

Firestore - Base de Datos en Tiempo Real

Sumario

- 1 Qué es Firestore
- 2 Registro en Firestore
- 3 Panel de Control de Firestore dentro de una Aplicación
- 4 Uso de Firestore con JavaScript
 - ◆ 4.1 Crear Proyecto en Firestore
 - ◆ 4.2 Guardar datos en Firestore
 - ◆ 4.3 Recuperar datos desde Firestore
 - ◆ 4.4 Actualizar y Borrar datos en Firestore
 - ◇ 4.4.1 Actualización de Datos
 - ◇ 4.4.2 Borrado de Datos
 - ◆ 4.5 Autenticación en Firestore
 - ◇ 4.5.1 Registro
 - ◇ 4.5.2 Login
 - ◇ 4.5.3 Administración (usuarios autenticados)
 - ◇ 4.5.4 Nuevo Producto (usuarios autenticados)
 - ◇ 4.5.5 Editar Producto (usuarios autenticados)
 - ◆ 4.6 Hosting en Firestore
 - ◇ 4.6.1 Configuración de las herramientas para trabajar desde línea de comandos
 - ◇ 4.6.2 Inicialización y envío de la aplicación a Firestore
 - ◇ 4.6.3 Desactivación del Hosting en Firestore

Qué es Firestore



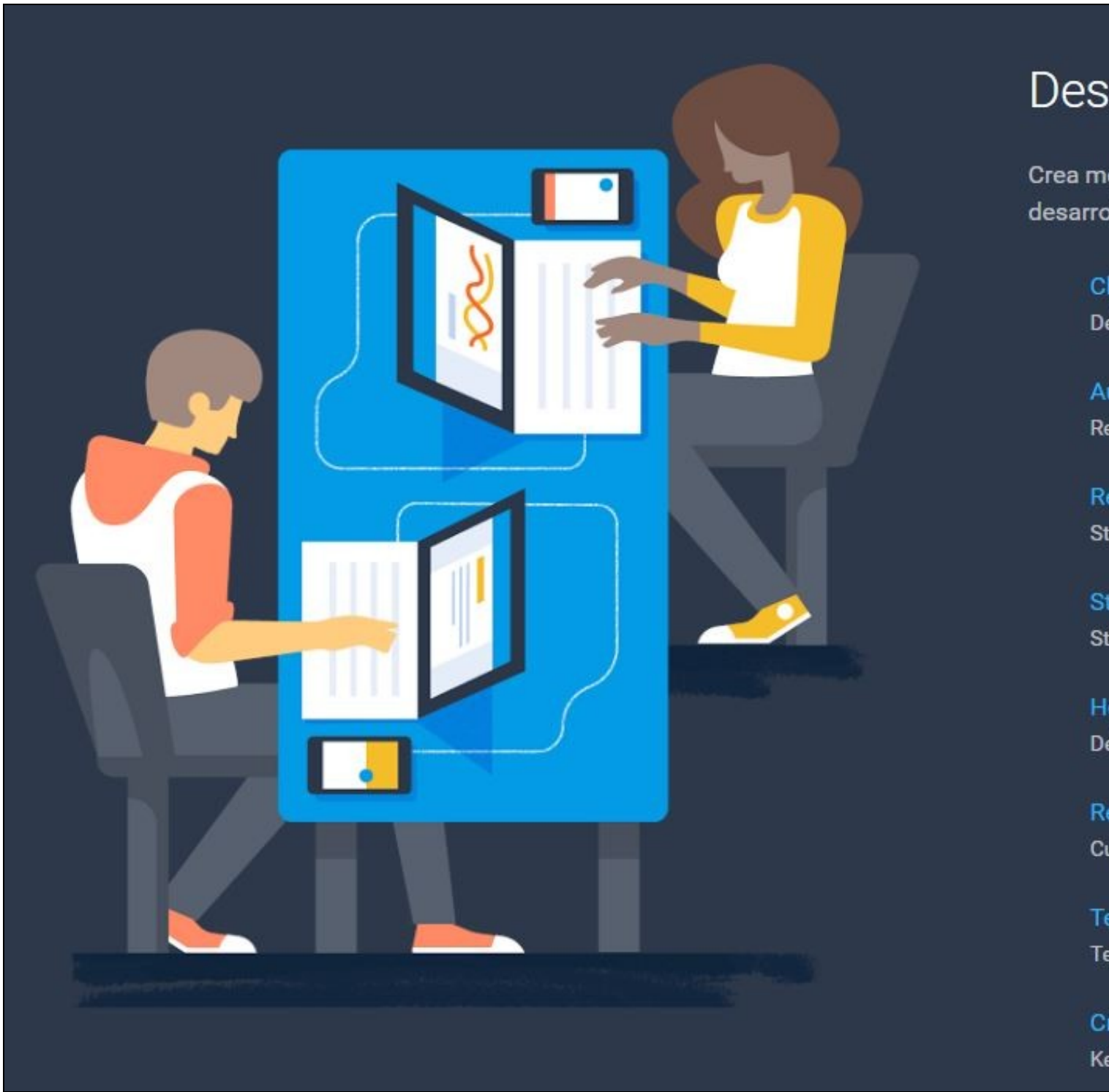
Firestore es un servicio web que nos proporciona un **backend** en la nube con una fuente de datos **NoSQL** en **tiempo real** y **librerías** para poder acceder a la base de datos desde aplicaciones **Web**, **IOS** o **Android**.

Los datos almacenados en **Firestore** se **sincronizan** con los clientes en **tiempo real**. Es básicamente con una base de datos remota que responde en tiempo real a los cambios realizados en los datos. De esta forma podemos escribir aplicaciones que almacenen o actualicen datos en Firestore y todos los clientes que utilicen dicha base de datos serán notificados en tiempo real de los cambios realizados.

Firestore también facilita la autenticación con el propio sistema de Firestore o mediante conectores como pueden ser Facebook, Twitter, etc.

Firestore nos permitirá programar aplicaciones centrándonos en la parte del frontend, dejando que todo el backend lo gestione Firestore.

