



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Aplicación de gestión de citas previas y citas
de atención inmediata de la UPV: poli[Cita]

Trabajo Fin de Grado
Grado en Ingeniería Informática

Autor: Raúl Utiel García

Tutoras: Manuela Albert Albiol
María Victoria Torres Bosch

2019/2020

Resumen

Este trabajo fin de grado presenta el desarrollo de una aplicación web para solicitar y gestionar diferentes tipos de citas previas y citas de atención inmediata, para que los miembros de la comunidad universitaria y personas externas puedan solicitar citas a los distintos servicios, unidades y centros de la UPV (como Registro General, Servicio de Alumnado, escuelas y facultades, etc.). Además, la aplicación deberá disponer de un monitor de llamadas que avisará a los solicitantes de las citas de que acaban de ser llamados por un gestor para atender su cita. La aplicación también tendrá otras utilidades extras para los solicitantes de las citas, como la integración con los calendarios iCal de la UPV, y avisos de recordatorio vía correo electrónico. Para realizar su desarrollo se han utilizado paquetes de bases de datos de Oracle a través de su módulo WEB TOOLKIT.

Palabras clave: cita, solicitante, gestor, llamada, UPV, paquete, Oracle.

Abstract

This final degree project presents the development of a web application to request and manage different types of scheduled appointments and immediate service appointments, so that members from inside and outside of the university community can request different types of appointments from the different services, units and centers of the UPV (such as General Registry, Student Service, schools and faculties, etc.). In addition, the application will make use of a call monitor to notify applicants that an agent is ready to attend their appointment. The application will also have other extra utilities for appointment applicants, such as integration with UPV iCal calendars, and appointment reminders via email. For the development of such application we have used Oracle's database packages through its WEB TOOLKIT module.

Keywords: appointment, applicant, agent, call, UPV, package, Oracle.

Tabla de contenidos

1.	Introducción	9
1.1.	Motivación.....	9
1.2.	Estructura de la memoria.....	9
1.3.	Colaboraciones	10
2.	Estado del Arte	12
3.	Metodología	14
4.	Análisis	15
4.1.	Requisitos de la aplicación	15
4.1.1.	Requisitos funcionales.....	15
4.1.2.	Requisitos no funcionales	18
4.2.	Modelado del Análisis.....	19
4.3.	Identificación y análisis de las posibles soluciones	21
4.3.1.	Adaptar la aplicación actual de Cita Previa.....	21
4.3.2.	Adquisición de <i>software</i> de mercado en la nube.....	22
4.3.3.	Adquisición de <i>software</i> de mercado (servidores UPV).....	22
4.3.4.	Implantación de una herramienta de <i>software</i> libre	23
4.3.5.	Solución propuesta	23
5.	Diseño de la solución.....	25
5.1.	Arquitectura del Sistema	25
5.2.	Diseño detallado	26
5.2.1.	Diagrama de navegación.....	26
5.2.2.	Prototipo de la interfaz gráfica de usuario	32
5.2.3.	Diagrama Entidad/Relación	35
5.3.	Estimación de esfuerzo y planificación.....	38
5.4.	Tecnología utilizada.....	40
6.	Implementación de la solución.....	42
6.1.	Estándar de desarrollo utilizado.....	42
6.2.	Buenas prácticas	48
6.3.	Objetos de BBDD implementados	49
7.	Pruebas.....	53
8.	Implantación.....	55
9.	Conclusiones	58
10.	Trabajos futuros.....	61

11. Referencias.....	63
12. Anexos	64
12.1. Anexo I. Ejemplo de uso del estándar multivista.....	64
12.2. Anexo II. Ejemplo de casos de pruebas funcionales.	67

Tabla de figuras

Figura 1. Formulario de solicitud de cita previa del ayuntamiento de Utrecht.	12
Figura 2. Esquema del modelo de ciclo de vida iterativo e incremental del software.	14
Figura 3. Diagrama de casos usos UML	20
Figura 4. Diagrama de componentes UML	26
Figura 5. Diagrama de navegación del portal del solicitante.	27
Figura 6. Diagrama de navegación del módulo de solicitud de atención inmediata.	28
Figura 7. Diagrama de navegación del portal del gestor	29
Figura 8. Diagrama de navegación del módulo del coordinador con sus puntos de acceso externos.	30
Figura 9. Diagrama de navegación del módulo del administrador.	31
Figura 10. Pantalla de Selección de Tipo de Cita.	32
Figura 11. Pantalla de Confirmación de Tipo de Cita	33
Figura 12. Pantalla de Asignación de Turno.	34
Figura 13. Pantalla de Confirmación de Email.	35
Figura 14. Diagrama entidad-relación de la base de datos	36
Figura 15. Tabla de estimación de esfuerzo inicial.	39
Figura 16. Arquitectura de la tecnología PL/SQL Web Toolkit de Oracle.	40
Figura 17. Acceso al Portal de Consultas UPV desde la página web de la UPV.	43
Figura 18. Portal de Consultas UPV con vista normal accedido desde la página web de la UPV.	44
Figura 19. Acceso al Portal de Consultas UPV desde la intranet de la UPV.	44
Figura 20. Portal de Consultas UPV con vista intranet accedida desde la intranet de la UPV.	44
Figura 21. Acceso al Portal de Consultas UPV desde la página web de la UPV visto desde un teléfono móvil.	45
Figura 22. Portal de Consultas UPV con vista móvil accedido desde la página web de la UPV visto desde un teléfono móvil.	46
Figura 23. Pantalla inicial del portal del solicitante de poli[Cita] con vista normal.	47
Figura 24. Pantalla inicial del portal del solicitante de poli[Cita] con vista intranet.	47
Figura 25. Pantalla inicial del portal del solicitante de poli[Cita] con vista móvil.	48
Figura 26. Enlace de acceso a poli[Cita] desde la web de la Escuela de Doctorado.	56
Figura 27. Selección del tipo de cita de la Escuela de doctorado.	56
Figura 28. Formulario de alta de la cita Gestión Defensa de Tesis de la Escuela de Doctorado.	57

1. Introducción

En este primer capítulo de introducción se recoge la motivación que impulsó a emprender este proyecto, así como sus objetivos principales. Además, se muestra la estructura de la memoria, y un apartado de colaboraciones de todas las personas que han participado en el proyecto.

1.1. Motivación

Este trabajo fin de grado forma parte de un proyecto que surge ante la necesidad de ofrecer un servicio de atención de cita previa de **calidad** en los diferentes centros y servicios de la UPV. El servicio de cita previa se necesita dar tanto a miembros de la universidad (alumnos, PDI, y PAS), como a personas externas, p. ej. alguien que desee pedir información personalizada al Servicio de Alumnado acerca de los trámites que debe realizar para ser admitido en algún grado de la UPV. El proyecto también contempla el concepto de cita de atención inmediata, es decir, citas que el solicitante pide en el mismo momento en el que se presenta para que le atiendan. Por contra, las citas previas, como su propio nombre indican deben ser anteriores o previas, y deben solicitarse con un mínimo margen de tiempo de antelación.

Se le pone el adjetivo **calidad** a cita previa, para dejar claro que no vale cualquier método para solicitar la cita previa, como p. ej. que el alumno solicite la cita previa vía correo electrónico, y el gestor del Servicio de Alumnado le conteste por el mismo canal. Se necesita un canal común, donde todas las citas queden registradas informáticamente, para que luego los gestores que las atiendan las tengan fácilmente accesibles, y así puedan llamar de forma electrónica a los solicitantes cuando les toque atenderlos. Con cada llamada electrónica que haga el gestor, el solicitante correspondiente de la cita podrá escuchar y ver en un panel informativo que acaba de ser llamado por un gestor para que le atienda.

Con esta solución, además, se pueden obtener luego estadísticas de uso de los distintos servicios de atención tanto a miembros de la universidad como a externos.

1.2. Estructura de la memoria

Con el fin de facilitar la comprensión al lector, esta memoria se estructura en 10 capítulos principales que se comentan a continuación:

- El *capítulo uno* corresponde a la introducción, donde se explica los aspectos introductorios de la memoria, como son la motivación, objetivos, estructura y colaboraciones acerca del trabajo.
- El *capítulo dos* corresponde al estado del arte, y en él se hace un estudio global acerca de las soluciones ya existentes para resolver problemas similares al que se plantea en este proyecto.
- En el *capítulo tres* se explica la metodología utilizada en el ciclo de vida del *software* a desarrollar.
- En el *capítulo cuatro* se detalla el análisis de la aplicación a desarrollar, el cual incluye la definición de requisitos, riesgos a considerar, solución de la propuesta, y la planificación del trabajo a hacer.

- El *capítulo cinco* muestra el diseño que tendrá la aplicación, dividiéndose éste en dos secciones según su nivel de abstracción: la primera más genérica con la arquitectura general del sistema, y la segunda con el diseño detallado.
- En el *capítulo seis* se explican algunos aspectos de la implementación de la aplicación llevada a cabo, tales como el estándar de desarrollo utilizado, buenas prácticas seguidas, y módulos *software* implementados.
- En el *capítulo siete* se comentan los diferentes tipos de pruebas practicadas a lo largo del proceso de desarrollo de la solución.
- El *capítulo ocho* explica el plan de implantación previsto y llevado finalmente a cabo.
- En el *capítulo nueve* se plasman las conclusiones extraídas tras la realización del trabajo final de grado, analizando si se han conseguido o no los objetivos iniciales, así como los problemas encontrados y las soluciones aplicadas a estos.
- En el *capítulo diez* se recoge una serie de mejoras para ampliar la funcionalidad de la aplicación.

1.3. Colaboraciones

El proyecto del que forma parte este trabajo fin de grado ha sido ejecutado por un equipo de trabajo heterogéneo del ASIC¹ del que formo parte. Es importante recalcar esto porque a partir de ahora en el documento se hará mención del término “proyecto” para hacer referencia al trabajo que ha sido realizado por todo el equipo en su conjunto. La siguiente tabla muestra la relación de participantes y sus funciones en el proyecto.

Participante	Rol	Tareas realizadas	Módulos implicados ²
Gustavo Pérez	Analista de Aplicaciones	Captura de requisitos. Planificación del proyecto Diseño de la solución <i>software</i> . Pruebas. Implantación	Todos (salvo <i>app</i>)
Víctor Braquehais	Analista de Aplicaciones	Captura de requisitos. Planificación del proyecto.	Todos
Raúl Utiel	Analista/Programador de Aplicaciones	Diseño e implementación de la solución <i>software</i> . Pruebas. Implantación.	Todos (salvo las modificaciones de la aplicación actual de cita previa ³ , y <i>app</i>)
Ignacio Carratalá	Analista de Aplicaciones	Diseño de la solución. Pruebas.	Modificaciones de la aplicación actual de cita previa

¹ ASIC son las siglas del Área de Sistemas de Información y Comunicaciones de la UPV.

² Los módulos se detallan más adelante en el documento en la sección 5.1 Arquitectura del Sistema.

³ Como se explica más adelante en el documento la solución propuesta fue [adaptar la aplicación actual de cita previa](#). Concretamente en el apartado [6.3 Objetos de BBDD implementados](#) se detallan los objetos de base de datos que fueron modificados relativos a la aplicación actual de cita previa.

Participantes empresa externa ODEC	Programadores	Implementación. Pruebas	Modificaciones de la aplicación actual de cita previa
Miguel Ángel Valero	Analista de Sistemas-Redes	Diseño e implementación de la solución <i>hardware</i> y de integración.	Solicitud de atención inmediata
José Luis Sepúlveda	Analista de Sistemas-Redes	Diseño e implementación de la solución <i>hardware</i> y de integración.	Panel informativo
José Ricardo Díaz Cano	Usuario que participa en la definición de requisitos.	Entrevistas para la definición de requisitos. Pruebas de aceptación.	Todos
María Isabel Carda Batalla	Usuario que participa en la definición de requisitos.	Entrevistas para la definición de requisitos. Pruebas de aceptación.	Todos

Señalar también, que en las entrevistas para la toma de requisitos y pruebas de aceptación también han participado en menor medida un conjunto de usuarios que representan algunas de las unidades UPV, principalmente del Servicio de Alumnado, de la Facultad de Bellas Artes, y del Registro General.

2. Estado del Arte

En la actualidad existen diferentes casos y ejemplos de aplicaciones de cita previa (aplicación online de cita previa de atención primaria¹, del SERVEF², etc.). Una característica común a la mayoría de estas aplicaciones es que comparten el mismo tipo de componentes que las forman: un portal para solicitar cita, otro portal de configuración y gestión de citas, y en ocasiones también un panel informativo donde se muestran las llamadas a los solicitantes cuando los gestores los llaman. La figura 1 muestra el formulario principal de solicitud de cita previa de Orchestra, una solución propietaria desarrollada por Qmatic que es utilizada por diversos ayuntamientos, como el de Elche, Burgos, o el de Utrecht (Países Bajos). En particular, esta figura muestra la implantación realizada por el ayuntamiento de Utrecht³. Como se puede observar, este formulario dispone de los habituales componentes de selección de fecha y hora de citas disponibles, en este caso para el tipo de cita Emigración previamente seleccionada.

Concertar una cita
Emigración (5 personas)

Fecha y hora (paso 3/5)
Elija una fecha en el calendario a continuación.
Luego elige una hora.

Fechas disponibles:

Abril 2020 »»

ma	Mar	Mié.	hacer	Vie.	Sáb.	entonces
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18 años	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

Tiempos disponibles:

- 14:10
- 14:20
- 2:30 p.m.
- 14:40
- 14:50
- 3:00 p.m.
- 15:10
- 15:20
- 3:30 p.m.
- 15:40
- 15:50
- 4:00 p.m.
- 16:10

[ANTERIOR](#) [SIGUIENTE](#) [PARA](#)

Figura 1. Formulario de solicitud de cita previa del ayuntamiento de Utrecht.

A pesar de que estas soluciones ofrecen una funcionalidad similar, su desarrollo varía dando lugar bien a desarrollos propios llevados a cabo por la propia organización, o a soluciones adquiridas por la organización. Si se dispone de la capacidad y tiempo de desarrollo, y se quiere una solución que se ajuste a todos los requisitos deseados y que sea totalmente personalizable e integrable con el resto de sistemas de la organización, el desarrollo propio de la solución es la

¹ http://www.san.gva.es/cita_previa/citaprevia_cas.html

² <https://sede.sepe.gob.es/citaprevia/solicitudCitaPrevia.do>

³ Enlace URL al formulario: <https://pki.utrecht.nl/onlinebooking/date-and-time>

mejor opción. Sino, podemos valorar adquirir una solución. Estas soluciones adquiridas, pueden ser soluciones de *software* propietario [1] o soluciones de *software* libre [2].

