

An abstract network graphic in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles in shades of green and grey, while the lines are thin and grey, creating a dense, organic structure.

Firebase

Firestore

{desafío}
latam_

An abstract network graphic in the bottom-right corner, similar to the one in the top-left. It consists of a web of interconnected nodes and lines, with nodes in green and grey and lines in grey, forming a complex, organic pattern.

Implementar una aplicación web que utilice el servicio de Firebase para resolver un problema planteado.

- Unidad 1: Vue Router
- Unidad 2: Vuex
- Unidad 3: Firebase
- Unidad 4: Pruebas Unitarias y end-to-end en un entorno Vue



Te encuentras
aquí



¿Qué aprenderás en esta sesión?

- *Implementa una aplicación web que utilice el servicio de autenticación con Firebase para resolver un problema planteado.*

¿Para qué utilizamos el
SDK que provee
Firebase?



¿El SDK de Firebase es
genérico en un proyecto?



¿Cuál es la desventaja de trabajar con el modo prueba en Firebase?



¿A qué se refieren las
colecciones en
Firestore?



Contexto antes de iniciar

En esta sesión seguiremos trabajando con Firebase, Firestore y añadiremos una capa a nuestros proyectos que consiste en generar un proceso de autenticación con Firebase.

Firebase Auth es un sistema web bastante robusto creado por Google para la integración de un sistema de usuarios en nuestras aplicaciones.



/* Firebase Auth */

Ejercicio guiado: "Autenticando usuarios con Firebase y Vue JS"



Autenticando usuarios con Firebase y Vue JS

A continuación, realizaremos un ejercicio en el cual integraremos los servicios de autenticación disponibles en Firebase. Para lograrlo, utilizaremos el conjunto de funciones que provee Firebase.

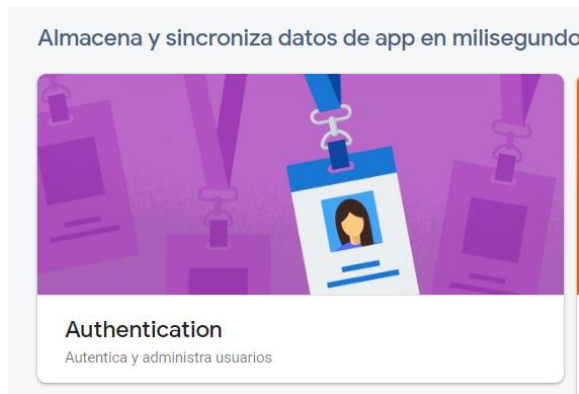
Nos enfocaremos en crear una aplicación sencilla que permite crear usuarios a través de un formulario y comprobar en la consola de Firebase que el ingreso de los datos esté funcionando.

Sigue los pasos...



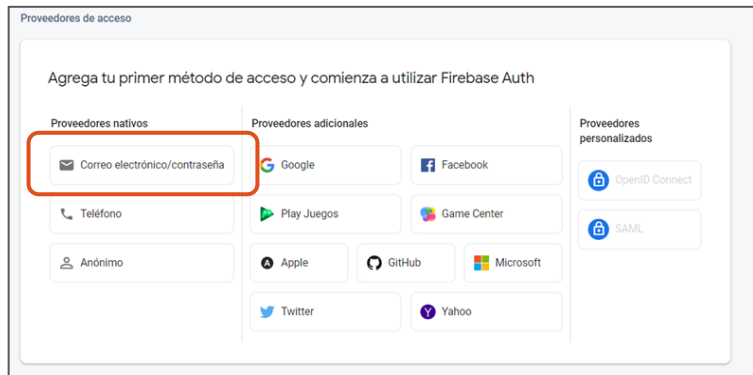
Sigue los pasos...

- **Paso 1:** Creamos un proyecto en Firebase y le definimos como nombre firebase-auth.
- **Paso 2:** Una vez creado el proyecto, nos dirigimos a la sección “Authentication” y damos click en comenzar



