

PFC - Aplicaciones Web para trabajo colaborativo:

**Diseño de la herramienta GIA
(Gestión Integral de Accesibilidad)**

**MEMORIA
Versión 1.1 (6.5.2012)**

**Consultor: Fatos Xhafa
Tutor: Juan Carlos Naranjo (ITACA)
Autor: Javier Cañadillas Pardo (jcanadillas)**

Dedico este trabajo,

A mi mujer, Marta, sin su apoyo, fuerza y cariño no habría conseguido seguir adelante.

A mi familia, por estar siempre ahí.

A todos los amigos que me han ayudado desinteresadamente en este periplo universitario (David, Sergio, Ángel, Héctor, etc.).

A los compañeros de las PECS por enseñarme, apoyarme y hacerme el camino más fácil.

A Zoe y Héctor, por darme el punto de partida de este proyecto fin de carrera.

A los amigos de la meseta y de la costa, por hacer más llevaderos los momentos duros del estudio.

A mi hijo, Bruno, que llegará en unos meses y con el que podré disfrutar de este logro a tiempo completo.

Javier Cañadillas Pardo, Junio del 2012

Índice

1	Plan de trabajo.....	5
1.1	Resumen.....	5
1.2	Descripción del proyecto.....	8
1.2.1	Objetivos del proyecto.....	8
1.2.2	Resultados esperados.....	8
1.2.3	Análisis de riesgos.....	9
1.2.4	Alcance de la propuesta.....	10
1.2.4.1	Estudio de las tecnologías de desarrollo web.....	10
1.2.4.2	Estudio de diseño de interfaces amigables.....	10
1.2.4.3	Estudio de herramientas de prototipaje.....	10
1.2.4.4	Instalación y desarrollo de la aplicación citada.....	11
1.2.5	Organización del proyecto.....	12
1.2.5.1	Relación de actividades.....	12
1.2.5.2	Calendario de trabajo.....	15
1.2.5.3	Hitos principales.....	15
1.2.5.4	Equipo de trabajo.....	15
1.2.5.5	Definición de roles.....	15
1.2.5.6	Mecanismos de control.....	16
1.2.6	Valoración económica.....	16
2	Especificación de requisitos.....	17
2.1	Actores.....	17
2.2	Casos de uso.....	17
2.3	Descripción textual de los casos de uso.....	18
2.3.1	CU_00 Gestión de usuarios.....	18
2.3.1.1	CU_01 Login.....	18
2.3.1.2	CU_02 Ejecutar aplicación.....	19
2.3.1.3	CU_03 Salir de aplicación.....	20
2.3.1.4	CU_04 Alta usuario.....	20
2.3.1.5	CU_05 Recordar contraseña.....	21
2.3.2	CU_10 Gestión de normativas.....	22
2.3.2.1	CU_11 Ver Listado normativas.....	22
2.3.2.2	CU_12 Añadir Normativa.....	23
2.3.2.3	CU_13 Modificar Normativa.....	24
2.3.2.4	CU_14 Eliminar Normativa.....	24
2.3.3	CU_20 Gestión de bloques / subbloques.....	25
2.3.3.1	CU_21 Ver Listado bloques / subbloques.....	25
2.3.3.2	CU_22 Añadir bloques.....	26
2.3.3.3	CU_23 Modificar bloques.....	26
2.3.3.4	CU_24 Eliminar bloques.....	27
2.3.3.5	CU_25 Añadir subbloques.....	28
2.3.3.6	CU_26 Modificar subbloques.....	28
2.3.3.7	CU_27 Eliminar subbloques.....	29
2.3.4	CU_30 Gestión de preguntas.....	30
2.3.4.1	CU_31 Ver Listado preguntas.....	30
2.3.4.2	CU_32 Añadir pregunta.....	31
2.3.4.3	CU_33 Modificar pregunta.....	32
2.3.4.4	CU_34 Eliminar pregunta.....	32
2.3.5	CU_40 Gestión de acciones.....	33
2.3.5.1	CU_41 Ver Listado acciones.....	33
2.3.5.2	CU_42 Añadir acción.....	34
2.3.5.3	CU_43 Modificar acción.....	34
2.3.5.4	CU_44 Eliminar acción.....	35
2.3.6	CU_50 Gestión de precios.....	36

2.3.6.1	CU_51 Ver Listado precios	36
2.3.6.2	CU_52 Añadir precio.....	36
2.3.6.3	CU_53 Modificar precio	37
2.3.6.4	CU_54 Eliminar precio.....	38
2.3.6.5	CU_55 Importar precios	38
3	Análisis.....	41
3.1	Modelo de datos.....	41
3.1.1	Base de datos	41
3.1.2	Diagrama de clases	44
3.1.3	Estado de los objetos	46
3.2	Subsistemas	47
4	Diseño.....	49
4.1	Necesidades de arquitectura.....	49
4.2	Implantación.....	50
4.3	Estructuración	52
4.3.1	Gestión de la base de Datos.....	53
4.3.2	Componentes comunes.....	53
4.3.3	Configuración	53
4.3.4	Estilos	54
4.3.5	Validaciones / Funcionalidades extra (jquery)	54
4.3.6	Secciones	54
4.3.7	Utilidades	54
5	Prototipo.....	55
6	Implementación.....	60
6.1	Tecnologías	60
6.2	Herramientas.....	61
7	Pruebas.....	63
8	Instalación de la aplicación	64
9	Conclusiones.....	67
10	Índice de ilustraciones	68
11	Índice de tablas	69
12	Bibliografía.....	70

1 Plan de trabajo

1.1 Resumen

El Proyecto Final de Carrera (PFC) está pensado para realizar un trabajo de síntesis de los conocimientos adquiridos en otras asignaturas de la carrera y que requiera ponerles en práctica conjuntamente en un trabajo concreto.

Antes de explicar en detalle la herramienta GIA, debemos centrarnos en ciertos conceptos que serán útiles para comprender mejor las funcionalidades de la misma.

Un Plan Integral de Accesibilidad (PIA¹) municipal es una línea de actuación realizada por un Organismo competente en la materia que está avalada por la Administración correspondiente sobre los espacios y edificios de uso público, los elementos de la cadena de transportes y los sistemas de comunicación que pretende hacer accesible de forma gradual el entorno existente con el objetivo de que todos los ciudadanos sin distinción alguna lo puedan utilizar de forma libre e independiente. Toda entidad local que decida promover la accesibilidad de su espacio público y aplicar un plan municipal integral de accesibilidad necesitará ejecutar diversas acciones encuadradas en tres fases diferenciadas:

Una primera **fase de estudio y evaluación** del espacio público, ejecutada por la empresa asesora de accesibilidad. Los técnicos de accesibilidad hacen un análisis de los colectivos y evalúan las necesidades del espacio público y de las demandas de los ciudadanos, que han transmitido a través de incidencias y quejas a su entidad local. A continuación, elaboran la propuesta técnica de accesibilidad integral, la cual es estudiada por la entidad promotora, generalmente un ayuntamiento, que decide sobre su ejecución.

Si la entidad promotora decide continuar con el proyecto de accesibilidad, empieza la **fase de ejecución del plan de accesibilidad**. En esta fase, el profesional (ya sea el técnico de accesibilidad si ha sido contratado para este cometido o la oficina técnica del propio ayuntamiento) se encarga de supervisar la ejecución correcta del plan de accesibilidad, siendo coordinado y controlado por la entidad promotora y en constante comunicación con los ciudadanos. Cuando la fase de ejecución finalice, el profesional confirma que el espacio público construido y/o modificado cumple con la propuesta de accesibilidad planificada. Todas las acciones llevadas a cabo son documentadas y proporcionadas al ayuntamiento, que tendrá poder de decisión sobre su publicación.

Una vez la ejecución finaliza, el espacio público está listo para ser usado por los ciudadanos. Nos encontraríamos en la **fase de uso y mejora continua** del espacio público. El ciudadano, usuario final del espacio, hace uso, valora y denuncia la falta de accesibilidad del espacio cuando sea necesario, ya sea a título personal o a través de asociaciones y colectivos. El profesional (ya sea el técnico de accesibilidad externo o el técnico de ayuntamiento) se encarga de supervisar la accesibilidad del espacio, y de realizar una mejora continua de posibles zonas no accesibles. La entidad pública ofrece normalmente un servicio de incidencias a la ciudadanía, de forma que estos puedan realizar las quejas y denuncias cuando lo consideren oportuno. Este servicio

¹ http://cedat.webs.upv.es/que_acc_actua1.php

de incidencias es usado para detectar necesidades reales de accesibilidad en el espacio público.

De manera esquemática el ciclo es el siguiente:



Figura 1: Ciclo de vida de un PIA

Actualmente para la elaboración de los Planes de Accesibilidad se realiza un trabajo de campo utilizando planos obtenidos de la cartografía municipal en CAD (*Computer-Aided Design*), y es frecuente encontrarnos con que no está actualizada y no coincide con lo que se observa, lo que conlleva tener que actualizar los planos mientras se toman los datos relacionados con la accesibilidad. Una vez obtenidos los datos de campo se realiza una comparativa con la normativa vigente en materia de accesibilidad utilizando unas bases de datos previamente elaboradas, con el trabajo añadido de mantener al día esa base de datos y completarla con las normativas de las diferentes comunidades autónomas, o países en los que se trabaja.

Con los datos contrastados se elaboran los presupuestos de las modificaciones propuestas utilizando software específico con las bases de precios vigentes. A continuación se elaboran los informes finales utilizando programas de maqueta con los datos de campo, las modificaciones propuestas y el presupuesto estimado, así mismo se elaboran planos con toda la información complementaria a los informes.

Finalmente se exportan los resultados a los diferentes formatos de sistemas GIS² que los Ayuntamientos disponen, aunque no todos los Ayuntamientos utilizan sistemas de información geográfica. Habitualmente las entidades locales que utilizan algún sistema de información geográfico es para uso interno y los ciudadanos no tienen acceso a esa información, por tanto, tampoco tiene acceso a la información relacionada con la accesibilidad.

Por lo tanto, las empresas de diagnóstico y asesoramiento técnico en accesibilidad y las entidades locales promotoras se encuentran con **numerosas limitaciones tecnológicas** que les impiden analizar, validar y gestionar de forma práctica y eficiente la accesibilidad de su entorno: la mayor parte de información de accesibilidad ni siquiera está en formato digital, y aquella que sí lo está, no sigue un formato unificado. El proceso para desarrollar un plan de accesibilidad completo es largo y

² <http://www.gis.com/>

tedioso, ya que no existe una única herramienta que englobe todas las funcionalidades necesarias para el análisis, planificación y gestión del plan de accesibilidad, y por lo tanto el profesional necesita del uso de múltiples herramientas y software distintos. Además, el ciudadano no tiene acceso a la información actualizada del estado de la accesibilidad del Municipio.

Todos estos factores se traducen en barreras físicas y tecnológicas tanto frente a los ciudadanos con discapacidad como frente a los profesionales de accesibilidad que pueden proporcionar soluciones a dichas limitaciones. Es por ello que surge la necesidad de contar con una herramienta como GIA, la cual permita a estos organismos de gestión en accesibilidad autorizados realizar un PIA de forma ágil e intuitiva, consiguiendo con ello mejorar la accesibilidad global dentro de un municipio y por consiguiente a los ciudadanos en general. De esta manera se dispondrá de información actualizada que reduzca las barreras arquitectónicas y mejore la accesibilidad universal de los espacios públicos.

Para desarrollar esta herramienta utilizaremos soluciones de código libre, ya que tienen como principal ventaja que no tienen costes de licencia pero por otra parte no disponen de soporte técnico. A pesar de ello, las tecnologías de código libre ofrecen herramientas flexibles y plenamente funcionales para la parte de desarrollo, así como aplicaciones de servidor de alto rendimiento en entornos de producción. Tienen además la ventaja de ser multiplataforma, por lo que podrían ser instaladas en diversos sistemas operativos sin que su productividad se viese afectada. Todas estas tecnologías cuentan además con amplia documentación, y comunidades de desarrolladores y usuarios que aportan continuas mejoras.

1.2 Descripción del proyecto

En esta sección nos centraremos en la definición de los objetivos del proyecto, de los resultados que esperamos alcanzar y de los riesgos que nos podemos encontrar durante su realización.

1.2.1 Objetivos del proyecto

Este Proyecto tiene como principal objetivo el desarrollo de una aplicación web que permita realizar, de manera cómoda y fácil, un Plan Integral de Accesibilidad.

Además la aplicación nos permitirá gestionar las acciones que se ejecutarán al no cumplirse una normativa (cuando se realice un PIA) y gestionar los precios (coste) asociado a las acciones anteriores.

Dicha aplicación será accesible vía web, usará herramientas no propietarias y será independiente de cualquier sistema anfitrión. De esta forma logramos que sea accesible desde cualquier sitio a través de un navegador web, no tendremos dependencias por usar herramientas propietarias y la haremos portable a cualquier plataforma ya sea Windows, Linux, Mac, etc.

Como Objetivos Generales podemos enumerar los siguientes:

- Desarrollo de una aplicación web para trabajo colaborativo
- Uso de tecnologías de desarrollo web estándar, no propietarias y portable

Como Objetivos Específicos podemos enumerar los siguientes:

- Control de normativas
- Control de acciones
- Control de precios (coste) asociados a acciones
- Importación de precios desde el aplicativo PRESTO³

1.2.2 Resultados esperados

A lo largo del Proyecto se deberá llegar a una serie de hitos y documentos que reflejen la progresión y el conocimiento que se va obteniendo:

- Análisis de necesidades por las cuales ha surgido la idea de este PFC.
- Análisis de requerimientos recogidos con las reuniones con el cliente y que habrá al desarrollar el aplicativo.
- Documento con el Plan de Trabajo a seguir a lo largo del proyecto además de las líneas esperadas: documentación, resultados finales y la planificación del mismo. Se trata de este mismo documento.
- Documentación asociada a la instalación y configuración de la aplicación.
- Manual de usuario con el funcionamiento de la aplicación

³ <http://www.soft.es/>

Por otro lado, una vez realizado el Proyecto se espera obtener:

- Una aplicación web que nos permita en gestionar normativas, precios y acciones para agilizar la realización de un Plan Integral de Accesibilidad.
- Unas interfaces amigables con una información bien estructurada.
- Una información contextual suficiente para que el usuario sepa en todo momento donde se encuentra y que puede hacer.

1.2.3 Análisis de riesgos

A partir de la información que tenemos para la elaboración del presente Plan de Proyecto, hemos identificado los siguientes riesgos:

RI-01: Requisitos incompletos

Descripción: Todas las acciones que se puedan llevar a cabo en la aplicación deben estar definidas como requisitos.

Impacto: Plazos de Desarrollo

Probabilidad: Alta

Acciones: Revisar los requisitos recogidos; elaborar un prototipo para identificar las posibles acciones que falten.

RI-02: Falta de Definición de los Requisitos

Descripción: Todas las acciones que se puedan llevar a cabo en la aplicación deben estar bien definidas, no debe quedar dudas sobre su comportamiento.

Impacto: Plazos de Desarrollo

Probabilidad: Media

Acciones: Revisar los requisitos recogidos; elaborar un prototipo para identificar las posibles definiciones que falten.

RI-03: Periodos de inactividad

Descripción: Puede haber paradas en el proyecto debido a enfermedad, trabajo o problemas familiares.

Impacto: Plazos de Desarrollo

Probabilidad: Baja

Acciones: Recuperar las horas perdidas realizando un esfuerzo los fines de semana.

RI-04: Pérdida de documentación

Descripción: El equipo donde se está realizando el proyecto se puede estropear por diversos motivos.

Impacto: Plazos de Desarrollo

Probabilidad: Baja

Acciones: Realizar copias semanales del trabajo sobre dispositivos externos; instalación de un nuevo equipo.

RI -05: Cargas funcionales

Descripción: Debido a la carga de funcionalidades que incluye la herramienta puede que en alguna no se profundice lo suficiente.

Impacto: Plazos de Desarrollo

Probabilidad: Baja

Acciones: Realizar una programación de las distintas funcionalidades para llegar al período de entrega.

RI -06: Poca madurez tecnológica

Descripción: Falta de conocimiento de ciertas tecnologías utilizadas en este proyecto.

Impacto: Plazos de Desarrollo

Probabilidad: Baja

Acciones: Realizar un proceso de documentación sobre estas tecnologías para evitar posibles cuellos de botella.

1.2.4 Alcance de la propuesta

Se considera dentro del alcance de este proyecto:

- Estudio de las tecnologías de desarrollo web
- Estudio de diseño de interfaces amigables
- Estudio de Herramientas de Prototipaje
- Instalación y Desarrollo de la aplicación antes citada

1.2.4.1 Estudio de las tecnologías de desarrollo web

Dado que nuestra aplicación se va a ejecutar en un servidor web, será necesario realizar un estudio de las tecnologías web no propietarias que hay actualmente y determinar cuál o cuáles son las más adecuadas para nuestra aplicación.

1.2.4.2 Estudio de diseño de interfaces amigables

Para que el uso de nuestras interfaces sean lo más amigable posible para el usuario final estudiaremos las recomendaciones actuales que hay en cuanto a Usabilidad Web.

1.2.4.3 Estudio de herramientas de prototipaje

Para la generación de un prototipo, buscaremos una herramienta que nos permita evaluar además de los requisitos recogidos de la aplicación, la usabilidad de nuestras interfaces.

1.2.4.4 Instalación y desarrollo de la aplicación citada

La aplicación que vamos a desarrollar constará de las siguientes características:

- **Gestor de precios:** este subsistema realiza todas las gestiones pertinentes referentes a precios, pudiendo realizar altas manuales o importar los precios desde un fichero EXCEL con un formato determinado por la herramienta PRESTO⁴.

Estos precios indican el coste de la realización de una modificación (acción) en un edificio o elemento urbanístico de un municipio para que cumpla con una determinada normativa.

- **Gestor de acciones:** este subsistema se encarga de realizar todas las operaciones relacionadas con las acciones que se ejecutarán al no cumplirse una normativa cuando se esté evaluando un PIA de un municipio.

Las acciones pueden llevar un coste (precio) asociado o no, lo cual quedaría reflejado en el informe que se entregará a los ayuntamientos al finalizar la ejecución del PIA.