Son soluciones de *software* propietario aquellas en las que no se tiene acceso a su código fuente, y habitualmente requieren el pago de una licencia por su uso. En general, estas soluciones son una buena opción si se quiere adquirir una solución robusta y no se dispone del tiempo o de la capacidad para poder desarrollarla. Por el contrario, tienen como desventajas que suelen implicar un importante desembolso económico, que no siempre responden al cien por cien a los requisitos deseados y necesitan desarrollos para personalizarlas, y que a menudo presentan problemas de integración con el resto de sistemas de la organización. En particular, la solución anteriormente mostrada (ver Figura 1) corresponde con el *software* propietario Orchestra de la empresa Qmatic, la cual tiene un coste elevado (109.672,8€ para la UPV¹). Es una solución de implantación en los servidores de la organización que dispone de un portal de solicitud de cita muy sencillo, y que los clientes pueden desarrollarse a medida con un API que proporciona el producto con el fin de poder ajustarlo a sus propios estándares visuales. Posee el resto de componentes fundamentales de una aplicación de cita previa, un portal de gestión y configuración de citas, que se integra con un panel informativo de llamadas.

Respecto a las soluciones de *software* libre, son soluciones de código abierto y que habitualmente suelen ser distribuidas de forma gratuita. Estas soluciones también son una buena opción si no se dispone del tiempo o de la capacidad para poder desarrollar la solución. Salvo por el tema económico, suelen presentar las mismas desventajas que cualquier solución adquirida: necesidades de personalización y problemas de integración con el resto de sistemas.

Por último, para las soluciones adquiridas, también podemos distinguir entre las soluciones que se implantan en los servidores de la propia organización, y las que se ofrecen como servicios por proveedores (*Software* como un Servicio [3]). Éstas últimas, están alojadas en los servidores de los proveedores y se consumen por el cliente bajo demanda. Son las llamadas soluciones en la nube. La principal ventaja de estas soluciones es que el cliente se despreocupa de la instalación, soporte, mantenimiento, y de los aspectos de seguridad relacionados con la aplicación. Por el contrario, el cliente necesita confiar en la disponibilidad del servicio, y en la confidencialidad y seguridad de sus datos, que le ofrece el proveedor.

¹ Presupuesto obtenido en marzo de 2017 que incluye el coste de implantación y el coste de licencias por nº de gestores concurrentes.

3. Metodología

En este capítulo se explica la metodología seguida en el desarrollo del proyecto. Primero, cabe señalar, que el proceso de desarrollo sigue las típicas grandes etapas de desarrollo de un proyecto *software*: análisis, diseño, implementación, pruebas, e implantación.

Estas etapas se han ejecutado siguiendo un ciclo de vida iterativo e incremental, es decir, se ha comenzado con el análisis, diseño e implementación de los requisitos básicos del sistema, e iterativamente se han seguido obteniendo versiones del producto *software* con funcionalidades adicionales. Siguiendo este modelo de ciclo de vida, la etapa de análisis de la segunda y sucesivas iteraciones se ha basado en el producto *software* obtenido en la iteración anterior que el cliente ha podido ver y probar, y a partir del cual se han requerido en algunas ocasiones nuevas funcionalidades.

En la figura 2 podemos apreciar un esquema de cómo funciona el ciclo de vida iterativo e incremental del *software* con sus diferentes iteraciones.

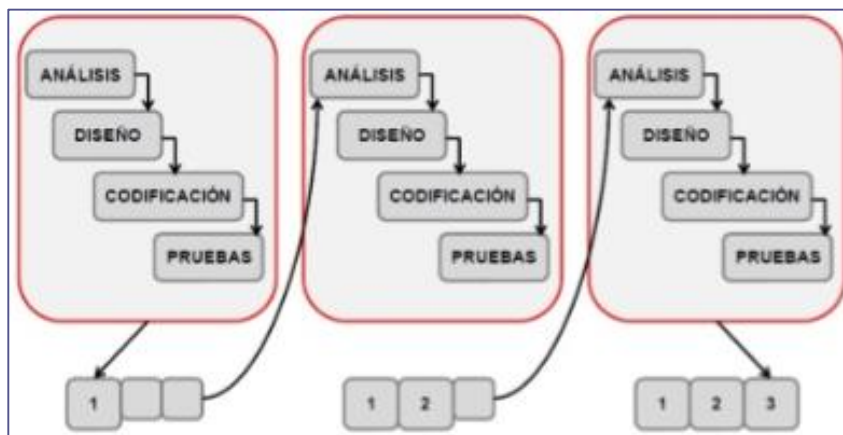


Figura 2. Esquema del modelo de ciclo de vida iterativo e incremental del *software*.

Una vez aclarado el modelo de ciclo de vida seguido, ahora podemos detallar la metodología *software* empleada. Se puede decir que en este proyecto se han aplicado prácticas de la filosofía ágil de desarrollo de *software* que encajan con el ciclo de vida de desarrollo iterativo e incremental utilizado. Las metodologías ágiles, se caracterizan por realizar entregas iterativas de “*software* que funciona” partiendo de la premisa de que los requisitos son cambiantes y evolucionan a lo largo del proyecto, por lo que es mejor ir haciendo pequeñas aproximaciones del “*software* final”, en vez de tratar de definir todos los requisitos al principio, y después tener que invertir grandes esfuerzos en adaptar cambios surgidos en estos.

Aunque no se ha seguido ninguna metodología ágil específica, como lo son Scrum, XP, Kanban, etc., sí se han aplicado los principios comentados en los que se basan las metodologías ágiles. Se han aplicado varias iteraciones en las fases de análisis, diseño, implementación y pruebas, hasta obtener el producto *software* final.

4. Análisis

En este capítulo de análisis, se identifican los requisitos, tanto funcionales como no funcionales, de la aplicación a desarrollar. A partir de los requisitos identificados, ya se puede hacer una modelización del análisis, es decir, se extraen las características esenciales que va a tener el sistema en un modelo representado mediante diagramas UML. Por último, se estudian posibles soluciones tecnológicas, y en base al resultado se selecciona la más adecuada.

4.1. Requisitos de la aplicación

Los requisitos de la aplicación han sido extraídos usando las técnicas tradicionales de entrevistas y reuniones con los usuarios que participan en la definición de requisitos de la aplicación. Estos requisitos los podemos agrupar por tipo: funcionales y no funcionales.

4.1.1. Requisitos funcionales

Los requisitos funcionales, como su propio nombre indica, son aquellos que definen la funcionalidad que va a ofrecer el *software* al usuario. A continuación, se muestra una tabla con los requisitos funcionales de la aplicación, agrupados por ámbito de aplicación para mayor claridad.

Requisitos solicitud de cita previa	
Req. #	Descripción
SolCP1	Debe permitir elegir la unidad en la que se solicita la cita y el tipo de servicio que se requiere.
SolCP2	Debe permitir elegir el día/hora dentro del calendario de disponibilidad de la unidad.
SolCP3	Debe permitir adjuntar un archivo en la solicitud de cita previa si el tipo de cita previa ha sido configurado de esa forma.
SolCP4	El alta de la solicitud de cita previa debe asignar un identificador de la cita que será generado a partir del nombre y apellidos del solicitante.
Requisitos solicitud de atención inmediata	
Req. #	Descripción
So1AI1	La solicitud de atención inmediata permitirá la solicitud de la primera cita disponible para el mismo día.
So1AI2	La solicitud de atención inmediata se hará <i>in situ</i> desde terminales táctiles instaladas en aquellas unidades que tengan esta modalidad de cita.
So1AI3	Posibilidad de lectura de los datos personales del solicitante en el proceso de solicitud mediante lectura de su carnet universitario con un lector de tarjetas.
Requisitos notificaciones al solicitante	
Req. #	Descripción

Not1	Las citas previas solicitadas deben notificarse a sus solicitantes por <i>email</i> .
Not2	Debe existir la posibilidad de enviar notificación por <i>email</i> cuando el gestor elimina la cita.
Not3	Debe enviarse recordatorios de la citas pendientes a los solicitantes por <i>email</i> .
Requisitos gestión de cita previa (solicitante)	
Req. #	Descripción
GesSol1	El solicitante debe poder consultar sus citas previas desde un mismo portal independientemente del tipo de cita de que se trate y de la unidad a la que pertenezca. En la consulta se deberá poder filtrar las citas solicitadas vigentes o pendientes de atender.
GesSol2	El solicitante puede anular o eliminar sus citas previas con fecha de atención futura.
Requisitos gestión de citas (gestor)	
Req. #	Descripción
Ges1	El portal de gestión de citas debe poder agrupar las citas de dos modos posibles: <ul style="list-style-type: none"> • Por unidad/es a las que pertenece el gestor, y tipos de citas de la unidad seleccionada. Para un tipo de cita determinado debe permitir el filtrado de citas por diversos campos de la misma: fecha de la cita, nombre del solicitante, si fue o no atendida, etc. • Por citas del día de la unidad a la que pertenece el gestor. En este caso las nuevas citas entrantes deben de aparecer automáticamente en la pantalla sin necesidad de consulta o refresco de la pantalla.
Ges2	Los gestores deben poder asignarse citas cuyos tipos de citas sean de su unidad.
Ges3	Los gestores de una unidad no podrán ver las citas de otras unidades.
Ges4	Los gestores deben poder anular o eliminar citas independientemente de su fecha.
Ges5	Para poder atender las citas el gestor debe asignarse un espacio de atención de la unidad a la que pertenece. Este espacio de atención es el que se mostrará en el panel informativo de llamadas cuando el gestor llame a los solicitantes que va a atender, para que así estos sepan dónde dirigirse.
Ges6	Los gestores deben poder llamar a los solicitantes a la hora de atender sus citas, de forma que cada llamada que haga un gestor a un solicitante debe visualizarse en el panel informativo de llamadas. Igualmente deben poder consultar las últimas llamadas realizadas.
Ges7	Los gestores deben poder registrar la asistencia o no asistencia del solicitante a su cita, es decir, marcar la cita como atendida o no atendida.
Ges8	Debe existir la posibilidad de que los gestores pongan observaciones y adjunten un archivo en la solicitud de cita.
Ges9	Debe existir un rol de coordinador, que además de poder realizar todas las acciones propias de un gestor, pueda realizar las siguientes acciones de configuración sobre los tipos citas de una unidad: <ul style="list-style-type: none"> • Configurar su capacidad, tiempo de atención, y disponibilidad horaria de atención.

	<ul style="list-style-type: none"> • Otorgar permisos de gestión a miembros de su unidad (convertirlos en gestores).
Requisitos configuración de citas	
Conf1	<p>Cada tipo de cita tendrá los siguientes parámetros configurables:</p> <ul style="list-style-type: none"> • Nombre y Descripción. • Diferentes rangos horarios de atención por día de la semana. • Calendario con los días no hábiles de atención. • Gestores que la pueden atender. • Tiempo de atención. • Cupo máximo, que será el máximo nº de gestores de la unidad atendiendo simultáneamente citas de ese tipo. • Cupo máximo por solicitante. Número máximo de citas de un tipo determinado que puede solicitar un solicitante sin tener en cuenta las citas vencidas (con fecha anterior a la actual). • Capacidad de citas de atención inmediata. Las citas de atención inmediata podrán configurarse con tres tipos de capacidad: <ul style="list-style-type: none"> ○ Capacidad ilimitada, con la que se atiende a todo el mundo que llega. ○ Capacidad limitada, con la que se atiende a un nº máximo de personas al día en un único rango horario definido. ○ Capacidad limitada por horas, con la que se atiende a un nº máximo de personas por hora.
Conf2	Debe existir la posibilidad de añadir campos personalizados para un tipo de cita en su formulario web de solicitud de cita.
Conf3	Debe existir la posibilidad de mantenimiento (altas y bajas) de los espacios de atención de la unidad por parte del coordinador del tipo de cita.
Conf4	Debe existir la posibilidad de mantenimiento (altas y bajas) de los videos multimedia del panel informativo de llamadas por parte del coordinador del tipo de cita.
Conf5	Debe existir la posibilidad de mantenimiento (altas y bajas) de los anuncios del panel informativo de llamadas por parte del coordinador del tipo de cita.
Requisitos panel informativo de llamadas	
Req. #	Descripción
Panel1	Debe visualizar las llamadas que los gestores de una unidad hacen a los solicitantes para atender sus citas. Cada llamada nueva que se visualice en el monitor debe mostrar el identificador de la cita, y el espacio donde se va a atender.
Panel2	Debe adaptarse correctamente las resoluciones específicas de los dispositivos (HD / FHD / 4K).
Panel3	Las nuevas llamadas que se visualizan deben reproducir un aviso sonoro, y destacarse visualmente de las anteriormente realizadas.
Panel4	Debe poder visualizar varios anuncios de la UPV en bucle.
Panel5	Debe tener la opción de poder visualizar varios videos multimedia en bucle.

Es importante aclarar, que los gestores gestionan “citas”, es decir, tanto citas previas, como citas de atención inmediata, solicitadas ambas tipologías de citas por los solicitantes. Igualmente, los apartados de requisitos de configuración de citas y del panel informativo de llamadas hacen referencia al término “cita” por lo que se refieren también a las dos tipologías de citas.

4.1.2. Requisitos no funcionales

Los requisitos no funcionales, no se refieren a las funciones específicas que el *software* debe realizar, sino que se refieren a propiedades o cualidades que el producto *software* debe tener. En general son restricciones o condiciones que el cliente impone al *software* que necesita. A continuación, se muestra una tabla con los requisitos no funcionales que debe tener la aplicación.

Req. #	Descripción
ReqNF1	<p>Las solicitudes de citas deben estar disponibles tanto para las personas miembros o vinculadas con la UPV (alumnos, PAS, PDI, etc.) como para personas externas a la UPV (cualquier persona que no pertenezca al grupo anterior):</p> <ul style="list-style-type: none"> • El acceso de las personas vinculadas a la UPV debe realizarse con las credenciales de la intranet de la UPV, ya sea autenticándose desde la misma intranet, o a través de una aplicación externa mediante algún mecanismo de SSO¹. • Las personas no vinculadas a la UPV no necesitaran ningún tipo de registro previo ni credenciales para poder acceder. Cuando den de alta una cita previa se les proporcionara un código de seguimiento para posteriormente poder consultarla si lo desean.
ReqNF2	<p>La aplicación debe ser plurilingüe:</p> <ul style="list-style-type: none"> • Castellano, valenciano e inglés para los solicitantes. • Castellano y valenciano para los gestores.
ReqNF3	<p>Acceso desde dispositivos móviles. Las pantallas web de la aplicación deben ser <i>responsive</i> y adaptarse a diferentes resoluciones, pudiendo visualizarse correctamente desde dispositivos móviles.</p>
ReqNF4	<p>Debe existir una <i>app</i> móvil para la solicitud de citas previas de los solicitantes.</p>
ReqNF5	<p>La aplicación de cita previa deberá integrarse tanto con BBDD, como con aplicaciones de la UPV.</p>
ReqNF6	<p>Integración con calendarios iCal² UPV.</p>
ReqNF7	<p>Se debe poder obtener informes de tiempos de servicio, volumen de citas, etc.</p>

¹ SSO son las siglas de *Single Sign-On*, que es un procedimiento de autenticación que permite a un usuario acceder a varios sistemas con una sola autenticación.

² iCal es la abreviatura de iCalendar, que es un estándar para el intercambio de información de calendarios.

