



INGENIERÍA EN DESARROLLO Y GESTIÓN DE SOFTWARE MULTIPLATAFORMA

Integradora

Evidencia DevOps - Rama ci_devops

Gestor DevOps (Gestión del Proceso)

Erick Salvador Ramírez Cárdenas Matrícula: 21855

Néstor Sebastián Ruiz Pérez Matrícula: 15267

Sergio Arath Flores Garza Matricula: 20251

Eugenio Espronceda Cavazos Matricula: 21848

Axel Gerardo Henríquez Carrizales Matricula: 21849

Alan Gabriel Puente Gutiérrez Matricula: 21854

William Martínez Lopez Matricula: 21851

SANTA CATARINA, N.L., 19 de septiembre 2025

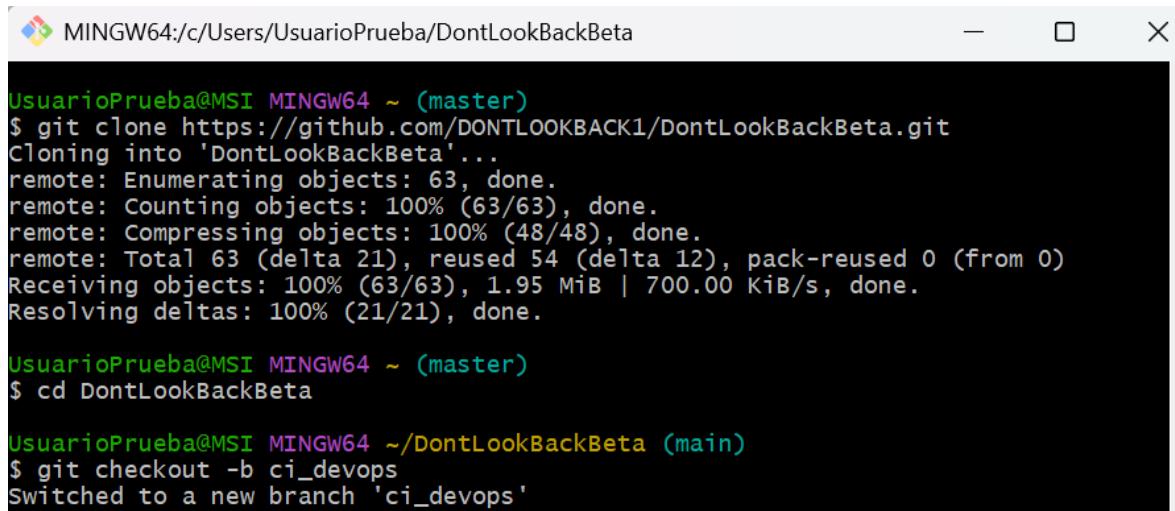
Introducción

El presente documento recoge las actividades realizadas como Gestor DevOps en el proyecto *DontLookBack*. Durante el proceso se habilitó el tablero ágil con plantillas para issues, pull requests, bugs y historias de usuario, lo que permitió organizar y controlar de manera efectiva las tareas del equipo. Se definió un flujo de trabajo Git claro, estableciendo las ramas principales (main, develop) y las ramas de funcionalidad (feature/*), así como la política de pull requests y convenciones de commits. Adicionalmente, se implementó un borrador de pipeline CI/CD con etapas de build y pruebas unitarias como placeholders, se configuró Git LFS para los recursos pesados y se estableció CODEOWNERS, asegurando la trazabilidad y control de los cambios en el repositorio.

Paso a Paso de la Configuración DevOps

Paso 1: Creación de la rama ci_devops

Se clonó el repositorio del equipo y se creó una rama independiente para trabajar en DevOps sin afectar main ni develop.



```
MINGW64:/c/Users/UsuarioPrueba/DontLookBackBeta
UsuarioPrueba@MSI MINGW64 ~ (master)
$ git clone https://github.com/DONTLOOKBACK1/DontLookBackBeta.git
Cloning into 'DontLookBackBeta'...
remote: Enumerating objects: 63, done.
remote: Counting objects: 100% (63/63), done.
remote: Compressing objects: 100% (48/48), done.
remote: Total 63 (delta 21), reused 54 (delta 12), pack-reused 0 (from 0)
Receiving objects: 100% (63/63), 1.95 MiB | 700.00 KiB/s, done.
Resolving deltas: 100% (21/21), done.

UsuarioPrueba@MSI MINGW64 ~ (master)
$ cd DontLookBackBeta

UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta (main)
$ git checkout -b ci_devops
Switched to a new branch 'ci_devops'
```

Captura de Terminal mostrando que se creó la rama ci_devops.