- **Gestor de normativas.** Es la parte más importante de la herramienta, ya que con ella se irán creando normativas que nos servirán para poder evaluar un PIA y realizar un informe que muestre la accesibilidad de un determinado municipio y el coste que llevaría (si es necesario) hacerlo accesible.

Cada normativa está compuesta de los siguientes elementos:

- **Bloques/Subbloques:** son los componentes de primer nivel que residen en la normativa y gracias a los cuales se va realizando una clasificación en ella. Por bloques pueden entenderse los elementos a evaluar, como por ejemplo, señales de tráfico, señales, vados, rampas, escaleras, etc.
- **Preguntas:** son componentes que residen a nivel de bloque/subbloque y permite refinar la normativa haciéndola más o menos exigente para alcanzar un nivel de accesibilidad dentro de un municipio.

Para conseguir este proceso de refinamiento, al crear la pregunta se especifica el tipo de respuesta (si o no, valores numéricos, etc.) esperada para que cumpla o no con la normativa. De esta manera a medida que vayamos haciendo preguntas más exigentes sobre medidas, por ejemplo, más se estará refinando la normativa y más completo será el proceso de evaluación del PIA.

⁴ <http://www.soft.es/>

1.2.5 Organización del proyecto

En esta sección cubriremos todas las actividades que tenemos que realizar durante el desarrollo del proyecto, incluiremos el calendario con los trabajos a realizar con los hitos principales del proyecto, el equipo de trabajo junto con sus roles, y por último estableceremos los mecanismos de control para que el proyecto se realice correctamente.

1.2.5.1 Relación de actividades

A continuación describiremos las diferentes actividades que hemos definido para la correcta realización del proyecto. Las dependencias que existen entre las actividades, se pueden consultar de forma gráfica en el fichero de "Microsoft Project 2007" que se adjunta con este documento (PFC.mpp).

AC-01: Lanzamiento del Proyecto

Descripción: Esta actividad consideramos que se realiza con la entrada en funcionamiento del aula virtual y los primeros contactos con el tutor para definir el proyecto que al final se va a seguir.

En este PFC en particular, también se ha tenido contacto con mi tutor dentro de ITACA (lugar donde trabajo) para definir (conjuntamente con el consultor) el proyecto que se va a seguir.

Estas reuniones ya se han llevado a cabo y se ha definido seguir el proyecto detallado que se muestra en el presente documento. A pesar que esta fase ha sido previa a la realización de este documento, indicamos su duración (que en este caso no será estimada) para que el Plan de Proyecto sea más completo.

Duración estimada: 10 días

Participantes: Consultor, Tutor y Alumno

Predecesoras: Ninguna

AC-02: Gestión y Coordinación del proyecto

Descripción: abarca todas las acciones que serán necesarias para la correcta marcha del proyecto. En caso de desviación, se aplicarán las medidas correctivas que creamos oportunas.

Duración estimada: Durante todo el Proyecto

Participantes: Consultor, Tutor y Alumno

Predecesoras: AC-01: Lanzamiento del Proyecto

AC-03: Análisis de necesidades

Descripción: En esta actividad se identificarán los diferentes actores, elementos relacionados y necesidades de cada ellos a través de entrevistas, cuestionarios, estados del arte.

Para esta herramienta realizamos el análisis de necesidades de los profesionales de gestión de accesibilidad, herramientas que usan, procesos de elaboración y ejecución del plan de accesibilidad, detección de nuevos conceptos, etc.

Al final de esta fase se deberá presentar un documento con las necesidades identificadas que deberá presentar el PFC que deberá ser validado por el consultor. Las fechas indicadas en el siguiente cuadro son orientativas, indicando el tiempo de realización del documento pero se entregará junto con el documento de requisitos.

Duración estimada: 5 días

Participantes: Consultor, Tutor y Alumno

Predecesoras: AC-01: Lanzamiento del Proyecto

AC-04: Definición de requisitos

Descripción: después del análisis conceptual, se abordarán los requisitos de los usuarios y de las herramientas, desde un punto de vista del producto final a obtener, definiendo no solo las capacidades del sistema, si no que incluirá la descripción de diferentes escenarios de aplicación y casos prácticos de uso que se llevarán a cabo en las fases de validación.

Al final de esta fase se deberá presentar un documento con los requisitos deseados que deberá presentar el PFC que deberá ser validado por el consultor.

Duración estimada: 5 días

Participantes: Consultor, Tutor y Alumno

Predecesoras: AC-03: Análisis de necesidades

AC-05: Diseño

Descripción: durante esta actividad se realizarán las siguientes tareas:

- Diseño del sistema de gestión profesional: definición de los subsistemas necesarios, definición de funcionalidades de cada subsistema, diseño de la arquitectura del sistema, flujo de información, etc.
- Se estudiará que solución tecnológica será la elegida para el desarrollo de la herramienta.
- Definición de los interfaces de comunicación entre los distintos elementos del sistema, es decir, como serán las herramientas de autor que necesitarán los profesionales.
- Realización de un primer prototipo no funcional, que será posteriormente validado, para poder realizar un refinamiento de todos aquellos procesos o funcionalidades que no hayan cumplido con las expectativas fijadas

Duración estimada: 6 días

Participantes: Alumno

Predecesoras: AC-04: Definición de requisitos

AC-06: Desarrollo

Descripción: abarca desde el desarrollo del prototipo a partir del diseño obtenido en la actividad anterior, pasando por una fase de testeo y finalizando con proceso de refinamiento del propio prototipo.

Duración estimada: 19 días

Participantes: Alumno

Predecesoras: AC-05: Diseño

AC-07: Validación

Descripción: esta fase abarca dos actividades fundamentales:

- Validación de los prototipos funcionales
- Validación de los prototipos pre-comerciales.

Duración estimada: 12 días

Participantes: Consultor, Tutor y Alumno

Predecesoras: AC-06: Desarrollo

AC-08: Evaluación

Descripción: abarca todas las acciones donde se evalúan la eficiencia y usabilidad de la herramienta en un entorno real.

Duración estimada: 15 días

Participantes: Consultor, Tutor y Alumno

Predecesoras: AC-07: Validación

AC-09: Defensa del PFC

Descripción: esta actividad representa la defensa que tendrá que hacer el alumno ante un tribunal (debate virtual) del TFC.

Duración estimada: 9 días

Participantes: Alumno, Tribunal

Predecesoras: AC-08: Evaluación

1.2.5.2 Calendario de trabajo

Se puede consultar con mayor detalle el archivo de Microsoft Project: PFC.mpp donde se puede apreciar claramente todas las características del calendario.

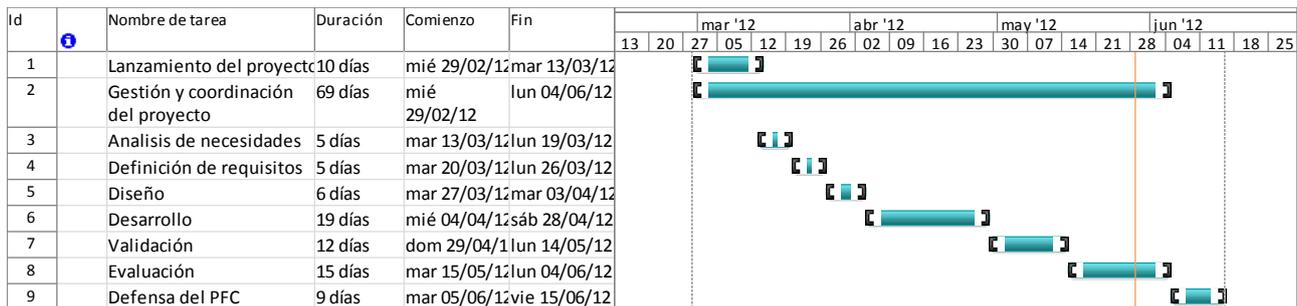


Figura 2: Calendario de trabajo

1.2.5.3 Hitos principales

Los Hitos definidos para este TFC, coinciden con las entregas parciales de las PECS

Hito	Descripción	Fecha
1	Propuesta de Proyecto	04/03/12
2	Planificación y Análisis de Requerimientos	26/03/12
3	Desarrollo	28/04/12
4	Entrega Final	04/06/12
5	Defensa del TFC	15/06/12

Tabla 1. Hitos del PFC

1.2.5.4 Equipo de trabajo

El equipo de trabajo de este proyecto se compone de dos personas:

- Consultor del proyecto: Fatos Xhafa que hará las veces de usuario final y validará que se ha cumplido todos los hitos que se han marcado para el proyecto y dando soporte para obtener la solución final de las diferentes fases.
- Tutor: Juan Carlos Naranjo, por parte de ITACA, se encargará de revisar la evolución en la etapa de desarrollo controlando que se cumplan los plazos establecidos.
- Alumno: Miguel Ángel Pineda Cruz que asumirá el resto de roles dentro de un equipo de trabajo.

1.2.5.5 Definición de roles

En la siguiente tabla se recoge los roles que cada persona del equipo de trabajo desempeña en el Proyecto.

Rol	Persona Asignada
Coordinador del Proyecto	Fatos Xhafa
Jefe de Proyecto	Juan Carlos Naranjo / Javier Cañadillas Pardo
Analista	Javier Cañadillas Pardo
Arquitecto	Javier Cañadillas Pardo
Desarrollador	Javier Cañadillas Pardo
Usuario Final	Javier Cañadillas Pardo

Tabla 2. Roles

1.2.5.6 Mecanismos de control

Para la consecución del objetivo final del Proyecto se establecen los siguientes mecanismos de control:

- El primer mecanismo de control serán los propios hitos que hemos establecidos como principales para el proyecto y que coinciden con las entregas parciales de las PECs.
- Como segundo mecanismo de control y de verificación del trabajo realizado se le informes de la evolución al consultor para que esté al tanto de la evolución del proyecto, además de borradores y guías de documentación para que valide que todo el proyecto avanza en el sentido que corresponde cumpliendo con los requisitos pedidos.

1.2.6 Valoración económica

El coste del proyecto se presupone nulo debido a que en todo momento es posible utilizar de forma adecuada software gratuito y de libre distribución; tanto para el entorno de producción (base de datos, tecnología y servidor de aplicaciones) como para el entorno de desarrollo (editores de código, visuales, entorno de trabajo...). Las horas dedicadas a la elaboración y seguimiento del mismo forman parte del calendario del PFC asignado.

Queda fuera del alcance del proyecto el mantenimiento y posibles mejoras que puedan realizarse en la herramienta.

2 Especificación de requisitos

En este apartado avanzaremos en el análisis de la aplicación que vamos a realizar, describiendo los actores y los casos de uso más importantes.

2.1 Actores

La información generada sobre accesibilidad urbanística en espacios públicos es heterogénea y proviene de fuentes muy diversas. El principal actor en el proceso de planificación de la accesibilidad universal del espacio público a través de GIA es:

- Técnicos de gestión en accesibilidad: personal técnico de empresas dedicadas a la gestión de la accesibilidad, que realizan un trabajo técnico de diagnóstico de la situación actual en cuanto a la arquitectura de la edificación pública y del urbanismo se refiere.

2.2 Casos de uso

A continuación vamos a presentar los casos de usos que se han detectado a partir del detalle de los requisitos funcionales de la aplicación.

En el diagrama siguiente podemos observar para el único actor involucrado en el manejo de la herramienta GIA (Técnico de Accesibilidad):



Figura 3: Casos de uso

2.3 Descripción textual de los casos de uso

En este párrafo vamos a describir detalladamente los casos de usos presentados anteriormente.

2.3.1 CU_00 Gestión de usuarios

En este apartado explicamos los casos de uso relacionados con la gestión de usuarios.

2.3.1.1 CU_01 Login

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera acceder a la aplicación.

Precondiciones: no estar identificado en el sistema.

Postcondiciones: el usuario accederá a la aplicación y se le mostrará la pantalla principal de normativas, con un listado de las mismas.

Casos de usos relacionados: CU_11: Ver Listado Normativas.

Escenario principal:

1. El usuario indica al sistema que quiere acceder a la aplicación.
2. El sistema muestra la pantalla "Inicio" de la aplicación.
3. El usuario registrado introduce sus datos de acceso.
4. El sistema comprueba que los datos son correctos.
5. El sistema muestra la pantalla principal de normativas.

Flujos alternativos:

- 4.1. El usuario no está registrado.
 - 4.1.1. El sistema muestra al usuario la pantalla "Inicio" que contendrá un mensaje de error con la descripción de dicho error.
 - 4.1.2. Finaliza la ejecución de este caso.
- 4.2. El usuario registrado está bloqueado.
 - 4.2.1. La aplicación muestra al usuario registrado la pantalla "Inicio" que contendrá un mensaje de error con la descripción de dicho error.
 - 4.2.2. Finaliza la ejecución de este caso.
- 4.3. Los datos están incompletos.
 - 4.3.1. El sistema muestra al usuario registrado la pantalla "Inicio" que contendrá un mensaje de error con los campos que son obligatorios.
 - 4.3.2. La ejecución del caso continúa en el punto 3.
- 4.4. Los datos no son válidos.
 - 4.4.1. El sistema muestra al usuario registrado la pantalla "Inicio" que contendrá un mensaje de error con la descripción de dicho error.
 - 4.4.2. La ejecución del caso continúa en el punto 3.
- 4.5. Error no esperado de la aplicación
 - 4.5.1. El sistema muestra al usuario registrado la pantalla "Inicio" que contendrá un mensaje de error con la descripción de dicho error.
 - 4.5.2. Finaliza la ejecución de este caso.

2.3.1.2 CU_02 Ejecutar aplicación

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario ejecute la aplicación.

Precondiciones: se ha realizado correctamente CU_01: Login.

Postcondiciones: el usuario accederá a la aplicación y se le mostrará la pantalla principal de normativas, con un listado de las mismas.

Casos de usos relacionados: CU_11: Ver Listado Normativas

Escenario principal:

1. El usuario registrado introduce sus datos de acceso.
2. El sistema comprueba que los datos son correctos.
3. El sistema muestra la pantalla principal de normativas.

Flujos alternativos:

- 2.1. Error no esperado de la aplicación
 - 2.1.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.

2.1.2. Finaliza la ejecución de este caso.

2.3.1.3 CU_03 Salir de aplicación

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario registrado quiera cerrar su sesión con la aplicación.

Precondiciones: se ha realizado correctamente CU_01: Login y CU_11: Ejecutar aplicación

Postcondiciones: al usuario se le muestra la pantalla de "Login" de la aplicación.

Casos de usos relacionados: ninguno.

Escenario principal:

1. El usuario registrado indica al sistema que quiere cerrar la sesión con la aplicación.
2. El sistema cierra la sesión del usuario con la aplicación.
3. La aplicación muestra la pantalla "Login".

Flujos alternativos:

- 2.1. Error no esperado de la aplicación
 - 2.1.1. El sistema muestra al usuario registrado la pantalla "Inicio" que contendrá un mensaje de error con la descripción de dicho error.
 - 2.1.2. Finaliza la ejecución de este caso.

2.3.1.4 CU_04 Alta usuario

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario registrado quiera darse de alta.

Precondiciones: ninguna

Postcondiciones: al usuario accede a la pantalla "Listado de normativas" con un mensaje de bienvenida al sistema.

Casos de usos relacionados: CU_11 Ver Listado normativas.

Escenario principal:

1. El usuario indica que quiere darse de alta en el sistema.
2. El sistema muestra el formulario de alta al usuario.
3. El usuario introduce sus datos, donde la contraseña será encriptada con el algoritmo MD5⁵.
4. El sistema valida los datos introducidos.
5. El sistema muestra la pantalla "Listado de Normativas" con un mensaje de bienvenida al nuevo usuario.

⁵ <http://es.wikipedia.org/wiki/MD5>

Flujos alternativos:

- 4.1. El usuario ya está registrado.
 - 4.1.1. El sistema muestra al usuario un mensaje de alerta, indicando la duplicidad del usuario en el sistema.
 - 4.1.2. La ejecución del caso continúa en el punto 3.
- 4.2. Los datos están incompletos
 - 4.2.1. El sistema muestra al usuario un mensaje de alerta, indicando el campo obligatorio que falta por rellenar.
 - 4.2.2. La ejecución del caso continúa en el punto 3.
- 4.3. Los datos no son válidos.
 - 4.3.1. El sistema muestra al usuario un mensaje de alerta, indicando el campo obligatorio que no es válido.
 - 4.3.2. La ejecución del caso continúa en el punto 3.
- 4.4. Error no esperado de la aplicación.
 - 4.4.1. El sistema muestra al usuario un mensaje de error con la descripción de dicho error.
 - 4.4.2. Finaliza la ejecución de este caso.

2.3.1.5 CU_05 Recordar contraseña

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario registrado olvide su password y solicite uno nuevo.

Precondiciones: CU_04: Alta usuario

Postcondiciones: el usuario podrá acceder al sistema con la nueva contraseña suministrada.

Casos de usos relacionados: ninguno.

Escenario principal:

1. El usuario indica que ha olvidado su contraseña.
2. El sistema muestra el formulario para recordar la dirección de correo electrónico con la cual se registro el usuario.
3. El sistema verifica la existencia de una cuenta con ese correo electrónico e identifica al usuario. Se envía un correo electrónico a este usuario con un identificador único que le permitirá resetear su contraseña.
4. El usuario recibe un nuevo correo. Este correo contiene un enlace que apunta a una página privada con un identificador único para este usuario.
5. El sistema obtiene el identificador del correo electrónico y lo compara con el que tiene guardado en la base de datos. Si coinciden, crea una contraseña aleatoria, la asigna al usuario y la envía por correo electrónico al usuario.
6. El usuario recibe un correo electrónico con la nueva contraseña, recuperando de nuevo el acceso a su cuenta en Vadeo.

Flujos alternativos:

- 3.1. El usuario (mail) no existe en el sistema.
 - 3.1.1. El sistema muestra al usuario un mensaje de alerta, indicando la no existencia del usuario en el sistema.
 - 3.1.2. La ejecución del caso continúa en el punto 2.
- 3.2. Error no esperado de la aplicación.

- 3.2.1 El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
- 3.2.2. Finaliza la ejecución de este caso.

