

Programación





UNIDAD II

A continuación se esbozan algunas orientaciones a tener en cuenta en el proceso de enseñanza de los estudiantes. Para ello, plantearemos algunas actividades generales que consideramos desde el equipo de Wonderly como pertinentes y potenciadoras para el abordaje de los contenidos. Se considera que las clases están planificadas para una duración estimada de 45 minutos reloj y el recurso primordial para el desarrollo de las mismas es el acceso a internet mediante una computadora/tablet. **Sin embargo, la invitación es que puedan adecuar la propuesta de forma singular y situada a la institución educativa y al grupo-clase en particular.**

En el desarrollo de las orientaciones se incluye y promueve el enfoque STEAM articulando al mismo tiempo el trabajo dentro de las aulas sobre las diversas competencias, tales como: pensamiento crítico, resolución de problemas, creatividad, cooperación y participación, pensamiento lógico-matemático, aprendizaje ensayo-error, participación activa y pensamiento computacional.

Consejo:

Para un mejor desenvolvimiento de cada encuentro, se sugiere que el educador revise detalladamente cada orientación como parte de la preparación previa para la clase

ORIENTACIONES GENERALES

CLASE 6

Objetivos de la clase	<ul style="list-style-type: none">• Introducir al lenguaje de programación Python• Aprender a crear y ejecutar el primer programa en Python
Temas	<ul style="list-style-type: none">• Programación• Almacena datos en variables• Python• Pseudocódigo para variables
Recursos	<p>Computadoras/tablets con acceso a internet Contenidos para trabajar en esta clase:</p> <ul style="list-style-type: none">• Video: Crea el primer programa• Misión: “Hola Mundo”• Desafío 2.1 Crea el primer programa• Misión: Reemplaza pseudocódigo por comandos en Python• Desafío 2.2 Crea el primer programa• Almacena datos en variables• Misión: Almacena datos en variables• Desafío 2.3 Almacena datos en variables• Pseudocódigo para variables• Misión: Llevar a Python un pseudocódigo• Desafío 2.4 Pseudocódigo para variables
Modalidad	<p>La dinámica será:</p> <ul style="list-style-type: none">• Trabajo con el grupo total• Trabajo en grupo
La propuesta	<p>La clase se dividirá en tres grandes momentos:</p> <ul style="list-style-type: none">• <u>1º momento</u>: Introducción• <u>2º momento</u>: Desarrollo de los contenidos• <u>3º momento</u>: Cierre

Clase 6

Explorando Python: creación de programas y manipulación de variables

Primer Momento: Introducción

Para estos minutos iniciales, puedes hacer una breve mención sobre la Unidad II que se comienza en el curso de Principios de Computación llamada: Programación. Para ello presenta la siguiente pregunta y abre al debate: *¿Qué es la programación y cómo piensan en que se puede crear un primer programa?*

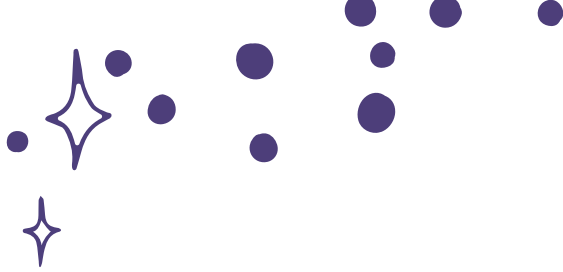
Dedica unos minutos para conversar y explicar qué es la programación: el proceso de escribir instrucciones para que una computadora las ejecute. Discute la importancia de la programación en el mundo moderno, desde el software hasta los dispositivos electrónicos. Ejemplificar con ejemplos cotidianos el papel de la programación en la vida diaria. Invita a los estudiantes a reflexionar en conjunto sobre estas temáticas.

¿Sabías que? La programación es el proceso de escribir un conjunto de instrucciones que una computadora puede entender y ejecutar para realizar una tarea específica. Estas instrucciones están escritas en un lenguaje de programación y pueden variar desde simples operaciones aritméticas hasta complejos algoritmos.

Los sistemas de navegación GPS en los automóviles utilizan algoritmos de programación para calcular la ruta más rápida de un punto a otro.

- Las redes sociales como Facebook o Instagram utilizan algoritmos para recomendar contenido a los usuarios basado en sus intereses y comportamientos anteriores.

- Los electrodomésticos inteligentes, como los termostatos programables o las luces controladas por voz, funcionan gracias al software que los programa para responder a ciertas condiciones o comandos.



Segundo Momento: Desarrollo de los contenidos

Visualización del video: Video: *Crea el primer programa*. Aquí se introduce el entorno de programación de Python. Se demuestra cómo escribir un programa simple en Python utilizando la función `print()` para mostrar "Hola, Mundo" en la pantalla, en la primera misión.

Se explica la estructura del programa, incluyendo la función `print()`, el uso de comillas dobles para delimitar cadenas de texto y la importancia del punto y coma al final de la instrucción. En Wonderly se va a trabajar con la pantalla dividida en donde se explica el paso a paso, permitiendo que los estudiantes practiquen después. Dedicar unos minutos a dialogar junto a los estudiantes los siguientes temas:

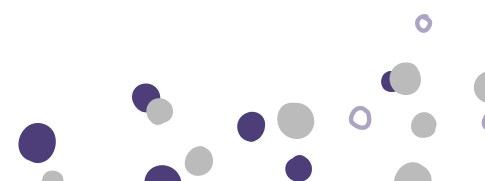
Python es un lenguaje de programación de alto nivel, interpretado y fácil de aprender. Es conocido por su sintaxis simple y legible, lo que lo hace ideal para principiantes y también para profesionales en el desarrollo de software.

