

## Entrega #2 - Control de versiones y CI/CD

**Grupo:** The devs

**Integrantes:**

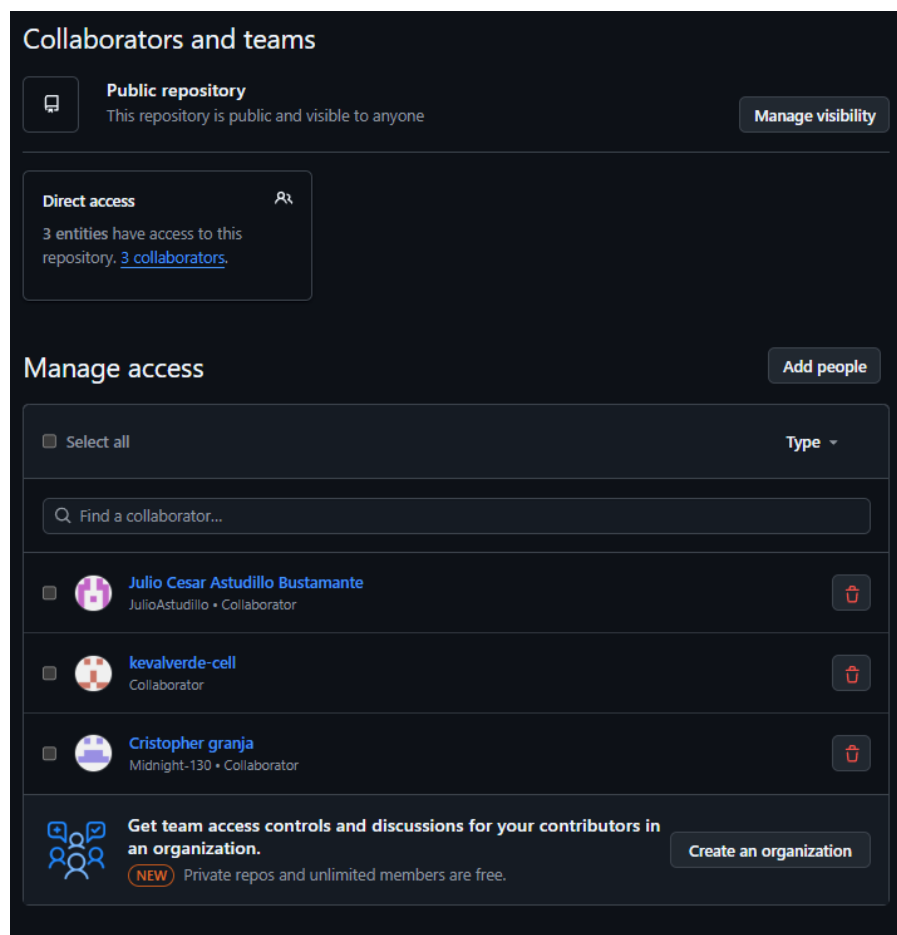
**-Julio César Astudillo Bustamante:** frontend

**-Kevin Eduardo Valverde mullo:** backend

**-Omar Enrique Téllez Sanabria:** Pruebas y CI/CD

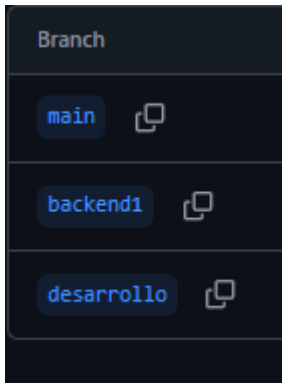
**-Cristopher Alexis Granja Astudillo:** UX/UI

