



TECNOLÓGICO
NACIONAL DE MÉXICO

TES SFP
TECNOLÓGICO DE ESTUDIOS SUPERIORES
SAN FELIPE DEL PROGRESO

TECNOLÓGICO DE ESTUDIOS SUPERIORES SAN FELIPE
DEL PROGRESO

Ingeniería Informática

CÓMPUTO EN LA NUBE

***Practica 1.1: Firebase - Autenticación, Hosting y Base de
Datos***

Semestre:

Julio - Agosto 2025 (Periodo: 07 de Julio – 30 de Agosto
2025)

Alumna: Geraldin Venegas Margarito

No. Cuenta: 2021330023

Grupo: 902

Docente: Ing. Lizeth Jose Ángeles

San Felipe del Progreso a 09 de Agosto de 2025

Índice del contenido:

Introducción	4
Objetivo de la práctica	4
Desarrollo	5
Requerimientos	5
Crear cuenta	5
Autenticar (correo).....	8
Hosting	10
Base de Datos	15
Análisis de la practica.....	21

Índice de Ilustraciones

Ilustración 1. Ingresar a Firebase y comenzar.	5
Ilustración 2. Comenzar un proyecto en Firebase.	5
Ilustración 3. Asignación de nombre de proyecto.	6
Ilustración 4. Crear proyecto.	6
Ilustración 5. Proyecto creado correctamente.	7
Ilustración 6. Interfaz del proyecto.	7
Ilustración 7. Authentication.	8
Ilustración 8. Comenzar Authentication.	8
Ilustración 9. Autenticar por correo.	9
Ilustración 10. Habilitar por correo y contraseña.	9
Ilustración 11. Autenticación por correo y contraseña habilitada correctamente.	9
Ilustración 12. Ir a configuración del proyecto.	10
Ilustración 13. Crear una app web.	10
Ilustración 14. Asignación de nombre y habilitar la opción de hosting.	10
Ilustración 15. Agregar el SDK de firebase.	11
Ilustración 16. Instalación de Firebase en la máquina física.	11
Ilustración 17. Instalar Firebase CLI.	12
Ilustración 18. Ejecutar el comando de instalación de CLI.	12
Ilustración 19. Comando para iniciar sesión.	12
Ilustración 20. Mensaje de confirmación exitosa de inicio de sesión.	13
Ilustración 21. Inicio de sesión correcto.	13
Ilustración 22. Inicializar proyecto_1.	14
Ilustración 23. Inicializar proyecto_2.	14
Ilustración 24. Subir app al hosting.	15
Ilustración 25. Copiar archivos a la carpeta public.	16
Ilustración 26. Subir app al hosting.	16
Ilustración 27. Abrir app en el navegador.	16
Ilustración 28. Registrar un nuevo usuario.	17
Ilustración 29. Registro de nuevo usuario exitoso.	17
Ilustración 30. Campos validados como obligatorios.	17
Ilustración 31. Intento de reservar laboratorio en una fecha pasada.	18
Ilustración 32. Notificación de error al intentar reservar con fecha pasada.	18
Ilustración 33. Reservar laboratorio.	18
Ilustración 34. Notificación de éxito al reservar.	19
Ilustración 35. Intento de reservar en el mismo horario.	19
Ilustración 36. Notificación de laboratorio ocupado en ese horario.	19
Ilustración 37. Usuario guardado correctamente.	20
Ilustración 38. Reservas guardadas en la BD correctamente.	20

Introducción

El presente proyecto tiene como finalidad llevar a cabo una práctica integral utilizando **Firebase**, una plataforma en la nube desarrollada por Google que proporciona herramientas esenciales para el desarrollo de aplicaciones web y móviles. A lo largo de esta actividad, se tiene previsto realizar la creación de un proyecto dentro de Firebase, implementar un sistema de **autenticación mediante correo electrónico y contraseña**, desplegar una **aplicación web con Firebase Hosting**, y vincular una **base de datos** que permita almacenar información en tiempo real. Esta práctica está orientada a comprender el funcionamiento de los servicios en la nube y su aplicación en proyectos reales, eliminando la necesidad de servidores locales o configuraciones complejas del lado del backend.

Durante el desarrollo de la actividad, se hará uso de herramientas como **Visual Studio Code** para la programación de la interfaz y la lógica de la aplicación, así como **Node.js** y **Firebase CLI** para gestionar la conexión y despliegue del proyecto. Se espera que el resultado final sea una aplicación funcional en la que el usuario pueda registrarse, iniciar sesión y realizar operaciones como la reserva de consultorios o laboratorios, todo ello con respaldo en una base de datos en la nube. Esta práctica permitirá adquirir habilidades técnicas en el uso de plataformas serverless, mejorar la comprensión del flujo de trabajo con Firebase y reforzar buenas prácticas en el desarrollo y despliegue de aplicaciones modernas. Además, brindará una visión más clara sobre la implementación de soluciones accesibles, escalables y seguras desde una sola plataforma centralizada.

Objetivo de la práctica

Implementar una solución básica utilizando **Firebase** que incluya: Creación de cuenta en Firebase, Autenticación mediante correo y contraseña, Configuración del Hosting para desplegar una aplicación y Integración de la base de datos en tiempo real o Firestore.

Desarrollo

Requerimientos

- Tener una cuenta de Google.
- Editor de código (Visual Studio Code recomendado).
- Node.js y npm instalados en el equipo.
- Proyecto web básico para desplegar (HTML/JS o React).
- Acceso a internet.

Crear cuenta

Paso 1: Ingresar a Firebase

- Accede a: <https://firebase.google.com/>
- Desliza hasta encontrar el apartado de “Prueba **Firebase** hoy mismo” y da clic en comenzar, como se muestra en la *ilustración 1*.



Ilustración 1. Ingresar a Firebase y comenzar.

Fuente: Elaboración Propia.

- Una vez dentro crear un nuevo proyecto, seleccionando la opción de “Comenzar con un proyecto de Firebase”, como se muestra en la *ilustración 2*.



Ilustración 2. Comenzar un proyecto en Firebase.

Fuente: Elaboración Propia.

- Asignarle un nombre a tu proyecto y da clic en continuar, como se muestra en la ilustración 3.

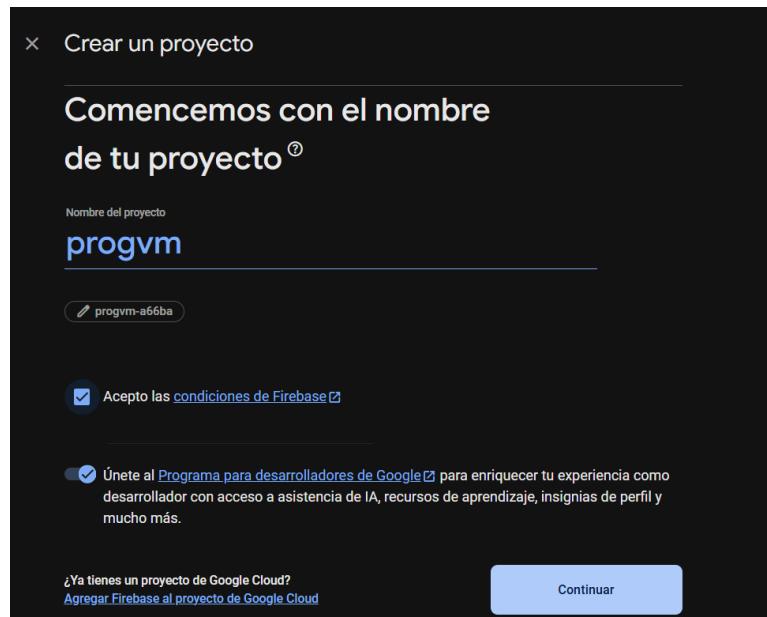


Ilustración 3. Asignación de nombre de proyecto.

Fuente: Elaboración Propia.

- Posteriormente dar clic en continuar hasta el apartado de: Configurar Google Analytics. En este apartado seleccionar la opción de México y crear proyecto, como se muestra en la *ilustración 4*.

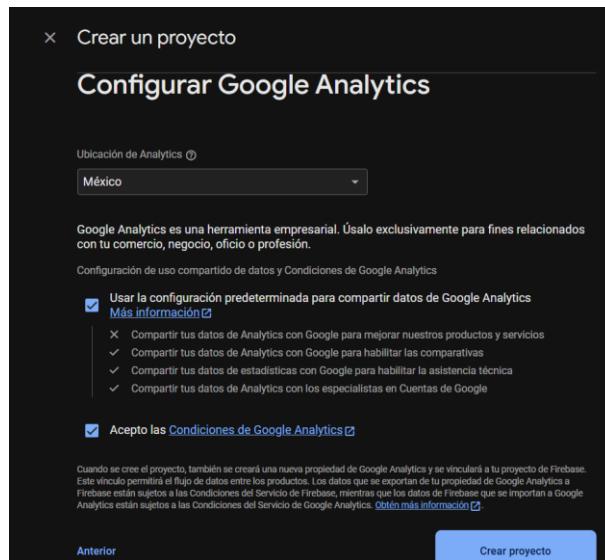


Ilustración 4. Crear proyecto.

Fuente: Elaboración Propia.

- Una vez que se creó el proyecto nos muestra el siguiente mensaje que se observa en la *ilustración 5*.

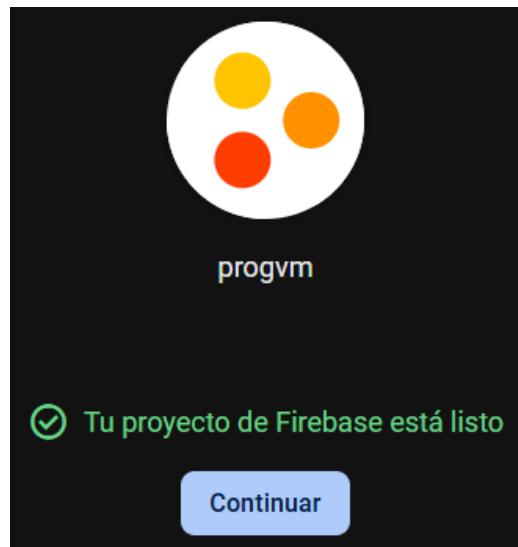


Ilustración 5. Proyecto creado correctamente.

Fuente: Elaboración Propia.

- Da clic en “Comenzar” y te mostrara la siguiente interfaz que se muestra en la *ilustración 6*.

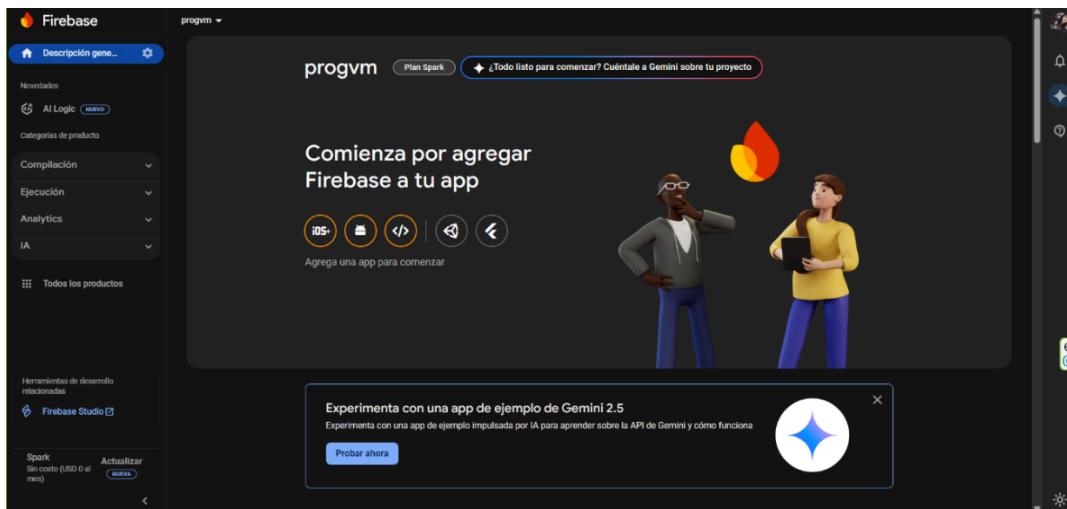


Ilustración 6. Interfaz del proyecto.

Fuente: Elaboración Propia.

Autenticar (correo)

Una vez creado el proyecto procedemos a autenticar nuestro proyecto con nuestro correo, para ello seguimos los siguientes pasos:

- En el menú lateral seleccionamos la opción de “Authenticaction”, como se muestra en la *ilustración 7*.

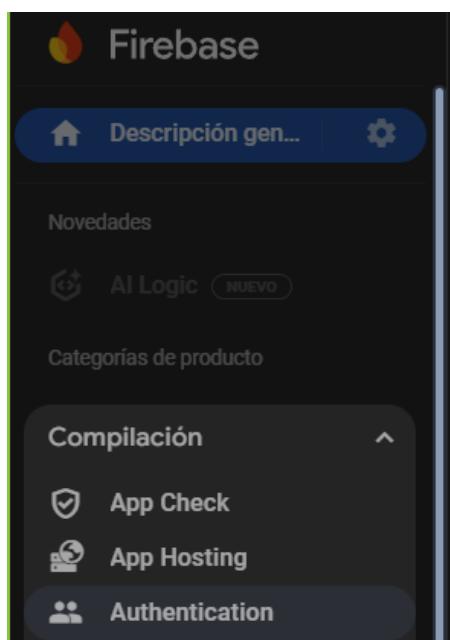


Ilustración 7. Authentication.

Fuente: Elaboración Propia.

- Posteriormente dar clic en “Comenzar”, como se muestra en la *ilustración 8*.

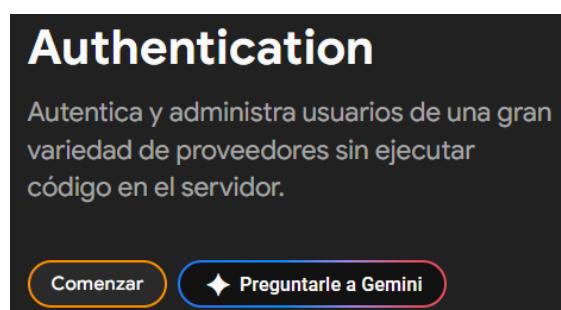


Ilustración 8. Comenzar Authentication.

Fuente: Elaboración Propia.

- Elegimos autenticar por correo, como se muestra en la ilustración 9.

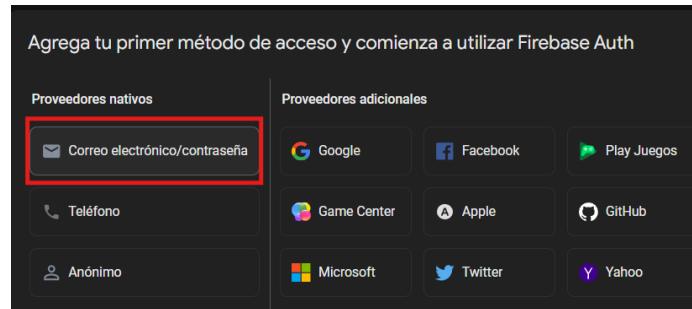


Ilustración 9. Autenticar por correo.

Fuente: Elaboración Propia.

- Habilitar la opción de “Correo electrónico/contraseña” y guardamos, como se muestra en la *ilustración 10*.

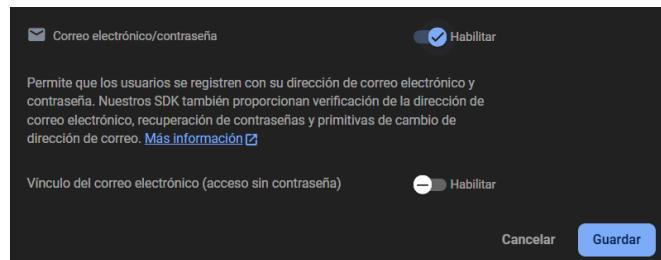


Ilustración 10. Habilitar por correo y contraseña.

Fuente: Elaboración Propia.

- Una vez que se habilito esta opción, nos mostrara lo siguiente que se encuentra habilitado como se observa en la *ilustración 11*.

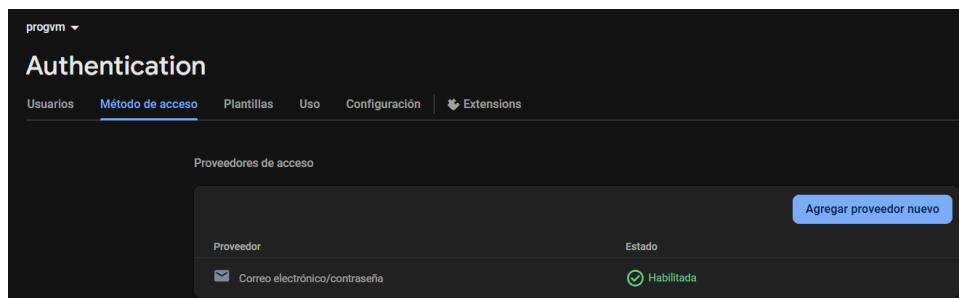


Ilustración 11. Autenticación por correo y contraseña habilitada correctamente.

Fuente: Elaboración Propia.

Hosting

Antes de configurar el hosting vamos a registrar una aplicación web, para ello nos dirigimos a las configuraciones del proyecto, como se muestra en la *ilustración 12*.

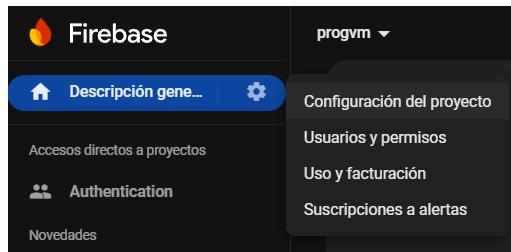


Ilustración 12. Ir a configuración del proyecto.

Fuente: Elaboración Propia.

- En configuraciones deslizamos hasta en la parte de abajo y creamos una app web, para ello seleccionamos el siguiente ícono </>, como se muestra en la *ilustración 13*.

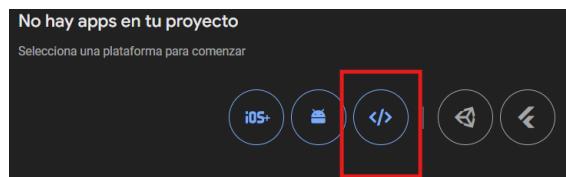


Ilustración 13. Crear una app web.

Fuente: Elaboración Propia.

- Asignamos un nombre a la app y activar **Firebase Hosting**, como se muestra en la *ilustración 14*.

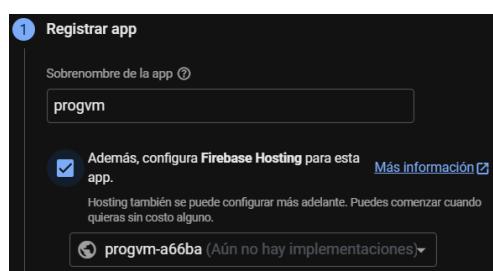


Ilustración 14. Asignación de nombre y habilitar la opción de hosting.

Fuente: Elaboración Propia.

- Dar clic en “Registrar app”, y nos aparecerá lo siguiente que se muestra en la *ilustración 15*.

- ✓ En el cual es necesario instalar firebase en el cmd de la computadora, para ello copiamos el comando que nos manda y lo ejecutamos. Además, es necesario guardar el código que nos arroja, ya que este lo ocuparemos más adelante.

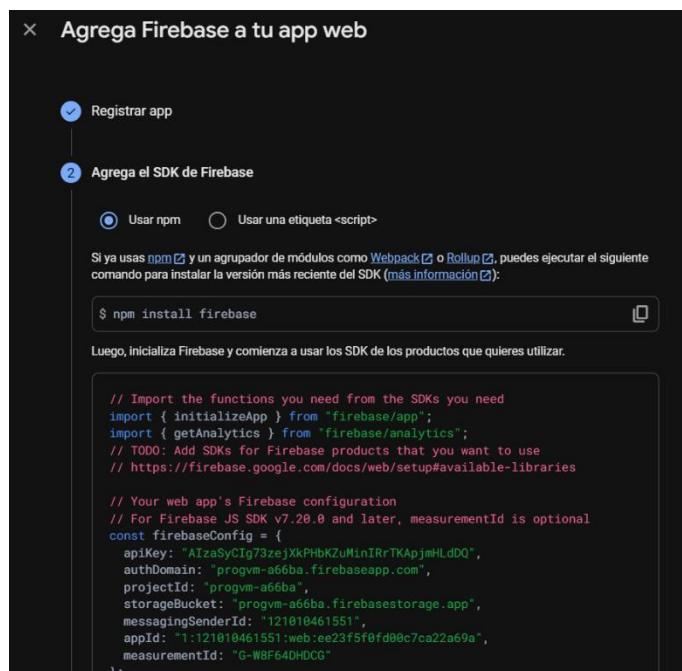


Ilustración 15. Agregar el SDK de firebase.

Fuente: Elaboración Propia.

- ✓ Instalación de Firebase en la máquina física.

```
npm install firebase
```

```
Microsoft Windows [Versión 10.0.26100.4351]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\GERALDINE>npm install firebase
[redacted] / idealTree:GERALDINE: sill idealTree buildDeps
```

Ilustración 16. Instalación de Firebase en la máquina física.

Fuente: Elaboración Propia.

- Posteriormente vamos a instalar el CLI en nuestra máquina, para ello ejecutamos el comando que nos manda automáticamente el cual se observa en la *ilustración 17*.

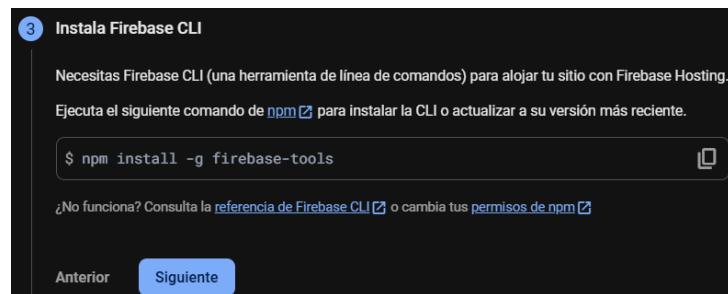


Ilustración 17. Instalar Firebase CLI.

Fuente: Elaboración Propia.

- ✓ Ejecutar el comando en el cmd.

```
C:\Windows\System32>npm install -g firebase-tools
[redacted] | idealTree:gaxios: sill placeDep node_modules/firebase-tools uuid@8.3.2 OK for: firebase-tools@14
```

Ilustración 18. Ejecutar el comando de instalación de CLI.

Fuente: Elaboración Propia.

- Por último, para realizar la implementación del Firebase Hosting, es necesario iniciar sesión con nuestra cuenta. Para ello en el cmd, ejecutamos el siguiente comando que se observa en la *ilustración 19*.

```
C:\Windows\System32>firebase login
i  The Firebase CLI's MCP server feature can optionally make use of Gemini in Firebase. Learn more about Gemini in Firebase and how it uses your data: https://firebase.google.com/docs/gemini-in-firebase#how-gemini-in-firebase-uses-your-data
✓ Enable Gemini in Firebase features? Yes [redacted]

i  Firebase optionally collects CLI and Emulator Suite usage and error reporting information to help improve our products. Data is collected in accordance with Google's privacy policy (https://policies.google.com/privacy) and is not used to identify you.
✓ Allow Firebase to collect CLI and Emulator Suite usage and error reporting information? Yes [redacted]

i  To change your preference at any time, run `firebase logout` and `firebase login` again.

Visit this URL on this device to log in:
https://accounts.google.com/o/oauth2/auth?client_id=563584335869-fgrhgmd47bqneki5j5i8b5pr03ho849e6.apps.googleusercontent.com&scope=email%20openid%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloudplatformprojects.readonly%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Ffirebase%20https%3A%2E%2Fwww.googleapis.com%2Fauth%2Fcloud-platform&response_type=code&state=3370522&redirect_uri=http%3A%2F%2Flocalhost%3A9005

Waiting for authentication...
```

Ilustración 19. Comando para iniciar sesión.

Fuente: Elaboración Propia.

- ✓ Al ejecutar el comando anterior, nos abrirá página donde debemos ingresar nuestro correo, permitir el acceso y aceptar términos y condiciones, al realizarlo correctamente nos aparece el siguiente mensaje.

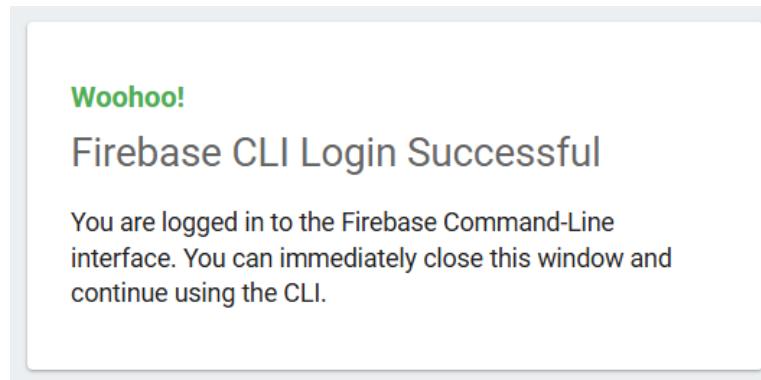


Ilustración 20. Mensaje de confirmación exitosa de inicio de sesión.

Fuente: Elaboración Propia.

- ✓ Y en el cmd nos deberá aparecer nuestro correo, como se muestra en la *ilustración 21*.

A screenshot of a terminal window showing a success message. The text reads: "Waiting for authentication... + Success! Logged in as geralvenmar@gmail.com". The text is in white on a black background.

Ilustración 21. Inicio de sesión correcto.

Fuente: Elaboración Propia.

- Una vez que se inicio sesión vamos a inicializar el proyecto ejecutando el comando de: **firebase init**, como se muestra en la *ilustración 22*.
 - ✓ Seleccionamos la opción de: **Hosting: Configure files for Firebase Hosting...**
 - ✓ Luego **Use an existing project**.

```
C:\Windows\System32>cd C:\Users\GERALDINE\Desktop\Computo en la nube\progvm
C:\Users\GERALDINE\Desktop\Computo en la nube\progvm firebase init

#####
##   # ##   # ##   # ##   # ##   # ##   # #
#####   # #####   # #####   # #####   # #####   # #####
##   # ##   # ##   # ##   # ##   # ##   # ##   # #
##   # ##   # ##   # ##   # ##   # ##   # ##   # #
##   # ##   # ##   # ##   # ##   # ##   # ##   # #

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

C:\Users\GERALDINE\Desktop\Computo en la nube\progvm

✓ Are you ready to proceed? Yes
✓ Which Firebase features do you want to set up for this directory? Press Space to select features, then Enter to confirm your choices. Hosting: Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys

== 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.

? Please select an option: (Use arrow keys)
> Use an existing project
Create a new project
Add Firebase to an existing Google Cloud Platform project
Don't set up a default project
```

Ilustración 22. Inicializar proyecto_1.

Fuente: Elaboración Propia.

- ✓ Seleccionar nuestro proyecto.
 - ✓ Colocar directorio público (public).
 - ✓ No sobreescribas index.html si ya lo tienes.

```
✓ Are you ready to proceed? Yes
✓ Which Firebase features do you want to set up for this directory? Press Space to select features, then Enter to
confirm your choices. Hosting: Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys

== 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.

✓ Please select an option: Use an existing project
✓ Select a default Firebase project for this directory: progvm-a66ba (progvm)
i Using project progvm-a66ba (progvm)

== 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? public
✓ Configure as a single-page app (rewrite all urls to /index.html)? No
✓ Set up automatic builds and deploys with GitHub? No
+ Wrote public/404.html
+ Wrote public/index.html

+ Wrote configuration info to firebase.json
+ Wrote project information to .firebaserc
+ Wrote .gitignore

+ Firebase initialization complete!

C:\Users\GERALDINE\Desktop\Computo en la nube\progvm>
```

Ilustración 23. Inicializar proyecto 2.

Fuente: Elaboración Propia.

- Por último, para subirlo al hosting ejecutamos el siguiente comando: **firebase deploy**, y este nos mandar un link donde se encontrará nuestra app web, como se muestra en la *ilustración 24*.

```
C:\Users\GERALDINE\Desktop\Computo en la nube\progvm>firebase deploy
=== Deploying to 'progvm-a66ba'...

i  deploying hosting
i  hosting[progvm-a66ba]: beginning deploy...
i  hosting[progvm-a66ba]: found 2 files in public
+  hosting[progvm-a66ba]: file upload complete
i  hosting[progvm-a66ba]: finalizing version...
+  hosting[progvm-a66ba]: version finalized
i  hosting[progvm-a66ba]: releasing new version...
+  hosting[progvm-a66ba]: release complete

+ Deploy complete!

Project Console: https://console.firebaseio.google.com/project/progvm-a66ba/overview
Hosting URL: https://progvm-a66ba.web.app

C:\Users\GERALDINE\Desktop\Computo en la nube\progvm>
```

Ilustración 24. Subir app al hosting.

Fuente: Elaboración Propia.

Nota: Al subir la app al hosting ya se estuvo configurando o programando dicha app, el cual se estuvo trabajando en Visual Studio, en el cual al realizar un cambio este se debe subir nuevamente al hosting, por lo tanto, se debe repetir el ultimo paso.

Base de Datos

1. Crear app.
 - Crear una carpeta y abrirla en Visual Studio,
 - Crear y configurar los siguientes archivos:
 - **Index.html:** Interfaz del sitio web.
 - **App.js:** Lógica de la app, es decir código que hace funcional nuestra aplicación.
 - **Style.css:** Estilos visuales de la app.
2. Copiar archivos (index.html, app.js, style.css) a la carpeta public y subirlos al hosting.

- Una vez configurados los archivos anteriores vamos a copiarlos a la carpeta public ubicada en el disco local (C:), como se muestra en la *ilustración 25*.

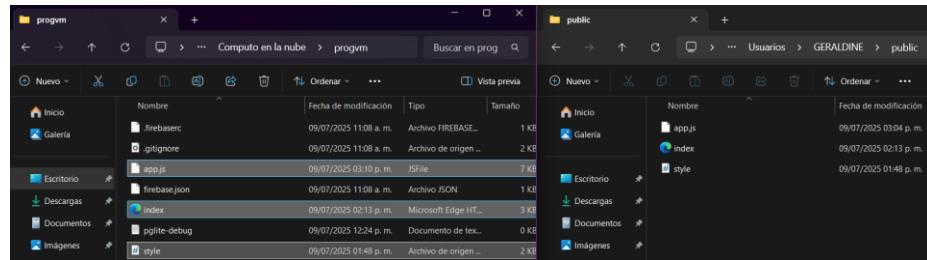


Ilustración 25. Copiar archivos a la carpeta public.

Fuente: Elaboración Propia.

- Subirlo al hosting.

```
+ Deploy complete!
Project Console: https://console.firebaseio.google.com/project/progvm-a66ba/overview
Hosting URL: https://progvm-a66ba.web.app
C:\Users\GERALDINE\Desktop\Computo en la nube\progvm>
```

Ilustración 26. Subir app al hosting.

Fuente: Elaboración Propia.

3. Abrir app web desde la url proporciona.

- Abrir la app en el navegador.

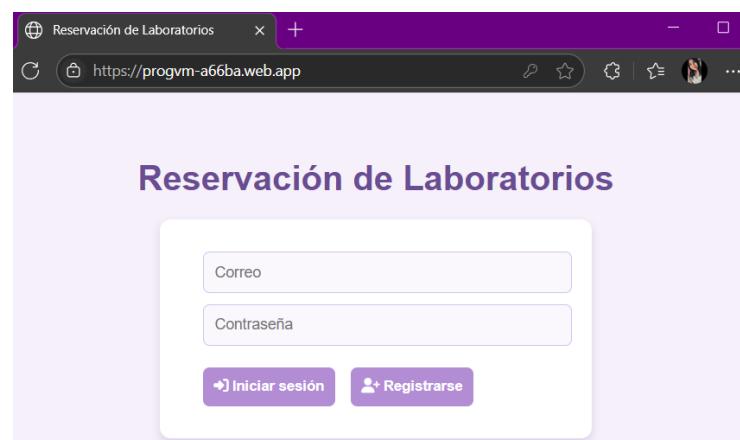


Ilustración 27. Abrir app en el navegador.

Fuente: Elaboración Propia.

4. Registrar datos y realizar reservas de consultorios.

- Registrar usuario.

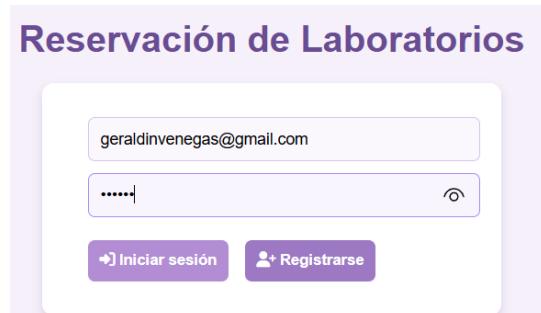


Ilustración 28. Registrar un nuevo usuario.

Fuente: Elaboración Propia.

Notificación de éxito al registrar nuevo usuario.



Registro exitoso

Usuario creado correctamente.

OK

Ilustración 29. Registro de nuevo usuario exitoso.

Fuente: Elaboración Propia.

- Iniciar sesión y realizar reservas.

Notificación al intentar reservar sin ingresar ningún valor.



Campos requeridos

Debes seleccionar la fecha, hora y laboratorio.

OK

Ilustración 30. Campos validados como obligatorios.

