

## ENTREGA FINAL - TELEMATICACHE

---

- \*Autor:\*\* Jose Alejandro Jimenez Vasquez
- \*Repositorio:\*\* 'll33311/telematicache' (<https://github.com/ll33311/telematicache>)
- \*Fecha:\*\* 11/11/2025

- --

### 1. Alcance General

---

- Plataforma integral para telemetria y control con tres capas principales:
  1. \*\*Sensores & Gateway (FastAPI)\*\*: recibe ingestas y comandos ('proyecto\_integrado/services/telemetry-gateway').
  2. \*\*Portal web Croody (Django)\*\*: UI, reportes y modulos CRUD ('proyecto\_integrado/Croody').
  3. \*\*DNS autoritativo BIND9\*\*: despliegue maestro/esclavo en AWS ('infra/dns/').
- Infraestructura descrita en 'arquitectura.md' y 'manual\_tecnico.md': VPC '10.50.0.0' con subred publica (NAT/Bastion) y privada (servicios).
- 'ns1' (maestro) -> instancia \_croody v1\_ ('i-0571eebcb74dfcdb7', IP privada '172.31.42.77').
- 'ns2' (esclavo) -> instancia \_bind-slave\_ ('i-074507051ca06113e', IP privada '172.31.71.231').
- Automatizacion por scripts Bash y GitHub Actions:
  - 'dns-lint.yml', 'bind-deploy.yml', 'terraform-ci.yml', 'full-stack-validate.yml'.
  - Evidencias en 'extras/evidencias\_finales.md' y 'extras/local\_ci\_report.md'.

- --

### 2. Trazabilidad de Indicativos

---

#### 2.1 Servidor & Protocolo (Requisito 1 y 4 del enunciado)

---

- Implementado en 'proyecto\_integrado/services/telemetry-gateway/app/main.py'.
- Endpoints:
  - 'POST /api/telemetry/ingest': registra mediciones cada 15 s transmitidas a todos clientes conectados.

- 'GET /api/telemetry/live|last|query': entrega datos en tiempo real/historicos.
- 'POST /api/telemetry/commands': admite 'LEFT/RIGHT/FORWARD/BACKWARD', valida colisiones.
- 'GET /api/telemetry/clients': lista clientes activos.
- Concurrencia con asyncio (queues + background tasks). Manejo de multiples clientes desconexiones.
- Usuario administrador autenticado via 'ADMIN\_API\_KEY'; comandos solo se aceptan des incluso si cambia la IP.
- Protocolo tipo texto/JSON documentado en 'manual\_tecnico.md' e 'informe\_tecnico\_entrega3.md' (formato de mensajes, codigos, secuencia tipo RFC).

## 2.2 Clientes (Indicativo 2)

---

- Clientes Python ('clients/python/') usan sockets Berkeley, construyen peticiones segun el protocolo y muestran respuestas de exito/error.
- UI Django integra telemetria en 'Croody/templates/landing/monitor.html', consumiendo '/api/telemetry/live'.

## 2.3 Funcionamiento & Comunicacion (Indicativo 3)

---

- **Robustez:** 'scripts/validate\_full\_stack.sh' levanta la pila con Docker Compose, ejecuta 'dig', 'curl', 'pytest tests/e2e/test\_gateway\_smoke.py' y recopila logs.
- **Coordinacion:** 'scripts/run\_local\_ci.sh' replica los workflows de GitHub (lint ETL, 'named-check\*', 'dig', Terraform fmt/validate/plan) para validar cada paso antes del merge.

## 2.4 Documentacion (Indicativo 4)

---

- Manuales:
  - 'manual\_tecnico.md' (procedimientos de despliegue y operacion).
  - 'arquitectura.md' (topologia VPC, decisiones clave).
  - 'docs/dns\_operacion.md' (runbook de BIND + failover).
  - 'plan\_practico\_endurecimiento.md', 'docs/indicativos\_resumen.md', 'docs/indicativos\_checklist.txt'.
- Matriz de cumplimiento: 'docs/matriz\_indicativos.md'.

