



**IES LOS SAUCES
BENAVENTE (ZAMORA)**



**TITULO GRADO SUPERIOR DE
DESARROLLO DE APLICACIONES WEB**

DEPARTAMENTO: INFORMÁTICA

TITULO PROYECTO: Autenticación de usuario con Node.js y Firebase

AUTOR: Luis Puente Fernández

CURSO: 2020/21

PERIODO: JUNIO

TIPO DE PROYECTO: IIED

CÓDIGO DE PROYECTO: I083

TUTORÍA COLECTIVA:

TUTORÍA INDIVIDUAL:

Autenticación de usuarios con Node.js y Firebase

Usando Typescript y Express js.



Luis Puente Fernández

Resumen


El proyecto consiste en un CRUD, del inglés, “Crear, Leer, Actualizar, Borrar”. Permitiendo registrarse, iniciar sesión, actualizar datos y borrar la cuenta.

Esta desarrollado con Node.js, Typescript, Firebase y TailwindCSS.

- **Node.js**, permite la ejecución de JavaScript en el lado del servidor, también se hace uso de sus módulos y **NPM** para instalarlos, los módulos más importantes son:
 - **Express**: Permitirá la creación del servidor con JavaScript. Así como gestionar las peticiones a este.
 - **EJS**: Permite, que, mediante código embebido, añadir lógica al HTML.
 - **Crypto-js**: Permite encriptar las contraseñas mediante varias funciones hash.
- **Typescript**, lenguaje de programación que extiende a JavaScript, permite añadir tipado, interfaces... La función de Typescript es hacer más entendible y ordenada la programación, pues una vez finalizado, el código fuente, será convertido, a JavaScript
- **Firebase**, plataforma de desarrollo de Google, entre sus múltiples funciones se encuentra, la **Cloud Firestore**, una base de datos no relacional (NoSQL) para su integración con Node.js es necesario un módulo llamado **firebase-admin**
- **TailWindCSS**, CSS framework, aunque no es necesario descargarlo, con un enlace al CDN funciona, descargarlo te permite ampliarlo o modificar ciertos valores que vienen por defecto.

Índice general

Objetivos.....	1
Estudio(s).....	1
Caso práctico: Análisis y diseño	1
Caso práctico: Implementación.....	2
Conclusiones y trabajo futuro	2
Webgráficas y referencias	2

	CÓDIGO DEL PROYECTO: i1083
	TÍTULO: Autenticación de usuarios con Node.js y Firebase
	AUTOR: Luis Puente Fernández

OBJETIVOS

Crear una aplicación web que al acceder a ella muestre una página principal, en la página estará la opción de registrarse e iniciar sesión, un usuario registrado podrá, cambiar su contraseña, editar su perfil (solo la descripción) y borrar la cuenta.


Como objetivo adicional sería que haya distintos tipos de usuarios, en el que el más básico sería el “usuario”, tipo por defecto para todos los que se registren y el que tenga todos los permisos sería el “Admin”, el cual sería capaz de crear usuarios, editarlos y cambiar su tipo y contraseña, borrarlos y mostrar un listado de todos los usuarios. También usuarios intermedios, con más o menos permisos, y ciertos permisos limitados, por ejemplo, crear o modificar usuarios del mismo nivel o inferior.

ESTUDIO(S)

Presentación

Nodejs es un entorno de ejecución de JavaScript multiplataforma, asíncrono y orientado a eventos. Está basado en el motor V8 de Google. Algunos aspectos, como su gran optimización (por su compilación en tiempo de ejecución, JIT) y eficiencia, su alta estabilidad o el uso de un solo lenguaje tanto para el servidor como para el cliente, le han dado una gran popularidad.

Firebase, es una plataforma para el desarrollo de aplicaciones web y móviles. Dispone de varias características, como un servicio de hosting, dos servicios de bases de datos, un servicio de autenticación, entre otros.

	CÓDIGO DEL PROYECTO: i1083
	TÍTULO: Autenticación de usuarios con Node.js y Firebase
	AUTOR: Luis Puente Fernández

TailWindCSS, librería CSS que permite un desarrollo ágil, basado en clases de utilidad que se pueden aplicar directamente en el código HTML. **Tailwind** no está basado en componentes, sino que provee clases predefinidas a más bajo nivel. Delegando la creación de componentes al usuario, permitiendo que los componentes se adapten al proyecto y evitando páginas con estilos muy parecidos entre sí.


