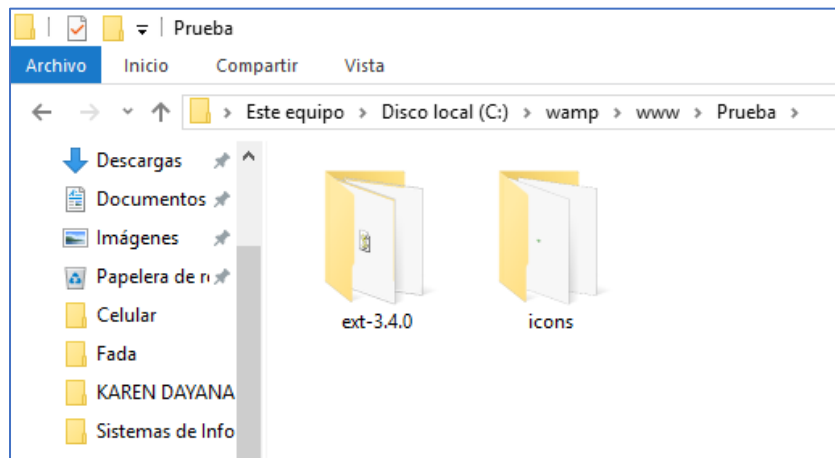
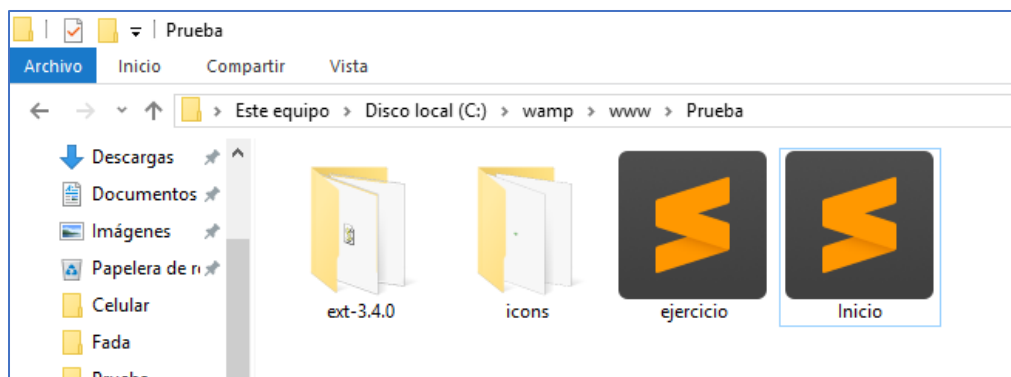


## Iniciando con JAVASCRIPT y EXT JS

1. Debemos adquirir las librerías del framework EXT JS, el cual es un marco de trabajo basado en el lenguaje JAVA SCRIPT y que nos permite crear paneles, ventanas, formularios y gráficos entre otros de manera más fácil, ágil y con efectos propios de esta librería.  
La librería la podemos obtener del Campus universitario en su versión 3.4.0, donde adicional a esta encontraremos una carpeta con iconos de apoyo para la creación de algunas GUI.
2. Luego de obtener los archivos, los ubicaremos en la carpeta raíz del Servidor Local de su elección (preferiblemente WAMP), en una carpeta que llamaremos “Prueba”.



3. Una vez tenemos los archivos en nuestro servidor local, procedemos a crear un archivo con extensión .html (pagina web) la cual llamaremos “Inicio” y que será nuestro lienzo para empotrar los elementos propios de la librería EXT JS. Adicionalmente crearemos un archivo con extensión .js (JAVA SCRIPT) el cual hará uso de las clases y métodos del Framework EXT JS para la creación de elementos (paneles, ventanas, formularios y gráficos entre otros).



4. Ahora en la pagina web “Inicio.html” procedemos a:
  - a- Llamamos los archivos JAVA SCRIPT los cuales poseen las clases y métodos que proporciona el Framework EXT JS para la creación de diversos elementos.

```
<script type="text/javascript" src="ext-3.4.0/adapters/ext-base.js"></script>
```

```
<script type="text/javascript" src="ext-3.4.0/ext-all.js"></script>
```

b- Llamamos los archivos CSS los cuales poseen los estilos que proporciona el Framework EXT JS para la creación de diversos elementos.