Fuente Elaboración Propia.

Ingresamos una fecha pasada.

Reservación de Laboratorios

Usuario: geraldinvenegas@gmail.com

07/07/2025

03:25 p. m.

Lab 1

Reservar

Mis Reservas:

Cerrar sesión

Ilustración 31. Intento de reservar laboratorio en una fecha pasada.

Fuente: Elaboración Propia.

Notificación al intentar reservar en una fecha pasada.

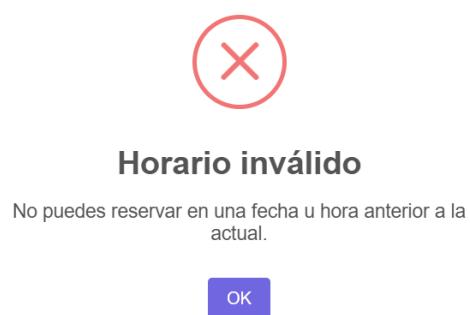


Ilustración 32. Notificación de error al intentar reservar con fecha pasada.

Fuente: Elaboración Propia.

Reservar un laboratorio.

Reservación de Laboratorios

Usuario: geraldinvenegas@gmail.com

09/07/2025

02:30 p. m.

Lab 4

Reservar

Mis Reservas:

Cerrar sesión

Ilustración 33. Reservar laboratorio.

Fuente: Elaboración Propia.

Notificación de éxito.



Reservación registrada

Tu reserva fue guardada correctamente.

OK

Ilustración 34. Notificación de éxito al reservar.

Fuente: Elaboración Propia.

Intento de reservar el mismo laboratorio en el mismo horario.

Usuario: geraldinvenegas@gmail.com

09/07/2025

02:35 p. m.

Lab 4

Mis Reservas:

- 2025-07-09 - 14:30 - Lab 4

Cerrar sesión

Ilustración 35. Intento de reservar en el mismo horario.

Fuente: Elaboración Propia.

Notificación de que el laboratorio se encuentra ocupado durante esa hora.



Horario ocupado

El Lab 4 ya está reservado en un rango de 1 hora dentro del mismo día.

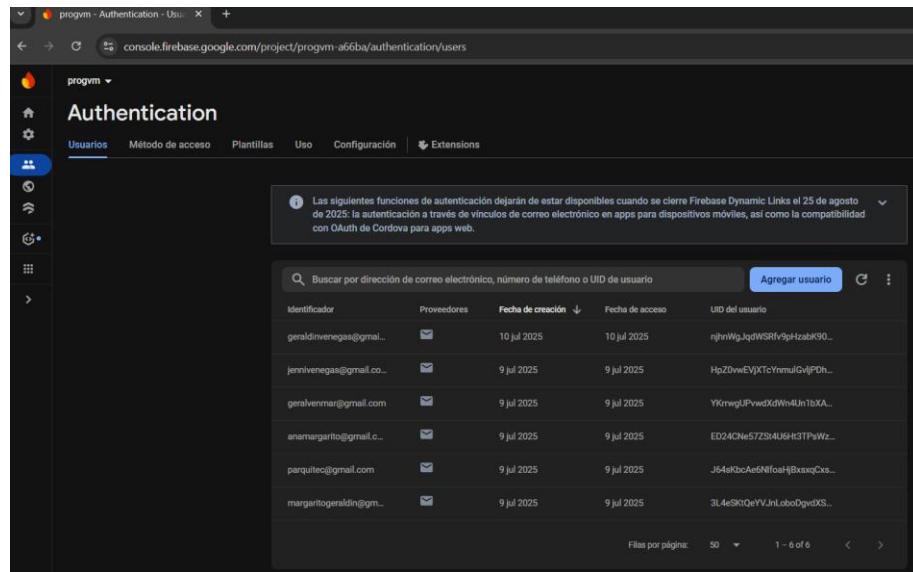
OK

Ilustración 36. Notificación de laboratorio ocupado en ese horario.

Fuente: Elaboración Propia.

5. Verificar que se guarden correctamente en la base.

- Nos dirigimos en Authentication, para verificar que se encuentre nuestro usuario.



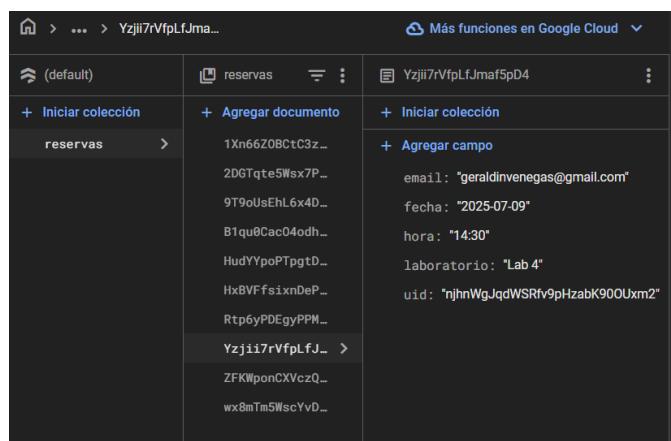
The screenshot shows the Firebase Authentication console for a project named 'progvm'. The 'Users' tab is selected. A message at the top states: 'Las siguientes funciones de autenticación dejarán de estar disponibles cuando se cierre Firebase Dynamic Links el 25 de agosto de 2025: la autenticación a través de vínculos de correo electrónico en apps para dispositivos móviles, así como la compatibilidad con OAuth de Cordova para apps web.' Below this, a table lists six users with their details:

Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID del usuario
geraldinvenegas@gmail...	✉️	10 jul 2025	10 jul 2025	njhnnWgJqdWSRv9pHzabK90...
jennivenegas@gmail.co...	✉️	9 jul 2025	9 jul 2025	HpZbwEVjXTCymnulGwjjDh...
geraldinvenegas@gmail.com	✉️	9 jul 2025	9 jul 2025	YKrmwgJLlwedXdWnLln1bKA...
anamargarito@gmail.c...	✉️	9 jul 2025	9 jul 2025	ED24CNe57ZSh4U6H3TPsWz...
parquitecto@gmail.com	✉️	9 jul 2025	9 jul 2025	J64ekbcAe5NlfoahBxxsqCx...
margaritogeraldin@gm...	✉️	9 jul 2025	9 jul 2025	3L4eSKtQeYYJnLoboDgv0XS...

Ilustración 37. Usuario guardado correctamente.

Fuente: Elaboración Propia.

- Nos dirigimos a **Firestore Database**, para visualizar que nuestra reserva se almaceno correctamente en la base de datos, como se muestra en la *ilustración 38*.



The screenshot shows the Firestore Database console. The path 'reservas' is selected. A document named 'Yzjiij7rVfpLfJmafSpD4' is expanded, showing its fields:

- email: "geraldinvenegas@gmail.com"
- fecha: "2025-07-09"
- hora: "14:30"
- laboratorio: "Lab 4"
- uid: "njhnWgJqdWSRv9pHzabK90OUxm2"

Ilustración 38. Reservas guardadas en la BD correctamente.

Fuente: Elaboración Propia.

Análisis de la práctica

La implementación de esta práctica permitió obtener una visión práctica y clara sobre el funcionamiento de los servicios de Firebase, una plataforma que se ha posicionado como una de las principales herramientas para el desarrollo de aplicaciones web y móviles. Desde la creación del proyecto hasta la gestión de datos mediante Firestore, cada fase del desarrollo aportó conocimientos valiosos sobre cómo construir una aplicación funcional sin depender de servidores físicos o configuraciones complicadas. Uno de los puntos clave fue la autenticación mediante correo y contraseña, la cual demostró ser sencilla de implementar y eficaz para proteger el acceso de los usuarios. A través de Firebase Authentication se logró establecer un sistema de registro e inicio de sesión que valida la identidad del usuario y restringe el acceso a ciertas funcionalidades de la aplicación.

Otro aspecto destacable fue la experiencia de desplegar la app mediante Firebase Hosting. Esta herramienta facilitó la publicación del proyecto en línea en cuestión de minutos, lo cual es ideal para pruebas, demostraciones o entornos de producción iniciales. A diferencia de otros métodos más tradicionales de despliegue, Firebase ofrece comandos intuitivos y una integración directa con el proyecto, lo que reduce el margen de error y mejora la productividad. Además, el uso de Firestore como base de datos permitió almacenar y consultar reservas de manera estructurada y en tiempo real, garantizando integridad de datos y validaciones como fechas inválidas o reservas duplicadas. Esta parte de la práctica reforzó la importancia de considerar validaciones tanto en el frontend como en el backend para ofrecer una experiencia de usuario fluida y segura.

Finalmente, uno de los aprendizajes más relevantes fue entender cómo interactúan los diferentes servicios de Firebase de forma conjunta: desde la autenticación, el almacenamiento de datos y hasta el despliegue en la nube, todo está pensado para integrarse de manera rápida y eficiente. Esta práctica no solo fortaleció conocimientos técnicos, sino que también fomentó buenas prácticas de desarrollo moderno como el uso de plataformas serverless, el manejo de datos centralizados y la seguridad en las aplicaciones web.



TECNOLÓGICO
NACIONAL DE MÉXICO

TES SFP
TECNOLÓGICO DE ESTUDIOS SUPERIORES
SAN FELIPE DEL PROGRESO

TECNOLÓGICO DE ESTUDIOS SUPERIORES SAN FELIPE
DEL PROGRESO

Ingeniería Informática

CÓMPUTO EN LA NUBE

Prácticas del Segundo Corte: Azure, GitHub, Firebase

Semestre:

Julio - Agosto 2025 (Periodo: 07 de Julio – 30 de Agosto
2025)

Alumna: Geraldin Venegas Margarito

No. Cuenta: 2021330023

Grupo: 902

Docente: Ing. Lizeth José Ángeles

San Felipe del Progreso a 09 de Agosto de 2025

Índice del contenido:

Introducción.....	6
Objetivo de la práctica	6
Desarrollo	7
Requerimientos.....	7
Crear cuenta en Azure para estudiantes.....	7
Creación de cuenta y repositorio en GitHub.....	12
Crear Static Web App en Azure y vincular GitHub	18
Modificar proyecto y reflejar cambios en tiempo real.....	24
Crear dominio personalizado y registros DNS (Azure + Firebase)	26
Creación de una Máquina Virtual en Microsoft Azure	31
Creación de Cuentas de Almacenamiento en Microsoft Azure.....	44
Conexión remota y transferencia de archivos a la máquina virtual de Azure	50
Análisis de la practica.....	55

Índice de Ilustraciones

Ilustración 1. Página de inicio de Azure y ruta para registrarse como estudiante.	7
Ilustración 2. Apartado de precios de Azure.	8
Ilustración 3. Apartado de Azure for Students.	8
Ilustración 4. Empezar gratis.	8
Ilustración 5. Iniciar sesión con cuenta institucional.	9
Ilustración 6. Formulario de verificación académica con datos personales_1.	9
Ilustración 7. Formulario de verificación académica con datos personales_2.	10
Ilustración 8. Formulario de verificación académica con datos personales_3.	10
Ilustración 9. Proceso de autenticación mediante Microsoft Authenticator_1.	11
Ilustración 10. Proceso de autenticación mediante Microsoft Authenticator_2.	11
Ilustración 11. Interfaz de bienvenida de Azure Education tras el registro exitoso.	12
Ilustración 12. Acceder a GitHub.	12
Ilustración 13. Instalación de GitHub Desktop.	13
Ilustración 14. Iniciar sesión.	13
Ilustración 15. Autorizar permisos a GitHub Desktop.	14
Ilustración 16. Abrir la cuenta en GitHub Desktop.	14
Ilustración 17. Abrir GitHub con la cuenta creada.	15
Ilustración 18. Crear carpeta en el Escritorio.	15
Ilustración 19. Crear repositorio.	15
Ilustración 20. Asignar nombre y ubicación del repositorio.	16
Ilustración 21. Carpeta creada del nuevo repositorio.	16
Ilustración 22. Archivos del programa Reservación de Consultorios.	16
Ilustración 23. Realizar commit.	17
Ilustración 24. Publicar en el repositorio.	17
Ilustración 25. Commit subido en el repositorio correctamente.	18
Ilustración 26. Búsqueda de Static Web App en Azure.	18
Ilustración 27. Crear una nueva app web estática.	19
Ilustración 28. Crear un nuevo recurso.	19
Ilustración 29. Asignación del nombre y tipo de plan.	20
Ilustración 30. Detalles de implementación.	20
Ilustración 31. Iniciar sesión con GitHub.	20
Ilustración 32. Detalles de implementación completa.	21
Ilustración 33. Resumen de la web estática - Crear.	21
Ilustración 34. Mensaje de implementación.	22
Ilustración 35. Creación de Static Web App correcto.	22
Ilustración 36. Visualizar recurso de la app estática.	22
Ilustración 37. Detalles de la app estática creada.	23
Ilustración 38. Visualización del sistema desde la url de la app estática.	23
Ilustración 39. Edición del index y visualización en Git.	24
Ilustración 40. Realizar commit: listas_de_laboratorio.	24
Ilustración 41. Publicar en el repositorio.	25

Ilustración 42. Visualizar el commit en el repositorio web.....	25
Ilustración 43. Sitio actualizado mostrando las nuevas opciones.....	25
Ilustración 44. Opción de Agregar dominio personalizado.....	26
Ilustración 45. Asignar nombre del dominio.....	26
Ilustración 46. Registros correspondientes al DNS.....	27
Ilustración 47. Copiar los registros y cerrar por ahora.....	27
Ilustración 48. Crear Zona DNS: "geralven.com".....	28
Ilustración 49. DNS creado correctamente.....	28
Ilustración 50. Abrir el DNS creado e ir a los Registros.....	29
Ilustración 51. Agregar el registro Tipo: A.....	29
Ilustración 52. Agregar el registro Tipo: TXT.....	30
Ilustración 53. Registros A y TXT agregados correctamente.....	30
Ilustración 54. Abrir PuTTYgen.....	31
Ilustración 55. Generar clave.....	31
Ilustración 56. Copiar clave.....	32
Ilustración 57. Frase de seguridad.....	32
Ilustración 58. Guardar claves.....	32
Ilustración 59. Claves guardadas.....	33
Ilustración 60. Crear un nuevo recurso.....	33
Ilustración 61. Seleccionar Máquina virtual.....	34
Ilustración 62. Asignar un grupo de recursos.....	34
Ilustración 63. Asignar nombre y región.....	34
Ilustración 64. Asignar nombre de usuario.....	35
Ilustración 65. Seleccionar un Origen de clave pública SSH.....	35
Ilustración 66. Pegar clave pública SSH.....	36
Ilustración 67. Seleccionar un tipo de disco del SO.....	36
Ilustración 68. Revisar y Crear.....	37
Ilustración 69. Crear máquina virtual.....	37
Ilustración 70. Creación correcta.....	38
Ilustración 71. Detalles del recurso de la maquina creada.....	38
Ilustración 72. Copiar dirección IP pública.....	39
Ilustración 73. PuTTY Connection Auth > Credentials.....	39
Ilustración 74. Selección de clave privada.....	40
Ilustración 75. Ingresar nombre de usuario.....	40
Ilustración 76. Ingresar dirección IP pública.....	41
Ilustración 77. Ingresar frase.....	41
Ilustración 78. Actualización de paquetes.....	42
Ilustración 79. Creación de user.....	42
Ilustración 80. Creación de archivo y carpeta.....	43
Ilustración 81. Mostrar contenido del archivo.....	43
Ilustración 82. Apartado de cuentas de almacenamiento.....	44
Ilustración 83. Asignar nombre a la cuenta de almacenamiento.....	44
Ilustración 84. Configuración del servicio principal.....	45
Ilustración 85. Habilitar Acceso de lectura de datos.....	45

Ilustración 86. Crear la cuenta de almacenamiento.....	45
Ilustración 87. Creación de almacenamiento correcto.	46
Ilustración 88. Ir al recurso > Crear.....	46
Ilustración 89. Cargar un archivo.....	46
Ilustración 90. Crear un nuevo contenedor.....	47
Ilustración 91. Cargar blob.	47
Ilustración 92. Mensaje de que se cargaron los blobs correctamente.....	48
Ilustración 93. Dirigirse a Contenedores.	48
Ilustración 94. Visualización de los Blobs.	48
Ilustración 95. Abrir el blob creado "documentos"	49
Ilustración 96. Abrir y descargar el archivo subido a la carpeta de documentos.....	49
Ilustración 97. Descarga correcta.	50
Ilustración 98. Instalación de WinSCP.....	50
Ilustración 99. Finalizar instalación.	51
Ilustración 100. Conexión guardada.....	51
Ilustración 101. Aviso de realizar la conexión.	52
Ilustración 102. Ingresar frase.....	52
Ilustración 103. Conexión entre máquinas correcta.	53
Ilustración 104. Transferencia de documento de la máquina física a la virtual.	53
Ilustración 105. Transferencia realizada correctamente.	54
Ilustración 106. Transferencia de documento de la máquina virtual a la física.	54
Ilustración 107. Visualización de los documentos en la máquina virtual.....	54

Introducción

En esta práctica se llevará a cabo el desarrollo de un sistema web utilizando tecnologías en la nube, integrando plataformas como Microsoft Azure, GitHub y Firebase. El objetivo principal será comprender y aplicar el despliegue de una aplicación web estática mediante Azure Static Web Apps, aprovechando los beneficios de los servicios gratuitos que ofrece Microsoft para estudiantes.

Como parte del proceso, se creará una máquina virtual en Azure, a la cual se accederá mediante el software PuTTY para realizar configuraciones y prácticas remotas. Además, se crearán cuentas de almacenamiento en la plataforma de Azure para gestionar recursos necesarios en la nube. También se configurará un repositorio en GitHub para alojar el código del sistema, estableciendo una integración continua con Azure para facilitar el despliegue automatizado.

Posteriormente, se realizarán modificaciones en el proyecto con el fin de verificar el correcto funcionamiento de la sincronización entre el repositorio de GitHub y la plataforma de Azure. Finalmente, se configurará un dominio personalizado a través de Firebase Hosting, el cual se vinculará con Azure DNS, permitiendo que la aplicación web esté disponible mediante una dirección de dominio específica. Esta práctica tiene como propósito que el estudiante se familiarice con procesos reales de despliegue, automatización y vinculación de servicios en la nube. De igual manera se realizará una conexión remota entre la máquina física con la máquina virtual de Ubuntu con la finalidad de transferir archivos entre las máquinas.

Objetivo de la práctica

Implementar un sitio web estático en Microsoft Azure con actualizaciones en tiempo real desde GitHub y vincular un dominio personalizado desde Firebase Hosting mediante registros DNS. Además, se configurará una máquina virtual y una cuenta de almacenamiento en Azure para reforzar el uso de servicios esenciales en la nube. Así como permitir la conexión remota con la máquina virtual y la máquina física para la transferencia de archivos.

Desarrollo

Requerimientos

- Acceso a internet.
- Navegador web (Google Chrome, Firefox, Edge, etc.)
- Cuenta institucional activa (para Azure for Students)
- Cuenta en GitHub (o crear una nueva)
- GitHub Desktop (aplicación de escritorio)
- Microsoft Authenticator (app para autenticación de dos factores)
- Acceso a Firebase Console
- Editor de código (Visual Studio Code o similar).
- PuTTYgen y PuTTY (cliente SSH para conexión remota a máquinas virtuales).
- Software WinSCP.

Crear cuenta en Azure para estudiantes

- Ingresar a la página oficial de [Microsoft Azure](#), como se muestra en la ilustración 1.

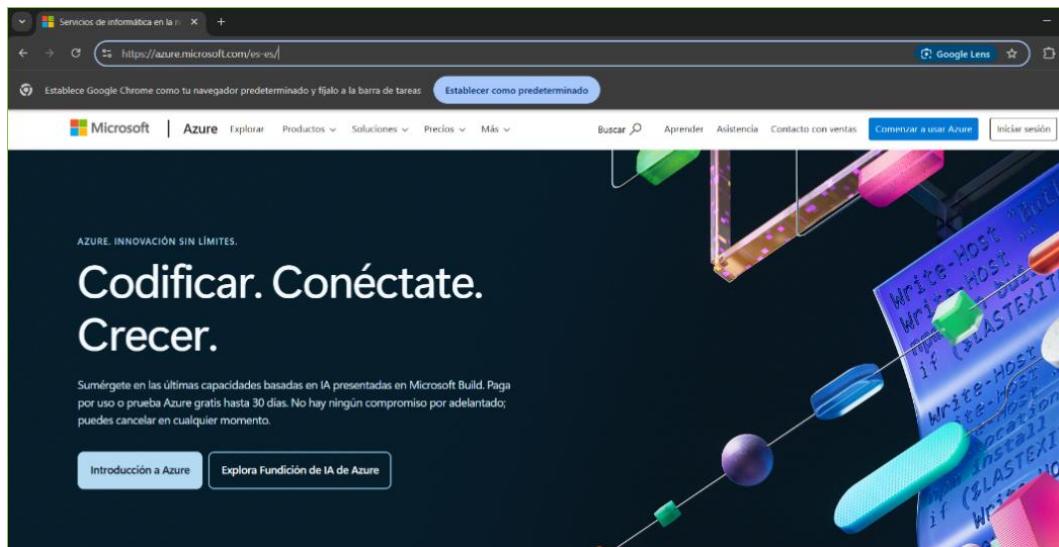


Ilustración 1. Página de inicio de Azure y ruta para registrarse como estudiante.

Fuente: Elaboración Propia.

- Ir al apartado de “**Precios**” > “**Servicios gratuitos de Azure**”, como se muestra en la *ilustración 2*.

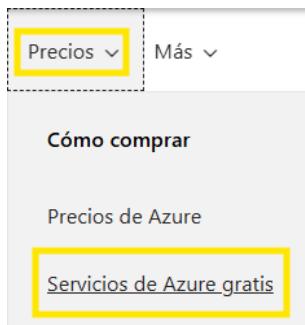


Ilustración 2. Apartado de precios de Azure.

Fuente: Elaboración Propia.

- Buscar y seleccionar “**Azure for Students**”, como se muestra en la *ilustración 3*.



Ilustración 3. Apartado de Azure for Students.

Fuente: Elaboración Propia.

- Clic en “**Más información**”, luego en “**Empezar gratis**”, como se muestra en la *ilustración 4*.

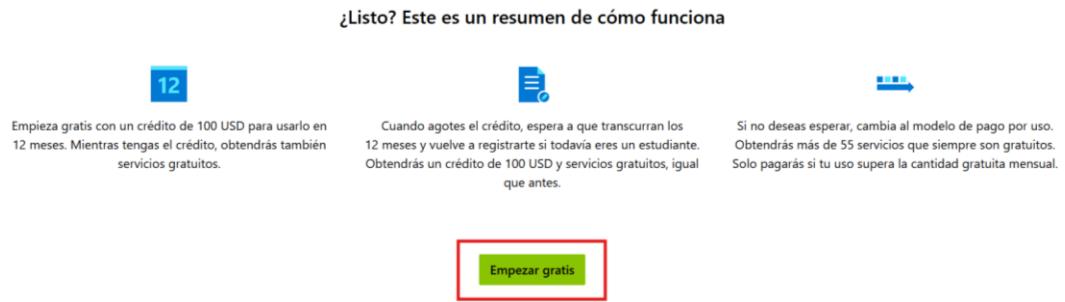


Ilustración 4. Empezar gratis.

Fuente: Elaboración Propia.

- Iniciar sesión con una cuenta institucional (correo de escuela), como se muestra en la *ilustración 5*.

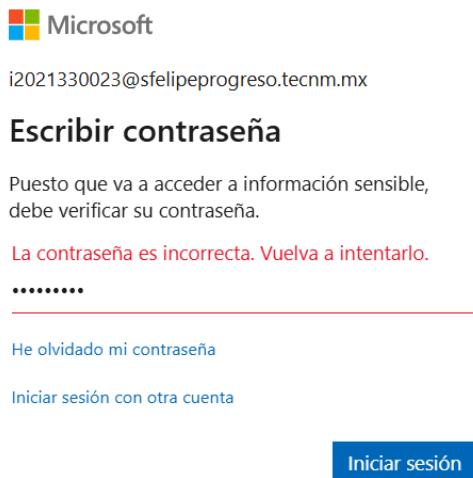


Ilustración 5. Iniciar sesión con cuenta institucional.

Fuente: Elaboración Propia.

- Completar el formulario con datos personales y verificar identidad, como se muestra en las siguientes ilustraciones.

Verificación académica

Para empezar, escriba su nombre según los registros de la escuela. Seleccione el país o región de la escuela y escriba el nombre de la escuela. Escriba su fecha de nacimiento según los registros de la escuela.

Nombre	Geraldin
Apellidos	Venegas Margarito
País o región	México
Si su país no aparece en la lista, la oferta no está disponible en su región. Más información	
Nombre de la escuela	Tecnológico De Estudios Superiores De San Felipe Del Progreso (San Felipe del Progreso, Méx., Mexico, Mé

Ilustración 6. Formulario de verificación académica con datos personales_1.

Fuente: Elaboración Propia.

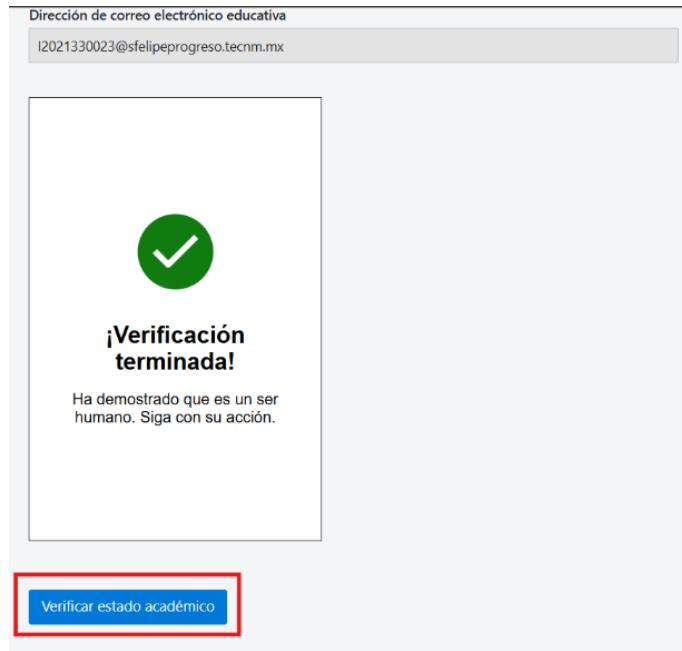


Ilustración 7. Formulario de verificación académica con datos personales_2.

Fuente: Elaboración Propia.

Código postal

Si continúa, acepta que, si usa el correo electrónico de su organización, es posible que su organización disponga de derechos para acceder y administrar sus datos y su cuenta. [Más información](#)

Entiendo que Microsoft puede ponerse en contacto conmigo sobre mi cuenta gratuita.

- Represento que soy la persona identificada en esta cuenta.
Acepto los términos y condiciones de la [Contrato de cliente de Microsoft](#) incluidos los Términos de usuario individuales complementarios.
- Me gustaría recibir información, sugerencias y ofertas sobre Azure y otros productos y servicios de Microsoft.
- Quiero que Microsoft comparta mi información con determinados partners para poder recibir información pertinente sobre sus productos y servicios.

Lea la [declaración de privacidad](#) para obtener información sobre cómo se administran los datos.

Registrarse

Ilustración 8. Formulario de verificación académica con datos personales_3.

Fuente: Elaboración Propia.

- Descargar la app **Microsoft Authenticator**, escanear el código y validar acceso, como se muestra en la *ilustración 9*.



Ilustración 9. Proceso de autenticación mediante Microsoft Authenticator_1.

Fuente: Elaboración Propia.

- Finalmente, dar clic en “**Listo**” para completar el registro, como se muestra en la *ilustración 10*.

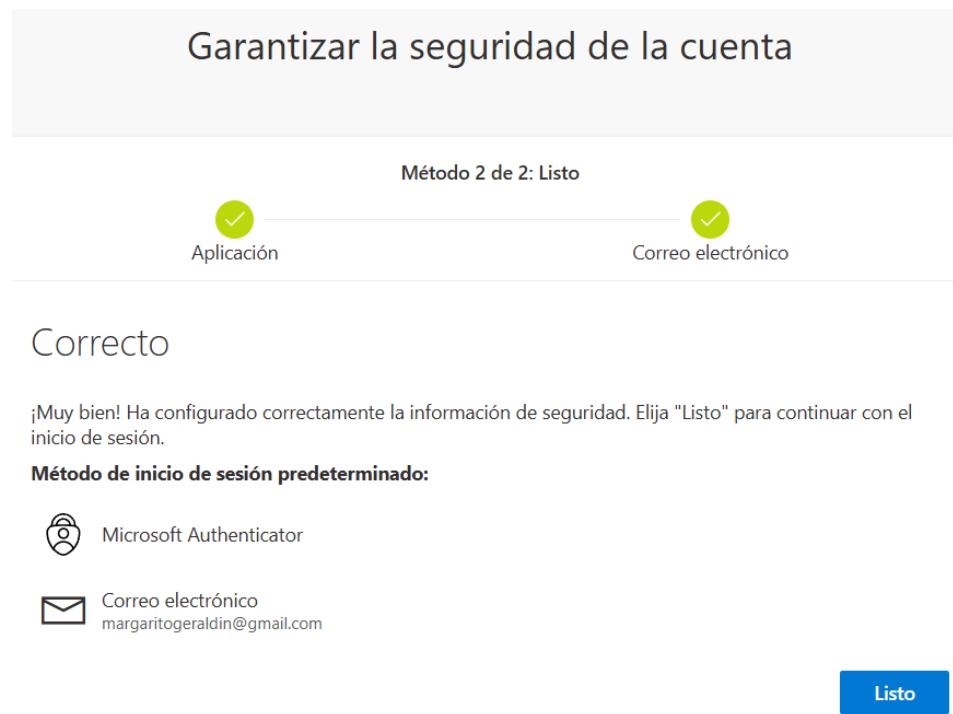


Ilustración 10. Proceso de autenticación mediante Microsoft Authenticator_2.

Fuente: Elaboración Propia.

- A continuación, en la *ilustración 11* se muestra la interfaz de Microsoft Azure.

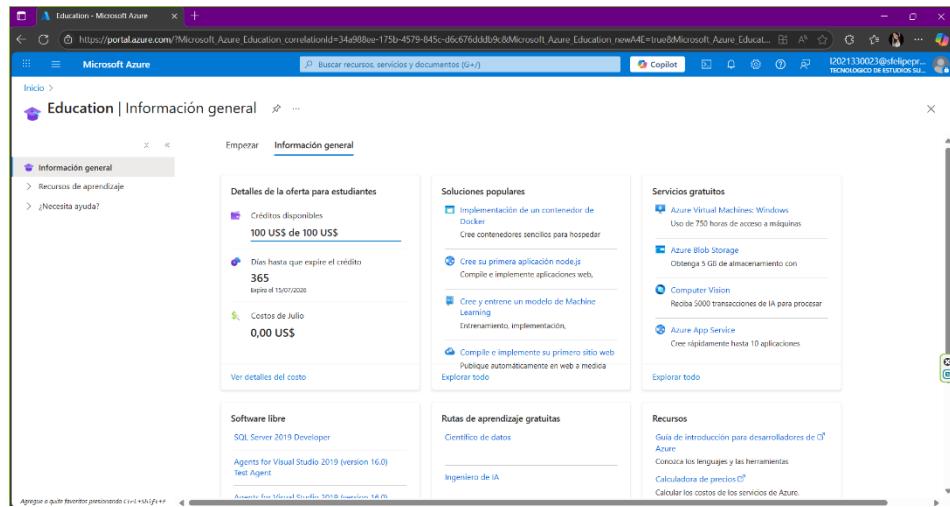


Ilustración 11. Interfaz de bienvenida de Azure Education tras el registro exitoso.

Fuente: Elaboración Propia.

Creación de cuenta y repositorio en GitHub

- Acceder a [GitHub](#) y crear una cuenta (o iniciar sesión), como se muestra en la *ilustración 12*.

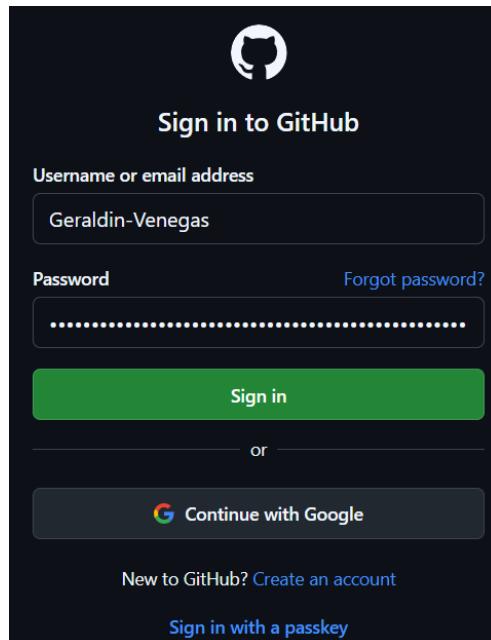


Ilustración 12. Acceder a GitHub.

Fuente: Elaboración Propia.

- Descargar e instalar **GitHub Desktop**, después iniciamos sesión con nuestra cuenta dando clic en "**Sign in to GitHub.com**", como se muestra en la *ilustración 13*.

