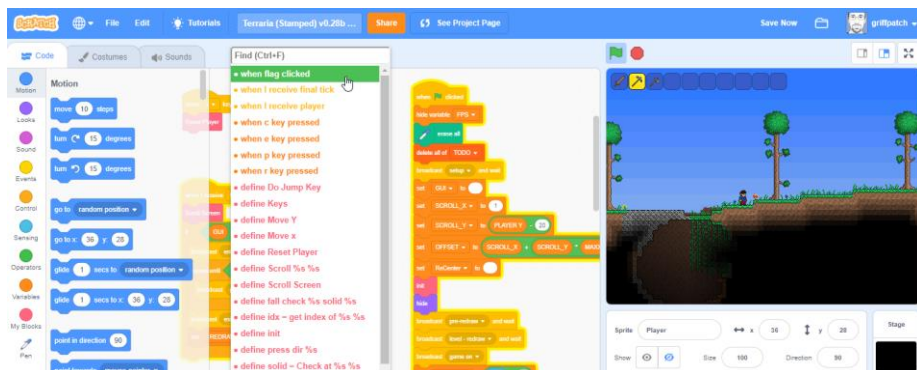
Los juegos 2D deben ser ilustrados por artistas experimentados, quienes trabajan tomando en consideración las limitaciones técnicas del hardware sobre el cual correrá el juego, esto incluye: Cantidad de colores disponibles, tamaño de los sprites, resolución final de los sprites y formatos a utilizar. Los artistas 2D también son los encargados de elaborar las animaciones del juego.

Se muestra el proceso de diseño 2D e ilustración de un personaje para un videojuego.



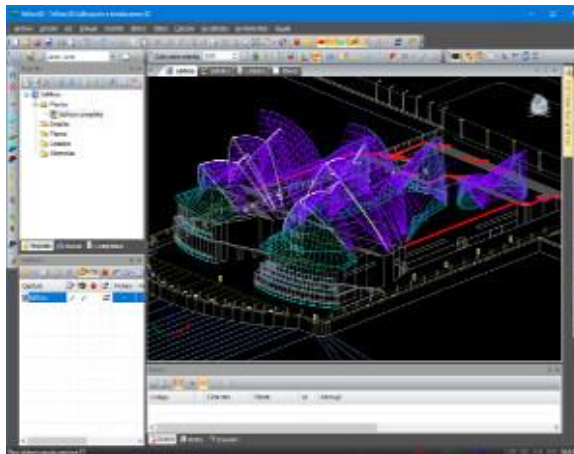
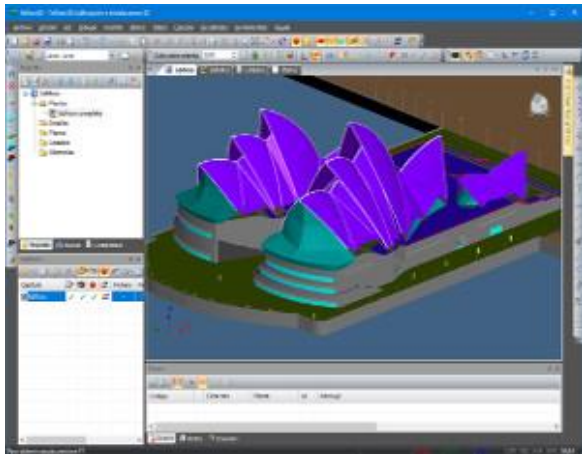
Interfaz

Es la forma en que se verán los elementos GUI y HUD, mediante los cuales el usuario interactuará con el juego. La Interfaz no es un elemento rígido dentro, ni para ningún juego, esta cambia de un juego a otro según su temática, la misma se crea siempre de una forma tal que el espectador pueda identificarse rápidamente con ella.



Modelado y animación 3D

Los artistas utilizan herramientas comerciales de modelado y animación tridimensional como Autodesk 3D Studio Max, Autodesk Maya, Autodesk Softimage XSI, Blender, etc. Pero además usan herramientas desarrolladas internamente que facilitan algunas de las funciones más comunes del proceso de creación de juegos.



Diseño de sonido

Los ingenieros de sonido se encargan de crear sonidos para cada objeto o personaje del juego. Pueden crear sonidos desde cero o utilizar sonidos del ambiente y modificarlos según sus necesidades.



PRUEBAS

Al igual que el software convencional, los videojuegos deben pasar por una etapa donde se corrigen los errores inherentes al proceso de programación y a diferencia de este, los videojuegos requieren un refinamiento de su característica fundamental, la de producir diversión de manera interactiva (jugabilidad). Generalmente esta etapa se lleva a cabo en dos fases:

- 🎮 **Pruebas Alpha:** Se llevan a cabo por un pequeño grupo de personas, que con anterioridad estén involucradas en el desarrollo, lo que puede incluir artistas, programadores, coordinadores, etc. El propósito es corregir los defectos más graves y mejorar características de jugabilidad no contempladas en el documento de diseño.

- 🎮 **Pruebas Beta:** Estas pruebas se llevan a cabo por un equipo externo de jugadores, bien sea que sean contratados para la ocasión o que sean un grupo componente del proyecto (grupo QA). De estas pruebas el videojuego debe salir con la menor cantidad posible de defectos menores y ningún defecto medio o crítico.



BIBLIOGRAFÍA

EcuRed. (2013). *Desarrollo de Videojuegos* - EcuRed. Desarrollo de Juegos.
https://www.ecured.cu/Desarrollo_de_Videojuegos

How to make a Video Game - Getting Started. (2017, 22 enero). [Vídeo]. YouTube.
<https://www.youtube.com/watch?v=j48LtUkZRjU&list=PLPV2KyIb3jR53Jce9hP7G5xC4O9AgnOuL>

Serrano, A. G. (2014). *Programacion de videojuegos con sdl en windows y linux* [Libro electrónico].
Ediversitas Multimedia, S.L.



Actividad 5. Creación de un Videojuego



Instrucciones: Realizar una infografía, donde se presentas el proceso de desarrollo de un videojuego que lograste comprender de la lectura. Puedes hacer uso de aplicaciones o de manera física:



Comienza con esto



<https://youtu.be/GXg6DORVZ3c>

1

Después de haber realizado la lectura anterior y de haber visto el video que se encuentra en el código QR de esta actividad, debes elaborar una infografía en la que representes las distintas etapas que se deben seguir para el desarrollo de un videojuego. Recuerda utilizar imágenes que permitan identificar con mayor facilidad cada uno de los procesos.

2

Una vez concluida la infografía debes anexar una pequeña reflexión a modo de conclusión en la que menciones la importancia del tema en relación con los temas de estudio de la materia.

3

No olvides utilizar el instrumento de evaluación correspondiente para la elaboración de la actividad.





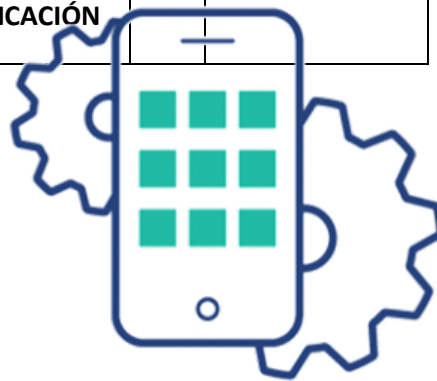
INSTRUMENTO DE EVALUACIÓN
LISTA DE COTEJO 6
Actividad 5: Creación de un Videojuego

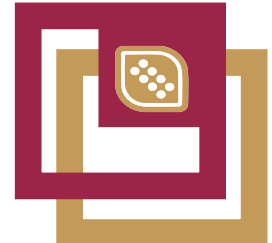


DATOS GENERALES

Nombre(s) del alumno(s)	Matrícula(s)
Producto: Infografía	Fecha
Submódulo 1: Aplicación Móvil y Juegos	Periodo: 2025 – 2026B
Nombre del docente	Firma del docente

CRITERIO	INDICADORES	VALOR OBTENIDO		PONDERACIÓN	CALIF	OBSERVACIONES Y/O SUGERENCIAS DE MEJORA
		SÍ	NO			
1	Presenta las 5 etapas en el desarrollado de videojuegos.			2		
2	Presenta ilustraciones acordes a la información que proporciona.			2		
3	Se aprecia el conocimiento y dominio del tema.			2		
4	Muestra dominio en el uso de las tecnologías de la Información y de la comunión.			2		
5	No presenta faltas ortográficas.			1		
6	Entrega en tiempo y forma.			1		
	CALIFICACIÓN					





Desarrollo de Videojuegos para Web



Detrás de cada videojuego está el código que lo ejecuta. Los **lenguajes para programar videojuegos** son una parte integral del desarrollo de este tipo de software. Los distintos códigos que se emplean para el diseño y desarrollo de videojuegos son responsables de cómo experimentamos e interactuamos con ellos.



Un código bien escrito puede proporcionar una **experiencia de juego fluida**. Conocer bien qué lenguajes se usan para programar videojuegos es el primer paso para **especializarte en su creación**.



Programando videojuegos: conceptos básicos

Un **lenguaje de programación de videojuegos**, secuencias de comandos, o código, es el sistema que emplean los programadores de videojuegos para dar forma a la configuración y a las mecánicas de un juego. Es decir, es la maquinaria que hace que un juego funcione de una manera determinada.

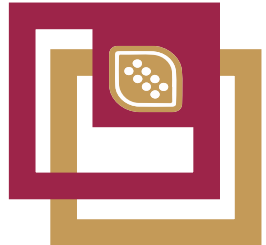


En este sentido, cada lenguaje de programación que se emplea en videojuegos tiene unos objetivos concretos y se usa por un motivo. Y es que, **los distintos códigos tienen una sintaxis diferente** y son más eficientes para realizar ciertas tareas.



No obstante, **muchos lenguajes de programación se superponen y son compatibles entre sí**, cada uno de los cuales contribuye a la experiencia general del juego. Por ese motivo no es raro, que los programadores, en el desarrollo de determinados juegos lleguen a emplear varios de los lenguajes que te vamos a mencionar en los siguientes apartados.





La Web rápidamente se ha convertido en una plataforma viable no solo para crear impresionantes juegos de alta calidad, sino también para distribuirlos.

El rango de juegos que pueden ser creados está a la par tanto de los juegos de escritorio como de SO nativos (Android, iOS). Con tecnologías Web modernas y un navegador reciente es totalmente posible hacer juegos de primera categoría para la Web. Y no estamos hablando sobre simples juegos de cartas



o juegos sociales multijugadores que en tiempos anteriores se podían hacer con Flash®. Estamos



hablando sobre juegos 3D shooters de acción, RPGs, y más. Gracias a las masivas mejoras de rendimiento en JavaScript con tecnología de compilación just-in-time y nuevas APIs, se pueden construir juegos que pueden correr en el navegador (o en dispositivos HTML5 como Firefox OS) sin problemas.

Ventajas

Como un desarrollador de juegos, de forma individual o como un gran estudio de juegos, es importante apuntar a la Web como tu próximo proyecto de juego.



- 🎮 El alcance de la Web es enorme; está en todas partes. Los juegos construidos con HTML5 funcionan en smartphones, tablets, PCs y Smart TVs.
- 🎮 Se mejoran la comercialización y la detección. No estás limitado a promocionar tu aplicación en la tienda de aplicaciones de otro. En su lugar, puedes anunciar y promover tu juego en toda la Web, así como en otros medios, aprovechando la inherente capacidad de enlace y capacidad de compartir de la Web para llegar a nuevos clientes.



- 🎮 Tienes control en lo más importante: Pagos. Usted no tiene que entregar más del 30% de sus ingresos a otros sólo porque su juego está en su ecosistema. En su lugar, cargue lo que quiera y utilice cualquier servicio de procesamiento de pagos que le guste.



El mercado de Java



Una vez más con más control, puedes actualizar tu juego cuando quieras. No hay que esperar sin aliento para la aprobación, mientras que alguien oculto dentro de otra empresa decide si tu solución de error crítico se enviará hoy o mañana.

¡Controla tus análisis! En lugar de confiar en otra persona para tomar todas las decisiones sobre qué



analítica necesita, puede recoger su propia o elegir la tercera parte que más le guste para recopilar información sobre sus ventas y el alcance de su juego.

- Consigues manejar tu relación con el cliente más de cerca. No es necesario tener comentarios de clientes filtrados a través de los mecanismos limitados de una tienda de aplicaciones. Involucrarse con sus clientes de la manera que desee, sin un intermediario.
- Tus jugadores pueden jugar su juego en cualquier lugar, en cualquier momento. Debido a que la Web es omnipresente, tus clientes pueden comprobar el estado de tu juego en sus teléfonos, tabletas, computadoras portátiles domésticas, sus escritorios de trabajo o cualquier otro dispositivo.

La forma en que se codifica un juego depende de muchos factores, como, por ejemplo, del tipo de proyecto que se esté desarrollando. No es lo mismo hacer un juego para consolas que para ordenador.



También, al contrario, un proyecto puede ser muy complejo y necesitar de **varios lenguajes de programación** para que todas las capas del juego funcionen como está previsto en el documento de diseño.



Como ves, los factores son variables y, al final, depende de qué tipo de juego, para que plataforma y cuán complejo quieras que sea tu videojuego. Vamos a ver ahora, en detalle, algunos de los **lenguajes para programar videojuegos más empleados en la actualidad**.



Funcionalidad de Videojuegos en el Móvil



Hoy en día el mercado de los videojuegos es un mercado inmenso que está en constante crecimiento, esto es gracias a los teléfonos móviles, ya que todo el mundo tiene uno y, cualquier móvil, sea de la marca que sea viene con una tienda de aplicaciones desde donde se pueden descargar miles de juegos. Muchos de estos juegos han sido creados por individuos que han tenido que investigar mucho para poder publicar un juego.



PRIMER PASO

Lo primero que hay que hacer es elegir con qué herramienta se hará, ya que dependiendo de la herramienta se utilizará un lenguaje de programación u otro. Puedes escoger entre utilizar un motor de videojuegos que te permite publicar el juego en diferentes plataformas o directamente elegir el framework necesario para la plataforma donde quieras publicarlo. La principal diferencia es que el motor de videojuegos te hace la tarea de creación más fácil, ya que llevan integrados un motor de físicas, detección de colisiones y muchas otras funcionalidades.



Si se elige el motor de videojuegos, la tarea será más difícil, ya que hay unos cuantos: Unity3D, Unreal Engine, Godott etc. y se ha de escoger uno que se adapte a nuestras necesidades. A continuación, se muestra un poco de información sobre cada uno:



🎮 **Unity 3D:** Es un motor de videojuegos donde se programa con C #, es muy potente y al mismo tiempo tiene una muy buena curva de aprendizaje, ya que hay una gran comunidad detrás y muchos tutoriales a disposición de todos. Además, se puede exportar a casi todas las plataformas existentes y te da la posibilidad de hacer juegos tanto con 3D como en 2D. Otra ventaja es que es gratuito si ganas menos de 100.000 \$ al año.



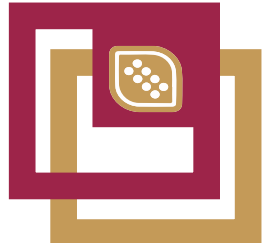
🎮 **Unreal Engine:** Es un motor de videojuegos muy famoso, se programa con C ++ lo que lo hace un poco más complicado. También te permite crear juegos con 3D y 2D con mucha facilidad y exportarlos a una gran cantidad de plataformas. Unreal Engine también es gratuito, pero solo hasta los primeros 3000 \$, después de ganar esta cantidad se debe pagar el 5% de las ganancias.



🎮 **Godot:** Godot es la alternativa open source gratuita que te permite hacer todo lo anterior y, además, puedes elegir con qué lenguaje el programa se puede hacer con lenguajes como C#, C++ o GDScript, un lenguaje basado en Python.



Si quieres hacerlo con el framework tienes que escoger la plataforma donde quieres publicar tu juego. En el caso de Android, lo puedes hacer con Android Studio y programar con Java o Kotlin, si prefieres publicarlo en iOS lo tienes que hacer con Xcode y programarlo con Swift.



VISUALES

Decidir qué modelos usar, si 3D o sprites (dibujos en el caso de un juego 2D) puede suponer un problema, ya que crearlos uno mismo y que queden bien puede ser más complicado que programar el juego entero. Pero gracias a muchos artistas caritativos que cuelgan sus creaciones en Internet de manera gratuita, se ha conseguido que no sean necesarias grandes habilidades artísticas para crear un juego estéticamente bonito. Podemos encontrar estos recursos en lugares como Itch, Kenney, Opengameart o PolyGoogle entre otros. Si en cambio, lo quieres crear tú, tienes disponibles programas gratuitos como Blender para hacer modelos 3D o Krita para hacer sprites.

PUBLICACIÓN

Una vez creado el juego se publicará en la tienda de aplicaciones, si lo queremos publicar en Google Play, se debe pagar 25 US y ya podrás publicar todas las aplicaciones que quieras. Si quieres publicarlo en la App Store, se debe pagar 99 US al año. Si se quiere publicar en Android aparte de Google Play hay otras tiendas de aplicaciones disponibles, aunque Google Play sea el oficial, en países como China o la India utilizan otras tiendas como la tienda de aplicaciones de Tencent (en el caso de China).

POSICIONAMIENTO

Una vez colgada la aplicación en la tienda de aplicaciones, nuestro objetivo será hacer que los usuarios la descubran, tenemos que dar una razón al usuario para descargar tu aplicación y no otro. Para lograr esto, utilizaremos ASO (App store optimization), sirve para mejorar el posicionamiento de nuestro juego dentro de la tienda de aplicaciones. No hay una única manera de hacer ASO, simplemente es seguir unas pautas que te ayudarán subir en el ranking de aplicaciones. Pautas como por ejemplo cómo escribir la descripción de la aplicación, de cómo diseñar el icono, etc.

Se debe considerar que el crear videojuegos suele tener finalidad económica. De acuerdo con SuperData, en 2020, las ganancias de la industria aumentaron en un 12% con respecto al año anterior; alcanzando los 139.000 mil millones de dólares globalmente. De este total, los juegos móviles tomaron la delantera, generando más de 73.8 mil millones de dólares americanos, mientras que, por su parte, los juegos para PC dejaron solo \$33.1 mil millones de dólares en ganancias, y el sector de las consolas \$19.7 mil millones.



Actividad 6. Diferencias y Similitudes



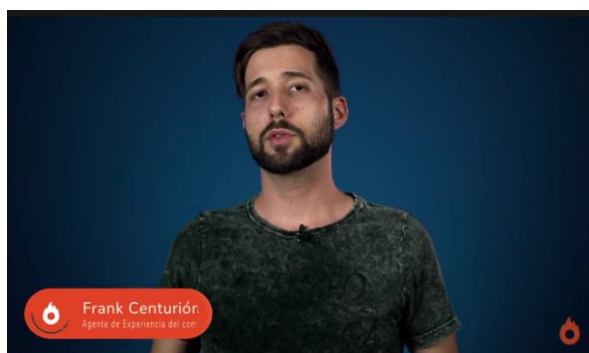
Instrucciones: Realizar un podcast, donde identifiques los diversos elementos necesarios para la programación de un videojuego en web y uno en el móvil. Considera lo siguiente:



- 1 Tomando en cuenta las lecturas previas, elabora un podcast en el que identifiques los diversos elementos necesarios para poder realizar un juego Web y un juego móvil.
- 2 Debes identificar las similitudes y diferencias, así como las ventajas y desventajas.
- 3 Considera que tu podcast no debe durar más de 3 minutos.



Comienza con esto



<https://youtu.be/4cWlc1KFIjQ>

No olvides utilizar el instrumento de evaluación correspondiente para la elaboración de la actividad.





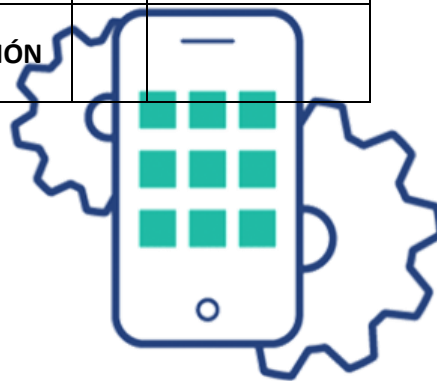
INSTRUMENTO DE EVALUACIÓN
LISTA DE COTEJO 7
Actividad 6: Diferencias y Similitudes

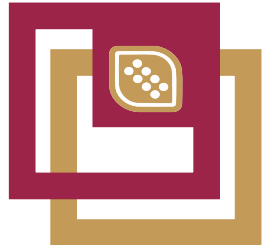


DATOS GENERALES

Nombre(s) del alumno(s)	Matrícula(s)
Producto: Podcast	Fecha
Submódulo 1: Aplicación Móvil y Juegos	Periodo: 2025 – 2026B
Nombre del docente	Firma del docente

CRITERIO	INDICADORES	VALOR OBTENIDO		PONDERACIÓN	CALIF	OBSERVACIONES Y/O SUGERENCIAS DE MEJORA
		SÍ	NO			
1	Se presenta y menciona sus datos esenciales.			2		
2	Identifica los elementos necesarios para la creación de videojuegos en el móvil.			2		
3	Identifica los elementos necesarios para la creación de videojuegos en la web.			2		
4	Identifica las diferencias en el desarrollo de videojuegos.			2		
5	Muestra dominio en el uso de las tecnologías de la Información y de la comunión.			1		
6	Entrega en tiempo y forma.			1		
CALIFICACIÓN						





Software de Desarrollo para Videojuegos

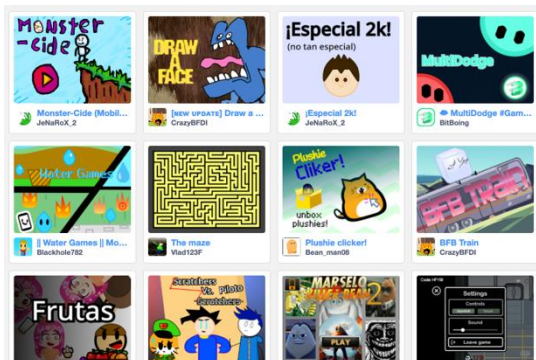
SCRATCH



Es un entorno de programación desarrollado por un grupo de investigadores del Instituto de Tecnología de Massachusetts (MIT) que aprovecha los avances en el diseño de interfaces para hacer que la programación sea atractiva y accesible para todo aquel que se enfrente por primera vez a programar.



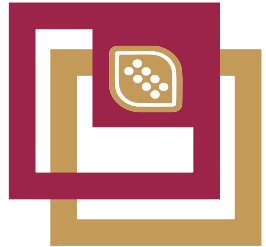
Programando con Scratch se pueden crear historias interactivas, juegos, animaciones, música y producciones artísticas. La página web del MIT también permite compartir creaciones e ideas con otros jóvenes con intereses similares. Desde su lanzamiento al público en mayo de 2007, la Web de Scratch se ha convertido en una gran comunidad en línea, donde se aprende, se intercambian ideas y se mezclan programas propios con los de otros.



Cada día, los scratchers de todo el mundo, suben al sitio más de 1.500 proyectos nuevos con el código de acceso libre (cualquiera que esté registrado puede ver cómo se ha creado el proyecto, bajárselo, modificarlo, etc). La edad de la mayoría de los visitantes del sitio web se ubica entre los 8 y los 16 años (con un pico en los 12 años) aunque un grupo considerable de adultos participa también. Scratch está disponible en múltiples idiomas, entre ellos el castellano.

Scratch promueve el pensamiento computacional y las habilidades en resolución de problemas; enseñanza y aprendizaje creativos, auto expresión y colaboración; e igualdad en informática.





En la computadora o en el teléfono, vamos a utilizar la Aplicación Web Progresiva (PWA)

SCRATCH

Para ello, debes abrir tu navegador preferido; de preferencia utiliza Google Chrome para hacer uso de las extensiones y datos que Google tiene con tu cuenta.

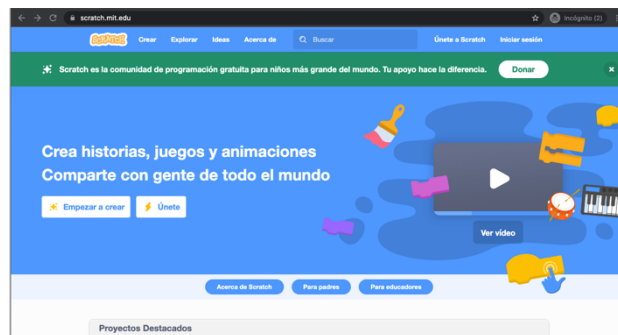


Escribe la URL: <https://scratch.mit.edu/>

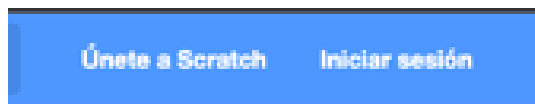


Después de ingresar

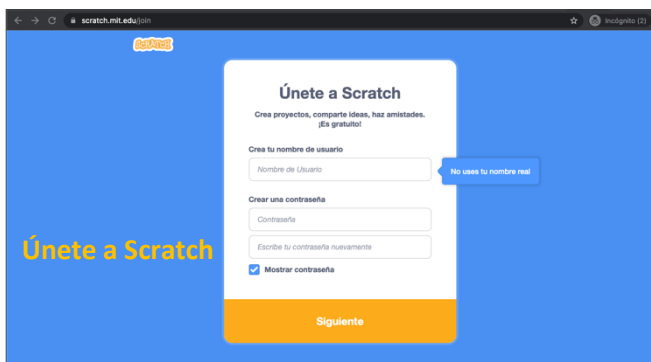
Puede aparecer en inglés, pero puedes usar la herramienta para traducir o elegir en la lista desplegable del idioma.



Seguidamente elige Scratch o si ya tienes



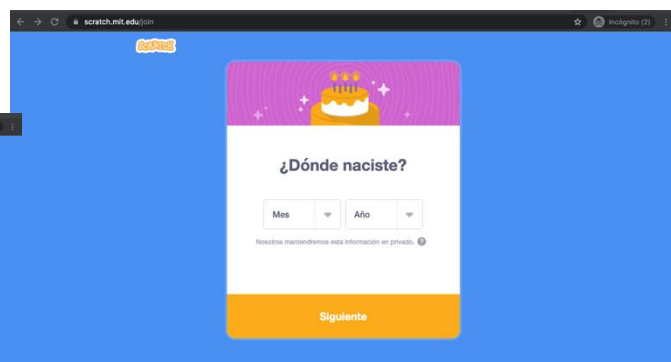
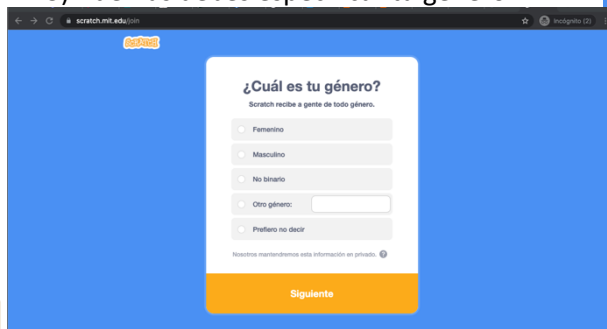
para crear una cuenta en cuenta, elige **Iniciar sesión**.

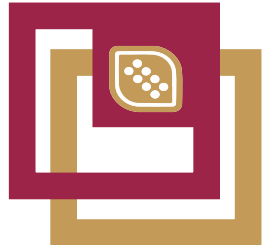


1) Debemos crear un nombre de usuario y tu respectiva contraseña, confírmala y NO LA OLVIDES.

2) Después de tener tu usuario, ahora es necesario dar tu fecha de nacimiento.

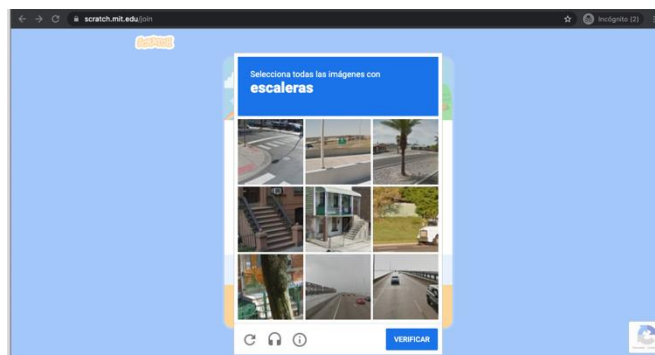
3) Además debes especificar tu género:



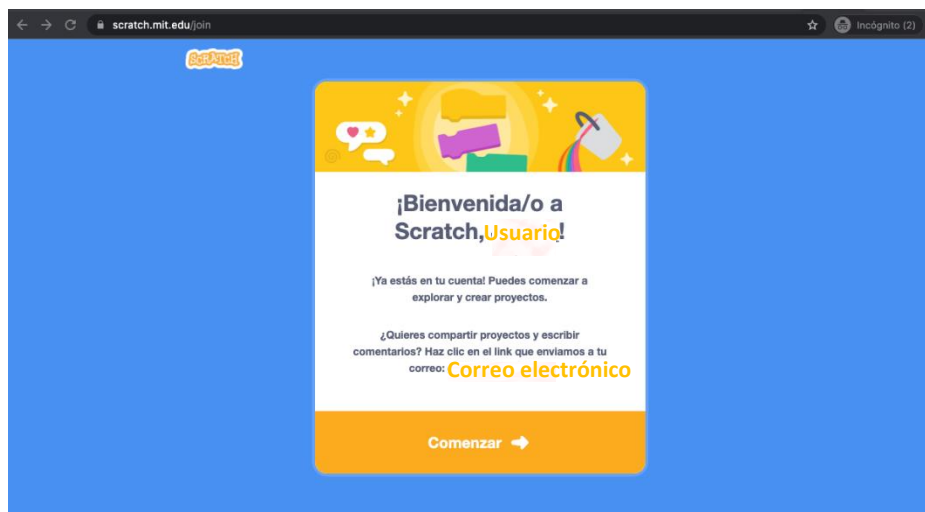


4) Casi para terminar, debemos proporcionar un correo electrónico al cual vincular la cuenta:

5) Ahora realizamos la comprobación (recuerda que te puede pedir que compruebes otras ilustraciones, presta atención a lo que te solicita).

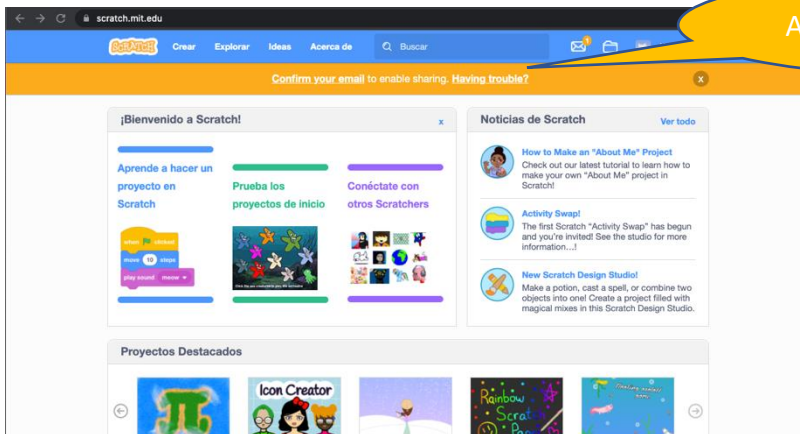
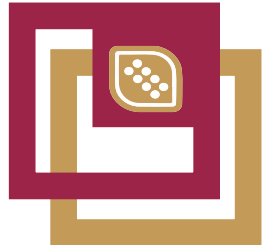


Si seguiste todos estos pasos, verás en tu pantalla la BIENVENIDA:



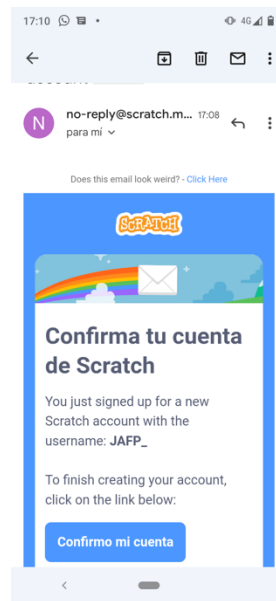
Para concluir es necesario confirmar el correo que te llegó a tu bandeja en el correo electrónico.



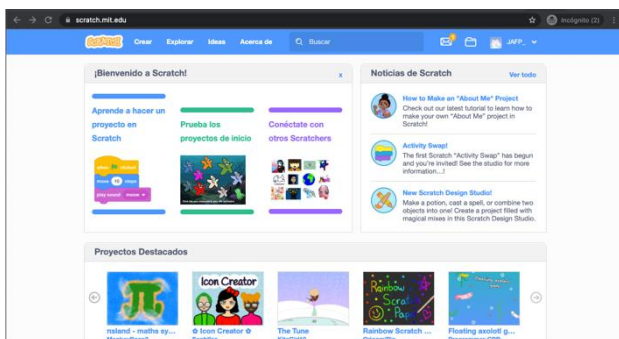


Aquí te pide la confirmación

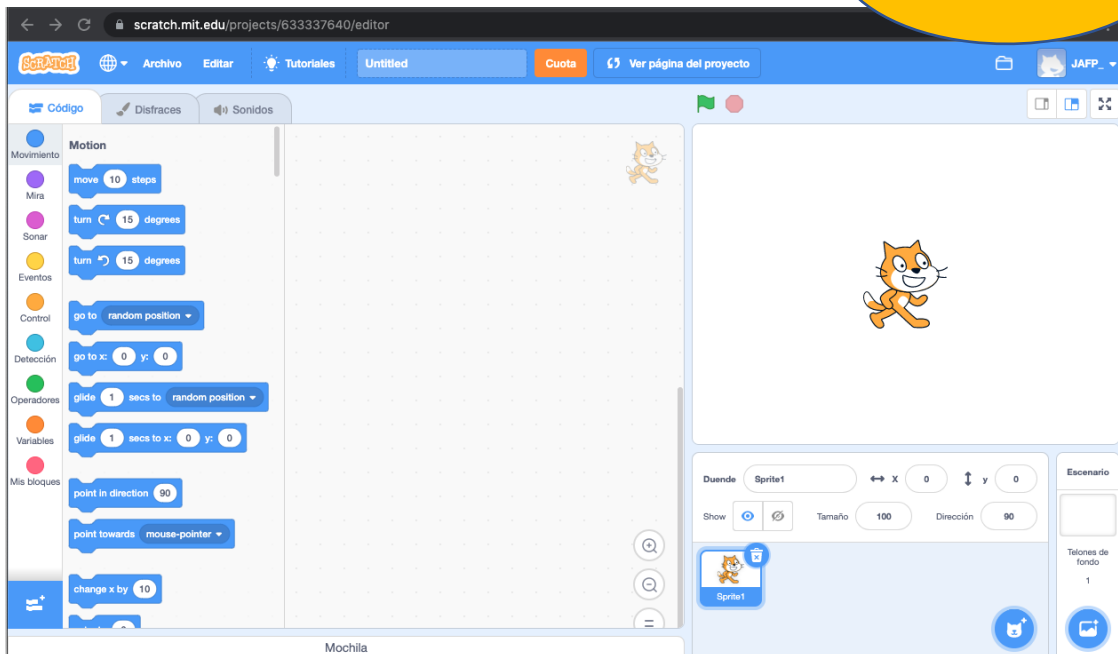
MENSAJE DE CONFIRMACIÓN

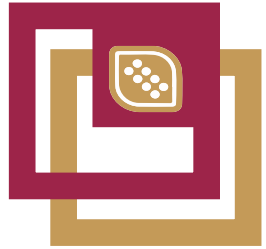


AHORA ESTAMOS LISTOS PARA USAR ESTA GRAN HERRAMIENTA:



Da clic en el menú crear y que empiece el juego





BIBLIOGRAFÍA

Desarrollar juegos móviles - "From Zero to Hero". (2020, 30 octubre). inLab FIB.
<https://inlab.fib.upc.edu/es/blog/desarrollar-juegos-moviles-zero-hero>

Introducción para desarrollo de juegos para la Web - Desarrollo de Videojuegos | MDN. (2014, 16 octubre). Introducción Para Desarrollo de Juegos Para La Web - Desarrollo de Videojuegos.
<https://developer.mozilla.org/es/docs/Games/Introduction>

School, T. (2021, 10 septiembre). *Lenguajes para programar videojuegos.* Tokio School.
<https://www.tokioschool.com/noticias/lenguajes-programar-videojuegos/>

Scratch - Imagine, Program, Share. (2020). Scratch. <https://scratch.mit.edu/>

Volk. (2021, 6 mayo). *Videojuegos móviles: Una evolución en la industria gamer* -.
<https://www.volkgames.com/videojuegos-moviles-una-evolucion-en-la-industria-gamer/>



Lectura No 5. Funcionalidades: Programación en Scratch

Instrucciones: Realiza el análisis de la siguiente información, posteriormente, desarrolla la práctica.



