

# Introducción a Javascript

## Módulo 01

# Aprender a programar

Podemos definir un programa como un conjunto de instrucciones que ejecuta un procesador de computadora. Todo programa tendrá un conjunto finito de instrucciones, las cuales se van ejecutando **1 a 1 en cadena hasta finalizar la ejecución**.

Para empezar a **trabajar con un programa es importante**, entender que un programa es un plan que se va a llevar a cabo.

# PseudoCódigo

El **pseudocódigo omite detalles** que quizás se vayan a trabajar con el lenguaje de programación elegido, pero porque estas no son esenciales para que comprendamos en sí de qué se trata y cuál es el fin del mismo.

Tiene la intención de planificar correctamente **aquello que llevarás a cabo en el lenguaje de programación que decidas implementar.**

**Se basa en convenciones de un lenguaje de programación cualquiera** pero lo hace entendible para las personas, y aparte lo realiza **independientemente de cualquier lenguaje específico de programación.**

```
algoritmo Sumar  
  
variables  
    entero a, b, c  
  
inicio  
    escribir( "Introduzca el primer número (entero): " )  
    leer( a )  
    escribir( "Introduzca el segundo número (entero): " )  
    leer( b )  
    c ← a + b  
    escribir( "La suma es: ", c )  
fin
```

# Aplicación: Cómo reconocerla?

Una aplicación es un **programa específico, que resuelve un problema concreto**. A menudo hablamos de aplicaciones contables, aplicaciones de gestión de **RR.HH**, **aplicaciones de liquidación de sueldos, etc.**

Una de las características principales de las aplicaciones es la interacción directa con el usuario.

Un sistema está formado **por un conjunto de programas, involucra también herramientas hardware (partes físicas: monitores, teclados, impresoras, etc), redes de comunicación, bases de datos, servidores.**



# Tipos de Aplicaciones: Aplicaciones de Escritorio

**Aplicaciones de escritorio** son aquellas que típicamente corren en un **sistema Windows**; las cuales pueden ser abiertas yendo a la lista de programas instalados en el sistema operativo. Dichas aplicaciones trabajan con ventanas, tienen un menú en la parte superior (**Con opciones tales como: archivos, herramientas, configuración, etc**).

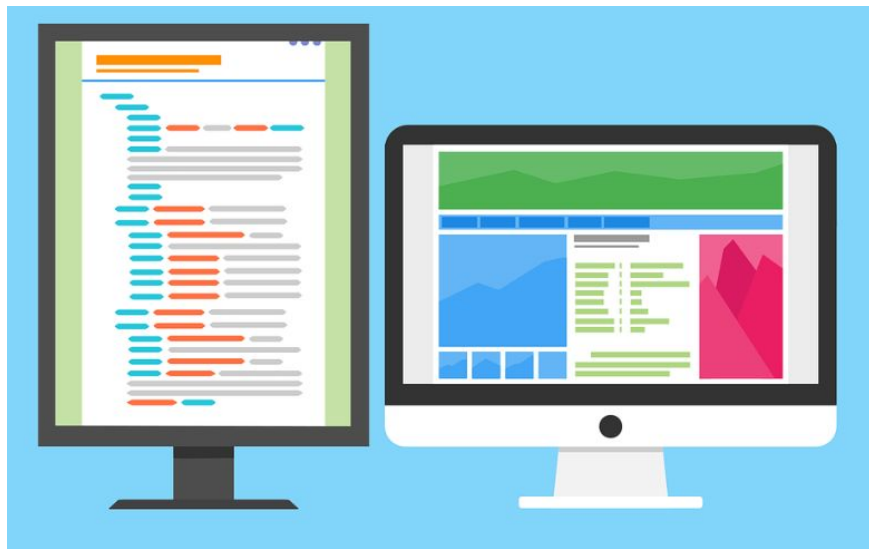
Estas aplicaciones permiten ingresar datos, obtener reportes de datos, etc. Existe mucha interacción con el teclado y el mouse de la computadora.

El botón **secundario del mouse** nos suele generar el conocido menú contextual, muy útil ya que representa un atajo para la ejecución de una funcionalidad específica.

Podemos decir que las aplicaciones de escritorio son las “**aplicaciones tradicionales**” de interfaz gráfica.

# Tipos de Aplicaciones: Aplicaciones de Consola

**Aplicaciones de Consola** son aquellas aplicaciones que utilizan una ventana de **MS-DOS** como **salida**. Quizás el definirlo de esta manera no te ayude demasiado a entender de qué se trata, pero básicamente **Visual Basic.Net** y **C#** quienes utilizan este tipo de consola para poder programar y crear aplicaciones de escritorio.



# Tipos de Aplicaciones:Aplicaciones Web

**Aplicaciones Web** son aquellas que son accedidas desde un browser (Internet Explorer, Firefox, Chrome, etc) a través de alguna dirección **web o url**.

El lenguaje web ha invadido **diferentes espacios**, y por esa razón no necesariamente esto es o puede ser un **sitio web**, sino qué empresas pueden requerir aplicaciones web para manejar cuestiones internas por el mero hecho de la facilidad que estas permiten de **acceder desde cualquier lugar mientras haya conexión a internet**.



