programa cultura digital

hacemos ESCUELA

Ministerio de EDUCACIÓN

GOBIERNO DE LA PROVINCIA DE CÓRDOBA
Seguimos haciendo

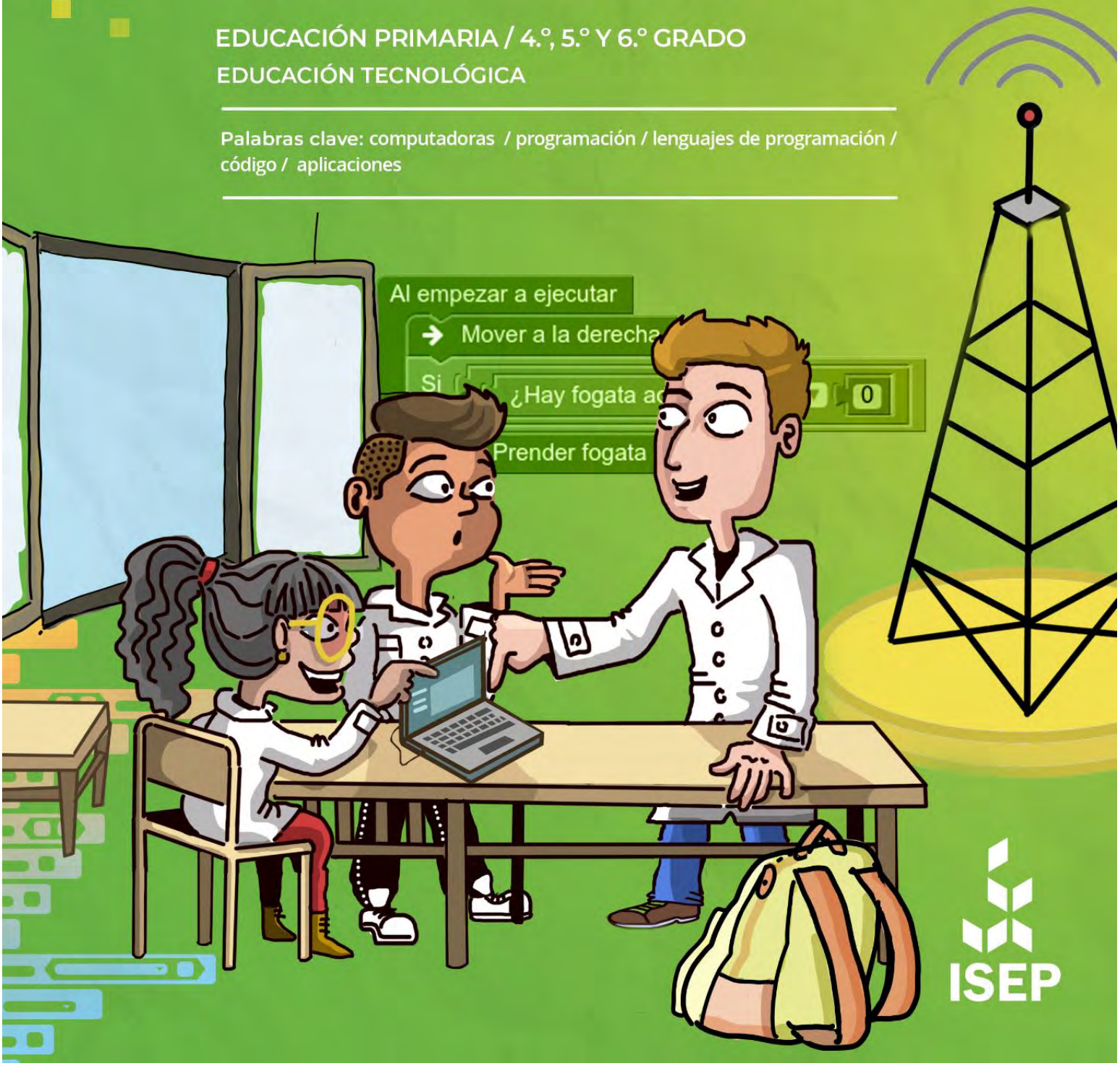
Programas y lenguajes de programación • Parte I

EJE: ALGORITMOS Y PROGRAMACIÓN

EDUCACIÓN PRIMARIA / 4.°, 5.° Y 6.° GRADO

EDUCACIÓN TECNOLÓGICA

Palabras clave: computadoras / programación / lenguajes de programación / código / aplicaciones



ISEP

Programas y lenguajes de programación • Parte I

Programa *Cultura Digital*

EDUCACIÓN PRIMARIA / SEGUNDO CICLO

EJE: ALGORITMOS Y PROGRAMACIÓN

Presentación

Esta clase está organizada en dos partes. Esta primera parte aborda el concepto de programa de computadora con el propósito de que los y las estudiantes puedan reconocerlo como un producto creado por personas que sirve para la resolución de algún problema o para la ejecución de una tarea determinada por parte de un dispositivo computacional. La propuesta también busca identificar la existencia de diferentes lenguajes de programación como construcciones artificiales de las personas para comunicar instrucciones a las computadoras. La segunda parte de la clase continuará trabajando sobre la idea de que estos lenguajes tienen características propias como, por ejemplo, determinadas reglas de escritura y un entorno de desarrollo. A través de la creación de un programa sencillo, se pretende que puedan comprender las particularidades de los lenguajes de programación y las tareas de las personas que programan.

Antes de empezar...

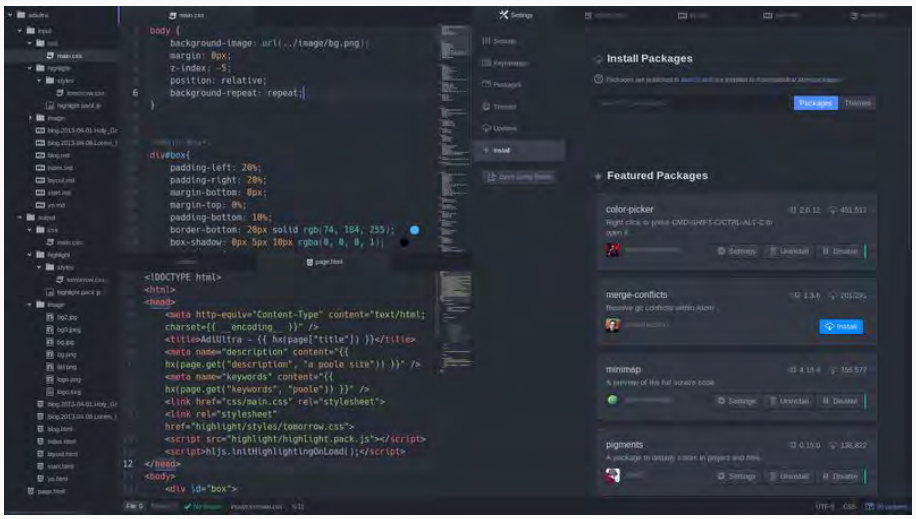
¿Qué diferencia hay entre programa y lenguaje de programación?