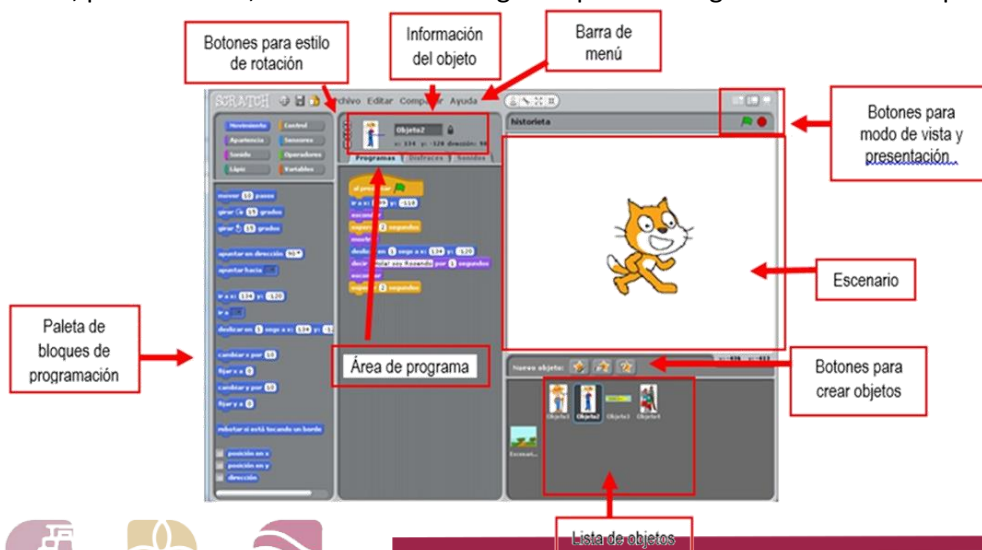
LECTURA 5. "FUNCIONALIDADES: PROGRAMACIÓN EN SCRATCH"

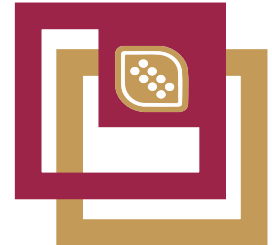


Es un entorno de programación desarrollado por un grupo de investigadores del Instituto de Tecnología de Massachusetts (MIT) que aprovecha los avances en el diseño de interfaces para hacer que la programación sea atractiva y accesible para todo aquel que se enfrente por primera vez a programar.



Atendiendo a las competencias definidas por la OCDE y la UNESCO, se puede comprobar que, utilizando Scratch, los estudiantes pueden desarrollarlas de una forma completa, favoreciendo un desarrollo integral del alumno a la vez que se prepara para un futuro profesional incierto, pero sin duda, rodeado de tecnología. Repasemos algunas de estas competencias.





Pensamiento creativo: Scratch favorece la curiosidad intelectual y el pensamiento creativo, habilidades ambas con importancia creciente en el actual mundo cambiante. Scratch implica a la gente joven en la búsqueda de soluciones innovadoras a problemas inesperados, no sólo aprendiendo cómo resolver problemas predefinidos sino enfrentándose a nuevos desafíos y creando nuevas soluciones.



Razonamiento abstracto: Con Scratch, como resultado directo de las clases de programación, el alumno se enfrenta a caminos de razonamiento divergentes, deben anticipar errores y evaluar rápidamente escenarios mentales. Este don sirve hoy en día tanto para manejar una cuenta de correo como para sacar el coche de un aparcamiento.



Pensamiento computacional: A medida que los alumnos aprenden a programar en Scratch, desarrollan el pensamiento crítico y computacional. Para construir los proyectos, los alumnos necesitan coordinar el tiempo y las interacciones entre los diferentes “sprites” (objetos programables). La habilidad de programar de forma interactiva proporciona a los estudiantes una experiencia directa con los conceptos fundamentales sobre sistemas a través de la percepción y con un feedback inmediato.



mover 10 pasos decir [Hola] esperar 1 segundos tocar sonido [miau] Sentencias	al tocar el apuntador del ratón al ratón presionado si <1> si <2> Expresiones Booleanas	si si si no Condiciones	por siempre por siempre si Ciclos	var1 fijar var1 a 0 Variables
al presionar al presionar tecla espacio al presionar Objeto 1 Hilos	al recibir enviar a todos Eventos	unir hola mundo letra 1 de mundo longitud de mundo Cadenas	número al azar entre 1 y 10 + raíz cuadrada de 10 Expresiones numéricas	lista añade cosa a lista longitud de lista Listas





Resolución de problemas: Scratch favorece la

identificación de problemas y su solución en un contexto significativo, innovador y atractivo. Crear un proyecto en Scratch requiere pensar en una idea, preguntarse

cómo dividir el problema en varios pasos y después implementarlos utilizando los bloques de código dispuestos para ello. Los estudiantes pueden cambiar dinámicamente el código y ver inmediatamente los resultados. A lo largo del proceso, los estudiantes experimentan y se entrenan en la resolución iterativa de problemas.

Pasos para la resolución de problemas



Aprendizaje autónomo: Tener una idea y pensar en cómo programarla requiere

**Aprendizaje
Autónomo**



persistencia y práctica. Cuando los jóvenes trabajan en los proyectos que surgen de sus ideas, encuentran un significado personal. Trabajar sobre sus ideas les proporciona una motivación interna para sobrellevar los desafíos y las frustraciones que encuentren en el proceso de diseño y resolución de problemas.

 **Trabajo colaborativo:** Debido a que los programas de Scratch son construidos con bloques gráficos, el código programado es más fácil de leer y de compartir que en otros lenguajes de programación. Los objetos visuales y el código modular propicia la colaboración, permitiendo a los estudiantes

trabajar juntos en proyectos e intercambiar objetos y código. Además, en la página web de Scratch los estudiantes pueden subir sus proyectos y recibir valoraciones y críticas y en la que también se pueden apoyar y colaborar. El concepto de compartir es parte integral de la interfaz de usuario de Scratch la cual tiene, en la parte superior de la página, tanto un menú como un icono de “Compartir” claramente visibles. Explorando los miles proyectos de la librería existente, los “Scratchers” tienen ideas para nuevos proyectos y aprenden nuevas técnicas de programación.



Comunicación: Una comunicación efectiva requiere, hoy en día, algo más que saber leer y escribir textos. Scratch implica a los jóvenes en la elección, manipulación e integración de distintos elementos multimedia para expresar su creatividad y para poner en práctica su capacidad de persuasión. Cuando los alumnos crean un proyecto, tienen una audiencia en mente y necesitan pensar en cómo responderá otra gente ante sus proyectos. Como los proyectos de Scratch son muy fáciles de revisar y cambiar, los estudiantes pueden modificar sus proyectos basándose en el feedback recibido por otras personas.



Manejo de TIC: Es habitual referirse a las personas jóvenes como “nativos digitales” debido a su aparente fluidez con las TIC. Pero, aunque interactúan con los medios digitales todo el tiempo, pocos son capaces de crear sus propios juegos, animaciones o simulaciones. Es como si pudieran “leer” pero no “escribir”. La competencia digital requiere no solamente tener habilidad para chatear, navegar o interactuar sino también la habilidad de diseñar, crear e inventar con los nuevos medios. Al trabajar en los proyectos de Scratch, los alumnos aprenden a crear, seleccionar y gestionar múltiples elementos multimedia, incluyendo texto, imágenes, animaciones y grabaciones de audio. A medida que los estudiantes ganan experiencia en multimedia, se vuelven más receptivos y críticos en los medios de comunicación que les rodean.

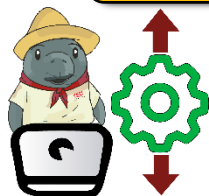


Life Long Learning (Aprendizaje a largo plazo): Con el uso de Scratch, el alumno es quien dirige su aprendizaje. De esa forma, se desarrolla la competencia de aprender a aprender y los estudiantes, al conseguir sus objetivos se percatan de que realmente son capaces de aumentar sus conocimientos por sí mismos, sin depender de instrucción externa.

¿Estás listo@
para iniciar?



Práctica Guiada “Mostrar y esconder”



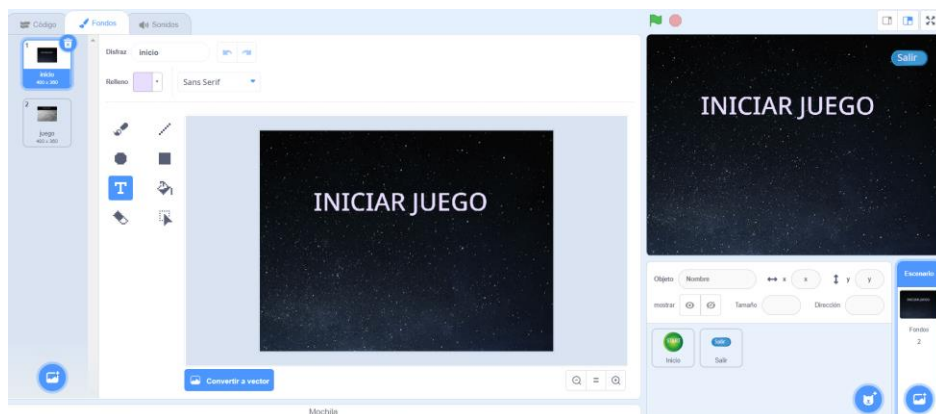
Instrucciones: Prepara tu equipo de cómputo con SCRATCH y sigue las indicaciones para crear un mini videojuego que permite mostrar objetos y ocultarlos.



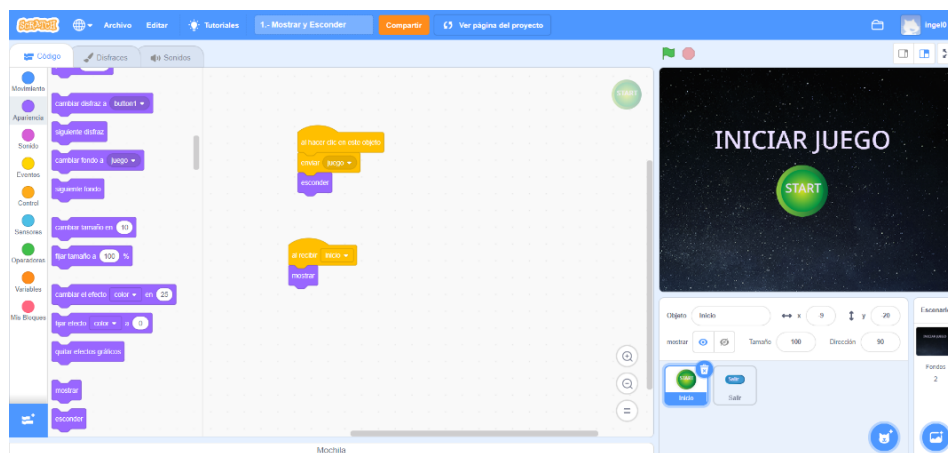
PRÁCTICA GUIADA. “MOSTRAR Y ESCONDER”

Con ayuda de tu maestro realiza la siguiente práctica en la que aprenderás como mostrar y esconder objetos en Scratch.

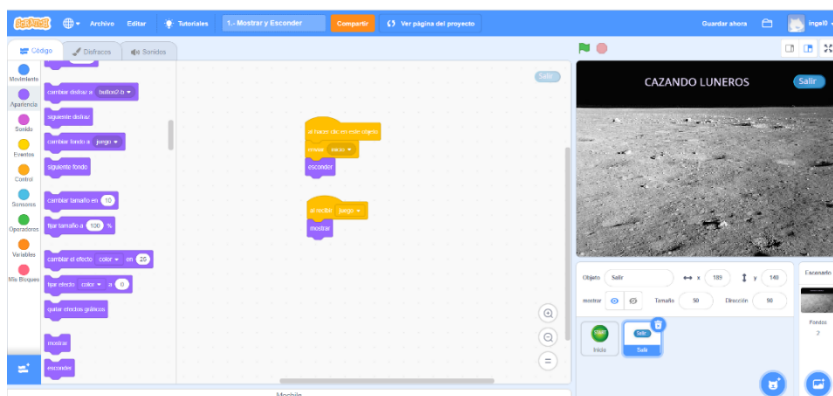
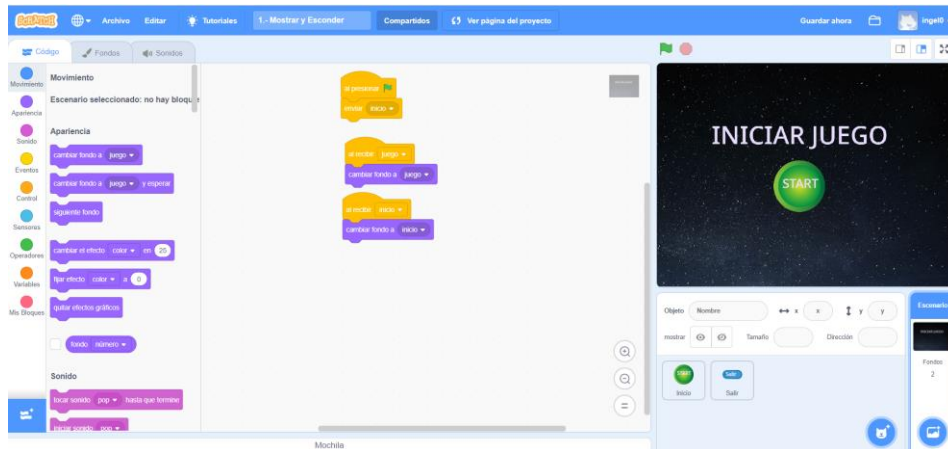
- 1.- Debes agregar dos fondos distintos, en los fondos puedes escribir textos dependiendo las necesidades. No olvides colocar el nombre correspondiente al disfraz (en este caso el fondo).



- 2.- En el fondo principal debes agregar los siguientes bloques para que se pueda realizar la interacción.



3.- Debes agregar dos botones un en la pantalla de inicio y otro en la pantalla del juego. A cada botón deberás agregar los códigos en bloques correspondientes, para que sean funcionales.



4.- Cada botón debe tener los bloques mostrados. Recuerda tener en cuenta el nombre de los objetos y los mensajes que devuelve cada uno de ellos. Tu maestro te ayudará a comprender mejor estas funcionalidades.

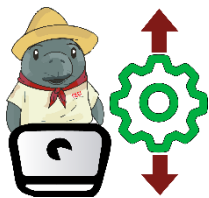


Revisa la práctica



<https://scratch.mit.edu/projects/629245836/>

Práctica Guiada “Disfraces y Sonidos”



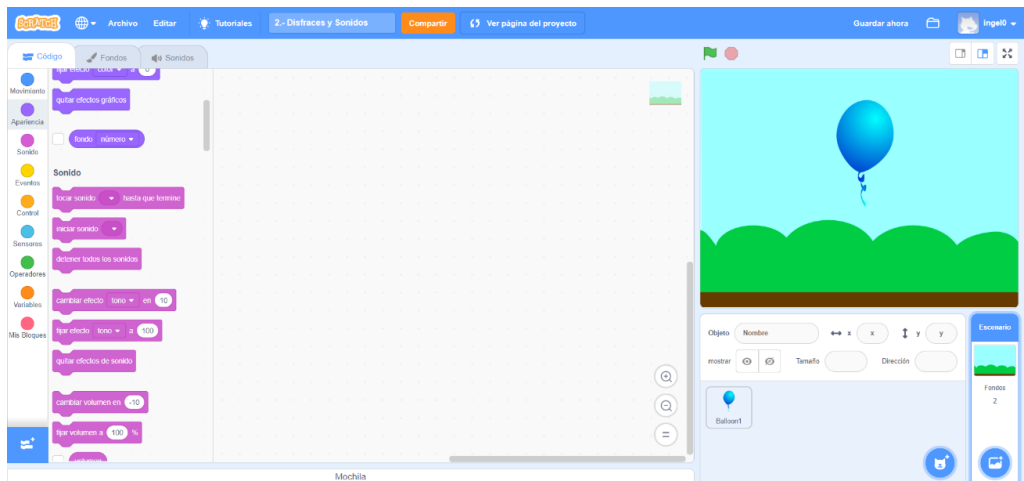
Instrucciones: Prepara tu equipo de cómputo con SCRATCH y sigue las indicaciones para crear un mini videojuego que permite mostrar trabajar personajes y sonidos.



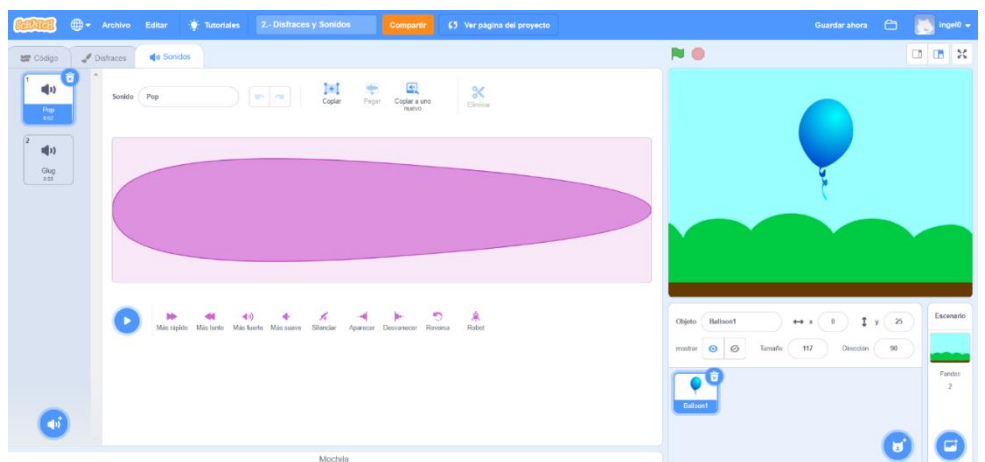
PRÁCTICA GUIADA. “DISFRACES Y SONIDOS”

En esta ocasión realizarás una práctica guiada junto con tu maestro, en la que agregaremos sonidos y haremos cambios de disfraces en los objetos.

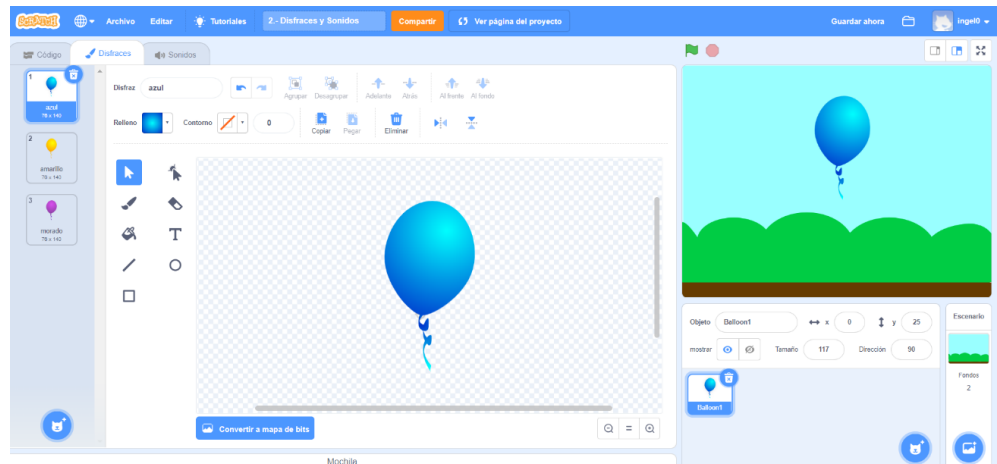
1.- Agrega un fondo y un objeto, debemos procurar que el objeto tenga diversos disfraces. Esto lo vemos al momento de seleccionar el disfraz.



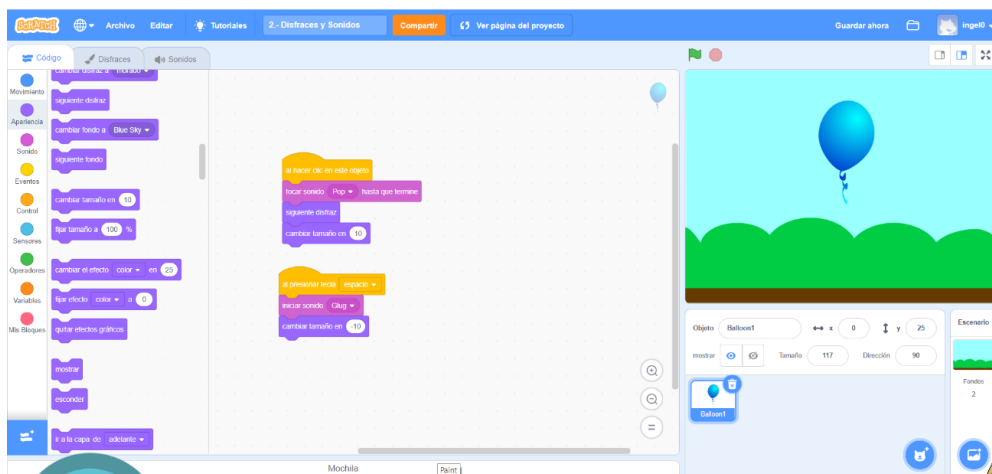
2.- Los disfraces pueden verse al hacer doble clic en el objeto, en la pestaña disfraces. Debemos tener en cuenta que es recomendable colocar un nombre específico y fácil de identificar a cada disfraz que vayamos a utilizar.



3.- Dentro del mismo objeto, debemos ir a la pestaña sonidos y elegir dos sonidos distintos para nuestra práctica.



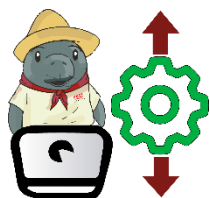
4.- Una vez agregado los elementos gráficos y de sonido, solo debemos agregar los bloques que hagan funcionar nuestra práctica, estos son agregados dentro del objeto que cambiará de disfraz.



Revisa la práctica



Práctica Guiada “Mover Objeto”



Instrucciones: Prepara tu equipo de cómputo con SCRATCH y sigue las indicaciones para crear un mini videojuego que permite mostrar mover objetos en un escenario.

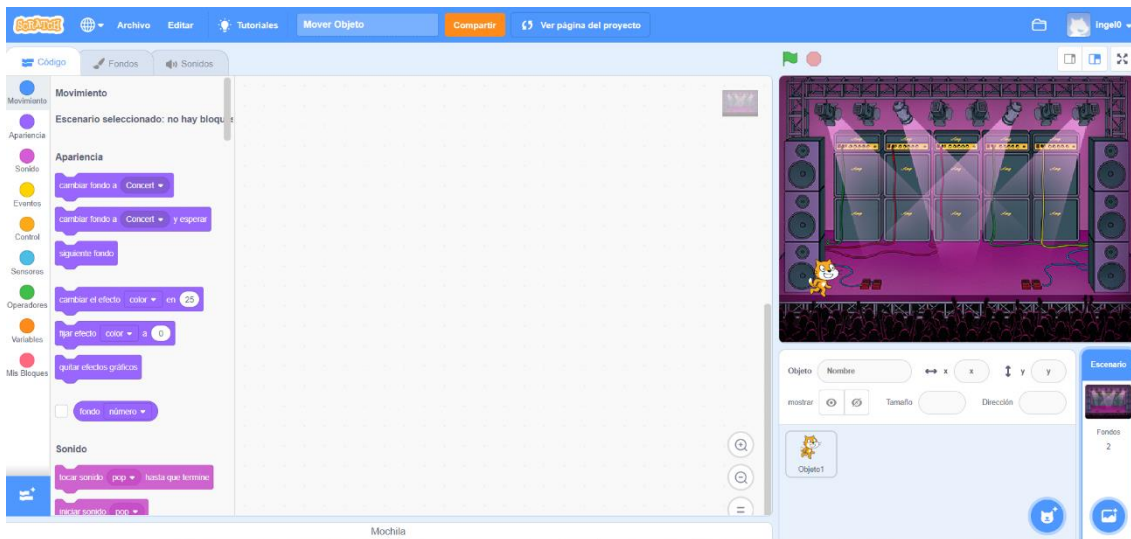


PRÁCTICA GUIADA. “MOVER OBJETO”

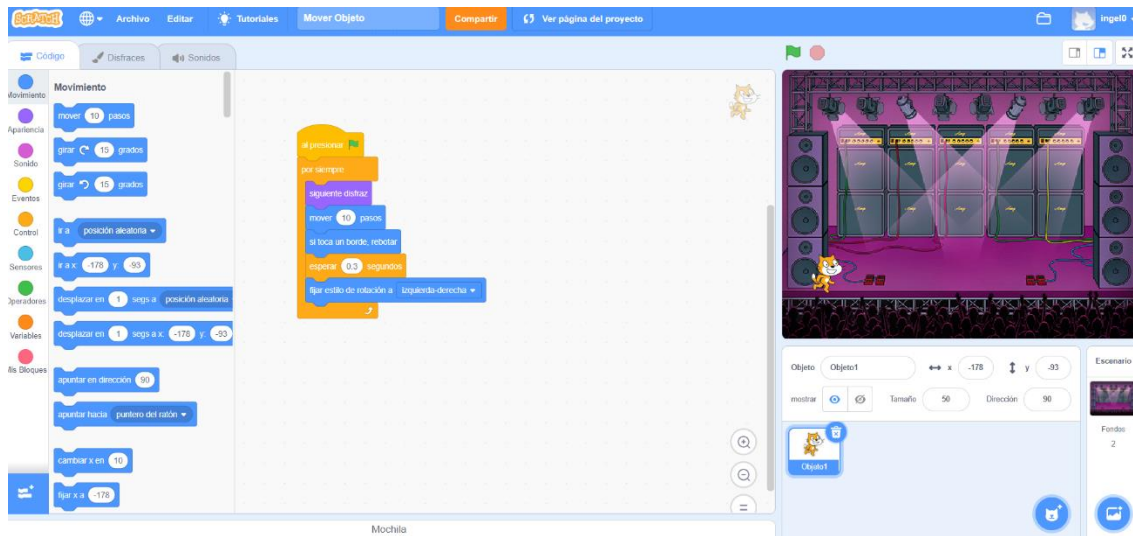
En esta ocasión realizaremos una práctica con ayuda de tu docente. En esta práctica moveremos objetos en Scratch de un punto a otro, este movimiento puede ser automático o al presionar alguna tecla o clic en el mouse.



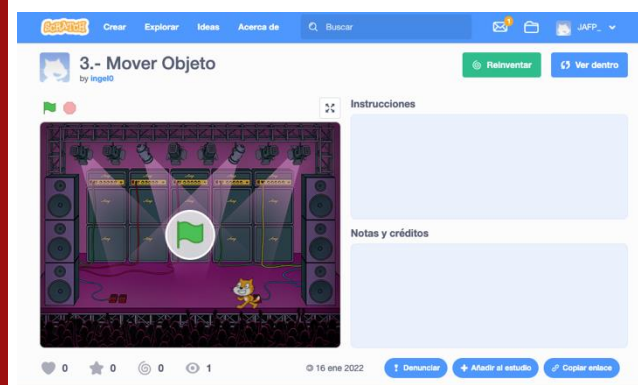
1.- Debemos agregar un fondo y un objeto. De preferencia, el objeto debe tener dos formas diferentes que demuestren movimiento.



2.- Una vez que se tienen los elementos gráficos solo debemos agregar los siguientes bloques dentro del objeto.

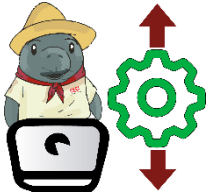


Revisa la práctica



<https://scratch.mit.edu/projects/628116027/>

Práctica Guiada "Sumar Puntos"



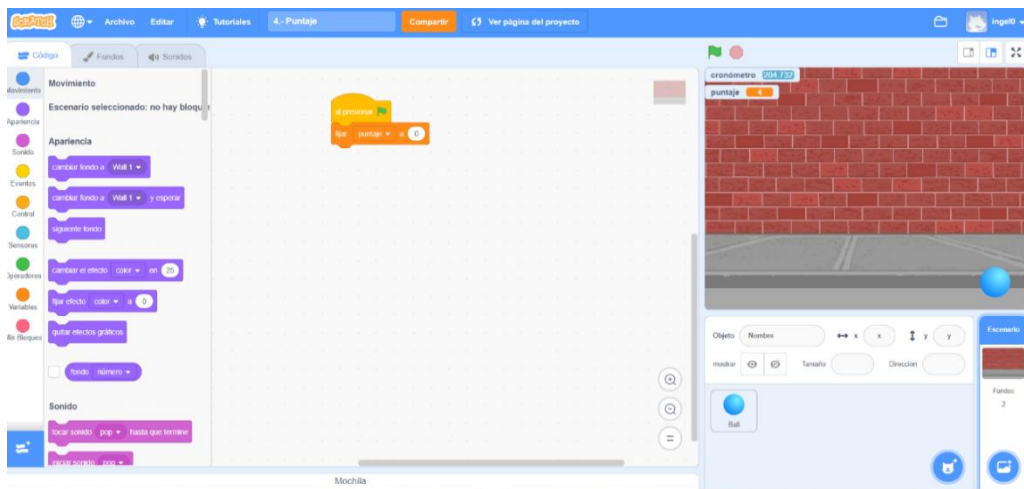
Instrucciones: Prepara tu equipo de cómputo con SCRATCH y sigue las indicaciones para crear un mini videojuego que permite programar acciones para sumar puntos.



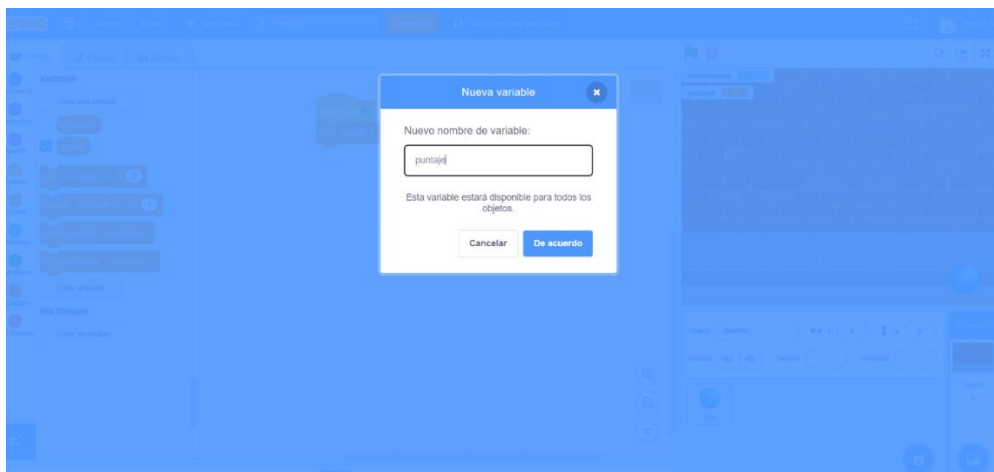
PRÁCTICA GUIADA. "SUMAR PUNTOS"



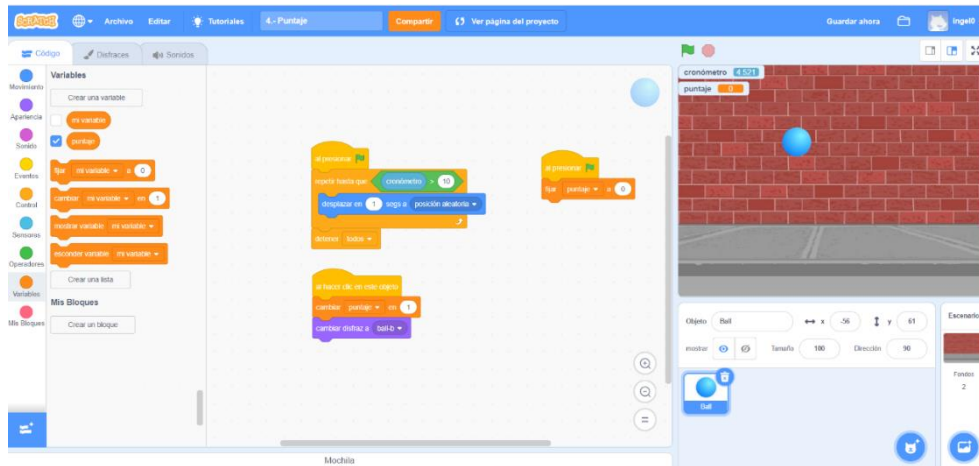
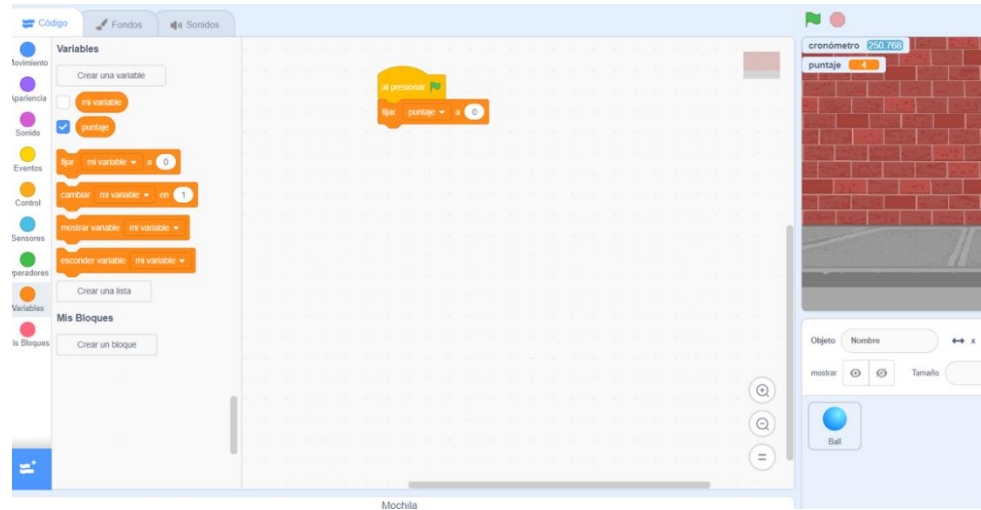
1.- Primero debemos agregar un fondo a nuestra práctica.