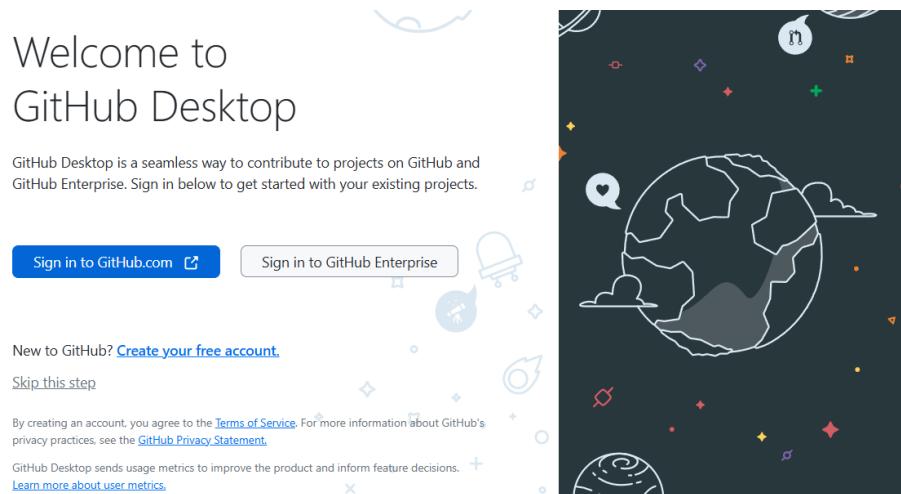


Ilustración 13. Instalación de GitHub Desktop.

Fuente: Elaboración Propia.

- Posteriormente nos abrirá una página donde se muestra nuestra cuenta dar clic en continuar, como se muestra en la *ilustración 14*.

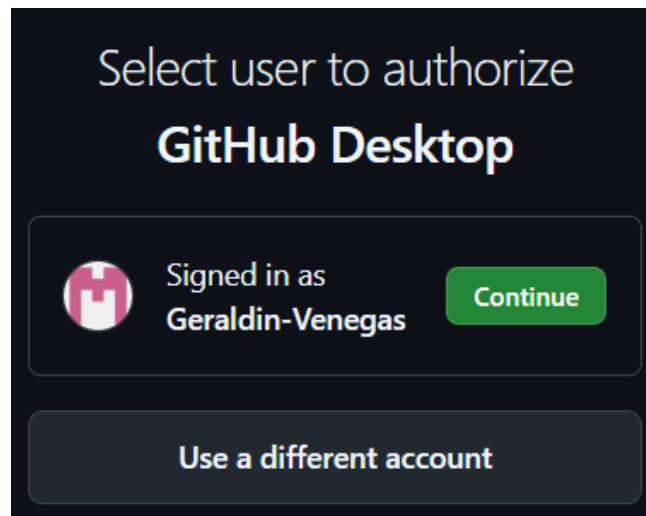


Ilustración 14. Iniciar sesión.

Fuente: Elaboración Propia.

- Autorizar que GitHub Desktop tenga acceso, dar clic en “Authorize desktop”, como se muestra en la *ilustración 15*.

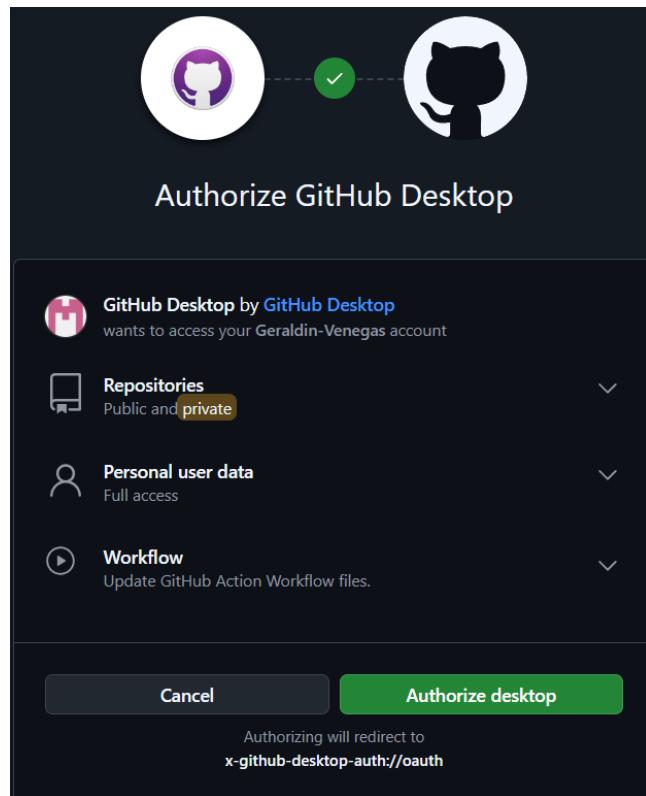


Ilustración 15. Autorizar permisos a GitHub Desktop.

Fuente: Elaboración Propia.

- Posteriormente nos aparecerá que la de abrir GitHub Desktop, dar clic en “Abrir”, como se muestra en la *ilustración 16*.

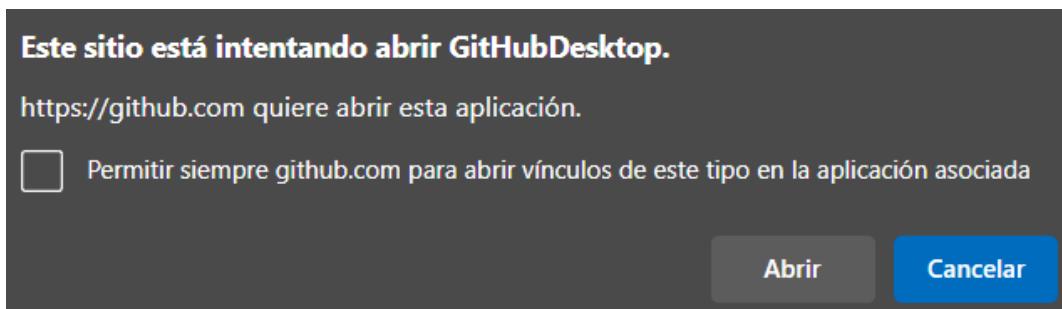


Ilustración 16. Abrir la cuenta en GitHub Desktop.

Fuente: Elaboración Propia.

- Al dar clic, los datos de la cuenta se mostrarán en la aplicación y dar clic en “Finish”, como se muestra en la *ilustración 17*.

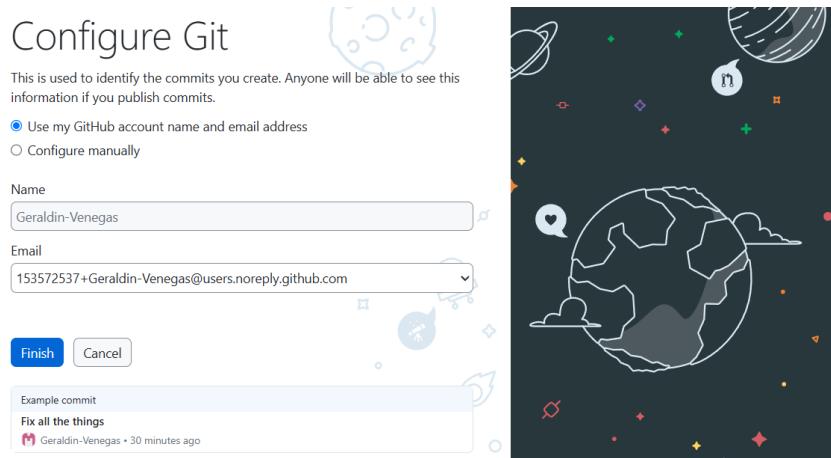


Ilustración 17. Abrir GitHub con la cuenta creada.

Fuente: Elaboración Propia.

- Crear una carpeta en el Escritorio con el nombre de GitHub, como se muestra en la *ilustración 18*.

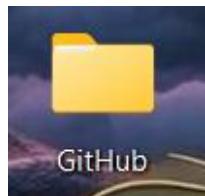


Ilustración 18. Crear carpeta en el Escritorio.

Fuente: Elaboración Propia.

- Una vez iniciado sesión vamos a crear un Repositorio, para ello ir a **File > New Repository**, asignar nombre (ej. geralvenmar) y seleccionar carpeta de destino, como se muestra en la *ilustración 19*.

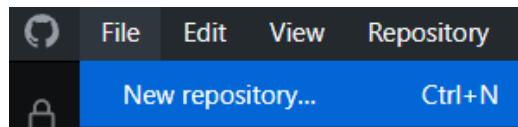


Ilustración 19. Crear repositorio.

Fuente: Elaboración Propia.

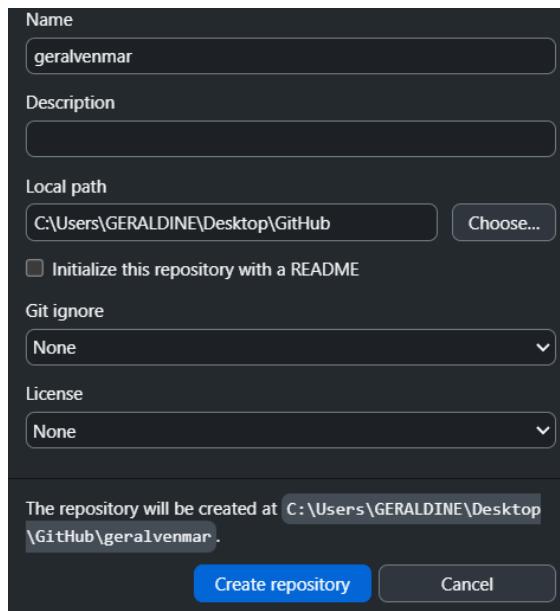


Ilustración 20. Asignar nombre y ubicación del repositorio.

Fuente: Elaboración Propia.

- Ya que se creó el repositorio nos aparecerá una carpeta con el nombre del repositorio, como se muestra en la *ilustración 21*.



Ilustración 21. Carpeta creada del nuevo repositorio.

Fuente: Elaboración Propia.

- De nuestro proyecto de Reservación de Consultorios, vamos a copiar los siguientes archivos: **app.js**, **index.html**, **style.css** y los pegamos en la carpeta que se creó, como se muestra en la *ilustración 22*.

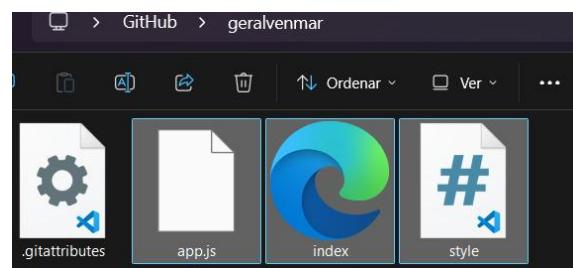


Ilustración 22. Archivos del programa Reservación de Consultorios.

Fuente: Elaboración Propia.

- Una vez pegado estos se deben mostrar en GitHub, procedemos a crear un commit para subirlos le asignamos "**actualización de código**" y presionar el botón de "**Commit 3 files to main**", como se muestra en la *ilustración 23*.

The screenshot shows a GitHub desktop application interface. In the top left, it says "Current repository: geralvenmar" and "Current branch: main". On the right, there's a "Publish repository" button. The main area shows a diff of an "app.js" file with three changes. A yellow box highlights the commit message input field containing "actualización de código" and the "Commit 3 files to main" button at the bottom.

```

@@ -0,0 +1,240 @@
1+ const firebaseConfig = {
2+   apiKey: "AIzaSyCIG73zejXkPHbKZuMinIRrTKApjmHldDQ",
3+   authDomain: "progvm-a66ba.firebaseio.com",
4+   projectId: "progvm-a66ba",
5+   storageBucket: "progvm-a66ba.firebaseio.storage.app",
6+   messagingSenderId: "121010461551",
7+   appId: "1:121010461551:web:ee23f5f0fd00c7ca22a69a",
8+   measurementId: "G-W8F64DHDCG"
9+ };
10+
11+
12+ firebase.initializeApp(firebaseConfig);
13+ const auth = firebase.auth();
14+ const db = firebase.firestore();
15+
16+ function register() {
17+   const email = document.getElementById("email");
18+   const password = document.getElementById("password");
19+
20+   // Validación de campos vacíos
21+   if (!email.value.trim() || !password.value.trim()) {
22+     if (!email.value.trim()) email.classList.add("input-error");
23+     if (!password.value.trim()) password.classList.add("input-error");
24+
25+     Swal.fire({
26+       icon: 'warning',

```

Ilustración 23. Realizar commit.

Fuente: *Elaboración Propia.*

- Posteriormente dar clic en "Publish repository", como se muestra en la *ilustración 24*.

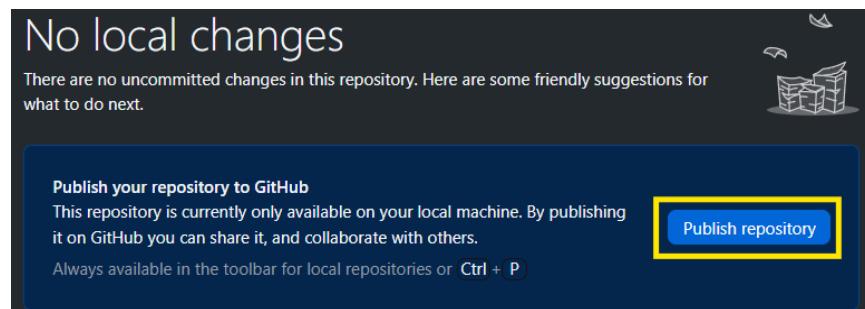


Ilustración 24. Publicar en el repositorio.

Fuente: *Elaboración Propia.*

- Una vez que se subió nos dirigimos GitHub, pero en la página buscamos el repositorio (`geralvenmar`) y verificamos que los archivos se subieron correctamente, como se muestra en la *ilustración 25*.

The screenshot shows a GitHub repository page for the user `geralvenmarr`. The repository name is `Geraldin-Venegas`. The commit history is displayed, showing a single commit titled "actualización de código" made by `Geraldin-Venegas` at 9290da9 · 1 minute ago. This commit includes three files: `app.js`, `index.html`, and `style.css`, all updated at 1 minute ago. The commit message is "actualización de código". The entire commit list is highlighted with a yellow box.

Ilustración 25. Commit subido en el repositorio correctamente.

Fuente: Elaboración Propia.

Crear Static Web App en Azure y vincular GitHub

- En Microsoft Azure buscar “**Static Web App**”, como se muestra en la *ilustración 26*.

The screenshot shows the Microsoft Azure portal interface. The top navigation bar has "Microsoft Azure" and a search bar. Below the navigation, there's a breadcrumb trail: "Inicio > Static Web Apps". The main content area displays a table of static web apps. At the top of the table, there are filters: "+ Crear", "Administrador vista", "Actualizar", "Exportar a CSV", "Abrir consulta", and "Asignar etiquetas". Below the table, a blue banner message reads: "You are viewing a new version of Browse experience. Click here to access the old experience." There are also filter buttons for "Filtrar por cualquier ca...", "Suscripción es igual a todo", "Grupo de recursos es igual a todo", "Ubicación es igual a todo", and "Agregar filtro".

Ilustración 26. Búsqueda de Static Web App en Azure.

Fuente: Elaboración Propia.

- Crear una nueva app web estática, para ello dar clic en el botón “Crear”, como se muestra en la *ilustración 27*.

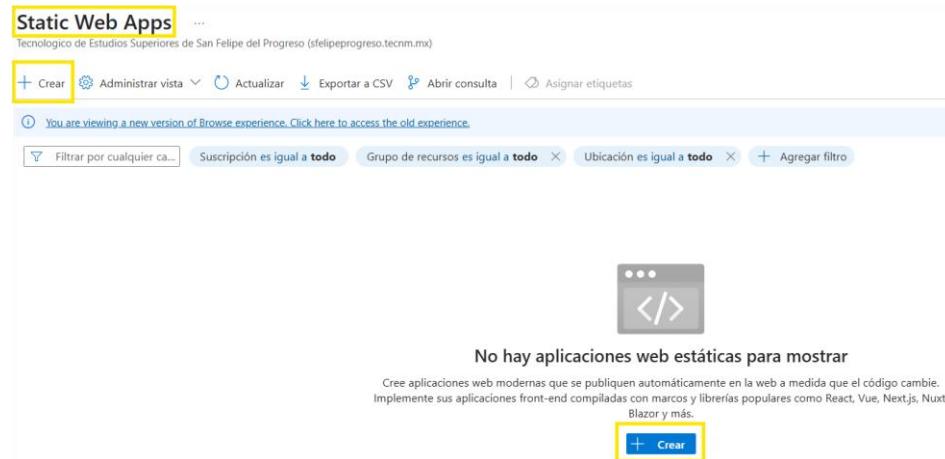


Ilustración 27. Crear una nueva app web estática.

Fuente: Elaboración Propia.

- Crear grupo de recursos: AzureGVM, como se muestra en la *ilustración 28*.

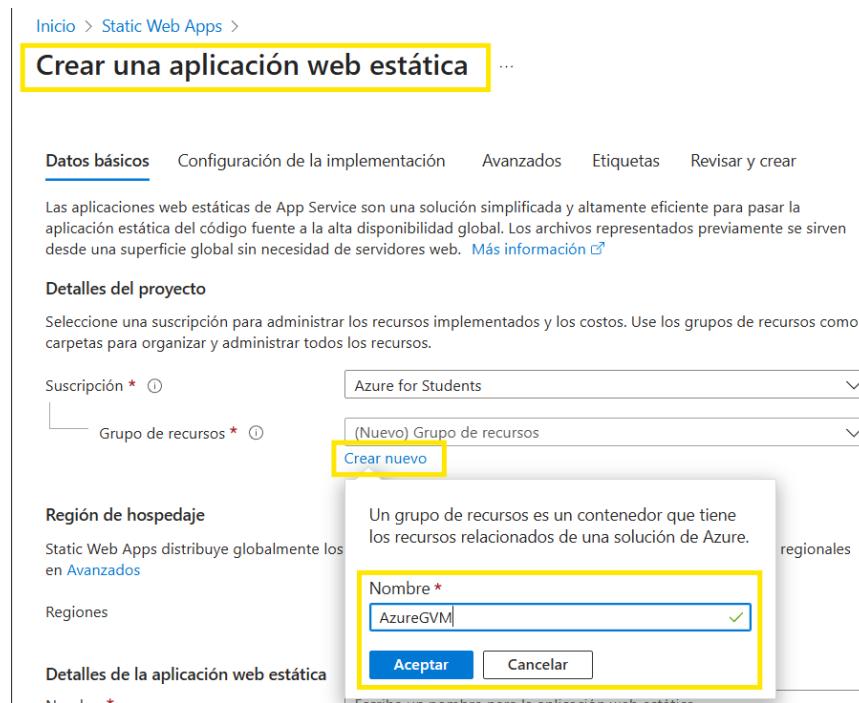


Ilustración 28. Crear un nuevo recurso.

Fuente: Elaboración Propia.

- Nombre de la app: AzureGVM, plan: **Gratis**, como se muestra en la *ilustración 29*.

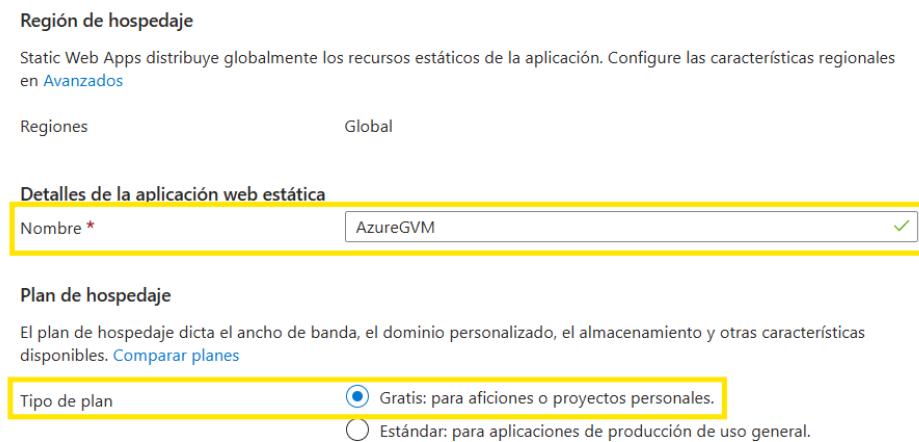


Ilustración 29. Asignación del nombre y tipo de plan.

Fuente: Elaboración Propia.

- Seleccionar como origen: **GitHub**, como se muestra en la *ilustración 30*.



Ilustración 30. Detalles de implementación.

Fuente: Elaboración Propia.

- Iniciar sesión en GitHub desde Azure, como se muestra en la *ilustración 31*.

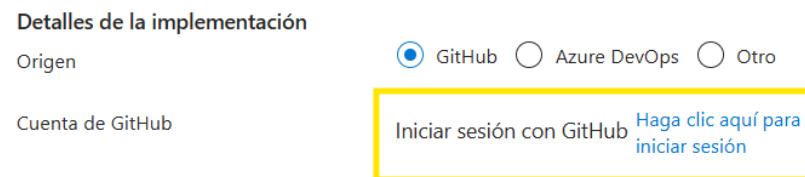


Ilustración 31. Iniciar sesión con GitHub.

Fuente: Elaboración Propia.

- Seleccionar Organización: Geraldin-Venegas, Repositorio: geralvenmar, Rama: main y finalmente dar clic en **Revisar y crear**, como se muestra en la *ilustración 32*.

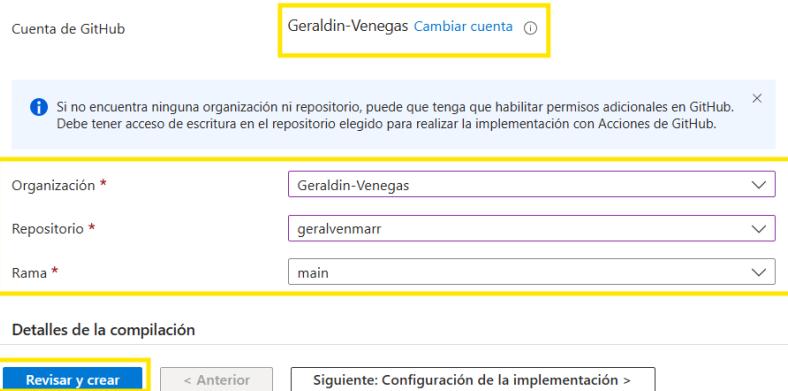


Ilustración 32. Detalles de implementación completa.

Fuente: Elaboración Propia.

- Posteriormente se muestra un resumen de la web estática, en este punto aún se pueden realizar cambios o en caso contrario crear la aplicación estática para ello dar clic en **Crear**, como se muestra en la *ilustración 33*.

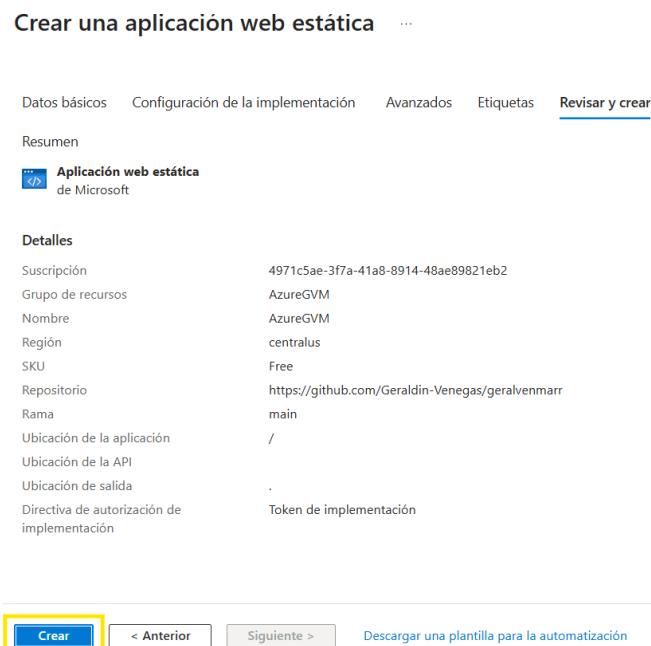


Ilustración 33. Resumen de la web estática - Crear.

Fuente: Elaboración Propia.

- Al dar clic en **Crear** nos aparecerá el siguiente mensaje en la parte superior derecha, como se muestra en la *ilustración 34*



Ilustración 34. Mensaje de implementación.

Fuente: Elaboración Propia.

- Una vez que se completó correctamente nos aparece lo siguiente con los detalles de la implementación, como se muestra en la *ilustración 35*.

Recurso	Tipo	Estado	Detalles de la operación
AzureGVM	Aplicación web estática	OK	Detalles de la operación

Ilustración 35. Creación de Static Web App correcto.

Fuente: Elaboración Propia.

- Dar clic en el recurso “**AzureGVM**” para ver solo detalles.

Detalles de implementación			
Recurso	Tipo	Estado	Detalles de la operación
AzureGVM	Aplicación web estática	OK	Detalles de la operación

Ilustración 36. Visualizar recurso de la app estática.

Fuente: Elaboración Propia.

- Al abrirlo se pueden visualizar los detalles de la app estática, entre ellos el dominio, como se muestra en la *ilustración 37*.

The screenshot shows the Microsoft Azure portal interface. In the top navigation bar, 'Microsoft Azure' is selected. Below it, the 'AzureGVM' resource is shown as an 'Aplicación web estática'. A search bar and several action buttons ('Ver aplicación en el explorador', 'Actualizar', 'Eliminar', 'Administrar token de implementación', 'Envíos sus comentarios') are visible. On the left, a sidebar lists various management options. The main content area is titled 'Información general' and contains detailed configuration settings. A red box highlights the 'Dominio' field, which is set to 'https://brave-bay-0ddf30510.2.azurestaticapps.net'. Other fields include 'Estado' (Listo), 'Entorno' (Producción), and 'Plan de hospedaje' (Free). At the bottom, there are buttons for 'Visitar el sitio', 'Agregar un dominio personalizado', 'Actualizar el plan de hospedaje', and 'Habilitar el perímetro de nivel empresarial'.

Ilustración 37. Detalles de la app estática creada.

Fuente: Elaboración Propia.

- Una vez que abrimos la dirección url se debe visualizar nuestro sistema de Reservación de Laboratorios y este debe funcionar correctamente, como se muestra en la *ilustración 38*.

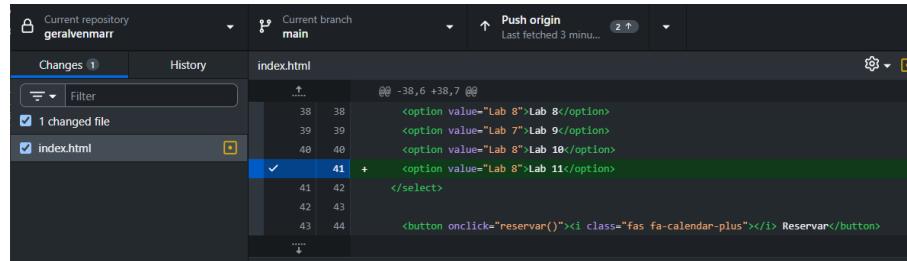
The screenshot shows a web browser window with the URL 'https://brave-bay-0ddf30510.2.azurestaticapps.net' in the address bar. The page itself has a light purple background and is titled 'Reservación de Laboratorios'. It contains two input fields: one for 'Correo' (Email) and one for 'Contraseña' (Password). Below these fields are two purple buttons: 'Iniciar sesión' (Sign in) and 'Registrarse' (Register). The overall design is clean and modern.

Ilustración 38. Visualización del sistema desde la url de la app estática.

Fuente: Elaboración Propia.

Modificar proyecto y reflejar cambios en tiempo real

- Ahora vamos a editar nuestro sistema para agregar más opciones de reservar laboratorios, para ello editar el archivo **index.html**, una vez realizado los cambios en GitHub se debe reflejar automáticamente que archivo se modificó, como se muestra en la *ilustración 39*.



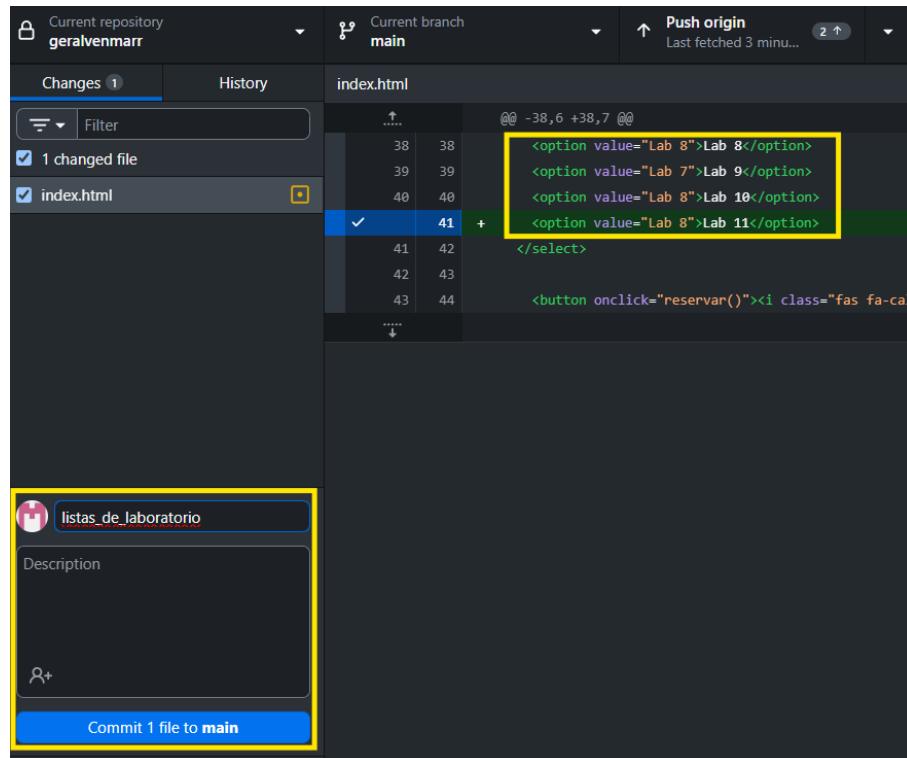
```
@@ -38,6 +38,7 @@
    <option value="Lab 8">Lab 8</option>
    <option value="Lab 7">Lab 9</option>
    <option value="Lab 8">Lab 10</option>
+   <option value="Lab 8">Lab 11</option>
</select>

<button onclick="reservar()"><i class="fas fa-calendar-plus"></i> Reservar</button>
```

Ilustración 39. Edición del index y visualización en Git.

Fuente: Elaboración Propia.

- Realizar un nuevo commit con mensaje: `listas_de_laboratorio`.



listas_de_laboratorio

Description

Commit 1 file to main

Ilustración 40. Realizar commit: `listas_de_laboratorio`.

Fuente: Elaboración Propia.

- Hacer **Push origin**.

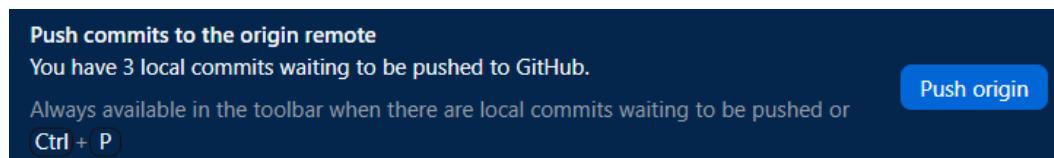


Ilustración 41. Publicar en el repositorio.

Fuente: Elaboración Propia.

- Al subir el commit este se debe reflejar en el repositorio web, como se muestra en la *ilustración 42*.

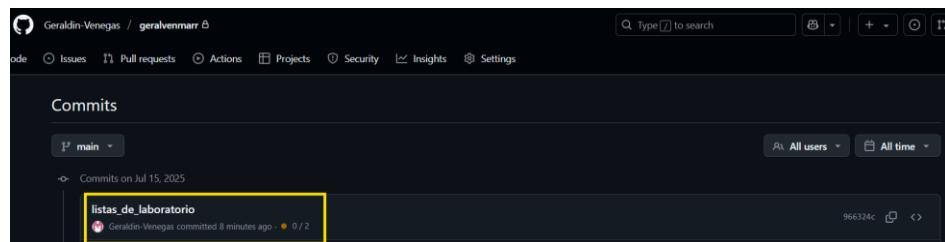


Ilustración 42. Visualizar el commit en el repositorio web.

Fuente: Elaboración Propia.

- Después actualizamos el sitio en Azure para verificar cambios.



Ilustración 43. Sitio actualizado mostrando las nuevas opciones.

Fuente: Elaboración Propia.

Crear dominio personalizado y registros DNS (Azure + Firebase)

En Firebase:

- Ir a **Hosting** > Agregar dominio personalizado.

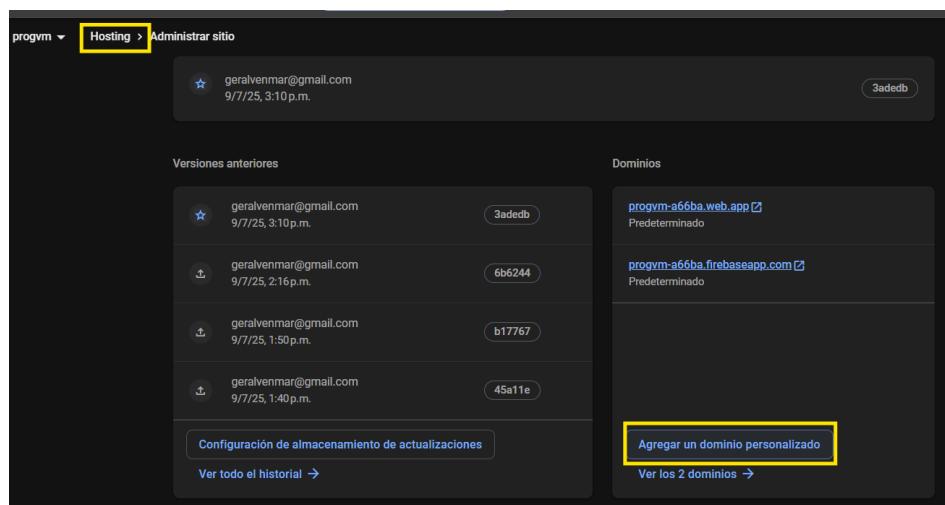


Ilustración 44. Opción de Agregar dominio personalizado.