Características

Nodejs:

- Permite llevar JavaScript al servidor, unificando el idioma de servidor y cliente
- Ligero y eficiente en el uso intensivo de datos en tiempo real
- Permite la construcción rápida y escalable de aplicaciones web
- Gran rendimiento, capaz de manejar un gran número de conexiones simultáneas
- Multiplataforma

Express:

- Permite la creación de un servidor de manera rápida
- Es casi un estándar en la mayoría de las aplicaciones web construidas con Node. Lo que permite que haya una extensa documentación en internet.
- Soporte de varios motores de plantillas, como EJS, Pug, hbs, [etc](#)
- Múltiples paquetes de middlewares para abordar casi cualquier problema de desarrollo web. Por ejemplo, para trabajar con cookies, sesiones, etc.

	CÓDIGO DEL PROYECTO: i1083
	TÍTULO: Autenticación de usuarios con Node.js y Firebase
	AUTOR: Luis Puente Fernández

TailwindCSS:

- Permite crear interfaces de manera simple y rápida.
- No impone decisiones de diseño lo que evita una apariencia similar entre las aplicaciones que la usan.
- Permite un mantenimiento y modificación de la web de forma más sencilla

Firestore (Firebase):


- Permite el uso de una base de datos, sin tener que preocuparse por su mantenimiento.
- Permite manejar un gran número de solicitudes, al tener una gran disponibilidad.
- Mediante el uso de reglas permite hacer de forma sencilla los datos accesibles a personas que cumplan con los requisitos de las reglas.
- Dispone de una opción de Google analytics, que permite hacer un seguimiento del uso de la base de datos.

Licencia – precios

A excepción de Firebase y TailwindCSS, todas las demás tecnologías son software libre y sin ningún tipo de coste.

TailwindCSS, es gratis, pero tiene componentes de pago.

Firebase, tampoco ha de suponer ningún coste, pero estará limitados bastantes aspectos como puede ser, en caso de la base de datos Firestore, las peticiones de lectura, escritura, borrado ... Aunque dichas limitaciones son muy pequeñas o incluso insignificantes para pequeños proyectos (la tarifa estándar permite, hasta 50.000 operaciones, de cualquier tipo, diarias). Los planes son ampliables y la propia

	CÓDIGO DEL PROYECTO: i1083
	TÍTULO: Autenticación de usuarios con Node.js y Firebase
	AUTOR: Luis Puente Fernández

página de Firebase proporciona una calculadora en la que te muestra en precio según lo parámetros que introduzcas, por ejemplo, más peticiones diarias o más almacenamiento.

Comparativa con otras, grado de implantación


Nodejs, nació con la idea de llevar JavaScript al servidor, aunque no se ha limitado a esto, debido en gran parte a su desarrollo. Las principales alternativas a Nodejs (enfocado a servidor), pueden considerarse, por ende, los lenguajes del servidor, como Java (JSP, Spring, Struts), Python (Django), PHP...

Sin embargo, Nodejs, difiere en gran medida de estos. Es un entorno de ejecución capaz de interpretar las peticiones y generar una respuesta según sea la petición. Además de no necesitar ningún servidor externo. Incluso dichas tecnologías no necesariamente han de ser una alternativa, por ejemplo, Nodejs, mediante un framework, es capaz de ejecutar PHP.

La tecnología más adecuada para ser considerada una alternativa es Deno. Considerado por algunos, el sucesor de Nodejs, nació con la idea de mejorar los fallos de este (su desarrollador es el mismo que el de Nodejs) como un gestor de paquetes propio, soporte por defecto para TypeScript, entre otros.

Ejemplos de aplicaciones desarrolladas que la utilizan

Netflix, Uber, PayPal, eBay, LinkedIn, Walmart, Godaddy. Incluso la Nasa utiliza Nodejs en los trajes espaciales.