Observaciones:

- o *Los correos que recibirá el usuario serán enviados desde la cuenta pfcGIA2012@gmail.com*

2.3.2 CU_10 Gestión de normativas

En este apartado explicamos los casos de uso relacionados con la gestión de normativas.

2.3.2.1 CU_11 Ver Listado normativas

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera ver el listado de normativas almacenadas en el sistema.

Precondiciones: se ha realizado correctamente CU_01: Login y CU_11: Ejecutar aplicación

Postcondiciones: se obtiene la lista con las normativas disponibles en el sistema para el usuario logado.

Casos de usos relacionados: CU_12: Añadir normativa, CU_13: Modificar normativa

Escenario principal:

1. El usuario ha pulsado la pestaña Normativas del menú o acaba de acceder al aplicativo.
2. El sistema muestra la pantalla principal de normativas.

Flujos alternativos:

4.1. Error no esperado de la aplicación

- 4.1.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
- 4.1.2. Finaliza la ejecución de este caso.

2.3.2.2 CU_12 Añadir Normativa

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera añadir una nueva normativa.

Precondiciones: se ha accedido desde CU_11: Ver Listado Normativas pulsando el botón "Nueva normativa".

Postcondiciones: nueva normativa registrada en la aplicación.

Casos de usos relacionados: CU_11: Ver Listado Normativas.

Escenario principal:

1. El usuario indica al sistema que quiere registrar una nueva normativa en la aplicación desde la pantalla "*Listado de normativas*".
2. El sistema muestra el formulario para dar de alta una normativa.
3. El usuario introduce la información de la nueva normativa.
4. El sistema valida los datos introducidos.
5. El sistema indica mediante un mensaje que el proceso de alta ha ido bien.

Flujos alternativos:

- 4.1. Los datos están incompletos.
 - 4.1.1. El sistema muestra al usuario un mensaje de alerta, indicando el campo obligatorio que falta por rellenar.
 - 4.1.2 La ejecución del caso continúa en el punto 3.
- 4.2. Error no esperado de la aplicación
 - 4.2.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
 - 4.2.2. Finaliza la ejecución de este caso.
- 4.3. Los datos no son válidos.
 - 4.3.1. El sistema muestra al usuario un mensaje de alerta, indicando el campo obligatorio que no es válido.
 - 4.3.2. La ejecución del caso continúa en el punto 3.

2.3.2.3 CU_13 Modificar Normativa

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera modificar una normativa.

Precondiciones: se ha accedido desde CU_11: Ver Listado Normativas.

Postcondiciones: se modifica algún campo de una normativa ya existente.

Casos de usos relacionados: CU_11: Ver Listado Normativas.

Escenario principal:

1. El usuario selecciona una de las normativas mostradas en la pantalla "*Listado de normativas*".
2. El sistema muestra el formulario para modificar una normativa.
3. El usuario modifica la información que cree oportuna en la normativa.
4. El sistema valida los datos introducidos.
5. El sistema indica mediante un mensaje que el proceso de modificación ha ido bien.

Flujos alternativos:

- 4.1. Los datos están incompletos.
 - 4.1.1. El sistema muestra al usuario un mensaje de alerta, indicando el campo obligatorio que falta por rellenar.
 - 4.1.2 La ejecución del caso continúa en el punto 3.
- 4.2. Error no esperado de la aplicación
 - 4.2.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
 - 4.2.2. Finaliza la ejecución de este caso.

2.3.2.4 CU_14 Eliminar Normativa.

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera eliminar una normativa.

Precondiciones: se ha accedido desde CU_11: Ver Listado Normativas.

Postcondiciones: se elimina una normativa existente.

Casos de usos relacionados: CU_13: Modificar normativa.

Escenario principal:

1. El usuario selecciona una de las normativas mostradas en la pantalla "*Listado de normativas*".
2. El usuario procede a la eliminación de la normativa.
3. El sistema valida que se puede realizar la eliminación de la normativa.
4. El sistema vuelve a la pantalla "*Listado de Normativas*" indicando mediante un mensaje que el proceso de eliminación ha ido bien.

Flujos alternativos:

- 4.1. Error no esperado de la aplicación

4.1.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.

4.1.2. Finaliza la ejecución de este caso.

2.3.3 CU_20 Gestión de bloques / subbloques

En este apartado explicamos los casos de uso relacionados con la gestión de bloques / subbloques.

2.3.3.1 CU_21 Ver Listado bloques / subbloques

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera ver el listado de bloques/subbloques almacenados en el sistemas y asociados a una normativa y tipo de proyecto.

Precondiciones: se ha realizado correctamente CU_01: Login, CU_11: Ejecutar aplicación y CU_12: Añadir normativa

Postcondiciones: se obtiene la lista con los bloques/subbloques disponibles en el sistema y asociados a una normativa y tipo de proyecto.

Casos de usos relacionados: CU_13: Modificar normativa

Escenario principal:

1. El usuario ha pulsado la pestaña Normativas del menú o acaba de acceder al aplicativo.
2. El sistema muestra la pantalla principal de normativas, con el listado de normativas.
3. El usuario selecciona una normativa existente.
4. El sistema muestra la pantalla de edición de normativas con un submenú operativo ("*Básicos*" y "*Bloques*").
5. El usuario selecciona la pestaña *Bloques* de submenú mostrado.
6. El sistema muestra la pantalla de listado de bloques / subbloques.
7. El usuario selecciona un tipo de proyecto.
8. El sistema cargará (si existen) los bloques asociados a la normativa seleccionada en el punto 3 y con la tipología elegida en el punto 7.
9. El usuario selecciona (si existen) un bloque del listado de bloques.
10. El sistema cargará (si existen) los subbloques asociados a la normativa seleccionada en el punto 3, con la tipología elegida en el punto 7 y el bloque marcado en el punto 9.

Flujos alternativos:

- 8.1. Error no esperado de la aplicación
 - 8.1.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
 - 8.1.2. Finaliza la ejecución de este caso.

2.3.3.2 CU_22 Añadir bloques

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera añadir un nuevo bloque.

Precondiciones: se ha accedido desde CU_21: Ver Listado bloques/subbloques pulsando el botón "Crear bloque".

Postcondiciones: nuevo bloque registrado en la aplicación.

Casos de usos relacionados: CU_21: Ver Listado bloques / subbloques.

Escenario principal:

1. El usuario indica al sistema que quiere registrar un nuevo bloque en la aplicación desde la pantalla "*Listado de bloques/subbloques*". Para ello primero selecciona un tipo de proyecto y pulsa el botón "*Crear Bloque*".
2. El sistema muestra el formulario (en la misma pantalla) para dar de alta un bloque.
3. El usuario introduce la información del nuevo bloque.
4. El sistema valida los datos introducidos.
5. El sistema indica mediante un mensaje que el proceso de alta ha ido bien.

Flujos alternativos:

- 4.1. Los datos están incompletos.
 - 4.1.1. El sistema muestra al usuario un mensaje de alerta, indicando el campo obligatorio que falta por rellenar.
 - 4.1.2 La ejecución del caso continúa en el punto 3.
- 4.2. Error no esperado de la aplicación
 - 4.2.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
 - 4.2.2. Finaliza la ejecución de este caso.

2.3.3.3 CU_23 Modificar bloques

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera modificar un bloque.

Precondiciones: se ha accedido desde CU_21: Ver Listado bloques/subbloques.

Postcondiciones: se modifica el nombre del bloque seleccionado.

Casos de usos relacionados: CU_21: Ver Listado bloques/subbloques.

Escenario principal:

1. El usuario selecciona uno de los bloques mostrados en la pantalla "*Listado de bloques/subbloques*".
2. El sistema muestra el formulario para modificar un bloque.
3. El usuario modifica (si cree oportuno) el nombre del bloque.
4. El sistema valida los datos introducidos.
5. El sistema indica mediante un mensaje que el proceso de modificación ha ido bien.

Flujos alternativos:

- 4.1. Los datos están incompletos.
 - 4.1.1. El sistema muestra al usuario un mensaje de alerta, indicando el campo obligatorio que falta por rellenar.
 - 4.1.2 La ejecución del caso continúa en el punto 3.
- 4.2. Error no esperado de la aplicación
 - 4.2.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
 - 4.2.2. Finaliza la ejecución de este caso.

2.3.3.4 CU_24 Eliminar bloques

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera eliminar un bloque.

Precondiciones: se ha accedido desde CU_21: Ver Listado bloques/subbloques.

Postcondiciones: se elimina un bloque existente.

Casos de usos relacionados: CU_21: Ver Listado bloques/subbloques.

Escenario principal:

1. El usuario selecciona una de los bloques mostrados en la pantalla "*Listado de bloques/subbloques*".
2. El usuario procede a la eliminación del bloque.
3. El sistema valida que se puede realizar la eliminación del bloque.
4. El sistema indica mediante un mensaje que el proceso de eliminación ha ido bien.

Flujos alternativos:

- 4.1. Error no esperado de la aplicación
 - 4.1.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
 - 4.1.2. Finaliza la ejecución de este caso.
- 4.2. El bloque tiene información vinculada.
 - 4.2.1. El sistema muestra al usuario un mensaje de alerta, indicando la imposibilidad de eliminar el bloque al estar vinculado a información.
 - 4.2.2 La ejecución del caso continúa en el punto 3.

2.3.3.5 CU_25 Añadir subbloques

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera añadir un nuevo subbloque.

Precondiciones: se ha accedido desde CU_21: Ver Listado bloques/subbloques pulsando el botón "Crear bloque".

Postcondiciones: nuevo subbloque registrado en la aplicación.

Casos de usos relacionados: CU_21: Ver Listado bloques / subbloques.

Escenario principal:

1. El usuario indica al sistema que quiere registrar un nuevo subbloque en la aplicación desde la pantalla "*Listado de bloques/subbloques*". Para ello selecciona un tipo de proyecto, un bloque y pulsa el botón "*Crear Bloque*".
2. El sistema muestra el formulario (en la misma pantalla) para dar de alta un subbloque.
3. El usuario introduce la información del nuevo subbloque.
4. El sistema valida los datos introducidos.
5. El sistema indica mediante un mensaje que el proceso de alta ha ido bien.

Flujos alternativos:

- 4.1. Los datos están incompletos.
 - 4.1.1. El sistema muestra al usuario un mensaje de alerta, indicando el campo obligatorio que falta por rellenar.
 - 4.1.2 La ejecución del caso continúa en el punto 3.
- 4.2. Error no esperado de la aplicación
 - 4.2.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
 - 4.2.2. Finaliza la ejecución de este caso.

2.3.3.6 CU_26 Modificar subbloques

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera modificar un subbloque.

Precondiciones: se ha accedido desde CU_21: Ver Listado bloques/subbloques.

Postcondiciones: se modifica el nombre del subbloque seleccionado.

Casos de usos relacionados: CU_21: Ver Listado bloques/subbloques.

Escenario principal:

1. El usuario selecciona uno de los subbloques mostrados en la pantalla "*Listado de bloques/subbloques*".
2. El sistema muestra el formulario para modificar un subbloque.
3. El usuario modifica (si cree oportuno) el nombre del subbloque.
4. El sistema valida los datos introducidos.
5. El sistema indica mediante un mensaje que el proceso de modificación ha ido bien.

Flujos alternativos:

- 4.1. Los datos están incompletos.
 - 4.1.1. El sistema muestra al usuario un mensaje de alerta, indicando el campo obligatorio que falta por rellenar.
 - 4.1.2 La ejecución del caso continúa en el punto 3.
- 4.2. Error no esperado de la aplicación
 - 4.2.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
 - 4.2.2. Finaliza la ejecución de este caso.

2.3.3.7 CU_27 Eliminar subbloques

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera eliminar un subbloque.

Precondiciones: se ha accedido desde CU_21: Ver Listado bloques/subbloques.

Postcondiciones: se elimina un subbloque existente.

Casos de usos relacionados: CU_21: Ver Listado bloques/subbloques.

Escenario principal:

1. El usuario selecciona una de los subbloques mostrados en la pantalla "*Listado de bloques/subbloques*".
2. El usuario procede a la eliminación del subbloque.
3. El sistema valida que se puede realizar la eliminación del subbloque.
5. El sistema indica mediante un mensaje que el proceso de eliminación ha ido bien.

Flujos alternativos:

- 4.1. Error no esperado de la aplicación
 - 4.1.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
 - 4.1.2. Finaliza la ejecución de este caso.
- 4.2. El subbloque tiene información vinculada.
 - 4.2.1. El sistema muestra al usuario un mensaje de alerta, indicando la imposibilidad de eliminar el subbloque al estar vinculado a información.
 - 4.2.2 La ejecución del caso continúa en el punto 3.

2.3.4 CU_30 Gestión de preguntas

En este apartado explicamos los casos de uso relacionados con la gestión de preguntas.

2.3.4.1 CU_31 Ver Listado preguntas

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera ver el listado de preguntas almacenadas en el sistema y asociadas a una normativa, tipo de proyecto, bloque y subbloque.

Precondiciones: se ha realizado correctamente CU_01: Login, CU_11: Ejecutar aplicación, CU_12: Añadir normativa, CU_22: Añadir bloque y CU_25: Añadir subbloque.

Postcondiciones: se obtiene la lista con las preguntas disponibles en el sistema y asociadas a una normativa, tipo de proyecto, bloque y subbloque.

Casos de usos relacionados: CU_13: Modificar normativa, CU_32: Añadir pregunta, CU_33: Modificar pregunta y CU_34: Eliminar pregunta

Escenario principal:

1. El usuario ha pulsado la pestaña Normativas del menú o acaba de acceder al aplicativo.
2. El sistema muestra la pantalla principal de normativas, con el listado de normativas.
3. El usuario selecciona una normativa existente.
4. El sistema muestra la pantalla de edición de normativas con un submenú operativo ("*Básicos*" y "*Bloques*").
5. El usuario selecciona la pestaña *Bloques* de submenú mostrado.
6. El sistema muestra la pantalla de listado de bloques / subbloques.
7. El usuario selecciona un tipo de proyecto.
8. El sistema cargará (si existen) los bloques asociados a la normativa seleccionada en el punto 3 y con la tipología elegida en el punto 7.
9. El usuario selecciona (si existen) un bloque del listado de bloques.
10. El sistema cargará (si existen) los subbloques asociados a la normativa seleccionada en el punto 3, con la tipología elegida en el punto 7 y el bloque marcado en el punto 9.
11. El usuario selecciona (si existen) un subbloque del listado de subbloques.
12. El sistema cargará (si existen) las preguntas asociadas a la normativa seleccionada en el punto 3, con la tipología elegida en el punto 7, al bloque marcado en el punto 9 y al subbloque marcado en el punto 11.

Flujos alternativos:

- 8.1. Error no esperado de la aplicación
 - 8.1.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
 - 8.1.2. Finaliza la ejecución de este caso.
- 12.1. Subir / Bajar nivel de preguntas
 - 12.1.1. El usuario sube / baja un nivel una pregunta seleccionada.
 - 12.1.2. El sistema permitirá este movimiento de nivel, si se cumplen las normas siguientes:
 - No se permite subir nivel si es la primera pregunta del listado

- No se permite bajar nivel si es la última pregunta del listado.
- 12.2. Anidar / Desanidar preguntas
- 12.2.1. El usuario anida / desanida una pregunta seleccionada.
- 12.2.2. El sistema permitirá este movimiento entre preguntas, si se cumplen las normas siguientes:
- No se permite anidar una pregunta si es la primera pregunta del listado o si la pregunta inmediatamente superior no es de tipo selección.
 - No se permite desanidar una pregunta si no se encontraba previamente anidada.

2.3.4.2 CU_32 Añadir pregunta

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera añadir una nueva pregunta.

Precondiciones: se ha accedido desde CU_31: Ver Listado preguntas pulsando el botón "Nueva pregunta".

Postcondiciones: nueva pregunta registrada en la aplicación.

Casos de usos relacionados: CU_31: Ver Listado Preguntas.

Escenario principal:

1. El usuario indica al sistema que quiere registrar una nueva pregunta en la aplicación desde la pantalla "*Listado de preguntas*".
2. El sistema muestra el formulario para dar de alta una pregunta.
3. El usuario introduce la información de la nueva pregunta.
4. El sistema valida los datos introducidos.
5. El sistema vuelve a la pantalla "*Listado de Preguntas*" indicando mediante un mensaje que el proceso de alta ha ido bien.

Flujos alternativos:

- 4.1. Los datos están incompletos.
 - 4.1.1. El sistema muestra al usuario un mensaje de alerta, indicando el campo obligatorio que falta por rellenar.
 - 4.1.2 La ejecución del caso continúa en el punto 3.
- 4.2. Error no esperado de la aplicación
 - 4.2.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
 - 4.2.2. Finaliza la ejecución de este caso.
- 4.3. Los datos no son válidos.
 - 4.3.1. El sistema muestra al usuario un mensaje de alerta, indicando el campo obligatorio que no es válido.
 - 4.3.2. La ejecución del caso continúa en el punto 3.

2.3.4.3 CU_33 Modificar pregunta

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera modificar una pregunta.

Precondiciones: se ha accedido desde CU_31: Ver Listado Preguntas.

Postcondiciones: se modifica algún campo de una pregunta ya existente.

Casos de usos relacionados: CU_31: Ver Listado Preguntas.

Escenario principal:

1. El usuario selecciona una de las preguntas mostradas en la pantalla *"Listado de preguntas"*.
2. El sistema muestra el formulario para modificar una pregunta.
3. El usuario modifica la información que cree oportuna en la pregunta.
4. El sistema valida los datos introducidos.
5. El sistema vuelve a la pantalla *"Listado de Preguntas"* indicando mediante un mensaje que el proceso de modificación ha ido bien.

Flujos alternativos:

- 4.1. Los datos están incompletos.
 - 4.1.1. El sistema muestra al usuario un mensaje de alerta, indicando el campo obligatorio que falta por rellenar.
 - 4.1.2 La ejecución del caso continúa en el punto 3.
- 4.2. Error no esperado de la aplicación
 - 4.2.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
 - 4.2.2. Finaliza la ejecución de este caso.

2.3.4.4 CU_34 Eliminar pregunta

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera eliminar una pregunta.

Precondiciones: se ha accedido desde CU_31: Ver Listado Preguntas.

Postcondiciones: se elimina una pregunta existente.

Casos de usos relacionados: CU_33: Modificar pregunta.

