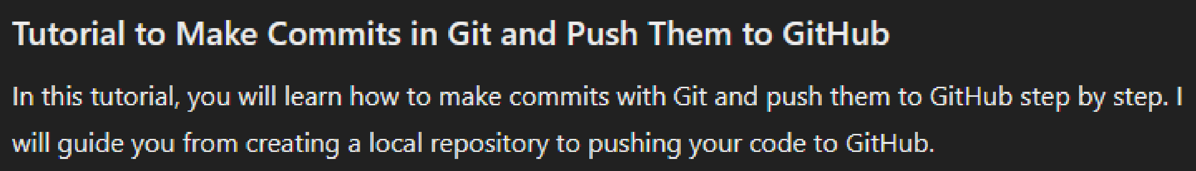
Entornos de Desarrollo

Unidad 0 - Primeros pasos con Git y GitHub

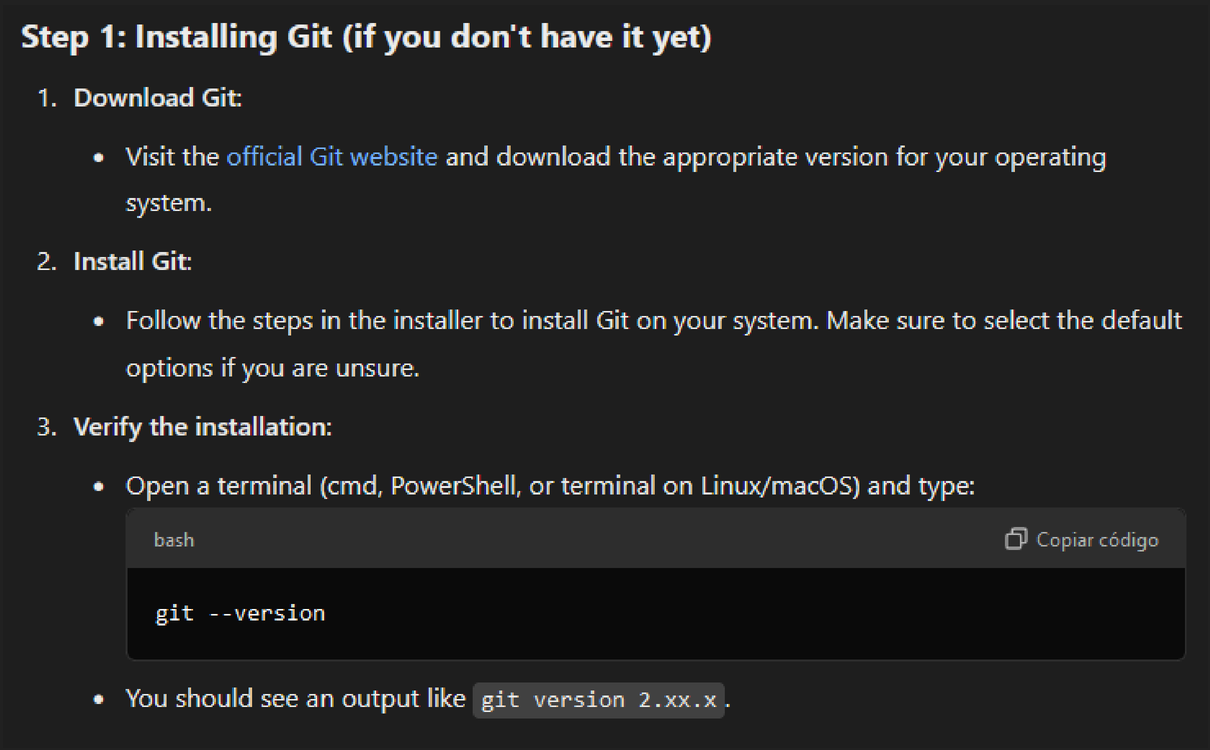
Creado por: **Fernando Parra Zurita**
Con la colaboración de: **ChatGPT**



Tutorial to Make Commits in Git and Push Them to GitHub

In this tutorial, you will learn how to make commits with Git and push them to GitHub step by step. I will guide you from creating a local repository to pushing your code to GitHub.