	CÓDIGO DEL PROYECTO: i1083
	TÍTULO: Autenticación de usuarios con Node.js y Firebase
	AUTOR: Luis Puente Fernández

CASO PRÁCTICO: ANÁLISIS Y DISEÑO

Entornos de desarrollo y Explotación

Tanto el entorno de desarrollo como el de explotación necesitan únicamente de Nodejs y NPM para funcionar. El SGBD es independiente.


Los componentes de cada entorno son los siguientes.

Entorno de desarrollo:

- **Sistema Operativo**, Windows 10
- **IDE**, Visual Studio Code
- **Navegador**, Google Chrome
- **Control de versiones**, GIT
- **Cliente SSH**, CLI SSH (incluido con Windows 10)
- **Sistema de transferencia de ficheros**: FileZilla Client
- **Otros**, Inkscape (Editor de gráficos vectoriales), Gimp (Editor de imágenes), Microsoft Word, Postman, StartUML

Entorno de Explotación (Servidor local virtualizado):

- **Sistema Operativo**, Ubuntu Server 20.04
- **Control de versiones**, GIT
- **Servidor SSH**, OpenSSH
- **Sistema de transferencia de ficheros**: sftp

	CÓDIGO DEL PROYECTO: i1083
	TÍTULO: Autenticación de usuarios con Node.js y Firebase
	AUTOR: Luis Puente Fernández

A, excepción de NPM y Node, todas las demás aplicaciones son prescindibles, sin embargo, facilitan el desarrollo del código o su paso a explotación.

IDE

El IDE usado es **Visual Studio Code**, la razón es en gran parte a que es un entorno ligero y tiene soporte nativo para Typescript (ambos son de Microsoft).

Las extensiones usadas para **Visual Studio Code**, son:

- Markdown Preview, que permite previsualizar archivos de markdown como el Readme.
- Tailwind CSS IntelliSense, muestra recomendaciones de las clases de tailwindcss
- ESLint, muestra las alertas y errores de ESLint en el propio editor


Lenguajes de programación, lenguajes de marcas utilizados

SGBD

El lenguaje de programación es Typescript, un super conjunto de JavaScript. También se usa JavaScript para la parte del cliente y el renderizado de la vista HTML.

El lenguaje de marcas utilizado es HTML, dentro de las plantillas EJS.

La base de datos es Firestore, de la plataforma Firebase. Es una base de datos no relacional, escalable. Las principales ventajas, es que te permite despreocuparte de

	CÓDIGO DEL PROYECTO: i1083
	TÍTULO: Autenticación de usuarios con Node.js y Firebase
	AUTOR: Luis Puente Fernández

la base de datos y su mantenimiento. Dispone de una garantía de tiempo de actividad extremadamente alta al pertenecer a Google.


También dispone de consultas sin conexión, aunque para esta aplicación, es necesario que el servidor disponga de conexión para recibir las peticiones, por lo que, la única necesidad por la que serían necesarias las consultas sin conexión es que el propio servicio de la base de datos se caiga.

Framework, librerías, ...

Las librerías o dependencias se componen de las dependencias normales y las de desarrollo.

Dependencias normales, necesarias para la ejecución de la aplicación una vez finalizado el desarrollo.

- **Cookie-parser:** middleware encargado de analizar las cookies y convertirlas a un objeto, necesario para el uso de la sesión
- **Express-session:** permite la creación y uso de sesiones de manera sencilla.
- **Crypto-js:** permite usar funciones Hash, las contraseñas se encriptan con una función sha256, de la librería.
- **EJS:** lenguaje de plantillas que permite generar un HTML, a partir de JavaScript embebido. Cuando Express responde a una petición, renderizará estas plantillas, que, según los datos, generaran el HTML. El cliente recibiría el HTML únicamente.
- **Express:** Permite la creación de un servidor, que se encargará de recibir las peticiones y en base a estas, renderizará las plantillas EJS.
- **Firebase-Admin:** librería para trabajar con la base de datos, Firestore.

