

EXERCISES QUE TRABAJAREMOS EN LA CUE

- EXERCISE 1: FUNDAMENTOS DE DESARROLLO WEB
- EXERCISE 2: LENGUAJE DE PROGRAMACIÓN
- EXERCISE 3: ACCESO A DATOS
- EXERCISE 4: FRAMEWORKS Y BIBLIOTECAS

EXERCISE 1: FUNDAMENTOS DE DESARROLLO WEB

El desarrollo web es el proceso de creación de contenido en un sitio web, ya sea una parte de una página web o un sitio web completo. Este proceso es similar a cualquier tipo de construcción donde primero se crea la base o estructura y luego se aplica una forma de estilo.

En el desarrollo web, el componente estructural es HTML y el componente de estilo es CSS. HTML es el lenguaje de marcado que define cómo se muestran el texto y las imágenes en una página web. Esto significa que HTML es el componente estructural, que también se utiliza para definir cómo se deben enviar los formularios para crear contenido nuevo o modificar contenido existente.



CSS es el componente de estilo del diseño web, es un lenguaje de hojas de estilo que determina cómo se muestran visualmente elementos como encabezados, párrafos, listas y botones.

Ahora aprendamos sobre estos dos pilares del diseño web con más detalle. "Hipertexto" se refiere a enlaces que conectan páginas web entre sí, ya sea dentro de un solo sitio web o entre sitios web. Los

enlaces son un aspecto fundamental de la Web. Al subir contenido a Internet y vincularlo a páginas creadas por otras personas, se convierte en un participante activo en la World Wide Web.

Un documento HTML se compone de una serie de "etiquetas", que consisten en el nombre del elemento rodeado por "<" y ">". El nombre de un elemento dentro de una etiqueta no distingue entre mayúsculas y minúsculas. Es decir, se puede escribir en mayúsculas, minúsculas o una mezcla. Por ejemplo, la etiqueta `<title>` se puede escribir como `<Title>`, `<TITLE>` o de cualquier otra forma.

Por otro lado, las hojas de estilo en cascada (CSS) es un lenguaje de hoja de estilo que se utiliza para describir cómo se deben representar los elementos en la pantalla.

El uso más común de CSS es diseñar páginas HTML con reglas simples que controlan fuentes, colores y tamaños. Sin embargo, CSS también se puede usar para crear diseños complejos, como tablas o marcos dentro de las páginas web. Puede usar CSS para cambiar la apariencia de sus páginas web y se verán diferentes según los estilos que haya aplicado.

CSS también tiene muchos otros usos, como asegurarse de que todos los navegadores representen imágenes y texto exactamente de la misma manera (por ejemplo, cambiando el color de fondo de una imagen para que coincida con lo que desea).

EXERCISE 2: LENGUAJE DE PROGRAMACIÓN

Los idiomas son los medios por los cuales nos comunicamos unos con otros. Son un conjunto de reglas y convenciones que rigen cómo se pueden organizar las palabras para formar oraciones. Estas reglas y convenciones varían de un idioma a otro, pero todas tienen algunos elementos comunes. Todo esto no solo es cierto para el lenguaje hablado y de señas, sino también para los lenguajes de programación.

Según la Enciclopedia Británica, un lenguaje de programación de computadoras se puede definir como "cualquiera de varios lenguajes para expresar un conjunto de instrucciones detalladas para una computadora digital. Aunque hay muchos lenguajes de computadora, relativamente pocos son ampliamente utilizados".

En este contexto, podemos entender la programación como el proceso de escribir software usando un lenguaje de programación apropiado o una combinación de ellos.

NIVELES DE LENGUAJES DE PROGRAMACIÓN

La enciclopedia mencionada anteriormente, describe los dos niveles de lenguajes de programación de la siguiente manera “los lenguajes de máquina y ensamblador se describen como de ‘bajo nivel’, lo que requiere que un programador administre explícitamente todas las características idiosincrásicas de almacenamiento y operación de datos de una computadora. Por el contrario, los lenguajes de alto nivel evitan que un programador se preocupe por tales consideraciones y proporcionan una notación que los programadores pueden escribir y leer más fácilmente”.