Git

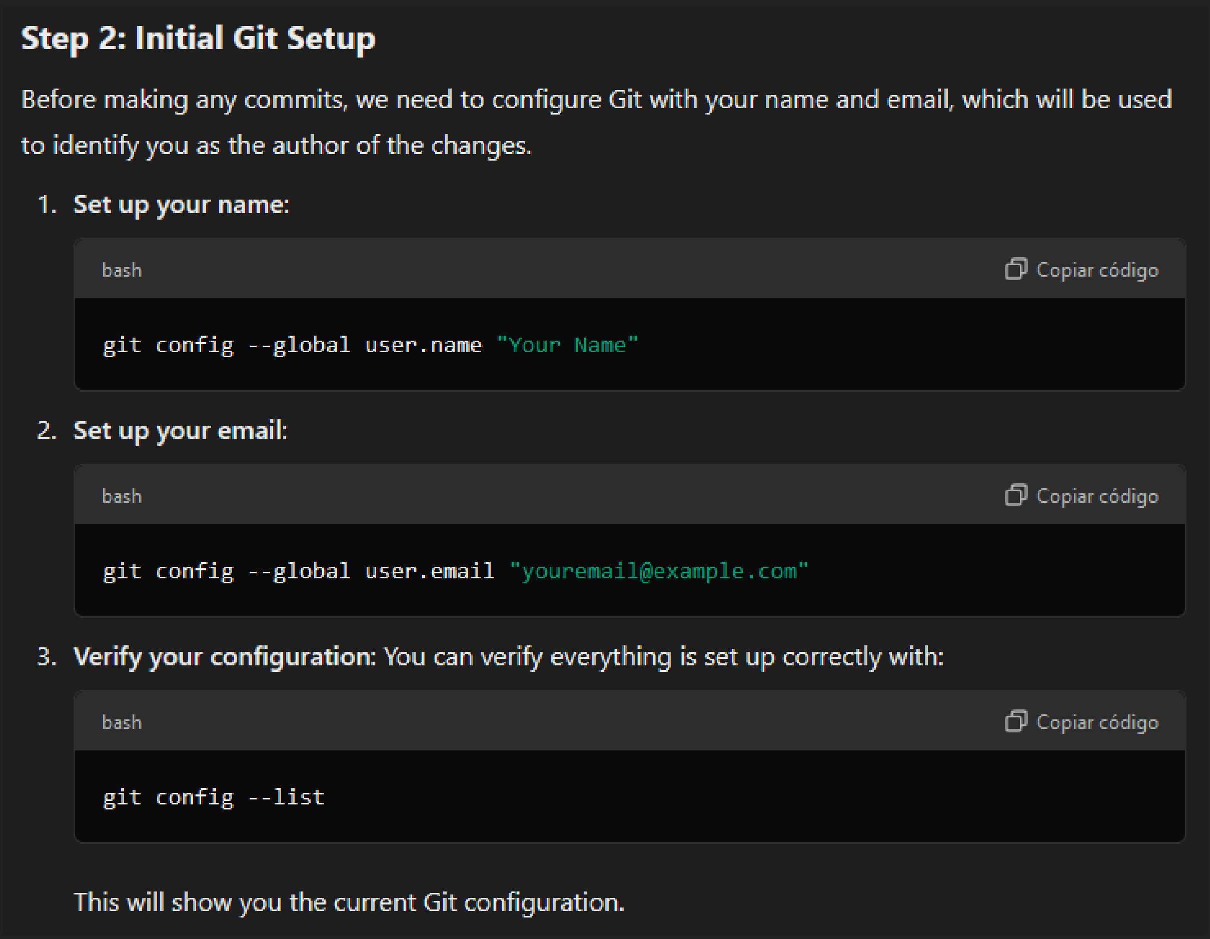


Step 1: Installing Git (if you don't have it yet)

1. **Download Git:**
   - Visit the official Git website and download the appropriate version for your operating system.
2. **Install Git:**
   - Follow the steps in the installer to install Git on your system. Make sure to select the default options if you are unsure.
3. **Verify the installation:**
   - Open a terminal (cmd, PowerShell, or terminal on Linux/macOS) and type:

```bash
git --version
```

   - You should see an output like `git version 2.xx.x`.



Step 2: Initial Git Setup

Before making any commits, we need to configure Git with your name and email, which will be used to identify you as the author of the changes.

