



UNIVERSIDADE DO ALGARVE

Faculdade de Ciências e Tecnologia

Instituto Superior de Engenharia

*WebSIG* para visualização e  
processamento de dados espaciais  
de interiores de edifícios

**David Elizalde Castells**

PROJECTO

MESTRADO EM GEOMÁTICA

RAMO CIÊNCIAS DA INFORMAÇÃO GEOGRÁFICA

Trabalho efetuado sob a orientação de:

Professor Doutor José I. Rodrigues

2012

UNIVERSIDADE DO ALGARVE

Faculdade de Ciências e Tecnologia

Instituto Superior de Engenharia

*WebSIG* para visualização e  
processamento de dados espaciais  
de interiores de edifícios

**David Elizalde Castells**

PROJECTO

MESTRADO EM GEOMÁTICA

RAMO CIÊNCIAS DA INFORMAÇÃO GEOGRÁFICA

Trabalho efetuado sob a orientação de:  
Professor Doutor José I. Rodrigues

2012

# ***WebSIG* para visualização e processamento de dados espaciais de interiores de edifícios**

## **Declaração de autoria de trabalho**

Declaro ser o autor deste trabalho, que é original e inédito. Autores e trabalhos consultados estão devidamente citados no texto e constam da listagem de referências incluída.

David Elizalde Castells

A Universidade do Algarve tem o direito, perpétuo e sem limites geográficos, de arquivar e publicitar este trabalho através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, de o divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

*Al Profesor Jose I. Rodrigues, por su orientación, disponibilidad y motivación. Por brindarme tan valiosa oportunidad de aprendizaje y transmitir el gusto y la curiosidad por los WebSIG y la programación.*

*A mis Padres, por su fuerza, valentía y determinación que los convierte no solo en grandes personas y padres sino en un ejemplo a seguir y admirar. Por su apoyo y amor constante.*

*A mi Hermano, Nacho, por la increíble persona que eres y lo que aun está por llegar! Por el orgullo que me haces sentir!*

*Especial agradecimiento a mis Padres y Hermano por su incalculable ayuda en la traducción portunhol-español de este proyecto.*

*A Sara, por sacar a relucir lo mejor de mi. Por hacer de cada día un día especial! A tu lado el verano no acaba, y la aventura no ha hecho más que comenzar! Te quiero!*

*Knowledge is of two kinds. We know a subject ourselves, or we know where we can find information on it.*

*SAMUEL JOHNSON*

## RESUMO

Os Sistemas de Informação Geográfica (SIG) oferecem, para além de outras características e potencialidades, a capacidade de permitir a visualização, compreensão, interpretação e análise de informação espacial sob diferentes perspetivas, descobrindo tendências, padrões ou simplesmente proporcionar uma perspetiva espacial que pode resultar de grande utilidade no momento de tomar decisões. Estas características têm motivado a utilização dos SIG como ferramentas de reconhecido valor na gestão de espaços interiores em grandes edifícios.

O progressivo desenvolvimento da Internet e da *Web* tem vindo a proporcionar condições para o aparecimento de novas formas de interação entre utilizadores e informação, tornando possível a criação de ferramentas online, complexas mas flexíveis, que facilitam a gestão e divulgação da informação.

A integração de aplicações no domínio dos SIG na *Web* está na origem do conceito de *WebSIG* sobre o qual se desenvolve o presente projeto. Os *WebSIG*, permitindo a utilização através de navegadores *Web* de funcionalidades próprias dos SIG proporcionam uma plataforma de acesso à informação geográfica independentemente dos sistemas operativos.

O principal objetivo do presente trabalho consistiu no desenvolvimento e implementação de uma aplicação *WebSIG*, chamada *MapUAlg*, que pretende servir como instrumento de divulgação e gestão dos espaços da Universidade do Algarve. Esta aplicação consiste numa interface de visualização, consulta e análise dos espaços exteriores (campus e edifícios) e interiores (laboratórios, salas, gabinetes, anfiteatros, etc.) da Universidade. Permite navegação interativa sobre a informação espacial bem como a consulta e atualização da informação alfanumérica associada. A realização de pesquisas temáticas, como por exemplo localização de salas de estudo, bares, anfiteatros ou pelo número de sala, conjuntamente com a possibilidade de consultas de estatísticas relacionadas com a utilização dos espaços, constituem algumas das principais ferramentas implementadas nesta aplicação.

A aplicação *MapUAlg* inclui também uma componente de visualização tridimensional dos edifícios. Para o efeito, foi implementada uma projeção geométrica paralela que, quando aplicada aos pisos de cada edifício produzem uma visualização em perspetiva. Para maior versatilidade e interatividade, o utilizador dispõe da possibilidade de definir a “direção de observação”. Esta perspetiva permite a visualização simultânea dos vários pisos que compõem um edifício, verticalmente alinhados, o que facilita a identificação das relações espaciais verticais entre espaços. Esta forma de visualização induz ainda uma mais fácil compreensão da localização de elevadores e escadas bem como das relações espaciais entre estes acessos verticais e os pisos servidos.

A aplicação informática *MapUAlg* foi integralmente desenvolvida para o presente projeto, utilizando única e exclusivamente componentes e ferramentas de carácter livre e de código aberto –*open source*–, proporcionando uma solução sem custos de licenciamento.

A plataforma escolhida para a implementação da aplicação foi *MapFish/Pylons*. Esta plataforma de desenvolvimento está orientada conforme o paradigma *Model-View-Controller* (MVC), segundo o qual há uma separação entre a lógica de negócio, os dados e a interface de utilização. Esta abordagem permite a utilização e/ou reutilização de componentes, reduzindo o tempo de desenvolvimento das aplicações. A organização modular da aplicação possibilita a incorporação de novas funcionalidades e capacidades.

Através da componente *Mapfish* é possível a criação de serviços *Web* de dados espaciais através dos quais se realiza o acesso aos objetos espaciais. É de salientar que estes serviços de dados, graças à natureza das ferramentas empregues, respeitam os standards propostos pelo Open Geospatial Consortium (OGC), concretamente os do serviço WFS (*Web Feature Service*).

A interface gráfica da aplicação foi integralmente desenvolvida a partir de ferramentas básicas para criação de páginas *Web*: *HTML* e *CSS*, com recurso à linguagem de programação *JavaScript*. As várias componentes da interface: manipulação e apresentação da informação espacial, os controlos que permitem a realização de operações próprias dos SIG bem como a estruturação dos conteúdos

visuais dentro do navegador *Web* foram programados com recursos disponíveis nas livrarias *OpenLayers*, *ExtJS* e *GeoExt*.

Os dados espaciais e alfanuméricos relativos aos espaços da Universidade do Algarve, usados no âmbito da aplicação *MapUAlg* foram organizados e carregados numa base de dados espaciais implementada em *PostgreSQL/PostGIS*. É através deste gestor de base de dados objeto-relacional que se realizam também as operações de consulta e atualização dos dados.

A informação espacial foi extraída a partir dos planos arquitetónicos dos edifícios da Universidade, disponíveis em formato digital (DWG) ou analógico (papel). Os dados gráficos passaram por um processo de georreferenciação, conversão de formatos e estruturação de modo a incorporar as adequadas geometrias e relações topológicas. Para além deste processo, os dados em suporte analógico passaram por um processo de digitalização (vetorização).

A informação alfanumérica associada a cada um dos espaços (laboratórios, anfiteatros, gabinetes, salas, áreas de circulação, elementos de estrutura, etc.) foi digitalizada a partir de fontes analógicas e levantamentos no local e classificada segundo um sistema de classificação de espaços previamente definido. Para além dos aspetos funcionais, a base de dados incorpora informação acerca dos responsáveis e utilizadores dos referidos espaços.

A aplicação *MapUAlg* implementa, também, um sistema de autenticação de utilizadores a partir do qual é permitido o acesso às diferentes funcionalidades, segundo três níveis de permissões: público, institucional e administrador. Os utilizadores públicos podem consultar a informação básica dos espaços de cada edifício bem como realizar pesquisas, fundamentalmente de localização. Os utilizadores institucionais têm, adicionalmente, acesso a informação mais completa dos espaços e a possibilidade de analisar estatísticas de utilização dos edifícios. Por último, os administradores podem editar e atualizar informação alfanumérica dos espaços. As atualizações espaciais são realizadas fora da aplicação, com recurso a softwares de SIG desktop.

Na atual fase do projeto, a aplicação *MapUAlg* apresenta estabilidade e funcionalidade suficiente, e está em condições para entrar em fase de produção,

encontrando-se provisoriamente disponível através do endereço  
<http://193.136.227.170>.

**PALABRAS CHAVE:**

SIG, *WebSIG*, Espaços interiores, *Software open source*, MapFish

## RESUMEN

Los Sistemas de Información Geográfica (SIG) ofrecen, entre otras, la capacidad de visualizar, interpretar y analizar información espacial bajo diferentes perspectivas, lo que ha motivado su uso como herramientas de reconocido valor para la gestión de espacios interiores en grandes edificios e instalaciones. El progresivo desarrollo de Internet y la *Web* viene proporcionando la aparición de nuevas formas de interacción entre usuarios e información, haciendo posible la creación de herramientas online que facilitan su gestión y divulgación.

La integración de aplicaciones SIG en la *Web* da origen al concepto *WebSIG*, que permitiendo la utilización a través de un navegador *Web* de funcionalidades propias los SIG y por tanto abstrayéndose del sistema operativo, proporcionan una plataforma de acceso a información geográfica para un elevado número de usuarios.

El principal objetivo de este trabajo fue el desarrollo de una aplicación *WebSIG*, llamada *MapUAlg*, que pretende servir como instrumento de divulgación y gestión de los espacios de la Universidade do Algarve. Esta permite la navegación interactiva sobre la información espacial de espacios –exteriores e interiores– de la Universidad, así como la consulta y actualización de la información alfanumérica asociada. Las búsquedas temáticas o por número de sala, conjuntamente con la consulta de estadísticas relacionadas con la utilización de los espacios, son algunas de las principales herramientas implementadas en esta aplicación *Web*.

*MapUAlg* incluye también un sistema de visualización tridimensional de los edificios, implementado por medio de una proyección geométrica paralela que aplicada a los pisos de cada edificio produce una visualización en perspectiva, pudiendo los usuarios definir la “dirección de observación”. Esto permite la visualización simultánea de los varios pisos que componen un edificio verticalmente alineados derivando en una mayor facilidad de lectura y facilitando la interpretación de las relaciones verticales entre los espacios así como la identificación de los accesos verticales.

### **PALABRAS CLAVE:**

SIG, *WebSIG*, Espacios interiores, *Software open source*, MapFish

## **ABSTRACT**

Geographical Information Systems (GIS) offer, beyond others, visualization, interpretation and analysis of spatial information under different perspectives. They can be a very useful tool when it comes to managing buildings and facilities indoor. The continuous development of internet and the *Web* encourages the emergence of new ways of interaction between people and information performed by complex data management online tools.

The combination of the *Web* and GIS gave birth to the *WebGIS* concept, which enables *Web* browsers with GIS capabilities, abstracting from the operative system, to provide an access platform to geographical information to a high number of internet users.

The main goal of this project is the development of a *WebGIS* application, titled *MapUAlg*, which aims to become a publishing and management tool for the University spatial information. This application allows an interactive navigation through the University dependencies as well as querying and updating the alphanumeric associated information. Thematic or room number searches, together with usage related statistics consultation are some of the main tools implemented in the application.

*MapUAlg* also incorporates a three-dimensional visualization of the buildings, implemented through a parallel geometric projection, applied to each building floor that performs a perspective visualization from which users can choose the “observation direction”. This allows the simultaneous view of several floors in a building, vertically aligned, that converts into an easier reading and interpretation of the vertical relations among the dependencies and identification of the vertical access.

### **KEYWORDS:**

GIS, WebGIS, Indoor areas, Open source software, MapFish

# TABLA DE CONTENIDOS

RESUMO .....	vi
RESUMEN .....	x
ABSTRACT.....	xi
ACRÓNIMOS .....	xvii
<b>1 INTRODUCCIÓN.....</b>	<b>1</b>
1.1 OBJETIVOS Y CONTRIBUCIONES DE ESTE TRABAJO.....	6
1.2 METODOLOGÍA Y ARQUITECTURA DE LA APLICACIÓN .....	7
1.3 ORGANIZACIÓN DEL PROYECTO .....	8
<b>2 CONCEPTOS PREVIOS .....</b>	<b>10</b>
2.1 SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIG).....	10
2.2 INTERNET, WEB Y WEBGIS.....	12
2.3 SOFTWARE LIBRE Y DE CÓDIGO ABIERTO.....	14
2.4 PARADIGMA CLIENTE / SERVIDOR.....	15
2.5 SERVICIOS DE DATOS Y ESTÁNDARES ABIERTOS .....	16
2.6 APLICACIONES WEBSIG .....	18
2.6.1 Paradigma MVC.....	19
2.6.2 Componentes de la arquitectura informática.....	20
2.6.2.1 Controller: El servidor.....	21
2.6.2.2 Model: Los datos. Almacenamiento, estructura y acceso.....	23
2.6.2.3 View: La interfaz de uso .....	27
<b>3 APLICACIÓN MAPUALG .....</b>	<b>30</b>
3.1 DATOS .....	32
3.1.1 Información espacial.....	32
3.1.2 Organización y estructuración de la información alfanumérica.....	36
3.2 BASE DE DATOS.....	38
3.3 ARQUITECTURA DE LA SOLUCIÓN .....	40
3.4 FUNCIONES DE LA APLICACIÓN .....	41
3.4.1 Funciones y usuarios .....	41
3.4.2 Interfaz de uso.....	43
3.4.2.1 Sitemap (mapa Web) .....	43
3.4.2.2 Wireframe (esquema de página).....	45

3.4.3	<i>Funciones públicas</i> .....	46
3.4.3.1	Navegación por los campus y edificios .....	46
3.4.3.2	Navegación por los espacios de los edificios .....	48
3.4.3.3	Visualización de la información parcial de los espacios.....	48
3.4.3.4	Realización de búsquedas.....	49
3.4.4	<i>Funciones para usuarios institucionales</i> .....	50
3.4.4.1	Visualización de la información completa de los espacios.....	50
3.4.4.2	Estadísticas .....	51
3.4.5	<i>Funciones de Administrador</i> .....	52
3.4.5.1	Edición de datos alfanuméricos .....	52
3.5	SERVIDOR.....	54
3.6	ACTUALIZACIÓN DE LOS DATOS ESPACIALES.....	57
<b>4</b>	<b>VISUALIZACIÓN TRIDIMENSIONAL.....</b>	<b>59</b>
4.1	MODELO DE PROYECCIÓN.....	60
4.2	IMPLEMENTACIÓN DE UNA PROYECCIÓN PARALELA ORTOGONAL.....	64
4.2.1	<i>Asignación de alturas, h, a los pisos.....</i>	<i>67</i>
4.2.2	<i>Resultados de la proyección paralela ortogonal.....</i>	<i>70</i>
4.3	IMPLEMENTACIÓN DE LA PROYECCIÓN EN LA APLICACIÓN MAPUALG.....	71
<b>5</b>	<b>CONCLUSIONES .....</b>	<b>73</b>
5.1	RESULTADOS.....	73
5.2	PROBLEMAS Y PRERREQUISITOS PARA EL DESARROLLO DE LA APLICACIÓN .....	74
5.3	TRABAJO FUTURO .....	74
<b>6</b>	<b>BIBLIOGRAFÍA .....</b>	<b>76</b>
<b>ANEXOS .....</b>	<b>78</b>	
ANEXO A	– Sistema completo de clasificación de espacios por uso .....	79
ANEXO B	– Tablas y atributos de la base de datos .....	81

# ÍNDICE DE FIGURAS

Figura 1.1 - Google Maps como ejemplo de un <i>WebSIG</i> utilizado diariamente por millones de usuarios en todo el mundo.....	2
Figura 1.2 - Página de inicio de la aplicación <i>MapUAlg</i> .....	4
Figura 1.3 - Vista de la visualización de espacios de la aplicación <i>MapUAlg</i> .....	5
Figura 1.4 - Visualización 3d de un edificio de 3 plantas.....	5
Figura 2.1 - Paradigma Cliente / Servidor .....	16
Figura 2.2 - Paradigma <i>Model-View-Controller</i> .....	19
Figura 2.3 - Componentes de la arquitectura informática según el paradigma MVC .....	20
Figura 2.4 - Objeto GeoJSON.....	26
Figura 2.5 - Representación del multipolígono definido en la Figura 2.4 .....	26
Figura 3.1 - Página de inicio de la aplicación <i>MapUAlg</i> con el mapa de los campus seleccionado.....	30
Figura 3.2 - Página de visualización de espacios de la aplicación <i>MapUAlg</i> para usuarios institucionales.....	31
Figura 3.3 - Página de visualización de espacios de la aplicación <i>MapUAlg</i> para administradores.....	32
Figura 3.4 - Plano arquitectónico en papel de un edificio de la Universidad .....	34
Figura 3.5 - Planta arquitectónica en formato DWG de un edificio de la Universidad .....	34
Figura 3.6- Proceso de adquisición y preparación de los datos espaciales.....	35
Figura 3.7- Leyenda de los mapas de espacios interiores.....	36
Figura 3.8 - Diagrama entidad/relación de la base de datos espaciales de <i>MapUAlg</i> .....	39
Figura 3.9 - Arquitectura de software de la solución informática .....	41
Figura 3.10 - Usuarios y autenticación .....	42
Figura 3.11 - Diagrama de contexto .....	43
Figura 3.12 - <i>Sitemap</i> (mapa <i>Web</i> ) .....	44
Figura 3.13 - <i>Wireframe</i> (esquema de página) general.....	45
Figura 3.14 - Selector de edificios .....	45
Figura 3.15 - Menú mapa (izquierda) y menú estadísticas (derecha).....	46
Figura 3.16 - De arriba abajo e izquierda a derecha: Mapas de navegación por los campus y edificios de Gambelas, Penha y Saude .....	47

Figura 3.17 - Mapa de navegación por los pisos de un edificio .....	48
Figura 3.18 - Información básica de un espacio seleccionado.....	49
Figura 3.20 - Búsqueda por número de sala .....	50
Figura 3.20 - Búsqueda temática por uso. Anfiteatros del complejo pedagógico de Gambelas.....	49
Figura 3.21 - Información completa de un espacio.....	50
Figura 3.22 - Visualización de las estadísticas de edificio para los espacios según sus usuarios (arriba) y según sus clases (abajo).....	52
Figura 3.23 - Edición de la información asociada a un espacio. Campos seleccionables (arriba). Sistema de protección contra clasificaciones incorrectas (centro). Posibilidad de añadir o quitar departamentos (abajo).....	53
Figura 3.24- Parámetros enviados en POST para la edición de un espacio.....	54
Figura 3.25 - Esquema de conexión directa de QGIS con la Base de Datos Espaciales .....	57
Figura 4.1 - Proyección de los pisos de un edificio. ....	60
Figura 4.2 - Diagrama de una proyección plana general. ....	61
Figura 4.3 - Proyección paralela (izquierda) y central (derecha). Imágenes adaptadas de I. Carlbom, J. Paciorek (1978). ....	62
Figura 4.4 - Punto P' sobre el nuevo referencial de coordenadas u y v en el plano $\pi$ . 63	
Figura 4.5 - Vista en planta de la situación del plano $\pi$ respecto a los campus de la Universidad. ....	66
Figura 4.6 - Vista en Alzado de la situación del plano $\pi$ respecto a los campus de la Universidad. ....	66
Figura 4.7 - Esquema de asignación de alturas a los pisos de un edificio .....	68
Figura 4.8 - <i>Bounding box</i> de un edificio y su diagonal .....	68
Figura 4.9 - Esquema para cálculo de altura, h, mínima para evitar superposición entre las proyecciones de los pisos de un edificio. ....	69
Figura 4.10 - Proyecciones paralelas ortogonales de un edificio utilizando diferentes planos de proyección definidos por los ángulos Az y V.....	70
Figura 4.11 - Proceso de proyección de los pisos de un edificio.....	72
Figura 4.12 - Representación de un edificio bajo la proyección paralela ortogonal con el control de selección de <i>features</i> activado, utilizando la librería OpenLayers. 72	

## **INDICE DE TABLAS**

Tabla 3.1-Sistema de clasificación de los espacios por uso.....	37
Tabla 3.2- Descripción de los controladores y sus acciones.....	55

# ACRÓNIMOS

AJAX	Assinchronous JavaScript
API	Application Programming Interface
ARPA	Advanced Research Projects Agency
BSD	Berkeley Software Distribution
CAD	Computer Assisted Design
CSS	Cascading Style Sheets
DWG	Drawing
ECMA	European Computer Manufacturers Association
FTP	File Transfer Protocol
GeoJSON	Geographical JavaScript Object Notation
GNU	GNU is Not Unix
GPL	General Public License
GPS	Global Positioning System
HTML	HyperText Markup Language
HTTP	HiperText Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
INSPIRE	Infrastructure for Spatial Information in the European Community
IPCC	Instituto Português de Cartografia y Cadastro
JSON	JavaScript Object Notation
MVC	Model-View-Controller
OGC	Open Geostatandar Consortium
OSGeo	Open Source Geospatial Foundation
OSM	Open Street Maps
RFC	Request for comments
SGBDE	Sistema de Gestión de Bases de Datos Espaciales
SIG	Sistema de Informação Geográfica
SMTP	Simple Mail Transfer Protocol

SQL	Structured Query Language
UAlg	Universidade do Algarve
UML	Unified Modeling Language
URL	Uniform Resource Locator
USB	Universal Serial Bus
WFS	Web Feature Service
WFS-T	Transactional Web Feature Service
WMS	Web Map Service
XML	Extensible Markup Language

# 1 INTRODUCCIÓN

Los Sistemas de Información Geográfica (SIG), desde su aparición en los años 60, no han dejado de evolucionar, convirtiéndose en una herramienta indispensable para la resolución de problemas complejos de planificación y gestión geográfica.

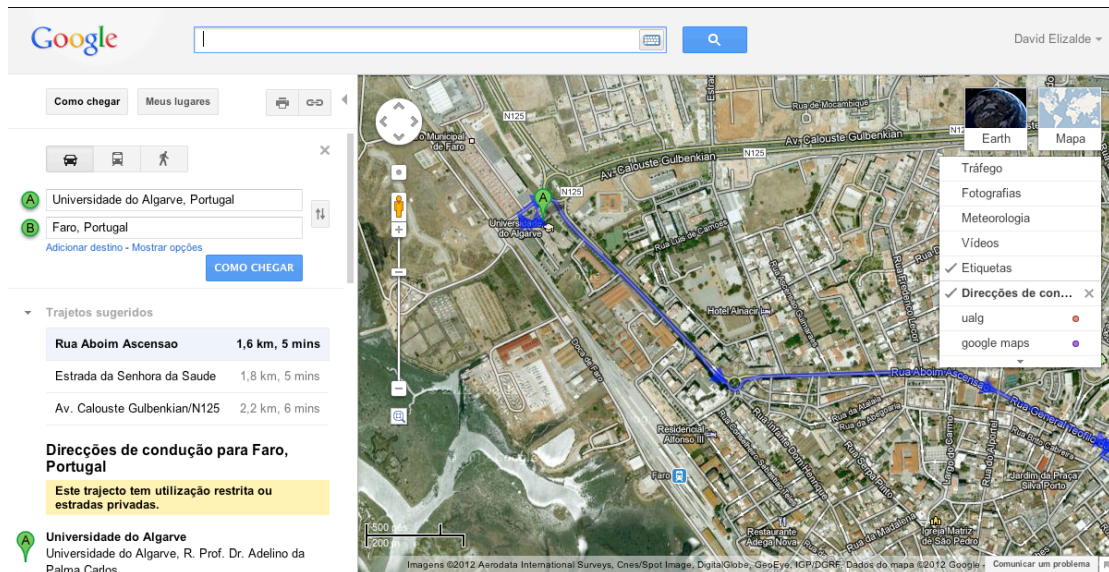
Tradicionalmente, los SIG eran concebidos como unidades de *software* dependientes del sistema operativo y con una reducida capacidad de interoperabilidad (Wong et al, 2002). Se entiende la interoperabilidad como la capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada (IEEE, 1990). El desarrollo de Internet constituyó un desafío a la interoperabilidad, lo que derivó en la exigencia de una solución en lo que al desarrollo de estándares se refiere (Shekhar et al, 2001). Con la creación del primer *WebSIG* surgió un nuevo concepto, capaz de proporcionar las funciones de un SIG en la *Web* a través del protocolo HTTP (Shanzhen et al.,2002; Putz, 1994). Los SIG se están beneficiando enormemente de la *Web*, la cual ha conseguido desbloquear el potencial de estos sistemas para el público en general. Hoy en día la gran mayoría de los usuarios de Internet utilizan aplicaciones de visualización de mapas e información espacial. En la Figura 1.1 se puede ver una captura de la popular página de Google Maps<sup>1</sup>, una aplicación de servicios *Web Mapping* ofrecida por Google, la cual es utilizada cada día por millones de usuarios en todo el mundo.

El uso de Sistemas de Información Geográfica para la gestión de espacios interiores e instalaciones comenzó a desarrollarse en los últimos años, suponiendo una novedosa aplicación de los SIG. Aplicaciones como la planificación o elección de espacios, la gestión de patrimonio, la creación de planes de emergencia o la localización de recursos, servicios y personal, son cada vez más buscadas en los SIG. Estas se aplican a todo tipo de instalaciones, desde hospitales o universidades hasta instalaciones militares y gubernamentales (Olimpo & Smith, 2009).

---

<sup>1</sup> <http://maps.google.pt/>

<sup>2</sup> <http://www.esri.com/software/arcgis/arcgisserver>



**Figura 1.1 - Google Maps como ejemplo de un *WebSIG* utilizado diariamente por millones de usuarios en todo el mundo.**

En la *Web* se pueden encontrar algunas aplicaciones para la visualización y gestión de espacios interiores, principalmente de campus universitarios, dado su interés divulgativo. Aparecen también, cada vez más, herramientas específicamente desarrolladas para la creación de este tipo de *WebSIG*.

Campus Place Finder (ESRI, 2012), es una herramienta diseñada para la creación de aplicaciones *Web* que permite a empleados y visitantes de un campus localizar geográficamente lugares y personal en el interior de los edificios. Esta se basa en el uso de ArcGIS Server<sup>2</sup> y JavaScript.

La Escuela Federal Politécnica de Lausanne, en Suiza, tiene a disposición del público en su página *Web*<sup>3</sup>, una aplicación basada en el uso de MapFish, ExtJS y PostgreSQL/PostGIS para localizar geográficamente, en las dependencias de la Universidad, locales, personas o servicios. La aplicación permite también el cálculo de la ruta más corta entre dos puntos de la Universidad.

Otra interesante utilidad fue presentada por Hui Li y Xiangyang Gong en 2012. Estos autores desarrollaron una aplicación *WebSIG* para *smartphone* que permite calcular el camino para encontrar un libro en una biblioteca. El carácter innovador de esta aplicación reside en la integración de la navegación en espacios exteriores –por medio

<sup>2</sup> <http://www.esri.com/software/arcgis/arcgisserver>

<sup>3</sup> <http://plan.epfl.ch/>

de GPS–, con la navegación en espacios interiores –utilizando la localización por señal Wi-Fi– (H. Li & X. Gong, 2012).

