✅ Fluxo Git Proposto — Projeto com JavaScript, HTML, CSS e DOM

🔀 Branches principais

- **`main`**: código pronto para produção. Apenas código 100% validado e testado.
- **`development`**: branch de integração contínua. Tudo que está funcionando e passou por revisão (estágio `inReview`).

🌿 Branches auxiliares (curtíssimos e com objetivo claro)

Tipo	Prefixo	Finalidade
Feature	`feature/`	Nova funcionalidade
Bugfix	`fix/`	Correção de bug
Hotfix	`hotfix/`	Correção urgente direto em `main`
Release	`release/`	Preparar deploy ou release estável

🔁 Etapas do Fluxo com `inReview`

▶️ 1. Nova Feature

```
git checkout development
git pull
git checkout -b feature/nome-claro
```

Trabalha na feature → commit → push:

```
git add .
git commit -m "feat: formulário de cadastro"
git push -u origin feature/formulario-cadastro
```

Abre um Pull Request (PR) para `development` com status "In Review"

- **Revisores validam código** (comentários, aprovações)
- Após aprovação → merge na `development`
- PR é então **movido da coluna `inReview` para `done`**

🐞 2. Bug Fix

```
git checkout development
git pull
git checkout -b fix/nome-do-bug
```

Corrige o bug → commit → push:

```
git commit -m "fix: correção na lógica do botão"
git push -u origin fix/botao-duplicando
```

- Abre PR para `development`
- Status: `inReview`
- Após revisão e aprovação → merge na `development`

🔥 3. Hotfix direto na `main`

```
git checkout main
git pull
git checkout -b hotfix/erro-prod
```

Resolve bug crítico em produção → commit → push:

```
git commit -m "hotfix: remove quebra no botão de login"
git push -u origin hotfix/erro-login
```

- PR direto para `main`
- Após merge:

```
git checkout development
git pull
git merge main
```

🚀 4. Release

```
git checkout development
git pull
git checkout -b release/v1.1
```

- Ajustes finais de versão
- PR de `release/v1.1` para `main` (status: `inReview`)
- Após aprovação:
  - Merge em `main` (deploy)
  - Merge de volta em `development`

📘 Regras e Convenções

📌 Nomes de branch

Use nomes descritivos e kebab-case (traço):

- `feature/formulario-login`
- `fix/valida-idade`

✅ Commits semânticos

Prefixos claros ajudam no changelog:

- `feat:` → nova funcionalidade
- `fix:` → correção
- `refactor:` → mudança interna
- `chore:` → manutenção (ex: `.gitignore`)
- `docs:` → documentação

✅ Visual atualizado com `inReview`