



Fundamentos de Programación en Videojuegos I

Un camino sin retorno...

Presentación de docentes y materia. Perfil del egresado. Industria Global, regional y local. Requisitos de aprobación, instancias y modalidades de evaluaciones. Fechas importantes, cronograma, control de notas y campus. Tips de cursada.



CUERPO DOCENTE

NÉSTOR ARRIOLA

VIERNES DE 18 A 20

PROFE 1

MICAELA PINTOS

MARTES DE 18 A 20

PROFE 2

LUCA MOYANO

MARTES DE 18 A 20

PROFE 1



PERFIL DEL EGRESADO/A



La posición ocupacional de la figura que es referencia del presente trayecto es la que suele denominarse Programador/a de Videojuegos que, integrando equipos de proyectos dedicados al desarrollo de videojuegos, a partir de especificaciones de diseño y funcionalidad, construyen la parte que les fuera asignada a desarrollar a partir de módulos existentes o adaptando y escribiendo uno nuevo que cumpla con los requerimientos, además de testear su trabajo, verificando lo producido y entregando lo realizado para la prueba de desarrollo final.

También puede desempeñarse en forma autónoma asumiendo la mayor parte de las tareas propias del proceso, resolviendo problemas de pequeñas organizaciones que requieren videojuegos de baja complejidad y reducida dimensión

Nuestros Objetivos: que el/la estudiante...

- Comprenda los fundamentos de la programación imperativa bajo el paradigma estructurado.
- Elabore modelos computacionales que permitan resolver problemas de complejidad baja relacionados con programas interactivos, tratamiento de secuencias y representaciones en el espacio bidimensional empleando los algoritmos y estructuras de datos más adecuados.
- Comprenda las reglas sintácticas, semánticas y de estilo de los lenguajes formales, especialmente las del lenguaje de programación GD Script y C#.
- Construya implementaciones de los modelos computacionales elaborados empleando los lenguajes programación GD Script y C#.
- Elabore casos y escenarios de prueba elementales que permitan evaluar el funcionamiento de las soluciones.
- Desarrolle habilidades relacionadas con la depuración de programas, especialmente las que permitan localizar y aislar errores.



Nuestros Objetivos: que el/la estudiante...

- Asimile y recuerde: ¡¡¡Programar es **ENTENDER** una situación o problema, proponer una **SOLUCIÓN** a través de una **ESTRATEGIA** y **COMUNICAR** dicha solución a las **PERSONAS** y **MÁQUINAS**. Es por eso que...
 - Tus programas deberían quedar claros a partir de la lectura.
- Sí leo el punto de entrada, tiene que quedar más que explícita la estrategia elegida.
- Usamos procedimientos, funciones y métodos para la claridad, legibilidad y expresar la estrategia.
 - Es importantísimo elegir nombres adecuados para los procedimientos, funciones y métodos que definimos.
- En lo posible, no anidar estructuras. Usamos procedimientos para dividir el problema en partes pequeñas y darles nombres adecuados.
- Definimos expresiones con nombres claros a partir de otras para comunicar mejor, y dividir el problema de las expresiones, también, en partes más pequeñas.



NUESTRO PLAN (tentativo)

CRONOGRAMA

Cronograma de clases - 1°
Cuatrimestre - 2025

1° PARCIAL

Evaluación individual



CONTROL

Control de Notas

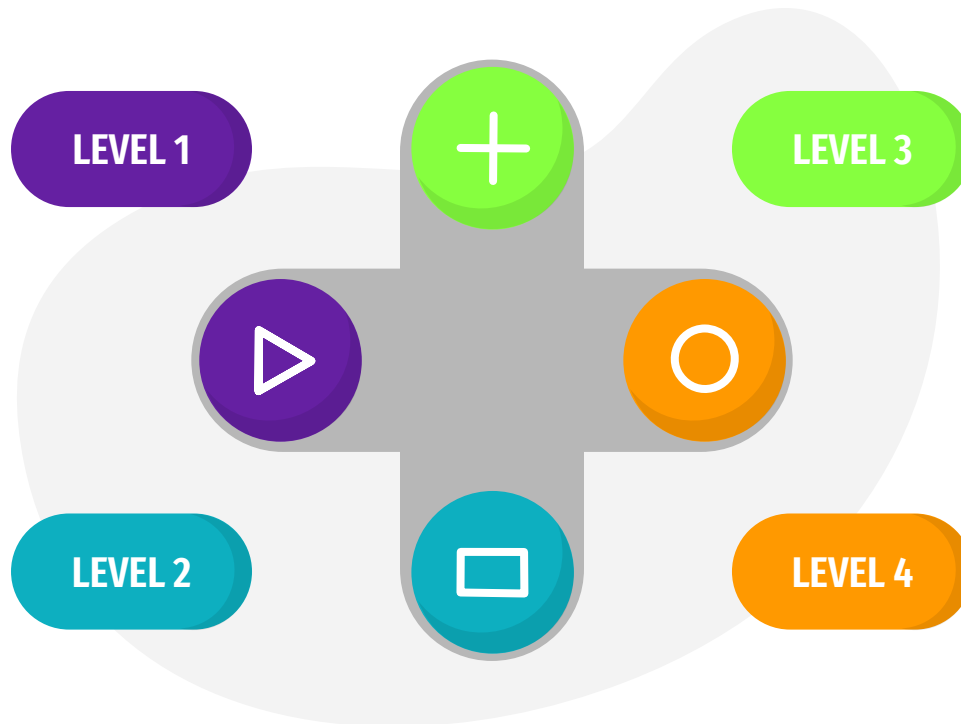
2° PARCIAL

Juego Grupal

Requisitos de la materia

75 % de asistencia al día de las evaluaciones.