2.- Después deberemos agregar una variable a nuestro código de bloques.



3.- Debemos colocar como visible la variable y el cronómetro para que aparezcan en la pantalla del juego.



4.- Por último, debemos agregar dentro del objeto que se moverá el código en bloque correspondiente.

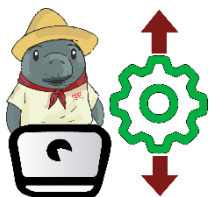


Revisa la práctica



Práctica 2. Vamo a Jugar

Instrucciones: Tomando como base las prácticas previas diseña y elabora un videojuego básico considerando lo estudiado en la Lectura 4. Dicho videojuego deberás elaborarlo utilizando la herramienta Scratch. Una vez concluido el juego deberás realizar una presentación del mismo ante el grupo.



PRÁCTICA 2. "VAMO A JUGAR"



Piensa el propósito del videojuego, deben desarrollar una propuesta utilizando Scratch como PWA.

Considera el siguiente recurso audiovisual como apoyo para desarrollar su propuesta de videojuego, además de la lectura 5; no olviden presentar sus datos completos.

Manos a la obra



<https://scratch.mit.edu/projects/633337640/editor>

TEAM
LEWIA

DESARROLLO DE SOFTWARE



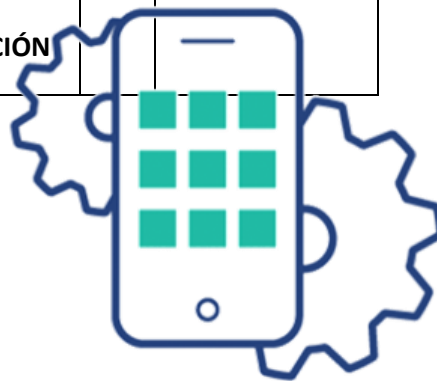
INSTRUMENTO DE EVALUACIÓN
LISTA DE COTEJO 8
Práctica 2: Vamo a Jugar



DATOS GENERALES

Nombre(s) del alumno(s)	Matrícula(s)
Producto: Propuesta de Videojuego en Scratch	Fecha
Submódulo 1: Aplicación Móvil y Juegos	Periodo: 2025 – 2026B
Nombre del docente	Firma del docente

CRITERIO	INDICADORES	VALOR OBTENIDO		PONDERACIÓN	CALIF	OBSERVACIONES Y/O SUGERENCIAS DE MEJORA
		SÍ	NO			
1	El videojuego demuestra la conceptualización y el diseño.			2		
2	Demuestran el uso de los objetos, inserta escenarios.			2		
3	El diseño propuesto, mueve los objetos y suma puntos.			2		
4	Se aprecia el conocimiento y dominio del tema.			2		
5	Muestra dominio en el uso de las tecnologías de la Información y de la comunión.			1		
6	Entrega en tiempo y forma.			1		
	CALIFICACIÓN					



INNOVANDO CON CREATIVIDAD



SIGA



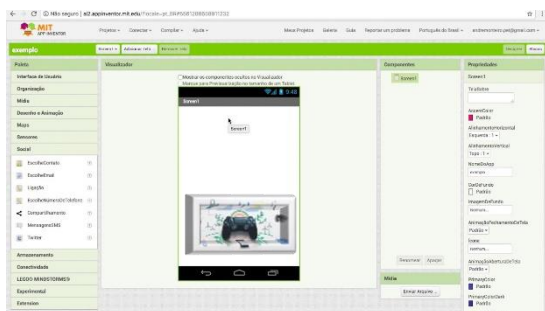
Propósito:

Desarrollar una aplicación móvil con temática sostenible en equipos de 5 integrantes donde presenta un catálogo de diversas plantas para el cuidado, uso, protección de éstas o hábitos, actividades de prevención, reducción, reciclado y reutilización de los desechos en el hogar utilizando el software de programación propuesto, para socializarlo en plenaria.

Instrucciones: Con los conocimientos que has adquirido a lo largo del submódulo, es momento de diseñar en equipo, una aplicación móvil en la que se hace conciencia en el cuidado de los recursos naturales, la sociedad y la sostenibilidad.

¿Qué debemos hacer?

Comienza con el Principio



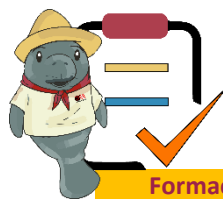
Considera los pasos estudiados en la Lectura 3. Diseña un App Móvil. Para después utilizar MIT Inventor como software para desarrollar su propuesta de la aplicación móvil, integra un videojuego si crees apropiado.

También, recuerda que tienes las prácticas guiadas que te apoyarán en el uso de herramientas sencillas para el desarrollo de la aplicación móvil.



piensa verde





COLEGIO DE BACHILLERES DE TABASCO PLANTEL No ____
RÚBRICA DE EVALUACIÓN INNOVANDO CON CREATIVIDAD



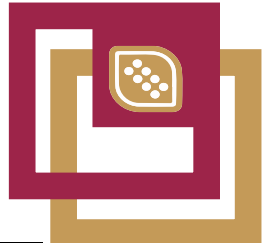
SIGA

Formación Laboral Básica Desarrollo de software		Módulo IV. Aplicaciones Móviles		Submódulo 1. Programación Móvil y Juegos	
Situación didáctica. Innovando con Creatividad.			Nombre de los estudiantes:		
Docente:					
Semestre. 6to Grupo.		Turno.		Fecha.	
Competencia laboral básica					
CLBDS7. Reconoce soluciones mediante software de aplicación, para aplicaciones móviles y juegos, relacionándose con sus semejantes de forma colaborativa, mostrando disposición al trabajo metódico y organizado en diferentes contextos.					
Evidencia de aprendizaje. Una aplicación móvil					

Instrucciones: Marque con una (X) para resaltar si cumplió con los criterios de la actividad. Sume los puntos para obtener la calificación.

INDICADORES	%	CRITERIOS			
		EXCELENTE	BUENO	SUFICIENTE	INSUFICIENTE
Fase de Diseño	30%	Presenta una temática, los objetos corresponden a la temática, presenta los sonidos acordes a la temática y se comprende la mecánica de la aplicación móvil. 30	Presenta al menos 3 de los 4 elementos que corresponden al diseño. 25	Presenta al menos 2 de los 4 elementos que corresponden al diseño. 15	Presenta al menos 1 de los 4 elementos que corresponden al diseño. 5
Fase de Producción	20%	Hace uso correcto de botones, Sitio web, sonidos y accesibilidad. 20	Hace uso correcto de al menos 3 elementos en la programación de la aplicación móvil. 15	Hace uso correcto de al menos 2 elementos en la programación de la aplicación móvil. 10	Hace uso correcto de al menos 1 elementos en la programación de la aplicación móvil. 5

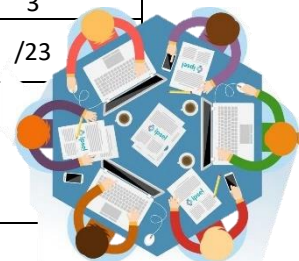




INDICADORES	%	CRITERIOS			
		EXCELENTE	BUENO	SUFICIENTE	INSUFICIENTE
Programación de Bloques	20%	Muestra la programación de los objetos, sonidos, acciones a través de los bloques de código. 20	Muestra la programación de los objetos y las acciones a través de los bloques de código. 15	Muestra la programación de los objetos a través de los bloques de código. 10	Muestra la programación de la aplicación móvil sin bloques. 5
Presentación y exposición	20%	Los estudiantes exponen los detalles de su aplicación móvil y muestran los elementos en su programación. 20	Los estudiantes exponen los detalles de su aplicación móvil y muestran algunos elementos en su programación. 15	Los estudiantes exponen los detalles de su aplicación móvil sin mostrar los elementos en su programación. 10	Solo exponen la aplicación móvil pero no describen los detalles en la programación de la aplicación móvil. 5
Todo el equipo ha participado en la creación de la aplicación móvil	10%	Todos y cada uno de los miembros del equipo han participado y colaborado en la creación de la aplicación móvil. 10	La mayoría de los integrantes del equipo han participado y colaborado en la creación de la aplicación móvil. 8	Pocos miembros del equipo han participado y colaborado en la creación de la aplicación móvil. 5	La construcción de la aplicación móvil ha sido realizada por un solo miembro del equipo y los demás observan. 3
Total	100%	/100	/83	/50	/23



Firma del evaluador. _____



Conflicto cognitivo	Menciona el software que consideras en la programación móvil. ¿Cuáles son las fases de desarrollo para aplicaciones móviles? Menciona el software que consideren en el diseño y desarrollo de juegos. ¿Los juegos solo sirven para divertirse? ¿Cuál es la utilidad de crear aplicaciones móviles?
----------------------------	---



Submódulo II. Diseño y Programación en Robótica



Propósito del Módulo

Evaluar soluciones con aplicaciones móviles y juegos, mediante software de aplicación; para solucionar problemas de su entorno, escuchando y respetando diferentes puntos de vista, con el fin de promover el bien común y estructurar propuestas de robótica básica, a través de su arquitectura y anatomía para desarrollarlos en su contexto, afrontando las consecuencias de sus actos.

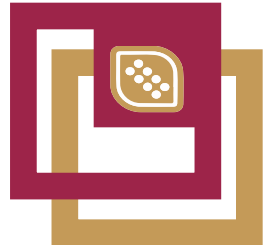
Aprendizajes Esperados

- ❖ Justifica los conocimientos adquiridos, aportando ideas creativas en la solución de problemas del entorno, para el diseño y desarrollo de diversos tipos de robot, afrontando los retos y asumiendo la frustración como parte del proceso.
- ❖ Selecciona el software de programación, en forma creativa, para movimientos y acciones a realizar por parte del robot, dando solución a los problemas de su contexto y expresando sus diversas opciones.

Competencia Laboral Básica

CLBDS8. Identifica propuestas de robótica básica, a través de sus fundamentos, arquitectura y anatomía; en cualquier ámbito, afrontando retos y asumiendo la frustración como parte de un proceso.





DOSIFICACIÓN PROGRAMÁTICA

Formación Laboral Básica: Desarrollo de Software

Módulo IV: Aplicaciones Móviles

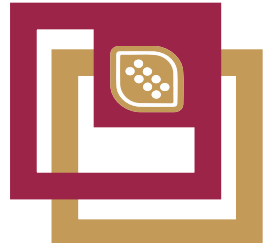
Submódulo II: Diseño y programación en robótica: C6PR

Semestre: 6to. **Turno:** Matutino/Vespertino **Periodo:** 2025 – 2026B



Submódulo	Momento	Tiempo (minutos)	Conocimientos	Semana	Fecha inicio	Observaciones
SUBMÓDULO II. DISEÑO Y PROGRAMACIÓN EN ROBÓTICA	Apertura	50 min 50 min 250 min	Encuadre. Evaluación diagnóstica. Robótica. • Definición. • Fases	8	23 – 27 de marzo de 2026	Evaluación sumativa
	Desarrollo	100 min 250 min	• Fases. • Tipos	9	13 – 17 de abril de 2026	17 de abril reunión de academia.
		350 min	Diseño de robot. • Mecánica.	10	20 – 24 de abril de 2026	
		350 min	• Eléctrica. • Electrónica.	11	27 de abril – 01 de mayo de 2026	Suspensión oficial: 1 de mayo de 2026
		350 min	Software para programación de robot.	12	04 – 08 de mayo de 2026	Suspensión oficial: 05 de mayo de 2026
		350 min	Diseño y desarrollo de robot.	13	11 – 15 de mayo de 2026	Suspensión oficial: 15 de mayo de 2026
		350 min	Diseño y desarrollo de robot. Situación didáctica.	14	18 – 22 de mayo de 2026	Evaluación intrasemestral.
		350 min	Diseño y desarrollo de robot. Situación didáctica.	15	25 – 29 de mayo de 2026	
		350 min	Diseño y desarrollo de robot. Situación didáctica.	16	01 – 05 de junio de 2026	





Criterios de Evaluación



Situación didáctica:

Automatizando Mi Creatividad

30%

Actividades:

20%

Actividad 1. Pon a prueba tus conocimientos.

5%

Actividad 2. Que no se te pase un paso.

5%

Actividad 3. Cuestionándome.

5%

Actividad 4. Anatomía del robot.

5%

Prácticas:

30%

Práctica 1. Hola mundo en Arduino.

5%

Práctica 2. ¿Qué me hace falta?

5%

Práctica 3. Semáforo.

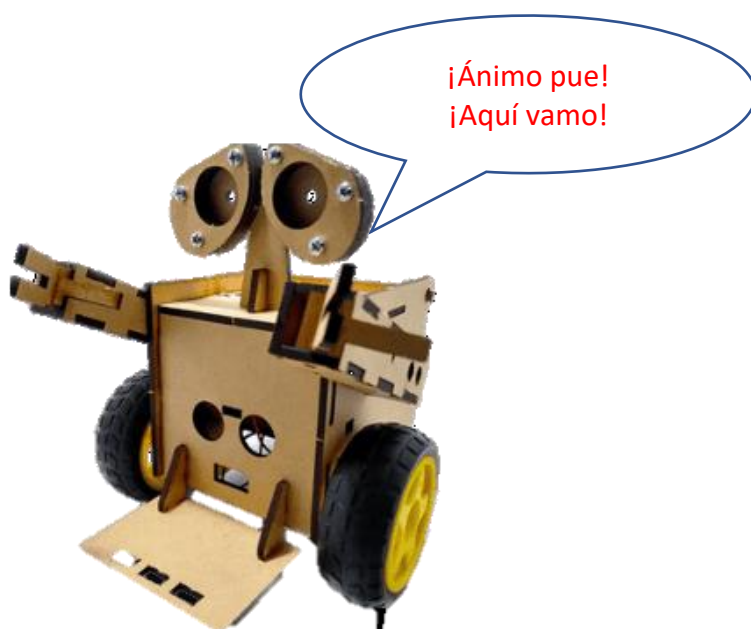
10%

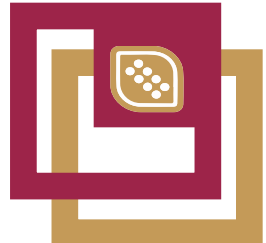
Práctica 4. Cableado de mi dron.

10%

Examen:

20%





Situación Didáctica

AUTOMATIZANDO MI CREATIVIDAD

Propósito:	Desarrollar un prototipo de robot en equipos de 5 integrantes que permita ayudar en la solución de un problema del entorno utilizando el software de programación propuesto, para socializarlo en plenaria.
Contexto:	El desarrollo de las tecnologías junto con la programación, han permitido atender tareas específicas, dando la oportunidad de que los usuarios puedan optimizar tiempo, contabilizar los recursos y notificar sobre situaciones que se tienen pendiente. Cada vez más está avanzado el campo de la robótica, los autómatas y tecnologías que desarrollan actividades propias del ser humano; esto ha causado un impacto en el medio ambiente, la agricultura y tareas propias en el desarrollo social, pero repercuten ecológicamente. Todos estos avances tecnológicos han tenido buen impacto en nuestra sociedad, por lo cual Colegio de Bachilleres de Tabasco, invita a los jóvenes de sexto semestre de la capacitación de desarrollo de software a que hagan su aportación en el diseño y desarrollo de un robot que sea de beneficio para la sociedad considerando el equilibrio ambiental y ecológico.
Conflicto cognitivo	Menciona el software que consideras en el diseño y programación de robots. ¿Cuáles son las fases de desarrollo para el diseño de un robot? Menciona los tipos de robots y ¿Qué formas pueden tomar los robots? ¿Qué aplicaciones en tu contexto puedes darle a la robótica?





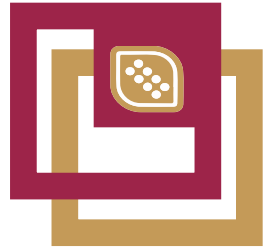
Evaluación Diagnóstica

Instrucciones: Lee cuidadosamente los siguientes cuestionamientos y selecciona en base a tus conocimientos la respuesta que consideres correcta.



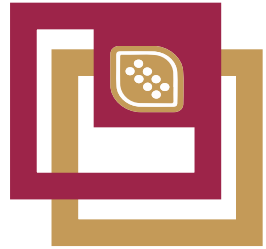
- ¿Es una entidad virtual o mecánica artificial, por lo general un sistema () electromecánico que, por su apariencia o sus movimientos, ofrece la sensación de tener un propósito propio?
A) Máquina B) Objeto C) Robot D) Tecnología ambiental
- C++, Java, Python, TinkerCad son algunos de los lenguajes de programación que usan () para la...
A) Programación de Base de Datos B) Programación de Robot C) Programación de aplicaciones D) Programación Web
- Ciencia que reúne diferentes campos tecnológicos, con el objetivo de diseñar () máquinas robotizadas capaces de realizar diferentes tareas automatizadas en función de la capacidad de su software.
A) Diseño de Sistemas B) Tecnologías de la comunicación C) Informática D) Robótica
- Es uno de los elementos que debe tener un robot para desplazar o mover las partes () que la integran.
A) Articulaciones B) Sensores C) Motor D) Control de voz
- ¿Cuál es una de las ventajas al desarrollar robots? ()
A) Sustituir al ser humano B) Realizar tareas peligrosas para el ser humano C) Para evitar saturación de trabajo D) Proporcionar diversión
- ¿Qué disciplinas de conocimiento emplea la robótica? ()
A) Lenguaje, mecánica y electrónica B) Mecánica, matemáticas y lenguaje C) Electrónica, lenguaje e informática D) Mecánica, electrónica e informática
- Aplicación online que incluye herramientas de software de Autodesk que permite a () los principiantes crear modelos 3D.
A) Java B) C++ C) Tinkercad D) HTML
- Consiste en utilizar software para crear una representación matemática de un () objeto o forma tridimensional.
A) Modelado 3D B) Programación secuencial C) Desarrollo de aplicaciones D) Sitios Web
- Son los componentes electrónicos que permiten dar una interacción con el robot ()
A) Capacitores B) Sensores C) Microprocesador D) Led's





10. Es uno de los elementos en el sistema mecánico: ()
A) Led's B) Sensores C) Actuadores D) Procesadores
11. Esta composición se encarga del diseño de la interconexión de un robot por medio de componentes electrónicos: ()
A) Mecánica B) Eléctrica c) Electromecánica D) Programación
12. Se utiliza más a menudo en la investigación de la robótica para asegurar que los algoritmos de control avanzado funcionan correctamente antes de pasarlos a un robot real ()
A) Diseño de Sistemas B) Tecnologías de la Información y de la Comunicación C) Informática D) Simulación
13. Es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios lenguajes. ()
A) Entorno de Desarrollo Integrado B) Lenguaje de programación C) Desarrollo de Software D) Control de aplicaciones
14. Herramienta dentro de Arduino IDE, o Bitbloq, plataforma Web, que son entornos diseñados para hacer programación gráfica en lugar de escrita. ()
A) Visual Studio Code B) ArduBlock C) Estructuras de control D) Entorno de Desarrollo Integrado
15. Es una aplicación web gratuita y fácil de usar que proporciona a la nueva generación de diseñadores e ingenieros las habilidades básicas para la innovación: diseño 3D, electrónica y codificación. ()
A) Visual Studio Code B) CorelDraw C) MIT App Inventor D) Tinkercad





Lectura No 1. Origen de la Robótica

Instrucciones: Realiza la Lectura 1. “Origen de la Robótica” subrayando las ideas principales, y al finalizar realiza la Actividad 1 – “Pon a prueba tus conocimientos”



LECTURA 1. “ORIGEN DE LA ROBÓTICA”



INTRODUCCIÓN A LA ROBÓTICA



¿Crees que el término robótica es reciente, pues no. El **origen de la robótica** se remonta a miles de años atrás, esto sucedió cuando la robótica no era reconocida como ciencia y los robots eran considerados como autómatas (Objeto mecánico que realiza una función una vez que se activa).

Por ejemplo, los antiguos egipcios unieron brazos mecánicos a las estatuas de sus dioses; los griegos construyeron estatuas que operaban con sistemas hidráulicos, los cuales eran utilizados para fascinar a los adoradores de los templos.



Una de las cosas que ha querido hacer el ser humano es crear vida artificial que le facilite su modo de vivir, satisfaciendo necesidades desde lo básico hasta lo más peligroso. Actualmente hay investigadores del área que están desarrollando robots que realizan actividades semejantes a la del ser humano.

El concepto de “robot” no es de ahora, fue usada por primera vez en el año 1921, por el escritor checo Karel Capek. Su origen proviene de la palabra esclava robota, que se refiere al trabajo realizado de manera forzada. George Devol, fue **quién inventó**

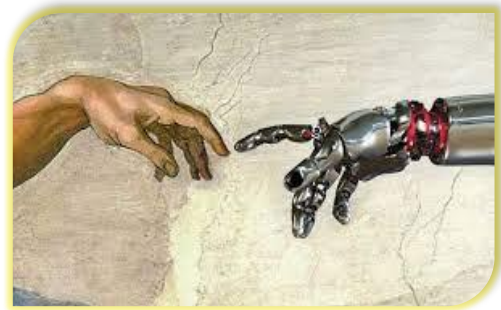
la Robótica Industrial, y patentó en 1948 un manipulador programable. Además, junto a Joseph F. Engelberger, fundó Unimation, la primera empresa de robótica de la historia.





Hoy en día si miramos a nuestro alrededor vemos muchos objetos mecánicos que realizan actividades para las cuales fueron diseñadas, permitiendo al ser humano resolver de manera rápida las necesidades que se les presenta. Viendo esto de lo básico, pero si exploramos otras áreas de investigación y conocimiento, estos avances tecnológicos han permitido desarrollar robots que realicen procesos automatizados a grandes escalas, indagar más allá de nuestro mundo, entre algunos ejemplos. La robótica ha estado en los grandes sucesos de la humanidad, contribuyendo en la obtención de nuevo conocimiento, así como apoyando en tareas difíciles para el hombre.

Lo que en un momento vimos en las películas de ciencia ficción, algunos expertos en el área están acercándose para crear Robots que hagan lo que en un momento fue inimaginable.



ROBOT

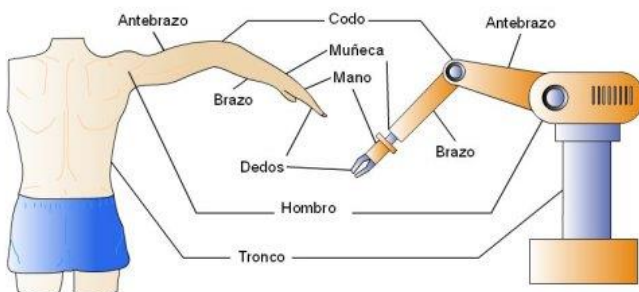


Antes de entrar a la programación de un robot, es importante que quede entendido que un **Robot** es una entidad virtual o mecánica artificial, capaz de realizar varias funciones o tareas complejas, manipular objetos y realizar automáticamente operaciones, incluyendo diferentes tipos de movimientos, en respuesta a su entorno.



Programación de un Robot

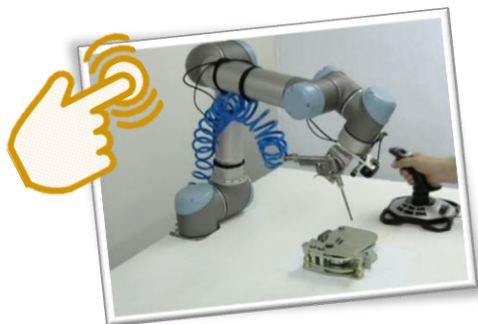
Ahora sí para el proceso de programación de un robot consiste en introducir en su sistema de control las instrucciones necesarias para que desempeñe las tareas para las que ha sido diseñado. Existen varios procedimientos de programación de robots.



Programación guiada o directa

El operario interviene guiando manualmente al robot, y hace que este vaya describiendo los movimientos y trace las trayectorias necesarias para cumplir su función.





Cada uno de los movimientos realizados se va almacenando en la memoria del robot, de forma que podrán ser repetidos posteriormente, ya sin intervención humana. En este tipo de programación guiada o directa es necesario disponer del propio robot para la elaboración del programa.

En muchas ocasiones se utiliza un sistema de guiado en forma de joystick para mover las articulaciones del robot. Es un tipo de programación sencilla, pero con el inconveniente de que no tiene en cuenta las variaciones del entorno.

Programación textual o indirecta

En este caso no es necesaria la presencia del robot para realizar el programa, puesto que este se lleva a cabo en un lenguaje informático. El programa consiste en un texto formado por un conjunto de instrucciones; cuando el programa sea grabado en la memoria del robot, este realizará las acciones indicadas en el mismo.



Este tipo de programación permite realizar operaciones más complejas y con mayor grado de precisión. Además, presenta la ventaja de que es posible establecer relaciones entre el robot y su entorno. Para ello basta con introducir en el programa los datos procedentes de los sensores de forma que el robot actúe en consonancia con los mismos, tal y como ocurre en los denominados «robots inteligentes».

A su vez, la programación textual puede ser de dos tipos: explícita y especificativa.

- La **programación textual explícita** se corresponde con los llamados lenguajes estructurados.



Consiste en programar de forma secuenciada y estructurada el conjunto de acciones que debe realizar el robot para llevar a cabo la tarea encomendada. En dichas instrucciones pueden introducirse también las características del medio.

- La **programación textual especificativa** está más en consonancia con los lenguajes de programación orientados a objetos. En este caso, el programa gira en torno a los elementos manipulados por el robot y las acciones que ha de realizar con ellos, teniendo en cuenta el ámbito en el que se desarrollan dichas acciones.





¿Cómo se programa un robot?

Para programar un robot se sigue un proceso semejante al de la elaboración de un programa informático destinado a cualquier otra aplicación. La mayoría de los **lenguajes de programación** están escritos en texto normal, lo que es fácil de entender para los humanos.



Los programas se compilan en código fuente para que un procesador los ejecute (o código de bytes, para que una máquina virtual los ejecute).



Primero será necesario establecer el que permita al robot llevar a cabo las que ha sido diseñado, tras lo cual se

algoritmo idóneo tareas para las traducirá dicho

algoritmo en un lenguaje de programación inteligible por el sistema de control del robot. Dicho lenguaje debe permitir especificar de forma clara y sencilla las tareas que debe realizar el robot.

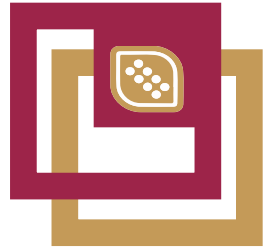
Los lenguajes de programación habituales (PASCAL, LOGO, VISUAL BASIC, C...) son lenguajes de propósito general, es decir, pueden emplearse para cualquier tipo de aplicación y, en particular, en la programación de robots.



La mayoría de **los robots industriales se programan** guiando un robot de un punto a otro a través de las fases de una operación, con cada punto almacenado en el sistema de control del robot.

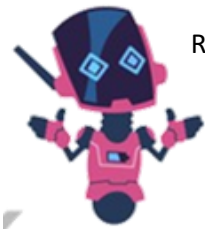
En los robots industriales no existe un lenguaje de programación generalizado. De hecho, en la actualidad existen multitud de lenguajes destinados a la programación de robots industriales, puesto que en la mayoría de los casos los propios fabricantes desarrollan el lenguaje destinado a su robot concreto. Algunos ejemplos son: VAL, RCL, AL, MAPLE, RAPT, LAMA, STRIPS... Suelen estar diseñados para programar un modelo de robot particular y no pueden emplearse en otros tipos de robots.





Hay numerosos lenguajes de programación disponibles, por ejemplo, **C/C++**, **Java**, **Fortran**, **Python**, etc. El lenguaje más popular en robótica es probablemente C/C++ (C++ es un sucesor del lenguaje C orientado a objetos).

Python también es muy popular debido a su uso en el periodo de pruebas de las máquinas y también porque puede ser usado para desarrollar paquetes ROS, Sistema Operativo de Robots (ROS) que no es propiamente un lenguaje de programación.



ROS es más bien un **middleware**, algo así como un «framework» de menor nivel basado en un sistema operativo existente. El principal sistema operativo soportado por ROS es **Ubuntu**. Tienes que instalar ROS en tu sistema operativo para poder usarlo.

Se trata de un sistema flexible para programar el software de los robots. Es una colección de herramientas, bibliotecas y protocolos que tienen como objetivo simplificar la tarea de crear un comportamiento robótico complejo y robusto en una amplia variedad de plataformas robóticas. En consecuencia, ROS se diseñó desde cero para fomentar la colaboración en el desarrollo de **software de robótica**.



Los subprogramas que se escriben (nodos) pueden ser escritos en cualquier idioma. Así, una aplicación puede tener un nodo escrito en Python comunicándose con un nodo escrito en C++.

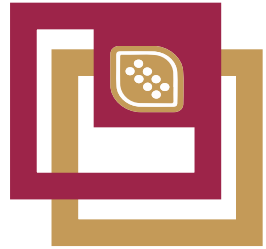


Checa dato



https://youtu.be/BGOuv4gCk_Y





BIBLIOGRAFÍA

A. (2022, 4 enero). *Cómo funciona la Robótica Industrial*. aula21 | Formación para la Industria. <https://www.cursosaula21.com/como-funciona-la-robotica-industrial/#:%7E:text=Hay%20numerosos%20lenguajes%20de%20programaci%C3%B3n,lenguaje%20C%20orientado%20a%20objetos>).

Historia de la robótica. (2021, 23 noviembre). BBVA.CH. <https://www.bbva.ch/noticia/historia-de-la-robotica/>

I. (2017, 14 junio). 2. *Lenguajes de Control Robótico*. Inteligencia Artificial. <https://inteligenciartificialmca.wordpress.com/2017/05/26/lenguajes-de-control-robotico/>

Turmero, P. (2014, 27 octubre). *El lenguaje V+*. Monografias.com. <https://www.monografias.com/trabajos102/lenguaje-v/lenguaje-v>

Technology Bit Perú. (2017, 28 noviembre). *Historia de la Robotica - Cursos y Talleres de Robótica en Lima*. Talleres Robotica Educativa en Lima. <https://www.roboticaeducativa.pe/2017/10/20/historia-de-la-robotica/#:%7E:text=Su%20origen%20proviene%20de%20la,de%20rob%C3%B3tica%20de%20la%20historia>.



Actividad 1. Pon a Prueba Tus Conocimientos



Instrucciones: Después de haber analizado la lectura 1. “Origen de la Robótica”, organízate en equipo de 3 a 5 integrantes y completa correctamente las siguientes oraciones y desarrolla tu investigación.



1.- El _____ se remonta a miles de años atrás, esto sucedió cuando la robótica no era reconocida como ciencia y los robots eran considerados como _____ (Objeto mecánico que realiza una función una vez que se activa).

2.- Un _____ es una _____ o mecánica artificial, capaz de realizar varias funciones o tareas complejas, manipular objetos y realizar automáticamente operaciones, incluyendo diferentes tipos de movimientos, en respuesta a su entorno.

3.- El proceso de programación de un robot consiste en introducir en su _____ las instrucciones necesarias para que desempeñe las tareas para las que ha sido diseñado.

4.- En este tipo de programación _____ es necesario disponer del propio robot para la elaboración del programa. En muchas ocasiones se utiliza un sistema de guiado en forma de _____ para mover las articulaciones del robot.

5.- El programa consiste en un texto formado por un _____; cuando el programa sea grabado en la memoria del robot, este realizará las acciones indicadas en el mismo.

Banco de respuesta:

joystick	guiada o directa	Conjunto de instrucciones	origen de la robótica
----------	------------------	---------------------------	-----------------------

autómatas	Sistema de control	Robot	Entidad virtual
-----------	--------------------	-------	-----------------





Una vez completadas las afirmaciones anteriores, desarrolla una presentación interactiva donde presentas al menos dos aplicaciones o programas diseñados al aprendizaje de robótica. Considera los siguientes puntos para tu presentación:

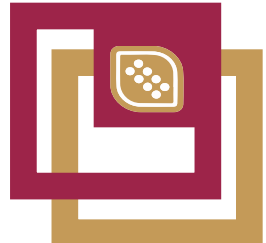
- ◆ Introducción a la robótica.
- ◆ Programación de robots
- ◆ 1er aplicación o programa diseñado para aprender robótica:
 - A. Nombre de la aplicación o programa.
 - B. Características o descripción.
 - C. Lenguaje utilizado o programación utilizada.
- ◆ 2da aplicación o programa diseñado para aprender robótica:
 - A. Nombre de la aplicación o programa.
 - B. Características o descripción.
 - C. Lenguaje utilizado o programación utilizada.



Consulta la web



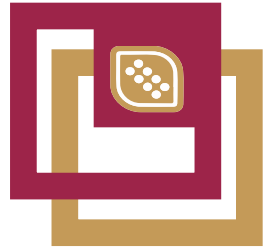
<https://www.bejob.com/10-aplicaciones-para-ensenar-programacion-robotica-en-colegios/>



INSTRUMENTO DE EVALUACIÓN
LISTA DE COTEJO 1
Actividad 1: "Pon A Prueba Tus Conocimientos"

DATOS GENERALES						
Nombre(s) del alumno(s)				Matrícula(s)		
Producto: Presentación Electrónica				Fecha		
Submódulo 2: Diseño y Programación en Robótica				Periodo: 2025 – 2026B		
Nombre del docente				Firma del docente		
CRITERIO	INDICADORES	VALOR OBTENIDO		PONDERACIÓN	CALIF	OBSERVACIONES Y/O SUGERENCIAS DE MEJORA
		SÍ	NO			
1	Presenta sus datos para identificarse.			1		
2	Incluye los solicitado para la 1er aplicación o software.			2		
3	Incluye los solicitado para la 2da aplicación o software.			2		
4	Presenta las ilustraciones de las aplicaciones o software.			1		
5	No presenta faltas ortográficas.			1		
6	Responde correctamente las afirmaciones de la actividad.			2		
7	Entrega en tiempo y forma.			1		
CALIFICACIÓN						





Lectura No 2. Elementos en la Robótica

Instrucciones: Realiza la Lectura 2. "Elementos en la Robótica" subrayando las ideas principales, y al finalizar realiza la Actividad 2 – "Pon a prueba tus conocimientos"



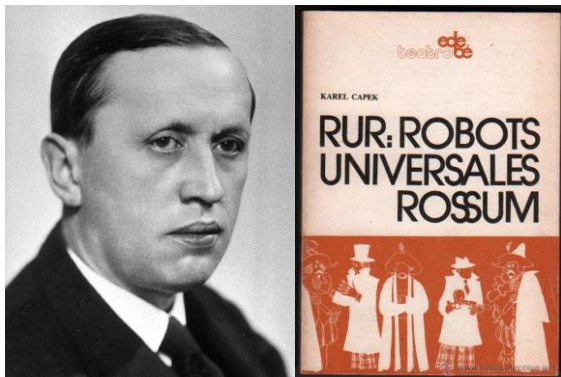
LECTURA 2. "ELEMENTOS EN LA ROBÓTICA"



PRINCIPIOS EN LA ROBÓTICA

La palabra robot se encuentra emparentada con la palabra gótica *arbi*, que quiere decir **herencia** y, también, con la palabra gótica *arbaiths*, que quiere decir **trabajo**, faena, pena, apuro. Una palabra alemana afín es *arbeit*, que quiere decir **trabajo**. El equivalente en eslavo antiguo es *rabota*; en checo y en polaco, *robota* quiere decir servidumbre, trabajo forzado.

