



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática



TFG del Grado en Ingeniería Informática

**Geoindoor
Documentación Técnica**



Presentado por Juan Pedro Pascual Vitores
en Universidad de Burgos — 5 de enero de 2018

Tutor: Carlos López Nozal

Índice general

Índice general	I
Índice de figuras	III
Índice de tablas	V
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	2
A.3. Estudio de viabilidad	5
Apéndice B Especificación de Requisitos	8
B.1. Introducción	8
B.2. Objetivos generales	9
B.3. Catalogo de requisitos	9
B.4. Especificación de requisitos	11
Apéndice C Especificación de diseño	70
C.1. Introducción	70
C.2. Diseño de datos	70
C.3. Diseño procedimental	72
C.4. Diseño arquitectónico	77
Apéndice D Documentación técnica de programación	81
D.1. Introducción	81
D.2. Estructura de directorios	81
D.3. Manual del programador	83
D.4. Compilación, instalación y ejecución del proyecto	88
D.5. Pruebas del sistema	91

Apéndice E Documentación de usuario	96
E.1. Introducción	96
E.2. Requisitos de usuarios	96
E.3. Instalación	96
E.4. Manual del usuario	96
Bibliografía	97

Índice de figuras

A.1. Gráfica Sprint 0	3
A.2. Gráfica Sprint 1	3
A.3. Gráfica Sprint 2	4
A.4. Gráfica Sprint 3	4
B.1. Gestión de edificio	13
B.2. Gestión de pois	14
B.3. Gestión de rutas	15
B.4. Visualización de edificios	16
B.5. Visualización de pois	16
B.6. Visualización de rutas	17
B.7. Diagrama de caso de uso	18
C.1. Diagrama de relacion de entidades	71
C.2. Diagrama de secuencia de la creación de un edificio	73
C.3. Diagrama de secuencia de la creación de una ruta predefinida	74
C.4. Diagrama de secuencia del dibujado de una ruta predefinida al seleccionarla	75
C.5. Diagrama de secuencia del borrado de una ruta	76
C.6. Diagrama de secuencia del dibujado de una ruta (Viewer)	77
C.7. Diagrama Modelo-Vista-Controlador [?]	78
C.8. Diagrama de flujo Modelo-Vista-Controlador [?]	78
C.9. Elementos integrantes del Modelo-Vista-Controlador en Architect .	79
C.10.Elementos integrantes del Modelo-Vista-Controlador en Viewer . .	80
C.11.Diagrama de navegabilidad de Geoindoor	80
D.1. Esquema de directorios Geoindoor	83
D.2. Eclipse Jee Neon	85
D.3. Terminal de bash de Git	85
D.4. Git GUI	86

D.5. Git en PowerShell	86
D.6. Comandos “node” y “npm” una vez instalado Node.js.	86
D.7. Comando “bower -v” una vez instalado Bower.	86
D.8. Comando “grunt” una vez instalado Grunt.	87
D.9. Comando “python -V” una vez instalado Python.	87
D.10. Diagrama de despliegue Geoindoor.	88
D.11. Heroku dashboard.	89
D.12. Página de inicio de la consola Firebase.	89
D.13. Panel de control Firebase.	90
D.14. Url de Geoindoor al lanzarla.	91
D.15. Test lanzado en Firefox a través de Selenium.	93
D.16. Resultados por usuario.	94
D.17. Ancho de banda del servidor y del usuario.	94
D.18. Apartado “Test Type” en Webserver Stress Tool.	95

Índice de tablas

A.1. Sueldos y salarios	5
A.2. Gatos fijos	6
A.3. Desglose gastos fijos	6
A.4. Beneficios - Gastos	6
B.1. CU-01 Gestión de edificio	20
B.2. CU-02 Emplazar edificio	21
B.3. CU-03 Detallar información de edificio	23
B.4. CU-04 Emplazar plano	25
B.5. CU-05 Redimensionar y mover plano	27
B.6. CU-06 Borrar edificio	28
B.7. CU-07 Gestión de pois	30
B.8. CU-08 Emplazamiento de pois	32
B.9. CU-09 Detallar información de pois	34
B.10.CU-10 Ubicar pois	36
B.11.CU-11 Enlazar pois	38
B.12.CU-12 Gestión de rutas	40
B.13.CU-13 Crear ruta	42
B.14.CU-14 Detallar información de la ruta	44
B.15.CU-15 Añadir pois a la ruta	46
B.16.CU-16 Trazar ruta	48
B.17.CU-17 Seleccionar ruta	49
B.18.CU-18 Limpiar ruta	51
B.19.CU-19 Borrar ruta	53
B.20.CU-20 Visualización de edificios	54
B.21.CU-21 Buscar edificio	56
B.22.CU-22 Visualizar información	57
B.23.CU-23 Visualización de pois	59
B.24.CU-24 Visualizar información	60
B.25.CU-25 Marcar un poi	61

<i>Índice de tablas</i>	VI
-------------------------	----

B.26.CU-26 Proyectar camino de forma automática	63
B.27.CU-27 Visualización de rutas	64
B.28.CU-28 Seleccionar ruta	66
B.29.CU-29 Mostrar ruta	67
B.30.CU-30 Limpiar ruta	69
D.1. Requisitos por test de interfaz.	93

Apéndice A

Plan de Proyecto Software

A.1. Introducción

La planificación es una parte muy importante del proyecto, ya que es la parte donde se estima el tiempo que se va consumir, el esfuerzo, y si ese proyecto va dar beneficios económicos.

He de decir que a la planificación de los tiempos y del trabajo se hace fundamental si se quiere ser productivo, existen factores no considerados como, que tipo de horas se ha invertido y con que lapsus de tiempo, que afectan a la productividad y a los tiempos. Existen muchos factores que pueden alterar estos parámetros, a continuación citaré algunos.

- Horas de desarrollo después de una jornada de trabajo.
- Horas de desarrollo dispersas, por ejemplo, se desarrolla durante una hora se para durante horas y se vuelve a retomar, lo que hace que la productividad disminuya .
- Curva de aprendizaje, en este proyecto el aprendizaje ha sido fundamental y en muchas ocasiones costoso, si se junta este factor con los 2 anteriores la curva de aprendizaje crece con facilidad.
- Horas de desarrollo de poca calidad, es importante tener un lugar donde trabajar donde no se pierda la concentración y el hilo de forma continua.

En este anexo se va analizar los factores por los cuales es conveniente o no realizar un proyecto.

- Planificación temporal

- Viabilidad del proyecto
- Viabilidad económica
- Viabilidad legal

A.2. Planificación temporal

La planificación del tiempo es fundamental, ya que es muchas veces las prisas no nos permiten razonar con serenidad, y tener mucho tiempo lleva a la procrastinación. Entonces lo ideal es llevar un trabajo constante y marcado, es decir crearse horarios de trabajo e intentar cumplirlos lo mejor posible.

Se ha utilizado una metodología ágil Scrum, basada en *Sprints*, y ZenHub, para el seguimiento de issues y tareas. La plataforma GitHub y ZenHub nos permiten administrar el trabajo y las tareas de forma más dinámica y sencilla. Se ha intentado marcar *Sprints* con una fecha final determinada, pero no ha sido posible debido a las dificultades a la hora de estimar el tiempo de trabajo y que no siempre existía la posibilidad de mantener un desarrollo en el tiempo, es decir, no siempre existían horas de calidad para aplicar al desarrollo. De todas formas si que se han determinado *Sprints* con un inicio y un final, los cuales detallaremos a continuación.

Estos son los pasos que se siguieron en el desarrollo:

- Aplicar una estrategia de desarrollo incremental a través de iteraciones y revisiones.
- La duración media de los *Sprints* debe ser de una semana.
- Al finalizar cada *Sprint* se entrega una parte del producto.
- Se realiza reuniones de revisión al finalizar cada *Sprint* y se vuelve a planificar *Sprint*.
- En la planificación del *Sprint* se genera una pila de tareas.
- Estas tareas se estiman y priorizan en un tablero.
- Para monitorizar el progreso del proyecto se utiliza gráficos

Se han seguido todos estos pasos, pero como ya se ha comentado anteriormente la finalización de los *Sprints* no estaba determinada así que su final se tomará cuando se acabaron todos los issues pertenecientes al mismo. Los gráficos en muchas ocasiones no son totalmente representativos del trabajo ni el tiempo utilizado ya que en muchas ocasiones el tiempo invertido no era de calidad, o no se podía invertir tiempo.

Sprint 0 (26/3/2017 - 3/4/2017)

El *Sprint* 0 fue el primer *Sprint* que se creo y sirvió de toma de contacto, se afianzaron las ideas y se especificaron los objetivos del proyecto. Se puede decir que ha servido para afianzar ideas sobre el proyecto.

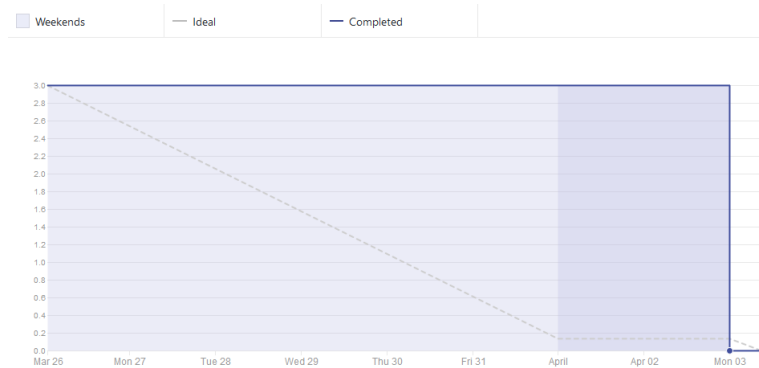


Figura A.1: Gráfica Sprint 0

Sprint 1 (10/4/2017 - 10/8/2017)

El *Sprint* 1 ha sido uno de los más importantes ya que es donde se ha centrado la mayoría del aprendizaje y una gran cantidad de desarrollo, este *Sprint* ha sido en el que se volcó por primera vez la herramienta Geoindoor en su conjunto, por lo que se crearon commits de gran tamaño.

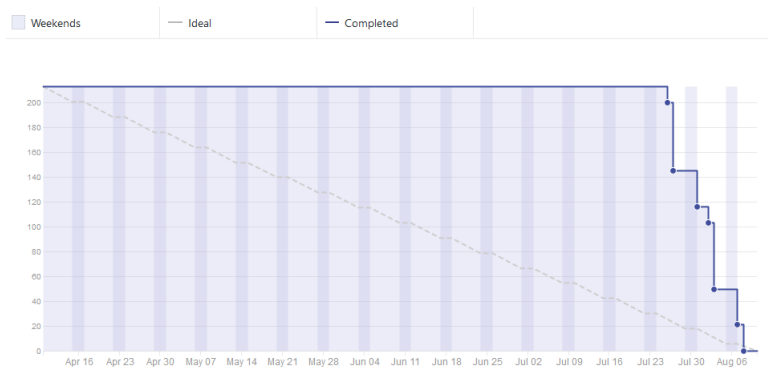


Figura A.2: Gráfica Sprint 1

Sprint 2 (10/8/2017 - 30/8/2017)

Si miramos el *Sprint* 2 podemos observar que la fase de aprendizaje está mas avanzada y que el desarrollo se hace menos costoso, pero también esta

cargado de menos tareas que el anterior. En este proyecto cabe destacar que el aprendizaje no acaba siempre se encuentra algo nuevo.

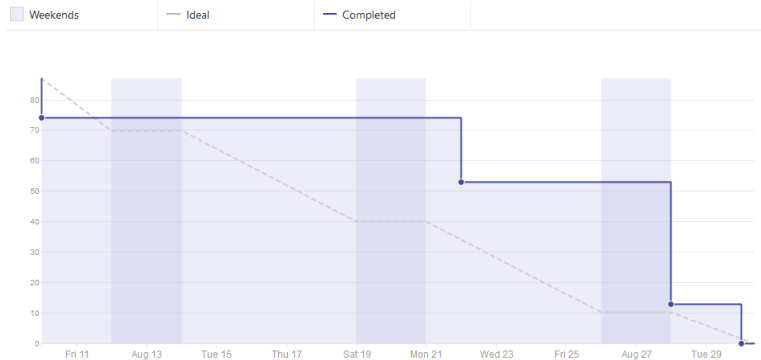


Figura A.3: Gráfica Sprint 2

Sprint 3 (25/9/2017 - 12/10/2017)

El *Sprint 3* está más dedicado al desarrollo en la parte del viewer, con lo cual se tuvo que volver a investigar y a analizar esa parte de la herramienta para entender cual era la mejor manera para introducir modificaciones. Una ventaja es que entre Architect y Viewer hay un gran parecido, por lo tanto el esfuerzo no fue tanto como la primera vez. También decir que se tuvieron que hacer cambios para que las búsquedas de edificios fueran mas efectivas, lo que llevo un gran esfuerzo.

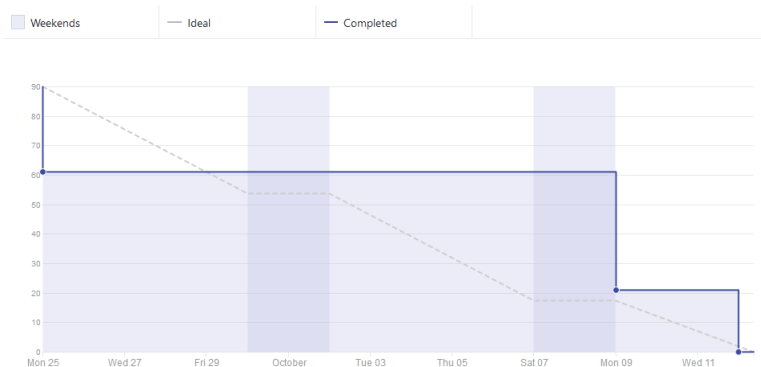


Figura A.4: Gráfica Sprint 3

Sprint 4 (17/10/2017 - *)

El *Sprint 4* es la última iteración antes de la entrega en ella se encuentran los últimos retoques a la herramienta así como, las pruebas realizadas y

elementos dedicados a la documentación. No tiene fecha final ya que abarca, como hemos dicho antes, los últimos pinceladas a la herramienta y la documentación. No encontramos gráfico por no tener fecha final.

A.3. Estudio de viabilidad

En cuanto al estudio de la viabilidad, se ha investigado y no hay mucha competencia en cuanto a geolocalización indoor basada en GPS. La mayoría de servicios de geolocalización indoor son de pago, y utilizan bacons para la localización. Lo que hace a Geoindoor uno de los sistemas más atractivos.

Viabilidad económica

Cabe destacar que este sistema en un primer momento no está pensado para obtener beneficio económico, pero se hará un análisis de la viabilidad económica basándonos en que se vende la herramienta Geoindoor a instituciones e infraestructuras como colegios, institutos, universidades, museos, centros comerciales, palacios de congreso, teatros, etc. . . .

A continuación haremos un análisis de la viabilidad económica que puede tener Geoindoor.

Para que el sistema Geoindoor sea rentable y fácil de vender se decide asociarse a un experto en marketing y publicidad, que nos ayudará a dar visibilidad al sistema, y nos servirá de contacto con los clientes para obtener beneficio a través de la venta de la herramienta a instituciones y empresas.

En cuanto a los ingresos se estima, a través de un estudio de mercado, que la venta de la herramienta generaría 60137 €.

Con esta información podemos analizar la viabilidad económica del proyecto de una forma más detallada:

**Seguridad social: Seguridad social mensual a cargo de la empresa.*

**Gasto mensual: Gasto mensual en sueldos.*

Puesto a desempeñar	Nº Trabajadores	Seguridad social	Gasto mensual
Programador	1	387€	1128,75€
Experto en Marketing	1	387€	1128,75€
Total mensuales:	2	774€	2257,50€

Tabla A.1: Sueldos y salarios

**Importe: Importe en base imponible.*

Servicios exteriores	Periodicidad	Importe	%IVA
Material de oficina	Mensual	90€	%0
Arrendamientos y cánones (alquiler)	Mensual	450€	%21
Reparaciones y conservación	Mensual	100€	%21
Servicios de profesionales independientes	Mensual	0€	%0
Primas de seguros	Mensual	118€	%0
Publicidad, propaganda y relaciones públicas	Mensual	450€	%21
Luz	Mensual	175€	%21
Teléfono	Mensual	124€	%21
Internet	Mensual	76€	%21
Otros gastos	Mensual	150€	%21
Total:		1733€	

Tabla A.2: Gatos fijos

Tipo de gasto	Mensual	Anual
Gastos de personal	3031,5€	36378€
Otros gastos de explotación	1733€	20796€
Amortización	97€	1164€
Intereses medios de préstamos	85€	1020€
Total gastos fijos:	4946,5€	59358€

Tabla A.3: Desglose gastos fijos

Año	1º año
Total beneficios:	60137€
Gastos personal	36378€
Gastos explotación	20796€
Amortización	1164€
Intereses	1020€
Total gastos:	59358€
Beneficio neto	779€

Tabla A.4: Beneficios - Gastos

Como se puede observar se obtiene un beneficio de 779€, con lo cual la conclusión, Geoindoor es un proyecto viable.

Viabilidad legal

En cuanto a la viabilidad legal AnyPlace posee una licencia MIT de código abierto lo que nos permite modificar libremente la herramienta. El uso de los mapas de Google Maps y sus facilidades también se encuentra abierto al público, así que en primera instancia, no nos encontraríamos ningún problema. Por otra parte hay un tema muy importante a tener en cuenta y es el derecho a la intimidad, y que puede afectar a la ley Orgánica 1/1982, de 5 de mayo, sobre protección civil del derecho al honor, a la intimidad personal y familiar y a la propia imagen.

