

# Desarrollo de una Aplicación de Gestión de Contenido Digital

Aplicación inspirada en Notion para facilitar la organización personal y profesional de los usuarios.



# Introducción

1

## Descripción del Proyecto

Aplicación para gestionar contenido digital de manera eficiente.

2

## Justificación

Notion se destaca por su versatilidad, utilidad, popularidad y simplicidad.

3

## Objetivo General

Crear una herramienta intuitiva y funcional.

# Objetivos Específicos

## Creación de Páginas

Permitir a usuarios crear y personalizar páginas.

## Bases de Datos Efectivas

Integrar funcionalidad para crear bases de datos.

## Gestión de Permisos

Proveer a administradores control de acceso a espacios.

# Historias de Usuario

## Usuario General

Registro, inicio de sesión, creación y edición de páginas.

## Administrador

Gestión de permisos y estadísticas de uso.



# Tarjetas CRC



Usuario

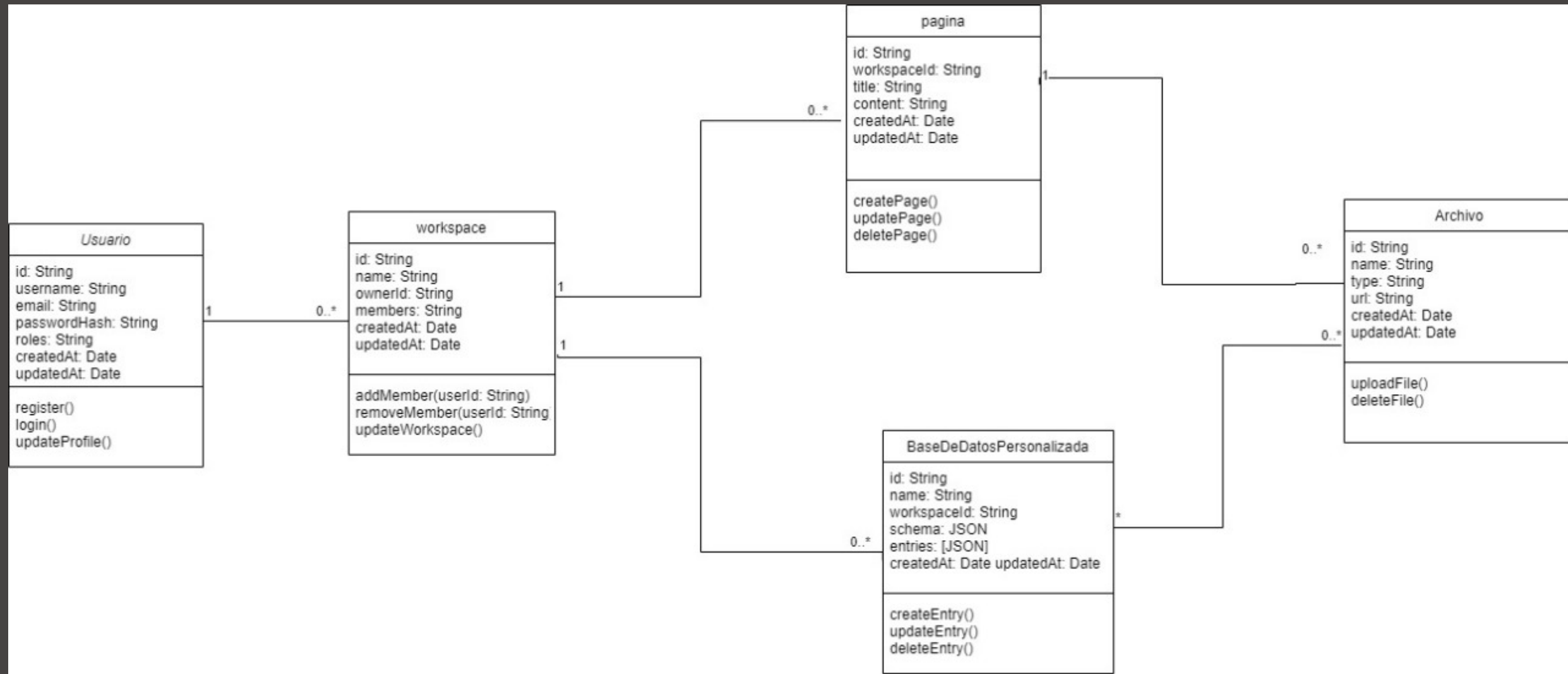
Class: User	
<b>Responsibilities:</b>  Register a new user (register()).  Log in (login()).  Update the user's profile (updateProfile()).	<b>Collaborators:</b>  Collaborates with the Workspace class, as a user can have multiple workspaces.



Espacio de Trabajo

Class: Workspace	
<b>Responsibilities:</b>  Create, modify, or delete a workspace (updateWorkspace()).  Add members to the workspace (addMember()).  Remove members from the workspace (removeMember()).	<b>Collaborators:</b>  Collaborates with the User class, as a user can own or be a member of several workspaces.  Collaborates with the Page class, as a workspace can contain multiple pages.  Collaborates with the CustomDatabase class, as a workspace can have custom databases.  Collaborates with the File class to manage files within the workspace.

# Diagrama de Clases



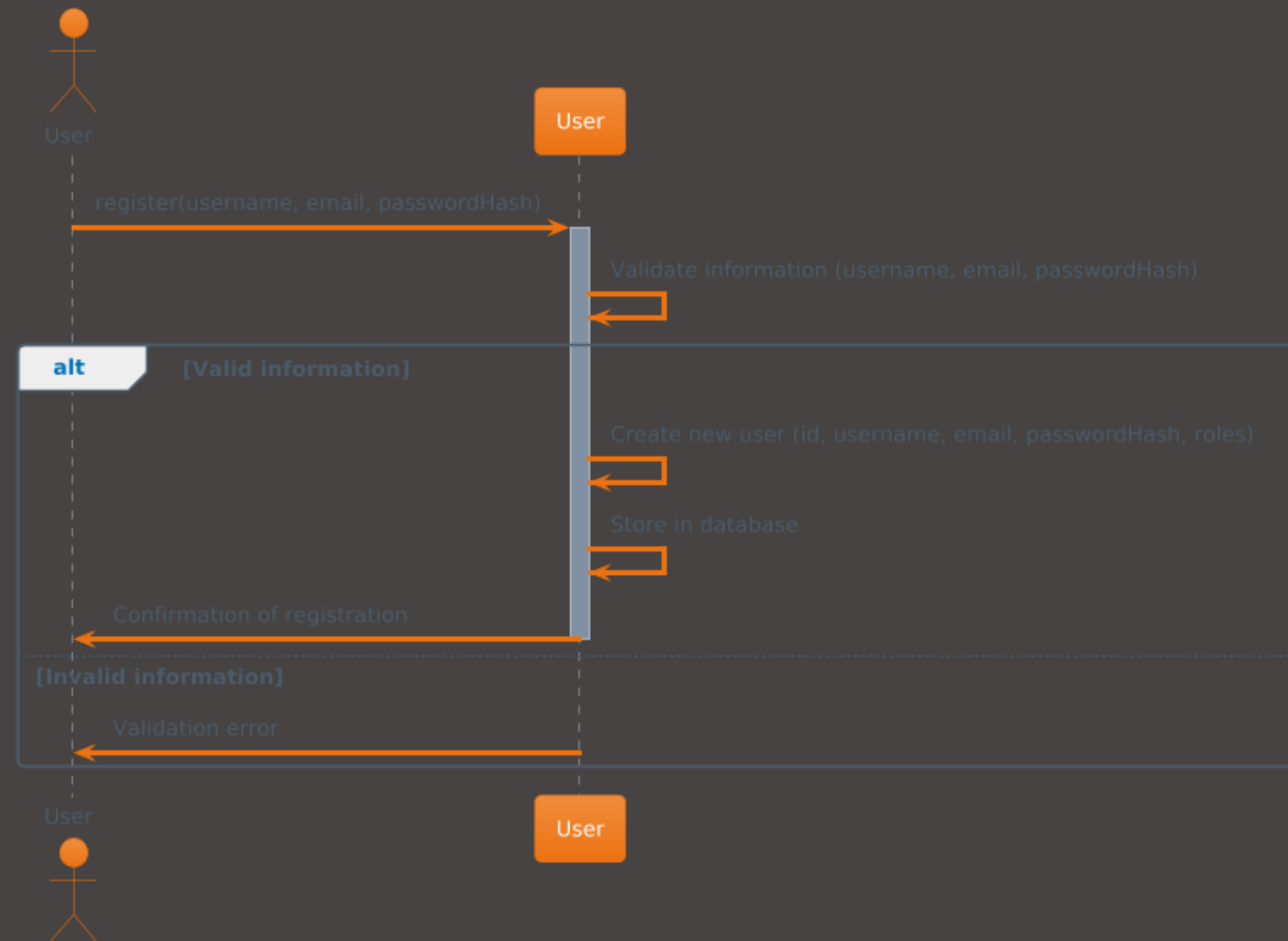
Representación de clases, atributos y relaciones.