	CÓDIGO DEL PROYECTO: i1083
	TÍTULO: Autenticación de usuarios con Node.js y Firebase
	AUTOR: Luis Puente Fernández


Dependencias de desarrollo

- **TypeScript:** Superconjunto de JavaScript. Permite el uso de tipos, objetos basados en clases, interfaces, etc. La dependencia corresponde al compilador
- **ESLint:** librería que analiza sintácticamente el código
- **Copyfiles:** permite copiar archivos en función de los parámetros. Necesario para copiar los archivos en la carpeta de salida al compilar
- **Nodemon:** reinicia de forma automática el servidor cada vez que se realizan cambios, evitando tener que reiniciar constantemente
- **Postcss:** permite convertir CSS mediante JavaScript, es una dependencia opcional de tailwindcss
- **Postcss-cli:** interfaz de línea de comandos de postcss
- **Tailwindcss:** librería de css
- **Autoprefixer:** dependencia para tailwindcss, añade prefijos al código css. Como -moz-, -ms-, -webkit-
- **Cssnano:** plugins de Postcss, minifica (elimina todos los espacios para reducir el tamaño del código) el código
- **@types:** archivos de declaración, permite a TypeScript usar paquetes JavaScript. Un archivo de declaración proporciona información sobre los archivos JavaScript.

Análisis y diseño

- Catálogo de requisitos

Usuarios

	CÓDIGO DEL PROYECTO: i1083
	TÍTULO: Autenticación de usuarios con Node.js y Firebase
	AUTOR: Luis Puente Fernández

Persona capaz de usar la aplicación.

- Usuario no registrado

Persona no identificada, capaz de usar la aplicación web

- Usuario registrado

Usuario que se ha registrado en la aplicación con un usuario y contraseña, de Tipo=USUARIO.

- Usuario SUPER_USUARIO

Usuario de tipo especial (SUPER_USUARIO), con capacidad para ver los demás usuarios.

- Usuario SEMI_ADMIN

Usuario con capacidad para crear, editar, y cambiar usuario del mismo nivel o inferior, además de cambiar las contraseñas a niveles más bajos.

- Usuario ADMIN

Usuario con capacidad total, cambiar tipos y contraseñas, editar, buscar, crear y borrar usuarios.

Páginas

Web pública


1. Un usuario no registrado podrá ver la **página de bienvenida**
2. Un usuario no registrado podrá **iniciar sesión**, convirtiéndose en un usuario registrado, con su tipo de usuario especificó. Pudiendo entrar en la parte privada
3. Un usuario no registrado podrá **registrarse**, convirtiéndose en un usuario de tipo “USUARIO”, podrá acceder a la parte privada una vez registrado
4. Un usuario no registrado podrá **acceder al repositorio de la aplicación**

Web privada

5. Un usuario registrado podrá **cerrar sesión**, pasando a ser un usuario no registrado
6. Un usuario registrado podrá **modificar su descripción**
7. Un usuario registrado podrá **cambiar su contraseña**
8. Un usuario registrado **podrá borrar su cuenta** (salvo un usuario administrador en concreto, por temas prácticos)

MTO Usuarios (SUPER USUARIO)

9. El usuario podrá **listar todos los demás usuarios**

	<p style="text-align: right;">CÓDIGO DEL PROYECTO: i1083</p> <p>TÍTULO: Autenticación de usuarios con Node.js y Firebase</p> <p>AUTOR: Luis Puente Fernández</p>
---	---

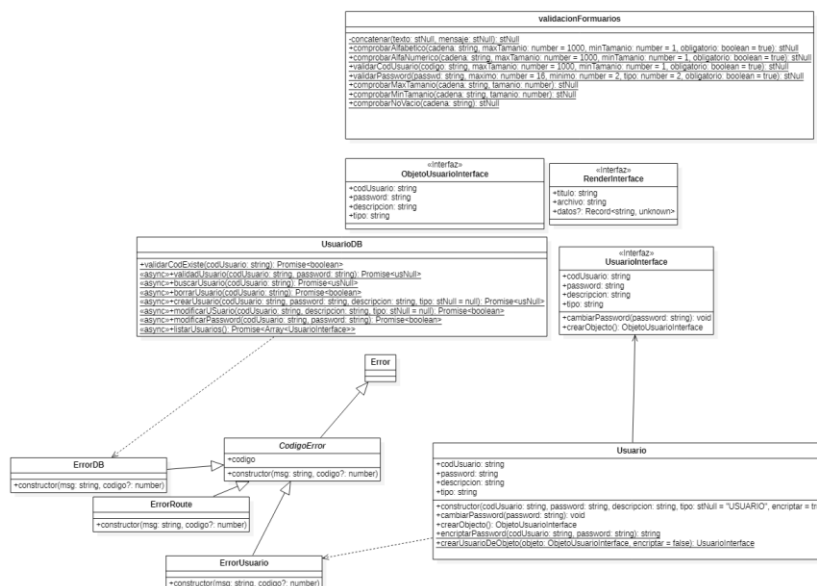
MTO Usuarios (SEMI ADMIN)


