



# Tema 10.1

## Javascript: Arrays

# Arrays

- ◆ **Array:** lista ordenada de
  - elementos **heterogéneos**
    - ◆ accesibles a través de un índice
      - de **0** a **length-1**
- ◆ **Tamaño máximo:**  $2^{32}-2$  elementos
- ◆ **Elementos**
  - **a[0]** es el primer elemento
  - .....
  - **a[a.length-1]** último elemento

```
var x = [1, 2, 3];
```

```
a[0]      => 1
```

```
a[1]      => 2
```

```
a[2]      => 3
```

```
a.length  => 3
```

# Elementos de un Array

- ◆ Elementos del array pueden contener cualquier valor u objeto
  - **undefined**
  - otro **array**
  - **objetos**
  - ...
- ◆ Indexar elementos que no existen
  - devuelve **undefined**
    - ◆ por ejemplo con índices mayores que **length**

```
var a = [1, undefined, 'a', , [1, 2]];
```

```
a[3];           => undefined
```

```
a[4];           => [1, 2]
```

```
a[9];           => undefined
```

```
a[4][1];        => 2
```

# Tamaño del Array

- ◆ Los arrays son dinámicos
  - pueden crecer y encoger
- ◆ Asignar un elemento fuera de rango
  - incrementa el tamaño del array
- ◆ El tamaño del array se puede modificar
  - con la propiedad **a.length**
    - ◆ **a.length = 3;**
      - modifica el tamaño del array

```
var a = [1, 3, 1];  
  
a;           => [1, 3, 1]  
  
a[4] = 2;    => [1, 3, 1, , 2]  
  
    // el array se reduce  
a.length = 2 => 2  
  
a           => [1, 3]
```

# Métodos de Array

Array hereda métodos de su clase

- ◆ **sort()**: devuelve array ordenado
- ◆ **reverse()**: devuelve array invertido
- ◆ **push(e1, ..., en)**
  - añade **e1, ...,en** al final del array
- ◆ **pop()**
  - extrae último elemento del array

```
var a = [1, 5, 3];  
  
a.sort()      => [1, 3, 5]  
  
a.reverse()   => [5, 3, 1]  
  
a.push(false) => 4  
a             => [5, 3, 1, false]  
  
a.pop()       => false  
a             => [5, 3, 1]
```

# Ejercicio

- ◆ Indicar el resultado de evaluar las siguientes expresiones

`[3, ,1, ,2, ,0][2]`      => se evalúa a 0, 1, 2, 3, undefined, null

`[3, ,1, ,2, ,0][3]`      => se evalúa a 0, 1, 2, 3, undefined, null

`[3, ,1, ,2, ,0][4]`      => se evalúa a 0, 1, 2, 3, undefined, null

`[3, ,1, ,2, ,0].sort()`      => se evalúa a [0,1,2,3], [3,2,1,0], [0,1,2,3, , , ], [0, ,2, ,1, ,3]

`[3, ,1, ,2, ,0].reverse()`      => se evalúa a [0,1,2,3], [3,2,1,0], [0,1,2,3, , , ], [0, ,2, ,1, ,3]

- ◆ Hacer una hoja Web similar al ejemplo anterior, que incluya
  - un script que muestre como JavaScript evalúa estas expresiones



# Tema 10.2

## Javascript: JSON

# JSON

- ◆ JSON: formato textual de representación de tipos y objetos JavaScript
  - <http://json.org/json-es.html>
- ◆ Un **objeto JavaScript** se transforma a un **string JSON** con
  - **JSON.stringify(object)**
- ◆ Un **string JSON** se transforma en el **objeto original** con
  - **JSON.parse(string\_JSON)**

```
var x = {a:1, b:{y:[false, null, ""]}}, y, z;
```

```
y = JSON.stringify(x);      => '{"a":1, "b":{"y":[false, null, ""]}}'
```

```
z = JSON.parse(y);          => {a:1, b:{y:[false, null, ""]}}
```



# Serialización de datos

- ◆ Serialización:
  - transformación **reversible** de un tipo u objeto (en memoria) en un **string equivalente**
- ◆ La serialización es un formato de intercambio de datos
  - **Almacenar datos** en un fichero
  - **Enviar datos** a través de una línea de comunicación
  - **Paso de parámetros** en interfaces REST
- ◆ En JavaScript se realiza desde ECMAScript 5 con
  - **JSON.stringify(...)** y **JSON.parse(...)**
- ◆ Otros formatos de serialización: XML, HTML, XDR(C), ...
  - XML está siendo desplazados por JSON
    - ◆ Hay bibliotecas de JSON para los lenguajes más importantes

# Características de JSON

## ◆ JSON puede serializar

- objetos, arrays, strings, números finitos, true, false y null
  - ◆ NaN, Infinity y -Infinity se serializan a null
  - ◆ Objetos Date se serializan a formato ISO
    - la reconstrucción devuelve un string y no el objeto original
- No se puede serializar
  - ◆ Funciones, RegExp, errores, undefined

## ◆ Admite filtros para los elementos no soportados

- ver doc de APIs JavaScript

```
JSON.stringify(new Date())    => '"2013-08-08T17:13:10.751Z"'
```

```
JSON.stringify(NaN)          => 'null'
```

```
JSON.stringify(Infinity)     => 'null'
```

# Ejercicio

- ◆ Indicar el resultado de evaluar las siguientes expresiones

```
JSON.stringify({a:6, b:Infinity}); => '{"a":6,"b":Infinity}'  
=> '{"a":6,"b":NaN}'  
=> '{a:6,b:null}'  
=> '{"a":6,"b":null}'
```

```
JSON.stringify({a:"", b:""}) => '{}'  
=> '{"a":"","b":""}'  
=> '{"a":"","b":""}'  
=> '{a:"",b:""}'
```

- ◆ Hacer una hoja Web similar al ejemplo anterior, que incluya
  - un script que muestre como JavaScript evalúa estas expresiones



