

MANUAL DE INSTALACIÓN DEL TABLERO

Proyecto MIAD_DSA_PROYECTO

1. Introducción

Este manual describe el proceso de instalación y ejecución del Tablero de Predicción de Riesgo de Abandono de Clientes, desarrollado con Dash y Plotly como parte de la Entrega Final del curso *Despliegue de Soluciones Analíticas*.

El objetivo del manual es permitir que cualquier usuario técnico pueda reproducir el entorno de ejecución del tablero localmente o en un servicio en la Nube (IaaS/PaaS), cumpliendo con los criterios de la rúbrica de *Tablero Desplegado*.

2. Requisitos previos

2.1 Software necesario

- Python 3.10+
- pip (administrador de paquetes)
- (Opcional) Git para clonar el repositorio
- (Opcional) Entorno virtual
 - venv (incluido en Python)
 - o conda

2.2 Sistema operativo compatible

- macOS
- Windows 10+
- Linux (Ubuntu recomendado)

2.3 Repositorio del proyecto

Clonar o descargar el repositorio:

https://github.com/FelipeRubiano/MIAD_DSA_PROYECTO

El tablero se encuentra en: MIAD_DSA_PROYECTO/dashboard/

3. Instalación local

3.1 Crear y activar entorno virtual

Desde la raíz del repositorio:

macOS / Linux:

```
python3 -m venv .venv  
source .venv/bin/activate
```

Windows:

```
python -m venv .venv  
.venv\Scripts\activate
```

3.2 Instalar dependencias

En la raíz del repositorio: `pip install -r requirements.txt`

Este archivo instala, entre otros:

- Dash
- Plotly
- Pandas
- NumPy
- scikit-learn
- joblib

3.3 Ubicación del modelo entrenado

El tablero usa un archivo `.pkl` ubicado en:

`MIAD_DSA_PROYECTO/models/RandomForest.pkl`

Este archivo debe mantenerse en esa ruta para que el tablero funcione.

4. Ejecutar el tablero localmente

Desde la raíz del repo (o desde `/dashboard`):

```
python dashboard/app.py
```

Si el entorno está correctamente instalado, aparecerá en la terminal:

Dash is running on <http://0.0.0.0:8050/>

Abrir el navegador y acceder a:

<http://localhost:8050>

4.1 Verificación

Al cargar el archivo CSV de prueba:

/data/processed/Bank Customer Churn Prediction.csv

El tablero debe:

- Mostrar KPIs
- Mostrar 4 visualizaciones
- Activar filtros
- Listar el Top 10 de clientes
- Actualizar resultados al mover el umbral

5. Instalación de la API (si se usa en conjunto)

Si el tablero interactúa con una API propia, esta debe instalarse y ejecutarse antes del tablero.

5.1 Ubicación

La API se encuentra en el directorio:

MIAD_DSA_PROYECTO/api/

5.2 Instalación

Dentro de la carpeta /api:

```
pip install -r requirements.txt
```

5.3 Ejecución

```
uvicorn main:app --host 0.0.0.0 --port 8001
```

La URL debe configurarse en el tablero (si aplica).

6. Despliegue en la nube (IaaS/PaaS)

Para cumplir con la rúbrica, el tablero debe estar desplegado en un servicio en la nube.

6.1 Opciones compatibles

- Railway (PaaS)
- Render (PaaS)
- AWS EC2 (IaaS)
- Google Cloud VM

6.2 Despliegue en Railway (opción recomendada y simple)

6.2.1 Crear servicio

1. Entrar a Railway
2. Crear un servicio de tipo **Python**
3. Subir el repositorio

6.2.2 Configurar comando de ejecución

En Railway → Variables → Start Command:

```
python dashboard/app.py
```

6.2.3 Configurar puerto

Railway usa variable de entorno:

PORT

Modificar app.run() así:

```
app.run(host="0.0.0.0", port=int(os.environ.get("PORT", 8050)))
```

6.2.4 Deploy automático

Railway construirá el proyecto y devolverá una URL pública.

Ejemplo:

<https://tablero-churn-production.up.railway.app>

Al abrir esa URL, el tablero debe funcionar igual que localmente.

6.3 Despliegue en AWS EC2 (IaaS)

Este despliegue suma puntos altos en rúbrica si fue utilizado.

6.3.1 Instalar dependencias en EC2

```
sudo apt update
```

```
sudo apt install python3 python3-venv python3-pip
```

```
git clone <repo>
```

```
cd MIAD_DSA_PROYECTO
```

```
python3 -m venv .venv
```

```
source .venv/bin/activate
```

```
pip install -r requirements.txt
```

6.3.2 Ejecutar el tablero

```
python dashboard/app.py
```

6.3.3 Abrir el puerto 8050 en el Security Group

Agregar regla:

Type: Custom TCP

Port: 8050

Source: 0.0.0.0/0

Acceder vía:

`http://<PUBLIC_IP>:8050`

7. Errores comunes y soluciones

El tablero no carga

- Verificar RandomForest.pkl en /models
- Revisar compatibilidad de Python

El archivo CSV no se procesa

- Debe ser UTF-8
- Debe contener variables numéricas
- Columnas categóricas deben coincidir con el entrenamiento

Despliegue en nube no abre

- Puerto no abierto
- Start command incorrecto
- Error en versiones de Python