PARCIALES: Los parciales se aprueban con 4 (70% del examen)



Promoción directa requiere obtener calificaciones no inferiores a 6 (seis) en las dos evaluaciones parciales o sus respectivos recuperatorios, y un promedio de 7 (siete) o más.

En el caso de no promocionar, aprobar el examen integrador o su (Nota min de aprobación: 4)
En el caso de no aprobar el integrador, aprobar el examen final. (Nota min de aprobación: 4)

IMPORTANTE

LOS RECUPERATORIOS DE LOS PARCIALES 1 Y 2, SE REALIZARÁN FINALIZANDO LA CURSADA.

ES POR ESTA RAZÓN QUE SOLO SE PUEDE RECUPERAR 1 PARCIAL, ES DECIR, COMO ESTUDIANTES ESTÁN OBLIGADOS A APROBAR UNO DE LOS PARCIALES SI O SI.



TIPS de cursada



1

El que tiene mala memoria se crea una de papel.

Armá un machete con los conceptos más importantes y la sintaxis para escribir programas, y tenerlo siempre a mano al momento de realizar actividades.

TIPS de cursada

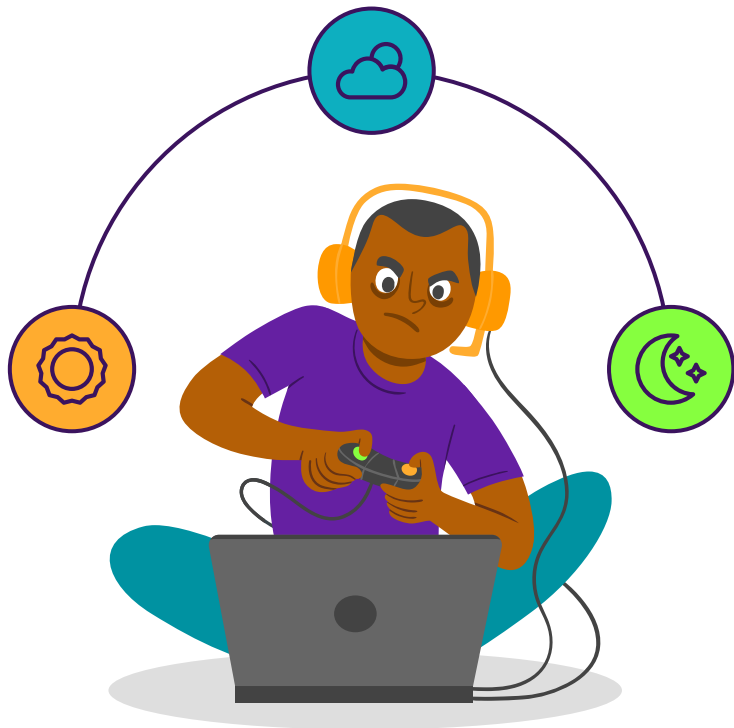


2

La práctica hace al maestro.

Volvé atrás y asegúrate de haber terminado todos los ejercicios realizados.

TIPS de cursada



3

Sí está mal, entonces hazlo bien.

Sí por falta de algún conocimiento que ahora adquiriste, encaraste ejercicios de forma incorrecta, volvé a realizarlos y terminalos de forma adecuada.

TIPS de cursada



4

El trabajo es siempre en equipo

Es una de las aptitudes más valoradas en las IT y la más difícil de cultivar.

TIPS de cursada



5

Corrección fraterna y docencia

No solo los profes corrigen las prácticas, compartan sus ejercicios entre ustedes, pídanse ayuda mutuamente, corrijanse y encontrarán en esa noble actividad una gran herramienta de aprendizaje.

AVISO de cursada

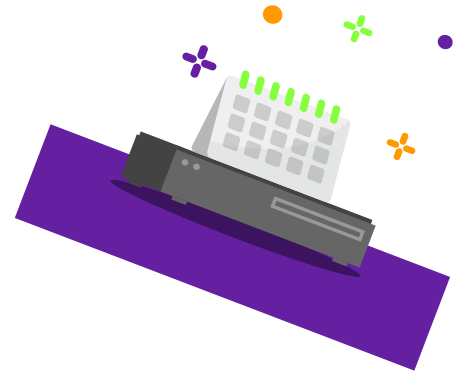


6

La situación educativa en el país es compleja, los/as docentes en este contexto estamos desbordados en las universidades lo que en ocasiones influye en algún aspecto de la cursada,.

Por esto y si bien toda nuestra comunicación debe y será a través del campus, podría pasar que no podamos responder de manera inmediata a las consultas que se hagan fuera del horario de clase, por eso les pedimos que procuren resolver todas las dudas en clase.

¿Por qué eligieron VJJ?



LA INDUSTRIA DE LOS VVJJ

QUÉ ES UNA INDUSTRIA

Actividad económica y técnica que consiste en transformar las materias primas hasta convertirlas en productos adecuados para satisfacer las necesidades del ser humano.



Implica un proceso productivo.

La industria convierte las materias primas en productos acabados.

Las empresas industriales requieren de recursos energéticos y de tecnología.