Su **página oficial** es: <https://firebase.google.com>



Des

Crea m
desarro

Cl

De

Ar

Re

Re

St

St

St

H

De

Re

Cu

Te

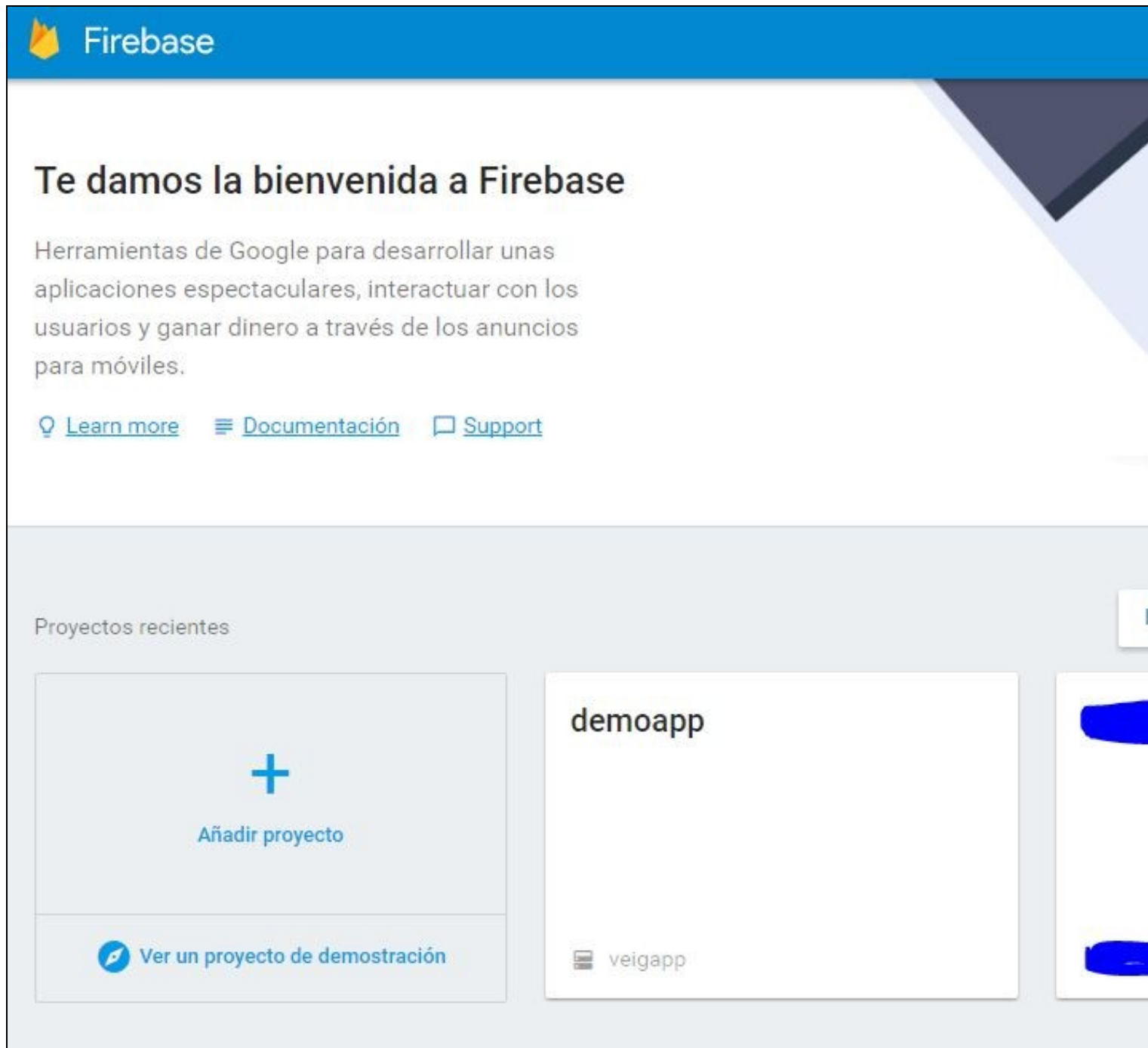
Te

Cl

Ke

Registro en Firebase

- Para **registrarnos en Firebase** lo podemos hacer con nuestra cuenta de Google (o del IES San Clemente en nuestro caso particular).
- Entraremos en <https://firebase.google.com/> y nos loguemos con la cuenta de Google o del Instituto..
- Aspecto de la consola que se muestra al loguearnos e ir a la Consola en Firebase:



The screenshot shows the Firebase console interface. At the top, there is a blue header with the Firebase logo and the word "Firebase". Below the header, a large heading reads "Te damos la bienvenida a Firebase". Underneath, a paragraph describes the tools as "Herramientas de Google para desarrollar unas aplicaciones espectaculares, interactuar con los usuarios y ganar dinero a través de los anuncios para móviles." Below this text are three links: "Learn more", "Documentación", and "Support".

The main content area is titled "Proyectos recientes" (Recent projects). It features a large light blue box on the left with a plus sign and the text "Añadir proyecto" (Add project). Below this box is a link "Ver un proyecto de demostración" (View a demo project). To the right, there are two project cards. The first card is titled "demoapp" and has a small icon at the bottom labeled "veigapp". The second card is partially visible on the right edge of the screenshot.

- **Proyecto de demostración:** <https://console.firebase.google.com/project/fir-demo-project/overview>
- **Tutorial para el uso de Firebase:** <https://firebase.google.com/docs/web/setup>

Panel de Control de Firebase dentro de una Aplicación

- **Aspecto del Panel de control - Funcionalidades más importantes:**
 - ◆ **Analytics:** estadísticas con las que podemos monitorar el estado de la aplicación y la cantidad de recursos que están siendo destinados para su funcionamiento (usuarios concurrentes, transferencia, etc).
 - ◆ **Authentication:** desde aquí se puede controlar cuáles son los mecanismos que estarán disponibles para la autenticación de usuarios, tanto email y clave como por redes sociales, junto con los datos de configuración.
 - ◆ **Database:** para acceder al modelo de datos. Se edita un JSON básicamente en esta sección.
 - ◆ **Storage:** para acceder al repositorio de archivos generados por los usuarios (como fotos, vídeos y archivos de audio) sin necesidad de utilizar código de servidor.
 - ◆ **Hosting:** es un espacio de alojamiento donde desplegar la app realizada. Una vez activado también se pueden controlar aquí las configuraciones del hosting.



Analytics

DESARROLLO



Authentication



Database



Storage



Hosting



Functions



Test Lab



Crash Reporting

AMPLIACIÓN



Notifications



Remote Config



Dynamic Links

BENEFICIOS



AdMob

Spark

Gratuito 0 \$/mes

[ACTUALIZAR](#)

Te damos la bienvenida a Firebase. En



Añade Firebase a tu aplicación de iOS



Añade Firebase a tu aplicación de Android

Descubre Firebase



Analytics

Consigue analíticas detalladas para analizar y calcular la interacción de los usuarios de tu aplicación

[Más información](#)

[EMPEZAR](#)



Authentic

Autentica y adm...
variedad de pro...
utilizar ningún e...

[Más informac](#)

Uso de Firebase con JavaScript

Ejemplos de código con Firebase: <https://firebase.google.com/docs/samples/#web>

Primeros pasos con Firebase (recomendable su lectura): <https://firebase.google.com/docs/database/web/start> Referencia de la **API de Firebase**: <https://firebase.google.com/docs/reference/js/>

Crear Proyecto en Firebase

- Guía de primeros pasos con Firebase, aquí: <https://firebase.google.com/docs/web/setup>
- Pulsamos en **Create a Project**

Crear proyecto ✕

Nombre del proyecto

País/Región ?

De forma predeterminada, los datos de Firebase Analytics mejoran otras funciones de Firebase y productos de Google. En la configuración puedes controlar cómo se comparten dichos datos en cualquier momento. [Más información](#)

CANCELAR **CREAR PROYECTO**

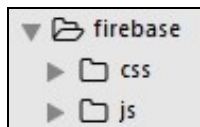
- A continuación pulsaremos el botón : **Añade Firebase a tu aplicación web.**



Guardar datos en Firebase

Vamos a realizar una prueba para **guardar datos en Firebase.**

- Creamos una **carpeta para nuestro proyecto** con los siguientes **directorios**:



- Crearemos un fichero **nuevoproducto.html** con el siguiente código:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <link rel="stylesheet" type="text/css" href="css/custom.css">

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
  <script src="https://www.gstatic.com/firebasejs/3.7.5/firebase.js"></script>
  <script src="js/nuevoproducto.js"></script>

  <title>Añadir un producto</title>
</head>
<body>
  <nav class="navbar navbar-inverse navbar-fixed-top">
    <div id="navbar" class="navbar-collapse collapse">
      <ul class="nav navbar-nav">
        <li><a href="index.html" title="Inicio">Inicio</a></li>
        <li><a href="administracion.html" title="">Administracion</a></li>
        <li class="active"><a href="nuevoproducto.html" title="">Añadir Productos</a></li>
      </ul>
    </div>
  </nav>
  <div class="container-fluid">
    <form id="formularioAlta">
      <div class="form-group">
        <label for="articulo">Articulo</label>
        <input type="text" class="form-control" id="articulo" placeholder="Nombre del artículo">
      </div>
      <div class="form-group">
```

```

        <label for="descripcion">Descripción</label>
        <textarea class="form-control" id="descripcion" placeholder="Descripción del artículo" rows="3"></textarea>
    </div>
    <div class="form-group">
        <label for="precio">Precio</label>
        <input type="text" class="form-control" id="precio" placeholder="Precio del artículo">
    </div>
    <div class="form-group">
        <label for="imagen">Elegir Imagen</label>
        <input type="file" id="imagen">
        <p class="help-block">Elija una imagen para este artículo</p>
        <img id="previsualizacion">
    </div>
    <button type="button" class="btn btn-primary" id="botonGuardar" disabled="disabled">Guardar</button>
</form>
</div>
</body>
</html>

```

- Crearemos el siguiente fichero **css/custom.css** con este código:

```

body{
    padding-top: 65px;
}

img{
    height:140px;
    width:140px;
}

```

- Crearemos el siguiente fichero **js/nuevoproducto.js** con este código:

```

$(document).ready(function()
{
    // Inicializar la base de datos
    var config = {
        apiKey: "AIzaSyANBMwjLUMhYwq2CDywa0Z47G2E3aJQ3ex",
        authDomain: "veigapp.firebaseio.com",
        databaseURL: "https://veigapp.firebaseio.com",
        projectId: "firebase-veigapp",
        storageBucket: "firebase-veigapp.appspot.com",
        messagingSenderId: "841479533091"
    };
    firebase.initializeApp(config);

    var database = firebase.database();

    var articulo;
    var descripcion;
    var precio;
    var imagen;