Paso 2: Crear estructura de carpetas base

Se organizaron las carpetas necesarias para código, tests, assets y documentación:

```
UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta (ci_devops)
$ mkdir -p docs src assets tests .github/workflows .github/ISSUE_TEMPLATE

UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta (ci_devops)
$ touch src/.gitkeep tests/.gitkeep assets/.gitkeep

UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta (ci_devops)
$ git add docs src assets tests .github/workflows .github/ISSUE_TEMPLATE

UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta (ci_devops)
$ git commit -m "chore: crear estructura de carpetas base - 2025-09-21 Nestor Ruiz"
[ci_devops 39fc537] chore: crear estructura de carpetas base - 2025-09-21 Nestor Ruiz
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 assets/.gitkeep
create mode 100644 src/.gitkeep
create mode 100644 tests/.gitkeep
```

Descripción:

- /docs → documentación y estrategia Git & CI
- /src → código principal de Godot 2D
- /assets → recursos del juego (usar Git LFS si son pesados)
- /tests → pruebas unitarias placeholder
- .github/workflows → pipelines CI/CD
- .github/ISSUE_TEMPLATE → plantillas para bugs y features

Captura de Terminal mostrando commit de la creación de carpetas.

Paso 3: Configuración de Git LFS

Para manejar archivos grandes en assets/:

```
UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta (ci_devops)
$ git lfs install
Updated Git hooks.
Git LFS initialized.

UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta (ci_devops)
$ git lfs track "assets/*"
Tracking "assets/*"
Pattern 'assets/*' matches forbidden file 'assets/.gitkeep'. If you would like t
o track assets/.gitkeep, modify '.gitattributes' manually.

UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta (ci_devops)
$ git add .gitattributes
warning: in the working copy of '.gitattributes', CRLF will be replaced by LF th
e next time Git touches it

UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta (ci_devops)
$ git commit -m "chore: activar Git LFS para assets - 2025-09-21 Nestor Ruiz"
On branch ci_devops
nothing to commit, working tree clean
```

Captura de Terminal mostrando configuración de Git LFS.

Paso 4: Crear pipeline CI/CD placeholder

```
UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta (ci_devops)
$ cat <<EOL > .github/workflows/ci.yml
name: CI

on:
  push:
    branches: [main, develop, ci_devops]
  pull_request:
    branches: [main, develop]

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: Build placeholder
        run: echo "☒ Build ejecutado"

  test:
    runs-on: ubuntu-latest
    needs: build
    steps:
      - uses: actions/checkout@v3
      - name: Test placeholder
        run: echo "❖ Test ejecutado"
EOL

UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta (ci_devops)
$ git add .github/workflows/ci.yml

UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta (ci_devops)
$ git commit -m "ci: agregar pipeline placeholder build y test - 2025-09-21 Nest
or Ruiz"
[ci_devops b8df84a] ci: agregar pipeline placeholder build y test - 2025-09-21 N
estor Ruiz
 1 file changed, 23 insertions(+)
 create mode 100644 .github/workflows/ci.yml
```

Captura de Terminal mostrando commit del pipeline.

Paso 5: Plantillas y CODEOWNERS

Se crearon plantillas para issues y se asignó responsabilidad en CODEOWNERS:

Archivo: .github/ISSUE_TEMPLATE/bug_report.md y feature_request.md

Archivo: .github/CODEOWNERS

```
UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta (ci_devops)
$ cat <<EOL > .github/ISSUE_TEMPLATE/bug_report.md
---
name: Reporte de Bug
about: Reportar un bug en la aplicación
---

**Descripción del bug**
---

**Pasos para reproducir**
---

**Comportamiento esperado**
---

**Capturas**
---
EOL

UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta (ci_devops)
$ cat <<EOL > .github/ISSUE_TEMPLATE/feature_request.md
---
name: Solicitud de Feature
about: Proponer nueva funcionalidad
---

**Descripción de la funcionalidad**
---

**Motivación**
---

**Notas adicionales**
---
EOL
```

```
UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta (ci_devops)
$ cat <<EOL > .github/CODEOWNERS
# Definir responsables de revisión
/docs/ @NestorRuiz
/src/ @NestorRuiz
/assets/ @NestorRuiz
/tests/ @NestorRuiz
.github/ @NestorRuiz
EOL

UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta (ci_devops)
$ git add .github/ISSUE_TEMPLATE .github/CODEOWNERS

UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta (ci_devops)
$ git commit -m "chore: agregar plantillas y CODEOWNERS con auditoría - 2025-09-21 Nestor Ruiz"
[ci_devops 63ed67f] chore: agregar plantillas y CODEOWNERS con auditoría - 2025-09-21 Nestor Ruiz
 3 files changed, 35 insertions(+)
 create mode 100644 .github/CODEOWNERS
 create mode 100644 .github/ISSUE_TEMPLATE/bug_report.md
 create mode 100644 .github/ISSUE_TEMPLATE/feature_request.md
```

Capturas de Terminal mostrando commit de plantillas y CODEOWNERS.

