Documentación laravel base + gitflow

Rehacer todo el proyecto con GitFlow y Laravel

Primero tener instalado:

- **XAMPP** (para PHP y MySQL) → Descargar aquí
- **Composer** (para dependencias PHP) → Descargar aquí
- **Node.js + npm** (para dependencias front) → Descargar aquí

Nosotros ya teníamos Xampp y composer instalados, solo nos quedaba Node.js + npm entonces instalamos desde ese link la versión sin docker que decia .msi, luego lo ejecutamos y instalamos todo por defecto

Verificación después de instalar node.js

Abrimos una terminal cmd y escribimos ***node -v*** , ***npm -v*** y esto nos debería mostrar la versión de cada uno de ellos y ya significa que está instalado :

```
C:\Users\pc>node -v
v22.18.0

C:\Users\pc>npm -v
10.9.3
```

luego creamos un proyecto laravel de cero con composer pero tambien se puede usando el comando **laravel new proyecto2025**:

```
C:\Users\pc>cd C:\xampp\htdocs
```

```
C:\xampp\htdocs>composer create-project laravel/laravel proyecto2025
Creating a "laravel/laravel" project at "./proyecto2025"
Installing laravel/laravel (v12.3.1)
  - Downloading laravel/laravel (v12.3.1)
  - Installing laravel/laravel (v12.3.1): Extracting archive
Created project in C:\xampp\htdocs\proyecto2025
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 112 installs, 0 updates, 0 removals
  - Locking brick/math (0.13.1)
  - Locking carbonphp/carbon-doctrine-types (3.2.0)
  - Locking dflydev/dot-access-data (v3.0.3)
  - Locking doctrine/inflector (2.1.0)
  - Locking doctrine/lexer (3.0.1)
  - Locking dragonmantank/cron-expression (v3.4.0)
  - Locking egulias/email-validator (4.0.4)
  - Locking fakerphp/faker (v1.24.1)
  - Locking filp/whoops (2.18.4)
  - Locking fruitcake/php-cors (v1.3.0)
  - Locking graham-campbell/result-type (v1.1.3)
  - Locking guzzlehttp/guzzle (7.10.0)
  - Locking guzzlehttp/promises (2.3.0)
```

luego entramos en la carpeta que acabamos de crear o sea proyecto 2025 y instalamos npm

```
C:\xampp\htdocs>cd proyecto2025

C:\xampp\htdocs\proyecto2025>npm install
```

```
added 89 packages, and audited 90 packages in 24s

22 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

- `added 89 packages`: se descargaron e instalaron los paquetes de npm (Vite, Tailwind, Alpine, etc., según el stack de Laravel).
- `audited 90 packages`: npm revisó si había vulnerabilidades de seguridad.
- `found 0 vulnerabilities`: no hay problemas detectados.

```
C:\xampp\htdocs\proyecto2025>npm run dev

> dev
> vite


  VITE v7.1.3  ready in 555 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help