- 10.** El usuario podrá **listar todos los demás usuarios**
- 11.** El usuario podrá **modificar la descripción de los demás usuarios** (han de ser del mismo nivel, o un nivel más bajo)
- 12.** El usuario podrá **cambiar la contraseña de los demás usuarios** (de nivel más bajo)
- 13.** El usuario podrá **cambiar el tipo de los usuarios** (el tipo máximo que podrá otorgar será el SEMI_ADMIN)
- 14.** El usuario podrá **crear nuevos usuarios** (el tipo que puede otorgar será el mismo o inferior)

MTO Usuarios (ADMIN)

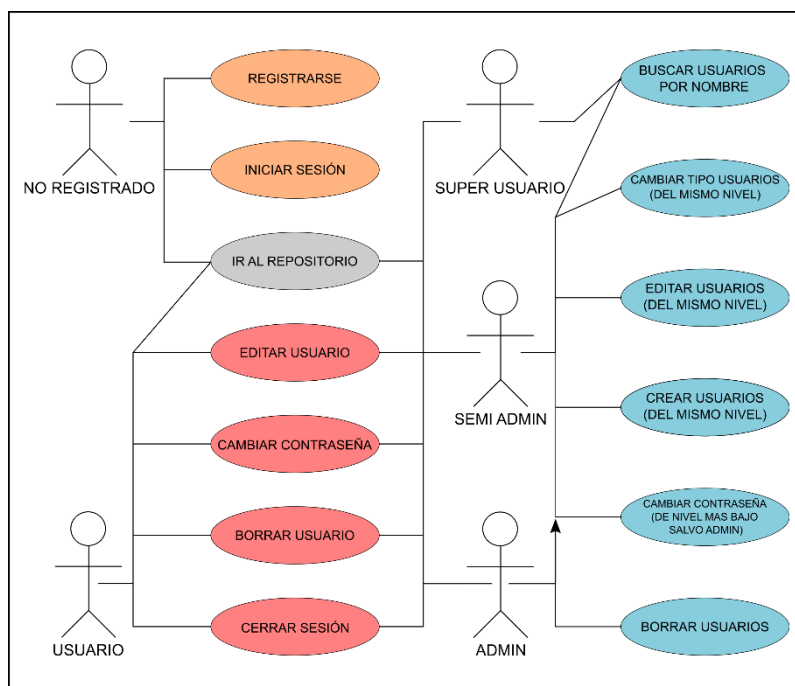
- 15.** El usuario podrá **listar todos los demás usuarios**
- 16.** El usuario podrá **borrar otros usuarios**
- 17.** El usuario podrá **cambiar la contraseña** de otro usuario
- 18.** El usuario podrá **cambiar el tipo** de un usuario
- 19.** El usuario podrá **crear nuevos usuarios** de cualquier tipo
- 20.** El usuario podrá **editar otros usuarios**

• Diagramas de clases



	<p>CÓDIGO DEL PROYECTO: i1083</p> <p>TÍTULO: Autenticación de usuarios con Node.js y Firebase</p> <p>AUTOR: Luis Puente Fernández</p>
---	--

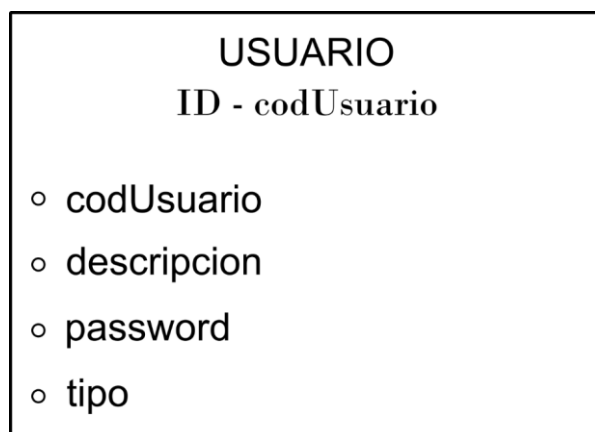
- Diagramas de casos de uso




En la aplicación final, solo está desarrollada la primera parte, parte naranja, gris y roja. También permite que el usuario Admin, buscar usuarios, pudiendo acceder a una lista de usuarios.