4.2. Modelado del Análisis

Una vez identificados todos los requisitos, ya podemos comenzar a construir los modelos del sistema. En concreto, se construye el diagrama UML de casos de uso [4] donde podemos ver fácilmente las acciones que realiza el sistema y su relación con los actores que participan en las mismas. En la figura 3 podemos apreciar que el sistema queda dividido en tres subsistemas, los relativos al solicitante, al gestor, y al coordinador. También podemos ver como el actor coordinador es una extensión del actor gestor, es decir, que el coordinador podrá realizar sus acciones propias como coordinador más las acciones propias de un gestor.

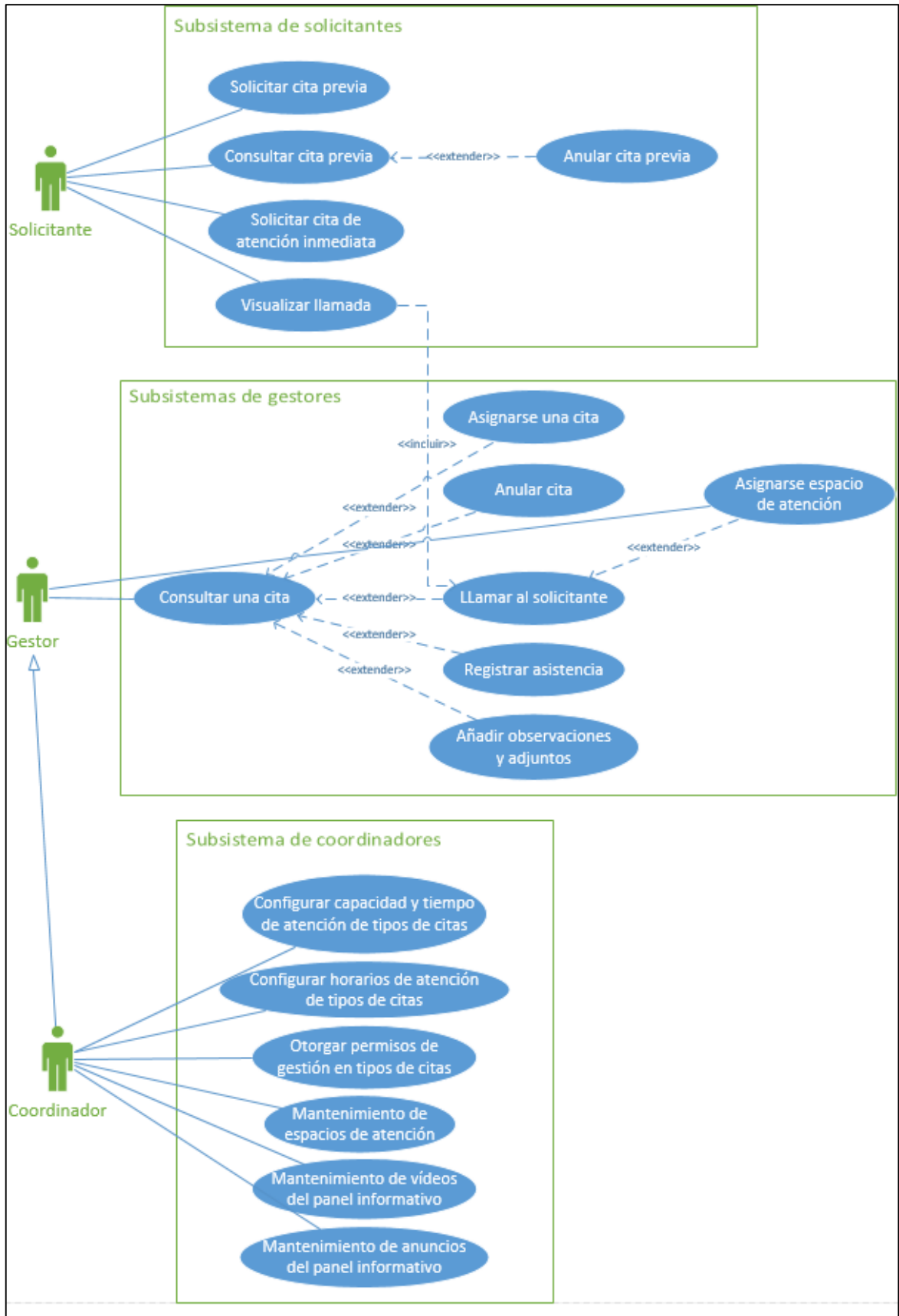


Figura 3. Diagrama de casos usos UML

Existen otras acciones relacionadas con la configuración de la aplicación que no son realizadas por usuarios finales de la aplicación (gestores y coordinadores), sino por los informáticos administradores de la misma. Por ello, no han sido incluidas explícitamente en el diagrama de casos de uso anterior. Estas acciones son:

- Crear los formularios de alta para nuevos tipos de citas.
- Configurar los campos de los formularios de alta de los diferentes tipos de citas.

4.3. Identificación y análisis de las posibles soluciones

Una vez analizado el problema, e identificadas las funcionalidades y características que va a tener nuestra aplicación, toca estudiar las posibles soluciones para abordarlo y resolverlo. A continuación, detallamos las diferentes soluciones que se exploraron para abordar el proyecto.

4.3.1. Adaptar la aplicación actual de Cita Previa

La UPV cuenta ya con una aplicación de cita previa que es utilizada por varios servicios de la UPV como son el:

- Servicio Médico
- Servicio de Enfermería
- Asesoramiento lingüístico a los usuarios del CAV
- Gabinete de Orientación Psicopedagógico Universitario – ICE
- Cita previa justificación de Ayudas Públicas Colaborativas
- Cita previa justificación de Ayudas Públicas Nacionales
- Servicio Integrado de Empleo
- Cita previa servicio de extranjería
- Otros

Esta aplicación de cita previa está construida a base de formularios web configurables, que hace que sean altamente personalizables a los diferentes tipos de citas. Por contra, la funcionalidad de gestión de las citas previas es muy básica, y carece de un panel informativo que muestre las llamadas a los solicitantes cuando los gestores los van a atender. En este sentido, habría que realizar desarrollos para adaptarla a todas las necesidades detectadas.

Estudiando la aplicación¹, se identifican las siguientes funcionalidades a desarrollar para satisfacer todos los requisitos detectados:

- Desde el punto de vista del solicitante de cita previa:
 - Estar accesible para usuarios externos de la UPV, y no sólo para personal o alumnado miembro de la UPV (ReqNF1).

¹ El estudio preliminar completo de la aplicación actual de Cita Previa está disponible en https://wiki.upv.es/confluence/display/AA/Estudio+preliminar?preview=/151781540/151781545/ESTUDIO_PREVIA_O_SV247365_CITA_PREVIA.docx (para poderse descargar el documento se necesita estar autenticado en la intranet y pedir permiso de acceso a guspeagupv@cc.upv.es)

- Notificar las citas vía *email*, y mandar un *email* recordatorio de las mismas (Not1, Not2 y Not3).
- Tener un único punto de acceso global que agrupe los tipos de citas por unidad o servicio, y no tener una pantalla de acceso por cada tipo de cita (SolCP1).
- Tener una única pantalla que permita consultar/gestionar todas las citas que tiene un solicitante de una cita, independientemente del tipo de cita que sean (SolCP1, GesSol1 y GesSol2).
- Disponer de paneles informativos para que el solicitante de la cita sepa que acaba de ser llamado por un gestor que va a atender su cita (Panel1, Panel2, Panel 3, Panel4 y Panel5).
- Desde el punto de vista del gestor de cita previa:
 - Permitir gestionar las citas, es decir, cambiarlas, anularlas, etc. (Ges1, Ges3, Ges4, Ges5, Ges7 y Ges8).
 - Poder asociar los tipos de citas a las personas que las atienden (Ges2).
 - Poder avisar a un solicitante que pasa a ser atendido, apareciendo dicha información en un panel informativo (Ges6).
 - Mantenimiento por parte del coordinador de los espacios de atención de la unidad, y anuncios y videos del panel informativo de llamadas (Conf3, Conf4, Conf5).
- Disponer de un punto de solicitud de atención inmediata, es decir, tipos de citas que se atienden en el momento (la primera disponible) in situ (SolAI1, SolAI2 y SolAI3).
- Desarrollar una *app* para solicitud y consulta/gestión de citas previas para el solicitante.

4.3.2. Adquisición de *software* de mercado en la nube

Existen muchos productos de mercado que ofrecen soluciones para llevar a cabo la reserva de cita previa. Muchos de estos productos se ofrecen a través de la nube, lo cual supone una ventaja en cuanto a la rapidez de implantación de la solución. Por el contrario, tienen como desventaja que, al ser productos estándar, casi nunca cumplen todos los requisitos necesarios, por lo que se necesita desarrollar personalizaciones o funcionalidades adicionales (siempre que el producto lo permita).

Aunque *a priori* parezca una opción viable, pocas soluciones de las revisadas cumplen la mayoría de requisitos necesarios. No ofrecen la atención de citas por espacios (Ges5), ni integran la atención de citas con un panel informativo de llamadas (Panel1). Tan solo hay una solución que sí parece viable, pero el proveedor no ha proporcionado información detallada, ni una estimación de precios (<http://www.wavetec.com/es/>). A raíz de la información existente en la web del proveedor no somos capaces de determinar si el producto podría integrar la autenticación UPV mediante algún mecanismo de SSO (ReqNF1), uno de los requisitos imprescindibles que se necesitan.

4.3.3. Adquisición de *software* de mercado (servidores UPV)

Existen múltiples soluciones de mercado específicas de cita previa de implantación en la propia organización, en este caso en los servidores del CPD de la UPV. Algunas de ellas cubren la mayoría de los requisitos necesarios, e incluso ofrecen otras funcionalidades adicionales no

contempladas en el análisis. Por contra, la parte de gestión de las citas suele ser poco usable y poco adaptable a estándares UPV. A continuación, hacemos un recorrido por las soluciones de este tipo estudiadas:

- Q-SIGE¹. Este producto cumple la mayoría de los requisitos necesarios, aunque algunos de ellos requieren de un desarrollo adicional por parte del proveedor. Parece una solución robusta, aunque el precio del *software* es elevado y puede ser motivo de descarte. El *software* de gestión de citas parece poco usable e intuitivo.
- Zinetic². Empresa de *software* que ofrece una solución de cita previa *a priori* viable en cuanto a requisitos. Tras verse una demo con el proveedor se observa que el aspecto y funcionalidad de la aplicación no se adecúa a las necesidades, por lo que queda descartada.
- Attendo³. Solución *a priori* viable en cuanto a requisitos, y que tiene por ventaja que el precio de licencia no depende del nº de centros ni gestores que la usan. Al final se ha descartado porque el proveedor no ha querido/podido realizar ninguna demo de la misma.
- Orchestra⁴. Solución robusta utilizada por organizaciones de relevancia. En cuanto a opciones de configuración, usabilidad y robustez parece la mejor solución. Por contra, es la solución más cara de todas. También hay que tener en cuenta que la solución no incluye el portal de solicitud de cita, es el propio cliente quien tiene que desarrollárselo a medida.

4.3.4. Implantación de una herramienta de *software* libre

Se ha buscado soluciones de *software* libre que pudiesen satisfacer todas las necesidades detectadas. Tras hacer un estudio de ellas se han descartado por la tecnología subyacente (que las hacen menos integrables con el resto de aplicaciones UPV al no ser la misma), o bien por carencias respecto a requisitos fundamentales como la imposibilidad de integrar la autenticación UPV mediante algún mecanismo de SSO (ReqNF1). Las soluciones estudiadas de este tipo fueron BookScheduler⁵ y las soluciones que ofrece el portal de administración electrónica de Sistema de Gestión de Colas ATENEO⁶ y de Cita previa para atención al ciudadano⁷.

4.3.5. Solución propuesta

A priori, la alternativa de implantar una solución de mercado en los servidores de la UPV parecía la más adecuada. Sin embargo, al estudiar los productos existentes, se ha visto que el coste es elevado, y que todos los requisitos necesarios son prácticamente imposibles de conseguir.

Las soluciones en la nube revisadas no se adecuan a algunos de los requisitos previamente identificados (Ges5 y Panel1). Es por ello por lo que se descarta este tipo de soluciones. También se han estudiado alternativas de *software* libre, pero han sido inviables por no satisfacer algunos

¹ Página web: <https://www.qsig.com/>

² Página web: <http://www.zinetic.es/>

³ Página web: <http://www.spainnova.com/soluciones/administracion-electronica/oficina-de-atencion-ciudadana>

⁴ Página web: <http://www.qmatic.com/es-es/soluciones/negocio/cita-previa-online/>

⁵ Página web: <http://www.bookedscheduler.com/>

⁶ Página web: <http://administracionelectronica.gob.es/ctt/ateneo/descargas#.WLPzVPK06zw>

⁷ Página web: <https://administracionelectronica.gob.es/ctt/citaprevia>

de los requisitos no funcionales identificados (ReqNF1), o por la tecnología utilizada distinta a la utilizada en el resto de aplicaciones de la UPV y que hace este tipo de soluciones menos integrable.

Adaptar la aplicación de Cita Previa UPV actual, permitiría obtener un producto que se ajustaría totalmente a las funcionalidades que se requieren, pero a costa de varios desarrollos necesarios que seguramente suponen un coste temporal claramente superior. Pese a este coste temporal adicional, esta solución ofrece como gran ventaja que es altamente integrable con otros servicios y aplicaciones de la UPV. Todo esto hace que esta sea la opción más viable en cuanto a cumplimiento de requisitos y flexibilidad de la aplicación, por lo que es la solución finalmente se decide desarrollar.

5. Diseño de la solución

Una vez realizado el análisis de la aplicación, en este capítulo se aborda la fase de diseño de la solución, es decir, cómo se ha desarrollado la solución. Para ello, el diseño se ha descompuesto en dos niveles de abstracción descritos en los siguientes dos apartados.

5.1. Arquitectura del Sistema

La arquitectura del sistema la estructura general del sistema. En ella se identifican los diferentes subsistemas o componentes esenciales de la solución. Cada subsistema es una asociación lógica de clases y operaciones interrelacionadas entre sí que responden a ciertas funcionalidades similares. A partir de los requisitos identificados y una vez analizada la solución en el capítulo anterior, se han identificado los siguientes subsistemas que va a tener la aplicación:

- **Portal del solicitante.** Portal web al que acceden los solicitantes para solicitar cita previa a los diferentes servicios y unidades de la UPV, y para posteriormente poder consultarlas y gestionarlas.
- **Módulo de solicitud de atención inmediata.** Módulo *software* que permite a los solicitantes pedir citas de atención inmediata *in situ*. Es un módulo web que se mostrará en una tableta en modo quiosco.
- **App.** App móvil para solicitar cita previa y que será una alternativa al portal de solicitud de cita previa.
- **Panel informativo de llamadas.** Módulo *software* que mostrará las llamadas que hacen los gestores a los solicitantes para atenderlos. Es un módulo web que se mostrará en una televisión en modo kiosco.
- **Portal del gestor.** Portal web donde los gestores consultarán y gestionarán las citas de las unidades a las que pertenecen. Los gestores también llamarán desde aquí a los solicitantes para atenderlos.
- **Módulo del coordinador.** Módulo que se enlaza desde el portal del gestor desde el cual los coordinadores configuran los tipos de citas de su unidad, y otorgan permisos de gestión sobre ellos a otros miembros de su unidad.
- **Módulo del administrador.** Módulo al que sólo pueden acceder los administradores de la aplicación (de perfil informático), y desde el cual se pueden crear los formularios web de alta de nuevos tipos de citas y configurar sus campos. La funcionalidad de este módulo no estaba contemplada inicialmente en los requisitos, pero fue una funcionalidad adicional incorporada al escoger la solución de adaptar la aplicación actual de cita previa, ya que esta ya tiene esta funcionalidad.