Se llama **programa** a todo conjunto de instrucciones escrito con un código determinado que puede ser interpretado y ejecutado por una máquina.

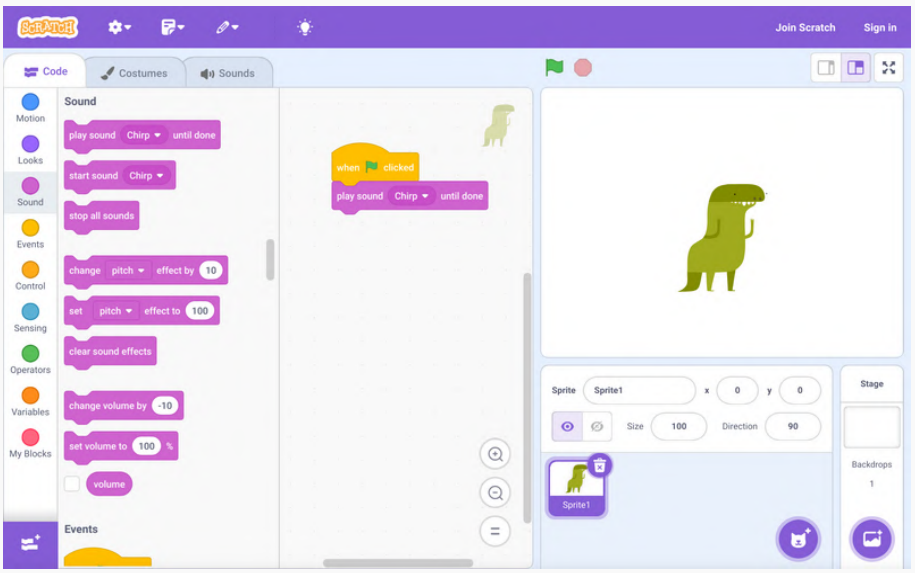
Un **lenguaje de programación** es el conjunto de reglas que se utilizan para escribir el código de un programa. Es lo que se llama la sintaxis del lenguaje. Por ejemplo, para escribir una carta o un cuento utilizamos letras que forman palabras, mientras que esas palabras forman oraciones y se escriben en un orden determinado. Además, utilizamos signos de puntuación como el punto, la coma, el punto y coma, los paréntesis, etc. En los lenguajes de programación, también hay reglas de escritura que se deben respetar para que el programa funcione.

Programar es crear un ***software*** (o programa) mediante la escritura, prueba y depuración del código. Es decir, la programación informática es el trabajo, que realiza una persona, de escribir instrucciones para una computadora. Estas instrucciones se denominan programas.

Los lenguajes de programación se pueden escribir en un editor de texto o en un **entorno de desarrollo integrado** (IDE, por sus siglas en inglés). El entorno de desarrollo integrado, que a menudo se lo denomina entorno de programación, es el soporte en el cual se escribe el programa. Es una aplicación de *software* que sirve para que una persona pueda escribir códigos de manera más sencilla y eficiente. Así como utilizamos un procesador de texto para escribir una planificación escolar, una secuencia didáctica o una monografía, quienes programan utilizan un entorno de desarrollo integrado para escribir los programas informáticos.



Entorno de desarrollo integrado de un lenguaje textual.



Entorno de desarrollo integrado de un lenguaje de bloques.

Para conocer otras propuestas didácticas acerca de este temática, pueden consultar los manuales para docentes de la colección "Ciencias de la Computación para el Aula", de Fundación Sadosky, particularmente:

- Secuencia didáctica "Programas". Areces, C.; Benotti L.; Cortez, J. y otros *(2018). Ciencias de la computación para el aula: 2do. ciclo de primaria: libro para docentes*. Ciudad Autónoma de Buenos Aires: Fundación Sadosky.
- Secuencia didáctica "¿Qué es programar?". Fundación Sadosky *(2024). Actividades para aprender a Program.AR*. Ciudad Autónoma de Buenos Aires: Fundación Sadosky.

También, pueden consultar las siguientes entradas de la Brújula de la informática:

- Programación > Conceptos > Programa
- Programación > Conceptos > Lenguaje de programación
- Programación > Herramientas > Entornos de programación textual
- Programación > Herramientas > Entornos de programación por bloques

Momento 1.
¿Qué son los programas?

Para el desarrollo de este momento, deberán contar con la ficha de la **Actividad 1**, disponible al final de la clase.

En este primer momento, se plantea la pregunta sobre qué son los programas. Se busca identificar las cualidades de los programas informáticos al diferenciarlos de otros conjuntos de instrucciones o secuencias que no son programas informáticos.

El o la docente inicia la clase preguntando al grupo si saben qué es un programa informático y qué ideas tienen acerca de eso. Se espera que este sea un momento de indagación; se sugiere entablar un diálogo que motive a los y las estudiantes a expresarse y escucharse. Ofrecemos aquí algunas preguntas a modo de ejemplo.

Para conversar

- ¿Alguien sabe qué es un programa de computadora? ¿Y el *software* qué es? ¿Las aplicaciones son programas?
- ¿Para qué sirve un programa? ¿Usaron alguna vez uno? ¿Cuándo? ¿Para qué?
- ¿Cómo se hacen los programas? ¿Quiénes los hacen?
- ¿Qué hace falta para que un programa funcione en la computadora o en el celular o en una *tablet* o en cualquier otro dispositivo?

Nota para el o la docente

Los términos ***software*** y **aplicación** (o *app*) se utilizan también como sinónimos de programa informático. Puede ocurrir que, al intentar responder sobre lo que entienden por programa informático, propongan definiciones o ejemplos utilizando estos términos, dado que los utilizan más comúnmente en su vida cotidiana, sobre todo, por el uso de las aplicaciones en los teléfonos celulares.