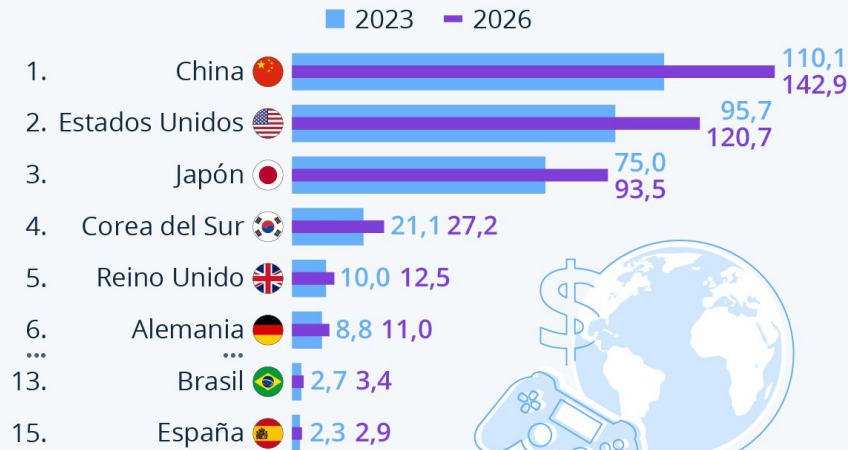
El producto manufacturado será de consumo final.

El concepto de industria está ligado al de actividad manufacturera.

VWJ, LA INDUSTRIA DE ENTRETENIMIENTO MÁS GRANDE DEL MUNDO

Los principales mercados de videojuegos

Países con los mayores ingresos generados por la industria de los videojuegos (en miles de millones de USD)



Estimaciones de agosto de 2023.

Fuente: Statista Market Insights



statista

Los datos del Statista Market Insights muestran que los videojuegos conforman una industria enormemente popular y rentable en todo el mundo, con una previsión de ingresos globales de cerca de 400.000 millones de dólares en 2023.

De los seis mayores mercados mundiales de videojuegos, tres son países asiáticos, con China a la cabeza, donde se prevé que la industria del gaming facture casi 143.000 millones en 2026.

EN EL MUNDO



CHINA

League of Legends
Oxygen Not Included
Valorant
Warframe

EN EL MUNDO



FRANCIA

Rayman
Tom Clancy
Star Wars
Assassin's Creed

EN EL MUNDO



USA

League of Legends
Valorant

EN EL MUNDO

SONY

JAPÓN

Gran Turismo
Grand Theft Auto
God of War

Naughty Dog
Santa Monica Studio
Guerrilla Games
PlayStation

EN EL MUNDO



JAPÓN

Donkey Kong
Super Mario
Valorant
Pokémon

EN EL MUNDO



USA

Crash Bandicoot
Guitar Hero
Call of Duty
Spyro el dragón
Warcraft
Overwatch
Candy Crush

EN LA REGIÓN



CHILE

Zeno Clash

Rock Of Ages

Abyss Odyssey

The Deadly Tower Of Monsters

Solseraph

The Eternal Cylinder

EN LA REGIÓN



URUGUAY

Kingdom Rush
Iron Marines Invasion
JunkWorld

EN LA REGIÓN



COLOMBIA

Cris Tales

EN LA REGIÓN



MÉXICO

Aztech Forgotten Gods
Mulaka
Hunter's Legacy

EN LA REGIÓN



BRASIL

Wonderbox: The Adventure Maker
Horizon Chase
Looney Tunes: World of Mayhem

EN ARGENTINA

Globant 

Hogwarts Legacy

Fortnite

FIFA

Fall Guys

HitMan

EN ARGENTINA



Mis ladrillos
Regnum Online
Master of Orion
Quantum league

EN ARGENTINA

etermax

**Apalabrados
Mezcladitos
Preguntados**

EN ARGENTINA



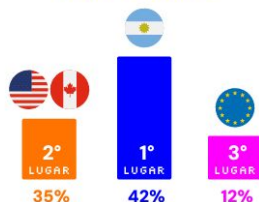
Peanuts: Snoopy's Town Tale
Garfield: Survival of the Fattest
Goosebumps HorrorTown
The Sandbox

EN ARGENTINA

RESUMEN ANUAL 2023



Principales Mercados



Tipos de Vj Desarrollados

CASUALES
40%

PUZZLES
33%

ACCIÓN
29%

Cantidad de Vj Publicados 2022



Tamaño estimado de la industria*

U\$S 91.545.600

*Calculado sobre la totalidad de RRHH en actividad

Tamaño de la empresa*



*Micro ≤ 7 // Pequeña 8-35 // Mediana 36-345 // Grande ≥ 346

Principales Plataformas



RR.HH

Profesionales en actividad

1646

MUJERES EN LA INDUSTRIA:

Empleados con rol profesional 21% MUJERES

Total de RRHH (socios + directivos incluidos) 17% MUJERES

Empleadas mujeres 66% PUESTOS STEM*

Cargos directivos 24% MUJERES

*ciencia, tecnología, ingeniería y matemáticas

Público objetivo

95% 19-35 años
78% 14-18 años
57% 36-60 años

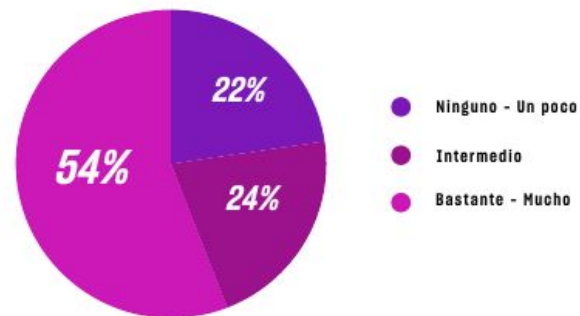
EN ARGENTINA

// RESUMEN ANUAL

Ranking de perfiles más difíciles de conseguir

1º	Comercialización
2º	Desarrollo
3º	Game Design
4º	Producción
5º	Arte

¿En qué medida considera que la subcontratación de personal en el exterior dificulta la búsqueda de personal para su propia empresa/emprendimiento?



