Informe SIPC: Programación web

Adán de la Rosa Lugo

Juan Siverio Rojas

@2020

# Introducción

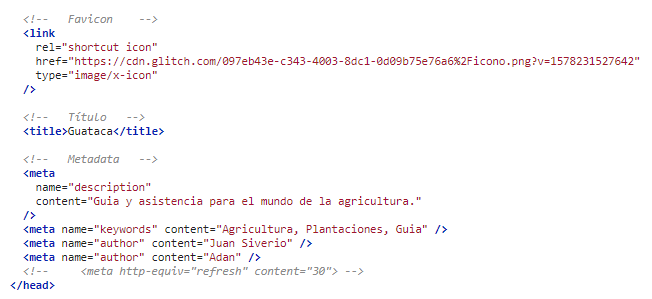
Para la realización de este proyecto, hemos decidido realizar una página web sobre guias de cultivo. En la página se da información relacionada con los cuidados como clima, plagas y terreno. Nuestra Web consta de 2 archivos html más un tercero creado con XALAN a partir de un XML y un XSL, un archivo css y dos archivos javascript. Destacar que hemos utilizado la herramienta de Glitch.com, un entorno de desarrollo de colaborativo donde podíamos trabajar de manera simultánea.

# Html5

**Cabecera**

En la cabecera de los html, incluimos el titulo de nuestra página, metadatos como quienes son los autores, descripción, palabras clave. Además del icono de la pagina, links a nuestros hojas de estilo CSS y Javascript y los enlaces referentes a Bootstrap, firebase y fuentes de texto.





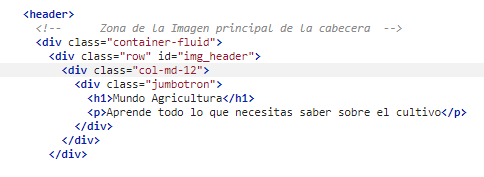
**Cuerpo**

Para esta práctica utilizamos algunos atributos como: **src, id, type, class,** entre otros. También hemos utilizado etiquetas **article, nav,h1,h2, div ,li** entre otras.

* **Titulo del Proyecto, mensaje principal y submensajes :** Hemos incluido un título para el proyecto**: Guataca,** haciendo referencia a la empresa ficticia que hemos inventado. Además, del mensaje principal y submensaje “Mundo agricultura” … en el header de la página aplicado sobre un elemento jumbotron de Bootstrap con una imagen de fondo.

****

En código desde la línea 79 a la 88 en el index.html:

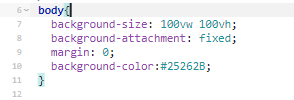


* **Formulario de Suscripción:** Formulario con dos entradas de texto para los campos de Email y Contraseña. Se comprueba que el formato de entrada es el correcto para ambos campos y se oculta la entrada escrita por el usuario en el campo de la contraseña.

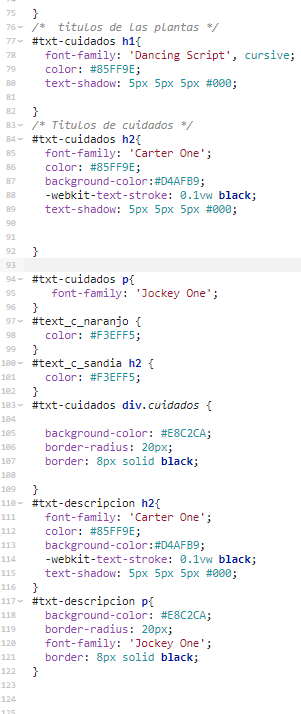


# CSS

* Hemos usado una hoja de estilos para la página que hemos llamado style.css. hemos utilizado una imagen de fondo para el jumbotron
* En el body utilizamos un color oscuro de fondo



* Los títulos tendrán diferentes tipografías, algunas de las que hemos usado hemos tenido que añadir el link en la cabecera. Personalizamos los div para agregarle diferentes colores y que fuiese mas agradable visualmente.