Así, la palabra robot significa "trabajo forzado" y tiene connotaciones de esclavitud.



El que acuñó el término robot fue el polaco Karel Capek (1890-1938) en 1921. En efecto, en su obra de teatro intitulada Los Robots Universales de Rossum, utiliza la palabra robot, para designar a ciertas creaciones, muy semejantes a seres humanos, que en la medida en que se vuelven más inteligentes son capaces de exterminar a la especie humana.



De acuerdo con el diccionario de la Universidad de Oxford, un robot es" ... un aparato mecánico que se parece y hace el trabajo de un ser humano ... "

Otro significado para la palabra robot lo tenemos en la definición del Instituto de Robots de América, " ... un manipulador reprogramable y multifuncional concebido para transportar materiales, piezas, herramientas o sistemas especializados, con movimientos variados y programados, con la finalidad de ejecutar tareas diversas ... "

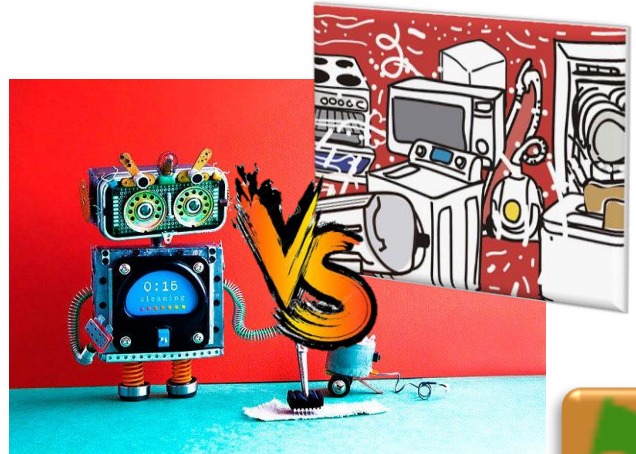




¿Qué es y qué no es un ROBOT?

Una de las definiciones más aceptadas considera a los robots como dispositivos mecánicos capaces de realizar tareas que podrían ser desempeñadas o no por seres humanos. Bajo esta definición, por ejemplo, está claro que las maquinas que intervienen en una cadena de ensamblaje de carros son robots. Como se puede apreciar, la inteligencia no es requisito para que una maquina o dispositivo sea clasificado como robot.

Pero en esta definición, también son robots las máquinas de siglos pasados construidas con madera o hierro, al igual que las maquinas inteligentes que surgieron en Japón, Estados Unidos y Europa a inicio de los 90's. A estos últimos se les conceden capacidades perceptivas (vista, oído y tacto, principalmente) reducidas en comparación con la capacidad de percepción de los seres vivos, pero son capaces de interactuar con su entorno.



La batidora, un electrodoméstico, realiza una función que podría desempeñar un ser humano. Como ellas, se pueden encontrar a la venta “robots de cocina” cuya única capacidad es triturar alimento. Pues bien: por definición la batidora es un robot, aun cuando no sea más que una herramienta eléctrica.

Sin embargo, en sociedades muy desarrolladas la batidora no se considera ya un robot, porque al compararla con los robots humanoides se convierte en algo muy simple. Se trata entonces de un problema de clasificación de contraste social. En el año 1985 nadie ponía en duda que una batidora fuera un robot de cocina, pero ahora resulta irónico decir que tanto una batidora como un humanoide son robots.

Un carro no es capaz de ir de Tabasco a la Ciudad de México sin conductor. ¿Será entonces un robot? Según el diccionario, un carro es una máquina, y aunque no sea capaz de realizar tal viaje por sí mismo, gracias a su sistema informático, electrónico y mecánico, sí es considerado como robot.



La palabra robótica fue acuñada por primera vez en el año 1921 por el escritor checo Karel Capek popularizado por el escritor ruso-americano de ciencia ficción Isaac Asimov de quién se deben las 3 leyes de la robótica:



Primera ley: “Un robot no le hará daño a un ser humano o, por inacción, permitir que un ser humano sufra daño.”

Segunda ley: “Un robot debe obedecer las órdenes dadas por los seres humanos, excepto si estas órdenes entran en conflicto con la primera ley.”



Tercera ley: “Un robot debe proteger su propia existencia en la medida que su protección no entre en conflicto con la primera y segunda ley.”

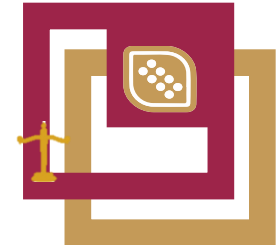


Las 6 leyes de la robótica de la Unión Europea

Aunque Isaac Asimov ya planteó las tres leyes principales, la Unión Europea se ha visto en la obligación de comenzar a crear una normativa debido a la rapidez con la que la inteligencia artificial irrumpe en nuestras vidas.



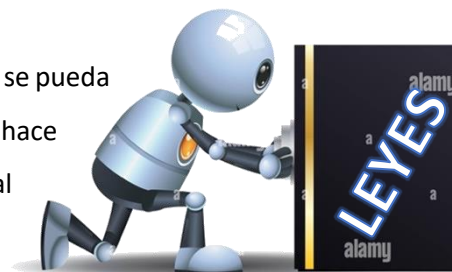
El Parlamento ha decidido regular nuestro futuro y para ello propone seis leyes que normalicen la convivencia con los robots. Todavía no son definitivas, ya que esta normativa tiene que pasar por el **filtro de la Comisión Europea**, pero su **principal objetivo es reducir el impacto** que supondrán la implantación en la sociedad de estas máquinas, ya que uno de los mayores efectos será la pérdida de trabajo.



1. Los robots deberán tener un interruptor de emergencia



Lo primero que tenemos que controlar es cualquier situación que se pueda dar de peligro. La evolución que están teniendo estas máquinas hace que la Inteligencia Artificial pueda aprender debido al entrenamiento, por lo que cualquier situación de peligro que se pueda producir debe ser evitada con un interruptor de emergencia.



2. Los robots no podrán hacer daño a los seres humanos

Obviamente, está fundamentada en la principal **Ley de Asimov** y completa la primera propuesta por la Unión Europea. La robótica siempre tiene que ser pensada para ayudar y proteger a los humanos, por lo que no se permitirá la creación de máquinas cuyos fines sean destruir o dañar a las personas.



3. No podrán generarse relaciones emocionales con los robots

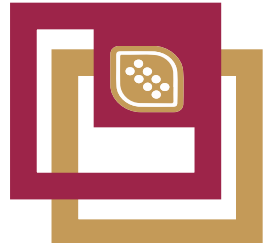
¿Has visto “**El hombre bicentenario**”? En la película, un robot interpretado por Robin Williams permanece colaborando con la misma familia durante décadas, dando como resultado estrechas relaciones personales entre las máquinas y las personas. Estos vínculos emocionales son la trama central de todo el film. Pues bien, la Unión Europea considera que esto no debe ser así, por lo que prohibirá crear cualquier relación emocional con los robots. El objetivo de esta norma es recordar a los humanos que las inteligencias artificiales no son capaces de sentir amor por nosotros, por lo que nosotros no debemos actuar diferente.



4. Los que sean más grandes deberán tener un seguro obligatorio

No todas las máquinas serán iguales ni tendrán las mismas características. Tampoco podrán realizar todas las mismas actividades. Es evidente que algunos de ellos tendrán más riesgo de causar ciertos daños materiales que otros robots. Por ello, la Unión Europea obligará a los dueños de los robots de mayor tamaño, que son considerados los que más riesgo tienen de provocar daños en el entorno, la contratación de un seguro obligatorio como podría suceder con los autos. Así, cualquier desperfecto podrá ser arreglado sin complicaciones.





5. Derechos y obligaciones para los robots



¿Cómo serán llamados ante las administraciones públicas? Parece que “Personas electrónicas” es el nombre escogido para su clasificación legal.

Siempre que se utilice la palabra “persona” implica que existirán una serie de derechos que de otra manera ni se pensarían. El debate se encuentra en cuáles serán sus derechos u obligaciones, pero según todo lo que estamos viendo, serán los robots los que asumirán las consecuencias de sus actos junto a sus propietarios o creadores.

6. Tendrán la obligación de pagar impuestos

Principalmente, se utilizarán para reducir el impacto de los robots en el empleo humano. Su entrada en el mercado laboral obligará a muchas empresas a despedir a sus empleados, ya que los robots realizarán el mismo trabajo a un precio menor. Por ello, el Parlamento Europeo considera necesario que las máquinas tributen en la seguridad social, para poder subvencionar las ayudas que se darán a estas personas que han sido despedidas.

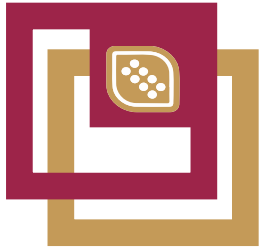


Seis son las primeras normas que se imponen a los robots. Pero, probablemente, no serán las últimas. Una nueva era de convivencia con la robótica ha comenzado, obligándonos a todos a adaptarnos a este nuevo entorno que se nos presenta. Renovarse o morir. No hay más.

Entonces...

Con todo esto, podemos considerar a la robótica como la ciencia y tecnología para diseñar y construir máquinas capaces de imitar tareas humanas llegando a recrear inteligencia. Hoy en día es una ciencia interdisciplinaria porque conjuga conocimientos de la ingeniería mecánica, ingeniería eléctrica, electrónica, ciencias de la computación, llegando hasta áreas como la biología.





Robótica: Tipos y Fases



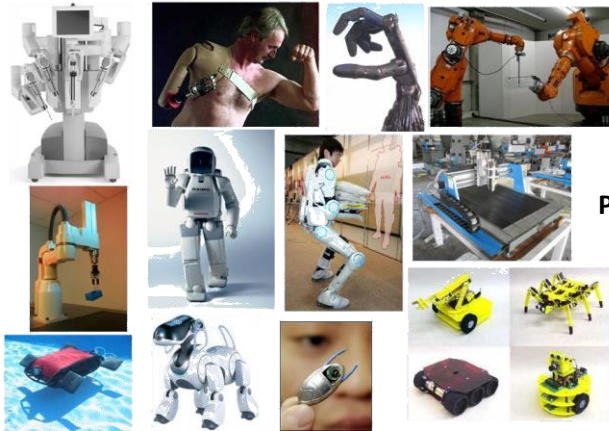
La robótica es la ciencia que estudia a los robots, y como tal, **concentra las distintas disciplinas necesarias para diseñar y fabricarlos**. Así, reúne conocimientos de distintas ramas de la ingeniería, de la electrónica, de la física, la informática, la mecánica, la animatrónica y otras áreas del saber semejantes.



Su cometido, claramente, es desarrollar los diferentes aspectos de un robot funcional: su autonomía e inteligencia propia, su resistencia y capacidad de operatividad, su programación y mecanismos de control.

TIPOS DE ROBOT

¿Qué tipos de robots hay?

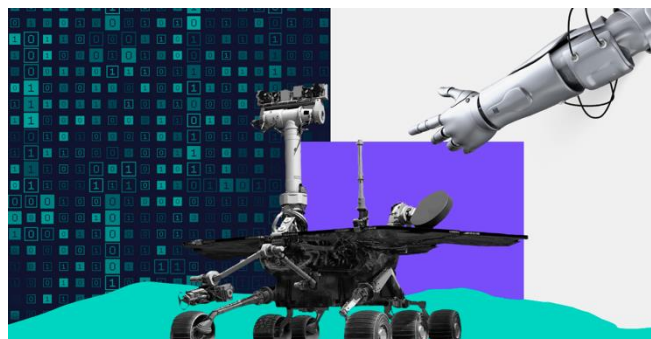


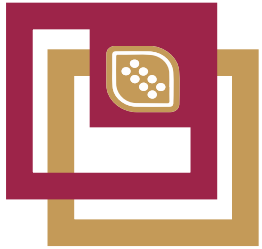
Los robots se clasifican generalmente en base a su pertenencia a las diferentes generaciones de robots construidos, que son:

Primera generación. Robots multifuncionales con un sistema simple de control, manual, de secuencia fija o secuencia variable.

Segunda generación. Robots de aprendizaje, que repiten secuencias de movimientos previamente ejecutadas por operadores humanos.

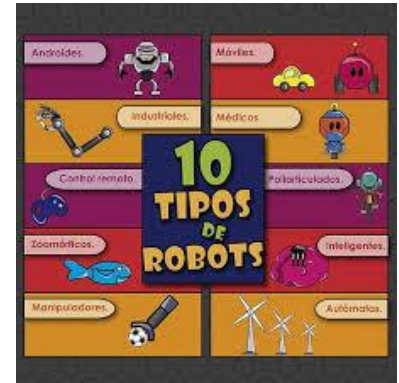
Tercera generación. Robots de control sensorizado, controlados por algún tipo de programa (software) que envía las señales al cuerpo robotizado para llevar a cabo determinadas tareas mecánicas.





Otra forma de clasificación responde a la **estructura del robot**, pudiendo hablar de robots:

- ◆ **Poliarticulados.** Tienen muchas piezas móviles.
- ◆ **Móviles.** Son de tipo rodante o automotor.
- ◆ **Zoomórficos.** Imitan la forma de algunos animales.
- ◆ **Antropomórficos.** Imitan la forma del ser humano.



Pueden ser de los siguientes tipos:

a) Andróides

Los andróides son robots u organismo sintético antropomorfo que, además de imitar la apariencia humana, simula algunos aspectos de su conducta de manera autónoma.



b) Móviles

Los robots móviles están provistos de patas, ruedas u orugas que los capacitan para desplazarse de acuerdo con su programación. Elaboran la información que reciben a través de sus propios sistemas de sensores y se emplean en determinado tipo de instalaciones industriales, sobre todo para el transporte de mercancías en cadenas de producción y almacenes. También se utilizan robots de este tipo para la investigación en lugares de difícil acceso o muy distantes, como es el caso de la exploración espacial y de las investigaciones o rescates submarinos.



c) Industriales

Los robots industriales están destinados a realizar de forma automática determinados procesos de fabricación o manipulación. Algunos de ellos son electrodomésticos capaces de realizar varias operaciones distintas de forma simultánea o consecutiva, sin necesidad de intervención humana.





Los robots industriales son muy utilizados en empresas que requieren automatizar sus procesos. Los países como Japón y Estados Unidos lideran la fabricación y consumo de robots industriales siendo Japón el número uno.

La Asociación Japonesa de Robótica Industrial (JIRA): Los robots son "dispositivos capaces de moverse de modo flexible análogo al que poseen los organismos vivos, con o sin funciones intelectuales, permitiendo operaciones en respuesta a las órdenes humanas".

El Instituto de Robótica de América (RIA): Un robot industrial es "un manipulador multifuncional y reprogramable diseñado para desplazar materiales, componentes, herramientas o dispositivos especializados por medio de movimientos programados variables con el fin de realizar tareas diversas".



d) Médicos



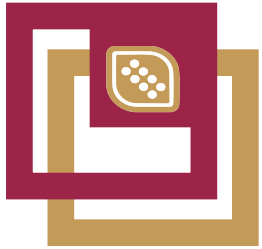
Los robots médicos son, fundamentalmente, los que trabajan de forma autónoma o semiautónoma para realizar tareas útiles para el bienestar de las personas.



e) Teleoperadores

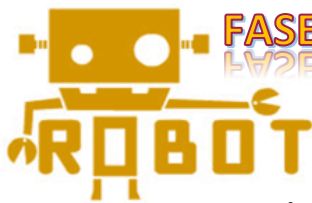
Los teleoperadores se controlan remotamente por un operador humano. Cuando pueden ser considerados robots se les llama "telerobots". Cualquiera que sea su clase, los teleoperadores son generalmente muy sofisticados y extremadamente útiles en entornos peligrosos tales como residuos químicos y desactivación de bombas.





f) Híbridos

Es un organismo cibernético en forma de un robot controlado por una computadora consistente en elementos biológicos y electrónicos. Los elementos biológicos son neuronas de roedor unidas a un chip de computadora.



FASES EN EL DESARROLLO DEL

Las fases **mecánica, eléctrica, electrónica e informática** se refieren a los **subsistemas o disciplinas fundamentales** que, al combinarse, forman cualquier **sistema robótico o de automatización** complejo.

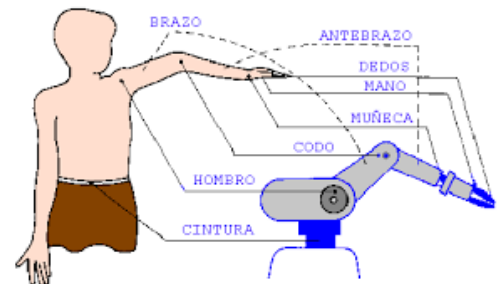
Aunque a menudo se mencionan como "fases de construcción", representan mejor los **componentes funcionales** del robot, cada uno a cargo de una tarea específica.

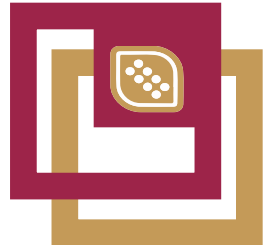


1. Fase Mecánica

Esta fase se refiere al **cuerpo físico, la estructura y el movimiento** del robot.

- **Función:** Proporcionar soporte estructural, movilidad y capacidad para realizar trabajo físico.
- **Componentes Clave:**
 - **Estructura:** El chasis, el armazón o el esqueleto del robot (hecho de metal, plástico, etc.).
 - **Cinemática:** Las articulaciones (uniones) y los eslabones que permiten el movimiento.
 - **Transmisión:** Componentes auxiliares como engranajes, poleas, y ejes que transfieren el movimiento desde los motores a las articulaciones.
 - **Actuadores:** Los elementos que generan el movimiento, como los motores, que convierten la energía eléctrica en fuerza física (movimiento mecánico).
- **Disciplina Clave:** Ingeniería Mecánica (análisis de cargas, cinemática, diseño estructural).

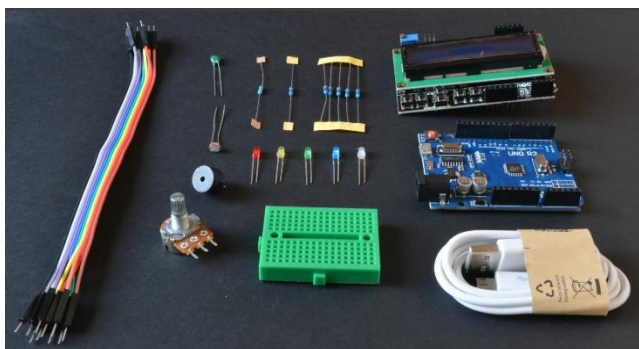




2. Fase Eléctrica

Esta fase se ocupa de la **generación, almacenamiento y distribución de la energía** que alimenta todo el sistema.

- **Función:** Suministrar la potencia necesaria a todos los componentes, especialmente a los actuadores (motores), que suelen requerir alta potencia.
- **Componentes Clave:**
 - **Fuente de Energía:** Baterías, fuentes de alimentación externas, o generadores.

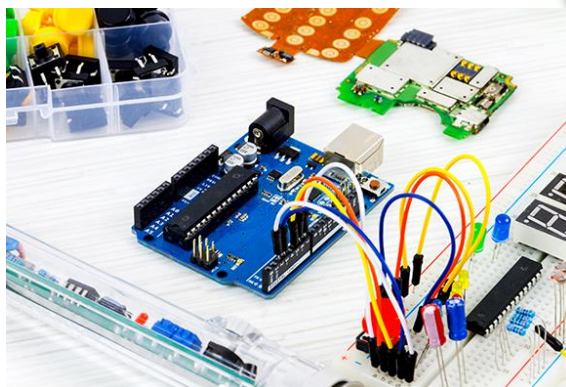


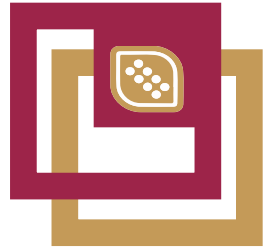
- **Cableado y Circuitos de Potencia:** Conductores y fusibles diseñados para manejar y distribuir grandes cantidades de corriente (alto voltaje/amperaje).
- **Propósito:** Proporcionar la "materia prima" (la energía eléctrica) para el funcionamiento.

3. Fase Electrónica

Esta fase se centra en el **control y procesamiento de señales de baja potencia** utilizadas para la lógica y la comunicación.

- **Función:** Controlar y manipular el flujo de corriente para procesar, transmitir y almacenar información. Actúa como la "inteligencia" o el "cerebro" del robot.
- **Componentes Clave:**
 - **Circuitos de Control:** Conjunto de circuitos que indican a cada parte del robot cómo y cuándo funcionar.
 - **Sensores:** Componentes electrónicos que detectan magnitudes físicas (luz, distancia, presión) y las convierten en señales eléctricas de control.
 - **Componentes Activos:** Transistores, diodos, resistencias y circuitos integrados (microcontroladores y microprocesadores) diseñados para manejar señales de baja potencia.
- **Distinción Clave con Eléctrica:** Mientras que la eléctrica maneja la **energía de potencia**, la electrónica se enfoca en el **control de la señal** para realizar tareas específicas.





4. Fase Informática

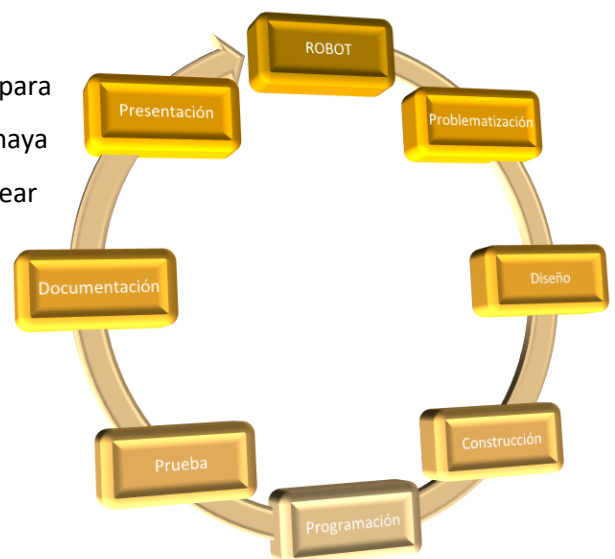
Esta fase se refiere al **software, los algoritmos y la lógica de programación** que dan vida al robot.

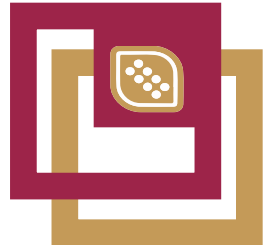
- **Función:** Ejecutar el *software* que controla el funcionamiento del robot. Es responsable de procesar los datos sensoriales, tomar decisiones y enviar instrucciones a los actuadores.
- **Componentes Clave:**
 - **Software de Control:** Algoritmos (ej. cinemática inversa, control PID) y programas que definen la lógica de movimiento y toma de decisiones.
 - **Lenguaje de Programación:** El código específico (ej. Python, C++) que permite vincular al robot con su usuario y entorno.
 - **Unidad Central de Procesamiento (CPU/Microcontrolador):** El componente que ejecuta el *software*.
- **Proceso Clave:**
 - **Captura de datos** (sensores)
 - **Procesamiento** (algoritmos)
 - **Toma de decisiones**
 - **Ejecución de acciones** (actuadores).



La robótica, por lo tanto, es una disciplina que **combina** estas cuatro áreas: **Mecánica** (el cuerpo), **Eléctrica** (la potencia), **Electrónica** (el sistema nervioso y el control) e **Informática** (el cerebro/inteligencia).

Para desarrollar un robot se debe realizar un estudio para saber si es viable o no en la realidad. Después que se haya definido que tipo de robot se va a diseñar, se puede emplear las siguientes fases para el desarrollo:



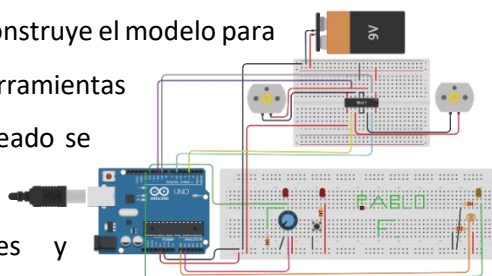


1. **Problemática:** se plantea un reto. Imagina y piensa en aquello que le gustaría realizar y las posibilidades a su alcance, por lo que investiga y explora. Se hace hincapié en la relevancia de imaginar dispositivos que favorezcan la resolución de problemas concretos. La creatividad cumple una función muy importante.

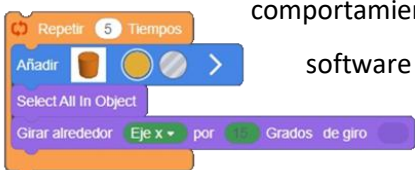


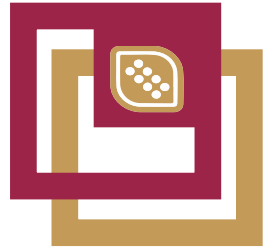
2. **Diseño:** se diseñan modelos de posibles soluciones a la problemática, dará origen al desarrollo de una maqueta, modelo, diseño. En esta fase es necesario incluir no solamente las características del dispositivo a crear, sino también cómo se realizaría su construcción. Es aquí donde se desarrollan ideas en proyectos.

3. **Construcción:** siguiendo el diseño planteado, se construye el modelo para darle movimiento haciendo uso de herramientas electrónicas y software. En base al diseño planteado se empezará a construir una solución al problema, valiéndose de piezas, conectores, sensores y conexiones.



4. **Programación:** en la que a través del software se programan los movimientos y comportamientos del modelo. Actividad basada en la utilización de un software de fácil uso que permite programar los movimientos y el comportamiento en general del modelo robótico. Deberá seguirse una secuencia ordenada de instrucciones, ingresarlos en el subsistema de control e ingresar el programa en dicho subsistema.

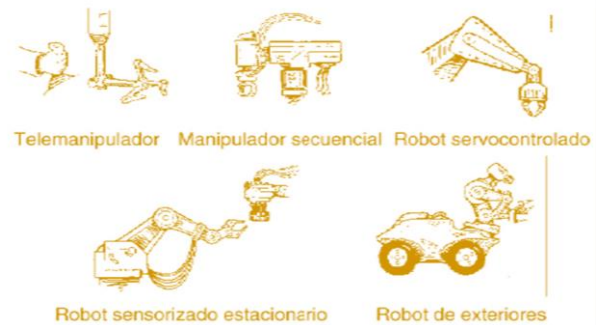


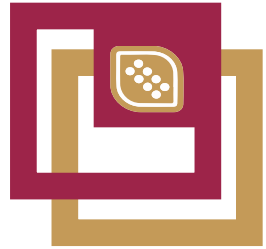


5. **Prueba:** en base al diseño, se construye el modelo haciendo uso del kit de robótica para darle movimiento. Verificar visualmente que el modelo implementado funciona. Comprobar que su funcionamiento cumple con un conjunto de especificaciones.

6. **Documentación:** se recopilan evidencias que prueban la funcionalidad del diseño a través de mano alzada, softwares especializados o procesador de texto/gráficos. Una vez que se ha probado el modelo que funciona como lo hemos diseñado, entonces se debe documentar el trabajo desarrollado.

7. **Presentación:** se presenta y explica el prototipo creado como alternativa de solución al problema planteado. La creación se comparte, acción que puede realizarse tanto de manera presencial o a través de medios telemáticos.





BIBLIOGRAFÍA

Historia de la robótica. (2021, 23 noviembre). BBVA.CH. <https://www.bbva.ch/noticia/historia-de-la-robotica/>

Introducción a la Robótica. (2004). Control y Robótica. http://platea.pntic.mec.es/vgonzale/cyr_0204/cyr_01/robotica/intro.htm

M., A. (2020, 20 abril). *TinkerCAD: ¡Te contamos todo lo que necesitas saber!* 3Dnatives. <https://www.3dnatives.com/es/tinkercad-software-200420202/#!>

Programación de Robot. (2011, 6 diciembre). Automática Industrial. <https://automaticaindustrial.wordpress.com/robotica/programacion-de-robot/#:%7E:text=El%20proceso%20de%20programaci%C3%B3n%20de,procedimientos%20de%20programaci%C3%B3n%20de%20robots.>

"Robótica". Autor: Equipo editorial, Etecé. De: Argentina. Para: Concepto.de. Disponible en: <https://concepto.de/robotica/>. Última edición: 3 de agosto de 2020. Consultado: 31 de enero de 2022

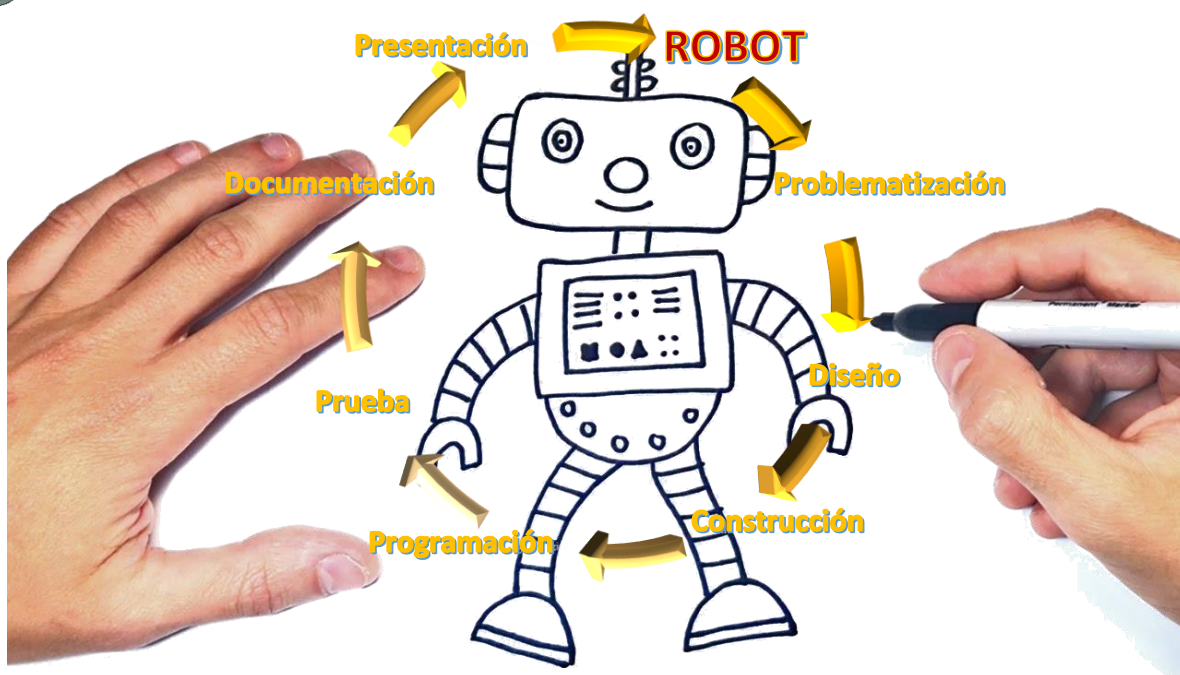
Technology Bit Perú. (2017, 28 noviembre). *Historia de la Robotica - Cursos y Talleres de Robótica en Lima.* Talleres Robotica Educativa en Lima. <https://www.roboticaeducativa.pe/2017/10/20/historia-de-la-robotica/#:%7E:text=Su%20origen%20proviene%20de%20la,de%20rob%C3%B3tica%20de%20la%20historia.>



Actividad 2. Que No Se Te Pase Un Paso



Instrucciones: Después de haber analizado la lectura 2. “Elementos en la Robótica”, especialmente “Robótica: Tipos y Fases” organízate en equipo de 3 a 5 integrantes para desarrollar de manera digital o física, una propuesta de robot considerando su contexto. Elige un tipo de robot y demuestra las fases.



En tu propuesta, mencionen el tipo de robot que permite dar respuesta a la problematización; además de ejemplificar las fases a través de una redacción o esquematización.

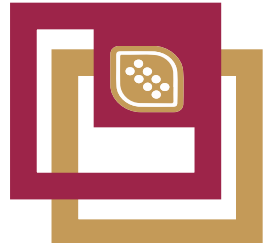
Consideren mencionar una propuesta de lenguajes o programas a partir de la fase de programación.

En el caso de la fase de prueba pueden describir cómo la realizarían, asimismo para la documentación y la presentación.

Finalmente expongan su propuesta en el grupo.

