



INFORMÁTICA

SERGIO FERREIRA  
DOCENTE

# Bienvenidos

## Tercer Periodo 2.020

### Clase de Informática

FECHA: OCTUBRE 26 a 30

GRADO: UNDECIMOS

SESION III P



**Goyavier**  
colegio campestre  
*Manifestación del alma en el arte y la ciencia*

## PRESENTACIÓN DE LA CLASE

**Tema:** Cuaderno de Apuntes virtual

**Trabajo a realizar:** Los estudiantes durante la clase actualizarán el archivo de su cuaderno virtual.

**Nota a calificar:**

Promedio – Revisión cuaderno de apuntes virtual.



PROFE, QUE HAY QUE HACER

## Actividad en clase


Revisar las guías trabajadas hasta el momento, junto con sus respectivas actividades.

A partir de las instrucciones dadas por el profesor, actualizar el cuaderno de apuntes virtual, publicarlo y dar a conocer la dirección URL para su revisión.

Para conocer los temas de las guías trabajadas hasta el momento, los invito a observar el contenido de esta presentación...



# Tema 1: Introducción al lenguaje de programación JavaScript

	NOMBRE: _____	FECHA: _____
	PROFESOR: SERGIO FERRERA GÓMEZ	ASIGNATURA: INFORMÁTICA
	ÁREA: TECNOLOGÍA E INFORMÁTICA	

**Indicadores de Desempeño:**  
Reconoce el concepto, la estructura y el tipo de variables de JavaScript a través de la creación y el diseño de páginas web.

## TERCER PERIODO GUÍA 1 INTRODUCCIÓN AL LENGUAJE DE PROGRAMACIÓN JAVASCRIPT

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos en el texto, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

JavaScript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con JavaScript podemos crear diferentes efectos e interactuar con nuestros usuarios.



### ¿Cómo nace JavaScript?

JavaScript nació con la necesidad de permitir a los autores de sitio web crear páginas que permitan intercambiar con los usuarios, ya que se necesitaba crear webs de mayor complejidad. El HTML solo permitía crear páginas estáticas donde se podía mostrar textos con estilos, pero se necesitaba interactuar con los usuarios.

En los años de 1990, Netscape creó LiveScript; las primeras versiones de este lenguaje fueron principalmente dedicadas a pequeños grupos de diseñadores Web que no necesitaban utilizar un compilador, o sin ninguna experiencia en la programación orientada a objetos.

A medida que estuvieron disponibles nuevas versiones de este lenguaje incluían nuevos componentes que dan gran potencial al lenguaje, pero lamentablemente esta versión solo funcionaba en la última versión del Navigator en aquel momento.

En diciembre de 1995, Netscape y Sun Microsystems (el creador del lenguaje Java) luego de unirse con el objetivo de desarrollar el proyecto en conjunto, reintroducen este lenguaje con el nombre de JavaScript. En respuesta a la popularidad de JavaScript, Microsoft lanzó su propio lenguaje de programación a base de script, VBScript (una pequeña versión de Visual Basic).

A pesar de las diferentes críticas que se le hacen al lenguaje JavaScript, este es uno de los lenguajes de programación más populares para la web.

### ¿Dónde puedo ver funcionando JavaScript?


- Entre los diferentes servicios que se encuentran realizados con JavaScript en Internet se encuentran:
- Correo
  - Chat
  - Buscadores de Información
  - Reloj
  - Contadores de visitas
  - Fechas
  - Calculadoras
  - Validadores de formularios
  - Detectores de navegadores e idiomas

ACTIVIDAD:  
Elaboración de un mapa conceptual acerca de JavaScript o cuestionario.

Cantidad de guías (2)



# Tema 2: Operadores lógicos relacionales, variables y tipos de datos.

	NOMBRE:	
	PROFESOR: SERGIO FERREIRA GOMEZ	TÉCNICO
	AREA: TECNOLOGÍA E INFORMÁTICA	ASIGNATURA: INFORMÁTICA
Indicadores de Desempeño: * Reconoce el concepto, la estructura y el tipo de variables de JavaScript a través de la creación y el diseño de páginas web.		

## TERCER PERIODO GUÍA 2 OPERADORES LOGICOS-RELACIONALES, VARIABLES Y TIPOS DE DATOS

JavaScript es un sencillo lenguaje de programación, creado y diseñado para utilizarse en las páginas web. Permite añadirle muchas funcionalidades, que las hacen más dinámicas, interactivas, útiles y atractivas.

