



Actividad 4 - Utilizando sistemas de control de versiones

Asignatura

Mantenimiento de Software

Presenta

Saray Alexandra Cotes Cotes

Docente

Diana Candia

Medellín – Colombia 2023

Link del Repositorio de GitHub:

<https://github.com/Saraycotes92/dreams.git>

Historia 1: Registrar un sueño

Como usuario, quiero poder registrar los detalles de mi sueño inmediatamente después de despertar, para no olvidar ningún detalle.

Funcionalidad: La aplicación permite a los usuarios ingresar los detalles de su sueño en un diario digital. Los usuarios pueden agregar títulos a sus sueños, describir lo que sucedió, las emociones que sintieron y cualquier otro detalle que crean que es importante.

Historia 2: Interpretación automática de sueños

Como usuario, quiero que la aplicación interprete automáticamente mi sueño basándose en los detalles que proporcioné para obtener una mejor comprensión de lo que podría significar.

Funcionalidad: La aplicación tiene una función de inteligencia artificial que analiza los detalles del sueño ingresados por el usuario y proporciona una interpretación basada en simbolismos comunes y teorías psicológicas.

Historia 3: Comparar sueños a lo largo del tiempo

Como usuario, quiero poder ver mis sueños anteriores y cómo han cambiado a lo largo del tiempo para identificar patrones o tendencias.

Funcionalidad: La aplicación almacena todos los sueños que los usuarios han ingresado y les permite ver su historial de sueños. Los usuarios pueden filtrar sus sueños por fecha, contenido, emoción y otros criterios.

Historia 4: Comunidad de interpretación de sueños

Como usuario, quiero poder compartir mi sueño con una comunidad de usuarios de la aplicación para obtener diferentes perspectivas sobre su interpretación.

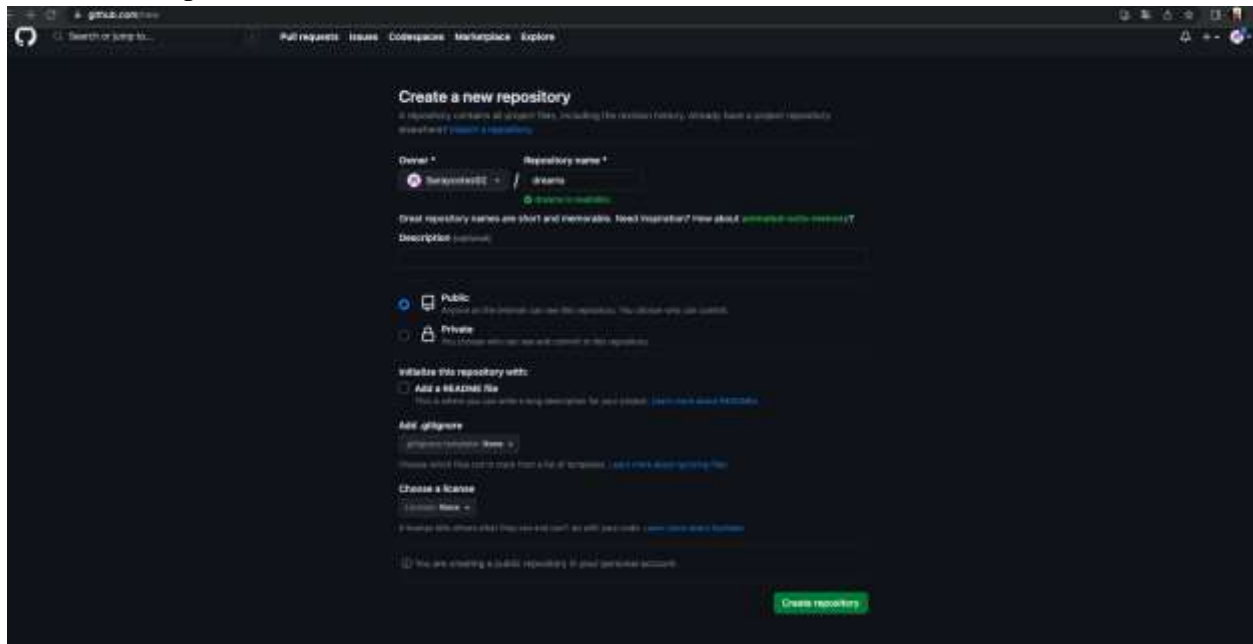
Funcionalidad: La aplicación cuenta con un foro de la comunidad donde los usuarios pueden compartir sus sueños y discutir sus interpretaciones. Los usuarios pueden optar por mantener ciertos detalles de su sueño privados si así lo desean.