RELAJATE Y PON A VOLAR TU
IMAGINACION
IMAGINACION







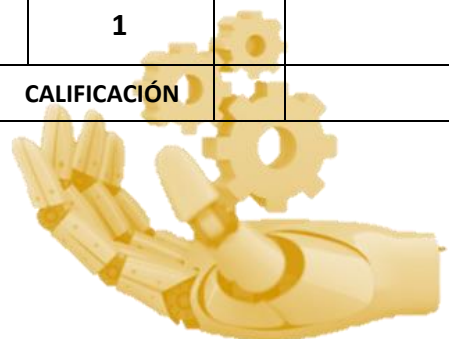
INSTRUMENTO DE EVALUACIÓN

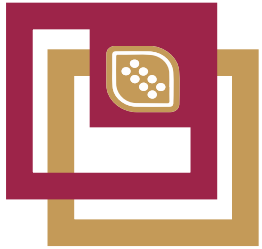
LISTA DE COTEJO 2

Actividad 2: "Que No Se Te Pase Un Paso"



DATOS GENERALES						
Nombre(s) del alumno(s)				Matrícula(s)		
Producto: Propuesta de robot				Fecha		
Submódulo 2: Diseño y Programación en Robótica				Periodo: 2025 – 2026B		
Nombre del docente				Firma del docente		
CRITERIO	INDICADORES	VALOR OBTENIDO		PONDERACIÓN	CALIF	OBSERVACIONES Y/O SUGERENCIAS DE MEJORA
		SÍ	NO			
1	Presenta sus datos para identificarse.			1		
2	El tipo de robot propuesto es consistente a la problematización.			2		
3	Expresa todas las fases en el desarrollo de un robot.			2		
4	Presenta las ilustraciones acordes al robot propuesto.			1		
5	No presenta faltas ortográficas.			1		
6	Expone correctamente todos los elementos utilizados.			2		
7	Entrega en tiempo y forma.			1		
CALIFICACIÓN						





Lectura No 3. La Mecánica del Robot

Instrucciones: Realiza la Lectura 3. “La Mecánica del Robot” subrayando las ideas principales, y al finalizar realiza la Actividad 3 – “Cuestionándome”

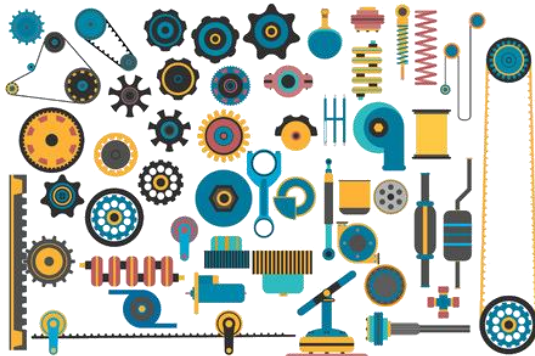


LECTURA 3. “LA MECÁNICA EN EL DISEÑO DE ROBOTS”



El sistema mecánico (ensamble mecánico) está constituido por mecanismos, actuadores (motores), tornillos, engranajes, poleas, ejes (articulaciones, grados de libertad, grados de movilidad) así como el equipo complementario, el cual permitirá la correcta realización de las tareas requeridas. En esta etapa se dota al robot de una estructura sólida, por lo que es necesario hacer buenas conexiones con las articulaciones mecánicas e incorporar motores para que puedan controlarse posteriormente los movimientos del robot, ya sea en forma manual o automática.

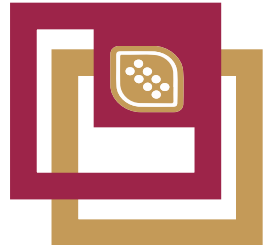
Para el montaje mecánico (etapa mecánica) de un prototipo de robot; se requieren dominar los conceptos



de engranajes, poleas, ejes, articulaciones, grados de libertad, de movilidad, etc. En esta fase se dota al robot de una estructura sólida, por lo que es necesario hacer buenas conexiones con articulaciones mecánicas e incorporar motores para que puedan controlarse posteriormente los movimientos del robot, ya sea en forma manual o automática.

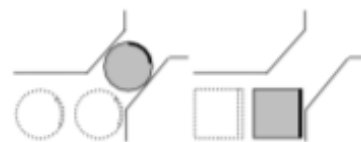


Un robot comprende una estructura física integrada por diferentes partes y subsistemas relacionados, de tal forma que ocasionan el movimiento y funcionamiento del robot, de acuerdo con los diferentes algoritmos empleados.



La estructura mecánica de un robot la integran principalmente **la morfología del robot, los actuadores y los efectores.**

MORFOLOGÍA DEL ROBOT



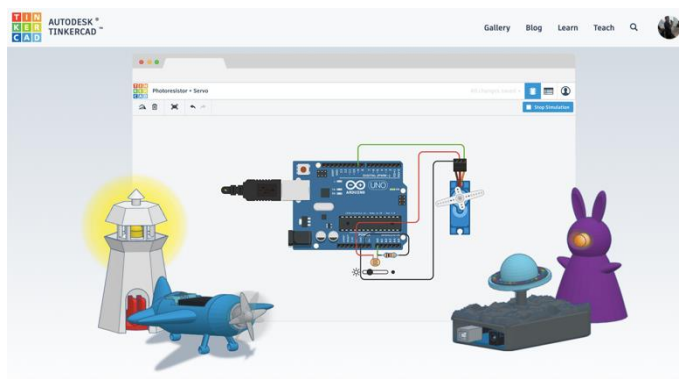
La morfología del robot corresponde a la forma física que presenta el robot con sus actuadores, efectores finales y sensores que son colocados sobre el robot, inclusive la carcasa que

Nombre	Figura	Características	Aplicaciones
Robot Cartesiano		Utiliza tres dispositivos deslizantes perpendiculares entre sí, para generar movimientos de acuerdo a los tres ejes cartesianos X, Y y Z.	- Control numérico. - Máquinas de fresado o dibujo. - Plotter.
Robot Cilíndrico		Se basa en una columna vertical que gira sobre la base. También tiene dos dispositivos deslizantes que pueden generar movimientos sobre los ejes Z e Y.	- Soldadura. - Traslado de material con alta precisión. - Montaje. - Traslado de material punto a punto desde una posición central.
Robot esférico o Polar		Utiliza un brazo telescópico que puede bascular en torno a un eje horizontal. Este eje telescópico está montado sobre una base giratoria. Las articulaciones proporcionan al robot la capacidad de desplazar el brazo en una zona esférica.	- Manipulación en máquinas herramientas. - Soldadura por punto. - Fundición a presión. - Soldadura por gas y por arco.
Robot de brazo articulado		Se trata de una columna que gira sobre la base. El brazo contiene una articulación, pero sólo puede realizar movimientos en un plano. En el extremo del brazo contiene un eje deslizante que se desplaza en el eje Z. El robot más común de este tipo se conoce como robot SCARA.	- Operaciones de ensamblaje. - Pintado en spray. - Máquinas de desbarbado. - Fundición a presión.
Robot antropomórfico		Esta constituido por dos componentes rectos que simulan el brazo o antebrazo humano, sobre una columna giratoria. Estos antebrazos están conectados mediante articulaciones que se asemejan al hombro y al codo.	- Exoesqueletos. - Teleoperación. - Máquinas de desbarbado. - Fundición.

cubre a todo el robot, si este tiene una. Un actuador es un elemento del robot que genera la fuerza necesaria para que se muevan sus efectores. Un efector es un elemento terminal del robot, por ejemplos pinzas, herramientas, ventosas y ganchos, cuya finalidad es afectar o producir alguna modificación en el entorno en el cual se encuentra el robot. La siguiente figura ilustra los sensores actuadores y efectores de un robot. (Velasco, 2006)

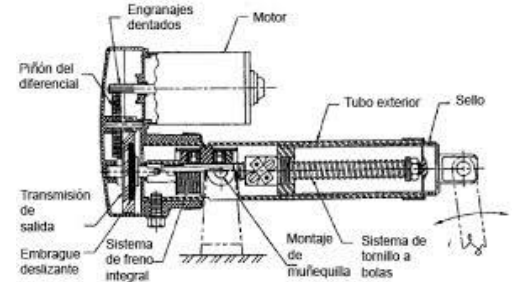


La morfología del robot puede favorecer o perjudicar su desempeño. Por ejemplo, un robot cúbico que requiere pasar por una zona estrecha con esquinas tendrá mayor problema para evitar las esquinas y pasar por el área estrecha, que un robot de estructura cilíndrica. Uno de los problemas de diseño de robots es que se desea contar con robots capaces de desplazarse en cualquier tipo de terreno, por muy complejo que éste sea, por ejemplo, en la exploración de volcanes, otros planetas o zonas de desastre. (Velasco, 2006).



MOTORES O ACTUADORES

Un actuador es un dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en energía mecánica. Los actuadores pueden verse como transductores; por ejemplo, el motor convierte energía eléctrica (se conecta a una fuente de alimentación) en energía mecánica rotacional (movimiento). Recuerdese que un transductor es cualquier elemento que convierte una forma de energía en otra forma de energía. Los elementos que conforman un actuador son los siguientes:



- Sistema de accionamiento: es el encargado de producir el movimiento.
- Sistema de transmisión: es el encargado de transmitir el movimiento del actuador a otros elementos.
- Sistema reductor: encargado de adecuar el torque y la velocidad del actuador a los valores requeridos.
- Sistema de control: encargado de enviar las órdenes al actuador para que se mueva de cierta manera.

Existen diferentes tipos de actuadores:

- Hidráulicos
- Neumáticos
- Eléctricos

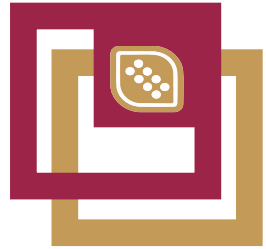


PRINCIPIOS EN LA ROBÓTICA

Los actuadores neumáticos transforman la energía acumulada en el aire comprimido en trabajo mecánico de movimiento circular o movimiento rectilíneo. Los actuadores neumáticos se clasifican en dos grandes grupos: cilindros y motores neumáticos.

Los actuadores hidráulicos obtienen su energía de un fluido a presión, generalmente algún tipo de aceite mineral. Los actuadores hidráulicos se clasifican en tres grandes grupos: cilindros hidráulicos, motores y válvulas hidráulicos. La principal ventaja de estos actuadores es su relación potencia/peso.





Los **actuadores eléctricos** transforman la energía eléctrica en energía mecánica rotacional. Podemos encontrar tres grandes grupos de actuadores eléctricos: motores de corriente continua, motores de corriente alterna y motores de paso a paso.

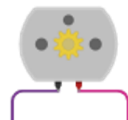


TIPOS DE MOTORES ELÉCTRICOS

Básicamente, cualquier objeto electrónico que hayas visto o conoces, tiene alguna pieza en movimiento. Por ejemplo: impresoras, coches de juguete, cepillos de dientes eléctricos, etc.; todos, contienen motores. Los hay de muchos tipos, pero principalmente encontrarás tres tipos de motores:



Motores DC (corriente continua): Si necesitas que algo gire, pero sin precisión, este es tu motor. Para encontrar un motor DC en la vida real, busca el ventilador que se encuentra dentro de tu ordenador. También puedes encontrar uno muy pequeño en tu teléfono móvil. El motor DC es el que hace que tu teléfono vibre, haciendo girar un bloque de metal cuyo peso está distribuido de manera no uniforme.



SERVOMOTOR



PASO A PASO



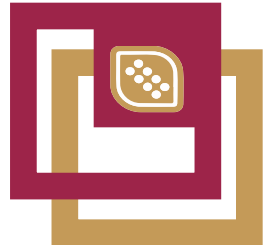
Un motor DC puede funcionar libremente en ambas direcciones, es muy fácil controlar su velocidad, pero no su posición. Tampoco es sencillo hacerlo parar de forma precisa. Viene con dos cables: alimentación y tierra.

MOTOR UNIVERSAL

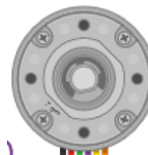


Con varios transistores dispuestos en un puente H, puedes incluso controlar la dirección sin tener que desconectar el motor.

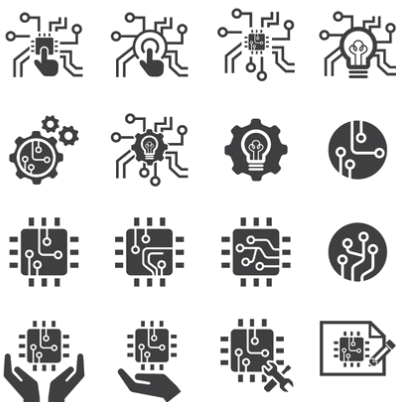




Motores paso a paso: Los motores paso a paso se pueden encontrar en cualquier objeto electrónico donde prima la precisión, como escáneres e impresoras. Un motor paso a paso puede, a diferencia del motor DC, ser muy preciso tanto en posición como en velocidad. La rotación completa de los motores paso a paso se divide en pasos equitativos y puedes controlar la parada del motor en cada uno de estos pasos. Los pasos se miden en grados, normalmente 1.8, 3.6 o 7.2. Cuanto más pequeños sean los pasos, más preciso será. Esto hace que sea muy útil cuando se necesita un posicionamiento repetido. Sin embargo, el motor paso a paso nunca será muy rápido. Un motor paso a paso tiene 4 o más cables. Por lo general, necesitas más de 5 voltios para alimentar un motor paso a paso.



Servomotores: Los servomotores son ampliamente utilizados en robótica y en radio-control. Tienen tres cables: uno para la energía, uno para tierra y otro para



controlarlos. Hay dos tipos de servomotores: rotación estándar y rotación continua. El estándar puede girar 180 grados y puede ser controlado como el motor paso a paso a una posición precisa. La rotación continua puede, al igual que el motor DC, rotar en ambas direcciones, no tan rápido; pero puedes controlar tanto la velocidad como la dirección sin tener que utilizar transistores (Tecnologías Creativas Castilla, 2020).



Blocks + Text

- Output
- Control
- Input
- Math
- Notation
- Variables

```

1 void setup()
2 {
3   pinMode(13, OUTPUT);
4 }
5
6 void loop()
7 {
8   digitalWrite(13, HIGH);
9   delay(1000); // Wait for 1000 millisecond(s)
10  digitalWrite(13, LOW);
11  delay(1000); // Wait for 1000 millisecond(s)
12 }

```

```

1 void setup()
2 {
3   pinMode(2, OUTPUT);
4   pinMode(8, OUTPUT);
5   pinMode(12, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(2, HIGH);
11  delay(5000); // Wait for 5000 millisecond(s)
12  digitalWrite(2, LOW);
13  delay(100); // Wait for 100 millisecond(s)
14  digitalWrite(8, HIGH);
15  delay(1000); // Wait for 1000 millisecond(s)
16  digitalWrite(8, LOW);
17  delay(100); // Wait for 100 millisecond(s)
18  digitalWrite(12, HIGH);
19  delay(5000); // Wait for 5000 millisecond(s)
20  digitalWrite(12, LOW);
21  delay(100); // Wait for 100 millisecond(s)
22 }

```

```

set pin 2 to HIGH
wait 5 secs
set pin 2 to LOW
wait 0.10 secs
set pin 8 to HIGH
wait 1 secs
set pin 8 to LOW
wait 0.10 secs
set pin 12 to HIGH
wait 5 secs
set pin 12 to LOW
wait 0.10 secs

```



Actividad 3. Cuestionándome



Instrucciones: Después de haber analizado la lectura 3. “La Mecánica del Robot”; elabora un cuestionario de mínimo 5 preguntas, evitando preguntas de respuestas abiertas. para desarrollar de manera digital, una vez elaborado tu propuesta de cuestionario; procede a elaborarlo utilizando una aplicación web.



Una vez concluida la lectura, elabora un cuestionario de mínimo 5 preguntas, evitando preguntas de respuesta abierta. Una vez que tengas tu



cuestionario procede a elaborarlo utilizando alguna herramienta en línea como Forms, Genially, Socrative, Kahoot!, etc.

Google Forms



Recuerda que es importante seleccionar las respuestas las respuestas correctas en los cuestionarios para que puedan ser calificados de forma automática.

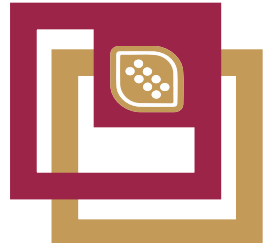


Una vez completado el cuestionario digital, comparte con algunos de tus compañeros el enlace y registra sus resultados en tus apuntes, con la finalidad de identificar que pregunta fue la más compleja de responder.



Comparte con tu docente los resultados obtenidos.







INSTRUMENTO DE EVALUACIÓN
LISTA DE COTEJO 3
Actividad 3: "Cuestionándome"



DATOS GENERALES						
Nombre(s) del alumno(s)				Matrícula(s)		
Producto: Cuestionario digital y respuestas				Fecha		
Submódulo 2: Diseño y Programación en Robótica				Periodo: 2025 – 2026B		
Nombre del docente				Firma del docente		
CRITERIO	INDICADORES	VALOR OBTENIDO		PONDERACIÓN	CALIF	OBSERVACIONES Y/O SUGERENCIAS DE MEJORA
		SÍ	NO			
1	Presenta sus datos para identificarse.			1		
2	El cuestionario fue diseñado en una aplicación web.			2		
3	Presenta al menos 5 preguntas bien elaboradas y un enlace para responderlo.			2		
4	Aplicó el cuestionario a sus compañeros del grupo.			1		
5	No presenta faltas ortográficas.			1		
6	Comparte los resultados de la aplicación del cuestionario.			2		
7	Entrega en tiempo y forma.			1		
CALIFICACIÓN						



Actividad Guiada: Jugando al Mecánico



Instrucciones: Considera todo lo aprendido hasta este momento, consulta el contenido audiovisual propuesto a continuación y; organizados en parejas, elaboren un robot con materiales disponibles en el hogar.



Recuerda que estamos aprendiendo los elementos básicos en la mecánica.

Propuesta 1



<https://youtu.be/lxAwXLXDvXM>

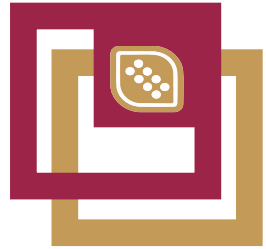


Propuesta 2



<https://youtu.be/v44WFPTk75c>





Lectura No 4. Energía y Fuerza de Robot

Instrucciones: Realiza la Lectura 4. “Energía y Fuerza de Robot” subrayando las ideas principales, y al finalizar realiza la Actividad 4 – “Infografía”



LECTURA 4. “ENERGÍA Y FUERZA DE ROBOT”



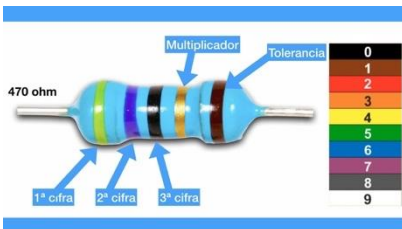
ELÉCTRICA

Otra de las etapas para el diseño de la interconexión de un robot por medio de componentes electrónicos, en la que se incluye diversos elementos de electrónica como diseño de circuitos, realización de cálculos de voltajes, potencias, resistencias, circuitos, etc. Aunque el tema es amplio y extenso, para nuestra materia nos enfocaremos en únicamente algunos componentes de electrónica básicos.

Circuitos

Los circuitos, como toda materia, están compuestos por átomos. Los átomos se componen de partículas subatómicas:

- Protones (con carga eléctrica positiva)
- Neutrones (sin carga)
- Electrones (con carga negativa)



Los átomos permanecen enlazados entre sí por fuerzas de atracción entre el núcleo y los electrones de un átomo en su capa exterior.

Cuando los átomos en un circuito son influenciados por la tensión, comienzan a reformarse y sus componentes ejercen un potencial de atracción conocida como una diferencia de potencial. Los electrones libres mutuamente atraídos avanzan hacia los protones y crean un flujo de electrones (corriente). Cualquier material en el circuito que restrinja este flujo se considera como **resistencia**.

¿Qué es la resistencia?

La resistencia es una medida de la oposición al flujo de corriente en un circuito eléctrico. La resistencia se mide en ohmios, que se simbolizan con la letra griega omega (Ω). Se denominaron ohmios en honor a Georg Simon Ohm (1784-1854), un físico alemán que estudió la relación entre voltaje, corriente y resistencia. Se le atribuye la formulación de la ley de Ohm. Todos los materiales resisten en cierta medida el flujo de corriente. Se incluyen en una de dos amplias categorías:



- **Conductores:** materiales que ofrecen muy poca resistencia, donde los electrones pueden moverse fácilmente. Ejemplos: plata, cobre, oro y aluminio.
- **Aislantes:** materiales que presentan alta resistencia y restringen el flujo de electrones. Ejemplos: goma, papel, vidrio, madera y plástico. El alambre de oro sirve como un excelente conductor.

Normalmente, se toman las mediciones de resistencia para indicar las características de un componente o un circuito.

- **Cuanto mayor sea la resistencia,** menor será el flujo de corriente. Si es anormalmente alta, una causa posible (entre muchas) podrían ser los conductores dañados por el fuego o la corrosión. Todos los conductores emiten cierto grado de calor, por lo que el sobrecalentamiento es un problema que a menudo se asocia con la resistencia.
- **Cuanto menor sea la resistencia,** mayor será el flujo de corriente.

Muchos componentes, tales como los elementos de calefacción y las resistencias, tienen un valor de resistencia fijo. Estos valores se imprimen a menudo en las placas de identificación de los componentes o en los manuales de referencia.

Cuando se indica una tolerancia, el valor de resistencia debe encontrarse dentro de la gama de la resistencia especificada. Cualquier cambio significativo en un valor de resistencia fijo generalmente indica un problema.

1.5 Ω Ohms

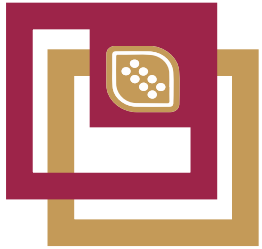
codigodecolor.com

COLOR	BANDA 1 (VALOR)	BANDA 2 (VALOR)	BANDA 3 (VALOR)	BANDA 4 (MULTIPLICADOR)	BANDA 5 (TOLERANCIA)
NEGRO	0	0	0	x1	+/- 1%
MARRON	1	1	1	x10	+/- 2%
ROJO	2	2	2	x100	
NARANJA	3	3	3	x1000	
AMARILLO	4	4	4	X10,000	
VERDE	5	5	5	X100,000	+/- 0.5%
AZUL	6	6	6	X1,000,000	+/- 0.25%
VIOLETA	7	7	7	X10,000,000	+/- 0.10%
GRIS	8	8	8		+/- 0.05%
BLANCO	9	9	9		
DORADO				0.1	+/- 5%
PLATEADO				0.01	+/- 10%
NINGUNO					+/- 20%

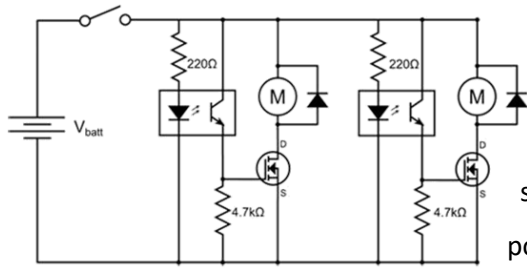
La "resistencia" puede parecer negativa, pero en la electricidad puede usarse beneficiosamente.

Ejemplos: la corriente debe luchar para fluir a través de las bobinas pequeñas de una tostadora, lo suficiente como para generar el calor que tuesta el pan. Las bombillas incandescentes antiguas fuerzan la

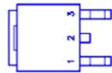
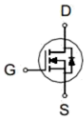
corriente para que fluya a través de filamentos muy delgados y generen luz. Algunos elementos de electrónica en un circuito soportan cierta intensidad de corriente por lo que es necesario utilizar resistencias para regular el flujo en cada componente. Para ello se utilizan resistencias con determinada capacidad, para saber la capacidad de las resistencias existe un código de colores que permite identificar la cantidad de ohms de la resistencia dependiendo los códigos de colores en ella.



ELECTRÓNICA



P0903BDG



1. GATE
2. DRAIN
3. SOURCE

La electrónica es la parte del robot en donde se distingue el microcontrolador que es el cerebro los sensores que son los que permiten detectar el entorno y los drivers de potencia que son los circuitos.

La **electrónica de control** al ser el cerebro de nuestro robot se encargará de:

- Generar señales de control de motores.
- Leer el estado de los sensores.
- Tomar las decisiones encargados de suministrar la energía necesaria a los motores para hacerlos rodar.

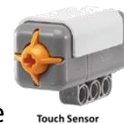
Para la **electrónica de potencia** los motores necesitan una corriente elevada que no puede ser suministrada por la electrónica de control. Esto quiere decir que los pines de salida de los micros anteriores no son capaces de hacer girar a un motor. Por ello la electrónica de potencia se encarga de adaptar las señales de control a los niveles efectivos para mover los motores. También puede hacer parte del trabajo del CPU y simplificar los programas de control. Existe el circuito Puente H que permite controlar la dirección de giro de un motor separando las señales de control del CPU y las entradas del motor.



Electrónica de Sensores

De forma similar a los seres vivos, los sensores facilitan la información necesaria para que los robots interpreten el mundo real. Todo robot debe tener al menos un sensor con el que interactuar. La mayoría de los sistemas robóticos incluyen al menos sensores de obstáculos (bumpers) y algún sensor de guiado por infrarrojos o ultrasonidos. Los sensores avanzados, además de detectar algo, son capaces de reportar una medida de lo detectado, como puede ser un sensor de temperatura, o un medidor de distancias ultrasónico por ejemplo (Robótica, 2021).

NXT Sensors



Touch Sensor



Light Sensor

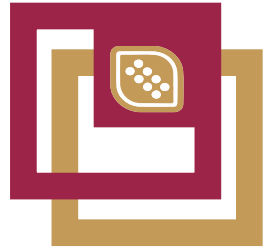


Sound Sensor

Ultrasonic Sensor

Un robot por su propia definición, debe ser una máquina capaz de relacionarse con el entorno que le rodea e interactuar con el mismo para tomar decisiones. Los seres humanos dispones de un sistema sensorial que nos permite interrelacionarnos con lo que nos rodea.





De la misma manera, los robots precisan de dicho sistema sensorial para adaptar sus movimientos y las acciones a realizar en función de los estímulos o respuestas que reciba de todo aquello que le rodea. Para conseguir esto, se utilizan diversos tipos de sensores que permiten dotar al robot de información precisa y al momento del entorno que le rodea.

Cómo funciona un sensor

Un sensor es un dispositivo diseñado para percibir información externa de una magnitud física o química y transformarla en un valor electrónico que sea posible introducir al circuito de control de un microcontrolador, como puede ser Arduino, de modo que el robot sea capaz de cuantificarla y reaccionar en consecuencia en base a su programación.

Tipos de sensores

Un sensor consta de algún elemento sensible a una magnitud física o química, como por ejemplo la intensidad o color de la luz, la temperatura, la presión, el magnetismo, la humedad,... y debe ser capaz, por su propias características, o por medio de dispositivos intermedios, de transformar esa magnitud física en un cambio eléctrico que se pueda alimentar en un circuito que la utilice directamente, o sino en una etapa previa que la condicione (amplificando, filtrando, etc.), para que finalmente se pueda utilizar para el control de un robot.

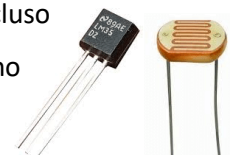


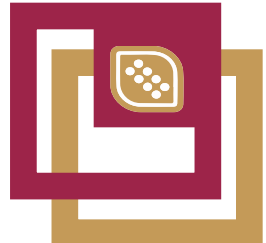
Tipos de Sensor de Velocidad



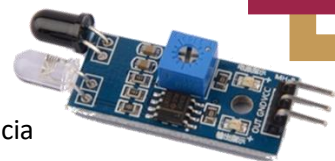
Sensores de velocidad: Son sensores que se utilizan para determinar la velocidad de movimiento de los robots. Estos sensores determinan la velocidad de giro de los motores. Este tipo de sensor se llama tacómetro. A partir de estas medidas, se utilizan actuadores (otro tipo de dispositivos), con el fin de modificar la velocidad de giro de los motores.

Sensores de luz: Dentro de los sensores de luz, existen una amplia gama de dispositivos con diferentes funciones, estamos hablando de fotodiodos, fotorresistores e incluso cámaras de video. Su función es determinar la cantidad de luz existente en el entorno del robot con el fin de actuar en función de los valores obtenidos.

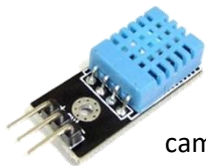
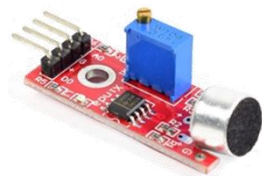




Sensores de distancia: Los sensores de distancia determinan la distancia existente a un objeto. Pueden ser de dos tipos, ya sea por infrarrojos o bien por ultrasonidos. Los basados en ultrasonidos ofrecen información de la distancia mientras que los que funcionan por infrarrojos suelen emplearse para detectar objetos en ciertos rangos de distancia y excepcionalmente también dan información de la distancia.



Sensores de sonido: Mediante estos sensores, los robots reaccionarán al sonido. Dentro de esta gama de sensores se engloban micrófonos y captadores piezoeléctricos. Estos sensores, además de usarse para captar sonido, se pueden usar también para medir distancias, al emitir el robot un sonido y medir el tiempo de rebote de dicho sonido en el obstáculo.



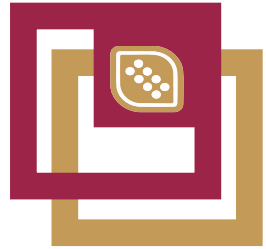
Sensores de temperatura y humedad: Los sensores de temperatura se pueden utilizar para diferentes aplicaciones en robótica. La más básica para activar dispositivos ante cambios de temperatura. Existen diferentes tipos de sensores de temperatura como son termistores, piro sensores o termorresistencias. En función del uso que se vaya a dar, se usarán de un tipo u otro. De la misma forma existen sensores de humedad que suelen ser capacitivos o resistivos para detectar cambios de humedad en el ambiente.

DHT11 pin	
1	VCC
2	DATA
3	NC
4	GND



Además de todos estos sensores que hemos comentado, podemos encontrar muchos más como de magnetismo, de ubicación geográfica, de gravedad o de presión y fuerza. El abanico es muy amplio. Lo que debemos tener claro es que un robot sin sensores es de muy poca utilidad a no ser que tenga una función muy específica que no requiera de sensores (Yañez, 2018).





BIBLIOGRAFÍA

Areatecnologia. (2021). *Areatecnologia*. Obtenido de Tipos de robots:
<https://www.areatecnologia.com/electronica/tipos-de-robots.html>

Control y Robótica para Tecnología. (2004). Control y Robótica.
http://platea.pntic.mec.es/vgonzale/cyr_0204/ctrl_rob/contenidos.htm

Gobierno de Canarias. (s.f.). Obtenido de Ejemplificación. La robótica.:
http://www3.gobiernodecanarias.org/aciisi/cienciasmc/web/u9/ejemplificacion_u9.html

Tecnologías Creativas Castilla. (Mayo de 2020). Creative Technologies Castilla. Obtenido de Arduino
Verkstad Education: <http://ctc-dev.verkstad.cc/es/course-literature/tipos-demotores/>

Robótica, S. (2021). *Super Robótica*. Obtenido de Sensores:
<http://www.superrobotica.com/sensores.htm>

Velasco, M. G. (Febrero de 2006). Diseño e implementación de un Sistema multi-robot con
capacidades de auto-ensamblado. Tonantzintla, Puebla, México: Instituto Nacional de Optica y
Electronica (INAOE).

Yañez, C. (2018). CEAC. Obtenido de La importancia de los sensores en robótica:
<https://www.ceac.es/blog/la-importancia-de-los-sensores-en-robotica>



Actividad 4. Anatomía del Robot



Instrucciones: Después de haber analizado la lectura 4. “Energía y Fuerza de Robot”; elabora una presentación electrónica en la que identifiques al menos 5 elementos fundamentales para la programación en un robot. Considera también la lectura 3. “Mecánica del Robot”.



HAIKU DECK

Después de realizar la lectura, comienza elaborando una infografía en la que representes los elementos que componen la fase eléctrica de la robótica.

Te sugerimos realizar una investigación más detallada sobre algunos otros elementos que pueden incluirse.

Continúa en el diseño de tu presentación electrónica y considera los componentes necesarios para que el robot funcione para el contenido de tu presentación electrónica.

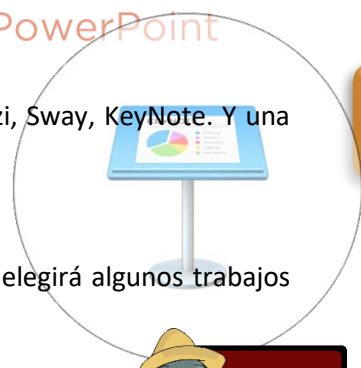


Utiliza herramientas tecnológicas como Genially, Canva, PowerPoint, Prezi, Sway, KeyNote. Y una vez terminada, presenta tu trabajo ante tus compañeros.

Una vez realizada la infografía procede a entregarla a tu docente el cual elegirá algunos trabajos para presentarlos en el aula.



PowerPoint

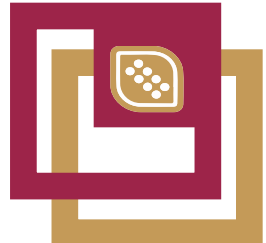


Considera esto:

MAKERSPACE



<https://youtu.be/QUasLDg0D0o>



 INSTRUMENTO DE EVALUACIÓN LISTA DE COTEJO 4 Actividad 4: "Anatomía del Robot" 						
DATOS GENERALES						
Nombre(s) del alumno(s)				Matrícula(s)		
Producto: Presentación electrónica				Fecha		
Submódulo 2: Diseño y Programación en Robótica				Periodo: 2025 – 2026B		
Nombre del docente				Firma del docente		
CRITERIO	INDICADORES	VALOR OBTENIDO		PONDERACIÓN	CALIF	OBSERVACIONES Y/O SUGERENCIAS DE MEJORA
		SÍ	NO			
1	Presenta sus datos para identificarse.			1		
2	Elabora una infografía presentando la fase eléctrica del robot.			2		
3	El contenido de la presentación electrónica muestra una relación directa de los componentes básicos programables y no programables en un robot.			2		
4	Aplicó el animaciones y transiciones a la presentación electrónica.			1		
5	No presenta faltas ortográficas.			1		
6	Comprende y expone correctamente los elementos que presenta.			2		
7	Entrega en tiempo y forma.			1		
CALIFICACIÓN						



Actividad Guiada: Articulando mi brazo



Instrucciones: Considera todo lo aprendido hasta este momento, consulta el contenido audiovisual propuesto a continuación y; organizados en parejas, elaboren un robot con materiales disponibles en el hogar.



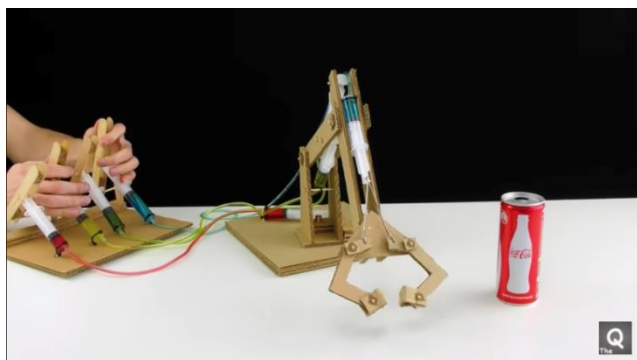
Recuerda que estamos aprendiendo los elementos básicos en la mecánica.

Propuesta 1



<https://youtu.be/ybFy-zyLYco>

Propuesta 2



<https://youtu.be/P2r9U4wkjcc>

Lectura No 5. Programación de Robots

Instrucciones: Realiza el análisis de la siguiente información, posteriormente, realiza la Actividad 5.

Timeline: Evolución de los Videojuegos.



