First Stage Basics

Que es una variable

Una variable es un espacio de memoria reservado para almacenar un valor determinado que corresponde a un tipo de dato soportado por el lenguaje de programación en el cual se trabaja.

¿Qué significa declarar una variable? Pon un ejemplo.

Declarar hay que definirla con un tipo y un identificador (es decir, un nombre para la variable).

En JavaScript En este caso estamos dando valor una variable

Var texto = "juan"

Var num =1;

En Java, en este caso estamos dando valor una variable

Int num =2

String Cliente="Isaac";

```
<script>
function saludo() {
    document.write("Hola, este es el resultado de la función saludo");
}
//llamamos a la función saludo() para que ejecute sus instrucciones
saludo();

</script>
```

¿Qué es un parámetro en una función? Pon un ejemplo.

son los valores que esta recibe por parte del código que la llama.

Al 'x' dentro del paréntesis le llamamos **parámetro** de la función (no es neces ario que se llame 'x', podríamos haberle puesto cualquier otro nombre) y al v alor que le pasamos, en este caso el de la variable 'radio', le llamamos **argu mento**.

```
function volEsfera(x) {
    var volumen = 4 / 3 * 3.14 * x * x * x;
    return volumen;
}
var radio = prompt("Introduce la longitud del radio de la esfera: ");
document.write("El volumen es: " + volEsfera(radio));
```

First Stage Basics

¿Cómo se llama a una función con diferentes parámetros? Pon un ejemplo que permita ejecutar una función, pasando diferentes valores.

Se llama sobre carga la función que tiene más de un parámetro

```
//definimos la función calculoPVP, que tiene como parámetros los valores de precio y descuento
function calculoPVP(precio, descuento) {
    var IVA = 1.21;
    var PVP = (precio - descuento) * IVA; //en la variable PVP hemos almacenado el PVP calculado
    return PVP; //la función devuelve el valor asignado a la variable PVP
}

document.write("Precio= " + calculoPVP(precio, descuento) + " €");
```

No tienen por qué coincidir los nombres de los parámetros y el de los argumentos, como hemos visto pero sí **debe haber el mismo número**. Es decir, si la función tiene 2 parámetros, debemos pasarle 2 argumentos, como vemos en el siguiente

Que es HTML y para se usa

HTML es un lenguaje de marcado de hipertexto o "HyperText Markup Language" por el desarrollo de sus iniciales en inglés, básicamente este lenguaje se escribe en su totalidad con elementos, estos elementos están constituidos por etiquetas, contenido y atributos, que explicaremos de una manera más detallada en algunas líneas más abajo.

Que es HTML 5

Es la última versión de HTML. El término representa dos conceptos diferentes:

- Se trata de una nueva versión de HTML, con nuevos elementos, atributos y comportamientos.
- Contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance. A este conjunto se le llama *HTML5 y amigos*, a menudo reducido a *HTML5*

¿Cuáles son los tags principales que conforman una página HTML? Describe cada uno de ellos.?

<HTML> … </HTML>

First Stage Basics

Delimita y engloba toda la página web, que consta de cabecera y cuerpo.

<HEAD> … </HEAD>

Delimita y engloba la cabecera de una página, que contiene un conjunto de informaciones que no se muestran en la ventana, entre ellas el título de la página, pero que pueden ayudar a los navegadores y a los buscadores para interpretar o a encontrar correctamente la página.

<TITLE> … </TITLE>

Dentro de la cabecera (HEAD), lo que se incluye aquí se muestra en la barra del título de la ventana del navegador.

Metadatos

La cabecera admite otras muchas etiquetas

<BODY> … </BODY>

Delimita y engloba el cuerpo de la página, que son el conjunto de informaciones (texto e imágenes) que se muestran en la página, así como las indicaciones de cómo deben mostrarse.
Admite atributos

Formatos de párrafo

El texto de la página se puede estructurar en encabezamientos de los diferentes apartados de la página, que pueden tener distintos niveles de 1 a 6 (siendo 1 el más importante) y párrafos normales.

<H1> ... </H1> o <H2> ... </H2> (hasta 6)

Párrafos que son encabezamientos (con distintos niveles).

