# **A) Repo por fila + trabajo por ramas (todo en GitHub)**

## **1) Crear el repositorio de la fila (una persona por fila)**

1. Entra en **GitHub → + → New repository**.
2. Nombre: gf-fila-01 (o similar para cada fila).
3. Marca **Add a README file** (opcional) y **Create repository**.

## **2) Añadir el index.html básico**

1. Dentro del repo: **Add file → Create new file**.
2. Nombre del archivo: index.html.

Pega este mínimo:  
  
 <!doctype html>

<html lang="es">

<head>

<meta charset="utf-8" />

<title>Galaxy Defenders</title>

</head>

<body>

<h1>Hola, Galaxy Defenders</h1>

<p>Página inicial.</p>

</body>

</html>

1. Abajo, **Commit changes** (puedes dejar el mensaje por defecto).  
    Asegúrate de que va a la rama main.

## **3) Invitar al compañero de la fila como colaborador**

1. **Settings → Collaborators** (o **Settings → Manage access**).
2. **Add people** → escribe su usuario → **Add**.
3. (El compañero debe **aceptar** la invitación que le llegará).
4. Por favor, añadirme a mi como colaboradora para verlo MariaCampusFp.

## **4) Cada miembro crea su propia rama desde la web**

1. Menú de ramas (caja que dice main en la vista **Code**).
2. Escribe el nombre de tu rama, por ejemplo: feat-menu o feat-jugador.
3. Pulsa **Create branch: feat-menu from ‘main’**.  
    (Repite el proceso tu compañero con su propia rama).

## **5) Editar index.html en tu rama (web editor)**

1. Asegúrate de estar en **tu** rama (no en main).
2. Abre index.html → botón del **lápiz** (Edit this file).
3. Haz un cambio pequeño (un párrafo, un style, etc.).
4. Abajo, **Commit changes**.  
   * Si GitHub te ofrece “Create a new branch for this commit”, elige tu rama (no crees otra).

## **6) Abrir un Pull Request (PR) hacia main**

1. Pestaña **Pull requests → New pull request**.
2. Base: main | Compare: **tu rama** (feat-menu, por ejemplo).
3. **Create pull request**.
4. Pon un título claro (ej: feat(menu): añade sección de inicio).
5. **Assign** un revisor (tu compañero).
6. **Create pull request**.

## **7) Revisar y aprobar el PR (el compañero)**

1. Entra al PR → pestaña **Files changed** para ver cambios.
2. Deja **comentarios** si hace falta.
3. Si está correcto: **Review changes → Approve → Submit review**.
4. El autor (o quien tenga permisos) hace **Merge pull request → Confirm**.
5. Tras el merge, en **Code**, cambia a main y verás el cambio integrado.

Repite 4–7 con la rama del otro miembro para que **ambos** trabajen por ramas y hagan PRs.

# **B) “Clonar” el repo del compañero usando Fork (sin consola)**

Aquí una de las personas “clona” el repo del compañero **desde GitHub** con **Fork** (copia en tu cuenta) y propone cambios mediante **Pull Request** hacia el repo original.

## **8) Hacer un Fork del repositorio del compañero**

1. Entra en el repo del compañero (por ejemplo: gf-fila-02).
2. Arriba a la derecha: **Fork** → **Create fork**.  
    (Ahora tienes una copia en tu cuenta: tuUsuario/gf-fila-02).

## **9) Editar index.html en tu fork**

1. En tu fork, abre index.html → botón del **lápiz**.
2. Realiza un cambio (ej.: añade un <p> o modifica estilo en línea).
3. **Commit changes** en main de tu fork (o crea una rama si prefieres y luego PR interno, pero no es necesario).

## **10) Enviar los cambios al repo original (PR “upstream”)**

1. En tu fork, verás un aviso **“This branch is X commits ahead of <origen>”** o un botón **Contribute**.
2. Pulsa **Contribute → Open pull request**.
3. Asegúrate de que **base repository** es el del compañero (el original) y **head repository** es tu fork.
4. Escribe un título y descripción (ej.: docs(index): añade párrafo de créditos).
5. **Create pull request** y **asigna** al compañero como revisor.

## **11) El compañero revisa y hace merge en su repo**

1. En el **repo original**, el compañero entra al PR.
2. Revisa, comenta si procede, **Approve**, y **Merge**.
3. ¡Listo! Ya has “clonado” (vía fork) y modificado el index.html del repo del otro.

## **12) (Opcional) Sincronizar tu fork con el original**

1. En tu fork → **Sync fork** o **Fetch upstream**.
2. **Update branch** para traer los últimos cambios del repo original.