Fuente: Elaboración Propia.

- Ingresar un nombre del dominio: *geralven.com*, dar clic en continuar.

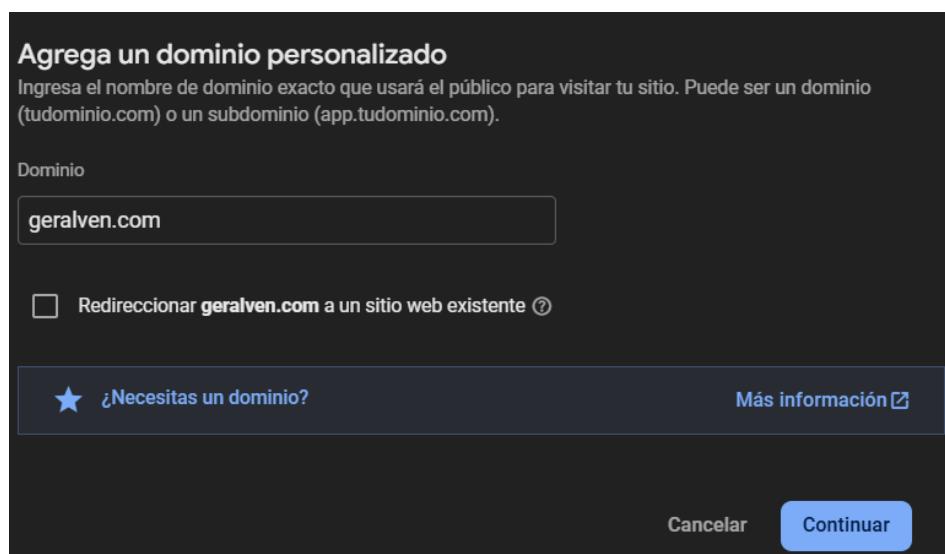


Ilustración 45. Asignar nombre del dominio.

Fuente: Elaboración Propia.

- Al crearse el dominio nos manda los registros correspondientes para el DNS.

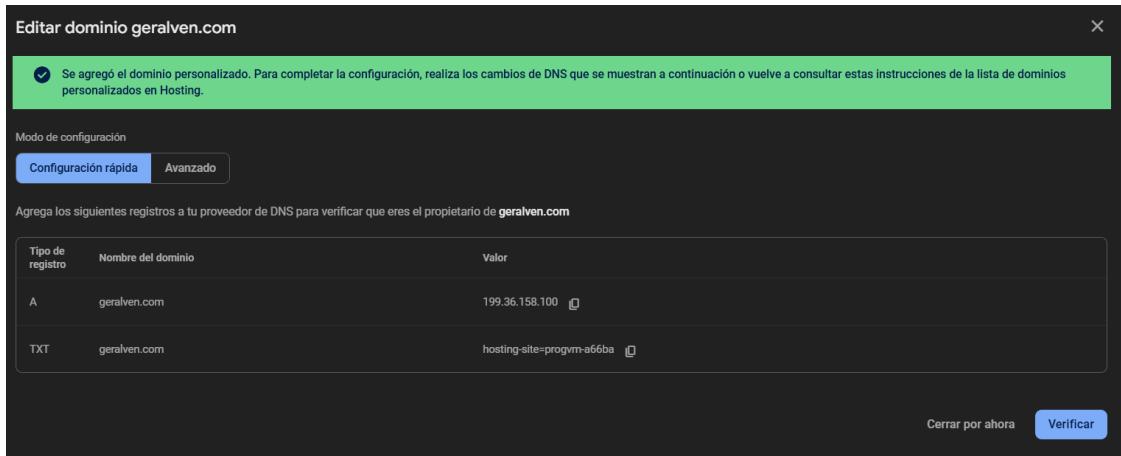


Ilustración 46. Registros correspondientes al DNS.

Fuente: Elaboración Propia.

- Copiar los registros A y TXT que proporciona Firebase.
- Dar clic en “**Verificar**” los registros, nos dice que los cambios se pueden tardar 24hrs en aplicarse, por lo tanto, dar clic en “**Cerrar por ahora**” y revisarlo más tarde.

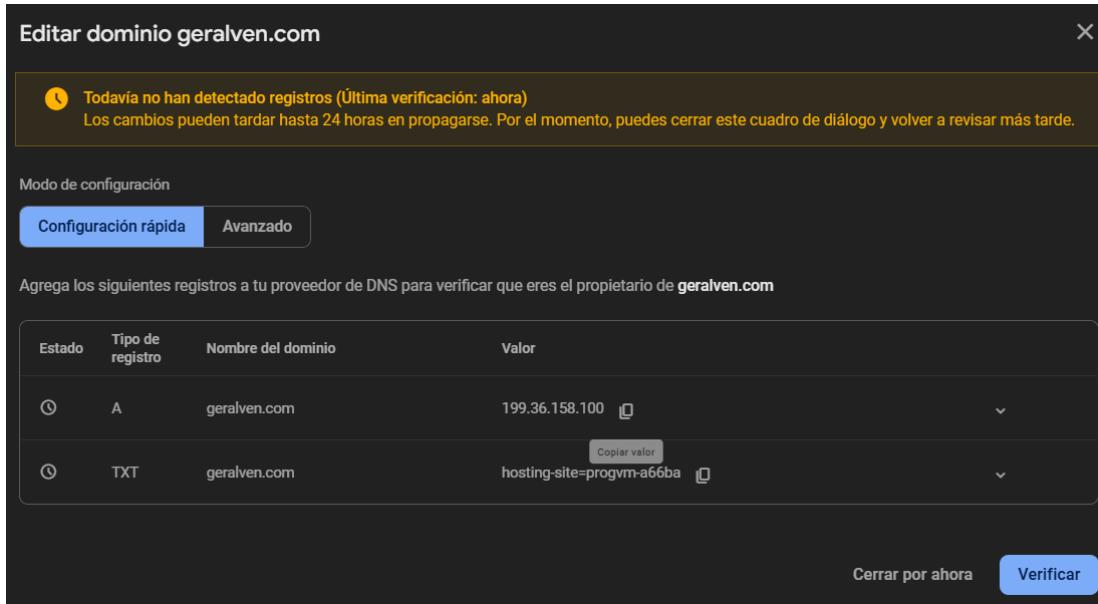


Ilustración 47. Copiar los registros y cerrar por ahora.

Fuente: Elaboración Propia.

En Azure:

- Buscar **Zona DNS** > Crear nueva zona: geralven.com.

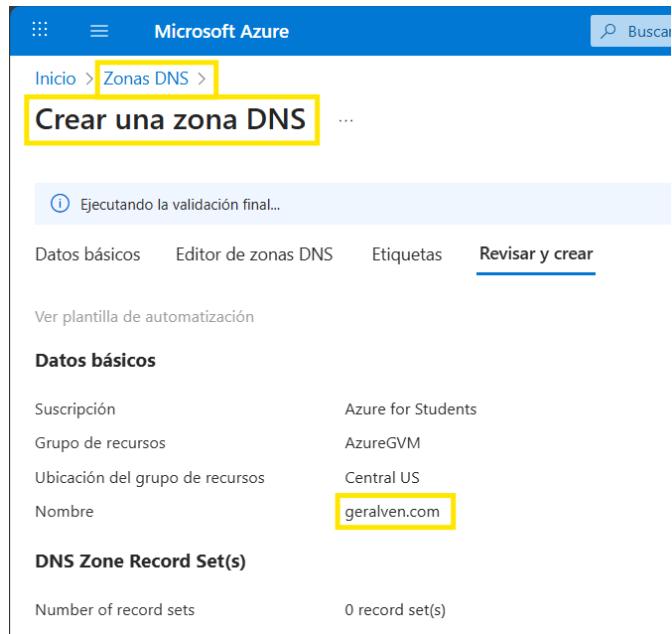


Ilustración 48. Crear Zona DNS: "geralven.com".

Fuente: Elaboración Propia.: Elaboración Propia.: Elaboración Propia. Elaboración Propia

- Al crearse se mostrará el recurso y el tipo, como se muestra en la *ilustración 49*.



Ilustración 49. DNS creado correctamente.

Fuente: Elaboración Propia.

- Al abrirlo nos dirigimos a **Registros DNS**.

Nombre	Tipo
@	NS
@	SOA

Ilustración 50. Abrir el DNS creado e ir a los Registros.

Fuente: Elaboración Propia.

- Agregar los dos registros que nos proporcionó Firebase:
 - **Registro A:** Nombre: www, Tipo: A, Dirección IP.

Agregar conjunto de registros

geralven.com

Nombre:

Tipo:

Conjunto de registros del alias:

TTL: Unidad de TTL:

Dirección IP:

Agregar

Ilustración 51. Agregar el registro Tipo: A.

Fuente: Elaboración Propia.

- **Registro TXT:** Nombre: www, Tipo: TXT, Valor.

The screenshot shows the Azure DNS Management portal for the domain 'geralven.com'. On the left, the navigation menu includes 'Información general', 'Registro de actividad', 'Control de acceso (IAM)', 'Etiquetas', 'Diagnosticar y solucionar problemas', 'Visualizador de recursos', 'Configuración' (with 'Bloqueos' expanded), 'Settings' (with 'Propiedades' expanded), and 'Administración de DNS' (with 'Registros' selected). The main pane displays a table of existing DNS records: one NS record for '@' with TTL 172800 and four SOA records for '@' with TTL 3600. A modal window titled 'Agregar conjunto de registros' is open, prompting for a 'Nombre' (www) and 'Tipo' (TXT - Text records). The 'Valor' field contains 'hosting-site=progvm-a66ba'. The 'Agregar' button at the bottom of the modal is highlighted with a green box.

Ilustración 52. Agregar el registro Tipo: TXT.

Fuente: Elaboración Propia.

- En la *ilustración 53* se visualizan los registros.

The screenshot shows the same Azure DNS Management portal interface as before, but now displaying the results of the previous addition. The table of records now includes the new TXT record for 'www' with TTL 3600 and the value 'hosting-site=progvm-a66ba'. The entire table area is highlighted with a green box.

Ilustración 53. Registros A y TXT agregados correctamente.

Fuente: Elaboración Propia.

- Como siguiente paso ingresamos a la página: www.whatsmydns.net, para visualizar en que estados y países se encuentre registrado nuestro dominio.

Nota: Primero no nos debe arrojar nada, en caso contrario crear otro que no exista, después que se haya validado ya nos debe aparecer nuestro dominio y al ingresar nos debe arrojar nuestro sistema.

Creación de una Máquina Virtual en Microsoft Azure

Para crear una máquina virtual desde Microsoft Azure, se realizó el siguiente procedimiento:

Generar clave SSH con PuTTYgen:

- Abrir **PuTTYgen**.

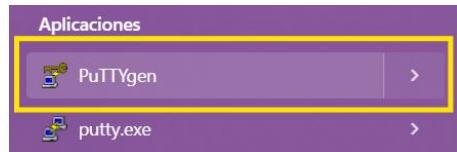


Ilustración 54. Abrir PuTTYgen.

Fuente: Elaboración Propia.: Elaboración Propia.: Elaboración Propia.

- Selecciona el tipo de clave como **RSA** > Dar clic en **Generate**.

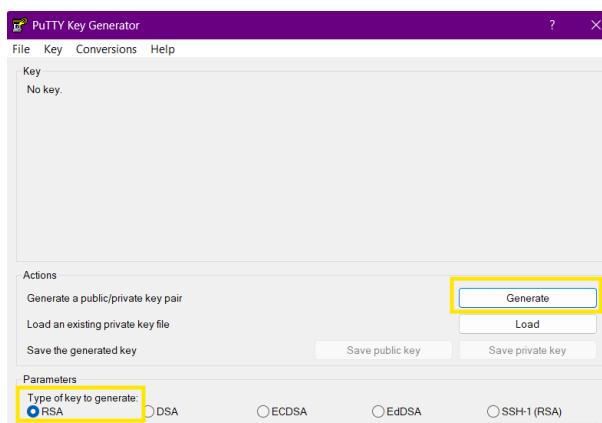


Ilustración 55. Generar clave.

Fuente: Elaboración Propia.: Elaboración Propia.: Elaboración Propia.

- Se genera la clave y se copia el contenido que aparece en el área superior (clave pública).

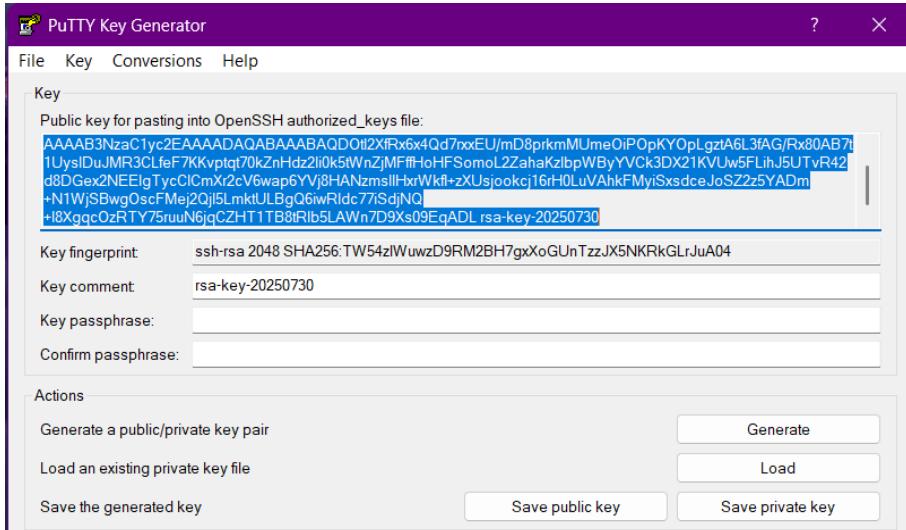


Ilustración 56. Copiar clave.

Fuente: Elaboración Propia.: Elaboración Propia.: Elaboración Propia.

- En el campo **Key passphrase** se escribe una frase de seguridad, y se repite en **Confirm passphrase**.

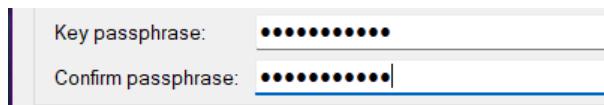


Ilustración 57. Frase de seguridad.

Fuente: Elaboración Propia.: Elaboración Propia.

- Se guarda tanto la **clave pública** como la **clave privada** en un lugar seguro.

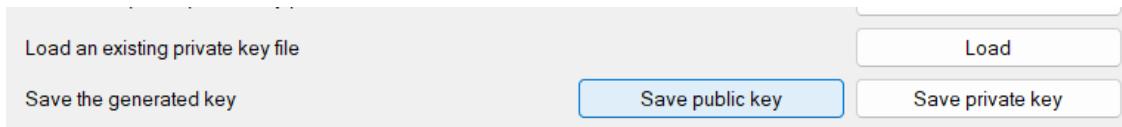


Ilustración 58. Guardar claves.

Fuente: Elaboración Propia.: Elaboración Propia.

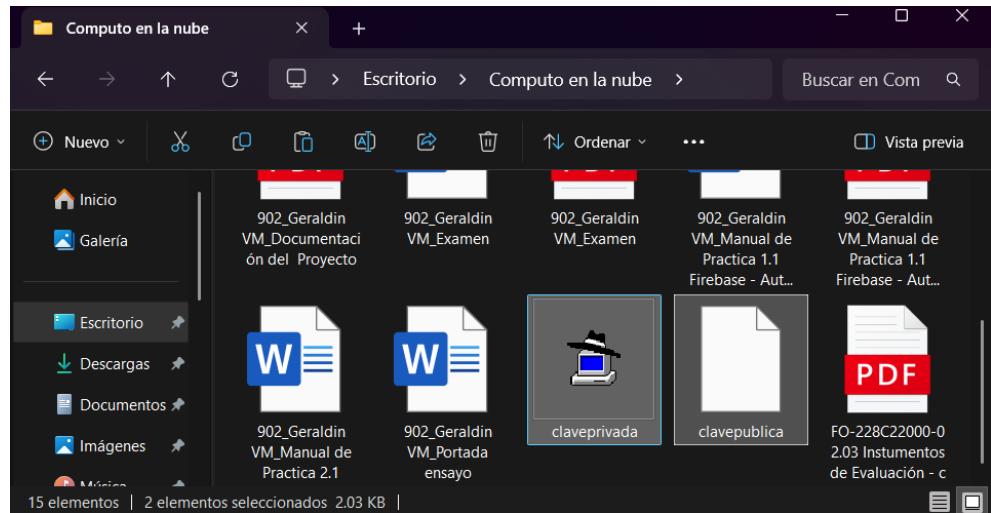


Ilustración 59. Claves guardadas.

Fuente: Elaboración Propia.: Elaboración Propia.

- Una vez guardadas, se cierra PuTTYgen.

Crear la máquina virtual en Azure:

- Acceder al portal de **Microsoft Azure** y crear un nuevo recurso.

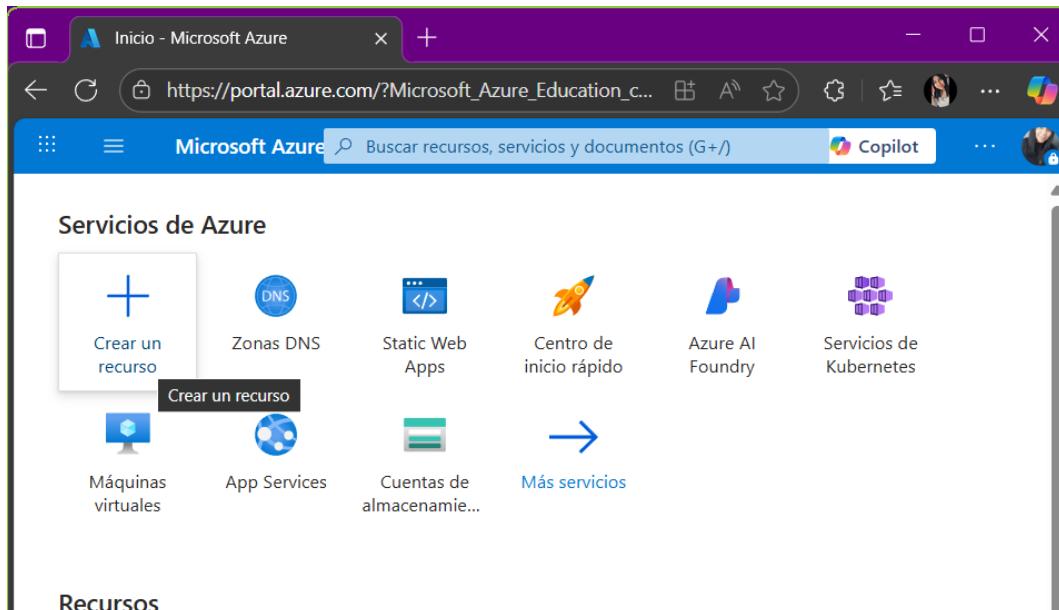


Ilustración 60. Crear un nuevo recurso.

Fuente: Elaboración Propia.: Elaboración Propia.

- Seleccionar **Máquina virtual**.



Ilustración 61. Seleccionar Máquina virtual.

Fuente: Elaboración Propia.: Elaboración Propia.

- En el grupo de recursos, seleccionar uno existente.

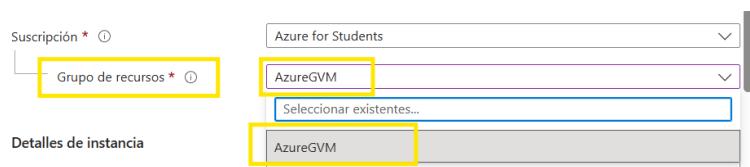


Ilustración 62. Asignar un grupo de recursos.

Fuente: Elaboración Propia.: Elaboración Propia.

- Agregar un nombre a la máquina virtual, en la región, se elige **Canadá Central**.

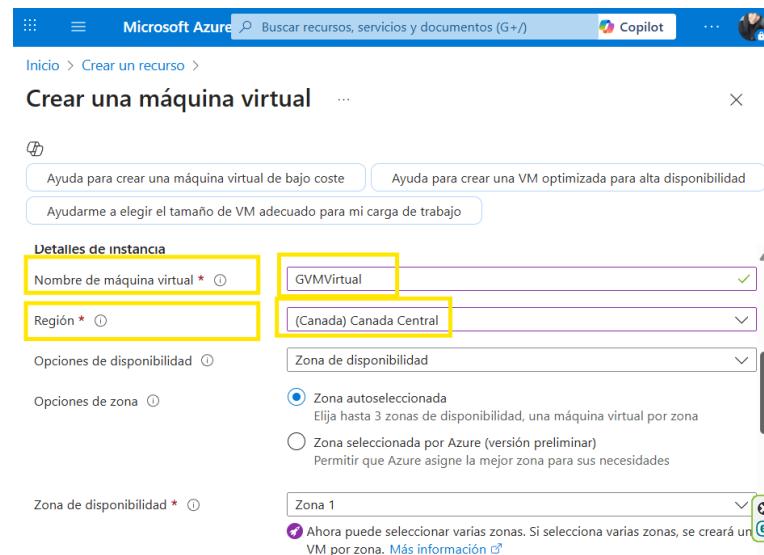


Ilustración 63. Asignar nombre y región.

Fuente: Elaboración Propia.: Elaboración Propia.

- En nombre de usuario se escribe: azuregeraldin.

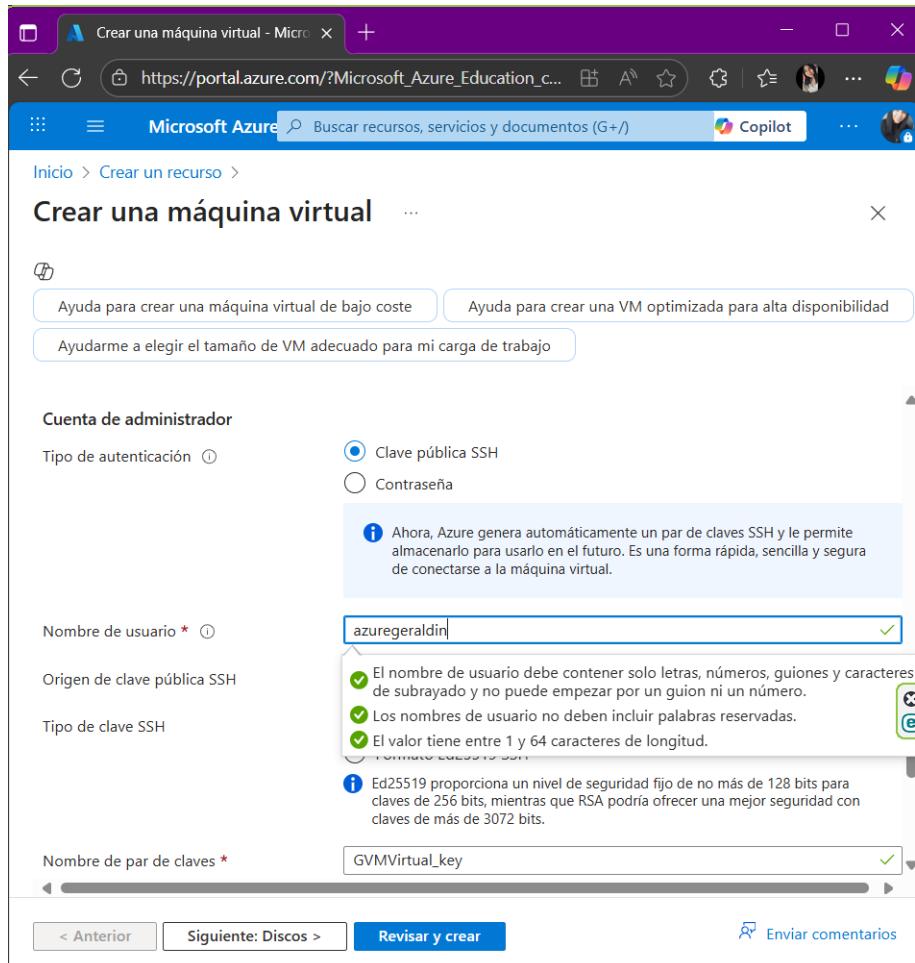


Ilustración 64. Asignar nombre de usuario.

Fuente: Elaboración Propia.: Elaboración Propia.

- En **Origen de clave pública SSH**, seleccionar **Usar la clave pública existente**.

The screenshot shows the 'Origen de clave pública SSH' dropdown menu. The option 'Usar la clave pública existente' is selected. Other options available are 'Generar un par de claves nuevo' and 'Usar la clave existente almacenada en Azure'.

Ilustración 65. Seleccionar un Origen de clave pública SSH.

Fuente: Elaboración Propia.: Elaboración Propia.

- Pegar la clave pública generada previamente.

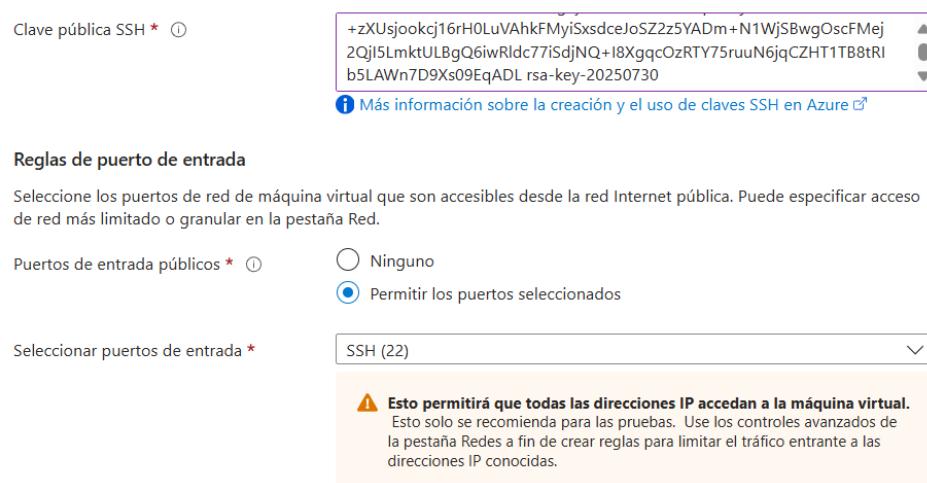


Ilustración 66. Pegar clave pública SSH.

Fuente: Elaboración Propia.: Elaboración Propia.

- En el tipo de disco para el sistema operativo, seleccionar: **SSD estándar (almacenamiento con redundancia local)**.

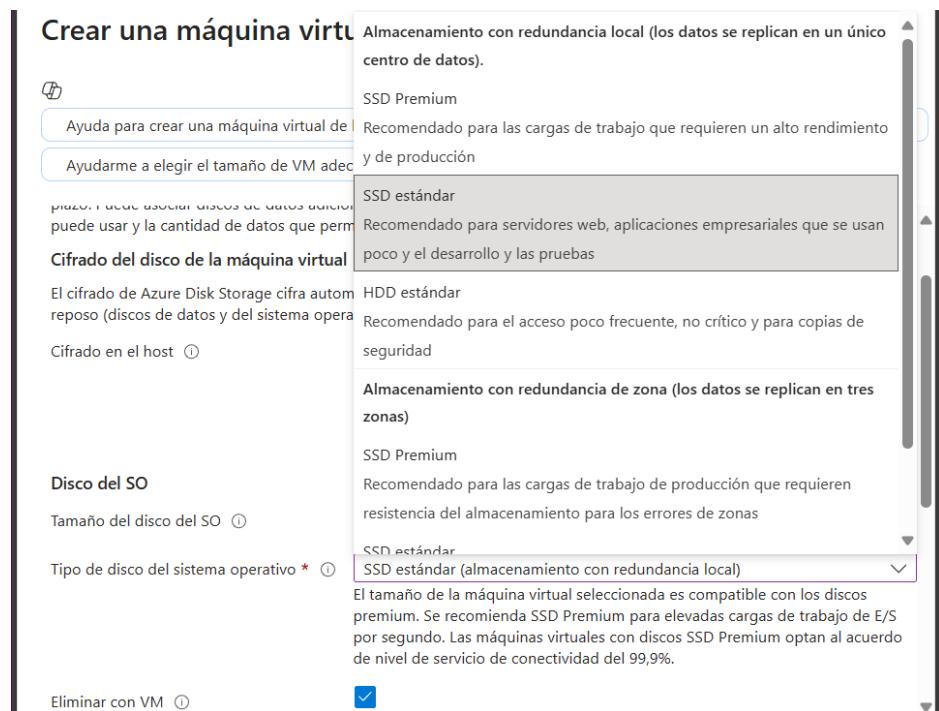


Ilustración 67. Seleccionar un tipo de disco del SO.

Fuente: Elaboración Propia.: Elaboración Propia.

- Mantener los demás parámetros por defecto y hacer clic en **Revisar y crear**.



Ilustración 68. Revisar y Crear.

Fuente: Elaboración Propia.: Elaboración Propia.

- Una vez se muestre el mensaje **Validación superada**, hacer clic en **Crear**.

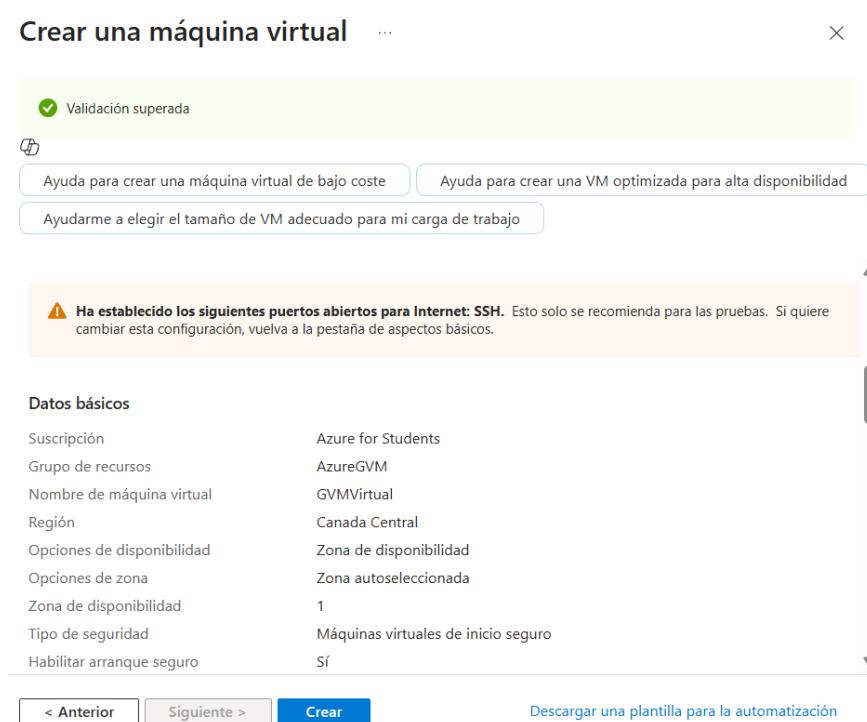


Ilustración 69. Crear máquina virtual.

Fuente: Elaboración Propia.: Elaboración Propia.

- Durante el proceso se mostrará "La implementación está en curso" hasta que finalice con "Se completó la implementación".

Información general

✓ Se completó la implementación

Nombre de implementación: CreateVm-canonical.ubuntu-24_04-lts... Hora de inicio: 30/7/2025, 14:16:48
Suscripción: Azure for Students Id. de correlación: 0427835d-f89d-45eb-b143-d37dc641b476

Detalles de implementación

Recurso	Tipo	Estado	Detalles de la operación
GVMVirtual	Microsoft.Compute/virtualMachi...	OK	Detalles de la operación
gvmvirtual975_z1	Microsoft.Network/networkInter... Created	Created	Detalles de la operación
GVMVirtual-nsg	Microsoft.Network/networkSecu...	OK	Detalles de la operación
GVMVirtual-vnet	Microsoft.Network/virtualNetwo...	OK	Detalles de la operación
GVMVirtual-ip	Microsoft.Network/publicIpAddr...	OK	Detalles de la operación

Pasos siguientes

- Configurar el apagado automático Recomendado
- Supervisar el estado, el rendimiento y las dependencias de red de la máquina virtual Recomendado
- Ejecutar un script dentro de la máquina virtual Recomendado

Botones: Ir al recurso | Crear otra VM

Ilustración 70. Creación correcta.

Fuente: Elaboración Propia.: Elaboración Propia.

- Hacer clic en **Ir al recurso** para consultar los detalles.