  LARAVEL v12.25.0  plugin v2.0.0

  ➜  APP_URL: http://localhost
```

- Esto levanta **Vite** (la herramienta que usa Laravel para compilar CSS y JS).
- Mientras tengas esta consola abierta, cada vez que cambies un archivo `.js` o `.css` se actualiza solo.

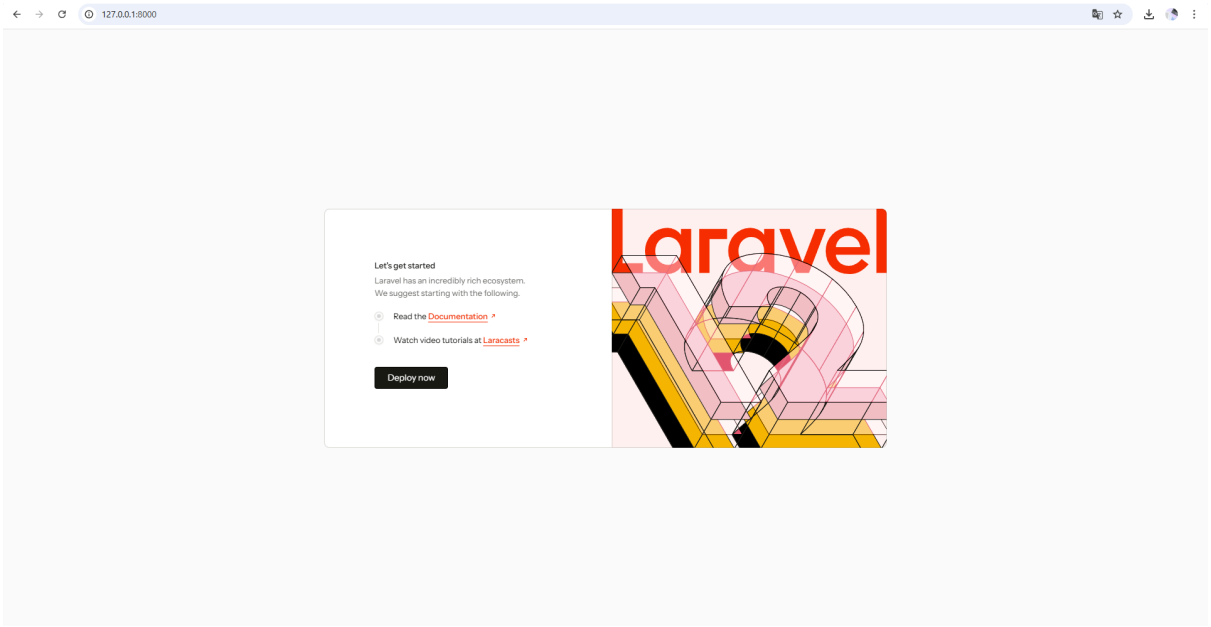
```
C:\xampp\htdocs\proyecto2025>php artisan serve

  INFO  Server running on [http://127.0.0.1:8000].

  Press Ctrl+C to stop the server

  2025-08-25 16:14:22 / ..... ~ 511.43ms
  2025-08-25 16:14:22 /favicon.ico ..... ~ 500.78ms
```

levantamos el servidor laravel



esto significa que está funcionando

`master`

```
PS C:\Users\pc\playcode> cd C:\xampp\htdocs
PS C:\xampp\htdocs> cd proyecto2025
PS C:\xampp\htdocs\proyecto2025> git init
Initialized empty Git repository in C:/xampp/htdocs/proyecto2025/.git/
PS C:\xampp\htdocs\proyecto2025> git remote add origin https://github.com/playcodecompany/Segunda_Entrega1.git
PS C:\xampp\htdocs\proyecto2025> git branch -M master
```

en la terminal de VSCode vamos al nuevo proyecto creado con cd y creamos un nuevo repo con git init, y luego conectamos el proyecto local al repo remoto

Crear `develop` y features

Crear la rama `develop` desde master:

```
PS C:\xampp\htdocs\proyecto2025> git checkout -b develop
Switched to a new branch 'develop'
PS C:\xampp\htdocs\proyecto2025> git push -u origin develop
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:      https://github.com/playcodecompany/Segunda_Entrega1/pull/new/develop
remote:
To https://github.com/playcodecompany/Segunda_Entrega1.git
 * [new branch]      develop -> develop
branch 'develop' set up to track 'origin/develop'.
```

con esto ya creamos develop desde master - Ahora a crear las features

Verificamos estar en develop:

```
PS C:\xampp\htdocs\proyecto2025> git checkout develop
Already on 'develop'
Your branch is up to date with 'origin/develop'.
```