**Flujo de trabajo con git**



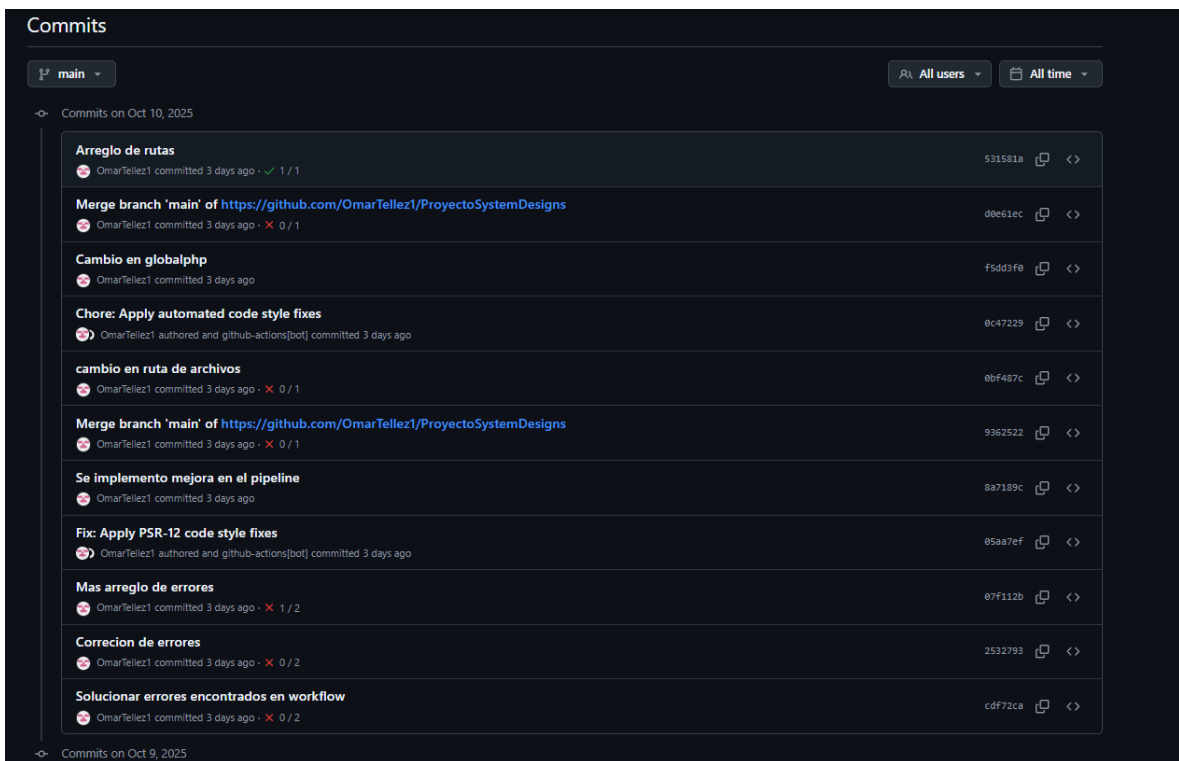
Omar Téllez es el dueño del repositorio del proyecto y sus compañeros de trabajo son colaboradores.

## Ramas del proyecto



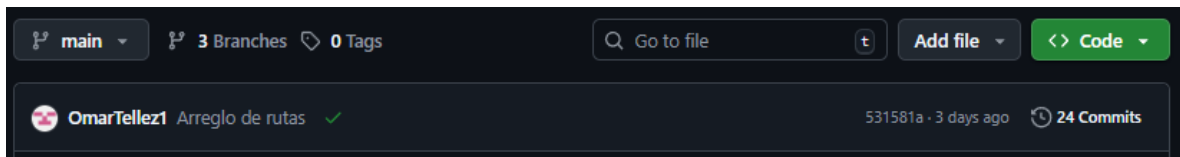
Existieron más pero a medida que avanzaba el proyecto, estas ramas se fusionaban a la rama de desarrollo mediante un pull request, y cuando se aprobaba el pull request, estas ramas cumplían con su función y se eliminaban.

## Commits

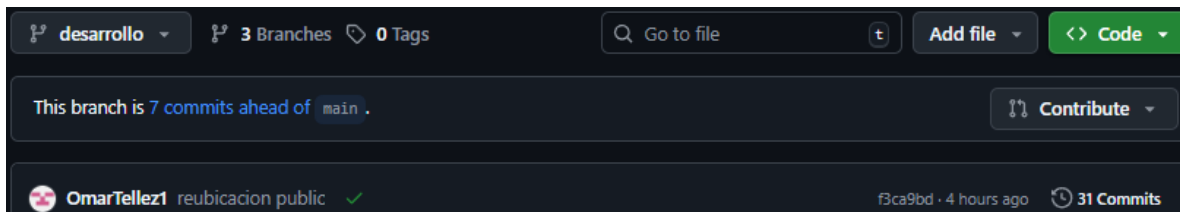


Un poco de commits de mucho más que se hicieron.

De la rama main ha habido un total de 24 commits



De la rama de desarrollo un total de 31 commits. Es decir la rama de desarrollo esta adelantada por 7 commits al momento que se hace este documento.