Escenario principal:

1. El usuario selecciona una de las preguntas mostradas en la pantalla *"Listado de preguntas"*.
2. El usuario procede a la eliminación de la pregunta.
3. El sistema valida que se puede realizar la eliminación de la pregunta.
4. El sistema vuelve a la pantalla *"Listado de Preguntas"* indicando mediante un mensaje que el proceso de eliminación ha ido bien.

Flujos alternativos:

- 4.1. Error no esperado de la aplicación

- 4.1.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
- 4.1.2. Finaliza la ejecución de este caso.
- 4.2. La pregunta tiene información vinculada.
 - 4.2.1. El sistema muestra al usuario un mensaje de alerta, indicando la imposibilidad de eliminar la pregunta al estar vinculada a información.
 - 4.2.2 La ejecución del caso continúa en el punto 3.

2.3.5 CU_40 Gestión de acciones

En este apartado explicamos los casos de uso relacionados con la gestión de acciones.

2.3.5.1 CU_41 Ver Listado acciones

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera ver el listado de acciones almacenadas en el sistema.

Precondiciones: se ha realizado correctamente CU_01: Login y CU_11: Ejecutar aplicación

Postcondiciones: se obtiene la lista con las acciones disponibles en el sistema.

Casos de usos relacionados: CU_42: Añadir acción, CU_43: Modificar acción

Escenario principal:

1. El usuario ha pulsado la pestaña Acciones del menú.
2. El sistema muestra la pantalla principal de acciones.
3. El usuario selecciona un tipo de proyecto.
4. El sistema carga (si existen) los bloques/subbloques existentes en el sistema que cumplen la tipología de proyecto seleccionada en el punto 3.
5. El usuario selecciona un subbloque del listado anterior.
6. El sistema carga (si existen) las acciones asociadas al tipo de proyecto seleccionado en el punto 3 y al subbloque seleccionado en el punto 5.

Flujos alternativos:

- 4.1. Error no esperado de la aplicación
 - 4.1.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
 - 4.1.2. Finaliza la ejecución de este caso.

2.3.5.2 CU_42 Añadir acción

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera añadir una nueva acción.

Precondiciones: se ha accedido desde CU_41: Ver Listado Acciones pulsando el botón "Nueva acción".

Postcondiciones: nueva acción registrada en la aplicación.

Casos de usos relacionados: CU_41: Ver Listado Acciones.

Escenario principal:

1. El usuario indica al sistema que quiere registrar una nueva acción en la aplicación desde la pantalla "*Listado de acciones*".
2. El sistema muestra el formulario para dar de alta una acción.
3. El usuario introduce la información de la nueva acción.
4. El sistema valida los datos introducidos.
5. El sistema vuelve a la pantalla "*Listado de Acciones*" indicando mediante un mensaje que el proceso de alta ha ido bien.

Flujos alternativos:

- 4.1. Los datos están incompletos.
 - 4.1.1. El sistema muestra al usuario un mensaje de alerta, indicando el campo obligatorio que falta por rellenar.
 - 4.1.2 La ejecución del caso continúa en el punto 3.
- 4.2. Error no esperado de la aplicación
 - 4.2.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
 - 4.2.2. Finaliza la ejecución de este caso.
- 4.3. Los datos no son válidos.
 - 4.3.1. El sistema muestra al usuario un mensaje de alerta, indicando el campo obligatorio que no es válido.
 - 4.3.2. La ejecución del caso continúa en el punto 3.

2.3.5.3 CU_43 Modificar acción

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera modificar una acción.

Precondiciones: se ha accedido desde CU_41: Ver Listado Acciones.

Postcondiciones: se modifica algún campo de una acción ya existente.

Casos de usos relacionados: CU_41: Ver Listado Acciones.

Escenario principal:

1. El usuario selecciona una de las acciones mostradas en la pantalla "*Listado de acciones*".
2. El sistema muestra el formulario para modificar una acción.
3. El usuario modifica la información que cree oportuna en la acción.
4. El sistema valida los datos introducidos.

5. El sistema vuelve a la pantalla *"Listado de Acciones"* indicando mediante un mensaje que el proceso de modificación ha ido bien.

Flujos alternativos:

4.1. Los datos están incompletos.

4.1.1. El sistema muestra al usuario un mensaje de alerta, indicando el campo obligatorio que falta por rellenar.

4.1.2 La ejecución del caso continúa en el punto 3.

4.2. Error no esperado de la aplicación

4.2.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.

4.2.2. Finaliza la ejecución de este caso.

2.3.5.4 CU_44 Eliminar acción

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera eliminar una acción.

Precondiciones: se ha accedido desde CU_41: Ver Listado Acciones.

Postcondiciones: se elimina una acción existente.

Casos de usos relacionados: CU_43: Modificar acción.

Escenario principal:

1. El usuario selecciona una de las preguntas mostradas en la pantalla *"Listado de acciones"*.

2. El usuario procede a la eliminación de la acción.

3. El sistema valida que se puede realizar la eliminación de la acción.

4. El sistema vuelve a la pantalla *"Listado de Acciones"* indicando mediante un mensaje que el proceso de eliminación ha ido bien.

Flujos alternativos:

4.1. Error no esperado de la aplicación

4.1.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.

4.1.2. Finaliza la ejecución de este caso.

4.2. La acción tiene información vinculada.

4.2.1. El sistema muestra al usuario un mensaje de alerta, indicando la imposibilidad de eliminar la acción al estar vinculada a información.

4.2.2 La ejecución del caso continúa en el punto 3.

2.3.6 CU_50 Gestión de precios

En este apartado explicamos los casos de uso relacionados con la gestión de precios.

2.3.6.1 CU_51 Ver Listado precios

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera ver el listado de precios almacenadas en el sistema.

Precondiciones: se ha realizado correctamente CU_01: Login y CU_11: Ejecutar aplicación

Postcondiciones: se obtiene la lista con las acciones disponibles en el sistema.

Casos de usos relacionados: CU_52: Añadir precio, CU_53: Modificar precio

Escenario principal:

1. El usuario ha pulsado la pestaña Precios del menú.
2. El sistema muestra la pantalla principal de precios.
3. El usuario selecciona un bloque/subbloque del listado mostrado (dichos listado son los bloques/subbloques cargados desde el fichero proveniente del aplicativo PRESTO).
4. El sistema carga (si existen) los precios asociados al bloque/subbloque seleccionado en el punto 3.

Flujos alternativos:

- 4.1. Error no esperado de la aplicación
 - 4.1.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
 - 4.1.2. Finaliza la ejecución de este caso.

2.3.6.2 CU_52 Añadir precio

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera añadir un nuevo precio.

Precondiciones: se ha accedido desde CU_51: Ver Listado Precios pulsando el botón "Nuevo precio".

Postcondiciones: nuevo precio registrado en la aplicación.

Casos de usos relacionados: CU_51: Ver Listado Precios.

Escenario principal:

1. El usuario indica al sistema que quiere registrar un nuevo precio en la aplicación desde la pantalla "*Listado de precios*".
2. El sistema muestra el formulario para dar de alta un precio.
3. El usuario introduce la información del nuevo precio.
4. El sistema valida los datos introducidos.
5. El sistema vuelve a la pantalla "*Listado de Precios*" indicando mediante un mensaje que el proceso de alta ha ido bien.

Flujos alternativos:

- 4.1. Los datos están incompletos.
 - 4.1.1. El sistema muestra al usuario un mensaje de alerta, indicando el campo obligatorio que falta por rellenar.
 - 4.1.2 La ejecución del caso continúa en el punto 3.
- 4.2. Error no esperado de la aplicación
 - 4.2.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
 - 4.2.2. Finaliza la ejecución de este caso.
- 4.3. Los datos no son válidos.
 - 4.3.1. El sistema muestra al usuario un mensaje de alerta, indicando el campo obligatorio que no es válido.
 - 4.3.2. La ejecución del caso continúa en el punto 3.

2.3.6.3 CU_53 Modificar precio

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera modificar un precio.

Precondiciones: se ha accedido desde CU_51: Ver Listado Precios.

Postcondiciones: se modifica algún campo de un precio ya existente.

Casos de usos relacionados: CU_51: Ver Listado Precios.

Escenario principal:

1. El usuario selecciona una de los precios mostrados en la pantalla "*Listado de precios*".
2. El sistema muestra el formulario para modificar un precio.
3. El usuario modifica la información que cree oportuna en el precio.
4. El sistema valida los datos introducidos.
5. El sistema vuelve a la pantalla "*Listado de Precios*" indicando mediante un mensaje que el proceso de modificación ha ido bien.

Flujos alternativos:

- 4.1. Los datos están incompletos.
 - 4.1.1. El sistema muestra al usuario un mensaje de alerta, indicando el campo obligatorio que falta por rellenar.
 - 4.1.2 La ejecución del caso continúa en el punto 3.
- 4.2. Error no esperado de la aplicación
 - 4.2.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
 - 4.2.2. Finaliza la ejecución de este caso.

2.3.6.4 CU_54 Eliminar precio

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera eliminar un precio.

Precondiciones: se ha accedido desde CU_51: Ver Listado Precios.

Postcondiciones: se elimina un precio existente.

Casos de usos relacionados: CU_53: Modificar precio.

Escenario principal:

1. El usuario selecciona una de los precios mostrados en la pantalla *"Listado de precios"*.
2. El usuario procede a la eliminación del precio.
3. El sistema valida que se puede realizar la eliminación del precio.
4. El sistema vuelve a la pantalla *"Listado de Precios"* indicando mediante un mensaje que el proceso de eliminación ha ido bien.

Flujos alternativos:

- 4.1. Error no esperado de la aplicación
 - 4.1.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
 - 4.1.2. Finaliza la ejecución de este caso.
- 4.2. El precio tiene información vinculada.
 - 4.2.1. El sistema muestra al usuario un mensaje de alerta, indicando la imposibilidad de eliminar el precio al estar vinculado a información.
 - 4.2.2 La ejecución del caso continúa en el punto 3.

2.3.6.5 CU_55 Importar precios

El sistema deberá comportarse tal y como se describe en el siguiente caso de uso, cada vez que un usuario quiera importar un listado de precios desde el aplicativo PRESTO.

Precondiciones: se ha accedido desde CU_51: Ver Listado Precios.

Postcondiciones: se carga el listado de precios proveniente del aplicativo PRESTO en el sistema.

Casos de usos relacionados: ninguno.

Escenario principal:

1. El usuario indica al sistema que quiere importar a la base de datos un listado de precios del aplicativo PRESTO, pulsando el botón *"Importar BD de Precios"* en la pantalla *"Listado de precios"*.
2. El sistema muestra el formulario para cargar el fichero desde Presto.
3. El usuario localiza el fichero y lo sube al sistema.
4. El sistema valida el fichero subido.
5. El sistema devuelve un listado con los precios importados, mostrando en color naranja los precios que ya existían y han sido modificados. Además de permitir la opción de Ignorar/Sustituir/Crear dicho precios en el sistema.

6. El usuario realiza las opciones pertinentes sobre los precios.
7. El sistema vuelve a la pantalla *"Listado de Precios"* indicando mediante un mensaje que el proceso de importación ha ido bien.

Flujos alternativos:

- 4.1. Error no esperado de la aplicación
 - 4.1.1. El sistema muestra al usuario registrado una pantalla que contendrá un mensaje de error con la descripción de dicho error.
 - 4.1.2. Finaliza la ejecución de este caso.
- 4.2. La pregunta tiene información vinculada.
 - 4.2.1. El sistema muestra al usuario un mensaje de alerta, indicando la imposibilidad de eliminar el precio al estar vinculado a información.
 - 4.2.2 La ejecución del caso continúa en el punto 3.
- 4.3. El fichero subido no cumple con el formato EXCEL.
 - 4.3.1. El sistema muestra al usuario un mensaje de alerta, indicando que el formato del fichero a subir al sistema no es el correcto.
 - 4.3.2 La ejecución del caso continúa en el punto 3.
- 4.4. Los datos están incompletos.
 - 4.4.1. El sistema muestra al usuario un mensaje de alerta, indicando el campo obligatorio que falta por rellenar.
 - 4.4.2 La ejecución del caso continúa en el punto 3.

Se adjunta con el presente documento, un fichero en formato EXCEL proveniente del aplicativo PRESTO, con nombre *"fichero ejemplo Formato PRESTO"* para realizar las pruebas pertinentes de carga de precios en el aplicativo, si fuera necesario.

En la siguiente tabla podemos ver un ejemplo de parte del fichero EXCEL creado desde el aplicativo PRESTO y adjunto con este documento:

00		Aparcamiento	
E01	u	Reservar una plaza de aparcamiento	831,64
E02	u	Construir un vado independiente	539,17
E03	u	Instalar un ascensor para subterráneo	15.310,68
E04	u	Dar tratamiento antideslizante al vado	260,81
E05	u	Reformar o rehacer vado	208,15
E06	u	Repavimentar vado	453,91
E07	u	Instalar señal vertical	123,15
E08	u	Ampliar plaza de aparcamiento en batería	43,32
E09	u	Pintar franja acceso a vado 1,50 m para plaza en línea o batería	109,00
E09.1	u	Pintar franja de seguridad 1,20 m para plaza en línea	87,20
E10	u	Pintar una señal en el suelo	5,82
E11	u	Ampliar plaza de aparcamiento en línea	37,21
E12	u	Independizar acceso de peatones y coches	558,79
E13	u	Rehacer vado para pendiente 6%	208,15
E13.1	u	Nuevo precio creado	1.000,00

01		Equipamiento	
01A		Mobiliario	
E14	u	Reservar/modificar zona en mostrador	201,80
E15	u	Sustituir los herrajes y mecanismos de accionamiento	7,76
E16	u	Reubicar bases y aparatos de control	12,27
E17	u	Disp. eléctricos temporizados con iluminación permanente	23,07
E18	u	Prolongar mobiliario hasta el suelo	95,52
E19	u	Instalar reposabrazos en los bancos.	45,39
E20	u	Sustituir un banco.	413,00
E21	u	Reubicar papeleras y buzones.	5,89
E22	u	Sustituir fotocopiadora.	1.199,00
E23	u	Sustituir fuente.	1.012,17
E24	u	Sustituir mecanismo de una fuente.	108,96
E25	NP	Sustituir mobiliario por contrastado y sin aristas (No presup.)	No Presup.
E26	u	Sustituir respaldo	124,47
E27	NP	Sustituir máquina expendedora (no presupuestado)	No Presup.
E28	NP	Sustituir cabina telefónica o de cajero (no presupuestado)	No Presup.
E29	NP	Sustituir teléfono (no presupuestado)	No Presup.
E30	u	Reubicar teléfono	5,89
E31	u	Eliminar escalón de acceso a una cabina	23,88
E32	u	Ampliar espacio previo de los aparatos (teléfono, etc.)	102,45

01B		Señalización	
E33	u	Instalar señal de emergencia	9,54
E34	u	Reubicar señal de emergencia	10,47
E35	u	Instalar alarma sonora y visual	978,44
E36	m	Señalizar superficies acristaladas o poco visibles	3,84
E37	m	Reubicar bandas de señalización	5,72
E38	u	Reservar espacios, ampliar o reubicar	54,44
E39	plnt	Directorio de planta	158,50
E40	u	Megafonía tipo viv50-80	1.514,09
E41	u	Megafonía tipo viv80-120	1.575,79
E42	u	Megafonía tipo viv120-180	1.637,49
E43	u	Megafonía tipo com50-80	2.024,04

Tabla 3: Formato fichero EXCEL proveniente de PRESTO

3 Análisis

En este capítulo describimos la funcionalidad implementada en la herramienta. A continuación definimos cómo se ha clasificado y tratado la información en el modelo de datos; detallamos la estructura de la base de datos, los componentes y subsistemas necesarios; describimos el interior del servidor web, así como el cliente web.

3.1 Modelo de datos

En este apartado realizaremos una taxonomía de la información utilizada en GIA. En primer lugar, comentaremos la Base de Datos y su estructura para mostrar cómo queda almacenada la información en el sistema. Continuaremos con un diagrama de clases y estado de los objetos utilizados, los cuales nos servirán para plasmar el diseño conceptual de la información y las relaciones entre los distintos componentes que se encargan del funcionamiento de GIA.

3.1.1 Base de datos

La herramienta GIA se basa, fundamentalmente, en una Base de Datos donde queda almacenada la información y a la cual se accede para obtener dicha información.

Podemos decir que es el elemento más importante de la aplicación, ya que la sección de interfaces puede realizarse con diferentes elementos y según estilos, pero sin una base de datos donde almacenar la información la herramienta no podría funcionar.

Toda la información sobre los elementos del sistema, normativas, precios, acciones, preguntas, así como todo lo referente a cada usuario, se tiene que tener almacenado en una Base de Datos.

El diseño más adecuado para la base de datos es un diseño relacional, donde la información se almacena en tablas que no pueden repetir nombre. Dichas tablas son, a su vez, conjuntos de registros. Cada registro representa un objeto del mundo real, poseyendo varias columnas que son sus campos o atributos.

El diseño realizado consta de seis partes bien diferenciadas.

La **primera** corresponde con la representación de Precios, en esta estructura se almacenarán todos los datos relativos a las precios bajo los cuales se asocia un coste a una determinada acción. Para ello se han creado tres tablas, la tabla Precios con la información referida a los mismos, en esta tabla existen dos campos, bloque y subbloque, que van referidos a dos tablas (bloques_presto y subbloque_presto), las cuales incluyen el valor de los precios según el bloque / subbloque seleccionado, y que ha sido precargado desde un fichero Excel exportado del aplicativo PRESTO.

La **segunda** parte es la referente a las acciones, la forma de proceder es muy similar al caso de las normativas. En este caso se tiene una sola tabla (acciones) donde se definen las acciones que se tomarían si no se cumple con una determinada normativa. Esta tabla va asociada a la tabla Precios debido a que la acción tomada puede implicar un coste.

El **tercer** bloque corresponde con la representación de las Normativas, en esta estructura se almacenarán todos los datos relativos a las normativas bajo las cuales se realizan los proyectos asociados a los planes de accesibilidad. Para ello se han creado cinco tablas, la tabla Normativas con la información referida a las mismas, esta tabla existen cuatro campos (uso, estado, ámbito y autonomía), que van referidos a sendas cuatro tablas (uso_normativas, estado_normativas, ámbito y autonomías), las cuales incluyen los usos de la normativa, los posibles estados de la misma, el ámbito que puede tener (Europeo, Nacional, Autonómico, etc.) y la autonomía (si procede) sobre la que va a tener efecto.