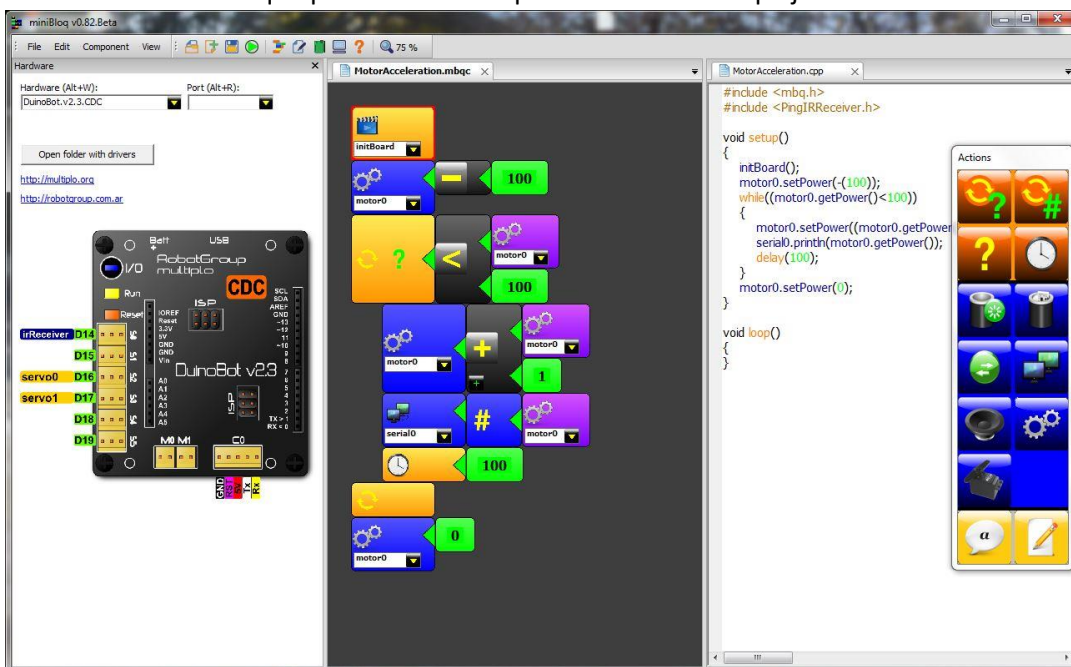
LECTURA 5. "PROGRAMACIÓN DE ROBOTS"

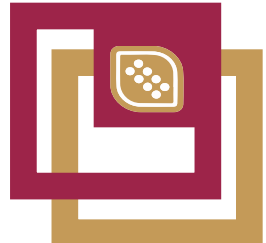


Existen en el mercado diferentes kits para comenzar o avanzar dentro de este campo, algunos sólo permiten su montaje y ejecución, sin embargo, otros los más complejos y flexibles permiten ser programados en distintas plataformas.



En la robótica hay un peso muy importante porque se desea dotar de comportamientos inteligentes a máquinas fabricadas por el ser humano para la realización de labores difíciles o repetitivas. Algunos ejemplos son la producción en cadena, el ensamblado de piezas o la exploración en entornos peligrosos. Las acciones llamadas funciones primitivas al ser combinadas en forma adecuada en una secuencia que permita resolver problemas más complejos.

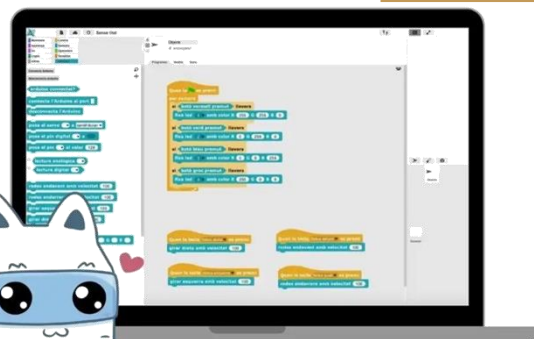




La **programación offline, o simulación**, se utiliza más a menudo en la investigación de la robótica para asegurar que los algoritmos de control avanzado funcionan correctamente antes de pasarlos a un robot real.

Sin embargo, también se utiliza en la industria para reducir el tiempo de inactividad y mejorar la eficiencia.

Puede ser un método particularmente útil para las PYMES, ya que es más probable que los robots se configuren varias veces que en los entornos de producción a gran escala.

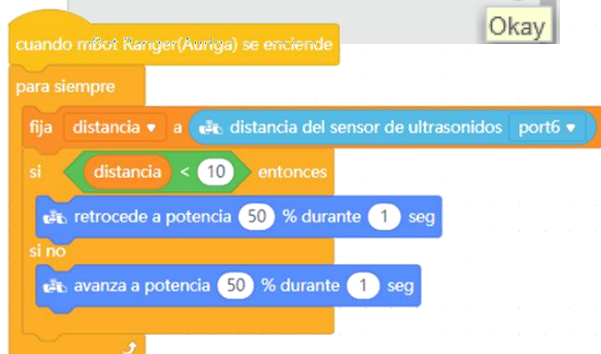


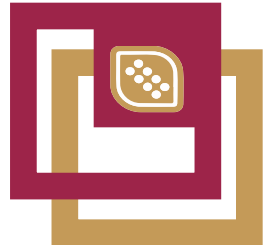
Programar offline significa que esto no interfiere demasiado

con la producción. La programación offline permite programar el robot usando una maqueta virtual del robot y en la tarea. Si el software de simulación es de uso intuitivo, puede ser una forma rápida de probar una idea antes de trasladarla al robot.



Algunos paquetes de simulación modernos son bastante sencillos de usar. La simulación se puede configurar en cuestión de minutos, una vez que se esté familiarizado con el software, utilizando una biblioteca de robots.

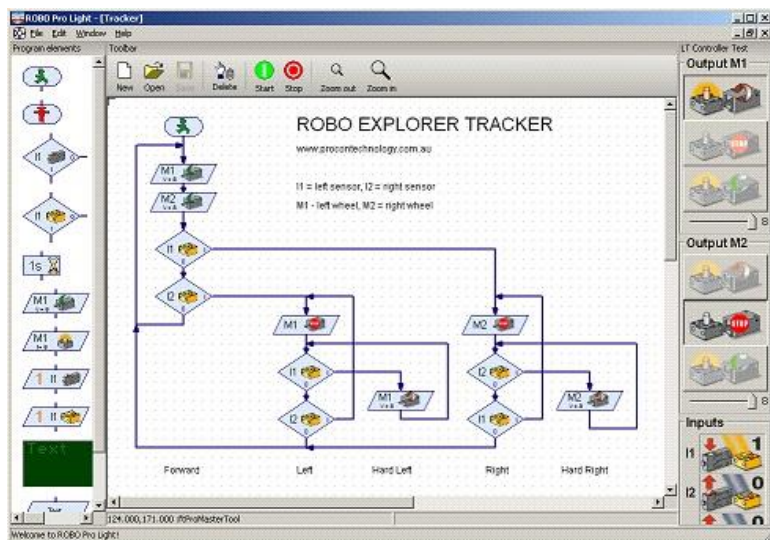




A continuación, algunas plataformas y entornos de desarrollo de simulación de robots:

1. LEGO® MINDSTORMS® EV3 Programmer App

Última aplicación gratuita diseñada para el ladrillo EV3 de LEGO Minsdstorm, que permite crear



programas para robots MINDSTORMS directamente desde dispositivos con iOS y Android Tablet o PC.

La aplicación para EV3 sustituye a las anteriores desarrolladas para NXT, pero mantiene el empleo de cajas, bloques y diagramas de flujo común a los modelos anteriores de



LEGO MINDSTORMS de la marca y al sistema de programación LabView de National Instruments, del que hereda su filosofía de programación con un lenguaje visual gráfico. Recomendado para sistemas hardware y software de pruebas, control y diseño, simulado o real y embebido.

Su principal característica es la facilidad de uso, válido tanto para programadores profesionales como para personas con pocos conocimientos en programación que pueden hacer programas relativamente complejos, imposibles para ellos de hacer con lenguajes tradicionales. Con LabVIEW pueden crearse programas, equivalentes a millones de páginas de código texto, para aplicaciones complejas, programas de automatizaciones de decenas de miles de puntos de entradas/salidas. Todo esto es extrapolable a la plataforma EV3, para que sirva de ejemplo es posible controlar el ángulo de giro de un motor paso a paso, con arranque y parada empleando únicamente un bloque funcional, mientras que con un lenguaje escrito se hace necesario implementar varias líneas de código más complejas de entender.

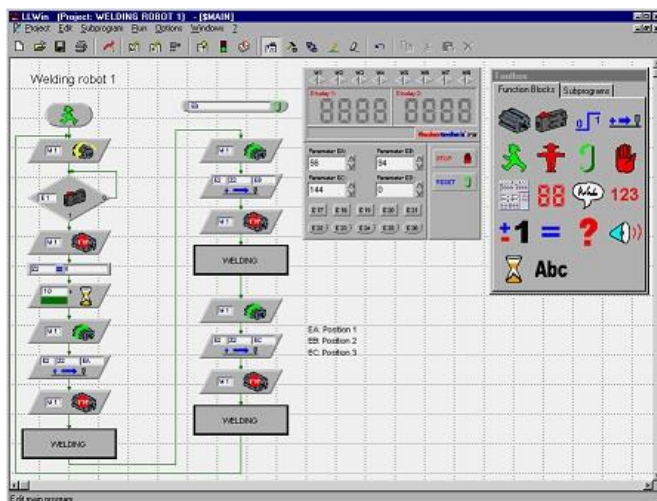


Se trata de una plataforma por tanto muy sencilla e intuitiva de emplear, diseñada para un robot con todas las posibilidades que un juguete de construcción puede ofrecer, ideal para iniciarse.



2. FischerTechnik ROBO PRO

Se trata de otra plataforma sencilla para principiantes a través de la programación mediante diagramas de flujo, que continua con la filosofía del programa LLWin empleado con anterioridad por esta marca, y muy extendido en los centros de enseñanza secundaria. Está constituido por diferentes módulos de software que intercambian datos no sólo mediante variables sino también con conexiones gráficas, con lo que el programa es más fácil de entender incluso aunque se trate programas complejos.

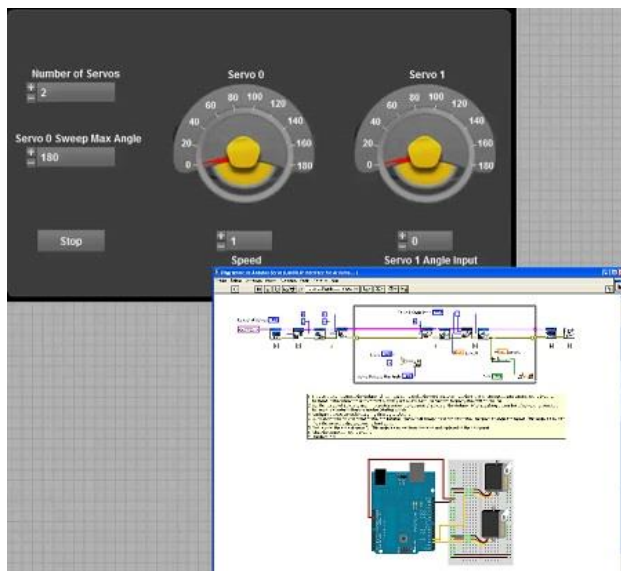


ROBO PRO proporciona todos los elementos clave de un lenguaje de programación moderno, por ejemplo: las matrices, las funciones, objetos, eventos asíncronos y procesamiento paralelo. Que lo convierten en una herramienta útil. Los programas se traducen directamente en lenguaje máquina para la ejecución de máxima eficacia.

La programación gráfica se realiza mediante diagramas de flujo y cajas donde se muestran las entradas y salidas que emplea el juguete físico, por lo que es fácil de entender lo que se está haciendo en cada momento de la ejecución.



Comparándolo con la plataforma anterior, basado como se ha dicho en LabView, se trata de una programación a más bajo nivel que la anterior, lo que significa que si bien en el anterior modelo para programar una acción compleja, marcha y parada de un motor, sólo es necesario emplear una caja, aquí es necesario crear una función más compleja que debe realizar todas las tareas correctamente, y que por tanto se acerca, salvando las diferencias, más a un lenguaje escrito con órdenes realizadas con líneas de texto.



3. S4A, Scratch for Arduino

S4A es un entorno desarrollado a partir del prestigioso y popular entorno Scratch desarrollado en el MIT Media Lab de muy amplia difusión en el mundo de la educación.

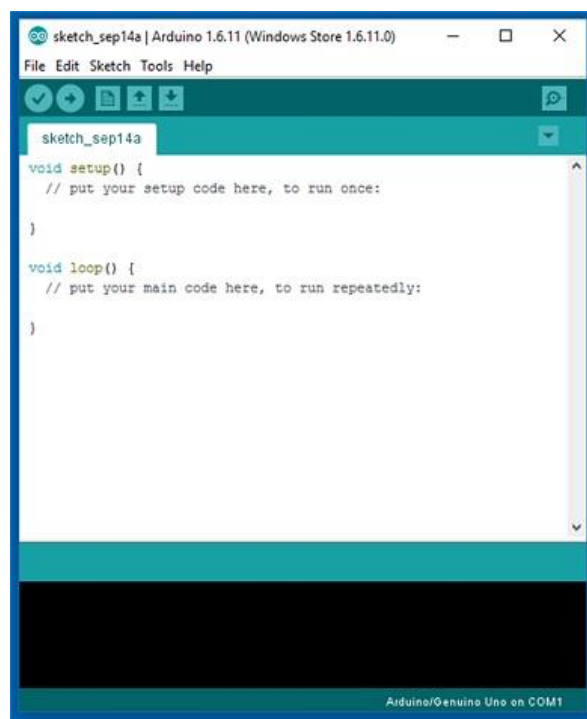
Se trata de una modificación de Scratch que permite programar la plataforma de hardware libre Arduino de una forma sencilla. Proporciona bloques nuevos para tratar con sensores y actuadores conectados a una placa.

4. Arduino IDE

El entorno de desarrollo integrado también llamado IDE (sigla en inglés de Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios lenguajes.

El lenguaje de programación de Arduino es C++. Aunque se hable de que hay un lenguaje propio de programación de Arduino, no es cierto, la programación se hace en C++, no es un C++ puro sino que es una adaptación que proviene de avr-libc que provee de una librería de C de alta calidad

para usar con GCC (compilador de C y C++) en los microcontroladores AVR de Atmel, pero Arduino ofrece unas librerías que facilitan la programación de los pines de entrada y salida y de los puertos de comunicación, así como otras librerías para operaciones específicas.

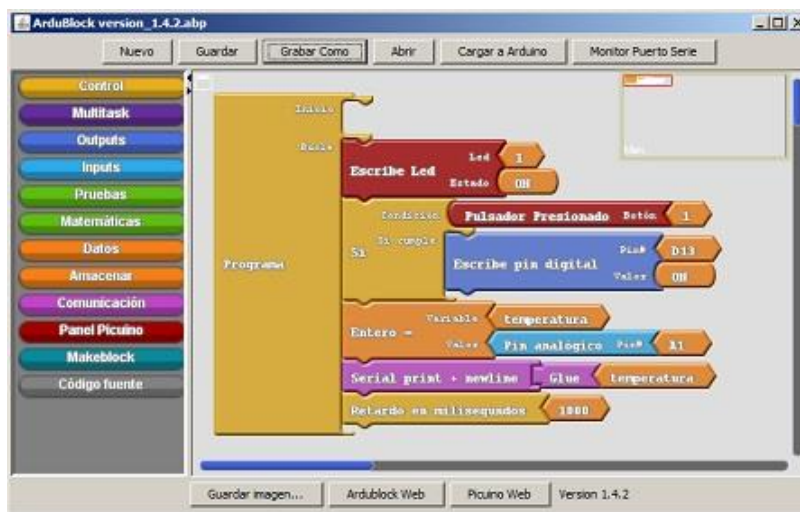




El propio IDE ya incluye estas librerías de forma automática, y no es necesario declararlas expresamente. Otra diferencia frente a C++ standard es la estructura del programa dividido en dos secciones.

El entorno Arduino IDE exige del programador un conocimiento previo avanzado para poder obtener resultados satisfactorios, C++ es un lenguaje de programación de propósito general asociado al sistema operativo UNIX, se trata de lenguaje de medio nivel que trata con objetos básicos como caracteres, números, etc... también con bits y direcciones de memoria, posee una gran portabilidad y se utiliza para la programación de sistemas: construcción de intérpretes, compiladores, editores de texto, etc. Se abre así un inmenso abanico de posibilidades con resultados que pueden alcanzar niveles profesionales.

5. ArduBlock o BitBloq

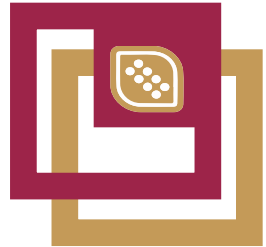


ArduBlock, herramienta dentro de Arduino IDE, o Bitbloq, plataforma Web, son entornos diseñados para hacer programación gráfica en lugar de escrita. Se trata por tanto de herramientas complementarias, que simplifican el trabajo convirtiendo un diagrama de flujo totalmente gráfico, en un

programa de C++ listo para compilar con Arduino, son en definitiva una puerta de entrada perfecta para los más jóvenes programadores en el mundo de la programación de robots.

Por tanto, las plataformas para Arduino serán objeto de estudio en futuros artículos al dar cabida en ellas los distintos niveles de conocimiento, desde el más básico, realizado con bloques funcionales y diagramas de flujo, hasta otro mucho más complejo donde será necesario contar con conocimientos previos para realizar programas escritos en C++.





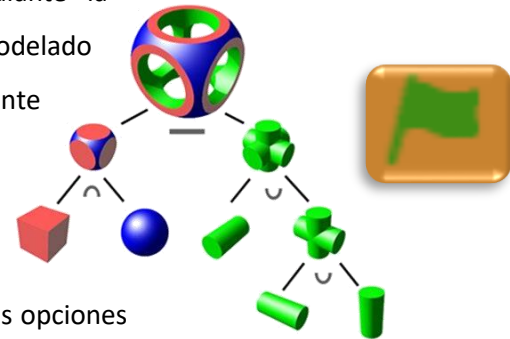
AUTODESK TINKERCAD



Tinkercad es una aplicación web gratuita y fácil de usar que proporciona a la nueva generación de diseñadores e ingenieros las habilidades básicas para la innovación: diseño 3D, electrónica y codificación.



Es una colección online que incluye herramientas de software de Autodesk que permite a los principiantes crear modelos 3D. Este software CAD se basa en una geometría sólida constructiva (CSG), que permite a los usuarios crear modelos complejos mediante la combinación de objetos más simples. Como resultado, este software de modelado 3D es fácil de usar y actualmente es disfrutado por muchos, particularmente maestros, niños, aficionados y diseñadores. ¡Además, es gratis!



TinkerCAD es una buena alternativa a otro software de modelado 3D como SketchUp o Fusion360, otra solución de Autodesk, si no necesita las opciones más avanzadas de estas soluciones. En realidad, la compañía líder de software adquirió TinkerCAD en 2013, dos años después de su lanzamiento por el ex ingeniero de Google Kai Backman y su cofundador Mikko Mononen. ¡La principal ventaja sobre estos dos programas es que es gratuito y además, también una posibilidad realmente interesante y es la de montar, programar y simular circuitos con Arduino!

La geometría sólida constructiva (CSG) es una técnica de modelado sólido que combina objetos primitivos para crear objetos visualmente más complejos.

¿Cuáles son las principales características de TinkerCAD?

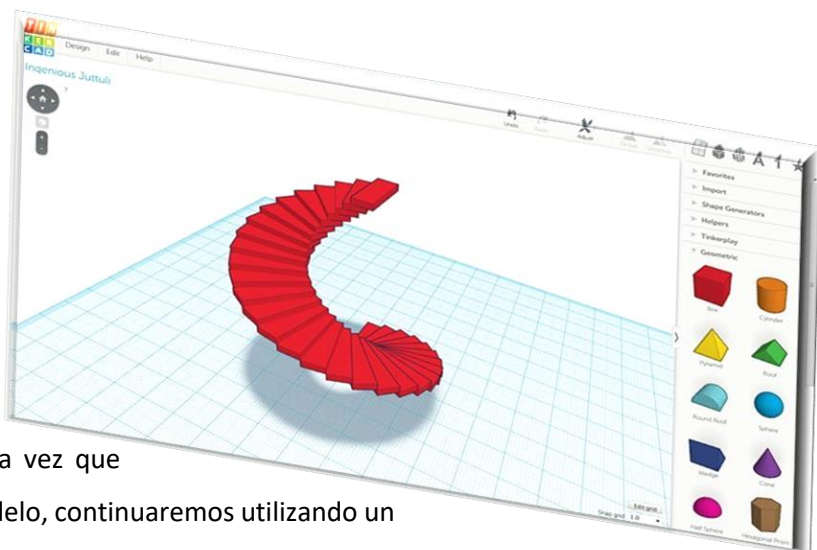
Aunque TinkerCAD es perfecto para principiantes, no significa que aquellos que tienen más experiencia con el modelado 3D no apreciarán este software. Dado que se basa en CSG para crear modelos sólidos, siempre podemos dificultar el modelo agregando más formas.



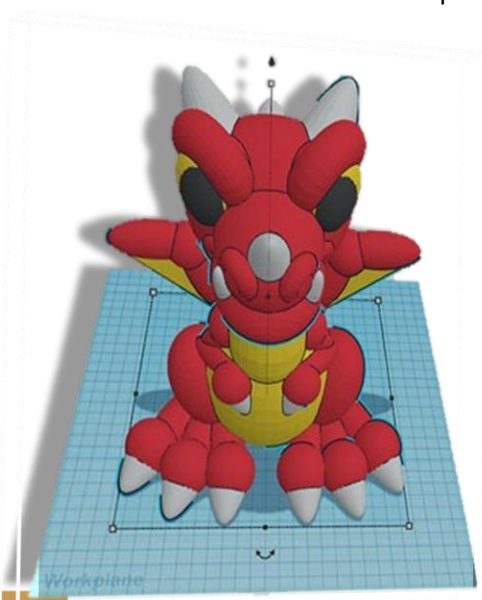


Además, el software permite agregar circuitos electrónicos a los diseños 3D para crear objetos con luz y movimiento. El resultado final incluso se puede simular en el software para verificar cómo responderán los componentes en la vida real. Otra capacidad de TinkerCAD es transformar un diseño 3D en modelos de ladrillo construibles, similar a la creación de legos. Finalmente, para aquellos que aman Minecraft, también incluye la capacidad de crear diseños compatibles con la aplicación.

Por lo tanto, TinkerCAD se puede utilizar para una gran variedad de aplicaciones, incluida la impresión 3D. Los modelos 3D se pueden guardar en tres formatos diferentes, STL, OBJ y SVG. Una vez que

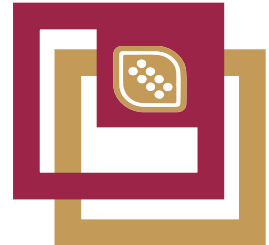


tengamos el archivo STL del modelo, continuaremos utilizando un software de laminado. El software de corte convierte el modelo 3D en una serie de capas delgadas y produce un archivo de código G que contiene instrucciones adaptadas a un tipo específico de impresora. En otras palabras, está dividiendo el objeto en una pila de capas planas y describe estas capas como movimientos lineales del extrusor de la impresora 3D.



Los usuarios a menudo complementan lo intuitivo que es este software CAD. La transformación, la duplicación y la modificación de la forma son fáciles de entender. Además, Autodesk ha puesto a disposición de su comunidad muchos recursos. Entre otras cosas, encontrarás inspiración, consejos y trucos para comenzar, en el blog de la empresa, ¡así como videos! Puedes encontrar más información en la web de TinkerCAD.





BIBLIOGRAFÍA

10 aplicaciones para enseñar programación robótica en colegios – Bejob. (2022). Bejob.
<https://www.bejob.com/10-aplicaciones-para-ensenar-programacion-robotica-en-colegios/>

12 programas gratis sobre robótica (Parte I) | Banana-Soft.com. (2022). Banana-Soft.
<https://banana-soft.com/es/12-programas-gratis-sobre-robotica-parte-i>

15 Apps para aprender programación y robótica en casa de forma divertida. (2021, 5 agosto). Sapos y Princesas. <https://saposyprincesas.elmundo.es/ocio-en-casa/apps-videojuegos/apps-para-aprender-programacion-y-robotica-en-casa/>

A. (2022, 4 enero). *Cómo funciona la Robótica Industrial.* aula21 | Formación para la Industria.
<https://www.cursosaula21.com/como-funciona-la-robotica-industrial/#:%7E:text=Para%20programar%20el%20robot%2C%20el,los%20puntos%20a%20toda%20ve>
locidad.



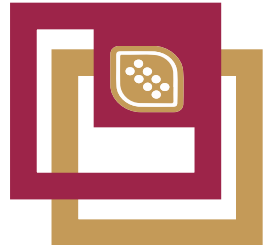
García, M. R. (2018). La Revista Digital del Portal de la Educación. *PLATAFORMAS DE PROGRAMACIÓN DE ROBOT.*
http://revistas.educa.jcyl.es/revista_digital/index.php?option=com_content&view=article&id=3789&catid=84&Itemid=87

M., A. (2020, 20 abril). *TinkerCAD: ¡Te contamos todo lo que necesitas saber!* 3Dnatives.
<https://www.3dnatives.com/es/tinkercad-software-200420202/#!>

Merino, M. (2020, 12 octubre). *Seis aplicaciones gratuitas para aprender robótica y programación.* Genbeta. <https://www.genbeta.com/desarrollo/seis-aplicaciones-gratuitas-para-aprender-robotica-programacion>

Programa y controle robots fácilmente con el software Robot Control de igus®. (2022). igus.
<https://www.igus.com.mx/info/robot-software>





Instalación y uso Básico de Tinkercad



Ahora que ya tienes la diferencia entre un IDE, como aplicaciones nativas y un PWA como desarrollo en web. Es momento de seguir las siguientes instrucciones para trabajar con la herramienta adecuada en la programación de robot.

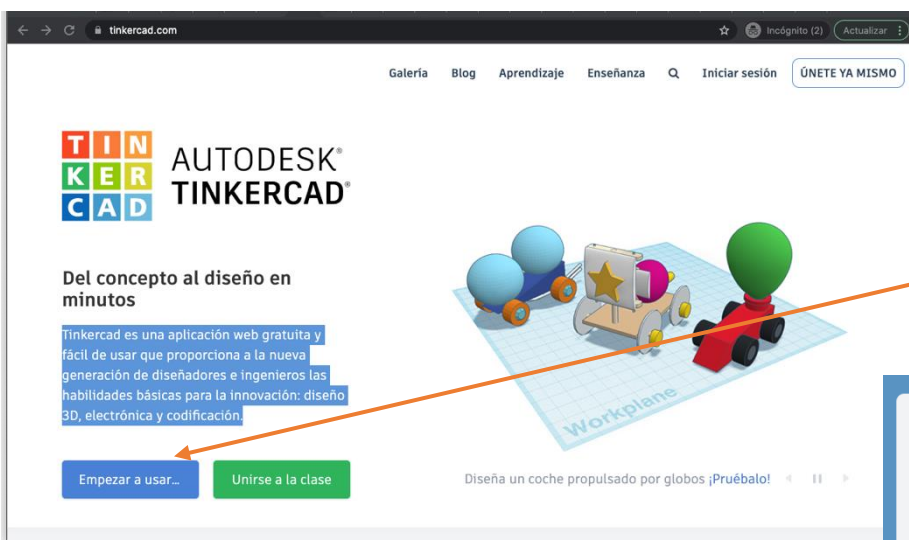
En la computadora o en el teléfono, vamos a utilizar la Aplicación Web Progresiva (PWA)

Tinkercad

Para ello, debes abrir tu navegador preferido; de preferencia utiliza Google Chrome para hacer uso de las extensiones y datos que Google tiene con tu cuenta.



Escribe la URL: <https://www.tinkercad.com/>



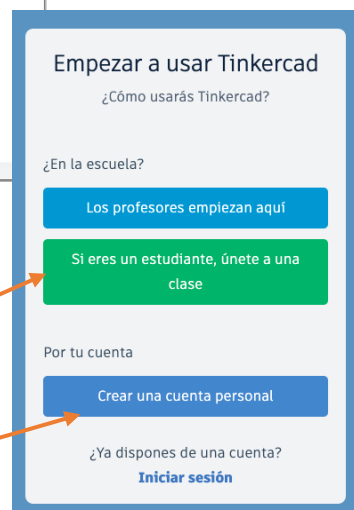
Elegimos **Empezar a usa...**

Al dar clic en el botón, nos mostrará la siguiente pantalla:

En caso de que tu profesor tenga una clase, elije la opción:

Si eres un estudiante, únete a una clase

O puedes tener tu propia cuenta en la aplicación, utilizando una cuenta personal; da clic en el botón: **Crear una cuenta personal.**



Después de ingresar

Una vez hecho login, debes leer y aceptar los términos y condiciones para poder utilizar la PWA y sincronizar tus dispositivos.

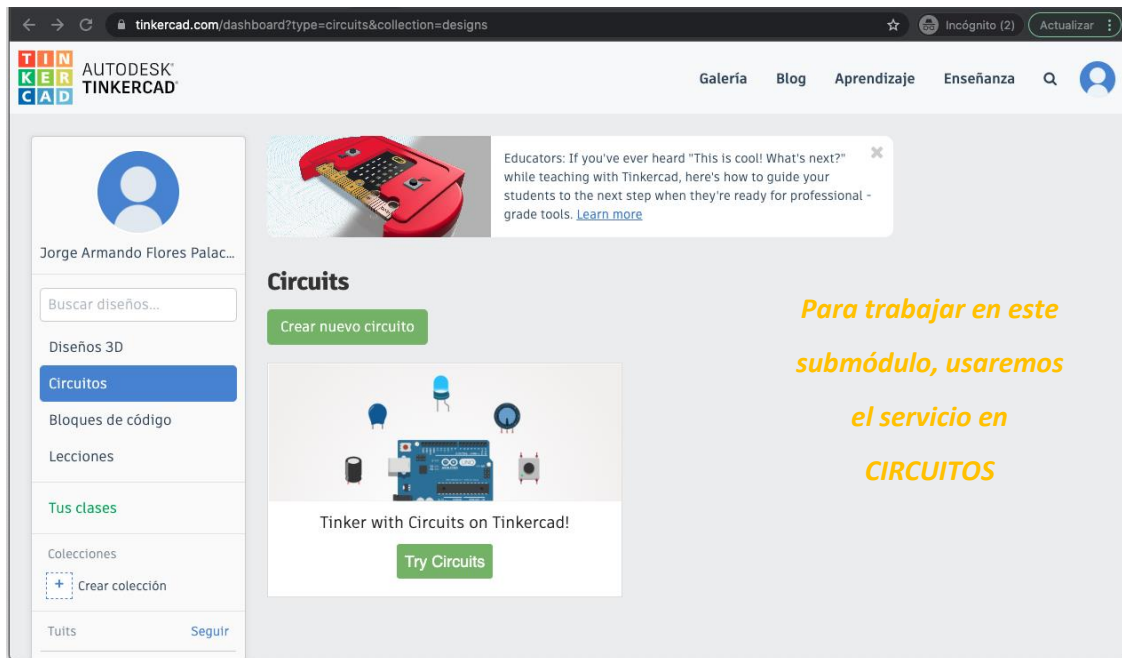
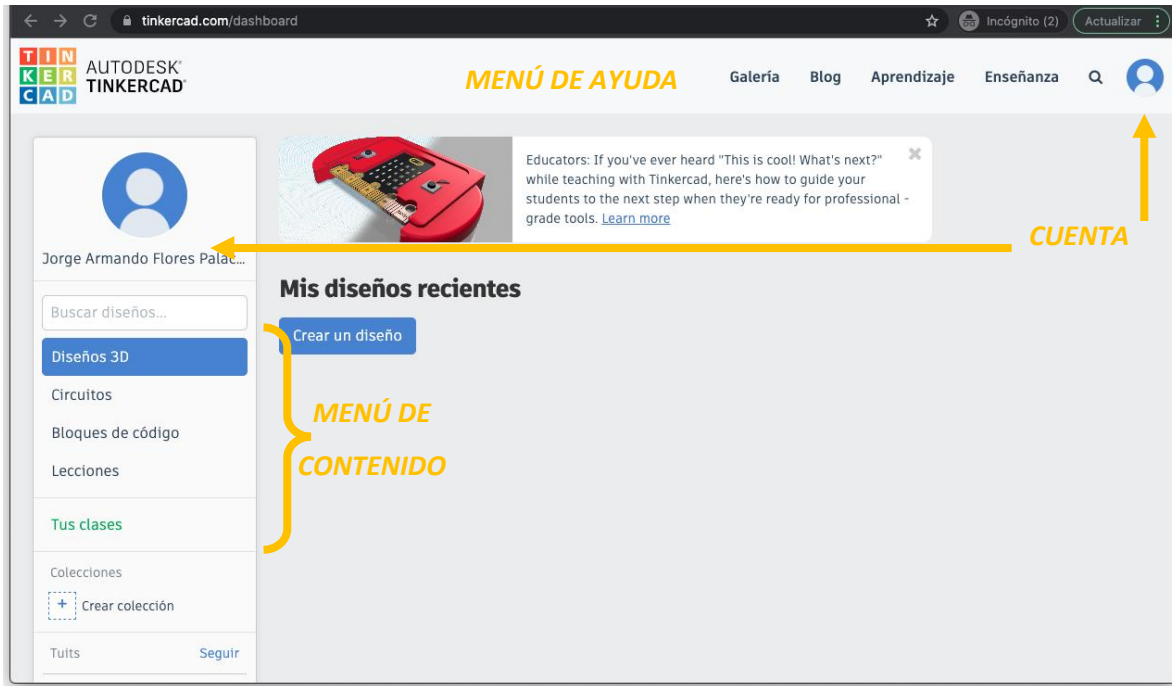
Español (Latinoamérica) Ayuda Privacidad Condiciones

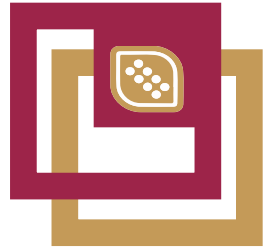




Entorno de TinkerCAD

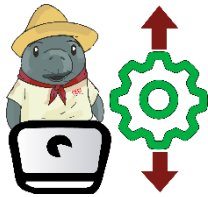
Después de la bienvenida, observarás la siguiente pantalla del contenido





Práctica 1. Hola mundo en Arduino

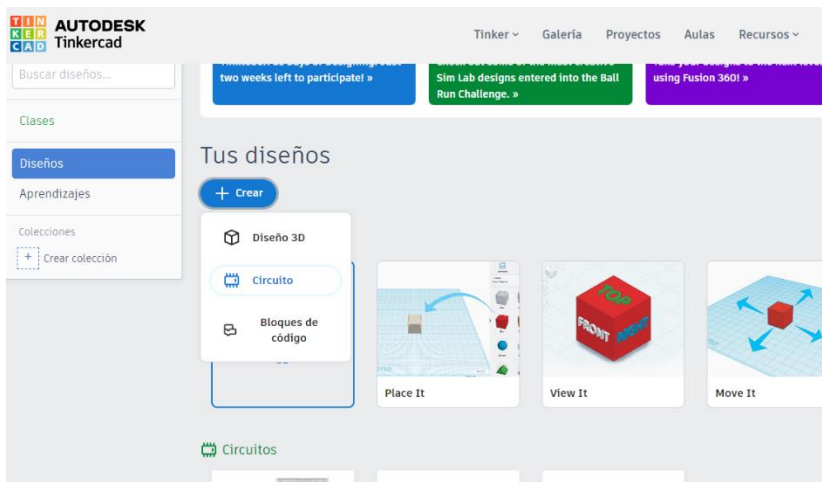
Instrucciones: De manera individual o en pareja, consideren lo estudiado hasta este momento y el material audiovisual propuesto a continuación para elaborar tu primera práctica programado en TinkerCAD. Una vez concluida la práctica, busca el esquema de conexión y deberás realizar una presentación de todo lo realizado ante el grupo.