Concluido este diálogo, el o la docente hace una síntesis de los aportes de los y las estudiantes y les propone completar la **Actividad 1**, en la que deberán identificar, a partir de una serie de imágenes, las tarjetas que corresponden a un programa informático. La **Actividad 1** se puede realizar con las tarjetas en papel o utilizando las imágenes del Anexo para verlas en un dispositivo, computadora o proyector.

Luego de que hayan completado la **Actividad 1**, el o la docente solicita que algunos estudiantes comenten sus respuestas con el resto de la clase y expliquen por qué decidieron, en cada caso, identificar como programa informático o no a las imágenes que se les presentaron. Para realizar una puesta en común, les plantea a sus estudiantes las siguientes preguntas.

Para conversar

- ¿Cómo pudieron identificar cuáles son programas informáticos?
- ¿Qué diferencias tienen los programas informáticos de las otras cosas que no son programas de computadora?
- ¿Por qué tienen esa forma particular de estar escritos?
- ¿Tienen palabras en algún idioma? ¿En cuál? ¿Por qué les parece que puede ser esto?

Posteriormente, les explica que los programas informáticos tienen ciertas características fundamentales: instrucciones en un orden definido, que están escritas por una persona o un grupo de personas para que sean entendidas por un dispositivo computacional, y que ese dispositivo computacional debe realizar alguna tarea a partir de esas instrucciones.

Para concluir

Un programa es una serie de instrucciones que dirige a una computadora para llevar a cabo tareas concretas, que pueden variar desde operaciones básicas hasta aplicaciones complejas. Su desarrollo requiere crear, probar y ajustar el código para garantizar que cumpla con su función prevista.

Momento 2.
¿Qué son los lenguajes de programación?

Para el desarrollo de este momento, deberán contar con la ficha de la **Actividad 2** y con la ficha de la **Actividad 3**, disponibles al final de la clase.

En este momento se plantea la pregunta sobre qué son los lenguajes de programación. Para eso, primero se busca que los y las estudiantes comprendan las diferencias que existen entre la forma en que se comunican las personas y la forma en la que se dan instrucciones a una computadora.

El o la docente comienza la clase explicando que las personas, por lo general, nos comunicamos a través de un sistema de comunicación verbal, como puede ser un determinado idioma. Además, en ciertos casos en los que la comunicación verbal no es posible, se utilizan otros sistemas, como, por ejemplo, un sistema de comunicación gestual, como el lenguaje de señas.

Para contextualizar los lenguajes de programación como tecnologías contemporáneas, el o la docente comenta que el desarrollo de tecnologías de la comunicación y las técnicas aplicadas a los sistemas de comunicación fueron evolucionando a lo largo de la historia de la humanidad.

Puede contarle a la clase que, en la Prehistoria, se utilizaban técnicas de pintura rupestre, que es el tipo de pintura que se hacía en las paredes de las cavernas o cuevas, para comunicar hechos de la vida de esas culturas y civilizaciones. Luego, miles de años después, se inventó el papiro, un soporte para la escritura parecido al papel actual que usamos en nuestras carpetas de la escuela, y que se realizaba con una planta que se encontraba a orillas del río Nilo, en Egipto. El papiro quedó en desuso cuando se perfeccionó la técnica de fabricación y se inventó el papel como lo conocemos hoy.

Una vez desarrollada esa explicación introductoria, el o la docente propone una serie de preguntas para estimular el pensamiento crítico y la comprensión básica de cómo las computadoras reciben y procesan instrucciones. Se busca que identifiquen las diferencias entre la comunicación **persona - persona** y la comunicación **persona - máquina**. A modo de ejemplo, proponemos algunas posibles preguntas, sin necesidad de abordar todas ellas y con el objetivo de retomar lo que el grupo vaya aportando.

Para conversar

- ¿Alguien sabe cómo reciben las instrucciones las computadoras?
- ¿Cómo piensan que se le puede dar una orden a una computadora para que haga algo? Por ejemplo, ¿cómo le dirían a una computadora qué hacer si quieren que realice un dibujo?
- Si una computadora pudiera entenderlos cuando hablan, ¿qué le dirían para que abra su juego favorito?
- ¿Por qué creen que necesitamos usar códigos o lenguajes especiales para que la computadora haga cosas? ¿No podría, simplemente, entendernos como si habláramos con ella?
- ¿Cómo creen que las computadoras recuerdan todas las instrucciones que les damos?
- ¿Qué creen que pasa si le dan a la computadora instrucciones confusas o equivocadas?

Una vez concluido el diálogo, se realiza una síntesis de lo conversado y se apunta que, cuando se necesita darle instrucciones a una máquina, hay una complicación: que las máquinas no entienden el lenguaje que hablan las personas.

El o la docente explica que, con la invención de la computadora, se planteó la necesidad de una nueva forma de comunicación: la comunicación entre las personas y las máquinas. Y refuerza la idea de que, para darle instrucciones a una computadora, necesitamos un sistema de comunicación que pueda ser entendido por la máquina. Es cuando surge la necesidad de desarrollar lenguajes de programación.

Luego, continúa explicando que estos lenguajes deben contener instrucciones bien detalladas y precisas, y en un determinado orden, de lo que debe hacer la máquina, porque las computadoras no entienden las cosas de la misma manera que las personas.

Propone un ejemplo sencillo para explicar por qué las instrucciones que les damos a las computadoras deben ser precisas y detalladas. Puede pedirle a un estudiante o a una estudiante que pase a dibujar un cuadrado en el pizarrón. Luego de eso, le explicará a todo el grupo que, si le pidiese a una computadora que haga lo mismo, la computadora, probablemente, no sabría qué es un cuadrado. Y suponiendo que supiera que un cuadrado es una figura plana, cerrada por cuatro líneas rectas iguales que forman cuatro ángulos rectos, aun así no sabría de qué color y tamaño dibujar el cuadrado. Se quedaría esperando las instrucciones precisas para dibujar el cuadrado del color y del tamaño que le indiquemos.