En **cuarto** lugar se encuentra la parte referente a los Bloques y Subbloques. Se tendrán dos tablas (bloques y subbloques) donde se definen los tipos de bloques y subbloques que contiene una determinada normativa, y los cuales almacenarán preguntas a rellenar posteriormente para la evaluación de la normativa dentro un proyecto.

El **quinto** bloque se corresponde con las Preguntas, e incluye todas las tablas referentes a las preguntas almacenadas en el sistema. Existe la tabla Preguntas con todos los detalles de cada pregunta, que además se relaciona con el bloque / subbloque. En esta tabla existen diferentes campos que nos indican el tipo de pregunta que estamos tratando, que tipo de respuesta se espera, que unidad tiene la pregunta (en el caso de ser de tipo Decimal o Entero), que tipo de acción se realizará cuando la respuesta no sea la esperada, etc. Todos estos campos están relacionados con tablas maestras (tipo_pregunta, tipo_respuesta, unidad, tipo_accion, acción, tipo_cantidad, etc.) que ayudarán en la gestión de dicha tabla aportando información fundamental.

En **último** lugar se encuentra el bloque USUARIOS con las tablas pertenecientes al control de usuarios. La tabla usuarios posee la información referente a los mismos y la tabla estado_usuarios contiene el listado de estados que puede tener un usuario. La tabla tipo_usuario indica el tipo de usuario que accede al sistema (Administrador, ayuntamiento, asociación, usuario estándar).

Las diferentes partes comentadas no están aisladas unas de otras, sino que están interconectadas a través de tablas que almacenan toda la información relativa a las relaciones existentes entre varias tablas. Además existen otras tablas que almacenan información no editable por el usuario (maestras), normalmente cargadas por el administrador, y que guardan información referente a tipologías de elementos y usuario, estados, etc. En este punto se han explicado las tablas más relevantes para el funcionamiento del sistema.

A continuación se encuentran detalladas todas las tablas descritas (incluidas las que almacenan las relaciones) con todos sus campos.

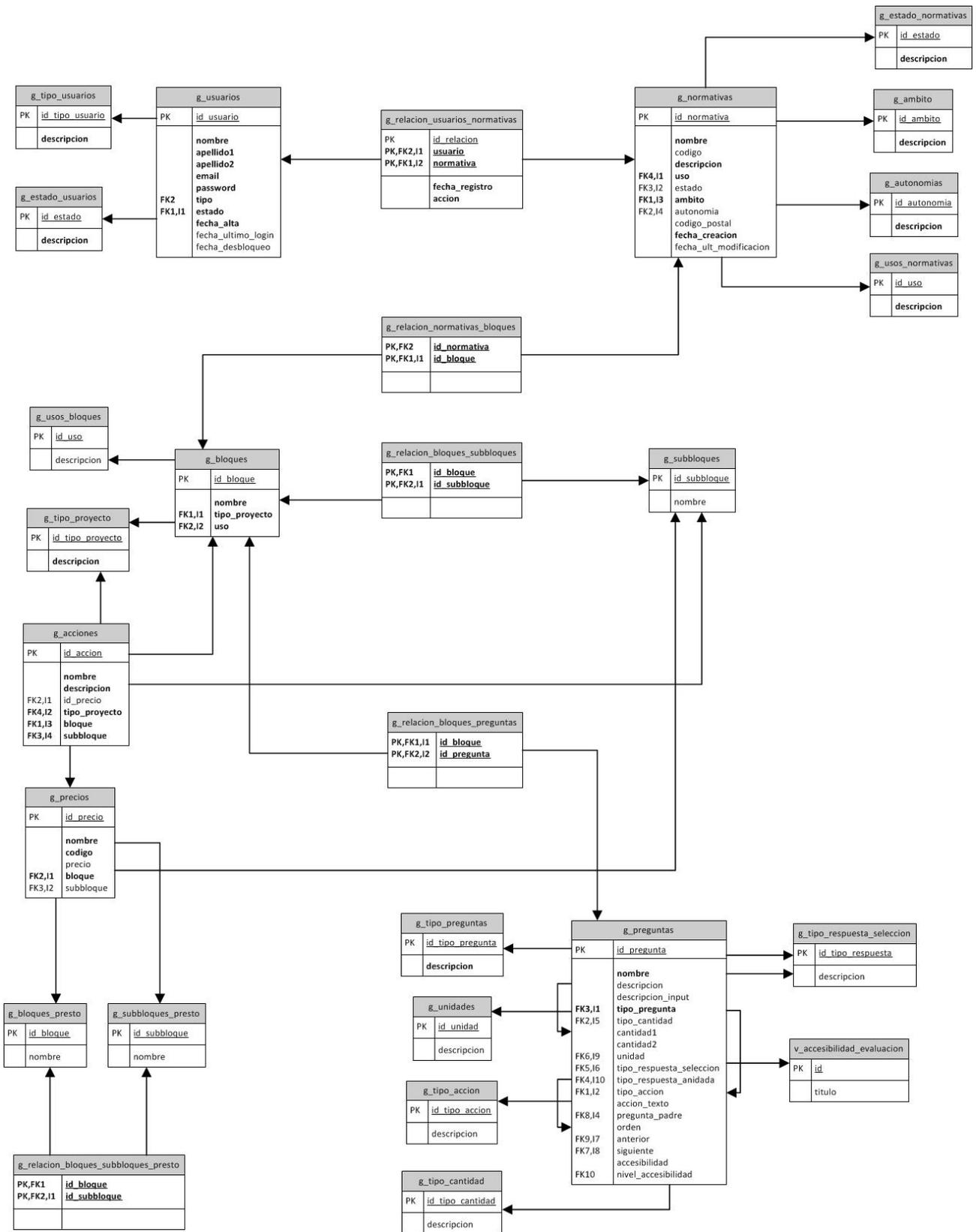


Figura 4: Base de datos

3.1.2 Diagrama de clases

El análisis de los casos de uso deriva en la definición de los conceptos básicos usados por el actor principal de la plataforma. En la figura 5, podemos ver el diagrama de clases obtenido. El diagrama de clases UML describe los tipos de objetos en el sistema y las relaciones que existen entre ellos. El diagrama también muestra los atributos principales de cada objeto, así como la cardinalidad de las relaciones.

Un análisis inicial del diagrama nos indica los objetos protagonistas del servicio, es decir, los que están presentes en la mayoría de los flujos de trabajo. Es fácilmente reconocible el objeto Normativas como eje central del servicio. Esto es correcto, puesto que toda la información del servicio se basa en la evaluación de datos a partir de normativas creadas.

Una normativa contiene bloques y subbloques que sirven para clasificar el ámbito sobre el cual se utilizará la normativa en un determinado proyecto perteneciente a un plan de accesibilidad. Una normativa también cuenta con preguntas (a nivel de bloque y subbloque) las cuales no sirve para refinarla y conseguir con ello que un proyecto supere (o no) una determinada normativa y por lo tanto declarar una edificación o urbanismo accesible.

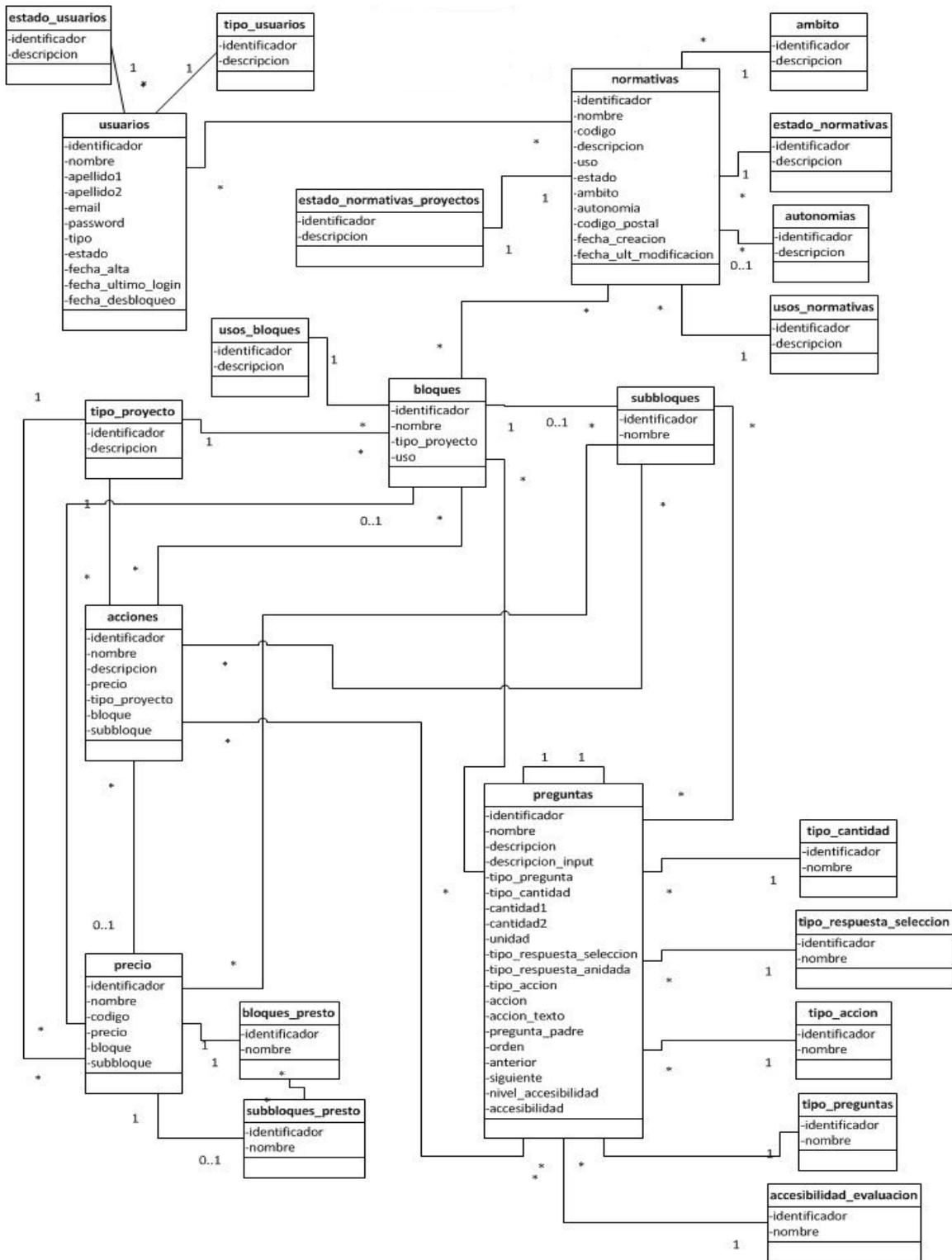


Figura 5: Diagrama de Clases

3.1.3 Estado de los objetos

Tras analizar los casos de uso y modelar los datos usados por la aplicación en un diagrama de clases (ver Figura 5) es necesario definir el comportamiento interno de los objetos más relevantes. Para ello haremos uso de los diagramas de estado. Los diagramas de estado son de gran utilidad para indicar la vida de un objeto, los estados que el objeto atraviesa desde el momento en que es creado hasta que es destruido, en respuesta a ciertos eventos.

Normativas

Las normativas se crean cuando un usuario rellena el formulario de creación correctamente. Sus posibles estados son (Ver figura 6):

- En Progreso: Estado inicial. La normativa ha sido creada pero no está activa.
- Eliminada: La normativa no es visible desde la aplicación, aunque permanece en el sistema.
- Activa: La normativa se considera completa y pasa a activa, para poder ser utilizada en el resto del sistema.

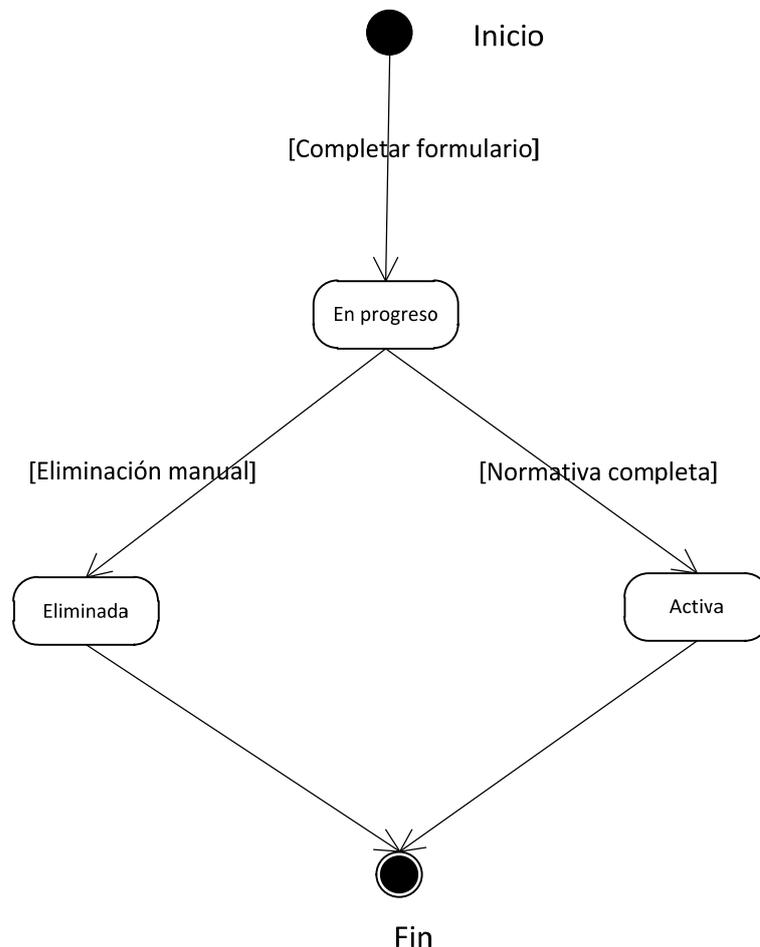


Figura 6: Diagrama de estados para normativas

3.2 Subsistemas

La herramienta GIA se compone de tres grandes subsistemas que definen la funcionalidad organizada por paquetes. Los subsistemas son Precios, Acciones y gestión de Normativas. En la figura 7 se puede apreciar una visión general de estos subsistemas.

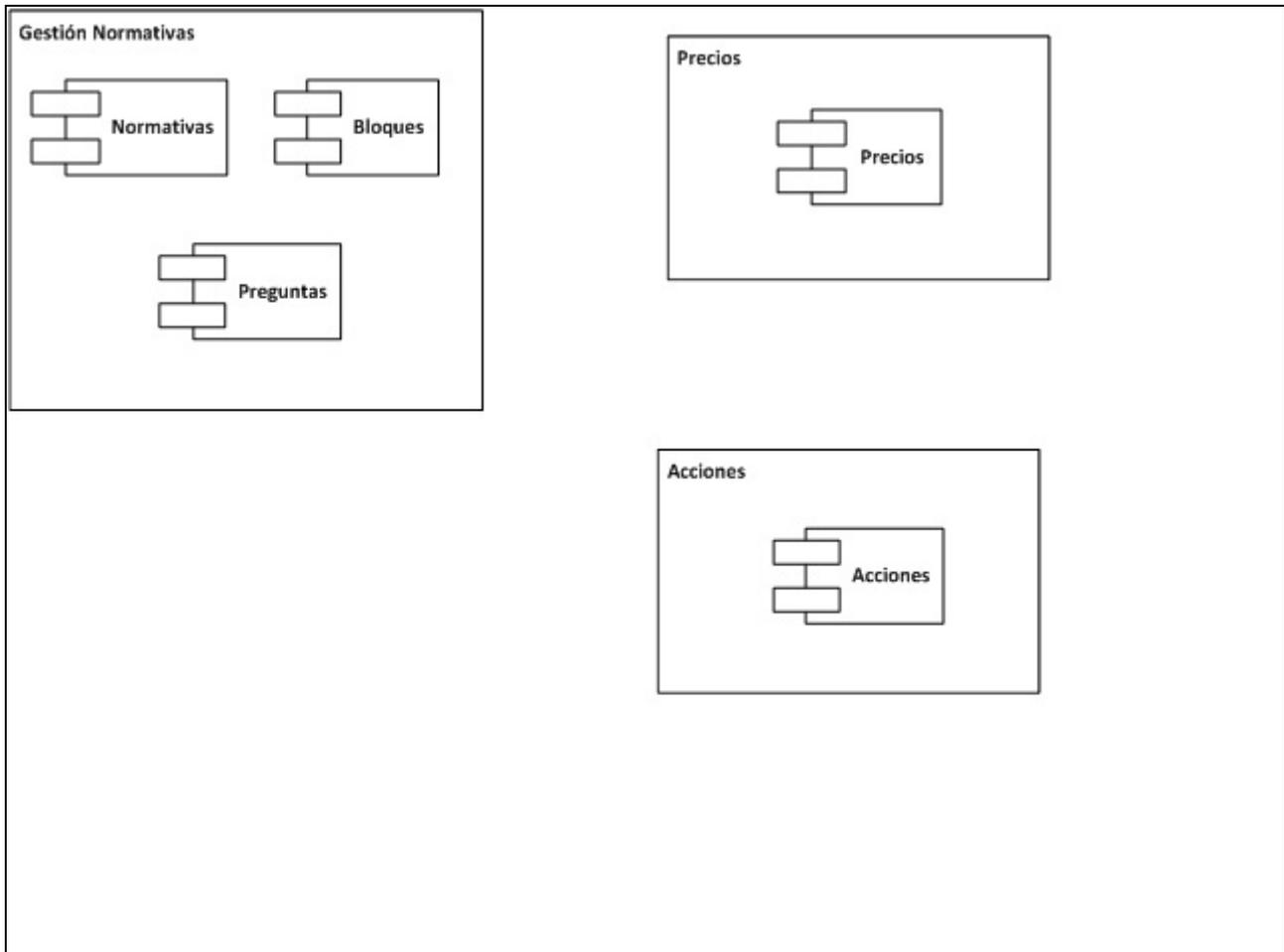


Figura 7: Subsistemas

Subsistema de Precios

Este subsistema cuenta con un módulo con el mismo nombre, el cual se encarga de la gestión de los precios que posteriormente se utilizarán en acciones correctivas para cumplir con las normativas. A través de él se podrán introducir los precios de manera manual o importándolos desde un fichero externo exportado desde el aplicativo PRESTO.