1. **Set up your name:**

```bash
git config --global user.name "Your Name"
```

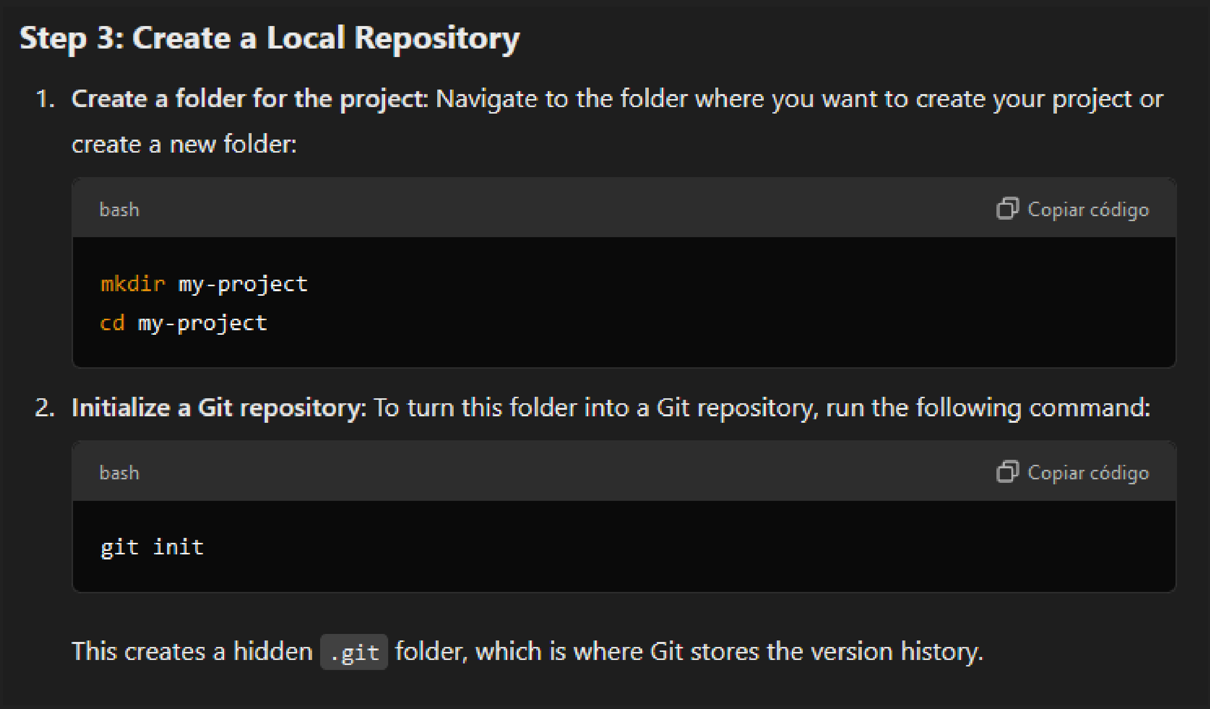
2. **Set up your email:**

```bash
git config --global user.email "youremail@example.com"
```

3. **Verify your configuration:** You can verify everything is set up correctly with:

```bash
git config --list
```

This will show you the current Git configuration.