Nota para el o la docente

Las primeras computadoras recibían las instrucciones de lo que debían realizar a través de tarjetas perforadas. Esta idea surgió, originalmente, para el funcionamiento de los telares, que eran máquinas para tejer que utilizaban tarjetas perforadas con las instrucciones del tejido que se necesitaba realizar. Las tarjetas perforadas eran tarjetas de cartón con agujeros que representaban datos en código binario (había agujero o no había agujero). Las computadoras leían las tarjetas usando máquinas lectoras que detectaban los agujeros y convertían la información en señales electrónicas. A medida que la tarjeta pasaba por el lector, los agujeros alineados permitían que la luz o un contacto eléctrico (según el diseño del lector) pasara a través de ellos o se interrumpiera, registrando así los datos. Cada tarjeta contenía una línea de datos o instrucciones, y se leían en secuencia para procesar programas o datos. Este método fue crucial en las primeras computadoras para introducir y almacenar información. En la actualidad, ya no se usa más este sistema de tarjetas perforadas porque se inventaron otras técnicas y se desarrollaron nuevas tecnologías para la entrada de datos en las computadoras.

Antes de comenzar la **Actividad 2**, el o la docente explica que los lenguajes de programación modernos, como los que se utilizan en la actualidad, tienen su origen a fines de la década de 1950; y que una posible clasificación de los lenguajes de programación es diferenciar entre los lenguajes textuales y los lenguajes de bloques.

Nota para el o la docente

Si bien existen diferentes tipos de lenguajes de programación y se pueden clasificar de distintas maneras, en esta clase solamente se va a diferenciar entre los lenguajes de programación textuales y los lenguajes de programación de bloques.

A continuación, se les propone a los y las estudiantes que resuelvan la **Actividad** 2 y se les presenta una línea de tiempo en la cual se mencionan algunos de los lenguajes de programación desde 1957 hasta la actualidad, junto con una breve información sobre cada uno de ellos. Este material de la línea de tiempo es el insumo que deben leer y consultar para la realización de la **Actividad 2**.

El o la docente explica que, en la **Actividad 2**, deberán completar una tabla buscando la información solicitada en la línea de tiempo.

Luego, una vez finalizada la **Actividad 2**, se hace una puesta en común y se verifica si la tabla fue completada de manera correcta. Posteriormente, se sugiere concluir la actividad con un intercambio de opiniones sobre qué información les pareció más interesante, si se imaginaban que podía haber programas de computadora tan viejos; quiénes pensaban que eran las personas que desarrollaban esos programas, o si tenían conocimientos previos sobre el tema.

En esta actividad se pretende recuperar ciertos datos sobre algunas mujeres que han trabajado en el desarrollo de lenguajes de programación, porque no es común encontrar esta perspectiva de género en las búsquedas sobre este tipo de información en internet. Se puede consultar también la clase “Programación para todos y todas” (Educación Primaria, Segundo Ciclo, Eje Ciudadanía), disponible en el sitio Hacemos Escuela en la cual se aborda la temática de las mujeres vinculadas con la historia de la programación.

Algunas posibles preguntas para iniciar el debate de cierre de la **Actividad 2** pueden ser las siguientes.

Para conversar

- ¿Qué fue lo que más les llamó la atención de la información que leyeron en la línea de tiempo?
- ¿Se imaginaban que podía haber lenguajes de programación tan antiguos?
- ¿Para qué se crearon esos programas? ¿Para qué sirven?
- ¿Quiénes crearon estos lenguajes de programación? ¿Personas trabajando solas?
- ¿Habían leído alguna vez que un lenguaje de programación haya sido creado por una mujer?
- ¿Creen que en la actualidad hay más hombres que mujeres que desarrollan lenguajes de programación o al revés? ¿Por qué será?

A continuación, el o la docente explica el desarrollo de la **Actividad 3**. Es un juego que se juega en grupos de **cuatro** o más estudiantes. Se arman equipos con la misma cantidad de integrantes (dos contra dos; tres contra tres o cuatro contra cuatro, por ejemplo).

El juego consiste en deducir, a través de diferentes pistas, qué tarjeta le tocó al equipo contrario.

Una vez finalizada la **Actividad 3**, el o la docente hace un cierre de la clase poniendo énfasis en que los lenguajes de programación no son entidades abstractas que surgen de ideas locas de científicos computacionales, matemáticos e investigadores; y que tienen un propósito y son creados por equipos de personas que estudian y trabajan mucho para eso. Explica, también, que cada lenguaje de programación tiene virtudes y limitaciones, y que son un reflejo de la creatividad humana para mejorar y perfeccionar la comunicación con las computadoras.

Para concluir

En términos sencillos, un lenguaje de programación es el sistema de instrucciones que usamos para comunicarnos con las computadoras. Este lenguaje nos permite enviarles instrucciones escritas en una forma que las computadoras pueden entender y, por lo tanto, logran realizar lo que queremos que hagan.

Los lenguajes de programación no son simplemente conceptos teóricos o inventos raros que surgen de ideas que tienen personas científicas o que trabajan en universidades o empresas de tecnología. Son herramientas creadas por personas, que estudian para eso, con el objetivo de resolver problemas específicos. A lo largo de la historia, cada lenguaje ha surgido en respuesta a necesidades particulares: desde la gestión de datos en los inicios de la computación hasta la creación de aplicaciones móviles, y la inteligencia artificial en la actualidad. Cada lenguaje refleja el contexto en el que fue desarrollado, las limitaciones tecnológicas de su época y los problemas que las personas que los crearon buscaban resolver. Así, los lenguajes de programación evolucionan junto con la tecnología y con las demandas de la sociedad, y son siempre un reflejo de la creatividad humana y de la búsqueda constante de soluciones.

FICHA TÉCNICA

Clase: Programas y lenguajes de programación. Parte I
Nivel: Primario
Cursos sugeridos: 4°, 5° y 6° grado
Espacio curricular: Educación Tecnológica

Eje: Algoritmos y Programación

Objetivos:

- Comprender el concepto de programa informático como conjunto de instrucciones escritas, de forma tal que pueden ser interpretadas y ejecutadas por una computadora.
- Reconocer los lenguajes de programación como tecnologías creadas por personas para resolver problemas o facilitar tareas con un fin determinado.
- Identificar algunos datos acerca de ciertos lenguajes de programación como, por ejemplo, el propósito de su creación y desarrollo, y su lugar de origen.
- Programar instrucciones básicas en un lenguaje de bloques para completar un programa que ya fue comenzado por otra persona.