## Configuración de CI/CD

CD:

```
1  deploy:
2    runs-on: ubuntu-latest
3    # Se ejecuta solo si el job 'build' fue exitoso
4    needs: build
5    steps:
6      - name: Checkout code
7        uses: actions/checkout@v4
8        with:
9          fetch-depth: 0
10     # Añadimos un paso para instalar la Heroku CLI manualmente
11     - name: Install Heroku CLI
12       run: curl https://cli-assets.heroku.com/install.sh | sh
13
14     - name: Deploy to Heroku
15       env:
16         HEROKU_API_KEY: ${ secrets.HEROKU_API_KEY }
17         HEROKU_APP_NAME: ${ secrets.HEROKU_APP_NAME }
18         HEROKU_EMAIL: ${ secrets.HEROKU_EMAIL }
19       run: |
20         # 1. Crea el archivo .netrc para la autenticación de Git
21         cat > ~/.netrc << EOF
22         machine api.heroku.com
23           login $HEROKU_EMAIL
24           password $HEROKU_API_KEY
25         machine git.heroku.com
26           login $HEROKU_EMAIL
27           password $HEROKU_API_KEY
28         EOF
29
30         # 2. Añade el repositorio de Heroku como un "remoto" de Git
31         heroku git:remote -a $HEROKU_APP_NAME
32
33         # 3. Sube el código de la rama actual a la rama 'main' de Heroku
34         git push heroku HEAD:main --force
35
```

CI:

```
1  name: PHP CI Pipeline
2
3  on:
4    push:
5      branches:
6        - main
7    pull_request:
8      branches:
9        - main
10
11 permissions:
12   contents: write # Esencial para que el bot pueda hacer commit
13
14 jobs:
15   build:
16     runs-on: ubuntu-latest
17
18     services:
19       mysql:
20         image: mysql:8.0
21         env:
22           MYSQL_ROOT_PASSWORD: root
23           MYSQL_DATABASE: control_asistencia
24         ports:
25           - 3306:3306
26         options: >-
27           --health-cmd="mysqladmin ping -h localhost"
28           --health-interval=10s --health-timeout=5s --health-retries=5
29
30     steps:
31       - name: Checkout code
32         uses: actions/checkout@v4
33
34       - name: Setup PHP
35         uses: shivammathur/setup-php@v2
36         with:
37           php-version: "8.2"
38           extensions: mysqli
39
40       - name: Install Dependencies
41         run: composer install
42
43       # — PASO DE AUTO-FORMATEO —
44       - name: Auto-fix code style issues
45         run: vendor/bin/phpcbf
46         continue-on-error: true
47
48       - name: Commit automatically fixed code
49         uses: stefanzweifel/git-auto-commit-action@v5
50         with:
51           commit_message: "Chore: Apply automated code style fixes"
52
53       # — PASOS DE VERIFICACIÓN sss —
54       - name: Run Linter (PHP Code Sniffer)
55         run: vendor/bin/phpcs
56
57       - name: Run Unit Tests (PHPUnit)
58         run: vendor/bin/phpunit
```

## Automatización:

```
1 on:
2   push:
3     branches:
4       - main
5   pull_request:
6     branches:
7       - main
```

La automatización principal reside aquí. Se ha configurado el pipeline para que se ejecute automáticamente cada vez que alguien:

-Hace un push (sube cambios) a la rama main.

-Crea o actualiza un pull request que apunta a la rama main.

## Build

```
1 jobs:
2   build:
3     runs-on: ubuntu-latest
4
5     services:
6       mysql:
7         image: mysql:8.0
8         env:
9           MYSQL_ROOT_PASSWORD: root
10          MYSQL_DATABASE: control_asistencia
11        ports:
12          - 3306:3306
13        options: --health-cmd="mysqladmin ping -h localhost"
14                --health-interval=10s --health-timeout=5s --health-retries=5
15
16      steps:
17        - name: Checkout code
18          uses: actions/checkout@v4
19
20        - name: Setup PHP
21          uses: shivammathur/setup-php@v2
22          with:
23            php-version: "8.2"
24            extensions: mysqli
25
26        - name: Install Dependencies
27          run: composer install
```

Estos son pasos necesarios para preparar el proyecto y su entorno para las pruebas.

## Lint

El linter es la herramienta que se encarga de revisar que tu código cumpla con las reglas de estilo y formato definidas para el proyecto.

```
1 - name: Run Linter (PHP Code Sniffer)
2   run: vendor/bin/phpcs
```

## Pruebas

Aquí es donde verificas que la lógica de tu aplicación funciona como se espera.

```
1 - name: Run Unit Tests (PHPUnit)
2   run: vendor/bin/phpunit
```

## Despliegue continuo

El despliegue se hace en la plataforma heroku.