JavaScript no es más que líneas de texto, que pueden ser escritas con cualquier editor, como el sencillo Bloc de notas de Windows. Este código es insertado dentro del código HTML o código fuente de las páginas web, encerrado entre etiquetas que le indican al navegador que son instrucciones que debe ejecutar.

### ¿Dónde y cómo insertar el código JavaScript en las páginas web?

Hay varias formas de insertar los códigos JavaScript en las páginas web. Podemos agregarlo en el área del HEAD de la página o encabezado y/o en el área del BODY o cuerpo.

### ¿Cuál es la diferencia?

Las páginas web se dividen en dos áreas, el HEAD y el BODY. El navegador al abrir una página lee primero todas las instrucciones escritas en el área del HEAD, nada escrito aquí se visualiza. Después muestra el contenido en el área del BODY, texto, imágenes y scripts.

### Estructura de una página y ubicación de los scripts

```
<html>
<head>
  <script>
  </script>
</head>
<body>
  <img alt="Logo" data-bbox="125 675 155 705"/>
  <script>
  </script>
  <script src="SCRIPT-EXTERNO.js"></script>
  <script async src="SCRIPT-EXTERNO.js"></script>
</body>
</html>
```

ÁREA DEL ENCABEZADO

CONTENIDO DE LA PÁGINA


orfpj.com

## ACTIVIDAD:

Desarrollo de los cinco ejercicios prácticos, dos capturas de pantalla por cada ejercicio (código + navegador).

Cantidad de guías (4)

# Tema 3: Estructuras de datos: Condicionales simples y compuestos

	AREA: TECNOLOGÍA E INFORMÁTICA	ASIGNATURA: INFORMÁTICA
<b>Indicadores de Desempeño:</b> <ul style="list-style-type: none"><li>✓ Reconoce el concepto, la estructura y el tipo de variables de JavaScript a través de la creación y el diseño de páginas web.</li><li>✓ Crea estructuras secuenciales, condicionales y switches usando operadores lógicos a través de los cuales se dinamizarán las páginas Web creadas y rediseñadas.</li></ul>		

## TERCER PERIODO GUÍA 3 ESTRUCTURAS DE DATOS: CONDICIONALES SIMPLES Y COMPUESTOS

**Condicionales Simples.** Un script consiste en una lista de enunciados que se van ejecutando a medida que se cargan. Sin embargo, en ocasiones nos interesa controlar el flujo de la ejecución estableciendo alternativas, es decir, que una serie de enunciados se ejecuten en algunas ocasiones y en otras no. Para permitir esto existen las estructuras condicionales.

if ... else

Por medio de **if** se puede indicar una condición que, de cumplirse, permite la ejecución de uno o más enunciados. Por medio de **else** se puede establecer una alternativa, aunque su uso es opcional. La sintaxis de esta estructura es la siguiente:

```
if(condición){
  ...enunciados a ejecutar si se cumple la condición...
} else {
  ...enunciados a ejecutar si NO se cumple la condición...
}
```

Ejemplo 1: Hacer un programa que lea un número e indique si es mayor o menor a 100.

```
<html>
<head>
<title>Ejemplo: Condicional simple</title>
</head>
<body>
<script language="javascript">
var num1;
num1=prompt('Ingrese un número:');
num1=parseInt(num1);
};
if (num1>100){
  document.write('El valor de la variable es mayor que 100');
} else {
  document.write('El valor de la variable es menor o igual que 100');
}
</script>
</body>
</html>
```

Recuerda que las actividades de la rama del verdadero o las del falso, NUNCA se realizan simultáneamente, siempre por separado dependiente del dato ingresado.

## ACTIVIDAD:

Desarrollo de los cuatro ejercicios prácticos, dos capturas de pantalla por cada ejercicio (código + navegador).

Cantidad de guías (4)

# Tema 4: Estructuras de datos: Condicional múltiple SWITCH

	NOMBRE:	GRADO: UNDECIMO
	PROFESOR: SERGIO FERREIRA GÓMEZ	FECHA:
	ÁREA: TECNOLOGÍA E INFORMÁTICA	ASIGNATURA: TECN E INFOR

**Indicadores de Desempeño**

- ✓ Crea estructuras secuenciales, condicionales y switches usando operadores lógicos a través de los cuales se dinamizarán las páginas Web creadas y rediseñadas.