# Tema 10.3

## Apps FirefoxOS



# FirefoxOS

## ◆ FirefoxOS es el nuevo SO

- para móviles y tabletas
  - ◆ Desarrollado por la Fundación Mozilla

## ◆ Sus aplicaciones se programan en

- HTML5, CSS y Javascript

## ◆ Recursos y tutoriales

- [https://marketplace.firefox.com/developers/docs/firefox\\_os](https://marketplace.firefox.com/developers/docs/firefox_os)
- [https://marketplace.firefox.com/developers/docs/quick\\_start](https://marketplace.firefox.com/developers/docs/quick_start)
- [https://developer.mozilla.org/es/docs/Aplicaciones/Comenzando\\_aplicaciones](https://developer.mozilla.org/es/docs/Aplicaciones/Comenzando_aplicaciones)
- [https://developer.mozilla.org/en-US/docs/Web/Apps/Introduction\\_to\\_open\\_web\\_apps](https://developer.mozilla.org/en-US/docs/Web/Apps/Introduction_to_open_web_apps)
- <https://developer.mozilla.org/en-US/Apps/Reference>



# Apps FirefoxOS



## ◆ Aplicaciones para instalar en móviles y tabletas

- Funcionando con o sin conexión a Internet
  - ◆ Construidas con tecnologías Web: URLs, HTTP, HTML, CSS, JavaScript, ...
- También pueden instalarse en PCs o navegadores

## ◆ Tipos de WebApps Firefox OS

- **hosted:** alojadas en un servidor
- **packaged:** se descargan empaquetadas
  - ◆ Hay tres tipos
    - **Plain:** aplicaciones Web convencionales
    - **Certified:** con acceso a algunos elementos restringidos
    - **Privileged:** con acceso a todos los recursos del móvil

# Hosted Apps



- ◆ Webapps publicadas en modo ejecutable en un servidor Web
  - Pueden ejecutarse desde un navegador
  - Además de ser publicadas en una tienda a través de su URL
    - ◆ para su instalación en un dispositivo FirefoxOS
  - Mas info:
    - ◆ <https://marketplace.firefox.com/developers/docs/hosted>
- ◆ Características
  - Necesitan conexión a Internet para ser ejecutadas
  - Solo puede haber una hosted WebApp por dominio Web
- ◆ Se gestionan a través del objeto predefinido **navigator.mozApps**
  - con métodos **install(...)** o **checkInstalled(...)**



# Packaged WebApps



- ◆ Webapps empaquetadas en un fichero ZIP
  - Pueden ser publicadas en una tienda
    - ◆ para su instalación en dispositivos Firefox OS
  - Mas info:
    - ◆ <https://marketplace.firefox.com/developers/docs/packaged>
    - ◆ [https://developer.mozilla.org/en-US/Apps/Developing/Packaged\\_apps](https://developer.mozilla.org/en-US/Apps/Developing/Packaged_apps)
- ◆ Se pueden publicar en tiendas o en servidores
  - Solamente se pueden instalar
    - ◆ No se pueden ejecutar directamente en un navegador
  - Pueden ser autonomas
    - ◆ Funcionando sin conexión a Internet
- ◆ Se instalan a través del objeto predefinido **navigator.mozApps**
  - con método **installPackage(...)**





# Tiendas de Apps

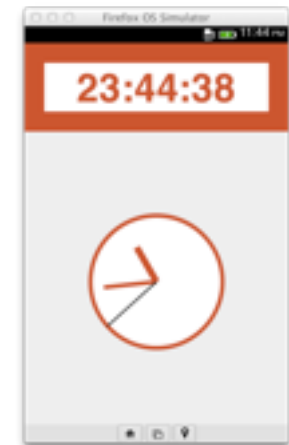
- ◆ Una tienda de aplicaciones para FirefoxOS
  - puede crearse en cualquier servidor que aloje Apps
    - ◆ Como las que hemos desarrollado
- ◆ Las tiendas FirefoxOS compiten entre si
  - serán más o menos populares
    - ◆ dependiendo de la aceptación de los usuarios
  - serán más o menos seguras
    - ◆ dependiendo del control de la seguridad realizado
- ◆ Tienda oficial de la fundación Mozilla
  - <https://marketplace.firefox.com>