EN LA INDUSTRIA



EMPECEMOS DESDE AHORA...

1

Define un cronograma de actividades en donde se vea claramente cuáles son las horas que le vas a dedicar a la facultad.

2

HÁBITOS DE ESTUDIO

Definir un objetivo de estudios, estudia siempre a la misma hora, no descuides tus horas de sueño, desactiva las notificaciones de tu teléfono.

3

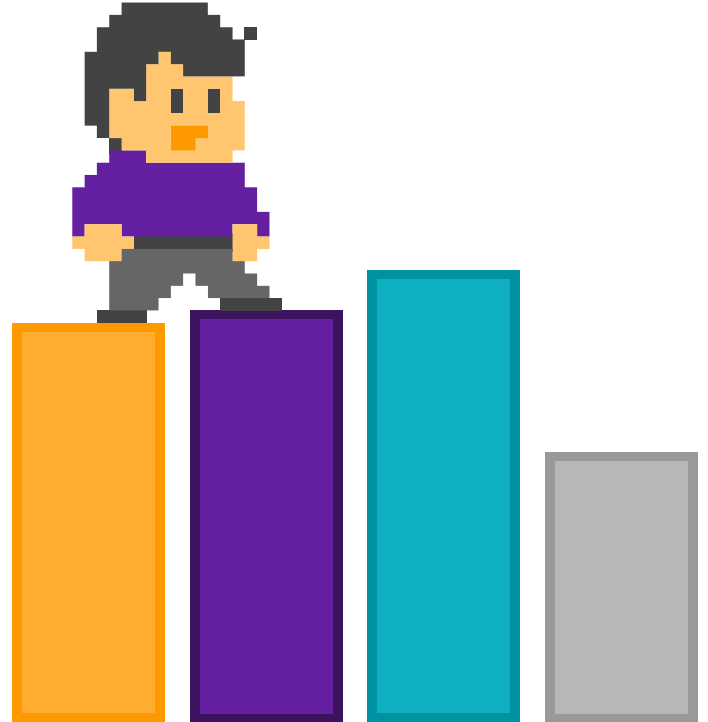
Grupos de estudios.

4

Asiste a clase, haz las prácticas, participa en clase, colabora con tus compañeros/as.

5

Como estudiantes universitarios y adultos, serán los únicos responsables de la calidad de su cursada. NUNCA NADIE LES PEDIRÁ EXCUSAS.



INTRODUCCIÓN A LA PROGRAMACIÓN: ¿qué es la programación? - ¿qué es el software? - ¿qué es el hardware? - dispositivos, memoria, procesador.



Computadora

Computadora

Una computadora es una máquina capaz de transformar datos en información.

La gracia es que pueden hacer esto rápido, muy rápido. De hecho, la palabra computadora viene de computar, que significa calcular.

Las computadoras, en un nivel más elemental, lo único que hacen son cálculos matemáticos, muchos, muchísimos, muy rápidamente (del orden del billón por segundo).

Los datos de entrada pueden provenir de distintas fuentes (teclado y mouse, sensores, etc.) y los datos de salida pueden también presentarse en diversas formas (pantalla, impresora, parlantes, etc.). En la vida moderna, las computadoras están tan presentes, y hacen tantas cosas, que muchas personas desconocen su verdadera naturaleza.

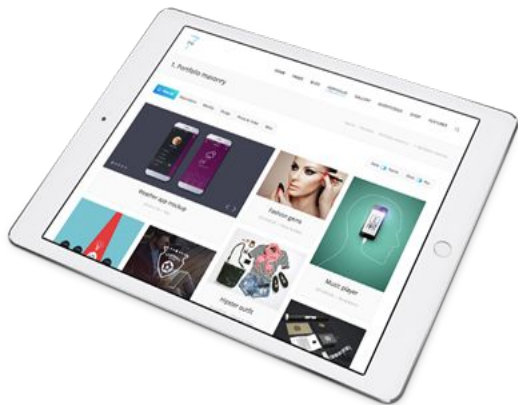
Computadora



Computadora



Computadora



Computadora



Hardware y Software

Claramente hay de muchas formas, y son muy distintas entre sí, en su función, en la forma de operarlas, etc.

Pero todas tienen partes internas similares. En particular hay dos aspectos claves de una computadora:

Hardware y Software

Claramente hay de muchas formas, y son muy distintas entre sí, en su función, en la forma de operarlas, etc.

Pero todas tienen partes internas similares. En particular hay dos aspectos claves de una computadora:

- **El Hardware:** El hardware comprende todos los elementos físicos que componen a la computadora. Es decir, el conjunto de circuitos, cables, perillas, palancas, botones, luces, displays, dispositivos de impresión, motores, imanes, placas metálicas, etc.
- **El Software:** El software es el conjunto de los programas de cómputo, archivos, procedimientos, reglas, documentación y datos asociados, que forman parte de las operaciones de un sistema de computación.

Computadora

Como regla del pulgar, si no anda, y lo puedo patear, es hardware.