El presente trabajo se encuadra en el contexto de un proyecto anteriormente iniciado con el objetivo de dotar a la Universidade do Algarve con una herramienta para la gestión de sus espacios físicos, en el cual se pueden identificar tres fases principales.

En una primera fase se realizó la identificación de los datos existentes. La mayoría de los planos de los edificios estaban disponibles en soporte digital, en formato DWG. Sin embargo, para algunos edificios fue necesario digitalizar los planos arquitectónicos. La información alfanumérica asociada a los diferentes espacios de la Universidad fue digitalizada a partir de fuentes analógicas y de la recogida de datos en el local. Los datos espaciales digitalizados fueron posteriormente convertidos a un formato vectorial topológico, con topología de polígonos. Se crearon también tablas de atributos que permitieron organizar los datos alfanuméricos asociados a estos polígonos.

Posteriormente, en una segunda fase, la información espacial georreferenciada y debidamente organizada fue incorporada en una base de datos espaciales, cuya estructura se presenta en la sección 3.2. Esta base de datos espaciales es una de las principales componentes del proyecto, y constituye un instrumento de apoyo para la gestión de la propia Universidad. En ella los datos están organizados según dos criterios: la identificación del uso de cada espacio y su asociación a las entidades responsables y usuarias.

La información contenida en esta base de datos quedará asociada al sistema integrado de gestión de la Universidade do Algarve, permitiendo unir la gestión de los espacios con la gestión financiera de la Universidad. Además de constituir un elemento de gestión, la base de datos de los espacios interiores de la UAlg permitirá el acceso a información de interés práctico para los miembros de la comunidad académica y sus visitantes gracias a los interfaces de consulta y análisis que se desarrollaron en la tercera fase.

La tercera fase se corresponde con el desarrollo e implementación de una aplicación informática en ambiente *Web*, con una interfaz gráfica para consulta y visualización

de datos espaciales y alfanuméricos así como para diversas funciones de análisis de los mismos. Esta aplicación se presentará con detalle en el capítulo 3.

A la aplicación informática se le llamó *MapUAlg*<sup>4</sup>, siendo por este nombre como se designará a lo largo de este documento. Esta permite a los usuarios realizar, a través de un navegador *Web*, operaciones propias de los SIG, como la navegación y visualización de los campus, edificios y espacios. Permite también la consulta y edición de la información asociada a los diferentes espacios y la visualización de estadísticas relacionadas con su utilización en un edificio.

En la Figura 1.2 se puede ver la actual página de inicio de la aplicación MapUAlg – con el mapa del campus de Gambelas seleccionado para visualizar y acceder a sus edificios–. Desde esta página se permite al usuario registrarse y visualizar los campus y los edificios de la Universidad. La página de inicio sirve también de puerta de acceso a la página de la aplicación para la visualización y consulta de los espacios, cuyo interfaz se muestra en la Figura 1.3.

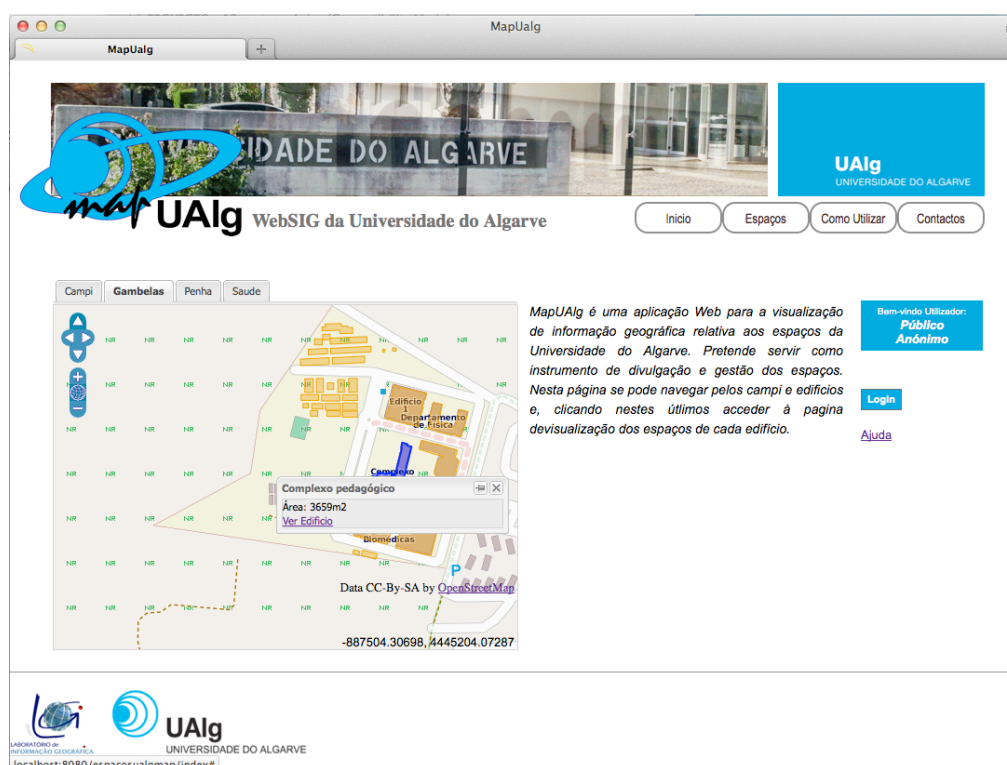
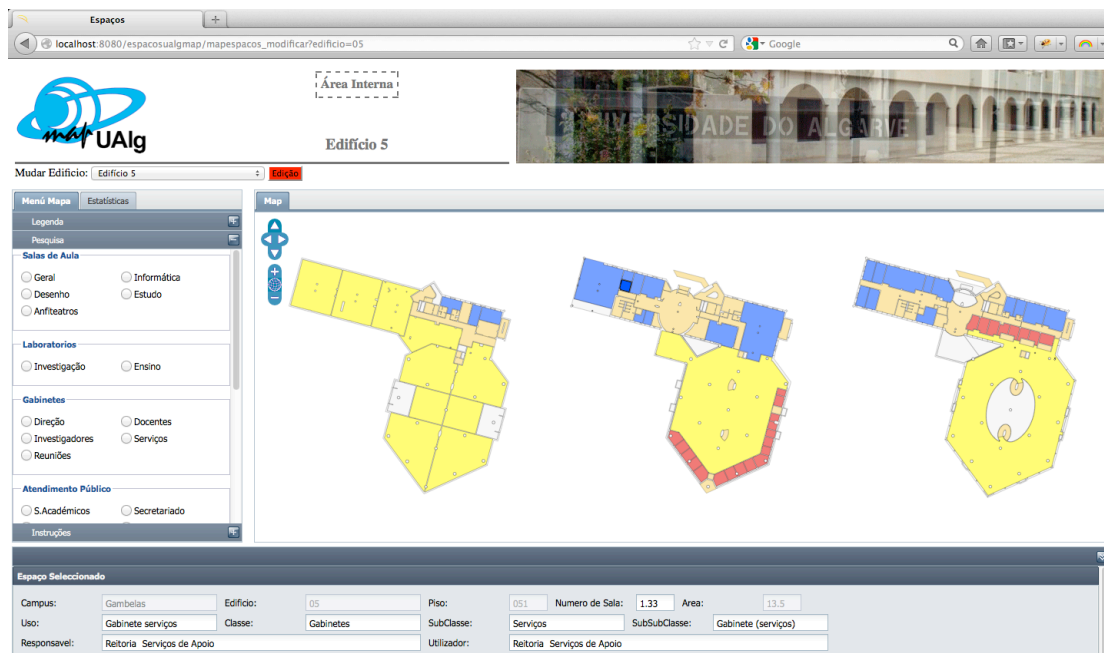


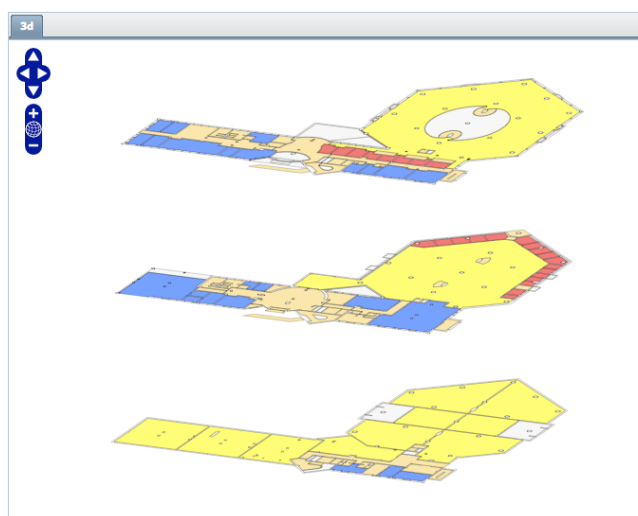
Figura 1.2 - Página de inicio de la aplicación *MapUAlg*

<sup>4</sup> <http://193.136.227.170/sigualg>



**Figura 1.3 - Vista de la visualización de espacios de la aplicación *MapUAlg***

Además de la representación en planta de los pisos de los edificios, se creó también un sistema paralelo de visualización de la información espacial de los edificios, con la particularidad de simular una perspectiva de planos superpuestos. Este sistema de visualización pretende ofrecer un método, de cierto modo innovador, para la visualización de información tridimensional a través de la *Web* creando perspectivas tridimensionales representables sobre sistemas pensados exclusivamente para las dos dimensiones. En la Figura 1.4 se muestra un ejemplo de representación tridimensional de un edificio y en el Capítulo 4 se presenta con detalle este sistema de visualización.



**Figura 1.4 - Visualización 3d de un edificio de 3 plantas**

## 1.1 OBJETIVOS Y CONTRIBUCIONES DE ESTE TRABAJO

El objetivo final de este proyecto es la creación de una herramienta que, tomando la forma de una base de datos espacial y una aplicación *WebSIG*, sirva como instrumento de divulgación y gestión de los espacios de la Universidad, constituyendo así un mecanismo que pueda ser empleado tanto por los servicios internos como por los funcionarios en la gestión de los espacios.

La recopilación y la posibilidad de acceso a la información geográfica de la Universidad en un mismo sistema supone una enorme ventaja en muchos aspectos, pudiéndose encontrar, gracias al carácter modular y flexible de la base de datos y la aplicación *Web* desarrollada, una gran variedad de usos y aplicaciones. Desde el nuevo alumno que no conoce las instalaciones y necesita encontrar la localización de un edificio o aula, hasta los servicios de gestión interna que necesitan saber cual es el área utilizada por un determinado departamento, para por ejemplo, poder ponderar los gastos relacionados con el consumo de electricidad entre otros. Por medio de un sistema de información como el desarrollado en este trabajo se torna posible por ejemplo: (i) localizar determinados tipos de aulas, laboratorios, despachos u otros espacios; (ii) ayudar en la elaboración de horarios y la asignación de los espacios para las clases; (iii) crear planes de emergencia y de accesibilidad; (iv) gestionar el patrimonio y los recursos humanos; entre muchas otras aplicaciones prácticas.

Desde el punto de vista técnico, tal vez el objetivo con carácter más innovador de este proyecto es la generación de un sistema de visualización de los pisos de los edificios en perspectiva tridimensional, utilizando para ello herramientas que tradicionalmente son destinadas a la visualización de información en dos dimensiones.

La aplicación informática está desarrollada bajo una perspectiva modular y se basa en el paradigma *Model-View-Controller* (ver sección 2.6.1), el cual permite la incorporación futura de nuevos componentes y funciones. Se trata de una aplicación dirigida principalmente al público en general, alumnos y visitantes que además de la localización de un aula, anfiteatro o servicio, pueden necesitar información práctica de navegación en el interior de un edificio. Esta funcionalidad podrá ser incorporada

en sistemas de navegación *indoor* –en interiores– para discapacitados visuales (Serrão, Rodrigues, Rodrigues e du Buf, 2012).

Destacar que para la implementación de la aplicación informática *WebSIG MapUAlg*, y la base de datos espacial se utilizó *software* libre y *open source*. Esto supuso otro de los grandes objetivos de este proyecto, dada la reducción de los costes de creación y mantenimiento.

Resulta necesario resaltar también cómo la solución desarrollada puede ser fácilmente adaptada para otro tipo de aplicaciones o ambientes, como pueden ser la gestión de instalaciones hospitalarias, hoteles, empresas públicas o privadas, industrias y otros grandes edificios.

## **1.2 METODOLOGÍA Y ARQUITECTURA DE LA APLICACIÓN**

Para la elaboración de este proyecto fue realizado un proceso de análisis previo de la información que pudiese resultar útil a los potenciales usuarios de la aplicación, resultando de ahí el modelo de datos que posteriormente se integraría en el *WebSIG*. Posteriormente se analizaron cuales podrían ser las funciones de utilidad para los futuros usuarios y cuales las herramientas que permiten implementarlas.

Como resultando de este análisis, se optó por el *framework* MapFish como solución para el servidor. Se trata de una extensión del servidor de aplicaciones *Web Pylons*, escrita en el lenguaje Python y que permite la creación de servicios de datos espaciales con las funciones de interrogación, edición y actualización. MapFish es compatible con los estándares, propuestos por el organismo Open Geospatial Consortium (OGC): WMS, WFS y GML entre otros, algunos de los cuales fueron adoptados para el desarrollo de este trabajo.

Las populares librerías de JavaScript OpenLayers, ExtJS y GeoExt fueron utilizadas en la implementación de la interface de la aplicación Web, ya que soportan los estándares adoptados y permiten también la creación de diseños y estructuras *Web* complejos y profesionales.

La base de datos se implementó en PostgreSQL, un sistema de gestión de bases de datos relacionales (SGBDR) desarrollado como proyecto de software libre y abierto,

siendo hoy en día uno de los SGBDR más avanzados. Gracias a su extensión PostGIS se hace posible la incorporación en la base de datos de información espacial así como de funciones específicas para datos georreferenciados.

En la primera fase del proyecto se utilizó el *software* CMS IntelliCAD<sup>5</sup> para la edición y georreferenciación de los planos de los edificios en soporte digital (formato DWG) así como para la conversión analógica-digital de los planos de los pisos. El *software desktop* de SIG Manifold GIS<sup>6</sup>, un potente, moderno y completo SIG que apuesta por precios de licenciamiento muy reducidos, sirvió para la conversión de los formatos vectoriales (CAD) en formatos vectoriales topológicos (SIG) así como para crear y completar las tablas de atributos. Los *software* mencionados y utilizados en esta fase del proyecto, a pesar de ser propietarios, suponen alternativas de bajo coste – sin perder por eso en potencialidad y rendimiento–, a los *software* propietarios más populares en este dominio actualmente, como son AutoCAD o ArcGIS.

La programación de la aplicación *MapUAlg* se realizó utilizando el lenguajes de programación Python para los componentes del servidor, y JavaScript, HTML y CSS para los del cliente. Se hizo uso de la ayuda del entorno de desarrollo Aptana Studio 3 y su extensión PyDev para la programación en Python, ambos de carácter gratuito y *open source*. Firebug, una extensión del navegador *Web* Firefox, fue imprescindible en la depuración del código JavaScript.

### 1.3 ORGANIZACIÓN DEL PROYECTO

Este proyecto se organiza en 6 capítulos. El primero, de introducción, pretende contextualizar y ofrecer una visión general sobre el trabajo realizado.

El segundo capítulo se destina a los conceptos previos, explicando los principales elementos que forman parte del marco en el que el proyecto se encuadra. SIG, Web y *WebSIG*, como conceptos fundamentales en el desarrollo de este proyecto, son ilustrados desde sus orígenes hasta su importancia hoy en día en la gestión de espacios. Se desarrollan aquí también las principales herramientas, modelos y arquitecturas empleadas en el desarrollo del proyecto.

---

<sup>5</sup> <http://www.intellicad.org>

<sup>6</sup> <http://www.manifold.net>

En el tercer capítulo se describe con detalle la aplicación *MapUAlg*. Los datos utilizados, la creación de la base de datos espaciales que los contiene, la arquitectura informática de la solución, el sistema de usuarios y finalmente, las funcionalidades implementadas en la aplicación, son los temas aquí tratados.

El cuarto capítulo está destinado al desarrollo del sistema de visualización en perspectiva de los edificios, donde se detallan cuáles son los modelos matemáticos utilizados para la implementación de las transformaciones geométricas necesarias para generar las vistas tridimensionales.

En el quinto capítulo se desarrollan las conclusiones. Se explica cuáles son las líneas de trabajo futuro para la mejora de la aplicación, presentando también un análisis de los resultados, de los problemas y de las limitaciones encontradas.

En el sexto capítulo se encuentran las referencias bibliográficas utilizadas tanto en la elaboración de la aplicación *Web* como en la redacción de este trabajo.

## 2 CONCEPTOS PREVIOS

Este capítulo tiene como objetivo presentar una visión general sobre los conceptos y elementos en los que este proyecto se basa, así como de las herramientas utilizadas, a través de las cuales se hace posible el desarrollo de este trabajo.

Se introducen los Sistemas de Información Geográfica, Internet y la *Web* como conceptos base, puesto que de su conjunción nacen los *WebSIG*, tema principal de este proyecto. Se hace alusión también a los conceptos de software libre y de código abierto dada que la mayoría de las herramientas utilizadas se identifican con esos dos términos. Los servicios de datos y los estándares abiertos como elementos inherentes al proyecto desarrollado son también expuestos en este capítulo. Se explica el modelo cliente/servidor, en el cual se fundamenta cualquier aplicación *Web*. Por último, y a través del paradigma *Model-View-Controller*, el cual sostiene la arquitectura de la aplicación *WebSIG* creada, se revisan los componentes y herramientas utilizadas en la elaboración de la aplicación informática.

### 2.1 SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIG)

Un Sistema de Información Geográfica es, tal y como Bonham-Carter (1994) definió, “un sistema informático para la gestión de datos espaciales”. Ya anteriormente Star & Estes (1990) habían definido cómo los SIG están diseñados para trabajar con datos referenciados por coordenadas espaciales o geográficas, convirtiéndose así en sistemas de bases de datos con capacidades específicas para datos espacialmente referenciados así como para un conjunto de operaciones y análisis orientados a la manipulación de este tipo de información.

Si nos remontamos a los orígenes de los SIG, llegaremos a Norteamérica donde, en la década de los 60 y con el comienzo de la era informática, se empezó a redefinir el análisis cartográfico. Las crecientes necesidades a nivel de gestión urbanística, junto a la vertiginosa velocidad con que la computación avanzaba, forjó la aparición de los primeros Sistemas de Información Geográfica.

Desde entonces estos no han parado de evolucionar paralelamente al desempeño de los ordenadores, especialmente en lo referente a las capacidades gráficas.

En su constante evolución, los SIG se han convertido en una herramienta indispensable para diversas aplicaciones tales como: la ordenación territorial, la planificación del uso del suelo, el modelado de ecosistemas y hábitats, el análisis criminológico, el análisis de mercados, y un sinfín de utilidades relacionadas con la información espacial.

Pinde Fu (2010) actualizó el concepto de los SIG definiéndolos como: “un sistema de hardware, software y procedimientos que permiten capturar, almacenar, editar, manipular, gestionar, analizar, compartir y visualizar datos georreferenciados”. Resaltando cómo las capacidades de los SIG van mucho más allá de la creación de mapas, indicó que: “Los SIG ofrecen un vasto conjunto de funciones analíticas que permiten revelar relaciones, patrones y tendencias que a simple vista se encuentran ocultas, facilitando así el pensamiento espacial en la resolución de problemas y la toma de decisiones”.

La mayoría de los *software desktop* de SIG corren de manera individual en ordenadores personales y son tecnologías destinadas a un público reducido. Estos requieren una gran inversión en tiempo y trabajo, tanto para la adquisición de datos espaciales como para el aprendizaje de su uso, para lo que es necesario tener amplios conocimientos sobre SIG. Estas herramientas –*software SIG desktop*– son prácticamente imprescindibles en el pre-procesado de los datos espaciales que posteriormente formarán parte de los *WebSIG*.

Se puede considerar el desarrollo de Internet y de la Web como uno de los más recientes pilares de la evolución de la civilización, dando acceso a la nueva era de la información, la cual ha hecho que cambiemos radicalmente la forma en que vivimos, trabajamos y nos relacionamos. Los SIG no se quedaron atrás, y de su fusión con las tecnologías *Web* nace lo que conocemos como *WebSIG*, principal elemento de este proyecto.

A continuación se introducirán los conceptos de Internet y la *Web* para así poder llegar a los *WebSIG*.

## 2.2 INTERNET, WEB Y WEBGIS

En los años 60, durante la guerra fría, el departamento de defensa de los Estados Unidos ARPA (Advanced Research Project Agency), creado para recuperar el liderazgo tecnológico frente a los soviéticos, inició un proyecto de investigación destinado a crear una red de ordenadores separados geográficamente que pudiesen intercambiar información. Fue en 1969 cuando se consiguió conectar cuatro universidades de la costa oeste, naciendo así, el predecesor de Internet ARPANet.

A finales de 1989, tras un gradual aumento del número de ordenadores que se incorporaban a esta red, existían más de 100.000 ordenadores interconectados.

El uso de esta red global presentaba cierta complejidad, estando por tanto limitado a algunos profesionales de la investigación y de agencias gubernamentales. En 1990 apareció un nuevo modo de usar la red, gracias a la invención por parte de Tim Berners-Lee del HTTP (Hypertext Transfer Protocol), HTML (Hypertext Markup Language) y la URL (Uniform Resource Locator) en un intento de crear de un sistema sencillo para compartir e intercambiar documentos con los compañeros de trabajo. Finalmente fue Berners Lee, quien al crear el primer servidor y navegador *Web*, introdujo el concepto de *World Wide Web*, convirtiendo Internet en algo accesible de utilizar sin la necesidad de grandes conocimientos técnicos.

No se deben confundir los términos Internet y *Web* (*World Wide Web*). Internet es una enorme red de redes que conecta millones de ordenadores en todo el mundo los cuales pueden comunicarse entre sí a través de una amplia variedad de protocolos como HTTP (Hypertext Transfer Protocol), SMTP (Simple Mail Transfer Protocol), FTP (File Transfer Protocol), etc.

La *World Wide Web* es un sistema de programas y documentos de hipertexto entrelazados entre sí a los cuales se puede acceder a través de Internet usando principalmente el protocolo HTTP.

El término *Web 2.0* fue acuñado por primera vez por Darcy DiNucci en 1999 y a pesar de que pueda sugerir una nueva versión de la *World Wide Web*, no se refiere la actualización de ninguna especificación técnica y sí a un cúmulo de cambios en la manera cómo los desarrolladores de software y los usuarios utilizan la *Web*. La *Web*

2.0 aboga por los contenidos generados por los usuarios y los flujos de información inversos, de abajo a arriba, tanto que su éxito se sustenta en la masa crítica de usuarios que comparten información. La *Web* se convierte también, en una plataforma para la computación y el desarrollo de *software*, así como para la aparición del concepto de *software como servicio*, donde las capacidades de los *software* se entregan en forma de servicios *Web*. Además, las aplicaciones *Web 2.0* proporcionan una interfaz gráfica amigable y fácil de usar, ofreciendo experiencias similares a las tradicionales aplicaciones *desktop*.

El desarrollo de Internet en los últimos años ha sido un factor determinante para la evolución de los SIG, permitiendo compartir información espacial de forma directa y descentralizando los servicios de datos espaciales, encontrándose estos disponibles a través de Internet. Probablemente el mayor logro haya sido el integrar las tecnologías SIG con la *Web* creando el concepto de *WebSIG* (Yang, Wong, Yang, Kafatos, & Li, 2004).

Esta combinación permite el acceso a la información espacial, así como a una gran variedad de herramientas de análisis y servicios de datos utilizando un simple navegador *Web* (Peng & Tsou, 910, 2003), consiguiendo así llegar a un mayor número de usuarios (Tang & Selwood, 2003). Según Pinde Fu (2010) la *Web* es la plataforma ideal para los SIG: se complementan mutuamente de manera extraordinaria permitiendo a profesionales, organizaciones y al público en general compartir y colaborar, convirtiendo los SIG en una tecnología cada vez más accesible a un amplio espectro de usuarios y sobre todo, a un bajo coste.

Las aplicaciones *WebSIG* proporcionan diversas funciones y controles a los usuarios, como si de un clásico *software desktop* SIG se tratase. Estas van desde los controles básicos de visualización (*pan and zoom*), selección de capas de datos y visualización de la información asociada a las diferentes entidades espaciales, hasta herramientas de mayor complejidad como la medición de distancias y áreas o la realización de análisis espaciales que pueden requerir un mayor nivel de procesamiento de datos (Drummond & French, 2008).

Algunos de los más populares *software open source* del lado del servidor para la creación de aplicaciones *WebSIG* son: GeoServer, MapServer y Mapfish. Como

*software* propietario destacar la importancia de ArcGIS Server. Las APIs (*Application Programming Interface*) orientadas al *Web Mapping* del lado del cliente surgieron en 2005 y son ahora imprescindibles en cualquier *WebSIG*, puesto que no solo permite proporcionar al usuario un gran control sobre los mapas sino que también simplifica la combinación *–mashups–* de diferentes fuentes de datos. OpenLayers es probablemente la API *open source* para *Web Mapping* más popular en el momento.

### 2.3 SOFTWARE LIBRE Y DE CÓDIGO ABIERTO

Vale la pena reservar esta sección para explicar los conceptos de software libre y de código abierto –en inglés, y a partir de ahora en este documento, será designado como *open source*–, ya que este trabajo fue desarrollado en su mayoría por medio de este tipo de herramientas. Estos dos términos, dada la creciente importancia y acogimiento de este tipo de software, son actualmente muy utilizados y muchas veces pueden resultar confusos. Esta sección se destina a aclarar estos conceptos y resumir las bases en las que se fundamentan, haciendo también referencia a las licencias bajo las que se distribuyen las herramientas *open source* utilizadas en el desarrollo de la aplicación *MapUAlg*.

Según la *Free Software Foundation*<sup>7</sup>, el *software* libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar y modificar el *software* y redistribuirlo modificado. El *software* libre suele estar disponible gratuitamente, sin embargo no es obligatorio que sea así, pudiendo ser distribuido comercialmente sin perder su carácter libre. Un *software* es libre cuando garantiza las siguientes cuatro libertades: la de usar el programa, con cualquier propósito; la de estudiar su funcionamiento y modificarlo, adaptándolo a necesidades específicas; la de distribuir copias del programa, ayudando así al prójimo; y finalmente la libertad de mejorar el programa y hacer públicas esas mejoras para los demás, de modo a que toda la comunidad se beneficie de las mismas.

---

<sup>7</sup> <http://www.fsf.org>

El término código abierto –*open source*–, fue acuñado por la OSI<sup>8</sup> (*Open Source Initiative*) y está estrechamente ligado al *software* libre. *Open source* se refiere al *software* que respeta las cuatro libertades definidas por la *Free Software Foundation*. Cualquier licencia de *software* libre también lo es de código abierto. La principal diferencia entre estos dos términos esta en su presentación: mientras que el *software* libre se envuelve en un discurso de ética, derechos y libertades, el código abierto se presenta desde un punto de vista puramente técnico, evitando las cuestiones morales.

Existen varios tipos de licencia bajo las cuales se distribuye *software* libre y abierto. Una licencia es una autorización formal con carácter contractual que el autor de un *software* otorga a un interesado para ejercer actos de explotación legales. A continuación se describen las principales licencias basadas en los principios de *software* libre.