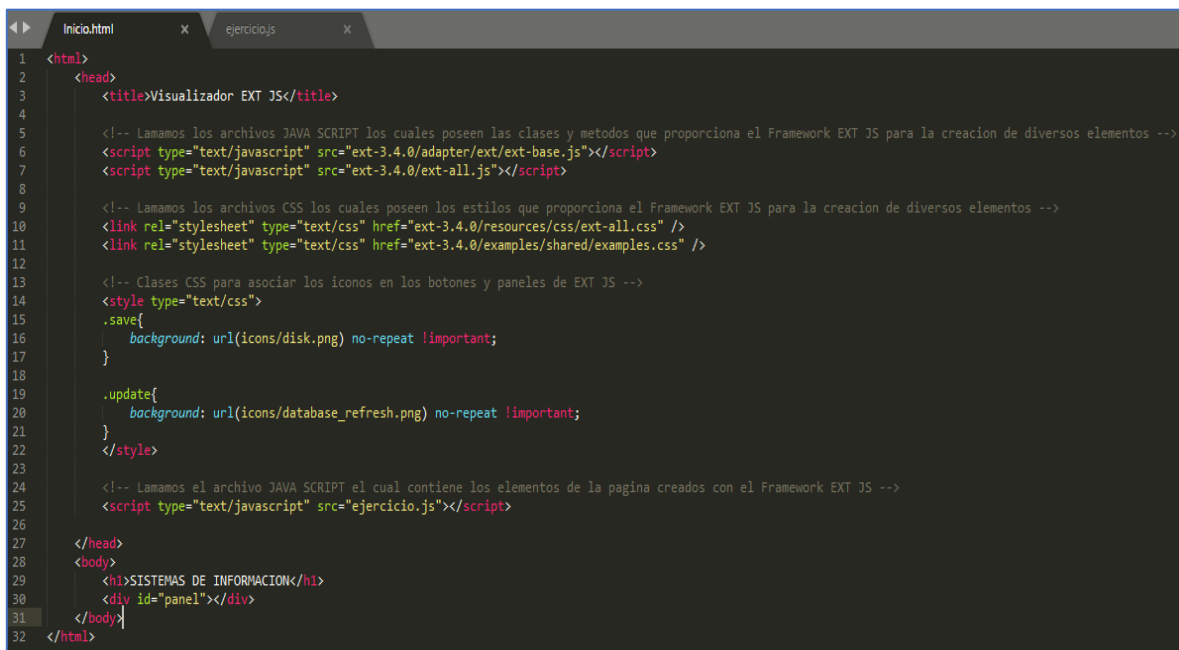
```
<link rel="stylesheet" type="text/css" href="ext-3.4.0/resources/css/ext-all.css" />
```

```
<link rel="stylesheet" type="text/css" href="ext-3.4.0/examples/shared/examples.css" />
```

c- Llamamos el archivo JAVA SCRIPT el cual contiene los elementos de la página creados con el Framework EXT JS.

```
<script type="text/javascript" src="ejercicio.js"></script>
```

Es de aclarar que estas líneas deben estar en una estructura html entre las etiquetas de head.