Código de alto nivel

```
class Triangle {  
    ...  
    float surface()  
        return b*h/2;  
}
```

Código de bajo nivel

```
LOAD r1,b  
LOAD r2,h  
MUL r1,r2  
DIV r1,#2  
RET
```

Código de máquina

```
0001001001000101  
0010010011101100  
10101101001...
```

EXERCISE 3: ACCESO A DATOS

¿DÓNDE SE ALMACENAN LOS DATOS EN INTERNET?

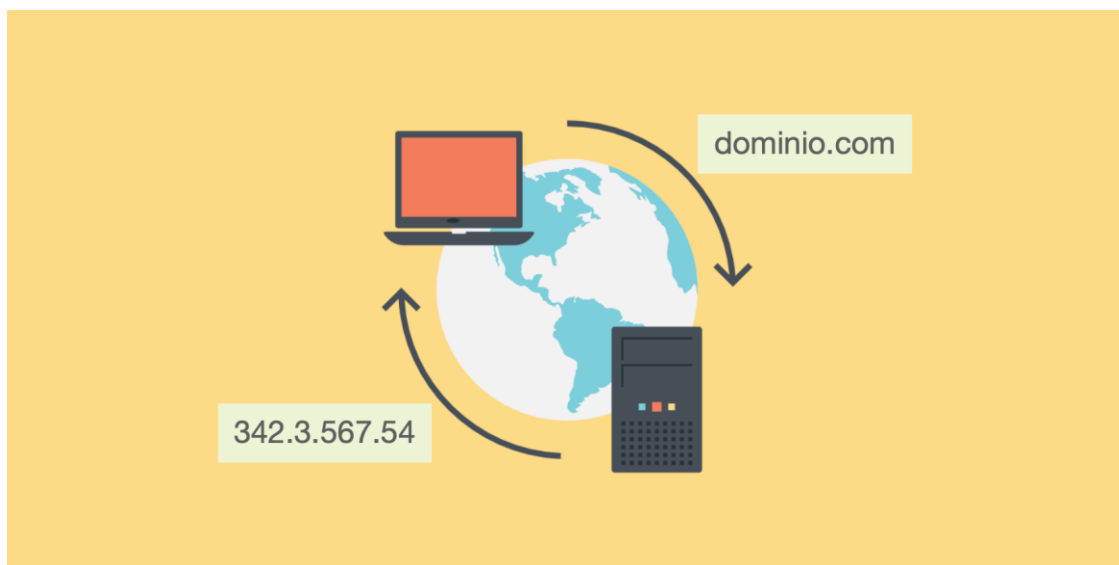
Todos los datos de lo que podemos consumir desde internet están alojados en bases de datos, pero ¿qué son las bases de datos? Según Oracle, una base de datos se puede definir como "una colección organizada de información estructurada, o datos, generalmente almacenados electrónicamente en un sistema informático. Una base de datos generalmente está controlada por un sistema de administración de bases de datos (DBMS). Los datos y el DBMS, junto con las aplicaciones asociadas con ellos, se denominan “sistema de base de datos”, a menudo abreviado como simplemente base de datos.

Los datos almacenados en una base de datos generalmente se modelan en filas y columnas en una serie de tablas. Esto permite que el procesamiento y la consulta de datos sean eficientes. Luego, se puede acceder, administrar, modificar, actualizar, controlar y organizar fácilmente los datos. La mayoría de las bases de datos utilizan lenguaje de consulta estructurado (SQL) para escribir y consultar datos".

SQL es un lenguaje de programación utilizado por casi todas las bases de datos relacionales para consultar, manipular y definir datos y para proporcionar control de acceso. SQL se desarrolló por primera vez en IBM en la década de 1970 con Oracle como uno de los principales contribuyentes, lo que condujo a la implementación del estándar SQL ANSI, SQL ha impulsado muchas extensiones de empresas como IBM, Oracle y Microsoft. Aunque SQL todavía se usa ampliamente en la actualidad, comienzan a aparecer nuevos lenguajes de programación.