Para poder representar estos subsistemas que componen la arquitectura del sistema se muestra en la figura 4 el diagrama de componentes UML [4]. Este diagrama visualiza los diferentes componentes del sistema, y las relaciones y dependencias entre ellos.

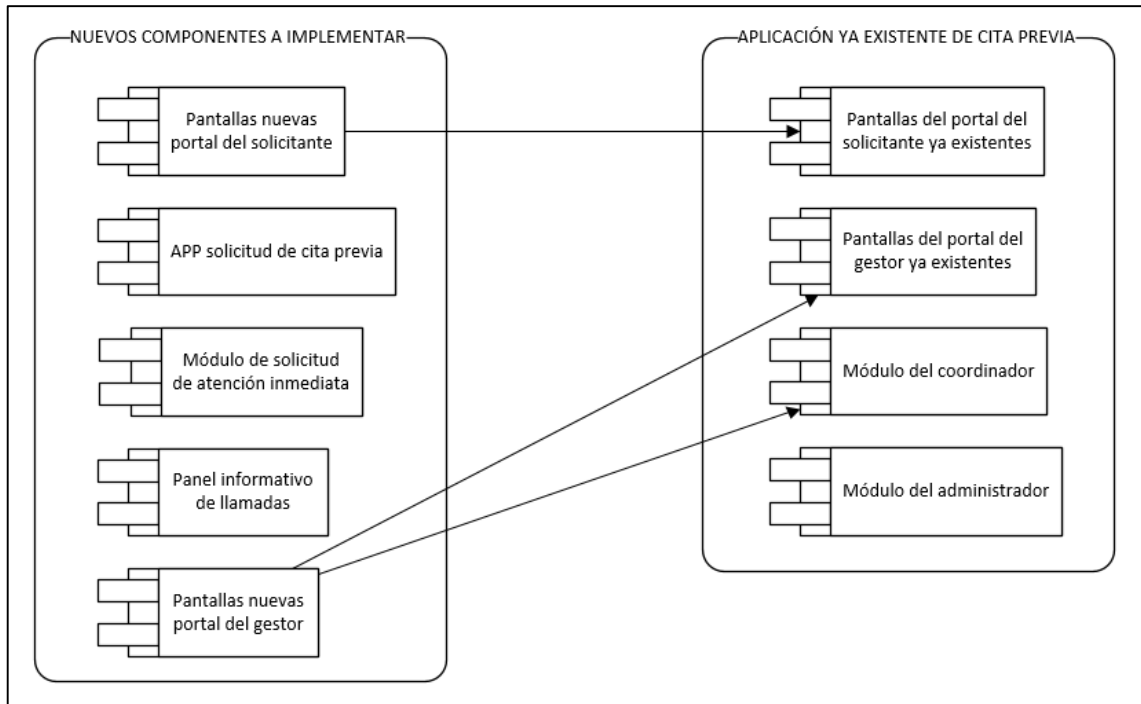


Figura 4. Diagrama de componentes UML

Como se puede apreciar en el diagrama hay dos partes bien diferenciadas, una parte con los componentes de la aplicación actual de cita previa, y otra con los nuevos componentes necesarios para poder implementar la solución. Ambas partes comparten los componentes relativos al portal del solicitante y a portal del gestor, a través de los cuales también se comunican.

5.2. Diseño detallado

Una vez representada la arquitectura del sistema, a continuación, se describe el diseño detallado de la solución. Para ello, presentamos el diagrama de navegación entre las distintas pantallas que tendrá la solución, el prototipo de la interfaz gráfica de usuario de uno de sus módulos, y el modelo de datos de la aplicación.

5.2.1. Diagrama de navegación

A partir del diagrama de componentes UML, hacemos una descomposición de cada uno de sus componentes en las distintas pantallas que tendrá la solución. Para poder ver la relación de pantallas que se obtienen de cada componente y su esquema de navegación, se muestra el diagrama de navegación para cada uno ellos, salvo para la *app*, que a día de hoy aún no se ha comenzado su desarrollo por temas de planificación. Para el panel informativo de llamadas tampoco se muestra un diagrama de navegación, ya que tan solo tiene una pantalla.

En la figura 5 se muestra el diagrama de navegación para el portal del solicitante. En el mismo, se puede apreciar la lógica de acceso para personas miembros o vinculadas a la UPV y para las personas externas a la UPV.

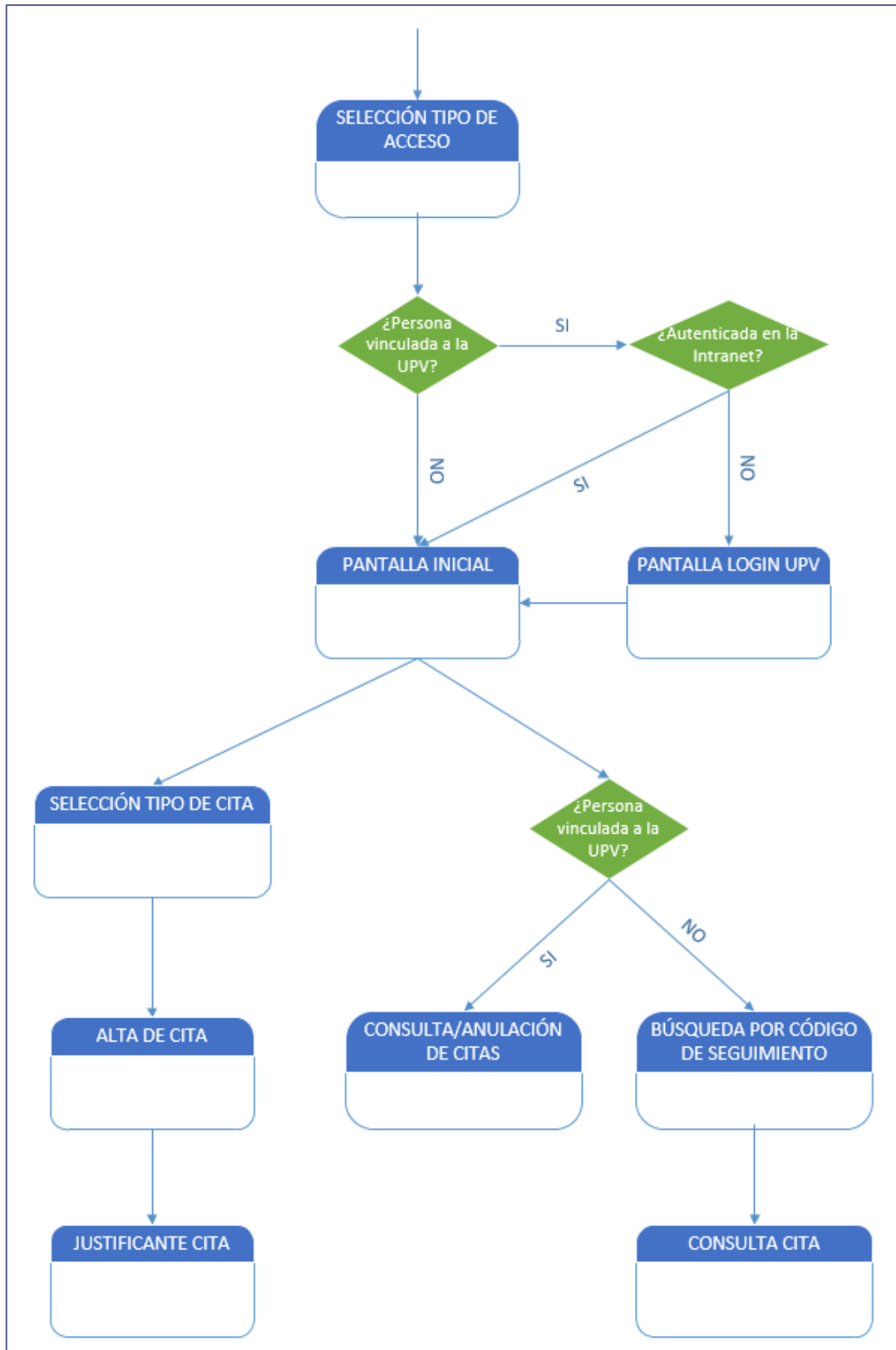


Figura 5. Diagrama de navegación del portal del solicitante.

En la figura 6 se muestra el diagrama de navegación para el módulo de solicitud de atención inmediata. Se puede el flujo de navegación en función de si el solicitante pasa su carnet universitario por el lector de tarjetas para la lectura automática de sus datos, o si los introduce manualmente.

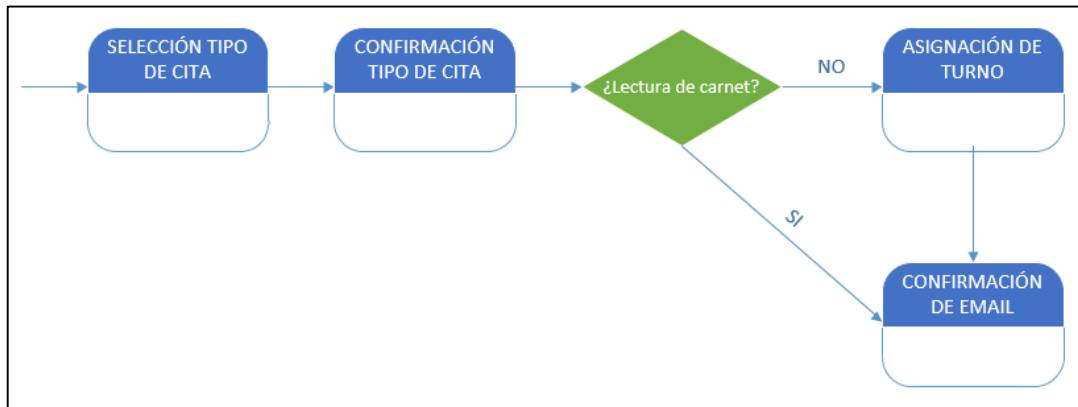


Figura 6. Diagrama de navegación del módulo de solicitud de atención inmediata.

En la figura 7 se muestra el diagrama de navegación para el portal del gestor. Se puede apreciar en el mismo la navegación a la pantalla Mi Espacio. Esta es la pantalla desde la cual el gestor se asigna un espacio de atención de la unidad a la que pertenece, condición necesaria para poder llamar a los solicitantes que va a atender (Ges5). A esta pantalla accede el gestor al inicio de su jornada desde la pantalla Selección Tipo Cita para configurarse el espacio de atención desde el que atenderá las citas. En caso de que se le olvide y no lo haga, cuando intente llamar a un solicitante desde la pantalla de Gestionar Cita, se le abrirá automáticamente antes para que así siempre llame al solicitante teniendo un espacio de atención asignado.

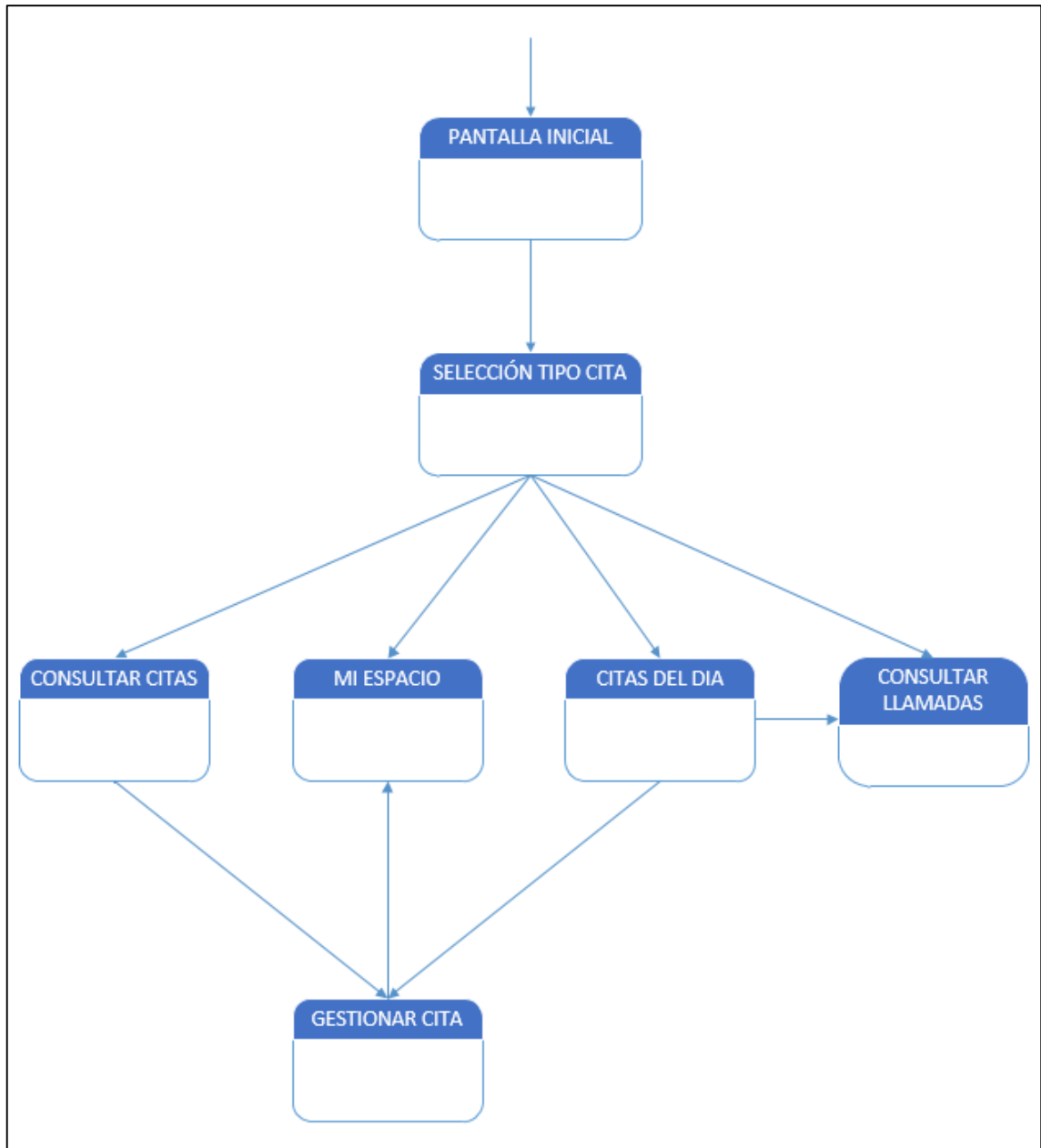


Figura 7. Diagrama de navegación del portal del gestor

En la figura 8 se muestra el diagrama de navegación del módulo del coordinador relacionado con la configuración de los tipos de citas. Se ha querido destacar los dos puntos de acceso externos al módulo, en función de a qué pantallas del mismo se va a acceder: uno desde el portal del gestor, y otro desde la intranet del gestor.