# Tema 10.4

## Reloj SVG como App FirefoxOS



# Reloj SVG como App

## ◆ Incluimos el reloj SVG como

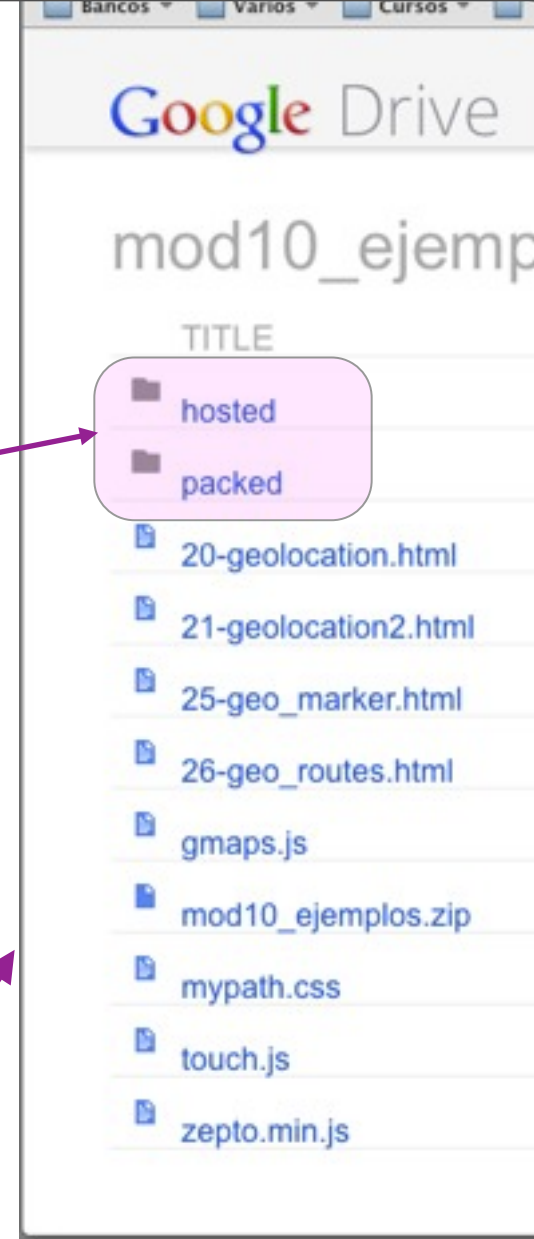
- **hosted App**
- **packaged App**



## ◆ Ambas están publicadas en

- **Dir. Google Drive de asignatura**

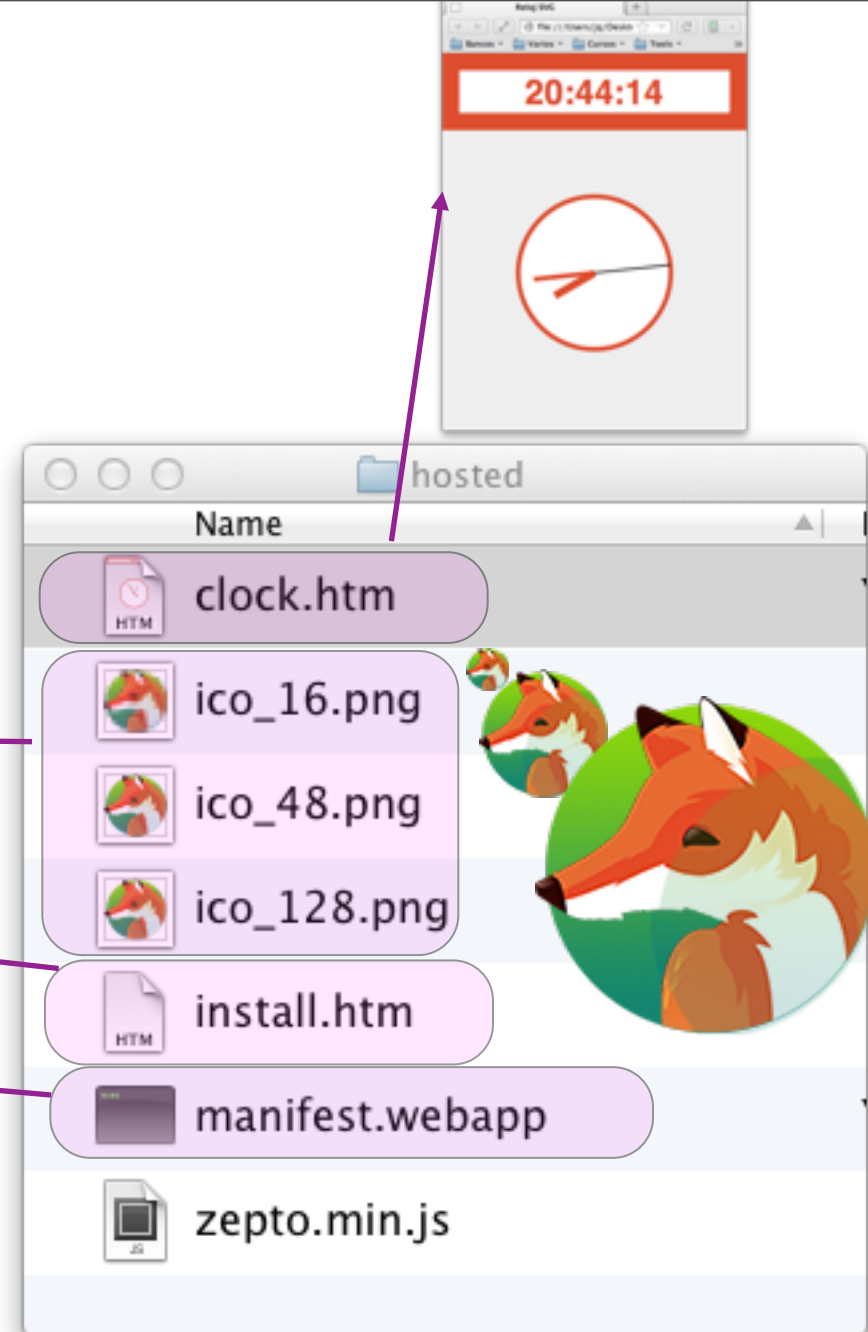
◆ [https://googledrive.com/host/0B48KCWfVwCIEcEdDcEVSTS1MS00/modulo10/mod10\\_ejemplos/](https://googledrive.com/host/0B48KCWfVwCIEcEdDcEVSTS1MS00/modulo10/mod10_ejemplos/)



# Reloj SVG: hosted App

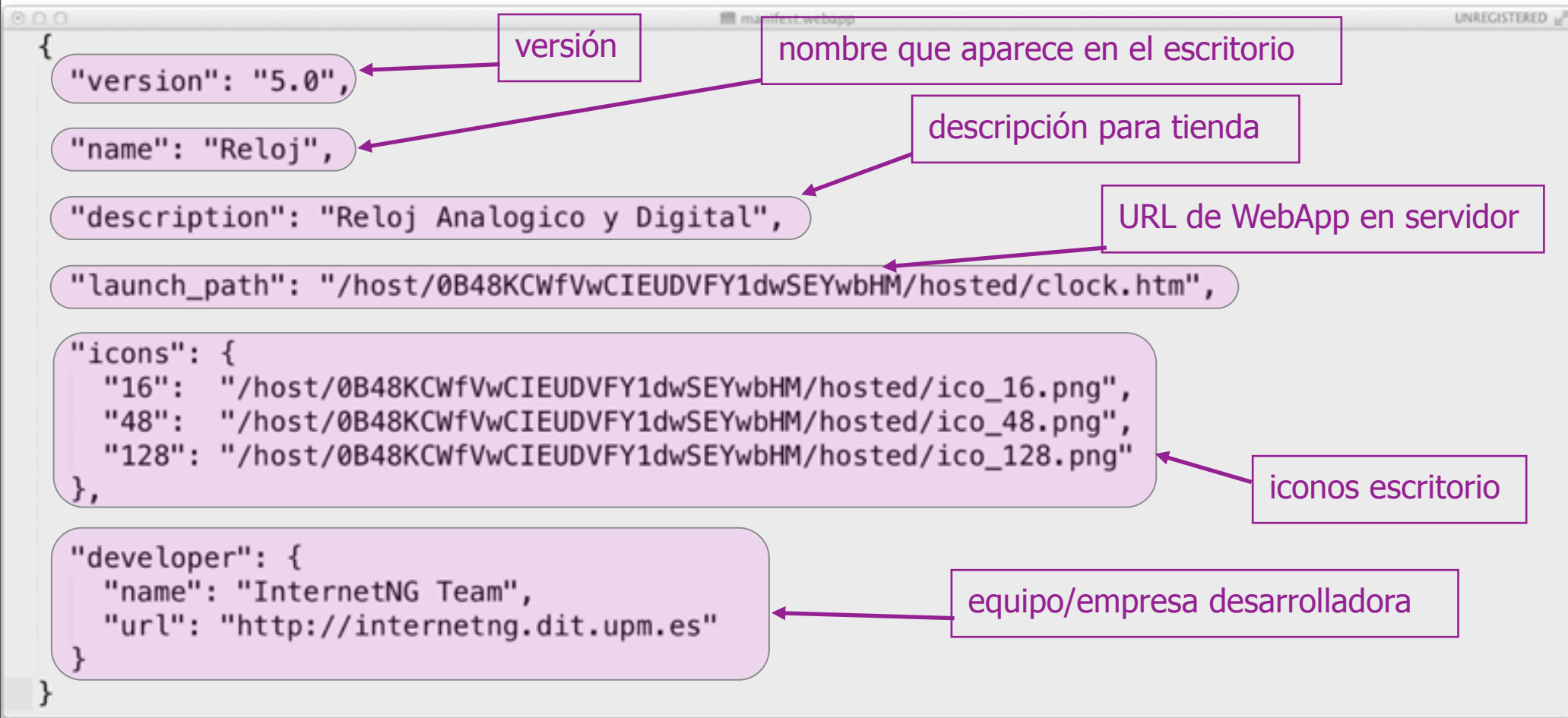
◆ Para crear una hosted App con Reloj SVG se debe añadir:

- **Iconos**
  - ◆ que se mostrarán en el escritorio
- **Instalador**
  - ◆ de la WebApp en el escritorio
- **Manifiesto**
  - ◆ con información de instalación



# manifest.webapp

- ◆ Fichero en formato JSON (JavaScript Object Notation)
  - Con información de instalación
- ◆ JSON representa datos como literales de objetos JavaScript
  - Todos los datos están serializados en el fichero como un string



# Instalar hosted App en simulador FirefoxOS

2. Introducir URL del instalador en el navegador y clicar (ejecutar instalador)

<https://googledrive.com/host/0B48KCWfVwCIEUDVFY1dwSEYwbHM/hosted/install.htm>

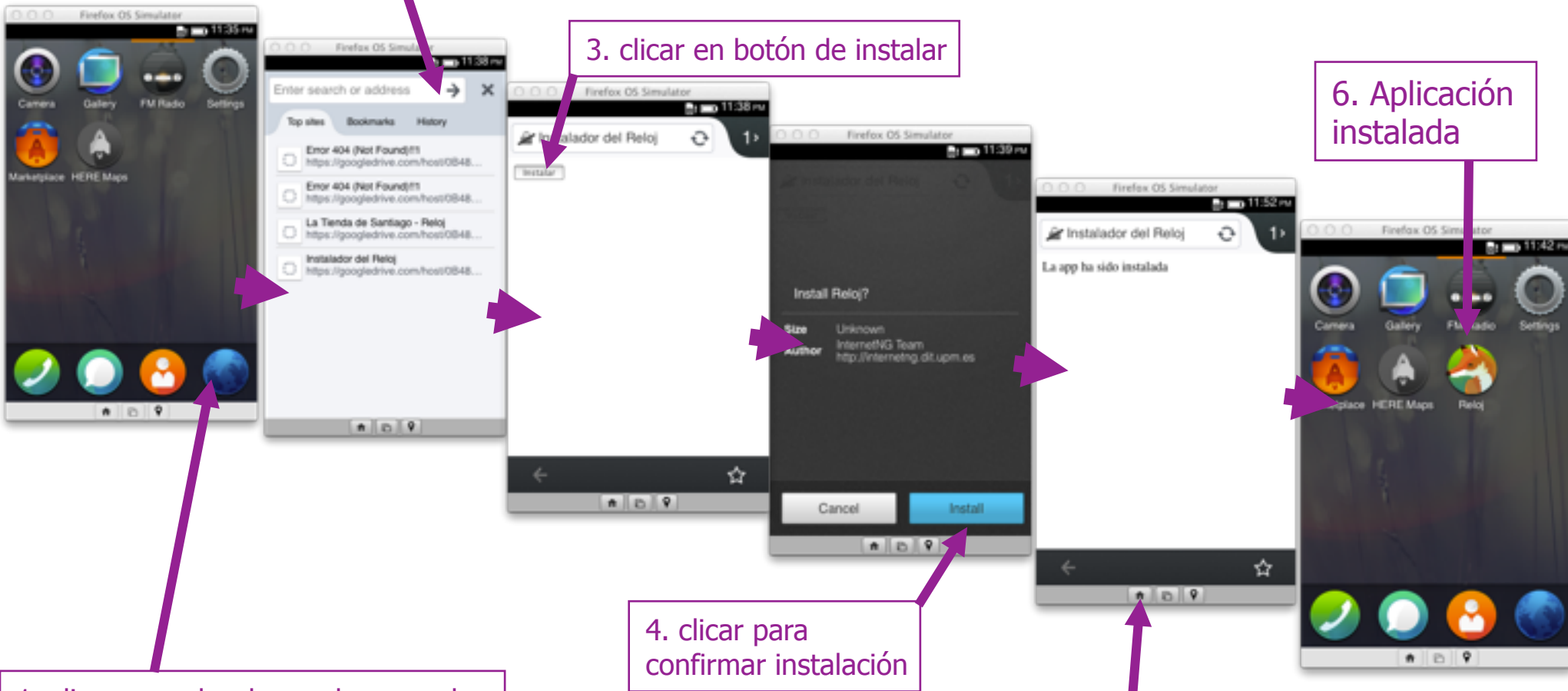
3. clicar en botón de instalar

6. Aplicación instalada

1. clicar para desplegar el navegador y poder instalar la hosted App

4. clicar para confirmar instalación

4. clicar para volver a escritorio



# install.html

```
<!DOCTYPE html>
<html><head><title>Instalador del Reloj</title>
<meta charset="UTF-8">
<script type="text/javascript" src="http://zeptojs.com/zepto.min.js" ></script>
</head>
<body>
  <div id="install">Estado de la instalación del Reloj 5.0</div>
<script>
  $(document).ready(function() {
    var gManifestName = location.href.replace("install.htm","") + 'manifest.webapp';