Step 3: Create a Local Repository

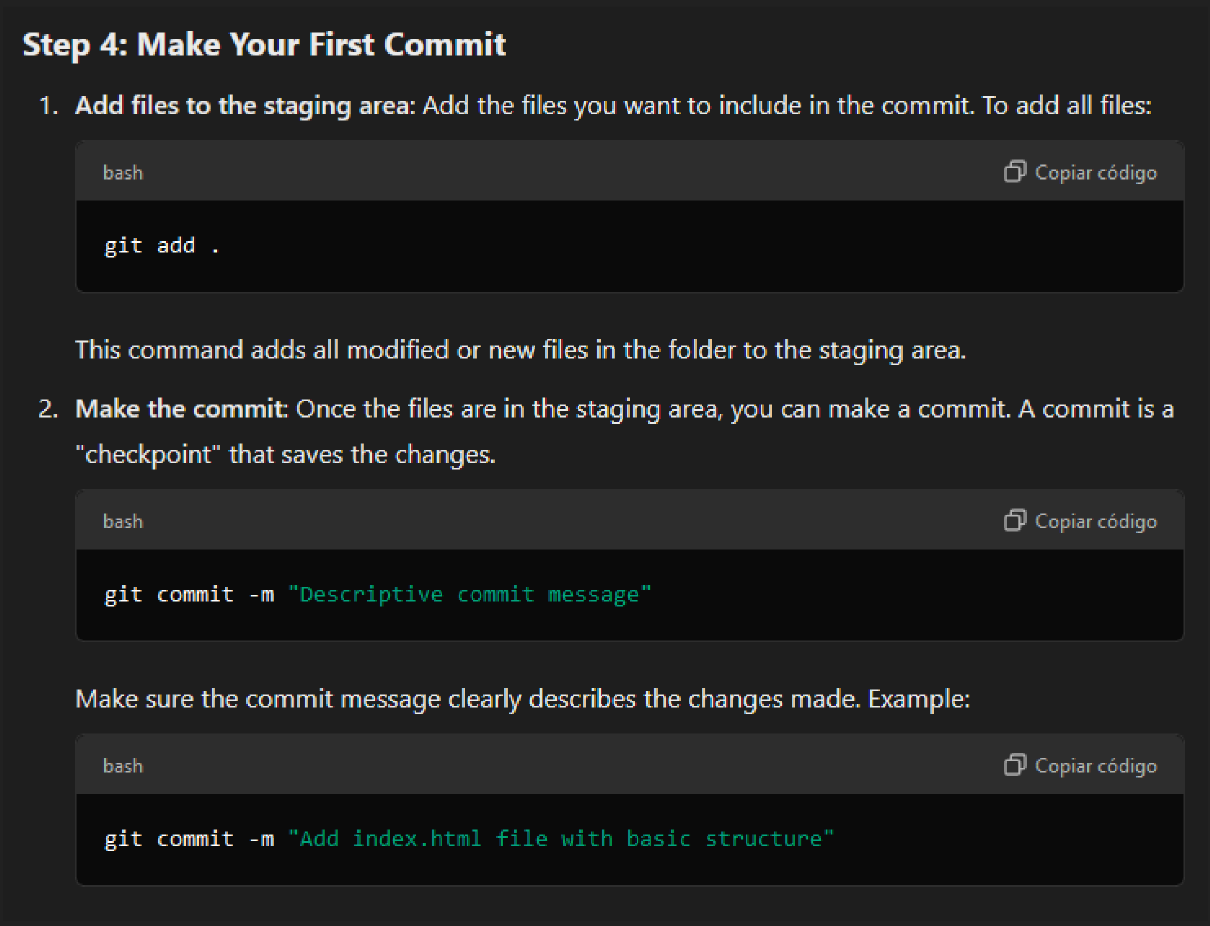
1. **Create a folder for the project:** Navigate to the folder where you want to create your project or create a new folder:

```bash
mkdir my-project
cd my-project
```

2. **Initialize a Git repository:** To turn this folder into a Git repository, run the following command:

```bash
git init
```

This creates a hidden `.git` folder, which is where Git stores the version history.



Step 4: Make Your First Commit

1. **Add files to the staging area:** Add the files you want to include in the commit. To add all files:

   ```bash
   git add .
   ```

   This command adds all modified or new files in the folder to the staging area.