<P>... </P>

Párrafos normales.

<P align="center">... </P>

El atributo align permite alinear el texto del párrafo. Se puede aplicar igual a las etiquetas <H1>, <H2>, etc ...

First Stage Basics

***
***

Permite partir un párrafo empezando una línea nueva pero sin dejar espacio.

<HR>

Pone una línea horizontal de separación. (admite atributos)

<BLOCKQUOTE>…</BLOCKQUOTE>

Sangra el párrafo.

Formatos de texto

El formato de caracteres permite cambiar tanto el tipo de fuente como su tamaño y aspecto.
Se pueden utilizar varias etiquetas HTML para dar distintos formatos a un grupo de caracteres:

Formatos Físicos:

- Negrita: <B>…</B>
- Cursiva: <I>…</I>
- Subrayado: <U>…</U>
- Teletipo: <TT>…</TT>
- Tachado: <STRIKE>…</STRIKE>
- Grande: <BIG>…</BIG>
- Pequeña: <SMALL>…</SMALL>
- Superíndice: […]
- Subíndice: _…

Formatos Lógicos:

- Cita: <CITE>…</CITE>
- Código: <CODE>…</CODE>
- Definición: <DFN>…</DFN>
- Énfasis: <EM>…</EM>
- Grueso: <STRONG>…</STRONG>
- Palabras clave: <KEY>…</KEY>
- Ejemplos: <SAMP>…</SAMP>
- Usuario: <KBD>…</KBD>
- Variables: <VAR>…</VAR>
- Ejemplo literal: <XMP>...</XMP> (ignora las etiquetas HTML de dentro)

First Stage Basics

Posibilidad de combinar etiquetas (anidándolas, esto es, una dentro de otra):

- <B>…<I>…</I>…</B> (Correcto)
- <B>…<I>…</B>…</I> (Incorrecto)

<FONT COLOR="red"> ... </FONT>

Indicación expresa del tipo de letra a usar, en este caso el color

<FONT SIZE="+1"> ... </FONT>

Indicación expresa del tipo de letra a usar, en este caso el tamaño
La etiqueta FONT permite combinaciones cualesquiera de los atributos COLOR, SIZE y FACE

Listas

<UL> ... </UL>

Lista numerada.

<OL> ... </OL>

Lista no numerada.

<LI> ... </LI>

Elementos de una lista.

Enlaces

Sirven para acceder desde una página a otra página o a otro recurso disponible

texto del enlace

Enlace absoluto a una página
texto del enlace

Enlace relativo a una página
 ...

Marcador (enlace interno) dentro de una página
texto del enlace

Enlace a un marcador de la misma página
texto del enlace

First Stage Basics

Enlace a un marcador de otra página (que puede darse con dirección absoluta o relativa)

texto del enlace

Enlace a otra página (absoluta o relativa, con o sin marcador) que se abra en otra ventana.

Imágenes

<img src="nombre.gif">

Muestra una imagen, que normalmente es de tipo GIF, JPG o PNG

¿Qué formas hay de enviar los datos de un formulario HTML?

En el lado del cliente, un formulario HTML no es más que una manera fácil de usar conveniente para configurar una petición HTTP para enviar datos a un servidor. Esto permite al usuario para proporcionar información a ser entregada en la petición HTTP.

El elemento <form> define cómo se enviarán los datos. Todos sus atributos están diseñados para que pueda configurar la solicitud que se enviará cuando un usuario pulsa un botón de envío. Los dos atributos más importantes son acción y método.

El atributo acción.