Información general

Información esencial

Grupo de recursos (mover) : AzureGVM
Estado : En ejecución
Ubicación : Canada Central (Zona 1)
Suscripción (mover) : Azure for Students
Id. de suscripción : 4971c5ae-3f7a-41a8-8914-48ae09821eb2
Zona de disponibilidad : 1
Etiquetas (editar) : Agregar etiquetas

Propiedades

Máquina virtual

Nombre del equipo	GVMVirtual
Sistema operativo	Linux (Ubuntu 24.04)
Generación de VM	V2
Arquitectura de VM	x64
Estado del agente	Ready
Versión del agente	2.14.0.1

Redes

Dirección IP pública	40.82.163.247 (Interfaz de red gvmvirtual975_z1)
Dirección IP pública (IPv6)	-
Dirección IP privada	10.0.0.4
Dirección IP privada (IPv6)	-
Red virtual/subred	GVMVirtual-vnet/default
Nombre DNS	Configurar

Ilustración 71. Detalles del recurso de la máquina creada.

Fuente: Elaboración Propia.: Elaboración Propia.

- Copiar la **Dirección IP pública** asignada a la máquina.

Sistema operativo	: Linux (Ubuntu 24.04)
Tamaño	: Stan (Copiar al Portapapeles 51B de memoria)
Dirección IP pública	: 40.82.163.247 
Red virtual/subred	: GVMVirtual-vnet/default
Nombre DNS	: Sin configurar
Estado de mantenimiento	: -
Hora de creación	: 30/7/2025, 20:16 UTC

Ilustración 72. Copiar dirección IP pública.

Fuente: Elaboración Propia.: Elaboración Propia.

Conectarse por SSH usando PuTTY:

- Abrir **PuTTY**.
- Ir a la sección: **Connection > SSH > Auth > Credentials**, y cargar la **clave privada** guardada anteriormente.

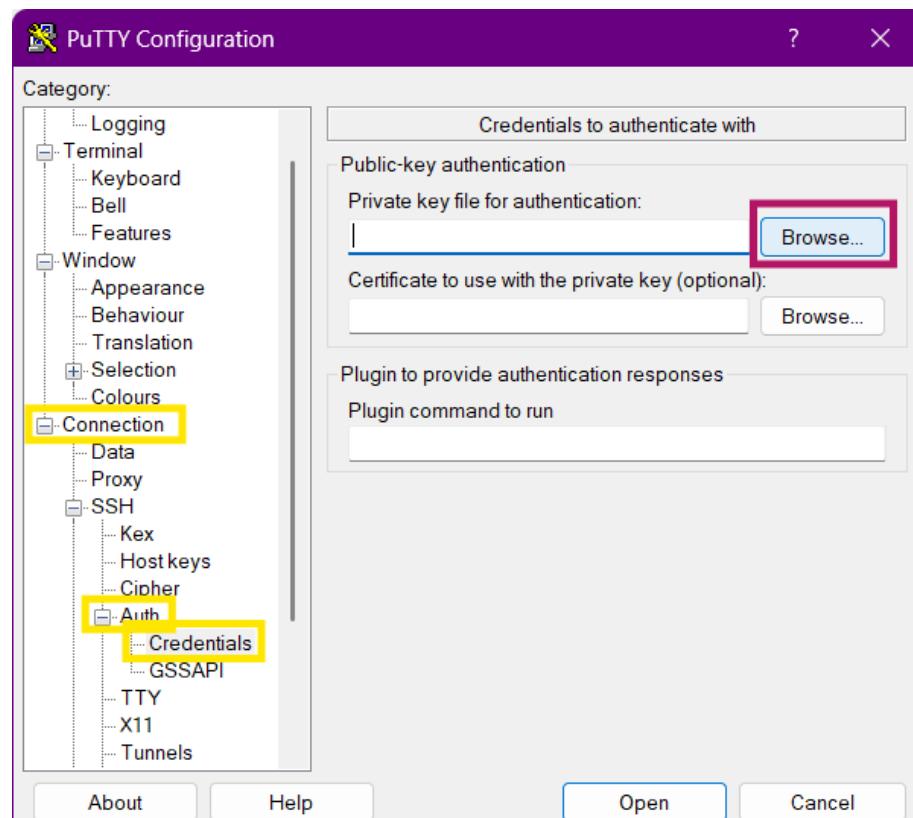


Ilustración 73. PuTTY Connection Auth > Credentials.

Fuente: Elaboración Propia.

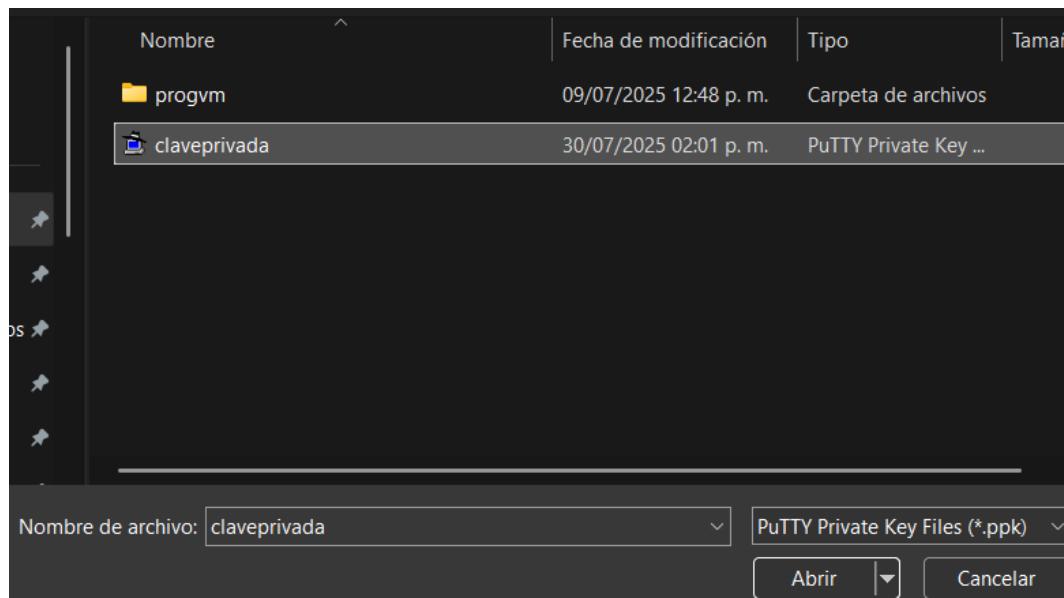


Ilustración 74. Selección de clave privada.

Fuente: Elaboración Propia.

- Luego ir a **Data** y colocar el nombre de usuario creado (azuregeraldin).

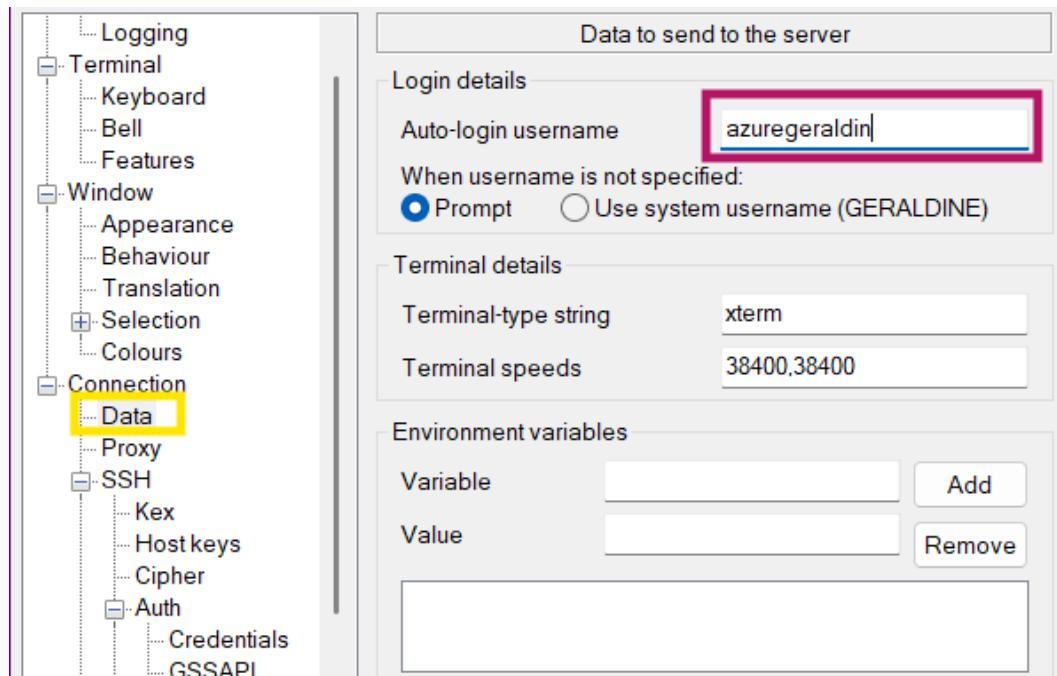


Ilustración 75. Ingresar nombre de usuario.

Fuente: Elaboración Propia.

- En la pestaña **Session**, introducir la IP pública, puerto **22**, tipo de conexión **SSH**, y guardar la sesión con un nombre (por ejemplo: **sesion_ubuntu**).

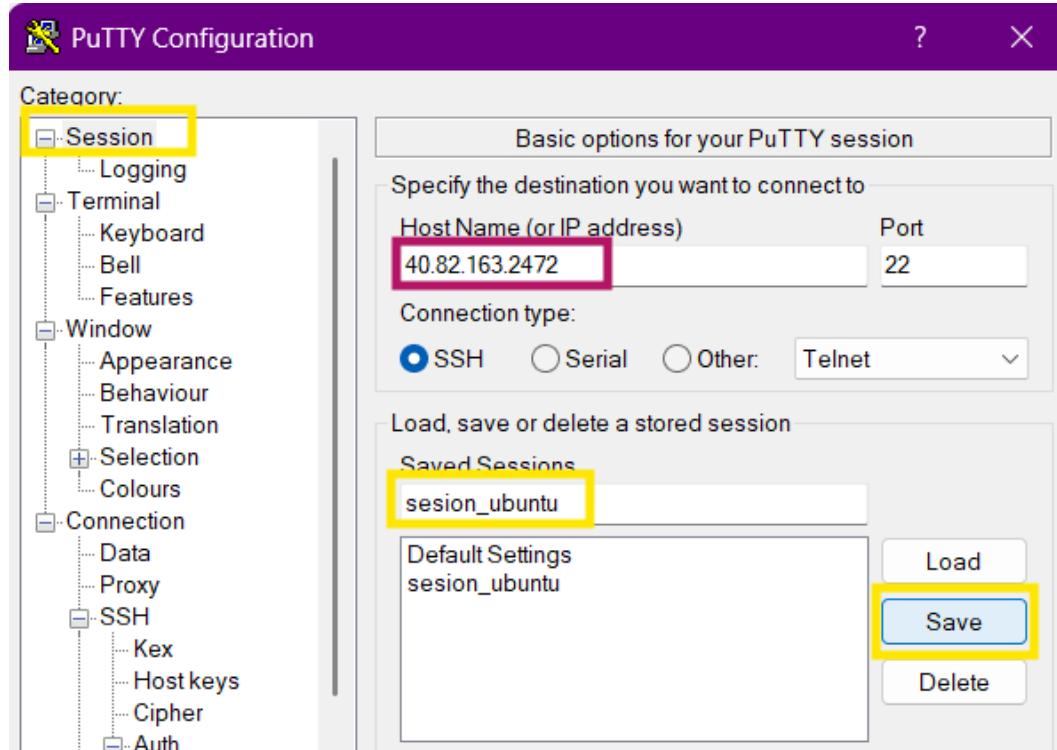


Ilustración 76. Ingresar dirección IP pública.

Fuente: Elaboración Propia.

- Hacer doble clic en la sesión guardada para iniciar la conexión.
- Ingresar la **frase de seguridad (passphrase)** cuando lo solicite.

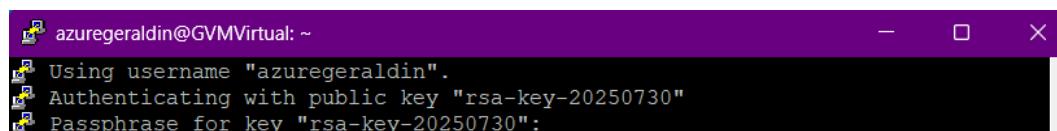
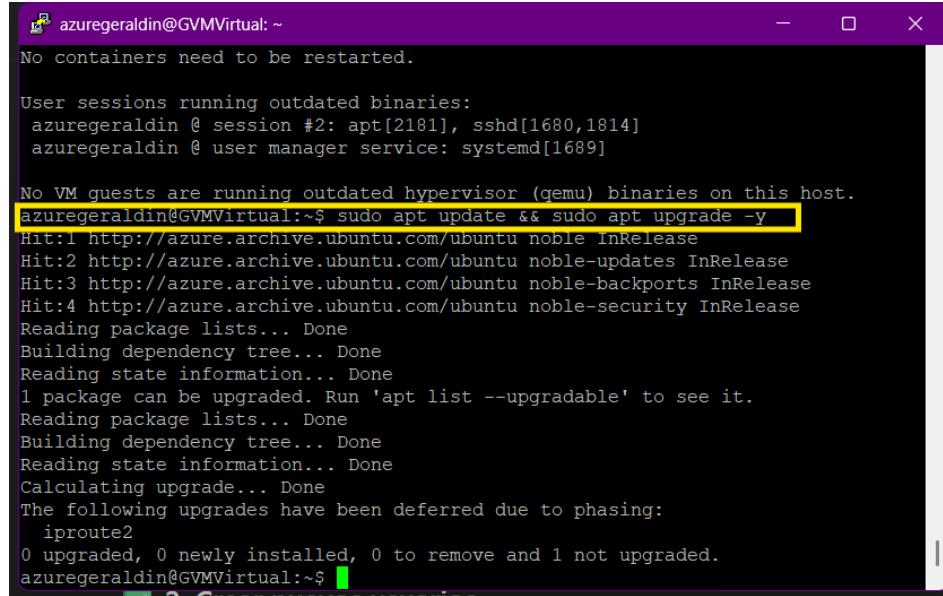


Ilustración 77. Ingresar frase.

Fuente: Elaboración Propia.

- Una vez dentro del sistema, vamos a actualizar y descargar los paquetes ingresando el siguiente comando: **sudo apt update && sudo apt upgrade -y**, como se muestra en la *ilustración 78*.



```

azuregeraldin@GVMVirtual: ~
No containers need to be restarted.

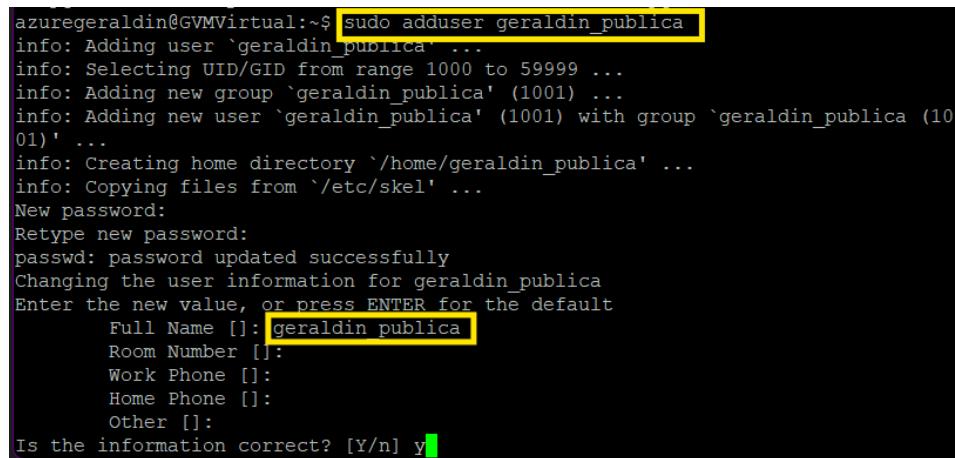
User sessions running outdated binaries:
azuregeraldin @ session #2: apt[2181], sshd[1680,1814]
azuregeraldin @ user manager service: systemd[1689]

No VM guests are running outdated hypervisor (gemu) binaries on this host.
azuregeraldin@GVMVirtual:~$ sudo apt update && sudo apt upgrade -y
Hit:1 http://azure.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://azure.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://azure.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://azure.archive.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following upgrades have been deferred due to phasing:
 iproute2
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
azuregeraldin@GVMVirtual:~$
```

Ilustración 78. Actualización de paquetes.

Fuente: Elaboración Propia.

- Crear un nuevo usuario: **sudo adduser geraldin_publica**, como se muestra en la *ilustración 79*.



```

azuregeraldin@GVMVirtual:~$ sudo adduser geraldin_publica
info: Adding user `geraldin_publica' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `geraldin_publica' (1001) ...
info: Adding new user `geraldin_publica' (1001) with group `geraldin_publica (1001)' ...
info: Creating home directory `/home/geraldin_publica' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for geraldin_publica
Enter the new value, or press ENTER for the default
    Full Name []: geraldin publica
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
```

Ilustración 79. Creación de user.

Fuente: Elaboración Propia.

- Crear una carpeta: **mkdir carpetita**.
- Crear un archivo dentro de la carpeta: **cd carpetita, touch archivito.txt**
- Editar el archivo con texto: **nano archivito.txt**
- Ingresar: **Hola soy Geral :)**, Guardar y salir (Ctrl + O, Enter y Ctrl + X), como se muestra en la *ilustración 80*.



Ilustración 80. Creación de archivo y carpeta.

Fuente: Elaboración Propia.

- Verificar el contenido del archivo: **cat archivito.txt**

```
azuregeraldin@GVMVirtual:~$ cd carpetita/
azuregeraldin@GVMVirtual:~/carpetita$ ls
archivito.txt
azuregeraldin@GVMVirtual:~/carpetita$ cat archivito.txt
Holaaaaa soy Geral :)
azuregeraldin@GVMVirtual:~/carpetita$
```

Ilustración 81. Mostrar contenido del archivo.

Fuente: Elaboración Propia.

Creación de Cuentas de Almacenamiento en Microsoft Azure

Una vez finalizada la conexión a la máquina virtual, se procedió a crear una cuenta de almacenamiento en Azure con los siguientes pasos:

1. En el portal de Azure, buscar **Cuentas de almacenamiento** y hacer clic en **Crear**.

The screenshot shows the Microsoft Azure portal interface. The top navigation bar is blue with the text "Microsoft Azure" and a search bar. Below it, the breadcrumb navigation shows "Inicio > Cuentas de almacenamiento". The main content area is titled "Cuentas de almacenamiento" and shows a list of storage accounts under "Tecnológico de Estudios Superiores de San Felipe del Progreso (sfelipeprogreso.tecnm.mx)". The list includes columns for Name, Type, Status, and Created. There are buttons for "Crear", "Restaurar", "Administrar vista", "Actualizar", "Exportar a CSV", "Abrir consulta", "Asignar etiquetas", and "Eliminar". A message at the top says "Está viendo una nueva versión de la experiencia de exploración. Haga clic aquí para acceder a la experiencia anterior." Below the list are three filter buttons: "Filtrar por cualquier ca...", "Suscripción es igual a todo", and "Grupo de recursos es igual a todo".

Ilustración 82. Apartado de cuentas de almacenamiento.

Fuente: Elaboración Propia.

2. Seleccionar un **grupo de recursos** ya existente, así mismo asignar un nombre a la cuenta de almacenamiento, en este caso: **gvmalmacenamiento**, y en **Región**, seleccionar: **(US) West US**, como se muestra en la *ilustración 83*.

The screenshot shows the "Crear una cuenta de almacenamiento" (Create a storage account) wizard. The first step is "Suscripción *", which is set to "Azure for Students". The second step is "Grupo de recursos *", which has a dropdown menu with "AzureGVM" selected and "Crear nuevo" (Create new) option available. The third step is "Detalles de la instancia". It has two fields: "Nombre de la cuenta de almacenamiento" with the value "gvmalmacenamiento" and "Región *" with the value "(US) West US". Below the region field is a note: "Implementación en una zona extendida de Azure".

Ilustración 83. Asignar nombre a la cuenta de almacenamiento.

Fuente: Elaboración Propia.

3. En el apartado de **Servicio principal**, elegir **Azure Blob Storage o Azure Data Lake Storage Gen2**, en rendimiento, seleccionar: **Estándar**, y en tipo de redundancia, elegir: **Almacenamiento con redundancia geográfica (GRS)**, como se muestra en la *ilustración 84*.

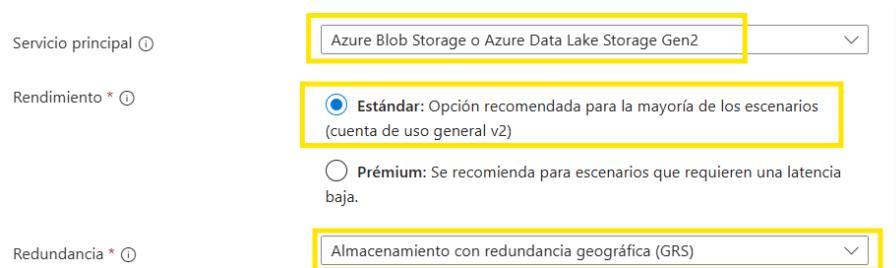


Ilustración 84. Configuración del servicio principal.

Fuente: Elaboración Propia.

4. Verificar que se encuentre habilitada la opción: **Acceso de lectura de datos...**, y hacer clic en **Revisar y crear**.

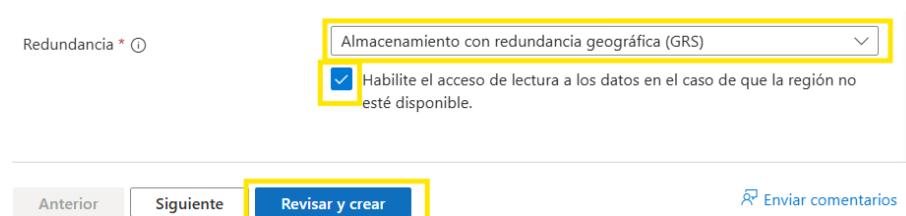


Ilustración 85. Habilitar Acceso de lectura de datos.

Fuente: Elaboración Propia.

5. Posteriormente dar clic en **Crear**.

Avanzado

Habilitar el espacio de nombres jerárquico	Deshabilitado
Habilitar SFTP	Deshabilitado
Habilitar el sistema de archivos de red v3	Deshabilitado
Permitir replicación entre inquilinos	Deshabilitado
Nivel de acceso	Hot



Ilustración 86. Crear la cuenta de almacenamiento.

Fuente: Elaboración Propia.

6. Una vez desplegada, hacer clic en **Ir al recurso**.

The screenshot shows the 'Información general' (General Information) page for a deployment named 'gvmalmacenamiento_1753909152175'. A green checkmark indicates that the implementation has completed successfully. Below this, there is a table titled 'Detalles de implementación' (Implementation Details) listing three resources: 'gvmalmacenamiento/default', 'gvmalmacenamiento/default', and 'gvmalmacenamiento'. All three resources are listed as 'OK' with their respective details. At the bottom of the page, there is a blue button labeled 'Ir al recurso' (Go to resource).

Ilustración 87. Creación de almacenamiento correcto.

Fuente: Elaboración Propia.

7. Dentro del recurso, hacer clic en **Cargar**.

The screenshot shows the 'Información general' (General Information) page for a storage account named 'gvmalmacenamiento'. In the top right corner, there is a blue button labeled 'Cargar' (Upload). Other buttons visible include 'Abrir en el Service Fabric Explorer' (Open in Service Fabric Explorer), 'Eliminar' (Delete), and 'Mover' (Move).

Ilustración 88. Ir al recurso > Crear.

Fuente: Elaboración Propia.

8. Arrastrar un archivo (imagen, PDF, etc.) al espacio de subida.

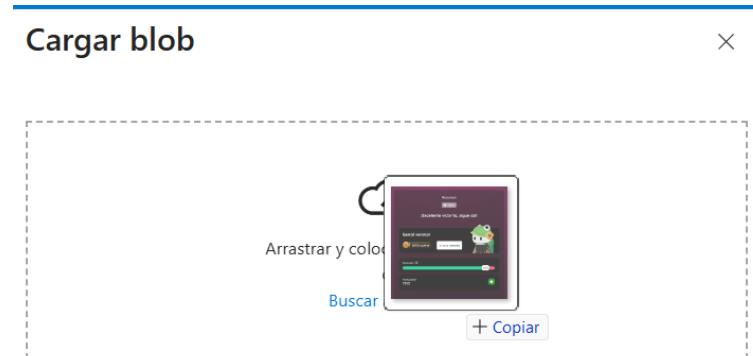


Ilustración 89. Cargar un archivo.

Fuente: Elaboración Propia.

9. Crear un nuevo contenedor:

- Nombre: documentos
- Nivel de acceso: **Privado**

Nuevo contenedor

Nombre

Nivel de acceso anónimo ⓘ

ⓘ El acceso anónimo a este contenedor se está bloqueando porque el acceso anónimo está deshabilitado en esta cuenta de almacenamiento.

Aceptar Cancelar

Ilustración 90. Crear un nuevo contenedor.

Fuente: Elaboración Propia.

10. Confirmar con el botón **Cargar**.

Cargar blob ×


1 archivo seleccionados: Examen.png
Arrastrar y colocar archivos aquí o [Buscar archivos](#)

Seleccione un contenedor existente

[Crear nuevo](#)

Sobrescribir los archivos si ya existen

▼ Avanzado

Cargar Enviar comentarios

Ilustración 91. Cargar blob.

Fuente: Elaboración Propia.

11. Aparecerá el mensaje: "Los blobs se han cargado correctamente".



Ilustración 92. Mensaje de que se cargaron los blobs correctamente.

Fuente: Elaboración Propia.

12. En el menú izquierdo, dirigirse a **Almacenamiento de datos > Contenedores**.

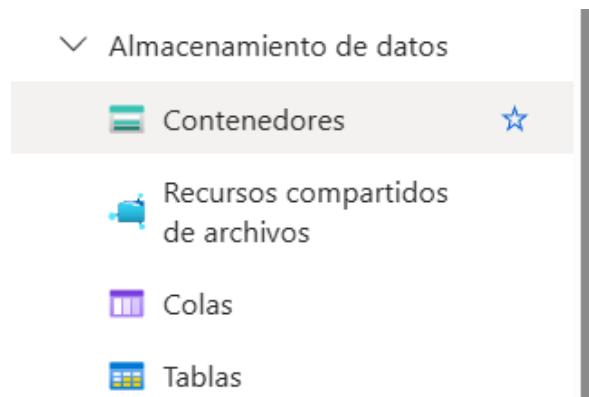


Ilustración 93. Dirigirse a Contenedores.

Fuente: Elaboración Propia.

13. Al abrir el contenedor llamado documentos, se visualiza el archivo cargado.

Mostrando todos los elementos 2				
<input type="checkbox"/>	Nombre	Última modificación	Nivel de acceso anónimo	Estado de con...
<input type="checkbox"/>	\$logs	30/7/2025, 2:59:54 p.m.	Privado	Disponible
<input type="checkbox"/>	documentos	30/7/2025, 3:01:59 p.m.	Privado	Disponible

Ilustración 94. Visualización de los Blobs.

Fuente: Elaboración Propia.

The screenshot shows the Azure Storage Explorer interface. At the top, there's a search bar labeled 'Buscar' and several action buttons: '+ Agregar directorio', 'Cargar', 'Cambiar nivel de acceso', and a three-dot menu. Below this is a sidebar with 'Información general' selected, followed by 'Diagnosticar y solucionar problemas', 'Control de acceso (IAM)', and 'Configuración'. The main area displays a list of blobs under the 'documentos' container. It includes a 'Método de autenticación' section with a link to 'Cambiar a la cuenta de usuario de Microsoft Entra'. There are filters for 'Agregar filtro' and 'Buscar blobs por prefijo'. A dropdown menu 'Solo mostrar blobs activos' is set to 'Mostrando todos los elementos 1'. The table lists one blob: 'Examen.png' with a modified date of '30/7/2025, 3:02:26 p.m.' and an access level of 'Frecuente (inferido)'.

Ilustración 95. Abrir el blob creado "documentos".

Fuente: Elaboración Propia.

14. Al dar clic sobre el archivo, se puede descargar directamente al ordenador.

The screenshot shows the properties for the 'Examen.png' blob. At the top, there are buttons for 'Guardar', 'Descartar', 'Descargar' (which is highlighted), 'Actualizar', 'Eliminar', and 'Cambiar nivel de acceso'. Below this is a navigation menu with 'Información general' selected, followed by 'Versiones', 'Instantáneas', 'Editar', and 'Generar SAS'. The main area displays various properties of the blob:

Propiedades	
URL	https://gvmalmacenamiento.blob.core.windows.net/documentos/Examen.png
ÚLTIMA MODIFICACIÓN	30/7/2025, 3:02:26 p.m.
HORA DE CREACIÓN	30/7/2025, 3:02:26 p.m.
ID. DE VERSIÓN	-
TIPO	Blob en bloques
TAMAÑO	96.29 KiB
NIVEL DE ACCESO	Frecuente (inferido)
ÚLTIMA MODIFICACIÓN DEL NIVEL DE ACCESO	N/D
ESTADO DEL ARCHIVO	-
PRIORIDAD DE REHIDRATACIÓN	-
SERVIDOR CIFRADO	true
ETAG	0x8DDCFAC632C16F6
DIRECTIVA DE INMUTABILIDAD DE NIVEL DE VERSIÓN	Deshabilitado
CONTROL DE CACHÉ	<input type="text"/>
TIPO DE CONTENIDO	image/png
CONTENT-MD5	WNnLx4l8rhjbKCWrzbvFOw...
CODIFICACIÓN DEL CONTENIDO	<input type="text"/>
IDIOMA DEL CONTENIDO	<input type="text"/>

Ilustración 96. Abrir y descargar el archivo subido a la carpeta de documentos.

Fuente: Elaboración Propia.

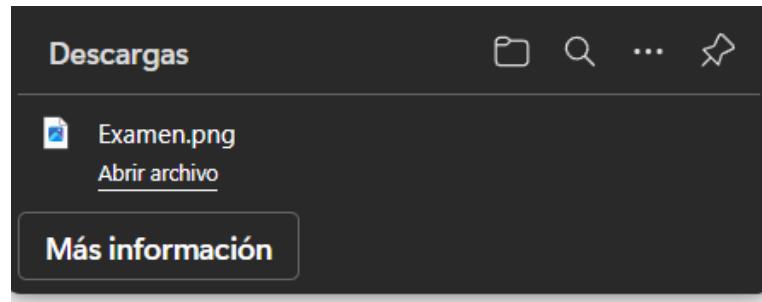


Ilustración 97. Descarga correcta.

Fuente: Elaboración Propia.

15. Se puede repetir el proceso de carga para subir múltiples archivos de distintos formatos.

Conexión remota y transferencia de archivos a la máquina virtual de Azure

A continuación, vamos a realizar una conexión entre nuestra máquina física y la máquina virtual de Ubuntu creada anteriormente, una vez realizada la conexión procedemos a transferir archivos de una maquina a otra, para ello se hará uso del software WinSCP con los siguientes pasos:

- Descargar el software WinSCP e iniciar el proceso de instalación.
- Durante el proceso de instalación automáticamente nos detecta la sesión guardada de la máquina virtual, para ello lo habilitamos a aceptamos.

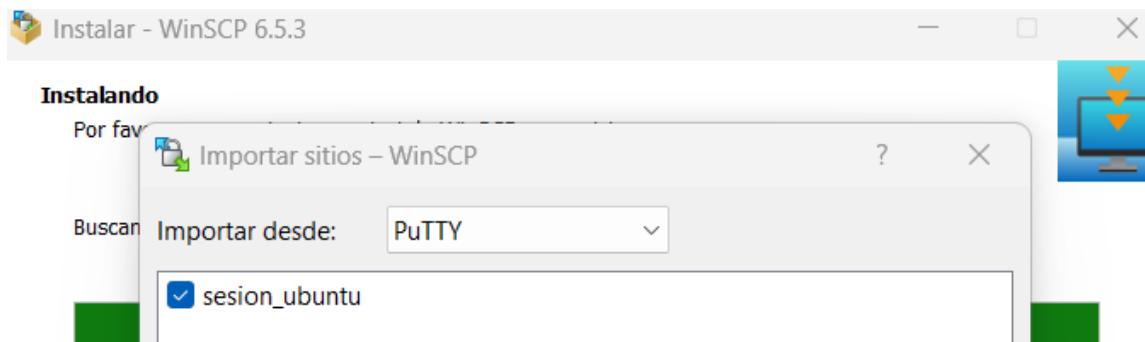


Ilustración 98. Instalación de WinSCP.

Fuente: Elaboración Propia.

- Finalizar instalación.



Ilustración 99. Finalizar instalación.

Fuente: Elaboración Propia.

- Al abrir WinSCP ya nos aparece la conexión guardada, para conectarnos dar clic en “Conectar”.

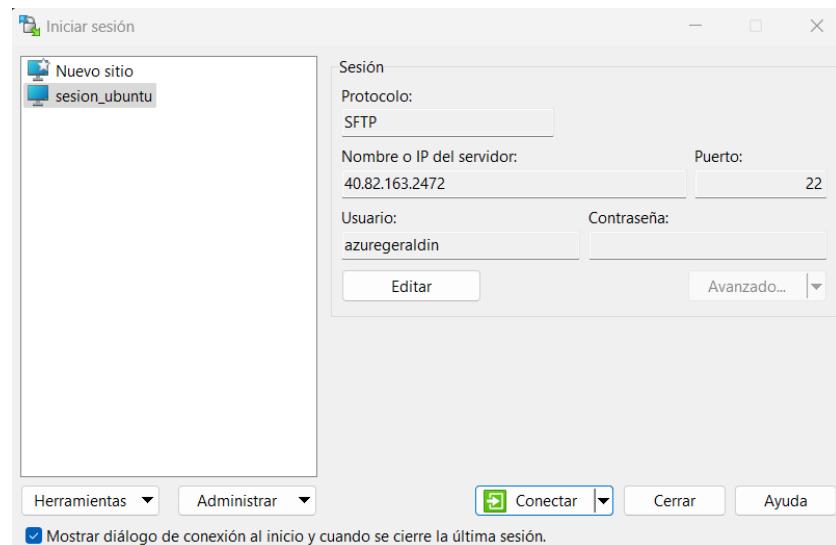


Ilustración 100. Conexión guardada.