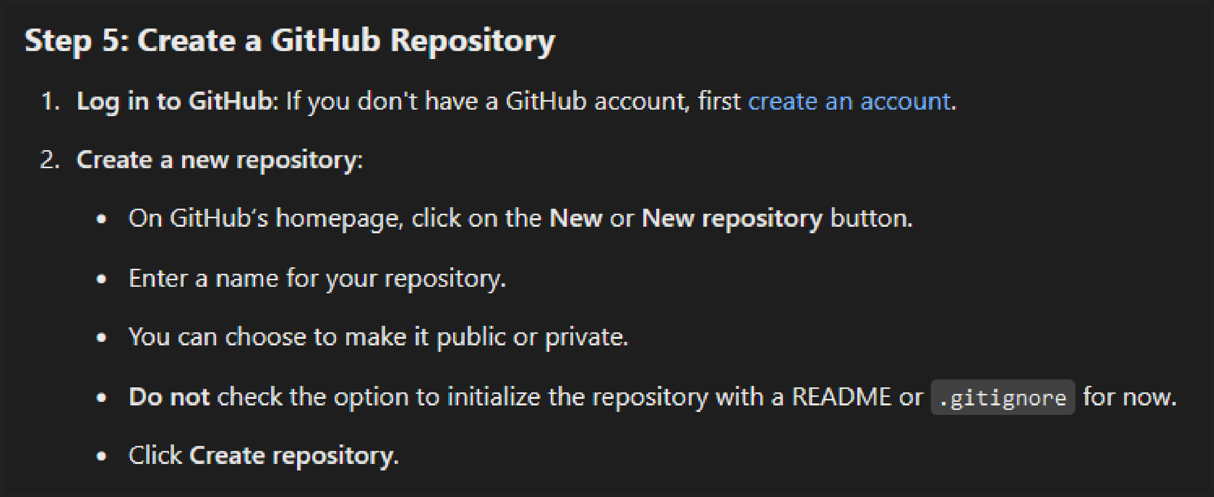
2. **Make the commit:** Once the files are in the staging area, you can make a commit. A commit is a "checkpoint" that saves the changes.

   ```bash
   git commit -m "Descriptive commit message"
   ```

   Make sure the commit message clearly describes the changes made. Example:

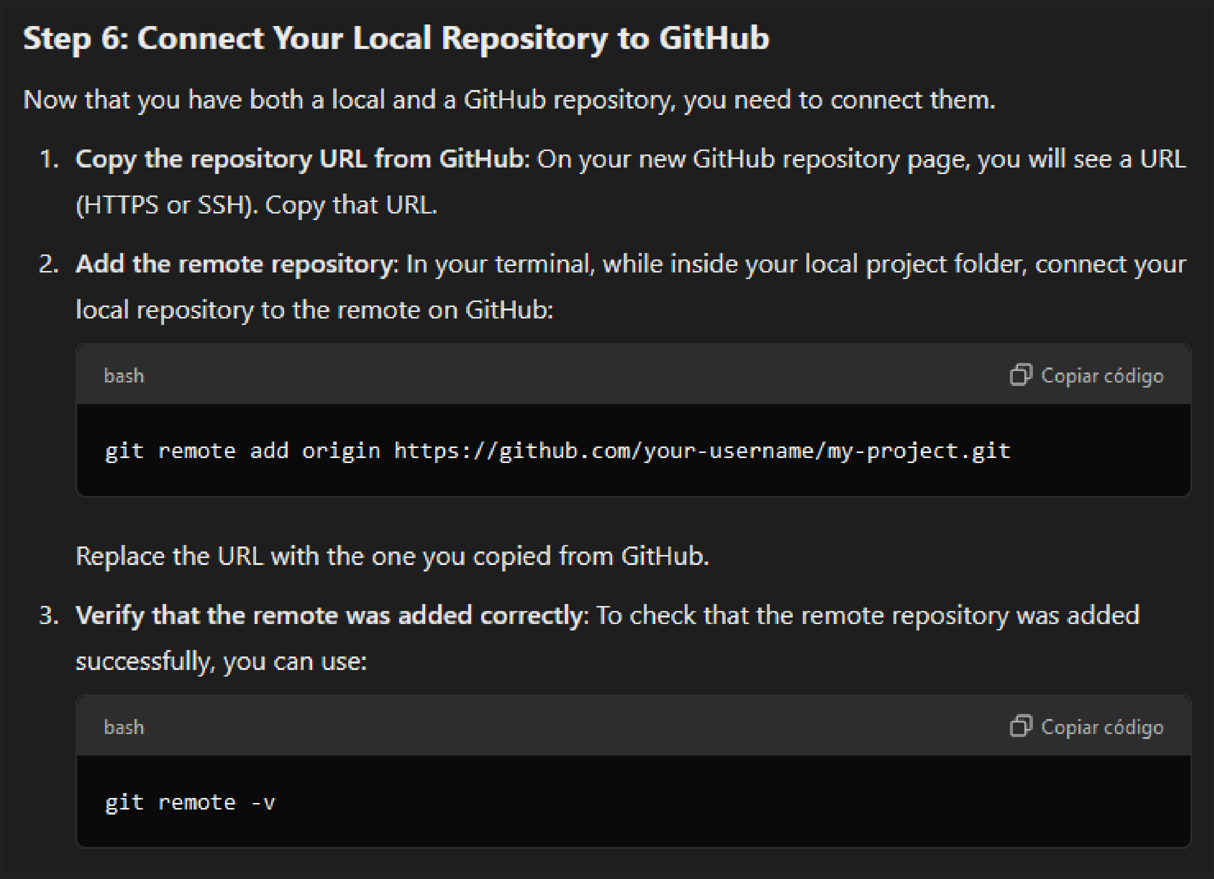
   ```bash
   git commit -m "Add index.html file with basic structure"
   ```