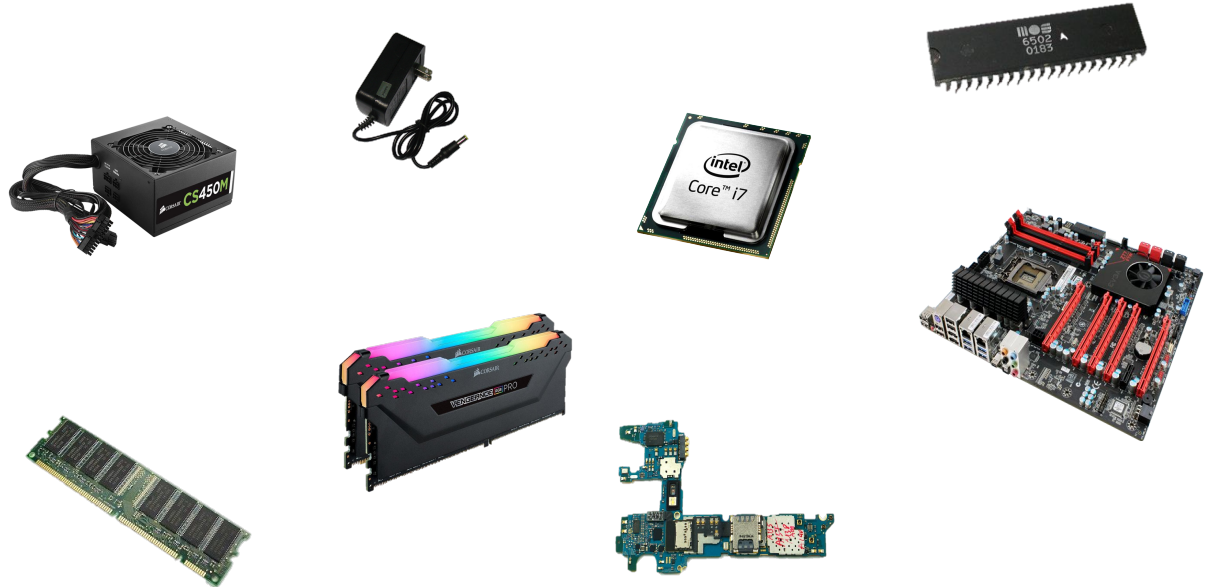
Si no anda, y solo lo puedo insultar, es software.



Hardware

La mayoría de las computadoras poseen una serie de partes más o menos estándar en términos físicos:

- Fuente de alimentación
- CPU
- Memoria RAM
- Motherboard
- Periféricos

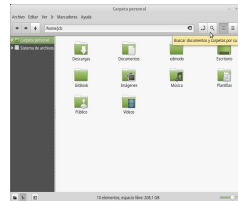
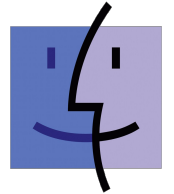
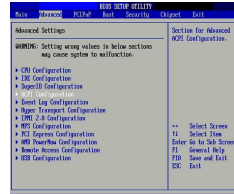


Software

Una computadora sin software no sirve para nada, pues sin electricidad, simplemente no funciona. Más aún, no solo debe haber electricidad, sino que debe haber electricidad en los lugares correctos en el momento correcto para que el equipo funcione de la forma esperada.

Entre el software al que podemos nombrar podemos mencionar:

- Firmware
- Sistema Operativo
- Programas
- Archivos



Mac™ OS

¿Qué es la programación?

La programación es una actividad fascinante que consiste en darle instrucciones a una computadora para que realice tareas específicas. Es como escribir una receta, pero en lugar de decirle a alguien cómo cocinar, le decimos a una máquina cómo hacer cálculos, mostrar imágenes en una pantalla, controlar robots, y mucho más.



Ok...

BE CUTE !!!
YOU M@#!? F%&*!!!



¿Qué es la programación?

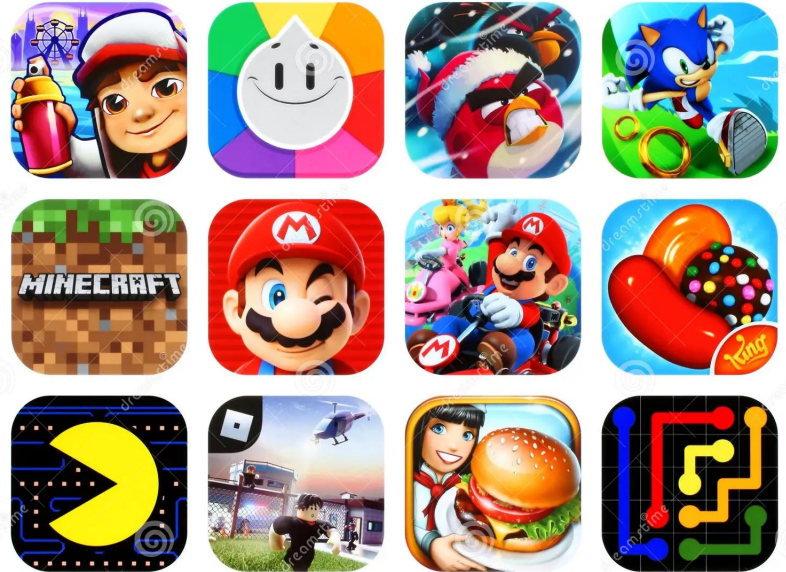
En términos más técnicos, la programación es el proceso de crear un conjunto de instrucciones, escritas en un lenguaje de programación, que la computadora puede entender y ejecutar. Estos lenguajes son como idiomas, pero diseñados específicamente para comunicarse con las máquinas.



¿Qué es un programa?

En general cuando hablamos de programa estamos hablando de un producto de software que es capaz de ejecutar en una computadora, y que resuelve un problema específico.