A lo que nos referimos es que Geoindoor no se hace responsable de la colocación de los planos por la parte de los usuarios, pero se guarda el derecho a eliminar y borrar cualquier cuenta o plano si esta no respeta el derecho a la intimidad de la gente.

Otro tema a tener en cuenta es si se permitiría subir planos de infraestructuras críticas o edificios públicos que se pudieran utilizar con fines dañinos para la sociedad.

Teniendo todos estos factores en cuenta, se concluye con la certeza de que Geoindoor es legal pero, tiene cierto potencial que lo puede convertir en peligroso y se debe controlar.

Especificación de Requisitos

B.1. Introducción

Este anexo recoge la especificación de requisitos, la cual define el comportamiento del sistema desarrollado, es utilizado tanto por el cliente y por el equipo para tener una concepto generalizado de lo que se quiere, para que ambas partes lleguen a una idea común. Para este fin muchas veces son utilizados diagramas los cuales aclaren el concepto y lo que se necesita.

Podemos distinguir entre requisitos funcionales y requisitos no funcionales.

- **Requisito funcional:** es aquel que especifica una función o un servicio que debe cumplir un sistema.
- **Requisito no funcional:** este especifica restricciones sobre el diseño y la implementación del sistema.

Las características de una buena especificación de requisitos son definidas por [?]:

- **Completa:** deben estar todos los requerimientos y todas sus relaciones.
- **Consistente:** todos los requerimientos y otros documentos de especificación se debe relacionar de forma coherente.
- **Inequívoca:** se debe ser claro y darse a entender.
- **Correcta:** el sistema debe cumplir con todos los requisitos especificados.
- **Trazable:** los requerimientos deben de estar ordenados y organizados de una manera tal, que sea sencilla su identificación.

- **Priorizable:** los requisitos deben estar organizados de forma jerárquica, según su importancia.
- **Modificable:** los requerimientos deben ser fácilmente modificables.
- **Verificable:** se debe poder probar todo requerimiento.

B.2. Objetivos generales

- Desarrollar un sistema que permita la localización del individuo en interiores.
- Desarrollar un sistema que permita la localización de lugares en interiores.
- Desarrollar un sistema que permita el trazado de rutas en interiores.
- Desarrollar un sistema que permita mostrar lugares y rutas predefinidas en interiores.

B.3. Catalogo de requisitos

A partir de los objetivos anteriormente citados obtenemos los siguientes requisitos.

Requisitos funcionales

- **RF-1 Gestión de edificios:** la herramienta debe permitir la gestión de un edificio.
 - **RF-1.1 Emplazar edificio:** la herramienta debe permitir la colocación de un edificio sobre el plano de localización
 - **RF-1.2 Detallar información del edificio:** se debe permitir adjuntar información al edificio (nombre, descripción, etc...).
 - **RF-1.3 Emplazar plano:** la herramienta debe permitir la colocación del plano del edificio sobre el plano de localización.
 - **RF-1.4 Redimensionar y mover el plano:** se debe permitir el movimiento y redimensión del plano añadido.
 - **RF-1.5 Borrar edificio:** se debe permitir el borrado de un edificio.
- **RF-2 Gestión de pois:** la herramienta debe permitir la gestión de pois.

- **RF-2.1 Emplazamiento de pois:** se debe permitir la colocación de puntos de interés sobre el plano.
- **RF-2.2 Detallar información del poi:** se debe permitir adjuntar información al poi (nombre, descripción, tipo, etc. . .).
- **RF-2.3 Ubicar poi:** la herramienta debe permitir la colocación del poi sobre una localización del plano de un edificio.
- **RF-2.4 Enlazar pois:** se debe permitir crear un camino entre pois.
- **RF-3 Gestión de rutas:** se debe permitir la gestión de rutas predefinidas.
 - **RF-3.1 Crear ruta:** se debe permitir crear una ruta predefinida o predeterminada.
 - **RF-3.2 Detallar información de la ruta:** se debe permitir agregar información a la ruta, como el nombre o la planta (el número de planta se añade de forma automática, lo que nos permite hacer rutas multinivel).
 - **RF-3.3 Añadir pois a ruta:** la herramienta debe ser capaz de añadir puntos de interés a la ruta.
 - **RF-3.4 Trazar ruta:** la herramienta tiene que ser capaz de mostrar el trazado de la ruta.
 - **RF-3.5 Seleccionar ruta:** la herramienta debe permitir elegir entre las diferentes rutas existentes.
 - **RF-3.6 Limpiar ruta:** la herramienta debe permitir dejar de mostrar el trazado de la ruta.
 - **RF-3.7 Borrar ruta:** la herramienta debe permitir el borrado de una ruta elegida.
- **RF-4 Visualización de edificios:** el edificio debe estar disponible para todos los usuarios si el usuario administrador o creador así lo decide.
 - **RF-4.1 Buscar edificio:** la herramienta debe permitir que el edificio se pueda encontrar y visualizar a partir de su identificador o nombre.
 - **RF-4.2 Visualizar información:** la herramienta debe permitir que la información del edificio sea visible a los usuarios visitantes, por lo tanto se debe visualizar también el plano.
- **RF-5 Visualización de pois:** los pois de un edificio deben ser visibles
 - **RF-5.1 Visualizar información:** la herramienta debe permitir que la información del poi sea visible a los usuarios visitantes.

- **RF-5.2 Marcar un poi:** la herramienta debe permitir a los usuarios marcar o seleccionar un poi.
- **RF-5.3 Proyectar camino de forma automática:** la herramienta debe permitir que al seleccionar un poi se pueda dibujar un camino automático desde la ubicación del usuario hasta el poi indicado.
- **RF-6 Visualización de rutas:** la herramienta debe permitir a los usuarios visualizar las rutas predefinidas del edificio seleccionado
 - **RF-6.1 Seleccionar ruta:** la herramienta debe permitir seleccionar rutas definidas del edificio.
 - **RF-6.2 Mostrar ruta:** la herramienta debe permitir mostrar el trazado de la ruta.
 - **RF-6.3 Limpiar ruta:** la herramienta debe permitir dejar de mostrar el trazado de la ruta.

Requisitos no funcionales

- **RF-1 Intuitiva:** la herramienta debe ser intuitiva y fácil de entender para que así el usuario se acostumbre fácilmente a su uso.
- **RF-2 Usable:** la herramienta debe ser fácil de usar, para atraer tanto a gente con voluntad de aprender como no.
- **RF-3 Portable:** la herramienta debe de necesitar de instalación alguna más allá de sus dependencias lógicas (Python y navegador).
- **RF-4 Rápida:** la herramienta debe proporcionar un servicio rápido, y con resultados tempranos para que el usuario este a gusto trabajando desde el primer instante.
- **RF-5 Para todo el mundo:** la herramienta debe tener como final, que un gran espectro de la población la use para ser así mas útil para la sociedad.

B.4. Especificación de requisitos

Actores

El sistema diferencia entre dos tipos de actores.

- **Usuario de Architect:** este usuario debe de estar autenticado para poder trabajar con la herramienta, se encarga de producir información (crear edificios, rutas, pois, etc...) que será vista por los usuarios de Viewer.

- **Usuario de Viewer:** este usuario no debe de estar autenticado para poder trabajar con la herramienta, y no se encarga de producir información, si no de su visualización.

Aclaraciones

- **Geoindoor:** herramienta o aplicación web que esta formada por dos sub aplicaciones, Architect y Viewer.
- **Architect:** herramienta o aplicación web que es utilizada por los usuarios para introducir y editar información sobre los edificios, que luego serán disponibles por los usuarios del Viewer.
- **Viewer:** herramienta o aplicación web que es utilizada por los usuarios para visualizar la información sobre los edificios (rutas, pois, etc...) introducidos a través del Architect.

