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](#_Toc209857743)

[2. Resumen 5](#_Toc209857744)

[3. Introducción 6](#_Toc209857745)

[4. Objetivos 7](#_Toc209857746)

[5. Solución Técnica 8](#_Toc209857747)

[5.1. Diseño de la arquitectura distribuida\*\* (alto nivel) 8](#_Toc209857748)

[5.1.1. Visión general y principios 8](#_Toc209857749)

[5.1.2. Componentes 8](#_Toc209857750)

[5.1.3. Flijo de datos y actualización en tiempo real 8](#_Toc209857751)

[5.1.4. Modelo de comunicación 8](#_Toc209857752)

[5.1.5. Decisiones de diseño 8](#_Toc209857753)

[5.1.6. Escalabilidad, disponibilidad y tolerancia a fallos 8](#_Toc209857754)

[5.1.7. Seguridad y privacidad 8](#_Toc209857755)

[5.1.8. Mapeo a estándares y compatibilidad 8](#_Toc209857756)

[5.1.9. *Roadmap* técnico 8](#_Toc209857757)

[5.2. Diseño e implementación del modelo de datos IndoorGML en PostgreSQL/PostGIS 9](#_Toc209857758)

[5.2.1. Mapeo del core IndoorGML 9](#_Toc209857759)

[5.2.2. Requisitos a modelo conceptual 9](#_Toc209857760)

[5.2.3. Modelo lógico: Mapeo IndoorGML 2.0 a relacional (PostgreSQL) 9](#_Toc209857761)

[5.2.4. Modelo físico 9](#_Toc209857762)

[5.2.5. Automatización y consistencia (funciones y triggers) 9](#_Toc209857763)

[5.2.6. Flujo geométrico de poligonos a grafo navegable DOOR TO DOOR\*\* NO ES ASI 9](#_Toc209857764)

[5.2.7. Rendimiento y escalabilidad 9](#_Toc209857765)

[5.2.8. Integración con herramientas (QGIS/PostGIS) 9](#_Toc209857766)

[5.2.9. Conformidad con Indoor 2.0 (y compatibilidad 1.1) 9](#_Toc209857767)

[5.2.10. Lecciones y decisioes de diseño 9](#_Toc209857768)

[5.2.11. Resumen operativo del SQL implementado 9](#_Toc209857769)

[5.2.12. Trabajo futuro inmediato (en linea con 2.0 part 1 y literartura) 9](#_Toc209857770)

[5.3. Ingesta de sensores y Complex Event Processing (CEP) 10](#_Toc209857771)

[5.3.1. Proposito y alcance 10](#_Toc209857772)

[5.3.2. Fuentes de datos y modelo de lectura 10](#_Toc209857773)

[5.3.3. Persistencia y unión con IndoorGML 10](#_Toc209857774)

[5.3.4. Motor CEP: Lecturas a cell\_score 10](#_Toc209857775)

[5.3.5. Conexión con el grafo IndoorGMl (actualización en tiempo real) 10](#_Toc209857776)

[5.3.6. Simulador de datos (estado actual del TFG) 10](#_Toc209857777)

[5.3.7. Operación: latencias, backpressure y calidad de datos (ESTO NO) 10](#_Toc209857778)

[5.3.8. Seguridad y privacidad 10](#_Toc209857779)

[5.3.9. Observabilidad y replay 11](#_Toc209857780)

[5.3.10. Ruta de producción 11](#_Toc209857781)

[5.3.11. Relación con arquitectura CDA/URJC 11](#_Toc209857782)

[5.3.12. Limitaciones y trabajo futuro 11](#_Toc209857783)

[5.4. Algoritmo de recomendación de rutas 11](#_Toc209857784)

[5.4.1. Modelo d egrafo y relación co IndoorGML 11](#_Toc209857785)

[5.4.2. Métrica de coste y objetivos (tiempo, seguridad, redundancia) 11](#_Toc209857786)

[5.4.3. Filtro previo por seguridad y por movilidad 11](#_Toc209857787)

[5.4.4. Rutas base: Dijkstra/A\* y salidas multiples 11](#_Toc209857788)

[5.4.5. Heuristica de robustez (redundancia bajo fallos) 11](#_Toc209857789)

[5.4.6. Personalización por perfil de usuario y contexto 11](#_Toc209857790)

[5.4.7. Actualización dinámica y re-planificación 11](#_Toc209857791)

[5.4.8. Salida del recomendador 11](#_Toc209857792)

[5.4.9. Validación empírica (enlazado con 5.6) 11](#_Toc209857793)

[6. Resultados y Conclusión 12](#_Toc209857794)

[7. Bibliografía 13](#_Toc209857795)

[8. Anexos 14](#_Toc209857796)

(Tabla de figuras o tablas)

# Resumen

# Introducción

# Objetivos

# Solución Técnica

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

### Visión general y principios

### Componentes

#### Capa IoT/AmI (edge)

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

#### Servicio de grafo/rutas

#### API del sistema

#### Clientes y actuadores

### Flijo de datos y actualización en tiempo real

### Modelo de comunicación

### Decisiones de diseño

### Escalabilidad, disponibilidad y tolerancia a fallos

### Seguridad y privacidad

### Mapeo a estándares y compatibilidad

### *Roadmap* técnico

## Diseño e implementación del modelo de datos IndoorGML en PostgreSQL/PostGIS

Aquí se documenta y explica la forma en que se ha materializado el modelo IndoorGML en PosstgreSLQ, además de eso se comenta como se generan y mantienen nodes y edges para la navegación (o cualquier otro uso basado en la localización). Como ya se ha mencionado este modelo se alinea con IndoorGML tanto sobretodo IndoorGML 2.0 (Part 1 – Conceptual Model).