Características de Python: Sintaxis simple y legible: Python utiliza una sintaxis limpia y fácil de entender, lo que facilita la lectura y escritura de código.

Amplia variedad de bibliotecas: Python cuenta con una amplia colección de bibliotecas que cubren una amplia gama de áreas, desde desarrollo web y científico hasta inteligencia artificial y análisis de datos.

Versatilidad en diferentes áreas de desarrollo: Python se utiliza en una variedad de campos, incluyendo desarrollo web, análisis de datos, automatización, inteligencia artificial y más.

Comparación con otros lenguajes de programación: Python se destaca por su facilidad de uso y su enfoque en la legibilidad del código. En comparación con otros lenguajes de programación como C++ o Java, Python tiende a ser más conciso y requiere menos líneas de código para lograr la misma tarea. Además, Python es conocido por su comunidad activa y su amplia documentación, lo que lo hace ideal para proyectos de código abierto y colaborativos.



Entendiendo Pseudocódigo. El pseudocódigo es una forma de describir algoritmos y programas utilizando una sintaxis simple que no se ejecuta en un editor de código. Añade que es una herramienta útil para planificar y entender programas antes de escribirlos en un lenguaje de programación real.

Puedes utilizar un ejemplo sencillo para demostrar cómo se vería el pseudocódigo para mostrar un mensaje en pantalla utilizando la instrucción DISPLAY (expresión).

Demostración: puedes reutilizar el ejemplo de la clase anterior, donde se muestra el pseudocódigo para mostrar mensajes como "Hola" y "Chicos".

Comparación con `println()` en Python. Aquí se explica el concepto de `println()` en un lenguaje de programación como Python, resaltando las similitudes y diferencias con la instrucción DISPLAY en pseudocódigo.

Actividad: Dividir a los estudiantes en parejas o grupos pequeños y pídeles que trabajen juntos para traducir otras instrucciones de pseudocódigo a `println()` en Python.

Lección: Almacena datos en variables. Aquí se explica que las variables son como contenedores donde podemos almacenar datos. Destaca su importancia en el desarrollo de programas. Utiliza ejemplos de la vida cotidiana para ilustrar el concepto de variables, como ajustar el volumen de la música o la dificultad en un videojuego.



Características de las Variables:

- Nombres, Valores y Tipos de Datos: Explicar que las variables tienen un nombre único que las identifica, un valor que puede cambiar durante la ejecución del programa y un tipo de datos que determina qué tipo de información pueden almacenar.
- Proporcionar ejemplos concretos de variables en un contexto de videojuegos, como el número de vidas, la puntuación o el nivel de dificultad.

Introducción a los Tipos de Datos. Aquí se presentan los tipos de datos básicos en Python, incluyendo enteros, flotantes, cadenas de texto y booleanos. Explicar cómo se representan estos tipos de datos en variables. Breve introducción a otros tipos de datos más complejos en Python, como listas y diccionarios, indicando que se explorarán en clases futuras.

Asignación de Valores a Variables:

Sintaxis de Asignación: Se explica la sintaxis de asignación en Python utilizando la palabra clave `var`, seguida del nombre de la variable, el signo igual y el valor que se asigna a la variable.

A su vez se utilizan ejemplos de código para demostrar cómo asignar valores a variables en Python y cómo imprimir los valores en pantalla utilizando `println()`.

Reasignación de Valores

Cambio de Valores: Se explica que los valores de las variables pueden cambiar durante la ejecución de un programa y muestra cómo asignar valores a variables existentes utilizando la misma sintaxis de asignación.

¡Actividades extras!

Proporcionar a los estudiantes una serie de ejercicios prácticos donde deben crear variables, asignarles valores e imprimir los valores en pantalla.



Ejercicio 1: Variables y Tipos de Datos

- Crea una variable llamada nombre y asígnale tu nombre.
- Crea una variable llamada edad y asígnale tu edad.
- Crea una variable llamada altura y asígnale tu altura en metros.
- Crea una variable llamada es_estudiante y asígnale True si eres estudiante o False si no lo eres.

Imprime en pantalla el valor de cada una de estas variables utilizando println().

Ejercicio 2: Operaciones con Variables

- Crea una variable llamada numero1 y asígnale un número entero.
- Crea una variable llamada numero2 y asígnale otro número entero.
- Calcula la suma de numero1 y numero2 y asígnala a una nueva variable llamada suma.
- Calcula la resta de numero1 menos numero2 y asígnala a una nueva variable llamada resta.

Imprime en pantalla el valor de suma y resta.



Tercer Momento: Cierre

En esta clase se empezó a explorar el fascinante mundo de la programación y descubriendo cómo las computadoras ejecutan instrucciones para realizar diversas tareas. Se aprendió que la programación es esencial en el mundo moderno, desde los sistemas de navegación en automóviles hasta las redes sociales y los electrodomésticos inteligentes. A su vez se conoció el entorno de programación de Python y se empieza a dar los primeros pasos en la creación del primer programa, utilizando la función `print()` para mostrar un mensaje en la pantalla.