Aprendizajes y contenidos:

- Identificación de qué es un programa informático a partir de sus características principales.
- Reconocimiento de los lenguajes de programación como tecnologías desarrolladas por las personas para poder comunicar instrucciones a las computadoras.
- Programación, testeo y ejecución de un programa con un objetivo específico a través de lenguaje de bloques.

Sobre la producción de este material

Los materiales de *Cultura Digital* y *Hacemos Escuela* se producen de manera colaborativa e interdisciplinaria entre los distintos equipos de trabajo.

Autoría: Pablo Mercanti

Equipo de producciones de materiales hipermediales y audiovisuales

Didactización: Nadia Gonnelli
Corrección literaria: Juan Pablo Spinassi
Diseño: Carolina Cena
Ilustración: Raúl Edgardo Soria Fernández
Coordinación de producción: María Florencia Scidá
Coordinación general: Paula Fernández, Luciana Dadone y Ana Gauna

Coordinación del Programa *Cultura Digital*: Natalia Zalazar
Coordinación de *Hacemos Escuela*: Fabián Iglesias

Citación:

Mercanti, P. y equipos de producción del ISEP. (2024). Programas y lenguajes de programación Parte I. Programa *Cultura Digital - Hacemos Escuela*. Para el Ministerio de Educación de la Provincia de Córdoba.

Este material está bajo una licencia Creative Commons Atribución-NoComercial 4.0 Internacional.

COMUNIDAD DE PRÁCTICAS:
La clase en plural

La Comunidad de prácticas es un espacio de generación de ideas y reinvención de prácticas de enseñanza, donde se intercambian experiencias para hacer escuela juntos/as. Los/as invitamos a compartir las producciones que resulten de la implementación de esta propuesta en sus instituciones y aulas, pueden enviarlas a hacemosescuela@isep-cba.edu.ar.

ISEP | INSTITUTO SUPERIOR de Estudios Pedagógicos

Los contenidos que se ponen a disposición en este material son creados y curados por el Instituto Superior de Estudios Pedagógicos (ISEP), con el aporte en la producción de los equipos técnicos de las diferentes Direcciones Generales del Ministerio de Educación de la provincia de Córdoba.

Secretaría DE INNOVACIÓN, DESARROLLO PROFESIONAL Y TECNOLOGÍAS EN EDUCACIÓN

Ministerio de EDUCACIÓN

GOBIERNO DE LA PROVINCIA DE CÓRDOBA Seguimos haciendo

Actividad 1.

¿Es un programa?

Identifiquen y marquen cuáles son programas informáticos y cuáles no.

ساحة السلام

باملونا ، تلك المدينة الخضراء الهادئة عاصمة اقليم
نابارا ومقصد الزوار والسائحين ، وايضاً مقصد الباحثين
عن العمل من ابناء الاقاليم ومن خارجه . كما أنها
تحتضن عدداً كبيراً من المهاجرين بتنوع جنسياتهم
ولغاتهم الاصلية . ولا شك أن لهذه المدينة
طبيعة خاصة من حيث تنوع وثراء التجربة اللغوية
بها ، حيث أن عدداً كبيراً من اهلها يمكنه
التعامل بلغتين أي أن مجتمعها يُعد ثنائ
اللغه ، وهذا يعطي للمدينة طابعاً مميزاً ويجعل
العقول اكثر قابلية لتفهم لغات اخرى واحترامها
اضف إلى ذلك أن هناك مدرسة للثنات تسهم
بما لا يدع مجالاً للشك في اثراء التجربة اللغوية
للمجتمع الباملوني .
ولكن هل اللغة العربية مكاناً في هذا المجتمع ؟
تناول اثارته عدة أحرف نقلها لطلاب المستوى
الثاني للغة العربية بمدرسة الثنات إلى قاعة

PANQUEQUES ESPONJOSOS

Ingredientes
(5 porciones aproximadas)

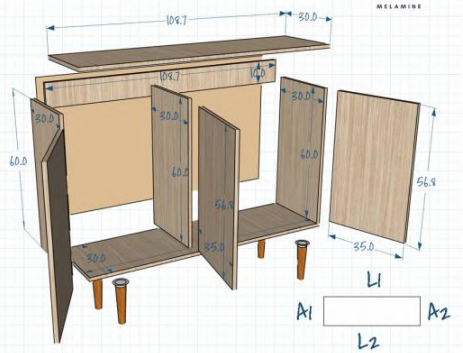
1 huevo
2 cucharadas de aceite
3/4 de taza de leche
2 cucharadas de azúcar
1 pizca de sal
1 taza de harina
1 cucharadita de polvo de hornear

ES IMPORTANTE LA CANTIDAD DE LECHE,
PARA ESTA RECETA
ES UN POCO MÁS DE LA MITAD.

```
setOfNumbers = []

print("How many random numbers do you want to generate?")
max = int(input())

for i in range (max):
    setOfNumbers.append(random.randrange(1,101,1))
setOfNumbers.sort()
print(setOfNumbers)

print("Which number do you want to find in the set of random numbers")
searchNumber = int(input())

firstPos = 0
lastPos = max-1
found = False

while (not found and firstPos <= lastPos):
    midPos = int((firstPos + lastPos)/2)

    if (searchNumber == setOfNumbers[midPos]) :
        found = True
    else :
        if (searchNumber < setOfNumbers[midPos]):
            lastPos = midPos - 1
        else :
            firstPos = midPos + 1


if (found) :
    print("Your item is in the list")
else :
    print("Your item is not in the list")
```

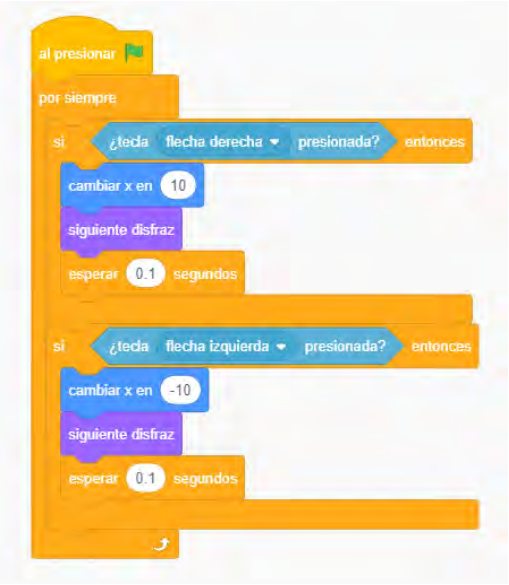
Melamina 15mm
Fresno Europeo

HAMMER
MELAMINE



DESCRIPCION	PIEZAS CANTIDAD	PIEZAS LARGO	PIEZAS ANCHO	TAPACANTO LI	TAPACANTO L2	TAPACANTO A1	TAPACANTO A2	PERF. CANT	PERF. LADO
TECHO	1	109.7	30	D	D	D	D		
PISO	1	109.7	30	D	D	D	D		
LATERAL IZQ/DER.	2	60	30	D	D				
COLUMNA CENTRAL	1	60	30	D					
TRAVESA	1	109.7	10	D					
PUERTAS	3	56.8	36	G	G	G	G	2	L