Nota: Hay algunas decisiones que no se alinean exactamente con el modelo concpetual, pero como esta en proceso me he tomado la libertad de realizar algún cambio simplemente por facilida a la hora de modelar la base de datos. Como cambiar nombres o nomenclaturas

PRIMAL(CELDAS) ->DUAL(NODOS)

### Objetivos y criterio de diseño

Objetivo funcional es representar el interior de edificios como un espacio celular y grafo de navegación.

Para esto se deben cumplir ciertos criterios:

(MIRAR DOCUMETNACION INDOOR GML AHÍ ESTA TODO)

UML DEL CORE

### Requisitos a modelo conceptual

Estos requisitos propuestos por ellos definen que conforma y como deben estar las entidades dentro del sistema. Gracias a ellos se pasa a modelar el la parte conceptual

(DOCUMENTACION INDOORGML)

### Modelo lógico: Mapeo IndoorGML 2.0 a relacional (PostgreSQL)

(AQUÍ UNA TABLA DE COCEPTO IDFOORGML VS IMPLEMENTACION EN POSTGRE)

Basicamente como he implementado este modelo de forma lógica, es decir que atributos son importantes.

### Modelo físico

Aquí es donde se detallan las decisiones importantes en el modelo, es donde mayor diferencia hay, por ejemplo:

Las celdas, se modelan en 2D, pero con una Z puesta solo por si acaso en un futuro se pasa a 3D.

Se modela las condiciones y restricciones como la de NO SOLAPE com funciones internas y triggers dentro de la base de datos.

Los índices y las vistas para que quede claro el contenido de código completo de base de datos.

### Automatización y consistencia (funciones y triggers)

Aquí explico el proceso con el que me he peleado tanto, que se automatice la creación del CORE del modelo a partir de las geometris de las celdas.

Se genera todo solo, nodos, edges, boudaries, todo de forma automática.

Menos la navegación, esa se elige a posteriore por uno mismo.

Es cierto que habría que optimizar pero funciona.

### Rendimiento y escalabilidad

Existen índices compuestos, existes procesos que pueden parecer extraños, CONSTRAINS TRIGGER DEFERRABLE que tiene si función, es decir tiene sentido que este así, porqu asu lo modele, pero no mucho mas allá de que estén ahí no es una decisión de diseño, sino que necesitba y eso existe entonces vino bien.

El rendimiento no es mucho la verdad pero cumple con lo que queria

### Integración con herramientas (QGIS/PostGIS)

Es lo que he usado para visualizar las geometrías, funciona muy bien y se conecta con PostgreSQL.

### Conformidad con Indoor 2.0 (y compatibilidad 1.1)

### Lecciones y decisioes de diseño

### Resumen operativo del SQL implementado

### Trabajo futuro inmediato (en linea con 2.0 part 1 y literartura) B

**# 4. Solución técnica / Resultados DEFINITIVO**

**## 5.5 Conexión BD ↔ motor de recomendación**

\* **\*\*5.5.1\*\*** Contratos de datos (JSON) y endpoints (API REST/WS).

\* **\*\*5.5.2\*\*** Consulta del grafo vía vistas; consistencia transaccional (*\*snapshot\**).

\* **\*\*5.5.3\*\*** Estrategias *\*pull\**/*\*push\** y cacheado.

\* **\*\*5.5.4\*\*** Gestión de errores y *\*fallbacks\**.

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

### Proposito y alcance

### Fuentes de datos y modelo de lectura

#### Sensorica y transporte

#### Contrato de evento (payload normalizado)

### Persistencia y unión con IndoorGML

### Motor CEP: Lecturas a cell\_score

#### Ventanas y noción de tiempo

#### Reglas base (Transparentes y auditables)

#### Propagación espacial y reglas compuestas (opcional)

### Conexión con el grafo IndoorGMl (actualización en tiempo real)

### Simulador de datos (estado actual del TFG)

### Operación: latencias, backpressure y calidad de datos (ESTO NO)

### Seguridad y privacidad

### Observabilidad y replay

### Ruta de producción

### Relación con arquitectura CDA/URJC

### Limitaciones y trabajo futuro

## Algoritmo de recomendación de rutas

### Modelo d egrafo y relación co IndoorGML

### Métrica de coste y objetivos (tiempo, seguridad, redundancia)

### Filtro previo por seguridad y por movilidad

### Rutas base: Dijkstra/A\* y salidas multiples

### Heuristica de robustez (redundancia bajo fallos)

### Personalización por perfil de usuario y contexto

### Actualización dinámica y re-planificación

### Salida del recomendador

### Validación empírica (enlazado con 5.6)

#### Implementación (resumen técnico reproducible)

# 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