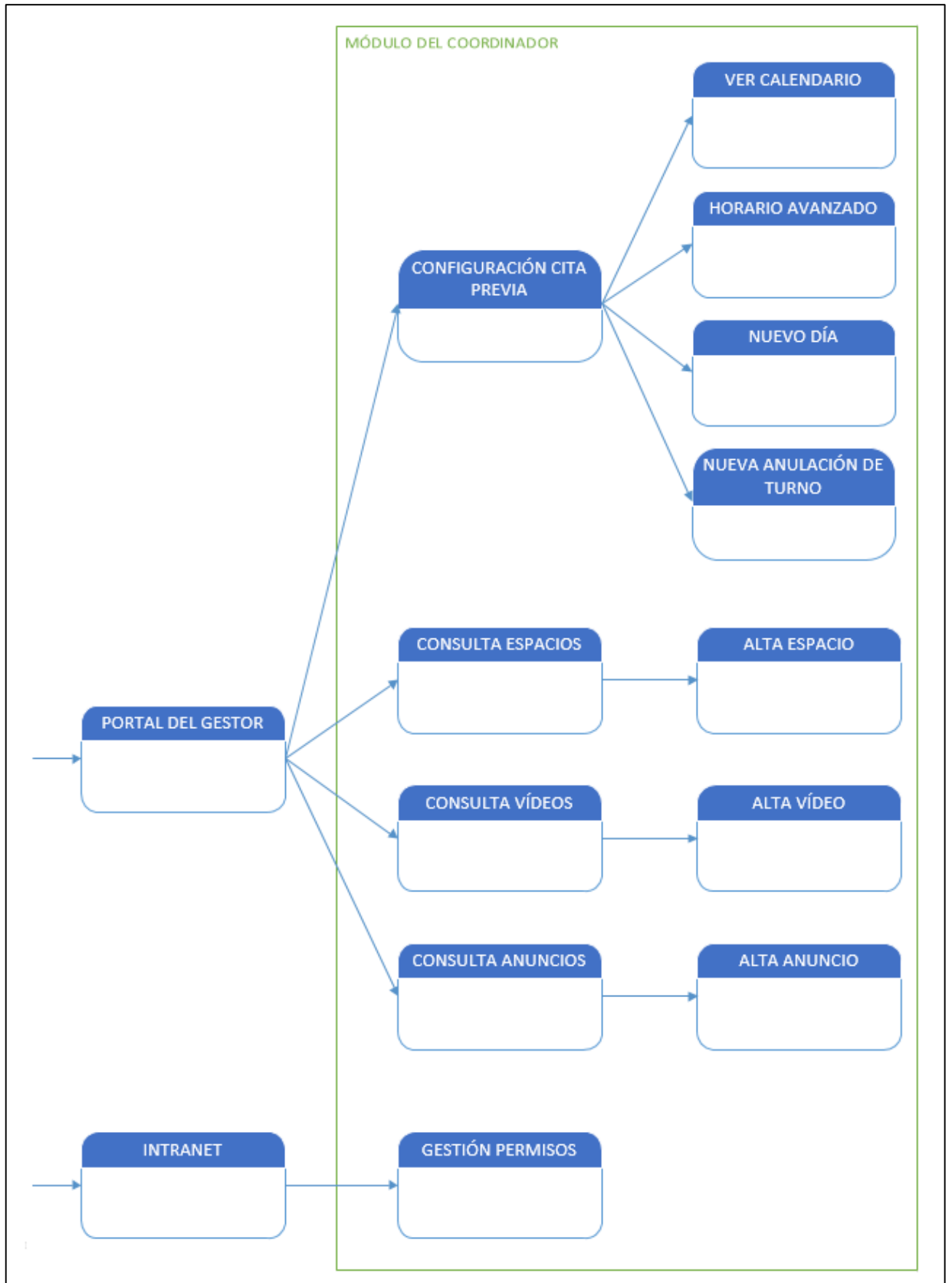


Figura 8. Diagrama de navegación del módulo del coordinador con sus puntos de acceso externos.

Hay algunas pantallas del módulo del coordinador cuyo nombre puede no dejar claro cuál es su funcionalidad:

- Ver Calendario. Es el mismo calendario que se muestra al solicitante cuando va a dar de alta una cita previa. El coordinador lo abre desde la pantalla de Configuración Cita Previa, para verificar que el calendario de atención que ha configurado para el tipo de cita previa es el que realmente quería.
- Horario Avanzado. Desde esta pantalla se configuran rangos horarios de atención adicionales al principal para el tipo de cita previa correspondiente. P. ej. podría configurarse un horario de mañanas de 9h a 14 h de lunes a viernes desde la pantalla de Configuración Cita Previa, y un horario adicional de tardes de 16h a 19h para los lunes y martes.
- Nuevo Día. Desde esta pantalla se dan de alta los días en los que no vaya a haber atención de citas por ser festivos o periodos vacacionales. También se configuran días en los que vaya a haber un horario de atención especial, como p. ej. horarios reducidos de Fallas.
- Nueva Anulación de Turno. Desde esta pantalla se configuran días o periodos de anulación de citas por causas de fuera mayor. P. ej. se podría forzar la anulación de todas las citas del día siguiente por baja médica del único gestor que atiende ese tipo de cita previa.
- Gestión de Permisos. En esta pantalla se muestra la relación de los tipos de citas previas de los que el usuario es coordinador, y sobre los cuales puede otorgar permisos de gestión a otros miembros de su unidad, es decir, convertirlos en gestores (Ges9).

Por último, en la figura 9 mostramos el diagrama de navegación del módulo del administrador. Con la pantalla Duplicar Formulario, se crea un formulario web de alta de cita previa de un nuevo tipo de cita que es copia de uno ya existente. Una vez creado el formulario, ya podemos configurar sus propiedades con la pantalla Configurar Formulario, y crear y configurar sus campos de formulario con la pantalla Nuevo Campo o cambiar sus propiedades con la pantalla Configurar Campo.

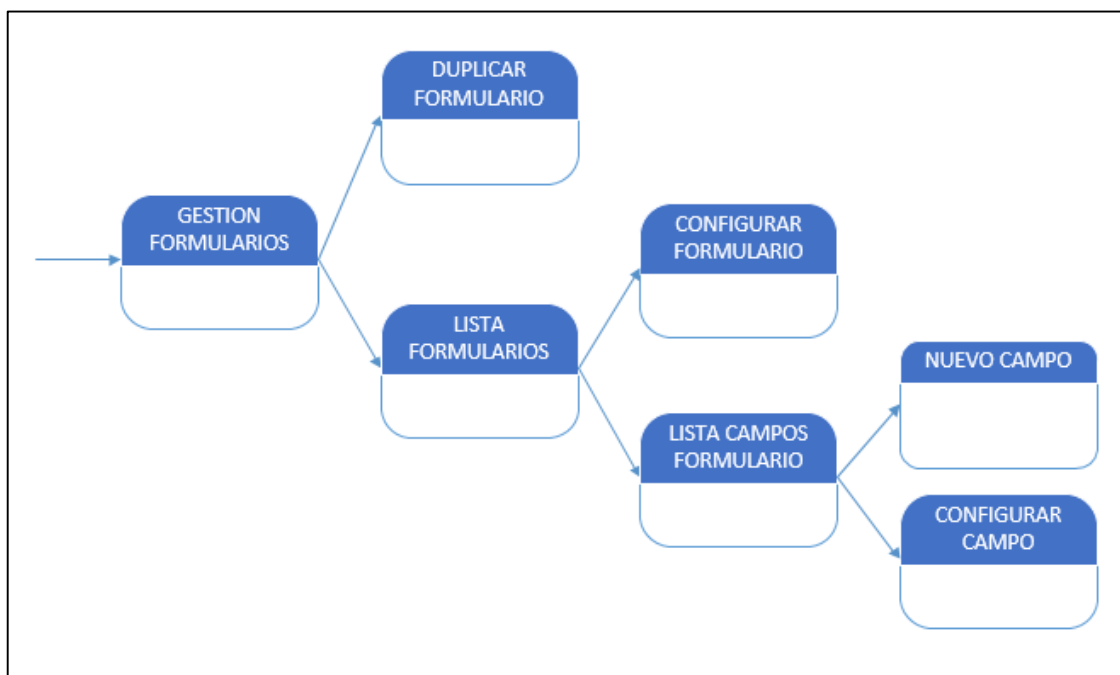


Figura 9. Diagrama de navegación del módulo del administrador.

5.2.2. Prototipo de la interfaz gráfica de usuario

En este apartado, a modo de ejemplo, presentamos un prototipo de interfaz gráfica de las cuatro pantallas que contiene el módulo de solicitud de atención inmediata. Es importante recordar de cara al diseño de la interfaz, que este módulo se ejecutará en terminales táctiles (So1AI2).

En la figura 10 se muestra la pantalla de Selección de Tipo de Cita desde la cual el usuario pulsa sobre el tipo de cita de atención inmediata del cual desea que le atiendan.

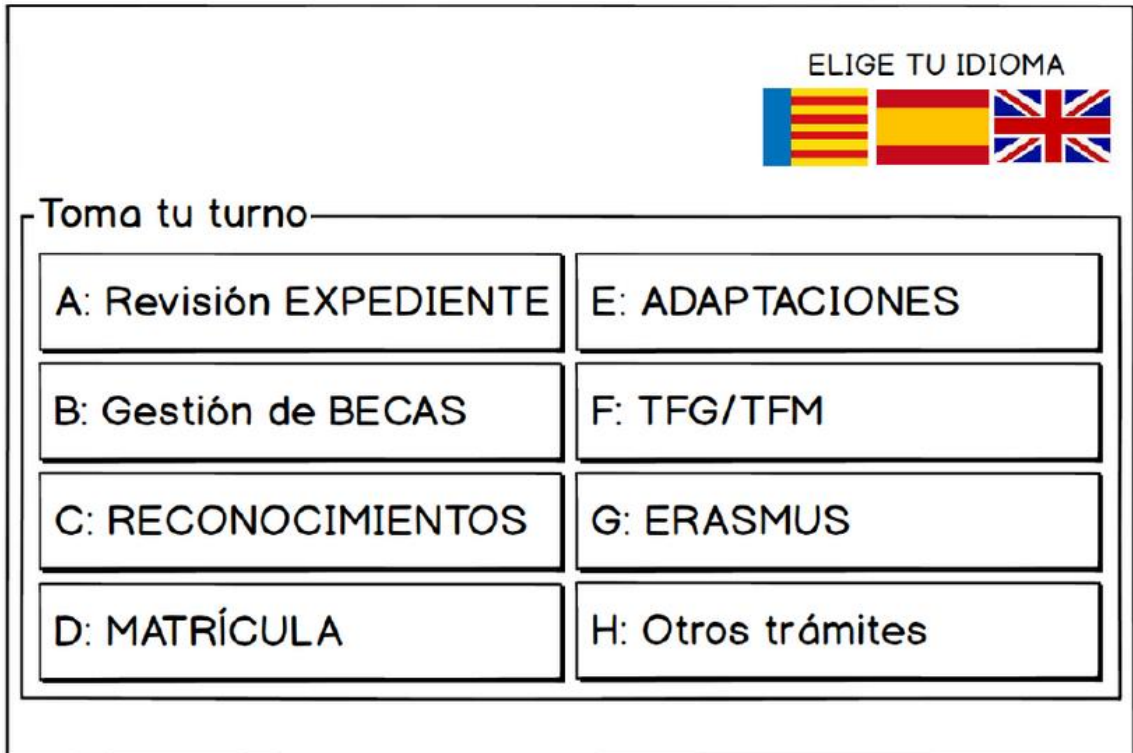


Figura 10. Pantalla de Selección de Tipo de Cita.

En la figura 11 se muestra la pantalla de Confirmación de Tipo de Cita desde la cual el solicitante introduce su nombre, y confirma el tipo de cita de atención inmediata del que desea que le atiendan. En caso de que pase su carnet universitario por el lector de tarjetas el nombre se introduce automáticamente y no necesita teclearlo.

ELIGE TU IDIOMA

Confirma que este es el tipo de atención que necesitas

TIPO DE ATENCIÓN

REVISIÓN DE EXPEDIENTE


Revisión de problemas en el expediente o dudas relacionadas con tus notas o títulos.

PASA TU CARNET POR EL LECTOR O ESCRIBE TU NOMBRE

TOMAR TURNO SALIR

Figura 11. Pantalla de Confirmación de Tipo de Cita

En la figura 12 se muestra la pantalla de Asignación de Turno. Esta pantalla, muestra el turno de atención asignado al solicitante que ha sido generado a partir del nombre que introdujo en la pantalla previa Selección de Tipo de Cita. Además, el solicitante puede introducir su dirección de *email* y pulsar tomar turno si desea que le llegue un email de confirmación de la cita con su turno asignado. Esta pantalla no se muestra si el solicitante de la cita pasó su carnet universitario por el lector de tarjetas en la pantalla anterior, ya que el dato del email se obtiene de su carnet.

ELIGE TU IDIOMA 

Toma nota de tu turno o introduce tu EMAIL si quieres que te enviemos un correo electrónico con tu código de turno.

TURNO ASIGNADO: AA01

INTRODUCE TU EMAIL Y PULSA ENVIAR

EMAIL:

HORA DE LLEGADA: 08:15


 ENVIAR

Figura 12. Pantalla de Asignación de Turno.

Por último, en la figura 13 se muestra la pantalla de Confirmación de Email. Esta pantalla visualiza un mensaje al solicitante indicándole que se le ha enviado un *email* de confirmación de la cita con su turno de atención asignado.



Figura 13. Pantalla de Confirmación de Email.

5.2.3. Diagrama Entidad/Relación

En este apartado se muestra el modelo lógico de datos sobre el que se sustentará la aplicación. Es importante señalar que el sistema gestor de bases de datos que se va a utilizar va a ser Oracle 12g, el cual trabaja con bases de datos relacionales. Por ello, se define un diagrama entidad/relación que representa el modelo lógico de datos para este tipo de bases de datos.

En la figura 14 se muestra el diagrama entidad-relación. Los cuadros con las cabeceras grises representan las tablas ya existentes de la aplicación actual de cita previa, mientras que los cuadros con las cabeceras azules representan las nuevas tablas necesarias para poder implementar la solución. Por cuestión de espacio, en algunas tablas sólo se han representado los campos más significativos (en esos casos se ha indicado esto poniendo como último campo los caracteres "...").