Diagrama de casos de uso

Geoindoor

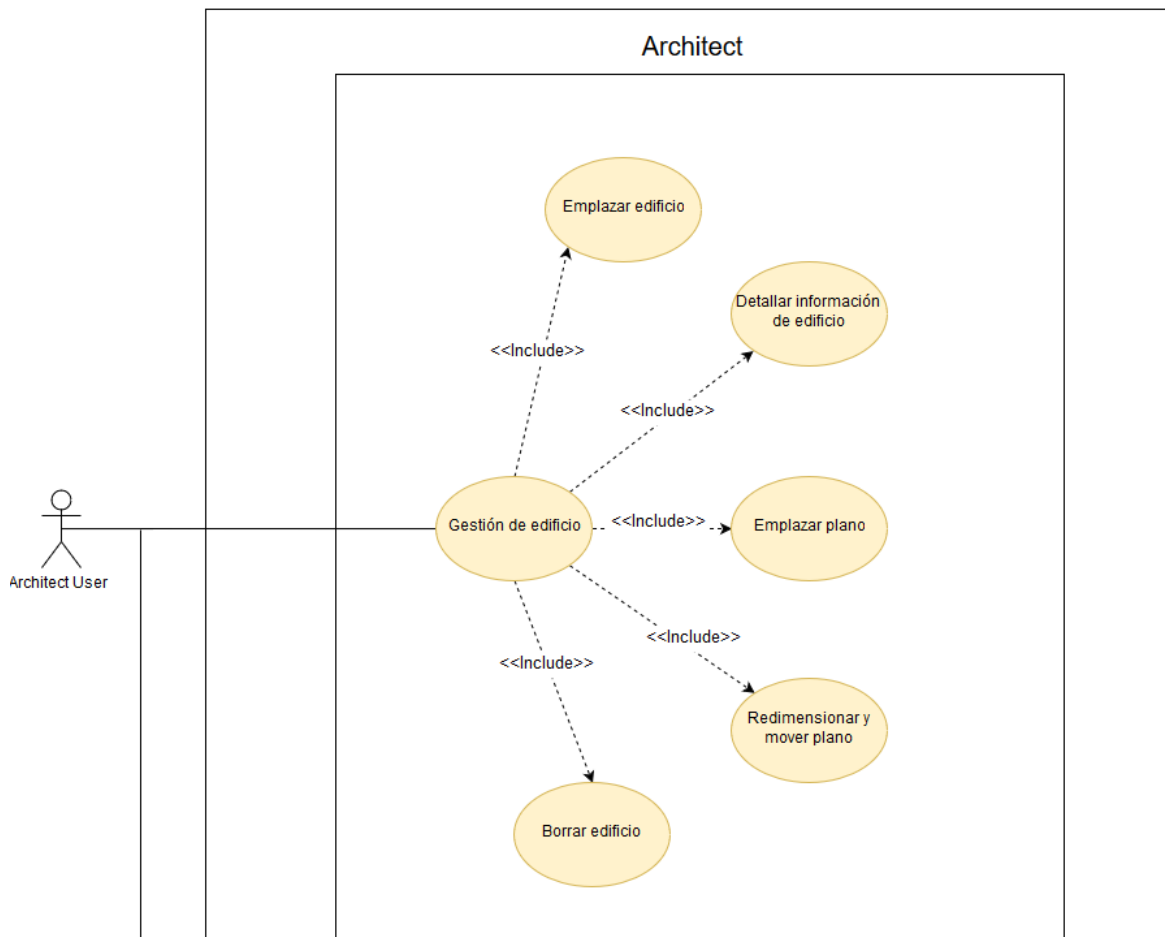


Figura B.1: Gestión de edificio

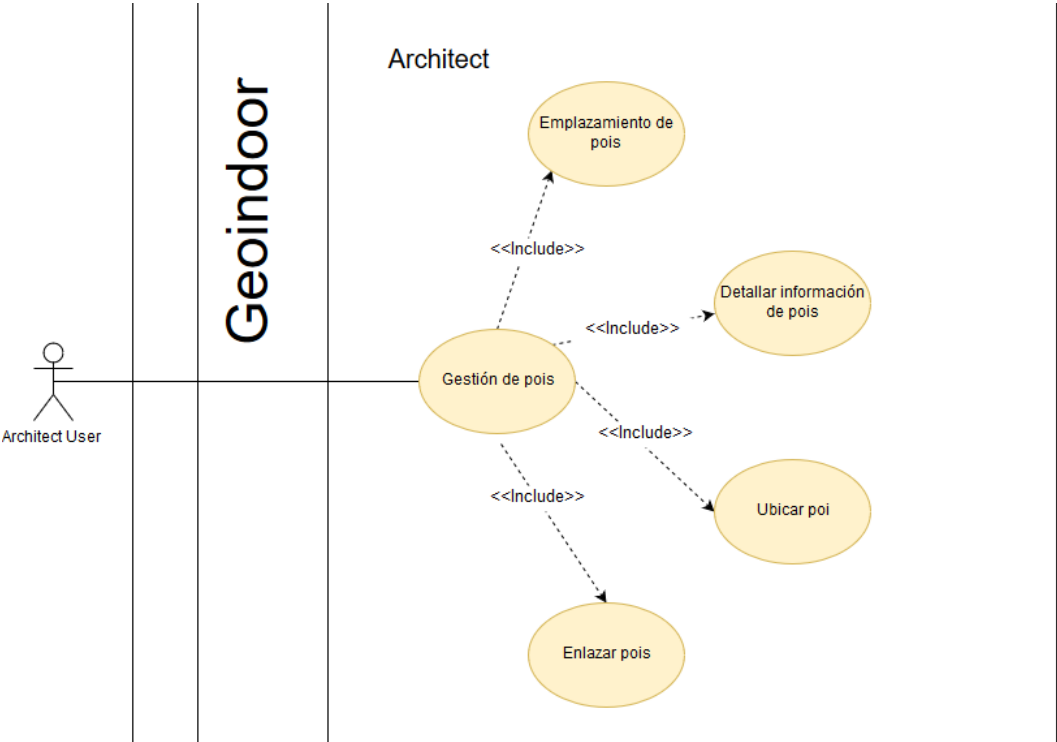


Figura B.2: Gestión de pois

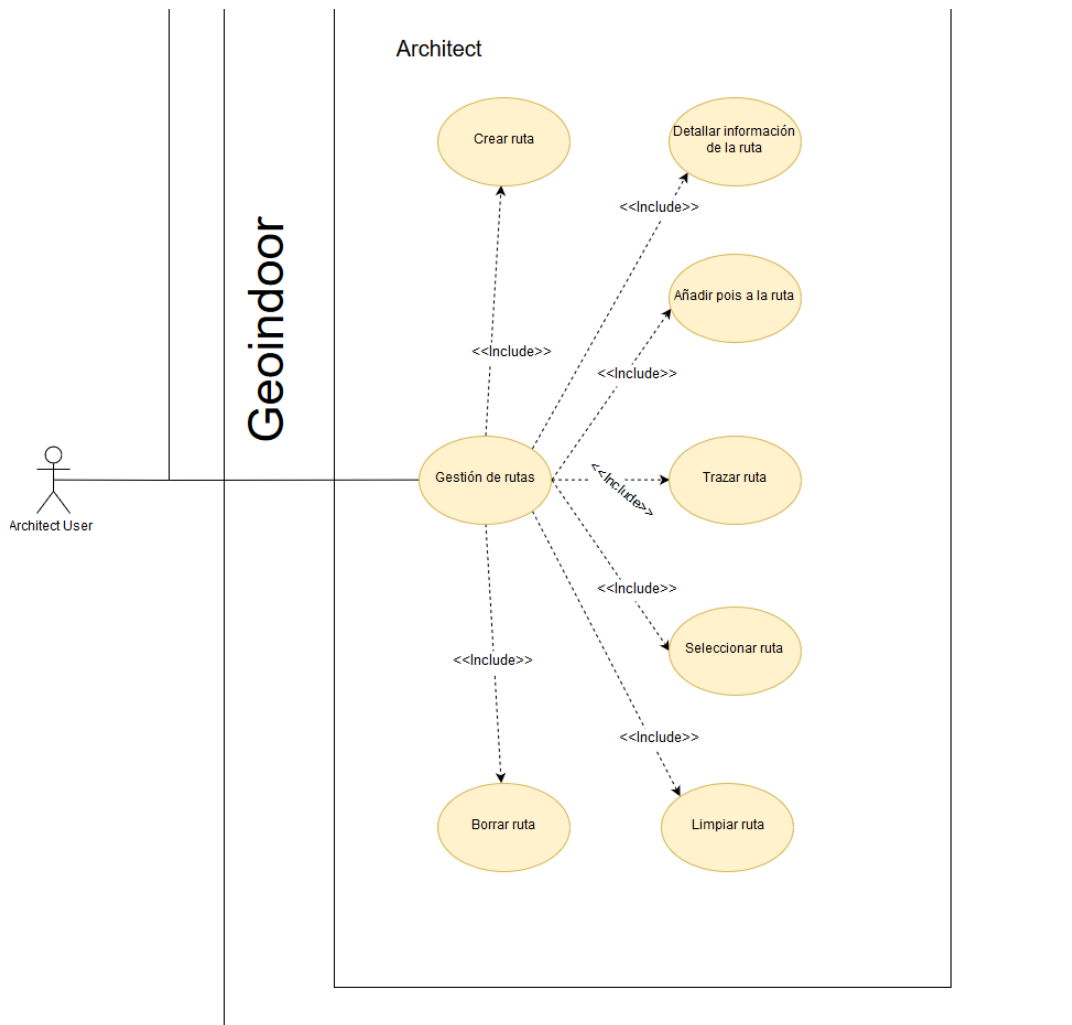


Figura B.3: Gestión de rutas

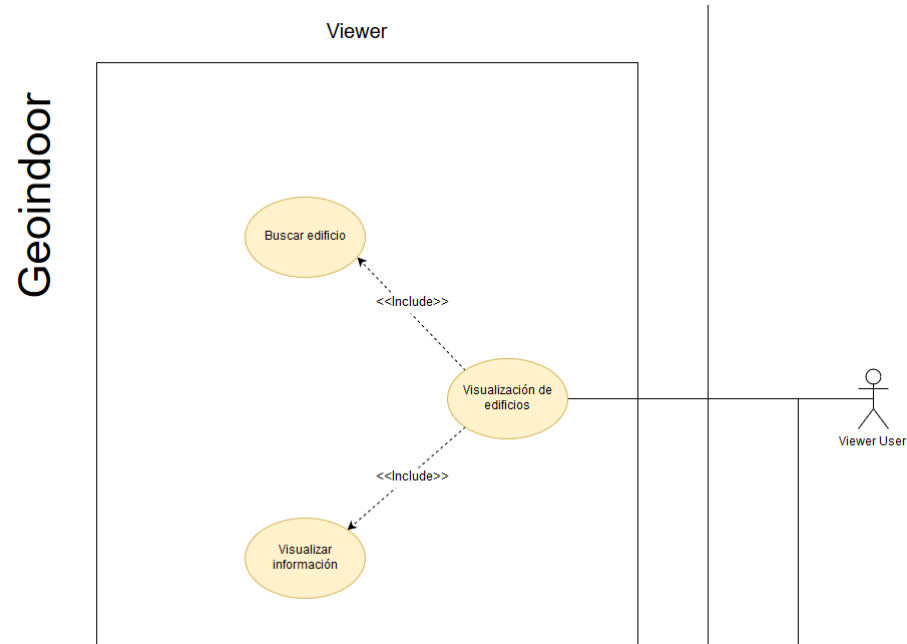


Figura B.4: Visualización de edificios

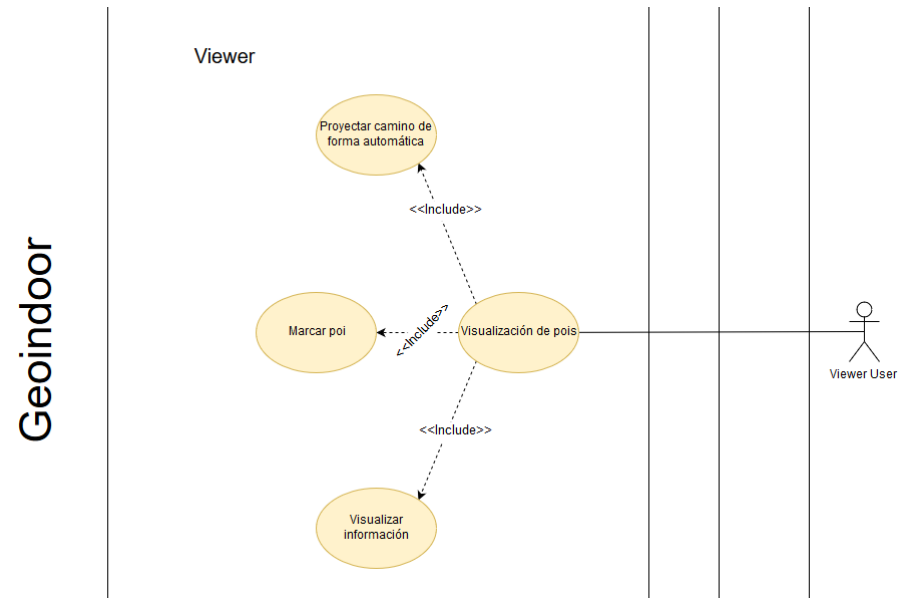


Figura B.5: Visualización de pois

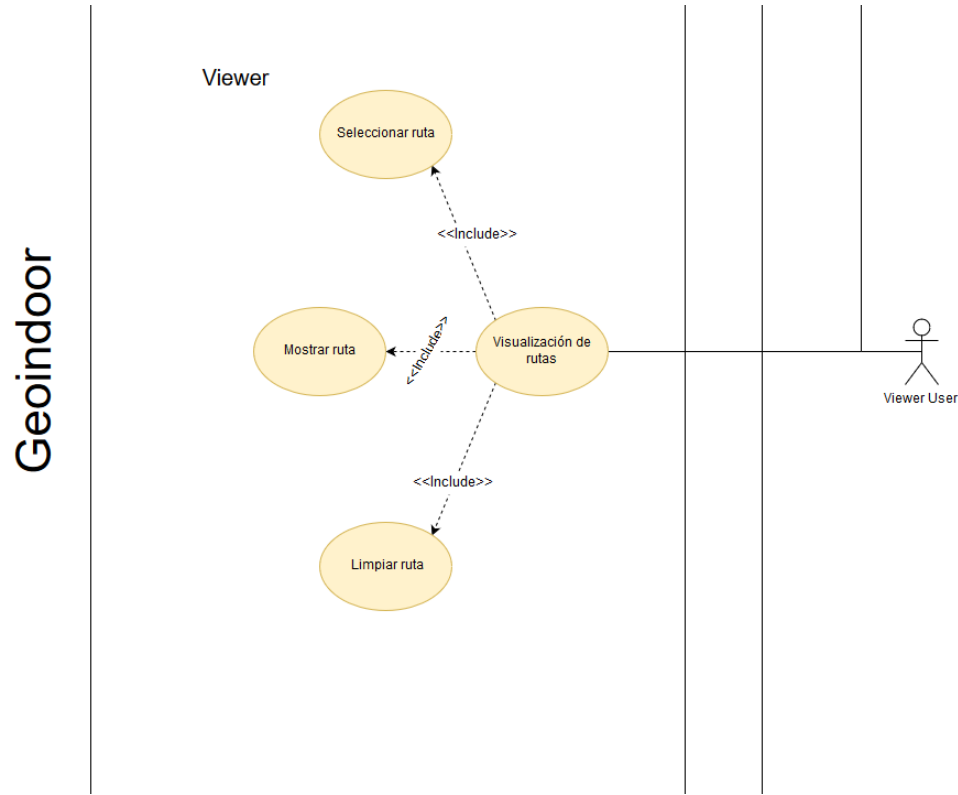


Figura B.6: Visualización de rutas

Diagrama de caso de uso (General)

Geoindoor

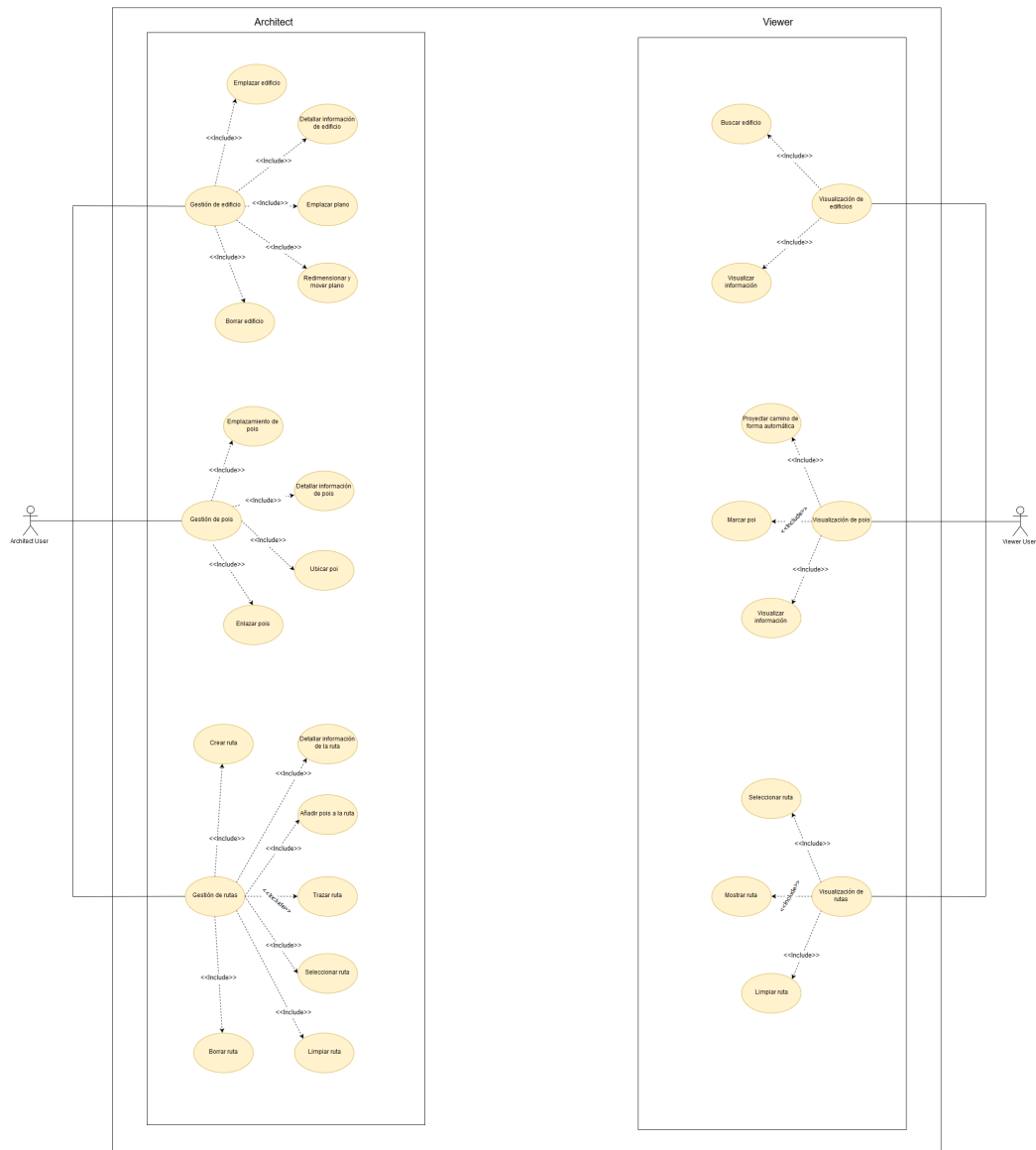


Figura B.7: Diagrama de caso de uso

Casos de uso

CU-01	Gestión de edificio
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-1, RF-1.2, RF-1.3, RF-1.4, RF-1.5
Descripción	Permite al usuario la administración y gestión de un edificio.
Precondiciones	<ul style="list-style-type: none">■ Se encuentran disponibles las bases de datos.■ Se encuentran disponibles las REST API.■ Se accede como usuario Architect (autenticado).
Acciones	<ol style="list-style-type: none">1. El usuario accede a la herramienta Architect.2. El usuario se autentica.3. El usuario interacciona con la herramienta para introducir la información concerniente a un edificio.4. Se introduce la información.
Postcondiciones	<ul style="list-style-type: none">■ La información se almacena en la base de datos.
Excepciones	<ul style="list-style-type: none">■ La bases de datos no están disponible.■ Las REST API no están disponibles.■ La información introducida no es válida

continúa en la página siguiente

continúa desde la página anterior

CU-01	Gestión de edificio
Importancia	Alta

Tabla B.1: CU-01 Gestión de edificio

CU-02	Emplazar edificio
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-1.1
Descripción	Permite al usuario la colocación de un edificio sobre el plano de localización.
Precondiciones	<ul style="list-style-type: none"> ■ Se encuentran disponibles las bases de datos. ■ Se encuentran disponibles las REST API. ■ Se accede como usuario Architect (autenticado).
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la herramienta Architect. 2. El usuario se autentica. 3. El usuario elige una posición en el plano. 4. El usuario pincha en la pestaña “Buildings” de su panel de control. 5. El usuario pincha en el botón “Add”, después en el icono del edificio y arrastra el puntero hasta una posición del plano. 6. El usuario pulsa sobre el botón “Add” del panel emergente, para añadir de forma definitiva el edificio.

continúa en la página siguiente

continúa desde la página anterior

CU-02	Emplazar edificio
Postcondiciones	<ul style="list-style-type: none"> ■ La información sobre el edificio se almacena en la base de datos. ■ La información posteriormente será visible.
Excepciones	<ul style="list-style-type: none"> ■ La bases de datos no están disponibles. ■ Las REST API no están disponibles. ■ La información introducida no es válida
Importancia	Alta

Tabla B.2: CU-02 Emplazar edificio

CU-03	Detallar información de edificio
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-1.2
Descripción	Permite al usuario adjuntar información al edificio (nombre, descripción, etc. . .).
Precondiciones	<ul style="list-style-type: none"> ■ Se encuentran disponibles las bases de datos. ■ Se encuentran disponibles las REST API. ■ Se accede como usuario Architect (autenticado).

continúa en la página siguiente

continúa desde la página anterior

CU-03	Detallar información de edificio
Acciones	<ol style="list-style-type: none">1. El usuario accede a la herramienta Architect.2. El usuario se autentica.3. El usuario elige una posición en el plano.4. El usuario pincha en la pestaña “Buildings” de su panel de control.5. El usuario pincha en el botón “Add”, después en el icono del edificio y arrastra el puntero hasta una posición del plano.6. El usuario rellena el formulario del panel emergente.7. El usuario pulsa sobre el botón “Add” del panel emergente, para añadir de forma definitiva el edificio.

continúa en la página siguiente

continúa desde la página anterior

CU-03	Detallar información de edificio
Postcondiciones	<ul style="list-style-type: none"> La información sobre el edificio se almacena en la base de datos. La información posteriormente será visible.
Excepciones	<ul style="list-style-type: none"> Las bases de datos no está disponibles. Las REST API no están disponibles. La información introducida no es válida
Importancia	Alta

Tabla B.3: CU-03 Detallar información de edificio

CU-04	Emplazar plano
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-1.3
Descripción	Permite al usuario la colocación del plano del edificio sobre el plano de localización.
Precondiciones	<ul style="list-style-type: none"> Se encuentran disponibles las bases de datos. Se encuentran disponibles las REST APIs. Se accede como usuario Architect (autenticado). Se tiene una imagen del plano o un pdf del mismo.

continúa en la página siguiente

continúa desde la página anterior

CU-04	Emplazar plano
Acciones	<ol style="list-style-type: none">1. El usuario accede a la herramienta Architect.2. El usuario se autentica.3. El usuario pincha sobre el icono de edificio que aparece en el plano de localización, si no, debe emplazar un edificio.4. El usuario pincha en la pestaña “Floors” de su panel de control.5. El usuario pincha en el text-area de “Floor Number”, para introducir a que planta pertenece el plano.6. El usuario pincha en el botón de “Floor Plan”.7. El usuario elige el plano a introducir.8. El usuario pulsa sobre el botón ✓ para emplazar definitivamente el plano.

continúa en la página siguiente

continúa desde la página anterior

CU-04	Emplazar plano
Postcondiciones	<ul style="list-style-type: none"> ▪ El plano del edificio se almacena en la base de datos. ▪ La información posteriormente será visible.
Excepciones	<ul style="list-style-type: none"> ▪ Las bases de datos no están disponibles. ▪ Las REST API no están disponibles. ▪ La información introducida no es válida
Importancia	Alta

Tabla B.4: CU-04 Emplazar plano

CU-05	Redimensionar y mover plano
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-1.4
Descripción	Permite al usuario el movimiento y redimensión del plano añadido.
Precondiciones	<ul style="list-style-type: none"> ▪ Se encuentran disponibles las bases de datos. ▪ Se encuentran disponibles las REST API. ▪ Se accede como usuario Architect (autenticado). ▪ Se tiene una imagen del plano o un pdf del mismo.

continúa en la página siguiente

continúa desde la página anterior

CU-05	Redimensionar y mover plano
Acciones	<ol style="list-style-type: none">1. El usuario accede a la herramienta Architect.2. El usuario se autentica.3. El usuario pincha sobre el icono de edificio que aparece en el plano de localización, si no, debe emplazar un edificio.4. El usuario pincha en la pestaña “Floors” de su panel de control.5. El usuario pincha en el text-area de “Floor Number”, para introducir a que planta pertenece el plano.6. El usuario pincha en el botón de “Floor Plan”.7. El usuario elige el plano a introducir.8. El usuario mueve y redimensiona la imagen a través de los iconos típicos de redimensión y giro de imagen.9. El usuario pulsa sobre el botón ✓ para emplazar definitivamente el plano.

continúa en la página siguiente

continúa desde la página anterior

CU-05	Redimensionar y mover plano
Postcondiciones	<ul style="list-style-type: none"> ▪ El plano del edificio se almacena en la base de datos. ▪ La información posteriormente será visible.
Excepciones	<ul style="list-style-type: none"> ▪ Las bases de datos no están disponible. ▪ Las REST API no están disponibles. ▪ La información introducida no es válida
Importancia	Alta

Tabla B.5: CU-05 Redimensionar y mover plano

CU-06	Borrar edificio
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-1.5
Descripción	Permite al usuario el borrado de un edificio.
Precondiciones	<ul style="list-style-type: none"> ▪ Se encuentran disponibles las bases de datos. ▪ Se encuentran disponibles las REST APIs. ▪ Se accede como usuario Architect (autenticado).

continúa en la página siguiente

continúa desde la página anterior

CU-06	Borrar edificio
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la herramienta Architect. 2. El usuario se autentica. 3. El usuario pincha sobre el icono de edificio que aparece en el plano de localización. 4. El usuario pincha en la pestaña “Buildings” de su panel de control. 5. El usuario pincha en el botón “Delete”, y posteriormente en el botón “Confirm Deletion” para borrar de forma definitiva el edificio.
Postcondiciones	<ul style="list-style-type: none"> ▪ La información del edificio posteriormente no será visible.
Excepciones	<ul style="list-style-type: none"> ▪ Las bases de datos no están disponibles. ▪ Las REST API no están disponibles.
Importancia	Alta