Nuevo archivo agregado

.github/PULL_REQUEST_TEMPLATE.md

Este template asegura que cada pull request incluya descripción de cambios, tipo de modificación, checklist de revisión y evidencia opcional.

```
UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta/.github (ci_devops)
$ nano PULL_REQUEST_TEMPLATE.md

UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta/.github (ci_devops)
$ git add PULL_REQUEST_TEMPLATE.md

UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta/.github (ci_devops)
$ git commit -m "chore: agregar template de Pull Request - 2025-09-21 Nestor Ruiz"
[ci_devops 11a8d84] chore: agregar template de Pull Request - 2025-09-21 Nestor Ruiz
 1 file changed, 22 insertions(+)
 create mode 100644 .github/PULL_REQUEST_TEMPLATE.md

UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta/.github (ci_devops)
$ git push origin ci_devops
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 957 bytes | 957.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/DONTLOOKBACK1/DontLookBackBeta.git
 2bde0ae..11a8d84 ci_devops -> ci_devops
```

Captura de Terminal mostrando el Pull Request template

Paso 6: Documentar estrategia Git & CI

Archivo: docs/Estrategia_Git_CI_v1.md

Flujo de ramas (main, develop, feature/*, ci_devops)

Convenciones de commit (feat, fix, chore, docs, ci)

Pipeline placeholder

Uso de Git LFS

```
UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta (ci_devops)
$ git add docs/Estrategia_Git_CI_v1.md

UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta (ci_devops)
$ git commit -m "docs: agregar estrategia Git & CI con comentarios de auditoría
- 2025-09-21 Nestor Ruiz"
[ci_devops 2bde0ae] docs: agregar estrategia Git & CI con comentarios de auditoría
- 2025-09-21 Nestor Ruiz
 1 file changed, 24 insertions(+)
 create mode 100644 docs/Estrategia_Git_CI_v1.md
```

Captura de Terminal mostrando commit del documento de estrategia.

Paso 7: Subir rama ci_devops al repositorio

```
UsuarioPrueba@MSI MINGW64 ~/DontLookBackBeta (ci_devops)
$ git push origin ci_devops
Enumerating objects: 26, done.
Counting objects: 100% (26/26), done.
Delta compression using up to 16 threads
Compressing objects: 100% (18/18), done.
Writing objects: 100% (21/21), 2.67 KiB | 683.00 KiB/s, done.
Total 21 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), completed with 2 local objects.
remote:
remote: Create a pull request for 'ci_devops' on GitHub by visiting:
remote:     https://github.com/DONTLOOKBACK1/DontLookBackBeta/pull/new/ci_devop
s
remote:
To https://github.com/DONTLOOKBACK1/DontLookBackBeta.git
 * [new branch]      ci_devops -> ci_devops
```

Captura de Terminal mostrando push exitoso y enlace para pull request.

Paso 8: Monitoreo de cambios del equipo

Aunque tu trabajo está aislado, puedes revisar commits en main o develop

The screenshot shows a GitHub commit history for a repository. At the top, it displays "4 commits", "8 files changed", and "1 contributor". Below this, a section titled "Commits on Sep 21, 2025" lists four commits:

- chore: crear estructura de carpetas base - 2025-09-21 Nestor Ruiz**
Nestor0425 committed 43 minutes ago
- ci: agregar pipeline placeholder build y test - 2025-09-21 Nestor Ruiz**
Nestor0425 committed 41 minutes ago
- chore: agregar plantillas y CODEOWNERS con auditoria - 2025-09-21 Nes...**
Nestor0425 committed 40 minutes ago
- docs: agregar estrategia Git & CI con comentarios de auditoria - 2025...**
Nestor0425 committed 39 minutes ago

At the bottom of the commit list, it says "Showing 8 changed files with 82 additions and 0 deletions." There are "Split" and "Unified" buttons for viewing the changes.

Dar seguimiento a los cambios recientes: en la captura se observan 4 commits consecutivos con diferentes tipos (chore, ci, docs), lo que refleja la organización del trabajo en tareas de estructura, automatización y documentación.

Identificar responsabilidades: cada commit muestra el autor, el mensaje descriptivo y la marca de tiempo, lo que facilita saber quién agregó qué y en qué momento.