Figura 14. Diagrama entidad-relación de la base de datos

Las tablas de la aplicación actual de cita previa tienen que ver con la configuración de los formularios web de alta de los diferentes tipos de citas, y con el registro de la información de las citas solicitadas. Son las siguientes:

- **WEB_FRMFORM.** En esta tabla se guarda la configuración general de los formularios de alta de los diferentes tipos de citas. Se configuran datos como el título del formulario, o el nombre de la tabla donde se guardan las citas que se dan de alta.
- **WEB_FRMCAMPO.** En esta tabla se guarda la configuración de los diferentes campos que tienen los formularios de alta de los diferentes tipos de citas. Se configuran datos como el orden que tiene un campo en el formulario, su etiqueta, su valor por defecto, etc.
- **WEB_FRM_CITACFG.** En esta tabla se guarda la configuración de los calendarios de solicitud de cita para los diferentes tipos de citas. Se configuran los días de la semana en los que se va a poder solicitar cita y sus rangos horarios, así como la duración de las citas, el cupo máximo, cupo máximo por solicitante, etc.
- **WEB_FRM_CITACFG_DIA.** En esta tabla se configuran excepciones en los calendarios de solicitud de cita de los diferentes tipos de citas. Se indican festivos o periodos vacacionales como el mes de agosto en los cuales no se podrá solicitar citas.
- **WEB_FRMPERMISOS.** En esta tabla se configuran las personas que tendrán permisos de gestión en la aplicación para atender los diferentes tipos de citas.
- **WEB_FRM_XXX_YYY_SC.** Esta tabla contiene los registros con la información de las citas solicitadas para un tipo de cita determinado, dónde:
 - XXXX es el acrónimo correspondiente de la unidad donde se atienden las citas. P. ej. FBA para la Facultad de Bellas Artes, ETSINF para Escuela Técnica Superior de Informática, etc.
 - YYYYY hace referencia a la naturaleza del tipo de cita. P. ej. SEC para secretaría, IE para relaciones internacionales y con empresa, etc.
 - _SC es un sufijo opcional que indica que la tabla guarda citas de atención inmediata. Si no lo lleva indica que la tabla guarda citas previas.

El resto de tablas, son las necesarias para dar soporte a todos los requisitos funcionales. Son las siguientes:

- **CP_CONF_CITAS.** Tabla con la configuración general de los diferentes tipos de citas: unidad donde se atiende el tipo de cita, nombre del tipo de cita, descripción del tipo de cita, tabla de datos asociada al tipo de cita, etc.
- **CP_CONF_UBICACION.** Tabla donde se definen las ubicaciones desde las que es posible atender las citas previas de cada unidad.
- **CP_LLAMADAS.** Tabla que guarda la información de las llamadas que hacen los gestores a los solicitantes para atender sus citas en la unidad correspondiente.
- **CP_CONF_CITAPREVIA.** Tabla de parametrización para definir el comportamiento de las nuevas pantallas de la aplicación implementadas.
- **CP_MONITOR_UNIDAD.** Tabla con la relación de direcciones IP desde donde se puede ejecutar el panel informativo de llamadas para cada unidad.
- **CP_INMEDIATA_UNIDAD.** Tabla con la relación de direcciones IP desde donde se puede ejecutar la solicitud de atención inmediata para cada unidad.

- CP_CONF_ANUNCIOS. Tabla con la configuración de los anuncios visibles desde los paneles informativos de llamadas.
- CP_CONF_VIDEOS. Tabla con la configuración de los videos que se reproducen desde los paneles informativos de llamadas.

5.3. Estimación de esfuerzo y planificación

Una vez se tiene la descomposición de las diferentes pantallas y modelo de lógico de datos que va a tener la aplicación, ya se puede hacer una estimación del esfuerzo o coste temporal que va a implicar. La estimación del esfuerzo se obtiene estimando el coste de implementación de cada pantalla y tabla de base de datos de manera individual, y posteriormente sumando el coste de todos ellos para así obtener una estimación global. A partir de la estimación del esfuerzo en horas, ya se puede comenzar a hacer una planificación temporal dividiendo las horas totales de esfuerzo por las horas que componen una jornada, para así obtener el número de jornadas necesarias. Con el número de jornadas necesarias, ya se puede hacer una planificación temporal teniendo en cuenta los días de la semana que se van a dedicar al proyecto, y el total de recursos humanos dedicados.

Como se vio en el capítulo de metodología, se ha seguido un modelo de ciclo de vida del *software* iterativo e incremental, que implica varias iteraciones en las distintas fases del proceso de desarrollo. En este proyecto, se hizo una estimación de esfuerzo inicial para la primera iteración en base a unos requisitos iniciales, que no recogen todos los requisitos finales especificados en el capítulo de análisis. Estos requisitos finales, son todos los requisitos recogidos tras varias iteraciones del proceso de desarrollo.

En la figura 15 podemos ver la tabla que calcula esa estimación de esfuerzo inicial en horas según el método comentado. En ella se pueden ver los distintos elementos a implementar (ventanas o pantallas, objetos de bbdd, procesos, etc.) y su valoración de esfuerzo de desarrollo en horas, en función de determinadas condiciones: lenguaje de programación a utilizar, complejidad, si es de sólo consulta, etc. Para ello se ha seguido un modelo de estimación de proyectos utilizado por el ASIC [1].


 UNIVERSITAT POLITÈCNICA DE VALÈNCIA ÀREA DE SISTEMES D'INFORMACIÓ I COMUNICACIONS		ESTIMACIÓN DE ESFUERZOS DE EVOLUTIVOS POR COMPONENTES					
Identificación Evolutivo	Cita Previa						
Nombre Evolutivo	Cita Previa						
ITEM	Nombre	Lenguaje	Nuevo	Actividad	Complejidad	Sólo Consulta	Valoración Desarrollo
TABLA / B.D.	Revisión modelo de datos	Oracle	SI	datos	Compleja	NO	5
TABLA / B.D.	Trigger Alta CP CITAS PERSONA	Oracle	SI	datos	Compleja	NO	5
VENTANA DINAMICA	SELECCIÓN TIPO CITA	PLWEB	SI	vista	Media	SI	8,55
VENTANA DINAMICA	SELECCIÓN TIPO CITA	PLWEB	SI	servicio	Media	SI	7,6
VENTANA DINAMICA	CONSULTAR CITAS (Solicitante)	PLWEB	SI	vista	Compleja	SI	15,2
VENTANA DINAMICA	CONSULTAR CITAS (Solicitante)	PLWEB	SI	servicio	Compleja	SI	19
PROCESOS	NOTIFICAR CITAS	PLWEB	SI	servicio	Compleja	NO	20
VENTANA DINAMICA	CITAS DEL DIA	PLWEB	SI	vista	Compleja	NO	16
VENTANA DINAMICA	CITAS DEL DIA	PLWEB	SI	servicio	Compleja	NO	20
VENTANA DINAMICA	PANEL LLAMADAS	PLWEB	SI	servicio	Compleja	SI	19
VENTANA DINAMICA	PANEL LLAMADAS	PLWEB	SI	vista	Compleja	SI	15,2
VENTANA DINAMICA	MI ESPACIO	PLWEB	SI	servicio	Simple	SI	4,75
VENTANA DINAMICA	MI ESPACIO	PLWEB	SI	vista	Simple	SI	4,75
VENTANA DINAMICA	GESTIONAR CITAS (Gestor)	PLWEB	SI	vista	Compleja	NO	16
VENTANA DINAMICA	GESTIONAR CITAS (Gestor)	PLWEB	SI	servicio	Compleja	NO	20
VENTANA DINAMICA	ANULAR CITAS (Gestor)	PLWEB	SI	servicio	Compleja	NO	20
VENTANA DINAMICA	ANULAR CITAS (Gestor)	PLWEB	SI	vista	Compleja	NO	16
VENTANA DINAMICA	DETALLE CITA (Gestor)	PLWEB	SI	servicio	Compleja	SI	19
VENTANA DINAMICA	DETALLE CITA (Gestor)	PLWEB	SI	vista	Compleja	SI	15,2
VENTANA DINAMICA	ENVIAR SMS (Gestor)	PLWEB	SI	servicio	Compleja	NO	20
VENTANA DINAMICA	ENVIAR EMAIL (Gestor)	PLWEB	SI	servicio	Simple	NO	5
VENTANA DINAMICA	CONFIGURAR UBICACION	PLWEB	SI	servicio	Simple	NO	5
VENTANA DINAMICA	CONFIGURAR UBICACION	PLWEB	SI	vista	Simple	NO	5
VENTANA DINAMICA	GESTION PERMISOS	PLWEB	SI	servicio	Compleja	NO	20
VENTANA DINAMICA	GESTION PERMISOS	PLWEB	SI	vista	Compleja	NO	16
VENTANA DINAMICA	CONFIGURACIÓN CITA PREVIA	PLWEB	SI	vista	Compleja	NO	16
VENTANA DINAMICA	CONFIGURACIÓN CITA PREVIA	PLWEB	SI	servicio	Compleja	NO	20
VENTANA DINAMICA	ESTADISTICAS CITAS	PLWEB	SI	servicio	Compleja	SI	19
VENTANA DINAMICA	ESTADISTICAS CITAS	PLWEB	SI	vista	Compleja	SI	15,2
VENTANA DINAMICA	SELECCIÓN TIPO CITA (S.A.I.)	PLWEB	SI	servicio	Compleja	NO	20
VENTANA DINAMICA	SELECCIÓN TIPO CITA (S.A.I.)	PLWEB	SI	vista	Compleja	NO	16
VENTANA DINAMICA	CONFIRMACIÓN TIPO DE CITA (S.A.I.)	PLWEB	SI	servicio	Compleja	NO	20
VENTANA DINAMICA	CONFIRMACIÓN TIPO DE CITA (S.A.I.)	PLWEB	SI	vista	Compleja	NO	16
VENTANA DINAMICA	ASIGNACIÓN DE TURNO (S.A.I.)	PLWEB	SI	servicio	Media	NO	8
VENTANA DINAMICA	ASIGNACIÓN DE TURNO (S.A.I.)	PLWEB	SI	vista	Media	NO	9
TOTAL HORAS ESTIMADAS							496,45
TOTAL JORNADAS ESTIMADAS							62,06

Figura 15. Tabla de estimación de esfuerzo inicial.

Como esta tabla no recoge los requisitos finales de la aplicación, sino sólo los iniciales, hay elementos que se han implementado y que no están en la tabla. Si nos fijamos en las figuras de los diagramas de navegación del anterior apartado, veremos que p. ej. para el portal del gestor no figura en la tabla los elementos correspondientes a la pantalla de consultar llamadas. También se ha querido destacar en azul aquellos elementos que no son nuevos (aunque así lo indique la tabla) y que sólo han sido modificados. Estos elementos marcados en azul son elementos que posteriormente se descubrió que ya existían en la aplicación actual de cita previa, que fue el punto de partida de la solución propuesta, y por eso sólo se han modificado. En rojo están marcados

elementos que responden a requisitos iniciales que finalmente fueron descartados en posteriores iteraciones del análisis.

En cuanto a la planificación temporal, no la hubo como tal. Varias circunstancias propiciaron el no hacer una planificación temporal que estaba condenada a no cumplirse. Estas circunstancias fueron:

- Cambios de prioridad en el desarrollo del proyecto. El proyecto quedó paralizado por dos años para dar prioridad a otros proyectos surgidos considerados en esos momentos más importantes para la UPV.
- Cambios drásticos de requisitos que implicaban nuevas iteraciones en el proceso de desarrollo.
- Participación de varios equipos de trabajo en el proyecto cuyos miembros participaban también en otros proyectos, lo que complicaba mucho una planificación temporal.
- Dependencia de una empresa externa para modificaciones en algunos de los componentes de la solución, unido a los retrasos continuos en las entregas de algunas de esas modificaciones.

5.4. Tecnología utilizada

Para la implementación de las diferentes pantallas se ha utilizado la tecnología PL/SQL Web Toolkit de Oracle. Esta tecnología permite implementar aplicaciones web con bases de datos Oracle, y ha sido utilizada por el ASIC para desarrollar otras aplicaciones web disponibles en la intranet del alumnado y profesorado de la UPV. Entre ellas podemos citar como ejemplos, la generación de diligencias de profesor, la generación de certificados y justificantes para los alumnos, o la solicitud de tutorías bajo demanda dirigida a alumnos y profesores.

Para explicar a modo de introducción esta tecnología [2], se muestra su arquitectura en la figura 16.

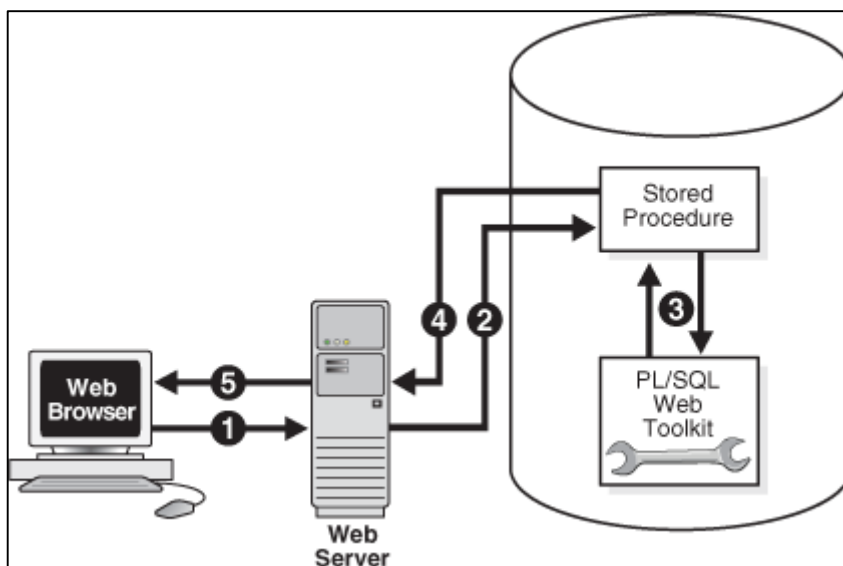


Figura 16. Arquitectura de la tecnología PL/SQL Web Toolkit de Oracle.

Esta arquitectura es muy similar a la arquitectura estándar que usan las aplicaciones web de tres capas. La única diferencia es que en este caso la lógica de negocio reside también en la base de datos y no en el servidor web. Éste último tan solo hace de enlace entre el navegador web del usuario y la base de datos. A continuación, explicamos el funcionamiento con más detalle de acuerdo con los pasos seguidos en la ilustración anterior:

1. El usuario visita una página web, sigue un enlace de hipertexto o envía los datos de un formulario. Esto hace que el navegador envíe una solicitud HTTP de una URL a un servidor HTTP.
2. El servidor HTTP llama a un procedimiento almacenado en una base de datos Oracle conforme a los datos codificados en la URL. Los valores de los parámetros de la URL son los datos que se pasan como valores en los parámetros del procedimiento almacenado.
3. El procedimiento almacenado invoca a procedimientos y funciones del módulo de Oracle PL/SQL Web Toolkit de Oracle. Estos procedimientos, como HTP.P, sirven para generar código HTML dinámicamente. El contenido de la página web generada variará de acuerdo con la lógica de negocio que se aplique en el procedimiento almacenado, y en ocasiones también en función de los valores de sus parámetros de entrada.
4. La página web generada dinámicamente se envía al servidor web.
5. El servidor web devuelve la página web al usuario en su navegador.