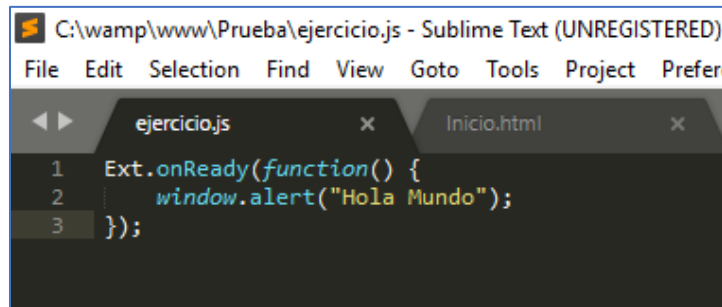
```
1 <html>
2 <head>
3   <title>Visualizador EXT JS</title>
4
5   <!-- Llamamos los archivos JAVA SCRIPT los cuales poseen las clases y metodos que proporciona el Framework EXT JS para la creacion de diversos elementos -->
6   <script type="text/javascript" src="ext-3.4.0/adapters/ext-base.js"></script>
7   <script type="text/javascript" src="ext-3.4.0/ext-all.js"></script>
8
9   <!-- Llamamos los archivos CSS los cuales poseen los estilos que proporciona el Framework EXT JS para la creacion de diversos elementos -->
10  <link rel="stylesheet" type="text/css" href="ext-3.4.0/resources/css/ext-all.css" />
11  <link rel="stylesheet" type="text/css" href="ext-3.4.0/examples/shared/examples.css" />
12
13  <!-- Clases CSS para asociar los iconos en los botones y paneles de EXT JS -->
14  <style type="text/css">
15    .save{
16      background: url(icons/disk.png) no-repeat !important;
17    }
18
19    .update{
20      background: url(icons/database_refresh.png) no-repeat !important;
21    }
22  </style>
23
24  <!-- Llamamos el archivo JAVA SCRIPT el cual contiene los elementos de la pagina creados con el Framework EXT JS -->
25  <script type="text/javascript" src="ejercicio.js"></script>
26
27 </head>
28 <body>
29   <h1>SISTEMAS DE INFORMACION</h1>
30   <div id="panel"></div>
31 </body>
32 </html>
```

5. Ahora podemos iniciar con la creación de (paneles, ventanas, formularios y gráficos entre otros), en el archivo “ejercicio.js” mediante las clases y métodos del Framework “EXT JS” del cual ya importamos o llamamos sus librerías.

6. Pero primero lo primero, como mostrar un mensaje o alerta. Para ello existen dos formas:

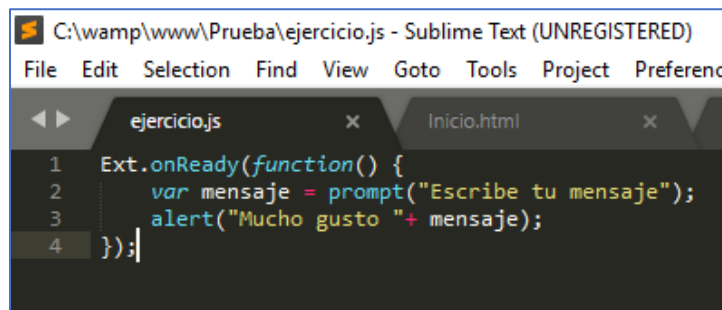
- Una es llamar la sentencia directamente desde el inicio del archivo.  
window.alert("Hola Mundo");
- Otra es llamar la sentencia desde un método principal, para el Framework “EXT JS” este método se llama “Ext.onReady” el cual llama una función la cual es la primera en ejecutarse, algo como el método MAIN en java.  
Ext.onReady(function() {

```
window.alert("Hola Mundo");  
});
```



```
C:\wamp\www\Prueba\ejercicio.js - Sublime Text (UNREGISTERED)  
File Edit Selection Find View Goto Tools Project Preferences  
ejercicio.js x Inicio.html x  
1 Ext.onReady(function() {  
2     window.alert("Hola Mundo");  
3 });
```

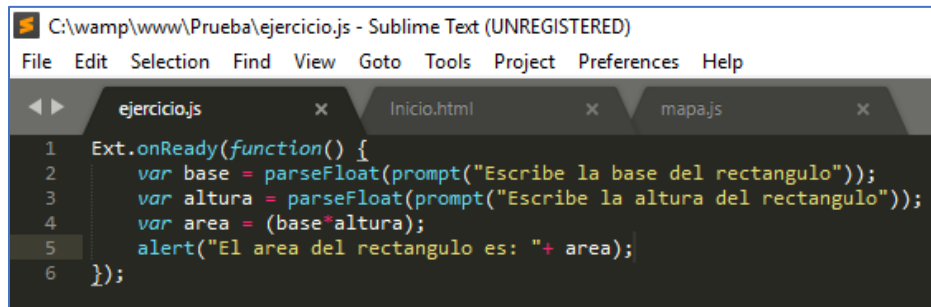
7. Ya sabiendo como mostrar un mensaje o alerta, ahora podemos proseguir en como capturar caracteres ingresados por teclado por parte del usuario.



```
C:\wamp\www\Prueba\ejercicio.js - Sublime Text (UNREGISTERED)  
File Edit Selection Find View Goto Tools Project Preferences  
ejercicio.js x Inicio.html x  
1 Ext.onReady(function() {  
2     var mensaje = prompt("Escribe tu mensaje");  
3     alert("Mucho gusto " + mensaje);  
4 });
```