Subsistema de Acciones

Este subsistema se encarga de la gestión de las acciones que posteriormente se utilizarán cuando determinada pregunta (asociada a una normativa) no cumpla con ella y se tenga que realizar una acción para que cumpla con la normativa. A través de él se podrán introducir acciones de manera manual, así como modificar precios asociados a acciones ya existentes.

Subsistema de Normativas

Es el subsistema principal dentro de la aplicación y se encarga de gestionar todo lo referente a normativas, las cuales serán utilizadas para evaluar un Plan de Accesibilidad y realizar el correspondiente informe.

Este subsistema cuenta con tres módulos:

- Normativas: realiza toda la gestión propia de normativas, almacenando información sobre ellos y sobre los bloques/subbloques que las componen.
- Bloques: este módulo se encarga de la gestión de los bloques (Edificación o Urbanismo) que dividen las distintas secciones de una normativa.
- Preguntas: este módulo se encarga de la gestión de preguntas, las cuales refinarán más los bloques existentes y con ello la propia normativa.

4 Diseño

El objetivo principal de este proyecto es crear un entorno colaborativo que permita a los organismos encargados de la gestión en accesibilidad realizar un PIA de forma ágil e intuitiva, consiguiendo con ello mejorar la accesibilidad global dentro de un municipio y por consiguiente a los ciudadanos en general. De esta manera se dispondrá de información actualizada que reduzca las barreras arquitectónicas y mejore la accesibilidad universal de los espacios públicos.

4.1 Necesidades de arquitectura

Dado que lo que queremos es diseñar una aplicación web en un entorno colaborativo, adoptaremos un estilo arquitectónico heterogéneo y que permita la independencia de sus componentes así como una transmisión de información centralizada.

Por este motivo hemos escogido la clásica arquitectura **cliente-servidor**, instalando en los diferentes clientes la herramienta y centralizando una base de datos común en la parte del servidor. De esta manera la capacidad de proceso está repartida entre los clientes y los servidores.

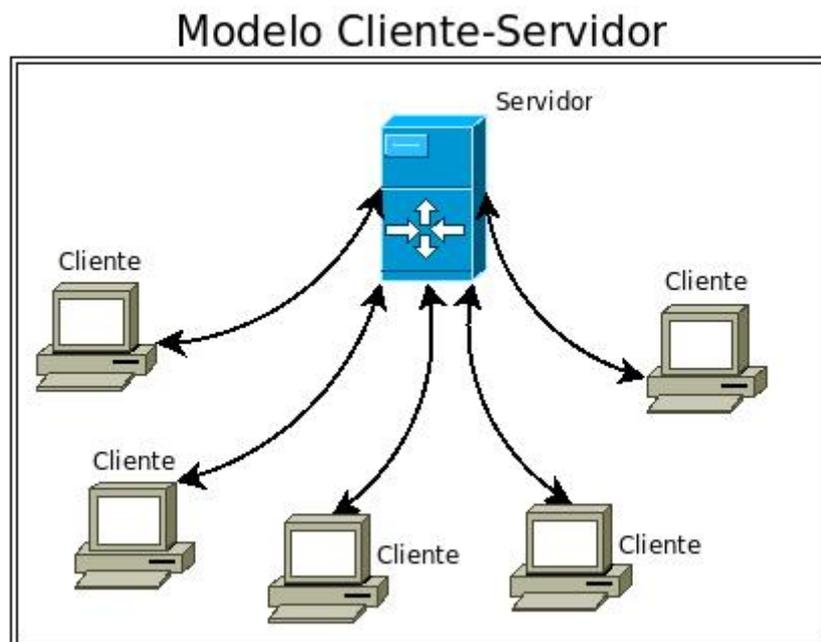


Figura 8: Modelo Cliente-Servidor

En nuestro caso la arquitectura Cliente-Servidor utilizada tiene **tres capas**:

- Clientes que interactúan con los usuarios finales (en nuestro caso, las documentos PHP)
- Servidores de aplicación que procesan los datos para los clientes (en nuestro caso, Apache)
- Servidores de la base de datos que almacenan los datos para los servidores de la aplicación (en nuestro caso, MySQL).

De esta manera, la capa del cliente solo es un interfaz que interactúa con el usuario, dejando para las otras dos capas la gestión de la aplicación. De esta manera conseguimos una mayor independencia entre las diferentes capas sin que afecte por ejemplo un posible cambio en las especificaciones de la bases de datos a la capa del cliente.

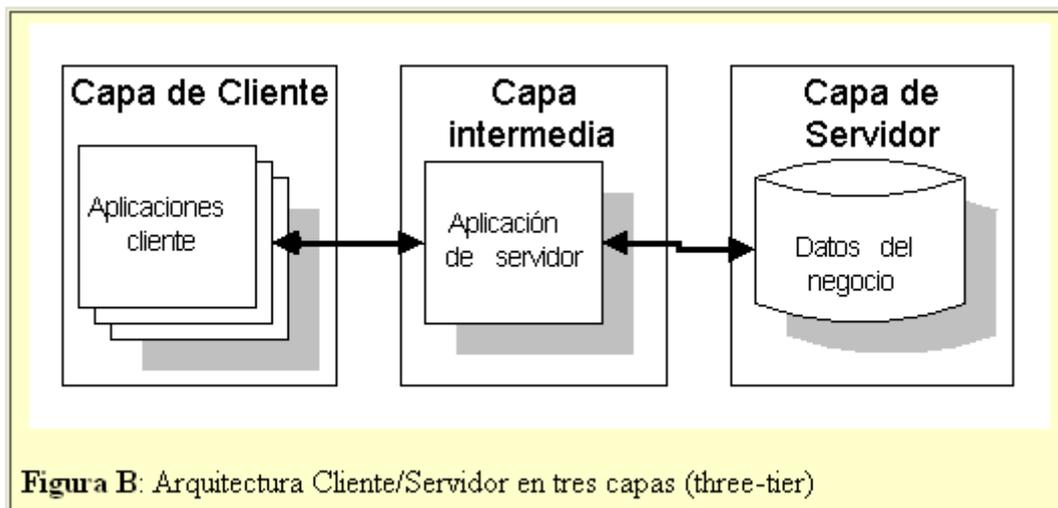


Figura 9: Arquitectura Cliente-Servidor en tres capas

4.2 Implantación

La arquitectura utilizada en GIA sigue el paradigma MVC (Modelo Vista Controlador), donde se separan los datos, la interfaz de usuario y la lógica de negocio en tres componentes distintos e independientes, definiendo la secuencia de ejecución que seguirá la herramienta para resolver las peticiones realizadas por los usuarios. El paradigma MVC y la arquitectura en tres capas son perfectamente compatibles ya que buscan la independencia entre capas las distintas capas que los forman fomentando el intercambio de información entre ellas.

En la figura siguiente podemos observar, de manera más sencilla, los distintos componentes de este paradigma y la relación que existe entre ambos.

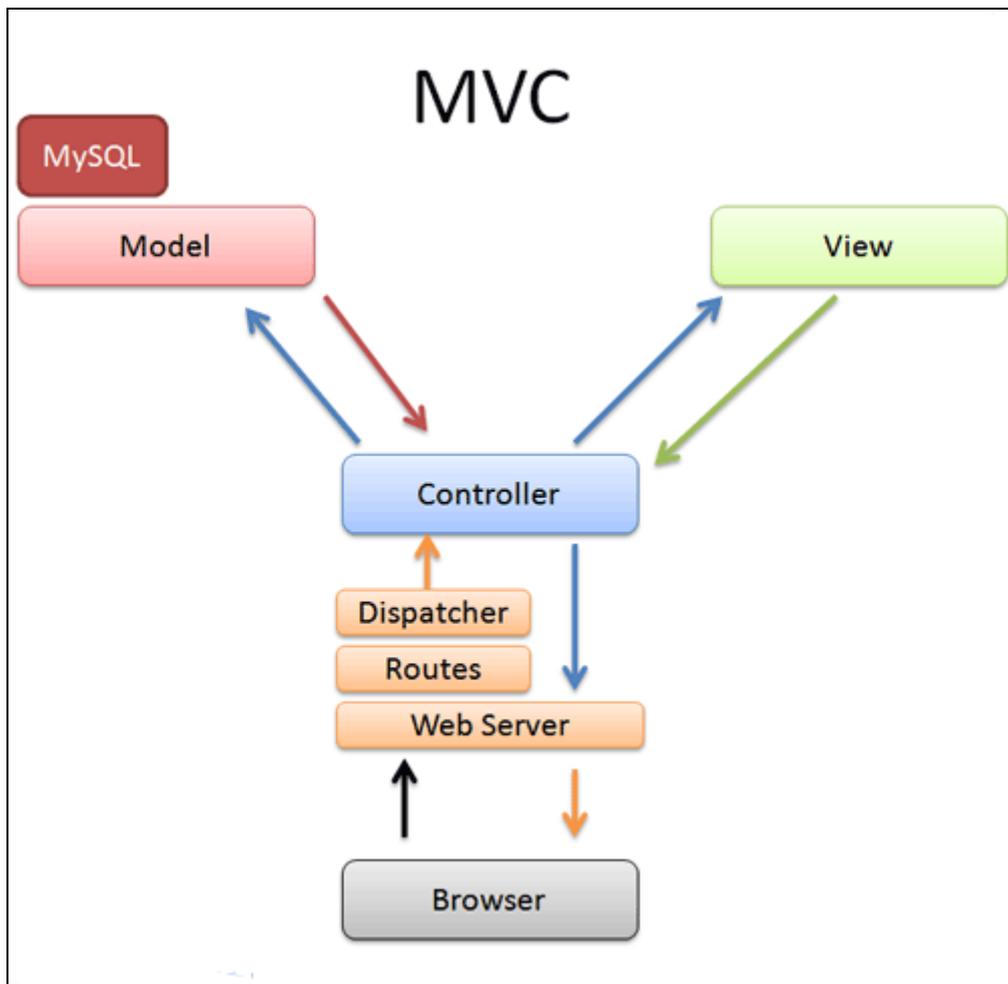


Figura 10: Paradigma MVC

En el caso particular de GIA, la implantación de los patrones arquitectónicos elegidos a dentro del aplicativo queda de la siguiente manera:

- Modelo (Base de Datos): es el sitio donde se almacena el código que obtiene la información de la base de datos (Normativas, Bloque, Subbloques, Preguntas, etc.).
- Vista (HTML): corresponde con las interfaces gráficas realizadas para la interacción con el usuario (técnico en accesibilidad).
- Controlador (PHP): es el sitio que recibe la petición desde el navegador decidiendo que página deberá ser mostrada. Esto implica invocar la lógica de negocio, obteniendo los datos (modelo) e inyectando la información en la vista para que la petición sea servida al usuario.

A continuación se describe la arquitectura y detalles de los subsistemas que componen GIA, el cual se compone de dos subsistemas internos: Servidor web y cliente web.

Servidor web:

El servidor web es el responsable de alojar la web de GIA. Obtiene consultas de los diferentes clientes, las filtra según las necesidades del servicio y las envía de vuelta a la web o el móvil, respectivamente. También es el responsable de alojar el contenido de la web, así como de procesar las distintas operaciones sobre los componentes que forman la herramienta según el perfil del usuario que acceda. Sus funcionalidades básicas son:

- Gestión de usuarios
- Gestión de normativas
- Gestión de precios
- Gestión de acciones

Cliente Web:

El cliente web consulta las páginas alojadas en el Servidor Web y las renderiza en el navegador web cliente. Toda la acción del usuario ocurre en el marco del navegador web. El usuario necesita estar registrado para poder interactuar con la plataforma.

Las funcionalidades que se pueden realizar varían según el usuario que haya accedido, a continuación mostramos un resumen de las mismas independientemente del acceso realizado:

- Mostrar Normativas
- Mostrar preguntas
- Mostrar acciones y precios operativos

4.3 Estructuración

En la estructuración del código siempre ha primado seguir la centralización y la no replicación de información, consiguiendo así un ahorro de trabajo ante posibles errores al tener todo ubicado en un único fichero y no replicado en varios.

Basándonos en los puntos anteriormente citados y siguiendo las arquitectura/paradigma comentado, procedemos a explicar la estructuración del código

4.3.1 Gestión de la base de Datos

Toda la gestión de la base de datos se realiza desde la carpeta **classes**. A continuación explicamos su estructuración:

- Subcarpeta BD: esta carpeta almacena un fichero (bd.php) que almacena todas las operaciones posibles a realizar contra la base de datos. Además, cuenta con los componentes DAOs⁶, uno por cada tabla principal (Acciones, bloques, normativas, precios y usuarios), donde se almacenarán todas las operaciones a realizar sobre cada tabla específica para ese DAO.
- Subcarpeta DTO: almacena un fichero DTO⁷ por cada tabla residente en la base de datos. Estos ficheros son un mapeo de las tablas físicas pero para ser utilizados dentro del aplicativo
- Controladores: El resto de carpetas incluyen un controlador por cada tabla principal (Acciones, bloques, normativas, precios y usuarios) para canalizar las llamadas desde los ficheros PHP (vista) hacia la base de datos (modelo).

4.3.2 Componentes comunes

Los componentes que se repiten en cada uno de los interfaces de la herramienta, como puede ser la cabecera (header), menú superior o pie (footer) se almacenan por separado y se referencian al crear la propia página. De esta manera se consigue no replicar el trabajo.

Estos ficheros se encuentran almacenados dentro de la carpeta **components**.

4.3.3 Configuración

Todos los ficheros comentados en este apartado se encuentran dentro de la carpeta **conf**.

Para facilitar la portabilidad de la herramienta o posibles cambios de Bases de datos, se ha creado un fichero de configuración en el cual se especifica las rutas sobre las que se encuentra desplegada la aplicación, el nombre de la base de datos, etc.

En este fichero también se configura el lenguaje de la herramienta, de momento solo está creada para castellano, pero cambiando un parámetro en el fichero **properties.php** y creando un fichero con todos los literales en el lenguaje elegido dentro la subcarpeta **lang** (language), se podría ver la herramienta en otro idioma. De momento solo está creado el fichero **labels_ES.php**.

Comentar también que en este fichero (**properties.php**) también almacena una variable para poder establecer el modo mantenimiento en la herramienta. De esta manera, si se necesita modificar algo de la herramienta y es necesario que los usuarios no puedan acceder, se activa este flag y no podrán tener acceso hasta que se vuelva a desactivar.

Por otra parte, tenemos el fichero **constants.php** donde se almacenan todas las constantes utilizadas a lo largo de la herramienta, de esta manera se tiene centralizado en un solo fichero y no en cada fichero que se utilice.

⁶ http://es.wikipedia.org/wiki/Data_Access_Object

⁷ http://en.wikipedia.org/wiki/Data_transfer_object

Por último, comentar el fichero **html_header.php**, que también es un componente común a todas las interfaces de la herramienta, unificando las llamadas a las hojas de estilo .css y a las funciones .js en un solo fichero.

4.3.4 Estilos

En la carpeta **css** se encuentran almacenados los ficheros que contienen las hojas de estilos utilizadas para dar formato a componentes dentro de la herramienta.

4.3.5 Validaciones / Funcionalidades extra (jquery)

En la carpeta **js** se encuentra almacenados varios ficheros de javascript, todos los que comienzan con "jquery" son los encargados de dar la funcionalidad extra aportada por JQuery⁸.

El fichero **funciones.js** almacena todas las funciones de validación realizadas en los múltiples formularios mostrados en la aplicación, de esta manera seguimos centralizando en un solo fichero muchas funcionalidades con la ventaja que eso conlleva.

El fichero **md5.js** nos aporta una funcionalidad para cifrar el password dado por el usuario con el algoritmo MD5⁹ antes de ser enviado al servidor, de esta manera se encripta antes del envío evitando posibles ataques cibernéticos.

4.3.6 Secciones

En esta carpeta (**sections**) se almacena toda la parte de la vista de la herramienta, estando dividida en carpetas por las secciones principales: Login, Normativas, Precio y Acciones.

Cada subcarpeta mantiene una estructura común, conteniendo una carpeta **gestión** que contiene ficheros para la gestión de las llamadas entre las distintos interfaces y seleccionando según que opciones a cual debe o no debe ir; y otra carpeta **components** que contiene los componentes comunes a esa sección.

Comentar también que la subcarpeta **mail** contiene dos plantillas utilizados para enviar los correos al usuario en el caso que no recuerde su contraseña e indicando su nueva contraseña asignada por el sistema.

4.3.7 Utilidades

En esta carpeta (**utils**) se almacenan utilidades necesarias para optimizar y mejorar el funcionamiento de la herramienta, tales como un manejador de excepciones y otro de errores, así como una librería que permite la lectura de ficheros EXCEL (necesaria en el apartado de Importación de Precios).

Además de la librería encargada de gestionar el envío de mails (**mail**).

⁸ <http://jquery.com/>

⁹ <http://es.wikipedia.org/wiki/MD5>

5 Prototipo

Para la realización del prototipo se ha usado una herramienta de Prototipaje Balsamiq Mockups¹⁰. Dicha Herramienta nos permite realizar un primer prototipo de la herramienta permitiendo desarrollar los interfaces utilizados de una manera fácil y amigable.

A continuación mostramos algunas capturas del prototipo (Login, Normativas, Acciones, Precios) comparándolo con la pantalla real, para que se pueda observar la utilidad de esta herramienta, en primer lugar, y lo similares que son la aplicación real al prototipo desarrollado.

En primer lugar de la pantalla de login de la aplicación, tanto en el diseño del prototipo como el diseño del aplicativo.