Aunque PL/SQL Web Toolkit de Oracle es la tecnología principal subyacente, cabe mencionar que para la implementación también se han utilizado otros lenguajes de programación, tecnologías, IDEs, y librerías de terceros, algunos bastante habituales en los desarrollos web:

- HTML5, CSS3, JavaScript y AJAX, como lenguajes y tecnologías para la generación de las interfaces web de usuario.
- El lenguaje de programación PL/SQL de Oracle, para la programación de la lógica de negocio y acceso a datos.
- Los IDEs de Oracle SQL Developer, SQL Navigator, y TOAD, para facilitar la programación con PL/SQL.
- La librería de terceros jkeyboard¹. Se utilizó la librería de terceros jkeyboard para poder utilizar el teclado virtual que se muestra en la solicitud de atención inmediata llamada. Dicha librería desarrollada en JQUERY, se adaptó y configuró para que el teclado tuviera la apariencia deseada. También se instaló la librería JQUERY como requisito del teclado virtual.
- El lenguaje de programación Python, para la implementación de los servicios necesarios para poder implantar el módulo de solicitud de atención inmediata en el dispositivo táctil.
- El IDE de Python Spyder, para facilitar la programación con este lenguaje.

¹ Código fuente de la librería disponible en <https://github.com/javidan/jkeyboard>

6. Implementación de la solución

En este capítulo se detalla cómo se ha implementado la solución. Para ello, se explica el estándar de desarrollo y buenas prácticas de programación utilizadas, así como los diferentes módulos *software* que se han generado.

6.1. Estándar de desarrollo utilizado

La implementación de la solución se ha hecho siguiendo un estándar de programación del ASIC para el desarrollo aplicaciones web PL/SQL llamado estándar multivista [3]. Este estándar permite un desarrollo más rápido, homogéneo y seguro de aplicaciones web usando los paquetes de Oracle a través del módulo WEB TOOLKIT de Oracle explicado en el capítulo anterior.

Este estándar se llama multivista porque permite cambiar sin ningún esfuerzo la apariencia de las páginas web en función del contexto desde el cual se acceden, y respetando siempre sus elementos principales y disposición. La “vista” que adopta la página web viene definida por el parámetro P_VISTA de la URL que la invoca. De este modo existen varias vistas en función del valor que toma este parámetro. He aquí las principales con las que se sigue desarrollando actualmente:

- Vista normal (P_VISTA=normal). Esta vista corresponde a la presentación de contenidos web de información pública de la universidad y que no requieren de autenticación previa en la intranet.
- Vista intranet (P_VISTA=intranet). Esta vista es la usada por todas las páginas de aplicaciones y utilidades corporativas que hacen uso de datos personales. Por lo tanto, a estas páginas se accede desde la intranet una vez autenticado en la misma.
- Vista móvil (P_VISTA=móvil). Esta vista es la que se utiliza para mostrar las páginas desde dispositivos móviles para que estas tengan un comportamiento *responsive* y así se adapten al tamaño de estos. Se utiliza tanto para contenidos públicos como para contenidos privados que requieren autenticación previa en la intranet.

Como a día de hoy la solución no está totalmente implantada en explotación, para mostrar el aspecto de las tres vistas se puede poner como ejemplo real el Portal de Consultas UPV. Desde este portal podemos seleccionar la escuela, estudio, o servicio universitario al cual vamos a poner nuestra consulta desde la aplicación poli[Consulta]. Dependiendo desde dónde o cómo accedamos al portal, éste se mostrará utilizando una vista u otra.

En la figura 17 podemos ver el enlace de acceso al Portal de Consultas UPV desde la página web de la UPV que es de dominio público.

Aplicación de gestión de citas previas y citas de atención inmediata de la UPV: poli[Cita]

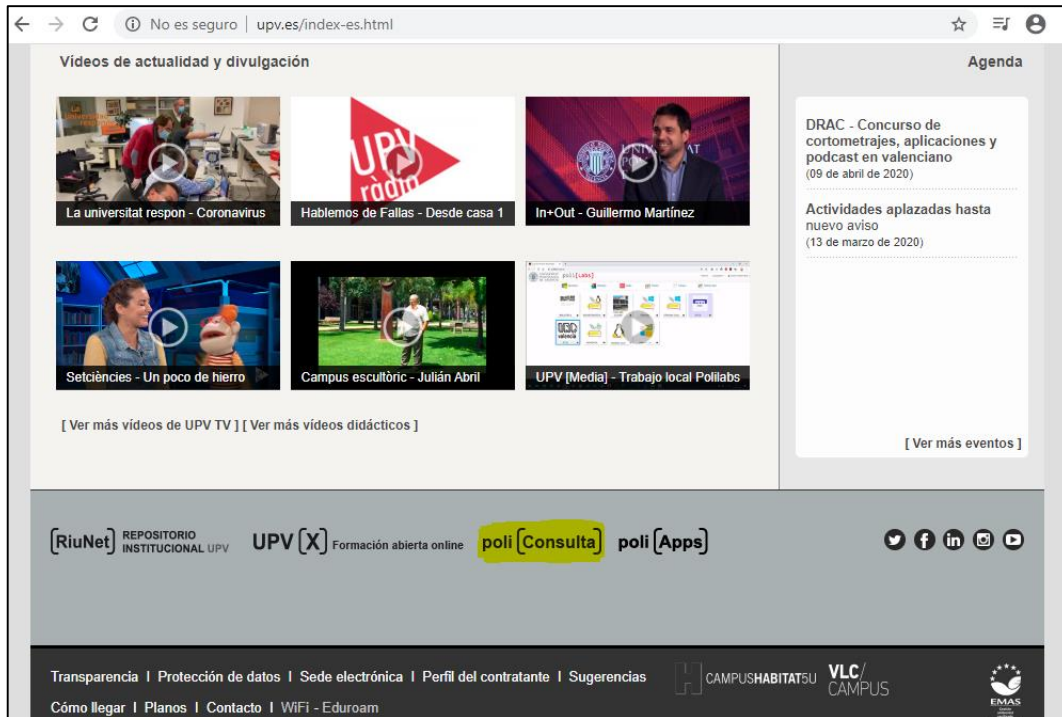


Figura 17. Acceso al Portal de Consultas UPV desde la página web de la UPV.

Una vez pulsamos el enlace acceso al Portal de Consultas UPV desde la página web de la UPV, podemos apreciar en la figura 18 como el Portal de Consultas UPV se abre con la vista normal.



Aplicación de gestión de citas previas y citas de atención inmediata de la UPV: poli[Cita]

Figura 18. Portal de Consultas UPV con vista normal accedido desde la página web de la UPV.

En la figura 19 se muestra el enlace de acceso al Portal de Consultas UPV desde la intranet de la UPV a la cual hay que autenticarse para acceder.

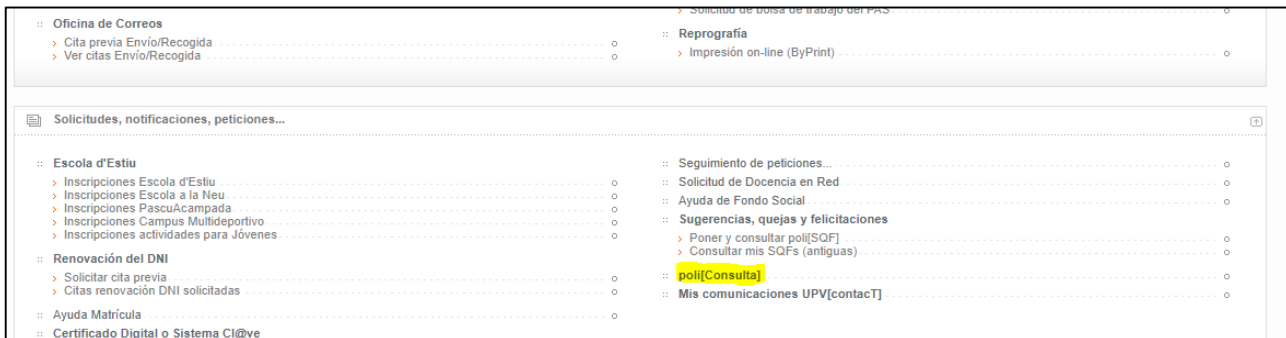


Figura 19. Acceso al Portal de Consultas UPV desde la intranet de la UPV.

Una vez pulsamos el enlace acceso al Portal de Consultas UPV desde la intranet de la UPV, podemos apreciar en la figura 20 como el Portal de Consultas UPV se abre con la vista intranet.



Figura 20. Portal de Consultas UPV con vista intranet accedida desde la intranet de la UPV.

En la figura 21 se muestra el enlace de acceso al Portal de Consultas UPV desde la página web de la UPV vista desde un teléfono móvil.

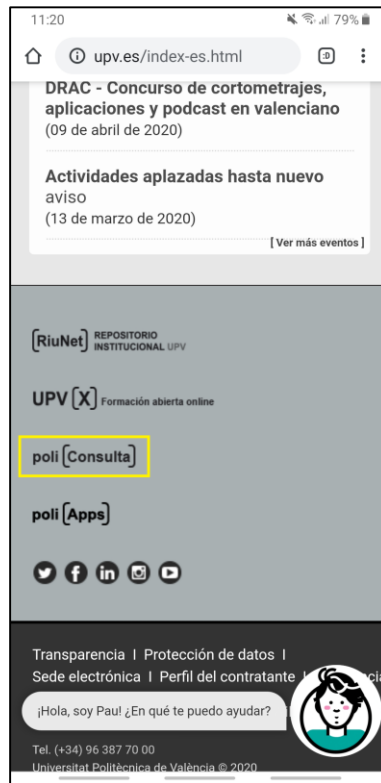


Figura 21. Acceso al Portal de Consultas UPV desde la página web de la UPV visto desde un teléfono móvil.

Una vez pulsamos el enlace acceso al Portal de Consultas UPV desde la web de la UPV vista desde un teléfono móvil, podemos apreciar en la figura 22 como el Portal de Consultas UPV se abre con la vista móvil.

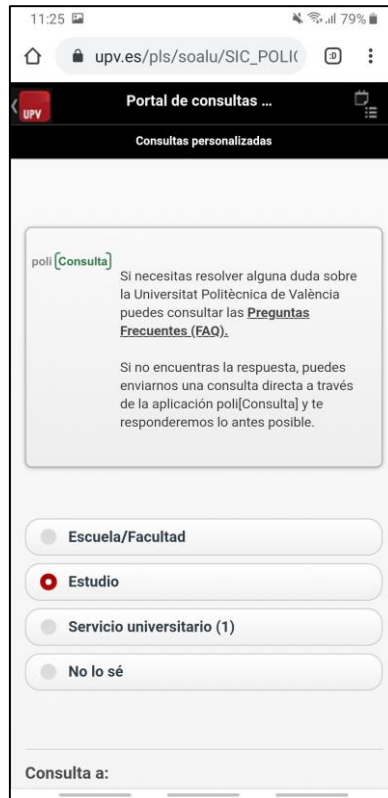


Figura 22. Portal de Consultas UPV con vista móvil accedido desde la página web de la UPV visto desde un teléfono móvil.

Aunque a día de hoy aún no se puede poner un ejemplo real de nuestra solución con las tres vistas, se puede forzar el valor del parámetro P_VISTA de la página inicial del portal del solicitante para ver cómo queda con cada vista. Podemos ver su código fuente en el Anexo I para entender con más detalle el estándar con este ejemplo.

En las figuras 23, 24, y 25 podemos apreciar cómo queda la página inicial del portal del solicitante de poli[Cita] para cada vista, forzando el valor de su parámetro P_VISTA a los valores normal, intranet y móvil respectivamente.

Aplicación de gestión de citas previas y citas de atención inmediata de la UPV: poli[Cita]

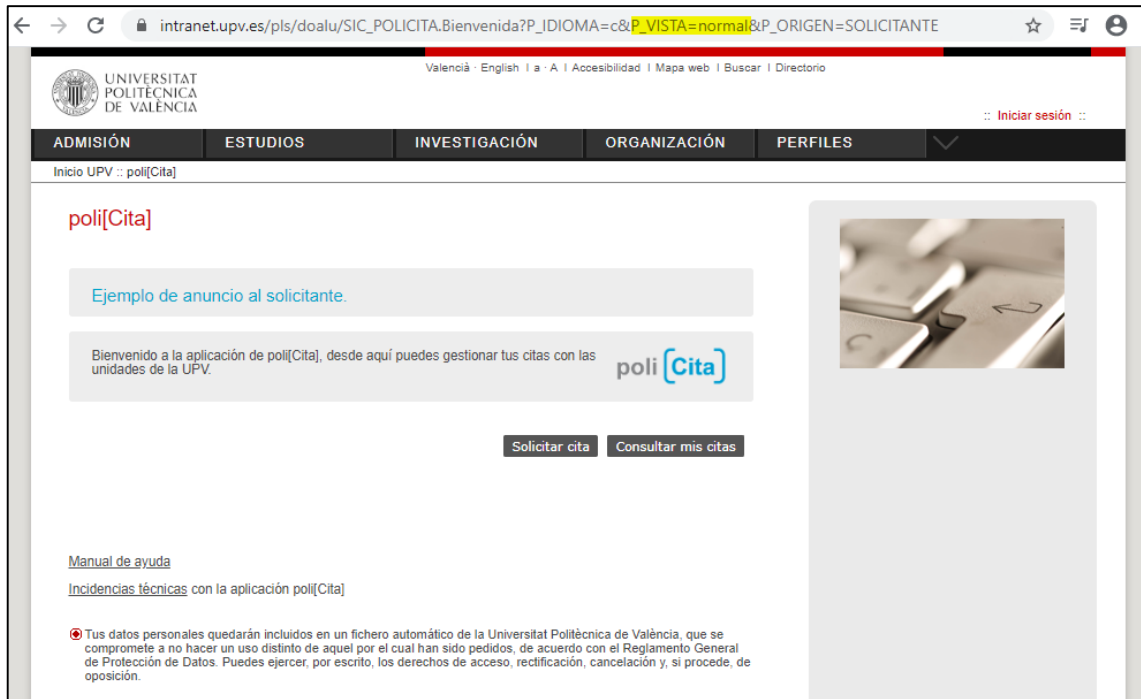


Figura 23. Pantalla inicial del portal del solicitante de poli[Cita] con vista normal.



Figura 24. Pantalla inicial del portal del solicitante de poli[Cita] con vista intranet.