    $("#imagen").change(function()
    {
        var descriptor=new FileReader();
        descriptor.readAsDataURL(this.files[0]);

        descriptor.onloadend = function()
        {
            imagen=descriptor.result;
            $("#previsualizacion").attr("src",imagen);
        };
    });

    $("#formularioAlta").change(function()
    {
        articulo=$("#articulo").val();
        descripcion=$("#descripcion").val();
        precio=$("#precio").val();
    });
}

```



```

        if (articulo && descripcion && precio)
        {
            $("#botonGuardar").prop("disabled", false);
        }
        else
        {
            $("#botonGuardar").prop("disabled", true);
        }
    });

$("#botonGuardar").click(function()
{
    articulo=$("#articulo").val();
    descripcion=$("#descripcion").val();
    precio=$("#precio").val();

    if (!imagen)
    {
        imagen="NONE";
    }

    // Indicamos que la referencia base de nuestra base de datos es productos (algo así como el padre)
    // del que colgarán el resto de nodos hijos.
    /*
    var usersRef = new Firebase('https://samplechat.firebaseio-demo.com/users');
    var fredRef = usersRef.child('fred');
    var fredFirstNameRef = fredRef.child('name/first');
    */
    var referencia=database.ref("productos");

    // De la siguiente forma el método sobrescribe los datos
    /*
    referencia.set(
    {
        articulo: articulo,
        descripcion: descripcion,
        precio: precio,
        imagen: imagen
    });
    */

    // Ahora estamos poniendo el articulo como clave en la colección
    // De esta manera podremos añadir nuevos articulos o actualizar uno ya existente.

    /*
    referencia.child(articulo).set(
    {
        descripcion: descripcion,
        precio: precio,
        imagen: imagen
    });
    */

    // Si queremos permitir que haya artículos con nombres duplicados entonces tendremos
    // que decirle a Firebase que utilice otra clave en lugar del nombre del articulo.
    // Usaremos el método push en lugar de set
    referencia.push(
    {
        articulo: articulo,
        descripcion: descripcion,
        precio: precio,
        imagen: imagen
    },function()
    {
        alert('El alta se ha realizado correctamente');
    });
});
});

```

- Aspecto de la base de datos en **Firestore** después de añadir **3 productos** desde **nuevoproducto.html**:

The screenshot shows the Firebase Realtime Database interface for a project named 'demoapp'. The left sidebar contains navigation options: Overview, Analytics, and a 'DESARROLLO' section with Authentication, Database (highlighted), Storage, Hosting, Functions, Test Lab, and Crash Reporting. Below that is an 'AMPLIACIÓN' section with Notifications, Remote Config, and Dynamic Links, and a 'BENEFICIOS' section with AdMob. At the bottom left, there is a 'Spark' offer for 'Gratuito 0 \$/mes' with an 'ACTUALIZAR' button.

The main area displays the 'Realtime Database' structure for the URL 'https://veigapp.firebaseio.com/'. The root node is 'veigapp', which contains a 'productos' node. This node has three child entries, each with the following fields:

- Entry 1:** Key: `-KgtvudDfQeRRLYoN2MI`
 - `articulo:` "Foto"
 - `descripcion:` "Es una foto"
 - `imagen:` "data:image/png;base64,iVBORw0"
 - `precio:` "123"
- Entry 2:** Key: `-KgtwSFPH-6jplRr5WE6`
 - `articulo:` "Router Gigabit"
 - `descripcion:` "Es un router Gigabi"
 - `imagen:` "data:image/jpeg;base64,/9j/4A"
 - `precio:` "245"
- Entry 3:** Key: `-KgtwXa5yT6EHRPyr9It`
 - `articulo:` "Raspberry PI"
 - `descripcion:` "Es la Raspberry versión 3"
 - `imagen:` "data:image/png;base64,iVBORw0"
 - `precio:` "35"

Recuperar datos desde Firebase

- Cómo recuperar datos desde Firebase para hacer un listado de Artículos en Venta.

- Crear un fichero `index.html`:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <link rel="stylesheet" type="text/css" href="css/custom.css">

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
  <script src="https://www.gstatic.com/firebasejs/3.7.5/firebase.js"></script>
  <script src="js/index.js"></script>

  <title>Gestión de Productos con Firebase</title>
</head>
<body>
  <nav class="navbar navbar-inverse navbar-fixed-top">
    <div id="navbar" class="navbar-collapse collapse">
      <ul class="nav navbar-nav">
        <li class="active"><a href="index.html" title="Inicio">Inicio</a></li>
        <li><a href="administracion.html" title="">Administracion</a></li>
        <li><a href="nuevoproducto.html" title="">Añadir Productos</a></li>
      </ul>
    </div>
  </nav>
  <div class="container-fluid" id="listado">
    <h1>Listado de Productos</h1>
  </div>
</body>
</html>
```

- Crear un fichero `js/index.js`:

```
$(document).ready(function()
{
  // Inicializar la base de datos
  var config = {
    apiKey: "AIzaSyANBMwjLUMhYwq2CDywa0Z47G2E3aJQ3ex",
    authDomain: "veigapp.firebaseio.com",
    databaseURL: "https://veigapp.firebaseio.com",
    projectId: "firebase-veigapp",
    storageBucket: "firebase-veigapp.appspot.com",
    messagingSenderId: "841479533091"
  };

  firebase.initializeApp(config);

  var database = firebase.database();

  // Fijarse que la ruta de partida ahora es la colección productos:
  var referencia=database.ref("productos");

  var productos={};

  /*
  Evento: value
```

The value event is used to read a static snapshot of the contents at a given database path, as they existed at the time of the read event. It is triggered once with the initial data and again every time the data changes. The event callback is passed a snapshot containing all data at that location, including child data. In our code example above, value returned all of the blog posts in our app. Everytime a new blog post is added, the callback function will return all of th

```

*/

referencia.on('value',function(datos)
{
    productos=datos.val();

    // Recorremos los productos y los mostramos
    $.each(productos, function(indice,valor)
    {
        var prevProducto='<div class="row"><div class="col-md-3 cabeceraProducto">';

        prevProducto+='<h2>'+valor.articulo+'</h2></div>';

        prevProducto+='<div class="row"><div class="col-md-3 cabeceraProducto">';
        prevProducto+='<h2>'+valor.precio+'?. </h2></div>';
        prevProducto+='</div>';

        prevProducto+='<div class="row">';
        prevProducto+='<div class="col-md-3 imagenFix">';
        if (valor.imagen=='NONE')
            prevProducto+='<img alt="Sin Fotografia"/>';
        else
            prevProducto+='';
        prevProducto+='</div>';

        prevProducto+='<div class="col-md-3">';
        prevProducto+='<p>'+valor.descripcion+'</p>';
        prevProducto+='</div>';
        prevProducto+='</div>';

        prevProducto+='</div>';
        prevProducto+='<div class="row espaciador">';
        prevProducto+='</div>';

        $(prevProducto).appendTo('#listado');
    });

},function(objetoError){
    console.log('Error de lectura:'+objetoError.code);
});

});

```

- Editar el fichero **css/custom.css**:

```

body{
    padding-top: 65px;
}

img{
    height:140px;
    width:140px;
}

.cabeceraProducto{
    border-bottom-style: solid;
    border-bottom-width: 1px;
    border-bottom-color: grey;
    background-color: black;
    color:white;
}

.imagenFix{
    min-height: 140px;
}

.espaciador{
    min-height: 30px;
}

```

- Aspecto de la página **index.html** con el listado funcionando:

Listado de Productos

Foto

123€.



Es una foto

Router Gigabit

245€.



Es un router Gigabit

Raspberry PI

35€.



Es la Raspberry versión 3.

Actualizar y Borrar datos en Firebase

- Como podemos **actualizar y borrar los datos en Firebase**.
- Partimos del ejemplo del **index.html** e **index.js** para crear los siguientes ficheros.