Tabla B.6: CU-06 Borrar edificio

CU-07	Gestión de pois
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-2, RF-2.1, RF-2.2, RF-2.3, RF-2.4
Descripción	Permite al usuario la gestión y administración de pois. <ul style="list-style-type: none">▪ Se encuentran disponibles las bases de datos.
Precondiciones	<ul style="list-style-type: none">▪ Se encuentran disponibles las REST API.▪ Se accede como usuario Architect (autenticado).
Acciones	<ol style="list-style-type: none">1. El usuario accede a la herramienta Architect.2. El usuario se autentica.3. El usuario pincha sobre el icono de edificio que aparece en el plano de localización, si no, debe emplazar un edificio y su respectivo plano.4. El usuario pincha en la pestaña “POIs” de su panel de control.5. El usuario interacciona con la herramienta para introducir la información concerniente a los pois de un edificio.

continúa en la página siguiente

continúa desde la página anterior

CU-07	Gestión de pois
Postcondiciones	<ul style="list-style-type: none"> La información sobre los pois del edificio se almacena en la base de datos. La información de los pois del edificio será visible.
Excepciones	<ul style="list-style-type: none"> Las bases de datos no están disponibles. Las REST API no está disponibles. No hay edificio donde ubicar los pois.
Importancia	Alta

Tabla B.7: CU-07 Gestión de pois

CU-08	Emplazamiento de pois
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-2.1
Descripción	Permite al usuario la colocación de puntos de interés sobre el plano.
Precondiciones	<ul style="list-style-type: none"> Se encuentran disponibles las bases de datos. Se encuentran disponibles las REST API. Se accede como usuario Architect (autenticado).

continúa en la página siguiente

continúa desde la página anterior

CU-08	Emplazamiento de pois
Acciones	<ol style="list-style-type: none">1. El usuario accede a la herramienta Architect.2. El usuario se autentica.3. El usuario pincha sobre el icono de edificio que aparece en el plano de localización, si no, debe emplazar un edificio y su respectivo plano.4. El usuario pincha en la pestaña “POIs” de su panel de control.5. El usuario pincha en el icono de “drag to add new POI” y arrastra hasta una posición del plano del edificio.
Postcondiciones	<ul style="list-style-type: none">■ Rellenar por parte del usuario, la información imprescindible del poi para su validación.■ La información sobre los pois del edificio se almacena en la base de datos.■ La información de los pois del edificio será visible.
Excepciones	<ul style="list-style-type: none">■ Las bases de datos no están disponibles.■ Las REST API no está disponibles.■ No hay edificio donde ubicar los pois.

continúa en la página siguiente

continúa desde la página anterior

CU-08	Emplazamiento de pois
Importancia	Alta

Tabla B.8: CU-08 Emplazamiento de pois

CU-09	Detallar información de pois
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-2.2
Descripción	Permite al usuario adjuntar información al poi (nombre, descripción, tipo, etc. . .). <ul style="list-style-type: none">■ Se encuentran disponibles las bases de datos.
Precondiciones	<ul style="list-style-type: none">■ Se encuentran disponibles las REST API.■ Se accede como usuario Architect (autenticado).
Acciones	<ol style="list-style-type: none">1. El usuario accede a la herramienta Architect.2. El usuario se autentica.3. El usuario pincha sobre el icono de edificio que aparece en el plano de localización, si no, debe emplazar un edificio y su respectivo plano.4. El usuario pincha en la pestaña “POIs” de su panel de control.5. El usuario pincha en el icono de “drag to add new POI” y arrastra hasta una posición del plano del edificio.6. El usuario en el panel emergente tiene varios campos que puede rellenar para adjuntar información al poi.7. El usuario pincha en el botón “Add” para añadir de forma definitiva el poi al plano y al edificio .
Postcondiciones	<ul style="list-style-type: none">■ La información sobre los pois del edificio se almacena en la base de datos.■ La información de los pois del edificio será visible.

continúa en la página siguiente

continúa desde la página anterior

CU-09	Detallar información de pois
Excepciones	<ul style="list-style-type: none"> ■ Las bases de datos no están disponibles. ■ Las REST API no está disponibles. ■ No hay edificio donde ubicar los pois.
Importancia	Alta

Tabla B.9: CU-09 Detallar información de pois

CU-10	Ubicar pois
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-2.3
Descripción	Permite al usuario la colocación del poi sobre una localización del plano de un edificio.
Precondiciones	<ul style="list-style-type: none"> ■ Se encuentran disponibles las bases de datos. ■ Se encuentran disponibles las REST API. ■ Se accede como usuario Architect (autenticado). <ol style="list-style-type: none"> 1. El usuario accede a la herramienta Architect. 2. El usuario se autentica. 3. El usuario pincha sobre el icono de edificio que aparece en el plano de localización, si no, debe emplazar un edificio y su respectivo plano.
Acciones	<ol style="list-style-type: none"> 4. El usuario pincha en la pestaña “POIs” de su panel de control. 5. El usuario pincha en el icono de “drag to add new POI” y arrastra hasta una posición del plano del edificio. 6. El usuario pincha en el poi y sin soltar lo cambia de localización, dejando de mantener el click para dejar el poi en la posición deseada.
Postcondiciones	<ul style="list-style-type: none"> ■ La información sobre los pois del edificio se almacena en la base de datos. ■ La información de los pois del edificio será visible.

continúa en la página siguiente

continúa desde la página anterior

CU-10	Ubicar pois
Excepciones	<ul style="list-style-type: none">▪ Las bases de datos no están disponibles.▪ Las REST API no está disponibles.▪ No hay edificio donde ubicar los pois.
Importancia	Alta

Tabla B.10: CU-10 Ubicar pois

CU-11	Enlazar pois
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-2.4
Descripción	Permite al usuario crear un camino entre pois.
Precondiciones	<ul style="list-style-type: none"> ▪ Se encuentran disponibles las bases de datos. ▪ Se encuentran disponibles las REST API. ▪ Se accede como usuario Architect (autenticado). ▪ Hay dos o más pois creados en un plano del edificio.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la herramienta Architect. 2. El usuario se autentica. 3. El usuario pincha sobre el icono de edificio que aparece en el plano de localización, si no, debe emplazar un edificio y su respectivo plano. 4. El usuario pincha en la pestaña “POIs” de su panel de control. 5. El usuario pincha sobre el icono de “toggle edge mode” para poner el modo camino a ON 6. El usuario pincha sobre un poi, y después pincha sobre otro para crear un camino en línea recta entre ellos. (Si se desea hacer caminos entre pois con curvas, existen pois intermedios llamados conectores que lo permiten.)
Postcondiciones	<ul style="list-style-type: none"> ▪ La información sobre los pois del edificio se almacena en la base de datos. ▪ La información de los pois del edificio será visible.

continúa en la página siguiente

continúa desde la página anterior

CU-11	Enlazar pois
	<ul style="list-style-type: none"> Las bases de datos no están disponibles.
Excepciones	<ul style="list-style-type: none"> Las REST API no está disponibles. No hay edificio donde ubicar los pois.
Importancia	Alta

Tabla B.11: CU-11 Enlazar pois

CU-12	Gestión de rutas
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-3, RF-3.1, RF-3.2, RF-3.3, RF-3.4, RF-3.5, RF-3.6, RF-3.7
Descripción	Permite al usuario la gestión y administración de rutas predefinidas.
Precondiciones	<ul style="list-style-type: none"> Se encuentran disponibles las bases de datos. Se encuentran disponibles las REST API. Se accede como usuario Architect (autenticado). Hay dos o más pois creados en un plano del edificio.

continúa en la página siguiente

continúa desde la página anterior

CU-12	Gestión de rutas
Acciones	<ol style="list-style-type: none">1. El usuario accede a la herramienta Architect.2. El usuario se autentica.3. El usuario pincha sobre el icono de edificio que aparece en el plano de localización, si no, debe emplazar un edificio y su respectivo plano.4. El usuario pincha en la pestaña “POIs” de su panel de control.5. El usuario interacciona con la herramienta para realizar la gestión de las rutas.
Postcondiciones	<ul style="list-style-type: none">■ La información sobre las rutas del edificio se almacena en la base de datos.■ La información de las rutas del edificio será visible.
Excepciones	<ul style="list-style-type: none">■ Las bases de datos no están disponibles.■ Las REST API no está disponibles.■ No hay edificio donde ubicar las rutas y pois para formarlas.

continúa en la página siguiente

continúa desde la página anterior

CU-12	Gestión de rutas
Importancia	Alta

Tabla B.12: CU-12 Gestión de rutas

CU-13	Crear ruta
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-3.1
Descripción	Permite al usuario la creación de rutas predefinidas.
Precondiciones	<ul style="list-style-type: none"> ▪ Se encuentran disponibles las bases de datos. ▪ Se encuentran disponibles las REST API. ▪ Se accede como usuario Architect (autenticado). ▪ Hay dos o más pois creados en un plano del edificio. <p>Se especifican todos los pasos necesarios para crear una ruta, teniendo en cuenta restricciones como; que el nombre de la ruta no este vacío, que una ruta tenga más de un poi, que no se repitan pois en la ruta etc. . .</p>
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la herramienta Architect. 2. El usuario se autentica. 3. El usuario pincha sobre el icono de edificio que aparece en el plano de localización, si no, debe emplazar un edificio y su respectivo plano. 4. El usuario pincha en la pestaña “POIs” de su panel de control. 5. El usuario pincha en el icono de “add route mode”. 6. El usuario rellena el campo que acaba de aparecer con el nombre de la ruta que desea crear. 7. El usuario pincha en los pois que quiere añadir a la ruta, y en el panel emergente, pulsa el botón “add POI to the route” para añadirlos. 8. El usuario pulsa el botón “Create route” para crear una ruta predefinida de forma definitiva.
Postcondiciones	<ul style="list-style-type: none"> ▪ La información sobre las rutas del edificio se almacena en la base de datos. ▪ La información de las rutas del edificio será visible.

continúa en la página siguiente

continúa desde la página anterior

CU-13	Crear ruta
Excepciones	<ul style="list-style-type: none"> ▪ Las bases de datos no están disponibles. ▪ Las REST API no está disponibles. ▪ No hay edificio donde ubicar las rutas y pois para formarlas.
Importancia	Alta

Tabla B.13: CU-13 Crear ruta

CU-14	Detallar información de la ruta
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-3.2
Descripción	Permite al usuario agregar información a la ruta, como el nombre o la planta.
Precondiciones	<ul style="list-style-type: none"> ▪ Se encuentran disponibles las bases de datos. ▪ Se encuentran disponibles las REST API. ▪ Se accede como usuario Architect (autenticado). ▪ Hay dos o más pois creados en un plano del edificio.

continúa en la página siguiente

continúa desde la página anterior

CU-14	Detallar información de la ruta
Acciones	<ol style="list-style-type: none">1. El usuario accede a la herramienta Architect.2. El usuario se autentica.3. El usuario pincha sobre el icono de edificio que aparece en el plano de localización, si no, debe emplazar un edificio y su respectivo plano.4. El usuario pincha en la pestaña “POIs” de su panel de control.5. El usuario pincha en el icono de “add route mode”.6. El usuario rellena el campo que acaba de aparecer con el nombre de la ruta que desea crear.7. El usuario pincha en los pois que quiere añadir a la ruta, y en el panel emergente, pulsa el botón “add POI to the route” para añadirlos.8. El usuario pulsa el botón “Create route” para crear una ruta predefinida de forma definitiva. En este momento es en el que se le agrega de forma automática el número de planta.

continúa en la página siguiente

continúa desde la página anterior

CU-14	Detallar información de la ruta
Postcondiciones	<ul style="list-style-type: none"> La información sobre las rutas del edificio se almacena en la base de datos. La información de las rutas del edificio será visible.
Excepciones	<ul style="list-style-type: none"> Las bases de datos no están disponibles. Las REST API no está disponibles. No hay edificio donde ubicar las rutas y pois para formarlas.
Importancia	Alta

Tabla B.14: CU-14 Detallar información de la ruta

CU-15	Añadir pois a la ruta
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-3.3
Descripción	Permite al usuario añadir puntos de interés a la ruta.
Precondiciones	<ul style="list-style-type: none"> Se encuentran disponibles las bases de datos. Se encuentran disponibles las REST API. Se accede como usuario Architect (autenticado). Hay dos o más pois creados en un plano del edificio.

continúa en la página siguiente

continúa desde la página anterior

CU-15	Añadir pois a la ruta
Acciones	<ol style="list-style-type: none">1. El usuario accede a la herramienta Architect.2. El usuario se autentica.3. El usuario pincha sobre el icono de edificio que aparece en el plano de localización, si no, debe emplazar un edificio y su respectivo plano.4. El usuario pincha en la pestaña “POIs” de su panel de control.5. El usuario pincha en el icono de “add route mode”.6. El usuario pincha en los pois que quiere añadir a la ruta, y en el panel emergente, pulsa el botón “add POI to the route” para añadirlos.
Postcondiciones	<ul style="list-style-type: none">■ La información sobre las rutas del edificio se almacena en la base de datos.■ La información de las rutas del edificio será visible.
Excepciones	<ul style="list-style-type: none">■ Las bases de datos no están disponibles.■ Las REST API no está disponibles.■ No hay edificio donde ubicar las rutas y pois para formarlas.

continúa en la página siguiente

continúa desde la página anterior

CU-15	Añadir pois a la ruta
Importancia	Alta

Tabla B.15: CU-15 Añadir pois a la ruta

CU-16	Trazar ruta
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-3.4
Descripción	Permite al usuario ver el trazado de la ruta que está creando.
Precondiciones	<ul style="list-style-type: none"> ■ Se encuentran disponibles las bases de datos. ■ Se encuentran disponibles las REST API. ■ Se accede como usuario Architect (autenticado). ■ Hay dos o más pois creados en un plano del edificio.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la herramienta Architect. 2. El usuario se autentica. 3. El usuario pincha sobre el icono de edificio que aparece en el plano de localización, si no, debe emplazar un edificio y su respectivo plano. 4. El usuario pincha en la pestaña “POIs” de su panel de control. 5. El usuario pincha en el icono de “add route mode”. 6. El usuario pincha en los pois que quiere añadir a la ruta, y en el panel emergente, pulsa el botón “add POI to the route” para añadirlos. Cada vez que el usuario añada un poi se irá mostrando el trazado de la ruta.
Postcondiciones	<ul style="list-style-type: none"> ■ La información sobre las rutas del edificio se almacena en la base de datos. ■ La información de las rutas del edificio será visible.

continúa en la página siguiente

continúa desde la página anterior

CU-16	Trazar ruta
Excepciones	<ul style="list-style-type: none"> ▪ Las bases de datos no están disponibles. ▪ Las REST API no está disponibles. ▪ No hay edificio donde ubicar las rutas y pois para formarlas.
Importancia	Alta

Tabla B.16: CU-16 Trazar ruta

CU-17	Seleccionar ruta
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-3.5
Descripción	Permite al usuario elegir entre las diferentes rutas existentes, ya sea para mostrar su trazado o modificarlo.
Precondiciones	<ul style="list-style-type: none"> ▪ Se encuentran disponibles las bases de datos. ▪ Se encuentran disponibles las REST API. ▪ Se accede como usuario Architect (autenticado). ▪ Hay rutas creadas en el edificio.

continúa en la página siguiente

continúa desde la página anterior

CU-17	Seleccionar ruta
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la herramienta Architect. 2. El usuario se autentica. 3. El usuario pincha sobre el icono de edificio que aparece en el plano de localización, si no, debe emplazar un edificio, su respectivo plano y crear al menos una ruta predefinida. 4. El usuario pincha en el desplegable que se encuentra arriba a la izquierda, y selecciona una ruta, se mostrará su trazado. Para editarla se debe ir a la pestaña “POIs” del panel de control, activar el botón “add route modez ya se permitiría su edición.
Postcondiciones	<ul style="list-style-type: none"> ▪ La información sobre las rutas del edificio se almacena en la base de datos, si se edita la ruta. ▪ La información de las rutas del edificio será visible.
Excepciones	<ul style="list-style-type: none"> ▪ Las bases de datos no están disponibles. ▪ Las REST API no está disponibles.
Importancia	Alta

Tabla B.17: CU-17 Seleccionar ruta

CU-18	Limpiar ruta
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-3.6

continúa en la página siguiente

continúa desde la página anterior

CU-18	Limpiar ruta
Descripción	Permite al usuario dejar de ver el trazado de la ruta.
Precondiciones	<ul style="list-style-type: none">▪ Se encuentran disponibles las bases de datos.▪ Se encuentran disponibles las REST API.▪ Se accede como usuario Architect (autenticado).▪ Hay rutas creadas en el edificio.
Acciones	<ol style="list-style-type: none">1. El usuario accede a la herramienta Architect.2. El usuario se autentica.3. El usuario pincha sobre el icono de edificio que aparece en el plano de localización, si no, debe emplazar un edificio, su respectivo plano y crear al menos una ruta predefinida.4. El usuario pincha en el desplegable que se encuentra arriba a la izquierda, y selecciona una ruta, se mostrará su trazado.5. El usuario pincha en el botón “Clear” que se encuentra arriba a la izquierda, para dejar de ver el trazado de la ruta seleccionada.

continúa en la página siguiente

continúa desde la página anterior

CU-18	Limpiar ruta
Postcondiciones	<ul style="list-style-type: none"> ■ No existe mayor repercusión más allá de la visual.
Excepciones	<ul style="list-style-type: none"> ■ Las bases de datos no están disponibles. ■ Las REST API no está disponibles.
Importancia	Alta

Tabla B.18: CU-18 Limpiar ruta

CU-19	Borrar ruta
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-3.7
Descripción	Permite al usuario el borrado de una ruta.
Precondiciones	<ul style="list-style-type: none"> ■ Se encuentran disponibles las bases de datos. ■ Se encuentran disponibles las REST API. ■ Se accede como usuario Architect (autenticado). ■ Hay rutas creadas en el edificio.