La Licencia General Pública o GPL (*General Public Licence*) es una de las más utilizadas: aproximadamente el 60% del *software* licenciado como libre emplea una licencia GPL. Esta posibilita la modificación y redistribución del *software*, pero únicamente bajo esa misma licencia.

La Licencia estilo BSD (*Berkeley Software Distribution*) es la licencia de *software* otorgada principalmente para los sistemas BSD. Esta tiene menos restricciones si se compara con la GPL, encontrándose muy cercana a las licencias de dominio público. La licencia BSD, al contrario que la GPL, permite el uso del código fuente en *software* no libre.

## 2.4 PARADIGMA CLIENTE / SERVIDOR

Típicamente las aplicaciones *WebSIG* presentan una arquitectura basada en el modelo cliente/servidor, donde la presentación de los datos, el procesado y la lógica de aplicación son procesos separados tanto lógica como físicamente. Existen tres niveles básicos en este modelo: Presentación, aplicación y datos.

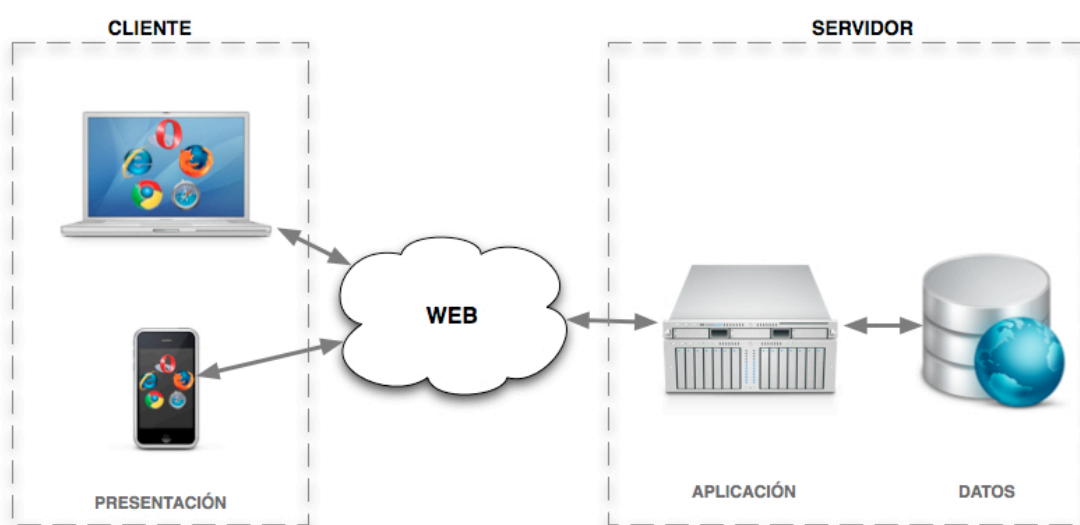
En el servidor se encuentra alojada la base de datos, a la cual se puede acceder a través de Internet. El nivel de aplicación representa un servidor de aplicaciones al cual

---

<sup>8</sup> <http://opensource.org/>

se pueden conectar de manera simultánea varios clientes y solicitar datos y contenidos, encargándose este de la comunicación con la base de datos.

Estos dos niveles pueden estar, como en el caso de este proyecto, implementados en un mismo servidor físico. La capa o nivel de presentación representa al cliente y puede estar localizada en cualquier ordenador o dispositivo que acceda vía Internet al servidor. En el caso de una aplicación *WebSIG*, el cliente estará normalmente representado por un navegador *Web*. En la Figura 2.1 se puede ver representada la arquitectura cliente/servidor así como la identificación de sus niveles de presentación, aplicación y datos.



**Figura 2.1 - Paradigma Cliente / Servidor**

## **2.5 SERVICIOS DE DATOS Y ESTÁNDARES ABIERTOS**

La capacidad de no solo compartir recursos sino también de cruzarlos, proporcionando así aplicaciones aun más útiles, supone un enorme beneficio para gobiernos, negocios y público en general. Con el desarrollo de Internet, tareas que tradicionalmente eran realizadas localmente por ordenadores personales, son ahora llevadas a cabo en grandes centros de computación conectados a Internet y disponibles como Servicios *Web* –*Web-based services*–.

Los Servicios *Web* utilizan un conjunto de protocolos y estándares que sirven para el intercambio de datos entre distintos elementos de *software*, que pueden estar

desarrollados en lenguajes de programación diferentes y ser ejecutados sobre cualquier plataforma. Esta interoperabilidad se consigue mediante la adopción de estándares. A continuación se describen algunas de las iniciativas que más han contribuido a la creación de estándares abiertos en lo que a información geográfica se refiere.

El Open Geospatial Consortium<sup>9</sup> (OGC) se fundó en 1994 como una organización de estándares internacionales voluntariamente consensuados para datos geoespaciales y servicios *Web*. Actualmente, 462 compañías, agencias gubernamentales y universidades participan en el proceso de consenso, desarrollando públicamente estándares de interfaz. Los estándares OGC respaldan las soluciones de interoperabilidad que “geo-habilitan” la *Web*, los sistemas inalámbricos, los servicios basados en localización y las principales tecnologías de la información. Los estándares permiten a los desarrolladores crear información y servicios espaciales complejos, accesibles y utilizables en todo tipo de aplicaciones. *Web Map Service* y *Web Feature Service* son dos de los estándares creados por la OGC que se utilizan en este proyecto.

*Web Map Service* (WMS) es un protocolo estándar, publicado en 1999, para servir a través de Internet capas geográficas de un área de interés por medio de imágenes mapa georreferenciadas generadas por un servidor de mapas y utilizando una base de datos espaciales. OpenStreetMaps utiliza este protocolo para servir los mapas que, como se verá más adelante, se utilizan en este proyecto.

*Web Feature Service* (WFS) es un protocolo estándar que ofrece un interfaz de comunicación que permite servir capas geográficas de un área de interés por medio de datos vectoriales. El WFS-T (*Transactional Web Feature Service*) permite la creación, eliminación y actualización de las entidades geográficas.

La iniciativa Infraestructura para la Información Espacial en la Comunidad Europea, INSPIRE<sup>10</sup> (*Infrastructure for Spatial Information in the European Community*), fue creada en 2001 por la Comisión Europea para promover la disponibilidad de la información de naturaleza espacial, utilizable en la formulación, implementación y evaluación de las políticas de la Unión Europea. Esta iniciativa pretende establecer un

---

<sup>9</sup> <http://www.opengeospatial.org/>

<sup>10</sup> <http://inspire.jrc.ec.europa.eu/>

marco legal para la creación gradual y armonizada de una infraestructura europea de información geográfica centrada en las políticas ambientales y con previsión de expansión para otros sectores como la agricultura o transportes. La creación de INSPIRE permite hacer disponibles servicios integrados de información de naturaleza espacial basados en una red distribuida de bases de datos conectadas por medio de normas y protocolos comunes que aseguren su compatibilidad. Estos servicios permitirán a cualquier usuario identificar y acceder a información espacial proveniente de diferentes fuentes de un modo interoperable. Resaltar la entrada como directiva europea<sup>11</sup> de la iniciativa INSPIRE en 2007 y su inclusión en la ley portuguesa<sup>12</sup> en 2009.

## **2.6 APLICACIONES *WEBSIG***

La *Web* era en sus inicios un lugar donde la información publicada era mayoritariamente estática y almacenada en ficheros que eran enviados a los clientes. Con la llegada de las *Web* dinámicas, al concepto de página *Web* se le sumó el de aplicación *Web*, refiriéndose a aquellas páginas que no solo proporcionaban información en tiempo real, como meteorología, sino que permitían la actualización de los contenidos y la interacción de los usuarios con la información.

Según Shklar, Leon (2003), se puede definir una aplicación *Web* como una aplicación basada en el modelo cliente/servidor, siendo el cliente un navegador *Web* que proporciona un servicio interactivo mediante la conexión con servidores de Internet (o Intranet). Las aplicaciones *Web* proporcionan contenidos dinámicamente diseñados en base a parámetros de una petición, comportamientos rastreados de los usuarios y aspectos de seguridad.

Con los *WebSIG* se indica una nueva forma de distribución de datos en los sistemas de información geográfica. La forma más elemental de un *WebSIG* es, como mínimo, un servidor y un cliente, donde el servidor es una aplicación y el cliente puede ser un navegador *Web*, una aplicación *desktop* o una aplicación móvil. Por tanto, una

---

<sup>11</sup> Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE) OJL 108, 25.4.2007, p. 1-14

<sup>12</sup> Decreto-Lei n.º 180/2009, de 7 de Agosto, Diário da República n.º 152, 1ª Série.

aplicación *WebSIG* es un SIG que utiliza la tecnología *Web* para comunicarse entre sus componentes basándose en el modelo cliente/servidor.

Las mayoría de las herramientas disponibles actualmente para la creación de aplicaciones *Web* basan su implementación en el paradigma *Model-View-Controller*, MVC (en español Modelo-Vista-Controlador), el cual resulta de gran utilidad a la hora de implementar sistemas de *software* interactivos ya que separa la interfaz de usuario de la lógica de negocio y de los datos (Leff, A.; Rayfield, J.T, 2001).

### 2.6.1 Paradigma MVC

El paradigma *Model-View-Controller*, MVC, define una arquitectura de diseño de *software* que pretende separar los datos de la interfaz de usuario y de la lógica de negocio, haciendo posible cambiar una sin afectar a la otra. Esto posibilita la construcción de aplicaciones flexibles y de más fácil mantenimiento. Permite, por ejemplo, cambiar completamente la apariencia de la aplicación sin necesidad de modificar las estructura de datos o la lógica de negocio y permite incluso la existencia de varios interfaces diferentes, como múltiples idiomas, o diferentes apariencias según los permisos del usuario (Leff, A.; Rayfield, J.T, 2001).

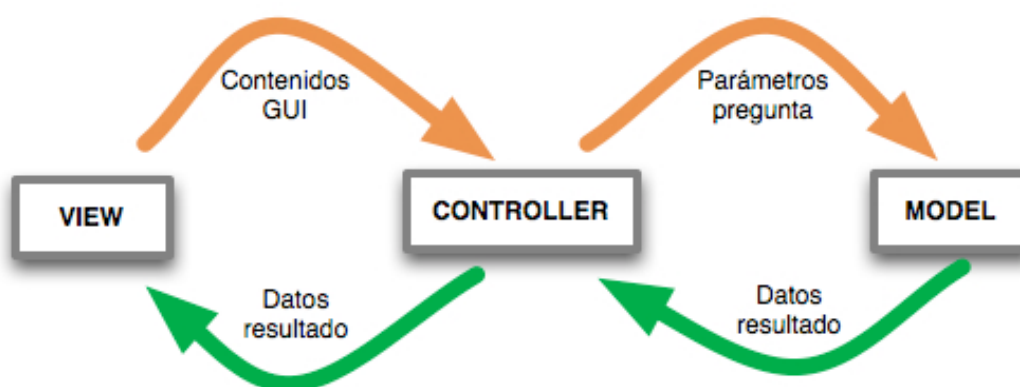


Figura 2.2 - Paradigma *Model-View-Controller*

En la Figura 2.2 se puede ver un diagrama del modelo MVC que muestra las relaciones entre el modelo, el controlador y la vista –*Model, Controller, View*–.

El *Model* representa los datos y las posibles conexiones con la base de datos y el *View* genera la interfaz con el usuario, que en el caso de un aplicación *Web* está

representada por el navegador *Web*. El controlador es el elemento que une todo, encargándose de la comunicación entre la vista y el modelo.

## 2.6.2 Componentes de la arquitectura informática

Para la implementación de la aplicación *MapUAlg* fue necesario escoger un conjunto de herramientas que se adaptasen tanto a las necesidades del proyecto como a los medios disponibles, siendo las opciones preferidas las que se presentan en esta sección.

Dado que la estructura de la aplicación *WebSIG* desarrollada está basada en el modelo de arquitectura informática MVC, se pueden organizar sus diferentes componentes en los tres grupos –*Model*, *View* y *Controller*– del paradigma.

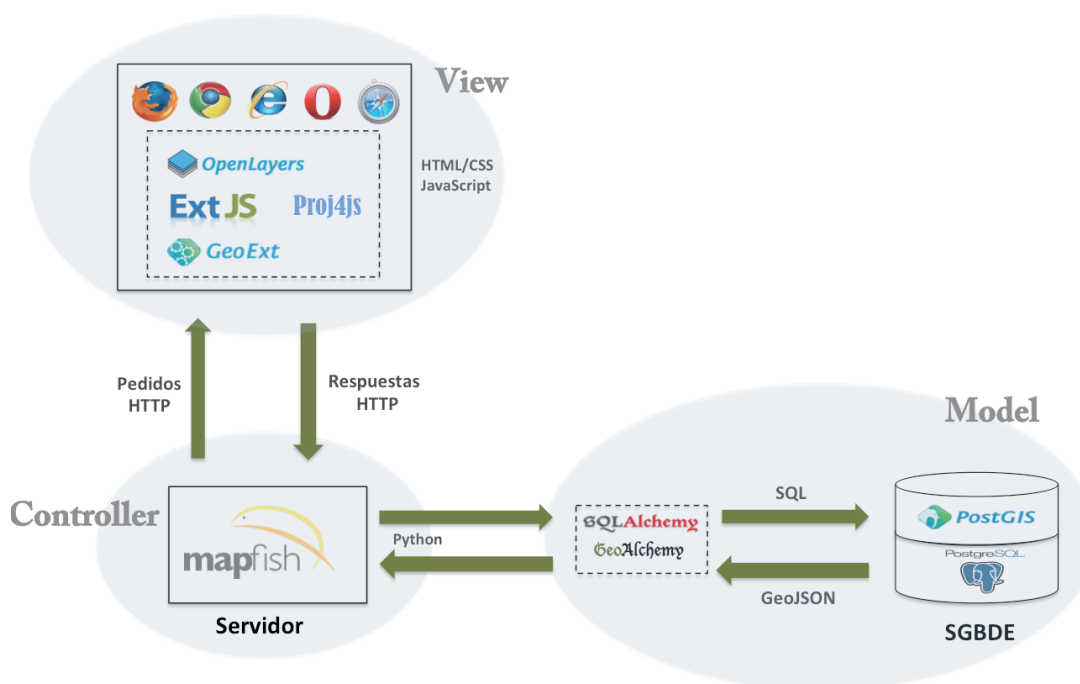


Figura 2.3 - Componentes de la arquitectura informática según el paradigma MVC

En la Figura 2.3 se pueden ver representados estos componentes agrupados según el paradigma MVC en *Model*, *Controller* y *View*. En el *Controller* se engloban: el servidor de aplicaciones *Web* MapFish; Python como lenguaje de programación de este servidor; y el protocolo HTTP, responsable por las comunicaciones *Web*. El *Model* representa todo lo relacionado con los datos: Su almacenamiento en la base de datos espacial desarrollada en PostgreSQL y PostGIS; su estructura en los diferentes formatos como GeoJSON o GML; y su acceso por medio del lenguaje SQL y la

librería SQLAlchemy. Por último en el *View* se agrupan los diferentes elementos que permitieron crear el interfaz con el usuario: Los lenguajes de diseño y programación *Web* –HTML, CSS– y las herramientas JavaScript –OpenLayers, GeoExt, ExtJS y Proj4js– que permiten el manejo y visualización de información alfanumérica y espacial.

### **2.6.2.1 Controller: El servidor**

El *Controller* se presenta como el principal responsable por el funcionamiento de la aplicación *Web*. Materializado principalmente como el servidor de aplicaciones *Web* MapFish, se encarga, como su nombre indica, de controlar los eventos y las relaciones entre el *Model* –los datos– y el *View* –la interfaz de uso–. En lenguaje de programación en el que MapFish está escrito, y por tanto las acciones que controlan las relaciones anteriormente mencionadas, es Python. Por último se puede englobar en el *Controller* el protocolo HTTP, responsable del intercambio de información en la *Web*, y en este caso, entre el *Controller* y el *View*.

#### MapFish

Por proporcionar un interesante conjunto de herramientas para el manejo de información espacial así como por su carácter *Open Source* y gratuito, se utilizó la extensión MapFish<sup>13</sup> del servidor de aplicaciones *Web* Pylons<sup>14</sup>.

Desde el año 2007 CampToCamp desarrolla MapFish, que basado en Pylons, es un *framework* para desarrollo *Web* escrito en Python, el cual fue adaptado para soportar tareas geoespaciales como la creación de servicios *Web* bajo los estándares OGC. MapFish fue específicamente diseñado para facilitar la conexión a una base de datos espacial PostgreSQL/PostGIS y devolver datos geográficos en formato GeoJSON. El *framework* se distribuye gratuitamente bajo una licencia LGPL y es compatible con los sistemas operativos de Microsoft Windows, Linux y Mac OS.

Existe un excelente manual del servidor Pylons (James Gardner, 2009), que, junto al tutorial de MapFish disponible en su página *Web* (MapFish, 2011) se convierte en la principal herramienta de referencia tanto para principiantes como para expertos.

---

<sup>13</sup> <http://mapfish.org>

<sup>14</sup> <http://www.pylonsproject.org>

El servidor MapFish está escrito en Python, lenguaje de programación explicado a continuación.

### Python

Python es un lenguaje de programación de alto nivel que se destaca por su sintaxis limpia, favoreciendo el código legible. Es un lenguaje multiparadigma, lo que significa que permite a los programadores adoptar el estilo de programación preferido, bien sea orientado a objetos, imperativo, funcional u otros estilos soportados a partir de extensiones. En el desarrollo de este trabajo, dada la estructura y programación del servidor *Web* MapFish, se adoptó una programación orientada a objetos. Esta se basa en el uso de clases, responsables por la creación de objetos, y sus funciones asociadas, que llevan a cabo las diferentes acciones que cada objeto puede realizar (van Rossum, 2005). En el ambiente de MapFish, se designa a las clases como controladores y a los métodos como acciones. Los controladores, a través de sus acciones, permiten el acceso a los datos, su procesado y su envío al *View*.

Python es considerado como una de los lenguajes preferidos para la programación de tareas reaccionadas con información geoespacial debido a su alto rendimiento y a la amplia variedad de herramientas de alto nivel desarrolladas para tal efecto (Westra, 2010). Su creación se remonta a finales de los años ochenta por Guido van Rossum en los Países Bajos. Destacar el manual de Python escrito por Guido van Rossum (2005) por su facilidad de comprensión y abordaje completo del Lenguaje de programación Python; así como el escrito por Erik Westra (2010), el cual habla de la utilidad del lenguaje para aplicaciones geoespaciales. Actualmente Python ya se encuentra en su versión 3 y se distribuye bajo una licencia propia de código abierto compatible con la licencia pública general de GNU.

### Protocolo HTTP

Resulta oportuno en esta sección hablar del *Hypertext Transfer Protocol* o HTTP (en español protocolo de transferencia de hipertexto), en el cual se basan las comunicaciones que soportan el concepto de la *Web*. En la aplicación desarrollada en este trabajo es el responsable por el intercambio de información entre el *Controller* y el *View* siendo el servidor de aplicaciones *Web* MapFish –*Controller*–, quien se

encarga de gestionar los pedidos y las respuestas HTTP. El protocolo está basado en transacciones y sigue un esquema de petición-respuesta entre un cliente y un servidor. HTTP fue desarrollado por el *World Wide Web Consortium* y la *Internet Engineering Task Force*, culminando en 1999 con la publicación del RFC 2616<sup>15</sup> que especifica la versión 1.1.

#### **2.6.2.2 Model: Los datos. Almacenamiento, estructura y acceso.**

El *Model* representa los datos. El sistema de almacenamiento de datos de la aplicación *Web* presentada es una base de datos espaciales desarrollada en PostgreSQL y PostGIS. El acceso a estos datos se realiza por medio del lenguaje SQL y con la ayuda de la librería de Python SQLAlchemy. Los datos se organizan según diferentes estructuras y formatos, siendo GeoJSON el preferido en esta aplicación. También se hablará del formato GML por su relevancia en el contexto de este proyecto.

#### *Bases de Datos Espaciales - PostgreSQL / PostGIS*

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados para su posterior uso. Las bases de datos relacionales son aquellas que cumplen con el modelo relacional, el cual permite establecer interconexiones entre los datos (guardados en tablas) pudiendo así relacionar datos de diferentes tablas. Los sistemas de gestión de bases de datos relacionales (SGBDR) son programas que permiten almacenar, acceder y modificar los datos de forma rápida y estructurada. PostgreSQL es el SGBDR utilizado en este proyecto. Su extensión PostGIS permite la habilitación espacial de PostgreSQL.

PostgreSQL<sup>16</sup> es un SGBDR *open source* que proporciona las características y rendimientos de los más avanzados SGBDR propietarios, como Oracle. La Universidad de California en Berkely fue responsable del inicio de su desarrollo en 1986, pero es en 1996 cuando pasó a formar parte del mundo *Open Source*. En los últimos años, y gracias a una gran comunidad de desarrolladores, se han ido implementando características avanzadas, como la ejecución de *queries* complejas, los

---

<sup>15</sup> <http://tools.ietf.org/html/rfc2616>

<sup>16</sup> [www.postgresql.org](http://www.postgresql.org)

disparadores (*triggers*) o la vistas (*views*). PostgreSQL está desarrollado para los sistemas operativos Microsoft Windows, Linux, Mac OS, FreeBSD y Solaris. Se distribuye gratuitamente bajo una licencia BSD y su mantenimiento está a cargo de la organización EnterpriseDB.

PostGIS<sup>17</sup> es una extensión de PostgreSQL que permite la integración de información espacial georeferenciada, su representación en múltiples formatos y estándares y la realización de complejas operaciones espaciales directamente en la base de datos (Tyler, 2005). Refrations Research comenzó a desarrollarlo en 2001, y junto con OSGeo (*Open Source Geospatial Foundation*) se encarga de su mantenimiento. PostGIS se distribuye gratuitamente bajo una licencia GPL.

PostgreSQL y PostGIS se encuentran perfectamente documentados ya que además de disponer de completas referencias oficiales en sus páginas *Web* (PostgreSQL Global Development Group, 2012), existen diversos manuales y guías, entre los cuales cabe destacar el escrito por Regina O. Obe, Leo S. Hsu (2011) por su enfoque práctico e su variedad de ejemplos de utilización.

### Acceso a los Datos

Resulta necesario, teniendo una base de datos, el poder efectuar consultas y modificaciones a los mismos con el fin de recuperar y actualizar de forma sencilla la información de interés.

El *Structured Query Language*, SQL (Lenguaje de Consulta Estructurado) es un lenguaje de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones. El manejo del álgebra y el cálculo relacional hace posible la realización de consultas con el fin de recuperar o modificar la información de interés en una base de datos. SQL es el lenguaje *de facto* para la mayoría de los SGBD tanto comerciales como *open source*.

SQLAlchemy<sup>18</sup> es una librería *open source* de Python destinada a servir como mapeador de objetos relacionales. Proporciona un conjunto de herramientas que permiten el acceso a las bases de datos por medio de un lenguaje simple y “Pythonico” con un alto rendimiento. De esta manera SQLAlchemy abstrae al

---

<sup>17</sup> <http://postgis.refrations.net/>

<sup>18</sup> <http://www.sqlalchemy.org/>

programador del lenguaje SQL, generándolo automáticamente para acceder a la base de datos. Esta librería, junto con GeoAlchemy<sup>19</sup> –una extensión de SQLAlchemy que permite trabajar con bases de datos espaciales– está incluida en el paquete de *software* de MapFish, por lo que su utilización resulta completamente integrada con el resto de los elementos del servidor.

### Estructuras de Datos

Las estructuras de datos son formas de organizar conjuntos de datos de manera a facilitar su manipulación. Las estructuras de datos espaciales organizan los datos referentes a la información espacial y las geometrías de una forma jerárquica. En este proyecto, GeoJSON fue la estructura de datos espacial preferida, dada su fácil integración con las herramientas utilizadas. GML es uno de los principales formatos estandarizados por la OGC y soportado también por los elementos que integran la aplicación desarrollada.

GeoJSON es un formato abierto utilizado para codificar estructuras de datos geográficas. Está basado en el formato JSON (*JavaScript Object Notation*) y es utilizado por MapFish en la codificación de la información espacial. En el ejemplo de la Figura 2.4 se puede ver en detalle una de las entidades –*features*– perteneciente al objeto GeoJSON –*FeatureCollection*, que representa un piso de un edificio–. Se puede observar como en la estructura de datos están definidas las propiedades del *feature* – que contiene todos sus atributos alfanuméricos– y su geometría, en la cual se especifica el tipo –*MultiPoligon* en este caso– y las coordenadas de los puntos que la conforman.

Las coordenadas siguen una estructura de listas anidadas, tal y como se observa en la Figura 2.4, pudiendo así definir tanto los varios polígonos que forman un multipolígono como los polígonos formados por uno exterior y uno o más interiores que forman “huecos” en el polígono exterior. La Figura 2.5 muestra la representación gráfica del multipolígono definido en la estructura GeoJSON de la Figura 2.4. Se trata de una pared construida alrededor de columnas formando así 2 polígonos, uno de ellos con un “hueco”.

---

<sup>19</sup> <http://geoalchemy.org/>



transvase de información entre los diferentes *software* que hacen uso de información geográfica.

### **2.6.2.3 View: La interfaz de uso**

La interfaz con el usuario de la aplicación *MapUAlg* es un página *Web* que utilizando un conjunto de imágenes, objetos gráficos y estructuras de presentación de datos, representa la información y las acciones disponibles a los usuarios. Su objetivo es proporcionar un entorno visual sencillo y completo para permitir la comunicación del usuario con la aplicación.

Las principales herramientas empleadas por el *View* para el desarrollo de los interfaces de utilización de la aplicación *MapUAlg* están escritas e implementadas en diferentes lenguajes que son interpretados por la mayoría de los navegadores *Web* de última generación. HTML y CSS son los lenguajes básicos en la programación de páginas *Web*. Son pues, los responsables de la organización y representación de los contenidos *Web*. JavaScript es el lenguaje de *scripting* que permite la creación de aplicaciones *Web* dinámicas.

MapFish propone un *Framework*, compuesto por OpenLayers, GeoExt y ExtJS, que presenta un enorme potencial a la hora de diseñar aplicaciones *Web* orientadas a la información geográfica así como de crear complejos interfaces de con el usuario. OpenLayers se encarga de soportar los componentes relacionados con los datos espaciales así como los controles sobre los mapas generados, ExtJS permite la creación de elementos complejos para aplicaciones *Web*, como paneles, tablas, formularios y *layouts*; mientras que GeoExt permite la integración de OpenLayers y ExtJS sacando así el máximo partido de ambas tecnologías.

#### HTML/CSS

HTML (*HyperText Markup Language*), o Lenguaje de Marcado de Hipertexto, es un lenguaje de marcación utilizado para la producción de páginas *Web*. Los documentos HTML pueden ser interpretados por los navegadores *Web* generando así la estructura y contenidos de la página.

CSS (*Cascading Style Sheets*), o sea, Hojas de Estilo en Cascada, es un lenguaje de estilo utilizado para definir la presentación de documentos escritos en un lenguaje de

marcación como HTML. La ventaja que presenta su utilización es la separación entre el formato y el contenido de los documentos.

### JavaScript

JavaScript es el lenguaje de programación que actualmente se utiliza para realizar todo tipo de acciones en las páginas *Web* debido a su flexibilidad. JavaScript es considerado un lenguaje de *scripting* ya que su funcionamiento está limitado dentro de un navegador *Web*, donde es interpretado, no teniendo acceso a ficheros locales, bases de datos o elementos de *hardware* (USB, etc.), lo que supondría un enorme riesgo de seguridad.

