

EJERCICIOS UT1

EJERCICIOS UT1_1

ACTIVIDAD 1.1

La vida y obra de Alan Turing; puesto que forma parte del origen del software moderno.

Nacido en Londres en 1912, fue un matemático considerado como uno de los precursores de la informática moderna. Sus primeros 13 años los pasó en la India, ya que su padre trabajaba en la Administración colonial. Al volver a su país estudia en King's College, y al terminar la carrera, se traslada a la Universidad de Princeton, es EEUU. Cabe destacar que aquí trabaja con el lógico Alonzo Church.

En 1937 publica un artículo donde define la máquina de Turing, la cual es una máquina calculadora de capacidad infinita. Muchas de sus teorías tuvieron algunas aplicaciones en la Segunda Guerra Mundial, ya que había una necesidad de descifrar los mensajes codificados de la Marina alemana.

También definió un método teórico para comprobar si una máquina podría ser capaz de pensar como un hombre, llamado Test de Turing. Además, realizó contribuciones a otras ramas de la matemática aplicada.

Finalmente murió en Reino Unido en el año 1954 por intoxicación. Hay varias especulaciones alrededor de su muerte sobre si fue un accidentado o intencional.

El firmware:

Los softwares están formados por distintas capas, las cuales interactúan y se comunican entre sí. Y cada una con una determinada función. Las capas mas altas se encargan de interactuar con el usuario, y estas si las podemos ver. Un ejemplo de esta capa principal es la interfaz de una Smart TV. Pues el firmware es la capa más baja de todas, la cual se encarga de comunicarse con el Hardware en sí. Está presente en dispositivos de memoria no volátil como la ROM o la Flash.

ACTIVIDAD 1.2

Indica si la siguiente afirmación es verdadera o falsa y razona tu respuesta:
El firmware puede considerarse un software

Sí, se considera software a pesar de tener algunas características que lo hacen distinto a resto de softwares, como es en sí el estar incrustado al Hardware.

EJERCICIOS UT1_2

ACTIVIDAD 2.1

Busca ejemplos de cada tipo de software basado en el método de distribución:

TIPO DE SOFTWARE	NOMBRE DE SOFTWARE	DESCRIPCIÓN
Shareware	Kaspersky	Es un antivirus
Freeware	Discord	Servicio de mensajería instantánea con chat de voz
Adware	OpenCandy	Este programa recomendaba softwares durante la instalación de otros programas
Software multimedia	Photoshop	Programa diseñado para la edición de imágenes
Software de uso específico	Pro Tools	Diseñado para crear música y edición de audio

EJERCICIOS UT1_3

ACTIVIDAD 3.1

¿Qué diferencia hay entre software, programa informático y aplicación informática?

El software es el término que se refiere a los componentes lógicos de una computadora o sistema informático, además de programas, datos o instrucciones entre otras cosas. En el software se pueden incluir programas y aplicaciones informáticas.

Un programa informático son instrucciones codificadas que se usan para realizar una tarea específica en la computadora.

Una aplicación es un programa diseñado para que los usuarios realicen determinadas actividades. Es una herramienta.

Por tanto, la relación sería que un software puede estar formado por programas y aplicaciones, y una aplicación puede estar formada por programas.

EJERCICIOS UT1_4

ACTIVIDAD 4.1

Busca 5 ejemplos de frameworks y los lenguajes de programación con los que trabaja.

- React: usa el lenguaje JavaScript
- Express.js: usa el lenguaje JavaScript
- Ruby on Rails: usa el lenguaje Ruby
- Django: usa el lenguaje Python
- Angular: usa el lenguaje TypeScript, que viene del JavaScript

Dar 1 ejemplo de aplicación famosa realizada con cada uno de estos frameworks.

- Facebook: fue creado con React
- Walmart: creado con Express.js
- GitHub: creado con Ruby on Rails
- Instagram: creada con Django
- LinkedIn: creada con Angular

ACTIVIDAD 4.2

Añade características nuevas, que no se hayan visto en clase, para cada lenguaje de programación de la lista anterior.