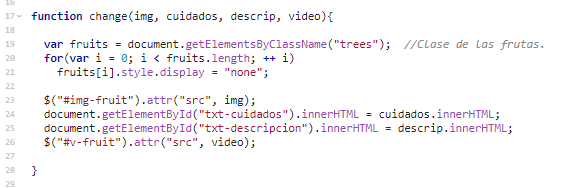
Quedando como resultado:



# Javascript

Para la parte de Javascript hemos hecho que la página cambie dinámicamente cuando el usuario selecciona una de las opciones del menú. Primero ocultamos todo el contenido de la página con todos los textos y mostramos solo los relacionados con la primera planta, naranjas.

A continuación, cuando el usuario hace click en la barra de menú, se carga en la página el contenido deseado con la función change

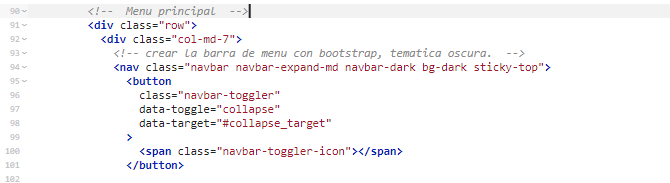


Con respecto a la validación del formato de la cuenta de correo especificado en la rúbrica de la práctica, se ha realizado una función en javascript que mediante una expresión regular comprueba que haya mínimo de dos caracteres antes y después de la arroba y del punto tal y como se solicitaba. Esta función funciona mediante un evento y escribe la cuenta de correo en rojo mientras no cumpla los requisitos, momento en el que cobrará su color negro normal.