- Modelo físico de datos

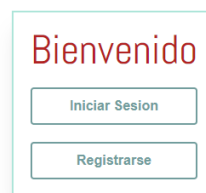
La colección de Usuarios está compuesta por documentos cuyo ID, corresponderá con el codUsuario del documento.



	CÓDIGO DEL PROYECTO: i1083
	TÍTULO: Autenticación de usuarios con Node.js y Firebase
	AUTOR: Luis Puente Fernández

- Diseño de Interfaz


La interfaz se compone de dos tonalidades, **colores rojizos** y **azulados**, y de colores blancos y grisáceos.



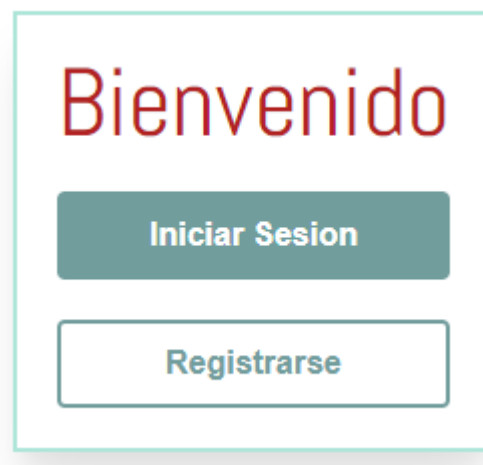
Las fuentes usadas son de Google Fonts, [Abel](#), usada para los títulos, [Hind](#), usada para los botones y [Mukta](#), fuente para todos los demás elementos.

Hay tres tipos de botones, de 2 tamaños (5 en total, un botón solo tiene un tamaño). Los tamaños corresponden al padding. El botón mas grande tiene el doble que el pequeño

- Botones para confirmar:


	<p>CÓDIGO DEL PROYECTO: i1083</p> <p>TÍTULO: Autenticación de usuarios con Node.js y Firebase</p> <p>AUTOR: Luis Puente Fernández</p>
---	--

- De normal el borde y las letras son azulados y el fondo transparente.
- Al pasar sobre estos el fondo pasa a ser azulado y las letras a ser blancas. El borde se vuelve transparente.

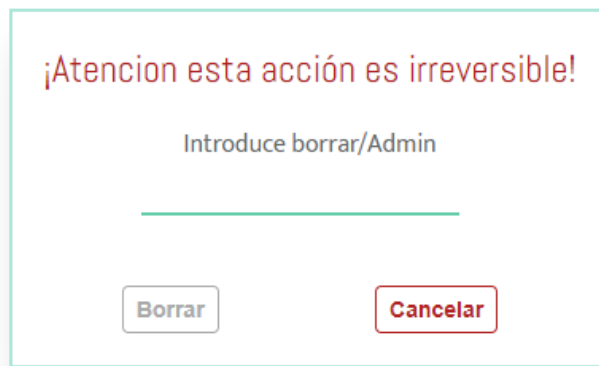


- Botones para cancelar:
 - De normal el borde y las letras son rojizos y el fondo transparente.
 - Al pasar sobre estos el fondo pasa a ser rojizo y las letras a ser blancas. El borde se vuelve transparente.



	CÓDIGO DEL PROYECTO: i1083
	TÍTULO: Autenticación de usuarios con Node.js y Firebase
	AUTOR: Luis Puente Fernández

- Botón desactivado (solo un tamaño):
 - El fondo es transparente, las letras y el borde grisáceos.