El principal objetivo para el cual fue desarrollado es la manipulación de páginas *Web*. Netscape es el responsable de su creación en 1995 y a pesar de su nombre, este no tiene nada que ver con el lenguaje de programación Java.

Al lado de JavaScript se encuentra ECMAScript, un lenguaje descrito en un documento creado por la organización ECMA cuando otros navegadores además de Netscape comenzaron a soportar JavaScript. ECMAScript describe el propósito principal del lenguaje de programación pero no define su integración con el navegador *Web*, siendo entonces JavaScript el lenguaje de ECMAScript sumado a un conjunto de herramientas de integración con las páginas y navegadores *Web*.

JavaScript está en continuo desarrollo y, dada la popularidad ganada en los últimos años, se ha convertido también en un lenguaje de *scripting* para otro tipo de aplicaciones *desktop* como Adobe Acrobat y Photoshop o incluso como lenguaje de programación del lado del servidor en herramientas como Node.js.

JavaScript será tal vez, junto de HTML y CSS, uno de los lenguajes de programación más documentados –tanto en Internet como por medio de libros editados– dada su popularidad, lo que dificulta el encontrar referencias de calidad. Eloquent JavaScript (Haverbeke, 2011), por su lenguaje directo y práctico así como por la calidad de sus ejemplos, es una de la más recomendables guías para este lenguaje.

## Bibliotecas y APIs JavaScript

OpenLayers<sup>20</sup> es una biblioteca pura de JavaScript dedicada a la representación de información geográfica en la mayor parte de los navegadores *Web* de última generación, sin depender del servidor. MetaCarta es responsable de su creación en 2005 y desde entonces no ha dejado de atraer a una gran comunidad de desarrolladores que colaboran en su mejora y evolución. OpenLayers presenta un API JavaScript para la construcción de aplicaciones *Web* geográficas complejas que utilizan métodos y protocolos abiertos de acceso a datos geográficos como WMS o WFS y proporcionan una fácil integración con los servicios de mapas más populares como Google, Yahoo, OSM o Bing (Hazzard, 2011). OpenLayers es de libre distribución y está desarrollado por la comunidad *open source* y se distribuye bajo una licencia BSD.

Una de las principales herramientas de consulta y referencia para esta librería es su API (OpenLayers, 2011). No obstante existe una práctica guía (Hazzard, 2011) para principiantes, que con inúmeros ejemplos prácticos, hace de su aprendizaje algo sencillo e intuitivo.

ExtJS<sup>21</sup> es una biblioteca pura JavaScript desarrollada por Sencha y orientada a la creación de aplicaciones *Web* interactivas. Permite la creación de *layouts* y estructuras que emulan las tradicionales aplicaciones desktop y ofrece una potente capacidad de interactuar con datos por medio de elementos como tablas, formularios y gráficas entre otros. Su API (Sencha, 2011), completa y estructurada, es la principal fuente de documentación para esta librería.

GeoExt<sup>22</sup> es otra biblioteca de JavaScript que junta la potencia e elasticidad de OpenLayers con la facilidad de uso de ExtJS, permitiendo así crear complejas aplicaciones SIG para la *Web*.

Proj4js<sup>23</sup> es una biblioteca JavaScript dedicada a la realización de transformaciones entre proyecciones cartográficas. Esta se integra con OpenLayers, facilitando así su utilización y la transformación de información geográfica proveniente de diferentes fuentes y distintos sistemas de proyección.

---

<sup>20</sup> <http://openlayers.org/>

<sup>21</sup> <http://www.sencha.com/products/extjs3/>

<sup>22</sup> <http://geoext.org>

<sup>23</sup> <http://trac.osgeo.org/proj4js/>

### 3 APLICACIÓN MAPUALG

Como resultado del desarrollo del principal objetivo de este proyecto se creó la aplicación *WebSIG MapUAlg*. Esta aplicación informática estará disponible al público a través de cualquier navegador *Web*. En el momento de entrega del presente trabajo se puede acceder a la primera versión de la aplicación *MapUAlg* por medio de la URL: <http://193.136.227.170/signalg>. Estando previsto la incorporación de un acceso directo en la página principal de la Universidade do Algarve<sup>24</sup>.

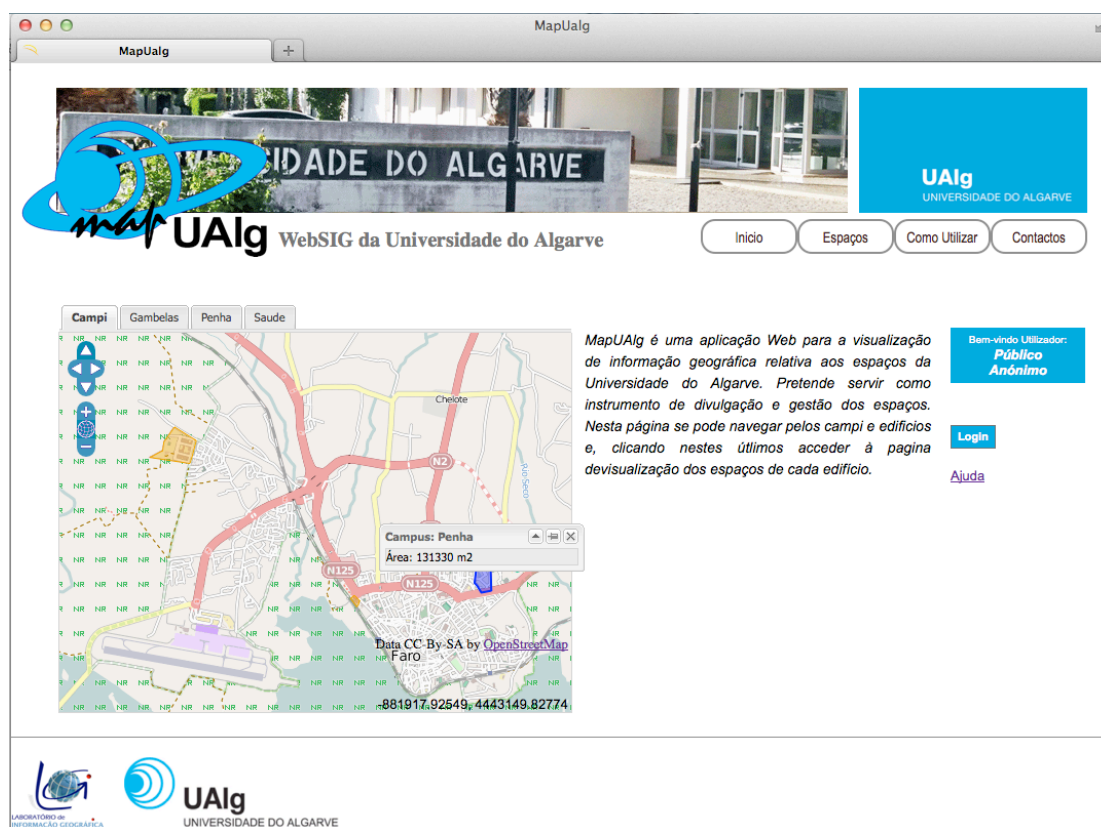
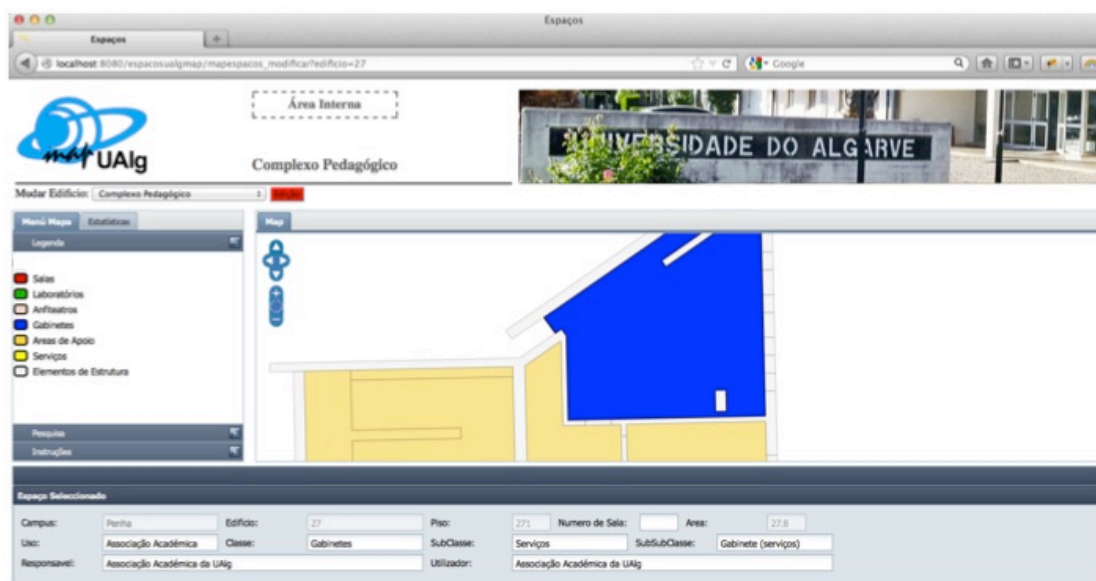


Figura 3.1 - Página de inicio de la aplicación *MapUAlg* con el mapa de los campus seleccionado.

La aplicación *MapUAlg* permitirá a cualquier usuario acceder a su página principal, la cual se puede ver en la Figura 3.1, donde podrá visualizar, además de información general sobre el proyecto, cuatro mapas: uno con los 3 campus y tres con los edificios

<sup>24</sup> <http://www.ualg.pt>

de cada uno de los campus. En la página principal se encuentra disponible un formulario de autenticación por medio del cual se podrán registrar los usuarios institucionales y administradores (ver sección 3.4.1). Será competencia de los administradores del proyecto *MapUAlg* distribuir las contraseñas de acceso una vez la aplicación sea lanzada al público. Desde la página de inicio, los usuarios públicos, es decir sin contraseña, tendrán acceso a la página de la aplicación para la visualización de los espacios de cada edificio, donde podrán navegar por los mapas del interior de los edificios, visualizar información asociada a cada espacio y realizar búsquedas de espacios. El aspecto de esta aplicación se pudo ver en la Figura 1.3. Los usuarios institucionales tendrán acceso a información adicional de cada espacio así como a la consulta de estadísticas de los edificios. En la Figura 3.2 se representa la aplicación de visualización de espacios para usuarios institucionales. Por último, los administradores tendrán la posibilidad de editar la información asociada a los espacios, tal y como se muestra en la Figura 3.3.



**Figura 3.2 - Página de visualización de espacios de la aplicación *MapUAlg* para usuarios institucionales.**

El presente capítulo, destinado a la explicación de la aplicación *MapUAlg*, se organiza de la siguiente manera. La primera sección explica cuál es la información espacial y alfanumérica utilizada y cuál fue el proceso de recopilación y tratamiento de la misma. La base de datos espacial, diseñada e implementada para almacenar, acceder y modificar esa información, se desarrollará en la segunda parte. La tercera sección

especifica la arquitectura de *software* adoptada para la solución. La cuarta se destina a explicar el diseño de la página *Web*, el sistema de usuarios y las distintas funciones de la aplicación. La quinta sección habla del servidor de la aplicación y finalmente, la sexta, está destinada a explicar el proceso de actualización de los datos espaciales.

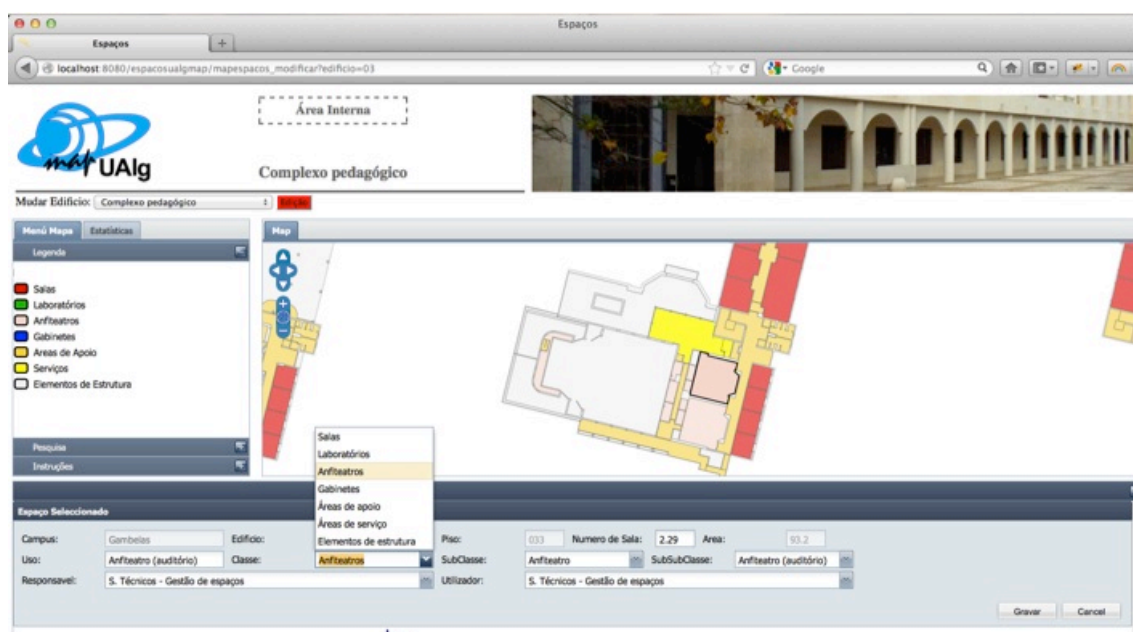


Figura 3.3 - Página de visualización de espacios de la aplicación *MapUAlg* para administradores.

## 3.1 DATOS

La aplicación *MapUAlg* se basa en la divulgación y el análisis de la información espacial, así como de los datos alfanuméricos asociados a esta, de la Universidade do Algarve. La obtención y preparación de estos datos conllevó un largo proceso de recopilación, tratamiento y clasificación de la información, tal y como se describe a continuación.

### 3.1.1 Información espacial

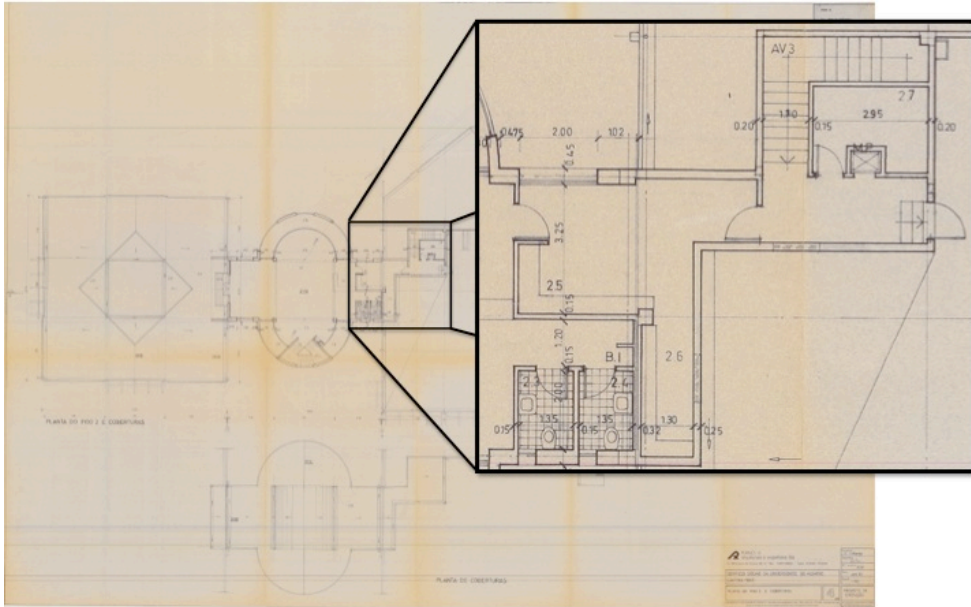
La información espacial necesaria para la realización de este proyecto se puede dividir en dos tipos: áreas exteriores y áreas interiores de los edificios, siendo esta última la que mayor relevancia tiene en el marco de este trabajo. La información

espacial de las áreas exteriores se refiere a los polígonos que representan las áreas de implantación de los edificios, así como las áreas de terreno pertenecientes a la Universidad, las cuales están repartidas en sus tres campus. Estos polígonos se obtuvieron por medio de una vectorización de los mismos sobre una ortofotografía aérea georreferenciada.

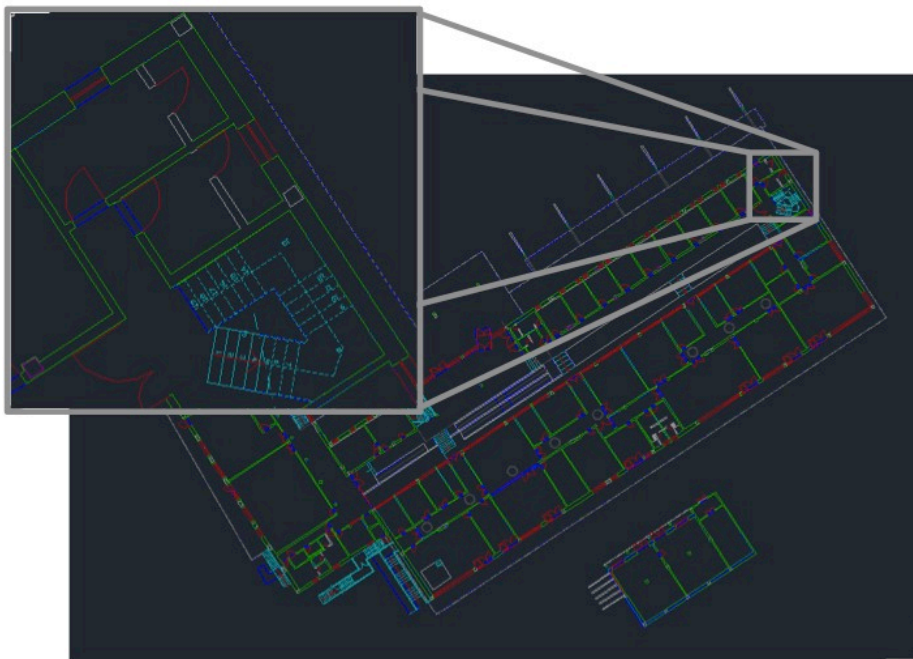
Los interiores de los edificios están compuestos por pisos y estos, a su vez, por espacios delimitados físicamente, principalmente, por paredes. Se determina así la división de los pisos en espacios funcionales, tales como aulas, despachos, laboratorios o elementos de circulación. Los espacios interiores también contemplan a los elementos de la estructura de los edificios, como paredes, puertas o ventanas. A todos estos elementos interiores se les designa a lo largo de este proyecto como “espacios”.

Toda esta información espacial es de naturaleza bidimensional, si bien y como se verá en capítulos posteriores, los pisos llevan asociada la información de su altura, la cual será utilizada para la creación de una perspectiva tridimensional simplificada. Existen otras cuestiones relacionadas con la altura, como los posibles desniveles dentro de un mismo piso, que no son consideradas en los datos utilizados para este trabajo.

Antes del inicio del proyecto de mapeado de los espacios de la Universidade do Algarve, la información espacial se reducía a planos arquitectónicos disponibles en papel, o bien, en el caso de los más edificios más recientes, en soporte digital, en el formato DWG (*Drawing*). La Figura 3.4 representa un plano arquitectónico en papel de un edificio de la Universidad. En la ampliación sobre la figura se puede ver como los elementos arquitectónicos se encuentran dimensionados, lo que permitió la digitalización de las plantas de los edificios. La Figura 3.5 muestra el plano en formato digital (DWG) de la planta de un edificio.



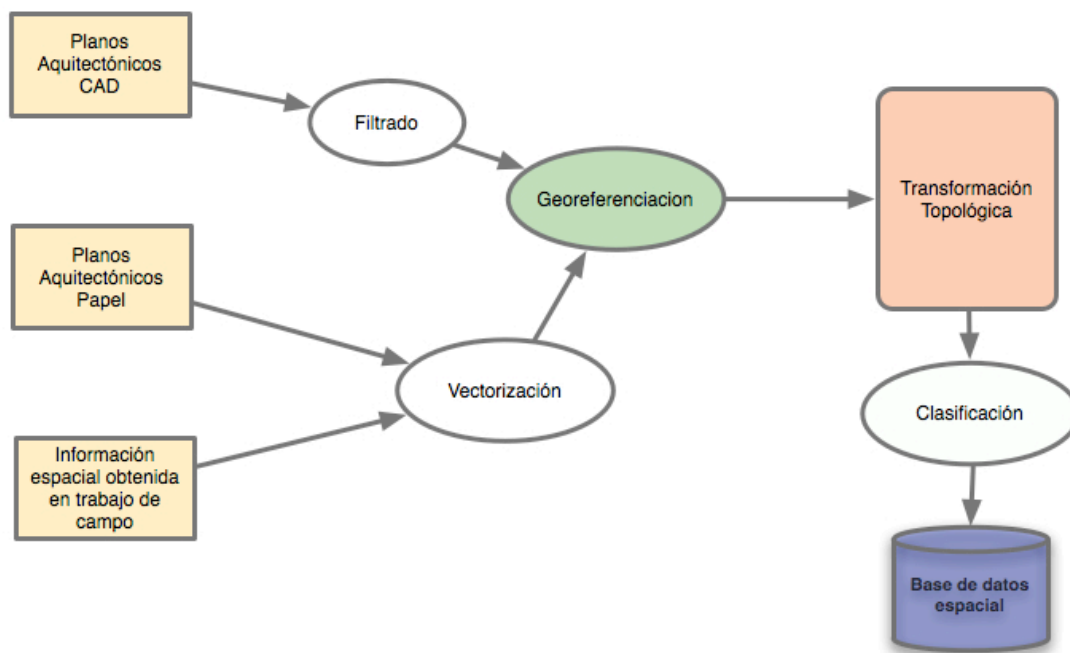
**Figura 3.4 - Plano arquitectónico en papel de un edificio de la Universidad**



**Figura 3.5 - Planta arquitectónica en formato DWG de un edificio de la Universidad**

Para la integración de esta información en un Sistema de Información Geográfica, fue necesario, como se puede ver en la Figura 3.6, un largo proceso de preparación de los datos provenientes de los diferentes formatos. Los planos en papel tuvieron que ser vectorizados y después georreferenciados. A la información disponible en soporte digital se le aplicó un filtrado, a fin de eliminar la información superflua para este

proyecto, para después georreferenciarla también. Seguidamente, a esta información en formato DWG se incorporó a un SIG, donde se aplicó un proceso de estructuración de modo a incorporar las geometrías y relaciones topológicas (polígonos) adecuadas. Una vez se integró la información espacial en un SIG, se asoció a cada espacio un conjunto de datos alfanuméricos conforme el sistema de clasificación que se explica en el próximo apartado, quedando así toda la información completa y preparada para su integración en una base de datos espaciales.



**Figura 3.6- Proceso de adquisición y preparación de los datos espaciales**

Tanto la información espacial disponible en papel como en soporte digital, pertenece a las plantas arquitectónicas de los proyectos de construcción de los edificios, es decir, en muchos de los casos esta información no se corresponde de manera exacta con la realidad, ya que durante los procesos de construcción se llevaron a cabo pequeñas modificaciones respecto a los planos originales. Esto, junto con las obras de modernización y reordenación llevadas a cabo en algunos de los edificios, hizo que fuese necesario realizar un exhaustivo trabajo de campo que permitiese verificar y corregir la información original.

Toda la información espacial de la Universidad fue georreferenciada sobre el sistema cartográfico Datum 73, Hayford Gauss.

### 3.1.2 Organización y estructuración de la información alfanumérica

Una vez recopilada e integrada la información espacial de los espacios de la Universidad en el *software desktop* de SIG Manifold, se procedió a la organización y estructuración de los datos alfanuméricos asociados a esta. Se consideraron dos criterios diferentes de clasificación: el primero basado en la utilización de los espacios, es decir, el uso para el cual están destinados; y el segundo conforme a los responsables de los espacios, diferenciando entre quienes son los responsables y quienes los usuarios de cada espacio.

Con la finalidad de poder identificar el uso de cada uno de los espacios, fue necesario crear un sistema de clasificación jerárquico de tres niveles: Clases, Subclases y Subsubclases. Estos tres elementos definen, con un nivel de detalle progresivo, cuál es el uso al que se destinan los espacios. Las siete clases definidas son: *Salas, Laboratorios, Anfiteatros, Gabinetes, Áreas de Apoyo, Servicios y Elementos de Estructura*.

Estos son las responsables de la clasificación visual de los espacios, es decir, de los colores utilizados para su representación, tal y como se muestra en la leyenda de los mapas representada en la Figura 3.7.



Figura 3.7- Leyenda de los mapas de espacios interiores

Cada Clase se divide en varias Subclases, tal y como se define en la Tabla 3.1. Las Subclases, a su vez, se dividen en Subsubclases, dando lugar al amplio sistema de clasificación según su uso. Este se puede consultar en su totalidad en el Anexo A.

**Tabla 3.1-Sistema de clasificación de los espacios por uso**

<u>Clase</u>	<u>SubClase</u>
<b>Salas</b>	Salas de aula
	Salas de informática
	Sala de estudio
	Sala de dibujo
<b>Laboratorios</b>	Laboratorio de investigación
	Laboratorio de enseñanza ( <i>ensino</i> )
<b>Anfiteatros</b>	Anfiteatro
<b>Gabinetes</b>	Dirección
	Docentes
	Investigadores
	Servicios
	Reuniones/Multiusos
<b>Áreas de apoyo</b>	I. sanitarias
	Servicios
	Depósito ( <i>Arrecadação</i> )
	Apoyo técnico
	Circulación
	Taller ( <i>Oficina</i> )
<b>Áreas de servicio</b>	Atendimento público
	Biblioteca
	Archivo
	Restauración y similares
<b>Elementos de estructura</b>	Espacio técnico
	Estructurales

El segundo criterio de clasificación identifica cuales son los responsables y usuarios de los espacios de la Universidad. Estos incluyen a las Unidades Orgánicas, Centros de Investigación así como a los diferentes órganos de gestión de la Universidad. Los departamentos implicados en el uso de los espacios también fueron identificados e incluidos en esta clasificación.

Con base a los dos criterios mencionados anteriormente se elaboró el siguiente sistema de clasificación de los espacios interiores:

- Cada espacio tiene un único uso definido por una Subsubclase.
- Cada Subsubclase tiene una Subclase y Clase únicas, haciendo que para cada espacio queden definidos tres niveles de uso.
- Cada espacio tiene un único responsable asociado.
- Cada espacio tiene un único usuario asociado y este a su vez un único responsable.
- Cada espacio, en caso de tener una unidad orgánica como responsable, tiene como mínimo un departamento asociado.
- Cada espacio con departamentos asociados tiene un indicador del porcentaje del espacio que utiliza cada departamento, siendo siempre la suma total de los porcentajes del cien por ciento.
- Cada espacio tiene un campo llamado “uso”, destinado a proporcionar información adicional sobre la utilización del espacio.

## 3.2 BASE DE DATOS

Conforme a la información disponible y a los criterios de clasificación explicados anteriormente, se diseñó e implementó una base de datos espacial para almacenar, actualizar y acceder a los datos espaciales de la Universidad. Como se explicó en la sección 2.6.2.2 el SGBDR utilizado fue PostgreSQL junto con su extensión para datos espaciales PostGIS.