Es importante notar que “un programa” no solo está compuesto de ese software. Hay muchos otros productos, como archivos de configuración, documentación, manuales, etc. que también forman parte de “un programa”, pero nos vamos a centrar en lo que llamamos generalmente “aplicación” (ese ícono al que le damos doble clic y abre una pantalla en nuestra computadora para que hagamos algo específico)



¿Qué es un programa?

Un programa es un texto que describe la solución a un problema computacional y que puede ser ejecutado por una computadora.

¿Qué es un programa?

Un programa es un **texto** que describe la solución a un problema computacional y que puede ser ejecutado por una computadora.

En las computadoras modernas, un programa se escribe, en general en un archivo de texto. El texto describe (de alguna forma) cómo se soluciona un problema.

¿Qué es un programa?

Un programa es un texto que describe la solución a un **problema computacional** y que puede ser ejecutado por una computadora.

Un problema computacional es aquel que puede ser resuelto por una computadora.

Por ej. sí mi problema es “mi novia no me quiere, me abandonó por otro y eso hace que la vida no tenga sentido...”, no voy a poder construir un programa que lo solucione, necesito probablemente años de terapia.

Un problema computacional entonces tiene que estar expresado como una transformación de información (recordemos, las computadoras toman información como entrada y producen información como salida).

¿Qué es un programa?

Un programa es un texto que describe la solución a un **problema computacional** y que puede ser ejecutado por una computadora.

“Dados 3 notas de un estudiante, calcular su promedio” es un problema computacional, hay información de entrada (las notas del estudiante) y se desea obtener una salida (el promedio). El programa describe entonces cómo se transforma esa entrada en la salida deseada.

Otros ejemplos podrían ser: “buscar la ruta con menos tránsito entre 2 puntos de un mapa”, “calcular cuál es la mejor inversión en bolsa según las últimas cotizaciones”, etc.

¿Qué es un programa?

Un programa es un texto que describe la solución a un problema computacional y que puede ser **ejecutado por una computadora.**

Para que un programa sea un programa, tiene que poder ser ejecutado por una computadora. Una descripción en español de como solucionar el problema, no alcanza, si una computadora no puede ejecutar dicha descripción.

Notar que el programa no soluciona el problema, la máquina, al ejecutar el programa descrito por el texto, es la que lo soluciona.

INTÉRPRETES, COMPILADORES Y LENGUAJES DE PROGRAMACIÓN.

PARADIGMAS.



Lenguajes naturales

Los lenguajes naturales son los lenguajes que utilizamos para comunicarnos de forma cotidiana.

El español, el inglés, el catalán, son lenguajes. Los símbolos que usamos (las letras y signos de puntuación) no son idénticos en cada uno de ellos (la Ñ, por ejemplo, es única del español). Las reglas de composición (palabras, reglas gramaticales) tampoco son iguales.

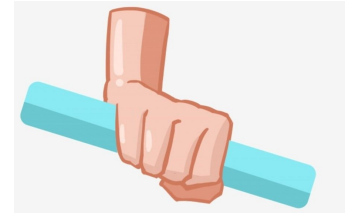
El problema con esos lenguajes es que son ambiguos. Hay muchas formas de decir lo mismo, y hay cosas que decimos, que solo pueden ser entendidas mediante un análisis del contexto y expresiones idiomáticas.

Cómo se interpreta la frase “lo voy a tomar”

Lenguajes naturales

Cómo se interpreta la frase “lo voy a tomar”.

- Puedo estar hablando de un vaso de un líquido
- Puedo estar hablando de un medio de transporte
- Puedo estar hablando de “asir” o “agarrar” algo.



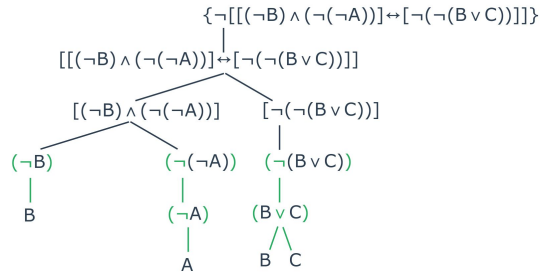
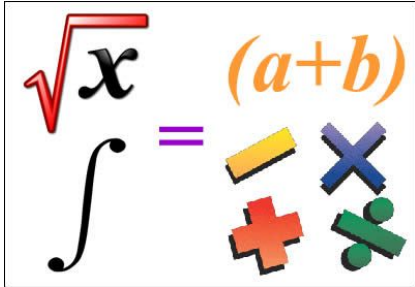
La frase sólo se entiende en el contexto.

Las computadoras no entienden de contexto, necesitan lenguajes no ambiguos.

Lenguajes formales

Un lenguaje formal es un conjunto de símbolos y reglas sintácticas que permiten construir expresiones bien formadas.

Ejemplos de lenguajes formales son el matemáticas, el lógico y los lenguajes de programación.



Lenguajes de programación

Un lenguaje de programación es un lenguaje, que no es ambiguo, y cuya función es permitir al programador comunicarse con la computadora.

Cuando escribimos un programa lo hacemos en un lenguaje de programación específico, y ese lenguaje asociará una sintaxis (símbolos y reglas) con una semántica (una interpretación sobre qué es lo que debe entenderse al leer una secuencia de símbolos determinada).

El lenguaje es lo que nos va a decir cómo debe escribirse la semántica “5+7”. Según las reglas que disponga podrá ser “Sumar 5 y 7”, “sumar 5, 7”, “sum 5 7” o cualquier cosa que se nos ocurra.

