# Tesis Doctoral - Simulaciones de IoT con NS-3

Este repositorio contiene las simulaciones realizadas para mi tesis doctoral, enfocada en la evaluación de protocolos de enrutamiento (AODV y DSDV) en redes de Internet de las Cosas (IoT) utilizando NS-3. El proyecto evalúa métricas como consumo de energía, latencia y cambios en la tabla de enrutamiento en escenarios con 20 nodos y 3 interferencias, sin nodos maliciosos.

## Descripción

El proyecto utiliza NS-3 para simular redes IoT con los protocolos AODV y DSDV. Los datos generados incluyen trazas de red (.pcap), consumo de energía (energy\_consumption.csv), posiciones de nodos móviles (mobile\_positions.csv), paquetes (packets\_normal.csv) y cambios en la tabla de enrutamiento (routing\_table\_changes.csv). Los resultados se procesan con scripts de Python para generar métricas resumidas y gráficos.

## Requisitos

* **NS-3**: Versión 3.36 o superior.
* **Python**: 3.8+ con las siguientes dependencias:
* pip install -r src/processing/requirements.txt
* **Sistema operativo**: Ubuntu 20.04 o superior (recomendado).
* **Git**: Para clonar y gestionar el repositorio.
* **Otras herramientas**: Bash para scripts de automatización.
* **Espacio de almacenamiento externo**: Google Drive, Zenodo o similar para datos crudos.

## Estructura del proyecto

/tesis-doctoral-ns3/  
├── src/  
│ ├── ns3/  
│ │ ├── simulacioniot.cc  
│ │ └── [otros scripts .cc si los tienes]  
│ ├── processing/  
│ │ ├── manual\_metrics\_dsr.py  
│ │ ├── post\_processing.py  
│ │ └── requirements.txt  
│ └── automation/  
│ ├── run\_simulations.sh  
│ └── [otros scripts de automatización]  
├── data/  
│ ├── raw/ # No subido a GitHub, solo enlace externo  
│ │ ├── aodv/  
│ │ │ ├── run1/  
│ │ │ ├── run2/  
│ │ │ └── ...  
│ │ ├── dsdv/  
│ │ │ ├── run1/  
│ │ │ ├── run2/  
│ │ │ └── ...  
│ ├── processed/  
│ │ ├── aodv\_metrics\_summary.csv  
│ │ ├── dsdv\_metrics\_summary.csv  
│ │ └── plots/ # Gráficos generados  
│ └── external\_data.md # Enlaces a datos crudos  
├── docs/  
│ ├── README.md # Este archivo  
│ ├── experiments.md # Registro detallado de experimentos  
│ ├── methodology.md # Explicación de la metodología  
│ └── results.md # Resumen de resultados  
├── examples/  
│ ├── example\_simulation.cc # Ejemplo de script NS-3  
│ └── example\_processing.py # Ejemplo de script de procesamiento  
├── .gitignore  
└── LICENSE

### Explicación de la estructura

* **src/**: Código fuente.
  + **ns3/**: Scripts de NS-3 (simulacioniot.cc).
  + **processing/**: Scripts de Python para procesar datos (manual\_metrics\_dsr.py, post\_processing.py) y dependencias (requirements.txt).
  + **automation/**: Scripts para automatizar simulaciones (run\_simulations.sh).
* **data/**:
  + **raw/**: Datos crudos (.pcap, energy\_consumption.csv, etc.) no incluidos en GitHub debido a su tamaño. Ver data/external\_data.md para enlaces.
  + **processed/**: Resultados procesados (CSV con métricas resumidas, gráficos en PNG/PDF).
  + **external\_data.md**: Enlaces a datos crudos en Google Drive o Zenodo.
* **docs/**: Documentación en Markdown.
  + README.md: Instrucciones generales (este archivo).
  + experiments.md: Detalles de cada experimento.
  + methodology.md: Metodología de las simulaciones.
  + results.md: Resumen de resultados.
* **examples/**: Ejemplos de scripts para facilitar la comprensión.
* **.gitignore**: Evita subir archivos generados o grandes.
* **LICENSE**: Licencia del proyecto (MIT).