Fuente: Elaboración Propia.

- En mi caso al realizar el paso anterior me marcaba error, por lo tanto, edite los datos de la conexión e ingrese nuevamente los datos guarde y conecte. Durante este proceso nos aparece el siguiente aviso dar clic en Aceptar.

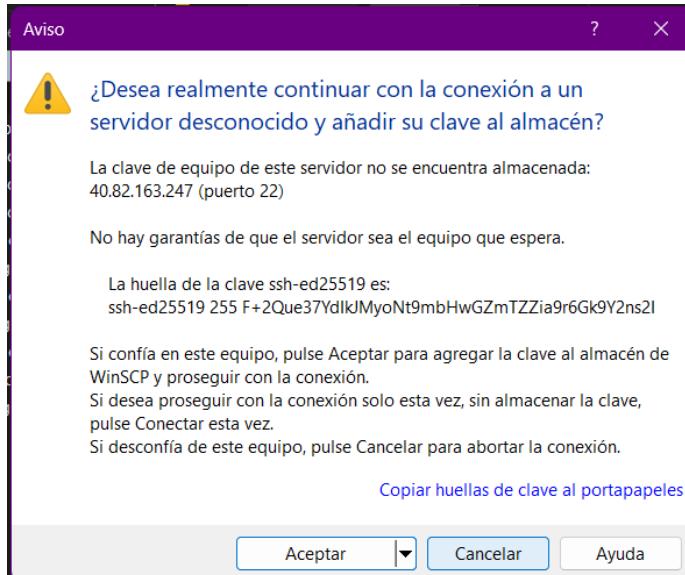


Ilustración 101. Aviso de realizar la conexión.

Fuente: Elaboración Propia.

- Ingresar la frase o contraseña y dar clic en aceptar.

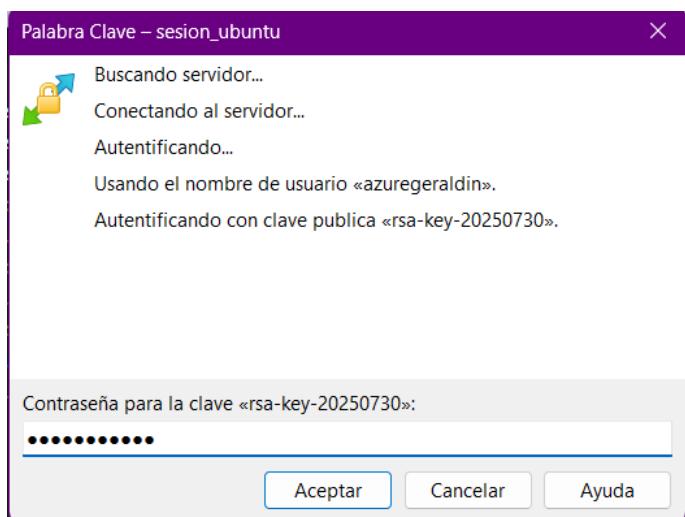


Ilustración 102. Ingresar frase.

Fuente: Elaboración Propia.

- Al dar clic en Aceptar nos aparecerá automáticamente la conexión entre las máquinas, en el lado izquierdo se muestran los documentos de la máquina física y en el lado derecho se visualizan los archivos que tenemos en la máquina virtual de **azuregeraldin**.

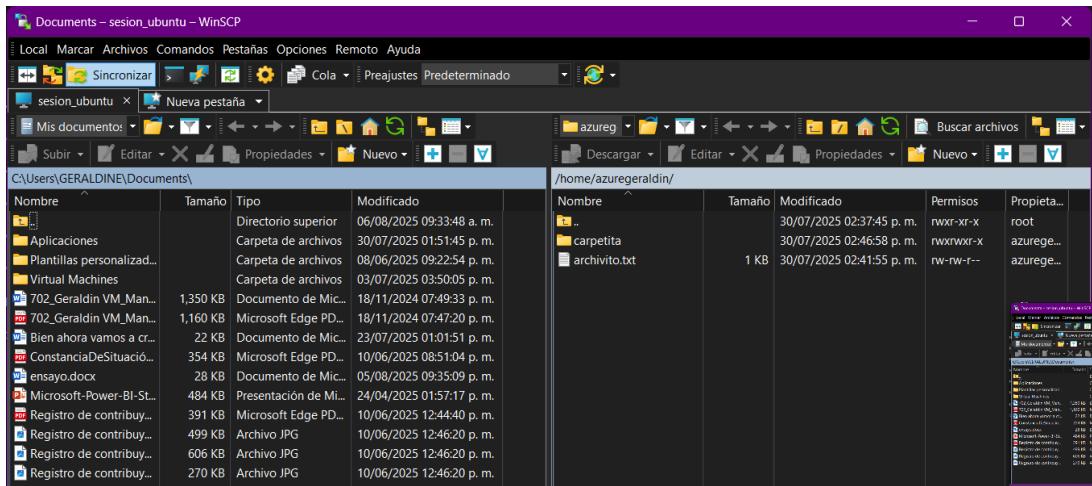


Ilustración 103. Conexión entre máquinas correcta.

Fuente: Elaboración Propia.

- Procedemos a compartir un archivo de la máquina física a la máquina virtual, para ello solo arrastramos el documento y nos aparece la siguiente ventana de confirmación.

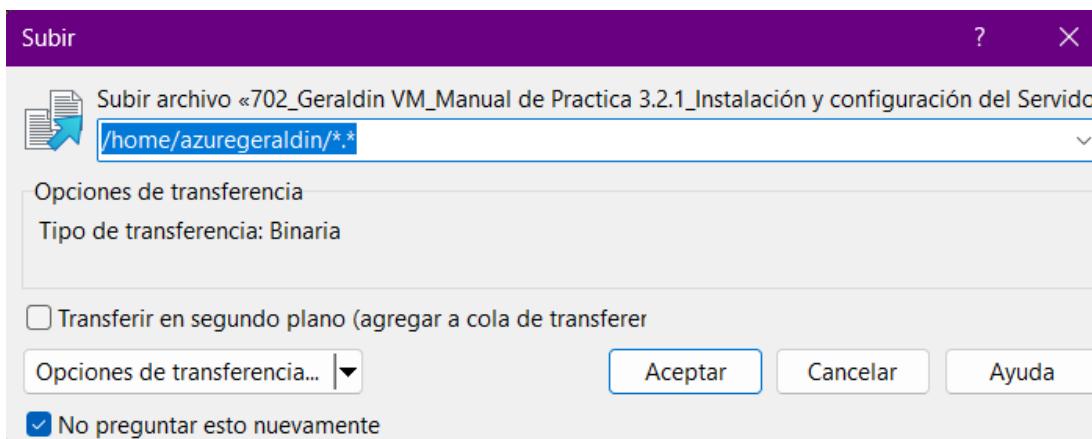


Ilustración 104. Transferencia de documento de la máquina física a la virtual.

Fuente: Elaboración Propia.

- Se realiza la transferencia.

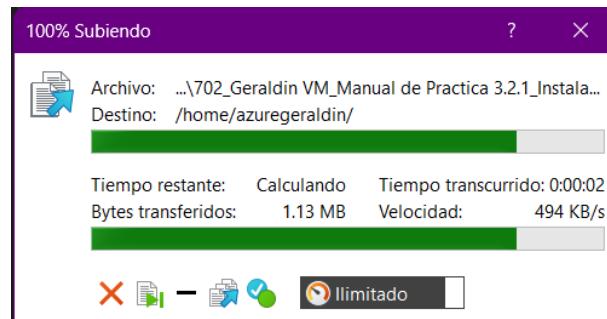


Ilustración 105. Transferencia realizada correctamente.

Fuente: Elaboración Propia.

- Se realiza una transferencia del documento “archivito.txt” de la máquina virtual a la máquina física, como se muestra en la *ilustración 106*.

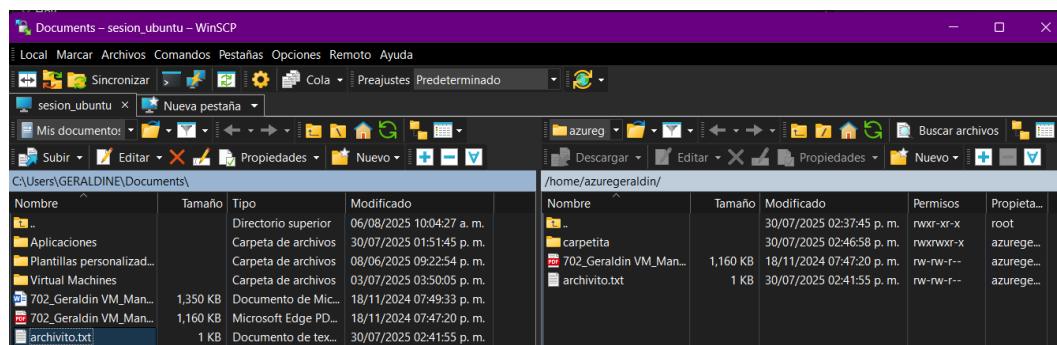


Ilustración 106. Transferencia de documento de la máquina virtual a la física.

Fuente: Elaboración Propia.

- Como último paso nos dirigimos a Putty en la máquina virtual mostramos los documentos, como se muestra en la *ilustración 107*.

```
azuregeraldin@GVMVirtual:~$ ls
'702_Geraldin VM_Manual de Practica 3.2.1_Instalación y configuración del Servidor FTP en Ubuntu.pdf'
archivito.txt
carpetita
azuregeraldin@GVMVirtual:~$
```

Ilustración 107. Visualización de los documentos en la máquina virtual.

Fuente: Elaboración Propia.

Análisis de la práctica

Durante la práctica se llevó a cabo un recorrido integral por diversas herramientas y servicios en la nube, comenzando con la creación de una cuenta en Azure para estudiantes, lo cual permitió acceder a recursos gratuitos y específicos para el desarrollo de proyectos académicos y profesionales. Posteriormente, se creó una cuenta en GitHub, plataforma esencial para el control de versiones y colaboración, donde se estableció un repositorio que serviría como base para el desarrollo del proyecto web.

Una vez listo el repositorio, se configuró una Static Web App en Microsoft Azure, integrándola directamente con GitHub para habilitar un flujo de trabajo continuo y automatizado. Esto facilitó que cualquier modificación realizada en el proyecto local se reflejara en tiempo real en la aplicación desplegada, garantizando un proceso ágil de desarrollo y pruebas.

Para personalizar y profesionalizar la presencia web, se procedió a crear un dominio personalizado, realizando la configuración de los registros DNS necesarios tanto en Azure como en Firebase, asegurando la correcta vinculación y resolución del dominio para la aplicación web. Además, se configuró una máquina virtual en Azure, lo que permitió comprender el manejo de infraestructura virtualizada y sus aplicaciones prácticas, así como la gestión de recursos computacionales en la nube.

Posteriormente, se implementaron cuentas de almacenamiento en Microsoft Azure, fundamentales para manejar y conservar datos de manera escalable y segura, complementando así los servicios de cómputo y hosting utilizados durante la práctica. En conjunto, estas actividades permitieron adquirir experiencia práctica en la administración y despliegue de aplicaciones en la nube, así como en la integración de múltiples servicios que garantizan un entorno de desarrollo moderno y eficiente. Por último se realizó una práctica sencilla donde se realizó la conexión entre la máquina física y la máquina virtual creada en Azure, lo que permitió la transferencia de archivos entre máquinas.



TECNOLÓGICO
NACIONAL DE MÉXICO

TES SFP
TECNOLÓGICO DE ESTUDIOS SUPERIORES
SAN FELIPE DEL PROGRESO

TECNOLÓGICO DE ESTUDIOS SUPERIORES SAN FELIPE
DEL PROGRESO

Ingeniería Informática

CÓMPUTO EN LA NUBE

Prácticas del 3er Corte

Semestre:

Julio - Agosto 2025 (Periodo: 07 de Julio – 30 de Agosto
2025)

Alumna: Geraldin Venegas Margarito

No. Cuenta: 2021330023

Grupo: 902

Docente: Ing. Lizeth José Ángeles

San Felipe del Progreso a 13 de Agosto de 2025

Índice del contenido:

Introducción.....	5
Objetivo de la práctica	5
Desarrollo	6
Requerimientos.....	6
Vercel Hosting	6
Autenticación	14
Base de datos.....	21
Subir archivos	26
Funcionalidad de Edición y Eliminación de Registros	31
Análisis de la practica.....	37

Índice de Ilustraciones

Ilustración 1. Crear Repositorio.....	6
Ilustración 2. Seleccionar Clone repository.....	7
Ilustración 3. Sellecionar repositorio a clonar	7
Ilustración 4. Seleccionar la carpeta y abrirla en Visual Studio.....	8
Ilustración 5. Creación del Index con el código del login.	8
Ilustración 6. Creación del styles.....	9
Ilustración 7. Crear commit con los cambios realizados.	10
Ilustración 8. Archivos subidos al repositorio.	10
Ilustración 9. Inicio de sesión en Vercel con GitHub.	11
Ilustración 10. Importar proyecto de GitHub.	11
Ilustración 11. Seleccionar el repositorio cloudGVM.	12
Ilustración 12. Deploy.....	12
Ilustración 13. Despliegue del proyecto en Vercel con dominio público.	13
Ilustración 14. Abrir el link de vercel.....	13
Ilustración 15. Ingresar a Supabase con GitHub.....	14
Ilustración 16. Crear un nuevo proyecto / organización.	14
Ilustración 17. Apartado de Autenticación.....	15
Ilustración 18. Configuración de proveedores de autenticación.....	15
Ilustración 19. Copiar URL del proyecto.	16
Ilustración 20. Copiar la clave pública.....	16
Ilustración 21. Crear capreta js con los archivos auth y dashboard.	16
Ilustración 22. Ingresar Url y clave en los archivos js.	17
Ilustración 23. Dashboard del panel del estudiante.....	17
Ilustración 24. Subir commit con las actualizaciones.....	18
Ilustración 25. Actualización subida al repositorio.....	18
Ilustración 26. Abrir dominio.....	19
Ilustración 27. Registrar un nuevo usuario.....	19
Ilustración 28. Mensaje de correo no confirmado.	20
Ilustración 29. Mensaje de sesión iniciada.....	20
Ilustración 30. Usuario almacenado correctamente.	20
Ilustración 31. Apartado de Database.....	21
Ilustración 32. Nombre de la tabla: estudiantes.	21
Ilustración 33. Columnas de la tabla de estudiantes.....	22
Ilustración 34. Tabla de estudiantes en Editor de tablas.....	22
Ilustración 35. View polices.....	23
Ilustración 36. Política de SELECT.	23
Ilustración 37. Política de INSERT.	24
Ilustración 38. Políticas de acceso creadas para el registro de estudiantes.....	24
Ilustración 39. Mensaje de estudiante registrado correctamente.	25
Ilustración 40. Visualización del estudiante en la lista.	25
Ilustración 41. Visualización en la base de datos.	25

Ilustración 42. Apartado de Storage.....	26
Ilustración 43. Nuevo bucket llamado: tareas.....	26
Ilustración 44. Crear nueva política para tareas.....	27
Ilustración 45. Política de tareas: Permitir acceso.....	27
Ilustración 46. Política de tareas: Permitir insertar.	28
Ilustración 47. Elegir archivo a subir.....	28
Ilustración 48. Archivo subido correctamente.	29
Ilustración 49. Visualización de archivos subidos.....	29
Ilustración 50. Subida y visualización de archivos en el panel.	30
Ilustración 51. Visualización de almacenamiento en Supabase.....	30
Ilustración 52. Creación de política actualizar en la tabla estudiantes.	31
Ilustración 53. Creación de política actualizar en la tabla estudiantes.	32
Ilustración 54. Generar commit con los cambios para editar y eliminar.	33
Ilustración 55. Cambios subidos al repositorio. Fuente: Elaboración Propia.	33
Ilustración 56. Visualización de estudiantes registrados previamente.	34
Ilustración 57. Relleno de formulario automáticamente para editar.	34
Ilustración 58, Mensaje de estudiante actualizado.....	35
Ilustración 59. Estudiante actualizado correctamente.....	35
Ilustración 60. Mensaje de confirmación para eliminar.	36
Ilustración 61. Mensaje de estudiante eliminado.	36
Ilustración 62. Estudiante eliminado correctamente.	36

Introducción

En la presente práctica se implementará un sistema web integrando servicios de autenticación, base de datos y almacenamiento de archivos, además de funcionalidades avanzadas de gestión de registros. El objetivo es lograr que el usuario pueda registrarse e iniciar sesión de manera segura, almacenar y consultar datos en una base de datos en la nube, así como subir archivos que queden vinculados a su cuenta. Una vez que el sistema cuente con registros almacenados, se añadirá la posibilidad de editarlos o eliminarlos directamente desde la interfaz, garantizando que cada usuario solo pueda modificar su propia información. El despliegue del proyecto se realizará en **Vercel Hosting**, aprovechando su facilidad para implementar aplicaciones web directamente desde un repositorio de código, permitiendo así actualizaciones rápidas y automáticas. De esta manera, el estudiante podrá comprender el ciclo completo de desarrollo y publicación de una aplicación web moderna, desde la configuración de los servicios backend hasta su disponibilidad en línea, fomentando buenas prácticas de seguridad, escalabilidad y usabilidad.

Objetivo de la práctica

Desarrollar y desplegar en **Vercel Hosting** un sistema web con autenticación de usuarios, gestión de datos en base de datos en la nube, carga de archivos y funcionalidades para editar y eliminar registros, asegurando que cada usuario solo pueda manipular su propia información.

Desarrollo

Requerimientos

- Acceso a internet.
- Navegador web actualizado (Google Chrome, Firefox, Edge, etc.)
- Cuenta en Vercel.
- Cuenta en la plataforma de base de datos (Supabase o similar).
- Editor de código (Visual Studio Code o similar).
- Repositorio en GitHub (o creación de uno nuevo).
- Herramientas para autenticación (por ejemplo, configuración de proveedores en Supabase Auth).
- Archivos y recursos para pruebas de carga (imágenes, documentos, etc.).

Vercel Hosting

1. Creación del Repositorio y Código Inicial.

1. Ingresar a GitHub y crear un repositorio llamado **cloudGVM**.

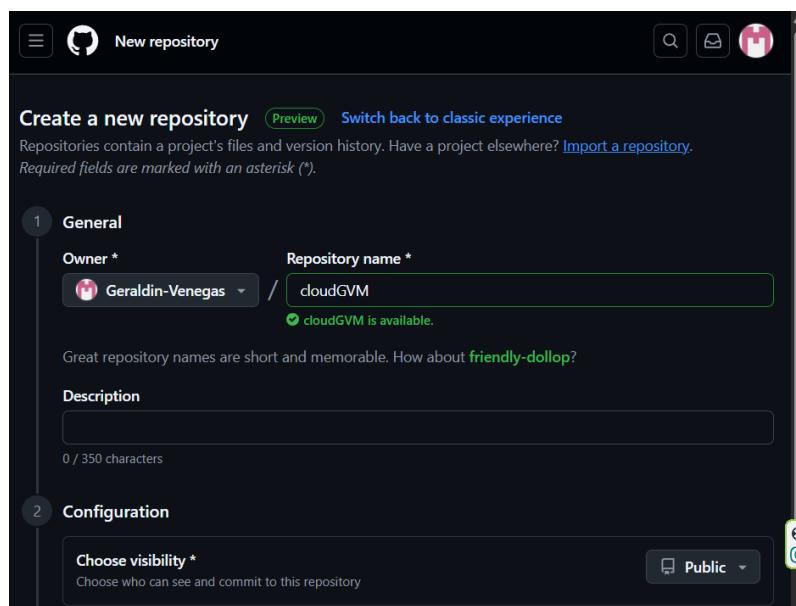


Ilustración 1. Crear Repositorio.

Fuente: Elaboración Propia.

2. En **GitHub Desktop**, ir a File > Clone repository, seleccionar el repositorio y dar clic en **Clone**.

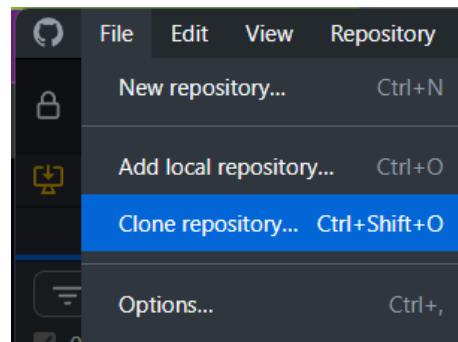


Ilustración 2. Seleccionar *Clone repository*.

Fuente: Elaboración Propia.

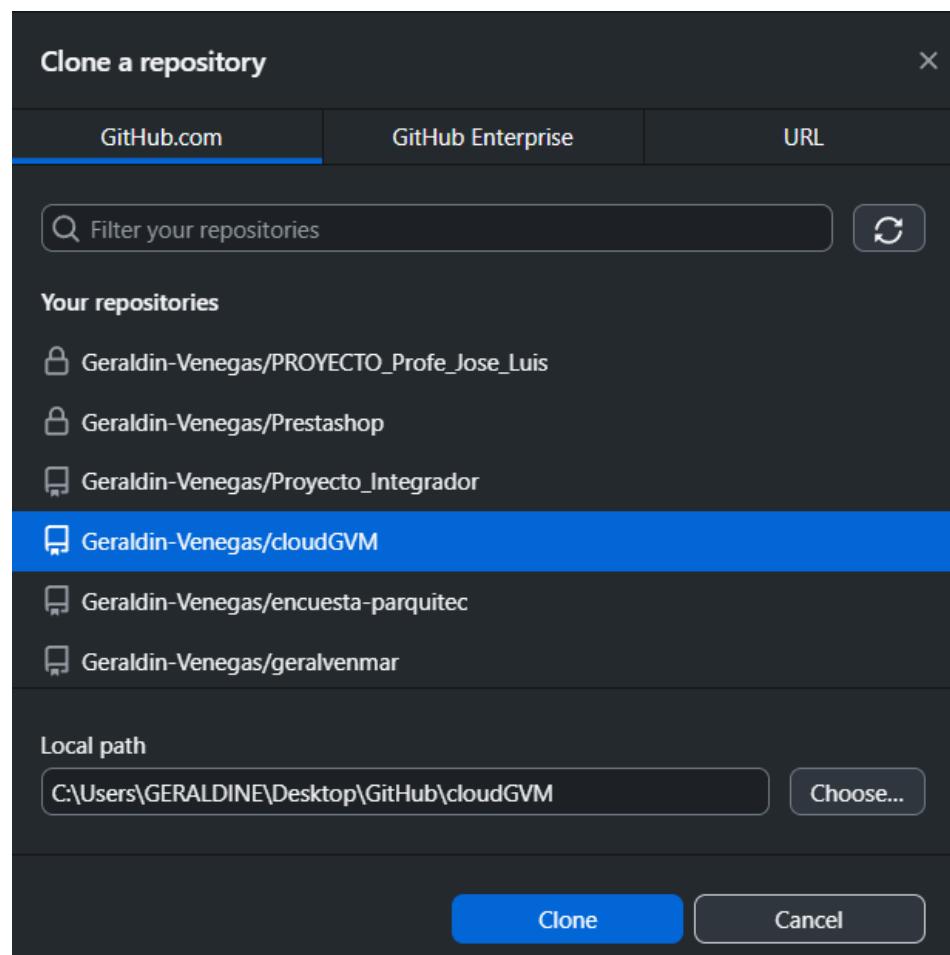


Ilustración 3. Seleccionar repositorio a clonar.

Fuente: Elaboración Propia.

3. Abrir la carpeta clonada en **Visual Studio Code** y crear:

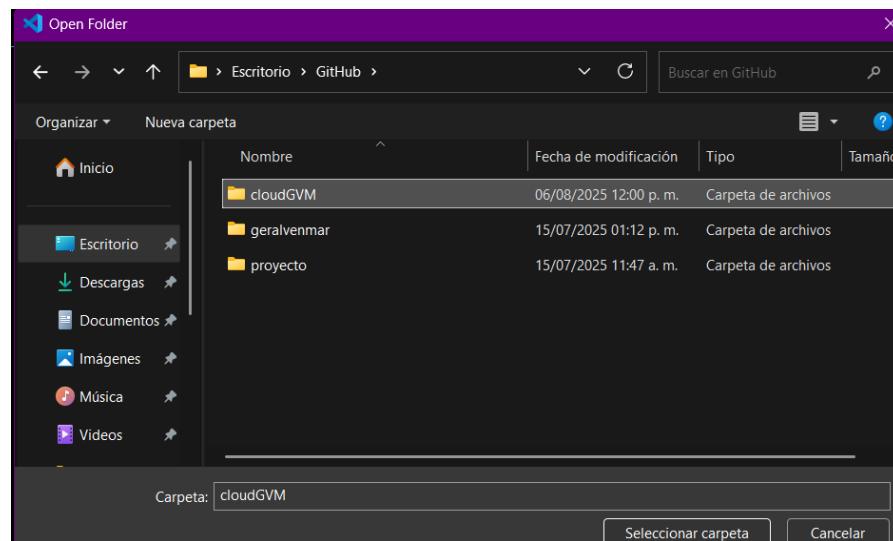


Ilustración 4. Seleccionar la carpeta y abrirla en Visual Studio.

Fuente: Elaboración Propia.

- **index.html** → Contendrá el formulario de inicio de sesión y registro.

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8" />
<title>Mi Clase en la Nube - Login</title>
<link rel="stylesheet" href="styles.css" />
</head>
<body>
<h1>Bienvenida a Mi Clase en la Nube</h1>
<div id="auth-container">
<div id="login-form">
<h2>Iniciar Sesión</h2>
<input type="email" id="email" placeholder="Correo" />
<input type="password" id="password" placeholder="Contraseña" />
<button onclick="login()">Entrar</button>
<p>¿No tienes cuenta? <a href="#" onclick="toggleForms()">Regístrate aquí</a></p>
</div>
<div id="register-form" style="display:none;">
<h2>Registrarse</h2>
<input type="email" id="reg-email" placeholder="Correo" />
<input type="password" id="reg-password" placeholder="Contraseña" />
<button onclick="register()">Crear cuenta</button>
<p>¿Ya tienes cuenta? <a href="#" onclick="toggleForms()">Inicia sesión aquí</a></p>
</div>
</div>
<script src="https://cdn.jsdelivr.net/npm/@supabase/supabase-js@2"></script>
<script src="js/auth.js"></script>
</body>
</html>
```

Ilustración 5. Creación del Index con el código del login.

Fuente: Elaboración Propia.

- **styles.css** → Contendrá los estilos de la página.

File Edit Selection View Go Run Terminal Help ← → ⌘

EXPLORER ... Welcome index.html # style.css X

✓ CLOUDGVM
↳ index.html
style.css

```
# style.css > ↳ img
1 body {
2   background-color: #f9f9f9;
3   font-family: 'Segoe UI', sans-serif;
4   display: flex;
5   justify-content: center;
6   padding: 40px;
7   margin: 0;
8 }
9
10 .contenedor {
11   background-color: white;
12   padding: 30px;
13   border-radius: 12px;
14   box-shadow: 0 5px 20px rgba(0,0,0,0.1);
15   width: 100%;
16   max-width: 600px;
17 }
18
19 h1, h2 {
20   color: #333;
21   margin-top: 20px;
22   text-align: center;
23 }
24
25 form {
26   display: flex;
27   flex-direction: column;
28   gap: 10px;
29   margin-bottom: 15px;
30 }
31
32 input, select {
33   padding: 10px;
34   font-size: 16px;
35   border: 1px solid #ccc;
36   border-radius: 8px;
37 }
```

> OUTLINE
> TIMELINE

Ilustración 6. Creación del styles.

Fuente: Elaboración Propia.

- Realizar un commit con el mensaje Código_de_inicio_de_sesión y publicarlo con **Publish Branch**.

The screenshot shows the GitHub Desktop application interface. At the top, it displays the repository "cloudGVM" and the branch "main". A message at the top indicates that an updated version of GitHub Desktop is available. The main area shows two changes: "index.html" and "style.css". The "index.html" file content is displayed in a diff view, showing code additions and modifications. A commit message box is open, containing the text "Código de inicio de sesión". Below the message box, there is a "Description" field and a "Commit 2 files to main" button.