- Contenido del fichero **administracion.html**:

```
<!DOCTYPE html>
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <link rel="stylesheet" type="text/css" href="css/custom.css">

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
  <script src="https://www.gstatic.com/firebasejs/3.7.5/firebase.js"></script>
  <script src="js/administracion.js"></script>

  <title>Listado Productos</title>
</head>
<body>
  <nav class="navbar navbar-inverse navbar-fixed-top">
    <div id="navbar" class="navbar-collapse collapse">
      <ul class="nav navbar-nav">
        <li><a href="index.html" title="Inicio">Inicio</a></li>
        <li class="active"><a href="administracion.html" title="">Administracion</a></li>
        <li><a href="nuevoproducto.html" title="">Añadir Productos</a></li>
      </ul>
    </div>
  </nav>
  <div class="container-fluid" id="listado">
    <h1>Edición de Productos</h1>
  </div>
</body>
</html>
```

- Contenido del fichero **js/administracion.js**:

```
// Inicializar la base de datos
var config = {
  apiKey: "AIzaSyANBMwjLUMhYwq2CDywa0Z47G2E3aJQ3ex",
  authDomain: "veigapp.firebaseio.com",
  databaseURL: "https://veigapp.firebaseio.com",
  projectId: "firebase-veigapp",
  storageBucket: "firebase-veigapp.appspot.com",
  messagingSenderId: "841479533091"
};

firebase.initializeApp(config);
var database = firebase.database();

var referencia=database.ref("productos");

var productos={};

/*
Evento: value

The value event is used to read a static snapshot of the contents at a given database path,
as they existed at the time of the read event. It is triggered once with the initial data and again every time the data changes.
The event callback is passed a snapshot containing all data at that location, including child data. In our code example above,
value returned all of the blog posts in our app. Everytime a new blog post is added, the callback function will return all of the posts
*/

referencia.on('value', function(datos)
```

```

{
  // Eliminamos el contenido del listado para actualizarlo.
  $("#listado div.row").remove();

  productos=datos.val();

  // Recorremos los productos y los mostramos
  $.each(productos, function(indice,valor)
  {
    var prevProducto='<div class="row" id="'+indice+'"><div class="col-md-3 cabeceraProducto">';

    prevProducto+='<h2>'+valor.articulo+'</h2></div>';

    prevProducto+='<div class="row"><div class="col-md-3 cabeceraProducto">';
    prevProducto+='<h2>'+valor.precio+' ?.</h2></div>';
    prevProducto+='</div>';

    prevProducto+='<div class="row">';
    prevProducto+='<div class="col-md-3 imagenFix">';
    if (valor.imagen=='NONE')
      prevProducto+='<img alt="Sin Fotografía"/>';
    else
      prevProducto+='';
    prevProducto+='</div>';

    prevProducto+='<div class="col-md-3">';
    prevProducto+='<p>'+valor.descripcion+'</p>';
    prevProducto+='</div>';
    prevProducto+='</div>';

    prevProducto+='<div class="row">';

    prevProducto+='<div class="col-md-3">';
    prevProducto+='<button type="button" class="btn btn-warning" onclick="editarProducto(\''+indice+'\')">Editar Producto</button>';
    prevProducto+='</div>';

    prevProducto+='<div class="col-md-3">';
    prevProducto+='<button type="button" class="btn btn-danger" onclick="borrarProducto(\''+indice+'\')">Borrar Producto</button>';
    prevProducto+='</div>';

    prevProducto+='</div>';
    prevProducto+='<div class="row espaciador">';
    prevProducto+='</div>';

    $(prevProducto).appendTo('#listado');
  });
},function(objetoError){
  console.log('Error de lectura:'+objetoError.code);
});

```

Actualización de Datos

- Primero añadiremos en el fichero **js/administracion.js** la **función** para realizar la edición de un producto:
- Contenido del fichero **js/administracion.js**:

```

// Inicializar la base de datos
var config = {
  apiKey: "AIzaSyANBMwjLUMhYwq2CDywa0Z47G2E3aJQ3ex",
  authDomain: "veigapp.firebaseio.com",
  databaseURL: "https://veigapp.firebaseio.com",
  projectId: "firebase-veigapp",
  storageBucket: "firebase-veigapp.appspot.com",
  messagingSenderId: "841479533091"
};

firebase.initializeApp(config);
var database = firebase.database();

var referencia=database.ref("productos");

var productos={};

```

```

/*
Evento: value

The value event is used to read a static snapshot of the contents at a given database path,
as they existed at the time of the read event. It is triggered once with the initial data and again every time the data changes.
The event callback is passed a snapshot containing all data at that location, including child data. In our code example above,
value returned all of the blog posts in our app. Everytime a new blog post is added, the callback function will return all of the po
*/

referencia.on('value',function(datos)
{
    // Eliminamos el contenido del listado para actualizarlo.
    $("#listado div.row").remove();

    productos=datos.val();

    // Recorremos los productos y los mostramos
    $.each(productos, function(indice,valor)
    {
        var prevProducto='<div class="row" id="'+indice+'"><div class="col-md-3 cabeceraProducto">';

        prevProducto+='<h2>'+valor.articulo+'</h2></div>';

        prevProducto+='<div class="row"><div class="col-md-3 cabeceraProducto">';
        prevProducto+='<h2>'+valor.precio+' ?.</h2></div>';
        prevProducto+='</div>';

        prevProducto+='<div class="row">';
        prevProducto+='<div class="col-md-3 imagenFix">';
        if (valor.imagen=='NONE')
            prevProducto+='<img alt="Sin Fotografía"/>';
        else
            prevProducto+='';
        prevProducto+='</div>';

        prevProducto+='<div class="col-md-3">';
        prevProducto+='<p>'+valor.descripcion+'</p>';
        prevProducto+='</div>';
        prevProducto+='</div>';

        prevProducto+='<div class="row">';

        prevProducto+='<div class="col-md-3">';
        prevProducto+='<button type="button" class="btn btn-warning" onclick="editarProducto(\''+indice+'\')">Editar Producto</button>';
        prevProducto+='</div>';

        prevProducto+='<div class="col-md-3">';
        prevProducto+='<button type="button" class="btn btn-danger" onclick="borrarProducto(\''+indice+'\')">Borrar Producto</button>';
        prevProducto+='</div>';

        prevProducto+='</div>';
        prevProducto+='<div class="row espaciador">';
        prevProducto+='</div>';

        $(prevProducto).appendTo('#listado');
    });

},function(objetoError){
    console.log('Error de lectura:'+objetoError.code);
});

function editarProducto(id)
{
    // Para pasar el ID a otro proceso lo hacemos a través de window.name
    window.name= id;

    // Cargamos la página editarproducto.html
    location.assign('editarproducto.html');
}

```

- Partiendo de una copia del fichero **nuevoproducto.html** vamos a crear **editarproducto.html**:

• Contenido del fichero **editarproducto.html**:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <link rel="stylesheet" type="text/css" href="css/custom.css">

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
  <script src="https://www.gstatic.com/firebasejs/3.7.5/firebase.js"></script>
  <script src="js/editarproducto.js"></script>

  <title>Edición de Productos</title>
</head>
<body>
  <nav class="navbar navbar-inverse navbar-fixed-top">
    <div id="navbar" class="navbar-collapse collapse">
      <ul class="nav navbar-nav">
        <li><a href="index.html" title="Inicio">Inicio</a></li>
        <li class="active"><a href="administracion.html" title="">Administracion</a></li>
        <li><a href="nuevoproducto.html" title="">Añadir Productos</a></li>
      </ul>
    </div>
  </nav>
  <div class="container-fluid">

    <form id="formularioEdicion">
      <div class="form-group">
        <label for="articulo">Articulo</label>
        <input type="text" class="form-control" id="articulo">
      </div>
      <div class="form-group">
        <label for="descripcion">Descripción</label>
        <textarea class="form-control" id="descripcion" rows="3"></textarea>
      </div>
      <div class="form-group">
        <label for="precio">Precio</label>
        <input type="text" class="form-control" id="precio">
      </div>
      <div class="form-group">
        <label for="imagen">Elegir Imagen</label>
        <input type="file" id="imagen">
        <p class="help-block">Elija una imagen para este articulo</p>
        <img id="previsualizacion">
      </div>
      <button type="button" class="btn btn-primary" id="botonActualizar">Actualizar Producto</button>
    </form>
  </div>
</body>
</html>
```

• Contenido del fichero **js/editarproducto.js**:

```
$(document).ready(function()
{
  // Inicializar la base de datos
  var config = {
    apiKey: "AIzaSyANBMwjLUMhYwq2CDywa0Z47G2E3aJQ3ex",
    authDomain: "veigapp.firebaseio.com",
    databaseURL: "https://veigapp.firebaseio.com",
    projectId: "firebase-veigapp",
    storageBucket: "firebase-veigapp.appspot.com",
    messagingSenderId: "841479533091"
  };

  firebase.initializeApp(config);
  var database = firebase.database();
  var referencia=database.ref("productos");
```

```

var productoId= window.name;
//console.log(productoId);

var articulo, descripcion, precio, imagen;
var producto={};

// Buscamos el artículo.
referencia.child(productoId).once('value',function(datos)
{
    producto=datos.val();

    articulo= producto.articulo;
    descripcion= producto.descripcion;
    precio=producto.precio;
    imagenEdicion=producto.imagen;

    $('#articulo').val(articulo);
    $('#descripcion').val(descripcion);
    $('#precio').val(precio);
    $('#previsualizacion').attr('src',imagenEdicion);
});

$('#imagen').change(function()
{
    var descriptor=new FileReader();
    descriptor.readAsDataURL(this.files[0]);

    descriptor.onloadend = function()
    {
        imagenEdicion=descriptor.result;
        $('#previsualizacion').attr("src",imagenEdicion);
    };
});

$('#botonActualizar').click(function()
{
    var articulo=$('#articulo').val();
    var descripcion=$('#descripcion').val();
    var precio=$('#precio').val();
    var imagen=imagenEdicion;

    // Guardamos los datos en referencia
    referencia.child(productoId).update(
    {
        articulo: articulo,
        descripcion: descripcion,
        precio: precio,
        imagen: imagen,
    }, alFinalizar);
});

function alFinalizar(error)
{
    if (error)
    {
        alert('Ha habido problemas al realizar la operación: '+error.code);
    }
    else{
        alert('Operación realizada con éxito !');
        location.assign('administracion.html');
    }
}
});

```

Borrado de Datos

- La parte de borrado se completa programando la función **borrarProducto** dentro del fichero **js/administracion.js**:
- Contenido del fichero **js/administracion.js**:

```
// Inicializar la base de datos
```

```

var config = {
  apiKey: "AIzaSyANBMwjLUMhYwq2CDywa0Z47G2E3aJQ3ex",
  authDomain: "veigapp.firebaseio.com",
  databaseURL: "https://veigapp.firebaseio.com",
  projectId: "firebase-veigapp",
  storageBucket: "firebase-veigapp.appspot.com",
  messagingSenderId: "841479533091"
};

firebase.initializeApp(config);
var database = firebase.database();

var referencia=database.ref("productos");

var productos={};

/*
Evento: value

The value event is used to read a static snapshot of the contents at a given database path,
as they existed at the time of the read event. It is triggered once with the initial data and again every time the data changes.
The event callback is passed a snapshot containing all data at that location, including child data. In our code example above,
value returned all of the blog posts in our app. Everytime a new blog post is added, the callback function will return all of the posts
*/

referencia.on('value',function(datos)
{
  // Eliminamos el contenido del listado para actualizarlo.
  $("#listado div.row").remove();

  productos=datos.val();

  // Recorremos los productos y los mostramos
  $.each(productos, function(indice,valor)
  {
    var prevProducto='<div class="row" id="'+indice+'"><div class="col-md-3 cabeceraProducto">';

    prevProducto+='<h2>'+valor.articulo+'</h2></div>';

    prevProducto+='<div class="row"><div class="col-md-3 cabeceraProducto">';
    prevProducto+='<h2>'+valor.precio+' ?.</h2></div>';
    prevProducto+='</div>';

    prevProducto+='<div class="row">';
    prevProducto+='<div class="col-md-3 imagenFix">';
    if (valor.imagen=='NONE')
      prevProducto+='<img alt="Sin Fotografía"/>';
    else
      prevProducto+='';
    prevProducto+='</div>';

    prevProducto+='<div class="col-md-3">';
    prevProducto+='<p>'+valor.descripcion+'</p>';
    prevProducto+='</div>';
    prevProducto+='</div>';

    prevProducto+='<div class="row">';

    prevProducto+='<div class="col-md-3">';
    prevProducto+='<button type="button" class="btn btn-warning" onclick="editarProducto(\''+indice+'\')">Editar Producto</button>';
    prevProducto+='</div>';

    prevProducto+='<div class="col-md-3">';
    prevProducto+='<button type="button" class="btn btn-danger" onclick="borrarProducto(\''+indice+'\')">Borrar Producto</button>';
    prevProducto+='</div>';

    prevProducto+='</div>';
    prevProducto+='<div class="row espaciador">';
    prevProducto+='</div>';

    $(prevProducto).appendTo('#listado');
  });
});

```

```
},function(objetoError){
    console.log('Error de lectura:'+objetoError.code);
});

function editarProducto(id)
{
    // Para pasar el ID a otro proceso lo hacemos a través de window.name
    window.name= id;

    // Cargamos la página editarproducto.html
    location.assign('editarproducto.html');
}

function borrarProducto(id)
{
    if (confirm("¿Está seguro/a de que quiere borrar este artículo?") == true)
    {
        referencia.child(id).remove();
    }
}
}
```

Autenticación en Firebase

- Vamos a dar un poco de **seguridad** a nuestra **aplicación** en **Firebase** para incorporar autenticación a los datos.
- **Firebase** nos permite usar **diferentes modos de autenticación E-mail, facebook, twitter**, etc.. para acceder a las aplicaciones:



Overview



Analytics

PROGRAMAR

Authentication

Database

Storage

Hosting

Functions

Test Lab

Crash Reporting

AUMENTAR

Notifications

Remote Config

Dynamic Links

GANAR

AdMob

Spark
Gratis \$0/mes

ACTUALIZAR



Proveedores de acceso

Proveedor

Correo electrónico/contraseña

Google

Facebook

Twitter

GitHub

Anónimo

Dominios de redireccionamiento de OAuth ?

Dominio autorizado

localhost








veigapp.firebaseio.com

127.0.0.1

auth.firebaseio.com

- Vamos a probar la **autenticación** a través de **E-mail y Password**.
- Para ello **activaremos** dicha opción en el **dashboard** de Firebase.

Proveedores de acceso

Proveedor	Estado
 Correo electrónico/contraseña	
<p>Permite a los usuarios registrarse con su dirección de correo electrónico y contraseña. SDK también proporcionan la verificación de la dirección de correo, la recuperación de y las primitivas de cambio de dirección de correo. Más información </p>	
 Google	Inhabilitado
 Facebook	Inhabilitado
 Twitter	Habilitado
 GitHub	Habilitado
 Anónimo	Inhabilitado

Registro

- Creamos la página **registro.html**:

Registro de Usuarios

Dirección E-mail:

Contraseña:

Confirmar Contraseña:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <link rel="stylesheet" type="text/css" href="css/custom.css">

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
  <script src="https://www.gstatic.com/firebasejs/3.7.5/firebase.js"></script>
  <script src="js/registro.js"></script>

  <title>Registro de Usuarios</title>
</head>
<body>
  <div class="container-fluid">
    <h2>Registro de Usuarios</h2>
    <form id="formularioRegistro">
      <div class="form-group">
        <label for="email">Dirección E-mail: </label>
        <input type="text" class="form-control" id="email" placeholder="Introduzca su e-mail">
      </div>

      <div class="form-group">
        <label for="password">Contraseña: </label>
        <input type="password" class="form-control" id="password" placeholder="Introduzca contraseña">
      </div>

      <div class="form-group">
        <label for="password2">Confirmar Contraseña: </label>
        <input type="password" class="form-control" id="password2" placeholder="Confirme contraseña">
      </div>

      <button type="button" class="btn btn-primary" id="botonRegistro">Registrar</button>
      <button type="button" class="btn btn-warning" id="botonCancelar">Cancelar</button>

    </form>
  </div>
</body>
</html>
```