## Instalación

Sigue estos pasos para configurar el entorno en Ubuntu.

### 1. Instalar dependencias del sistema

sudo apt-get update  
sudo apt-get install git build-essential python3 python3-pip

### 2. Instalar NS-3

git clone https://gitlab.com/nsnam/ns-3-dev.git  
cd ns-3-dev  
./ns3 configure --enable-examples --enable-tests  
./ns3 build

### 3. Clonar este repositorio

git clone https://github.com/tu-usuario/tesis-doctoral-ns3.git  
cd tesis-doctoral-ns3

### 4. Instalar dependencias de Python

pip install -r src/processing/requirements.txt

## Configuración del proyecto

1. **Mover scripts existentes** (si tienes una estructura previa, por ejemplo, en /home/diego/Descargas/ID/SIMULACIONES 9/MOD2/PROYECTO 20250603\_0855):

* cp /home/diego/Descargas/ID/SIMULACIONES\ 9/MOD2/PROYECTO\ 20250603\_0855/scripts/simulacioniot.cc src/ns3/  
  cp /home/diego/Descargas/ID/SIMULACIONES\ 9/MOD2/PROYECTO\ 20250603\_0855/scripts/manual\_metrics\_dsr.py src/processing/  
  cp /home/diego/Descargas/ID/SIMULACIONES\ 9/MOD2/PROYECTO\ 20250603\_0855/scripts/post\_processing.py src/processing/  
  cp /home/diego/Descargas/ID/SIMULACIONES\ 9/MOD2/PROYECTO\ 20250603\_0855/scripts/run\_simulations.sh src/automation/

1. **Copiar datos procesados** (por ejemplo, métricas resumidas):

* cp /home/diego/Descargas/ID/SIMULACIONES\ 9/MOD2/PROYECTO\ 20250603\_0855/int\_no\_mal/AODV/run1/metrics/\*.csv data/processed/  
  cp /home/diego/Descargas/ID/SIMULACIONES\ 9/MOD2/PROYECTO\ 20250603\_0855/int\_no\_mal/DSDV/run1/metrics/\*.csv data/processed/

1. **Subir datos crudos a un servicio externo**:
   * Comprime los datos crudos:
   * cd /home/diego/Descargas/ID/SIMULACIONES\ 9/MOD2/PROYECTO\ 20250603\_0855/int\_no\_mal  
     tar -czf aodv\_data.tar.gz AODV  
     tar -czf dsdv\_data.tar.gz DSDV
   * Sube aodv\_data.tar.gz y dsdv\_data.tar.gz a Google Drive o Zenodo.
   * Crea data/external\_data.md con los enlaces:
   * # Datos Crudos  
       
     Los datos crudos generados por las simulaciones están disponibles en:  
     - \*\*AODV\*\*: [Enlace a Google Drive/Zenodo]  
     - \*\*DSDV\*\*: [Enlace a Google Drive/Zenodo]
2. **Crear .gitignore**: Crea un archivo .gitignore con:

* # Archivos generados por NS-3  
  build/  
  \*.o  
  \*.log  
  \*.tr  
  \*.pcap  
    
  # Datos crudos (muy grandes)  
  data/raw/\*  
    
  # Archivos de Python  
  venv/  
  \_\_pycache\_\_/  
  \*.pyc  
    
  # Archivos temporales  
  \*.swp  
  \*.bak  
  post\_processing.log  
    
  # Dependencias innecesarias  
  tzdata/

1. **Crear requirements.txt**: En src/processing/requirements.txt, añade:

* pandas  
  matplotlib

## Cómo ejecutar

1. **Correr una simulación**:

* cd src/automation  
  ./run\_simulations.sh
* Esto ejecuta simulacioniot.cc y genera datos crudos en data/raw/aodv/ o data/raw/dsdv/.

1. **Procesar resultados**:

* cd src/processing  
  python post\_processing.py
* Los resultados procesados (CSV, gráficos) se guardan en data/processed/.