¡El mundo de la programación es infinito y recién se comienza! Se aprendió a cómo almacenar datos y manipularlos dentro de los programas. Se exploró los diferentes tipos de datos y cómo asignar y reasignar valores a variables.

¡No olvides de completar las misiones y desafíos juntos a los estudiantes, resolver dudas!

¡Anexo de otra idea para trabajar en el aula!

CONSEJO

¡Educadores! Esta actividad tiene vinculación con los temas vistos en este encuentro. A modo de sugerencia, puede ser integrada como otra actividad a trabajar o dedicar su desarrollo en profundidad en otra clase.

Actividad: Calculadora Simple

- Explicar a los estudiantes que van a crear una calculadora simple utilizando Python que pueda realizar operaciones de suma, resta, multiplicación y división.
- Dividir a los estudiantes en parejas o grupos pequeños
- Guiar a los estudiantes a través de los siguientes pasos:
 - a) Solicitar al usuario que ingrese dos números y la operación que desea realizar (suma, resta, multiplicación o división).
 - b) Realizar la operación seleccionada utilizando los números proporcionados.
 - c) Imprimir el resultado de la operación en la pantalla.
- Ayuda a los estudiantes a escribir el código paso a paso, explicando cada parte a medida que avanzan.
- A medida que los estudiantes avanzan en la actividad, anímalos a probar diferentes combinaciones de números y operaciones para ver cómo funciona su calculadora.
- Después de que todos los grupos hayan completado la actividad, invítalos a compartir sus calculadoras y a discutir cualquier problema o desafío que hayan enfrentado durante el proceso de programación.



ORIENTACIONES GENERALES

CLASE 7

Objetivos de la clase	<ul style="list-style-type: none">• Comprender las expresiones matemáticas• Fomentar la resolución de problemas• Aplicar los conceptos en ejercicios prácticos
Temas	<ul style="list-style-type: none">• Expresiones matemáticas• Procedimientos y constantes matemáticas• Almacenar cadenas en variables
Recursos	<p>Computadoras/tablets con acceso a internet Contenidos para trabajar en esta clase:</p> <ul style="list-style-type: none">• Video: Expresiones matemáticas• Misión 5: Cálculo de calorías• Desafío 2.5 Expresiones matemáticas• Video: Procedimientos y constantes matemáticas• Misión: Calculemos el área de la pizza• Desafío 2.6 Procedimientos y constantes matemáticas• Video: Almacenar cadenas en variables• Misión: Almacenar cadenas en variables
Modalidad	<p>La dinámica será:</p> <ul style="list-style-type: none">• Trabajo con el grupo total• Trabajo en grupo
La propuesta	<p>La clase se dividirá en tres grandes momentos:</p> <ul style="list-style-type: none">• <u>1º momento</u>: Introducción• <u>2º momento</u>: Desarrollo de los contenidos• <u>3º momento</u>: Cierre

Clase 7

Explorando expresiones matemáticas y operadores en programación

Primer Momento: Introducción

En esta clase, se hará énfasis en descubrir los secretos numéricos que se esconden en cada ecuación y operación. El objetivo es introducir a los estudiantes al uso de operadores matemáticos en programación y cómo aplicarlos para realizar cálculos. Puedes proponer una breve discusión sobre la importancia de los ordenadores y cómo han facilitado los cálculos matemáticos y preguntar a los estudiantes si están familiarizados con los operadores matemáticos y cómo se utilizan en programación. ¿Quién está listo para sumergirse en el misterio de los números?

Actividad introductoria: El Laberinto de los Números

Materiales Necesarios: Cartulinas o papel grande para dibujar el laberinto. Marcadores o rotuladores para escribir los números y operaciones. Espacio amplio en el salón de clase para colocar el laberinto.

- Dibujar un laberinto en el suelo del salón utilizando cartulinas o papel grande. El laberinto debe tener caminos, bifurcaciones y áreas de descanso. En cada sección del laberinto, escribe números y operaciones matemáticas como suma, resta, multiplicación y división.
- Dividir a los estudiantes en grupos pequeños. Cada grupo selecciona un representante para ser el "explorador" del laberinto.
- Explicar a los estudiantes que están a punto de embarcarse en una aventura matemática en busca del tesoro numérico escondido en el laberinto. Cada grupo comienza en el punto de inicio del laberinto



- Los exploradores avanzan por el laberinto resolviendo problemas matemáticos en cada sección para determinar el camino correcto. Cuando se encuentren con un número u operación, deben resolver el problema matemático correspondiente para determinar qué dirección tomar. Si resuelven el problema correctamente, avanzan hacia adelante. Si no, deben retroceder y probar otro camino
- A lo largo del laberinto, añadir obstáculos matemáticos que los estudiantes deben superar. Por ejemplo, una sección podría requerir la resolución de un problema más difícil para avanzar. También puedes agregar secciones de "descanso" donde los estudiantes puedan tomar un respiro y repasar conceptos matemáticos básicos.
- El objetivo final es llegar al final del laberinto y descubrir el tesoro numérico escondido.
- Después de completar el laberinto, los estudiantes pueden reflexionar sobre los problemas que encontraron más difíciles y cómo los resolvieron.