continúa en la página siguiente

continúa desde la página anterior

CU-19**Borrar ruta**

Acciones

1. El usuario accede a la herramienta Architect.
2. El usuario se autentica.
3. El usuario pincha sobre el icono de edificio que aparece en el plano de localización, si no, debe emplazar un edificio, su respectivo plano y crear al menos una ruta predefinida.
4. El usuario pincha en el desplegable que se encuentra arriba a la izquierda, y selecciona una ruta, se mostrará su trazado.
5. El usuario pincha en el botón “x” que se encuentra arriba a la izquierda, para borrar de forma definitiva la ruta seleccionada.

continúa en la página siguiente

continúa desde la página anterior

CU-19	Borrar ruta
Postcondiciones	<ul style="list-style-type: none"> La información de la ruta borrada no será visible.
Excepciones	<ul style="list-style-type: none"> Las bases de datos no están disponibles. Las REST API no está disponibles.
Importancia	Alta

Tabla B.19: CU-19 Borrar ruta

CU-20	Visualización de edificios
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-4, RF-4.1, RF-4.2
Descripción	Permite al usuario el borrado de una ruta.
Precondiciones	<ul style="list-style-type: none"> Se encuentran disponibles las bases de datos. Se encuentran disponibles las REST API. Se accede como usuario Viewer (no autenticado). Hay edificios creados.

continúa en la página siguiente

continúa desde la página anterior

CU-20	Visualización de edificios
	El usuario puede localizar un edificio a través de un desplegable que se encuentra en la parte superior de la interfaz, además de la manera que se va explicar a continuación.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la herramienta Viewer. 2. El usuario pincha sobre clusters que aparecen en el plano de localización hasta ver un icono “G”, el cual significa que, ahí se encuentra un edificio. 3. El usuario pincha sobre el icono “G” y podrá visualizar la información sobre el edificio.
Postcondiciones	<ul style="list-style-type: none"> ■ La información del edificio se ve tal como el usuario administrador (Architect), la ha introducido.
	<ul style="list-style-type: none"> ■ Las bases de datos no están disponibles.
Excepciones	<ul style="list-style-type: none"> ■ Las REST API no está ■ No existen edificios.
Importancia	Alta

Tabla B.20: CU-20 Visualización de edificios

CU-21	Buscar edificio
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-4.1
Descripción	Permite al usuario encontrar y visualizar un edificio a partir de su identificador o nombre.

continúa en la página siguiente

continúa desde la página anterior

CU-21	Buscar edificio
Precondiciones	<ul style="list-style-type: none">■ Se encuentran disponibles las bases de datos.■ Se encuentran disponibles las REST API.■ Se accede como usuario Viewer (no autenticado).■ Hay edificios creados.
Acciones	<ol style="list-style-type: none">1. El usuario accede a la herramienta Viewer.2. El usuario tiene 2 opciones para buscar un edificio:<ul style="list-style-type: none">■ Localizando el edificio en el mapa: el usuario pincha sobre clusters que aparecen en el plano de localización hasta ver un icono “G”, el cual significa que, ahí se encuentra un edificio.■ Mediante el nombre o el identificador del edificio: el usuario puede localizar un edificio a través de un desplegable que se encuentra en la parte superior de la interfaz, introduciendo su nombre o identificador.

continúa en la página siguiente

continúa desde la página anterior

CU-21	Buscar edificio
Postcondiciones	<ul style="list-style-type: none"> La información del edificio se ve tal como el usuario administrador (Architect), la ha introducido. Las bases de datos no están disponibles.
Excepciones	<ul style="list-style-type: none"> Las REST API no está No existen edificios.
Importancia	Alta

Tabla B.21: CU-21 Buscar edificio

CU-22	Visualizar información
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-4.2
Descripción	Permite al usuario visualizar la información de un edificio.
Precondiciones	<ul style="list-style-type: none"> Se encuentran disponibles las bases de datos. Se encuentran disponibles las REST API. Se accede como usuario Viewer (no autenticado). Hay edificios creados.

continúa en la página siguiente

continúa desde la página anterior

CU-22	Visualizar información
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la herramienta Viewer. 2. El usuario pincha sobre clusters que aparecen en el plano de localización hasta ver un icono “G”, el cual significa que, ahí se encuentra un edificio. 3. El usuario pincha sobre el icono “G” y se debe visualizar toda la información del edificio, número de planta, el plano del edificio, los pois, sus rutas predefinidas etc. . . . Es decir debe ver toda la información que el usuario administrador (Architect) desee mostrar.
Postcondiciones	<ul style="list-style-type: none"> ■ La información del edificio se ve tal como el usuario administrador (Architect), la ha introducido. ■ Las bases de datos no están disponibles.
Excepciones	<ul style="list-style-type: none"> ■ Las REST API no está ■ No existen edificios.
Importancia	Alta

Tabla B.22: CU-22 Visualizar información

CU-23	Visualización de pois
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-5, RF-5.1, RF-5.2, RF-5.3

continúa en la página siguiente

continúa desde la página anterior

CU-23	Visualización de pois
Descripción	Permite al usuario visualizar los pois de un edificio.
Precondiciones	<ul style="list-style-type: none">▪ Se encuentran disponibles las bases de datos.▪ Se encuentran disponibles las REST API.▪ Se accede como usuario Viewer (no autenticado).▪ Hay edificios creados y pois.
Acciones	<ol style="list-style-type: none">1. El usuario accede a la herramienta Viewer.2. El usuario pincha sobre clusters que aparecen en el plano de localización hasta ver un icono “G”, el cual significa que, ahí se encuentra un edificio.3. El usuario pincha sobre el icono “G” y se visualiza la información del edificio. Además visualizan los pois correspondientes al edificio y la planta.
Postcondiciones	<ul style="list-style-type: none">▪ La información de los pois del edificio se ve tal como el usuario administrador (Architect), la ha introducido.
Excepciones	<ul style="list-style-type: none">▪ Las bases de datos no están disponibles.▪ Las REST API no está▪ No existen edificios o pois.

continúa en la página siguiente

continúa desde la página anterior

CU-23	Visualización de pois
Importancia	Alta

Tabla B.23: CU-23 Visualización de pois

CU-24	Visualizar información
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-5.1
Descripción	Permite al usuario visualizar la información de los pois un edificio.
Precondiciones	<ul style="list-style-type: none"> ▪ Se encuentran disponibles las bases de datos. ▪ Se encuentran disponibles las REST API. ▪ Se accede como usuario Viewer (no autenticado). ▪ Hay edificios creados y pois.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la herramienta Viewer. 2. El usuario pincha sobre clusters que aparecen en el plano de localización hasta ver un icono “G”, el cual significa que, ahí se encuentra un edificio. 3. El usuario pincha sobre el icono “G” y se visualiza la información del edificio. 4. El usuario pincha sobre un poi (los poi tienen iconos representativos, para identificarlos). Al pinchar en el poi aparece un panel desplegable en el que se puede visualizar su información, además pulsando en el icono “i” se visualizará la descripción del poi.

continúa en la página siguiente

continúa desde la página anterior

CU-24	Visualizar información
Postcondiciones	<ul style="list-style-type: none"> La información de los pois del edificio se ve tal como el usuario administrador (Architect), la ha introducido.
Excepciones	<ul style="list-style-type: none"> Las bases de datos no están disponibles. Las REST API no está No existen edificios o pois.
Importancia	Alta

Tabla B.24: CU-24 Visualizar información

CU-25	Marcar un poi
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-5.2
Descripción	Permite al usuario marcar o seleccionar un poi.
Precondiciones	<ul style="list-style-type: none"> Se encuentran disponibles las bases de datos. Se encuentran disponibles las REST API. Se accede como usuario Viewer (no autenticado). Hay edificios creados y pois.

continúa en la página siguiente

continúa desde la página anterior

CU-25	Marcar un poi
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la herramienta Viewer. 2. El usuario pincha sobre clusters que aparecen en el plano de localización hasta ver un icono “G”, el cual significa que, ahí se encuentra un edificio. 3. El usuario pincha sobre el icono “G” y se visualiza la información del edificio. 4. El usuario pincha sobre un poi para marcarlo, los pois tienen iconos representativos, para identificarlos.
Postcondiciones	<ul style="list-style-type: none"> ■ La información de los pois del edificio se ve tal como el usuario administrador (Architect), la ha introducido.
Excepciones	<ul style="list-style-type: none"> ■ Las bases de datos no están disponibles. ■ Las REST API no está ■ No existen edificios o pois.
Importancia	Alta

Tabla B.25: CU-25 Marcar un poi

CU-26	Proyectar camino de forma automática
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-5.3
Descripción	Permite al usuario al marcar un poi, que se dibuje un camino automático desde la ubicación del usuario hasta el poi indicado.

continúa en la página siguiente

continúa desde la página anterior

CU-26	Proyectar camino de forma automática
Precondiciones	<ul style="list-style-type: none">■ Se encuentran disponibles las bases de datos.■ Se encuentran disponibles las REST API.■ Se accede como usuario Viewer (no autenticado).■ Hay edificios creados y pois.
Acciones	<ol style="list-style-type: none">1. El usuario accede a la herramienta Viewer.2. El usuario pincha sobre clusters que aparecen en el plano de localización hasta ver un icono “G”, el cual significa que, ahí se encuentra un edificio.3. El usuario pincha sobre el icono “G” y se visualiza la información del edificio.4. El usuario pincha sobre un poi para marcarlo.5. El usuario pincha sobre el icono de navegación que aparece en el panel emergente, para proyectar el camino.

continúa en la página siguiente

continúa desde la página anterior

CU-26	Proyectar camino de forma automática
Postcondiciones	<ul style="list-style-type: none"> La información de los pois del edificio se ve tal como el usuario administrador (Architect), la ha introducido.
Excepciones	<ul style="list-style-type: none"> Las bases de datos no están disponibles. Las REST API no está No existen edificios o pois. No se puede localizar al usuario.
Importancia	Alta

Tabla B.26: CU-26 Proyectar camino de forma automática

CU-27	Visualización de rutas
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-6, RF-6.1, RF-6.2, RF-6.3
Descripción	Permite al usuario visualizar las rutas predefinidas del edificio seleccionado.
Precondiciones	<ul style="list-style-type: none"> Se encuentran disponibles las bases de datos. Se encuentran disponibles las REST API. Se accede como usuario Viewer (no autenticado). Hay edificios creados y rutas predefinidas.

continúa en la página siguiente

continúa desde la página anterior

CU-27	Visualización de rutas
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la herramienta Viewer. 2. El usuario pincha sobre clusters que aparecen en el plano de localización hasta ver un icono “G”, el cual significa que, ahí se encuentra un edificio. 3. El usuario pincha sobre el icono “G” y se visualiza la información del edificio. 4. El usuario pincha sobre el desplegable que aparece arriba a la derecha para seleccionar las rutas predefinidas del edificio.
Postcondiciones	<ul style="list-style-type: none"> ▪ La información de las rutas predefinidas del edificio se ve tal como el usuario administrador (Architect), la ha introducido.
Excepciones	<ul style="list-style-type: none"> ▪ Las bases de datos no están disponibles. ▪ Las REST API no está ▪ No existen edificios o rutas predefinidas.
Importancia	Alta

Tabla B.27: CU-27 Visualización de rutas

CU-28	Seleccionar ruta
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-6.1

continúa en la página siguiente

continúa desde la página anterior

CU-28	Seleccionar ruta
Descripción	Permite al usuario elegir entre las diferentes rutas predefinidas existentes en un edificio.
Precondiciones	<ul style="list-style-type: none">▪ Se encuentran disponibles las bases de datos.▪ Se encuentran disponibles las REST API.▪ Se accede como usuario Viewer (no autenticado).▪ Hay edificios creados y rutas predefinidas.
Acciones	<ol style="list-style-type: none">1. El usuario accede a la herramienta Viewer.2. El usuario pincha sobre clusters que aparecen en el plano de localización hasta ver un icono “G”, el cual significa que, ahí se encuentra un edificio.3. El usuario pincha sobre el icono “G” y se visualiza la información del edificio.4. El usuario pincha sobre el desplegable que aparece arriba a la derecha para seleccionar entre las rutas predefinidas del edificio.

continúa en la página siguiente

continúa desde la página anterior

CU-28	Seleccionar ruta
Postcondiciones	<ul style="list-style-type: none"> La información de las rutas predefinidas del edificio se ve tal como el usuario administrador (Architect), la ha introducido.
Excepciones	<ul style="list-style-type: none"> Las bases de datos no están disponibles. Las REST API no está No existen edificios o rutas predefinidas.
Importancia	Alta

Tabla B.28: CU-28 Seleccionar ruta

CU-29	Mostrar ruta
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-6.1
Descripción	Permite al usuario ver el trazado de la ruta.
Precondiciones	<ul style="list-style-type: none"> Se encuentran disponibles las bases de datos. Se encuentran disponibles las REST API. Se accede como usuario Viewer (no autenticado). Hay edificios creados y rutas predefinidas.

continúa en la página siguiente

continúa desde la página anterior

CU-29	Mostrar ruta
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la herramienta Viewer. 2. El usuario pincha sobre clusters que aparecen en el plano de localización hasta ver un icono “G”, el cual significa que, ahí se encuentra un edificio. 3. El usuario pincha sobre el icono “G” y se visualiza la información del edificio. 4. El usuario pincha sobre el desplegable que aparece arriba a la derecha. 5. El usuario selecciona una ruta predefinida, lo que hará que se muestre su trazado.
Postcondiciones	<ul style="list-style-type: none"> ■ La información de las rutas predefinidas del edificio se ve tal como el usuario administrador (Architect), la ha introducido.
Excepciones	<ul style="list-style-type: none"> ■ Las bases de datos no están disponibles. ■ Las REST API no está ■ No existen edificios o rutas predefinidas.
Importancia	Alta

Tabla B.29: CU-29 Mostrar ruta

CU-30	Limpiar ruta
Versión	1.0
Autor	Juan Pedro Pascual Vitores
Requisitos asociados	RF-6.1

continúa en la página siguiente

continúa desde la página anterior

CU-30	Limpiar ruta
Descripción	Permite al usuario dejar de ver el trazado de la ruta.
Precondiciones	<ul style="list-style-type: none">▪ Se encuentran disponibles las bases de datos.▪ Se encuentran disponibles las REST API.▪ Se accede como usuario Viewer (no autenticado).▪ Hay edificios creados y rutas predefinidas.
Acciones	<ol style="list-style-type: none">1. El usuario accede a la herramienta Viewer.2. El usuario pincha sobre clusters que aparecen en el plano de localización hasta ver un icono “G”, el cual significa que, ahí se encuentra un edificio.3. El usuario pincha sobre el icono “G” y se visualiza la información del edificio.4. El usuario pincha sobre el desplegable que aparece arriba a la derecha.5. El usuario selecciona una ruta predefinida, lo que hará que se muestre su trazado.6. El usuario pincha sobre el botón “x” para dejar de ver el trazado de la ruta predefinida.

continúa en la página siguiente

continúa desde la página anterior

CU-30	Limpiar ruta
Postcondiciones	<ul style="list-style-type: none">■ La información de las rutas predefinidas del edificio se ve tal como el usuario administrador (Architect), la ha introducido.
Excepciones	<ul style="list-style-type: none">■ Las bases de datos no están disponibles.■ Las REST API no está■ No existen edificios o rutas predefinidas.
Importancia	Media

Tabla B.30: CU-30 Limpiar ruta

Especificación de diseño

C.1. Introducción

En este apartado, nos detendremos a detallar la estructura de datos, los procedimientos y la arquitectura, que sigue la herramienta Geoindoor.

C.2. Diseño de datos

Este proyecto trabaja con una base de datos no sql por lo tanto la manera en la que están estructurados los datos, es un poco diferente a la que estamos acostumbrados con bases de datos sql. Los datos en vez de estar estructurados en tablas (BD sql), se estructuran en documentos¹.

Además Geoindoor utiliza el servicio web de bases de datos de Firebase, por lo que la BD esta formada por un único documento. Se usa JSON para el almacenamiento de datos, lo que nos lleva a utilizar un sistema de *clave-valor* para la ordenación.

En conclusión podríamos resumir las características de la BD de Geoindoor en 5 puntos:

- Base de datos no sql.
- Documentos en vez de entidades relacionales.
- Estructura de un único documento.
- Almacenamiento de datos a través *clave-valor*

¹Un documento es un conjunto de entidades de bases de datos que se agrupan en una entidad

- Utilización de JSON para almacenamiento de datos

Entonces la manera de analizar el diseño de datos no se puede realizar desde el punto de vista relacional. De todas formas, a continuación se muestran diagramas que permiten entender mejor el diseño y estructura de datos de Geoindoor.