    var request = navigator.mozApps.checkInstalled(gManifestName);

    request.onsuccess = function() {
      if (request.result) { $("#install").text("La app ya esta instalada"); // App instalada
      else { // App no instalada
        $("#install").html("<input type='button' value='Instalar'>");

        $("#install").click(function() {
          var req = navigator.mozApps.install(gManifestName);
          req.onsuccess = function(data) {
            $("#install").text("La app ha sido instalada").unbind('click');
          }
          req.onerror = function(errObj) {
            alert("Error de instalacion: " + this.error.name);
          }
        });
      }
    }

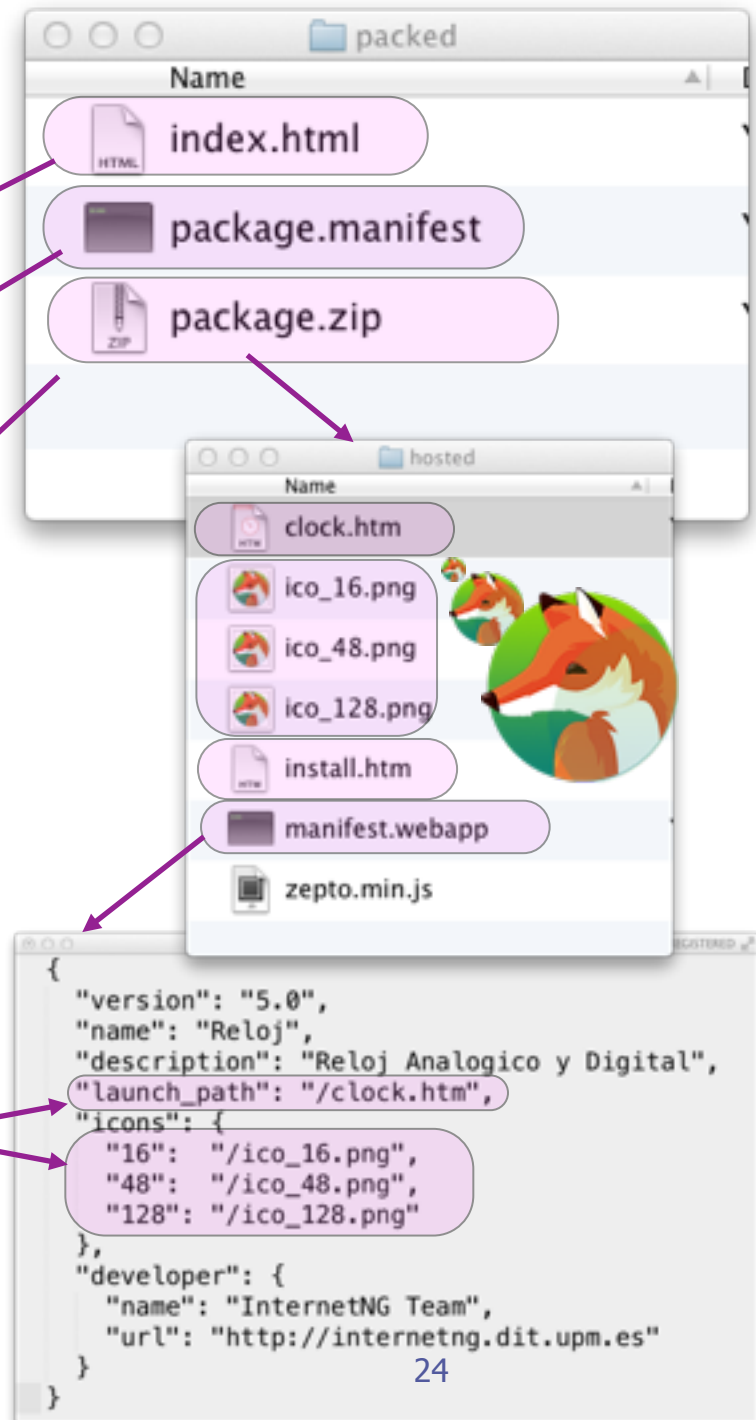
    request.onerror = function() { alert('Error de instalacion: ' + this.error.message); }
  });
</script>
</body>
</html>
```



# Reloj SVG: packaged App

## ◆ Componentes

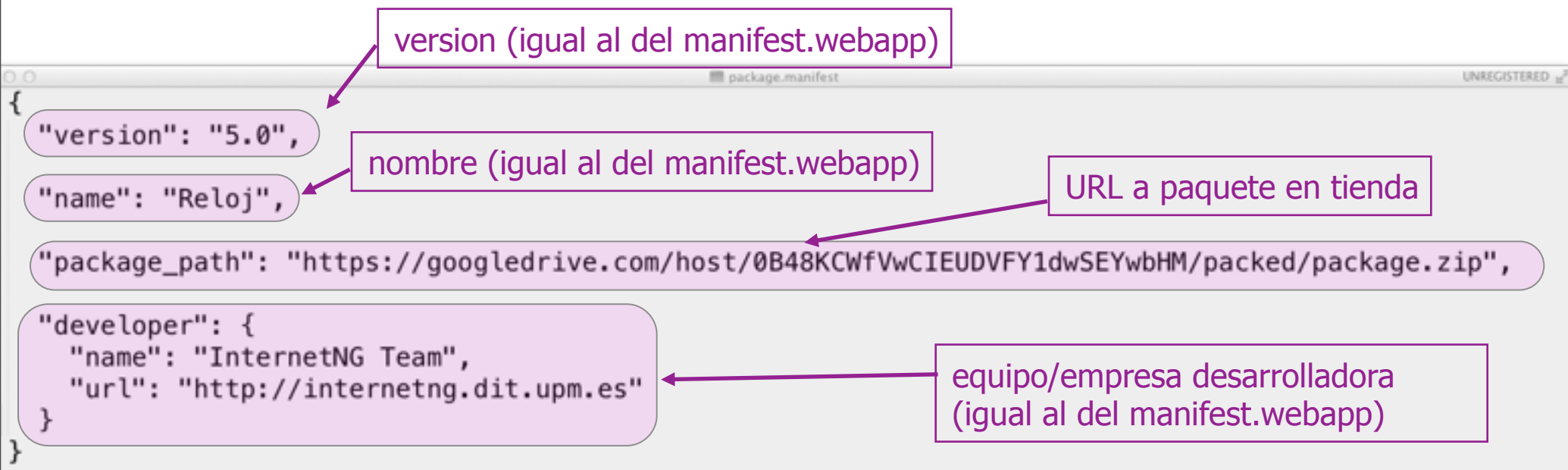
- **Instalador**
  - ◆ de la WebApp en el escritorio
- **Manifiesto**
  - ◆ con información de instalación del paquete
- **App empaquetada**
  - ◆ **Zip** con todos los ficheros de la App
    - los mismos ficheros de la hosted App
      - Salvo manifiesto: URLs relativos
  - ◆ **OJO**: Los ficheros deben empaquetarse sin más, sin estar en ningún directorio