8. Lo que vimos esta bien si solo se trata de mostrar mensajes sobre cadenas de caracteres, pero si lo que queremos es realizar operaciones y cálculos sobre los datos ingresados por el usuario por medio del prompt. En este caso debemos realizar una conversión explícita de tipos de datos teniendo lo siguiente:
- a. Convertir Cadena en Decimal → `parseFloat(cadena)`  
Toma la "cadena" y la transforma en un número en coma flotante, si es posible.  
`parseFloat("123.456") = 123.456`  
`parseFloat("123ABC") = 123`  
`parseFloat("ABC") = NaN`
  - b. Convertir Cadena en Entero → `parseInt(cadena, número)`  
Devuelve números enteros, el segundo argumento nos permite escoger la base de numeración (entre 2 y 36)  
`parseInt("ABC", 16) = 2748` `ABC16 = 274810`  
Si no especificamos el segundo argumento, por defecto es 10.  
Si la cadena empieza por 0x y no existe el segundo argumento, se entiende que es 16.  
Si la cadena empieza por 0 y no existe el segundo argumento, se entiende que es 8.
  - c. Convertir Números en Cadenas → `toString(argumento)`  
Si argumento = número  
Devuelve una cadena que contiene el número  
Puede haber un argumento opcional:

(13).toString(16) = "d" siendo 1310 = d16  
(13).toString(2) = "1101" siendo 1310 = 11012

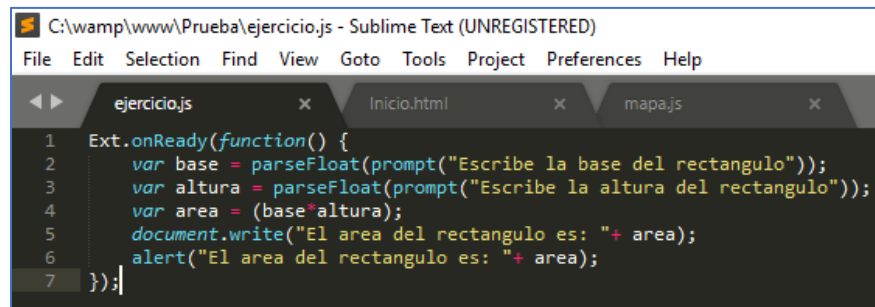


```
C:\wamp\www\Prueba\ejercicio.js - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

ejercicio.js x Inicio.html x mapajs x

1 Ext.onReady(function() {
2   var base = parseFloat(prompt("Escribe la base del rectangulo"));
3   var altura = parseFloat(prompt("Escribe la altura del rectangulo"));
4   var area = (base*altura);
5   alert("El area del rectangulo es: "+ area);
6 });
```

9. JavaScript también nos permite escribir sobre el documento de la pagina web HTML, en caso de no existir la necesidad de mostrar una alerta. Para ello utilizamos la sentencia "document.write(cadena)".



```
C:\wamp\www\Prueba\ejercicio.js - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

ejercicio.js x Inicio.html x mapajs x

1 Ext.onReady(function() {
2   var base = parseFloat(prompt("Escribe la base del rectangulo"));
3   var altura = parseFloat(prompt("Escribe la altura del rectangulo"));
4   var area = (base*altura);
5   document.write("El area del rectangulo es: "+ area);
6   alert("El area del rectangulo es: "+ area);
7 });
```