Este atributo define el lugar donde los datos se envian. Su valor debe ser una dirección URL válida. Si no se proporciona este atributo, los datos serán enviados a la dirección URL de la página que contiene el formulario.

buenas prácticas de HTML

A través de la declaración del tipo de documento, o **DOCTYPE**

```
<!DOCTYPE html>
```

Cierre de etiquetas

Una de las primeras cosas que se estudian cuando se está aprendiendo **HTML** es que las etiquetas deben cerrarse, y en el mismo orden en el que se abrieron.

```
<div>
   <p>Contenido</p>
```

```
  <p>Contenido</p>
</div>
```

Destacar que en html5 se introducen etiquetas vacias sin que necesite etiqueta de cierre

<area>, <base>,
, <col>, <embed>, <hr>, <img>, <input>, <keygen>, <link>, <menuitem>, <meta>, <param>, <source>, <track>, <wbr>

Usos de miniscula en los nombres de las etiquetas, atributos y valores

```
<section>
  <h1>Web App Design</h1>
  <p>Contenido</p>
<section>
```

Usa el atributo ALT en imágenes

Las imágenes deben incluir siempre dicho atributo, que proporciona, como su nombre indica, un **texto alternativo** a la imagen. Su uso mejora la accesibilidad ya que, por ejemplo, los lectores de pantalla se basan en este atributo para proporcionar un contexto a las imágenes.

Dicho valor debe ser lo suficientemente descriptivo al contenido. Si la imagen no tiene ninguna relevancia (como un icono, por ejemplo), es recomendable seguir incluyendo el atributo, aunque esté vacío.

```
<img src="img/logo.png" alt="Logo - Web App Design">
```

Usa una indentación consistente

Indentar es utilizar espacios a la derecha de cada línea con el objetivo de mejorar la **legibilidad** del código, por lo que es un aspecto importante y que deberías tener siempre en cuenta al programar.

```
<div>
  <h1>Web App Design</h1>
  <ul>
    <li>Elemento 1</li>
    <li>Elemento 2</li>
    <li>Elemento 3</li>
  </ul>
```

```
</div>
```

Separa el contenido de la presentación

No uses **estilos en línea**. HTML es el contenido, y CSS proporciona la presentación visual de dicho contenido. No los mezcles.

Por ello, siempre es recomendable utilizar hojas de estilo **externas**, junto con clases para aplicar estilos según sea necesario.

```
<p class="alerta">Información</p>
```

Usa etiquetas semánticas y evita la Divitis

Se podría definir* Divitis *como esa mala práctica de usar* divs *para organizar todo el contenido de la página. Aunque funciona, empeora la legibilidad y sobrecarga el código.

```
<header></header>
<nav></nav>
<main>
   <article></article>
   <article></article>
</main>
<footer></footer>
```

Omite el valor de atributos booleanos

Según el estándar, y a pesar de lo que pueda parecer, estos atributos no admiten los valores *true* o *false*, se representan incluyendo (*true*) u omitiendo (*false*) su nombre.

```
<input type="checkbox" name="ejemplo" checked>
```

Especifica la codificación de caracteres

especificar correctamente la **codificación de caracteres** nos permitirá la correcta visualización del contenido de nuestra página, incluyendo

tildes y caracteres especiales como *ñ*, indicando al navegador cómo debe interpretarlos.

Para especificarla, debemos añadir la siguiente línea al *header* de la página:

<meta charset="UTF-8"/>

En este caso indicamos como sistema de codificación **UTF-8** al ser el más extendido y usado.

Valida tu código

La última práctica que vamos a mencionar en este artículo es **validar** nuestro código para comprobar que cumple con los estándares. Y qué mejor forma para validarlo que usando la herramienta realizada por la misma organización que establece dichos estándares:

Que es CSS y para qué sirve

es un lenguaje de hojas de estilos creado **para** controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML.

CSS sirve para organizar la presentación y aspecto de una página web.

CSS ·3

Cascading Style Sheets es lo mismo que css pero css 3 agrega más cantidades de atributos

Indica a qué tipo de elementos puedes aplicar propiedades en CSS y cómo se hacen referencia a ellos?