¿CÓMO ACCEDEMOS A LOS DATOS WEB?

Todos los sitios web en Internet se almacenan en servidores. Los servidores son computadoras que asignan información. Cuando un usuario intenta ir a un sitio web, su navegador envía una petición a un DNS, que es un servidor de nombres de dominio, este tipo de servidor traduce el nombre de dominio de un sitio web en una dirección IP. Esto se hace porque los sitios web se guardan utilizando las ubicaciones de sus direcciones IP. Por ejemplo, las direcciones IP del DNS público de Google son 2001:4860:4860::8888 y 2001:4860:4860::8844.



Entonces, cuando escribimos "google.com" en nuestros navegadores web, primero se contacta con un servidor DNS, traduciendo el nombre de dominio a una dirección IP que se puede encontrar en un servidor normal. Luego, nuestro navegador contacta con esa dirección IP real y le envía la información que está asignada en el servidor.

EXERCISE 4: FRAMEWORKS Y BIBLIOTECAS

A medida que aprendas más sobre los diferentes lenguajes y metodologías de programación, es muy probable que encuentres conceptos como frameworks (o "*marcos*" en español) y libraries ("*bibliotecas*").

Los desarrolladores suelen utilizar los términos "frameworks" y "bibliotecas" como si fueran lo mismo, pero hay una diferencia. Tanto las bibliotecas como los frameworks son código reutilizable escrito por otra persona que tiene el propósito de ayudarte a resolver problemas comunes de la programación de una manera más fácil.

Podemos explicar la diferencia entre estos conceptos con un ejemplo. Una *biblioteca* es como construir tu casa desde cero, tienes la opción de diseñarla y construirla como quieras. Puedes decidir el formato de los cuartos y la cantidad de baños.

Por otro lado, un *framework* es como comprar una casa nueva, no tienes que lidiar con problemas de construcción, pero no puedes elegir cómo ordenar tus habitaciones porque la casa ya está construida.



LIBRERÍA



FRAMEWORK

FRAMEWORKS

Los frameworks de software generalmente incluyen clases, funciones, objetos y bibliotecas pre-escritas que se pueden usar en un proyecto sin tener que escribir ningún código por cuenta propia. Este tipo de herramientas ayudan a administrar el código de tu aplicación, lo cual es importante porque te brinda la posibilidad de cambiar partes de tu aplicación sin tener que volver a escribir todo.

Ejemplos populares de frameworks variarán según el lenguaje de programación, en el desarrollo web, Angular y Vue son ejemplos importantes. En otro tipo de desarrollo llamado "Backend", tenemos frameworks como Django o Express.js, solo por nombrar algunos. En sí, hay una gran variedad de opciones.

BIBLIOTECAS

Las bibliotecas, por otro lado, son colecciones de código pre-escrito que los usuarios pueden usar para optimizar tareas. Por lo general, las bibliotecas contendrán una serie de clases que tú mismo has escrito o descargado del Internet. Como mencionamos anteriormente, los frameworks generalmente recurren a sus propias bibliotecas para realizar ciertas acciones.

UNA DIFERENCIA TÉCNICA IMPORTANTE

Puede que, al momento de definir ambos conceptos, encontremos similitudes que nos opacan ver las diferencias, pero hay una gran diferencia técnica entre un framework y una biblioteca. Esta diferencia radica en un concepto llamado inversión de control. Cuando usas una biblioteca, quien está a cargo del flujo (*el flujo es la ejecución de las instrucciones*) de la aplicación eres tú, ya que tú eliges cuándo y dónde llamar a la biblioteca. Por otra parte, al usar un framework es este mismo quien está a cargo del flujo del código. Con los frameworks, tú tienes ciertos lugares en donde puedes conectar tu código, pero el framework llamará al código que conectaste según sea necesario.