Para el diseño de la base de datos y las relaciones entre los diferentes elementos de información se creó un diagrama de Entidad/Relación (DER). Un DER es una herramienta para el modelado de datos en el cual se representan las entidades relevantes en el sistema de información, así como sus interrelaciones y propiedades (o atributos). En la Figura 3.8 se puede ver el DER que representa el modelo de datos utilizado en la aplicación *MapUAlg*, donde los rectángulos se corresponden con las entidades, los óvalos con sus atributos y los rombos representan las relaciones entre entidades, las cuales están condicionadas a ser unitarias o múltiples –simbolizado por un 1 o un N respectivamente– y opcionales u obligatorias –representado por una línea simple o doble respectivamente–.

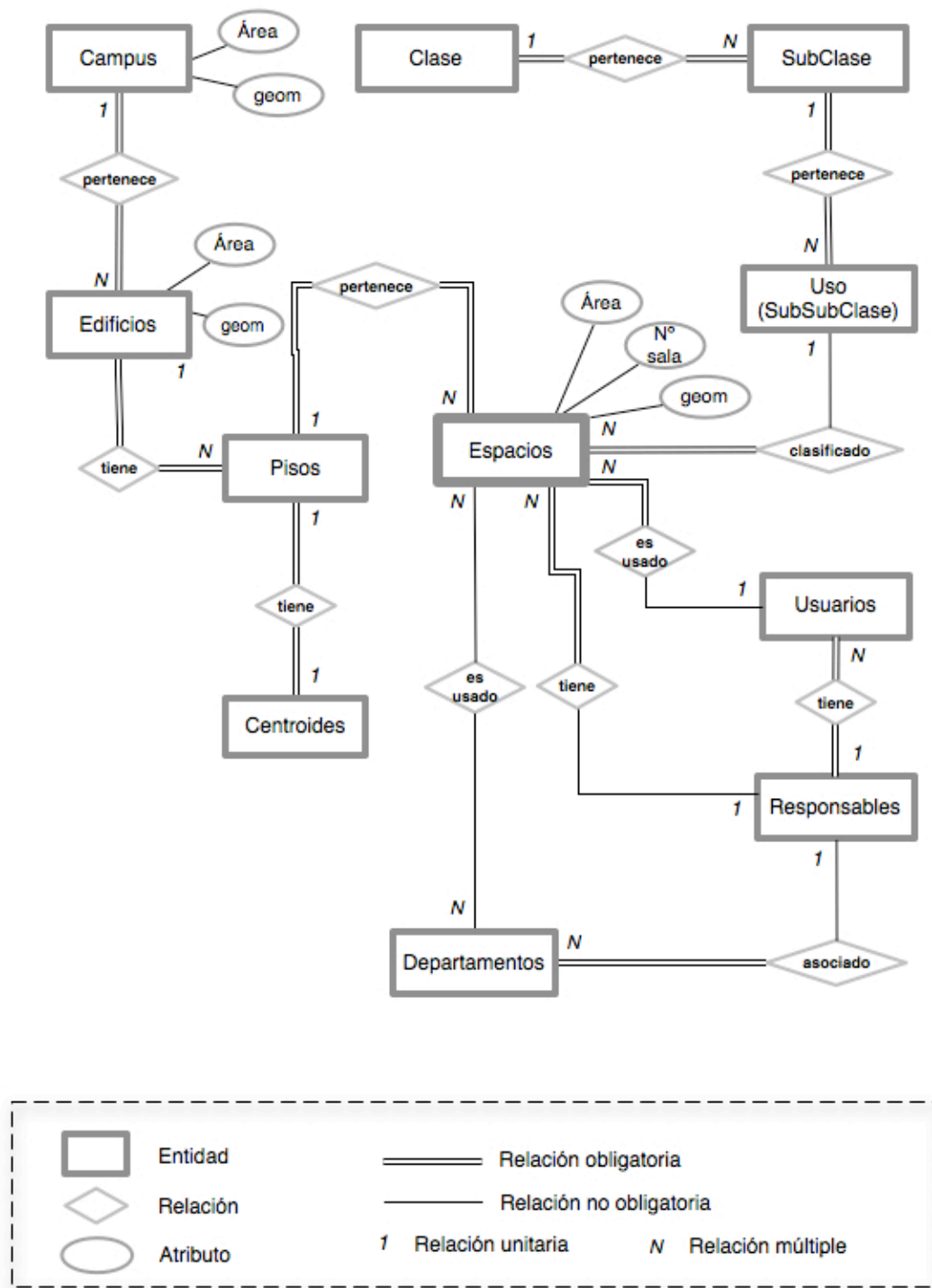


Figura 3.8 - Diagrama entidad/relación de la base de datos espaciales de *MapUAlg*

Como entidad principal, se encuentran los *espacios* de la Universidad, perteneciendo obligatoriamente cada uno de ellos a un único *piso*, que a su vez, pertenece a un único *edificio*, y este a su vez a un único *campus*. *Espacios*, *edificios* y *campus* son las tres entidades que poseen una componente de información espacial, designada como *geom*. Cada *espacio* obligatoriamente se clasifica, mediante un único valor de

*Subsubclase*, que a su vez se cataloga con un valor de *subclase* y *clase*. Un *responsable* y un *usuario* son también relaciones obligatorias y unitarias para cada *espacio*. Los *usuarios*, a su vez, se agrupan por *responsables*, siendo que cada *usuario* tiene un único *responsable* asociado. Los *responsables* pueden tener varios *usuarios* asociados. Un espacio puede tener uno o más departamentos asociados, lo cual solo ocurre cuando el responsable del espacio es una unidad orgánica. Cada piso tiene un centroide asociado, cuyas coordenadas sirven como elemento auxiliar en la representación de las etiquetas de los pisos.

Una vez diseñada la base de datos se crearon las tablas necesarias para su implementación. Estas se pueden consultar en su totalidad en el Anexo B.

### 3.3 ARQUITECTURA DE LA SOLUCIÓN

Para la implementación de la aplicación *WebSIG* se utilizaron diferentes paquetes de *software* libre y *open source*. La Figura 3.9 muestra un esquema de la arquitectura de los principales componentes de la solución informática.

Los datos espaciales, junto con los alfanuméricos, se almacenan en un sistema de gestión de bases de datos espaciales (SGBDE) implementado mediante PostgreSQL 9.0 y su extensión espacial PostGIS 1.5.

El navegador *Web* del usuario podrá ser cualquiera entre los más populares –Firefox, Google Chrome, Internet Explorer, Opera y Safari–, siempre que se encuentre actualizado en sus últimas versiones. El navegador solicita y recibe información del servidor –MapFish, Pylons– mediante el protocolo HTTP.

Pylons se encuentra bajo la extensión MapFish, en su versión 2.2, la cual ofrece junto con la gran productividad de Pylons, la integración de un conjunto de herramientas orientadas a la creación de servicios *Web* de datos espaciales. El servidor está por tanto compuesto por varias herramientas extensibles y escritas en Python. Entre ellas cabe destacar SQLAlchemy y GeoAlchemy, ya que son las responsables del mapeado de la información y de la conexión, con la ayuda de la librería de Python Psicopg2, con la base de datos.

Volviendo al cliente, encontramos otro conjunto de herramientas que son ejecutadas en el navegador *Web*. Se trata de varios API JavaScript que permiten, desde la

conexión con servicios de datos espaciales externos (OpenStreetMap) o la gestión de la petición y posterior representación de la información alojada en la base de datos, hasta la creación de un interfaz *Web* interactiva y amigable con el usuario.

Estas herramientas son ExtJS, OpenLayers y GeoExt, las cuales son proporcionadas en su conjunto por el paquete MapFish *Client*, si bien, fueron actualizadas manualmente –por responder mejor a los objetivos del proyecto– a las versiones 3.4, 2.11 y 1.1 respectivamente.

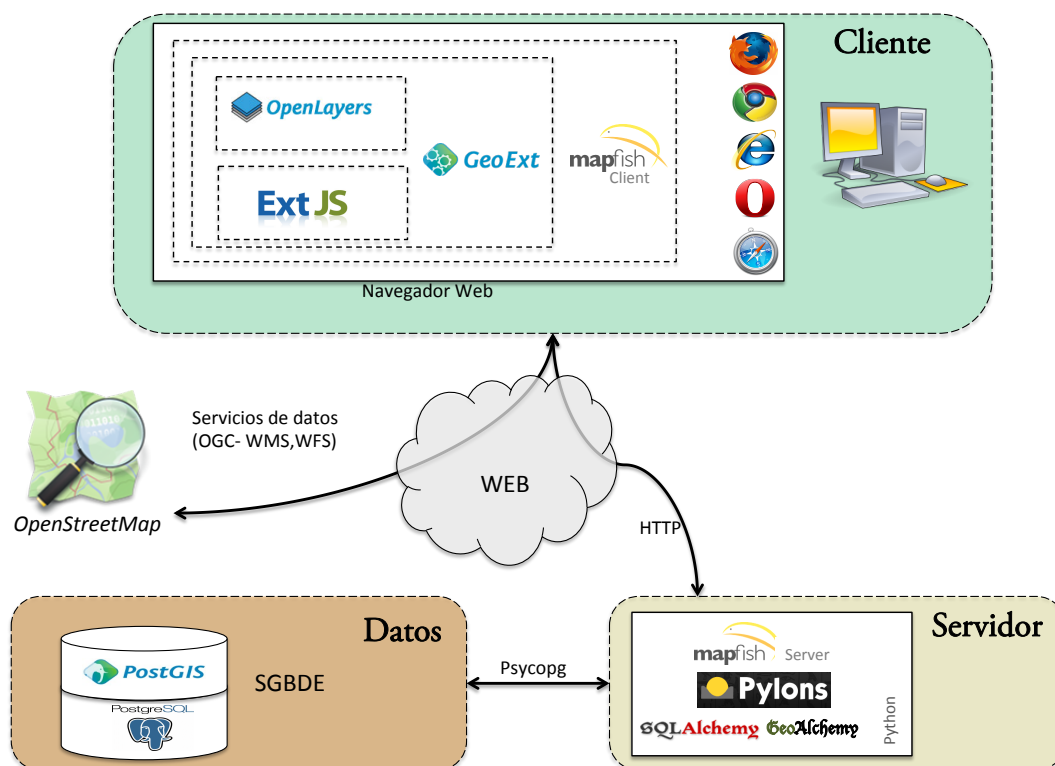
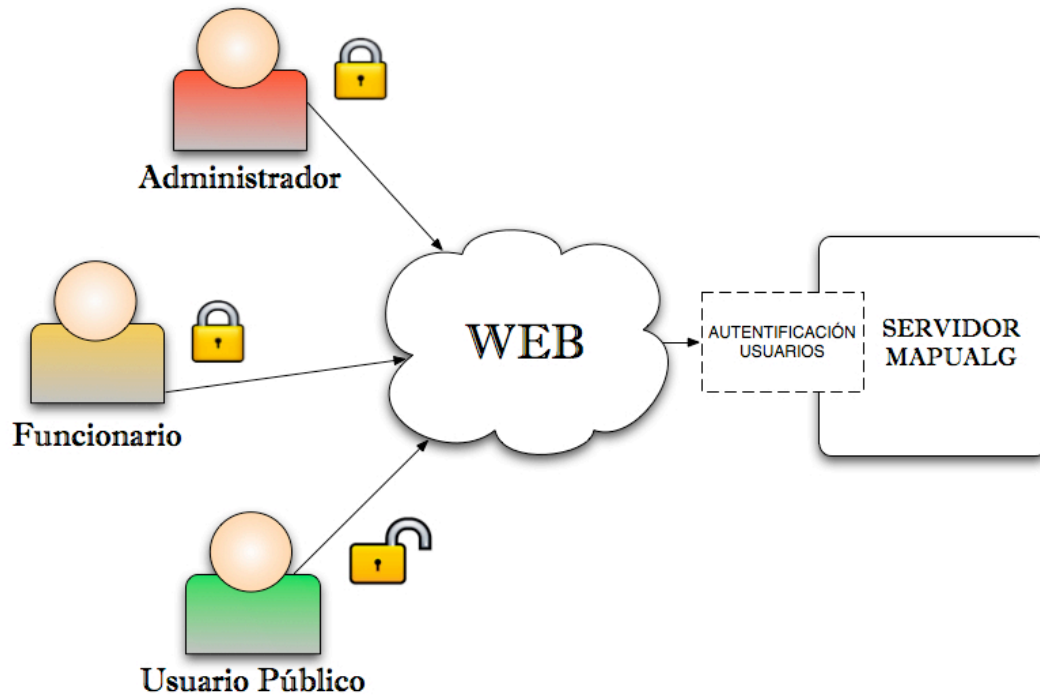


Figura 3.9 - Arquitectura de software de la solución informática

## 3.4 FUNCIONES DE LA APLICACIÓN

### 3.4.1 Funciones y usuarios

Las diferentes funciones de la aplicación *MapUAlg* presentan un acceso condicionado según un sistema de usuarios definido. Este control de acceso se realiza en el nivel de aplicación, mediante un sistema de autenticación de usuarios.



**Figura 3.10 - Usuarios y autenticación**

Tal y como se puede ver en la Figura 3.10, existen tres tipos de usuarios: usuario público, usuario institucional y administrador. El usuario público representa a cualquier persona con acceso a Internet y a un navegador *Web*. Este tendrá acceso a las funciones clasificadas como públicas, las cuales proporcionan información de interés, tanto para alumnos como para visitantes, invitados o futuros estudiantes. El usuario institucional representa a los trabajadores de la Universidad, como pueden ser profesores, investigadores o personal administrativo. La información a la que tendrán acceso será de gran utilidad para la gestión de los espacios. Por último, el administrador es un usuario designado para realizar el mantenimiento de la aplicación, así como para la actualización o corrección de los datos alfanuméricos asociados a los espacios.

Este sistema de usuarios presenta la siguiente estructura jerárquica: el usuario público, en último lugar, con acceso a funciones básicas; el usuario institucional, en medio, tiene acceso a las funciones básicas y a las funciones adicionales de gestión; y el administrador, en primer lugar, tiene acceso a todas las funciones de la aplicación.

En la Figura 3.11 se muestra un diagrama de contexto –propio del lenguaje UML (*Unified Modeling Language*)– en el cual quedan representadas las relaciones de las funciones de la aplicación con los grupos de usuarios definidos anteriormente.

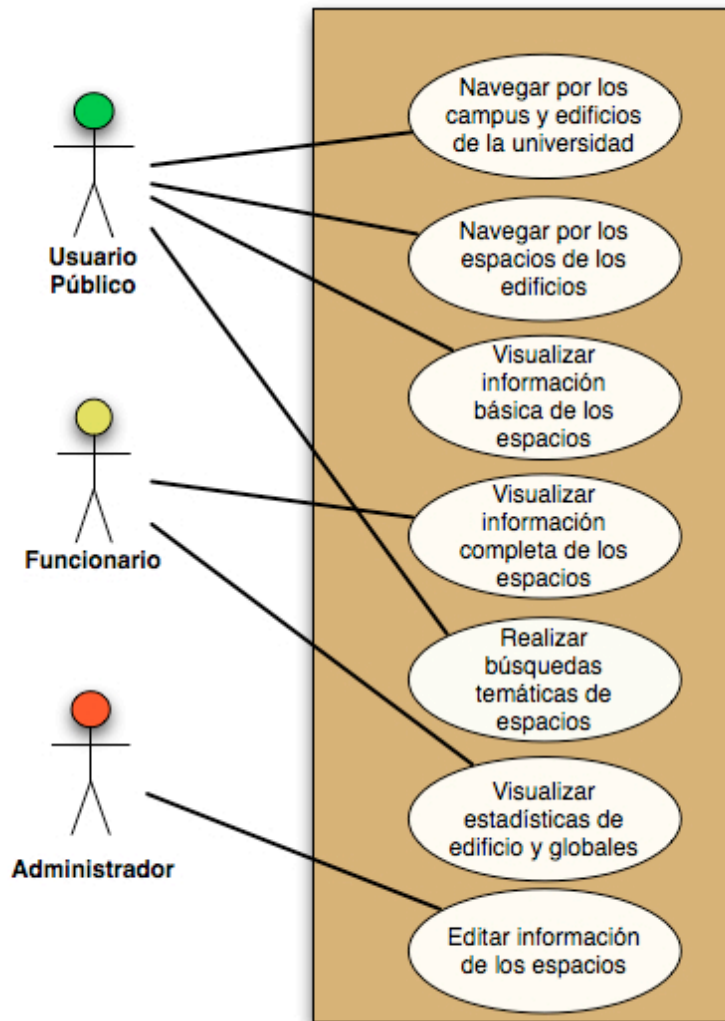


Figura 3.11 - Diagrama de contexto

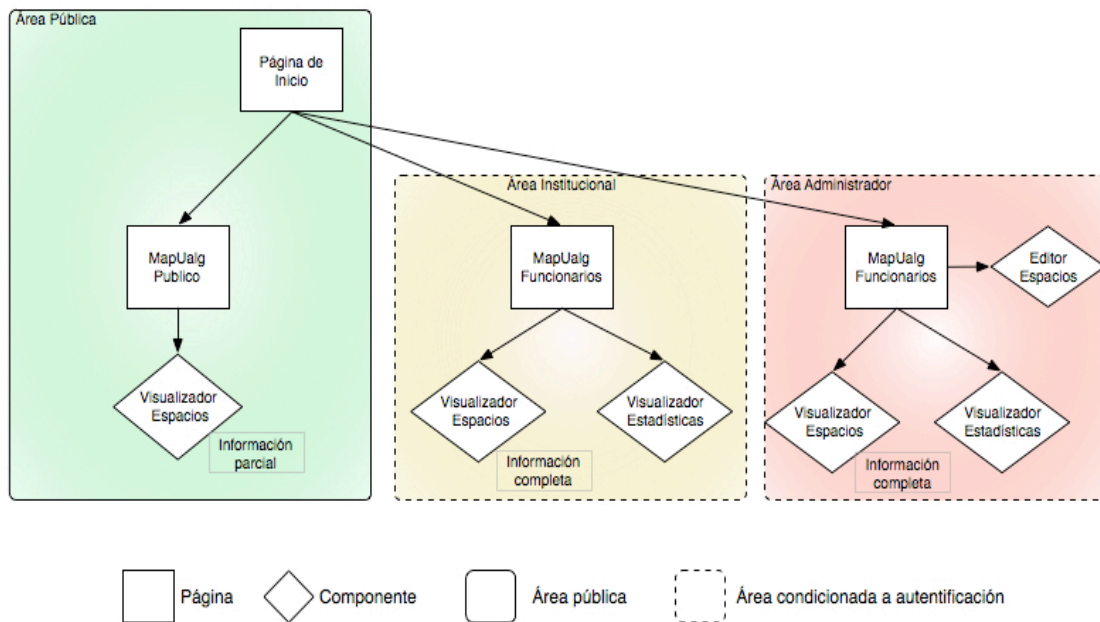
### 3.4.2 Interfaz de uso

Para una mejor comprensión de las funciones de la aplicación es necesario explicar primero la forma en la que un usuario interactúa con esta. Para ello se presentan, en este punto, dos elementos del diseño: un *sitemap* y un *wireframe*. A través de estas dos herramientas se puede entender, de una manera más visual, los procesos que se explicarán más adelante en las funciones de la aplicación.

#### 3.4.2.1 Sitemap (mapa Web)

Un *sitemap* –o mapa *Web*– es una herramienta de gran valor, tanto para el diseño como para el posterior uso de una aplicación o sitio *Web*, ya que organiza y

representa jerárquicamente sus páginas y componentes. En la Figura 3.12 se puede ver el *sitemap* de la aplicación *MapUAlg*. La página de inicio es, por definición, el punto de entrada para cualquier usuario de la aplicación. Esta página contiene la presentación del proyecto, los mapas de los campus y edificios de la Universidad, los formularios de autenticación de usuarios y los hipervínculos a la páginas de visualización de los espacios interiores de los edificios.



**Figura 3.12 - Sitemap (mapa Web)**

La página del área pública de la aplicación *MapUAlg* de visualización de espacios interiores permite a los usuarios, además de visualizar el interior de los edificios, acceder a parte de su información asociada. En la Figura 1.3 se puede ver una pantallazo de esta parte de la aplicación. La aplicación *MapUAlg* destinada a usuarios institucionales (ver Figura 3.2) permite la visualización de los espacios, su información completa y, además, la obtención de estadísticas. Si el usuario está autenticado como administrador podrá también al área de administración, donde tendrá permitida la edición de la información de los espacios (ver Figura 3.3).

### 3.4.2.2 Wireframe (esquema de página)

Un *wireframe* –o esquema de página– es una representación de la estructura visual de un sitio o aplicación *Web* que muestra la organización de los contenidos y de la interfaz con el usuario. En la Figura 3.13 se muestra el esquema genérico de la aplicación *MapUAlg*.

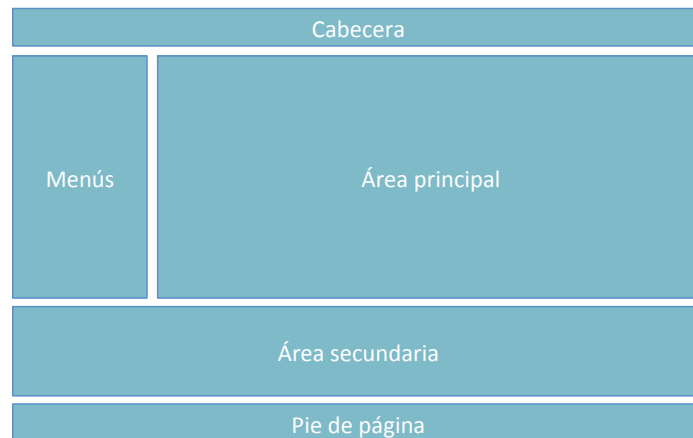


Figura 3.13 - *Wireframe* (esquema de página) general

En la cabecera, además del título identificativo de la aplicación y del nombre del edificio que se está visualizando, se encuentra un menú desplegable –*select*– que permite navegar entre todos edificios de la Universidad. La Figura 3.14 muestra en detalle este elemento de navegación.

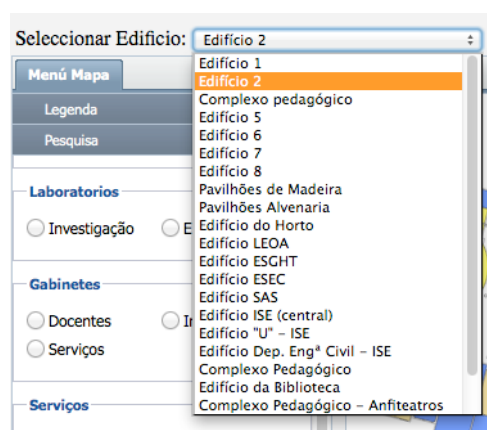
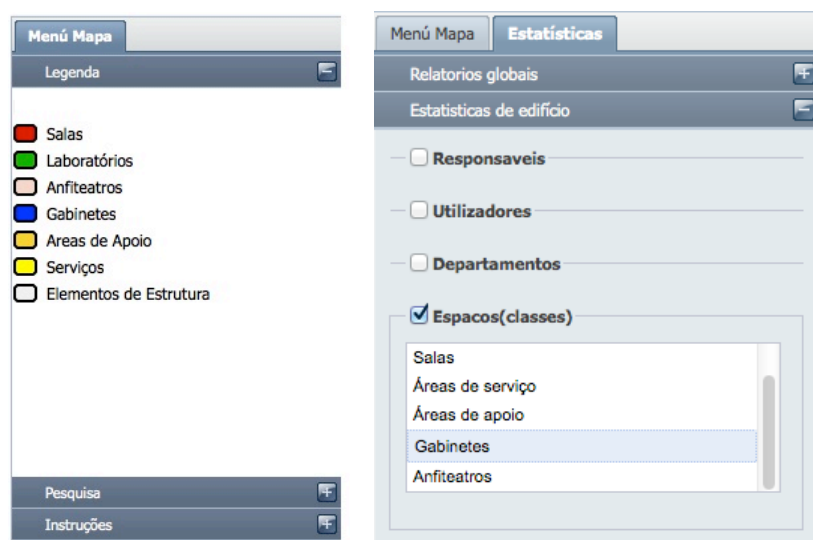


Figura 3.14 - Selector de edificios

La información útil para el uso de la aplicación y sus elementos de exploración se encuentran contenidos en el área del menú. Existen dos menús principales: el de

mapas y el de estadísticas. La información de como utilizar la aplicación, la leyenda de los mapas y el submenú de búsqueda de espacios, son los elementos principales del menú de mapas. En el menú estadísticas se encuentran las diferentes opciones para visualizar las estadísticas de uso del edificio. La Figura 3.15 muestra el detalle del menú mapa (izquierda) y el menú estadísticas (derecha).



**Figura 3.15 - Menú mapa (izquierda) y menú estadísticas (derecha)**

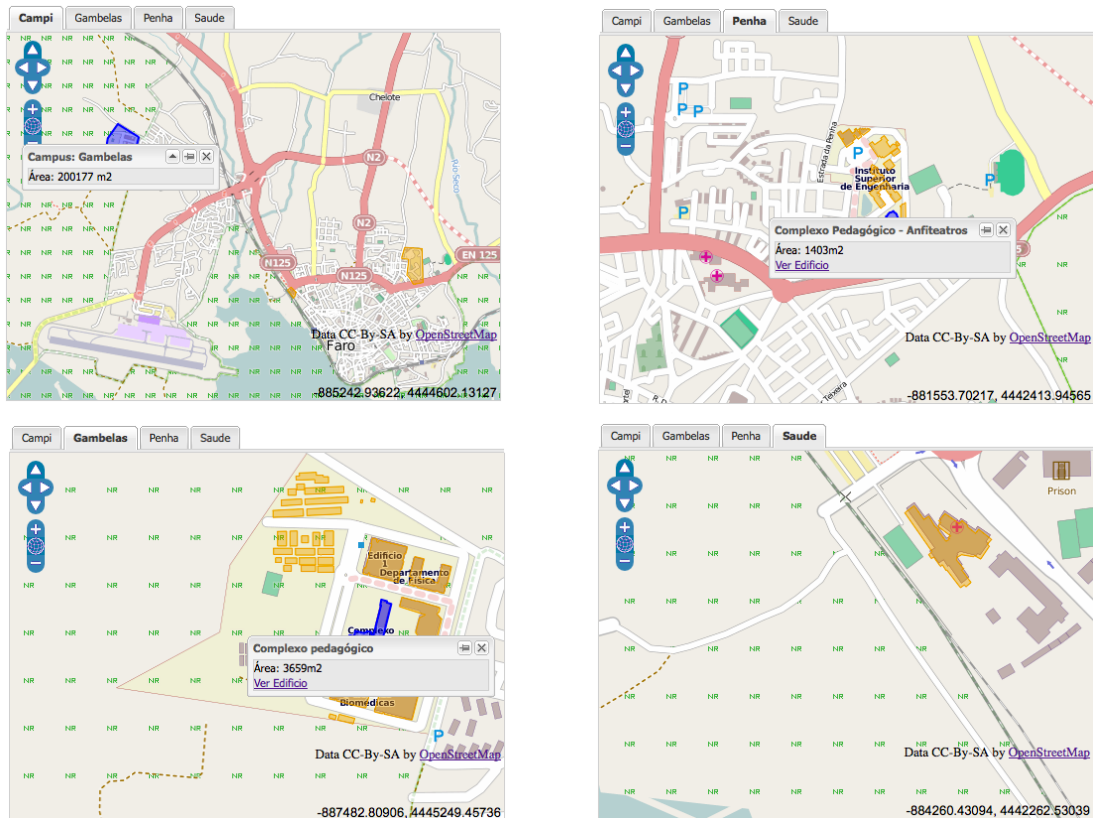
El área principal de la aplicación se destina a la representación de los mapas de los espacios de los edificios. El área secundaria se destina a la visualización de diversas informaciones, según lo requiera la aplicación. Cuando se navega por el mapa de los espacios, este área muestra la información asociada a los espacios seleccionados, mientras que cuando se solicitan las estadísticas del edificio, el área secundaria sirve para su visualización. El pie de página se destina a mostrar información de contacto y referencias a los organismos involucrados en la creación de la aplicación.

