Texto

Descripción generada automáticamenteI

**Escuela Superior de Ciencias Experimentales y Tecnología**

**Curso Académico 2025/26**

**Directores:**

**Jorge Luis Huarachi Salbador**

**Arquitectura para la Recomendación de Rutas**

**Trabajo de Fin de Grado**

**GRADO EN INGENIERÍA**

**DE TECNOLOGÍAS INDUSTRIALES**

Imagen en blanco y negro

El contenido generado por IA puede ser incorrecto.

**Grado en Ingeniería de Tecnologías Industriales**

**Trabajo de Fin de Grado**

El presente trabajo, titulado ***Arquitectura Distribuida para la recomendación de Rutas***, constituye la memoria correspondiente a la asignatura Trabajo de Fin de Grado que presenta **D/Dª. *NOMBRE DEL AUTOR/A*** como parte de su formación para aspirar al Título de Graduado/a en Ingeniería de Tecnologías Industriales. Este trabajo ha sido realizado en ***NOMBRE DEL CENTRO*** en el ***NOMBRE DEL DEPARTAMENTO*** bajo la dirección de ***NOMBRE DEL/LOS DIRECTOR/ES***

Móstoles, (DÍA) de (MES) de (AÑO)

# Contenido

[1. Contenido 2](#_Toc209649289)

[2. 3](#_Toc209649290)

[3. 3](#_Toc209649291)

[4. 3](#_Toc209649292)

[4.1. 3](#_Toc209649293)

[4.1.1. 3](#_Toc209649294)

(Tabla de figuras o tablas)

# Resumen

# Introducción

# Objetivos

# Solución Técnica

## Diseño de la arquitectura distribuida\*\* (alto nivel)

### Visión general y principios

El sistema propuesto persigue **recomendar rutas de evacuación en tiempo real** en espacios cerrados, integrando sensórica IoT, un **modelo espacial indoor** normalizado y un **motor de rutas** que reacciona ante cambios del entorno. Los principios arquitectónicos que lo guían son: (i) **tiempo real** y baja latencia en la propagación sensor→decisión, (ii) **resiliencia** y capacidad de degradación controlada ante fallos parciales, (iii) **interoperabilidad** mediante estándares abiertos (IndoorGML, HTTP/REST, MQTT/LoRaWAN), y (iv) **trazabilidad** extremo a extremo de cada lectura y decisión.

El modelado espacial se alinea con **IndoorGML**: la norma 1.1 (1) define el núcleo (*CellSpace*, *State/Transition* y el grafo de accesibilidad), mientras que **IndoorGML 2.0 (Part 1 – Conceptual Model)** (2) actualiza conceptos y abre la puerta a codificaciones SQL/JSON que facilitan su mapeo a SGBD relacionales y vistas para navegación. Esta compatibilidad garantiza intercambiabilidad del modelo indoor y sostenibilidad del diseño a medio plazo.

**Figura sugerida 4.1-A.** *Vista de principios arquitectónicos*: un diagrama tipo “arco de requisitos no funcionales” (RT, resiliencia, interoperabilidad, trazabilidad) y su trazado a componentes.

### Componentes

#### Capa IoT/AmI (edge)

La capa de borde está compuesta por **balizas BLE** y **sensores ambientales** (humo, temperatura, CO₂, TVOC, IAQ) desplegados en salas y pasillos. Para cobertura campus y bajo consumo, se emplea **LoRaWAN** con gateways (p.ej., Dragino LPS8-N) asociados a **The Things Network (TTN)**. La localización basada en **RSSI BLE** es viable pero sensible a **multitrayecto, interferencias y fluctuaciones**, por lo que se recomienda combinarla con conocimiento topológico (celdas IndoorGML) y filtrado temporal. ( )

Evidencia operacional previa URJC/SENIALAB: registro de *gateways* y balizas en TTN, decodificación (*payload formatter*) y tabla de inventario por edificio/planta. (Lorawan)

**Figura sugerida 4.1-B.** Topología edge: sensores → gateway LoRaWAN → TTN.

#### Ingesta y Complex Event Processing (CEP)

Las tramas uplink llegan a TTN y se **reenviarán** por **webhooks** a un middleware HTTP(s) que normaliza, decodifica y persiste lecturas. Sobre ese flujo se aplican reglas CEP (ventanas, umbrales, propagación a celdas adyacentes) para derivar un **índice de “seguridad” por celda** que alimenta el peso del grafo de evacuación. La literatura CEP en **gestión de emergencias** avala la necesidad de acciones complejas disparadas por patrones de eventos.