Segundo Momento: Desarrollo de los contenidos

Visualización del vídeo: Expresiones matemáticas. Misión 5: Cálculo de calorías. Desafío 2.5 Expresiones matemáticas. Aquí se muestra a los estudiantes ejemplos de operadores matemáticos en Python, como la suma, resta, multiplicación, división y módulo. Se explica cómo se utilizan estos operadores y proporcionar ejemplos de su uso en código.

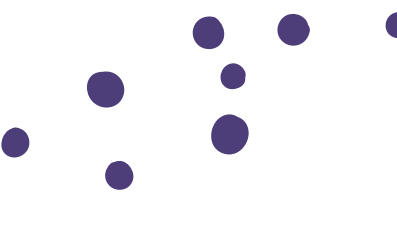
Aquí algunos: ¡El Desafío de las Mascotas!

- Preparar una serie de problemas relacionados con el cuidado de las mascotas, como calcular la cantidad de agua que necesita un gato en un día, determinar el tiempo total de juego para un perro en una semana, etc. Asegúrate de que los problemas sean variados en dificultad y que requieran el uso de expresiones matemáticas y variables para resolverlos.
- Dividir a los estudiantes en parejas o grupos pequeños, según la preferencia y el tamaño de la clase.



- Proporcionar a cada grupo uno de los problemas de cuidado de mascotas que has preparado. Explicar claramente el problema y asegúrate de que los estudiantes entiendan lo que se les pide hacer.
- Los grupos trabajan juntos para desarrollar un algoritmo que resuelva el problema dado.
- Animar a los estudiantes a discutir y planificar cómo abordarán el problema, utilizando expresiones matemáticas y variables según sea necesario.
- Una vez que hayan desarrollado un plan, los grupos escriben el código en el lenguaje de programación que estén utilizando (puede ser pseudocódigo si aún no han comenzado a programar). Después de escribir el código, deben probarlo con diferentes valores de entrada para asegurarse de que funcione correctamente en diferentes situaciones.
- Al final del tiempo asignado, cada grupo presenta su solución al problema a la clase.

Visualización del video: Procedimientos y constantes matemáticas. Misión: Calculemos el área de la pizza. Desafío 2.6 Procedimientos y constantes matemáticas. Aquí los estudiantes aprenderán a calcular el área de una pizza utilizando procedimientos y constantes matemáticas en un entorno de programación. El objetivo principal es practicar el uso de procedimientos para modularizar el código y el uso de constantes matemáticas para realizar cálculos precisos.



Constantes Matemáticas: se introduce el concepto de constantes matemáticas, como pi (π), que es una constante utilizada para calcular el área de un círculo. Puedes explicar que pi es una constante predefinida en la mayoría de los lenguajes de programación y su valor es aproximadamente 3.14159.

Recordatorio: preguntar sobre cualquier desafío que hayan enfrentado durante la actividad y cómo lo resolvieron.

Visualización del vídeo: Almacenar cadenas en variables. Misión: Almacenar cadenas en variables

Desafío 2.7 Almacenar cadenas en variables. Aquí se explorarán las variables de Cadena en Programación. Puedes dedicar unos minutos a una breve discusión sobre la importancia de almacenar texto en programación y cómo se pueden utilizar variables de cadena en diferentes contextos.


Presentación de Variables de Cadena: Se muestran ejemplos de variables de cadena en Python y explica cómo se declaran y utilizan. A su vez se destaca la importancia de usar comillas dobles para delimitar las cadenas de texto y cómo pueden contener letras, números y símbolos.

Actividad Práctica:

Dividir a los estudiantes en parejas o grupos pequeños.

Asignar a cada grupo un escenario o problema donde tengan que utilizar variables de cadena para almacenar información relevante.

Los estudiantes deben trabajar juntos para escribir el código que resuelve el problema y mostrar los resultados.





Tercer Momento: Cierre

Para cerrar este encuentro, invita a los estudiantes a reflexionar sobre los conceptos que han aprendido durante la clase. Puedes hacer preguntas como: "¿Qué concepto o técnica te resultó más interesante o útil?" o "¿Cómo crees que estos conocimientos podrían aplicarse en situaciones de la vida real?" Esto les permite consolidar lo aprendido y comprender su relevancia más allá del aula.

No olvides de animar a los estudiantes a compartir cualquier pregunta adicional que tengan o cualquier idea que deseen explorar más a fondo: ¿Hay algo que no entendieron completamente y les gustaría discutir más?"

No olvides de fomentar a los estudiantes a seguir explorando y aprendiendo sobre matemáticas y programación fuera del aula. Puedes sugerir recursos adicionales, como libros, tutoriales en línea o clubes de programación, para que continúen su desarrollo.

¡Anexo de otra idea para trabajar en el aula!

CONSEJO

¡Educadores! Esta actividad tiene vinculación con los temas vistos en este encuentro. A modo de sugerencia, puede ser integrada como otra actividad a trabajar o dedicar su desarrollo en profundidad en otra clase.

Actividad: Desafío matemático en tiempo real

Pasos a seguir:

- Preparar una serie de preguntas matemáticas variadas, desde problemas simples hasta problemas más desafiantes que requieren el uso de múltiples operadores y expresiones.
- Crear tarjetas o hojas de papel con las preguntas impresas claramente para cada grupo.
- Dividir a los estudiantes en equipos o grupos pequeños. Entrega a cada grupo una serie de tarjetas con las preguntas matemáticas. Establece un límite de tiempo, por ejemplo, 15-20 minutos, para que los grupos resuelvan tantas preguntas como sea posible
- Mientras los grupos trabajan, circula por el aula para ofrecer ayuda y orientación según sea necesario.
- Invitar a cada grupo a presentar sus respuestas y explicar cómo llegaron a ellas.