### **3.4.3 Funciones públicas**

#### ***3.4.3.1 Navegación por los campus y edificios***

Los usuarios públicos podrán acceder a la navegación por los campus y edificios de la Universidad, función que se encuentra disponible en la página principal a través de cuatro mapas seleccionables uno a uno como los que se pueden ver en la Figura 3.16.

El primer mapa permite visualizar el área de los tres campus y los tres siguientes contienen los edificios de cada campus: Gambelas, Penha y Saúde. Todos estos mapas tienen como capa base el mapa proporcionado por OpenStreetMap a través de un servicio WMS.



**Figura 3.16 - De arriba abajo e izquierda a derecha: Mapas de navegación por los campus y edificios de Gambelas, Penha y Saude**

Los usuarios pueden también seleccionar los elementos del mapa –campus o edificios–, lo que activa una pequeña ventana informativa –*popup*– donde se muestra la información básica del elemento –nombre y área– y un hipervínculo que, en el caso de los edificios, reencamina hacia la aplicación de visualización de los espacios de ese edificio.

Dado que el servicio WMS de OpenStreetMaps proporciona los mapas bajo la proyección Google Mercator, fue necesario incluir en la aplicación un proceso de transformación del sistema cartográfico de los edificios y campus – Datum 73, Hayford Gauss– al del servicio OSM –Google Mercator–, lo cual fue posible gracias a la librería de JavaScript Proj4js.

### 3.4.3.2 Navegación por los espacios de los edificios

En la página de la aplicación de visualización de espacios los usuarios pueden visualizar y navegar por los espacios de los pisos de cada uno de los edificios. Para poder visualizar simultáneamente todos los pisos de un mismo edificio se optó por aplicar un desplazamiento *–offset–* sucesivo en el eje *x* de cada piso, de forma que estos aparezcan representados uno al lado de otro. La Figura 3.17 muestra, en detalle, un ejemplo del mapa de navegación de espacios.



Figura 3.17 - Mapa de navegación por los pisos de un edificio

### 3.4.3.3 Visualización de la información parcial de los espacios

En estos mapas existe también un control que permite a los usuarios hacer clic encima de un espacio, quedando este seleccionado mediante el cambio de color y del grosor del contorno del polígono que lo representa. Este control solamente permite seleccionar un espacio de cada vez y cuando se realiza clic en otro espacio, este aparece como seleccionado, dejando de estarlo el anterior. Para deseleccionar un espacio basta con volver a clicar sobre este o sobre un punto cualquiera del mapa fuera del área de los espacios. El control de selección, además de resaltar visualmente el espacio, activa la visualización de la información alfanumérica básica asociada al espacio seleccionado, la cual, como se puede ver en la Figura 3.18, se compone de los siguientes campos: campus, edificio, piso, número de sala, área, uso, clase y responsable.

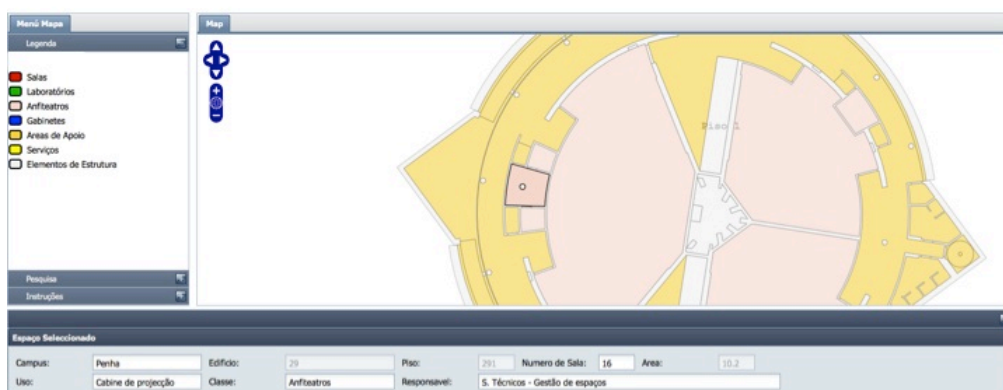


Figura 3.18 - Informação básica de un espacio seleccionado

### 3.4.3.4 Realización de búsquedas

La realización de búsquedas de espacios dentro de un mismo edificio, tanto por número de sala como por el uso al que se destina, es otra de las funciones accesibles a los usuarios públicos a través del menú de mapas. Para buscar un espacio con un número de sala determinado, el usuario selecciona el número de sala deseado en un menú desplegable *–select–*. De este modo la aplicación realiza un *zoom* sobre la sala pretendida, quedando esta, a su vez, seleccionada. Para buscar los espacios destinados a un uso en concreto, se escoge este entre los disponibles en el menú, y automáticamente, todos los espacios del edificio destinados a dicho uso aparecerán seleccionados. En la Figura 3.19 y la Figura 3.20 se muestra un ejemplo de búsqueda por uso y por número de sala respectivamente.

Ambos tipos de búsqueda se ejecutan en el navegador *Web*, lo que resulta posible gracias a los mecanismos disponibles en las bibliotecas *OpenLayers* y *ExtJS*, que permiten la pesquisa de entidades en el mapa a partir de sus atributos.

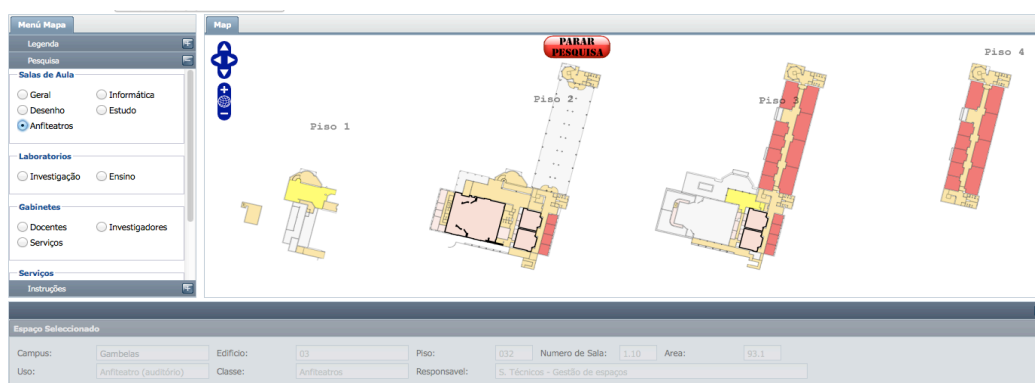


Figura 3.19 - Búsqueda temática por uso. Anfiteatros del complejo pedagógico de Gambelas

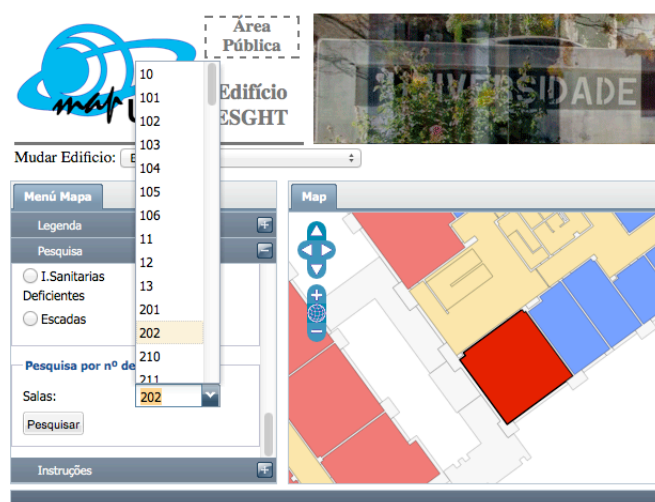


Figura 3.20 - Búsqueda por número de sala

### 3.4.4 Funciones para usuarios institucionales

#### 3.4.4.1 Visualización de la información completa de los espacios

Esta función muestra, al igual que la explicada anteriormente en la visualización de la información parcial de los espacios, la información asociada a un espacio cuando este es gráficamente seleccionado. En este caso la información mostrada se corresponde con la información completa del espacio, estando compuesta por los siguientes campos: campus, edificio, piso, número de sala, área, uso, clase, subclase, subsubclase, responsable, utilizador y departamentos. En el caso de los departamentos, por tratarse de un campo que solamente tiene valor cuando el espacio pertenece a una unidad orgánica, el espacio designado para su visualización se oculta cuando no existen departamentos. En la Figura 3.21 se puede ver un ejemplo de selección y visualización de la información asociada un espacio.

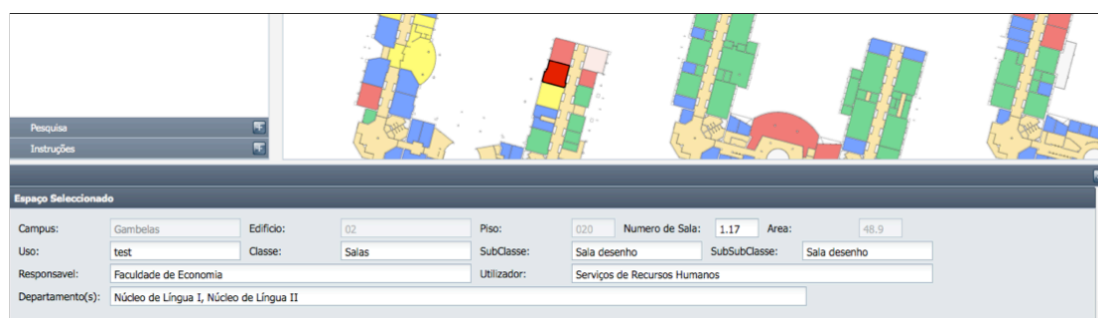


Figura 3.21 - Información completa de un espacio

### 3.4.4.2 Estadísticas

La obtención y visualización de estadísticas relacionadas con la utilización de los espacios de la Universidad es otra de las funciones a las que los usuarios institucionales pueden acceder. Estas estadísticas muestran y analizan los datos relativos al edificio que se está visualizando en el momento. Para acceder a ellas, el usuario clicca sobre la pestaña del menú *estadísticas* donde podrá visualizar las casillas para la selección *checkbox* de las diferentes estadísticas disponibles. Estos *checkbox* permiten escoger las estadísticas que se quieren obtener según cuatro de los parámetros con los que se clasifican los espacios: responsables, utilizadores, departamentos y clases. Cada parámetro muestra una lista con los valores adquiere en ese edificio, de tal manera que cuando alguno de ellos es seleccionado, se calculan las estadísticas de dicho parámetro.

En todas las estadísticas se muestran dos tipos de información: espacial y numérica. La información espacial se representa con la selección de todos los espacios implicados en la estadística solicitada – resaltándolos en el mapa del edificio–. Como se puede ver en el ejemplo de la Figura 3.22 (arriba), si se requieren los datos relacionados con los espacios de los cuales, por ejemplo, los servicios académicos son responsables, todos los espacios que cumplan con esa condición se mostraran seleccionados. El segundo tipo de información expuesta, la numérica, se visualiza en el área secundaria de información de la aplicación, donde se muestra el número total de espacios que cumplen la condición seleccionada, así como la suma de sus áreas. Se visualiza también una tabla que analiza esta misma información con un nivel superior de detalle, mostrándose, en el caso de los responsables, utilizadores y departamentos, cual es la división de estos espacios por clase; y en el caso de los espacios (clases), cual es la división por responsable.

En el primer ejemplo, reflejado en la parte superior de la Figura 3.22, se puede observar el caso de obtención de estadísticas de los espacios del edificio de la Escola Superior de Gestão Hotelaira e Turismo, de los cuales los servicios académicos son usuarios: de los 7 espacios y 224,2 m<sup>2</sup> que cumplen esta condición, uno corresponde a un depósito *arrecadação* con 50.3 m<sup>2</sup> y 6 a *serviços* con un total de 173.9 m<sup>2</sup>.

En el segundo ejemplo, mostrado en la parte inferior de la Figura 3.22, se solicitan las estadísticas para la clase despachos *gabinetes*, resultando en un total de 59 espacios

y 1019,8 m<sup>2</sup> , de los cuales 53 tienen como responsable a la Escola Superior de Gestão Hotelera y Turismo y seis a los Serviços Académicos.

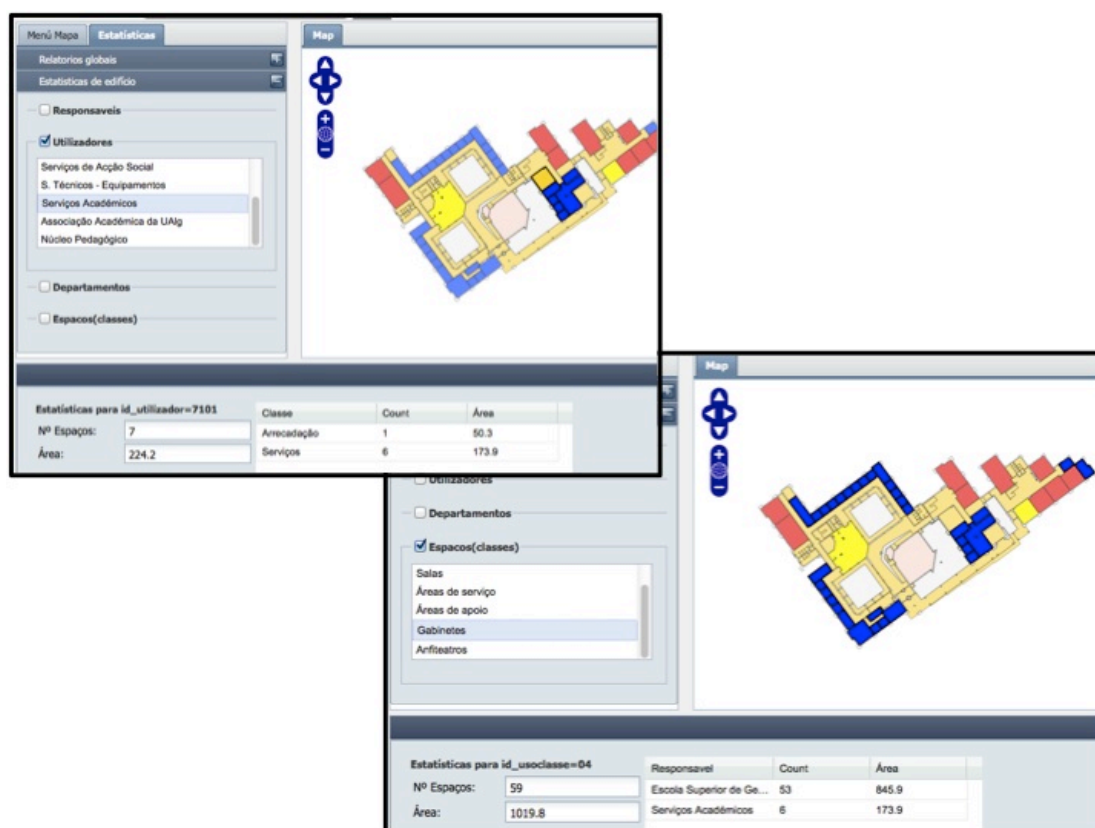


Figura 3.22 - Visualización de las estadísticas de edificio para los espacios según sus usuarios (arriba) y según sus clases (abajo)

### 3.4.5 Funciones de Administrador

#### 3.4.5.1 Edición de datos alfanuméricos

Esta es la función que permite a los usuarios administradores editar algunos de los datos alfanuméricos asociados a los espacios por medio del formulario situado en el área secundaria de la aplicación, donde se muestra la información de los espacios seleccionados.

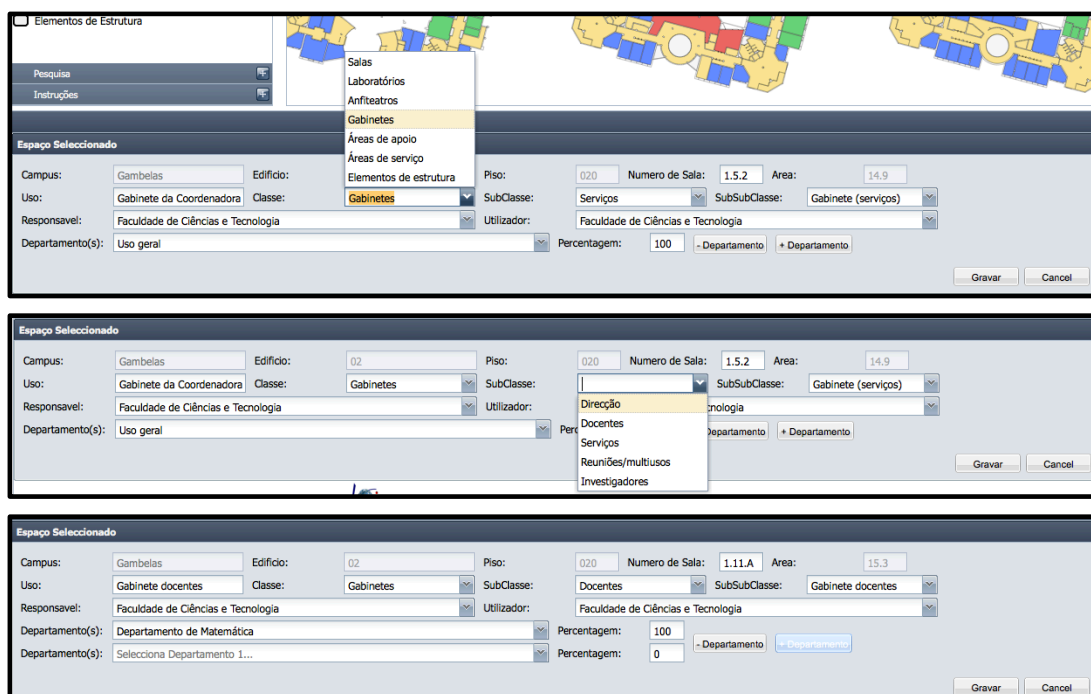
Cuando la funcionalidad está activada, los campos de visualización de la información no modificable (campus, edificio, piso y área) se bloquean –imposibilitando su modificación–, puesto que no será habitual modificar esta información, ya que un espacio no puede cambiar su localización o geometría. Los restantes campos pasarán

a ser editables mediante la inserción de texto o a través de menús seleccionables. Los campos editables mediante la inserción de texto –*textbox*– son el uso y el número de sala. Los campos con menús seleccionables –*select*– son aquellos que implican una clasificación de los espacios. En la Figura 3.23 (arriba) se puede ver un menú seleccionable desplegado. Clase, subclase, subsubclase, responsable, utilizador y departamentos son los campos que presentan esta característica.

Por existir una relación jerárquica entre algunos de estos campos, el sistema filtra dinámicamente las posibles selecciones, imposibilitando la realización de combinaciones incorrectas de selecciones.

En la Figura 3.23 (centro) se puede observar como, si está seleccionada la clase despachos (*gabinetes*), el menú desplegable de las subclases sólo muestra los cinco valores asociados a los despachos: *direcção*, *docentes*, *serviços*, *reuniões/multiusos* e *investigadores*.

En el caso de querer modificar un espacio perteneciente a una unidad orgánica, se muestran los departamentos asociados de forma individual, y se da también la posibilidad de añadir o retirar departamentos (Figura 3.23, abajo).



**Figura 3.23 - Edición de la información asociada a un espacio. Campos seleccionables (arriba). Sistema de protección contra clasificaciones incorrectas (centro). Posibilidad de añadir o quitar departamentos (abajo)**

Una vez completado el formulario con la información actualizada de un espacio, el botón *grabar* se encargará de enviar el formulario al servidor. La Figura 3.24 muestra una captura de los parámetros enviados al servidor. Este recoge la información enviada y construye y ejecuta una consulta *–query–* para la modificación de los campos solicitados en la base de datos.

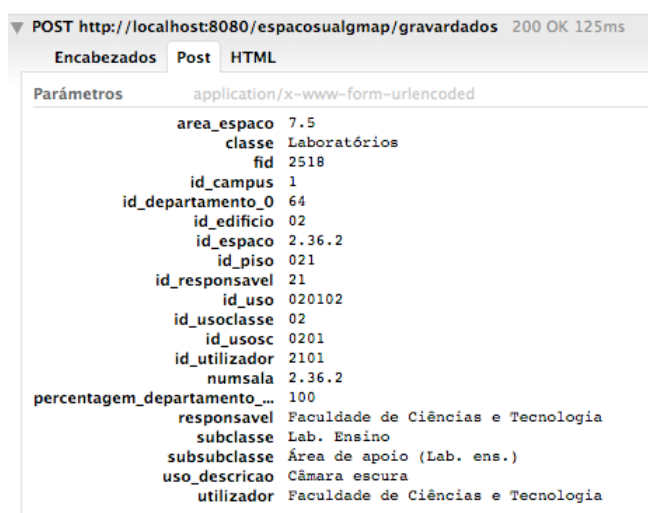


Figura 3.24- Parámetros enviados en POST para la edición de un espacio.

Para evitar recargar la página para visualizar la información actualizada después de modificar un espacio, las modificaciones se ejecutan también localmente en el cliente, cambiando tanto la información alfanumérica del espacio como el color de representación del mismo *–en el caso de que se haya cambiado la clase–*.

### 3.5 SERVIDOR

Un servidor de aplicaciones *Web* es un *software* ejecutado en un ordenador, denominado servidor, integrado en una red de computadores y que ejecuta aplicaciones proporcionando servicios de aplicación a los computadores cliente (ver sección 2.4). Generalmente, el servidor de aplicaciones gestiona la mayor parte de la lógica de negocio y el acceso a los datos.

El servidor de aplicaciones *Web* utilizado en este proyecto es, tal y como se vio en la sección 2.6.2.1, MapFish, una extensión de Pylons preparada para el manejo de datos espaciales, donde se encuentra la tecnología necesaria para poder acceder, procesar y publicar la información espacial y alfanumérica almacenada en la base de datos. El

computador, servidor, donde se aloja MapFish es el mismo donde se encuentra instalada la base de datos espaciales.

El funcionamiento del servidor está basado en la arquitectura Modelo-Vista-Controlador (ver sección 2.6.1). Los pedidos provenientes de los clientes son enviados a los controladores adecuados, los cuales son clases Python, y cuyos métodos se denominan acciones, siendo estas las que manejan la lógica de aplicación. El controlador comunica con el modelo para obtener la información necesaria de la base de datos, procesarla y enviarla a los *templates –View–*, quienes se encargan de generar el HTML necesario para la visualización de las páginas. Las peticiones de los clientes al servidor pueden ser de tipo síncronas o asíncronas y todas ellas se basan en el protocolo HTTP. Mediante las peticiones síncronas, el cliente solicita al servidor el envío de información –normalmente una página o aplicación *Web–* existiendo también la posibilidad de enviar parámetros que el servidor pueda necesitar para la generación de los contenidos solicitados. Las peticiones asíncronas, basadas en la tecnología AJAX (*Asynchronous JavaScript and XML*), permiten a los clientes obtener información, contenidos o actualizaciones de datos sin la necesidad de cargar de nuevo la página *Web*. Este tipo de peticiones son utilizadas en la aplicación para cargar los mapas, encontrar los departamentos de un espacio seleccionado, calcular las estadísticas de edificio y editar la información de los espacios.

La Tabla 3.2 describe las acciones –métodos en las clases de Python– de cada uno de los controladores implementados en el servidor, indicando también, cuales son los parámetros que estas esperan recibir del cliente.

**Tabla 3.2- Descripción de los controladores y sus acciones**

<u>Controlador</u>	<u>Acción</u>	<u>Parámetros</u>	<u>Descripción</u>
<b>Espacosualgmap</b>	index	-	Carga la página de inicio
	mapespacos	- Edificio	Carga página de aplicación de visualización de espacios según el usuario autenticado
	departamentos	- Piso - Id espacio	Devuelve lista con los departamentos asociados a un espacio

	gravar_dados	-Información de un espacio (ver Figura 3.24)	Graba en la base de datos la información modificada de un espacio.
	no_edificio_geral	- Id de campo seleccionado (reponsable o utilizador ou usoclase o departamento) - Edificio	Devuelve el número de espacios y la suma de sus áreas de los espacios del edificio que cumplen la condición seleccionada.
<b>Estatísticas</b>	no_edificio_detalhe	- Id de campo seleccionado (reponsable o utilizador o usoclase o departamento) - Edificio	Devuelve una lista con el número de espacios y la suma de sus áreas de los espacios del edificio que cumplen la característica seleccionada organizados por responsable o clase.
	getuniquevaluesdepartamentos	-Edificio	Devuelve una lista con los departamentos existentes en un edificio.
	getespacosdepartamento	-Edificio -Departamento	Devuelve una lista con los espacios asociados a un departamento en un edificio
<b>Espacosualg</b>	index	-Edificio	Devuelve GeoJSON con la geometría y atributos del edificio solicitado
<b>Edificiosualg</b>	index	-Campus	Devuelve GeoJSON con la geometría y atributos de los edificios del campus solicitado.
<b>Campiualg</b>	index	-Edificio	Devuelve GeoJSON con geometría y atributos de los campus de la Universidad.
<b>Pisocentroide</b>	index	-Edificio	Devuelve GeoJSON con la geometría y atributos de los centroides de los pisos del edificio solicitado.

### 3.6 ACTUALIZACIÓN DE LOS DATOS ESPACIALES

Resulta importante narrar el plan previsto para la actualización de los datos espaciales de la Universidad, ya que estos, aunque se prevé que sea de manera poco frecuente, pueden sufrir alteraciones. Estas pueden resultar de la necesidad de modificar las geometrías debido a la realización de obras de mejora o a la reorganización física de los espacios. También se puede dar el caso de alteración por la adición de espacios de nueva construcción o por la substracción de espacios que sean eliminados de las dependencias de la Universidad.

Si bien la aplicación *MapUAlg* permite la edición de los datos alfanuméricos asociados a los espacios, la edición de las geometrías no es posible. Resaltar, no obstante, que las herramientas utilizadas para la implementación de la aplicación permiten perfectamente la creación de funciones que satisfagan estas necesidades de actualización, pero que, debido a la naturaleza esporádica de las actualizaciones de geometría, se consideró que no era necesario su desarrollo.

Para cubrir esta necesidad de actualización se optó por utilizar la aplicación QuantumGIS. Se trata de un *software SIG desktop* que permite la conexión directa con una base de datos PostgreSQL/PostGIS para la visualización y edición de la información geográfica. La Figura 3.25 muestra un esquema de conexión en el que la aplicación QuantumGIS se conecta directamente al sistema de base de datos PostgreSQL/PostGIS.