# Diagramas uml

## Diagramas de secuencia

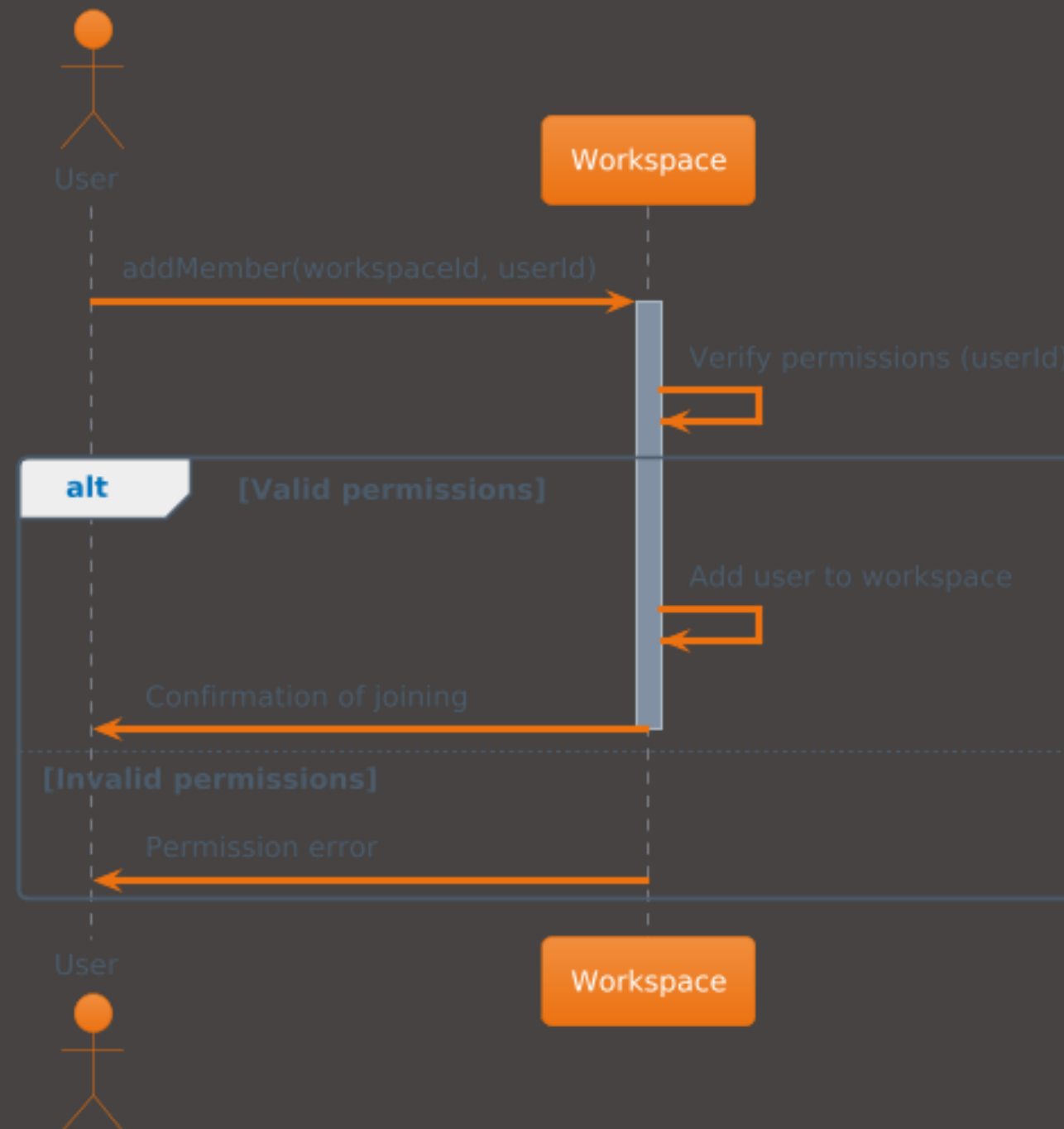
### Registrar usuario



# Diagramas uml

## Diagramas de secuencia

### Ingresar al espacio de trabajo

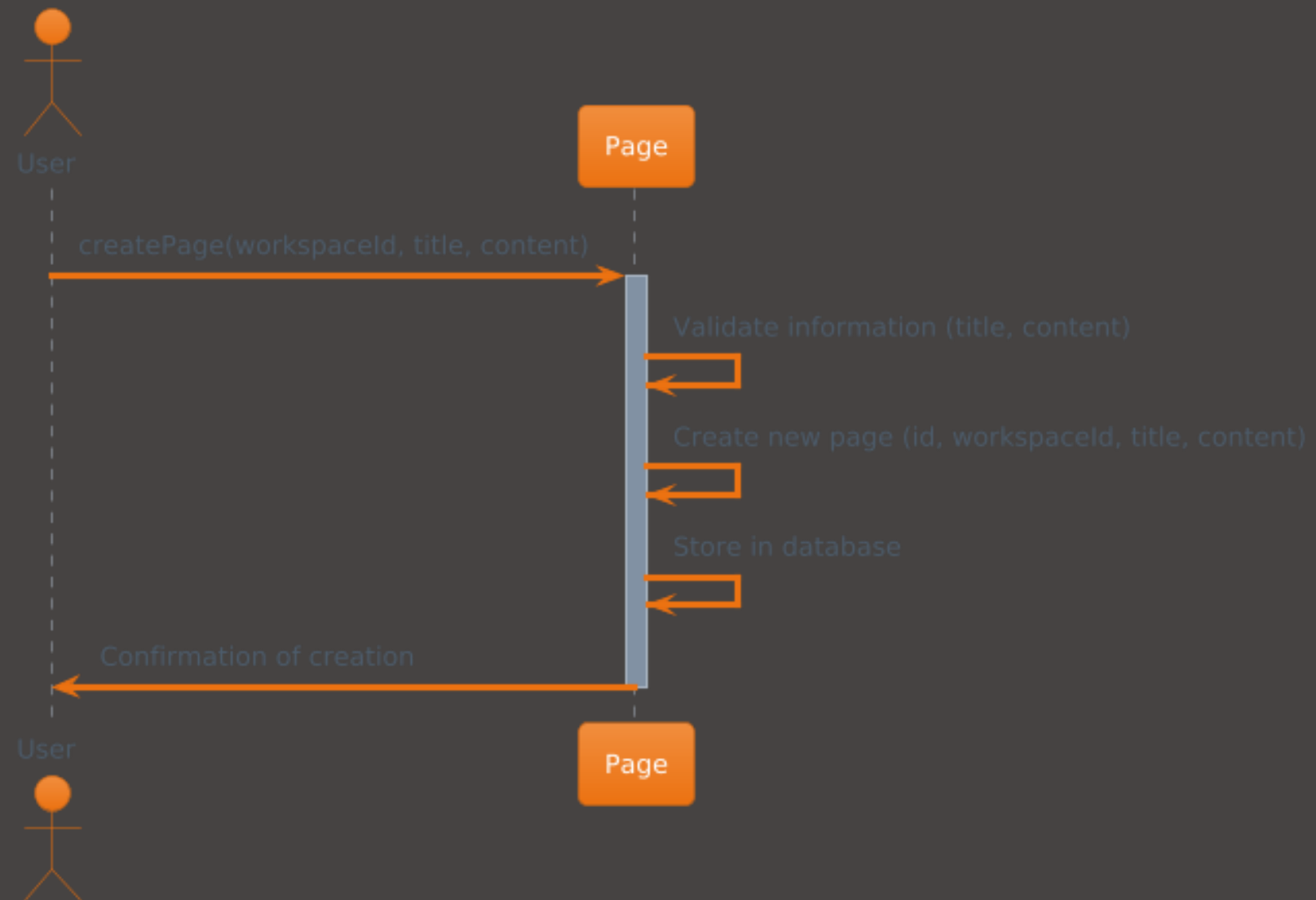




# Diagramas uml

## Diagramas de secuencia

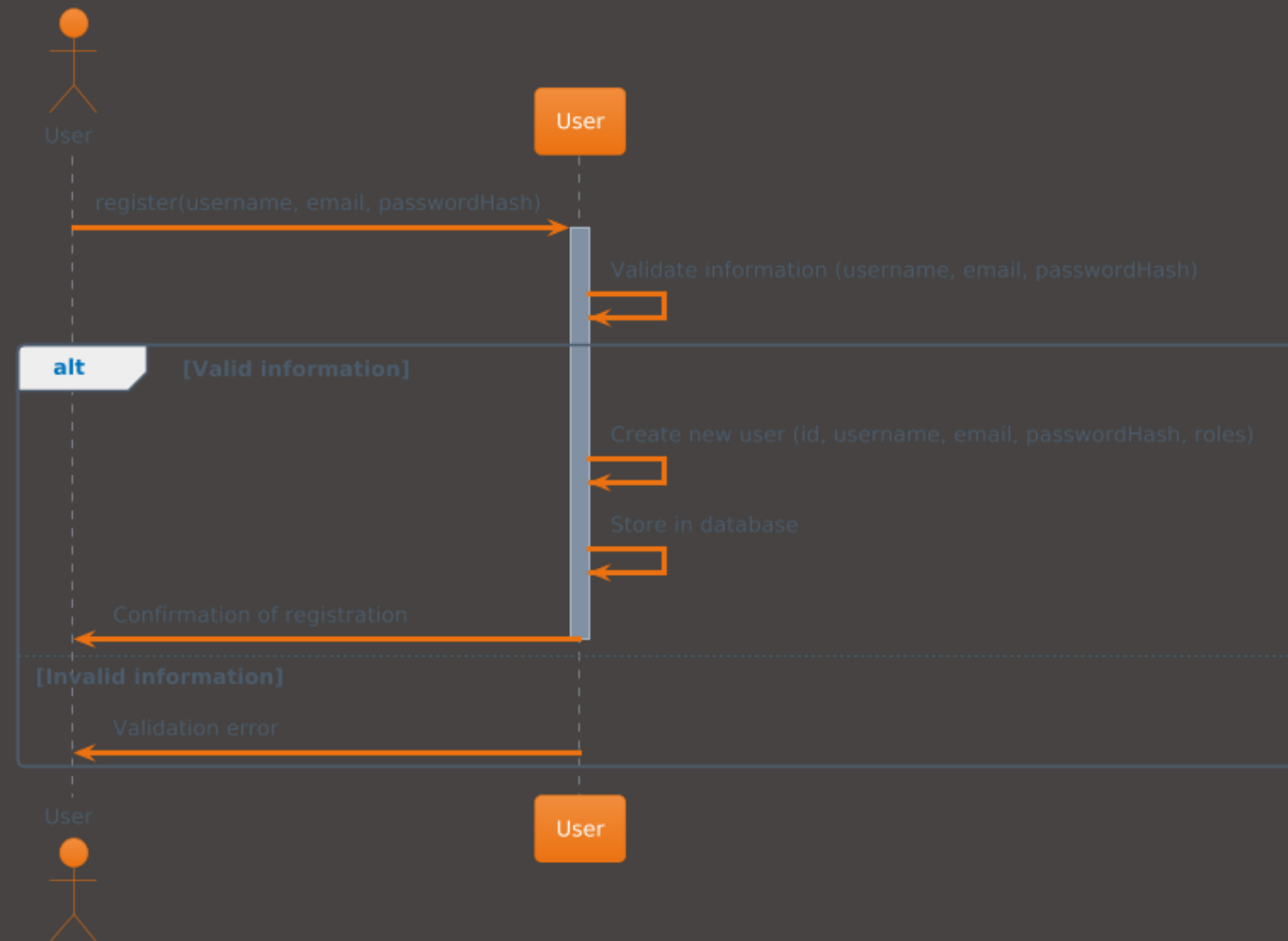
### Crear pagina



# Diagramas uml

## Diagramas de secuencia

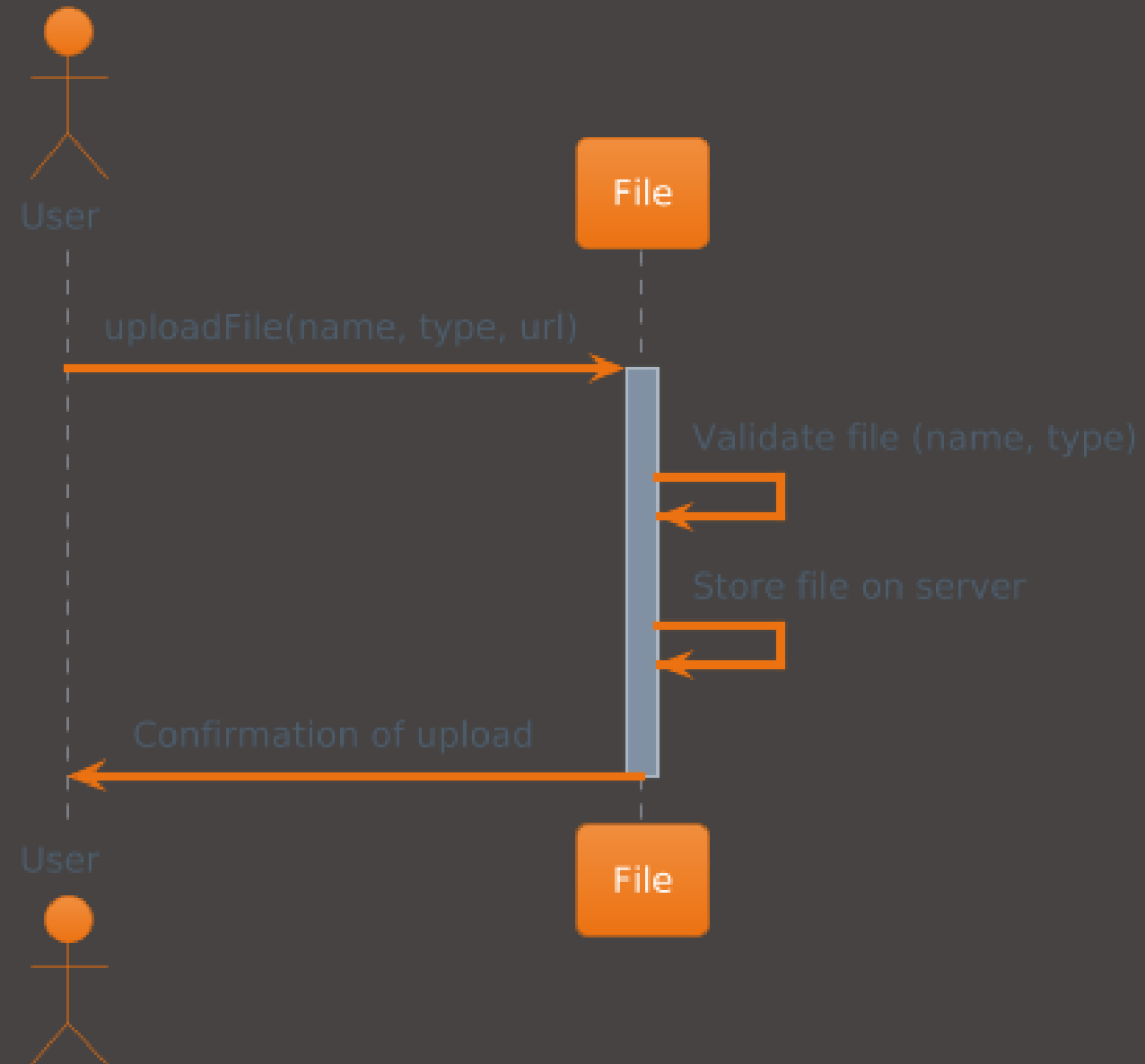
### Registrar usuario



# Diagramas uml

## Diagramas de secuencia

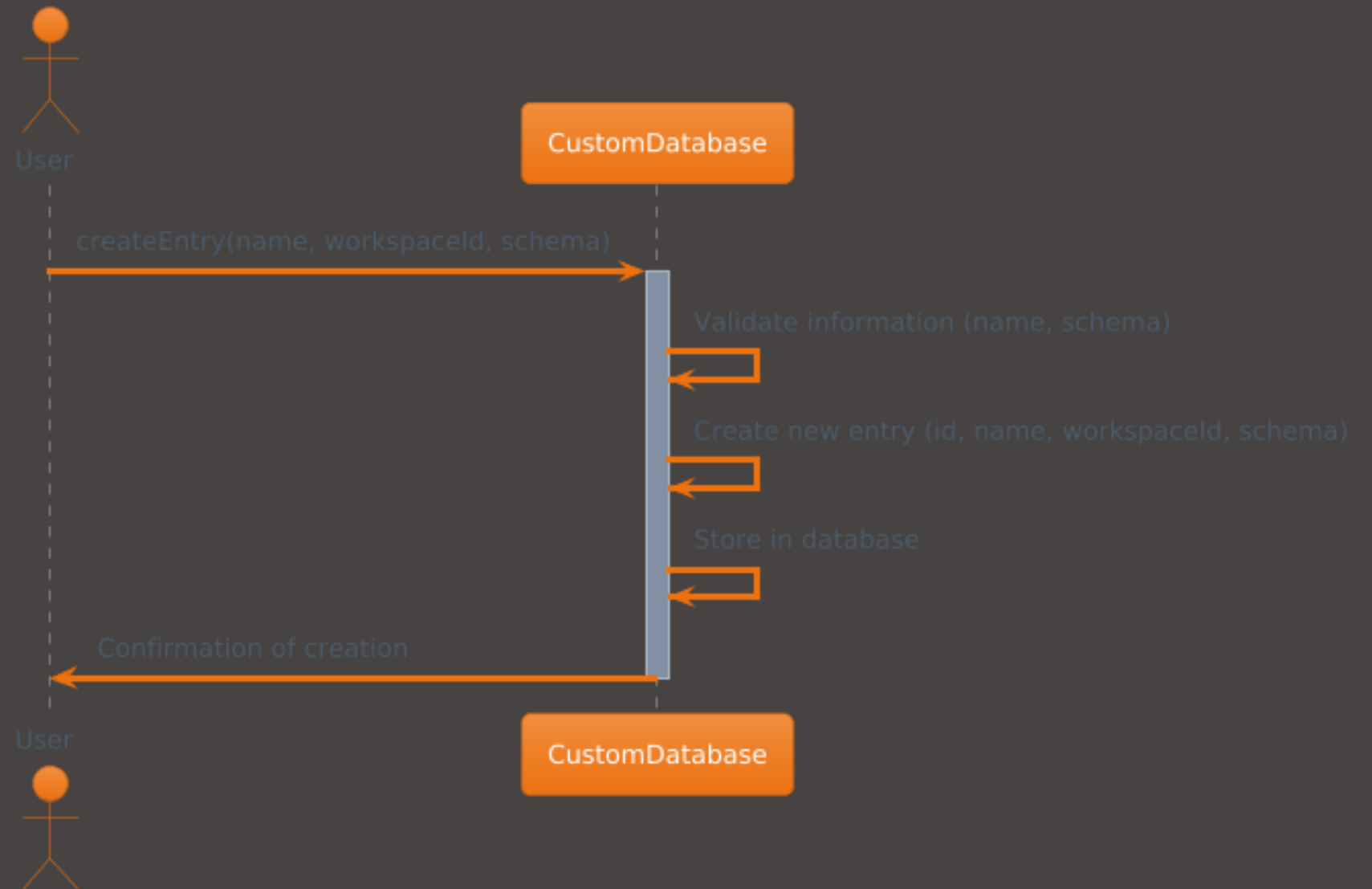