• Crearemos la página `js/registro.js`:

```
// Inicializar la base de datos
var config = {
  apiKey: "AIzaSyANBMwjLUMhYwq2CDywa0Z47G2E3aJQ3ex",
  authDomain: "veigapp.firebaseio.com",
  databaseURL: "https://veigapp.firebaseio.com",
  projectId: "firebase-veigapp",
  storageBucket: "firebase-veigapp.appspot.com",
  messagingSenderId: "841479533091"
};

firebase.initializeApp(config);

var email,password,passwordConfirm;

function exito()
{
  alert('Se ha creado la cuenta de usuario correctamente. ');
  location.assign('index.html');
}

function alFinalizar(error)
{
  // console.log(error);

  if (error!==undefined)
  {
    // Códigos de error:
    // auth/invalid-email
    // auth/weak-password
    // auth/email-already-in-use
    switch(error.code)
    {
      case 'auth/email-already-in-use':
        alert('ERROR: No se puede crear la nueva cuenta de usuario, por que el e-mail ya está en uso !');
        break;
      case 'auth/invalid-email':
        alert('ERROR: El e-mail facilitado no es un e-mail correcto. ');
        break;
      default:
        alert('Se ha producido un error al crear el usuario.\n\n'+error+'\n\n');
        break;
    }
  }
}

$(function()
{
  // Programamos el click de los botones del formulario:
  $("#botonRegistro").click(function()
  {
    email=$("#email").val();
    password=$("#password").val();
    passwordConfirm=$("#password2").val();

    if (password != passwordConfirm)
    {
      alert("Error: Las contraseñas son distintas!");
    }
    else
      firebase.auth().createUserWithEmailAndPassword(email,password).then(exito).catch(alFinalizar);
  });

  $("#botonCancelar").click(function()
  {
    location.assign('index.html');
  });
});
```


Login

- Creamos la página **login.html**:



```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <link rel="stylesheet" type="text/css" href="css/custom.css">

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
  <script src="https://www.gstatic.com/firebasejs/3.7.5/firebase.js"></script>
  <script src="js/login.js"></script>
  <title>Login de Usuarios</title>
</head>
<body>
  <div class="container-fluid">
    <h1>Login/registro de Usuarios</h1>
    <div id="spinner"></div>
    <form id="formularioRegistro">
      <div class="form-group">
        <label for="email">Dirección E-mail: </label>
        <input type="text" class="form-control" id="email" placeholder="Introduzca su e-mail">
      </div>

      <div class="form-group">
        <label for="password">Contraseña: </label>
        <input type="password" class="form-control" id="password" placeholder="Introduzca contraseña">
      </div>

      <button type="button" class="btn btn-success" id="botonLogin">Login</button>
      <button type="button" class="btn btn-primary" id="botonRegistro">Registro</button>
      <button type="button" class="btn btn-warning" id="botonCancelar">Cancelar</button>
    </form>
  </div>

</body>
</html>
```

- Crearemos la página **js/login.js**:

```
// Inicializar la base de datos
var config = {
  apiKey: "AIzaSyANBMwjLUMhYwq2CDywa0Z47G2E3aJQ3ex",
  authDomain: "veigapp.firebaseio.com",
  databaseURL: "https://veigapp.firebaseio.com",
  projectId: "firebase-veigapp",
```

```

        storageBucket: "firebase-veigapp.appspot.com",
        messagingSenderId: "841479533091"
    };

    firebase.initializeApp(config);

    function exito()
    {
        $("#spinner").html("");
        location.assign('index.html');
    }

    $(function()
    {
        $("#botonLogin").click(function()
        {
            $("#spinner").html("<img src='img/spinner.gif' style='width:25px; height:25px;'/>");
            var email=$("#email").val();
            var password=$("#password").val();

            firebase.auth().signInWithEmailAndPassword(email, password).then(exito).catch(function(error)
            {
                $("#spinner").html("");
                //console.log(error);
                alert ("Error detectado:\n\n"+error.message);
            });
        });

        $("#botonRegistro").click(function()
        {
            location.assign('registro.html');
        });

        $("#botonCancelar").click(function()
        {
            location.assign('index.html');
        });
    });
});

```

Administración (usuarios autenticados)

- Editamos la página `js/administracion.js` para tener en cuenta el usuario logueado:

```

// Inicializar la base de datos
var config = {
    apiKey: "AIzaSyANBMwjLUMhYwq2CDywa0Z47G2E3aJQ3ex",
    authDomain: "veigapp.firebaseio.com",
    databaseURL: "https://veigapp.firebaseio.com",
    projectId: "firebase-veigapp",
    storageBucket: "firebase-veigapp.appspot.com",
    messagingSenderId: "841479533091"
};

firebase.initializeApp(config);
var database = firebase.database();
var referencia=database.ref("productos");
var productos={};

// Chequeamos la autenticación antes de acceder al resto de contenido de este fichero.
firebase.auth().onAuthStateChanged(function(user) {
    if (user)
    {
        console.log(user);
        console.log('Usuario: '+user.uid+' está logueado con '+user.providerData[0].providerId);
        var logueado='<li><p class="navbar-text navbar-center">'+user.email+'</p></li>';
        logueado+='<li><button type="button" class="btn btn-warning navbar-btn" id="botonLogout">Salir</button></li>';

        $(logueado).appendTo('.nav');
        $("#botonLogout").click(desconectar);
    }
});

```

```

} else
{
    console.log('Usuario no logueado');
    location.assign('login.html');
}
});

function desconectar()
{
    firebase.auth().signOut().then(function() {
        location.assign('index.html');
    }, function(error)
    {
        alert("Error al intentar desconectarse.");
    });
}

```

```

/*
Evento: value

```

The value event is used to read a static snapshot of the contents at a given database path, as they existed at the time of the read event. It is triggered once with the initial data and again every time the data changes. The event callback is passed a snapshot containing all data at that location, including child data. In our code example above, value returned all of the blog posts in our app. Everytime a new blog post is added, the callback function will return all of the posts.

```

referencia.on('value',function(datos)
{
    // Eliminamos el contenido del listado para actualizarlo.
    $("#listado div.row").remove();

    productos=datos.val();

    // Recorremos los productos y los mostramos
    $.each(productos, function(indice,valor)
    {
        var prevProducto='<div class="row" id="'+indice+'"><div class="col-md-3 cabeceraProducto">';

        prevProducto+='<h2>'+valor.articulo+'</h2></div>';

        prevProducto+='<div class="row"><div class="col-md-3 cabeceraProducto">';
        prevProducto+='<h2>'+valor.precio+' ?.</h2></div>';
        prevProducto+='</div>';

        prevProducto+='<div class="row">';
        prevProducto+='<div class="col-md-3 imagenFix">';
        if (valor.imagen=='NONE')
            prevProducto+='<img alt="Sin Fotografía"/>';
        else
            prevProducto+='';
        prevProducto+='</div>';

        prevProducto+='<div class="col-md-3">';
        prevProducto+='<p>'+valor.descripcion+'</p>';
        prevProducto+='</div>';
        prevProducto+='</div>';

        prevProducto+='<div class="row">';

        prevProducto+='<div class="col-md-3">';
        prevProducto+='<button type="button" class="btn btn-warning" onclick="editarProducto(\''+indice+'\')">Editar Producto</button>';
        prevProducto+='</div>';

        prevProducto+='<div class="col-md-3">';
        prevProducto+='<button type="button" class="btn btn-danger" onclick="borrarProducto(\''+indice+'\')">Borrar Producto</button>';
        prevProducto+='</div>';

        prevProducto+='</div>';
        prevProducto+='<div class="row espaciador">';
        prevProducto+='</div>';
    });
}

```

```

        $(prevProducto).appendTo('#listado');
    });

    },function(objetoError){
        console.log('Error de lectura:'+objetoError.code);
    });

function editarProducto(id)
{
    // Para pasar el ID a otro proceso lo hacemos a través de window.name
    window.name= id;

    // Cargamos la página editarproducto.html
    location.assign('editarproducto.html');
}

function borrarProducto(id)
{
    if (confirm("¿Está seguro/a de que quiere borrar este artículo?") == true)
    {
        referencia.child(id).remove();
    }
}

```

- Aspecto de la **administración** con el **usuario logueado**:

Inicio
Administración
Añadir Productos
veiga@iessanclemente.net
Salir

Edición de Productos

Dibujos animados	10 €.
	Es una foto sin más.
Editar Producto	Borrar Producto
Router Gigabit	278 €.
	Es un router Gigabit

Nuevo Producto (usuarios autenticados)