Se puede aplicar como selector universal con asterisco *{}

Selector de tipo p{}

Selector descendente p span{}

Selector de clase primero aplicamos class =”nombre” en html y la aplicamos en css de esta manera .nombre{} estilo que queremos

Selectores de ID lo aplicamos en html id=”name” lo aplicamos en css con una #name

First Stage Basics

Indica 5 frameworks o librerías de CSS y explica su función principal
Bulma
Uikit
Boostrap
Foundation
Meterialize

Buenas prácticas CSS
Organizar la estructura de arriba hacia abajo
/*** generic classes*******/***
styles goes here…
/*** header *******/***
styles goes here…
/*** nav menu *******/***
styles goes here…
/*** main content *******/***
styles goes here…
/*** footer *******/***
Nombra correctamente los selectores
Separar las palabras mediante guiones o mayuscula
Legible el css
Combina elementos
Utilizar selectores descendientes
Utilizar nombres descriptivos

No utilices como nombre de un selector una característica visual

Valida tu css
https://jigsaw.w3.org/css-validator/

Agrega los prefijos de los navegadores en propiedades que no sean estables

Que es http

Es el protocolo de transferencia de hipertexto, el protocolo de comunicación que permite la tranferencia de información world wide web

Se usa en la web

Diferencia entre HTTP y HTTPS

First Stage Basics

Las principales diferencias son:

Las principales diferencias son:

HTTP	HTTPS
La URL empieza con "http://"	La URL empieza con "https://"
No utiliza conexión segura	Utiliza conexión segura
Envía los datos a través del puerto 80	Envía los datos a través del puerto 443
No requiere validación de dominio	Requiere mínimo la validación de dominio y, para algunos certificados, incluso se requiere la validación de documentos legales
Funciona a nivel de aplicación	Funciona a nivel de transporte

Los principales métodos de petición

GET

El método GET solicita una representación de un recurso específico. Las peticiones que usan el método GET sólo deben recuperar datos.

HEAD

El método HEAD pide una respuesta idéntica a la de una petición GET, pero sin el cuerpo de la respuesta.

POST

El método POST se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.

PUT

El modo PUT reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.

DELETE

El método DELETE borra un recurso en específico.

CONNECT

El método CONNECT establece un túnel hacia el servidor identificado por el recurso.

OPTIONS

First Stage Basics

El método OPTIONS es utilizado para describir las opciones de comunicación para el recurso de destino.

TRACE

El método TRACE realiza una prueba de bucle de retorno de mensaje a lo largo de la ruta al recurso de destino.

PATCH

El método PATCH es utilizado para aplicar modificaciones parciales a un recurso.

¿Cuáles son los distintos códigos de respuesta de una respuesta HTTP?

100 continúe

Esta respuesta provisional indica que todo hasta ahora está bien y que el cliente debe continuar con la solicitud o ignorarla si ya está terminada.

101 Switching Protocol

Este código se envía en respuesta a un encabezado de solicitud Upgrade por el cliente e indica que el servidor acepta el cambio de protocolo propuesto por el agente de usuario.

102 Processing (WebDAV)

Este código indica que el servidor ha recibido la solicitud y aún se encuentra procesandola, por lo que no hay respuesta disponible.

JAVASCRIPT

¿Qué es JS? ¿Para qué se usa?

es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web

usado en muchos entornos sin navegador, tales como node.js, Apache CouchDB y Adobe Acrobat.

¿Qué tipo de variables existen? Define cada una de ellas.

```
var iva = 16;       // variable tipo entero

var total = 234.65;  // variable tipo decimal
```

First Stage Basics

```
var mensaje = "Bienvenido a nuestro sitio web"; TIPO DE TEXTO

var días = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado", "Domin
go"]; //arrays
```

Booleano

```
var clienteRegistrado = false;

var ivaIncluido = true;
```

¿Qué diferencia hay entre comparar dos valores con == o con ===?
Los operadores `===` y `!==` son los operadores de comparación *estricta*. Esto significa que si los operandos tienen tipos diferentes, no son iguales. Por ejemplo,

```
1 === "1" // false
1 !== "1"  // true
null === undefined // false
```

Los operadores `==` y `!=` son los operadores de comparación *relajada*. Es decir, si los operandos tienen tipos diferentes, JavaScript trata de convertirlos para que fueran comparables. Por ejemplo,

```
1 == "1" // true
1 != "1" // false
null == undefined // true
```

¿Qué son los callbacks? ¿Para qué se usan? Crea 2 ejemplos que sirvan para entender los callbacks de forma clara.

Cuando nos referimos a una función de callback entendemos que es una fu nción que se pasa a otra función como un parámetro. Esta función enviada por parámetro se ejecuta dentro de la función que la contiene para comple tar algún tipo de acción. De esta forma, podemos realizar ciertas acciones si empre y cuando se ejecute previamente la función primaria.