# package.manifest

- ◆ Fichero en formato JSON (JavaScript Object Notation)
  - Con informacion sobre la webapp empaquetada
- ◆ JSON representa datos como literales de objetos JavaScript
  - Todos los datos están serializados en el fichero como un string



# Instalar hosted App en simulador FirefoxOS

2. Introducir URL del instalador en el navegador y clicar (ejecutar instalador)  
<https://googledrive.com/host/0B48KCWfVwCIEUDVFY1dwSEYwbHM/packed/>

3. clicar en botón de instalar

6. Aplicación instalada

1. clicar para desplegar el navegador y poder instalar la hosted App

4. clicar para confirmar instalación

4. clicar para volver a escritorio



```
<!DOCTYPE html>
<html>
<head>
  <title>La Tienda de Santiago - Reloj</title>
  <meta charset="UTF-8">
</head>

<body>
  <p>Página de Auto-Instalación del Reloj 5.0</p>
```

```
<script>
  (function() {
    var manifestUrl = location.href.replace("index.html","") + 'package.manifest';

    if (! navigator.mozApps.installPackage) {
      alert("ERROR: Esta aplicación no es compatible con tu dispositivo.");
      return;
    }

    var req = navigator.mozApps.installPackage(manifestUrl);
    req.onsuccess = function() {
      alert("Instalacion completada."+this.result.origin);
    };
    req.onerror = function() {
      alert("Error de instalación: "+this.error.name);
    };
  })();
</script>
</body>
</html>
```

# Ejercicio

- ◆ Publicar el cronómetro desarrollado en el Módulo 8
  - como aplicaciones hosted y packaged
    - ◆ en el directorio público de Google Drive
      - y comprobar que se instala correctamente en el simulador de FirefoxOS

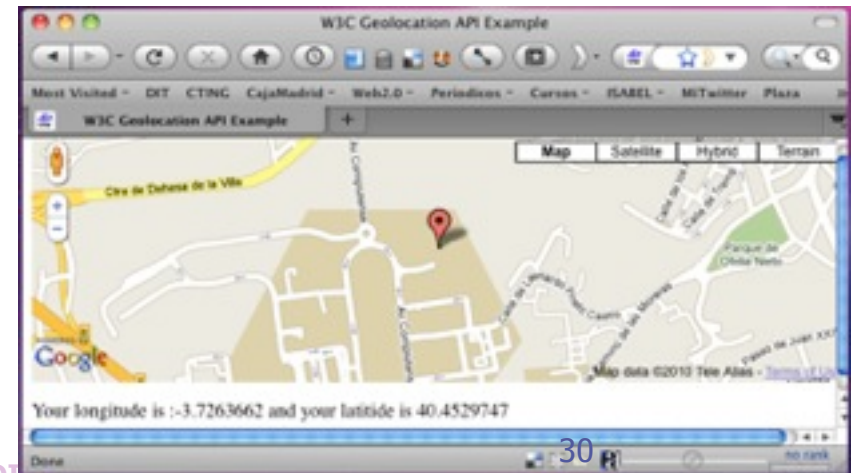
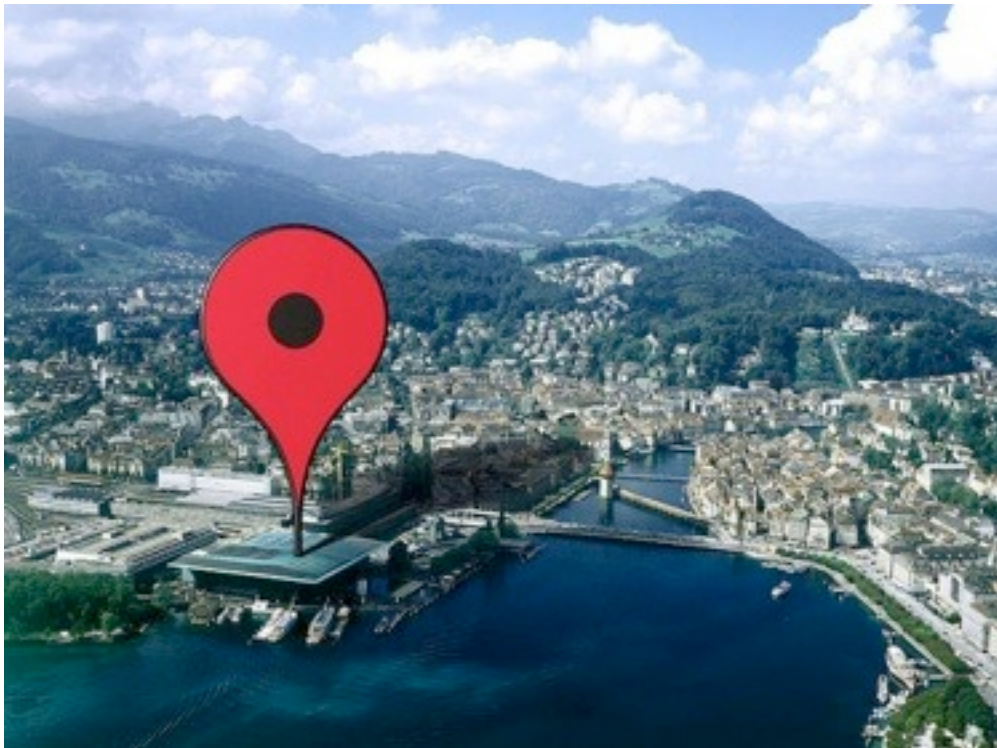


# Tema 10.5

## Geolocalización HTML5

# Geolocalización y Sensores

- ◆ HTML5 puede soportar geolocalización
  - En todo tipo de clientes
    - ◆ PCs, móviles tabletas, .....
- ◆ El interfaz de geolocalización
  - da acceso tambien a otros sensores
    - ◆ Brújula, acelerometro, .....

