

Table of contents

1 Operación y despliegue	1
1.1 Deploy automático	1
1.2 Comandos locales	1
1.3 Backups	2
2 Servicios por entorno	2
2.1 Logs	3
2.2 Observabilidad	3
2.3 Roles y permisos	3

1 Operación y despliegue

1.1 Deploy automático

1. Merge a main.
2. GitHub Actions ejecuta:
 - `make lint`
 - `make test`
 - `alembic upgrade --sql` check para migraciones pendientes.
3. SSH al VPS (VPS_HOST, VPS_USER, VPS_KEY).
4. `git pull, make migrate, make seed_min`.
5. `sudo systemctl restart illanes00-ep` (hasta migrar al unit templado `illanes00-ep@production`).
6. Smoke tests (`/healthz, /api/0.6.9/health, /api/0.6.9/hello`).

1.2 Comandos locales

- `make dev`: corre Uvicorn con autoreload.
- `make seed_min`: carga dimensiones básicas.
- `make smoke`: prueba endpoints (`/healthz, /api/<ver>/health, /api/<ver>/hello`).

- **Pipeline ENUSC**

```
ENUSC_YEARS=2022,2023 ENUSC_LOAD_DB=0 PYTHONPATH=. python -c "from etl.pipelines.enusc.pip
```

– Variables clave:

- * `ENUSC_LOAD_DB`: 1 para escribir en Postgres (requiere credenciales configuradas); 0 deja solo parquet/CSV.
- * `ENUSC_DB_APPEND_ONLY`: fuerza modo append en tablas bronze/silver/gold.
- * `ENUSC_ONLY_MODULES, ENUSC_SKIP_MODULES`: controlan la exportación bronze vs. modulares.

- Artefactos se depositan en `data/bronze/enusc/`, `data/silver/enusc/`, `data/gold/enusc/` y reportes `data/silver/enusc/reportes/` (manifest de módulos, resumen personas/hogares).
- CLI auxiliar:
 - * `python -m etl.pipelines.enusc.cli ingest --years 2022 2023` (interpreta flags y exporta usando las clases nuevas).
 - * `python -m etl.pipelines.enusc.cli preview-columns --limit 1` entrega conteo de columnas por módulo sin ejecutar el pipeline completo.
 - * `python -m etl.pipelines.enusc.cli compare-manifests data/silver/enusc/reportes/mod backups/enusc/modulos_manifest.json` avisa de módulos/archivos añadidos o faltantes entre corridas.

1.3 Backups

- Ejecutar `scripts/backup_environment.py` para cada entorno:

```
.venv/bin/python scripts/backup_environment.py --env production
.venv/bin/python scripts/backup_environment.py --env development
.venv/bin/python scripts/backup_environment.py --env test
```

- Los artefactos se guardan en `backups/<entorno>/<timestamp>/` (ej. producción: `backups/production/20251020-180032/`).
- Cada carpeta contiene `database.dump` (formato `pg_dump -F c`), `metadata.tar.gz` con `config/`, `data/meta/`, `data/schema/`, `docs/` y una copia del `.env`.
- Programar un `cron` o `systemd timer` pendiente para automatizar la ejecución diaria y sincronizar los dumps a almacenamiento externo.
- Para `test` primero crear el rol y la base desde postgres:

```
sudo -u postgres createuser -P ep_app_test
sudo -u postgres createdb epdatos_test -O ep_app_test
```

2 Servicios por entorno

- Desarrollo (`APP_PORT=8211`): `systemctl --user restart illanes00-ep@development`
- Test (`APP_PORT=8210`): `systemctl --user restart illanes00-ep@test`
- Habilitar arranque automático: `systemctl --user enable illanes00-ep@development`, `systemctl --user enable illanes00-ep@test`
- Asegura `loginctl enable-linger illanes00` para que las unidades de usuario persistan tras `logout`.
- Producción seguirá en `sudo systemctl restart illanes00-ep` hasta migrar a `illanes00-ep@production`.

2.1 Logs

- Producción (legacy): `sudo journalctl -u illanes00-ep -n 200 --no-pager`
- Nuevas unidades de usuario: `journalctl --user -u illanes00-ep@development -n 200`,
`journalctl --user -u illanes00-ep@test -n 200`

2.2 Observabilidad

- Sentry se inicializa automáticamente cuando `SENTRY_DSN` está definido en `.env`.
- El backend reporta errores y tiempos vía `SentryAsgiMiddleware`; el entorno se etiqueta con `ENVIRONMENT` y la versión con `VERSION` + commit.
- Ajusta `SENTRY_TRACES_SAMPLE_RATE` / `SENTRY_PROFILES_SAMPLE_RATE` para habilitar trazas y perfiles (0.0 por defecto).
- Recomiendo crear reglas de alerta (correo/Slack) directamente en Sentry para caídas frecuentes o spikes de 5xx.

2.3 Roles y permisos

- Usuario `illanes00` posee `sudo` con `NOPASSWD` para `systemctl status|restart illanes00-ep`.
- Base de datos `epdatos` pertenece al rol `illanes00-ep`.