First Stage Basics

```
function Sumar(a, b, callback) {
    return callback(a + b);
}
//seleccionamos el id para realizar un evento con un click
document.querySelector("#operar").addEventListener('click', function() {
    var a = parseInt(document.querySelector("#a").value),
        b = parseInt(document.querySelector("#b").value);
    //llamamos a la funcion sumar para que realize y recogiendo dos valores y
 dando una respuesta por el alert
    Sumar(a, b, function(r) {
        alert('El resultado es ' + r);
    })
});
function a(name, callback) {
    // Simulamos un tiempo de espera en la ejecución de la función a
    setTimeout(function() {
        console.log("Se ha ejecutado la función a " + name + " Luego se ejecuta
ra la funcion b");
        //Creamos un parametro lo llamamos como funcion
        callback();
    }, 1000);
}

function b() {
    console.log("Se ha ejecutado la función b una vez que se ha ejecutado a"
);
}
//para que se ejecute primero la funcion a tiene  que encerrar a la funcion b
 para ser callback
a('robert', b);
```

¿Qué principales ventajas y desventajas tienen los callbacks?

El callback es importante aquí porque tenemos que esperar a la respuesta del servidor antes de avanzar en el código.

Callbacks son conceptualmente simples. Pasas una función que quieras que se ejecute después.

Callbacks corren donde sea. No requiere de un transpilador o polyfill.

First Stage Basics

- **Composición tosca** — No se componen tan elegantemente como las alternativas que mostraremos después. Las llamadas anidadas pueden llevar a realizar un código con aun más anidaciones dentro — comúnmente llamado *callbak hell*, aunque esta preocupación puede ser mitigada extrayendo código para separar las funciones.

- **Flujo poco intuitivo**— *callbacks* requieren que te muevas dentro del código para comprender el flujo del mismo. Los patrones alternativos que se presentan a continuación proporcionan una experiencia de lectura más lineal.

- **Tosco manejo de errores** —Como se pude ver en el ejemplo anterior, se tiene que pasar el error hacia otro función en vez de usar el tradicional try/catch.

¿Qué diferencia hay entre un callback y una promesa?

Callbacks. La llamada asíncrona recibe como parámetro una función. Cuando el resultado esté disponible, invocará a esa función pasándole como parámetro el resultado.

Promises. La llamada asíncrona retorna como resultado "provisional" una Promise. Este es un objeto que contendrá en el futuro el resultado.

Diferencia entre set Interval y set timeout

The setTimeout() method calls a function or evaluates an expression after a specified number of milliseconds.

description

The setInterval() method calls a function or evaluates an expression at specified intervals (in milliseconds).

The setInterval() method will continue calling the function until clearInterval() is called, or the window is closed.

Indica 5 frameworks o librerías de JS y explica su función principal.

Angular:

Empezaremos por uno que quizá ya has usado o escuchado, Angular: este es un framework robusto de código abierto que está desarrollado en typescript; actualmente está siendo mantenido por Google por lo que tendremos Angular para bastante rato.

First Stage Basics

Angular está basado en componentes por lo que busca desarrollar SPA'S o *single page applications*, es decir, páginas que trabajen bajo un solo «html» y donde se van cargando los componentes que vayas necesitando.

React:

Otra librería que no se queda atrás, es React. Nació a raíz de ciertos problemas que se presentaron dentro de Facebook a la hora de mantener los códigos de los anuncios, aquí fue cuando Jordan Walke, Ingeniero en esta misma empresa empezó con el prototipo de React JS con el fin de darle solución final a estos inconvenientes.