# Tipos de Aplicaciones: Aplicaciones Mobile

**Aplicaciones Mobile** son aquellas que funcionan sobre dispositivos mobile (tablets, celulares, etc). Se trata nada más y nada menos de las famosas "apps". Corren en sistema operativos móviles como **Android, iOS, BlackBerry OS, Windows Phone, etc.**

Resuelven de forma más simple gestiones y operaciones del usuario. Por ejemplo la app de **mercado libre, instagram, uber, rappi, cabify** seguramente son aquellas que más utilices en tu teléfono o dispositivo móvil.





# Lenguajes de Programación

Se trata de un lenguaje formal, con **reglas estrictas de escritura**, el cual permite comunicarle a una computadora que es lo que debe hacer con absoluto detalle.

**Todo lenguaje de programación se conforma por un conjunto de símbolos, signos de puntuación, operadores, valores, palabras clave e identificadores** que permiten escribir las instrucciones a ejecutar. A través de los lenguajes de programación podemos **crear nuestros programas**.

**Existen docenas y docenas de lenguajes de programación** hoy día, muchos con similitudes entre sí, como también así con sus diferencias, pero lo más importante es entender como Javascript y todo lo que este lenguaje de programación hoy deriva es el centro del universo tecnológico.

Es impensado casi en todos los ámbitos que un programador aunque este sea o maneje otro lenguaje específicamente no sepa Javascript, y vamos a contarte por qué.

# Javascript como Lenguaje de Programación

**Javascript**, es el centro de todo porque es un lenguaje de programación interpretado por el propio navegador (**Chrome, Firefox, Opera, IE, etc**), sin necesidad de absolutamente nada más.

**La web domina el mundo de la tecnología**, desde la creación de interfaces para fábricas de autos, cajeros automáticos, o simplemente aplicaciones para lograr que los empleados puedan desde cualquier lugar donde hay conexión a internet resolver cualquier problema laboral o trabajar sin necesidad de movilizarse, lo cual reduce costos y mejora el rendimiento.

Entre las tantas cosas qué podemos hacer con Javascript están,

- **Abrir ventanas**
- **Mostrar mensajes**
- **Validar datos en un formulario**
- **Hacer una galería de imágenes**

# Implementar JS: Manera Interna

Lo haremos a través de la **etiqueta script**, simplemente ubicamos la misma tanto en el head como en el body de cualquier documento html y comenzaremos a trabajar con **nuestro código**.

```
<script>  
alert('hola soy js')  
</script>
```

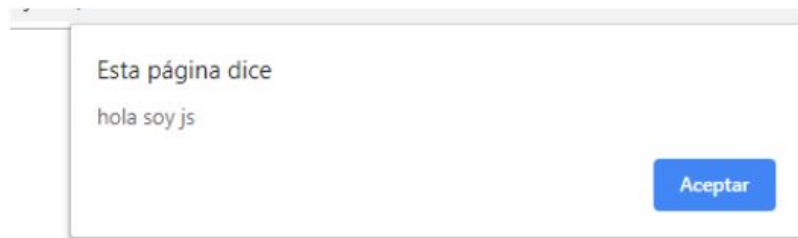
Cuándo trabajamos con elementos que están en **nuestro html** siempre esto debe estar debajo de estos elementos y puede encontrarse **dentro del mismo body**

# Implementar JS: En línea

La idea es hacerlo dentro de las propias etiquetas de **HTML**, por ejemplo en el ejemplo siguiente la misma alerta anterior se dispara al momento de **levantar la página en el navegador**.

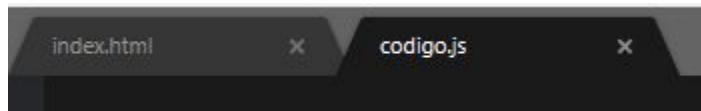
```
<body onload="alert('hola soy js')">
```

Lo malo de trabajar de esta forma es que solo **afecta al HTML** en el cual se encuentra y **es muy engorroso si tenemos un sitio web al momento de hacer cualquier cambio o en el usual mantenimiento del mismo** .

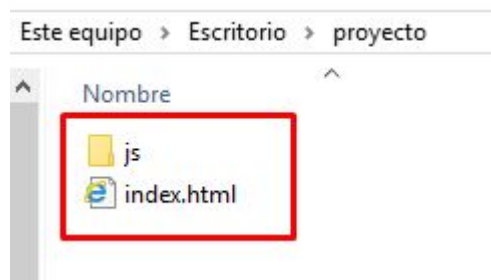


# Implementar JS: Externa

Esta forma, es la ideal pues trabajaremos con un archivo **externo de .js** , por ejemplo guardaremos los siguientes elementos para trabajar,



Para , lograr un proceso **más ordenado te recomendamos generar una carpeta js**, como en la siguiente imagen:



# Implementar JS: Externa

Dentro de este **archivo.js**, lo que haremos será empezar a trabajar con nuestro lenguaje de programación. Pero para poder hacerlo debemos vincularlo con tu **HTML**,