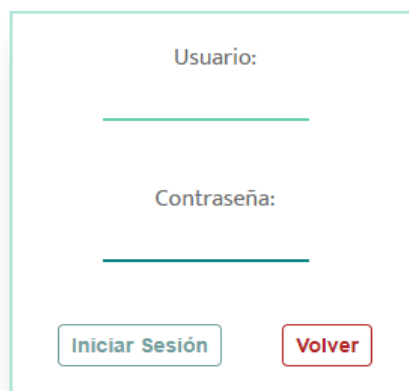


¡Atencion esta acción es irreversible!

Introduce borrar/Admin


Los inputs pueden ser de 3, también hay tres tipos de etiquetas asociadas para estos inputs:

- Input normal:
 - Borde inferior de color turquesa, que se oscurece al pasar o seleccionar el input
 - La etiqueta es de color grisáceo

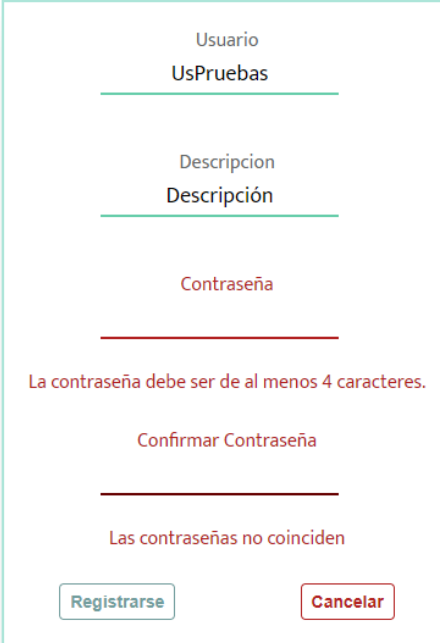


Usuario:

Contraseña:

	CÓDIGO DEL PROYECTO: i1083
	TÍTULO: Autenticación de usuarios con Node.js y Firebase
	AUTOR: Luis Puente Fernández

- Input error:
 - Borde inferior de color rojizo, que se oscurece al pasar o seleccionar el input
 - La etiqueta es también de color rojizo



Usuario
UsPruebas

Descripción
Descripción

Contraseña


La contraseña debe ser de al menos 4 caracteres.

Confirmar Contraseña

Las contraseñas no coinciden

Registrarse Cancelar

- Input desactivado:
 - Borde inferior de color azulado
 - Etiqueta de color gris oscuro

	CÓDIGO DEL PROYECTO: i1083
	TÍTULO: Autenticación de usuarios con Node.js y Firebase
	AUTOR: Luis Puente Fernández

CodUsuario

Admin

Descripcion

Administrador

Tipo de Usuario

ADMIN

Editar

Cancelar

Los títulos pueden ser de dos tamaños, los títulos de bienvenida son más grandes.

Por favor Admin, introduce tu contraseña


Una vez reautenticado, no se volvera a solicitar que vuelves a introducir la contraseña, hasta que canceles o realices el cambio correctamente

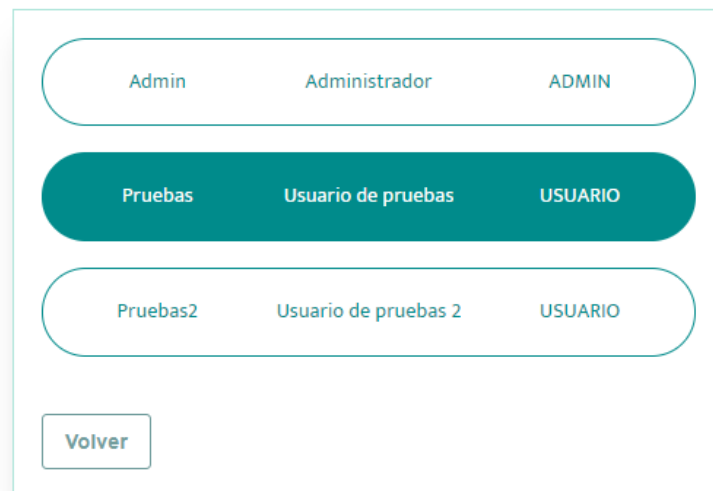
Si sales de alguna forma, que el servidor no contempla, alguien podrá cambiar tus credenciales sin necesidad de autenticarse

Entrar

Cancelar

Por último, está el listado de usuarios para los Administradores. Los colores y el comportamiento son similar a los botones normales, son más redondeados.