ORIENTACIONES GENERALES

CLASE 8

Objetivos de la clase	<ul style="list-style-type: none">• Comprender las operaciones básicas con cadenas• Explorar los métodos de cadena en Python• Ejercitar y practicar
Temas	<ul style="list-style-type: none">• Operaciones con cadenas – Parte I y Parte II• Condicionales con if, else y booleanos
Recursos	<p>Computadoras/tablets con acceso a internet Contenidos para trabajar en esta clase:</p> <ul style="list-style-type: none">• Video: Operaciones con cadenas – Parte I y Parte II• Misión: Realiza concatenaciones y subcadenas y utiliza la función “upper”• Desafío 2.8 Operaciones con cadenas – Parte I y II• Video: Condicionales con if, else y booleanos• Desafío 2.9 Condicionales con if, else y booleanos
Modalidad	<p>La dinámica será:</p> <ul style="list-style-type: none">• Trabajo con el grupo total• Trabajo en grupo
La propuesta	<p>La clase se dividirá en tres grandes momentos:</p> <ul style="list-style-type: none">• <u>1º momento</u>: Introducción• <u>2º momento</u>: Desarrollo de los contenidos• <u>3º momento</u>: Debate y reflexión

Clase 8

Explorando operaciones con cadenas en programación



Primer Momento: Introducción

Para estos minutos iniciales, puedes introducir la temática a trabajar que será la manipulación de texto en nuestros programas. Las cadenas, o strings, son una parte fundamental de la mayoría de los programas, ya que permiten trabajar con información textual de diversas maneras.

Dirígete a los estudiantes:

“Imaginen que están trabajando en un programa que necesita procesar nombres, direcciones de correo electrónico, mensajes de texto o incluso documentos completos. En todos estos casos, necesitarán habilidades para manipular y trabajar con cadenas de texto de manera efectiva. En esta clase se explorará cómo podemos realizar diversas operaciones con cadenas, como la concatenación, la búsqueda de subcadenas, la extracción de partes específicas de una cadena y la modificación de su contenido. También veremos cómo podemos trabajar con la codificación y decodificación de cadenas, especialmente en el contexto de Unicode, que nos permite manejar caracteres de diferentes idiomas y símbolos especiales. A lo largo de la clase, practicaremos con ejemplos prácticos y desafíos que les ayudarán a comprender mejor cómo aplicar estas operaciones en situaciones reales de programación”

Segundo Momento: Desarrollo de los contenidos

Visualización de los vídeos: *Operaciones con cadenas – Parte I y Parte II. Misión: Realizar concatenaciones y subcadenas y utilizar la función “upper”*. Aquí se presentan las Operaciones con Cadenas en Python, como concatenación, extracción de subcadenas, conversión a minúsculas/mayúsculas, etc. Se explica cómo se realizan estas operaciones y cuándo pueden ser útiles en el desarrollo de aplicaciones.

Actividad práctica:

- **Formateo de Texto:** Solicitar a los estudiantes que escriban una función que tome un texto como entrada y devuelva una versión en la que todas las palabras comienzan con mayúscula. Solicitar a los estudiantes que escriban una función que tome un número de teléfono como entrada en un formato específico y devuelva una versión del número en un formato estándar, por ejemplo, eliminando los espacios y agregando el código de país si falta.
- **Búsqueda de Palabras Claves:** proponer a los estudiantes que escriban una función que tome un texto y una palabra como entrada, y devuelva la cantidad de veces que esa palabra aparece en el texto. Pídeles que escriban una función que tome una lista de correos electrónicos y devuelva una lista con solo aquellos correos que contengan un dominio específico, por ejemplo, "gmail.com".



- Validación de Entradas de Usuario: desafía a los estudiantes a escribir una función que tome un correo electrónico como entrada y verifique si está en un formato válido. Solicitar a los estudiantes que escriban una función que tome una contraseña como entrada y verifique si cumple con ciertos criterios de seguridad, como tener al menos 8 caracteres, al menos una letra mayúscula y al menos un número.
- Generación de Salidas Formateadas: Desafía a los estudiantes a escribir una función que tome una lista de nombres y edades y devuelva una cadena formateada que liste cada nombre con su respectiva edad. Pídeles que escriban una función que tome un número y devuelva una cadena que indique si es par o impar.

Visualización del video: *Condicionales con if, else y booleanos. Desafío 2.9 Condicionales con if, else y booleanos.* En el mundo de la programación, muchas veces nos encontramos en situaciones en las que necesitamos que nuestros programas tomen decisiones en función de ciertas condiciones. Esto es donde entran en juego los condicionales, una herramienta poderosa que permite controlar el flujo de nuestro código y hacer que nuestras aplicaciones sean más inteligentes y adaptables. Uso de condicionales con `if`, `else` y booleanos en programación. Estas son herramientas fundamentales que nos permiten escribir código que puede evaluar diferentes situaciones y actuar en consecuencia.



¿Qué son los condicionales? son estructuras de control que nos permiten tomar decisiones en nuestro código. Nos permiten ejecutar cierto bloque de código si se cumple una condición específica, y otro bloque de código si esa condición no se cumple. El condicional más básico en programación es el `if`. Con el `if`, podemos verificar si una condición es verdadera y ejecutar un bloque de código en consecuencia.