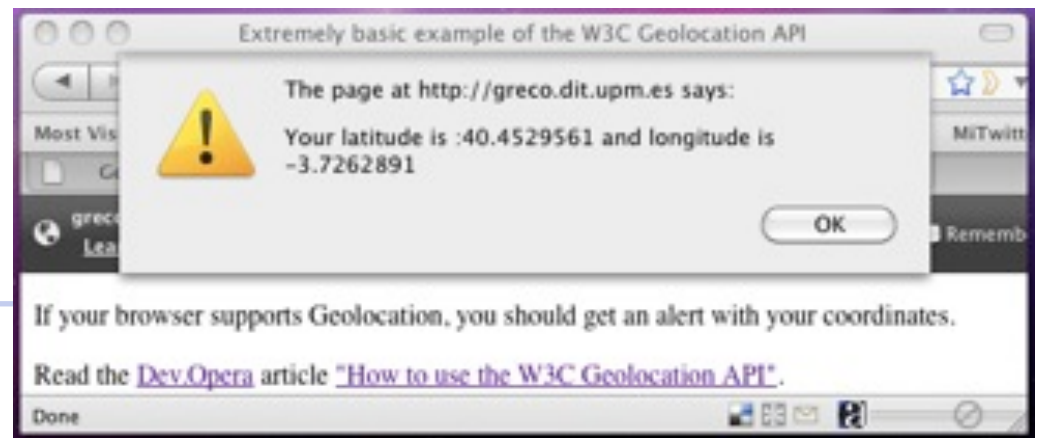




# Geolocalización

- ◆ La geolocalización se realiza siguiendo jerarquía de consultas
  - GPS -> antena WIFI -> antena GSM o 3G -> IP fija -> .....
  - ◆ Se devuelve la respuesta más precisa
- ◆ La geolocalización está accesible en el objeto **navigator.geolocation**
  - con método **getCurrentPosition(successFunction, errorFunction)**
    - ◆ Permite conocer
      - Latitud y longitud en formato decimal
      - Altitud y precisión de la altitud
      - Dirección y velocidad
- ◆ Norma y tutoriales
  - <http://dev.w3.org/geo/api/spec-source.html>
  - <http://dev.opera.com/articles/view/how-to-use-the-w3c-geolocation-api/>
  - <http://code.google.com/apis/maps/index.html>

# Ejemplo Geolocation



```
<!DOCTYPE html>
<html>
<head>
<title>Example of W3C Geolocation API</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript">

  if (navigator.geolocation) { //Check if browser supports W3C Geolocation API
    navigator.geolocation.getCurrentPosition (successFunction, errorFunction);
  } else { alert('Geolocation is not supported in this browser.')}

  function successFunction(position) {
    var lat = position.coords.latitude;
    var long = position.coords.longitude;
    alert('Your latitude is :'+lat+' and longitude is '+long);
  }

  function errorFunction(position) { alert('Error!'); }
</script>
</head>
<body>
<p>If your browser supports Geolocation, you should get an alert with your coordinates.</p>

<p>Read the <a href="http://dev.opera.com">Dev.Opera</a> article <a
href="http://dev.opera.com/articles/view/how-to-use-the-w3c-geolocation-api/">"How
to use the W3C Geolocation API"</a>.
</body>
</html>
```



# Funciones alert(), confirm() y prompt()

## ◆ Interacción sencilla basada en “pop-ups”

### ◆ alert(msj):

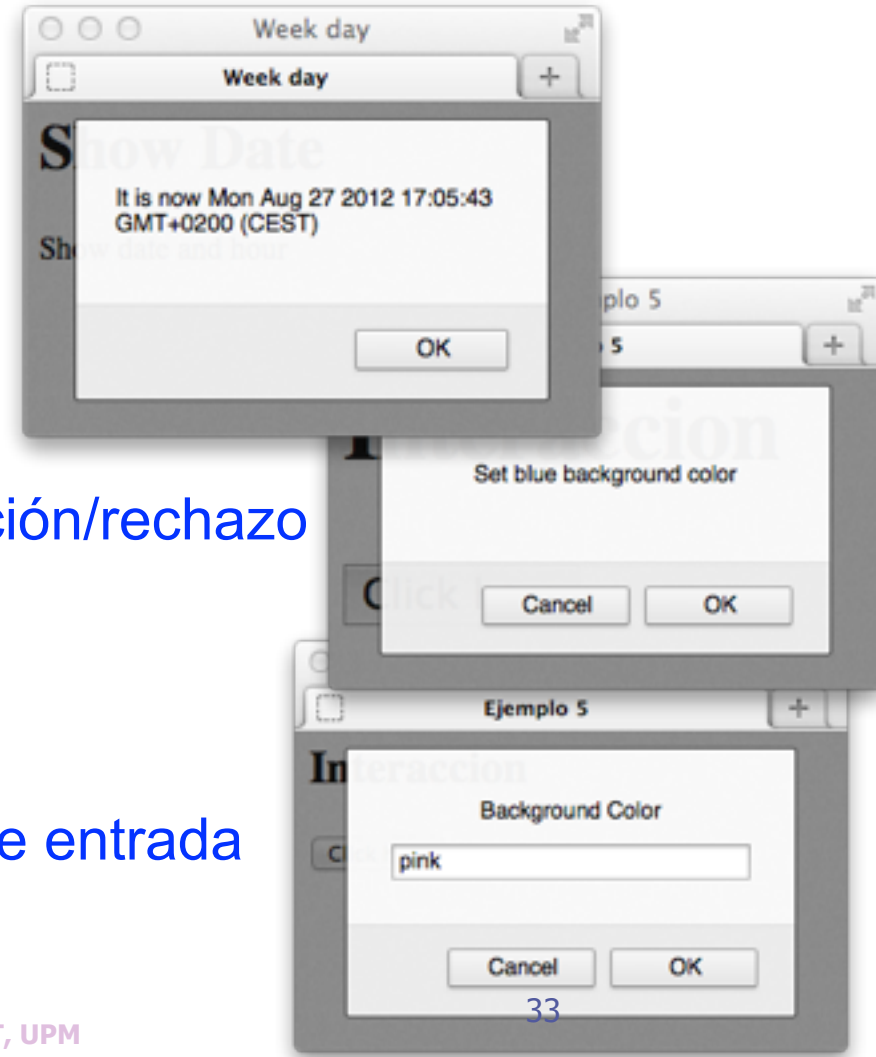
- presenta un mensaje al usuario
  - ◆ Retorna al pulsar OK

### ◆ confirm(msj):

- Presenta mensaje y pide confirmación/rechazo
  - ◆ Devuelve true/false

### ◆ prompt(msj):

- Presenta mensaje y pide un dato de entrada
  - ◆ Devuelve parametro introducido