### TERCER PERIODO GUÍA N° 4

#### ESTRUCTURAS DE DATOS: CONDICIONAL MÚLTIPLE SWITCH

La estructura de control SWITCH de JavaScript es utilizada para tomar decisiones en función de distintos estados o valores de una variable. Las estructuras de control son la manera con la que se puede dominar el flujo de los programas, para hacer cosas distintas en función de los estados de las variables.

SWITCH, una estructura un poco más compleja que permite hacer múltiples operaciones dependiendo del estado de una variable. En esta guía veremos que SWITCH nos sirve para tomar decisiones en función de distintos estados de las variables. Esta expresión se utiliza cuando tenemos múltiples posibilidades como resultado de la evaluación de una sentencia.

La estructura SWITCH se incorporó a partir de la versión 1.2 de Javascript (Netscape 4 e Internet Explorer 4). Su sintaxis es la siguiente.

```
switch (expresión) {
  case valor1:
    Sentencias a ejecutar si la expresión tiene como valor a valor 1
    break
  case valor2:
    Sentencias a ejecutar si la expresión tiene como valor a valor 2
    break
  case valor3:
    Sentencias a ejecutar si la expresión tiene como valor a valor 3
    break
  default:
    Sentencias a ejecutar si el valor no es ninguno de los anteriores
}
```

La expresión se evalúa, si vale valor1 se ejecutan las sentencias relacionadas con ese caso. Si la expresión vale valor2 se ejecutan las instrucciones relacionadas con ese valor y así sucesivamente, por tantas opciones como deseemos. Finalmente, para todos los casos no contemplados anteriormente se ejecuta el caso por defecto.

Veamos un ejemplo de uso de esta estructura. Supongamos que queremos indicar que día de la semana es. Si el día es 1 (lunes) mostrar un mensaje indicándolo, si el día es 2 (martes) debemos sacar un mensaje distinto y así sucesivamente para cada día de la semana, menos en el 6 (sábado) y 7 (domingo) que queremos mostrar el mensaje "es fin de semana". Para días mayores que 7 indicaremos que ese día no existe.

```
switch (día_de_la_semana) {
  case 1:
    document.write("Es Lunes")
    break
  case 2:
    document.write("Es Martes")
    break
  case 3:
    document.write("Es Miércoles")
    break
  case 4:
    document.write("Es Jueves")
    break
  case 5:
    document.write("Es viernes")
    break
  default:
    document.write("Es fin de semana")
}
```

## ACTIVIDAD:

Desarrollo de (3) tres ejercicios prácticos, dos capturas de pantalla por cada ejercicio (código + navegador).

Cantidad de guías (2)



# Tema 5: Estructuras repetitivas: While (mientras), For (para) y Do while (hacer mientras)

GOYAVIER  
PROFESOR: SERGIO FERREIRA GOMEZ  
AREA: TECNOLOGIA E INFORMÁTICA ASIGNATURA: INFORMÁTICA

Indicadores de Desempeño:  
Crea estructuras secuenciales, condicionales y switches usando operadores lógicos a través de los cuales se dinamizarán las páginas Web creadas y rediseñadas.

## TERCER PERIODO GUÍA 5 SENTENCIAS REPETITIVAS: WHILE, FOR, DO WHILE.

Las estructuras repetitivas son aquellas que sirven para evitar la molestia de andar repitiendo unas acciones varias veces. Estas sirven para que una acción se ejecute un determinado número de veces, y depende del cumplimiento de una determinada acciones (una condición).

Las estructuras repetitivas permiten ejecutar acciones que pueden descomponerse en otras acciones primitivas, esta estructura debe tener una entrada y una salida en la cual se ejecute la acción un determinado o indeterminado número de veces. Las estructuras repetitivas pueden ser: WHILE (mientras), FOR (para), DO WHILE (hacer mientras).

### WHILE

#### • Condición

Una expresión que se evalúa antes de cada paso del bucle. Si esta condición se evalúa como verdadera, se ejecuta sentencia. Cuando la condición se evalúa como false, la ejecución continúa con la sentencia posterior al bucle while.

#### • Sentencia

Una sentencia que se ejecuta mientras la condición se evalúa como verdadera. Para ejecutar múltiples sentencias dentro de un bucle, utiliza una sentencia block {...} para agrupar esas sentencias.

**Ejemplo:** El siguiente bucle while itera mientras n es menor que tres.

```
1 n = 0;  
2 x = 0;  
3 while (n < 3) {  
4     n ++;  
5     x += n;  
6 }
```

## ACTIVIDAD:

Consultar un ejercicio para cada uno de los ciclos, una captura de pantalla por cada ejercicio (solo código).

Cantidad de guías (3)