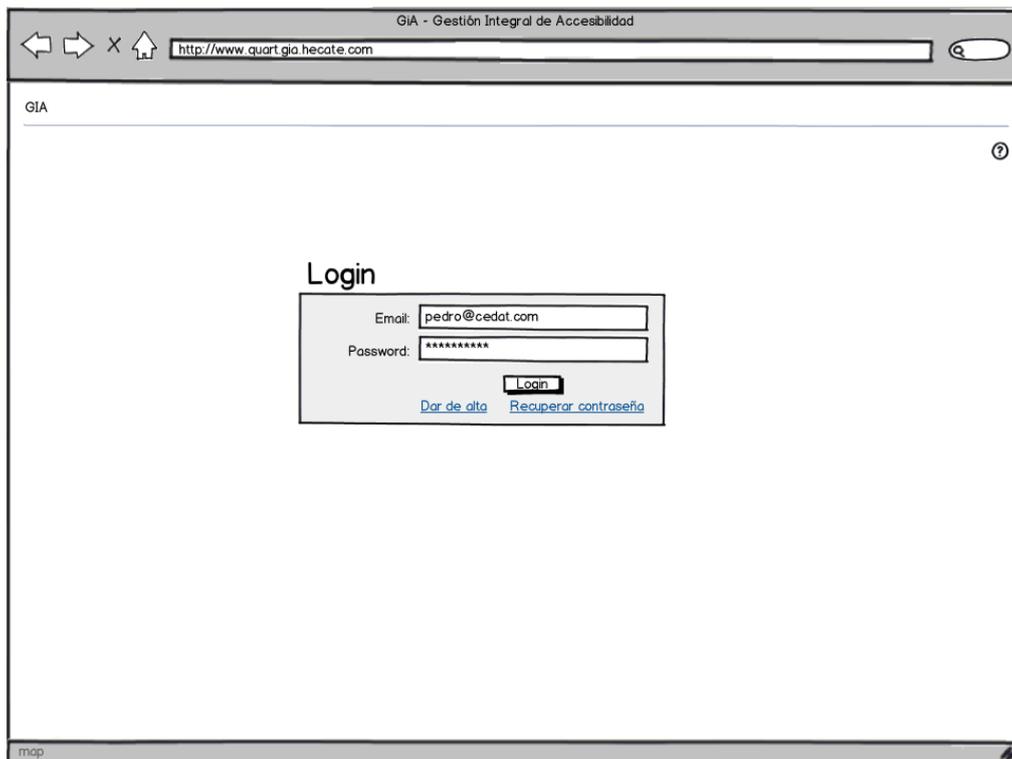


Figura 11: Login del prototipo

¹⁰ <http://www.balsamiq.com/products/mockups>



Figura 12: Login del aplicativo

Continuamos con la pantalla que muestra los listados de las normativas existentes en el sistema para el usuario que ha accedido.

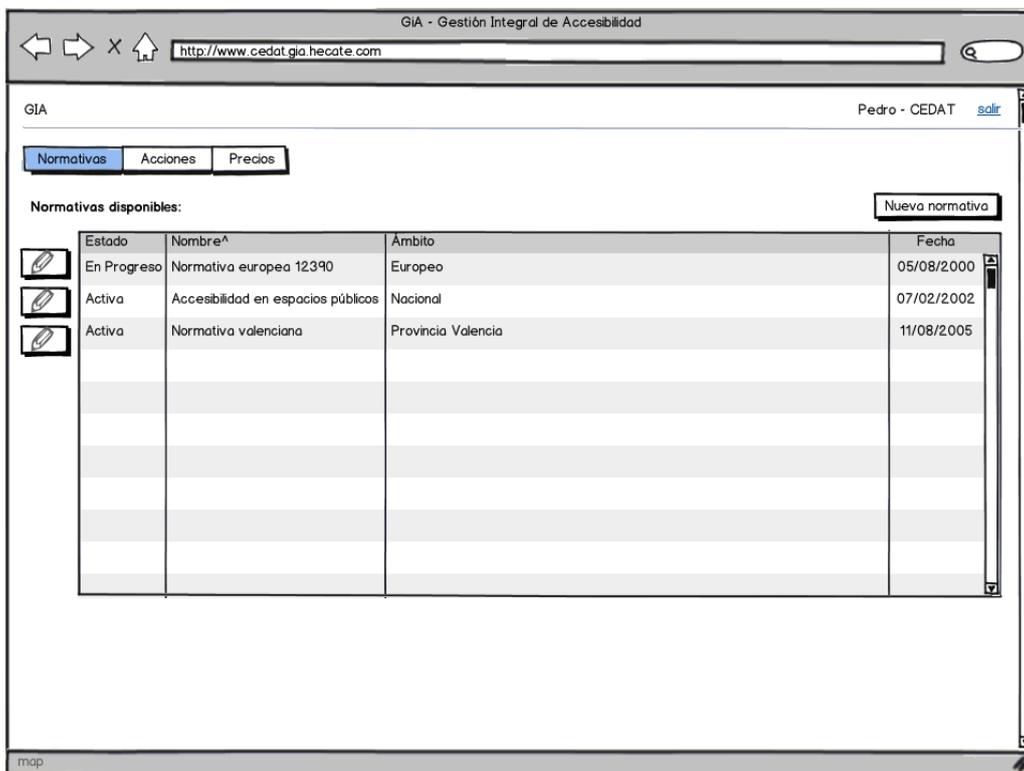


Figura 13: Listado de normativas del prototipo

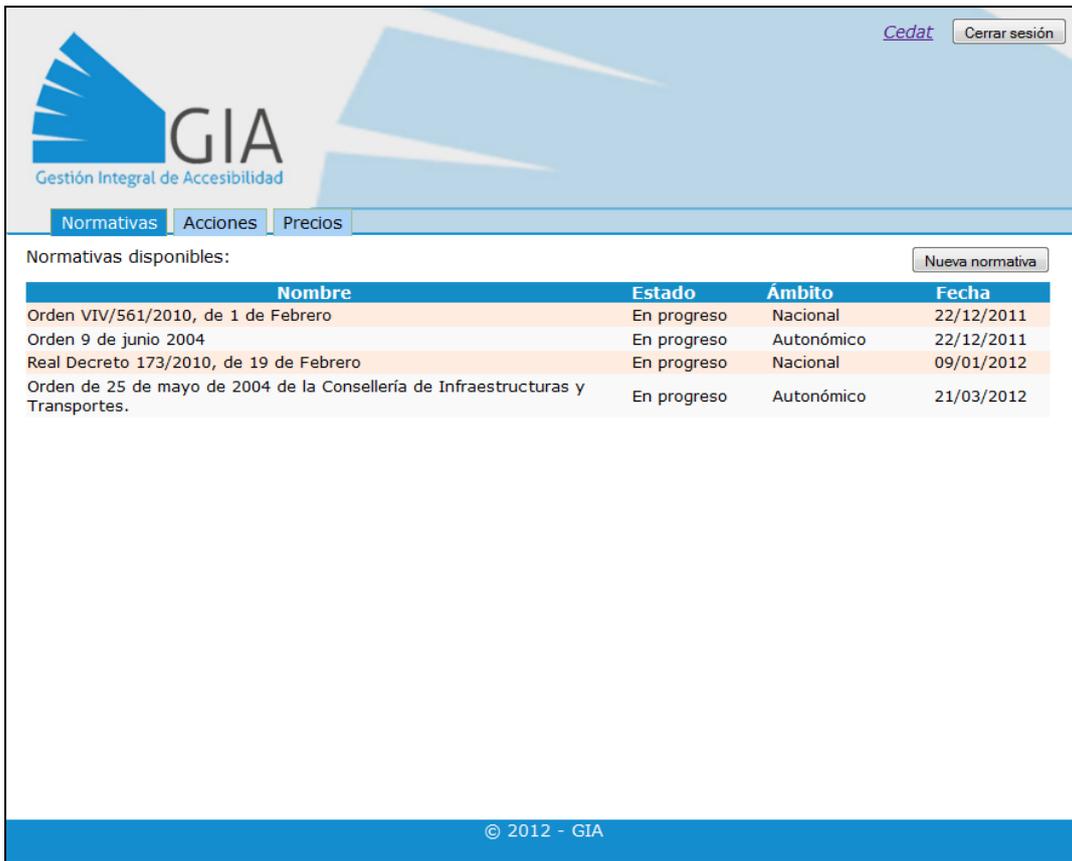


Figura 14: Listado de normativas del aplicativo

Proseguimos con la pantalla que muestra los listados de las acciones existentes en el sistema.

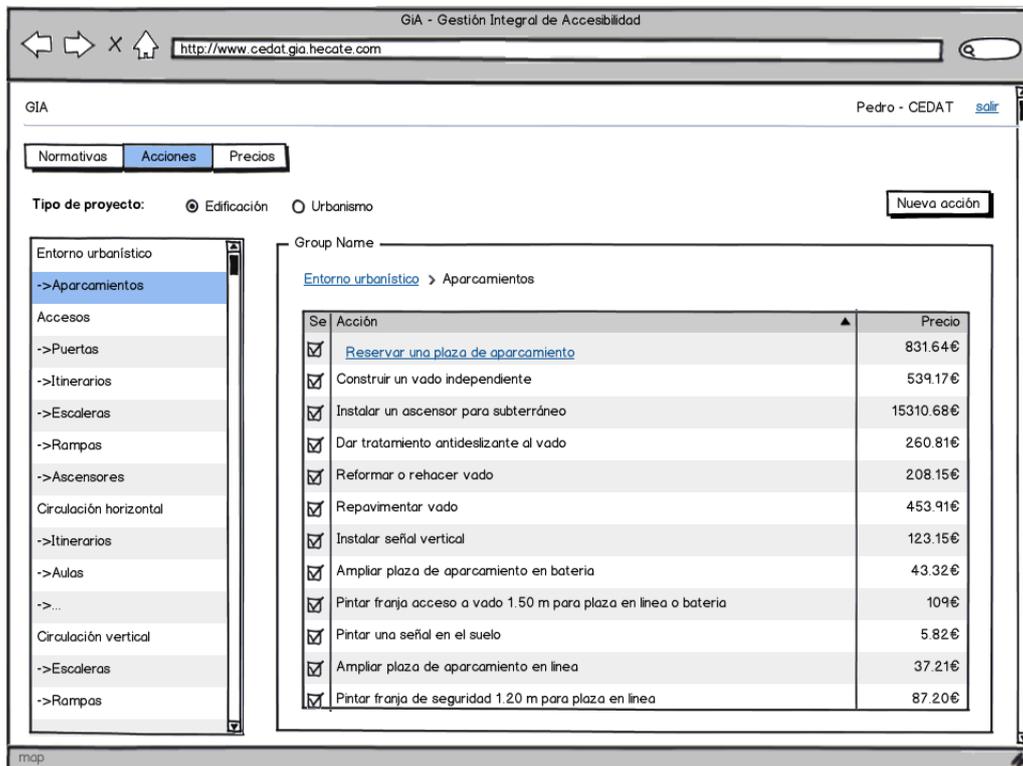


Figura 15: Listado de acciones del prototipo

[Cedat](#)

GIA
 Gestión Integral de Accesibilidad

Tipo de proyecto: Edificación Urbanismo

1.- Accesos de uso público
La rampa de acceso cumple alguna de las siguientes
 Si el acceso tiene escalera, ¿cuenta con una rampa
 Si existe un desnivel para acceder desde el espaci
2.- Itinerarios de uso público
 circulaciones horizontales
 Circulaciones verticales:
 Aparatos elevadores
 circulaciones verticales:
 ascensores

1.- Accesos de uso público > La rampa de acceso cumple alguna de las siguientes

Acción	Precio
accion #1	831.64 €
accion #2	350.81 €

© 2012 - GIA

Figura 16: Listado de acciones del aplicativo

Finalizamos este apartado mostrando las pantallas que muestran los listados de los precios existentes en el sistema.

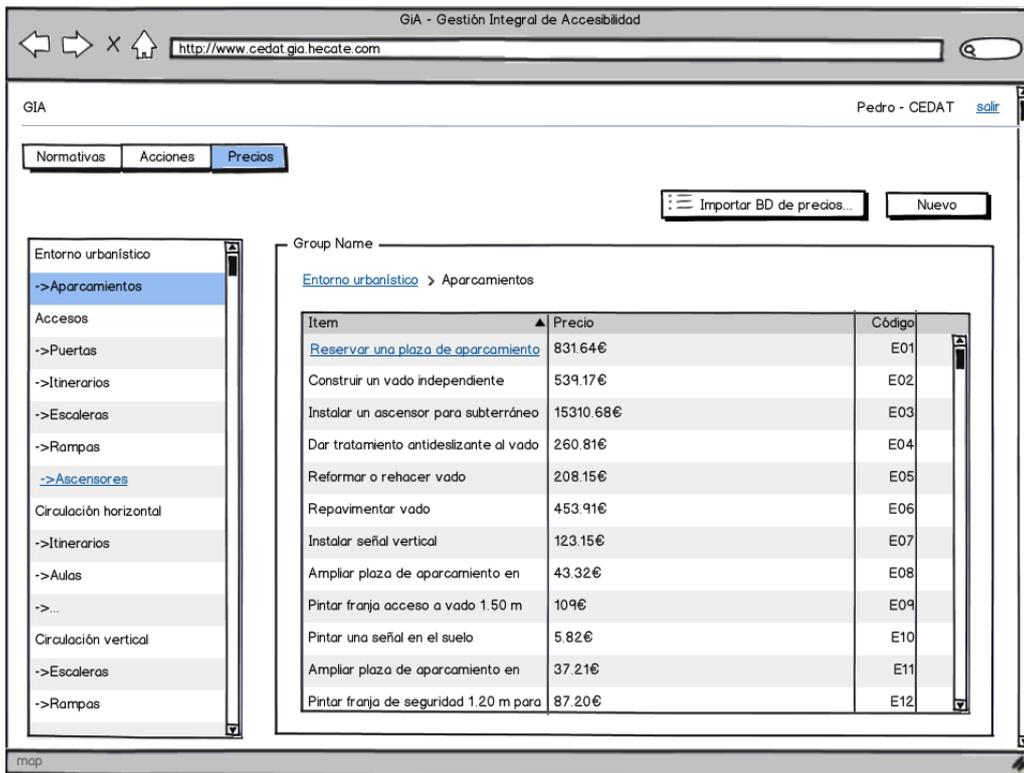


Figura 17: Listado de precios del prototipo

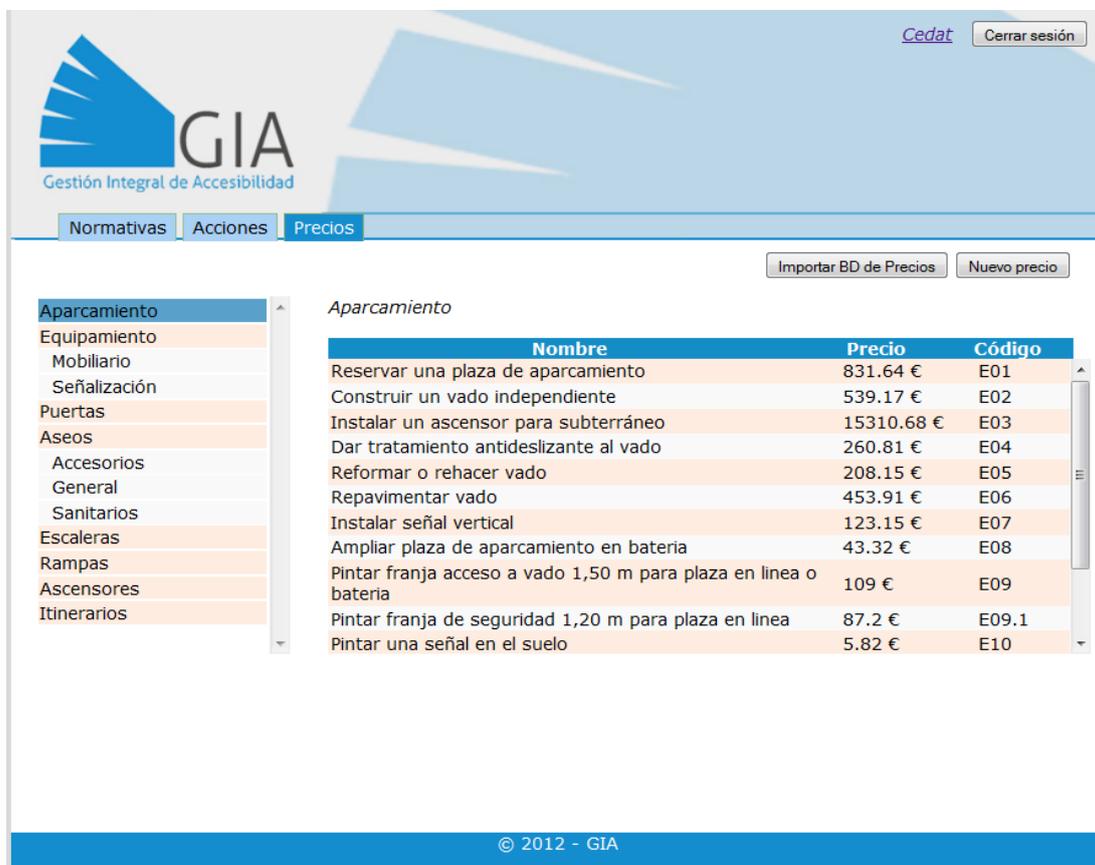


Figura 18: Listado de precios del aplicativo

6 Implementación

A continuación realizaremos¹¹ una descripción breve de las tecnologías que hemos utilizado en la implementación del prototipo, así como las herramientas utilizadas.

6.1 Tecnologías

- HTML5, siglas de HyperText Markup Language versión 5 (*“Lenguaje de marcado de hipertexto”*). Se ha utilizado en la elaboración de las páginas web.
- CSS, siglas de Cascading Style Sheets (*“hojas de estilo en cascada”*). Se trata de un lenguaje que es usado para estructurar la información en los documentos HTML, XML y XHTML.
- PHP, se trata de un lenguaje de programación interpretado diseñado originalmente para la creación de páginas web dinámicas.
- JQuery, se trata de una biblioteca de javascript que nos simplifica el trabajo en el tratamiento de eventos, animaciones, interacciones, etc., en la capa de presentación.
- JSON, acrónimo de *Javascript Object Notation*. Se trata de un formato ligero para el intercambio de datos, mediante objetos de Javascript y que no requiere el uso de XML.
- Javascript, se trata de un lenguaje de programación interpretado, que puede ser utilizado tanto desde la parte servidora como cliente. En el prototipo, se ha usado en la parte del cliente para mejorar las páginas web.
- AJAX, siglas de Asynchronous Javascript And XML (*“Javascript asíncrono y XML”*). Se trata de una técnica de desarrollo web para crear aplicaciones interactivas o RIA (*Rich Internet Applications*).