Heroku es una Plataforma como Servicio (PaaS) que permite a los desarrolladores construir, desplegar, monitorear y escalar aplicaciones web y APIs de una manera sencilla y rápida.

```
1  deploy:
2    runs-on: ubuntu-latest
3    # Se ejecuta solo si el job 'build' fue exitoso
4    needs: build
5    steps:
6      - name: Checkout code
7        uses: actions/checkout@v4
8        with:
9          fetch-depth: 0
10     # Añadimos un paso para instalar la Heroku CLI manualmente
11     - name: Install Heroku CLI
12       run: curl https://cli-assets.heroku.com/install.sh | sh
13
14     - name: Deploy to Heroku
15       env:
16         HEROKU_API_KEY: ${ secrets.HEROKU_API_KEY }
17         HEROKU_APP_NAME: ${ secrets.HEROKU_APP_NAME }
18         HEROKU_EMAIL: ${ secrets.HEROKU_EMAIL }
19       run: |
20         # 1. Crea el archivo .netrc para la autenticación de Git
21         cat > ~/.netrc << EOF
22         machine api.heroku.com
23           login $HEROKU_EMAIL
24           password $HEROKU_API_KEY
25         machine git.heroku.com
26           login $HEROKU_EMAIL
27           password $HEROKU_API_KEY
28         EOF
29
30         # 2. Añade el repositorio de Heroku como un "remoto" de Git
31         heroku git:remote -a $HEROKU_APP_NAME
32
33         # 3. Sube el código de la rama actual a la rama 'main' de Heroku
34         git push heroku HEAD:main --force
```

## Workflows

Aquí podemos ver los procesos automatizados que se realizan al hacer pull request en la rama main

All workflows				Filter workflow runs
Showing runs from all workflows				
45 workflow runs				Event ▾ Status ▾ Branch ▾ Actor ▾
✓ <b>correccion deployment</b>	main	20 minutes ago	1m 13s	...
PHP CI Pipeline #44: Commit <a href="#">e7eb911</a> pushed by <a href="#">OmarTellez1</a>				
✗ <b>correccion heroku</b>	main	52 minutes ago	57s	...
PHP CI Pipeline #40: Commit <a href="#">74d63ba</a> pushed by <a href="#">OmarTellez1</a>				
✗ <b>correccion</b>	main	1 hour ago	49s	...
PHP CI Pipeline #39: Commit <a href="#">4c21e5f</a> pushed by <a href="#">OmarTellez1</a>				
✓ <b>Merge branch 'main' of https://github.com/OmarTellez1/ProyectoSystemD...</b>	main	Today at 2:57 PM	54s	...
PHP CI Pipeline #38: Commit <a href="#">a254907</a> pushed by <a href="#">OmarTellez1</a>				
✓ <b>Merge branch 'main' of https://github.com/OmarTellez1/ProyectoSystemD...</b>	main	Today at 2:46 PM	49s	...
PHP CI Pipeline #37: Commit <a href="#">f61cf95</a> pushed by <a href="#">OmarTellez1</a>				
✓ <b>Correccion error salir sesion</b>	main	Today at 2:21 PM	46s	...
PHP CI Pipeline #36: Commit <a href="#">756da5b</a> pushed by <a href="#">OmarTellez1</a>				
✓ <b>Stop tracking .env file</b>	main	Today at 12:04 PM	47s	...
PHP CI Pipeline #35: Commit <a href="#">24aa539</a> pushed by <a href="#">OmarTellez1</a>				
✓ <b>cambios en el gitignore</b>	main	Today at 12:00 PM	54s	...
PHP CI Pipeline #34: Commit <a href="#">5117957</a> pushed by <a href="#">OmarTellez1</a>				

## Estado actual del codigo

Login:

## Primero Tu

Ingrese sus datos de Acceso

Ingresar

## Perfil



## Introducción

En esta segunda entrega, hemos logrado avances significativos en la profesionalización de nuestro flujo de trabajo. El principal objetivo ha sido establecer un sistema robusto de control de versiones con Git y configurar un pipeline de Integración y Despliegue Continuo (CI/CD) utilizando GitHub Actions. El progreso se centra en la automatización de la construcción del proyecto, la ejecución de pruebas y la verificación de la calidad del código, sentando las bases para un desarrollo colaborativo más eficiente y seguro.

## Conclusión

A lo largo de esta entrega, hemos aprendido la importancia crítica de automatizar los procesos de desarrollo. La implementación del pipeline de CI/CD nos ha permitido detectar errores de manera temprana y asegurar un estándar de calidad constante en nuestro código.

### Enlace a repositorio de GitHub:

<https://github.com/OmarTellez1/ProyectoSystemDesigns>

### Enlace a demo desplegada:

<https://mi-app-asistencia-aa9b764d7d3a.herokuapp.com/vistas/escritorio.php>