- --

## 3. DNS Autoritativo y VPC en AWS

```
-----  
  
1. **Generacion de llaves y plantillas:**  
```bash  
scripts/dns/setup_bind.sh \  
--domain croody.app \  
--master-ip 172.31.42.77 \  
--slave-ip 172.31.71.231  
``  
2. **Zona activa:** `infra/dns/bind-master/zones/croody.app.db` (serial `2025111800`,  
registros `ns1/ns2` actualizados).  
3. **Variables de entorno:** `infra/dns/bind-master/env.example` y `bind-  
slave/env.example` sincronizadas (ALLOW_TRANSFER, NOTIFY_TARGETS y MASTER_IP correctos).  
4. **Runbook:** `docs/dns_operacion.md` detalla variables criticas, comandos `dig`,  
de failover y checklist post-cambio.  
5. **Secrets y AWS:**  
- Tabla completa en `docs/secrets_map.md` (IPs privadas, TSIG, AWS/ECR, SSH,  
Cloudflare).  
- Terraform (`infra/terraform/`) define VPC, subredes, rutas, NAT y bastion. Validado  
en `terraform-ci.yml`.  
- --
```

#### 4. Procedimiento de Despliegue Rapido

---

```
1. **Preparar entorno local**  
```bash  
python3 -m venv .venv  
source .venv/bin/activate  
pip install -r proyecto_integrado/requirements.txt  
``  
2. **Servicios de aplicacion**  
```bash  
cd proyecto_integrado  
cp .env.example .env # ajustar SECRET_KEY, TG_DB_URL, ADMIN_API_KEY  
docker compose up --build  
``  
- Verificar `http://localhost:8080`, `/api/telemetry/healthz`, `/api/telemetry/liv  
3. **DNS/BIND local**  
```bash  
cd infra/dns  
cp bind-master/env.example bind-master/.env  
cp bind-slave/env.example bind-slave/.env  
source bind-master/keys/tsig.env
```

```

docker compose up -d bind-master bind-slave
dig @127.0.0.1 croody.app SOA
dig @127.0.0.1 croody.app AXFR
```
4. **Validacion integral**
```bash
scripts/run_local_ci.sh      # genera extras/local_ci_report.md
scripts/validate_full_stack.sh # genera extras/evidencias_finales.md
```
5. **Produccion**
- Actualizar secrets en GitHub ('BIND_*', 'AWS_*', 'BIND_*_ECR_REPO', 'BIND_SSH_*', 'VALIDATION_BASE_URL', 'CF_*').
- 'git push' y monitorear 'dns-lint', 'bind-deploy', 'terraform-ci', 'full-stack-validate'.

```

- --

## 5. Evidencias Generadas

---

- 'extras/local\_ci\_report.md': estado de cada etapa (Terraform, Docker, dig, full-stack-validate).
  - 'extras/evidencias\_finales.md': resultados de Terraform fmt/validate/plan, Docker Compose, 'dig', 'curl', 'pytest' y logs finales.
  - 'docs/indicativos\_resumen.md' + 'docs/matriz\_indicativos.md': resumen textual del cumplimiento.
  - Capturas finales (pendientes) deben agregarse en 'extras/' si se requieren pruebas visuales.
- --

## 6. Artefactos y Rutas Clave

---

| Componente                             | Ruta                                                            |
|----------------------------------------|-----------------------------------------------------------------|
| Telemetry Gateway (FastAPI)            | ' proyecto_integrado/services/telemetry-gateway '               |
| Portal Croody (Django)                 | ' proyecto_integrado/Croody/ '                                  |
| Clientes de referencia                 | ' clients/python/ '                                             |
| Scripts de despliegue / automatizacion | ' deploy.sh', 'deploy_from_scratch.sh', 'scripts/deployment/ '  |
| DNS autoritativo (Docker)              | ' infra/dns/ '                                                  |
| Infraestructura como codigo            | ' infra/terraform/ '                                            |
| Workflows GitHub Actions               | '.github/workflows/*.yml'                                       |
| Documentacion principal                | ' manual_tecnico.md', 'arquitectura.md', 'docs/introduction.md' |

- --

## 7. Observaciones Finales

-----

- Todos los valores sensibles estan documentados sin exponer secretos reales ('docs/secrets\_map.md').
- Ejecutar 'scripts/run\_local\_ci.sh' antes de cada entrega para adjuntar un reporte actualizado.
- Documentos legados no usados fueron archivados en 'trash/docs\_legacy/' para mantener repositorio limpio.

"\*\*Nota:\*\* Este documento fue redactado en Markdown y convertido a PDF ('entrega.pdf') para conservar el formato solicitado."

"\*\*Responsable:\*\* Jose Alejandro Jimenez Vasquez."