* Medidas solo referenciales



Serie Cooler

Serie de estabilizadores electrónicos de voltaje, diseñados para variaciones significativas de voltaje, donde la regulación se necesita con más precisión. Apropiados para proteger equipos de oficina, electrodomésticos y equipos de comunicaciones.

Características

Códigos Productos	003-0001-000030 003-0001-000041	
Capacidad	VA	2000 VA
	Watts	1400 W
Calidad de regulación	V. Entrada	95-135 VAC
	V. Salida	115 +/- 6%
	Frecuencia	60 Hz
	Supresores	80 J 150 VAC
	Conmutación	8 ms con Relés
	Cable conexión	cable de conexión 1.2 m con clavija
Medidas	Peso	4.2 kg
	Dimensiones	143 x 215 x 90 mm

Para descargar las imágenes pueden hacer clic aquí.

Actividad 2.

Algunos lenguajes de programación a lo largo de la historia

Lean la línea de tiempo disponible en la **Ficha 1**, en la cual podrán aprender un poco sobre algunos de los lenguajes de programación más conocidos a lo largo de la historia.

Luego, deben completar la siguiente ficha. Pueden tener a mano la línea de tiempo para consultar algún dato por si no lo recuerdan.

Lenguaje de programación	Año en que fue creado	Lugar en el que fue creado	Personas que lo desarrollaron	¿Cuál fue el propósito para el cual se desarrolló?
Cobol				
Logo				
JavaScript				
Python				

Scratch				

Ficha 1.
Algunos lenguajes de programación a lo largo de la historia

Breve historia de algunos de los lenguajes de programación más importantes

1957

FORTRAN

Desarrollado por John Backus y su equipo en IBM, en Estados Unidos.
Este es uno de los primeros lenguajes de programación. Fue creado para ayudar a los científicos e ingenieros a hacer cálculos matemáticos complejos rápidamente. Su nombre es una mezcla de palabras que significan "traducción de fórmulas" (***For****mula* ***Tran****slation*), porque ayuda a las máquinas a entender matemáticas.

John Backus

Fuente: Wikimedia.

John McCarthy

Fuente: Wikimedia.

1958

LISP

Desarrollado en Estados Unidos por John McCarthy y su equipo de trabajo en el Instituto Tecnológico de Massachusetts (MIT).
Fue creado con el propósito de ayudar en el desarrollo de inteligencia artificial y resolver problemas lógicos. Se centró en el manejo de ideas y datos de manera avanzada.
John McCarthy fue un científico computacional considerado el padre de la inteligencia artificial, porque fue la primera persona que propuso ese nombre. En 1957, McCarthy fundó, junto con Marvin Minsky, el Laboratorio de Inteligencia

Artificial del MIT.

1959

COBOL

Este lenguaje de programación fue desarrollado en 1959, y todavía lo utilizan empresas y programadores. Sirve para el manejo de datos y transacciones comerciales en negocios y gobiernos. Ayudó a automatizar cálculos y reportes en empresas.
COBOL fue creado gracias a un ambicioso proyecto en el que participaron fabricantes de computadoras, empresas y organismos usuarios de estas, y el Departamento de Defensa de Estados Unidos. El objetivo era crear un lenguaje común para todas las computadoras de la época y también las futuras. La principal desarrolladora de COBOL fue Grace Hopper, científica computacional y oficial de la Marina de Estados Unidos. Inventora del lenguaje de programación FLOW-MATIC, también se la considera la madre de COBOL.

Grace Hopper y estudiantes, frente a una UNIVAC I

Fuente: Wikimedia.

Seymour Papert con un robot, la "Tortuga Logo", para que los niños jueguen y resuelvan problemas.

1967

LOGO

Este lenguaje de programación fue diseñado por Wallace Feurzeig, Seymour Papert y Cynthia Solomon. Logo fue creado para ayudar a los niños a aprender a programar. Usa una tortuga en la pantalla que sigue instrucciones para dibujar formas, lo que hace el aprendizaje divertido.

Fuente: Wikipedia.

1972

C

Es un lenguaje de programación de propósito general que influenció muchos lenguajes posteriores. Fue creado por Dennis Ritchie y Brian Kernighan en Bell Labs, Nueva Jersey, Estados Unidos. Es un lenguaje muy importante porque muchas otras herramientas y lenguajes modernos están basados en él. Se usa para hacer todo tipo de programas, desde juegos hasta sistemas operativos. El propósito fue crear un lenguaje eficiente para desarrollar sistemas operativos y aplicaciones. Sirvió como base para muchos lenguajes modernos y es considerado el primer lenguaje de programación verdaderamente moderno y portátil.

Dennis Ritchie (de pie)**, en Bells Labs en 1972**

Fuente: Computer History Museum.

1991

PYTHON

Este lenguaje de programación fue creado por un científico computacional neerlandés llamado Guido van Rossum, en la Fundación Centro Matemático (CWI), de Ámsterdam, Países Bajos.

Guido Van Rossum en una conferencia de Google en 2006

Fuente: Wikimedia.

Anteriormente, él mismo había creado un lenguaje de programación llamado ABC, y Python fue un sucesor de ese otro lenguaje. Python es uno de los lenguajes más utilizados en el mundo porque es fácil de aprender, intuitivo y potente. Además, el código escrito en Python pretende ser tan comprensible como cualquier texto en inglés. Python se encuentra entre los lenguajes de programación más demandados en las ofertas de trabajo. ¿Sabías que su nombre proviene de un grupo de comedia británico que le gusta mucho a Guido van Rossum?