Implementación previa: *webhook* TTN → **Flask** en servidor SENIALAB, decodificación y guardado en BBDD (servicio *systemd* y tareas *cron*); dashboards en Grafana para validación. (Lorawan)

El repositorio de estado integra (i) **modelo indoor** (celdas, límites, estados/transiciones) conforme a IndoorGML mapeado a SQL y (ii) **series temporales** de sensores. PostGIS habilita tipos/índices/funciones espaciales para consultas eficientes (centroides, contención, adyacencias) sobre *CellSpace* y *CellSpaceBoundary* y soporta vistas del **grafo de accesibilidad**.

En el proyecto se han definido **vistas** para nodos (v\_dual\_nodes) y aristas (v\_dual\_edges\_idx) y *triggers* que actualizan el *score* de seguridad por celda a partir de lecturas. (Se detalla en §4.2).

**Figura sugerida 4.1-D.** Esquema lógico BD: tablas IndoorGML↔SQL y tablas de lecturas.

#### Servicio de grafo/rutas

Servicio responsable de **construir/actualizar** el grafo navegable (nodos=centroides de CellSpace, aristas=puertas/pasillos) y de computar rutas mínimas condicionadas por **seguridad** y **redundancia** (véase §4.4). Obtiene su entrada de las **vistas** en BD, aplicando filtros de umbral y políticas de accesibilidad (p. ej., evitar escaleras).

#### API del sistema

Una **API REST** expone: (i) consulta del grafo actual y (ii) cálculo de ruta desde una celda origen a la salida óptima. **FastAPI** facilita contracts OpenAPI y handlers asíncronos para peticiones concurrentes de usuarios y paneles de operación.

#### Clientes y actuadores

**App móvil / panel de operador**: consumo de rutas y estado de seguridad por celda; posibilidad *push* (WS) para avisos.

**Señalización dinámica** (luces direccionales/pantallas) como actuadores externos de las decisiones CEP.

**Tabla sugerida 4.1-T1.** *Contratos JSON mínimos*: Node {id, geom, safety}, Edge {u,v,cost}, Route {origin, target, path[], cost, redundancy}.

### Flijo de datos y actualización en tiempo real

El **pipeline** opera como sigue: **telemetría** (BLE/LoRa) → **TTN** → **webhook** HTTP(S) → *middleware* (normalización/decodificación) → **CEP** (cálculo de seguridad por celda) → **BD** (vistas de grafo actualizadas) → **servicio de rutas** → **entrega** a clientes por REST/WS. Con TTN *webhooks* se garantiza entrega directa por evento (uplink) al servidor, cumpliendo los requisitos de latencia del sistema. [The Things Industries+1](https://www.thethingsindustries.com/docs/integrations/webhooks/?utm_source=chatgpt.com)

En SENIALAB se dispone ya de esta cadena con **datos reales** de beacons ambientales; para este TFG, cuando no haya balizas disponibles, se utilizarán **datos simulados** con el mismo contrato y frecuencia.

**Figura sugerida 4.1-E.** *Diagrama de actividad* con tiempos objetivo por tramo (uplink→persistencia ≤ 1–2 s; persistencia→ruta ≤ 0.5–1 s).

### Modelo de comunicación

**Edge→Nube**: **LoRaWAN** (LPWA, bajo consumo, alcance campus/edificio) con **TTN** como *network server*; alternativa BLE→*gateway* local en escenarios sin LoRa. [LoRa Alliance®+1](https://lora-alliance.org/about-lorawan-old/?utm_source=chatgpt.com)

**Ingesta interna**: HTTP(S) **webhooks** de TTN hacia *middleware* Python. [The Things Industries](https://www.thethingsindustries.com/docs/integrations/webhooks/?utm_source=chatgpt.com)

**Mensajería interna opcional**: **MQTT** para desacoplar productores/consumidores si la tasa de eventos crece (broker ligero tipo Mosquitto). [MQTT+1](https://mqtt.org/?utm_source=chatgpt.com)

**Exposición a clientes**: **REST**/WebSocket (FastAPI) con esquemas JSON versionados. [fastapi.tiangolo.com](https://fastapi.tiangolo.com/?utm_source=chatgpt.com)

**Tabla sugerida 4.1-T2.** *Esquemas JSON*: Reading, CellSecurityUpdate, GraphSnapshot, RouteRequest/Response.

### Decisiones de diseño

**Persistencia y vistas**: el grafo se materializa en **vistas** sobre tablas IndoorGML para evitar duplicidades y permitir recalcular pesos con cada actualización de seguridad; índices espaciales y por atributos garantizan consultas sub-segundo. [PostGIS](https://postgis.net/workshops/postgis-intro/introduction.html?utm_source=chatgpt.com)

**Desacoplo por eventos**: *webhooks* de TTN activan ingesta inmediata; si el volumen crece, una **cola/MQTT** amortigua picos (*backpressure*). [The Things Industries+1](https://www.thethingsindustries.com/docs/integrations/webhooks/?utm_source=chatgpt.com)

**Política de *thresholds***: definición de umbral de transitabilidad por celda; degradación controlada cuando faltan datos (persistir última observación válida X minutos y marcar “degradado”).