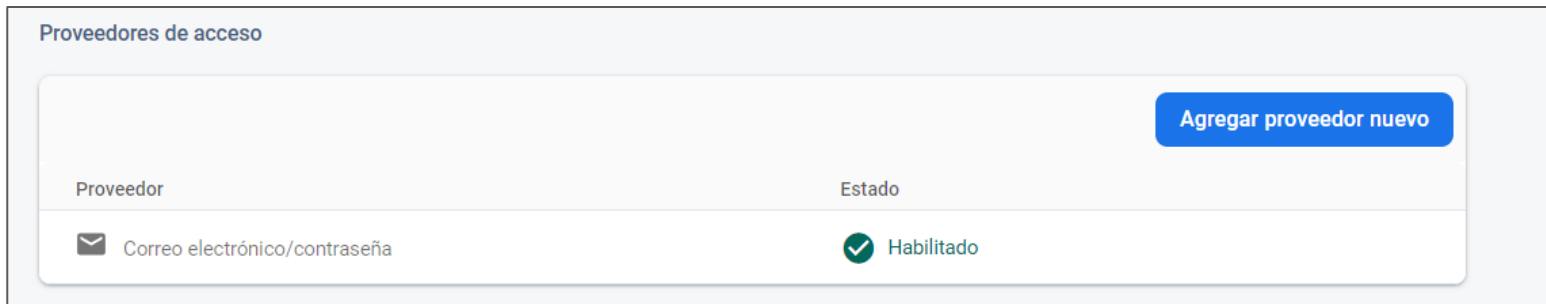
Sigue los pasos...

- **Paso 3:** Ahora, debemos seleccionar el medio por el cual haremos la autenticación, supongamos que queremos hacerla vía correo electrónico y contraseña.



Sigue los pasos...

- **Paso 4:** Si todo resulta de manera correcta, veremos el siguiente mensaje en la consola de Firebase.



Luego de ver este mensaje procedemos a crear la aplicación de Vue JS y habilitamos en los presets la presencia de Vue Router.

Sigue los pasos...

- **Paso 5:** Luego de crear la aplicación de Vue JS con vue router como dependencia, procedemos a conectar el proyecto de Firebase con el SDK. La configuración del SDK lo agregamos en un archivo en el directorio `/src` con el nombre `firebaseconfig.js`.

Recordemos que en la aplicación de Vue debemos correr el comando:
`npm install firebase`



Sigue los pasos...

- **Paso 6:** Creamos un archivo llamado `auth.js` en el directorio `/src` y añadiremos las siguientes líneas de código.

```
import {
  getAuth,
  createUserWithEmailAndPassword,
  signInWithEmailAndPassword,
  onAuthStateChanged }
from 'firebase/auth';

const auth = getAuth(firebaseApp);

export { auth,
  createUserWithEmailAndPassword,
  signInWithEmailAndPassword,
  onAuthStateChanged
};
```



Sigue los pasos...

- **Paso 7:** En el template del archivo App.vue generamos el siguiente código.

```
<template>
  <div>
    <h2>Registro</h2>
    <input v-model="email" type="email" placeholder="Correo electrónico">
    <input v-model="password" type="password" placeholder="Contraseña">
    <button @click="register">Registrarse</button>
  </div>
</template>
```



Sigue los pasos...

- **Paso 8:** El script del componente App.vue quedará de la siguiente manera.

```
<script>
import { createUserWithEmailAndPassword, auth } from '@/auth';

export default {
  data() {
    return {
      email: '',
      password: ''
    };
  },
  methods: {
    async register() {
      try {
        const userCredential = await createUserWithEmailAndPassword(auth, this.email, this.password);
        const user = userCredential.user;
        console.log('Usuario registrado:', user);
      } catch (error) {
        console.error('Error al registrar:', error.message);
      }
    }
  }
};
</script>
```

Resultado parcial de ejecución

Hasta este punto, veremos el siguiente resultado en el navegador.



← → ↻ ⓘ localhost:8080

Registro

<input type="text" value="Correo electrónico"/>	<input type="password" value="Contraseña"/>	<input type="button" value="Registrarse"/>
---	---	--

```
Usuario registrado: App.vue:25  
  ▶ UserImpl {providerId: 'firebase', proactiveRefresh: ProactiveRefresh, reloadU  
    serInfo: {...}, reloadListener: null, uid: 'ZKkKuk5gib0s19oM0qseLy0Zu493', ...}  
>
```



Resultado parcial de ejecución

En la consola de Firebase veremos los usuarios registrados.