- Editamos la página `js/nuevoproducto.js` para tener en cuenta el usuario logueado:

```
$(document).ready(function()
{
    // Inicializar la base de datos
    var config = {
        apiKey: "AIzaSyANBMwjLUMhYwq2CDywa0Z47G2E3aJQ3ex",
        authDomain: "veigapp.firebaseio.com",
        databaseURL: "https://veigapp.firebaseio.com",
        projectId: "firebase-veigapp",
        storageBucket: "firebase-veigapp.appspot.com",
        messagingSenderId: "841479533091"
    };
    firebase.initializeApp(config);
    var database = firebase.database();

    var articulo;
    var descripcion;
    var precio;
    var imagen;

    // Chequeamos la autenticación antes de acceder al resto de contenido de este fichero.
    firebase.auth().onAuthStateChanged(function(user)
    {
        if (user)
        {
            console.log(user);
            console.log('Usuario: '+user.uid+' está logueado con '+user.providerData[0].providerId);
            var logueado='<li><p class="navbar-text navbar-center">'+user.email+'</p></li>';
            logueado+='<li><button type="button" class="btn btn-warning navbar-btn" id="botonLogout">Salir</button></li>';

            $(logueado).appendTo('.nav');
            $("#botonLogout").click(desconectar);

        } else
        {
            console.log('Usuario no logueado');
            location.assign('login.html');
        }
    });

    function desconectar()
    {
        firebase.auth().signOut().then(function()
        {
            location.assign('index.html');
        }, function(error)
        {
            alert("Error al intentar desconectarse.");
        });
    }

    $("#imagen").change(function()
    {
        var descriptor=new FileReader();
        descriptor.readAsDataURL(this.files[0]);

        descriptor.onloadend = function()
        {
            imagen=descriptor.result;
            $("#previsualizacion").attr("src",imagen);
        };
    });

    $("#formularioAlta").change(function()
    {
        articulo=$("#articulo").val();
    });
});
```

```

descripcion=$("#descripcion").val();
precio=$("#precio").val();

if (articulo && descripcion && precio)
{
    $("#botonGuardar").prop("disabled", false);
}
else
{
    $("#botonGuardar").prop("disabled", true);
}
});

$("#botonGuardar").click(function()
{
    articulo=$("#articulo").val();
    descripcion=$("#descripcion").val();
    precio=$("#precio").val();

    if (!imagen)
    {
        imagen="NONE";
    }

    // Indicamos que la referencia base de nuestra base de datos es productos (algo así como el padre)
    // del que colgarán el resto de nodos hijos.
    /*
    var usersRef = new Firebase('https://samplechat.firebaseio-demo.com/users');
    var fredRef = usersRef.child('fred');
    var fredFirstNameRef = fredRef.child('name/first');
    */
    var referencia=database.ref("productos");

    // De la siguiente forma el método sobrescribe los datos
    /*
    referencia.set(
    {
        articulo: articulo,
        descripcion: descripcion,
        precio: precio,
        imagen: imagen
    });
    */

    // Ahora estamos poniendo el articulo como clave en la colección
    // De esta manera podremos añadir nuevos articulos o actualizar uno ya existente.

    /*
    referencia.child(articulo).set(
    {
        descripcion: descripcion,
        precio: precio,
        imagen: imagen
    });
    */

    // Si queremos permitir que haya artículos con nombres duplicados entonces tendremos
    // que decirle a Firebase que utilice otra clave en lugar del nombre del articulo.
    // Usaremos el método push en lugar de set
    referencia.push(
    {
        articulo: articulo,
        descripcion: descripcion,
        precio: precio,
        imagen: imagen
    },function()
    {
        alert('El alta se ha realizado correctamente!');
    });
});

```

));

- Aspecto de la **Añadir Productos** con el **usuario logueado**:

[Inicio](#) [Administracion](#) **Añadir Productos** [veiga@iessanclemente.net](#) [Salir](#)

Artículo

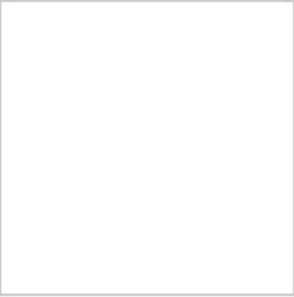
Descripción

Precio

Elegir Imagen

 Ningún archivo seleccionado

Elija una imagen para este artículo



Editar Producto (usuarios autenticados)

- Editamos la página `js/editarproducto.js` para tener en cuenta el usuario logueado:

```
$(document).ready(function()
{
    // Inicializar la base de datos
    var config = {
        apiKey: "AIzaSyANBMwjLUMhYwq2CDywa0Z47G2E3aJQ3ex",
        authDomain: "veigapp.firebaseio.com",
        databaseURL: "https://veigapp.firebaseio.com",
        projectId: "firebase-veigapp",
        storageBucket: "firebase-veigapp.appspot.com",
        messagingSenderId: "841479533091"
    };

    firebase.initializeApp(config);
    var database = firebase.database();
    var referencia=database.ref("productos");

    var productoId= window.name;
    //console.log(productoId);

    var articulo, descripcion, precio, imagen;
    var producto={};

    // Chequeamos la autenticación antes de acceder al resto de contenido de este fichero.
    firebase.auth().onAuthStateChanged(function(user)
    {
        if (user)
        {
            console.log(user);
            console.log('Usuario: '+user.uid+' está logueado con '+user.providerData[0].providerId);
            var logueado='<li><p class="navbar-text navbar-center">'+user.email+'</p></li>';
            logueado+='<li><button type="button" class="btn btn-warning navbar-btn" id="botonLogout">Salir</button></li>';

            $(logueado).appendTo('.nav');
            $("#botonLogout").click(desconectar);

        } else
        {
            console.log('Usuario no logueado');
            location.assign('login.html');
        }
    });

    function desconectar()
    {
        firebase.auth().signOut().then(function()
        {
            location.assign('index.html');
        }, function(error)
        {
            alert("Error al intentar desconectarse.");
        });
    }

    // Buscamos el artículo.
    referencia.child(productoId).once('value',function(datos)
    {
        producto=datos.val();

        articulo= producto.articulo;
        descripcion= producto.descripcion;
        precio=producto.precio;
        imagenEdicion=producto.imagen;

        $('#articulo').val(articulo);
        $('#descripcion').val(descripcion);
        $('#precio').val(precio);
        $('#previsualizacion').attr('src',imagenEdicion);
    });
});
```



```

});

$("#imagen").change(function()
{
    var descriptor=new FileReader();
    descriptor.readAsDataURL(this.files[0]);

    descriptor.onloadend = function()
    {
        imagenEdicion=descriptor.result;
        $("#previsualizacion").attr("src",imagenEdicion);
    };
});

$("#botonActualizar").click(function()
{
    var articulo=$("#articulo").val();
    var descripcion=$("#descripcion").val();
    var precio=$("#precio").val();
    var imagen=imagenEdicion;

    // Guardamos los datos en referencia
    referencia.child(productoId).update(
    {
        articulo: articulo,
        descripcion: descripcion,
        precio: precio,
        imagen: imagen,
    }, alFinalizar);
});

function alFinalizar(error)
{
    {
        if (error)
        {
            alert('Ha habido problemas al realizar la operación: '+error.code);
        }
        else{
            alert('Operación realizada con éxito !');
            location.assign('administracion.html');
        }
    }
});

```

Hosting en Firebase

- Firebase ofrece hosting de nuestra aplicación. <https://firebase.google.com/docs/hosting/?hl=es>
- Información de cómo configurar el hosting en la sección **Hosting** del menú izquierdo.
- Es necesario tener instalado previamente **Node.js**: <https://nodejs.org/en/>
- Tendremos que instalar **FIREBASE CLI** desde **npm**. <https://firebase.google.com/docs/cli/?hl=es>

Configuración de las herramientas para trabajar desde línea de comandos

```

# Instalamos firebase-tools a nivel global
npm install -g firebase-tools

# Cerramos la línea de comandos y volvemos a abrirla para ejecutar el comando "firebase login" para permitir el acceso de nuestra app
# Se abrirá una ventana de navegador, dónde tendremos que autorizar el acceso a nuestra base de datos.
firebase login

# Obtendremos algo como esto:
D:\xampp\htdocs\web_dev\www.veiga.local\firebasegoogle>firebase login
? Allow Firebase to collect anonymous CLI usage information? Yes

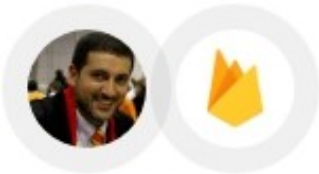
Visit this URL on any device to log in:
https://accounts.google.com/o/oauth2/auth?client_id=563584335869-fgrhgmd47bq.....

Waiting for authentication...



# Se abrirá en el navegador una ventana de autorización:



```



- Aspecto de la **ventana de autorización** de acceso a Firebase:







▾ Firebase CLI quiere hacer lo siguiente:

-  Conocer tu identidad en Google. 