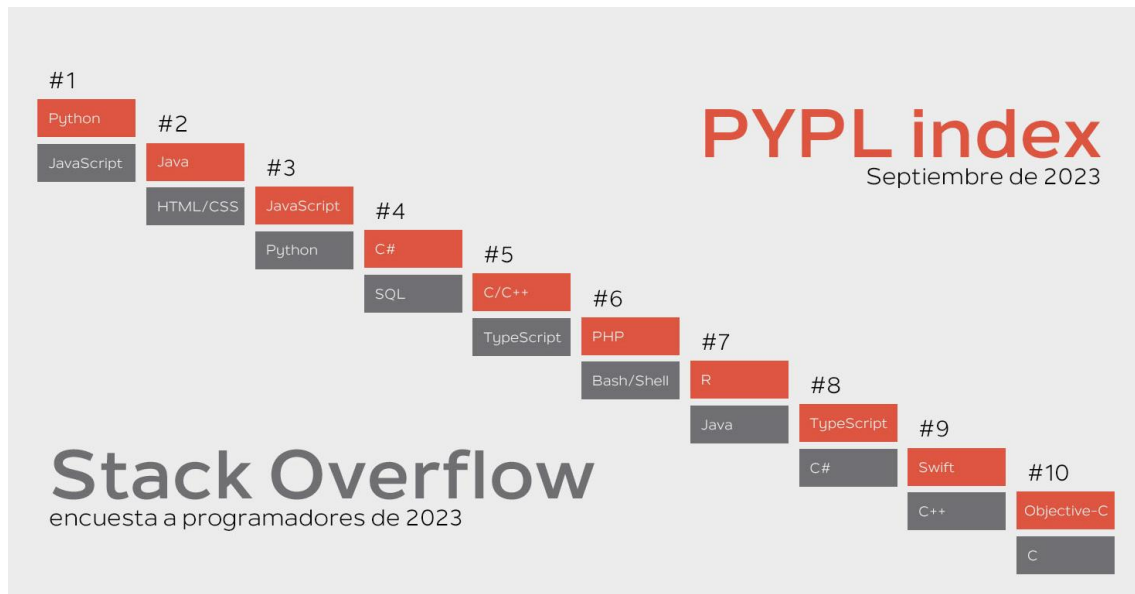
- Python:
 - Es interpretado, de forma que no hay que compilar su código antes de ejecutarlo
 - Versátil, se puede usar en numerosos ámbitos
 - Tiene una comunidad grande y activa
- JavaScript:
 - Es compatible con la mayoría de los navegadores web
 - Convierte webs en interactivas
 - Es asincrónico

- Java:
 - Está orientado a objetos
 - Se puede ejecutar en múltiples plataformas
 - Tiene un gran enfoque en la seguridad
 - Gran comunidad, bibliotecas y frameworks
- C#:
 - Al igual que Java, está orientado a objetos
 - Facilita el desarrollo de aplicaciones empresariales y servicios web
 - Gran seguridad
- Ruby:
 - Lenguaje dinámico
 - Tiene una sintaxis fácil de comprender
 - Orientado a objetos
 - Multiplataforma
- PHP:
 - Es eficaz en la interacción con bases de datos
 - Creación de sitios web dinámicos
 - Fácil de usar y aprender
 - Amplia disponibilidad en servidores web

Busca más lenguajes de programación y cita sus características más importantes.

- Rust:
 - Gran seguridad
 - Alto rendimiento
- Swift:
 - Enfocado en el desarrollo de aplicaciones móviles
 - Tiene una sintaxis clara y moderna
 - Admite programación funcional y orientada a objetos
- C++:
 - Alto rendimiento
 - Orientado a objetos
 - Utilizado en aplicaciones de sistemas y desarrollo de juegos
- R:
 - Enfocado en el análisis de datos y estadísticas
 - Tiene una comunidad muy activa
 - Ofrece capacidad de generar gráficos y visualizar datos

Busca algún gráfico que muestre el porcentaje de los lenguajes de programación más usados en el mundo. ¿Cuál es el más usado en la actualidad?



Como podemos apreciar en la imagen, el lenguaje más usado actualmente en la actualidad es Python, seguido de JavaScript y Java.

EJERCICIOS UT1_5

ACTIVIDAD 5.1

Elabora una lista que incluya al menos 3 lenguajes de programación interpretados, junto con al menos otros 3 que sean compilados, generando código objeto.

Lenguajes interpretados:

- Python
- JavaScript
- Ruby

Lenguajes compilados:

- C
- C++
- Rust

ACTIVIDAD 5.2

Cita las ventajas y desventajas:

Lenguajes compilados:

Ventajas:

- Son más rápidos que los interpretados
- Aplican optimizaciones de código a nivel máquina
- Más portabilidad
- Mayor privacidad del código fuente

Desventajas:

- El proceso de compilación puede ser lento
- Dificultad de depuración
- Menos flexible

Lenguajes interpretados

Ventajas:

- Facilidad de desarrollo
- Facilidad de depuración
- Portable

Desventajas:

- Suelen ser más lentos que los compilados
- Están menos optimizados
- Menor privacidad del código fuente