PRÁCTICA 1. "HOLA MUNDO EN ARDUINO"

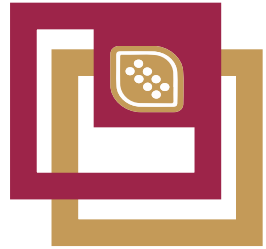


1) Ingresa a TinkerCad.com, en el menú "Tus Diseños" y elige "Crear/Circuito".

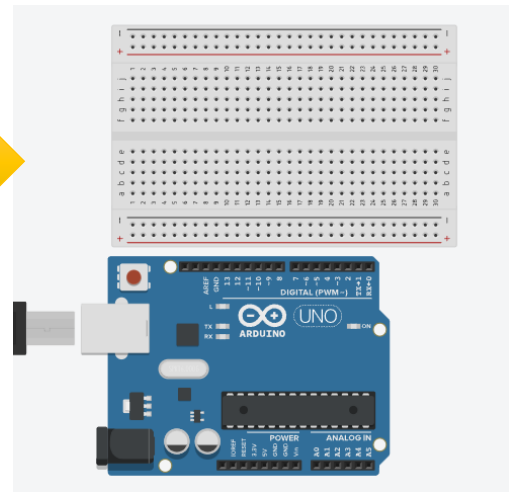
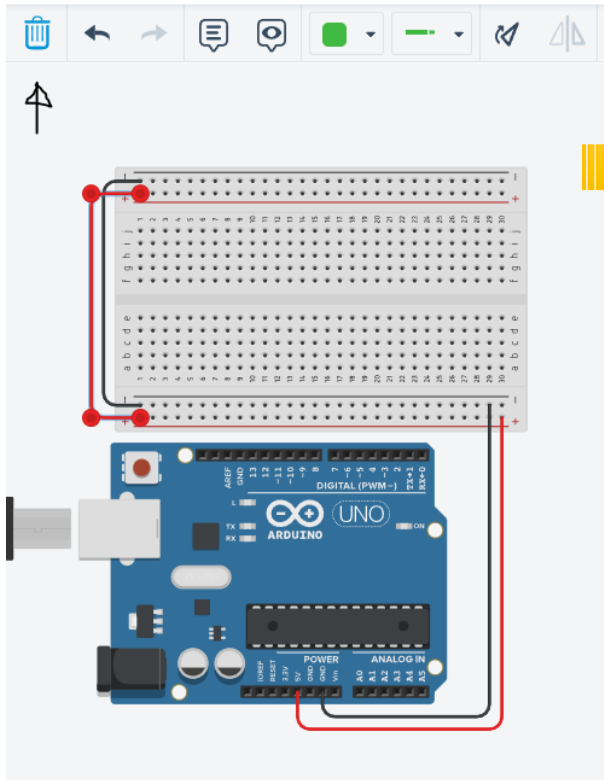


2) En el menú de componentes da clic a la opción "Arduino" y selecciona la "Placa de pruebas", arrastra la placa virtual hacia el área de trabajo.





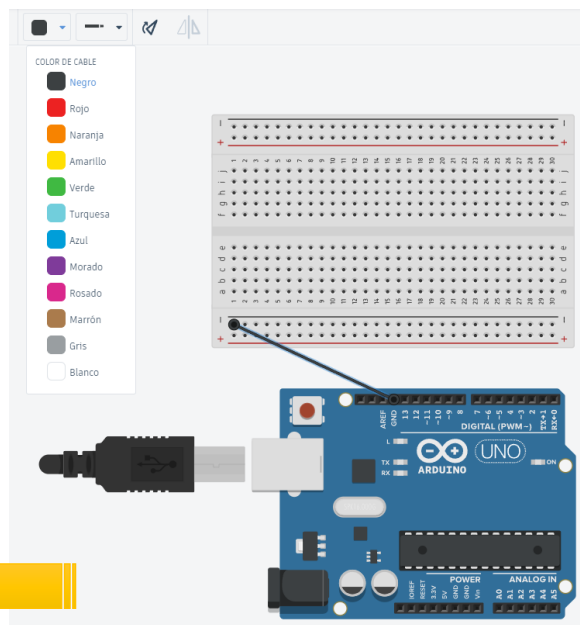
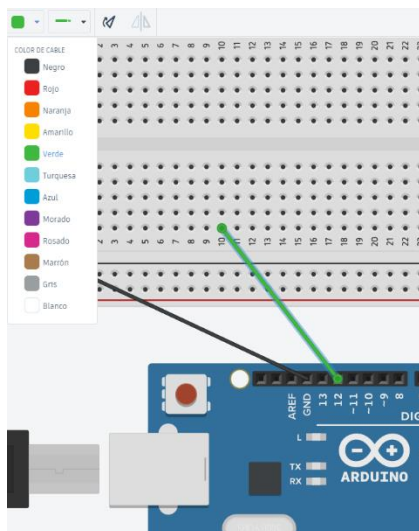
3) Selecciona con un clic los cables (negros y rojos) y elimínalos con el botón **eliminar** que se encuentra en la parte superior del área de trabajo.

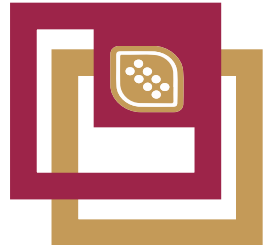


4) Ahora conectamos un cable **negativo** que va de la tierra de la placa Arduino (**GND**) a la zona de alimentación del bus del **protoboard**, línea negativa y lo cambiamos a color negro como se muestra en la imagen:

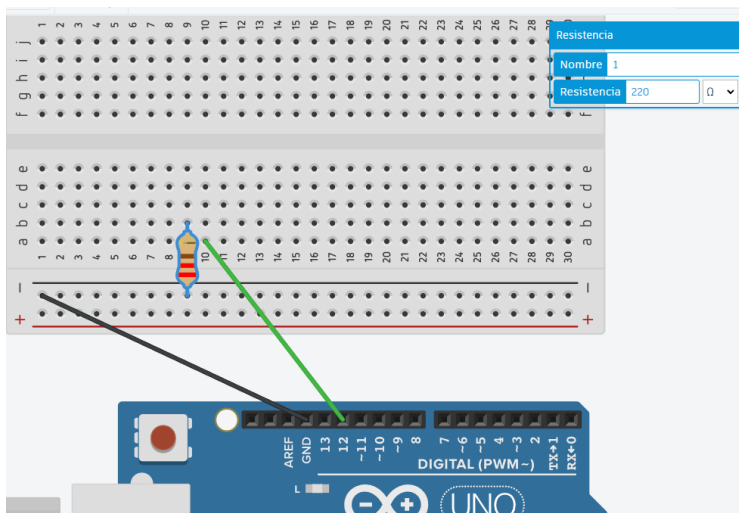


5) Conectamos el cable **positivo** del puerto 12 de la placa Arduino al puerto 10 del **protoboard** y el cable lo configuramos en color verde.





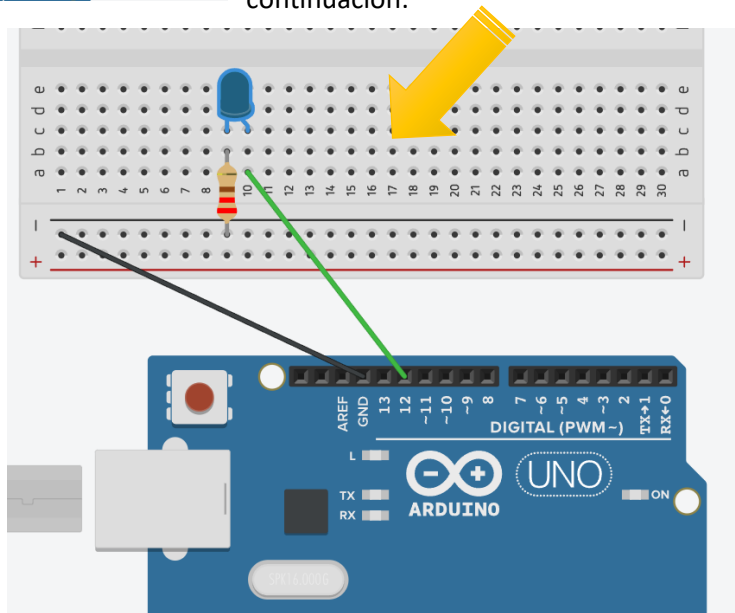
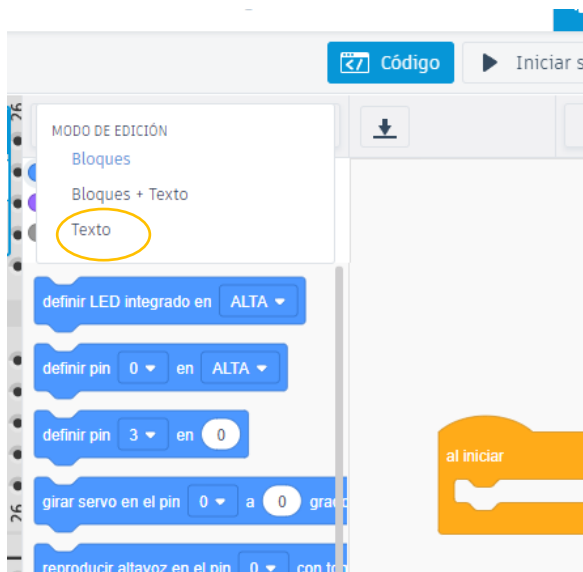
6) Agregamos una **resistencia**; en el menú, elige la opción de componentes básicos y selecciona la resistencia. Llévela al protoboard y conéctala como se muestra en la imagen.

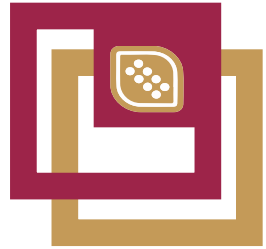


En el cuadro de diálogo, configura la resistencia a 200 Ω .

7) Entre los componentes básico, elige el **LED** y colócalo en el protoboard, dejando el **cátodo** en la línea negativa y el **ánodo** en la línea positiva, como se muestra a continuación:

8) A continuación, escribiremos en código para que encienda el LED. Para ello, debes dar clic en el menú **código** y elige la opción **texto**:





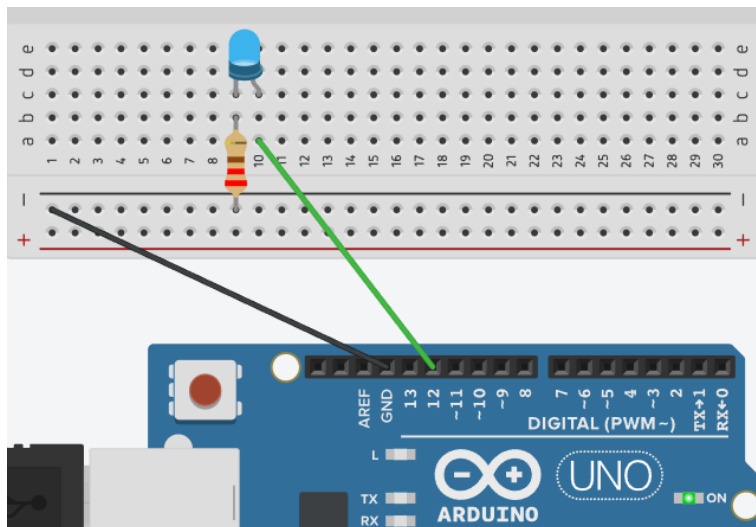
9) Ahora modifica el código para que quede de la siguiente manera:

```

1 // C++ code
2 //
3 void setup()
4 {
5     pinMode(12, OUTPUT);
6 }
7
8 void loop()
9 {
10    digitalWrite(12, HIGH);
11    delay(1000); // Wait for 1000 millisecond(s)
12    digitalWrite(12, LOW);
13    delay(1000); // Wait for 1000 millisecond(s)
14 }
    
```

Como habrás observado, el código se genera automáticamente para encender y apagar el LED, pero no logra encender el circuito, ya que hay que especificarle que el LED está conectado en el pin 12 de la placa Arduino, como se muestra en el código.

10) Por último, probaremos nuestro primer circuito presionando el botón “Iniciar simulación”, de esta manera podrás observar cómo se enciende la placa Arduino y como parpadea el LED.



¡FELICIDADES! Has terminado tu primera práctica en Arduino.





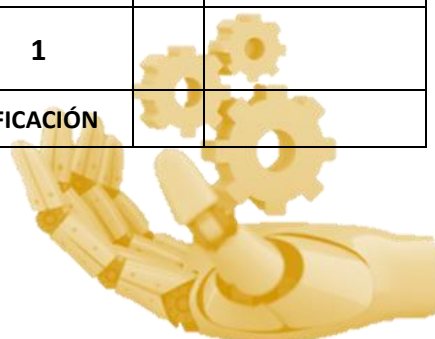
INSTRUMENTO DE EVALUACIÓN
LISTA DE COTEJO 5
Práctica 1: "Hola mundo en Arduino"



DATOS GENERALES

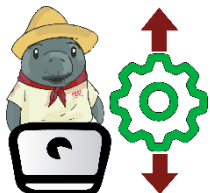
Nombre(s) del alumno(s)	Matrícula(s)
Producto: Práctica en documento o presentación	Fecha
Submódulo 2: Diseño y Programación en Robótica	Periodo: 2025 – 2026B
Nombre del docente	Firma del docente

CRITERIO	INDICADORES	VALOR OBTENIDO		PONDERACIÓN	CALIF	OBSERVACIONES Y/O SUGERENCIAS DE MEJORA
		SÍ	NO			
1	Presenta sus datos para identificarse.			1		
2	Presenta ilustraciones del diseño del circuito y el código en bloques.			2		
3	Demuestra el código en el lenguaje C++ y logra explicar la estructura utilizada junto a las instrucciones.			2		
4	Presenta el circuito funcionando correctamente.			1		
5	No presenta faltas ortográficas.			1		
6	Comprende y expone correctamente los elementos que presenta.			2		
7	Entrega en tiempo y forma.			1		
CALIFICACIÓN						



Práctica 2. ¿Qué me hace falta?

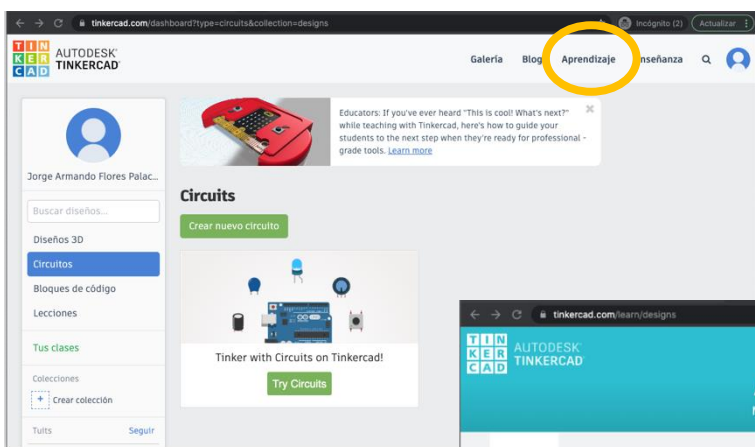
Instrucciones: De manera individual o en pares, consideren lo estudiado hasta este momento y el blog con lecciones de aprendizaje en TinkerCAD. Ahora vamos a encontrar los componentes faltantes. Una vez concluida la práctica, deberás realizar una presentación de este ante el grupo.



PRÁCTICA 2. "¿QUÉ ME FALTA?"



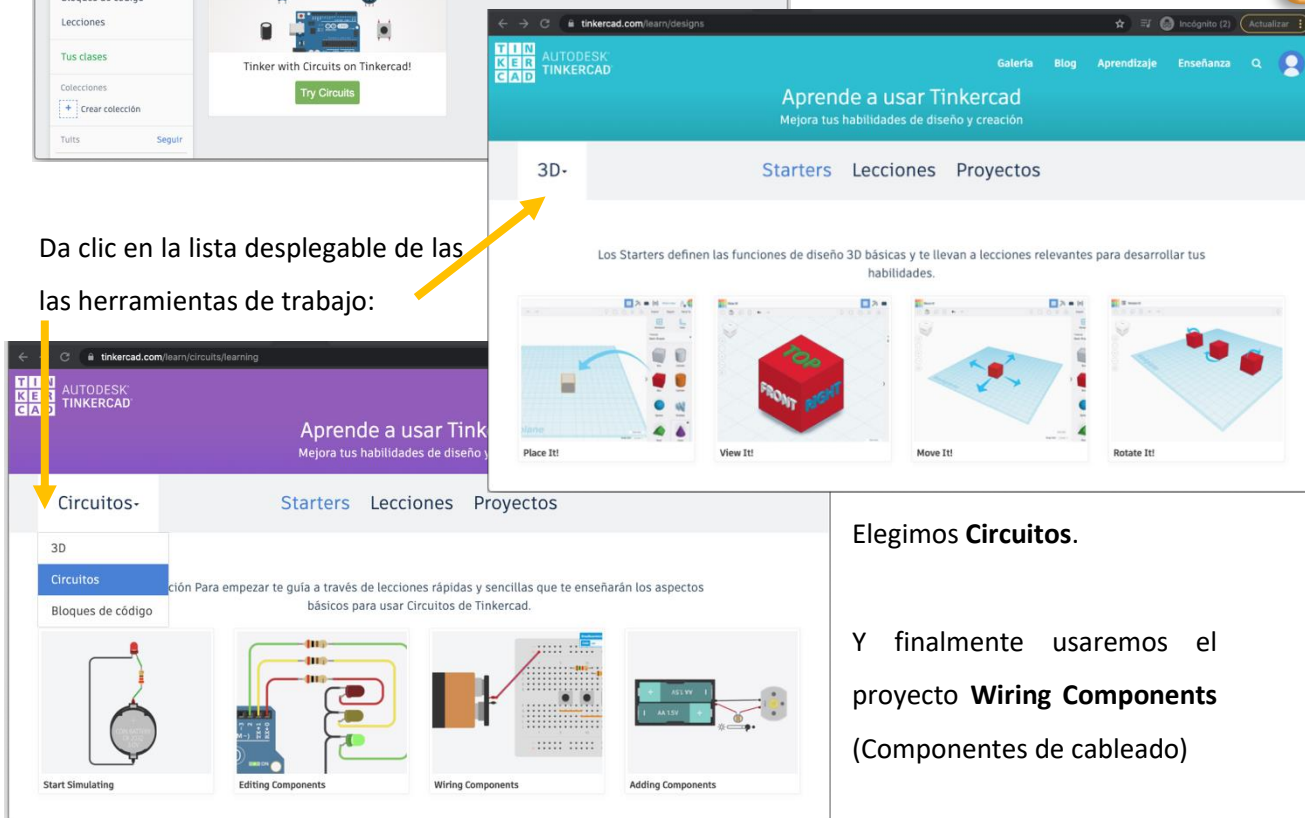
En el menú principal, elijan la opción **Aprendizaje**.



Te llevará a la ventana donde podrás elegir entre varias opciones como aprender a usar TinkerCAD:



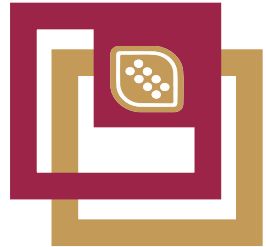
Da clic en la lista desplegable de las herramientas de trabajo:



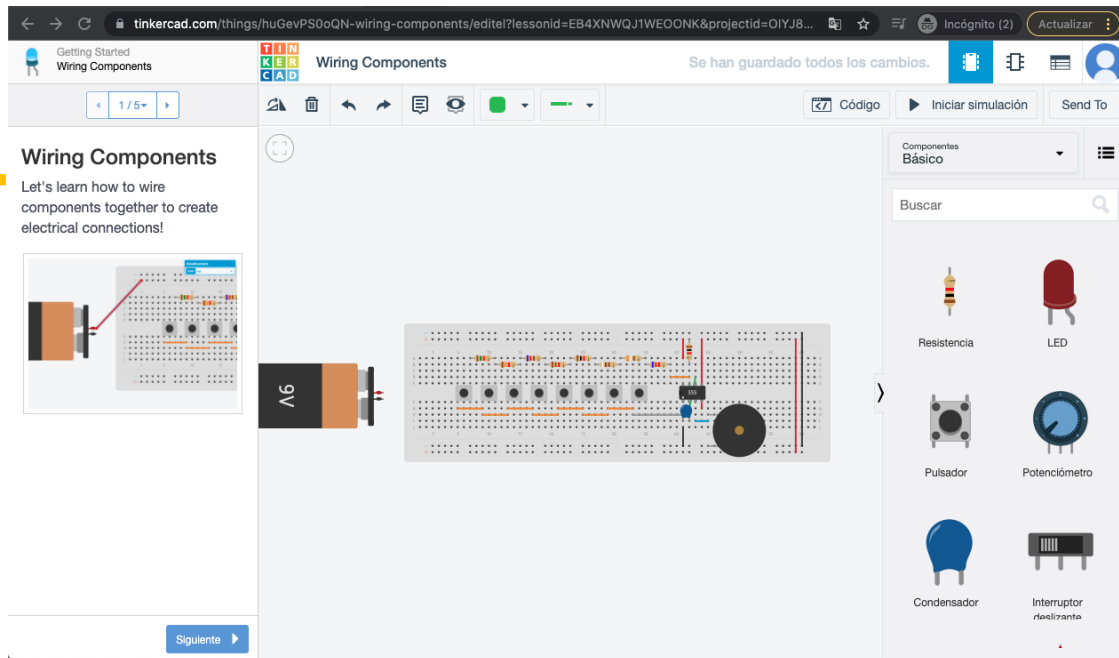
Elegimos **Circuitos**.

Y finalmente usaremos el proyecto **Wiring Components** (Componentes de cableado)





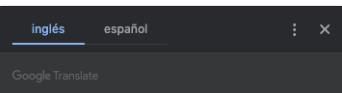
Nos llevará a la siguiente ventana



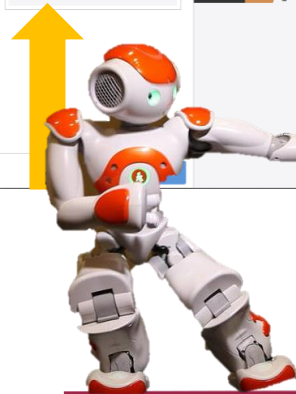
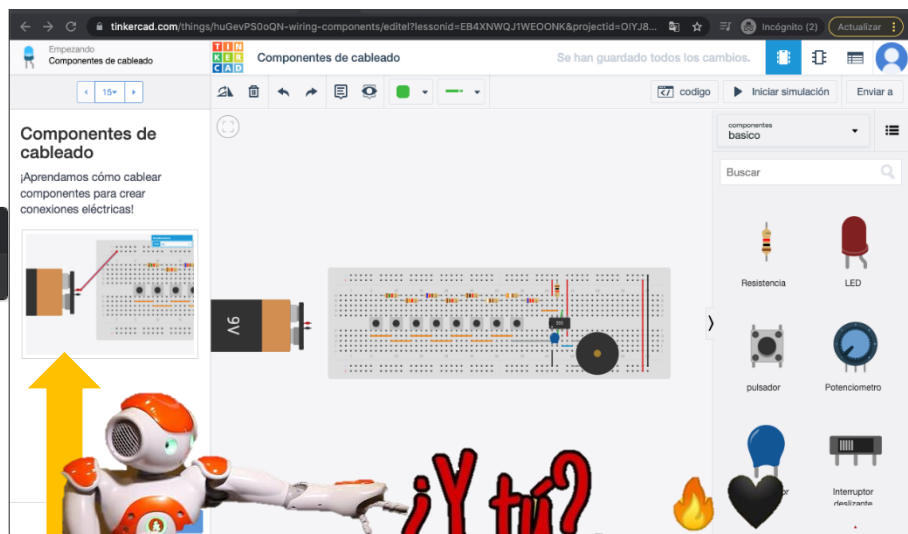
OBJETIVO:

Descubrir los componentes eléctricos o electrónicos que faltan para que el circuito funcione e **Iniciar la simulación**. Además, describe que componentes mecánicos, eléctricos y electrónicos contiene el proyecto y presenten el esquema de conexión.

Puedes utilizar la extensión del traductor del navegador

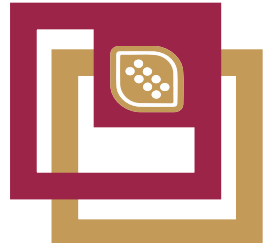


para traducir el tutorial que te permite el software:

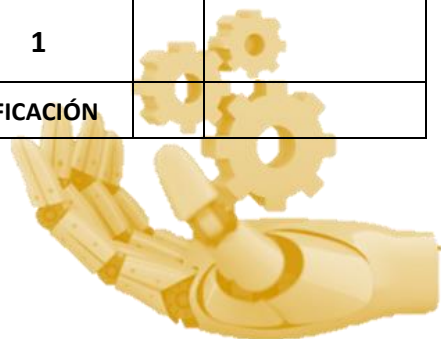


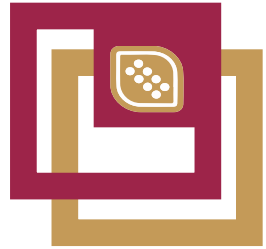
¿Y tú?
¿estás listo?





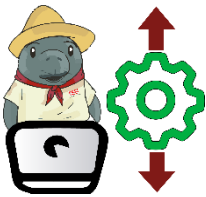
 INSTRUMENTO DE EVALUACIÓN LISTA DE COTEJO 6 Práctica 2: “¿Qué me hace falta?” 						
DATOS GENERALES						
Nombre(s) del alumno(s)				Matrícula(s)		
Producto: Práctica en documento o presentación				Fecha		
Submódulo 2: Diseño y Programación en Robótica				Periodo: 2025 – 2026B		
Nombre del docente				Firma del docente		
CRITERIO	INDICADORES	VALOR OBTENIDO		PONDERACIÓN	CALIF	OBSERVACIONES Y/O SUGERENCIAS DE MEJORA
		SÍ	NO			
1	Presenta sus datos para identificarse.			1		
2	Presenta ilustraciones del diseño del del circuito terminado y funcionando.			2		
3	Demuestra el esquema de conexión e identifica sus elementos.			2		
4	Muestra organización y considera las opiniones de sus compañeros.			1		
5	No presenta faltas ortográficas.			1		
6	Comprende y expone correctamente los elementos que presenta.			2		
7	Entrega en tiempo y forma.			1		
CALIFICACIÓN						





Práctica 3. Semáforo

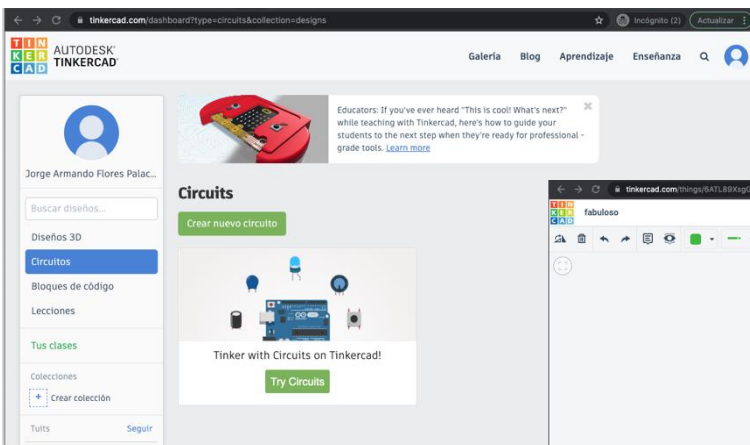
Instrucciones: Organizados en equipos de máximo 5 integrantes, consideren lo estudiado hasta este momento y el material audiovisual propuesto a continuación para elaborar un semáforo programado en TinkerCAD. Una vez concluida la práctica, busca el esquema de conexión y deberás realizar una presentación de todo lo realizado ante el grupo.



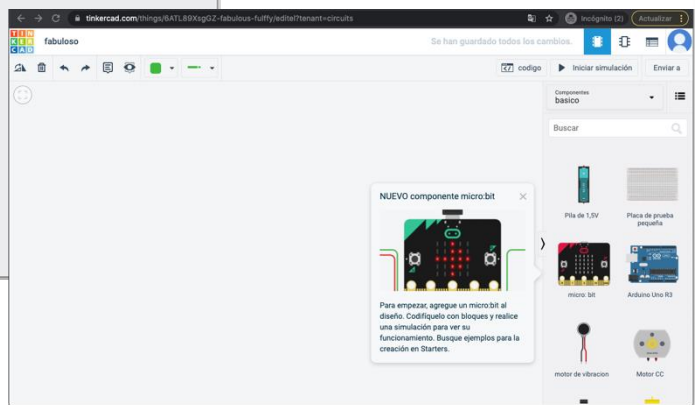
PRÁCTICA 3. "SEMÁFORO"



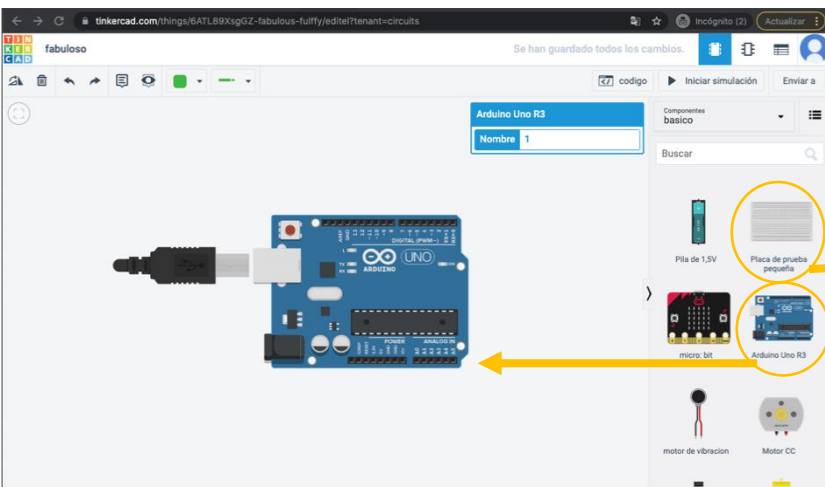
En el menú de diseños, elijan la opción CIRCUITOS.



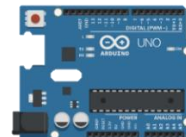
Y seguidamente, dar clic en el botón **"Crear nuevo circuito"** Para ver la siguiente ventana:



Ahora vamos a seleccionar y arrastrar la placa **Arduino Uno R3** y la dejamos en el centro de nuestro espacio de trabajo:

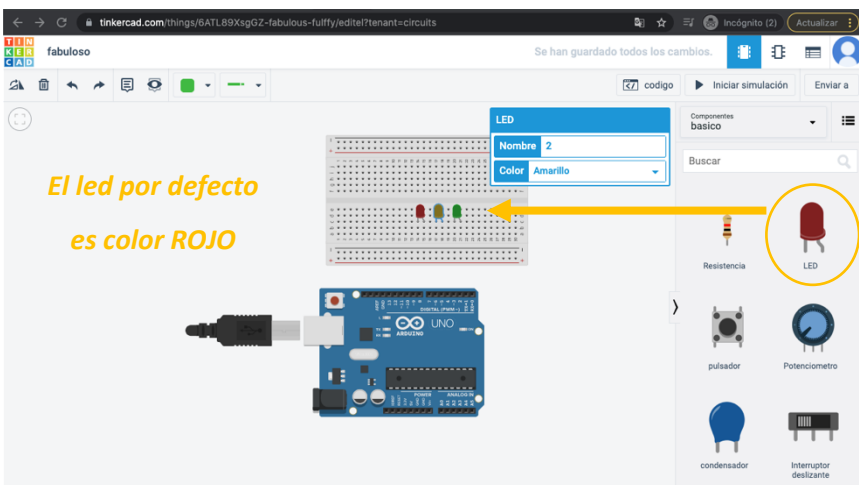


Continúa seleccionando una **Placa de prueba pequeña** (también conocida como Protoboard):





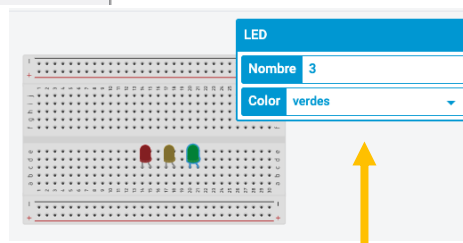
Procedemos a colocar los leds en la Placa de prueba pequeña, seleccionamos el ícono del LED



*El led por defecto
es color ROJO*

Arrastra el LED a la placa y suéltalo. Repite el mismo procedimiento hasta tener los tres leds, solo recuerda separarlos

Puedes cambiar el nombre de los Leds, al igual que los nombres de los demás componentes al seleccionarlos; aparece un cuadro de diálogo con algunos datos que permiten su configuración. Por ejemplo, para los LED, puede cambiar el NOMBRE y el COLOR.



Continuamos colocando las **resistencias**, recuerda: selecciona el ícono de la resistencia, arrástralo a la **placa de prueba pequeña** y posicónala en línea con el primer led y el polo negativo de la placa:

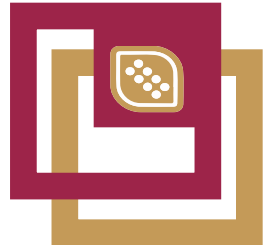


*El polo negativo está
representado por el
signo (-)*

*Verás que los colores de la
resistencia cambiarán de
acuerdo con el valor asignado*

*También, modifica el
valor de la resistencia,
Por defecto está en 1 k Ω
(kilo Ohm); por 500 Ω*





Después de colocar los elementos,
es necesario agregar los
cables o conectores:

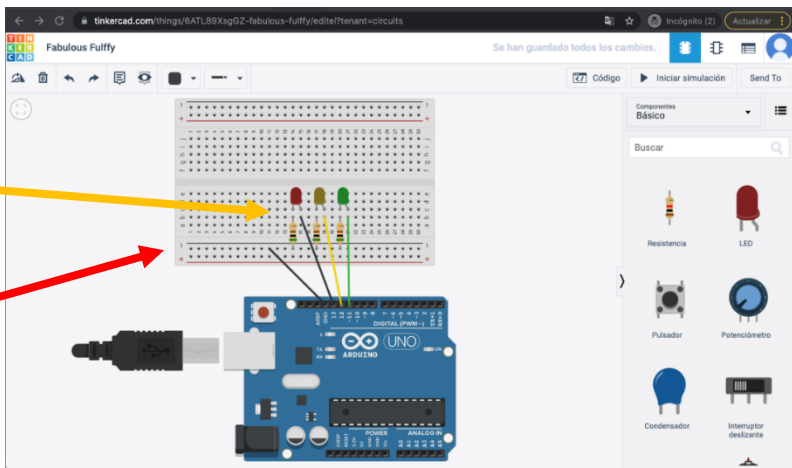
Usaremos la
herramienta cable

Y con el mouse, da clic en los pines (espacios) de
la placa del Arduino con la placa de prueba
pequeña como se observa en la figura

Aquí debes
considerar que los
pines de la placa de
Arduino están
enumerados y esos
índices servirán para
la programación.