React actualmente es mantenido por Facebook y la comunidad de software libre ya que es de código abierto y al igual que Angular, está basado en componentes y busca crear interfaces de usuario que faciliten el uso de las SPA. También intenta ayudar a los desarrolladores a construir aplicaciones que usan datos que cambian todo el tiempo, dentro de los sitios más importantes hechos en React

Vue JS:

Ahora hablaremos de uno de los contendientes más jóvenes de la lista, Vue JS, este es un framework ligero bastante reciente en el mercado pero que está tomando mucha fuerza. Vue JS también está basado en componentes y enfocado a las SPA y se autodenomina como un framework progresivo, es decir que podemos ir utilizando las partes de librería que necesitemos, aunque esto no suene nada nuevo ya que tanto ReactJS como Angular cuentan con una organización parecida en su código base, lo que diferencia a VueJS de otras alternativas, es lo bien desacoplados que se encuentran sus componentes y lo que les permite seguir trabajando bien, aunque se sigan incluyendo más módulos.

EmberJS:

Al igual que Angular, este es un framework bastante robusto y de código abierto. Actualmente es mantenido por Ember core Team, este no está basado en componentes si no que sigue el estándar del MVVM o patrón modelo-vista-modelo que busca desacoplar lo más posible la lógica funcional de una aplicación de su interfaz.

Ember está considerado como un Framework para la web, pero también es posible crear aplicaciones de escritorio y móviles. Un claro ejemplo de una aplicación de escritorio desarrollada en Ember es la popular Apple Music. Ember también es usado por muchos sitios populares como:

First Stage Basics

Meteor:

Este es un framework un poco más completo que los anteriormente mencionados, ya que se abarca desde el backend hasta el frontend. Está escrito en node.js y también es de código abierto, y actualmente es mantenido por Meteor Development Group.

El objetivo de Meteor es mediante JavaScript crear aplicaciones y código multiplataforma (Web, Android y iOS) con el fin de reducir los tiempos de desarrollo. Meteor se integra con MongoDB para el manejo de base de datos, introduce un concepto poco convencional que es el manejo de bases de datos en el cache e introduce un concepto de manejo de datos dentro del cliente sin requerir actualizaciones del servidor.

Del lado del cliente, Meteor se apoya con jQuery y es compatible con cualquier librería JavaScript para la interfaz.

Buenas prácticas JS

No usar taquigrafía

```
if (variableExiste) {
    x = false;
}
llamarOtraFuncion();
```

Utilice === en vez de ==
Usa JSLint es un depurador
Coloque su script al final de la página
Declara las variables fuera de la sentencia FOR
La manera más rápida de construir una cadena

```
var contenedor = document.getElementById('contenedor');
var longitud = algunArray.length;
var contenido = '';
for(var i = 0; i < longitud;  i++) {
   contenido += 'mi numero: ' + i;
   console.log(i);
}
contenedor.innerHtml = contenido;
```

First Stage Basics

La manera más rápida de construir una cadena
Reducir variables globales
Comenta tu código
Usar {} en vez de New Object()
Usar [] en vez de New Array()
Omitir var keyword para declarar muchas variables

Local Storage

nos permite almacenar datos de manera local en el navegador y sin necesidad de realizar alguna conexión a una base de datos

El LocalStorage **suele usarse mucho** en aplicaciones web desarrolladas completamente con JavaScript, con tecnologías como Angular, aunque también puede aplicarse a cualquier web en la cual necesitemos compartir datos entre secciones.

Que es php y para que se usa

es un lenguaje de programación interpretado que **se** utiliza **para** la generación de páginas web de forma dinámica. Éste código **se** ejecuta al lado del servidor y **se** incrusta dentro del código HTML. Cabe destacar que es un lenguaje de código abierto, gratuito y multiplataforma

Define 15 buenas prácticas de PHP.

1. EL MANUAL OFICIAL DE PHP SERÁ TU MEJOR AMIGO, SOBRETODO SUS COMENTARIOS.

2. TENER ACTIVADO ERROR REPORTING SIEMPRE QUE TRABAJEMOS CON NUESTRO CÓDIGO.

UTILIZA ENTORNOS DE DESARROLLO.

NO UTILICES CÓDIGO REDUNDANTE.

SANGRA TU CÓDIGO, LO AGRADECERÁS.

DALE UNA ESTRUCTURA COHERENTE A TU CÓDIGO.

First Stage Basics

DEBEMOS UTILIZAR ESTÁNDARES PARA ENCAPSULAR NUESTRO CÓDIGO.