### Subir archivo



# Diagramas uml

## Diagramas de secuencia

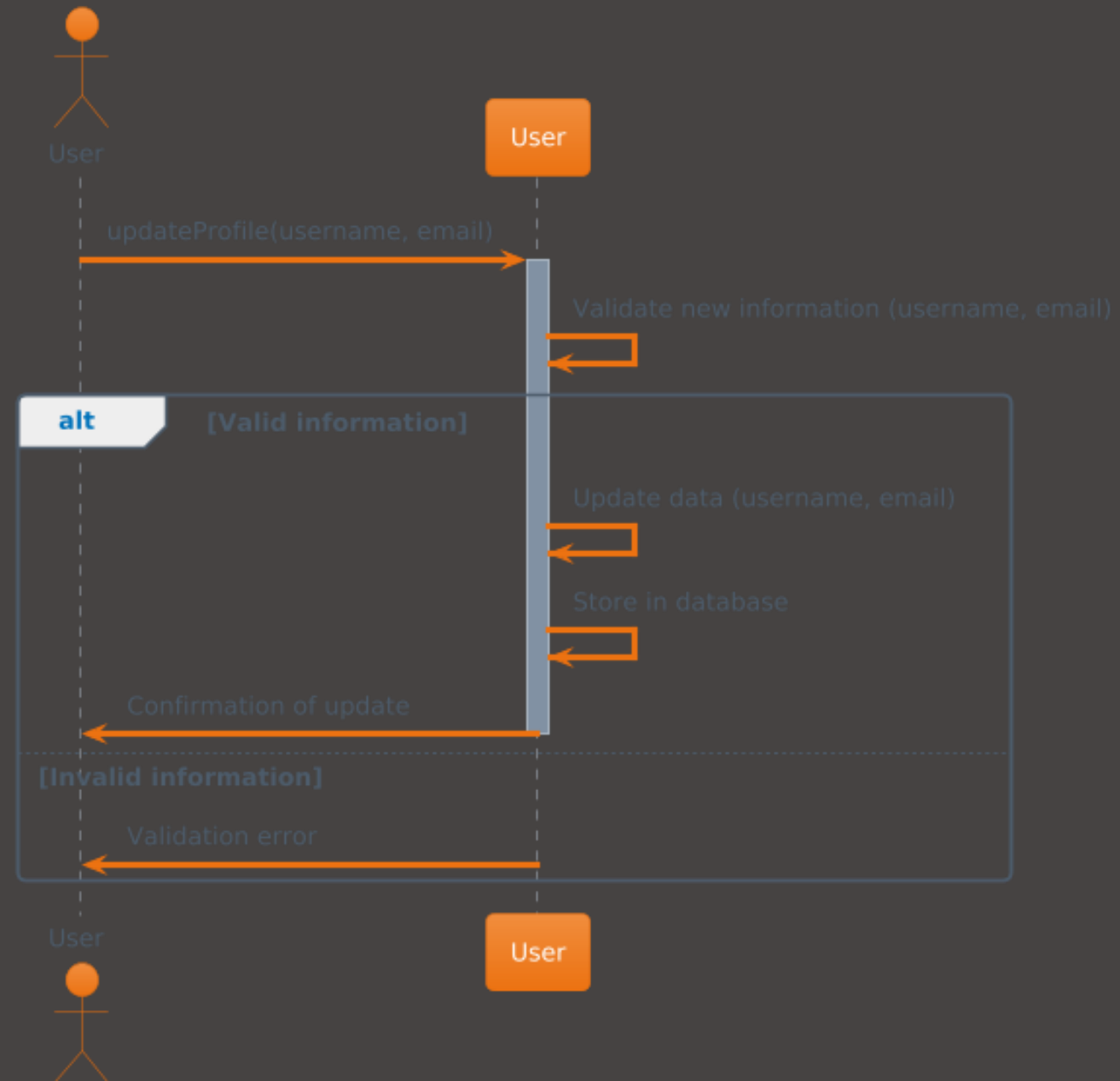
Crear base de datos



# Diagramas uml

## Diagramas de secuencia

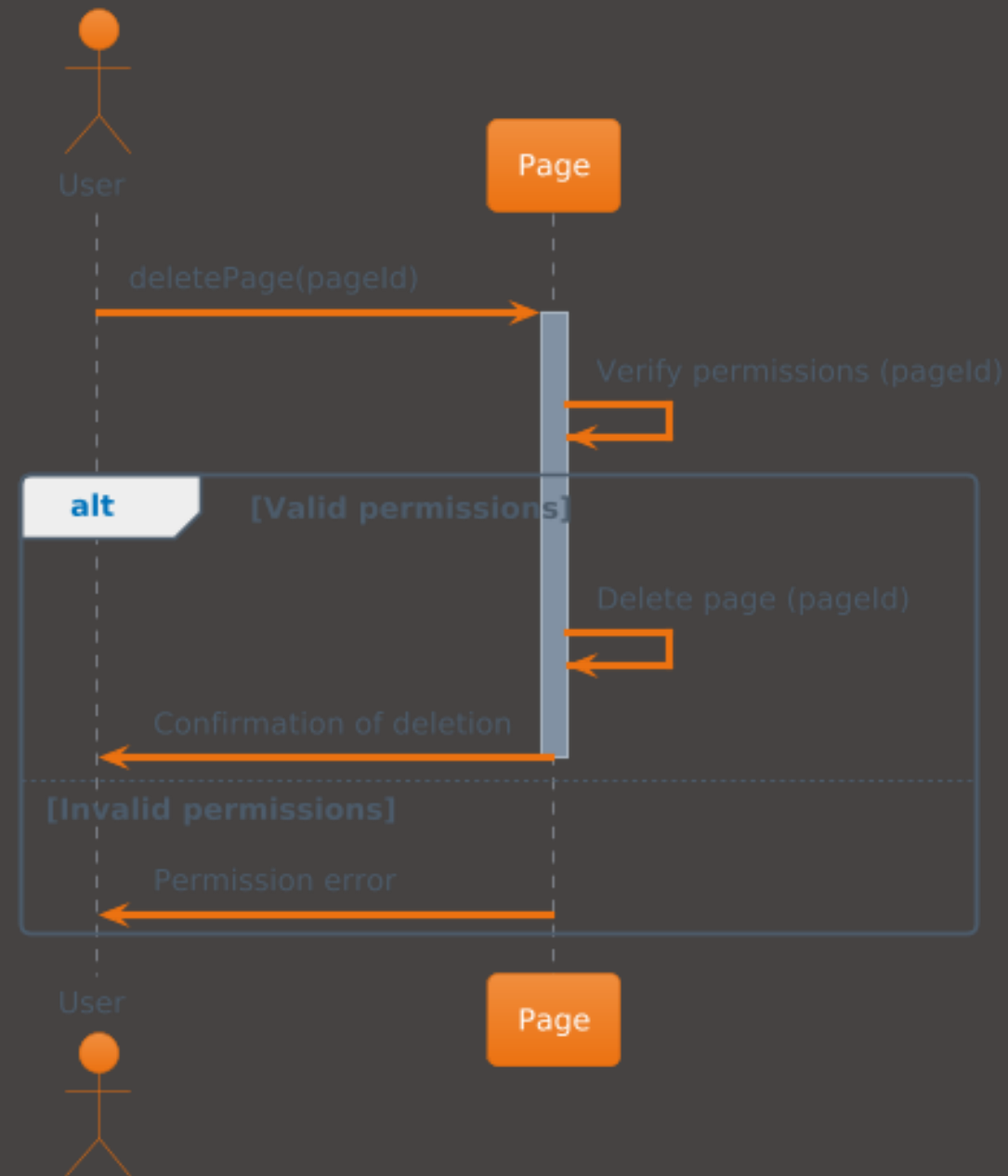
### Actualizar perfil de usuario



# Diagramas uml

## Diagramas de secuencia

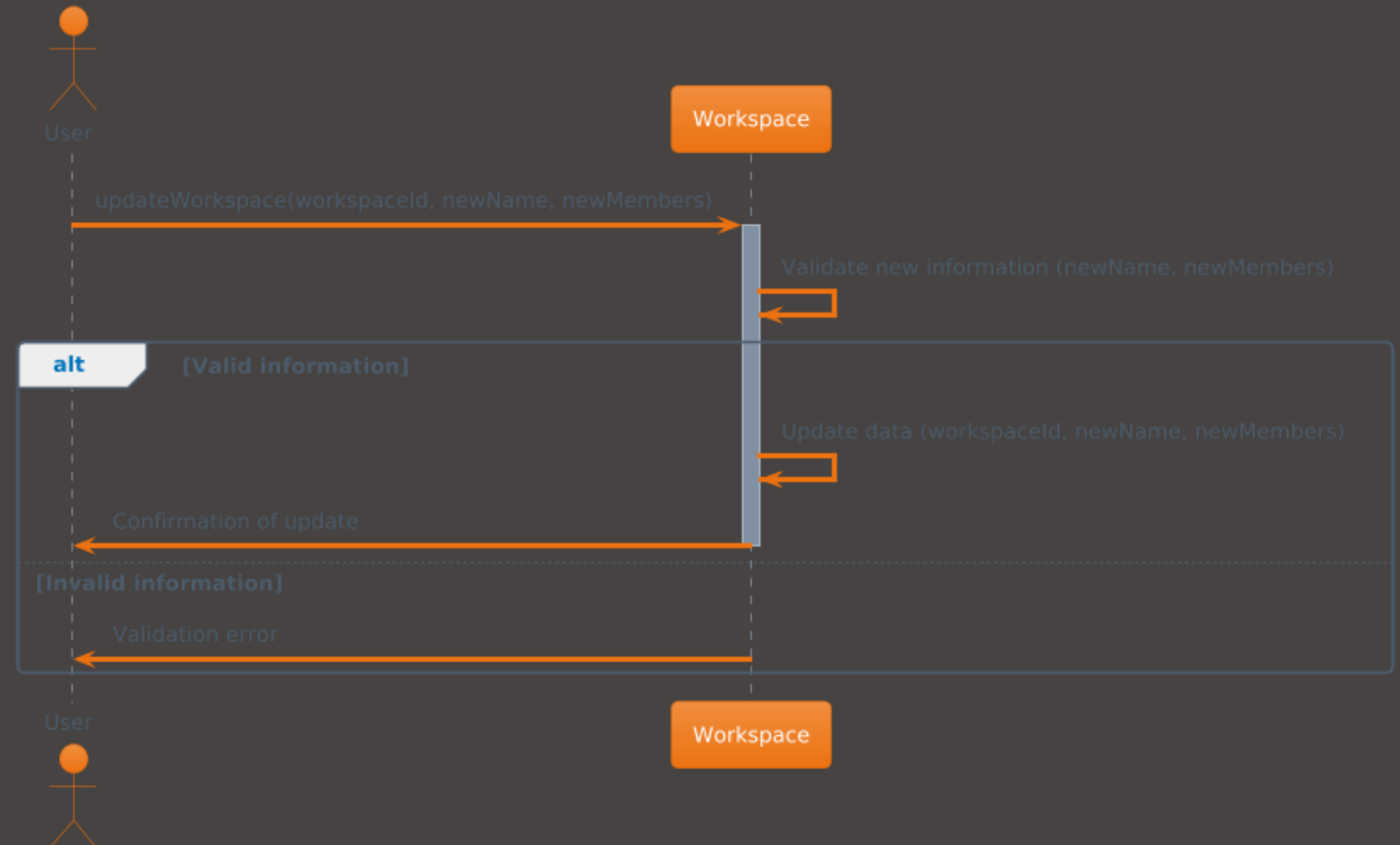
### Borrar pagina



# Diagramas uml

## Diagramas de secuencia

### Actualizar espacio de trabajo

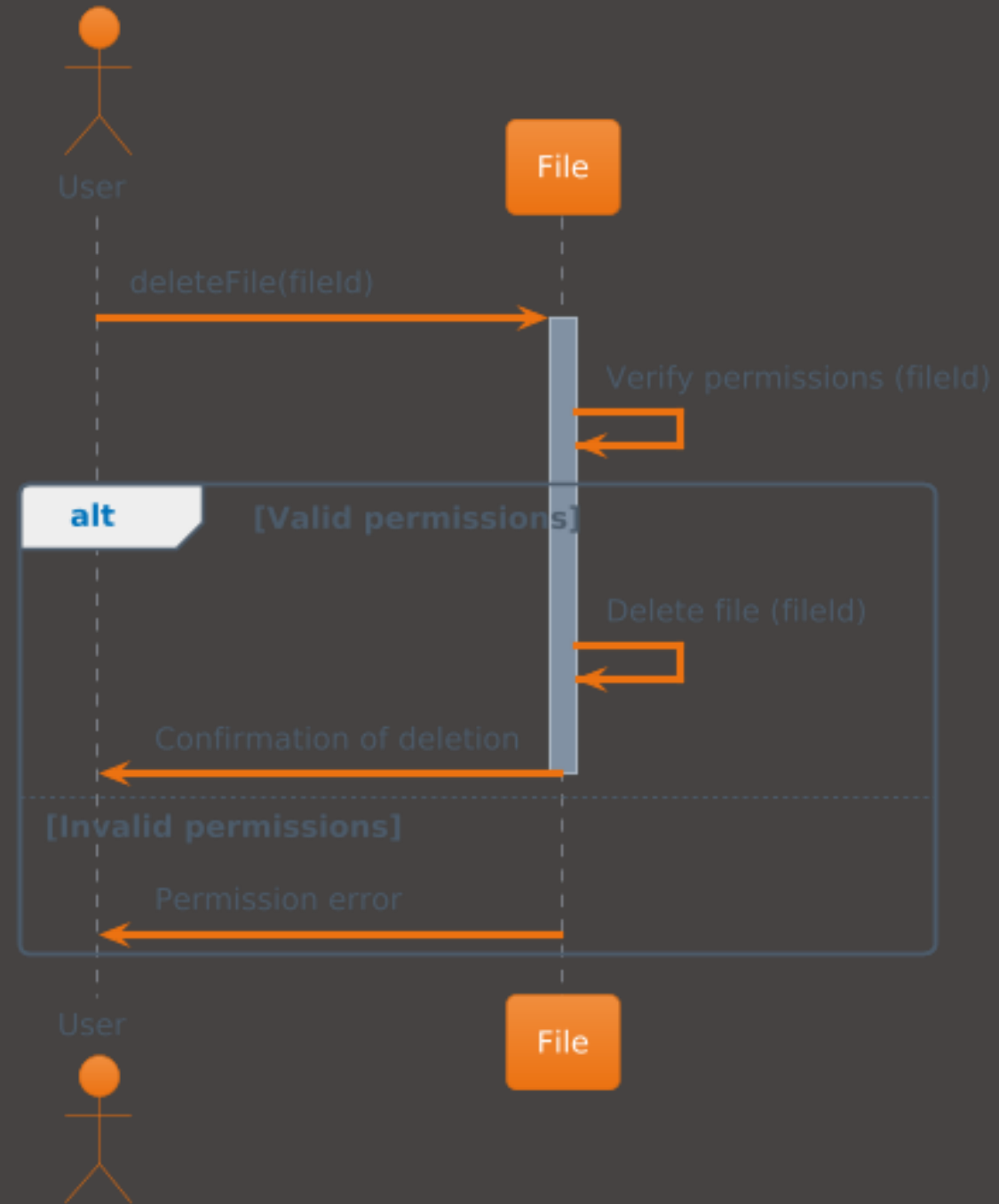




# Diagramas uml

## Diagramas de secuencia

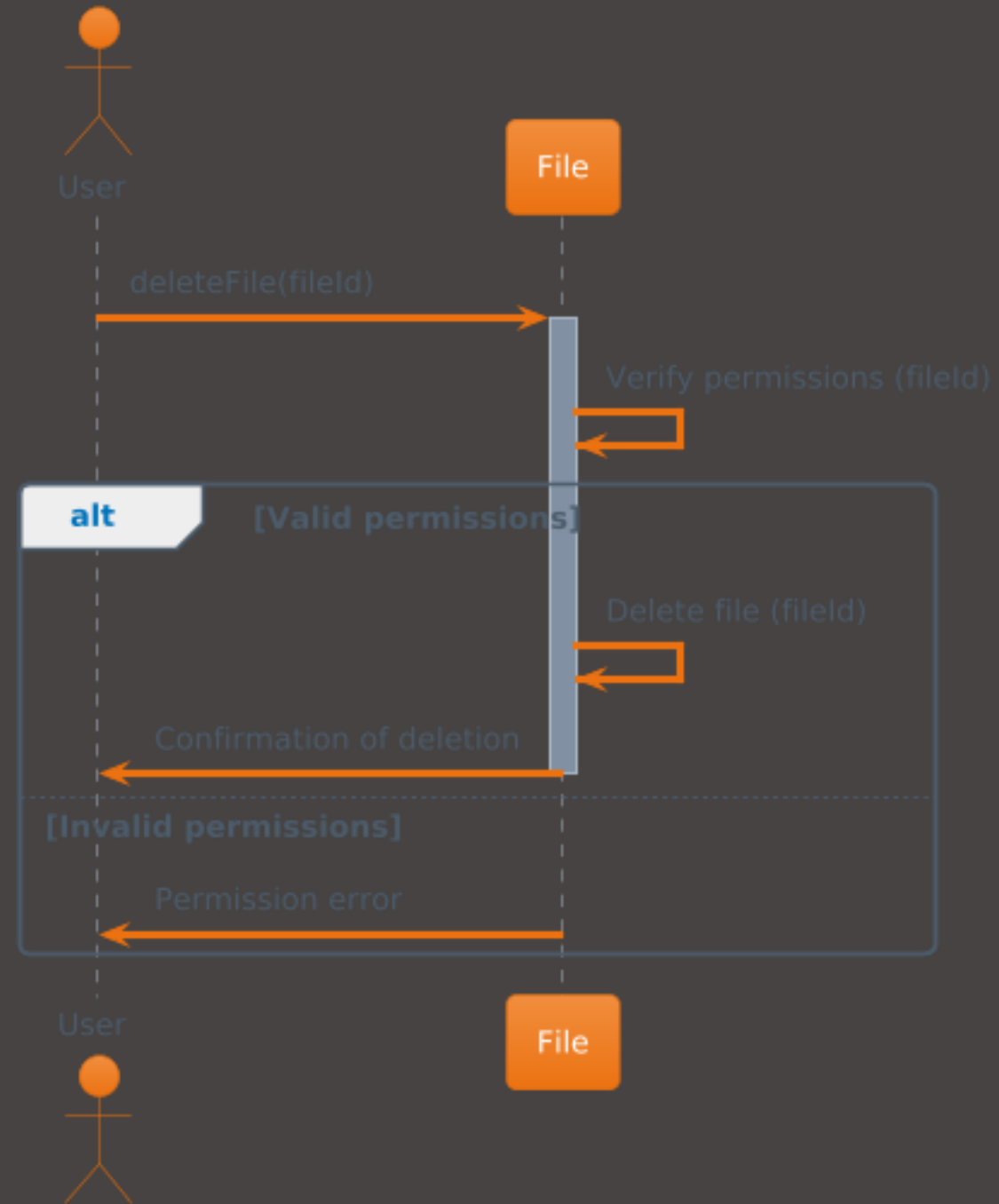
Borrar archivo