	<p>CÓDIGO DEL PROYECTO: i1083</p> <p>TÍTULO: Autenticación de usuarios con Node.js y Firebase</p> <p>AUTOR: Luis Puente Fernández</p>
---	--




El estilo se adapta en cierta medida a las pantallas, salvo algunas excepciones, como un listado de usuarios grande alargará la pantalla en vez de quedarse fijo.

- Estructura de almacenamiento

```

/docs
/src
  /config
  /controller
  /core
  /errors
  /interfaces
  /model
  /public
  /routes
  /scripts
  /views
index.ts

```

	CÓDIGO DEL PROYECTO: i1083
	TÍTULO: Autenticación de usuarios con Node.js y Firebase
	AUTOR: Luis Puente Fernández

- Presupuesto y financiación

El presupuesto para desarrollar una aplicación enfocada al ámbito profesional y comercial varía según los precios, ya antes mencionados y el servidor.

Cualquier servidor Linux (o Windows, aunque menos recomendable por el consumo de recursos), puede correr el programa.

El precio del servidor varía en función de sus recursos. En función de los precios de los planes, que más se ajustan a los requisitos el servidor tendría un coste de más o menos 8,30€/mes (10\$). Precio obtenido de [upcloud](https://www.upcloud.com/).


La aplicación, en el estado actual, ha necesitado de 60 horas. A estas 60 horas hay que añadirle horas adicionales para finalizar la aplicación. Que en total podrían suponer unas 300-350 horas. En función de las horas y considerando el salario mínimo en España (aproximadamente 7,50€ por hora). El costo de desarrollo de la aplicación 2.250-2625€ en total, sin contar posibles retrasos de la aplicación.

CASO PRÁCTICO: IMPLEMENTACIÓN

Repositorio de software

GitHub

Aplicación desarrollada

	CÓDIGO DEL PROYECTO: i1083
	TÍTULO: Autenticación de usuarios con Node.js y Firebase
	AUTOR: Luis Puente Fernández

Aplicación

Explicación del trabajo realizado y Documentación

Mirar **README** del [proyecto](#)

Manual de instalación

Es necesario Nodejs y NPM, además de la clave privada de la base de datos.

Y recomendable el certificado SSL y la clave privada de este.


En caso de tener el código fuente dentro de la carpeta del proyecto ejecutar **npm run build**, que generará en la carpeta raíz del proyecto una carpeta llamada **dist**, con los archivos compilados.

Una vez con los archivos ya compilados, ejecutamos el comando **npm start**, en caso de ejecutarse correctamente no se mostrará nada y la consola se quedara sin permitirte introducir más comandos, esto significa el programa está corriendo, en caso de error, como por ejemplo tener el puerto ocupado, se mostrará el error en la consola.

Para acceder por defecto a la aplicación entra en: <https://localhost:3000/>

CONCLUSIONES Y TRABAJO FUTURO

Si bien la aplicación desarrollada no ha llegado al punto que me hubiera gustado, con el tema de permisos y tipos de usuario. El estudio sobre las tecnologías, y el objetivo en el uso de estas, si ha logrado su objetivo.

	CÓDIGO DEL PROYECTO: i1083
	TÍTULO: Autenticación de usuarios con Node.js y Firebase
	AUTOR: Luis Puente Fernández

Cosas como el estudio de la contraparte al SQL, el uso de alguna librería css o entender cómo funcionan las peticiones y son procesadas por un servidor de una forma más clara.

Sobre el futuro, en caso de ser posible, profundizar más en tecnologías parecidas como Deno o relacionadas con la web como lo son WebAssembly y Assemblyscript.

WEBGRAFÍAS Y REFERENCIAS

- [Documentación EJS](#)
- [Documentación Express](#)
- [Documentación TypeScript](#)
- [Documentación Node](#)
- [Documentación Firebase](#)
- [Documentación TailwindCSS](#)
- [Documentación ESLint](#)
- [Documentación Copyfiles](#)
- [Developer Mozilla](#)