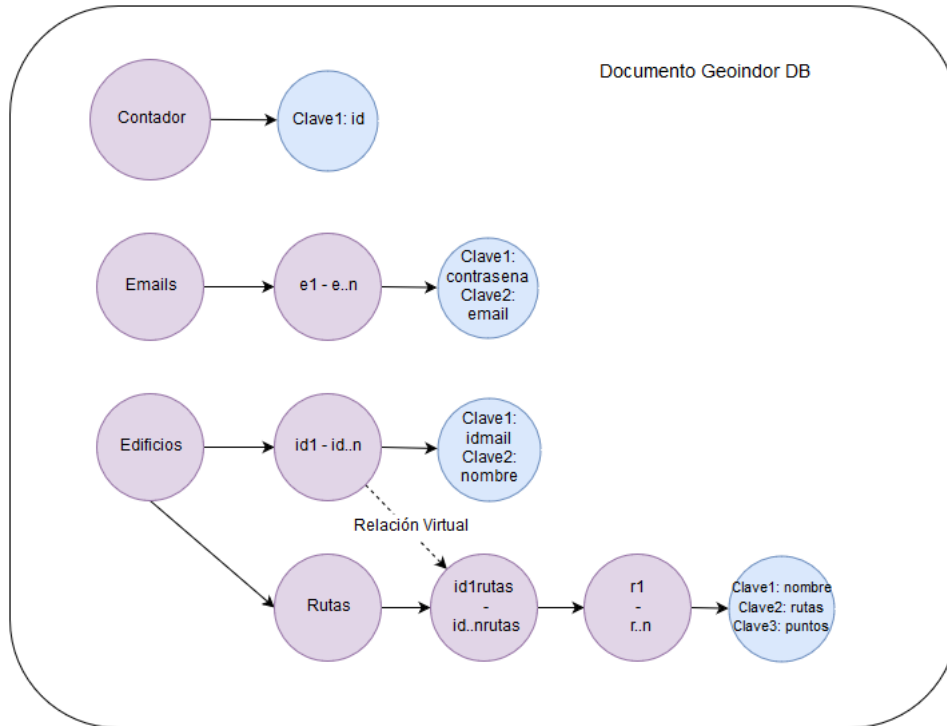


Figura C.1: Diagrama de relacion de entidades

Estos diagramas se leen de izquierda a derecha, para saber que entidad contiene a cual. Las entidades moradas contienen otras entidades, mientras que las azules guardan la relación *clave-valor*, donde se almacenan realmente los datos. Todas las entidades se encuentran en un mismo documento (Geoindoor DB).

La relación virtual indica que hay una relación lógica entre esas entidades, es decir, hay relación de datos entre entidades. En este caso la entidad “id1 - id.n”, que identifica a los edificios, esta relacionada con “id1rutas - id.nrutas”, que identifica las rutas de cada edificio. De esta manera podríamos decir que la relación entre “id1 - id.n” y “id1rutas - id.nrutas” es de 1 - *.

Geoindoor DB entidades:■ **Contador**

- Clave 1: id

■ **Emails**

- **e1 - e..n**: identificadores de email.
 - Clave 1: contraseña
 - Clave 2: email

■ **Edificios**

- **id1 - id..n**: identificadores de edificio.
 - Clave 1: idmail
 - Clave 2: nombre
- **Rutas**
 - **id1rutas - id..nrutas**: identificadores de ruta de edificio.
 - ◊ **r1 - r..rn**: identificadores de ruta.
 - Clave 1: nombre
 - Clave 2: rutas
 - Clave 3: puntos

C.3. Diseño procedimental

En el diseño procedimental explica como interaccionan las distintas partes del algoritmo. A continuación se muestran unos diagramas de secuencia que describen el comportamiento de los algoritmos utilizados para el correcto funcionamiento de la herramienta Geoindoor. No se exponen todos los diagramas de secuencia resultantes de el análisis de los algoritmos, por que no son relevantes para el proyecto Geoindoor.

Para los diagramas de la parte de Architect de Geoindoor, el usuario ya esta logeado una vez comienza la acción.

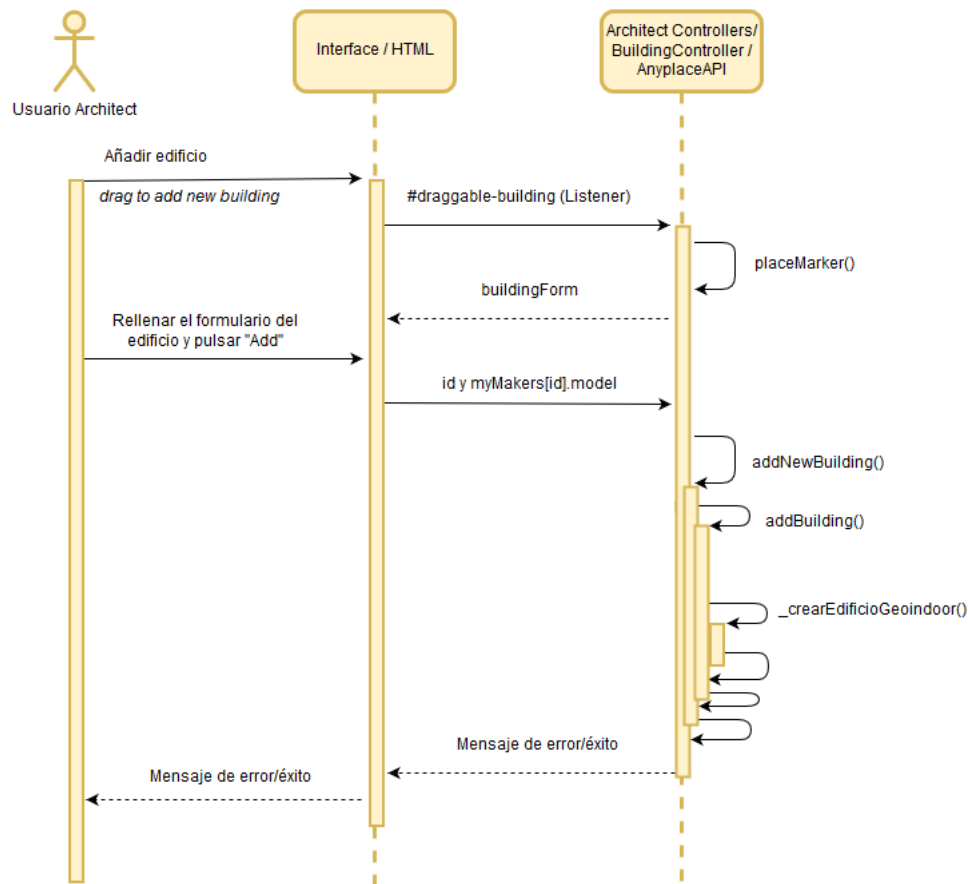


Figura C.2: Diagrama de secuencia de la creación de un edificio

Para que el siguiente diagrama tenga sentido hay que ubicarle en un contexto. Debe haber un edificio creado y con al menos dos pois.

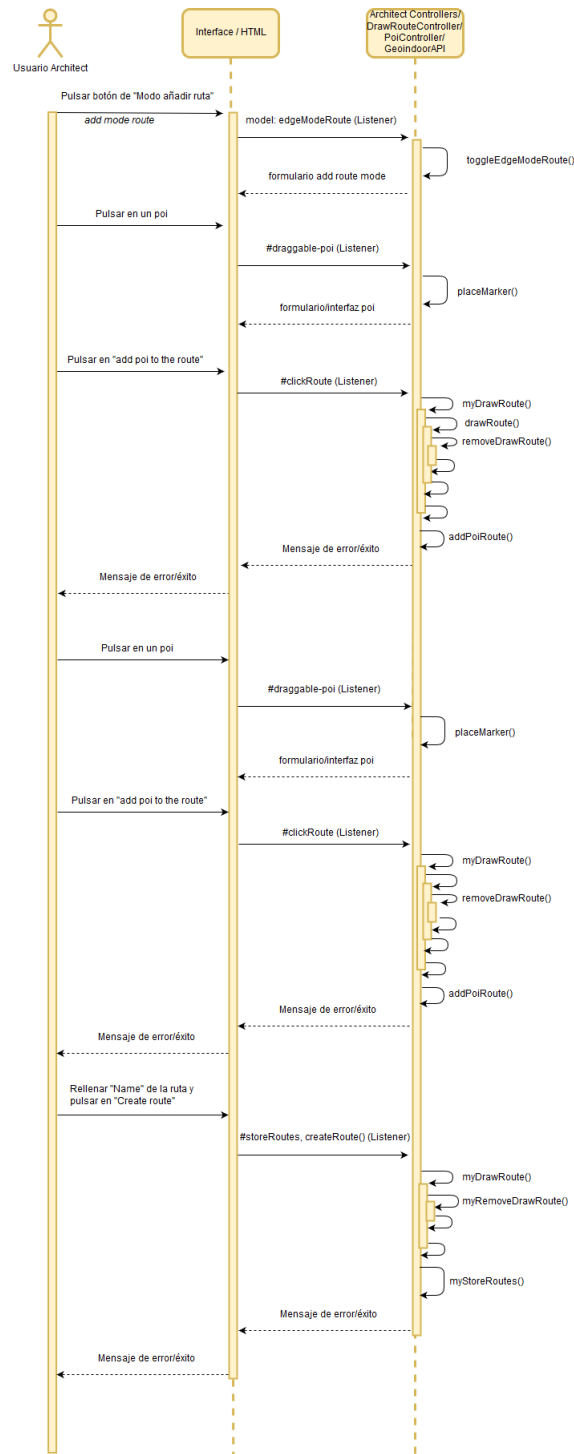


Figura C.3: Diagrama de secuencia de la creación de una ruta predefinida

Para el siguiente diagrama el contexto cambia. Ahora deber haber un edificio con al menos una ruta predefinida.

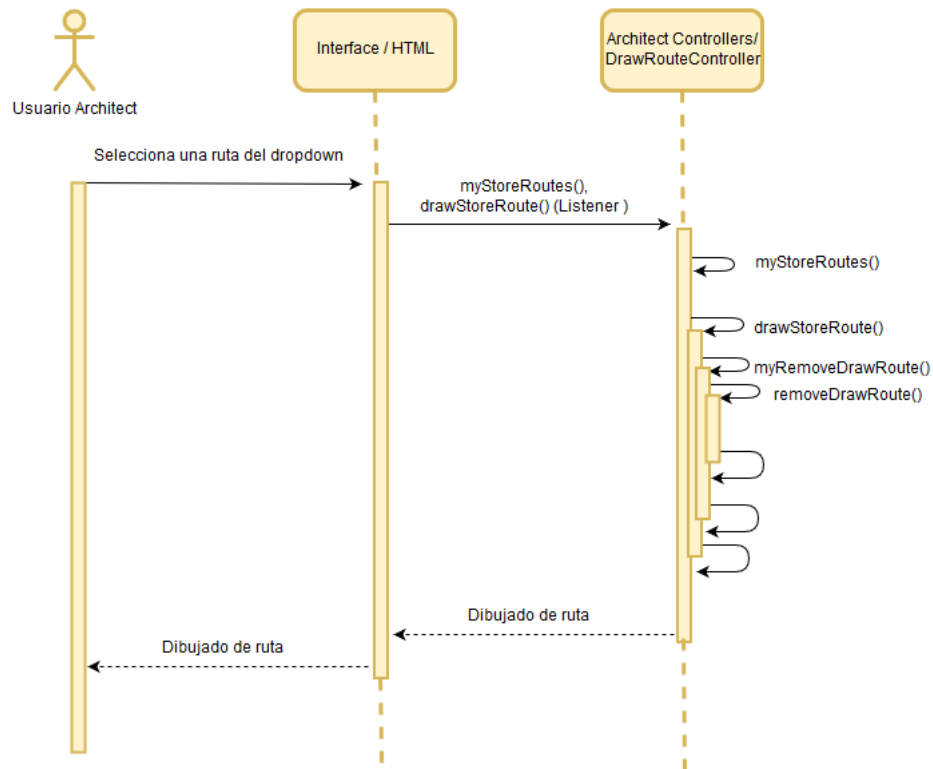


Figura C.4: Diagrama de secuencia del dibujado de una ruta predefinida al seleccionarla

El siguiente diagrama tiene el mismo contexto que el anterior.

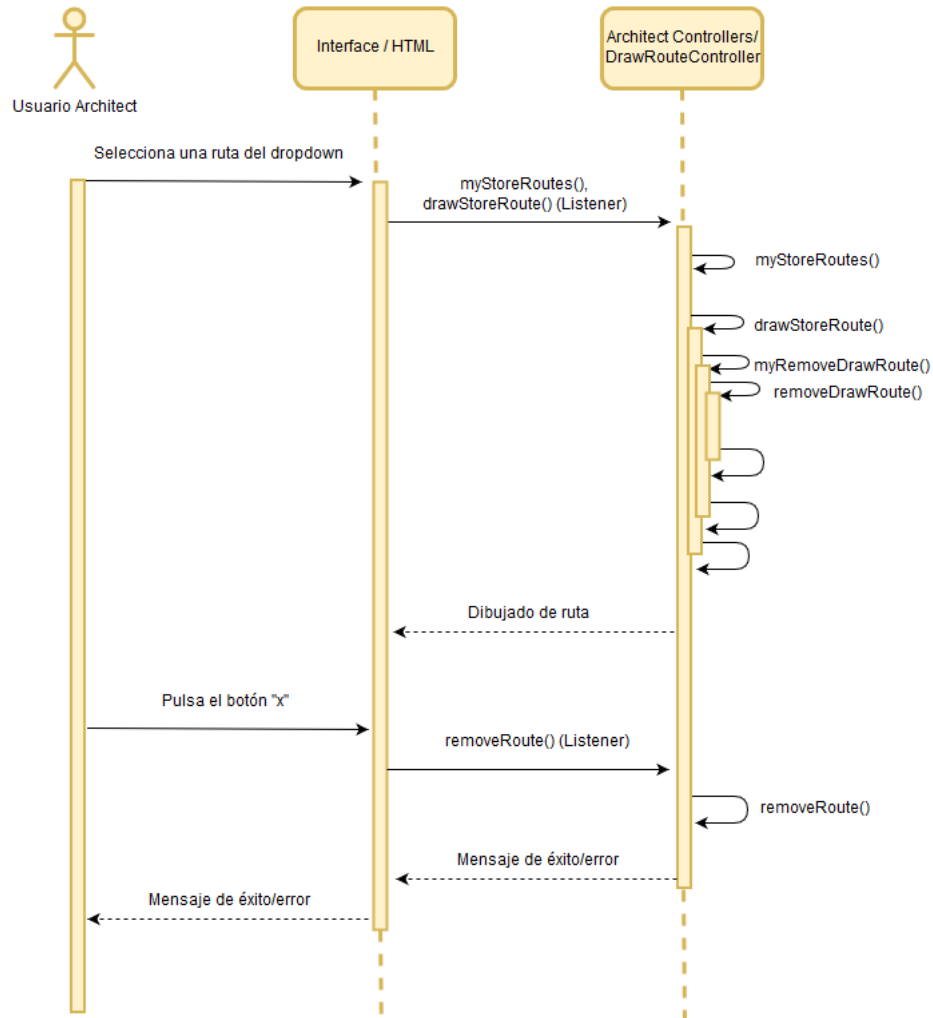


Figura C.5: Diagrama de secuencia del borrado de una ruta

En este diagrama, nos encontramos en Geoindoor Viewer, en un edificio con al menos una ruta predefinida.

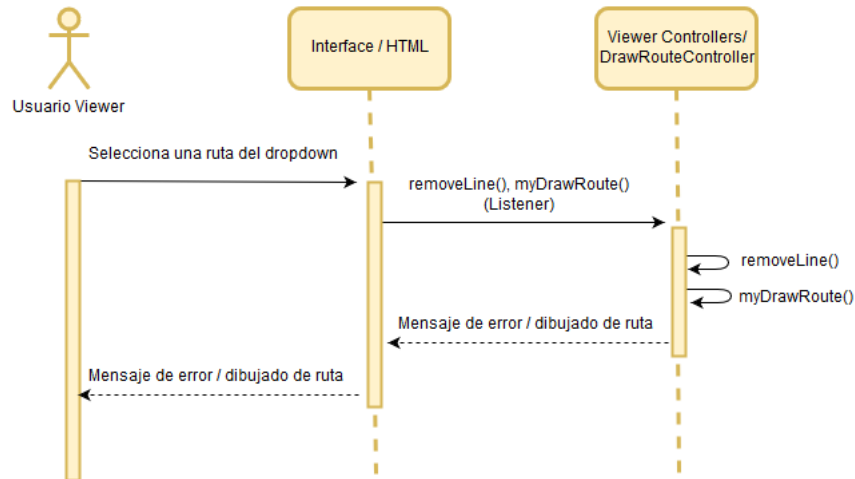


Figura C.6: Diagrama de secuencia del dibujo de una ruta (Viewer)

C.4. Diseño arquitectónico

En este apartado se hablara de como esta diseñado el sistema Geoindoor a nivel de su arquitectura de software.

El patrón que se ha elegido para implementar y que en gran parte ha sido impuesto por la herramienta en la que se ha basado Geoindoor (AnyPlace), es el patrón Modelo-Vista-Controlador (MVC).

El patrón MVC busca que la interfaz de usuario sea totalmente independiente de la lógica del sistema, pero con Geoindoor y AnyPlace, esta independencia, en ciertos aspectos se vuelve muy difícil de conseguir, ya que JavaScript y AngularJS dependen en gran medida de la interfaz HTML, lo que en varias ocasiones ha hecho que haya una relación de dependencia entre lógica e interfaz.

Modelo Vista Controlador

Este patrón se basa en separar los datos y la lógica del sistema de la interfaz del usuario. Para tal finalidad el MVC propone la construcción de 3 componentes: **modelo**, **vista** y **controlador** [?].

- **Modelo:** es el encargado de mantener los datos, es decir, es el que se encarga de la representación y gestión (actualizaciones, borrados, con-

sultas, creaciones) de la información. Envía a la *Vista* la información que le solicita el *Controlador*.

- **Controlador:** Es el encargado de controlar la interacción entre la *Vista* y el *Modelo*. Responde a eventos como acciones del usuario, y transmite las peticiones al modelo.
- **Vista:** es el componente encargado de mostrar la información recibida del *Modelo* al usuario y transmitir las peticiones del usuario al *Controlador*.