1991

HTML

Este lenguaje fue desarrollado por Tim Berners-Lee, un científico computacional inglés, quien también inventó la World Wide Web. HTML es el lenguaje estándar para la creación de páginas web y aplicaciones web, y se utiliza en conjunto con un lenguaje de diseño gráfico llamado CSS y con un lenguaje de programación llamado JavaScript, que permite darles interactividad a las páginas y aplicaciones web.
HTML es un lenguaje de marcado y no tiene funciones aritméticas o variables, como poseen los lenguajes de programación.
HTML sirve para generar hipervínculos o enlaces entre diferentes palabras, para acceder a diferentes partes de un documento.

Tim Berners-Lee en Londres en 2014

Fuente: Wikipedia.

Brendan Eich

1995

JAVASCRIPT

Este lenguaje fue creado por Brendan Eich, un programador estadounidense que trabajaba en

Fuente: Wikipedia.

una empresa llamada Netscape. Brendan Eich comenzó a trabajar en la creación de un lenguaje para mejorar el navegador Netscape Navigator 2.0. Llamó a su nuevo lenguaje de programación LiveScript, y tuvo mucho éxito.
Pero justo antes de que el navegador web se presentase a los usuarios, Netscape decidió cambiar el nombre del lenguaje a JavaScript y se unió a otra empresa llamada Sun Microsystems para seguir mejorándolo.
JavaScript es muy utilizado para desarrollar interfaces web interactivas. Todos los sitios que vemos en internet y las aplicaciones en los dispositivos móviles utilizan este lenguaje para que sus contenidos se vean más dinámicos y atractivos.

2007

SCRATCH

Este es un lenguaje de programación de bloques. Fue desarrollado por Mitchel Resnick, un físico y científico computacional estadounidense, quien trabaja como director del equipo de Aprendizaje Permanente en el Laboratorio de Medios del Instituto Tecnológico de Massachusetts (MIT), Estados Unidos.
Resnick y su equipo de trabajo desarrollaron este lenguaje con el objetivo de hacer más fácil la enseñanza y el aprendizaje de la programación en las escuelas.

Mitchel Resnick

Fuente: Wikimedia.

Scratch se ejecuta en un entorno de programación visual y, para escribir el código, se utilizan bloques de colores que se arrastran y se encastran, conectándose entre sí de forma vertical.
Scratch está inspirado en las ideas previas de Seymour Papert y del lenguaje qué él había creado: Logo. Su nombre proviene de una técnica de pasar música en la cual los DJ hacen remezclas de los discos y los giran sobre las bandejas de tocadiscos.

Pilas Bloques, uno

2016

de los proyectos de la Fundación Sadosky

Fuente: Wikipedia.

PILAS BLOQUE

Pilas Bloques es un lenguaje de programación de bloques. Está basado en Blockly, un lenguaje similar a Scratch, pero que fue creado anteriormente por Google.

Pilas Bloques fue desarrollado en Argentina por Program.AR–Fundación Sadosky con la colaboración de personas que trabajan en el desarrollo del sistema operativo Huayra, del Ministerio de Educación y de diferentes universidades.

Esta aplicación hace uso del programa Pilas Engine,herramienta para programar textualmente y que fue creada por el programador argentino Hugo Ruscitti.

Su nombre se debe a que se basa en Pilas Engine, pero incorpora bloques.

Pilas Bloques surgió con el propósito de tener una aplicación propia, y pensada para el contexto de nuestro país, que sirviera para enseñar programación a cualquier niña, niño o adolescente de la escuela argentina.

¿Saben qué lenguaje de programación es?

¡En esta actividad, vamos a jugar! El juego se realiza en grupos de cuatro o más estudiantes. Se arman equipos con la misma cantidad de integrantes: dos contra dos; tres contra tres o cuatro contra cuatro, por ejemplo.

El objetivo es deducir, a través de diferentes pistas, qué lenguaje de programación es el que está en la tarjeta que le tocó al equipo contrario.

Instrucciones

Hay un mazo con 10 tarjetas; una por cada lenguaje de programación de los que se trabajaron en la Actividad 2.

Se mezcla el mazo con todas las tarjetas y se reparten 5 tarjetas para cada equipo. No se deben mostrar las tarjetas al equipo contrario. Cada tarjeta contiene cinco pistas diferentes. Se decide -al azar o acordando entre los equipos- qué equipo comenzará leyendo las pistas y qué equipo deberá descubrir de qué lenguaje de programación se trata.

Una vez que fue leída la pista 1, el equipo contrario puede arriesgarse a decir el nombre del lenguaje de programación sobre el cual trata esa tarjeta. Si no acierta, ya no puede continuar pidiendo pistas y es el turno de adivinar del equipo contrario. Si no se arriesga a decir el nombre, puede pedir que les lean la segunda pista; y así sucesivamente hasta la última pista. Cuando ya se leyeron todas las pistas, se cambia de turno. Ahora, el equipo que leyó es el que debe responder, y el otro equipo leerá las pistas de su primera tarjeta.

Si responden correctamente cuál es el nombre del lenguaje de programación sobre el que trata esa tarjeta, ganarán puntos. Si logran decir la respuesta correcta habiendo utilizado solamente la primera pista, ganarán 20 puntos. Con cada pista adicional que necesiten, son cinco puntos menos. Es decir, con la primera pista ganarán 20 puntos; con la segunda pista, ganarán 15 puntos; con la tercera pista, ganarán 10 puntos; con la cuarta pista, ganarán cinco puntos, y con la quinta pista, ganarán solamente 1 punto.

Las tarjetas a continuación son para imprimir y recortar. ¡A jugar!

Fortran

FORTRAN - PISTA 1

Fue creado para ayudar a los científicos e ingenieros a hacer cálculos matemáticos complejos rápidamente.

FORTRAN - PISTA 2

Su diseño inicial se centró en la manipulación de números y de fórmulas.

FORTRAN - PISTA 3

Es uno de los lenguajes de programación más antiguos, creado en la década de 1950.

FORTRAN - PISTA 4

Fue desarrollado por John Backus y su equipo en IBM, Estados Unidos.

FORTRAN - PISTA 5

Su nombre es una abreviatura de "Formula Translation".

Lisp

LISP - PISTA 1

Este lenguaje tiene una forma muy especial de escribir instrucciones, con muchos paréntesis.

LISP - PISTA 2

Fue creado para hacer que las computadoras puedan trabajar con ideas y conceptos, no solo con números.