**Caché en memoria** del grafo para acelerar rutas bajo alta concurrencia; invalidación por versión de *snapshot*.

**Figura sugerida 4.1-F.** *Patrones de consistencia*: “snapshot transaccional” para lecturas de grafo en el servicio de rutas.

### Escalabilidad, disponibilidad y tolerancia a fallos

**BD** con replicación y copias en caliente (HA); *retries* exponenciales y **circuit breakers** en servicios.

**Degradación elegante**: si falla la ingesta o la seguridad no está disponible, el servicio vuelve a **rutas estáticas** (solo longitud) y etiqueta la recomendación como “degradada”.

**Observabilidad**: *logging* correlacionado (IDs de lectura/ruta), métricas de latencia (p50/p95/p99) y *tracing* distribuido.

**Tabla sugerida 4.1-T3.** *SLO propuestos*: ingestión < 2 s (p95), cálculo de ruta < 500 ms (p95), disponibilidad anual ≥ 99.5 %.

### Seguridad y privacidad

**Minimización** de datos de localización (granularidad por *cell*, no por coordenada exacta), **retención** limitada y **seudonimización** de identificadores de dispositivo.

**Cifrado** extremo a extremo (LoRaWAN/TTN→HTTPS) y **RBAC** en API; segregación de datos de identificación frente a datos operativos.

**Figura sugerida 4.1-G.** *Diagrama de flujo de datos personales* (DPIA breve) con puntos de control.

### Mapeo a estándares y compatibilidad

El **núcleo de datos** adopta IndoorGML 1.1 (clases *CellSpace*, *State*, *Transition* y grafo de accesibilidad). La arquitectura prevé la **adopción progresiva** de **IndoorGML 2.0 (Part 1)**, manteniendo las mismas entidades conceptuales y permitiendo explorar **codificaciones SQL/JSON** cuando el *Part 2 – Encoding* se estabilice, sin reescribir la capa de negocio. [OGC Documentos Públicos+2OGC Documentos Públicos+2](https://docs.ogc.org/is/19-011r4/19-011r4.html?utm_source=chatgpt.com)

**Tabla sugerida 4.1-T4.** *Matriz de conformidad*: IndoorGML→Tablas/Vistas (qué se implementa hoy, qué queda en backlog).

### *Roadmap* técnico

**Simulación → piloto real**: sustituir generador de lecturas por **balizas LoRaWAN** ya registradas en TTN (inventario disponible) y activar *webhook* hacia el *middleware* del TFG.

Red LoRaWAN 2793f3b1237e806ebb5…

**Validación en campo**: pruebas de humo/temperatura controladas por zonas, verificación de latencia extremo a extremo y calidad de rutas.

**Escalado funcional**: multi-planta/3D, reglas CEP más ricas (difusión por conectividad IndoorGML), y personalización por accesibilidad.

**Operación**: endurecer HA en BD, brokers MQTT si aumenta la tasa de eventos, y mejoras de observabilidad.

**Figura sugerida 4.1-H.** *Hoja de ruta*: cronograma por hitos (piloto TTN, validación, 3D, operación).

## Modelado relacional IndoorGML en PostgreSQL

En esta sección documento cómo he llevado el *core* de IndoorGML (1.1) y el módulo de **Navegación** a un esquema objeto-relacional en PostgreSQL/PostGIS, y cómo ese esquema se operacionaliza para generar y mantener el **grafo navegable** que consume el recomendador. Indico, además, cómo quedará preparada la migración a **IndoorGML 2.0 (Part 1 – Conceptual Model)** y a sus futuras codificaciones SQL/JSON.

### Mapeo del core IndoorGML

#### Clases conceptuales a tablas relacionales

La Tabla 4-1 resume el mapeo que implemento entre las clases del **Core** y **Navigation** de IndoorGML y mi modelo relacional:

# Resultados y Conclusión

# Bibliografía

1. **Jiyeong Lee, Ki-Joune Li, Sisi Zlatanova, Thomas H. Kolbe, Claus Nagel, Thomas Becker, Hye-Young Kang.** *OGC® IndoorGML 1.1.* Open Geospatial Consortium (OGC). Wayland, MA : Open Geospatial Consortium (OGC)., 2020 (fecha de publicación oficial). 19-011r4..

2. **Sisi Zlatanova (Editor), Abdoulaye Diakite (Editor), Taehoon Kim (Editor), Ki-Joune Li (Editor).** *OGC IndoorGML 2.0 Part 1 – Conceptual Model.* Wayland, MA : Open Geospatial Consortium (OGC), 2025 (fecha de publicación oficial: 26 de junio de 2025). 22-045r5.

# Anexos