UTILIZA UNA NOMENCLATURA COMPRENSIBLE.

COMENTA TODO LO QUE PUEDAS.

UTILIZA LÍMITES PARA TUS SCRIPTS.

USA OBJETOS.

RECUERDA BORRAR LOS FICHEROS PHPINFO().

PREVÉ ATAQUES EN TUS FORMULARIOS DE LOGIN.

ENCRIPTA TUS CONTRASEÑAS.

PRUEBA SISTEMAS DE MAPEO DEL TIPO ORM.

Que es una API y para que se usa

- Es una forma de describir la forma en que los programas o los sitios webs intercambian datos.
- El formato de intercambio de datos normalmente es **JSON** o XML.

First Stage Basics

Ofrecer datos a aplicaciones que se ejecutan en un movil

Ofrecer datos a otros desarrolladores con un formato más o menos estándar.

Ofrecer datos a nuestra propia web/aplicación

Consumir datos de otras aplicaciones o sitios Web

El primer método representa una **petición GET** a la ruta diseases. La respuesta será un listado de enfermedades. Y esta respuesta se va a procesar gracias a la clase DiseasesResponse.

El segundo método es una **petición POST** a la ruta upload/photo. Esta petición se hace enviando ciertos parámetros. Entre ellos, una variable String que representa una imagen codificada en base64. Se asume que la API está lista para subir la foto a través de esta ruta.

Que es un Json

- **Una Cadena** es una secuencia de ceros o más caracteres Unicode.
- **Un Objeto** es una colección desordenada de cero o más pares **nombre:valor**, donde un **nombre** es una cadena y un **valor** es una cadena, número, booleano, nulo, objeto o arreglo.
- **Un Arreglo** es una secuencia desordenada de ceros o más valores.

Que diferencia hay entre Json y Jsonp

```
//JSON
{"name":"stackoverflow","id":5}
//JSONP
func({"name":"stackoverflow","id":5});
```

JSon es una forma conveniente de transportar datos entre aplicaciones, especialmente cuando el destino es una aplicación de Javascript.
JSONP le permite especificar una function de callback que pasa su object JSON. Esto le permite eludir la misma política de origen y cargar JSON desde un server externo al javascript en su página web.

Que es sql y para que se usa

La programación **SQL** permite interactuar con una base de datos. El lenguaje de consulta estructurado (**SQL**) es el lenguaje de base de datos más implementado y valioso **para** cualquier persona involucrada en la

programación informática o que **usa** bases de datos **para** recopilar y organizar información.

Que es crud

En informática, **CRUD** es el acrónimo de "Crear, Leer, Actualizar y Borrar" (del original en inglés: Create, Read, Update and Delete), **que** se usa para referirse a las funciones básicas en bases de datos o la capa de persistencia en un software.

Cuales son las principales sentencia SQL

Guarda la definición de todos los objetos almacenados en la base de datos; sus características, restricciones, privilegios, relaciones entre ellos, etc.
Las **sentencias SQL** pertenecen a dos categorías **principales**: Lenguaje de Definición de Datos, DDL y Lenguaje de Manipulación de Datos, DML.

Que es poo y para que se usa

Es un paradigma de programación que **usa** objetos y sus interacciones, **para** diseñar aplicaciones y programas informáticos. ... En la actualidad, existe variedad de lenguajes de programación que soportan la orientación a objetos.

¿Qué es una clase?

Una **clase** es la descripción de un conjunto de objetos similares; consta de métodos y de datos **que** resumen las características comunes de dicho conjunto

Que es un objeto

un objeto es una unidad dentro de un programa de computadores que consta de un estado y de un comportamiento, que a su vez constan respectivamente de datos almacenados y de tareas realizables durante el tiempo de ejecución.

¿Qué es un atributo?

Los **atributos** son las caracterísiticas individuales **que** diferencian un objeto de otro y determinan su apariencia, estado u otras cualidades

First Stage Basics

¿Qué es localStorage? ¿Para qué se usa?

El principal objetivo de Angular es aumentar las aplicaciones web basadas en el modelo vista controlador con el fin de hacer que el desarrollo y las pruebas sobre estos sean más sencillos.