-  Consultar tu dirección de correo electrónico. 

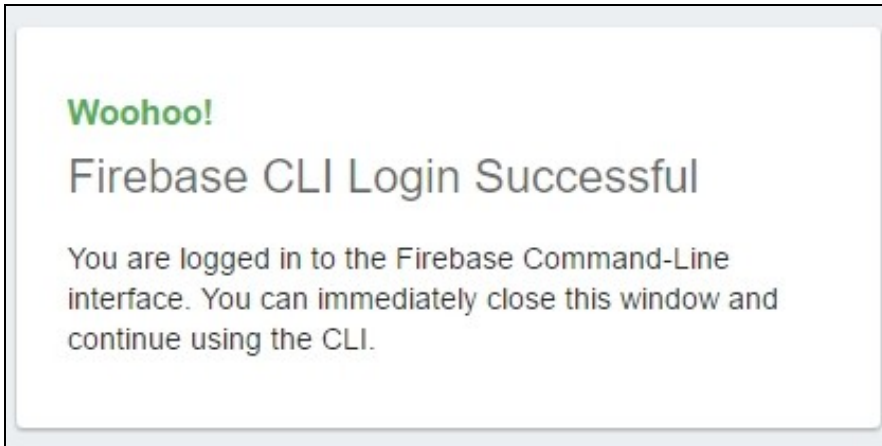
-  Ver y administrar tus datos en los servicios de Google Cloud Platform 

-  Ver tus proyectos de Cloud Platform 

-  Ver y administrar toda tu configuración y tus datos de Firebase 

Al hacer clic en Permitir, autorizas a esta aplicación y a Google a utilizar tu información según lo establecido en sus respectivas condiciones de servicio y políticas de privacidad. Puedes cambiar este y otros [permisos de la cuenta](#) en cualquier momento.

- Aspecto de la **ventana de confirmación de autorización** de acceso a Firebase:



- Por último se mostrará en la línea de comandos el mensaje de autenticación con éxito.
- A partir de este momento podremos interactuar con Firebase a través de la línea de comandos.

```
D:\xampp\htdocs\web\_dev\www.veiga.local\firebasegoogle>firebase login
? Allow Firebase to collect anonymous CLI usage information? Yes

Visit this URL on any device to log in:
https://accounts.google.com/o/oauth2/auth?client_id=563584335869-fgrhgm47bq.....

Waiting for authentication...

+ Success! Logged in as veiga@iessanclemente.net
```

Inicialización y envío de la aplicación a Firebase

- **Listado de aplicaciones con firebase list:**

```
# Para ver todas nuestras aplicaciones de Firebase lo haremos con '''firebase list''':
D:\xampp\htdocs\web\_dev\www.veiga.local\firebasegoogle>firebase list
```

A screenshot of a terminal window showing the output of the 'firebase list' command. The output is a table with three columns: Name, Project ID / Instance, and Permissions. The first row shows 'demoapp' with project ID 'firebase-veigapp' and instance 'veigapp (instance)', and the user is listed as 'Owner'.

Name	Project ID / Instance	Permissions
demoapp	firebase-veigapp veigapp (instance)	Owner

- A Firebase le tendremos que indicar qué carpeta es la que queremos subir al servidor.
- Para ello creamos una **carpeta en nuestro proyecto** llamada **deploy** y **hacemos una copia de todo nuestro proyecto allí dentro**.
- Entramos desde la línea de comandos en la carpeta principal de nuestro proyecto (**no entrar en la carpeta deploy**).
- Una vez dentro de la carpeta del proyecto ejecutaremos el comando **firebase init** :

```
# Le indicamos que queremos Hosting, para ello usamos los cursores y el espaciador para dejar marcado solamente Hosting.

D:\xampp\htdocs\web\_dev\www.veiga.local\firebasegoogle>firebase init
```

```
#####  #####  #####  #####  #####  #####  #####  #####
##      ##  ##      ##  ##      ##      ##  ##      ##
#####  ##  #####  #####  #####  #####  #####  #####
##      ##  ##      ##  ##      ##      ##  ##      ##
##      #####  ##  #####  #####  ##      ##  #####  #####
```

You're about to initialize a Firebase project in this directory:

```
D:\xampp\htdocs\web\_dev\www.veiga.local\firebasegoogle
```

? Are you ready to proceed? Yes

? What Firebase CLI features do you want to setup for this folder? (Press <space> to select)

() Database: Deploy Firebase Realtime Database Rules

() Functions: Configure and deploy Cloud Functions

>(*) Hosting: Configure and deploy Firebase Hosting sites

Seleccionamos a continuación el proyecto dónde vamos a alojar los ficheros.

Indicamos también el directorio deploy

Y que no queremos sobrescribir el index.html que ya tenemos.

=== Project Setup

First, let's associate this project directory with a Firebase project.

You can create multiple project aliases by running firebase use --add,

but for now we'll just set up a default project.

? What Firebase project do you want to associate as default? demoapp (firebase-veigapp)

=== Hosting Setup

Your public directory is the folder (relative to your project directory) that

will contain Hosting assets to be uploaded with firebase deploy. If you

have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? deploy

? Configure as a single-page app (rewrite all urls to /index.html)? No

+ Wrote deploy/404.html

? File deploy/index.html already exists. Overwrite? No

i Skipping write of deploy/index.html

i Writing configuration info to firebase.json...

i Writing project information to .firebaserc...

+ Firebase initialization complete!

• Contenido del fichero **firebase.json** creado después de la **inicialización**:

```
# Contenido del fichero firebase.json:
```

```
{
  "hosting": {
    "public": "deploy"
  }
}
```

• Para **subir la aplicación al servidor** se hace con **firebase deploy**:

```
D:\xampp\htdocs\web\_dev\www.veiga.local\firebasegoogle>firebase deploy
```

=== Deploying to 'firebase-veigapp'...

i deploying hosting

i hosting: preparing deploy directory for upload...

+ hosting: 15 files uploaded successfully

i starting release process (may take several minutes)...

+ Deploy complete!

Project Console: <https://console.firebase.google.com/project/firebase-veigapp/overview>

Hosting URL: <https://veigapp.firebaseio.com>





- La URL de nuestra aplicación será: <https://veigapp.firebaseio.com>
- La URL del Dashboard será: <https://console.firebase.google.com/project/firebase-veigapp/overview>

- Aspecto de la aplicación en el Dashboard.





dominio

Dominio	Tipo
veigapp.firebaseio.com	Predeterminado

Historial de implementaciones

Estado	Hora	Implementación
 Actual	6 abr. 2017 14:18	 veiga@iessanclemente.net -A8eeX
 Implementado	6 abr. 2017 14:14	 veiga@iessanclemente.net rMk2kY

- Se podrán realizar todos los envíos que queramos con **firebase deploy** y desde el **DashBoard** podremos **revertir** o **eliminar** la **versión** que queramos.



 Implementado	6 abr. 2017 14:18		veiga@iessanclemente.net -A8eeX
 Implementado	6 abr. 2017 14:14		veiga@iessanclemente.net rMk2kY

Desactivación del Hosting en Firebase



- Si queremos **desactivar el hosting** podremos hacerlo con **firebase hosting:disable**

```
D:\xampp\htdocs\web\_dev\www.veiga.local\firebasegoogle>firebase hosting:disable
? Are you sure you want to disable Firebase Hosting?
  This will immediately make your site inaccessible! Yes
+ Hosting has been disabled for firebase-veigapp. Deploy a new version to re-enable.
```

```
D:\xampp\htdocs\web\_dev\www.veiga.local\firebasegoogle>
```

Historial de implementaciones		
Estado	Hora	Implementación
 Inhabilitada	6 abr. 2017 14:20	 veiga@iessanclemente.net L9KFx4

- Para volver a **reactivar el hosting**, simplemente realizaremos un nuevo **firebase deploy**

Historial de implementaciones		
Estado	Hora	Implementación
 Actual	6 abr. 2017 14:23	 veiga@iessanclemente.net sl6rjy
 Inhabilitada	6 abr. 2017 14:20	 veiga@iessanclemente.net L9KFx4

Veiga (discusión) 14:27 6 abr 2017 (CEST)