Historia 5: Recursos de aprendizaje sobre sueños

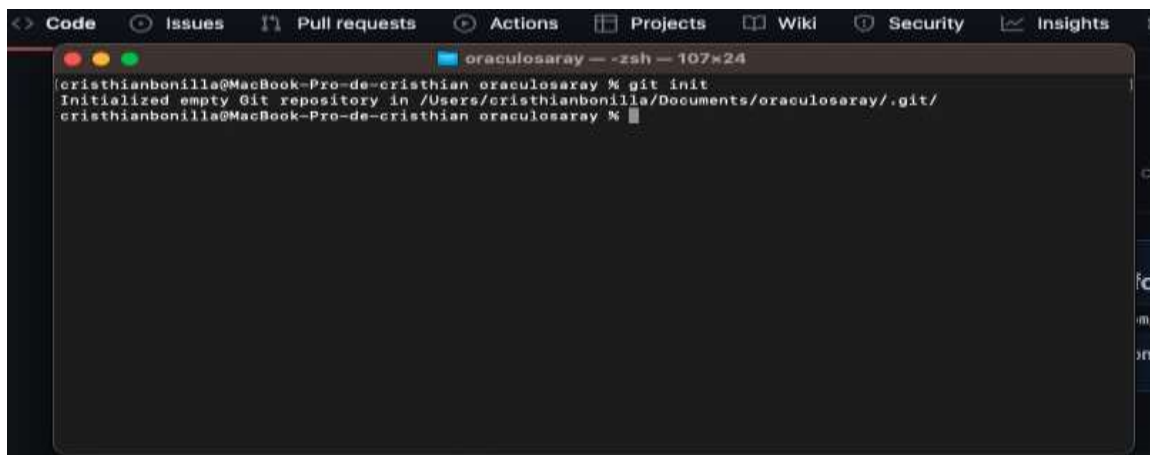
Como usuario, quiero tener acceso a recursos y artículos sobre la interpretación de sueños para ampliar mis conocimientos sobre el tema.

Funcionalidad: La aplicación cuenta con una biblioteca de recursos y artículos sobre la interpretación de sueños, la psicología de los sueños y la importancia de los sueños en diferentes culturas. Los usuarios pueden guardar artículos para leer más tarde y buscar temas específicos.

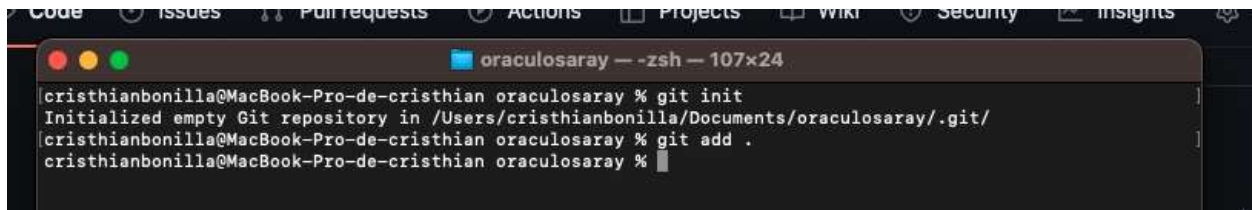
Creamos el repositorio manualmente en GitHub



Ejecutar **git init**: Este comando se utiliza para crear un nuevo repositorio Git. Al ejecutarlo en un directorio, se inicializa un repositorio vacío y se crea una carpeta oculta llamada “git” que contiene todos los archivos y metadatos necesarios para el control de versiones.