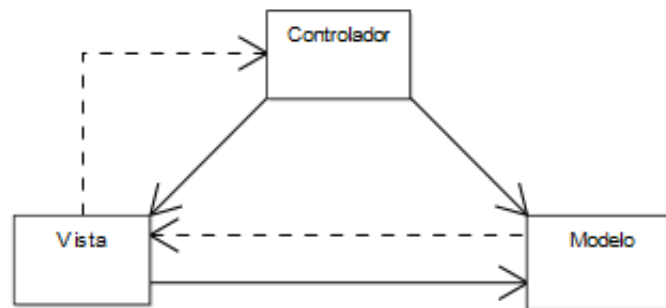


Figura C.7: Diagrama Modelo-Vista-Controlador [?]

La abstracción del funcionamiento de un sistema Modelo-Vista-Controlador, puede ser representada de la siguiente forma.

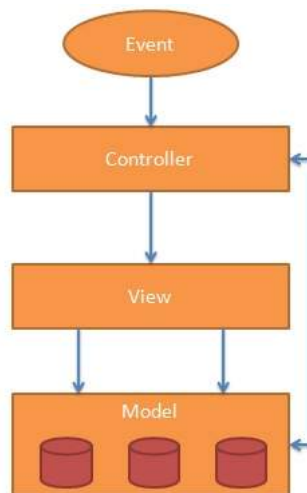


Figura C.8: Diagrama de flujo Modelo-Vista-Controlador [?]

Modelo-Vista-Controlador en Geoindoor

En cuanto a como esta implementado el Modelo-Vista-Controlador en Geoindoor podemos categorizar cada elemento participante, en cada uno de los estratos del MVC.

- **Modelo:** en el modelo podemos introducir el servicio de base de datos de Firebase (Geoindoor), la REST API creada en Heroku (Geoindoor), la BD de AnyPlace y la REST API de AnyPlace. Ambas API REST nos permiten interactuar con las respectivas bases de datos.
 - BD Geoindoor
 - REST API Geoindoor
 - BD AnyPlace
 - REST API AnyPlace
- **Controlador:** en el componente de controladores podemos añadir el fichero “*app.js*”², todos los ficheros de la carpeta “*controllers*” y los ficheros de la carpeta “*scripts*” que son utilidades, usadas por los controladores.
- **Vista:** en la vista añadiremos el fichero “*index.html*”. Bootstrap, CSS y demás elementos para el estilo se podrían introducir en componentes que forman la *Vista*.

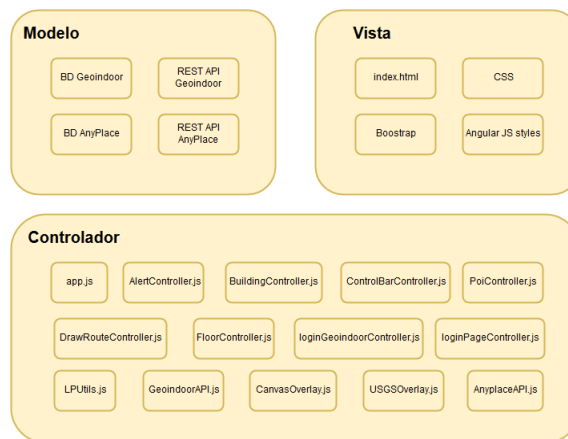


Figura C.9: Elementos integrantes del Modelo-Vista-Controlador en Architect

²app.js: es el fichero en el que se declara cada aplicación, y esta basada en un modulo, a partir de el cual surgen los controladores.

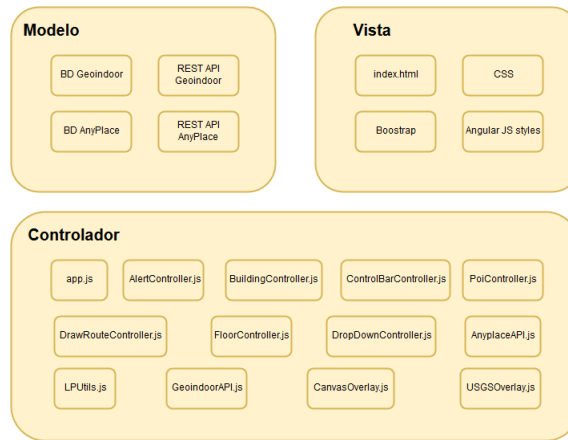


Figura C.10: Elementos integrantes del Modelo-Vista-Controlador en Viewer

Navegabilidad

En cuanto a la navegabilidad del sistema, es muy simple, una vez arrancada la herramienta o sistema Geoindoor, aparecerá una pantalla de inicio, dónde encontramos dos botones que permitirán al usuario elegir si trabajar con la herramienta Architect o Viewer, de Geoindoor.

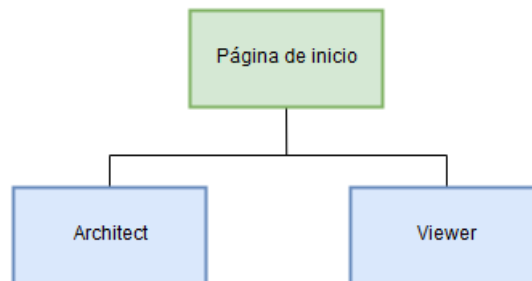


Figura C.11: Diagrama de navegabilidad de Geoindoor

No se considera la fácil navegación (idas y vueltas intermitentes) entre la herramienta Architect y Viewer, ya que para el usuario no tiene ningún sentido esa navegación.

Documentación técnica de programación

D.1. Introducción

En este apartado se hablará de la estructura final de directorios del proyecto, se detallará información importante para los desarrolladores, como la forma de compilar, instalar y ejecutar el proyecto. Además se explicarán las pruebas realizadas a Geoindoor.

D.2. Estructura de directorios

La estructura de directorios del proyecto en github es la siguiente:

- **./**: directorio raíz donde se encuentra el proyecto Geoindoor.
- **./BaseDeDatos**: directorio donde se encuentra el documento de la exportación de la base de datos.
- **./Despliegue Geoindoor**: directorio donde se encuentra el diagrama del despliegue de la aplicación Geoindoor.
- **./Geoindoor**: directorio donde se encuentra la aplicación Geoindoor pero sin el contexto para el lanzamiento de la herramienta.
- **./Paquete Geoindoor**: directorio donde se encuentra la aplicación Geoindoor con el contexto necesario para el lanzamiento de la herramienta y el zip que contiene la aplicación Geoindoor.
- **./Planos Museo**: directorio donde se encuentran los planos del museo de la evolución, los cuales serán utilizados para la presentación de Geoindoor.

- **./Server REST API**: directorio donde se encuentra el sistema de ficheros necesario, para crear una REST API en Heroku, la REST API de Geoindoor.¹
- **./TestEstresGeoindoor**: directorio donde se encuentran los test de estrés de Geoindoor.
- **./TestGeoindoor**: directorio donde se encuentran los test de integración de Geoindoor.
- **./Documentacion**: directorio donde se encuentra la documentación de Geoindoor, escrita en la latex.

En cuanto a la estructura de directorios del proyecto se merece especial análisis el **./Paquete Geoindoor**.

- **Paquete Geoindoor/PaqueteGeoindoor/**: encontramos los lanzadores de la herramienta Geoindoor.
- **Paquete Geoindoor/PaqueteGeoindoor/geoindoor/**: encontramos la página de inicio, con sus respectivas dependencias y las carpetas de las aplicaciones Architect y Viewer.
- **Paquete Geoindoor/PaqueteGeoindoor/geoindoor/architect**: encontramos la página de login y la página de inicio de Architect. Además encontramos las carpetas “bower_components” y “node_modules” donde se encuentran las dependencias de bower y node.
- **Paquete Geoindoor/PaqueteGeoindoor/geoindoor/architect/controllers**: directorio donde se encuentran los controladores de la aplicación.
- **Paquete Geoindoor/PaqueteGeoindoor/geoindoor/architect/scripts**: directorio donde se encuentran los ficheros que contienen las utilidades.²
- **Paquete Geoindoor/PaqueteGeoindoor/geoindoor/viewer**: encontramos la página de inicio de Viewer. Además encontramos las carpetas “bower_components” y “node_modules” donde se encuentran las dependencias de bower y node.
- **Paquete Geoindoor/PaqueteGeoindoor/geoindoor/viewer/controllers**: directorio donde se encuentran los controladores de la aplicación.
- **Paquete Geoindoor/PaqueteGeoindoor/geoindoor/viewer/scripts**: directorio donde se encuentran los ficheros que contienen las utilidades.³

¹Merece especial mención el fichero **./Server REST API/index.js** en el cual están las funciones y métodos para la REST API.

²Se desestima hablar de las restantes carpetas, ya que no son de tanto interés.

³Se desestima hablar de las restantes carpetas, ya que no son de tanto interés.

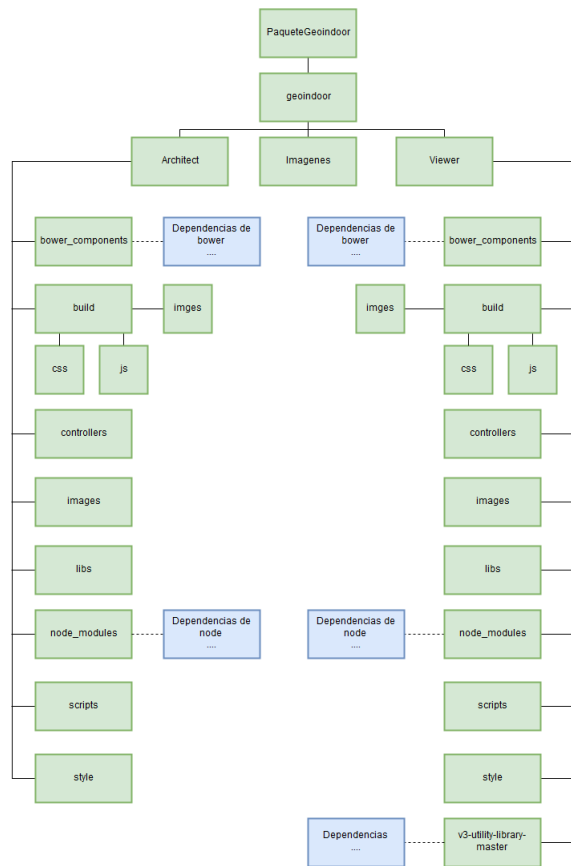


Figura D.1: Esquema de directorios Geoindoor

D.3. Manual del programador

En el proyecto se han utilizado varias herramientas que nos han facilitado el desarrollo, y que habitualmente requieren de instalación.

- **Java** (Tests)
- **Eclipse** (Tests)
- **Git**
- **Node.js**
- **Bower**
- **Grunt**
- **Heroku**

- **Python**

Java

Se ha necesitado instalar Java DK 8 (Development Kit) y Java RE 8 (Runtime Environment) para las pruebas de integración a través de selenium. JDK es un software que nos permite crear programas en java [?], mientras que JRE son las utilidades que nos permiten correr programas java [?].

Para la instalación de ambos es, necesario ir a la pagina web oficial ([JDK](#), [JRE](#)), ahí debemos buscar la versión que deseemos, y descargarla asegurando que es compatible con nuestro sistema operativo. Una vez descargada, se siguen los pasos del asistente para acabar con la instalación. Posteriormente debemos configurar las variables de entorno [?].

- **JAVAPATH**: la ruta completa donde está instalado JDK.
- **CLASSPATH**: la ruta en la que están las bibliotecas o clases de usuario.
- **PATH**: variable en la que se añade la ruta donde esta el JDK.

Eclipse

Para realizar nuestros test de integración con selenium hemos necesitado un IDE (Integrated Development Environment) para realizar la tarea, se ha usado Eclipse Jee Neon ⁴.

Para la instalación debemos ir a la página oficial ([Eclipse](#)) y descargar la versión que queramos, asegurándonos que es compatible con nuestro sistema operativo. Una vez descargado, ejecutamos y seguimos los pasos del asistente hasta finalizar la instalación o en el caso de no haber asistente, se descomprime en el lugar deseado.

Git

Git es un software de control de versiones que ha sido utilizado conjuntamente con GitHub para llevar el control de versiones de Geoindoor. También ha sido utilizado para poner en producción, los cambios en la REST API de Geoindoor (Heroku).

Para instalar Git debemos ir a la página oficial ([Git](#)) y descargar la versión que deseemos, fijándonos en que sea compatible con nuestro sistema operativo. Una vez hecha la descarga se sigue el asistente hasta finalizar la instalación. Con la instalación vienen dos herramientas que nos permiten trabajar con Git.

⁴Eclipse Jee Neon es un IDE que nos permite desarrollar programas java en un entorno determinado, con un contexto determinado.

APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN⁸⁵

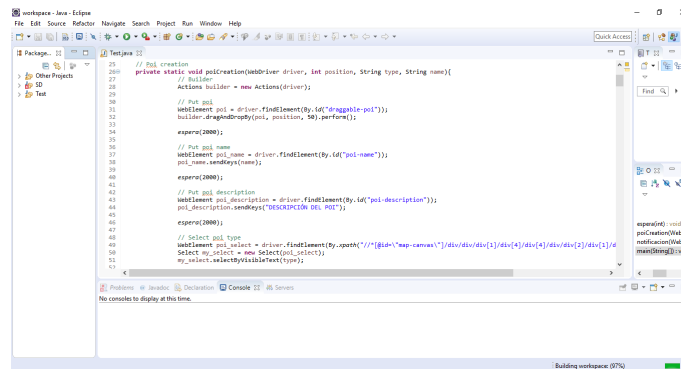


Figura D.2: Eclipse Jee Neon

- **Git bash:** consola con la cual se puede trabajar en el control de versiones a través de comandos de Git.
- **Git GUI:** interfaz gráfica que nos permite trabajar con Git de forma más amigable.

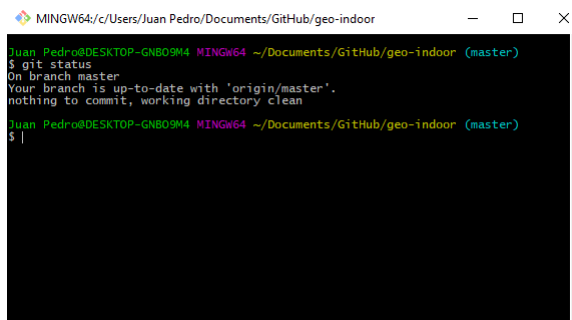


Figura D.3: Terminal de bash de Git

Contiene los comandos git utilizados para poner en producción los ficheros en Heroku.

Node.js

Para poder utilizar bower y grunt tenemos que tener instalados el manejador de paquetes “npm” y para ello debemos tener instalado Node.js. Para instalar Node.js tenemos que ir a la pagina oficial ([Node.js](https://nodejs.org/)) y descargar una versión igual o superior a la 6.11.0 (se recomienda la versión 6.11.0) compatible con nuestro sistema operativo. Una vez se haya descargado, ejecutamos y seguimos los pasos del instalador hasta haber finalizado la instalación.

APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN⁸⁶

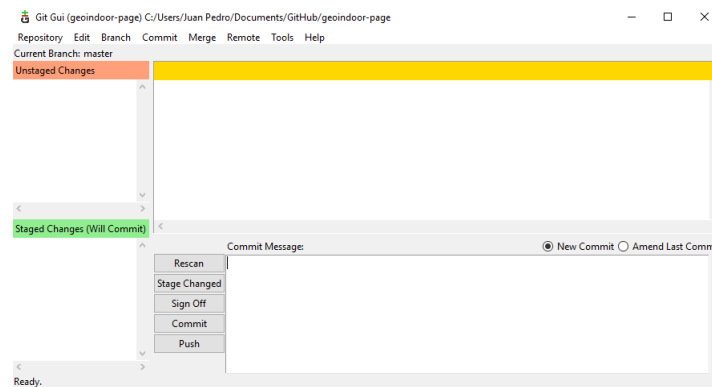


Figura D.4: Git GUI

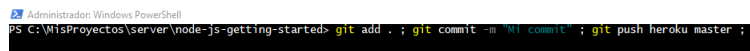


Figura D.5: Git en PowerShell

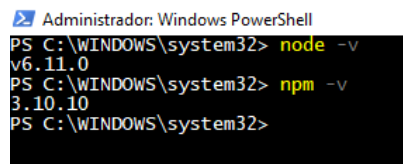


Figura D.6: Comandos “node” y “npm” una vez instalado Node.js.

Bower

Bower es un gestor de paquetes para la web, que utilizamos para manejar los paquetes del proyecto, para su instalación, necesita que previamente este instalado node, npm y git. Para la instalación basta con abrir el terminal o consola y ejecutar “npm install -g bower”. [?]

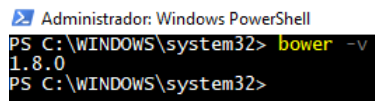


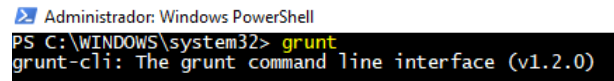
Figura D.7: Comando “bower -v” una vez instalado Bower.

Grunt

Grunt es un corredor de tareas (Task runner), que nos ha permitido compilar el proyecto de forma sencilla y rápida, ayudándonos a tener siempre

centralizados los scripts y sus dependencias. Nos permite hacer cambios en caliente ya que recompila el proyecto para poder probarlo.

Para la instalación de grunt es necesario tener previamente instalado npm (Node.js). Para instalarlo debemos ejecutar el comando “npm install -g grunt-cli” [?].



```

Administrador: Windows PowerShell
PS C:\WINDOWS\system32> grunt
grunt-cli: The grunt command line interface (v1.2.0)
    
```

Figura D.8: Comando “grunt” una vez instalado Grunt.