El uso de booleanos: son tipos de datos que pueden tener solo dos valores: `True` o `False`. Estos son fundamentales para el funcionamiento de los condicionales, ya que nos permiten expresar condiciones que pueden ser evaluadas como verdaderas o falsas.

¡A practicar!

Determinar si un número es positivo, negativo o cero: solicitar a los estudiantes que escriban un programa que tome un número como entrada y determine si es positivo, negativo o cero. Deben imprimir un mensaje indicando el resultado.

Verificar si un año es bisiesto: desafía a los estudiantes a escribir un programa que tome un año como entrada y determine si es bisiesto o no. Un año es bisiesto si es divisible por 4, pero no por 100, excepto si también es divisible por 400.

Calcular el precio final de un producto con descuento: proponer a los estudiantes que escriban un programa que tome el precio original de un producto y el porcentaje de descuento aplicado. Deben calcular el precio final después del descuento e imprimirlo.



Clasificar un triángulo según sus lados: Pídeles que escriban un programa que tome la longitud de los tres lados de un triángulo como entrada y determine si es equilátero (todos los lados iguales), isósceles (dos lados iguales) o escaleno (ningún lado igual).

Verificar si una contraseña cumple con ciertos criterios: desafía a los estudiantes a escribir un programa que tome una contraseña como entrada y verifique si cumple con ciertos criterios, como tener al menos 8 caracteres, al menos una letra mayúscula y al menos un número.

Tercer Momento: Cierre

Antes de finalizar hoy se ha trabajado sobre cómo utilizar if, else y booleanos para tomar decisiones en base a diferentes situaciones, aplicando este conocimiento resolviendo desafiantes ejercicios prácticos. Dedicar unos minutos a repasar y resolver dudas de los estudiantes. ¡Estimula siempre a seguir aprendiendo!



ORIENTACIONES GENERALES

CLASE 9

Objetivos de la clase	<ul style="list-style-type: none">• Comprender los fundamentos de la lógica de programación.• Aplicar estructuras condicionales y operadores lógicos en programas de software.• Desarrollar habilidades para resolver problemas utilizando la lógica y la programación
Temas	<ul style="list-style-type: none">• Condicionales anidadas• Booleanos compuestos con operadores lógicos• Definir un procedimiento
Recursos	<p>Computadoras/tablets con acceso a internet Contenidos para trabajar en esta clase:</p> <ul style="list-style-type: none">• Video: Condicionales anidadas• Misión: Condicionales anidadas• Desafío 2.10 Condicionales anidadas• Video: Booleanos compuestos con operadores lógicos• Misión: Booleanos compuestos con operadores lógicos – Parte 1• Misión: Booleanos compuestos con operadores lógicos – Parte 2• Desafío 2.11 Booleanos compuestos con operadores lógicos• Vídeo: Definir un procedimiento• Misión: Creando procedimiento• Desafío 2.12 Definir un procedimiento
Modalidad	<p>La dinámica será:</p> <ul style="list-style-type: none">• Trabajo con el grupo total
La propuesta	<p>La clase se dividirá en tres grandes momentos:</p> <ul style="list-style-type: none">• <u>1º momento</u>: Introducción• <u>2º momento</u>: Desarrollo de temas• <u>3º momento</u>: Cierre

Clase 9

Desafío de la lógica: navegando por las decisiones y operaciones lógicas en el mundo de la programación

Primer Momento: Introducción

En esta clase, se seguirá avanzando con el mundo de la computación y programación, en donde se explorará los temas de: condicionales, operadores lógicos y la definición de procedimientos.

Puedes proponer una actividad interactiva para despertar el interés de los estudiantes. Por ejemplo, podrías mostrarles un video corto sobre cómo los programas informáticos pueden tomar decisiones basadas en condiciones. Este video puede presentar situaciones cotidianas donde las decisiones son necesarias y cómo estas decisiones se pueden traducir en código de programación. Por ejemplo: mostrar cómo un programa de control de temperatura decide si encender o apagar un sistema de calefacción según la temperatura ambiente.



Segundo Momento: Desarrollo de los contenidos

Visualizaciones de los videos: *Condicionales anidadas, booleanos compuestos con operadores lógicos y por último definir un procedimiento*. Realización de las misiones y desafíos respectivos.

Aquí puedes presentar los conceptos de condicionales, operadores lógicos y definición de procedimientos de una manera interactiva y visual. Utiliza ejemplos simples y divertidos para explicar cada concepto, como decidir qué ropa usar según el clima o planificar un juego con diferentes niveles.

Demostración: realizar una demostración en vivo de cómo utilizar condicionales y operadores lógicos para resolver un problema práctico. Por ejemplo, puedes simular un juego de adivinanzas donde el programa debe determinar si la respuesta dada por el usuario es correcta o incorrecta utilizando condicionales y operadores lógicos.

En el caso de un procedimiento para simplificar el código y promover la reutilización de código. Por ejemplo, puedes definir un procedimiento para calcular el promedio de una lista de números y luego utilizarlo en diferentes partes del programa.

Práctica Guiada: proporcionar a los estudiantes ejercicios prácticos para que practiquen la creación de condicionales, el uso de operadores lógicos y la definición de procedimientos.

Por ejemplo:

- Crear un programa que determine si un número dado es par o impar.