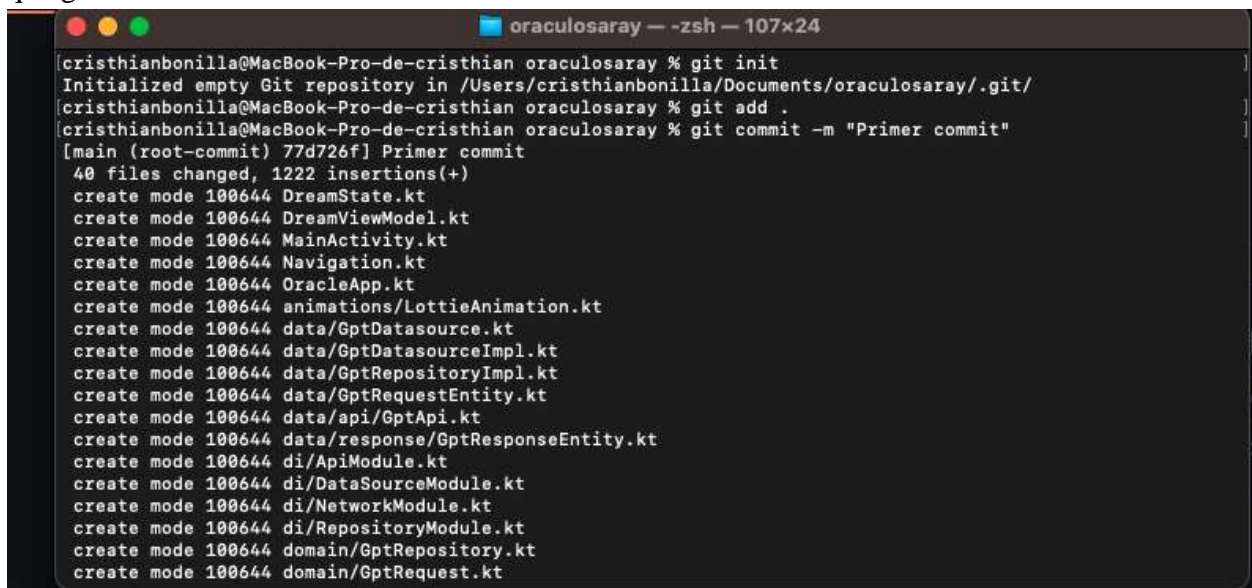


Ejecutar **git add.**: Con este comando, puedes agregar cambios o archivos específicos al área de preparación (también conocida como "staging area") para ser incluidos en el próximo commit. Puedes indicar los archivos de forma individual o utilizar patrones para seleccionar múltiples archivos.

A terminal window titled 'oraculosaray — zsh — 107x24' showing the execution of 'git init' and 'git add .' commands. The output of 'git init' indicates an initialized empty repository in the current directory.

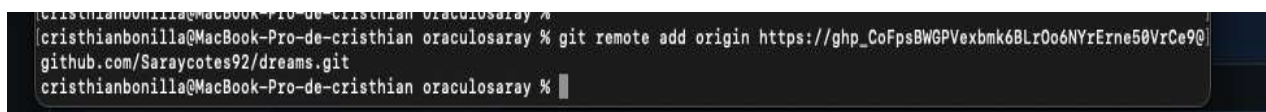
```
cristhianbonilla@MacBook-Pro-de-cristhian oraculosaray % git init
Initialized empty Git repository in /Users/cristhianbonilla/Documents/oraculosaray/.git/
cristhianbonilla@MacBook-Pro-de-cristhian oraculosaray % git add .
cristhianbonilla@MacBook-Pro-de-cristhian oraculosaray %
```

Ejecutar **git commit -m “primer commit”**: Después de haber agregado los cambios al área de preparación, puedes utilizar este comando para crear un nuevo commit que registre esos cambios en la historia del repositorio. La opción -m permite agregar un mensaje descriptivo al commit, que generalmente resume los cambios realizados.

A terminal window titled 'oraculosaray — zsh — 107x24' showing the execution of 'git commit -m "Primer commit"'. The output lists 40 files changed with 1222 insertions, including various Kotlin files for a project named 'Dreams'.

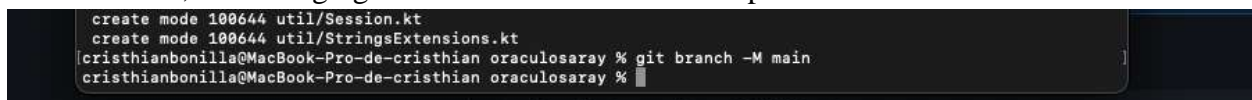
```
cristhianbonilla@MacBook-Pro-de-cristhian oraculosaray % git init
Initialized empty Git repository in /Users/cristhianbonilla/Documents/oraculosaray/.git/
cristhianbonilla@MacBook-Pro-de-cristhian oraculosaray % git add .
cristhianbonilla@MacBook-Pro-de-cristhian oraculosaray % git commit -m "Primer commit"
[main (root-commit) 77d726f] Primer commit
40 files changed, 1222 insertions(+)
create mode 100644 DreamState.kt
create mode 100644 DreamViewModel.kt
create mode 100644 MainActivity.kt
create mode 100644 Navigation.kt
create mode 100644 OracleApp.kt
create mode 100644 animations/LottieAnimation.kt
create mode 100644 data/GptDatasource.kt
create mode 100644 data/GptDatasourceImpl.kt
create mode 100644 data/GptRepositoryImpl.kt
create mode 100644 data/GptRequestEntity.kt
create mode 100644 data/api/GptApi.kt
create mode 100644 data/response/GptResponseEntity.kt
create mode 100644 di/ApiModule.kt
create mode 100644 di/DataSourceModule.kt
create mode 100644 di/NetworkModule.kt
create mode 100644 di/RepositoryModule.kt
create mode 100644 domain/GptRepository.kt
create mode 100644 domain/GptRequest.kt
```