¹¹ Para la descripción de las tecnologías se ha buscado información principalmente en <http://es.wikipedia.org>

En la figura 19 podemos ver una representación gráfica de las tecnologías utilizadas separadas por su ámbito de uso (Cliente o Servidor).

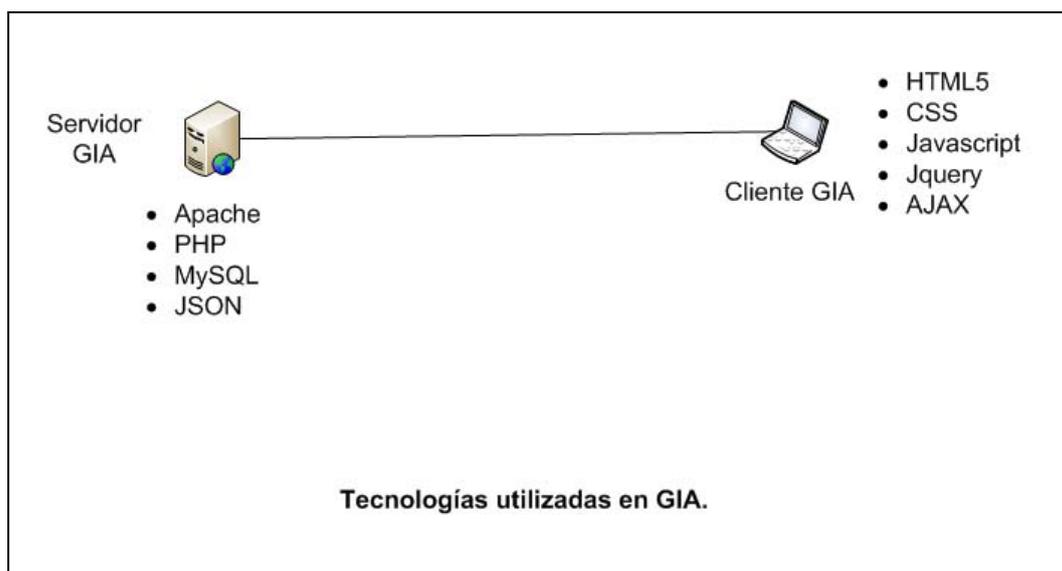


Figura 19: Tecnologías utilizadas en GIA

En conclusión, se puede resumir que las aplicaciones con las que se desarrollará la Herramienta GIA serán aplicaciones web: un lenguaje de scripting PHP, así como las tecnologías anteriormente comentadas, (Javascript, JQuery, AJAX, HTML5, JSON) para conseguir una mejor escalabilidad en una aplicación en la que el número de usuarios simultáneos puede ser muy alto. Como base de datos se utilizará el motor MySQL.

Como se comenta en el documento de instalación, tanto el servidor web como la base de datos están incluidos dentro del paquete XAMPP (plataforma de software libre). Indicamos también que para el desarrollo de la parte del cliente hemos utilizado el entorno de desarrollo Aptana Studio¹² (también gratuito), basado en Eclipse y especialmente diseñado para lenguajes de scripting con finalidad de crear aplicativos web.

6.2 Herramientas

- Aptana¹³, es un entorno de desarrollo integrado gratuito basado en Eclipse y desarrollado por Aptana Inc. Toda la herramienta se ha desarrollado con dicha plataforma.
- Xampp¹⁴, es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.
- MySQL¹⁵, es un sistema de gestión de bases de datos relacional, multihilo y multiusuario.

¹² <http://www.aptana.com/>

¹³ <http://www.aptana.com/>

¹⁴ <http://www.apachefriends.org/es/xampp.html>

¹⁵ <http://www.mysql.com/>

- Apache¹⁶, es un servidor web HTTP de código abierto, para plataformas Unix, Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

¹⁶ <http://www.apache.org/>

7 Pruebas

Durante el desarrollo de la aplicación se han ido realizando diversos tipos de pruebas.

En primer lugar, las Pruebas Unitarias. Las hemos ido realizando a medida que se iban implementando las funcionalidades de la aplicación. Dichas pruebas han consistido en verificar que cada una de las funciones implementadas han funcionado tal como se esperaba con la definición de un conjunto de casos de uso de prueba.

En segundo lugar, las Pruebas de Integración, nos ha permitido verificar que un módulo funciona correctamente, tanto individualmente como de forma colaborativa con otros módulos. Para la toma de requisitos y diseño del prototipo (en la figura inferior llamado Mockup) también hemos realizado unas pruebas “ligeras” de integración para conseguir un diseño del prototipo lo más parecido al producto final, validando los distintos módulos a desarrollar después de la reunión con el cliente.

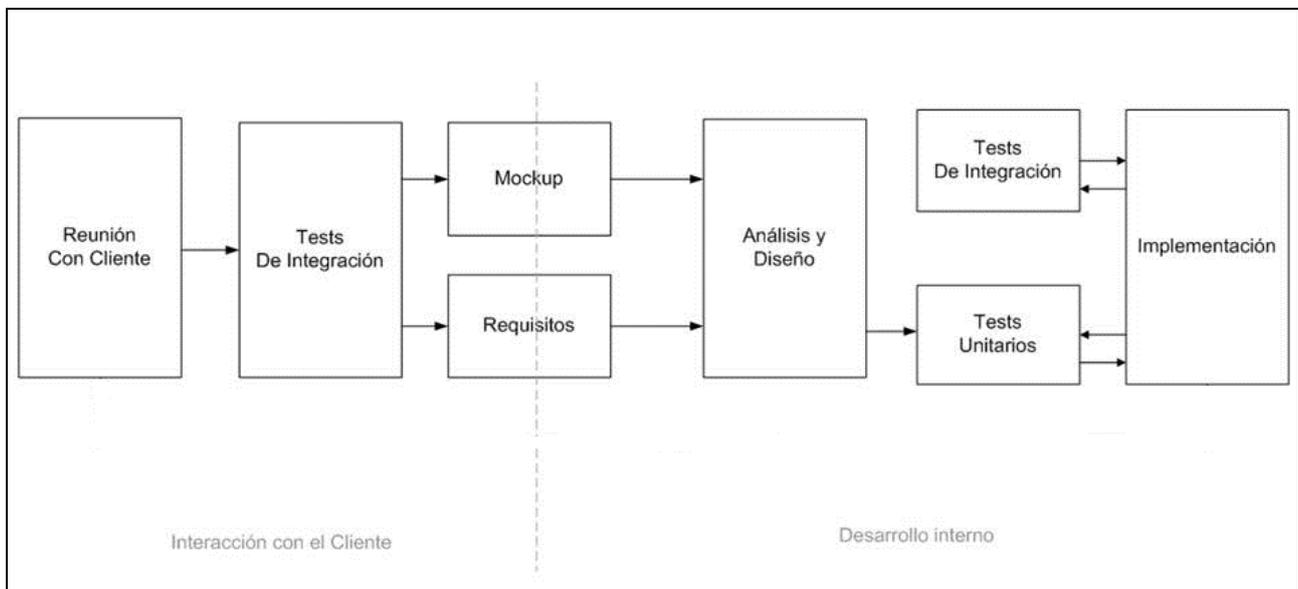


Figura 20: Esquema de desarrollo

Para finalizar, las Pruebas funcionales, nos han permitido verificar que una funcionalidad que debe de ofrecer la aplicación, se encuentra implementada y que funciona correctamente.

8 Instalación de la aplicación

Para que la aplicación funcione correctamente es necesario seguir los siguientes pasos

- a) Para empezar la puesta en marcha de la herramienta, necesitamos descargar e instalar un servidor para poder desplegar nuestra herramienta.

El servidor elegido es XAMPP¹⁷ ya que cuenta con una base de datos MySQL, un servidor web Apache y un intérprete de lenguaje de script PHP. Nos descargaremos el "installer" de este servidor desde este enlace:

<http://sourceforge.net/projects/xampp/files/XAMPP%20Windows/1.7.7/xampp-win32-1.7.7-VC9-installer.exe/download>

A continuación, realizamos la instalación del servidor avanzando a través de las pantallas de la instalación pulsando el botón siguiente hasta llegar al final.

- b) Una vez finalizada la instalación se habrá creado en la unidad C: una carpeta con nombre **xampp**, accedemos a esta carpeta y posteriormente a la carpeta **htdocs**. En este momento estaremos en la ruta C:\xampp\htdocs\ y descomprimos el fichero gia.zip (incluido en la entrega).
- c) Para poder remitir correos, debemos activar openssl en php. Para ello debemos editar el fichero de configuración **php.ini** ubicado en la ruta C:\xampp\php y añadir o descomentar las siguiente líneas:

```
extensión=php_curl.dll  
extensión=php_openssl.dll
```

- d) Ahora tenemos que activar el servidor de aplicaciones Apache y la Base de datos MySQL, para ello pulsamos Inicio → Programas → Apache Friends → XAMPP → XAMPP Control Panel y pulsamos el botón Start para los módulos Apache y MySQL.

¹⁷ <http://www.apachefriends.org/es/xampp.html>

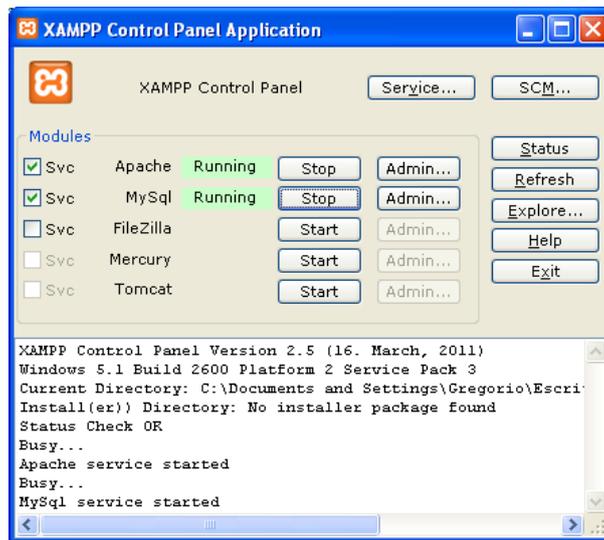


Figura 21: Panel de Control de XAMPP

En este momento, tendremos activos el servidor Apache y la base de datos MySQL.

- e) Ahora tenemos que importar la base de datos suministrada en la entrega, para ello accedemos a la base de datos MySQL instalada a través de este enlace:

<http://localhost/phpmyadmin/>

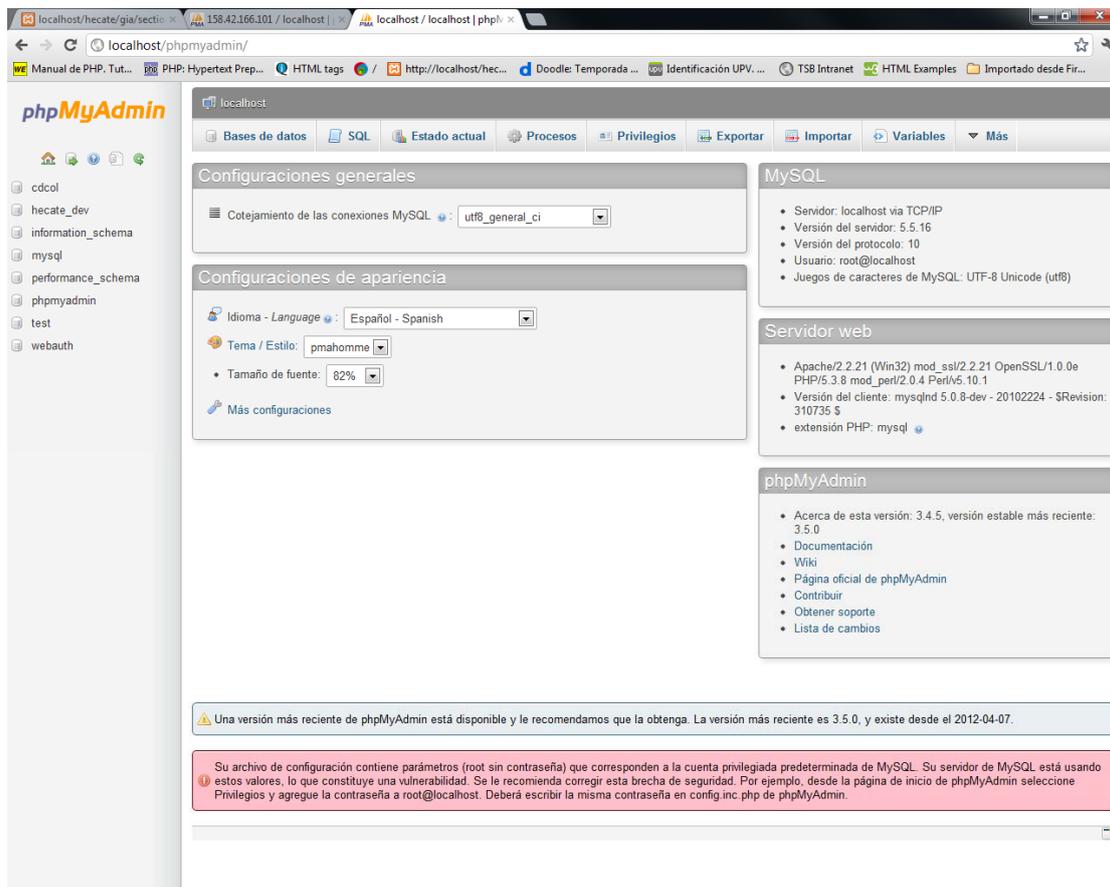


Figura 22: Consola de la base de datos MySQL en XAMPP

Y realizamos las siguientes operaciones:

- Configurar el idioma de la base de datos (combo idioma – language). Es opcional, pero será más sencillo moverse a través de la base de datos ya que por defecto se configura en alemán.
 - Crear la base de datos con nombre “bd_gia” (Pestaña Bases de datos → crear base de datos) con cotejamiento utf8_general_ci.
 - Importar la base de datos suministrada en la entrada (bd_gia.sql.zip) dentro de la base de datos que acabamos de crear (“bd_gia”). Para ello, accedemos a la base de datos bd_gia que acabamos de crear, y pulsamos la Pestaña Importar, seleccionamos el fichero bd_gia.sql.zip (adjunto en la entrega) y continuamos.
- f) Configuramos las opciones de la pantalla, con una resolución de 1280 x 1024, en el caso que sea posible. Si no, a la máxima resolución permitida por el dispositivo.
- g) En este punto ya tendremos todo el entorno de la herramienta preparado para poder ejecutarla. Para ello, ejecutamos la misma desde este enlace:

<http://localhost/gia>

Se puede acceder a la herramienta con un usuario nuevo, creándolo con la herramienta, o bien a través del siguiente usuario de pruebas:

- Usuario: accesibl@upvnet.upv.es
- Password: hecate2011gia

9 Conclusiones

A continuación procederé a enumerar las conclusiones que he podido obtener con la realización de este Proyecto Fin de Carrera (PFC).

- La realización de este Proyecto me ha permitido profundizar en el estudio de diversas tecnologías web.
- También me ha permitido ver como el uso de nuevas tecnologías tiene un coste alto, debido a su aprendizaje.
- Y como de importante es, determinar correctamente los factores de riesgos que puedan hacer peligrar un proyecto.

Además del enriquecimiento personal a nivel de conocimiento que ha conllevado este PFC me gustaría destacar el fin del mismo, poder hacer un mundo más accesible intentando salvar las barreras arquitectónicas existentes.

10 Índice de ilustraciones

Figura 1: Ciclo de vida de un PIA	6
Figura 2: Calendario de trabajo	15
Figura 3: Casos de uso	18
Figura 4: Bases de datos	43
Figura 5: Diagrama de clases	45
Figura 6: Diagrama de estados de una normativa	46
Figura 7: Subsistemas	47
Figura 8: Modelo Cliente-Servidor	49
Figura 9: Arquitectura Cliente-Servidor en 3 capas	50
Figura 10: Paradigma MVC	51
Figura 11: Login del prototipo	55
Figura 12: Login del aplicativo	56
Figura 13: Listado de normativas del prototipo	56
Figura 14: Listado de normativas del aplicativo	57
Figura 15: Listado de acciones del prototipo	57
Figura 16: Listado de acciones del aplicativo	58
Figura 17: Listado de precios del prototipo	59
Figura 18: Listado de precios del aplicativo	59
Figura 19: Tecnologías usadas en GIA	61
Figura 20: Esquema de desarrollo	63
Figura 21: Panel de control de XAMPP	65
Figura 22: Consola de la Base de Datos MySQL en XAMPP	65

11 Índice de tablas

Tabla 1: Hitos del PFC	15
Tabla 2: Roles	15
Tabla 3: Formato de fichero EXCEL proveniente de PRESTO	40

12 Bibliografía

- Thomas A. Powell (2001), Manual de Referencia HTML 4. Editorial Osborne McGraw-Hill.
- Eguíluz Pérez, Javier (2009), CSS Avanzado. Editorial www.librosweb.es
- Bear Bibeault, Yehuda Katz (2010), Jquery 1.4. Editorial Anaya Multimedia.
- Material de la asignatura *Metodología y Gestión de proyectos informáticos* de la UOC.
- Material de la asignatura *Ingeniería del Software orientada a objeto* de la UOC.
- Material de la asignatura *Ingeniería del Software de componentes y sistemas distribuidos* de la UOC.
- Este documento se ha basado, estructuralmente hablando, en la plantilla de la Memoria del alumno Miguel Ángel Pineda Cruz, suministrada por el tutor Fatos Xhafa a través del foro del PFC.
- <http://es.wikipedia.org/wiki/Cliente-servidor>
- http://es.wikipedia.org/wiki/Modelo_Vista_Controlador
- <http://validator.w3.org/check>
- <http://jsonlint.com/>
- http://docente.ucol.mx/sadanary/public_html/bd/cs.htm
- <http://fernando-cabrera.blogspot.com.es/2008/07/arquitecturas-web-vs-mvc.html>
- <http://www.w3schools.com>
- <http://jquery.com/>