# Diagramas uml

## Diagramas de secuencia

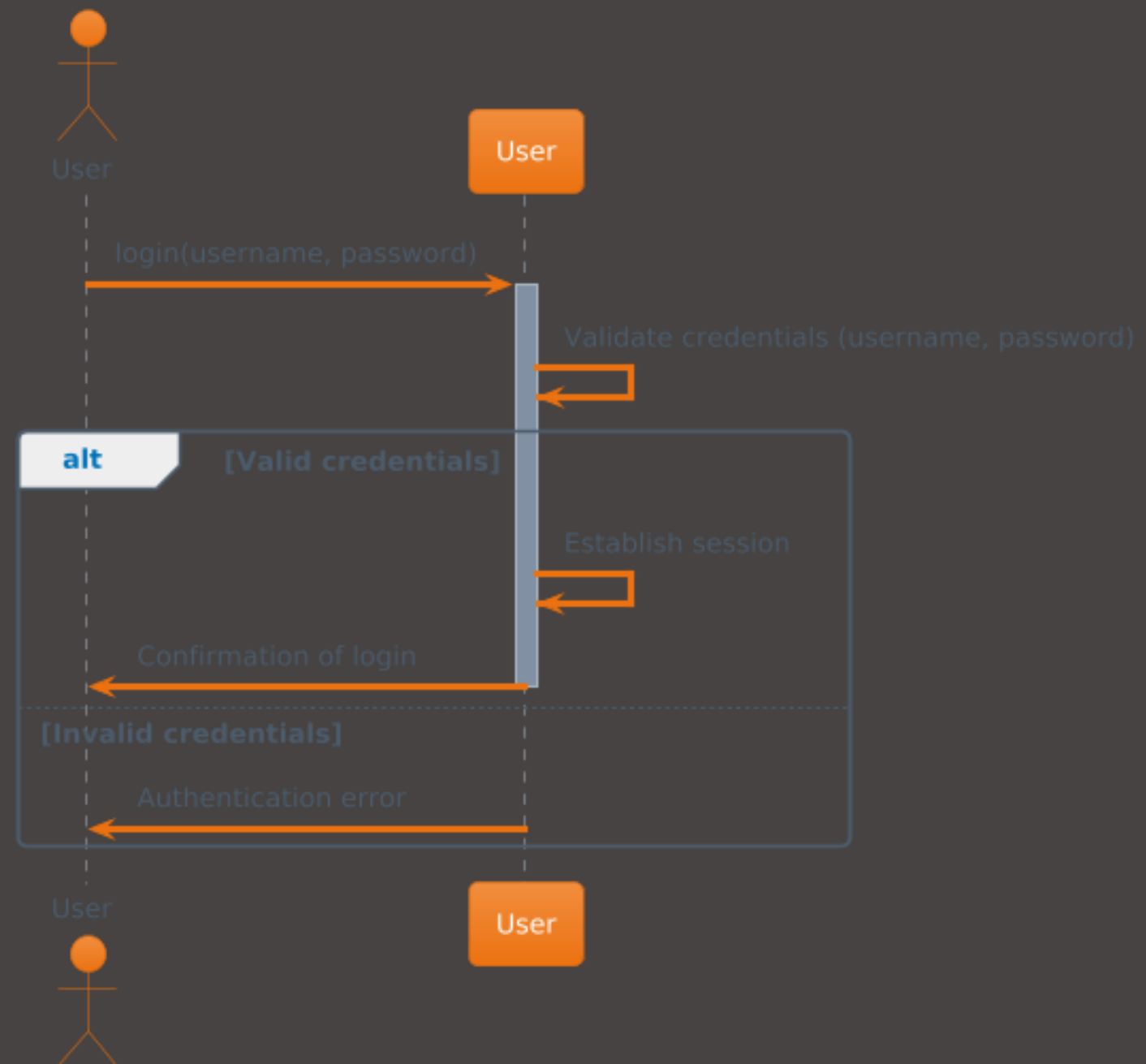
Actualizar base de datos



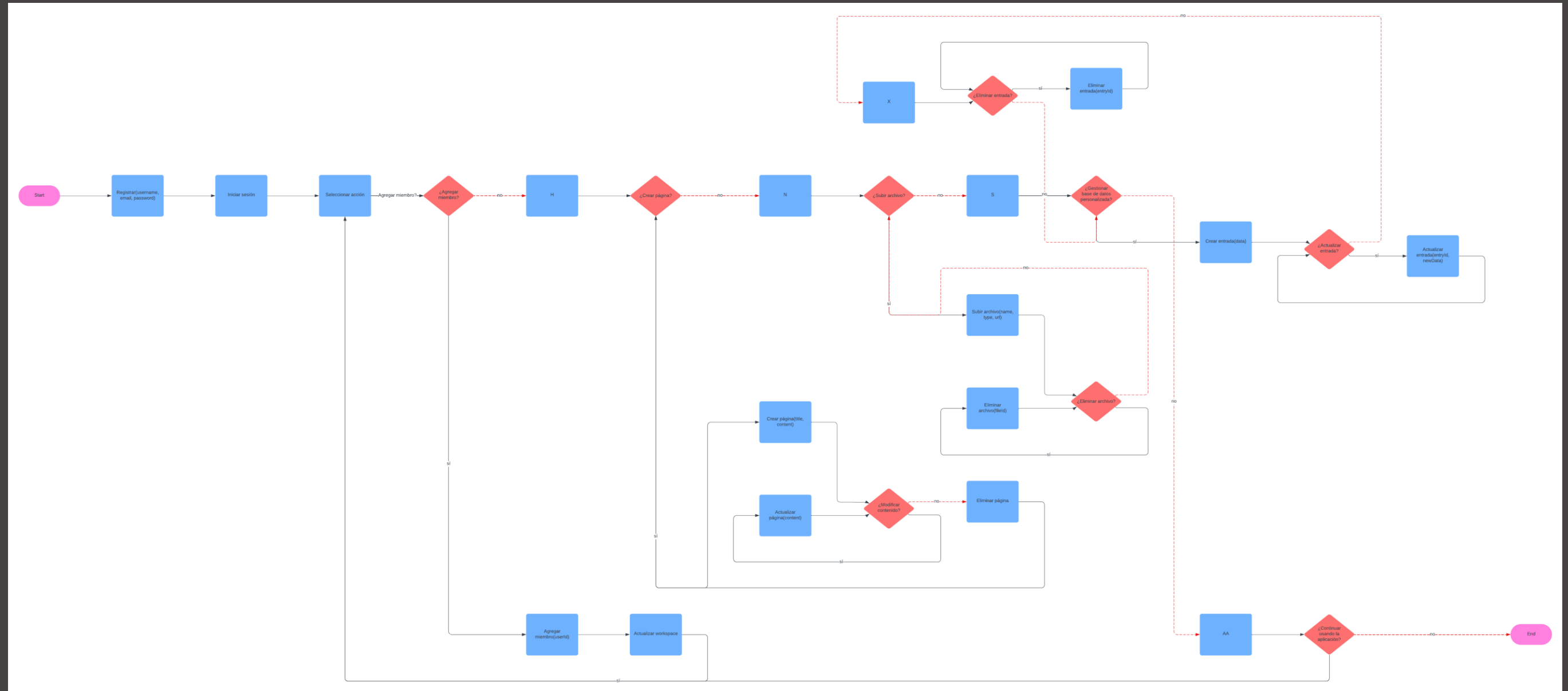
# Diagramas uml

## Diagramas de secuencia

### Inicio de sesión



# Diagrama de actividades



# Funcionalidades Mínimas

1

## Registro y Sesión

Registro de usuarios e inicio de sesión seguro.

2

## Páginas y Búsqueda

Creación, edición y búsqueda de información.

3

## Permisos y Estadísticas

Gestión de permisos y visualización de estadísticas.



# Conclusión

## Resumen

1