git branch -M main: Este comando se utiliza para renombrar la rama actual del repositorio. En este caso, se está cambiando el nombre de la rama a "main". Puedes utilizar otro nombre en lugar de "main" si lo prefieres.

A terminal window titled 'oraculosaray' showing the execution of 'git remote add origin' with a GitHub repository URL.

```
cristhianbonilla@MacBook-Pro-de-cristhian oraculosaray % git remote add origin https://ghp_CoFpsBWGPVexbmk6BLr0o6NYrErne50VrCe9Q@github.com/Saraycotes92/dreams.git
cristhianbonilla@MacBook-Pro-de-cristhian oraculosaray %
```

git branch -M main <https://github.com/Saraycotes92/dreams.git> con token personal: Con este comando, se está agregando un enlace remoto a tu repositorio local.

A terminal window titled 'oraculosaray' showing the execution of 'git branch -M main'. The output shows the creation of 'util/Session.kt' and 'util/StringsExtensions.kt' files.

```
create mode 100644 util/Session.kt
create mode 100644 util/StringsExtensions.kt
cristhianbonilla@MacBook-Pro-de-cristhian oraculosaray % git branch -M main
cristhianbonilla@MacBook-Pro-de-cristhian oraculosaray %
```

git push -u origin main: Este comando envía los commits locales al repositorio remoto especificado (en este caso, "origin") en la rama principal ("main"). La opción -u establece una conexión de seguimiento entre la rama local y la rama remota, lo que permite utilizar simplemente `git push` en futuros envíos.

```
github.com/Saraycotes92/dreams.git
cristianbonilla@MacBook-Pro-de-cristhian oraculosaray % git push -u origin main
Enumerating objects: 53, done.
Counting objects: 100% (53/53), done.
Delta compression using up to 10 threads
Compressing objects: 100% (48/48), done.
Writing objects: 100% (53/53), 15.10 KiB | 3.77 MiB/s, done.
Total 53 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), done.
To https://github.com/Saraycotes92/dreams.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
cristianbonilla@MacBook-Pro-de-cristhian oraculosaray %
```

git status: Al ejecutar este comando, obtenemos la información sobre el estado actual del repositorio. Te mostrará los archivos modificados, los archivos nuevos que aún no se han agregado al área de preparación y cualquier otro detalle relevante.

```
cristianbonilla@MacBook-Pro-de-cristhian oraculosaray % git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
cristianbonilla@MacBook-Pro-de-cristhian oraculosaray %
```

git log: Con este comando, puedes ver un registro detallado de todos los commits realizados en el repositorio. Te mostrará información como el autor, la fecha, el mensaje del commit y un identificador único (SHA) para cada commit.

```
github.com/Saraycotes92/dreams.git
cristianbonilla@MacBook-Pro-de-cristhian oraculosaray % git log
commit 63434e0310906835b6ae40aff974847e3ed18a8a (HEAD -> main, origin/main)
Author: Saray Cotes <saraycotes@gmail.com>
Date: Thu May 25 11:33:24 2023 -0500

    cambiar colores

commit 40048c732204f5d3c5b0d7593ac419b821dd5236
Author: Saray Cotes <saraycotes@gmail.com>
Date: Thu May 25 11:32:24 2023 -0500

    primer commit
cristianbonilla@MacBook-Pro-de-cristhian oraculosaray %
```

Finalmente añadimos **.gitignore**

```
git add .gitignore
```

Y se hace commit del gitignore

```
git commit -m "Agregado archivo .gitignore"
```

Bibliografía

- Channon S., Straub B, (Apress). (2021). Pro Git - Versión 2.1.22-4- g2264d13. (p7-57)