Authentication					
Users Sign-in method Templates Usage Settings Extensiones NUEVA					
<div>🔍 Buscar por dirección de correo electrónico, número de teléfono o UID de usuario Agregar usuario ↻ ⋮</div>					
Identificador	Proveedores	Fecha de creación	↓	Fecha de acceso	UID de usuario
correo2@correo2.com	✉	18 ago 2023		18 ago 2023	ULM9QMwLk7YWEMf7KKoRGrID...
sdsdss@sdsd.com	✉	18 ago 2023		18 ago 2023	tNaqxpnsU4WekKh7C58steB03va2
correo@corre.com	✉	18 ago 2023		18 ago 2023	ZKkKuk5gibOs19oM0qseLy0Zu493
Filas por página: 50 1 – 3 of 3 < >					

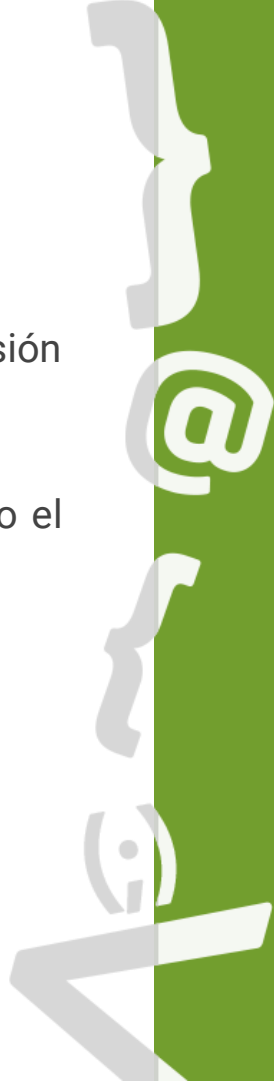
Hasta este punto del
ejercicio ¿Qué te ha
costado más poder
comprender?



Integrando vue router

A continuación, seguiremos trabajando en el mismo ejercicio anterior, en esta ocasión integraremos Vue router.

- **Paso 9:** Creemos una vista llamada `RegistroView.vue`. Esta vista contendrá todo el código que definimos en el archivo `App.vue`.



Códigos de la vista Registro.vue

```
<template>
  <div>
    <h2>Registro</h2>
    <input v-model="email" type="email" placeholder="Correo electrónico">
    <input v-model="password" type="password" placeholder="Contraseña">
    <button @click="register">Registrarse</button>
  </div>
</template>
```

```
<script>
import { createUserWithEmailAndPassword, auth } from '@/auth';

export default {
  data() {
    return {
      email: '',
      password: ''
    };
  },
  methods: {
    async register() {
      try {
        const userCredential = await createUserWithEmailAndPassword(auth, this.email, this.password);
        const user = userCredential.user;
        console.log('Usuario registrado:', user);
      } catch (error) {
        console.error('Error al registrar:', error.message);
      }
    }
  }
};
</script>
```



Códigos del componente App.vue

```
<template>
  <div>
    <router-link to="/">Home</router-link> |
    <router-link to="/registro">Registro</router-link> |
    <router-view />
  </div>
</template>

<script>

export default {
  name: 'App'
}
</script>
```



Integrando vue router

- **Paso 10:** Ahora veremos el código del router en el archivo index.js

```
import { createRouter, createWebHistory } from 'vue-router';

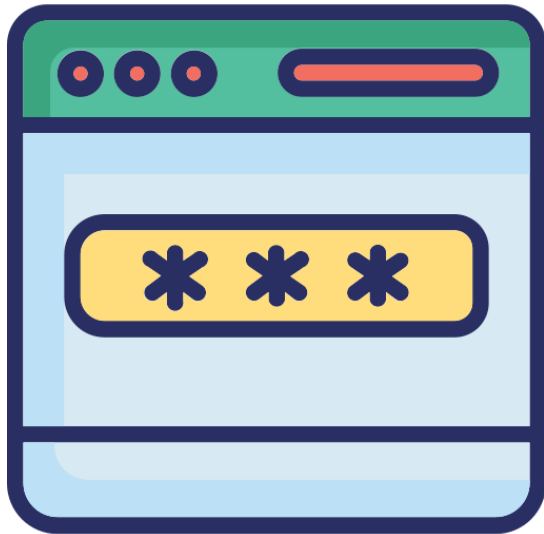
const routes = [
  { path: '/',
    component: () => import(/* webpackChunkName: "homeview" */ '../views/HomeView.vue')
  },
  {
    path: '/registro',
    component: () => import(/* webpackChunkName: "registroview" */ '../views/RegistroView.vue'),
  },
  // ...otras rutas...
];

const router = createRouter({
  history: createWebHistory(),
  routes
});

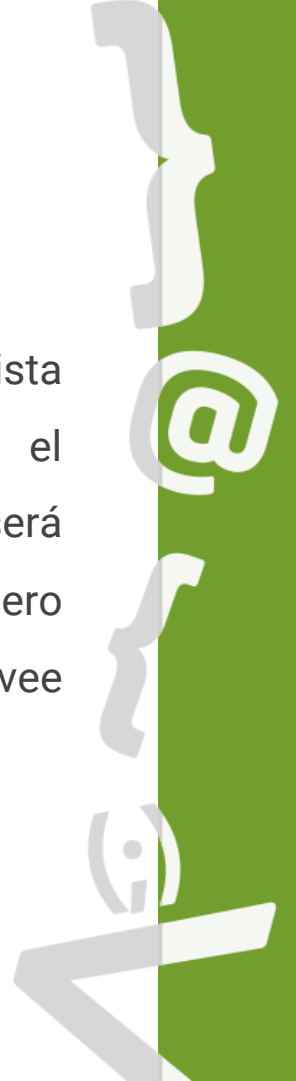
export default router;
```