## 10. Variables Estructuras de Control

JavaScript permite utilizar variables, de tipos numéricos, cadenas de caracteres o booleanos. No obstante, cuando declaramos una variable no se le asigna un tipo, sino que es el propio intérprete el que, según el valor que se le asigne, le da un tipo u otro. La declaración es opcional, de manera que se puede hacer de dos maneras:

- Simplemente asignándole directamente un valor, por ejemplo, `x = 42`
- Utilizando la palabra clave `var`, por ejemplo, `var x = 42`

Por lo demás, la sintaxis del lenguaje es prácticamente igual que la de Java (es decir, igual que la de C). Así, tanto los operadores como las estructuras de control son exactamente iguales. Recordemos las estructuras de control:

### Estructuras condicionales

- La estructura `if ... else ...` es así:

`if (condición)`

`{`

    bloque de instrucciones 1

```
}  
else  
{  
    bloque de instrucciones 1  
}
```

- **La estructura switch es:**

```
switch (expresión)  
{  
    case caso1 :  
        instrucciones;  
        break;  
    case caso2:  
        instrucciones;  
        break;  
    ...  
    default : instrucciones;  
}
```

### **Estructuras iterativas**

- **La estructura while es así:**

```
while (condición)  
{  
    instrucciones  
}
```

- **La estructura do ... while es así:**

```
do {  
    instrucciones  
} while (condición);
```

- **La estructura for es así:**

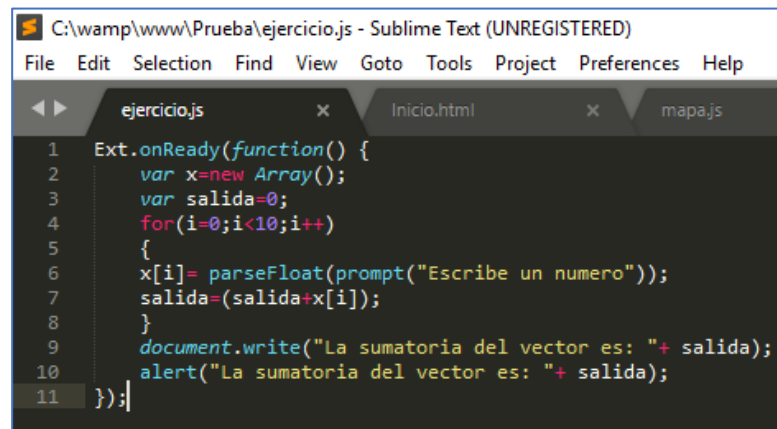
```
for (expresión inicial; condición de cumplimiento; expresión de incremento)
```

```
{  
  
    instrucciones  
  
}
```

## 11. Arrays

Un arreglo o vector es un conjunto de elementos colocados de forma adyacente en la memoria de manera que nos podemos referir a ellos con un solo nombre común. Para crear un vector debemos crear una variable que lo identifique, adicionalmente debemos crear una instancia de la clase “array” utilizando la palabra reservada “new”. `var x=new Array();`

Ya creado el vector debemos manejarlo mediante una estructura iterativa para así asignar valores a cada posición de vector o realizar operaciones con los valores de cada posición de vector, para obtener o asignar un valor a una posición del vector debemos manejar un índice de tipo entero entre paréntesis el cual indexa el vector desde la posición 0 para el primer elemento. Ejemplo: Obtener el valor de la posición 3 del vector X la sentencia sería `“alert(“el valor de la posición 3 del vector X es: “+ x[2]);”`



```
C:\wamp\www\Prueba\ejercicio.js - Sublime Text (UNREGISTERED)  
File Edit Selection Find View Goto Tools Project Preferences Help  
ejercicio.js Inicio.html mapa.js  
1 Ext.onReady(function() {  
2     var x=new Array();  
3     var salida=0;  
4     for(i=0;i<10;i++)  
5     {  
6         x[i]= parseFloat(prompt("Escribe un numero"));  
7         salida=(salida+x[i]);  
8     }  
9     document.write("La sumatoria del vector es: "+ salida);  
10    alert("La sumatoria del vector es: "+ salida);  
11 });
```