1. **Visualizar resultados**:
   * Consulta los archivos en data/processed/plots/ (gráficos en PNG/PDF).
   * Revisa el resumen en docs/results.md.

## Subir el proyecto a GitHub

1. **Crear un repositorio en GitHub**:
   * Ve a [github.com](https://github.com/), inicia sesión y haz clic en **New repository**.
   * Nombra el repositorio tesis-doctoral-ns3.
   * Elige **privado** (si no quieres que sea público) y marca **Add a README file**.
   * Haz clic en **Create repository**.
2. **Configurar Git localmente** (si no lo has hecho):

* git config --global user.name "Tu Nombre"  
  git config --global user.email "tu.email@ejemplo.com"

1. **Subir los archivos**:

* cd tesis-doctoral-ns3  
  git add .  
  git commit -m "Estructura inicial del proyecto de tesis"  
  git push origin main

1. **Manejar nuevos experimentos**:
   * Crea una rama para cada experimento:
   * git checkout -b experimento-aodv-20nodos
   * Modifica scripts o añade resultados procesados.
   * Sube la rama:
   * git add .  
     git commit -m "Experimento AODV con 20 nodos"  
     git push origin experimento-aodv-20nodos
   * Crea un **Pull Request** en GitHub desde experimento-aodv-20nodos a main y fusiónalo.
2. **Alternativa con GitHub Desktop**:
   * Descarga e instala [GitHub Desktop](https://desktop.github.com/).
   * Clona el repositorio: **File > Clone Repository**.
   * Arrastra los archivos a la interfaz, escribe un mensaje de commit y haz clic en **Commit to main**.
   * Haz clic en **Push origin**.

## Acceso a datos crudos

Los datos crudos (.pcap, energy\_consumption.csv, etc.) son demasiado grandes para GitHub y están almacenados externamente. Consulta data/external\_data.md para los enlaces a Google Drive o Zenodo.

## Experimentos

Los detalles de cada experimento están en docs/experiments.md. Ejemplo: - **AODV, 20 nodos, sin nodos maliciosos**: - Fecha: 03/06/2025 - Script: src/ns3/simulacioniot.cc - Parámetros: 20 nodos, 10 minutos, 3 interferencias. - Resultados: data/processed/aodv\_metrics\_summary.csv - **DSDV, 20 nodos, sin nodos maliciosos**: - Fecha: 03/06/2025 - Script: src/ns3/simulacioniot.cc - Parámetros: 20 nodos, 10 minutos, 3 interferencias. - Resultados: data/processed/dsdv\_metrics\_summary.csv

## Metodología

La metodología, incluyendo la configuración de NS-3 y los protocolos AODV/DSDV, está en docs/methodology.md.

## Resultados

Un resumen de los resultados (consumo de energía, latencia, etc.) está en docs/results.md. Los gráficos generados se encuentran en data/processed/plots/.

## Ejemplos

Consulta examples/ para: - example\_simulation.cc: Ejemplo de script NS-3. - example\_processing.py: Ejemplo de script de procesamiento.

## Consejos para nuevos experimentos

1. Crea una nueva rama:

* git checkout -b experimento-nuevo

1. Modifica simulacioniot.cc o los parámetros en run\_simulations.sh.
2. Guarda los datos crudos en Google Drive/Zenodo y los procesados en data/processed/.
3. Actualiza docs/experiments.md con los detalles.
4. Sube los cambios y crea un Pull Request.

## Licencia

Este proyecto está licenciado bajo la [Licencia MIT](LICENSE).

## Contacto

* **Nombre**: [Tu Nombre]
* **Correo**: [tu.email@ejemplo.com]
* **GitHub**: [tu-usuario](https://github.com/tu-usuario)

## Recursos adicionales

* [Tutorial de Git y GitHub para principiantes](https://www.youtube.com/watch?v=HiXLkL42fys) (en español).
* [ProGit en español](https://git-scm.com/book/es/v2) (primeros capítulos).
* [NS-3 Documentation](https://www.nsnam.org/documentation/).