

# GUÍA PARA EL EQUIPO - RANDOM FOREST CLASSIFIER

## CONFIGURACIÓN INICIAL

### 1. Clonar repositorio:

```
git clone https://github.com/NIU1748951/random-forest-classifier.git
cd random-forest-classifier
```

### 2. Entorno virtual:

Abrir una terminal (cmd o powershell en windows)

```
python -m venv venv
# Activar:
# Windows: venv\Scripts\activate.bat
# Linux/Mac: source venv/bin/activate
Instalar dependencias:
pip install -r requirements.txt
```

Importante que cuando lo activeis, a la hora de correr el script o descargar las dependencias hayais activado el venv (paso 2). Os debería salir (venv) en verde a la izquierda

### 3. ESTRUCTURA DEL PROYECTO:

```
├── src/                # Código fuente
│   ├── nodes/         # Nodos (Node, Leaf, Parent)
│   ├── utils/         # Métricas y helpers
│   └── random_forest.py # Clase principal
├── tests/             # Pruebas unitarias
└── requirements.txt    # Dependencias
```

### 4. FLUJO DE TRABAJO:

Actualizar localmente antes de empezar:

```
git checkout main git pull origin main
```

Crear rama nueva:

```
git checkout -b tipo/descripcion
```

Ejemplo: `git checkout -b feature/nodes`

Hacer commits:

```
git add archivos
git commit -m "feat: mensaje descriptivo"
git push origin nombre-rama
```

Crear Pull Request (PR) en GitHub:

- Compara tu rama con main.
- Describe cambios y etiqueta revisor.
- **NO** fusionar sin aprobación.

## 5. Pruebas:

```
pytest tests/ # Ejecutar ANTES de cada PR
```

## 6. SOLUCIÓN DE PROBLEMAS:

Conflictos en Git:

Sincronizar con main:

```
git checkout main
git pull origin main
git checkout tu-rama
git merge main
```

**Resolver conflictos manualmente.**

Commit y push:

```
git add .
git commit -m "fix: conflictos"
git push
```

**¡IMPORTANTE!**

Trabajar SIEMPRE en ramas.