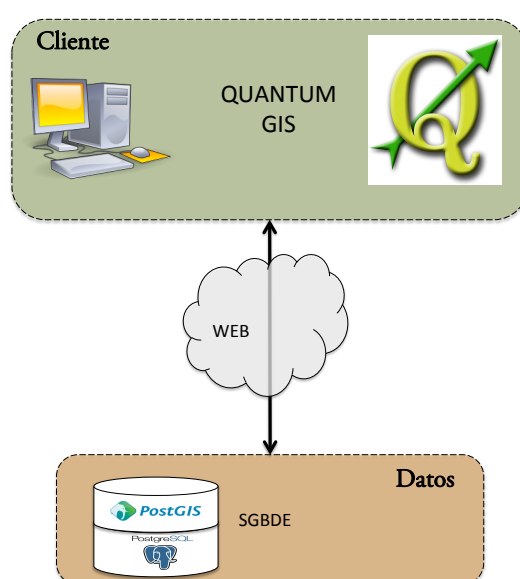


Figura 3.25 - Esquema de conexión directa de QGIS con la Base de Datos Espaciales

Estas tareas serán realizadas, cuando sean requeridas, por un técnico especializado en el uso de este *software* y familiarizado con la aplicación *MapUAlg*.

## 4 VISUALIZACIÓN TRIDIMENSIONAL

Desde la aparición de la computación gráfica una de las principales preocupaciones y líneas de desarrollo ha sido cómo representar objetos tridimensionales en una pantalla, pues estas son superficies bidimensionales (I.Carlbom; J.Paciorek, 1978). Hoy en día y con la aparición de la *Web* y con el incremento de las velocidades de conexión a Internet así como de las capacidades gráficas de los ordenadores, aparecen, cada vez más, nuevas herramientas para la publicación de información tridimensional en la *Web*, siendo algunas de estas:

X3DOM<sup>25</sup>, un *framework* experimental y *open source* para la publicación de escenarios 3D en la *Web* y que permite la manipulación del contenido tridimensional; y WebGL<sup>26</sup> (*Web Graphics Library*), un API JavaScript que proporciona una fácil integración de gráficos tridimensionales interactivos en la *Web*.

Normalmente, cualquier tecnología que pretenda representar información tridimensional necesita que esta esté estructurada en modelos de datos espaciales diseñados para soportar las tres dimensiones. Dos de las estructuras más utilizadas para datos tridimensionales son: X3D<sup>27</sup>, basado en el estándar XML; y CityGML<sup>28</sup>, un modelo orientado a la representación de objetos urbanos tridimensionales y cuya versión 2.0 ya está adoptada por la OGC.

La peculiaridad del sistema de visualización aquí presentado reside en su capacidad de representar información espacial bidimensional –las plantas de los edificios– bajo una perspectiva tridimensional, utilizando una estructura de datos geográficos bidimensional –GeoJSON– y una plataforma de visualización que únicamente tolera información plana –OpenLayers–.

Para alcanzar este objetivo se optó por proyectar las plantas de los pisos de los edificios obteniendo así una visión en perspectiva de los mismos (ver Figura 4.1). A

---

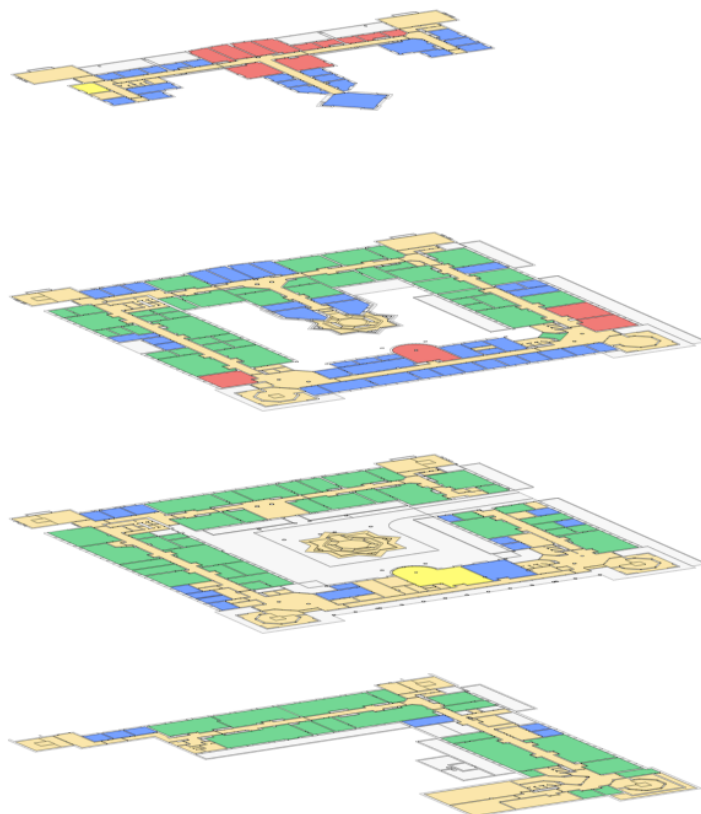
<sup>25</sup> <http://www.x3dom.org/>

<sup>26</sup> <http://www.khronos.org/webgl/>

<sup>27</sup> <http://www.web3d.org/x3d/>

<sup>28</sup> <http://www.citygml.org/>

continuación se explica el modelo de proyección empleado para la transformación geométrica de los edificios así como la implementación de esta proyección en la aplicación MapUAlg.



**Figura 4.1 - Proyección de los pisos de un edificio.**

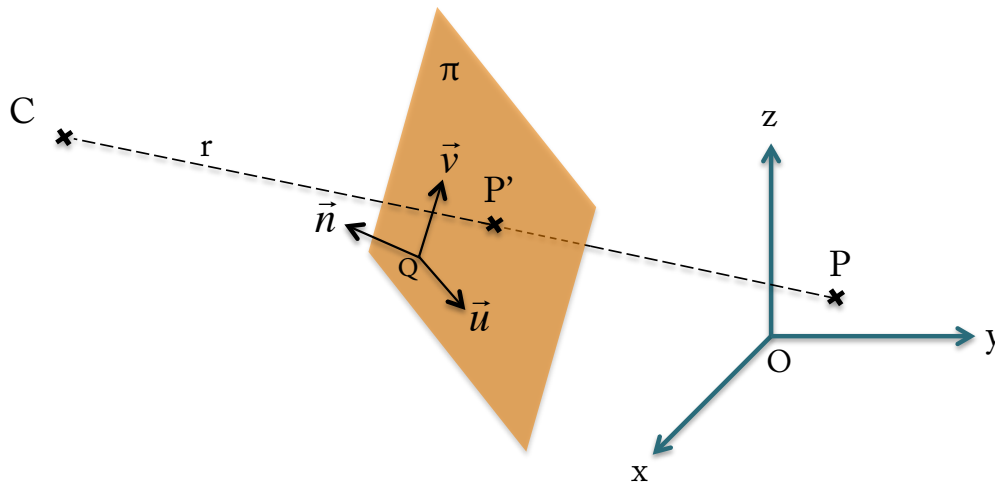
## **4.1 MODELO DE PROYECCIÓN**

De manera a poder visualizar las plantas de los edificios bajo una perspectiva tridimensional, inducida a partir de información bidimensional, se implementó un procedimiento para realizar una proyección geométrica de los pisos. Una proyección geométrica es un sistema de representación que permite visualizar en un plano un objeto situado en el espacio. La proyección utilizada en este proyecto es una proyección geométrica plana, la cual se obtiene con el trazado de líneas, llamadas proyectantes, por cada punto del objeto y encontrando así la imagen formada por la intersección de estas líneas con un determinado plano de proyección. Concretamente, se trata de una proyección paralela ortogonal, pues las proyectantes son paralelas entre si y perpendiculares al plano de proyección. Previamente al proceso de

proyección de los espacios, fue necesario asignar a cada piso de los edificios una determinada altura, tal y como se explica en la sección 4.2.1. A continuación se describe con detalle el modelo matemático empleado en el desarrollo del método de proyección.

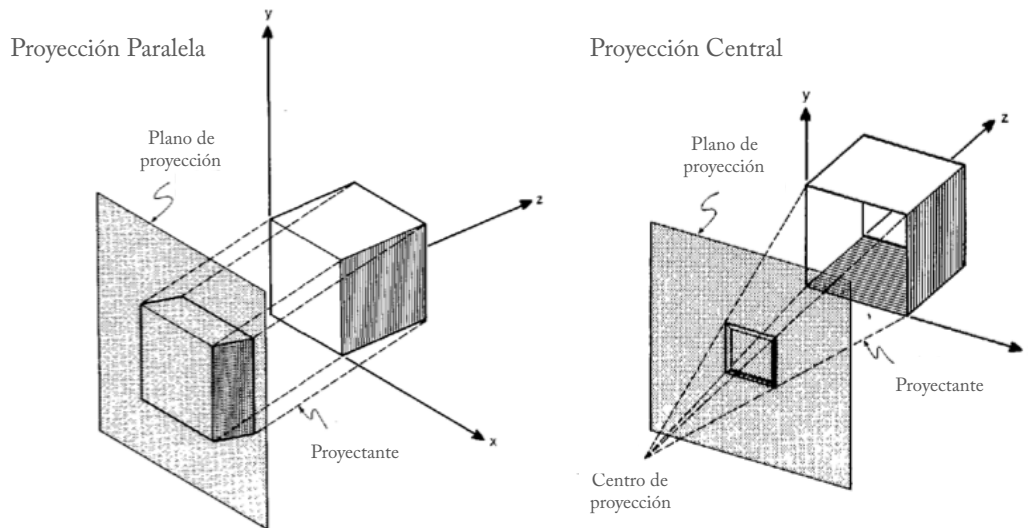
Se define el plano de proyección como un plano  $\pi$  no paralelo a ninguno de los planos coordenados ( $XOY$ ,  $XOZ$  y  $YOZ$ ) cuya ecuación general es:

$$\pi \equiv Ax + By + Cz - D = 0 \quad (1)$$



**Figura 4.2 - Diagrama de una proyección plana general.**

Tal y como se muestra en la Figura 4.2, si se considera un punto  $P$ , su proyección sobre el plano  $\pi$  corresponde con el punto  $P'$ , el cual resulta de la intersección de una recta,  $r$ , con el plano  $\pi$ . Si para cada punto  $P$ , la recta  $r$  toma siempre la misma dirección, la proyección es paralela, y si esta a su vez es perpendicular al plano  $\pi$ , la proyección es ortogonal. Sin embargo, si cada recta  $r$  contiene al punto  $C$ , la proyección es central. La Figura 4.3 muestra un ejemplo de proyección paralela (izquierda) y de proyección central (derecha). Se adoptó para este proyecto una proyección paralela y ortogonal.



**Figura 4.3 - Proyección paralela (izquierda) y central (derecha). Imágenes adaptadas de I. Carlbom, J. Paciorek (1978).**

Considerando la ecuación vectorial de la recta:

$$P' = P + \alpha \vec{\omega} \quad (2)$$

donde  $\vec{\omega}$  designa un vector del espacio vectorial real  $\mathfrak{R}^3$ .

$P'$  será el punto de intersección de la recta  $r$  con el plano  $\pi$  si  $P'$  es la solución del sistema de ecuaciones:

$$\begin{cases} P' = P + \alpha \vec{\omega} \\ Ax' + By' + Cz' - D = 0 \end{cases} \quad (3)$$

Considerando  $\omega = (\omega_x, \omega_y, \omega_z)$ ,  $P = (x, y, z)$  y  $P' = (x', y', z')$  el sistema (3) puede ser escrito de la siguiente forma:

$$\begin{cases} x' = x + \alpha \omega_x \\ y' = y + \alpha \omega_y \\ z' = z + \alpha \omega_z \\ Ax' + By' + Cz' = D \end{cases} \quad (4)$$

Ya que se pretende resolver en relación a  $P' = (x', y', z')$  y a  $\alpha$ , en notación matricial, el sistema (4) puede tomar la forma:

$$\begin{bmatrix} 1 & 0 & 0 & -\omega_x \\ 0 & 1 & 0 & -\omega_y \\ 0 & 0 & 1 & -\omega_z \\ A & B & C & 0 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ \alpha \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ D \end{bmatrix} \quad (5)$$

Resolviendo el sistema se obtiene la siguiente expresión para el valor de  $\alpha$ :

$$\alpha = \frac{D - Ax - By - Cz}{A\omega_x + B\omega_y + C\omega_z} \quad (6)$$

Dado que el vector  $\vec{N} = (A, B, C)$  es normal al plano  $\pi$ , la Ec. 6. es equivalente a la siguiente ecuación vectorial:

$$\alpha = \frac{D - \vec{N} \cdot \vec{OP}}{\vec{N} \cdot \vec{\omega}} \quad (7)$$

que facilitará la implementación de la aplicación informática.

Una vez calculado el escalar real  $\alpha$ , las coordenadas de la proyección de P sobre el plano  $\pi$ , P', se obtiene a partir de la Ec. 2.

Según la elección del vector  $\vec{\omega}$  tendremos dos tipos de proyecciones: una proyección central si  $\vec{\omega} = \vec{PC}$ , siendo C el centro de proyección; y una proyección paralela si  $\vec{\omega} \equiv \text{constante}$ , la cual será ortogonal en el caso de  $\vec{\omega}$  ser perpendicular al plano  $\pi$  y oblicua en los restantes casos. Este vector define la dirección y el sentido de observación del punto P y por tanto de su proyección en el plano  $\pi$ .

Para poder visualizar la proyección de los pisos es necesario definir un referencial cartesiano  $(Q, \vec{u}, \vec{v})$  sobre el plano  $\pi$  y calcular las coordenadas de cada punto P' en este referencial.

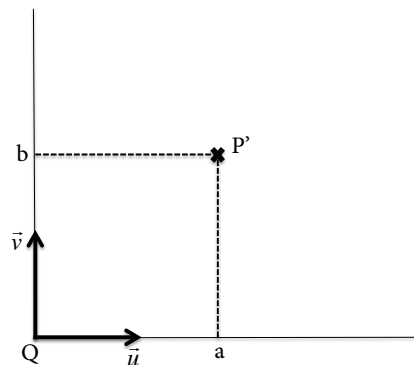


Figura 4.4 - Punto P' sobre el nuevo referencial de coordenadas u y v en el plano  $\pi$

Quedando el punto proyectado P' definido como:

$$\overline{QP'} = a\vec{u} + b\vec{v} \quad (8)$$

Y sus componentes como:

$$a = \overline{QP'} | \vec{u} \quad \text{y} \quad b = \overline{QP'} | \vec{v} \quad (9)$$

Por definición, las coordenadas (a, b) de un P' en este referencial coinciden con los coeficientes escalares de combinación lineal, donde  $\vec{u} \perp \vec{v}$  y  $|\vec{u}| = |\vec{v}| = 1$ .

A continuación se abordara la implementación en la aplicación de una proyección paralela ortogonal.

## 4.2 IMPLEMENTACIÓN DE UNA PROYECCIÓN PARALELA ORTOGONAL

El vector  $\vec{\omega}$ , el cual consideraremos unitario, es definido a partir del azimut, Az, de su proyección en el plano cartográfico XOY y de un ángulo vertical, V, con este plano. Estos parámetros pueden ser fácilmente definidos por el utilizador y representan la dirección y altura de “observación” de los edificios. Con estas condiciones, las componentes del vector  $\vec{\omega}$  quedaran definidas como:

$$\omega_x = \cos(V) \text{sen}(Az) \quad (10)$$

$$\omega_y = \cos(V) \cos(Az) \quad (11)$$

$$\omega_z = \text{sen}(V) \quad (12)$$

En este caso se optó por una proyección paralela y ortogonal, lo que equivale a considerar la normal al plano  $\vec{n} \equiv \vec{\omega}$ , siendo  $\vec{n} = \frac{\vec{N}}{\|\vec{\omega}\|}$  un vector unitario.

Se decidió posicionar al plano de proyección en el centro, M, del rectángulo mínimo – designado como *bounding box*– que contiene el área de implantación de los edificios,

de manera a crear un único plano de proyección común a todos los edificios de la Universidad. Este punto está definido como  $M = (M_x, M_y, M_z)$ . siendo  $M_x=16330$ ;  $M_y=292289$ ; y  $M_z = 0$ , ya que se sitúa en el plano horizontal. Otra posibilidad sería situar el punto M en el centroide de cada edificio, creando un plano diferente para cada edificio.

En la Figura 4.5 se pueden ver los 3 campus, el rectángulo *bounding box*, su centroide M, y la posición del plano de proyección  $\pi$  definido por el azimut de la proyección de la normal  $\vec{N}$  sobre el plano cartográfico,  $\vec{N}_{xy}$ . En la Figura 4.6 se observa también, en una vista de Alzado, la orientación vertical del plano  $\pi$ . Esta orientación está definida por el ángulo vertical de la normal  $\vec{N}$  con el plano cartográfico. Al posicionar el plano  $\pi$ , el azimut  $Az \in ]-\pi, \pi[$  y el ángulo vertical  $V \in ]0, \pi/2[$ .

Una vez definido el vector  $\vec{N}$  normal al plano  $\pi$  y el punto M, el cual pertenece a este plano, se puede calcular el valor de D a partir de la Ec. 1 como:

$$D = \vec{N} \cdot (M_x, M_y, M_z) = AM_x + BM_y + CM_z = AM_x + BM_y = \omega_x M_x + \omega_y M_y \quad (13)$$

Conocido el valor de D, se calcula  $\alpha$  por medio de la Ec. 7 como:

$$\alpha = \frac{D - \vec{N} \cdot \overline{OP}}{\vec{N} \cdot \vec{\omega}} = \frac{D - \vec{\omega} \cdot \overline{OP}}{\vec{\omega} \cdot \vec{\omega}} = D - \vec{\omega} \cdot \overline{OP} \quad (14)$$

ya que  $\vec{N} \equiv \vec{\omega}$ .

Una vez conocido el valor de  $\vec{\omega}$  y calculado el valor de  $\alpha$ , se puede obtener la proyección P' de un punto P sobre el plano  $\pi$  a través de la Ec. 2.

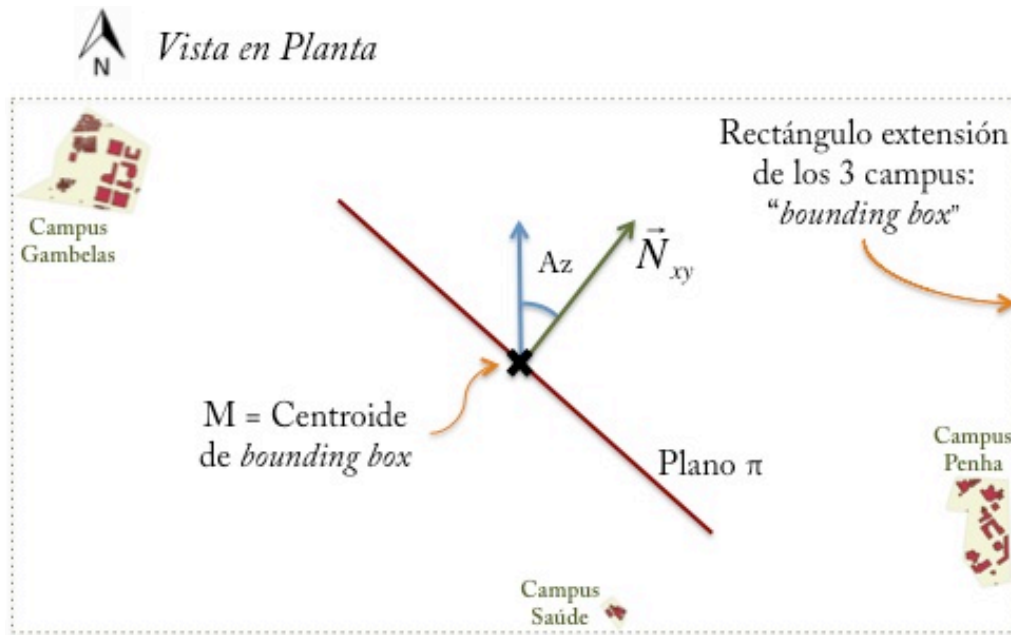


Figura 4.5 - Vista en planta de la situación del plano  $\pi$  respecto a los campus de la Universidad.

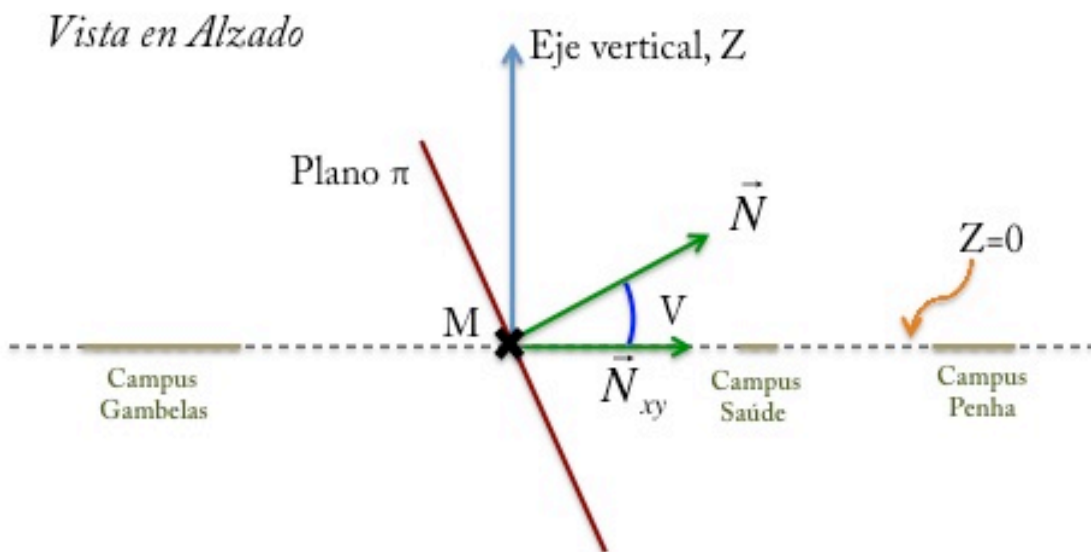


Figura 4.6 - Vista en Alzado de la situación del plano  $\pi$  respecto a los campus de la Universidad.

Con el punto P ya proyectado es necesario ahora realizar la transformación de sus coordenadas cartográficas  $P'=(X',Y',Z')$  para las coordenadas planas  $(u,v)$ .

Para la definición del referencial ortonormal  $(Q,\vec{u},\vec{v})$  en el plano  $\pi$ , se consideró  $Q=M$  y se definió el vector  $\vec{v}$  como colineal con la proyección del vector  $\vec{e}_z = (0,0,1)$  en el plano  $\pi$  e  $\vec{u}$  perpendicular a  $\vec{v}$ . Para ello, se proyectó en el plano  $\pi$  un punto S de coordenadas  $(M_x, M_y, 100)$ , es decir, S es un punto en la vertical del punto M,

obteniendo el punto proyectado  $S'$ . El vector unitario  $\vec{v}$  queda definido entonces como:

$$\vec{v} = \overline{QS'} / \|\overline{QS'}\|, \quad (15)$$

y el vector unitario  $\vec{u}$  como el producto vectorial de  $\vec{v}$  con la normal al plano  $\vec{n}$ :

$$\vec{u} = \vec{v} \times \vec{n} \quad (16)$$

Con los vectores unitarios  $\vec{u}$  y  $\vec{v}$  que definen el nuevo sistema de referencia plano calculados, se obtienen las nuevas coordenadas de los puntos proyectados en el plano por medio de la Ec. 9.

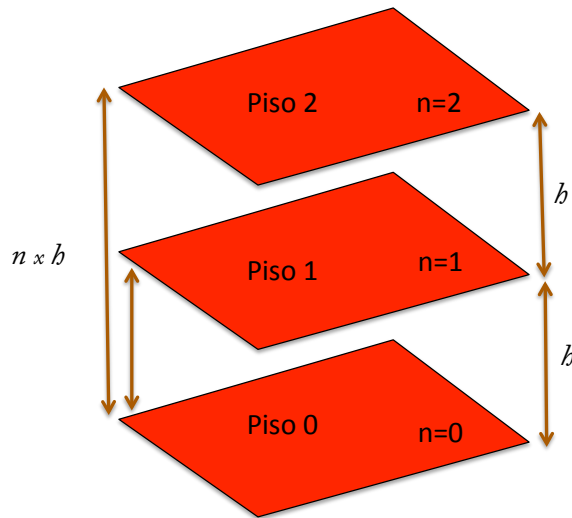
Se implementó entonces un algoritmo que transforma todos los polígonos que componen los pisos de un edificio, proyectando los puntos que definen cada polígono. Los resultados de este algoritmo se muestran a continuación y la integración de este algoritmo en la aplicación MapUAlg se explica en el apartado 4.3.

#### 4.2.1 Asignación de alturas, $h$ , a los pisos

Es necesario resaltar el método empleado para la asignación de alturas a los diferentes pisos de un edificio, pues este valor es el que evita la superposición de sus proyecciones. Tal y como se muestra en el esquema de la Figura 4.7, al piso inferior de cada edificio se le asigna el valor de altura 0, a los siguientes pisos se le aplica el valor

$$H_n = n \times h \quad (17)$$

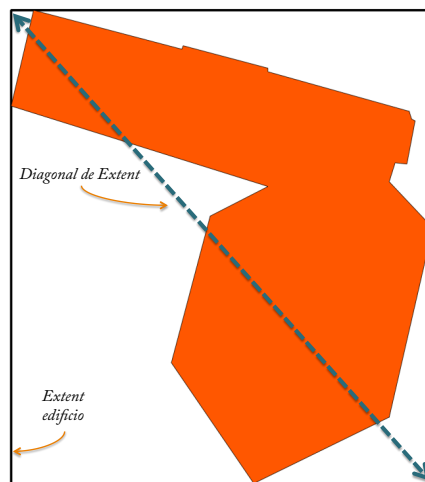
siendo  $n$  el número de piso (1, 2, 3, etc.) y  $h$  un valor de altura suficiente para evitar la superposición de las proyecciones de los pisos.



**Figura 4.7 - Esquema de asignación de alturas a los pisos de un edificio**

El cálculo del mínimo valor que debe tomar  $h$  para evitar la superposición de las proyecciones se efectuó como se describe a continuación.

Se definió la máxima diagonal de un edificio como la diagonal del rectángulo que conforma su extensión –o *bounding box*–, tal y como se muestra en la Figura 4.8.