Inicio de sesión /login



Para el inicio de sesión, crearemos una vista llamada `LoginView.vue`, esta contendrá el formulario de inicio de sesión. Este, será prácticamente igual que el de registro, pero modificaremos las funciones que provee Firebase en esta acción.



Código de la vista LoginView.vue

```
<template>
  <div>
    <h2>Inicio de Sesión</h2>
    <input v-model="email" type="email" placeholder="Correo electrónico">
    <input v-model="password" type="password" placeholder="Contraseña">
    <button @click="singIn">Iniciar sesión</button>

  </div>
</template>
```

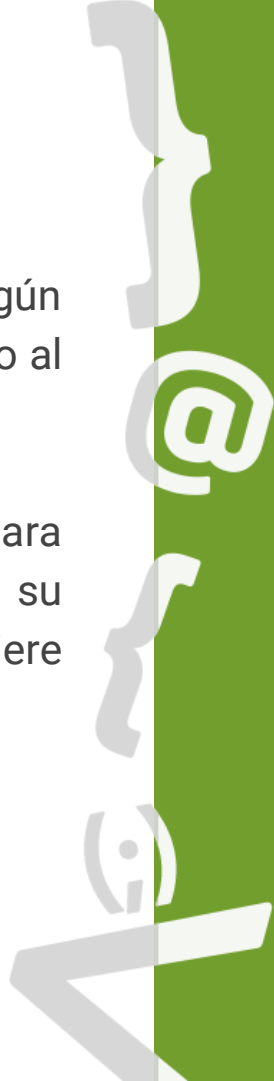
```
<script>
import { signInWithEmailAndPassword, getAuth } from '@/auth';

export default {
  data() {
    return {
      email: '',
      password: ''
    };
  },
  methods: {
    async singIn() {
      const { email, password } = this;
      const auth = getAuth();
      const { user } = await signInWithEmailAndPassword(auth, email, password);
      console.log(user)
    },
  },
};
</script>
```



Restricción de rutas y guardianes

- La restricción de una ruta consiste en la creación de condicionales que según los valores definimos si el usuario procede a su destino deseado o redirigirlo al destino que nosotros definamos.
- Para abordar este concepto modificaremos los parámetros de la ruta raíz para asignarle un metadato que declara que se necesitará de un login para su acceso y en el guardián declararemos que si la ruta destino requiere autenticación, redirigiremos al usuario a la vista “/about”.



Sigamos con los pasos...

- **Paso 11:** Para intencionar la ruta protegida utilizaremos la vista AboutView.vue. Nuestro archivo de rutas quedará de la siguiente manera.

```
{
  path: '/about',
  component: () => import(/* webpackChunkName: "aboutview" */ '../views/AboutView.vue'),
  meta: {
    login: true
  }
},
```

```
router.beforeEach((to, from, next) => {
  const authRequired = to.meta.login
  if (authRequired) {
    next("/about")
    router.push('/')
  }
  else next()
})
```

Guardián beforeEach

Parámetro	Tipo	Definición
to	Object	Objeto de ruta a la cual se quiere acceder.
from	Object	Objeto de la ruta de donde se viene.
next	Function	Función encargada de resolver, es decir, del direccionamiento.

¿Tienes dudas? ¡Este es
el momento de
revisarlas!



Resumen

- Firebase nos provee un servicio de autenticación en cual podemos registrar y dar el servicio de inicio de sesión a los usuarios.
- Además, podemos definir rutas protegidas en nuestras aplicaciones en función de que solo tengan acceso aquellos usuarios que hayan iniciado sesión respectivamente.

A partir de lo visto ¿Qué importancia identifican en implementar un sistema de autenticación?





Próxima sesión...

- *Firebase Hosting*

{desafío}
latam_

*Academia de
talentos digitales*