LISP - PISTA 3

A los investigadores que trabajan en inteligencia artificial les gusta mucho este lenguaje, porque ayuda a construir programas que pueden "pensar".

LISP - PISTA 4

Fue desarrollado en Estados Unidos por John McCarthy y su equipo de trabajo en el Instituto Tecnológico de Massachusetts (MIT)

LISP - PISTA 5

El nombre de este lenguaje suena como una palabra en inglés que significa "lista", porque trabaja mucho con listas.

Cobol

COBOL - PISTA 1

Este lenguaje fue creado especialmente para ayudar a las empresas a manejar datos de ventas y otros números importantes.

COBOL - PISTA 2

A pesar de ser muy antiguo, sigue siendo utilizado en sistemas de gestión de información en bancos y en grandes empresas.

COBOL - PISTA 3

Aunque parece complicado, su diseño hace que los programas se parezcan a las instrucciones en inglés.

COBOL- PISTA 4

Grace Hopper, mujer pionera en la informática, fue la principal creadora de este lenguaje de programación.

COBOL- PISTA 5

Su nombre significa "Common Business-Oriented Language", que quiere decir “Lenguaje Común Orientado a los Negocios”.

Logo

LOGO - PISTA 1

Se basa en un concepto de aprendizaje constructivista, donde los niños y las niñas aprenden haciendo y experimentando con la

LOGO - PISTA 2

Es un lenguaje educativo que se centra en la enseñanza de conceptos básicos de programación a través de gráficos.

LOGO - PISTA 3

Usa una tortuga que sigue instrucciones para dibujar formas en la pantalla, facilitando el aprendizaje visual.

LOGO - PISTA 4

Fue creado por Seymour Papert y su equipo para hacer la programación accesible y divertida para estudiantes jóvenes.

LOGO - PISTA 5

Su nombre viene de una palabra griega que significa "palabra", "discurso" o “lógica”.

C

C - PISTA 1

Este lenguaje es conocido por su eficiencia y se usa ampliamente en el desarrollo de sistemas y aplicaciones.

C - PISTA 2

Es muy bueno para hacer programas que tienen que controlar el *hardware* de las computadoras.

C - PISTA 3

Muchos otros lenguajes modernos están basados en este lenguaje.

C - PISTA 4

Fue creado por Dennis Ritchie y Brian Kernighan en Bell Labs, Nueva Jersey, Estados Unidos.

C - PISTA 5

El nombre de este lenguaje es una de las primeras letras del alfabeto.

Python

PYTHON - PISTA 1

Este lenguaje de programación es muy popular en ciencia de datos, desarrollo web y automatización.

PYTHON - PISTA 2

Su diseño hace que sea fácil de leer y escribir, lo que lo convierte en un programa muy

PYTHON - PISTA 3

Un programador neerlandés quería crear un lenguaje que fuera fácil de aprender y usar para hacer tareas de

PYTHON - PISTA 4

El nombre de ese programador es Guido van Rossum, y fue quien creó este lenguaje de

PYTHON - PISTA 5

El nombre de este lenguaje de programación está inspirado en un grupo de comedia británico

HTML

HTML - PISTA 1

Se utiliza casi siempre junto con un lenguaje de diseño gráfico llamado CSS y con un lenguaje de programación llamado JavaScript.

HTML - PISTA 2

Utiliza "etiquetas" para definir elementos como encabezados, párrafos e imágenes.

HTML - PISTA 3

Este lenguaje de marcado define cómo se presentan textos, imágenes y otros elementos en los sitios web.

HTML - PISTA 4

Fue creado por Tim Berners-Lee, un científico computacional inglés, como parte de la primera propuesta para la World Wide Web.

HTML - PISTA 5

Su nombre surge de las siglas de las siguientes palabras en inglés: Hyper Text Markup Language.

JavaScript

JAVASCRIPT - PISTA 1

Este lenguaje sirve para el desarrollo de páginas web interactivas.

JAVASCRIPT - PISTA 2

Se ejecuta en el navegador web, permitiendo que las páginas web respondan a acciones como clics y desplazamientos.

JAVASCRIPT - PISTA 3

Este lenguaje se suele utilizar junto con el lenguaje de marcado HTML y el lenguaje de diseño gráfico CSS.

JAVASCRIPT - PISTA 4

Fue creado por Brendan Eich, un programador estadounidense que trabajaba en la empresa Netscape, que luego trabajó en Google y, más tarde, como director ejecutivo del navegador web Brave.

JAVASCRIPT - PISTA 5

Antes de conocerse con su nombre actual, este lenguaje de programación se llamaba LiveScript.

Scratch

SCRATCH - PISTA 1

Este lenguaje de programación fue desarrollado para que los principiantes puedan crear animaciones y juegos sin escribir códigos complejos.

SCRATCH - PISTA 2

Su desarrollo estuvo basado en las ideas previas de Seymour Papert y su lenguaje de programación Logo.

SCRATCH - PISTA 3

Es un lenguaje educativo que usa bloques gráficos que se ensamblan como piezas de un rompecabezas para enseñar programación a estudiantes de escuelas primarias y secundarias en todo el mundo.

SCRATCH- PISTA 4

Este lenguaje fue desarrollado por Mitchel Resnick y su equipo en el Laboratorio de Medios del Instituto Tecnológico de Massachusetts (MIT), en Estados Unidos.

SCRATCH- PISTA 5

Su nombre proviene del nombre en inglés de una técnica usada por los DJ, que consiste en girar los discos sobre las bandejas de tocadiscos para remezclar música.

Pilas Bloques

PILAS BLOQUES - PISTA 1
Es un lenguaje de programación diseñado para ayudar a los principiantes a entender los conceptos básicos de programación sin tener

PILAS BLOQUES - PISTA 2

Está basado en Blockly, un lenguaje creado anteriormente por Google.

PILAS BLOQUES - PISTA 3

También se basó en un desarrollo previo de un programador llamado Hugo Ruscitti.

PILAS BLOQUES - PISTA 4

Fue creado con el propósito de tener una aplicación pensada para el contexto de nuestro país, para enseñar programación a los y las estudiantes de Argentina.

PILAS BLOQUES - PISTA 5

Creado por Fundación Sadosky en colaboración con profesionales de diferentes universidades argentinas (UBA, UNC, UNQ y UTN).