Avances significativos en el desarrollo del proyecto.

## Logros

2

Definición de objetivos, funcionalidades y diseño.

## Próximos Pasos

3

Desarrollo de funcionalidades mínimas y mejoras.

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.Scanner;

public class MainApp {
    public static void main(String[] args) {
        // Using try-with-resources to automatically close the Scanner
        try (Scanner scanner = new Scanner(System.in)) {
            // Welcome message
            System.out.println("Welcome to the console application.");
            // Ask the user if they want to register or log in
            System.out.println("What would you like to do? (1) Register, (2) Login");
            int option = scanner.nextInt(); // Capture the user's choice (1 or 2)
            scanner.nextLine(); // Clear the input buffer after reading the integer

            // Prompt the user to enter their username
            System.out.print("Enter your username: ");
            String username = scanner.nextLine();

            // Prompt the user to enter their password
            System.out.print("Enter your password: ");
            String password = scanner.nextLine();

            // Determine the command to execute based on the chosen option
            String command = option == 1 ? "register" : "login";

            // Build the command to execute the Python script with the given arguments
            String[] commandArray = {"python3", "user_management.py", command, username, password};
            Process process = Runtime.getRuntime().exec(commandArray); // Execute the command

            // Read and print the output from the executed process (the Python script)
            BufferedReader reader = new BufferedReader(new InputStreamReader(process.getInputStream()));
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line); // Print each line of output from the script
            }

            // Wait for the process to finish
            process.waitFor();
        } catch (Exception e) {
            // Display an error message if something goes wrong with the process
            System.out.println("Error interacting with the authentication system.");
        }
    }
}
```