Heroku

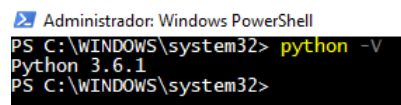
Heroku es el servicio web que hemos utilizado para crear la REST API de Geoindoor en Node.js, para poder trabajar con ello debemos descargarnos “Heroku CLI” para poder comunicarnos con ser servicio.

Para ello debemos descargarnos e instalar la versión que corresponda a nuestro sistema operativo, desde la página oficial ([Heroku CLI](#)), una vez hecha la descarga, se ejecuta el fichero y se siguen los pasos del instalador hasta finalizar la instalación.

Heroku

Python es un lenguaje de programación interpretado que ha sido utilizado para simular un servidor http, para el desarrollo de Geoindoor no se hace necesario, pero si ofrece comodidad.

Para su instalación vamos a su página oficial ([Python](#)) y nos descargamos la versión (mayor o igual a 3.6.1) que corresponda a nuestro sistema operativo. A continuación ejecutamos el fichero descargado y seguimos los pasos del auto instalador hasta finalizar la instalación. En la mayoría de distros de Linux, Python viene ya instalado.



```

Administrador: Windows PowerShell
PS C:\WINDOWS\system32> python -V
Python 3.6.1
PS C:\WINDOWS\system32>
    
```

Figura D.9: Comando “python -V” una vez instalado Python.

D.4. Compilación, instalación y ejecución del proyecto

En este apartado se va explicar como compilar, instalar y ejecutar el proyecto Geoindoor desde el principio. Es decir se explicara como se creo la REST API en Heroku de Geoindoor y como se creo la base de datos en Firebase.

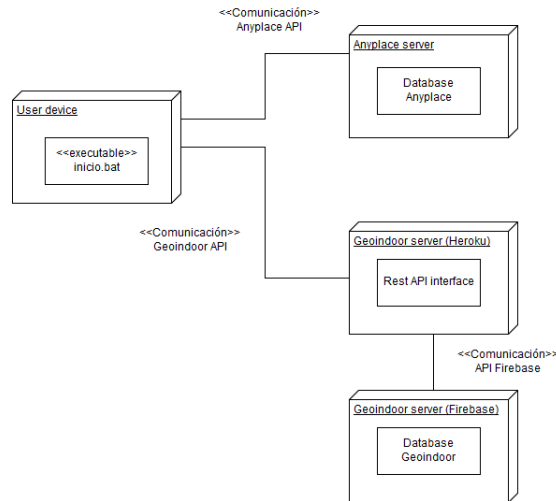


Figura D.10: Diagrama de despliegue Geoindoor.

Heroku Server REST API y Firebase Base de datos

Lo primero que debemos hacer es clonar o descargarnos el repositorio de Geoindoor desde GitHub.

Una vez descargado el repositorio Geoindoor debemos ir a la página oficial de **Heroku** y hacernos una cuenta. Una vez hecha, debemos crearnos una app en Node.js en Heroku, que se hace mediante los siguientes comandos

- ***heroku login***: login con la cuenta de Heroku.
- ***git clone https://github.com/heroku/node-js-getting-started.git***: clonación de la plantilla.
- ***cd node-js-getting-started***
- ***heroku create***: creación de la app.
- ***git push heroku master***: poner la app en el repositorio.
- ***heroku ps:scale web=1***: asegurarnos de que esta funcionando al menos una instancia de la app.

APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN⁸⁹

Una vez hecho esto basta sustituir los ficheros de la app recién creada con los ficheros `./Server REST API/` del repositorio de Geoindoor.

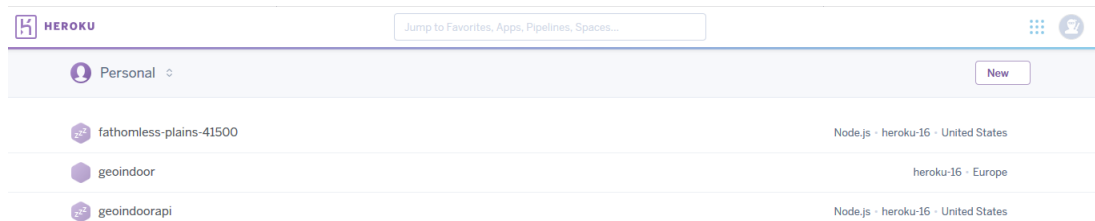


Figura D.11: Heroku dashboard.

Ahora para crear el servicio de base de datos en **Firestore**, debemos ir a la página oficial y crearnos una cuenta (a través de la cuenta de gmail), desde la página de inicio de la consola creamos un nuevo proyecto, con el nombre que deseemos. Ahora en el apartado de “Database” en el panel de control, importamos la base de datos que encontramos en `./BaseDeDatos/`.

En este momento debemos enlazar nuestra REST API con la base de datos de Firestore, para ello pegamos en el `index.js` de nuestra REST API la siguiente porción de código con la información correspondiente, o cambiamos la que está por la correcta.

```
firebase.initializeApp({
  serviceAccount: "nombre_del_fichero_de_nuestra_clave_Sin_extension",
  databaseURL: "https://geoindoordb.firebaseio.com"
});
```



Figura D.12: Página de inicio de la consola Firebase.



Figura D.13: Panel de control Firebase.

Para generar la clave (serviceAccount) debemos ir al panel de control de nuestra aplicación web en Firebase, y desde Configuración/Cuentas de servicio debemos crear la clave. Una vez creada la ubicamos en la raíz de nuestra REST API y borramos la anterior.

De esta forma quedaría enlazada la REST API con la base de datos de Firebase.

[?]

Geoindoor cliente

Ahora vamos a analizar como compilar, instalar y ejecutar la parte del cliente de Geoindoor.

En primer lugar debemos poner **./Paquete Geoindoor** en el directorio en el que se quiera trabajar, para lanzar la herramienta se debe ejecutar el lanzador (inicioWindows.bat o inicioLinux.sh), Geoindoor se lanzará en el puerto 9000 (localhost). En Windows, a través de Google Chrome y en Linux a través de Firefox.

Si no se dispone de python instalado, puede usar mismamente **./Geoindoor** y ubicarlo en el lugar en el que se quiera trabajar, que proporcione un servicio http, ya sea XAMPP, WAMP u otra herramienta que simule un servidor.

Tanto en Viewer como en Architect la forma de comenzar el desarrollo es la misma.

- Abirmos la consolola y nos colocamos en la raíz de las aplicaciones, las carpetas ***/architect/** y ***/viewer/**
- Ejecutamos **bower install**, si no existiera el fichero “bower.json”.

- Ejecutamos **grunt**.

Si nuestro servicio http funciona bien, nos dirigimos con el navegador a las raíz de la carpeta ***/architect/** y ***/viewer/** para comprobar que Geoindoor esta en funcionamiento. Si aún así no funciona, podemos dirigirnos manualmete a ***/architect/index.html** y ***/viewer/index.html**

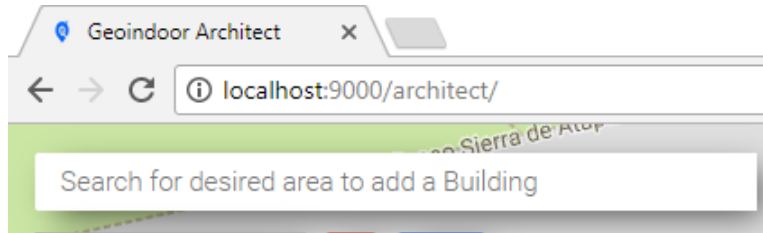


Figura D.14: Url de Geoindoor al lanzarla.

D.5. Pruebas del sistema

La pruebas realizadas al sistema Geoindoor, se pueden dividir en dos.

- Pruebas de integración.
- Pruebas de estrés.

Pruebas de integración

Para las pruebas de integración se ha utilizado Selenium, que es un automatizador de los navegadores, lo que nos permite automatizar acciones sobre la herramienta Geoindoor y ver los resultados.

Las pruebas de integración que se han hecho a Geoindoor son las siguientes.

Test	Requisito
Gestión de edificios	■ RF-1.1 Emplazar edificio.
	■ RF-1.2 Detallar información del edificio.
	■ RF-1.3 Emplazar plano.
	■ RF-1.4 Redimensionar y mover el plano
Gestión de pois	■ RF-2.1 Emplazamiento de pois.
	■ RF-2.2 Detallar información del poi.
	■ RF-2.3 Ubicar poi.
	■ RF-2.4 Enlazar pois.
Gestión de rutas	■ RF-3.1 Crear ruta.
	■ RF-3.2 Detallar información de la ruta.
	■ RF-3.3 Añadir pois a ruta.
	■ RF-3.4 Trazar ruta.
	■ RF-3.5 Seleccionar ruta.
	■ RF-3.6 Limpiar ruta.
	■ RF-3.7 Borrar ruta.

continúa en la página siguiente

continúa desde la página anterior

Test	Requisito
------	-----------

Tabla D.1: Requisitos por test de interfaz.

Para la ejecución del test, debemos importar en eclipse el proyecto “Test” que se encuentra en **./TestGeoindoor/Test.zip**. Una vez importado, debemos asegurarnos que tenemos importadas las librerías de Selenium, si no es así debemos hacerlo. Una vez importadas las librerías tenemos que ubicar el driver (Google Chrome, Mozilla Firefox, Safari, etc. . .), en el lugar que deseemos, sin olvidar que después hay que indicar la ruta del driver en el código de nuestro test. Una vez realizados todos estos pasos, solo nos quedaría lanzar Geoindoor a través de sus lanzadores (inicioLinux.sh o inicioWindows.sh), y ejecutar el test desde eclipse.

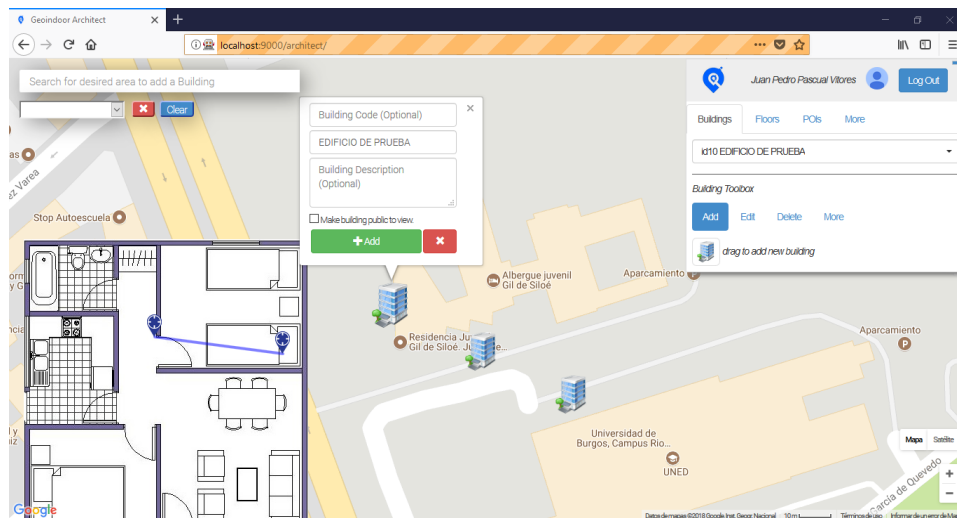


Figura D.15: Test lanzado en Firefox a través de Selenium.

Los test quedan recogidos en **./TestGeoindoor/Test.mp4** y el código en java, para el test con Selenium, en **./TestGeoindoor/Test.zip**. Se proporciona también las librerías de Selenium (**./TestGeoindoor/selenium-java-3.7.1.zip**) y los drivers de Firefox (**./TestGeoindoor/geckodriver-v0.19.1-win64.zip**) y Google Chrome (**./TestGeoindoor/chromedriver-win32.zip**). Además de la cuenta en Geoindoor para tests (**./TestGeoindoor/TestCuenta.txt**)

Pruebas de estrés

Las pruebas de estrés han sido realizadas a través de la herramienta Web-server Stress Tool, una aplicación de prueba HTTP-client/server, que detecta

APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN94

problemas de rendimiento en un sitio web o servidor web [?].

La prueba de estrés consiste en simular 10 usuarios simultáneos, 5 clicks por usuario y 12 segundos entre clicks.

User No.	Clicks	Hits	Errors	Avg. Click Time [ms]	Bytes	kbit/s
1	5	5	0	34	4.595	216,10
2	5	5	0	25	4.595	294,31
3	5	5	0	24	4.595	311,34
4	5	5	0	27	4.595	269,21
5	5	5	0	18	4.595	420,00
6	5	5	0	25	4.595	290,49
7	5	5	0	24	4.595	302,58
8	5	5	0	35	4.595	211,21
9	5	5	0	18	4.595	415,13
10	5	5	0	20	4.595	372,29

Figura D.16: Resultados por usuario.

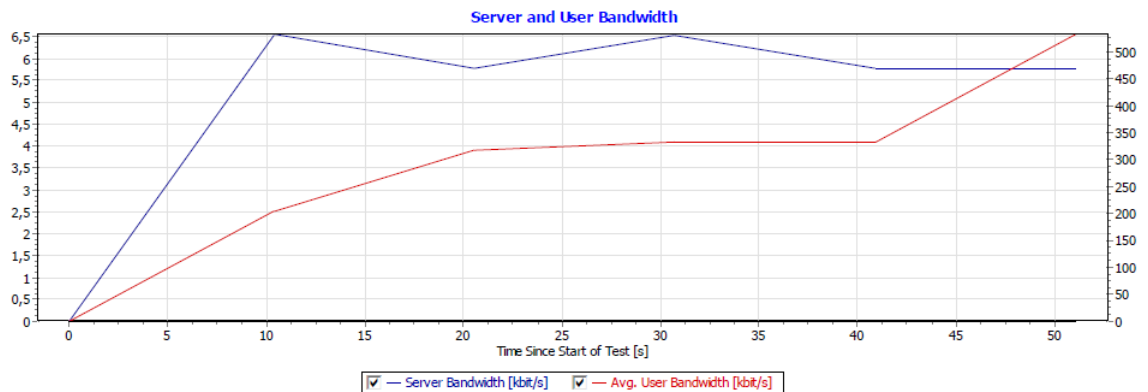


Figura D.17: Ancho de banda del servidor y del usuario.

Para ejecutar el test debemos instalar Webserver Stress Tool, una vez instalado, abrimos Webserver Stress Tool y en el apartado de “Test type” elegimos la casilla “CLICKS”, a continuación en el campo “Run until” ponemos el valor

5. En el apartado de “User simulation” en “Number Of Users”, ponemos 10 y en “Click delay” 12.⁵

Ahora en el apartado “URLs” ponemos la url de Geoindoor, en nuestro caso `http://localhost:9000`. Navegamos hasta el apartado “Browser Settings” y en “Browser Simulation / Use agent”, marcamos la casilla, y ponemos *Mozilla/5.0 (compatible; Webserver Stress Tool 8; Windows)*, en el caso que queramos hacer el test con Firefox en Windows, si no, se debe poner el correspondiente al navegador y sistema operativo. A continuación marcamos la casilla “Use Timeouts” y le ponemos un valor de 120. De esta manera se replica el test que se ha hecho a Geoindoor.

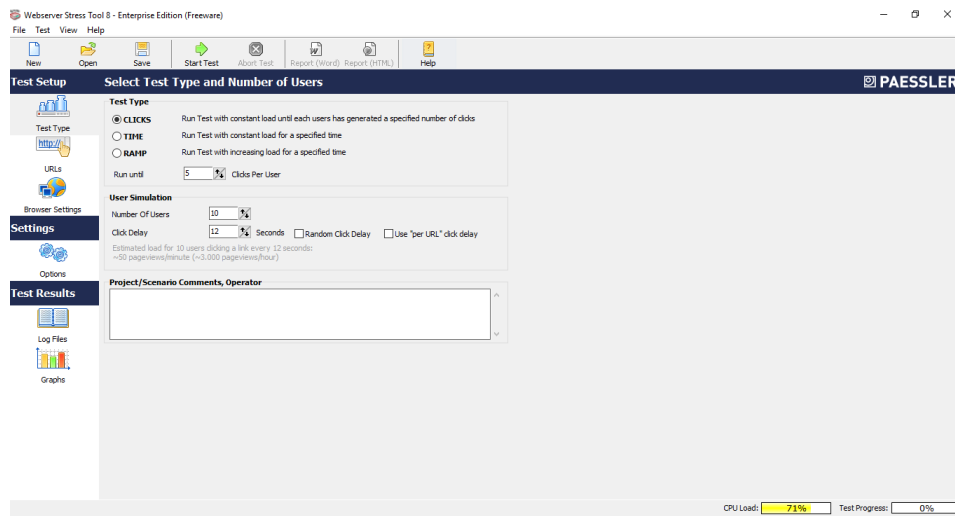


Figura D.18: Apartado “Test Type” en Webserver Stress Tool.

En `./TestEstresGeoindoor/GeoindoorWebStress.{pdf doc}` se encuentra un informe más detallado de los resultados del test, así como la herramienta Webserver Stress Tool (`./TestEstresGeoindoor/webstress.zip`). Además en `./TestEstresGeoindoor/TestEstres.{pdf tex}` se encuentran las restricciones en cuanto a rendimiento de Firebase y Heroku.

⁵En la versión de pago no hay limitación de valores en estos campos, pero en la versión gratuita sí.

Documentación de usuario

- E.1. Introducción**
- E.2. Requisitos de usuarios**
- E.3. Instalación**
- E.4. Manual del usuario**

Bibliografía
