

convención de nombres de ramas (HU)

este documento define cómo nombrar ramas y cómo redactar commits para mantener orden y trazabilidad en el repositorio.

estructura de la rama

plantilla general

```
<tipo-trabajo>/<stack>/HU-<id>-<ambiente>[-breve-descripcion]
```

- **tipo-trabajo:** `feature` | `bugfix` | `hotfix` | `refactor` | `chore` | `spike`
- **stack:** `app-movil` | `backend` | `data-base` | `doc`
- **HU-<id>:** identificador de la historia de usuario (p. ej., `HU-1234`)
- **ambiente:** `dev` | `qa` | `main` | `project`
- **breve-descripcion** (opcional): en *kebab-case*, 3–6 palabras, **sin** acentos ni ñ (usar `n`).

nota: todo en minúsculas salvo el prefijo `HU-`. evitar caracteres especiales distintos de `-`.

ejemplos (sin descripción)

```
feature/app-movil/HU-19630-dev
feature/app-movil/HU-19630-qa
feature/app-movil/HU-19630-main
feature/app-movil/HU-19630-project
bugfix/backend/HU-19752-qa
refactor/data-base/HU-20401-dev
chore/doc/HU-20045-main
feature/doc/HU-01-dev
```

ejemplos (con breve-descripcion opcional)

```
feature/backend/HU-23110-qa-optimize-taxpayer-endpoint
bugfix/app-movil/HU-19752-dev-fix-audio-playback-error
refactor/data-base/HU-21007-main-simplify-stored-procedures
```

definición de campos

- **feature:** trabajo nuevo (funcionalidad).
- **bugfix:** corrección de defectos.
- **hotfix:** corrección urgente en producción.

- **refactor**: cambios internos sin alterar comportamiento observable.
- **chore**: tareas de mantenimiento (build, tooling, dependencias).
- **spike**: investigación/experimento corto.
- **app-movil**: código de la app móvil.
- **backend**: servicios o apis del lado del servidor.
- **data-base**: esquemas, funciones, vistas y migraciones de bd.
- **doc**: documentación, guías, manuales.
- **HU-**: id numérico entregado por el gestor de tareas (ej.: **HU-19630**).
- **ambiente**:
 - **dev**: desarrollo
 - **qa**: pruebas/quality assurance
 - **main**: rama estable principal
 - **project**: integración temporal por proyecto (cuando aplique)
- **breve-descripcion** (opcional para la rama, **obligatoria** en el título del PR):
 - usar 3–6 palabras en *kebab-case*.
 - sin acentos ni ñ (usar **n**, ej.: **movil**, **optimizacion**).
 - ser específica y corta (evitar artículos innecesarios).

validación (regex de ejemplo)

para validar automáticamente el nombre de la rama (con descripción opcional de 3–6 palabras):

```
^(feature|bugfix|hotfix|refactor|chore|spike)/(app-movil|backend|data-base|doc)/HU-\d{1,6}-(dev|qa|main|project)(?:-[a-z0-9]+(?:-[a-z0-9]+){2,5})?$$
```

- **\d{1,6}**: id HU de 1 a 6 dígitos (ajustar según necesidad).
- el sufijo **(?:-[a-z0-9]+(?:-[a-z0-9]+){2,5})?** fuerza que, si hay descripción, tenga **entre 3 y 6** palabras (*kebab-case*).

commits (conventional commits – inglés)

usar mensajes de *conventional commits* en **inglés**, con **scope** alineado al **stack**. incluir el id de la historia (**HU-xxxx**) al final del *subject* o en el *body*.

formato

```
<type>(<scope>): <short imperative summary>

<body - optional>
```

<footer - optional>

types: `feat`, `fix`, `refactor`, `chore`, `docs`, `test`, `build`, `ci`, `perf`, `style` ...

scopes sugeridos: `app-mobile`, `backend`, `database`, `docs` (en inglés).

ejemplos

```
feat(app-mobile): add offline cache for invoices (HU-19630)
fix(backend): handle null payer id in GET /billing/taxpayers (HU-19752)
refactor(database): split large procedure into smaller units (HU-21007)
chore(docs): update branching convention
```

recomendaciones rápidas

- una rama por historia de usuario.
- rebase frecuente sobre `main` (o `qa` si aplica) para evitar *merge hell*.
- pull request con título: `[HU-19630] optimize taxpayer endpoint` y descripción con checklist.
- cerrar la rama al *mergear* el pull request.

nota: "conventional commit - inglés". los ejemplos de rama con `-ambiente` son plantillas; reemplace `-ambiente` por `-dev`, `-qa`, `-main` o `-project` según corresponda.