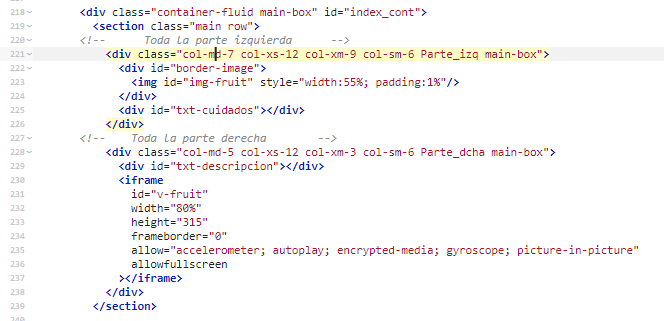


# Bootstrap

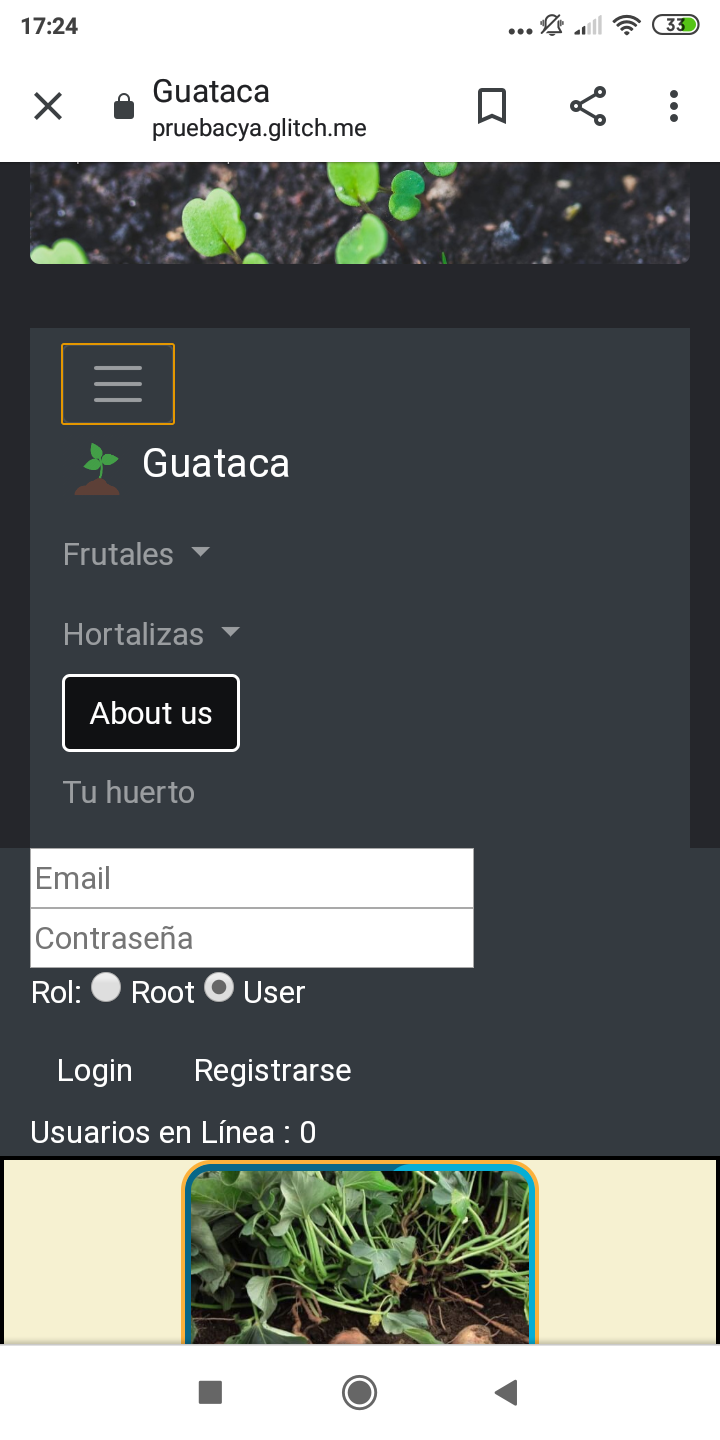
Hemos utilizado varios elementos de Bootstrap como el ya mencionado jumbotron o el elemento navbar. El cual nos ha permitido definir una barra de menú responsiva que, además, comprime todo el contenido del menú a un botón desplegable cuando la pagina es abierta en dispositivos con pantallas de móvil.