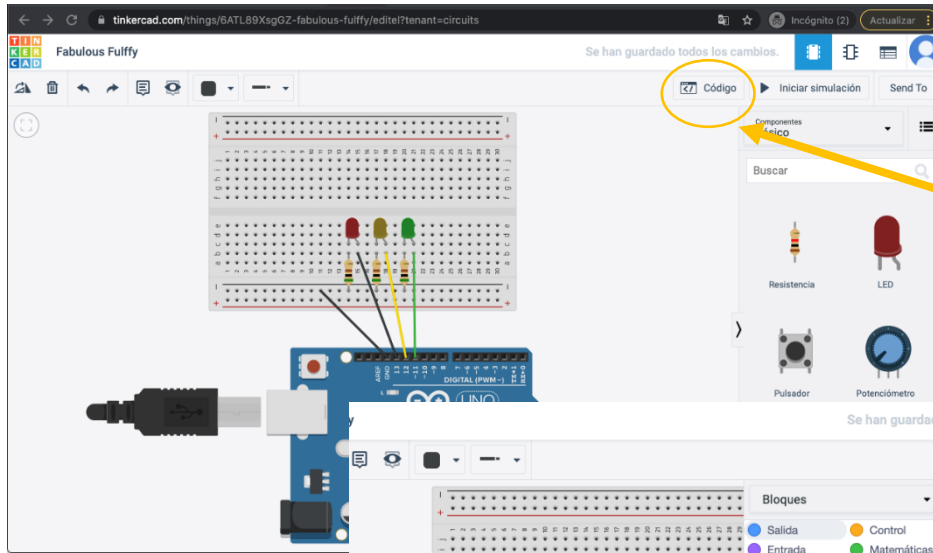
En el caso de los cables que se conectan con
la placa de prueba pequeña, se conectan en
el polo positivo del led, y disponible.

Para cerrar el circuito, colocamos un cable
que va del pin de la **tierra (GND)** en la placa
Arduino al pin **negativo** donde están
conectadas las terminales de las
resistencias.



A programar el circuito

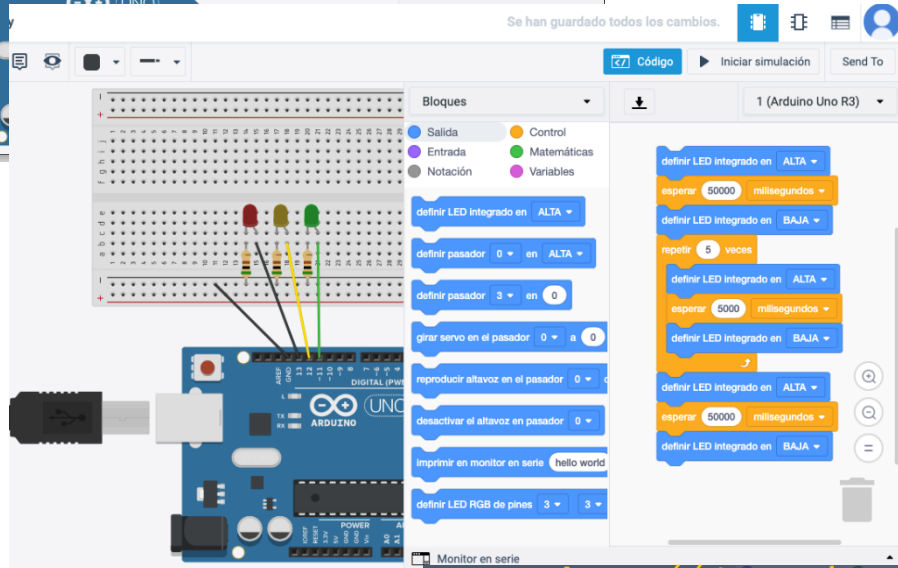
Usando los bloques



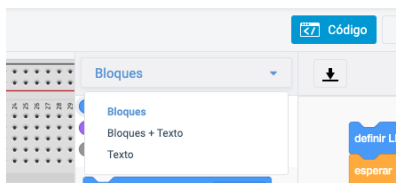
Comenzamos dando clic en el botón **Código** para visualizar los bloques:

Toma los bloques como se muestra a continuación y modifica los datos del tiempo:

Considera que 1 segundo son 1000 milisegundos.

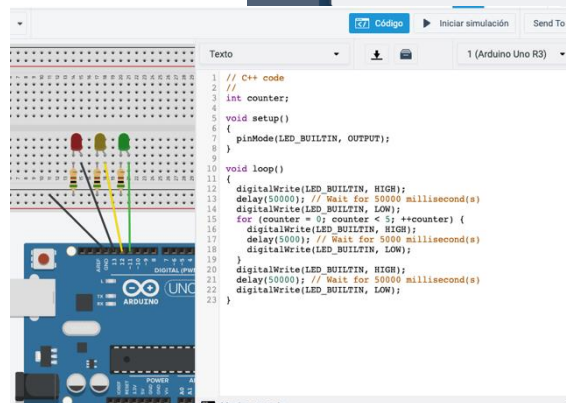


Continuamos dando clic en el menú desplegable



Bloques, para elegir **Texto**:

Te aparecerá el código en lenguaje C++ de la siguiente manera. Debemos modificar algunas líneas.



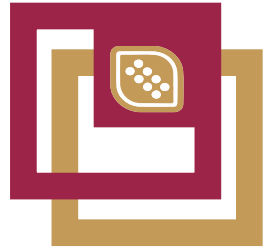
¿Está seguro?

¿Seguro que desea cerrar el editor de bloques? Se perderán todos los bloques que tenga en este momento. Se mantendrá el código del editor de texto y se podrá editar.

Continuar

Cancelar





A programar el circuito

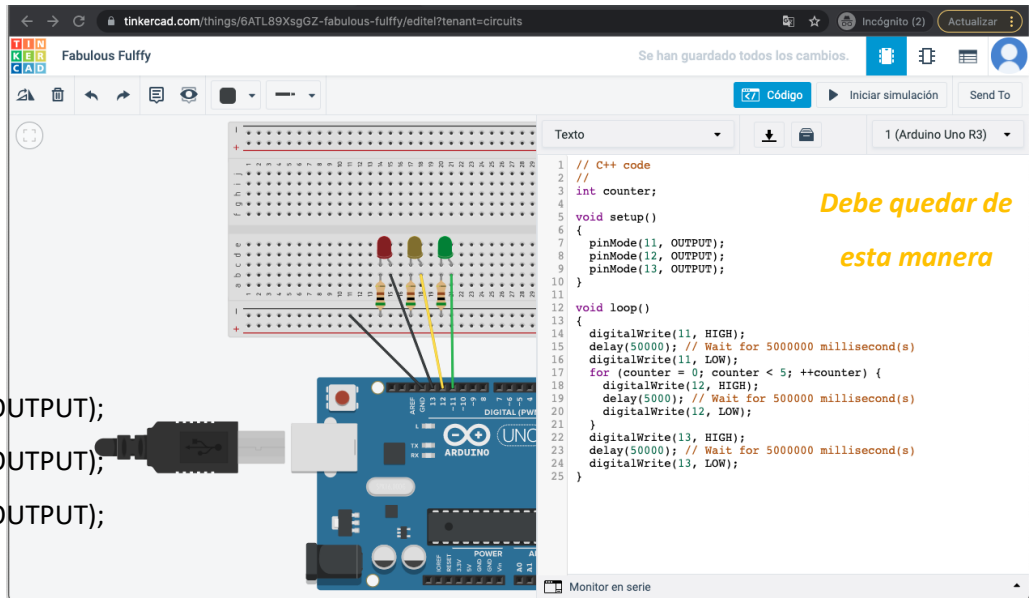
Usando el lenguaje

El código que debes ingresar es:

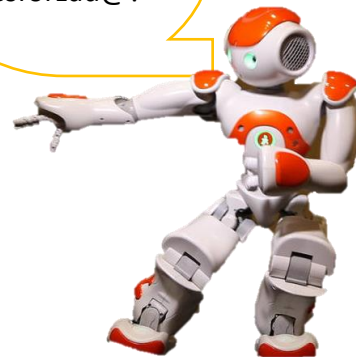
```
// C++ code
//
int counter;

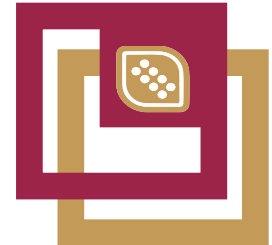
void setup()
{
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
}
```

```
void loop()
{
  digitalWrite(11, HIGH);
  delay(50000); // Wait for 50000 millisecond(s)
  digitalWrite(11, LOW);
  for (counter = 0; counter < 5; ++counter) {
    digitalWrite(12, HIGH);
    delay(5000); // Wait for 5000 millisecond(s)
    digitalWrite(12, LOW);
  }
  digitalWrite(13, HIGH);
  delay(50000); // Wait for 50000 millisecond(s)
  digitalWrite(13, LOW);
}
```

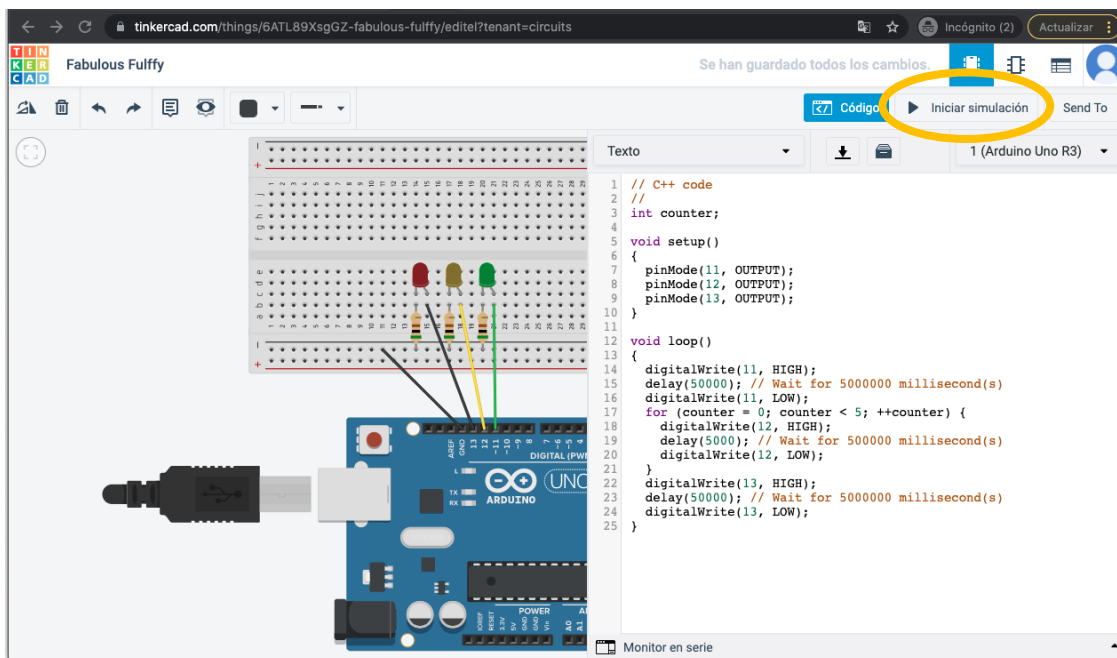


¡Puedes
hacerlo mi
esforzad@!

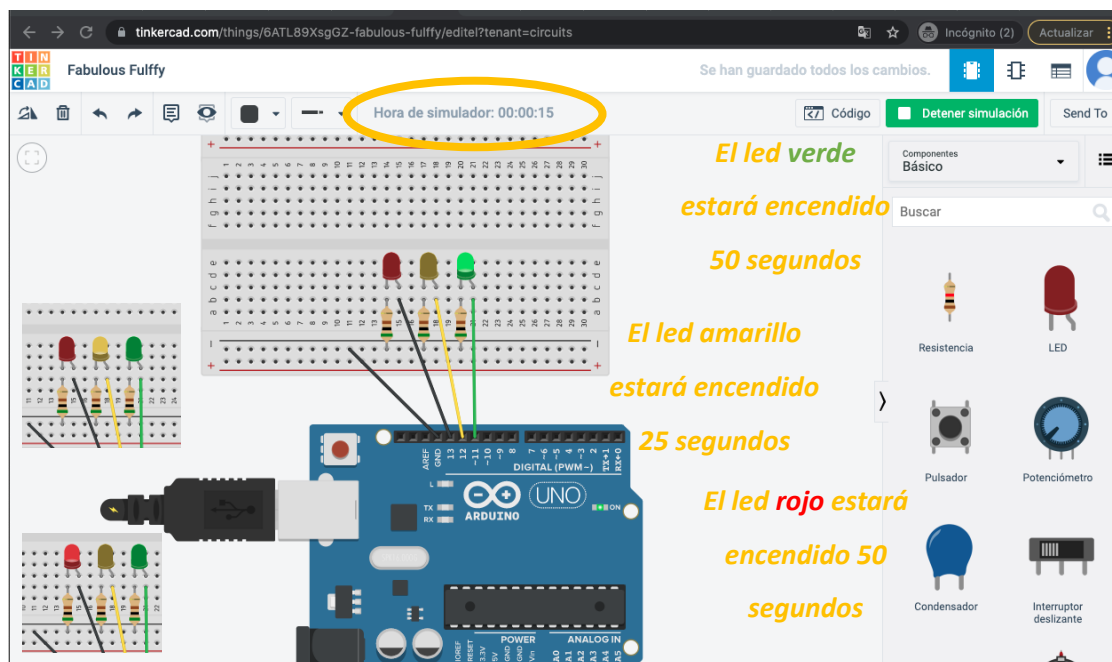




Concluimos nuestra práctica al dar clic en Iniciar simulación:



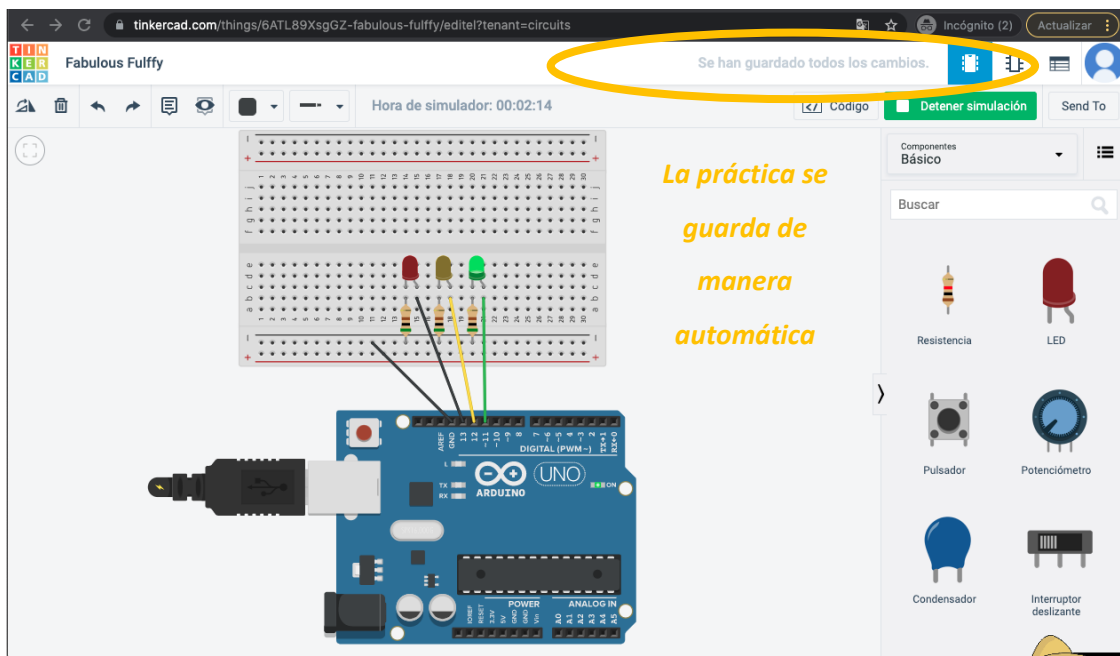
Comenzará a ejecutarse la simulación y se activará el cronómetro:



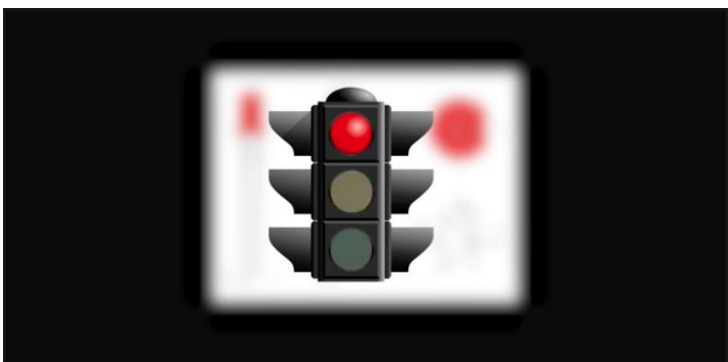
Led verde comienza a encenderse y estará encendido en el intervalo de 0 a 49 segundos.

Después, el led amarillo se encenderá en el intervalo de 50 a 1 minuto, 15 segundos, mientras que el led verde y rojo están apagados. Para finalmente, el led rojo quedará encendido en el intervalo de 1 minutos, 16 segundos a 2 minutos 05 segundos. Después, los leds volverán a encenderse.





Semáforo con TinkerCAD Circuits



<https://youtu.be/zIJpr6NFxrE>

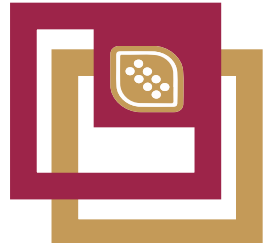
Responde las preguntas de conclusión:

¿Consideras que la robótica te permite observar el funcionamiento de la tecnología en tu contexto?

¿Qué se te dificultó al realizar la práctica?

¿Qué consideras debes mejorar para desarrollar tus habilidades en la robótica?

¿Cómo aplicas lo aprendido en tu vida cotidiana, académica o laboral?





INSTRUMENTO DE EVALUACIÓN
LISTA DE COTEJO 7
Práctica 3: "Semáforo"



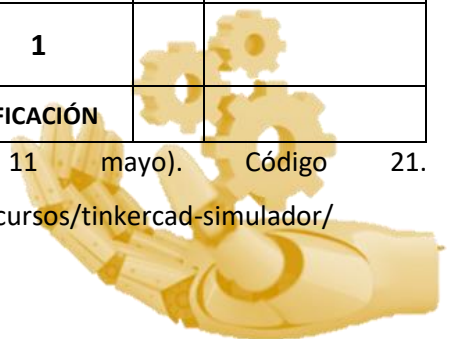
DATOS GENERALES						
Nombre(s) del alumno(s)				Matrícula(s)		
Producto: Práctica en documento o presentación				Fecha		
Submódulo 2: Diseño y Programación en Robótica				Periodo: 2025 – 2026B		
Nombre del docente				Firma del docente		

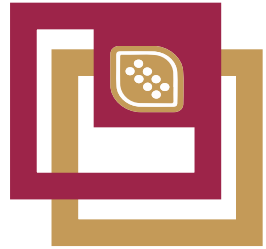
CRITERIO	INDICADORES	VALOR OBTENIDO		PONDERACIÓN	CALIF	OBSERVACIONES Y/O SUGERENCIAS DE MEJORA
		SÍ	NO			
1	Presenta sus datos para identificarse.			1		
2	Presenta ilustraciones del diseño del semáforo y el código en bloques.			2		
3	Demuestra el código en el lenguaje C++ y logra explicar la estructura utilizada junto a las instrucciones.			2		
4	Muestra organización y considera el trabajo de sus compañeros.			1		
5	No presenta faltas ortográficas.			1		
6	Comprende y expone correctamente los elementos que presenta.			2		
7	Entrega en tiempo y forma.			1		
CALIFICACIÓN						



Tinkercad. Simulador. (2018, 11 mayo). Código 21.

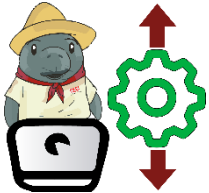
<https://codigo21.educacion.navarra.es/recursos/tinkercad-simulador/>





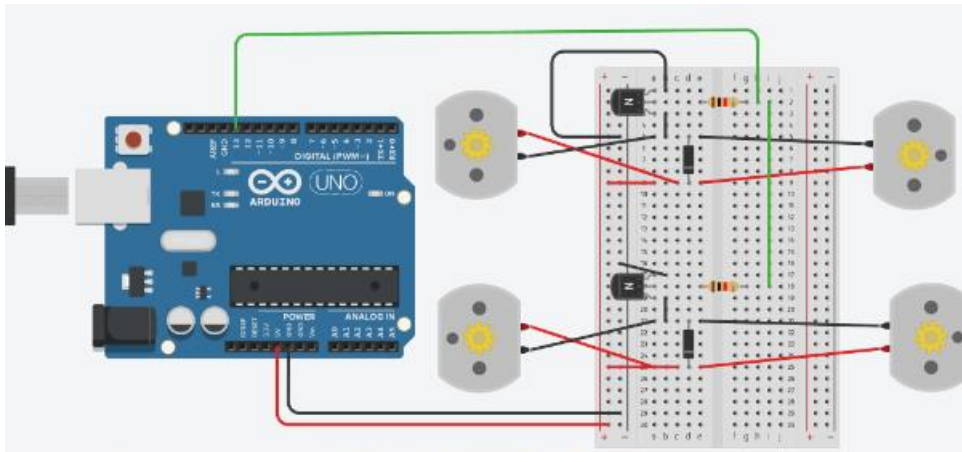
Práctica 4. Cableado de mi dron

Instrucciones: De manera individual o en equipo de 5 integrantes como máximo, consideren lo estudiado hasta este momento. Ahora vamos a desarrollar un circuito de un dron. Una vez concluida la práctica, deberás realizar una presentación de este ante el grupo.



PRÁCTICA 4. "CABLEADO DE MI DRON"

Circuito por realizar:



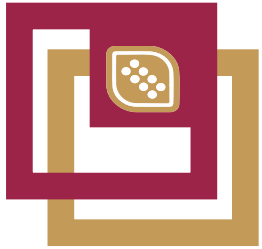
Componentes.

- 1 placa de pruebas.
- Arduino Uno R3.
- 4 motores DC.
- 2 transistores NPN (BJT).
- 3 diodos.
- 2 resistencias de 1 kΩ.
- Cables: rojo, negro.

Propósito

Diseña el circuito presentado, elijan los componentes que se les describen y programen el circuito para hacerlo funcionar con el simulador.

Describan qué hace, elaboren una presentación de sus resultados y presenten sus conclusiones ante el grupo.



Código.

A programar el circuito

```
1 // Código
2 //
3 void setup()
4 {
5   pinMode(LED_BUILTIN, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(LED_BUILTIN, HIGH);
11  delay(1000); // Espera por 1000 milisegundos
12  digitalWrite(LED_BUILTIN, LOW);
13  delay(1000); // Espera por 1000 milisegundos
14 }
```

Usando el lenguaje



CONCLUSIÓN:

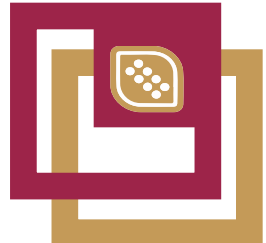
¿Cuáles son los componentes mecánicos, eléctricos y electrónicos que identificas en el circuito?

¿Qué aprendiste?

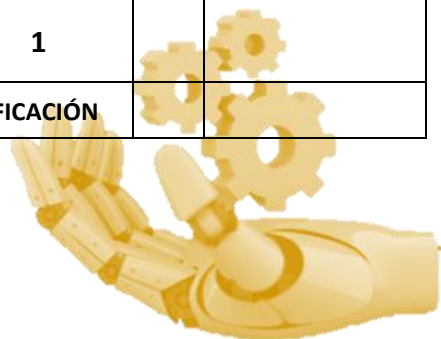
¿Cómo puedes aplicarlo en tu vida cotidiana y profesional?

¿Hay algo que consideras se puede mejorar?





 INSTRUMENTO DE EVALUACIÓN LISTA DE COTEJO 8 Práctica 4: "Cableado de mi dron" 						
DATOS GENERALES						
Nombre(s) del alumno(s)				Matrícula(s)		
Producto: Práctica en documento o presentación				Fecha		
Submódulo 2: Diseño y Programación en Robótica				Periodo: 2025 – 2026B		
Nombre del docente				Firma del docente		
CRITERIO	INDICADORES	VALOR OBTENIDO		PONDERACIÓN	CALIF	OBSERVACIONES Y/O SUGERENCIAS DE MEJORA
		SÍ	NO			
1	Presenta sus datos para identificarse.			1		
2	Presenta ilustraciones del diseño del del circuito terminado y funcionando.			2		
3	Demuestra el esquema de conexión e identifica sus elementos.			2		
4	Muestra organización y considera las opiniones de sus compañeros.			1		
5	No presenta faltas ortográficas.			1		
6	Comprende y expone correctamente los elementos que presenta.			2		
7	Entrega en tiempo y forma.			1		
CALIFICACIÓN						



AUTOMATIZANDO MI CREATIVIDAD



SIGA



Instrucciones:

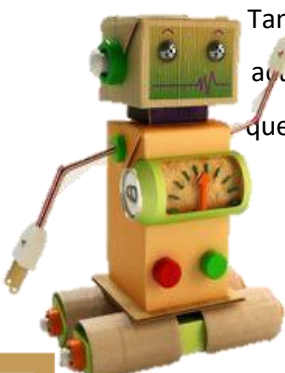
Con los conocimientos que has adquirido a lo largo del submódulo, es momento de diseñar en equipo, un prototipo de robot que permita ayudar en la solución de un problema de su entorno, deberás considerar utilizar el software de programación propuesto u otra herramienta que te permita desarrollar la programación al prototipo.

Propósito:

Desarrollar un prototipo de robot en equipos de 5 integrantes que permita ayudar en la solución de un problema del entorno utilizando el software de programación propuesto, para socializarlo en plenaria.

¿Qué debemos hacer?

Considera los pasos estudiados en la Lectura Fases de Desarrollo de un Robot y comiencen observando y analizando un problema de su entorno, el cual pueda ser solucionado con la implementación de la robótica.



También, recuerda que tienes las actividades guiadas y las prácticas que te apoyarán en el uso de herramientas sencillas para el desarrollo del robot simulado.





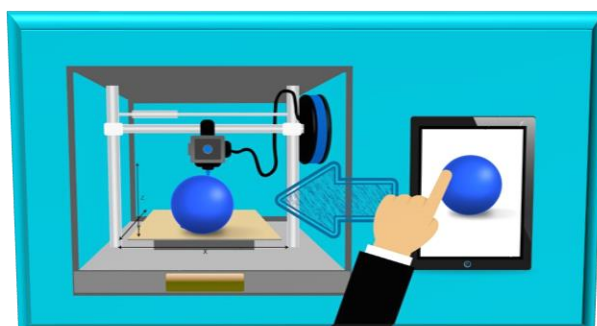
Considera lo siguiente

Nuestro

trabajo consiste en realizar la programación de robot propuesta que pretende resolver el problema observado y analizado.

Algunas propuestas para programar en la simulación del robot que realice **una tarea específica**:

- ◆ Sistemas de riego.
- ◆ Brazo para Incubadora.
- ◆ Medición de humedad.
- ◆ Separa por monedas (alcancía inteligente)
- ◆ Controlador de iluminación (lámpara)
- ◆ Alarma de los electrodomésticos
- ◆ Dispensador de agua
- ◆ Ventilador inteligente. (de acuerdo con la temperatura aumenta la velocidad)
- ◆ Detector de incendio con alarma.



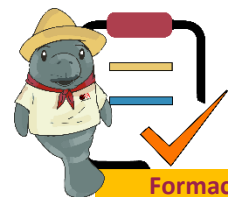
Recuerda que un prototipo no necesariamente es el producto concluido y funcionando, sino que es el primer ejemplar que se fabrica de una figura, un invento u otra cosa, y que sirve de modelo para fabricar otras iguales, o molde original con el que se fabrica.

Considera



<https://youtu.be/m207F-6i8UI>





COLEGIO DE BACHILLERES DE TABASCO PLANTEL No ____
RÚBRICA DE EVALUACIÓN AUTOMATIZANDO MI CREATIVIDAD



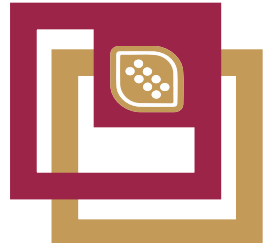
SIGA

Formación Laboral Básica Desarrollo de software	Módulo IV. Aplicaciones Móviles	Submódulo 2. Diseño y Programación en Robótica
Situación didáctica. Automatizando Mi Creatividad.		Nombre de los estudiantes:
Docente:		
Semestre. 6to Grupo.	Turno.	Fecha.
CLBDS8. Identifica propuestas de robótica básica, a través de sus fundamentos, arquitectura y anatomía; en cualquier ámbito, afrontando retos y asumiendo la frustración como parte de un proceso.		
Evidencia de aprendizaje. Prototipo de robot		

Instrucciones: Marque con una (X) para resaltar si cumplió con los criterios de la actividad. Sume los puntos para obtener la calificación.

INDICADORES	%	CRITERIOS			
		EXCELENTE	BUENO	SUFICIENTE	INSUFICIENTE
Fase de Desarrollo	30%	Demuestra observación en un problema del contexto y desarrolla las 7 fases para la construcción de un robot. 30	Presenta la observación de un problema del contexto y desarrolla al menos 5 fases para la construcción de un robot. 25	Presenta la observación de un problema del contexto y desarrolla al menos 3 fases para la construcción de un robot. 15	Solamente presenta la observación de un problema del contexto sin considerar las fases de desarrollo. 5
Fase de Producción	20%	El prototipo considera la parte eléctrica, electrónica y mecánica. 20	El prototipo considera solamente dos de las 3 partes de la robótica. 15	El prototipo considera al menos una de las 3 partes de la robótica. 10	Solamente presenta un prototipo de robot sin considerar el fundamento de la robótica. 5



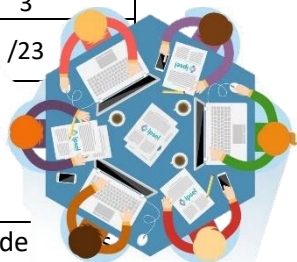


INDICADORES	%	CRITERIOS			
		EXCELENTE	BUENO	SUFICIENTE	INSUFICIENTE
Programación de Bloques	20%	Muestra la programación de los circuitos, placas, componentes eléctricos a través de los bloques de código. 20	Muestra la programación de los circuitos y los componentes eléctricos a través de los bloques de código. 15	Muestra la programación de los circuitos a través de los bloques de código. 10	Muestra la programación del robot sin bloques. 5
Presentación y exposición	20%	Los estudiantes exponen los detalles de su prototipo de robot y muestran los elementos en su programación. 20	Los estudiantes exponen los detalles de su prototipo de robot y muestran algunos elementos en su programación. 15	Los estudiantes exponen los detalles de su prototipo de robot sin mostrar los elementos en su programación. 10	Solo exponen el prototipo de robot, pero no describen los detalles en la programación del robot. 5
Todo el equipo ha participado en la creación del prototipo de robot	10%	Todos y cada uno de los miembros del equipo han participado y colaborado en la creación del prototipo de robot. 10	La mayoría de los integrantes del equipo han participado y colaborado en la creación del prototipo de robot. 8	Pocos miembros del equipo han participado y colaborado en la creación del prototipo de robot. 5	La construcción del prototipo de robot ha sido realizada por un solo miembro del equipo y los demás observan. 3
Total	100%	/100	/83	/50	/23



Firma del evaluador. _____

Conflicto cognitivo	Menciona el software que consideras en el diseño y programación de
	¿Cuáles son las fases de desarrollo para el diseño de un robot?
	Menciona los tipos de robots y ¿Qué formas pueden tomar los robots?
	¿Qué aplicaciones en tu contexto puedes darle a la robótica?



Anexos

HIMNO DEL COBATAB



¡Oh!, Colegio de Bachilleres, impetuosa y querida
institución casa fiel del conocimiento,
hoy te canto este himno con amor,
eres rayo de esperanza del mañana,
eres la voz de la verdad.

¡Oh!, Colegio de Bachilleres, eres
luz en medio de la oscuridad.
//Colegio de bachilleres conducta
clara y firme decisión
Colegio de bachilleres tu misión
para siempre es ser mejor.//

En Tabasco se ha sembrado
la semilla que un día germinara,
el impulso de la vida modernista
en progreso de toda la sociedad,
es tu memorable historia gran
orgullo para toda la región,
educación que genera cambio,
ejemplo digno en cada generación.



PORRA INSTITUCIONAL

¡Somos!
¡Somos!

Jóvenes Bachilleres
Jóvenes Bachilleres

Con Valor y Lealtad
De Norte a Sur
De Este a Oeste

Somos líderes Bachilleres del Sureste
COBATAB Unido, COBATAB Fortalecido.
Este encuentro lo gano porque lo gano
Como dijo el Peje, me canso ganso.

¡Somos!
¡Somos!

Jóvenes Bachilleres
Jóvenes Bachilleres

¡Somos!
¡Somos!

Jóvenes Bachilleres
Jóvenes Bachilleres

COBATAB Unido, COBATAB Fortalecido.



“Educación que genera cambio”

COBACHITO



Colegio de Bachilleres,
Está de fiesta señores
Pues todos sus estudiantes
Hoy celebran con honores.

Que ya llegó la alegría
Es hora de motivar
Bailemos con algarabía
Cobachito nos guiará.

Allá por el acahual
En los ríos de Tabasco
Aconchado en unas ramas
O nadando sin parar.

Un manatí se ha ganado
El cariño de la gente
Cobachito le han llamado
Y no para de bailar.

Cobachito, con él vamos a ganar
Cobachito, eres espectacular
Cobachito, respetamos tu hábitat
Cobachito, mascota del COBATAB.

Mientras la orquesta se escucha
Y la porra se emociona
Los jóvenes bachilleres
A una voz ovacionan.

Con orgullo representan
A una gran institución
COBATAB está presente
Y Cobachito ya llegó.

Cobachito...





COBATAB

COLEGIO DE BACHILLERES
DE TABASCO

“Educación que Genera Cambio”