Del texto a la ejecución

¿Qué distingue a un archivo de texto con código de un programa ejecutable?

Del texto a la ejecución

¿Qué distingue a un archivo de texto con código de un programa ejecutable?

Hay tres formas distintas de que ese programa se vuelva ejecutable.

- Mediante un intérprete.
- Mediante un compilador.
- Mediante alguna combinación de las anteriores.

Intérprete

1

Un intérprete es una aplicación que es capaz de leer el texto de un programa y luego realizar en la computadora cálculos asociados.

Intérprete

1

Un intérprete es una aplicación que es capaz de leer el texto de un programa y luego realizar en la computadora cálculos asociados.

2

Por ejemplo, si nuestro programa dice algo como “sumar 5 y 7”, el intérprete es capaz de leer ese texto y realizar “ $5+7$ ” en el CPU de la computadora.

Intérprete

1

Un intérprete es una aplicación que es capaz de leer el texto de un programa y luego realizar en la computadora cálculos asociados.

2

Por ejemplo, si nuestro programa dice algo como “sumar 5 y 7”, el intérprete es capaz de leer ese texto y realizar “5+7” en el CPU de la computadora.

3

Un intérprete está pensado para comprender un lenguaje de programación específico. O sea, habrá un intérprete para Python, un intérprete para PHP, etc.

Intérprete

1

Un intérprete es una aplicación que es capaz de leer el texto de un programa y luego realizar en la computadora cálculos asociados.

2

Por ejemplo, si nuestro programa dice algo como “sumar 5 y 7”, el intérprete es capaz de leer ese texto y realizar “5+7” en el CPU de la computadora.

3

Un intérprete está pensado para comprender un lenguaje de programación específico. O sea, habrá un intérprete para Python, un intérprete para PHP, etc.

4

El texto del programa a interpretar tiene que cumplir con las reglas planteadas por ese lenguaje de programación, para que el programa pueda ser interpretado de forma adecuada.

Intérprete

1

Un intérprete es una aplicación que es capaz de leer el texto de un programa y luego realizar en la computadora cálculos asociados.

2

Por ejemplo, si nuestro programa dice algo como “sumar 5 y 7”, el intérprete es capaz de leer ese texto y realizar “5+7” en el CPU de la computadora.

3

Un intérprete está pensado para comprender un lenguaje de programación específico. O sea, habrá un intérprete para Python, un intérprete para PHP, etc.

4

El texto del programa a interpretar tiene que cumplir con las reglas planteadas por ese lenguaje de programación, para que el programa pueda ser interpretado de forma adecuada.

5

Los creadores de un lenguaje de programación suelen proveer herramientas específicas para trabajar con el lenguaje, lo cual puede incluir un intérprete.

Compiladores

- Los archivos poseen lo que se conoce como “cabecera” que le indica al sistema operativo cómo debe interpretarlo. En algunos sistemas operativos está dado por la extensión de archivo, un sufijo (en general oculto a simple vista) del nombre de un archivo en la computadora.
- Por ej. mi archivo.txt es un archivo que contiene texto (.txt es la extensión para textos)
- Mientras que programa.exe es un archivo de un programa ejecutable (.exe es la extensión para programas)
- Por supuesto, el binario asociado a un programa como texto, y a un programa como un programa ejecutable suele ser muy distinto (usan una codificación distinta del binario al momento de interpretarse).

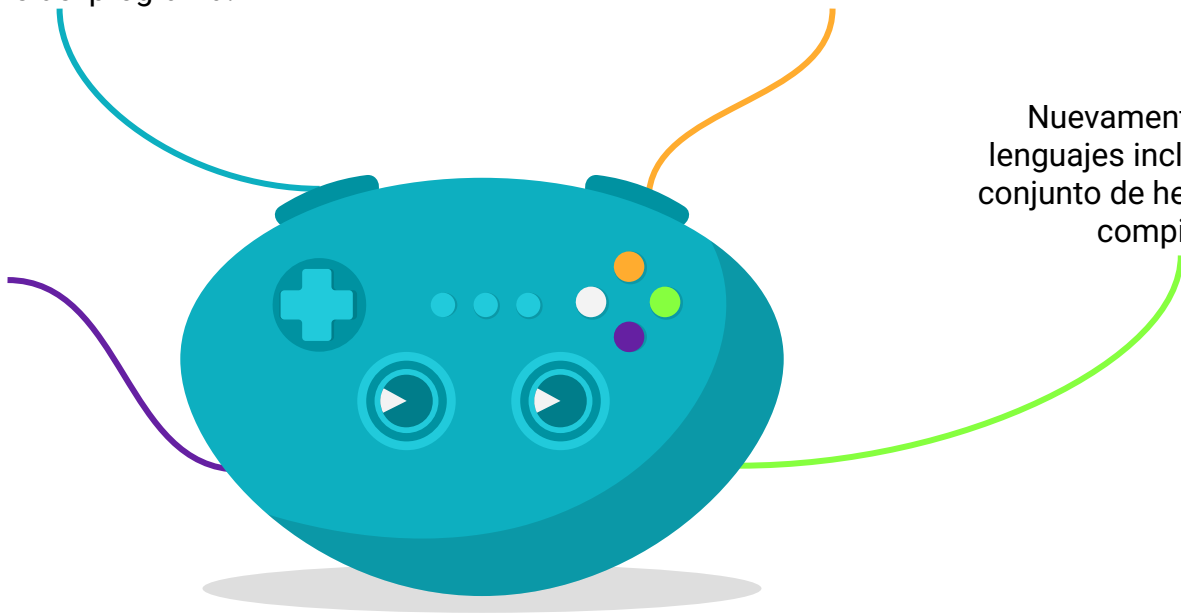
Compiladores

Decimos que el texto que describe el programa es el código fuente de un programa. Una versión de aplicación ejecutable generada a partir de esa descripción es el código objeto o código ejecutable del programa.