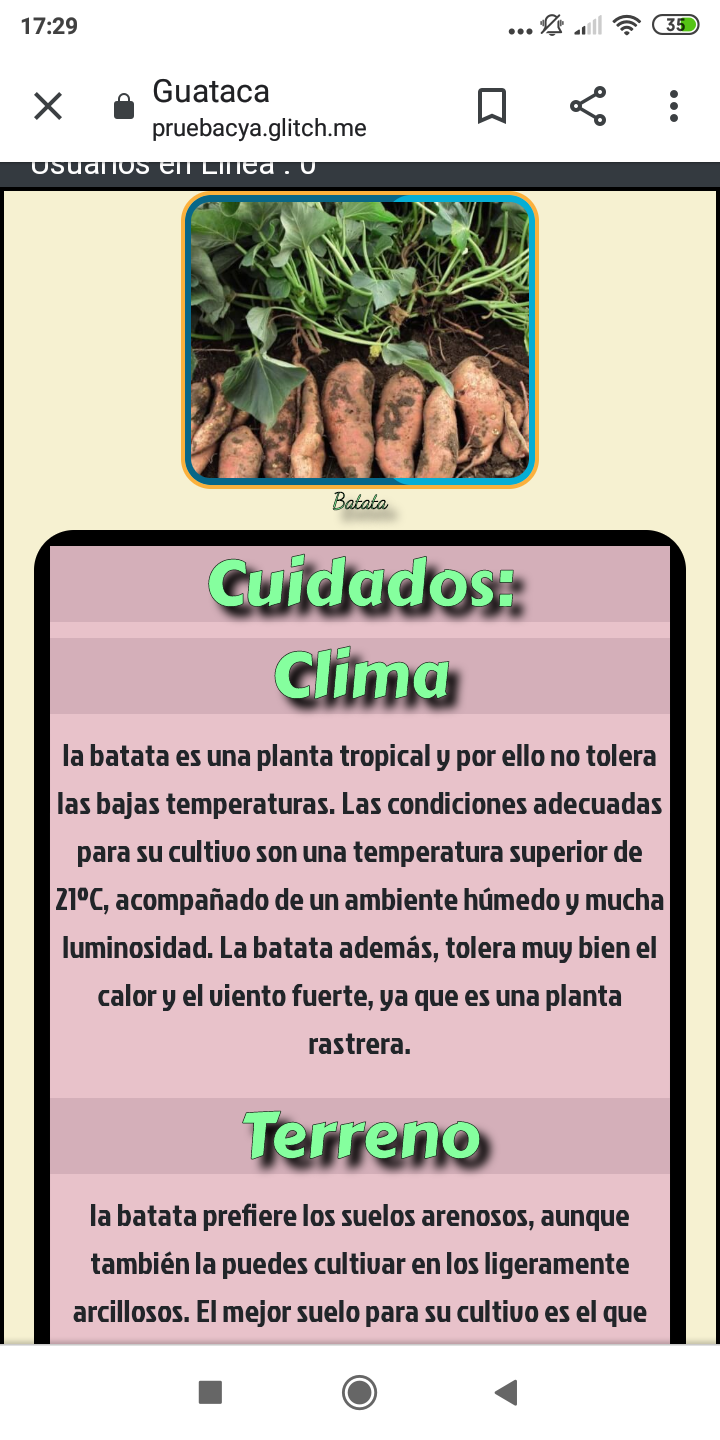
Además, hemos definido un grid para el contenido principal de la página, la cual esta dividida en dos partes que hemos llamado parte izquierda y parte derecha.



Viéndose el menú en un dispositivo móvil de la siguiente manera:



Y el contenido de la página:

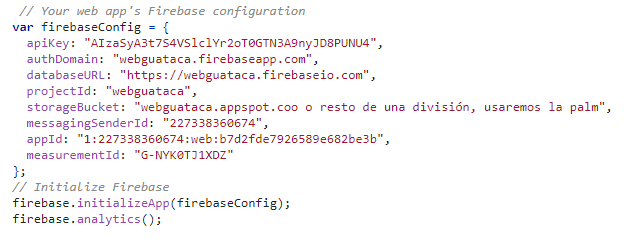


Mientras que en la web se vería:

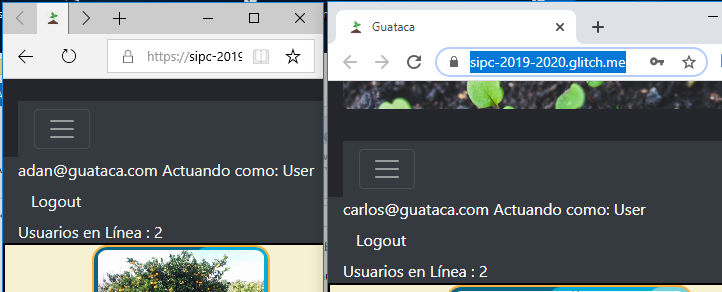


# Firebase

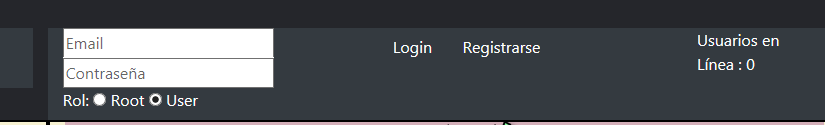
Con respecto a firebase, se han utilizado las funciones de **base de datos y autenticación, además de la función de analytics**. La cabecera firebase obtenida para nuestro proyecto es la siguiente:



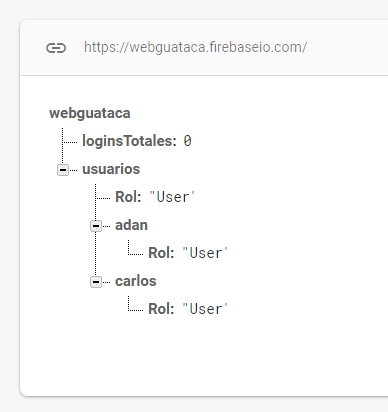
* La página registrar usuarios nuevos para poder después hacer login con ellos.
* Al crear la cuenta de usuario se puede especificar el rol que tendrá, que podrá ser o ROOT o USER.
* La página muestra en todo momento el número de usuarios en línea que haya y esto se hace usando las funciones de actualización de tiempo real de la base de datos firebase.
* También muestra el rol con que se había creado la cuenta de usuario. Por ejemplo, en la siguiente imagen se aprecia como están logueados simultáneamente el usuario [adan@guataca.com](mailto:adan@guataca.com) y [carlos@guataca.com](mailto:carlos@guataca.com) ambos con rol de USER, uno desde google Chrome y otro desde Microsoft Edge .



* El botón de logout, solo está disponible si el usuario está logueado y si no lo está desaparece y aparecen los de login o registrarse(crear cuenta nueva).



La base de datos en firebase es la siguiente:



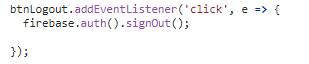
* Para crear una cuenta nueva utilizamos el método **firebase.auth.createUserWithEmailAndPassword(email,pass)** de la api Firebase asociada al botón Registrarse.



* Para hacer login **firebase.auth.signInWithEmailAndPassword(email,pass)** asociada a login.



* Para desconectar:



* Un método importate que usamos es firebase .auth().onAuthStateChanged que ejecutará una función cada vez que cambio el estado del usuario. Esto nos permite, modificar cualquier contenido de la web, actualizar variables,etc.

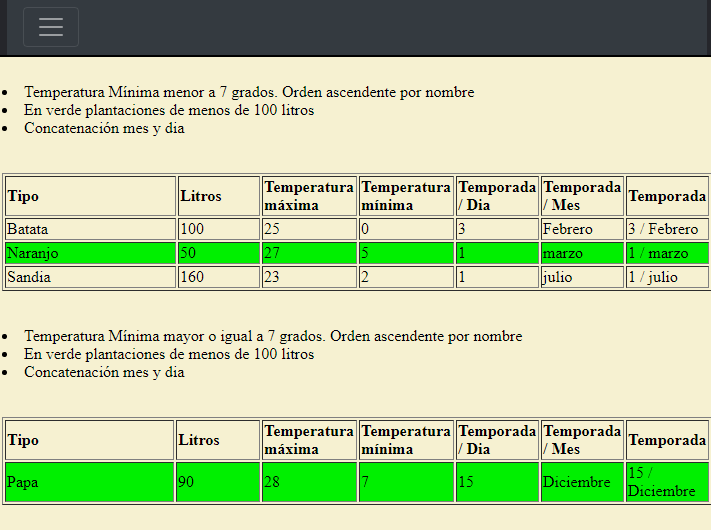


# xml / xsl

El enlace “Tu huerto” del menú de opciones, nos abrirá un html que lleva embebido otro html creado con XALAN a partir de un XML y un XSL.

Para probar diferentes funcionalidades:

* Se ha discriminado la base de datos “XML” creando dos tablas diferentes en base a la temperatura mínimo.
* También se han ordenado por orden ascendente basándose en el nombre.
* Se pinta en verde la fila cuyo valor litros sea menor de 100.
* Y se han concatenado campos día y mes de la temporada.



Todo ello realizado con código XSL.