```

@@ -0,0 +1,32 @@
1 + <!DOCTYPE html>
2 + <html lang="es">
3 + <head>
4 +   <meta charset="UTF-8" />
5 +   <title>Mi Clase en la Nube - Login</title>
6 +   <link rel="stylesheet" href="styles.css" />
7 + </head>
8 + <body>
9 +   <h1>Bienvenida a Mi Clase en la Nube</h1>
10 +
11 +   <div id="auth-container">
12 +
13 +     <div id="login-form">
14 +       <h2>Iniciar Sesión</h2>
15 +       <input type="email" id="email" placeholder="Correo" />
16 +       <input type="password" id="password" placeholder="Contraseña" />
17 +       <button onclick="login()">Entrar</button>
18 +       <p>¿No tienes cuenta? <a href="#" onclick="toggleForms()">Regístrate aquí</a></p>
19 +     </div>
20 +
21 +     <div id="register-form" style="display:none;">
22 +       <h2>Registrarse</h2>
23 +       <input type="email" id="reg-email" placeholder="Correo" />
24 +       <input type="password" id="reg-password" placeholder="Contraseña" />
25 +       <button onclick="register()">Crear cuenta</button>
26 +       <p>Ya tienes cuenta? <a href="#" onclick="toggleForms()">Inicia sesión aquí</a></p>
27 +     </div>
28 +
29 +   <script src="https://cdn.jsdelivr.net/npm/@supabase/supabase-js@2"></script>
30 +   <script src="js/auth.js"></script>
31 +
32 + </body>
</html> @@
```

Ilustración 7. Crear commit con los cambios realizados.

Fuente: Elaboración Propia.

5. Verificamos que los archivos se subieron en el repositorio.

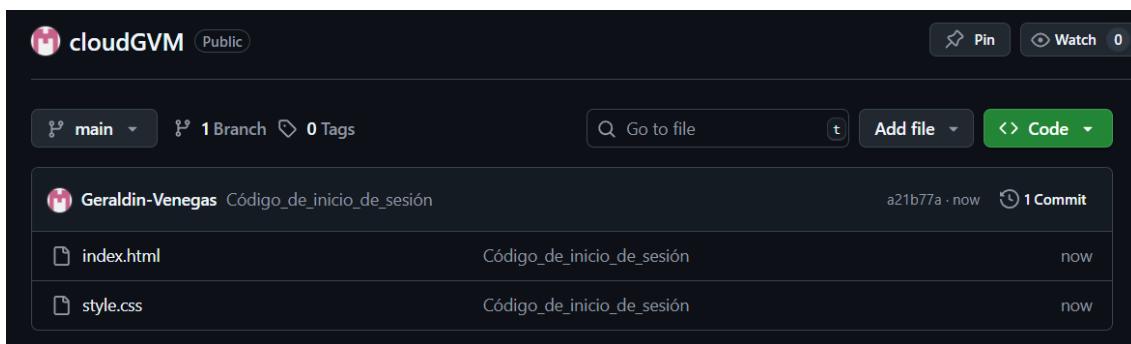


Ilustración 8. Archivos subidos al repositorio.

Fuente: Elaboración Propia.

2. Despliegue en Vercel

1. Entrar a vercel.com e iniciar sesión con GitHub.

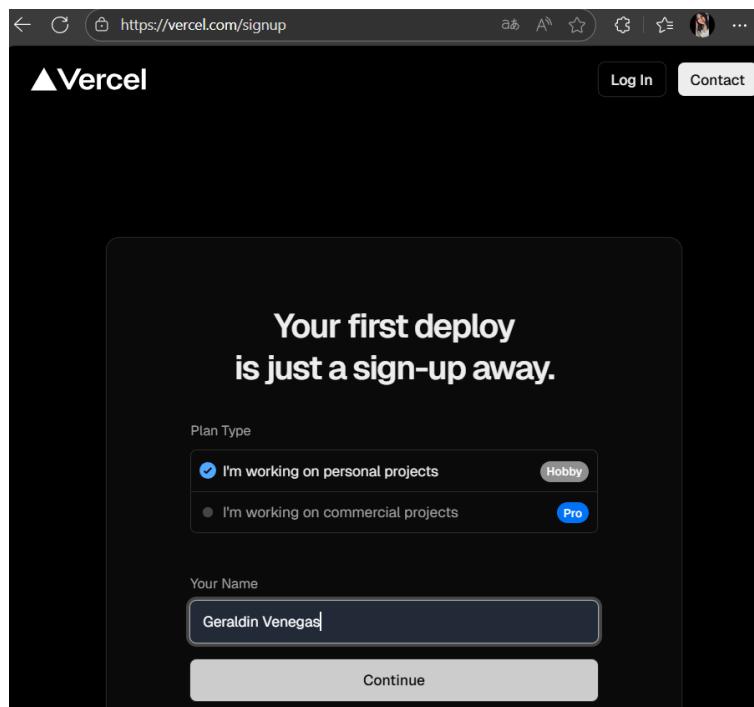


Ilustración 9. Inicio de sesión en Vercel con GitHub.

Fuente: Elaboración Propia.

2. En el dashboard, seleccionar **Import Project**.

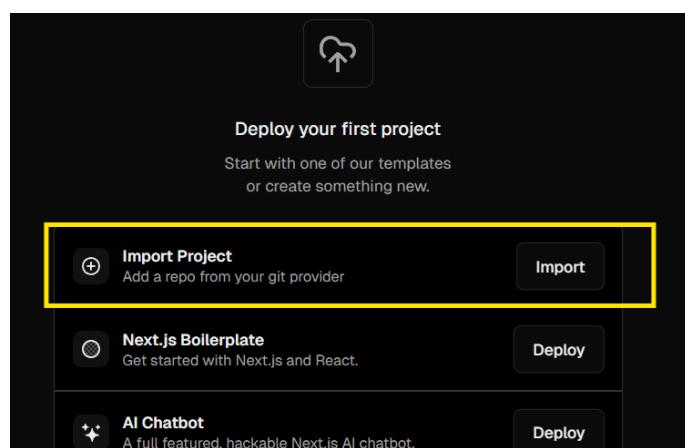


Ilustración 10. Importar proyecto de GitHub.

Fuente: Elaboración Propia.

3. Instalar GitHub y seleccionar el repositorio **cloudGVM**.

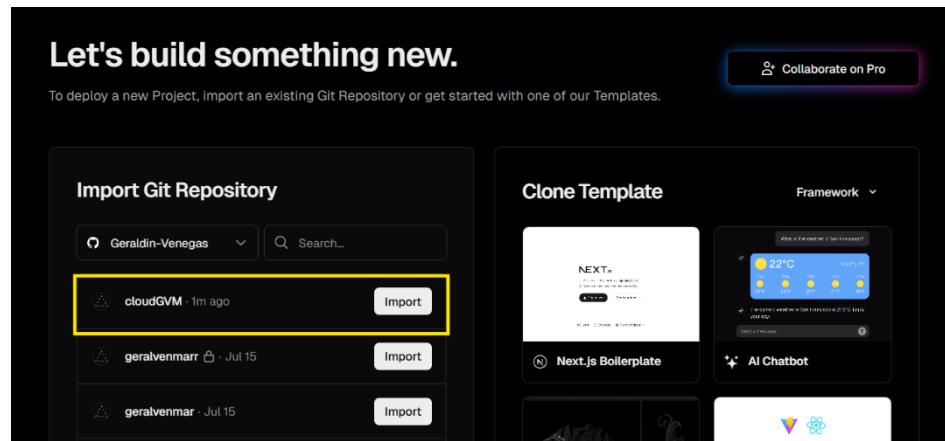


Ilustración 11. Seleccionar el repositorio cloudGVM.

Fuente: Elaboración Propia.

4. Dar clic en **Deploy** y luego en **Continue to Dashboard**.

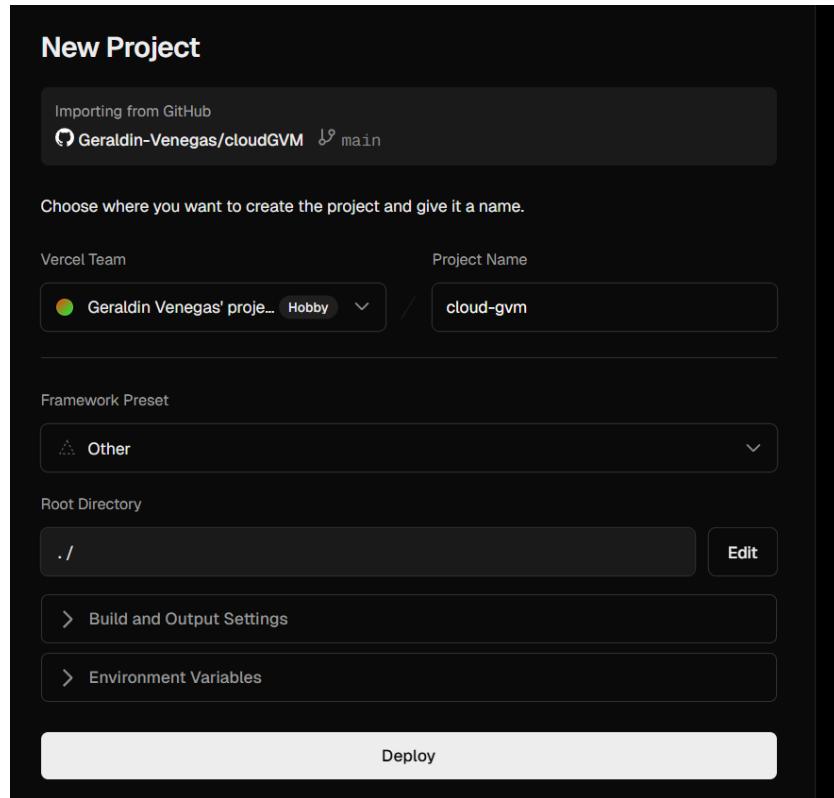
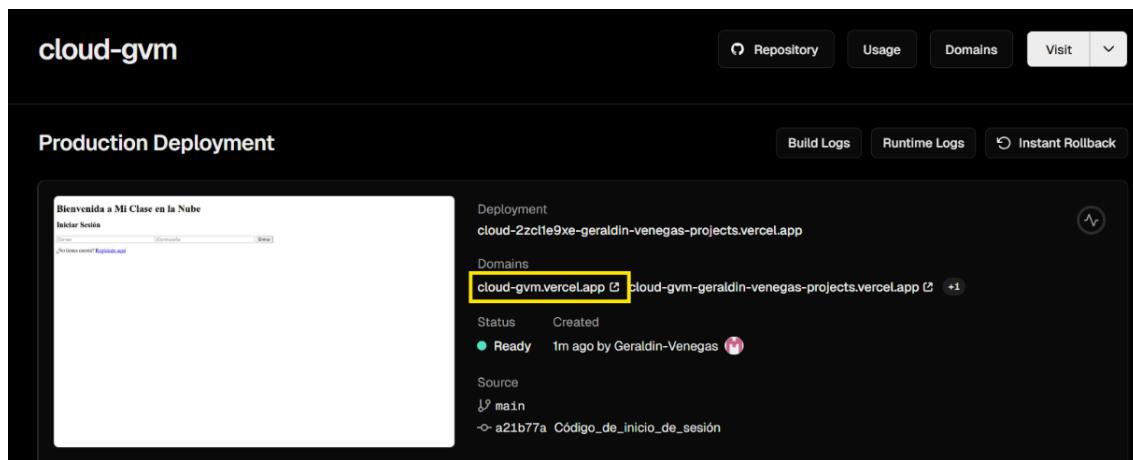


Ilustración 12. Deploy.

Fuente: Elaboración Propia.

5. Copiar y guardar el dominio asignado para acceder al login.

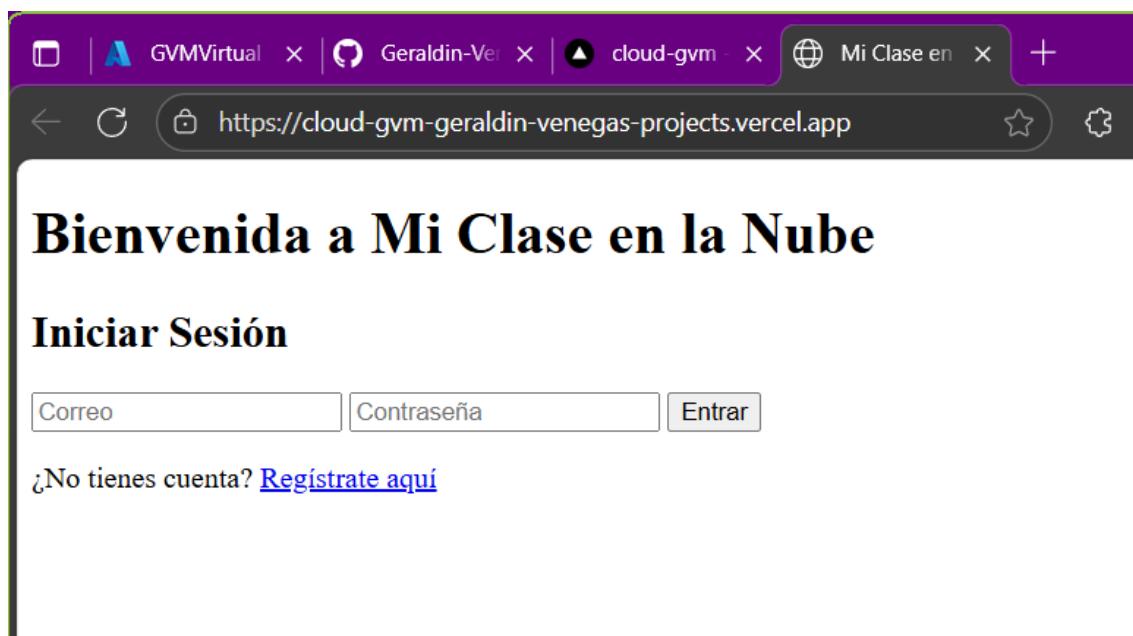


The screenshot shows the Vercel dashboard for a project named 'cloud-gvm'. The main section is titled 'Production Deployment'. It displays a preview of the application's landing page, which says 'Bienvenida a Mi Clase en la Nube' and 'Iniciar Sesión'. Below the preview, there is a table with two rows. The first row contains 'Status' (Ready), 'Created' (1m ago by Geraldin-Venegas), and a profile icon. The second row contains 'Source' (main branch) and a commit hash (a21b77a). At the top right of the dashboard, there are tabs for 'Repository', 'Usage', 'Domains', 'Visit', and a dropdown menu. Below the dashboard, there are buttons for 'Build Logs', 'Runtime Logs', and 'Instant Rollback'.

Ilustración 13. Despliegue del proyecto en Vercel con dominio público.

Fuente: Elaboración Propia.

6. Abrimos el link y ya podemos visualizar nuestro login.



The screenshot shows a web browser window with multiple tabs open. The active tab is for a project named 'cloud-gvm' on Vercel. The URL in the address bar is 'https://cloud-gvm-geraldin-venegas-projects.vercel.app'. The page content is a login form with the heading 'Bienvenida a Mi Clase en la Nube' and 'Iniciar Sesión'. It features two input fields: 'Correo' and 'Contraseña', and a 'Entrar' button. Below the form, there is a link '¿No tienes cuenta? [Regístrate aquí](#)'. The browser's toolbar includes icons for back, forward, and search.

Ilustración 14. Abrir el link de vercel.

Fuente: Elaboración Propia.

Autenticación

Configuración de Autenticación en Supabase

1. Ingresar a supabase.com e iniciar sesión con GitHub.



Ilustración 15. Ingresar a Supabase con GitHub.

Fuente: Elaboración Propia.

2. Crear una nueva organización:

- Nombre: *Geraldin-Venega's Org.*
- Tipo: Personal /
- Plan: Gratis
- Contraseña y región por defecto.

The screenshot shows the 'Crear un nuevo proyecto' (Create a new project) interface. It includes fields for 'Organización' (Organization) set to 'Org de Geraldin-Venegas' (Gratis), 'Nombre del proyecto' (Project name) set to 'Geraldin-Venegas's Project', 'Contraseña de la base de datos' (Database password) with a strength indicator, 'Región' (Region) set to 'Este de EE. UU. (Virginia del Norte)', and 'OPCIONES DE SEGURIDAD' (Security options) and 'CONFIGURACIÓN AVANZADA' (Advanced configuration) sections at the bottom. Buttons for 'Cancelar' (Cancel) and 'Crear nuevo proyecto' (Create new project) are at the bottom right.

Ilustración 16. Crear un nuevo proyecto / organización.

Fuente: Elaboración Propia.

3. En **Authentication > Iniciar sesión / Proveedores**, activar:

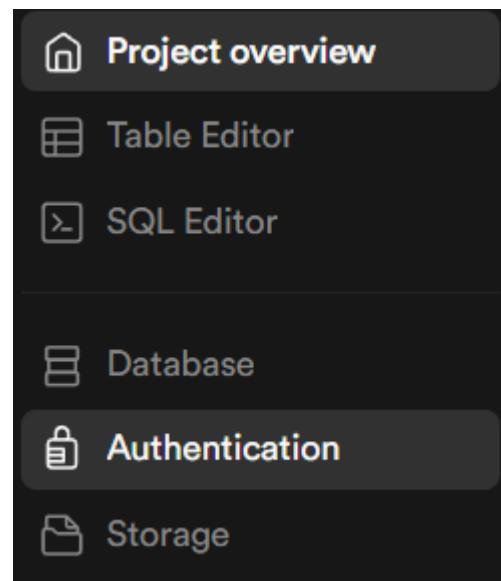


Ilustración 17. Apartado de Autenticación.

Fuente: Elaboración Propia.

- Permitir que nuevos usuarios se registren.
- Confirmar correo electrónico.

A screenshot of the Firebase Authentication configuration interface. The left sidebar shows "Autenticación" with "Iniciar sesión / Proveedores" selected. The main area is titled "Iniciar sesión / Proveedores" and includes a sub-section "Registros de usuarios". It lists four configuration options with toggle switches:

- Permitir que los nuevos usuarios se registren (Enabled)
- Permitir vinculación manual (Disabled)
- Permitir inicios de sesión anónimos (Disabled)
- Confirmar correo electrónico (Enabled)

Below each option is a brief description.

Ilustración 18. Configuración de proveedores de autenticación.

Fuente: Elaboración Propia.

4. En **Configuración > API de datos**, copiar la URL.

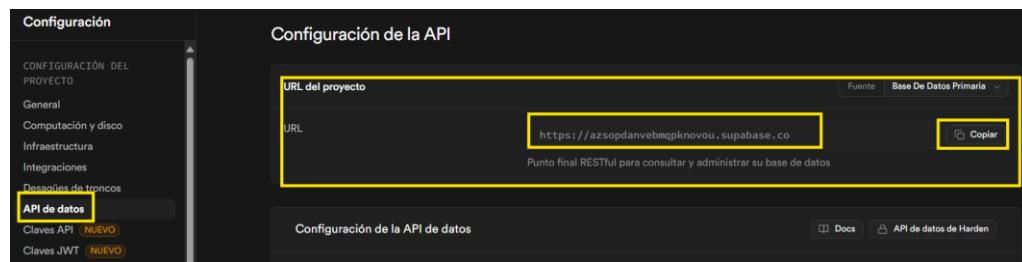


Ilustración 19. Copiar URL del proyecto.

Fuente: Elaboración Propia.

5. En **Claves API**, copiar la clave pública.

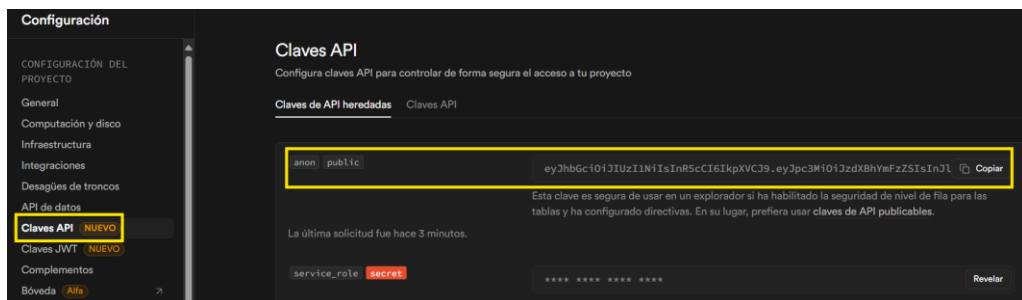


Ilustración 20. Copiar la clave pública.

Fuente: Elaboración Propia.

Integración del Código con Supabase

1. En el proyecto, crear la carpeta js con:

- **auth.js**
- **dashboard.js**

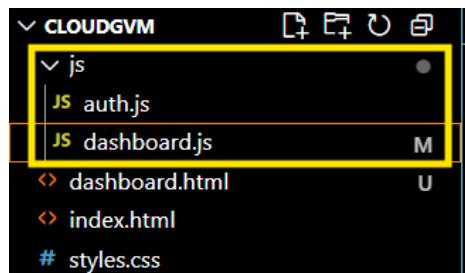
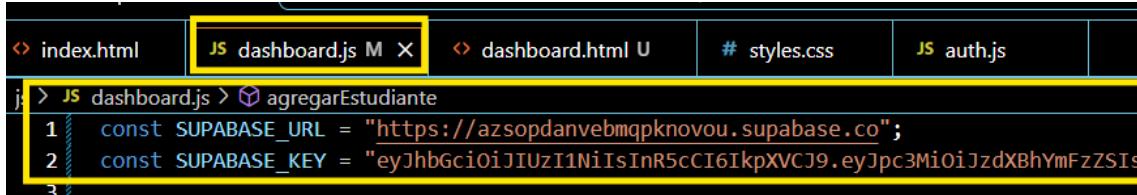


Ilustración 21. Crear carpeta js con los archivos auth y dashboard.

Fuente: Elaboración Propia.

2. Al inicio de ambos, agregar:

```
const SUPABASE_URL = "URL copiada";  
const SUPABASE_KEY = "Clave pública copiada";
```



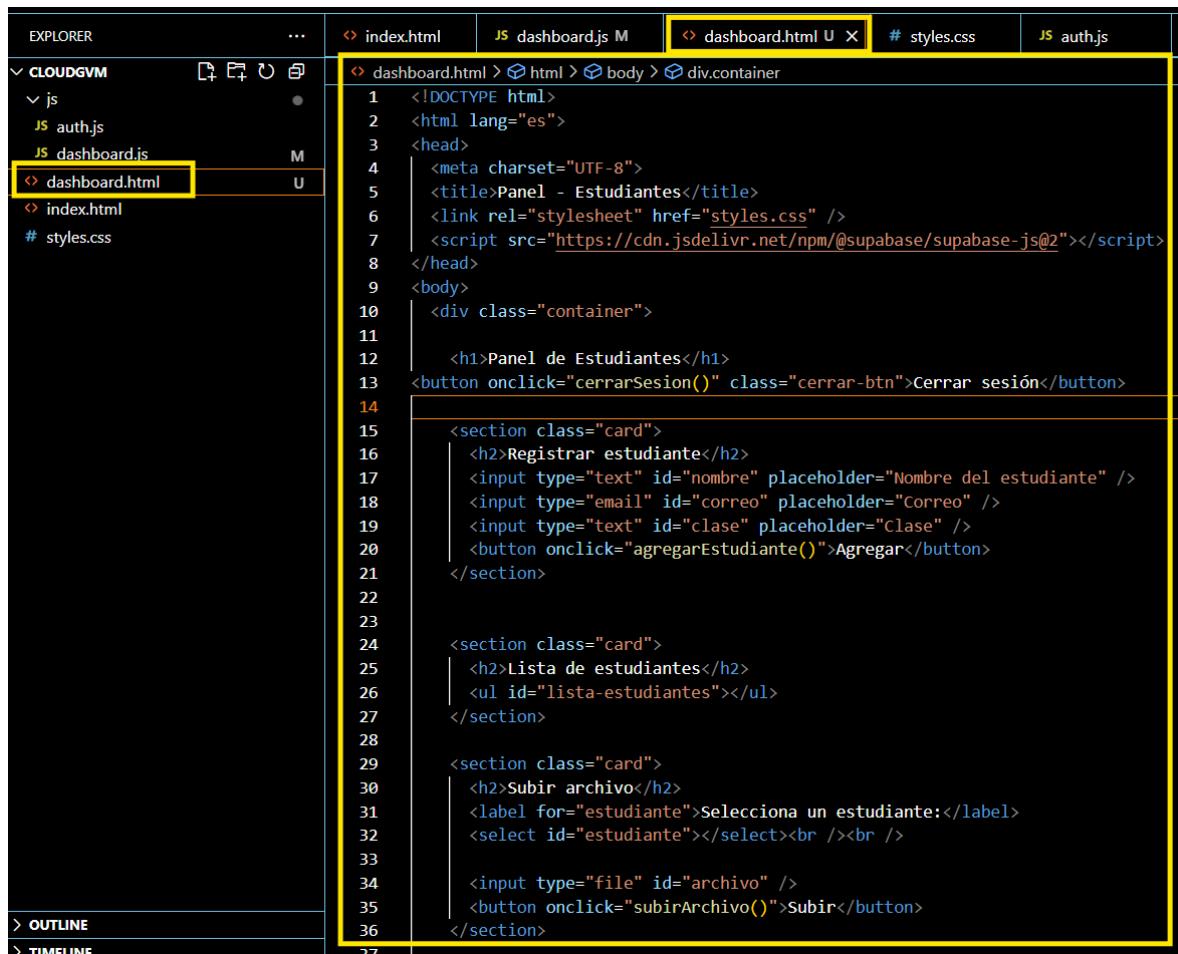
The screenshot shows a code editor with several tabs: index.html, JS dashboard.js (highlighted with a yellow box), dashboard.html, styles.css, and auth.js. The dashboard.js tab contains the following code:

```
1 const SUPABASE_URL = "https://azsopdanvebmqpknovou.supabase.co";  
2 const SUPABASE_KEY = "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIs  
3
```

Ilustración 22. Ingresar Url y clave en los archivos js.

Fuente: Elaboración Propia.

3. Crear el archivo **dashboard.html** con el panel de estudiantes.



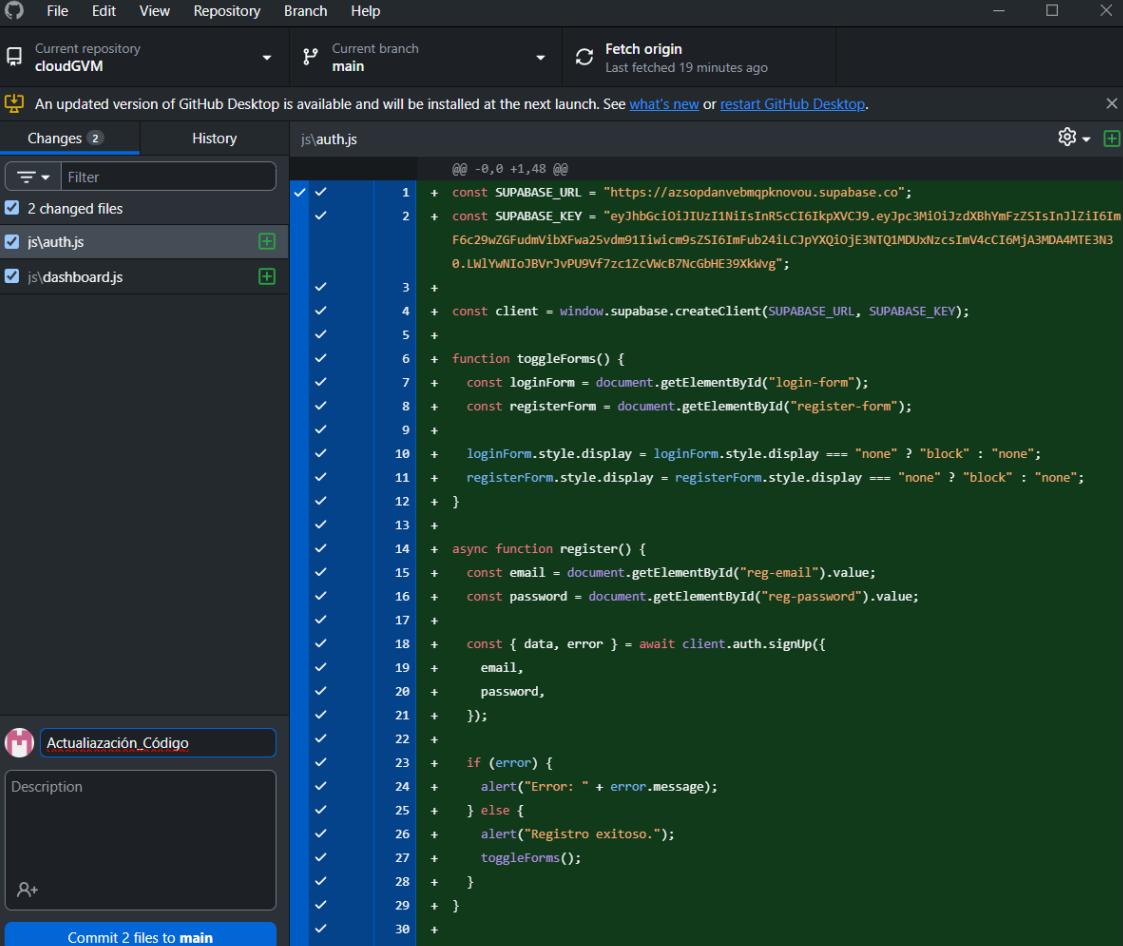
The screenshot shows a code editor with several tabs: index.html, JS dashboard.js, dashboard.html (highlighted with a yellow box), styles.css, and auth.js. The dashboard.html tab contains the following code:

```
1 <!DOCTYPE html>  
2 <html lang="es">  
3   <head>  
4     <meta charset="UTF-8">  
5     <title>Panel - Estudiantes</title>  
6     <link rel="stylesheet" href="styles.css" />  
7     <script src="https://cdn.jsdelivr.net/npm/@supabase/supabase-js@2"></script>  
8   </head>  
9   <body>  
10    <div class="container">  
11      <h1>Panel de Estudiantes</h1>  
12      <button onclick="cerrarSesion()" class="cerrar-btn">Cerrar sesión</button>  
13  
14      <section class="card">  
15        <h2>Registrar estudiante</h2>  
16        <input type="text" id="nombre" placeholder="Nombre del estudiante" />  
17        <input type="email" id="correo" placeholder="Correo" />  
18        <input type="text" id="clase" placeholder="Clase" />  
19        <button onclick="agregarEstudiante()">Agregar</button>  
20      </section>  
21  
22  
23      <section class="card">  
24        <h2>Lista de estudiantes</h2>  
25        <ul id="lista-estudiantes"></ul>  
26      </section>  
27  
28  
29      <section class="card">  
30        <h2>Subir archivo</h2>  
31        <label for="estudiante">Selecciona un estudiante:</label>  
32        <select id="estudiante"></select><br /><br />  
33  
34        <input type="file" id="archivo" />  
35        <button onclick="subirArchivo()">Subir</button>  
36      </section>  
37
```

Ilustración 23. Dashboard del panel del estudiante.

Fuente: Elaboración Propia.

4. Subir los cambios a GitHub y verificar la actualización en la página de Vercel.



The screenshot shows the GitHub Desktop application interface. At the top, it displays the current repository as "cloudGVM" and the branch as "main". A message at the top indicates an update is available. Below this, the "Changes" tab is selected, showing two changed files: "js\auth.js" and "js\dashboard.js". The "js\auth.js" file is open, displaying a diff of 31 lines. The commit message is "Actualización Código".

```
@@ -0,0 +1,31 @@
1 + const SUPABASE_URL = "https://azsopdanvbmqpknovou.supabase.co";
2 + const SUPABASE_KEY = "eyJhbGciOiJIUzI1NiisInR5cCI6IkpxVCJ9eyJpc3MiOiJzdXBhYmFzZSIsInJlZii6Im
3 + F6c29wZGFudmVibFua25vdm911iwcm9SZi61mfub241LCJpYXQ1OjE3NTQ1MDUxNzcsImV4cCI6MjA3MDA4MTE3N
4 + 0.LW1YMWIoJBVrJvPU9Vf7zc12cVcB7NcGbHE39XKwvg";
5 +
6 + function toggleForms() {
7 +   const loginForm = document.getElementById("login-form");
8 +   const registerForm = document.getElementById("register-form");
9 +
10 +   loginForm.style.display = loginForm.style.display === "none" ? "block" : "none";
11 +   registerForm.style.display = registerForm.style.display === "none" ? "block" : "none";
12 + }
13 +
14 + async function register() {
15 +   const email = document.getElementById("reg-email").value;
16 +   const password = document.getElementById("reg-password").value;
17 +
18 +   const { data, error } = await client.auth.signUp({
19 +     email,
20 +     password,
21 +   });
22 +
23 +   if (error) {
24 +     alert("Error: " + error.message);
25 +   } else {
26 +     alert("Registro exitoso.");
27 +     toggleForms();
28 +   }
29 + }
30 +
31 + async function login() {
```

Ilustración 24. Subir commit con las actualizaciones.

Fuente: Elaboración Propia.

5. Verificamos que se actualizo en GitHub



Ilustración 25. Actualización subida al repositorio.

Fuente: Elaboración Propia.

6. Probamos nuestro login y verificamos que funcione la autenticación.

- Abrimos nuestro link del sistema.



Ilustración 26. Abrir dominio.

Fuente: Elaboración Propia.

- Registramos un usuario.



Ilustración 27. Registrar un nuevo usuario.

Fuente: Elaboración Propia.

- Intentamos ingresar con el nuevo usuario.
- Nos manda un mensaje en el cual es necesario confirmar el correo, por lo tanto, la autenticación del correo funciona correctamente.

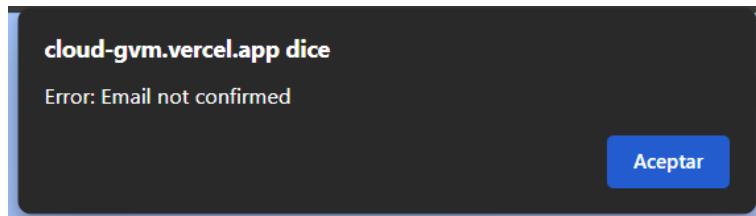


Ilustración 28. Mensaje de correo no confirmado.

Fuente: Elaboración Propia.

- Una vez confirmado en el correo nos aparece el mensaje de “sesión iniciada”.

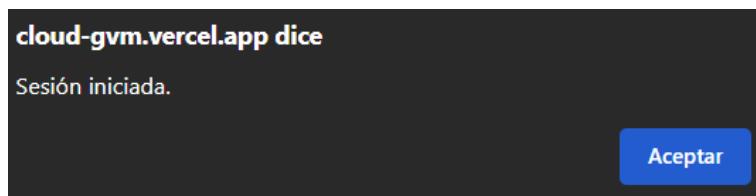


Ilustración 29. Mensaje de sesión iniciada.

Fuente: Elaboración Propia.

- Verificamos que se haya almacenado el usuario correctamente.

Authentication		Users					
MANAGE							
Users							
CONFIGURATION							
Policies		UID	Display name	Email	Phone	Providers	
		1e44be49-02ed-48e5-9149-2e3b37304c18	-	margaritogeraldin@gmail.com	-	Email	

Ilustración 30. Usuario almacenado correctamente.

Fuente: Elaboración Propia.

Base de datos

Creación de la Base de Datos

1. En Supabase, ir a **Database > Schema Visualizer > Create new table.**

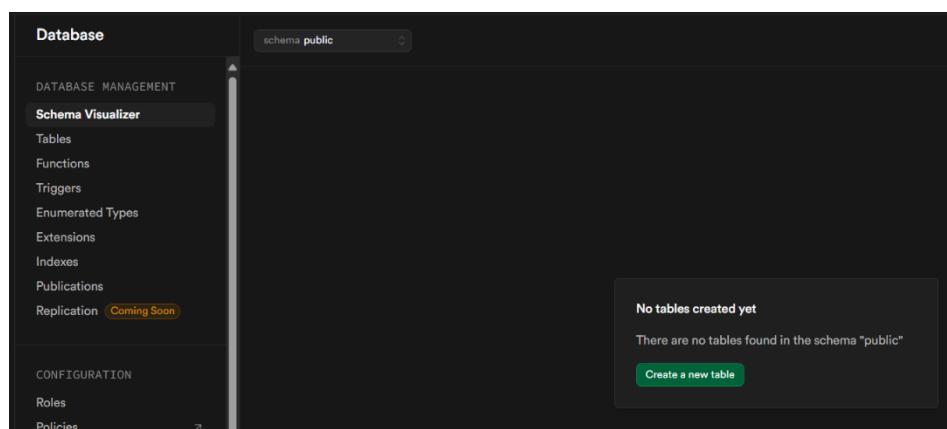


Ilustración 31. Apartado de Database.

Fuente: Elaboración Propia.

2. Nombre: **estudiantes**.

A screenshot of the 'Create a new table' form. The title is 'Cree una nueva tabla en public'. There are two input fields: 'Nombre' with 'estudiantes' and 'Descripción' with 'Opcional'. At the bottom, there's a checked checkbox for 'Habilitar la seguridad de nivel de fila (RLS)' with the note 'Recomendado' and a descriptive text: 'Restrinja el acceso a su tabla habilitando RLS y escribiendo políticas de Postgres.'.

Ilustración 32. Nombre de la tabla: estudiantes.

Fuente: Elaboración Propia.

3. Columnas:

- o id → uuid
- o created_at → timestamp
- o nombre → text

- correo → text
- user_id → uuid
- clase → text

Columns

Name	Type	Default Value	Primary
id	uuid	gen_random_uuid()	<input checked="" type="checkbox"/>
created_at	timestamp	now()	<input type="checkbox"/>
nombre	text	NULL	<input checked="" type="checkbox"/>
correo	text	NULL	<input checked="" type="checkbox"/>
user_id	uuid	gen_random_uuid()	<input checked="" type="checkbox"/>
clase	text	NULL	<input checked="" type="checkbox"/>

Add column

Foreign keys

Add foreign key relation

Cancel Save

Ilustración 33. Columnas de la tabla de estudiantes.

Fuente: Elaboración Propia.

4. Una vez guardado la tabla, en editor de tablas nos debe aparecer.

Editor de tablas

esquema público

+ Nueva tabla

Buscar tablas...

estudiantes

Ora identificación uuid

Filtro Ordenar Insertar

Ilustración 34. Tabla de estudiantes en Editor de tablas.

Fuente: Elaboración Propia.

5. En Editor de tablas > estudiantes > **View polices**.

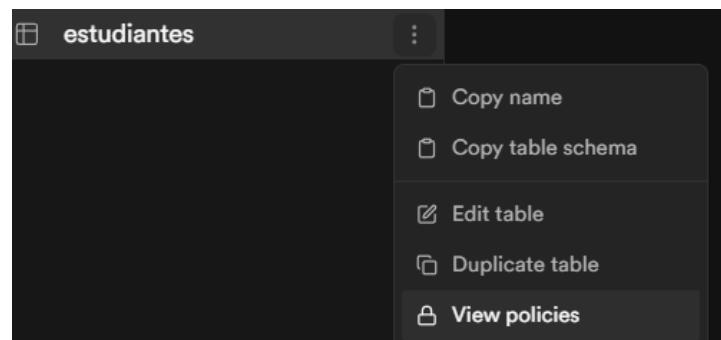


Ilustración 35. View polices.

Fuente: Elaboración Propia.

6. Crear políticas:

- o **SELECT** y **INSERT** para roles authenticated con condición: **user_id = auth.uid()**.

The screenshot shows the 'Create a new Row Level Security policy' dialog. The fields are as follows:

- Policy Name: Seleccionar
- Table: public.estudiantes
- Policy Behavior: Permissive
- Policy Command: SELECT (radio button selected)
- Target Roles: authenticated

Below the dialog, the generated SQL code is displayed:

```
1 create policy "policy_name"
2 on "public"."estudiantes"
3 as PERMISSIVE
4 for SELECT
5 to public
6 using (
7 user_id = auth.uid ( )
8 );
```

Ilustración 36. Política de SELECT.

Fuente: Elaboración Propia.

Create a new Row Level Security policy

Policy Name: Insertar

Table on clause: public.estudiantes

Policy Behavior as clause: Permissive

Policy Command for clause: INSERT SELECT UPDATE DELETE ALL

Target Roles to clause: authenticated

```

    USE OPTIONS ABOVE TO EDIT
1  create policy "Insertar"
2  on "public"."estudiantes"
3  as PERMISSIVE
4  for INSERT
5  to authenticated
6  with check (
7    user_id = auth.uid ( )
8  );

```

Ilustración 37. Política de INSERT.

Fuente: Elaboración Propia.

- Visualización de las políticas creadas.

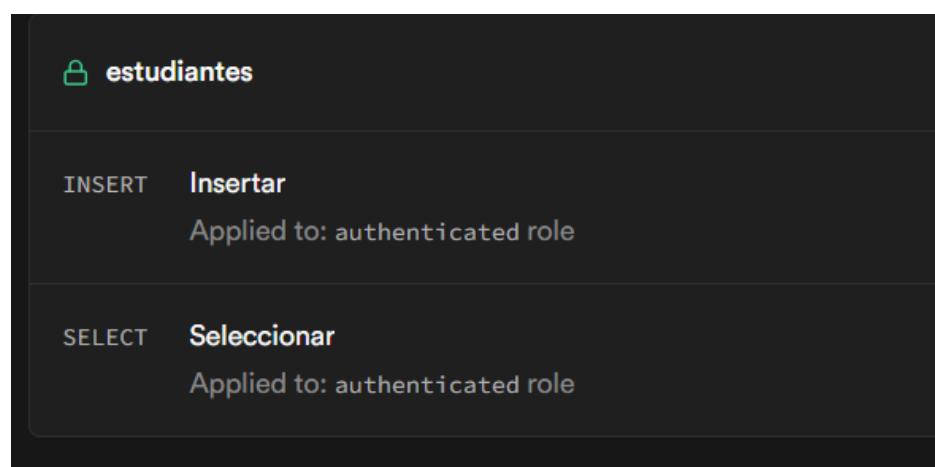


Ilustración 38. Políticas de acceso creadas para el registro de estudiantes.

Fuente: Elaboración Propia.

Registro de Estudiantes

1. En el panel (dashboard.html), registrar un estudiante.



Ilustración 39. Mensaje de estudiante registrado correctamente.

Fuente: Elaboración Propia.

Al dar clic en “Aceptar” se visualiza en la lista de estudiantes.



Ilustración 40. Visualización del estudiante en la lista.

Fuente: Elaboración Propia.

2. Verificar que se muestre en la lista y en la base de datos de Supabase.

estudiantes				
	+	Insert		
Filter	Sort	Insert		
id	uuid	created_at	timestampz	nombre
bf5096db-1024-4077-bf95-38ab0d116774	2025-09-06 20:41:18.556533+0	Geraldin_GVM		correo
				venegasgeraldin4@gmail.com

Ilustración 41. Visualización en la base de datos.

Fuente: Elaboración Propia.

Subir archivos

Configuración de Almacenamiento de Archivos

1. En Supabase, ir a **Storage > New bucket** llamado tareas.

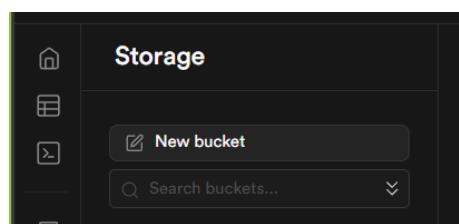


Ilustración 42. Apartado de Storage.

Fuente: Elaboración Propia.

A screenshot of the 'Create storage bucket' dialog box. It has a title bar 'Create storage bucket' and a close button. The 'Name of bucket' field is filled with 'tareas'. A note says 'Buckets cannot be renamed once created.' Below it, a note says 'Only lowercase letters, numbers, dots, and hyphens'. There are three options: 'Standard bucket' (selected), 'Analytics bucket' (disabled with a note about being in alpha), and 'Public bucket' (disabled with a note about anyone reading without authorization). An 'Additional restrictions' section is collapsed. At the bottom are 'Cancel' and 'Create' buttons.

Ilustración 43. Nuevo bucket llamado: tareas.

Fuente: Elaboración Propia.

2. Crear política:

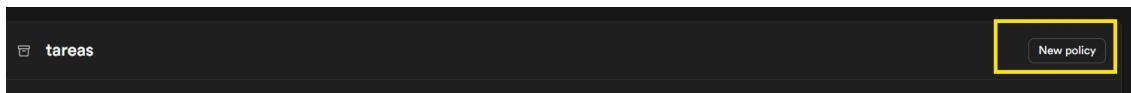


Ilustración 44. Crear nueva política para tareas.

Fuente: Elaboración Propia.

- Nombre: Permitir acceso.
- Operación: SELECT.
- Rol: authenticated.
- Definición: **bucket_id = 'tareas' and auth.uid() = owner**

A screenshot of the Firebase Rules Editor. It shows a configuration for the 'tareas' bucket. Under 'Allowed operation', 'SELECT' is checked and other options like 'INSERT', 'UPDATE', and 'DELETE' are unchecked. Below this, a list of functions is shown: upload, download, list, update, move, copy, remove, createSignedUrl, createSignedUrls, and getPublicUrl. Under 'Target roles', 'authenticated' is selected. In the 'Policy definition' section, a SQL conditional expression is defined: 1 bucket_id = 'tareas' and auth.uid () = owner. At the bottom right, there are 'View templates' and 'Review' buttons.

Ilustración 45. Política de tareas: Permitir acceso.

Fuente: Elaboración Propia.

- Nombre: Permitir insertar.
- Operación: INSERT.
- Rol: authenticated.
- Definición: **bucket_id = 'tareas' and auth.uid() = owner**

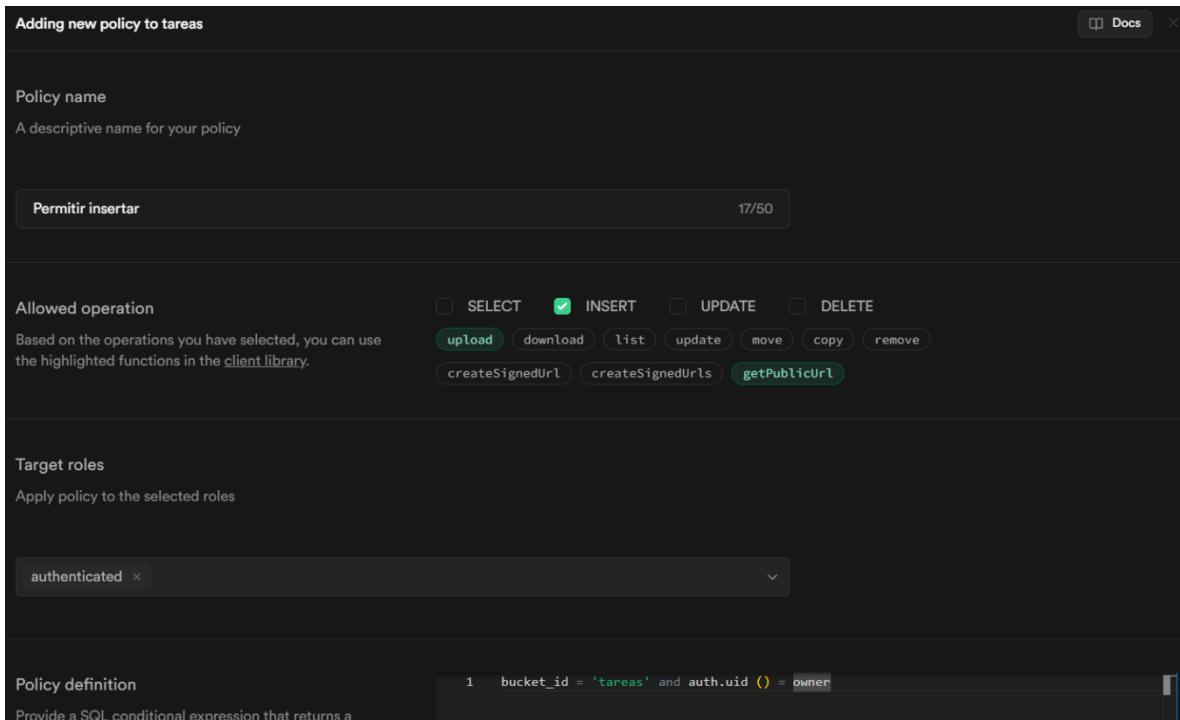


Ilustración 46. Política de tareas: Permitir insertar.

Fuente: Elaboración Propia.

3. Subir un archivo desde el panel y verificar que aparezca en la lista con opción de Elegir archivo, al dar clic nos abre el explorador de archivos donde podemos elegir el archivo a subir.

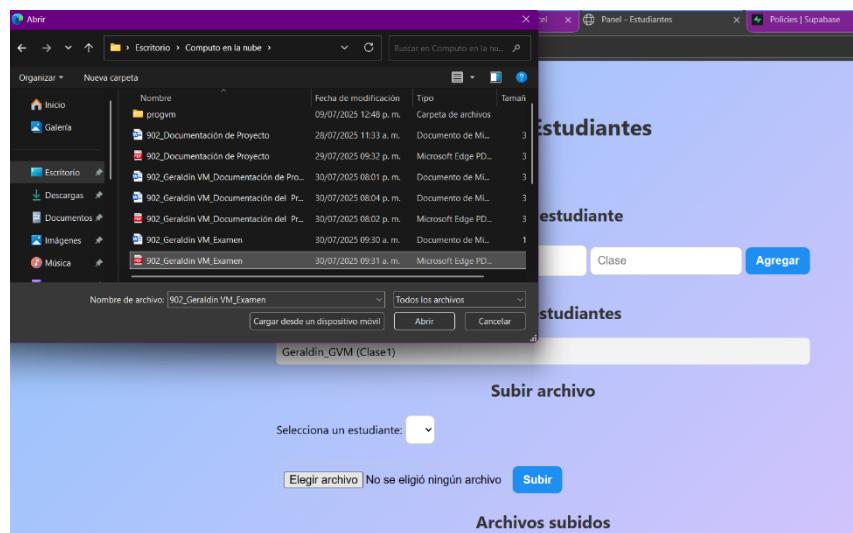


Ilustración 47. Elegir archivo a subir.

Fuente: Elaboración Propia.

4. Ya que se eligió el archivo dar clic en el botón de “Subir” y nos parece el mensaje de que el archivo se subió correctamente.

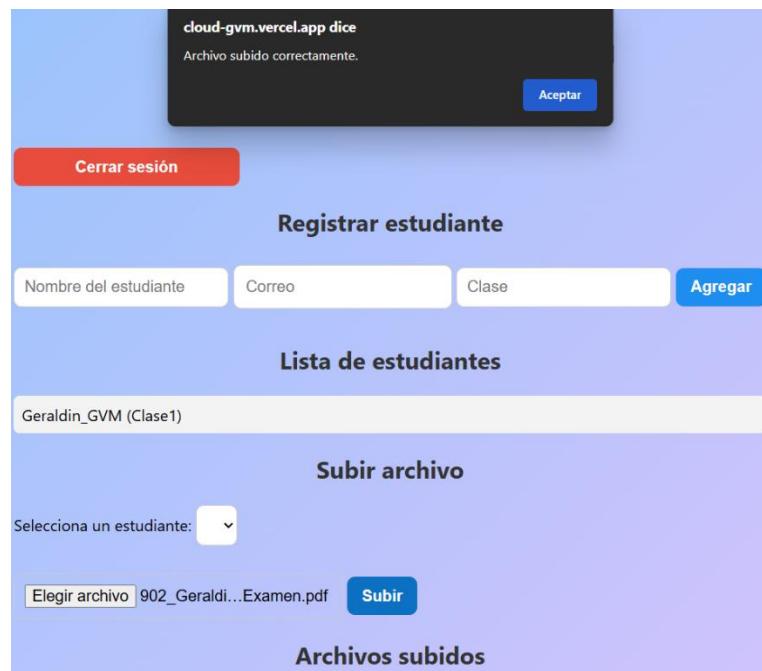


Ilustración 48. Archivo subido correctamente.

Fuente: Elaboración Propia.

5. Al dar clic en “Aceptar” se enlista el archivo subido, se puede subir diferentes tipos de archivos ya sea PDF, Exel, Imagen, Video, etc.

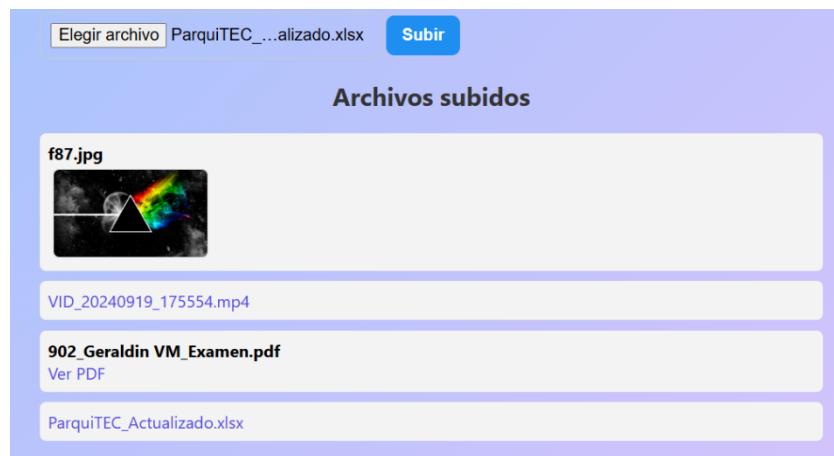


Ilustración 49. Visualización de archivos subidos.

Fuente: Elaboración Propia.

6. Se registra un nuevo usuario y accedemos, este se encuentra completamente vacío, es decir no nos muestra lo realizado en la otra cuenta, ya que son cuentas independientes. Registramos un estudiante y subimos un archivo.

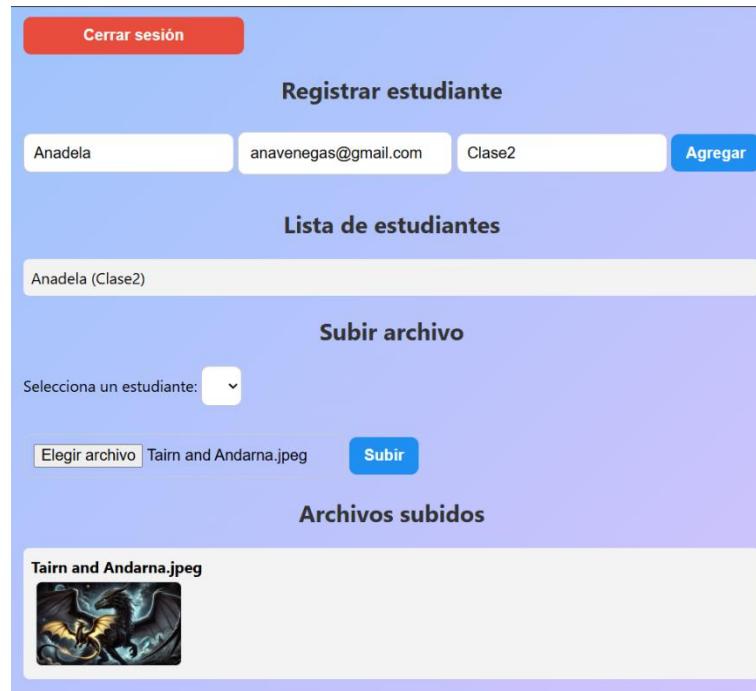


Ilustración 50. Subida y visualización de archivos en el panel.

Fuente: Elaboración Propia.

7. En Supabase se puede visualizar dos almacenamientos que corresponde a cada uno de los usuarios.

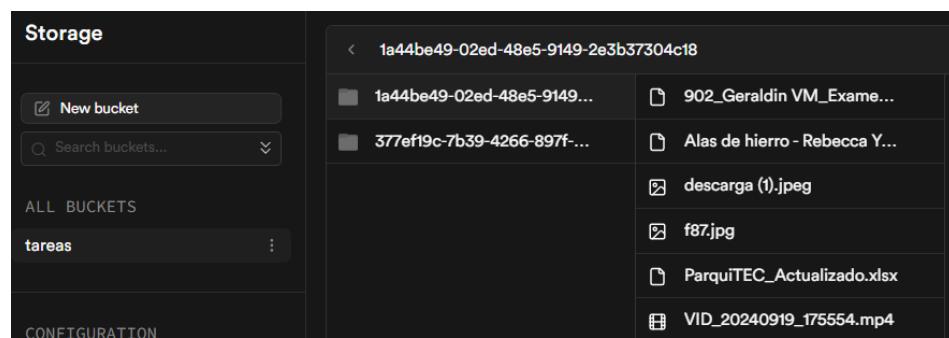


Ilustración 51. Visualización de almacenamiento en Supabase.

Fuente: Elaboración Propia.

Funcionalidad de Edición y Eliminación de Registros

Para ampliar las capacidades del sistema, se implementó la opción de **editar** y **eliminar** registros de estudiantes desde la interfaz web, garantizando que cada usuario pueda modificar únicamente sus propios datos.

1. Configuración de Políticas en Supabase

- En la tabla **estudiantes**, se creó la política:
 - **Nombre:** actualizar
 - **Policy Command:** UPDATE
 - **Condición SQL:** auth.uid() = user_id

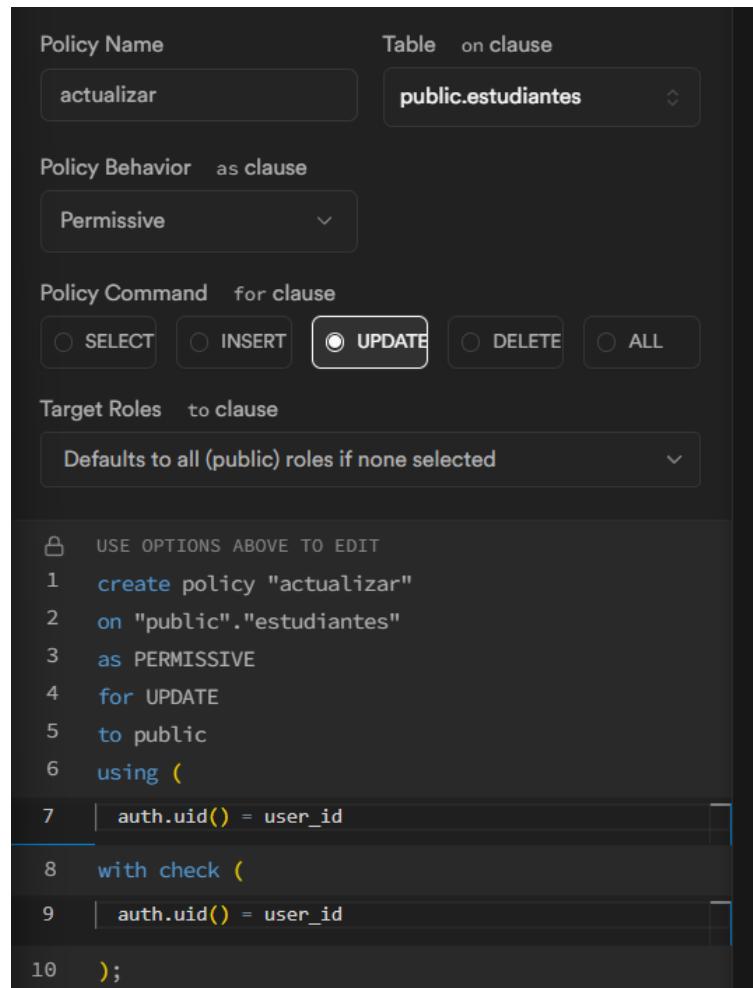


Ilustración 52. Creación de política *actualizar* en la tabla *estudiantes*.

Fuente: Elaboración Propia.

- Se creó la política:
 - **Nombre:** eliminar
 - **Policy Command:** DELETE
 - **Condición SQL:** auth.uid() = user_id

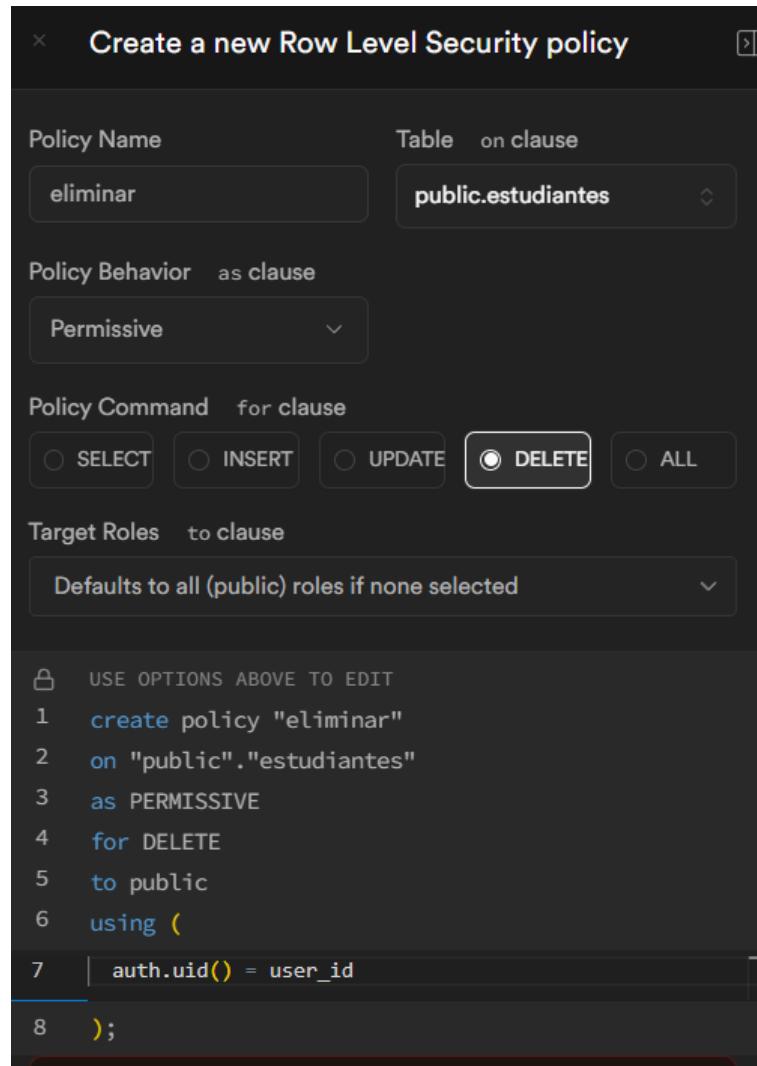


Ilustración 53. Creación de política actualizar en la tabla estudiantes.

Fuente: Elaboración Propia.

2. Modificación del Código

- Se actualizaron los archivos JavaScript para mostrar en la tabla dos nuevos botones por cada registro: **Editar** y **Eliminar**.
- Se subió un commit con los cambios realizados.

The screenshot shows a GitHub commit interface. The repository is named 'cloudGVM'. The current branch is 'main'. The commit message is 'actualización'. The commit has 3 changes:

- dashboard.html**: Changes include adding a CSS link for 'font-awesome' and removing a button with the ID 'cerrarSesion'.
- js/dashboard.js**: Changes include adding a dropdown menu with options for 'Clase 1' through 'Clase 4'.
- styles.css**: Changes include adding a table with the ID 'tabla-estudiantes'.

Ilustración 54. Generar commit con los cambios para editar y eliminar.

Fuente: Elaboración Propia.

- Verificamos que se subieron en el repositorio.

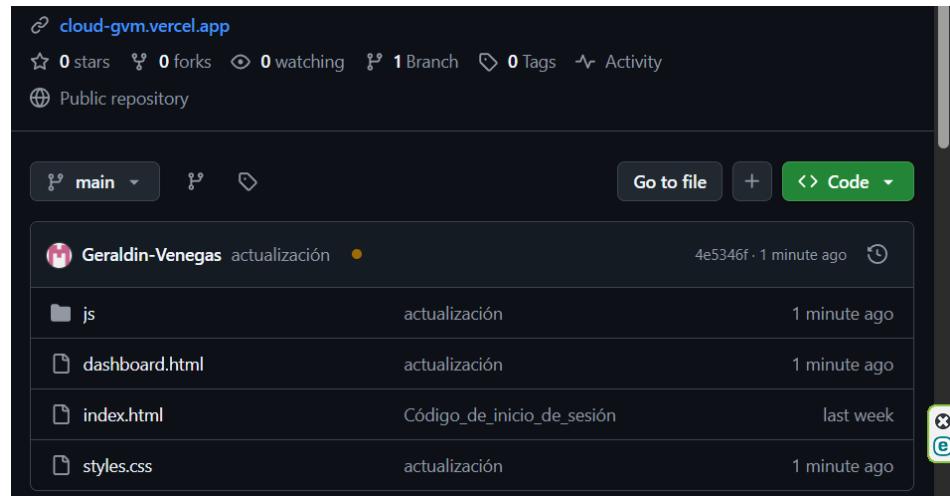


Ilustración 55. Cambios subidos al repositorio. Fuente: Elaboración Propia.

- Una vez que se subieron los cambios volvemos a cargar la página para que se actualice y probamos que funcionen correctamente.
- Al ingresar podemos visualizar los registros que ya tiene el usuario, vamos a editar el registro de Yazmin, y cambiaremos los datos por uno nuevo.

The screenshot shows a user interface for managing students. At the top, a header reads "Panel de Estudiantes". Below it, a sub-header says "Registrar estudiante". There are three input fields: "Nombre del estudiante" (Name of student), "Correo" (Email), and "Seleccione una clase" (Select a class). A blue "Agregar" (Add) button is located below the input fields. In the center, a section titled "Lista de estudiantes" (List of students) displays a table with four rows of data. The columns are "Nombre" (Name), "Correo" (Email), "Clase" (Class), "Editar" (Edit), and "Eliminar" (Delete). The data is as follows:

Nombre	Correo	Clase	Editar	Eliminar
Yazmin	yazvenegas@gmail.com	Clase 4	<button>Editar</button>	<button>Eliminar</button>
Jenni_JVM	margaritojenni@gmail.com	Clase 3	<button>Editar</button>	<button>Eliminar</button>
Ana_AVM	margaritoana@gmail.com	Clase 2	<button>Editar</button>	<button>Eliminar</button>
Geraldin_GVM	margaritogeral@gmail.com	Clase 1	<button>Editar</button>	<button>Eliminar</button>

Ilustración 56. Visualización de estudiantes registrados previamente.

Fuente: Elaboración Propia.

- Al dar clic en “Editar” se llena nuevamente el formulario con los datos cargados.

The screenshot shows the "Registrar estudiante" (Register Student) form. It includes fields for "Nombre" (Name) containing "Yazmin", "Correo" (Email) containing "yazvenegas@gmail.com", and "Clase" (Class) containing "Clase 4". A green "Guardar cambios" (Save changes) button is at the bottom left. An orange "Cerrar sesión" (Close session) button is at the top right.

Ilustración 57. Relleno de formulario automáticamente para editar.

Fuente: Elaboración Propia.

- Editamos la información correo: yazmin07@gmail.com, clase 1 y guardamos cambios.

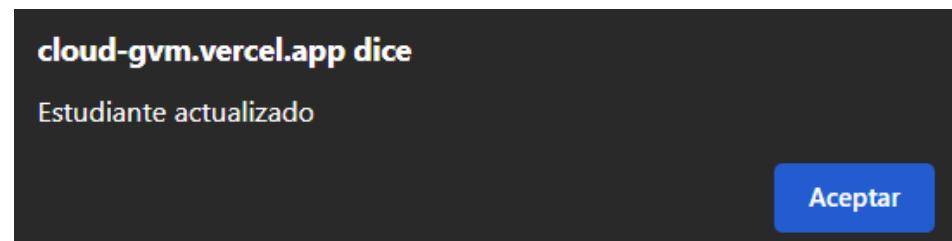


Ilustración 58, Mensaje de estudiante actualizado.

Fuente: Elaboración Propia.

- Al dar clic en aceptar se visualizan los cambios en la tabla.

The image shows a user interface for managing students. At the top, a header reads "Panel de Estudiantes". Below it, a section titled "Registrar estudiante" contains input fields for "Nombre del estudiante", "Correo", and "Seleccione una clase" (with a dropdown arrow). A blue "Agregar" button is located below these fields. To the right, an orange "Cerrar sesión" button is visible. Below this section is a table titled "Lista de estudiantes" with the following data:

Nombre	Correo	Clase	Editar	Eliminar
Yazmin	yazmin07@gmail.com	Clase 1	Editar	Eliminar
Jenni_JVM	margaritojenni@gmail.com	Clase 3	Editar	Eliminar
Ana_AVM	margaritoana@gmail.com	Clase 2	Editar	Eliminar

Ilustración 59. Estudiante actualizado correctamente.

Fuente: Elaboración Propia.

- Ahora vamos a eliminar el resto del estudiante: Jenn_JVM, Al presionar eliminar nos mostrara un mensaje donde debemos confirmar que queremos eliminar el registro.

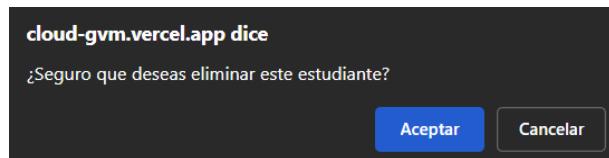


Ilustración 60. Mensaje de confirmación para eliminar.

Fuente: Elaboración Propia.

- Al dar clic en aceptar, nos mostrar el mensaje de estudiante eliminado.



Ilustración 61. Mensaje de estudiante eliminado.

Fuente: Elaboración Propia.

- Visualizamos que en la tabla ya no se encuentra el registro de Jenni_JVM.

Lista de estudiantes				
Nombre	Correo	Clase	Editar	Eliminar
Yazmin	yazmin07@gmail.com	Clase 1	Editar	Eliminar
Ana_AVM	margaritoana@gmail.com	Clase 2	Editar	Eliminar
Geraldin_GVM	margaritogeral@gmail.com	Clase 1	Editar	Eliminar

Ilustración 62. Estudiante eliminado correctamente.

Fuente: Elaboración Propia.

3. Mejoras Visuales

- Durante el proceso se mejoró el diseño de la página, añadiendo nuevos estilos CSS para una mejor presentación y experiencia de usuario.

Análisis de la práctica

Durante la elaboración de estas prácticas se configuró un entorno completo para el desarrollo y despliegue de un sistema web funcional. La elección de **Vercel Hosting** permitió que el proceso de publicación fuera rápido y que las actualizaciones del código se reflejaran casi de inmediato en el sitio web, optimizando el flujo de trabajo y la experiencia de desarrollo.

La implementación de **autenticación de usuarios** fue un punto clave, ya que se garantizó que cada persona pudiera acceder solo a su cuenta, protegiendo la información y personalizando la interacción con la aplicación. Posteriormente, la integración de una **base de datos en la nube** permitió almacenar información estructurada de los estudiantes, asegurando su persistencia y disponibilidad en cualquier momento.

Se habilitó la **subida de archivos**, vinculando cada elemento a un usuario específico, lo que abrió la puerta a funcionalidades de gestión personalizada. Además, se añadieron botones de **edición y eliminación** de registros, junto con políticas de seguridad que restringieron dichas acciones únicamente al propietario de los datos. Esto mejoró notablemente la usabilidad del sistema y reforzó la seguridad a nivel de backend.

En conjunto, la práctica permitió desarrollar un sistema web completo que cubre aspectos esenciales como la autenticación, el manejo de datos y archivos, y la administración de contenido dinámico. El resultado fue una aplicación funcional y segura, desplegada en línea y lista para ser utilizada en entornos reales, fortaleciendo así las competencias del estudiante en el desarrollo de software moderno.