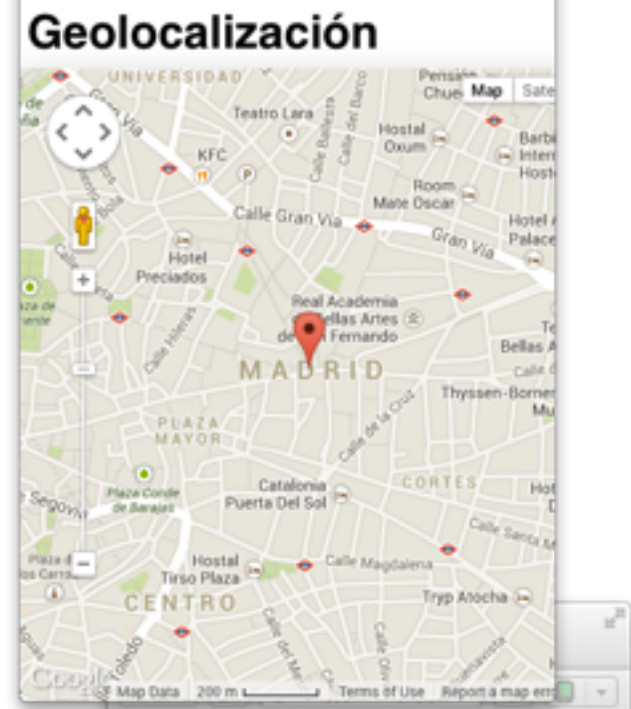


# Tema 10.6

## Aplicación geolocalizada en Google Maps

# Geolocalización con gmaps.js

- ◆ Aplicación de geolocalización
  - Carga un mapa centrado en nuestra posición
    - ◆ que se indica con un marcador
- ◆ Usamos librería gmaps.js para acceso a Google Maps
  - librería muy potente y sencilla de utilizar
    - ◆ <http://hpneo.github.io/gmaps/>
      - SE recomienda consultar documentación y ejemplos
  - La librería de Google es bastante mas compleja
- ◆ Se añade al mapa un manejador de eventos click/tap
  - que calcula la ruta hasta el punto indicado



# Geo-mapa

```
<link rel="stylesheet" type="text/css" href="mypath.css" />
```

```
<script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=true"></script>
<script type="text/javascript" src="gmaps.js"></script>
```

```
<script type="text/javascript" src="zepto.min.js"></script>
<script type="text/javascript">
  var map, lat, lng;
```

```
$ (function(){
```

```
  function geolocalizar(){
    GMaps.geolocate({
      success: function(position){
        lat = position.coords.latitude; // guarda coords en lat y lng
        lng = position.coords.longitude;

        map = new GMaps({ // muestra mapa centrado en coords [lat, lng]
          el: '#map',
          lat: lat,
          lng: lng
        });
        map.addMarker({ lat: lat, lng: lng}); // marcador en [lat, lng]
      },
      error: function(error) { alert('Error: '+error.message); },
      not_supported: function(){ alert("No soporta geolocalización"); },
    });
  };
};
```

```
  geolocalizar();
});
```

```
</script>
```

```
</head><body>
```

```
  <h1>Geolocalización</h1>
```

```
  <div id="map"></div>
```

```
</body></html>
```



# Geo-mapa

```
<script type="text/javascript">
  var map, distancia, lat, lng;

  $(function(){

    function enlazarMarcador(e){
      .....
    };

    function geolocalizar(){
      GMaps.geolocate({
        success: function(position){
          lat = position.coords.latitude; // guarda coords en lat y lng
          lng = position.coords.longitude;

          map = new GMaps({ // muestra mapa centrado en coords [lat, lng]
            el: '#map',
            lat: lat,
            lng: lng,
            click: enlazarMarcador, // eventos click y tap sobre el mapa
            tap: enlazarMarcador
          });
          map.addMarker({ lat: lat, lng: lng}); // marcador en [lat, lng]
        },
        error: function(error) { alert('Error: '+error.message); },
        not_supported: function(){ alert("No soporta geolocalización"); },
      });

      geolocalizar();
    });
  });
</script>
```





```
<script type="text/javascript">
  var map, distancia, lat, lng;
```

```
$(function(){
```

```
function enlazarMarcador(e){
```

```
  // muestra ruta entre marcas anteriores y actuales
```

```
  map.drawRoute({
    origin: [lat, lng], // origen en coordenadas anteriores
    // destino en coordenadas del click o toque actual
    destination: [e.latLng.lat(), e.latLng.lng()],
    travelMode: 'driving',
    strokeColor: '#000000',
    strokeOpacity: 0.6,
    strokeWeight: 5
  });
```

```
  lat = e.latLng.lat(); // guarda coords para marca siguiente
  lng = e.latLng.lng();
```

```
  map.addMarker({ lat: lat, lng: lng}); // pone marcador en mapa
};
```

```
function geolocalizar() {
```

```
  ....
  click: enlazarMarcador, // eventos click y tap sobre el mapa
  tap: enlazarMarcador
  ....
};
```

```
  geolocalizar();
```

```
});
```

```
</script>
```

# Geo-mapa



```
body{
  font-family: 'Droid Sans', 'Helvetica', Arial, sans-serif;
}
```

```
#latlng{
  display: block;
  margin: 0;
  padding: 0;
  position: absolute; /* posición absoluta a navegador */
  top: 0;             /* ajusta a borde de navegador */
  left: 0;
  right: 0;
  height: 50px;
}
```

```
#map{
  display: block;
  margin: 0;
  padding: 0;
  position: absolute; /* posición absoluta a navegador */
  top: 50px;          /* 50px debajo de borde de navegador */
  left: 0;            /* ajusta a borde de navegador */
  right: 0;
  bottom: 0;
  background: rgba(0,255,0,0.5); /* verde si no hay mapa */
}
```

## Estilo CSS multi-terminal



# Ejercicio entregable semana 5

- ◆ Modificar la aplicación con geolocalización en Google Maps para que
  - cuando se cierra la App, guardar la ruta en una variable de localStorage
    - ◆ para que al volver a abrirla reproduzca la ruta anterior
      - en vez de cargar un mapa limpio, como hace ahora
  - Además se debe incluir un boton de inicializar
    - ◆ que cargue un mapa limpio centrado en nuestra posición
      - colocando un marcador en dicho punto
- ◆ **Recomendación:** guardar la ruta como un array de posiciones
  - donde cada posición es un array de [lat, lng]
    - ◆ ejemplo: [[-1.040, 11.029], [0.040, 10.029], [5.040, 12.030]]
  - guardar el array en localStorage serializado con JSON.stringify
    - ◆ recuperarlo de localStorage deserializando con JSON.parse
- ◆ Publicar la App como aplicaciones hosted y packaged
  - en el directorio público de Google Drive
    - ◆ y comprobar que se instala correctamente en el simulador de FirefoxOS