GitHub



Step 5: Create a GitHub Repository

1. **Log in to GitHub**: If you don't have a GitHub account, first create an account.
2. **Create a new repository**:
   - On GitHub's homepage, click on the **New** or **New repository** button.
   - Enter a name for your repository.
   - You can choose to make it public or private.
   - **Do not** check the option to initialize the repository with a README or `.gitignore` for now.
   - Click **Create repository**.



Step 6: Connect Your Local Repository to GitHub

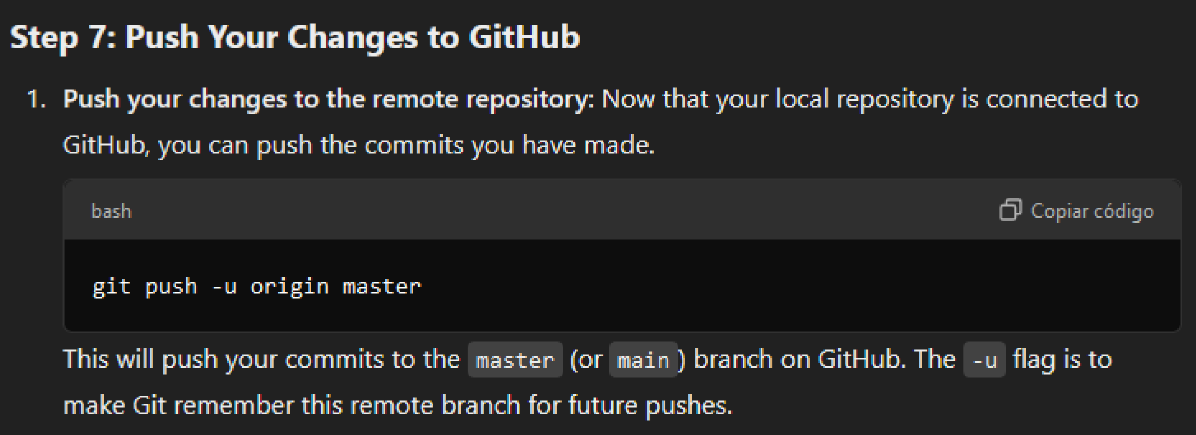
Now that you have both a local and a GitHub repository, you need to connect them.

1. **Copy the repository URL from GitHub**: On your new GitHub repository page, you will see a URL (HTTPS or SSH). Copy that URL.
2. **Add the remote repository**: In your terminal, while inside your local project folder, connect your local repository to the remote on GitHub:

   ```bash
   git remote add origin https://github.com/your-username/my-project.git
   ```

   Replace the URL with the one you copied from GitHub.
3. **Verify that the remote was added correctly**: To check that the remote repository was added successfully, you can use:

   ```bash
   git remote -v
   ```

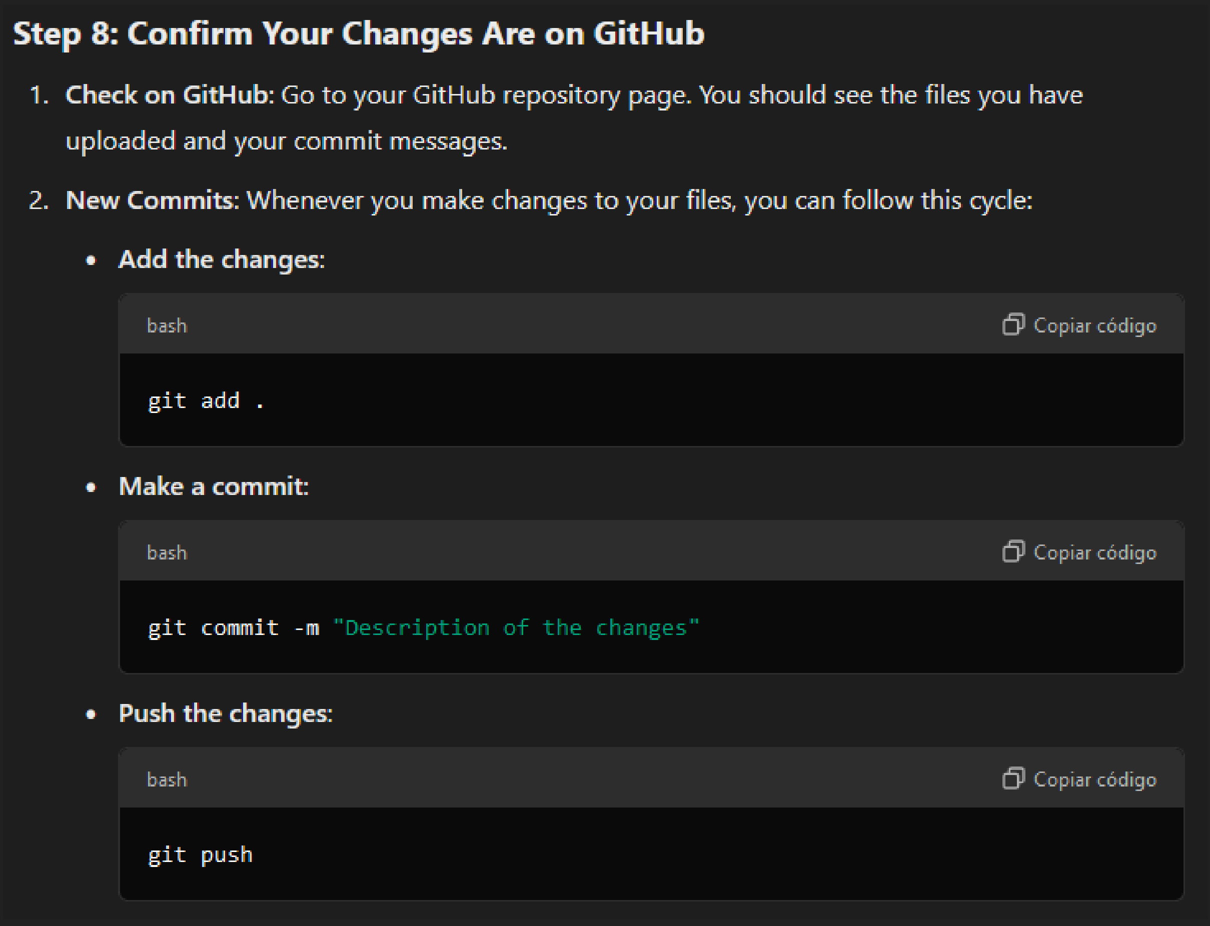


Step 7: Push Your Changes to GitHub

1. **Push your changes to the remote repository**: Now that your local repository is connected to GitHub, you can push the commits you have made.

   ```bash
   git push -u origin master
   ```

   This will push your commits to the `master` (or `main`) branch on GitHub. The `-u` flag is to make Git remember this remote branch for future pushes.



Step 8: Confirm Your Changes Are on GitHub

1. **Check on GitHub**: Go to your GitHub repository page. You should see the files you have uploaded and your commit messages.
2. **New Commits**: Whenever you make changes to your files, you can follow this cycle:
   - **Add the changes**:

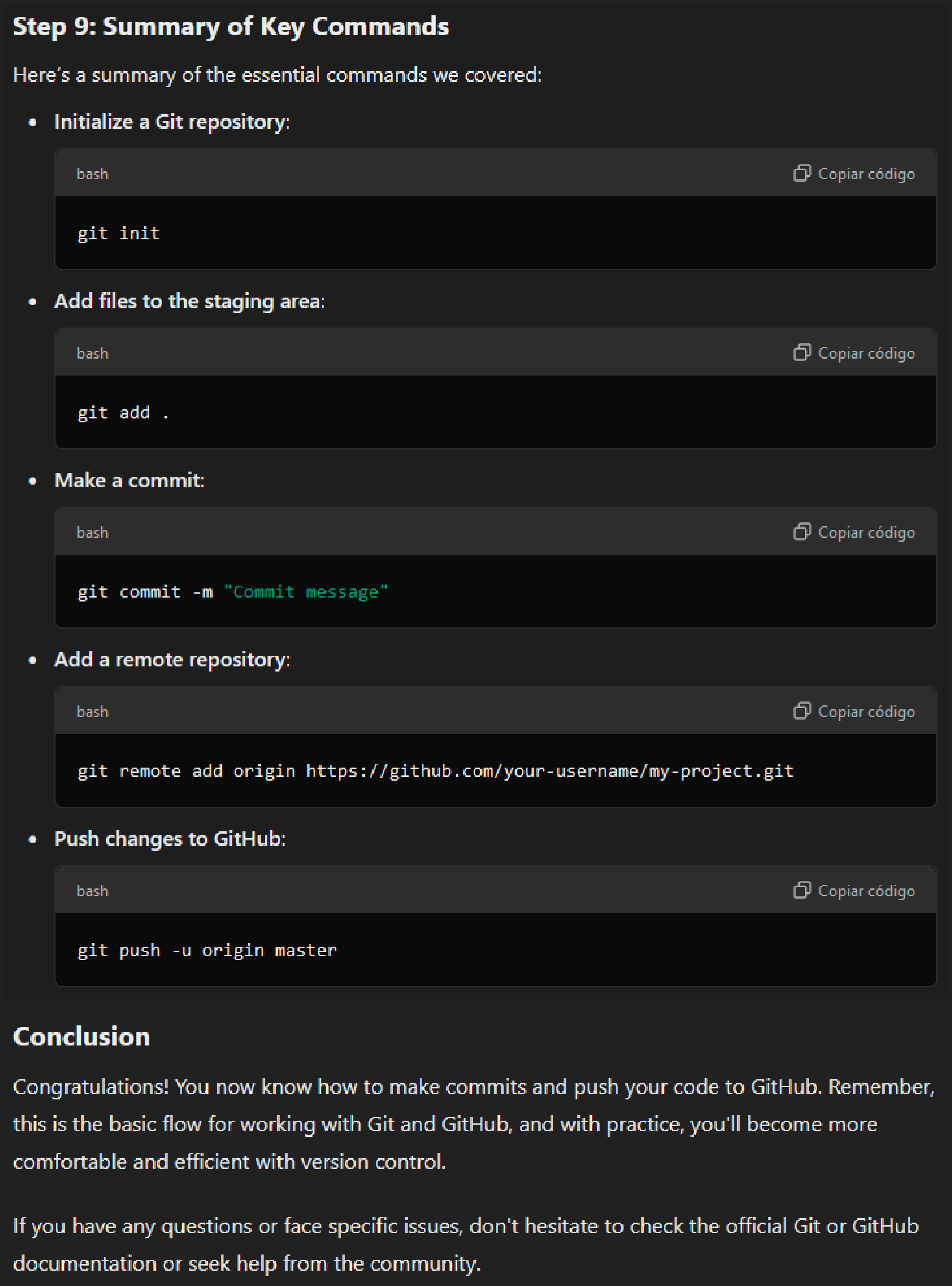
     ```bash
     git add .
     ```

   - **Make a commit**:

     ```bash
     git commit -m "Description of the changes"
     ```

   - **Push the changes**:

     ```bash
     git push
     ```



Step 9: Summary of Key Commands

Here's a summary of the essential commands we covered:

- **Initialize a Git repository**:

```bash
git init
```

- **Add files to the staging area**:

```bash
git add .
```

- **Make a commit**:

```bash
git commit -m "Commit message"
```

- **Add a remote repository**:

```bash
git remote add origin https://github.com/your-username/my-project.git
```

- **Push changes to GitHub**:

```bash
git push -u origin master
```

Conclusion

Congratulations! You now know how to make commits and push your code to GitHub. Remember, this is the basic flow for working with Git and GitHub, and with practice, you'll become more comfortable and efficient with version control.

If you have any questions or face specific issues, don't hesitate to check the official Git or GitHub documentation or seek help from the community.