**Figura 4.8 - *Bounding box* de un edificio y su diagonal**

La Figura 4.9 muestra una vista en alzado del esquema de proyección ortogonal sobre el plano  $\pi$  de un edificio de dos pisos. Este esquema muestra el peor de los casos, en el que la diagonal calculada anteriormente es perpendicular a la componente horizontal del plano de proyección. Se puede comprobar que para evitar la superposición entre las proyecciones de los pisos es necesario que la distancia entre sus proyecciones,  $d$ , sea mayor que 0, lo que se traduce en que la distancia  $H_0$  tendrá

que ser menor que la altura entre pisos,  $h$ . Se puede obtener el valor de  $H_0$  de la siguiente manera:

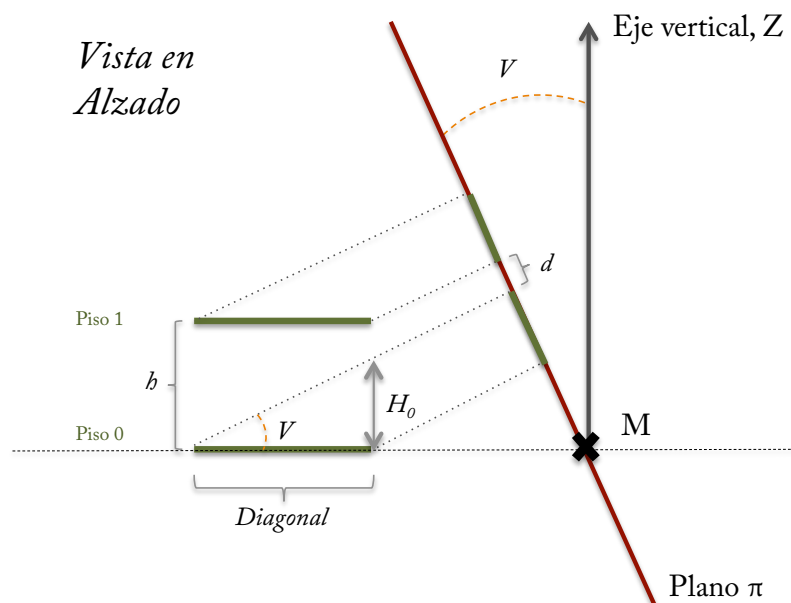
$$\tan(V) = \frac{H_0}{diagonal} \Rightarrow H_0 = diagonal \cdot \tan(V) \quad (18)$$

siendo  $V$  el ángulo vertical por medio del cual se definió el plano  $\pi$  y  $diagonal$  la medida de la diagonal de la bounding box.

Conocido el valor de  $H_0$ , se optó por asignar a  $h$  un valor un 10% mayor que  $H_0$ , garantizando así la no superposición entre pisos:

$$h = 1.1H_0 \quad (19)$$

Resaltar que el ángulo  $V$  de la Figura 4.9 tiene la misma amplitud que el ángulo  $V$  entre la normal al plano,  $\vec{N}$ , y el plano horizontal definido anteriormente.



**Figura 4.9 - Esquema para cálculo de altura,  $h$ , mínima para evitar superposición entre las proyecciones de los pisos de un edificio.**

## 4.2.2 Resultados de la proyección paralela ortogonal

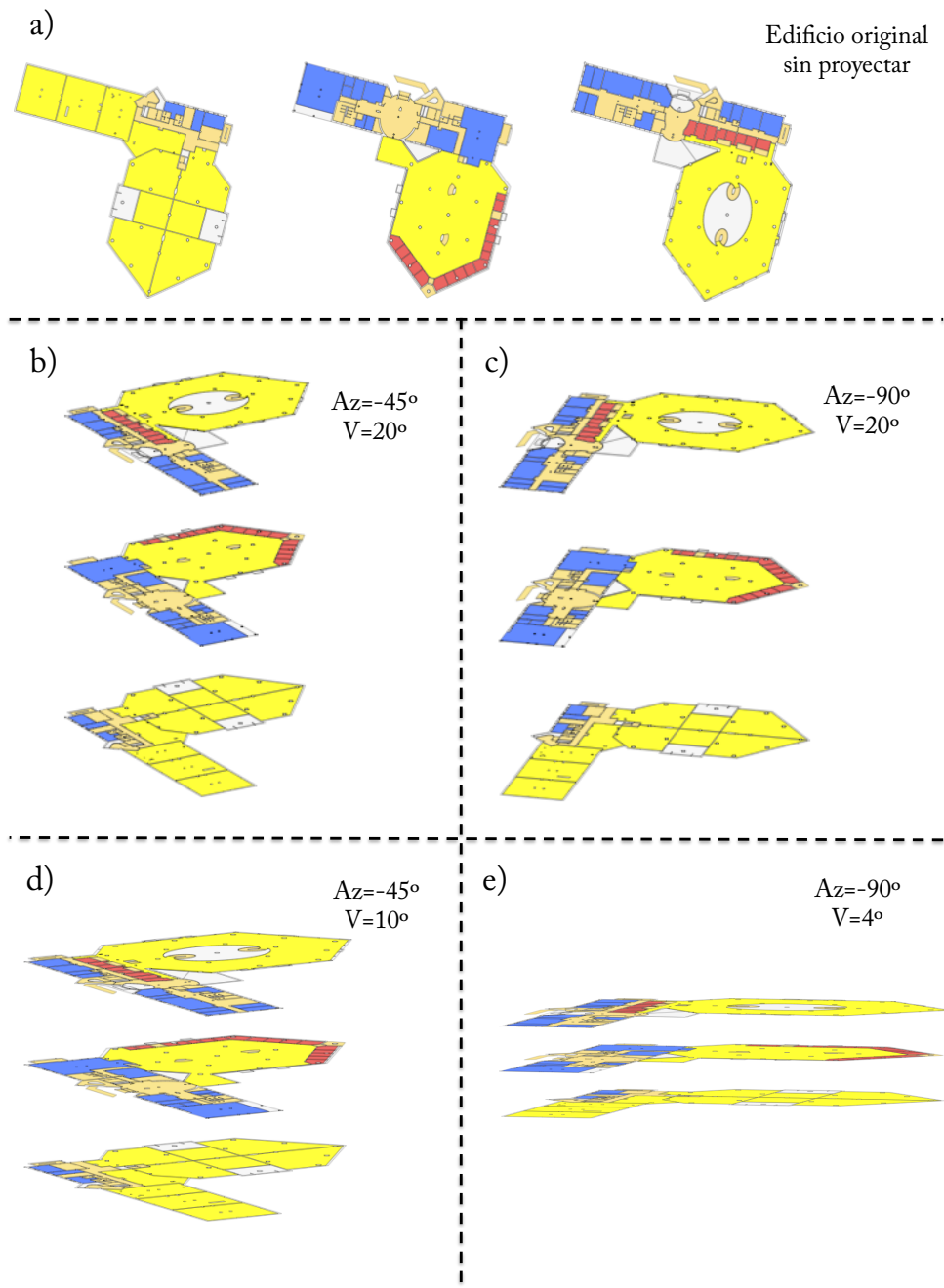


Figura 4.10 - Proyecciones paralelas ortogonales de un edificio utilizando diferentes planos de proyección definidos por los ángulos  $Az$  y  $V$ .

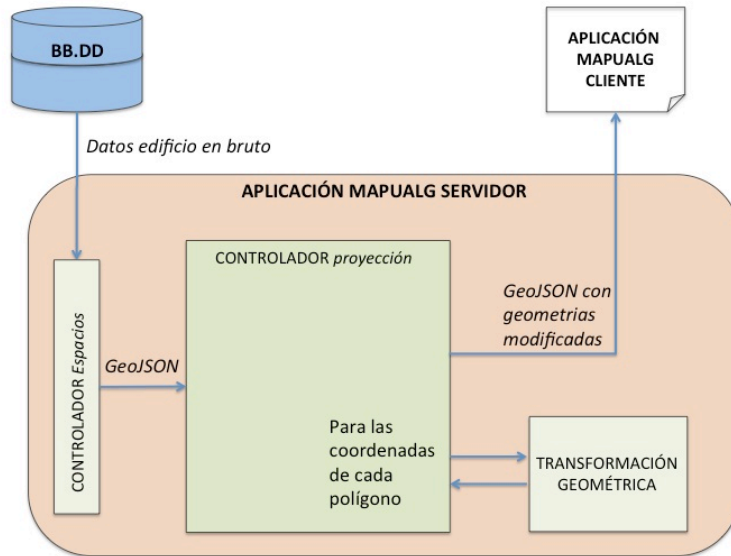
La Figura 4.10 muestra algunos de los resultados de la aplicación de la proyección paralela ortogonal, explicada anteriormente, en un edificio. Se puede comprobar como a través de los dos ángulos que definen el plano de proyección,  $Az$  y  $V$ , se controla la dirección –respecto al Norte– y la verticalidad del plano de proyección

respectivamente. Se comprueba también como en ningún caso se superponen los pisos. En la Figura 4.10 a) se muestra las plantas de los pisos de un edificio sin proyectar; en b) se representa el resultado de aplicar la proyección con  $Az = -45^\circ$  y  $V = 20^\circ$ ; c) muestra la misma transformación con un diferencia de  $45^\circ$  en la dirección de observación, o sea,  $Az = -90^\circ$ , pudiendo comprobarse una rotación en la representación del edificio; en d) se aumenta en  $10^\circ$  la verticalidad del plano de proyección de b); y por último, en e) se aplica un valor pequeño  $-4^\circ$  al ángulo vertical,  $V$ , de plano de proyección. Se puede comprobar como en ningún caso los pisos se superponen

### 4.3 IMPLEMENTACIÓN DE LA PROYECCIÓN EN LA APLICACIÓN MAPUALG

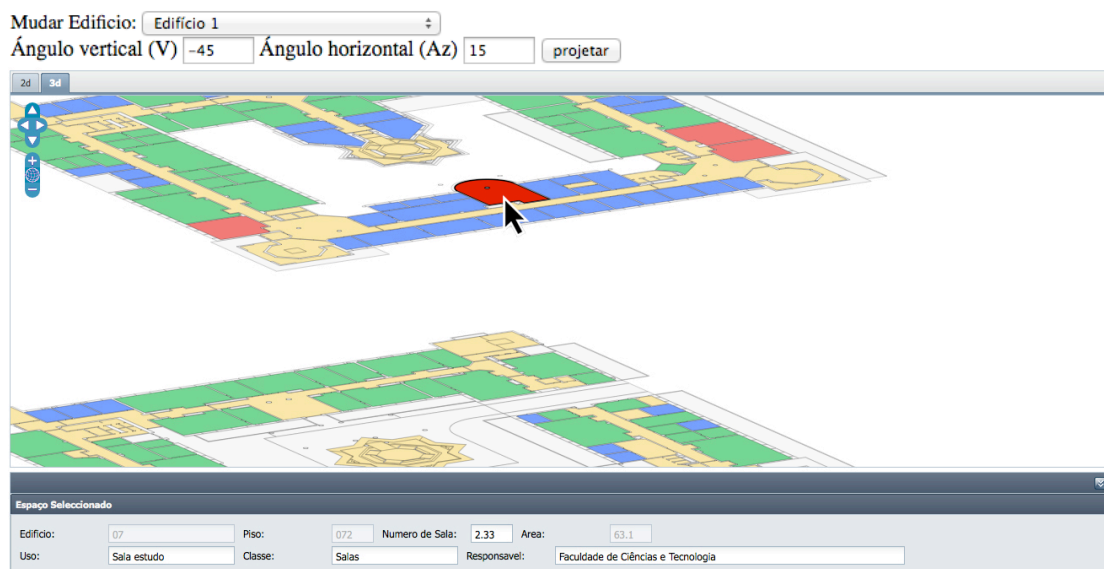
El controlador *espaçosualg*, tal y como se detalla en la tabla 3-2 fue implementado en MapFish para la extracción, de la base de datos, de la información completa de los espacios de un edificio presentándola bajo el formato GeoJSON. Esta información se intercepta en otro controlador llamado *proyección*, el cual accede a cada una de las *features* –compuestas por polígonos– para extraer sus coordenadas y enviárselas a un módulo que calcula la proyección perpendicular ortogonal de cada uno de los puntos que definen los polígonos. Se crea así un nuevo objeto GeoJSON con las coordenadas proyectadas. Este nuevo objeto será el que se envía a la aplicación cliente –navegador *Web*– para su representación. La Figura 4.11 muestra el esquema del proceso de proyección de geometrías de un edificio explicado.

Para la implementación del algoritmo de proyección en perspectiva, se utilizó la ayuda de la librería NumPy para Python, la cual soporta y opera con arrays y matrices multidimensionales y proporciona, a su vez, una gran cantidad de funciones para la realización de operaciones matemáticas de alto nivel.



**Figura 4.11 - Proceso de proyección de los pisos de un edificio**

La Figura 4.12 muestra la representación por medio de OpenLayers y GeoExt de un edificio proyectado para el cual el usuario elige los ángulos V y Az que definen el plano de proyección. Se puede observar como los elementos de los pisos proyectados se comportan de la misma manera que los elementos planos explicados en el capítulo 3, pudiendo por ejemplo, al ser seleccionados, visualizarse su información asociada.



**Figura 4.12 - Representación de un edificio bajo la proyección paralela ortogonal con el control de selección de *features* activado, utilizando la librería OpenLayers.**

## 5 CONCLUSIONES

### 5.1 RESULTADOS

El uso Internet y, concretamente, de aplicaciones *WebSIG* supone una gran ventaja a la hora de divulgar y gestionar información espacial de interiores de edificios de carácter público, como es el caso de una universidad. Entre otras numerosas aplicaciones, esta permite por ejemplo: al nuevo alumno, localizar las clases de su nuevo horario lectivo; al administrador, obtener una perspectiva de los recursos y su distribución espacial; o incluso al visitante, localizar el anfiteatro donde tiene lugar un seminario.

La adopción de estándares abiertos permitieron la interoperabilidad y por tanto la integración de diferentes tecnologías y fuentes de datos geográficos en el mismo proyecto. Las tecnologías libres y *open source* utilizadas en la realización de este proyecto permitieron la construcción de un *WebSIG* personalizado y no supusieron ningún coste de licenciamiento, suponiendo en ese sentido buenas elecciones.

Se puede concluir de la realización de este proyecto, que existen actualmente suficientes tecnologías abiertas y *open source* como para permitir el desarrollo de aplicaciones *WebSIG* completas que nada tienen que envidiar a las desarrolladas con tecnologías propietarias. De echo, son tantas las opciones existentes que el proceso de elección de las herramientas adecuadas se torna complicado.

Como resultado final de este trabajo, se puede afirmar que se ha obtenido un completo *WebSIG*, de fácil uso y apariencia amigable, cuya utilización y funcionalidad puede resultar de gran ayuda para la divulgación y gestión de los espacios de la Universidad do Algarve. Este presenta un interfaz de uso que, entre otros, permite: obtener una visión global de la localización de campus y edificios; la navegación por los edificios con las plantas de los pisos situadas en un mismo plano, estando la correspondiente al piso más bajo a la izquierda y la del piso más alto a la derecha; el acceso a los datos de cada espacio, a datos agregados y a estadísticas; la visualización interactiva en 3D, la cual realza las relaciones verticales entre los espacios de cada edificio.

## 5.2 PROBLEMAS Y PRERREQUISITOS PARA EL DESARROLLO DE LA APLICACIÓN

El aprendizaje del lenguaje de programación JavaScript fue uno de los pilares sin el cual este proyecto no se podría haber llevado a cabo. Resulta prácticamente imprescindible el empleo de JavaScript para el desarrollo de aplicaciones cliente *WebSIG* de carácter *open source*. Sin embargo, el aprendizaje de este lenguaje es relativamente rápido e intuitivo, sobre todo si se tiene experiencia o nociones previas de programación y de otros lenguajes, como por ejemplo C, cuya sintaxis es similar a la de JavaScript. Esta facilidad y flexibilidad ofrecida por JavaScript puede tornarse fácilmente en contra del programador, ya que sin una estricta técnica y organización el código puede crecer de forma descontrolada confundiendo hasta al propio programador.

HTML y CSS, por ser los lenguajes básicos para creación de páginas *Web*, se convierten en fundamentales en el desarrollo de una aplicación *WebSIG*. Su aprendizaje es relativamente simple y están ampliamente documentados.

Para la programación del servidor, dado a que se optó por utilizar Pylons, fue imprescindible el uso del lenguaje Python. A pesar de su enorme simplicidad y limpieza a la hora de programar, para poder comprender y explotar lo mejor posible los recursos del servidor Pylons, resulta muy importante tener unos buenos conocimientos de base de programación orientada a objetos.

## 5.3 TRABAJO FUTURO

Los sistemas de información están siempre en constante evolución, tanto para mejorar su rendimiento e interfaz como para la incorporación de nuevas utilidades que permitan incrementar las capacidades analíticas a los usuarios.

A pesar de que actualmente la aplicación *MapUAlg* presenta estabilidad y funcionalidad suficiente para entrar en su fase de producción, existen algunas mejoras que se pueden realizar. Entre ellas se destacan:

- La creación de un sistema de control de acceso de usuarios integrado con el sistema de usuarios ya existente de la Universidade do Algarve.
- La implementación de un sistema de impresión que permita la creación de diseños de impresión de los mapas conforme a las necesidades de los usuarios.

El sistema de visualización 3D desarrollado supone un punto de partida interesante para aplicaciones y mejoras sobre el mismo, entre las cuales vale la pena destacar las siguientes:

- La creación de un sistema de navegación basado en el cálculo y visualización de los caminos más cortos entre dos puntos de un mismo edificio. La visualización sobre las representaciones tridimensionales de los pisos supone una ventaja a la hora de interpretar los accesos verticales. La integración de la información de la ruta calculada con otros dispositivos de navegación, como GPS o localización por Wi-Fi, y sintetizadores de voz, supone la base para una posible aplicación práctica destinada a discapacitados visuales.
- La mejora de la interactividad del usuario con la visualización tridimensional por medio de controles intuitivos de los ángulos que definen la dirección de observación de los edificios, como por ejemplo por medio de barras deslizantes *–slide bars–*.
- La inclusión de una herramienta que dibuje sobre el mapa de los pisos en tres dimensiones, cuando es solicitado por el usuario para un elemento de acceso vertical (escaleras y ascensores), líneas verticales que conectan estos accesos de manera a obtener una mejor interpretación de las relaciones verticales.
- La inclusión, en la visualización de los pisos, de los elementos clave de acceso, como salidas y escaleras de emergencia.
- La inclusión de un sistema de visualización de fotografías 360° georreferenciadas en los interiores y exteriores de los edificios, permitiendo a la aplicación, en su componente divulgativa, ofrecer una mejor experiencia informativa sobre los espacios de la Universidad.

## 6 BIBLIOGRAFÍA

- Bonham, Carter (1994) - **Geographic Information Systems for Geoscientists: Modeling with GIS**. Kidlington: Elsevier.
- Carlbom, Ingrid; Paciorek, Joseph (1978) - Planar Geometric Projections and Viewing Transformations. **Computer Surveys**, Vol.10, nº 4.
- DiNucci, Darcy (1999) - Fragmented future. **Print** Vol 53, nº 4, p. 32.
- Drummond, W. J.; French, S. P. (2008) - The Future of GIS in Planning. **Journal of the American Planning Association**. nº74, p. 161-174
- ESRI (2012) - **Campus Place Finder (ArcGIS 10)**.  
WWW:<URL:http://www.arcgis.com/home/item.html?id=57a49d765f004bfe86aba7a7937bf280> [Última consulta: 9 Jul 2012].
- Fu, Pinde (2010) - **Web GIS: Principles and Applications**. Redlands, CA, USA: ESRI Press.
- Gardner, James (2009) - **The Definitive Guide to Pylons**. USA: Apress.
- Haverbeke, Marijn (2011) - **Eloquent JavaScript**. San Francisco, CA, USA: No Starch Press Inc.
- Hazzard, Erik (2011) - **OpenLayers 2.10, Beginner's Guide**. Birmingham, UK: Packt Publishing Ltd.
- Institute of Electrical and Electronics Engineers (1990) - **IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries**. New York, NY.
- Leff, A.; Rayfield, J.T. (2001) - Web-application development using the Model/View/Controller design pattern. **Enterprise Distributed Object Computing Conference, EDOC '01**. Proceedings. Fifth IEEE International. pp.118-127 WWW:<URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=950428&isnumber=20561>
- Li Hui; Gong, Xiangyang (2011) - An approach to integrate outdoor and indoor maps for books navigation on the intelligent mobile device, en **IEEE 3rd International Conference on Communication Software and Networks, ICCSN**. 27-29 May. pp.460-465 WWW:<URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6013872&isnumber=6013532> [Última consulta: 30 Mayo 2012].
- MapFish (2011) - **MapFish 2.2 documentation**. WWW:<URL:http://mapfish.org/doc/index.html> [Última consulta: 18 Agosto 2012].

- O. Obe, Regina; S. Hsu, Leo (2011) - **PostGIS in Action**. Stamford, CT, USA: Manning Publications Co.
- Olimpio, Julio; Smith, Dennis (2009) - Location, location, location. Improve Asset Management by taking GIS inside the building. Conference. IBM Pulse, ESRI.
- OpenLayers (2011) – **OpenLayers 2.11 API documentation**.  
WWW:<URL:http://dev.openlayers.org/releases/OpenLayers2.11/doc/apidocs/files/OpenLayers-js.html> [Última consulta: 23 Agosto 2012].
- Peng, Z.-R.; Tsou, M.-H. (2003). **Internet GIS: Distributed Geographic Information Services for the Internet and Wireless Networks**. New Jersey: John Wiley & Sons Ltd.
- PostgreSQL Global Development Group (2012) – **PostgreSQL 8.4.13 Documentation** WWW:<URL:http://www.postgresql.org/docs/8.4/interactive/index.html> [Última consulta: 02 Jun 2012].
- Sencha (2011) – ExtJS 3.4 API Documentation. WWW:<URL:http://docs.sencha.com/ext-js/3-4/#/api> [Última consulta: 23 Agosto 2012].
- Serrão, M., Rodrigues, J.M.F., Rodrigues, J.I. and du Buf, J.M.H (2012) - **Indoor localization and navigation for blind persons using visual landmarks and a GIS**, aceptado para la 4º Int. Conf. on Software Development for Enhancing Accessibility and Fighting Info-exclusion, Porto, Portugal, 19-22 Jul.
- Shklar, Leon; Rosen, Richard (2003) - **Web Application Architecture: Principles, protocols and practices**. West Sussex: John Wiley & Sons Ltd.
- Star, J.; Estes, J. (1990) - **Geographic Information Systems**. Englewood Cliffs, New Jersey: Prentice Hall.
- Tang, W.; Selwood, J. (2003) - **Connecting Our World: GIS Web Services**. Redlands, CA, USA: ESRI Press.
- Tyler M. (2005) - Web mapping illustrated Using Open Source GIS Toolkits. **Oreilly Media first edition**, California, USA. pp 25-45.
- van Rossum, Guido (2005) - **Guía de Aprendizaje de Python, Release 2.4.1a0**. Python Software Foundation. Fred L. Drake, Jr., editor.
- Westra, Erik (2010) - **Python Geospatial Development**. Birmingham, UK: Packt Publishing Ltd.
- Yang, C.; Wong, D.; Yang, R.; Kafatos, M.; Li, Q. (2004) - Performance-improving techniques in web-based GIS. **International Journal of Geographical Information Science**, Vol: 19. p. 319-342.

## **ANEXOS**

## ANEXO A – Sistema completo de clasificación de espacios por uso

<u>Clase</u>	<u>SubClase</u>	<u>SubSubClase</u>
<b>Salas</b>	Salas de aula	Sala de aula (general)
	Salas de informática	Sala de aula informática
	Sala de estudio	Sala de estudio Área de estudio
	Sala de dibujo	Sala de dibujo
	Gimnasio	Gimnasio
<b>Laboratorios</b>	Laboratorio de investigación	Laboratorio de investigación Área de apoyo (Lab. Inv)
	Laboratorio de enseñanza ( <i>ensino</i> )	Laboratorio de enseñanza Área de apoyo (Lab. Ens)
<b>Anfiteatros</b>	Anfiteatro	Anfiteatro (Auditorio) Anfiteatro (Área apoyo)
<b>Gabinetes</b>	Dirección	Gabinete (dirección)
	Docentes	Gabinete docentes
	Investigadores	Gabinete investigadores
	Servicios	Gabinete (servicios)
	Reuniones/Multiusos	Sala de reuniones/multiusos
<b>Áreas de apoyo</b>	I. sanitarias	Inst. sanit. Inst. sanit. - H Inst. sanit. - M Inst. sanit. – minusválidos ( <i>deficientes</i> )
	Servicios	Apoyo - Servicios Apoyo - Equipamientos
	Depósito ( <i>Arrecadação</i> )	Depósito
	Apoyo técnico	Apoyo técnico
	Circulación	Escaleras Ascensor Pasillo Atrio
	Taller ( <i>Oficina</i> )	Taller Almacén

<b>Áreas de servicio</b>	Atendimento público	Servicios académicos Secretariado Reproducción documental Papelería
	Biblioteca	Biblioteca
	Archivo	Archivo
	Restauración y similares	Cantina Restaurante Bar Alojamiento
<b>Elementos de estructura</b>	Espacio técnico	Espacio técnico Armario
	Estructurales	Patio Ventana Puerta Pared Balcón Muro / guarda Espacio verde Parquin Cubierto

## ANEXO B – Tablas y atributos de la base de datos

<u>Tabla</u>	<u>Atributo</u>	<u>Descripción</u>
<b>Campus</b>	id_campus campus_nombre	Nº identificador del campus Nombre del campus
<b>Campus_geom</b>	fid the_geom id_campus campus_area campus_nombre	Nº identificador de la entidad geométrica Geometría Nº identificador del campus Área del campus (m <sup>2</sup> ) Nombre del Campus
<b>Departamentos</b>	id nombre_departamento departamento_sigla	Nº identificador del departamento Nombre del departamento Siglas del departamento
<b>Departamentos_espacios</b>	id_campus id_edificio id_piso numsala id_espacio id_departamento porcentaje nombre_departamento	Nº identificador del campus Nº identificador del edificio Nº identificador piso Nº de sala Nº identificador del espacio Nº identificador departamento Porcentaje usado del espacio Nombre del departamento
<b>Edificios</b>	id_campus id_edificio edificio_nombre edificio_nombre2	Nº identificador del campus Nº identificador del edificio Nombre abreviado del edificio Nombre completo del edificio
<b>Edificios_geom</b>	fid the_geom id_edificio área_implantación	Nº identificador de la entidad geométrica Geometría Nº identificador del edificio Área de implantación del edificio (m <sup>2</sup> )
<b>Edificios_pisos</b>	id_campus id_edificio id_piso piso_nome  dx  dy	Nº identificador del campus Nº identificador del edificio Nº identificador del piso Nombre del piso Desplazamiento del piso en el eje x para su visualización (metros) Desplazamiento del piso en el eje y para su visualización (metros)

<b>Espacios_geom</b>	fid the_geom id_campus id_edificio id_piso id_espacio numsula id_uso uso_descripción uso_etiqueta id_responsable id_usuario área_espacio id_usosc id_usoclase clase subclase subsubclase responsable usuario	N° identificador de la entidad geométrica Geometría N° identificador del campus N° identificador del edificio N° identificador del piso N° identificador del espacio Numero de Sala N° identificador subsubclase Descripción del uso del espacio Etiqueta que describe el uso del espacio N° identificador del responsable N° identificador del usuario Área del espacio (m <sup>2</sup> ) N° identificador de la subclase N° identificador de la clase Nombre de la clase Nombre de la subclase Nombre de la subsubclase Nombre del responsable Nombre del usuario
<b>Responsables</b>	id_responsable responsable_nombre responsable_sigla	N° identificador del responsable Nombre del responsable Sigla del responsable
<b>Usoclasses</b>	id_usoclase usoclasse_nombre	N° identificador de la clase Nombre de la clase
<b>Usosubclases</b>	id_usosc usosc_nome	N° identificador de la subclase Nombre de la subclase
<b>Usosubsubclases</b>	id_uso uso_nombre	N° identificador de la subsubclase Nombre de la subsubclase
<b>Usuarios</b>	id_usuario usuario_nombre usuario_sigla id_responsable	N° identificador del usuario Nombre del usuario Sigla del usuario N° identificador del responsable