```
PS C:\xampp\htdocs\proyecto2025> git checkout -b feature/base-project
Switched to a new branch 'feature/base-project'
PS C:\xampp\htdocs\proyecto2025> git add .
PS C:\xampp\htdocs\proyecto2025> git commit -m "Base del proyecto Laravel lista"
[feature/base-project 350ff78] Base del proyecto Laravel lista
 56 files changed, 13041 insertions(+)
 create mode 100644 .editorconfig
 create mode 100644 .env.example
 create mode 100644 .gitattributes
 create mode 100644 .gitignore
 create mode 100644 app/Http/Controllers/Controller.php
 create mode 100644 app/Models/User.php
 create mode 100644 app/Providers/AppServiceProvider.php
 create mode 100644 artisan
 create mode 100644 bootstrap/app.php
 create mode 100644 bootstrap/cache/.gitignore
 create mode 100644 bootstrap/providers.php
 create mode 100644 composer.json
 create mode 100644 composer.lock
 create mode 100644 config/app.php
 create mode 100644 config/auth.php
 create mode 100644 config/cache.php
 create mode 100644 config/database.php
 create mode 100644 config/filesystems.php
 create mode 100644 config/logging.php
 create mode 100644 config/mail.php
 create mode 100644 config/queue.php
```

y con eso creamos la feature/base-project dentro de develop y añadimos todo el laravel que ya teniamos instalado del paso que hicimos al principio

```
PS C:\xampp\htdocs\proyecto2025> git checkout develop
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
PS C:\xampp\htdocs\proyecto2025> git merge feature/base-project
Updating ab625b8..350ff78
Fast-forward
 .editorconfig                                       |   18 +
 .env.example                                        |   65 +
 .gitattributes                                      |   11 +
 .gitignore                                          |   24 +
 app/Http/Controllers/Controller.php                 |    8 +
 app/Models/User.php                                 |   48 +
 app/Providers/AppServiceProvider.php                |   24 +
 artisan                                             |   18 +
 bootstrap/app.php                                   |   18 +
 bootstrap/cache/.gitignore                          |    2 +
 bootstrap/providers.php                             |    5 +
 composer.json                                       |   75 +
 composer.lock                                       | 8383 ++++++++++++++++++++
 config/app.php                                      |  126 +
 config/auth.php                                     |  115 +
 config/cache.php                                    |  108 +
 config/database.php                                 |  174 +
 config/filesystems.php                              |   80 +
 config/logging.php                                  |  132 +
 config/mail.php                                     |  118 +
 config/queue.php                                    |  112 +
 config/services.php                                 |   38 +
 config/session.php                                  |  217 +
 database/.gitignore                                 |    1 +
 database/factories/UserFactory.php                  |   44 +
 .../0001_01_01_000000_create_users_table.php        |   49 +
 .../0001_01_01_000001_create_cache_table.php        |   35 +
 .../0001_01_01_000002_create_jobs_table.php         |   57 +
 database/seeders/DatabaseSeeder.php                 |   23 +
```

```
PS C:\xampp\htdocs\proyecto2025> git push origin develop
Enumerating objects: 77, done.
Counting objects: 100% (77/77), done.
Delta compression using up to 12 threads
Compressing objects: 100% (59/59), done.
Writing objects: 100% (76/76), 87.35 KiB | 4.85 MiB/s, done.
Total 76 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/playcodecompany/Segunda_Entrega1.git
   ab625b8..350ff78  develop -> develop
```

aca verificamos denuevo estar en develop y mergeamos lo de la feature base a develop

Ahora `develop` contiene el proyecto base y ya está en GitHub.