- Escribir un programa que clasifique a los estudiantes según sus calificaciones en diferentes categorías (excelente, bueno, suficiente, insuficiente) utilizando condicionales anidadas.
- Definir un procedimiento para calcular el área de diferentes formas geométricas (círculo, rectángulo, triángulo) y luego utilizarlo para resolver problemas de geometría

Actividades adicionales para desarrollar los temas de condicionales, operadores lógicos y definición de procedimientos:

CONSEJO

¡Educadores! Estas actividades tienen vinculación con los temas vistos en este encuentro. A modo de sugerencia, pueden ser integradas como otras actividades a trabajar o dedicar su desarrollo en profundidad en otras clases.

Juego de Roles:

- Organizar un juego de roles donde los estudiantes actúen como programadores y simulan diferentes situaciones donde necesitan utilizar condicionales y operadores lógicos para tomar decisiones. Por ejemplo, puedes asignar roles como el programador, el cliente y el usuario final, y crear escenarios donde el programador debe escribir código para satisfacer las necesidades del cliente y del usuario final.

Proyectos de Simulación:

- Dividir a los estudiantes en grupos y proporcione un conjunto de problemas de simulación del mundo real que requieran el uso de condicionales, operadores lógicos y procedimientos para resolverlos. Por ejemplo, pueden simular un sistema de control de tráfico, un sistema de gestión de inventario o un juego de estrategia.

Creación de Mini Juegos:

- Animar a los estudiantes a crear sus propios mini juegos utilizando condicionales, operadores lógicos y procedimientos. Por ejemplo, pueden crear un juego de adivinanzas, un juego de laberinto o un juego de preguntas y respuestas. Esto les permitirá aplicar los conceptos aprendidos de una manera divertida y creativa.

Desafíos de Resolución de Problemas:

- Presentar desafíos de resolución de problemas que requieran el uso de condicionales, operadores lógicos y procedimientos para resolverlos. Por ejemplo, puedes proporcionar problemas como determinar si un año es bisiesto, calcular el costo total de una compra en función de descuentos aplicados o determinar la ruta más corta para llegar a un destino en un mapa.

Competencias de Programación:

- Organizar competencias de programación donde los estudiantes compitan para resolver problemas utilizando condicionales, operadores lógicos y procedimientos en un tiempo determinado. Puedes proporcionar una serie de problemas de dificultad creciente y premiar a los estudiantes que logren resolver la mayor cantidad de problemas en el menor tiempo posible



Tercer Momento: Cierre

Para finalizar este encuentro, puedes invitar a los estudiantes a reflexionar sobre lo que han aprendido durante la clase. Pregunta si hay algo que encontraron especialmente interesante o desafiante. Anímalos a pensar en situaciones cotidianas donde podrían aplicar los conceptos aprendidos hoy. Pregúntales cómo creen que estos conocimientos podrían ser útiles en su vida diaria o en futuros proyectos de programación



ORIENTACIONES GENERALES

CLASE 10

Objetivos de la clase	<ul style="list-style-type: none">• Recapitular y reflexionar sobre los contenidos trabajados• Explorar tendencias actuales y futuras• Fomentar la participación activa de los estudiantes
Temas	<ul style="list-style-type: none">• Integración de contenidos• Actividades lúdicas• Reflexiones finales
Recursos	<p>Computadoras/tablets con acceso a internet</p> <p>Contenidos para trabajar en esta clase:</p> <ul style="list-style-type: none">• Videos trabajados durante el curso
Modalidad	<p>La dinámica será:</p> <ul style="list-style-type: none">• Trabajo con el grupo total
La propuesta	<p>La clase se dividirá en tres grandes momentos:</p> <ul style="list-style-type: none">• <u>1º momento</u>: Introducción• <u>2º momento</u>: Actividades finales• <u>3º momento</u>: Reflexion y cierre

Clase 10

Luces, cámaras, computación: Una visión panorámica del mundo de la computación

Primer Momento: Introducción

Para este encuentro final donde hoy se pondrán a prueba los conocimientos del fascinante mundo de la computación. Durante este curso, se han explorado los fundamentos esenciales de la informática, desde conceptos básicos hasta temas más avanzados. Ahora, es el momento de demostrar todo lo que han aprendido a lo largo de este viaje, que vale aclarar que estas clases, son solo el comienzo, aún queda camino por recorrer. Por ello, se proponen dos actividades para realizar una revisión e integración de los contenidos trabajados en estas clases.



Segundo Momento: Actividades finales

Opción 1: Batalla de Bits: ¡Demostrando los conocimientos en Computación!

Preparación:


- Diseñar una lista de preguntas relacionadas con los temas clave abordados en el curso. Estas preguntas pueden ser de diferentes tipos, como opción múltiple, verdadero o falso, preguntas de respuesta corta, completar el espacio en blanco, etc. Asegúrate de que las preguntas cubran la amplia gama de temas y dificultades para desafiar a los estudiantes y evaluar su comprensión del material.
- Organizar las preguntas en categorías si es posible, para darle estructura al juego y facilitar su seguimiento.

Organización de equipos:

- Dividir a los estudiantes en equipos equitativos. Puedes hacerlo de forma aleatoria asignando números o colores a cada estudiante y luego formando equipos con una combinación equilibrada de estos números o colores. Si prefieres, también puedes permitir que los estudiantes elijan sus propios equipos.