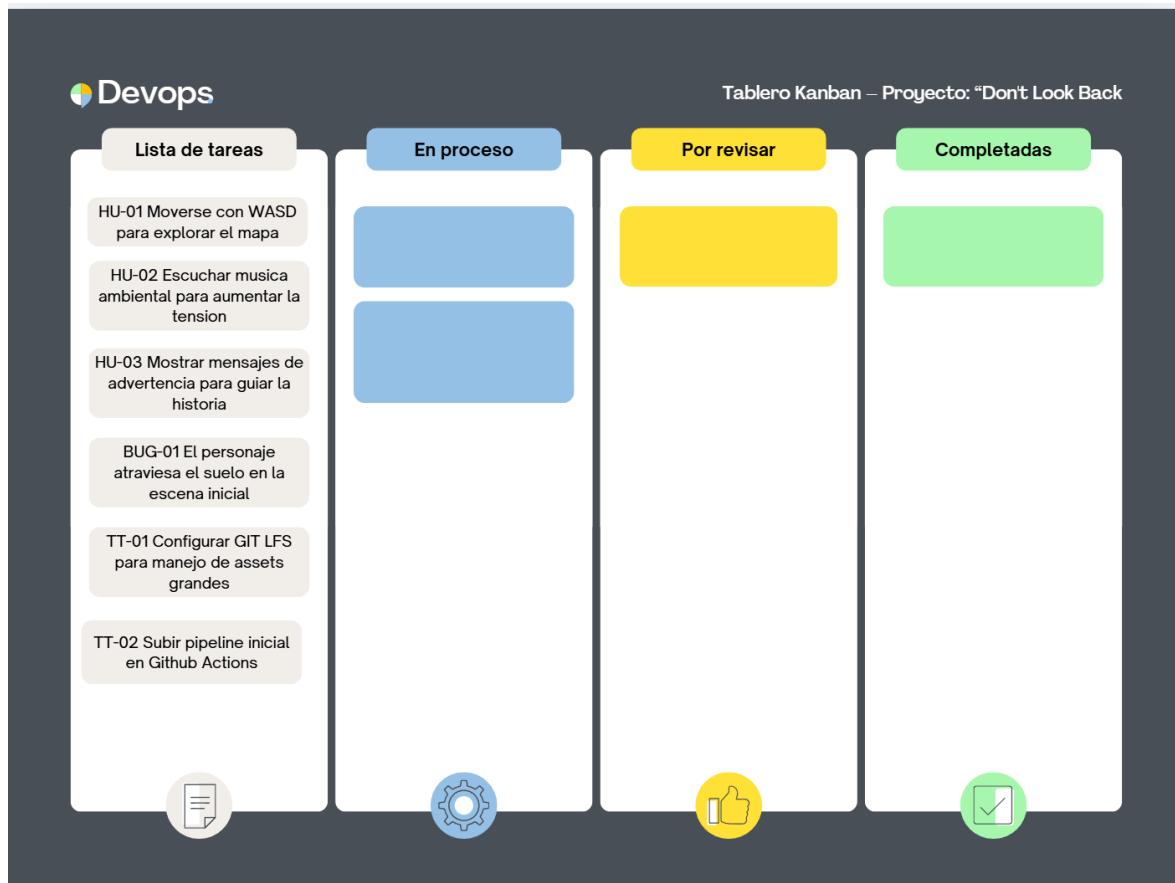
Controlar impacto de cambios: la sección "8 changed files with 82 additions and 0 deletions" ayuda a dimensionar la magnitud de los ajustes en el código.

Asegurar calidad y auditoría: al revisar mensajes de commit claros y estandarizados (como chore:, ci:, docs:), se mantiene la trazabilidad y la auditoría del proyecto.

Evidencia – Tablero Kanban

El tablero Kanban permitió organizar las tareas del proyecto Dont Look Back en columnas de Backlog, In Progress y Done, facilitando la visualización del avance, la priorización de tareas y la asignación de responsabilidades dentro del equipo. Esta herramienta fue clave para aplicar la metodología ágil y dar seguimiento al progreso de manera colaborativa.

📸 Captura del tablero del Videojuego Dont Look Back



https://www.canva.com/design/DAGzejxuGsE/OJH15Df5ONC_klkb8ayLRQ/view?utm_content=DAGzejxuGsE&utm_campaign=designshare&utm_medium=link2&utm_source=uniquelinks&utllId=h9ffb355e7c

Conclusión

Como Gestor DevOps, logré establecer la base de la infraestructura de colaboración y automatización para el equipo. Se definió un flujo de trabajo Git estandarizado, se implementaron plantillas para issues y pull requests, se configuró Git LFS para manejar recursos pesados, se agregó un pipeline CI/CD inicial para asegurar calidad de código, y se incorporó CODEOWNERS para dar trazabilidad y responsabilidad en los cambios.

El monitoreo de commits demuestra que el proyecto cuenta ahora con un marco de trabajo organizado, transparente y auditabile, lo que facilitará la integración continua, reducirá riesgos de errores y permitirá al equipo enfocarse en el desarrollo del producto con confianza en los procesos establecidos.