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Desarrollo</title>
  <script src="js/codigo.js"> </script>
</head>

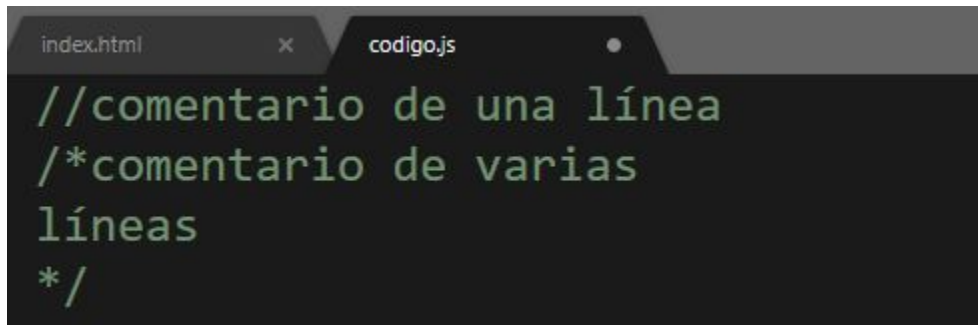
<body>

</body>
</html>
```

# Sintaxis básica de JS

**Javascript** se maneja a través de sentencias que le dicen al navegador (**Chrome, Firefox, etc**) qué hacer. En base a eso, es interesante también saber que tenemos la posibilidad de **hacer comentarios**.

**Estos no afecta a nuestro programa** y nos permiten hacer que un código deje de ejecutarse o dejar mensajes que facilitan el desarrollo o la comunicación entre quienes estamos trabajando en una **determinada aplicación**

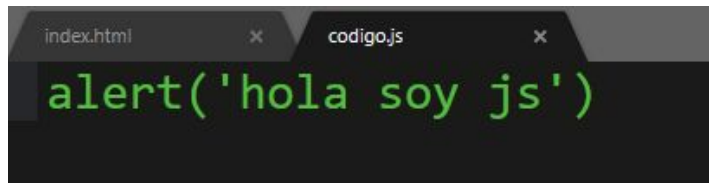
A screenshot of a code editor with two tabs: 'index.html' and 'codigo.js'. The 'codigo.js' tab is active, showing three lines of JavaScript comments in a light green font on a dark background. The first line is a single-line comment starting with '//' and the second two lines are a multi-line comment starting with '/\*' and ending with '\*/'.

```
//comentario de una línea
/*comentario de varias
líneas
*/
```

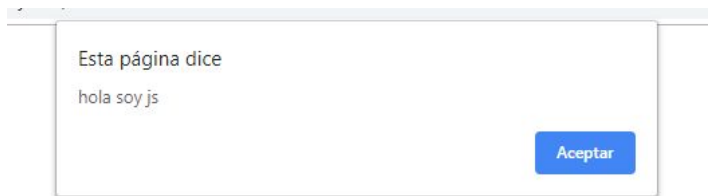
# Primer Script: Ventana de Alerta

Esta ventana alerta al usuario sobre diferentes situaciones, si bien hoy en día las ventanas de alertas fueron reemplazadas en su gran mayoría por **ventanas modales más vistosas**.

La realidad es que no ayudarán a aprender la **sintaxis de JS** y son la base de todo lo que aprenderás más adelante. En tu **codigo.js** escribirás lo siguiente,

A screenshot of a web browser's developer console or a code editor. It shows two tabs: 'index.html' and 'codigo.js'. The 'codigo.js' tab is active, displaying the JavaScript code `alert('hola soy js')` in green text on a dark background.

El resultado será el siguiente,

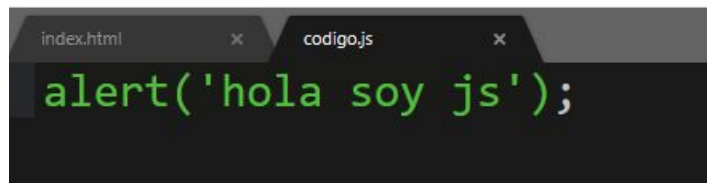




# Primer Script: Ventana de Alerta

Si bien no es obligatorio se recomienda el uso de ; (punto y coma) al final de cada sentencia para poder así **evitar errores y separarla de las otras**.

En otros lenguajes de programación como **PHP**, estos separados son obligatorios, pero en este caso no es necesario, de todas formas para realizar un código **más ordenado y prolijo terminaremos nuestra sentencia así**,

A screenshot of a code editor with two tabs: 'index.html' and 'codigo.js'. The 'codigo.js' tab is active, showing the JavaScript code `alert('hola soy js');` in green text on a dark background.

```
index.html x  codigo.js x  
alert('hola soy js');
```

# Revisión

- Repase los conceptos básicos de un **lenguaje de programación**.
- Trabaje con **pseudocódigos** para empezar a adaptarse a esta lógica.
- Implemente una **ventana de alerta en línea, de forma interna y externa**.
- Aplique todas las propiedades en el **proyecto integrador**
- Realice las preguntas necesarias al/la docente antes de continuar



# ¡Muchas gracias!

¡Sigamos trabajando!