Dinámica del juego:

- Establecer un formato para el juego. Puedes optar por un concurso tipo "quiz show" donde los equipos compiten para responder preguntas y acumular puntos

- 
- Cada equipo tendrá la oportunidad de seleccionar una pregunta de una categoría específica y recibir puntos según la dificultad de la pregunta
 - Definir las reglas del juego, como el tiempo asignado para responder cada pregunta, el método para determinar qué equipo responde primero, cómo se asignan los puntos, etc.
 - Puedes agregar elementos adicionales para hacer el juego más dinámico, como rondas de bonificación, preguntas sorpresa, o comodines que los equipos puedan utilizar para obtener ventaja.

Tiempo y lugar:

- Programar el juego en una sesión específica al final del curso, asegurándote de que haya suficiente tiempo para completar todas las rondas planificadas.
- Elegir un lugar adecuado para llevar a cabo el juego, preferiblemente una sala de clases o un auditorio con suficiente espacio para todos los equipos y con las facilidades necesarias, como pizarra o proyector para mostrar las preguntas.

Opción 2: Panel de discusión: Conectando Bits: explorando el futuro de la computación

Invitación a los expertos:

- Contactar a expertos en el campo de la computación o profesionales que estén dispuestos a participar en el panel de discusión. Puedes buscar profesionales locales, ex alumnos destacados, o incluso expertos que estén dispuestos a participar de forma remota a través de plataformas de videoconferencia.

Temas de discusión:

- Coordinar con los expertos para establecer los temas que se abordarán durante el panel. Estos pueden incluir tendencias actuales en tecnología, desafíos y oportunidades en la industria, ética en la informática, impacto de la inteligencia artificial, etc.
- Proporcionar a los expertos una guía clara sobre los temas que se espera que aborden, pero también permitirles la libertad de añadir su experiencia y conocimientos adicionales.

Moderación:

- Designar a un moderador que se encargue de dirigir la discusión y mantenerla en marcha. El moderador debe ser alguien con experiencia en facilitar conversaciones y mantener el orden.
- El moderador debe preparar preguntas tanto para los expertos como para los estudiantes para fomentar la interacción y mantener la conversación en movimiento.
- Asegúrate de que el moderador esté familiarizado con los temas a tratar y capaz de mantener un tono neutral y respetuoso durante toda la discusión.

Interacción con los estudiantes:

- Abrir la discusión para que los estudiantes puedan hacer preguntas y participar activamente en la conversación. Esto les brindará la oportunidad de obtener información de primera mano y de interactuar con profesionales del campo.



- Establecer un ambiente de respeto y apertura donde los estudiantes se sientan cómodos compartiendo sus preguntas e inquietudes.
- Animar a los estudiantes a participar activamente haciendo preguntas pertinentes y contribuyendo con sus propias experiencias y perspectivas.

Al seguir estos pasos y adaptar el panel de discusión a las necesidades específicas de tus estudiantes y del curso, podrás organizar una clase final enriquecedora y estimulante que proporciona una valiosa oportunidad para el intercambio de ideas y la exploración de temas relevantes en el campo de la computación.

Tercer Momento: Reflexión y cierre

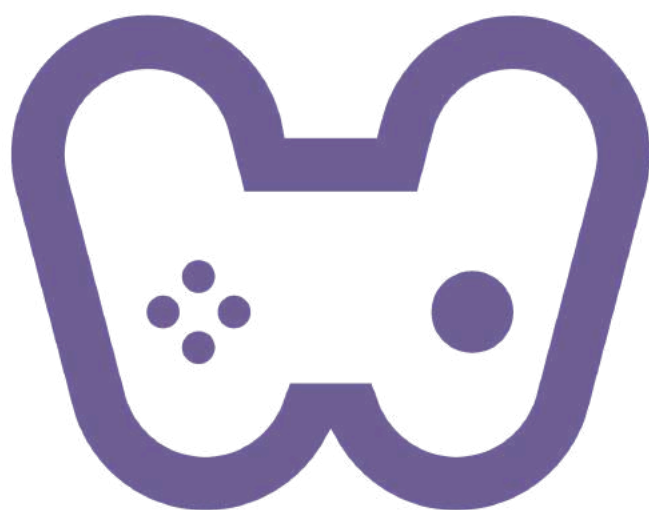
Para finalizar, puedes realizar un relevamiento general del curso, con el objetivo de conocer las opiniones y/o demandas de los estudiantes. Algunos interrogantes para debatir: ¿Les gustó el curso? ¿Qué fue lo que mas les gusto y que menos les gusto? ¿Qué aprendieron del mundo de la computación? ¿Son contenidos que le pueden llegar a servir? ¿Les gustaría seguir aprendiendo con los otros cursos que ofrece Wonderly?

Recuerden que el aprendizaje no termina aquí. El mundo de la computación y programación es un campo vasto y apasionante, y siempre hay más por descubrir y explorar. Continúen alimentando la curiosidad de los estudiantes, que sigan aprendiendo y nunca dejen de desafiarse a sí mismos. ¡Continuará esta ruta de aprendizaje con más contenidos próximamente!

CONSEJO DIDÁCTICO

¡Educadores! Pueden proponer una dinámica oral compartiendo las preguntas o de forma escrita en donde los estudiantes destinen unos minutos para responderlas.





El placer de aprender tecnología