Un compilador transforma de código fuente a código objeto.

Un compilador es un programa capaz de tomar un archivo de texto que contiene un programa escrito en un lenguaje de programación específico, y producir un archivo de aplicación ejecutable.

Nuevamente, algunos lenguajes incluyen entre su conjunto de herramientas un compilador.



¿Compilados o interpretados?

- Un lenguaje de programación no es ni interpretado, ni compilado. El mismo programa puede interpretarse o compilarse, según la herramienta que se use.
- En general mucha gente habla erróneamente de “lenguajes compilados” o “lenguajes interpretados” porque los creadores del lenguaje suelen incluir una de las dos herramientas (y no ambas). Pero nada impide que se genere un intérprete para un lenguaje que se provee con un compilador, o viceversa.
- Más aún, muchas herramientas hacen cosas intermedias. Por ejemplo, realizan una pre-compilación a un lenguaje intermedio, más simple de interpretar, y luego interpretan ese código intermedio (Python y Java son casos de esto).

Intérprete y compilador

El intérprete y el compilador no entienden cualquier cosa. Para una computadora no es lo mismo decir:

- “suma 5 y 7”
- “sumar 5, 7”
- y “sumar 5 y 7”.

Aunque las personas lo entendamos como sinónimos de la misma idea.

PARADIGMAS

Los lenguajes de programación son los que determinan cómo se escribe un programa en código.

El código, sin embargo, intentan reflejar abstracciones de ideas que están en la mente del programador. Y cómo tales, estas ideas pueden ser muy dispares pues provienen de lugares muy distintos.

Un paradigma de programación es una forma de concebir un programa (en términos abstractos) para describir la solución a un problema computacional.

Existen muchos paradigmas, algunos más estandarizados y difundidos que otros.

Principios de programación imperativa

La programación imperativa es un paradigma de programación donde se le indica a la computadora una secuencia de pasos a seguir para lograr un resultado determinado. Es como darle una receta de cocina, donde cada paso debe seguirse en un orden específico para obtener el plato deseado.

Los programas de este paradigma tienen como concepción fundamental la idea sobre elementos clave:

Principios de programación imperativa

Estado

En la programación imperativa, los programas tienen un estado que cambia a medida que se ejecutan. Este estado se representa mediante variables que almacenan datos.



Principios de programación imperativa

Estado

En la programación imperativa, los programas tienen un estado que cambia a medida que se ejecutan. Este estado se representa mediante variables que almacenan datos.

Instrucciones

Son las acciones que modifican el estado del programa. Estas pueden ser asignaciones, llamadas a funciones, condicionales o bucles.



Principios de programación imperativa

Estado

En la programación imperativa, los programas tienen un estado que cambia a medida que se ejecutan. Este estado se representa mediante variables que almacenan datos.

Instrucciones

Son las acciones que modifican el estado del programa. Estas pueden ser asignaciones, llamadas a funciones, condicionales o bucles.



Secuencia

Las instrucciones se ejecutan en un orden específico, generalmente de arriba hacia abajo.

Principios de programación imperativa

Estado

En la programación imperativa, los programas tienen un estado que cambia a medida que se ejecutan. Este estado se representa mediante variables que almacenan datos.

Instrucciones

Son las acciones que modifican el estado del programa. Estas pueden ser asignaciones, llamadas a funciones, condicionales o bucles.



Secuencia

Las instrucciones se ejecutan en un orden específico, generalmente de arriba hacia abajo.

Variables

Son contenedores que almacenan datos de diferentes tipos (enteros, flotantes, cadenas, etc.).

Principios de programación imperativa

Expresiones

Combinan variables, operadores y valores para producir un nuevo valor.

5



Principios de programación imperativa

Expresiones

Combinan variables, operadores y valores para producir un nuevo valor.

5

Comandos

Son instrucciones completas que realizan una acción, como asignar un valor a una variable o ejecutar un bloque de código.

6



Principios de programación imperativa

Expresiones

Combinan variables, operadores y valores para producir un nuevo valor.

Comandos

Son instrucciones completas que realizan una acción, como asignar un valor a una variable o ejecutar un bloque de código.

5

6



7

Funciones

Son bloques de código reutilizables que realizan una tarea específica.

Principios de programación imperativa

Expresiones

Combinan variables, operadores y valores para producir un nuevo valor.

Funciones

Son bloques de código reutilizables que realizan una tarea específica.

Comandos

Son instrucciones completas que realizan una acción, como asignar un valor a una variable o ejecutar un bloque de código.

Procedimientos

Similar a las funciones, pero no devuelven un valor.



Principios de programación imperativa

Un programa en este paradigma es una combinación de comandos, funciones, procedimientos, variables y expresiones, que se disponen con una estructura secuencial. Además, existen elementos que permiten cambiar el flujo secuencial del programa, para obtener flujos distintos.



ACTIVIDAD Y TAREA

RESOLVER LOS EJERCICIOS DEL NIVEL PRINCIPIANTE

PROGRAMANDO EN LA COMPUTADORA:

Dieta a base de churrascos

- Desafío 1
- Desafío 2
- Desafío 3

REPETICIÓN:

Más churrascos para Duba

- Desafío 1
- Desafío 2
- Desafío 3