```
PS C:\xampp\htdocs\proyecto2025> git checkout develop
Already on 'develop'
Your branch is up to date with 'origin/develop'.
PS C:\xampp\htdocs\proyecto2025> git checkout -b feature/login
Switched to a new branch 'feature/login'
PS C:\xampp\htdocs\proyecto2025> git checkout -b feature/index
Switched to a new branch 'feature/index'
PS C:\xampp\htdocs\proyecto2025> git checkout -b feature/play
Switched to a new branch 'feature/play'
PS C:\xampp\htdocs\proyecto2025> git checkout -b feature/panel
Switched to a new branch 'feature/panel'
PS C:\xampp\htdocs\proyecto2025> git checkout -b feature/match
Switched to a new branch 'feature/match'
PS C:\xampp\htdocs\proyecto2025> git checkout -b feature/register
Switched to a new branch 'feature/register'
```

verificamos estar en develop denuevo y creamos las features para cada view

Conclusión

Primero, preparamos el entorno de desarrollo: verificamos tener XAMPP instalado para PHP y MySQL, Composer para gestionar dependencias de PHP, y Node.js con npm para el front-end. También teníamos instalado Laravel y dejamos el proyecto base (`proyecto2025`) listo, con todas las dependencias de frontend instaladas y compiladas, y el servidor local funcionando para probar la aplicación.

A continuación, inicializamos el repositorio Git con git init y lo conectamos con GitHub (`Segunda_Entrega1`). Creamos y subimos la rama `master` con el commit inicial del README, y después creamos la rama **`develop`**, que es la rama de desarrollo donde se integrarán todas las funcionalidades. Esto establece la estructura base de GitFlow.

En GitFlow, se sigue un flujo de trabajo organizado: la rama `master` contiene las versiones finales y estables del proyecto; la rama `develop` integra todas las features desarrolladas; y cada feature se trabaja en su propia rama independiente (`feature/*`). Cada feature branch se crea a partir de `develop`, se desarrollan los cambios (vistas, CSS, JS, etc.), se hacen commits y se sube al remoto antes de hacer merge nuevamente a `develop`. Luego se puede borrar la rama local de feature para mantener el repositorio limpio.

En nuestro proyecto, cada vista o funcionalidad tiene su propia rama feature: `feature/login` para la vista iniciosesion, `feature/index` para el index/página principal, `feature/play` para la vista jugar, `feature/panel` para panel, `feature/match` para partida y `feature/register` para registro. Cada rama incluye los archivos necesarios para que la vista funcione correctamente, como los archivos `.blade.php` y CSS/JS relacionados.

Este flujo tiene varias ventajas: los cambios están aislados, evitando que se rompa el proyecto; permite colaboración simultánea, ya que varios pueden trabajar en distintas features; mantiene `develop` siempre funcional; y asegura que `master` permanezca limpio y estable para futuras versiones finales.

Debemos crear las ramas feature para las demás vistas o funcionalidades restantes, hacer commits regularmente, probar y luego mergear a `develop`. Una vez que todas las features estén completas y estables, se hace el merge final a `master` para dejar la versión de entrega lista.

Cambiarte a la feature branch

- Si la rama ya existe: `git checkout feature/nombre-de-la-feature`
- Si todavía no existe, creala desde `develop`:
    - `git checkout develop`
    - `git checkout -b feature/nombre-de-la-feature`
- Esto asegura que trabajes aislada del resto del proyecto.
- Todos los cambios que hagas aquí no afectarán `develop` ni `master` hasta que hagas merge.

Trabajar en la feature

- Hacés los cambios en los archivos de Laravel: vistas (`.blade.php`), controladores, CSS, JS, rutas, etc.
- Probá localmente: `php artisan serve`
- Comprobá que todo funciona antes de guardar los cambios en Git.

Guardar los cambios (commit)

- Primero agregás los archivos modificados: `git add .`
- Luego hacés el commit con un mensaje descriptivo:
    - `git commit -m "Descripción clara de lo que hiciste"`

`git add .` → marca todos los archivos modificados para incluirlos en el commit.
`git commit -m` → crea un snapshot de tu trabajo con un mensaje que explique los cambios.

Subir la feature al remoto

```
git push -u origin feature/nombre-de-la-feature
```

- Esto sube tu feature branch a GitHub.
- `-u` hace que tu rama local quede vinculada con la rama remota, así en futuros pushes solo necesitás `git push`.

Mergear la feature a `develop`

Cuando la feature está completa y probada:

- `git checkout develop`
- `git merge feature/nombre-de-la-feature`
- `git push origin develop`

Ahora `develop` tiene tu feature integrada y lista para usar con las demás.

Limpiar la rama feature (opcional)

Si ya mergeaste tu feature y no la necesitás más localmente:

- `git branch -d feature/nombre-de-la-feature`

Mantiene el repositorio limpio.
La rama remota sigue en GitHub si alguien más la necesita.

- Los archivos ya están en la rama `develop` gracias al merge, así que no se pierde ningún trabajo.
- Solo desaparece el nombre de la rama en tu repositorio local.
- Limpiar = borrar la rama local de feature.
- No borra los commits ni archivos que ya están en `develop`.
- Mantiene tu repo más ordenado y evita tener muchas ramas locales innecesarias.