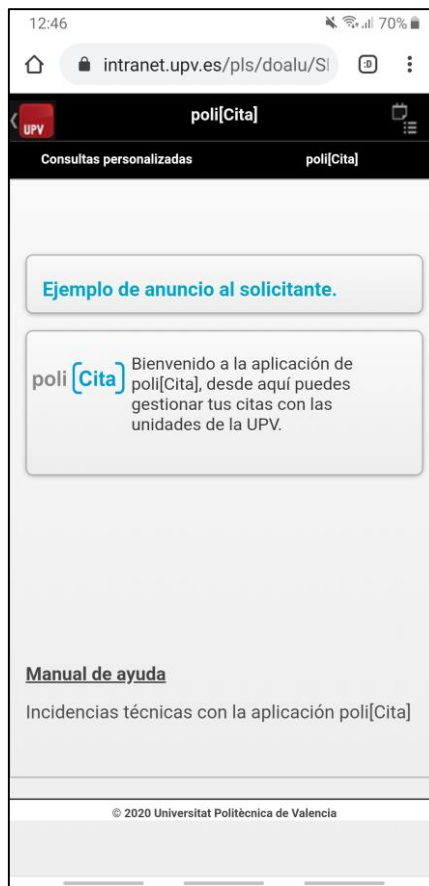


Figura 25. Pantalla inicial del portal del solicitante de poli[Cita] con vista móvil.

Además de la utilidad “multivista”, este estándar ofrece una lista de componentes¹ para poder desarrollar las páginas web con apariencia y estructura similar. Algunos de estos componentes sirven para construir la estructura general de la página, y luego hay otros que sirven para representar información en ella, como listados simples, listados multi-columna, fichas, y formularios.

6.2. Buenas prácticas

En el desarrollo de la solución se ha intentado seguir en la medida de lo posible algunas buenas prácticas de desarrollo que ayudan a crear un *software* de calidad. El crear *software* de calidad implica que éste será más eficiente, entendible, flexible y fácil de mantener, y también será más robusto y menos propenso a errores. Podemos afirmar que se han seguido las siguientes buenas prácticas de desarrollo:

- Código muy cohesionado y poco acoplado. Con cohesión nos referimos al grado en el que cada procedimiento, función o módulo hace lo que debe hacer y nada más, es decir, que tiene un único objetivo. Con acoplamiento nos referimos al grado de dependencia que los diferentes procedimientos y funciones tienen entre sí. Un código altamente

¹ En http://www.upv.es/pls/doalu/stdes.mv_inicio podemos ver ejemplos explicativos de estos componentes.

cohesionado y poco acoplado será más sencillo de modificar futuro, menos propenso a errores, y más fácil de entender por otra persona que no sea el autor.

- Encapsular al máximo posible el código. Con esto nos referimos a que no haya código duplicado y esté encapsulado en funciones o procedimientos para así poder ser reutilizado tantas veces como se desee. Esto nos ayudará a tener un código mucho más comprensible, ya que cuando lo analicemos nos abstraeremos de muchas funcionalidades secundarias que son realizadas por funciones o procedimientos a los que se llaman, y nos centraremos sólo en la funcionalidad principal. El código también será menos propenso a errores, y evitaremos cosas como p. ej., hacer un cambio y que se nos olvide aplicar dicho cambio a todos los fragmentos de código duplicado.
- Separación por capas. Como se vio, la tecnología PL/SQL Web nos obliga a tener las distintas capas de nuestra aplicación, presentación, lógica de negocio y datos, en el mismo lugar, los procedimientos almacenados de base de datos Oracle. Esto no impide que el código se pueda organizar a nivel lógico para intentar separar estas tres capas lo máximo posible. En este sentido, se ha implementado la capa de presentación en procedimientos separados de los procedimientos y funciones que contienen la lógica de negocio y datos.
- Código comentado. Se ha comentado el código con los aspectos clave de su funcionalidad para que sea más comprensible y fácil de mantener.
- Nomenclatura de los módulos *software*. Los distintos objetos de base de datos (funciones, procedimientos almacenados, paquetes, tablas, disparadores, etc.) implementados han seguido unas reglas de nomenclatura de acuerdo a un estándar de base de datos [4] para que, a partir de su nombre, se pueda saber de qué tipo de objeto se trata y a qué esquema pertenece. En este sentido todos los nombres de los objetos llevan un prefijo que hace referencia al esquema de base de datos al que pertenece. P. ej. el prefijo “CP_” hace referencia al esquema CITA, que contiene todos los objetos de la aplicación, salvo los paquetes que contienen los procedimientos de pantallas, cuyos nombres empiezan por “SIC_”. Por otro lado, todos los nombres de objetos de base de datos, salvo las tablas, llevan también un sufijo que hace referencia al tipo de objeto del que se trata: “_FU” para funciones, “_PR” para procedimientos, “_TR” para disparadores, etc.
- Herramienta de control de versiones. Este tipo de herramientas permiten guardar diferentes versiones del código que luego son recuperables. Además, tienen otras utilidades, como evitar que dos programadores accedan concurrentemente al mismo código y solapen modificaciones, o poder comparar código de diferentes versiones. En este caso se ha utilizado la herramienta de control de versiones Subversion¹.

6.3. Objetos de BBDD implementados

Como se comentó, para implementar la solución se ha tecnología PL/SQL Web Toolkit de Oracle. Esta tecnología permite implementar aplicaciones web utilizando procedimientos almacenados de base de datos Oracle. Estos procedimientos, son objetos de la base de datos, que, junto a otros como funciones, paquetes, tablas, vistas, etc. implementan todos ellos la solución.

Por otro lado, y tal como se explicó, la solución se ha basado en adaptar la aplicación de Cita Previa ya existente. Por tanto, tenemos objetos de base de datos que ya existían y cuyo código ha

¹ Página web: <https://subversion.apache.org/>



sido modificado para satisfacer parte de los requisitos necesarios, así como nuevos objetos de base de datos que se han implementado desde cero para satisfacer el resto de los requisitos.

Estos son los nuevos objetos de base de datos que se han implementado desde cero:

- Paquete SIC_POLICTA. Este paquete contiene los procedimientos almacenados que implementan las nuevas pantallas del portal del solicitante y del portal del gestor. También contiene el procedimiento que implementa el panel informativo de llamadas, y los procedimientos de las pantallas que componen el nuevo módulo de solicitud de atención inmediata.
- Función de base de datos CP_GENERAR_IDENTIFICADOR_FU. Esta función genera los identificadores de las nuevas citas solicitadas a partir del nombre y apellidos del solicitante. Cuando se va a insertar una nueva cita en la tabla de datos del tipo de cita se dispara un disparador de tipo “before insert”. Este disparador llama a la función, y pone el identificador devuelto por ésta como valor del campo IDENTIFICADOR de la tabla.
- *Jobs* de base datos. Un *job* es un objeto de base de datos Oracle con el que se puede definir una tarea programada, y que por identificador tiene un código numérico en vez de un nombre. Se han implementado dos *jobs* en función de su propósito:
 - Job de recordatorio a los solicitantes. Este *job* se ejecuta diariamente a primera hora de la mañana y llama al procedimiento almacenado CP_NOTIF_CITA_PREVIA_PR. Este procedimiento comprueba para cada tipo de cita, las citas que están pendientes y que les quedan un número determinado de días para atenderse, y envía a sus solicitantes un recordatorio por correo electrónico.
 - Job de borrado de llamadas. Este Job se ejecuta diariamente a primera hora de la mañana y llama al procedimiento almacenado CP_BORRADO_LLAMADAS_PR. Este procedimiento borra los registros de llamadas de gestores a solicitantes cuya antigüedad es superior a un número determinado de días.
- Vistas WEB_FRM_PC_XXXn. Estas vistas muestran para cada tipo de cita las citas previas que se han solicitado. Están diseñadas para que un solicitante vea en sus calendarios iCal UPV¹ un calendario con las citas previas que ha solicitado, estén pendientes o no. Para cada vista, XXX hace referencia a las siglas de la unidad correspondiente, y n hace referencia a un n° secuencial que diferencia cada uno de los tipos citas de esa unidad.
- Tablas. Tal y como se detalla en el apartado Diagrama Entidad/Relación del anterior capítulo del Diseño, se han creado las siguientes tablas nuevas: CP_CONF_CITAS, CP_CONF_UBICACIÓN, CP_LLAMADAS, CP_MONITOR_UNIDAD, CP_INMEDIATA_UNIDAD, CP_CONF_CITAPREVIA, CP_CONF_ANUNCIOS, y CP_CONF_VIDEOS.
- Disparadores y secuencias. Para cada una de las nuevas tablas creadas con identificador auto numérico se ha tenido que crear un disparador y una secuencia para poder convertir su identificador en auto numérico. El disparador de tipo “before insert” cada vez que se dispara genera un nuevo valor de la secuencia, y dicho valor se asigna al identificador de la tabla para que este simule su comportamiento auto numérico. Estos identificadores

¹ Los calendarios iCal UPV están accesibles desde la intranet de la UPV desde el enlace [Calendarios UPV en formato iCal](#)

auto numéricos se tuvieron que implementar de este modo ya que la versión 11g de Oracle con la que se empezó a implementar no tiene campos de tipo auto numérico.

Y estos son objetos de base de datos que ya existían y que han sido modificados:

- Paquete de base de datos SIC_FRM. Este paquete contiene procedimientos almacenados que implementan pantallas del portal del solicitante y del portal del gestor que han sido modificadas. En este paquete se han hecho las siguientes modificaciones:
 - Se ha modificado la pantalla de consulta de citas de los solicitantes para que se pueda consultar citas de cualquier tipo de cita de cualquier unidad, y se le ha añadido el filtro de citas vigentes (GesSol1).
 - Se ha modificado la pantalla de gestión de citas, que agrupa todas las citas de un mismo tipo, para que cuando se anule o elimine una cita se envíe un *email* de notificación al solicitante (Not2).
- Paquete de base de datos SIC_FRMCITAS. Este paquete contiene el procedimiento que implementa la pantalla de configuración de horarios de atención de los diferentes tipos de citas. Se ha modificado para que permita configurar más de un rango horario de atención por día de la semana (Conf1).
- Paquete de base de datos SIC_FRMADMIN. Este paquete contiene los procedimientos que implementan las pantallas del módulo del administrador. También contiene el procedimiento de la pantalla de Gestión Permisos con la que el coordinador da permisos de gestión a otros miembros de su unidad. Realmente en este paquete no se ha implementado ninguna funcionalidad nueva, tan solo se separó del paquete SIR_FRMCITAS para facilitar la gestión del código.
- Tablas de datos. Se ha modificado la tabla WEB_FRM_CITACFG para incluir los nuevos campos necesarios para poder tener más de un rango horario de atención por día: re1_horaini, re1_horafin, re1_duracion, re1_cupo, etc.

Además de todos los objetos de base de datos relacionados con la tecnología PL/SQL Web Toolkit de Oracle, también se han implementado los servicios necesarios para poder obtener los datos de las lecturas de carnets universitarios desde el módulo de solicitud de atención inmediata. Estos servicios, son servicios normales de Linux ya que está previsto que el módulo de solicitud inmediata se implante en una RasPad¹, con Raspbian². Estos servicios son:

- Readusb.service. Este servicio recoge las lecturas del dispositivo de lectura conectado al puerto USB de la RasPad, y almacena el código de lectura ROMCode en un fichero de texto local: rom.txt.
- Webservice.service. Este servicio es un servicio web que lee la última entrada de lectura del rom.txt, y consulta otro servicio web a partir del ROMCode para obtener y publicar los datos el propietario del carnet leído: nombre y apellidos, *email*, DNI... Este servicio se publica por HTTP localmente a través de *localhost* y públicamente a las IPs autorizadas. Es el servicio que es invocado desde la pantalla de Confirmación Tipo de Cita cuando se

¹ RasPad es una tableta Raspberry Pi. Más detalles en <https://www.raspad.com/products/all-in-one-raspberry-pi-display>

² Raspbian es una distribución del sistema operativo GNU/Linux basado en Debian. Más detalles en <https://www.raspberrypi.org/downloads/raspbian/>

detecta con el dispositivo de lectura que se ha hecho una lectura, para así volcar los datos del propietario en las pantallas sin necesidad de introducirlos manualmente.

- Kiosk.service. Este servicio no tiene que ver con la lectura de carnets. Es el servicio de arranque del navegador Google Chrome que mostrará las pantallas del módulo de solicitud de atención inmediata con bloqueo de pantalla de usuario (modo Kiosco).

Por último, cabe señalar que, aunque el desarrollo de la *app* como alternativa al portal web para la solicitud y consulta/gestión de citas previas del solicitante es un requisito a satisfacer (ReqNF4), ésta aún no ha sido desarrollada a día de hoy por temas de planificación de proyectos.

7. Pruebas

A lo largo del proceso de desarrollo de la solución, se han ido realizando diferentes tipos de pruebas en función de la fase en que se encontraba en cada momento. Así pues, se han realizado los siguientes tipos de pruebas:

- Pruebas unitarias. Se han realizado las pruebas unitarias de los diferentes elementos implementados (procedimiento, funciones, pantallas, etc.) para validar su correcto funcionamiento. Han sido realizadas por el propio programador que las ha implementado.
- Pruebas de integración. Una vez realizadas las pruebas unitarias de los diferentes módulos, e integrados estos entre sí, se han hecho las pruebas de integración del *software* resultante. También han sido realizadas por el propio programador que ha implementado e integrado estos módulos.
- Pruebas de regresión. Cada vez que se ha integrado un nuevo módulo, además de la prueba de integración, se ha realizado la prueba de regresión correspondiente. Estas pruebas de regresión consisten en realizar un conjunto de pruebas ya realizadas previamente, para validar que no se ha producido ningún efecto colateral no deseado al integrar el nuevo módulo. Estas pruebas han sido practicadas por el programador y el analista.
- Pruebas funcionales. Una vez completado el desarrollo se han practicado los diferentes casos de prueba funcionales para validar que éste cumple con todos los requisitos funcionales especificados en la fase de análisis. Estas pruebas han sido realizadas por el programador y el analista. Podemos ver un ejemplo completo de conjunto de casos de pruebas funcionales en el Anexo II.
- Pruebas de sistema. Una vez desarrollada completamente la solución, se han practicado los diferentes casos de prueba para validar que el *software* implementado cumple con los estándares de calidad establecidos. Son los llamados requisitos no funcionales. Este tipo de pruebas han sido realizadas por el programador y el analista. Las pruebas que se han practicado de este tipo son:
 - Pruebas de seguridad. En este apartado podemos mencionar las pruebas que se han realizado de inyección SQL y XSS (Cross-Site Scripting). Con estas pruebas se ha asegurado que no se pueda inyectar código SQL o JavaScript malicioso a través de los campos de los formularios.
 - Pruebas de rendimiento en condiciones normales.
 - Pruebas de recuperación ante fallos.
 - Pruebas de usabilidad de las diferentes pantallas. Para ello se ha observado cómo los usuarios interaccionan con las pantallas desarrolladas.
 - Pruebas de integración con otros sistemas o servicios externos. Aquí podemos mencionar como ejemplo, las pruebas que se realizaron de integración con el servicio web de lectura del carnet universitario en la solicitud de atención inmediata. También podemos mencionar como ejemplo, las pruebas que se han realizado para verificar que el panel de llamadas y la solicitud de atención inmediata, han seguido funcionando y visualizándose correctamente en los navegadores de los dispositivos donde finalmente se van a implantar, y que son distintos al navegador del equipo utilizado para su desarrollo.

- Pruebas de aceptación. Se ha validado con un conjunto de usuarios que la solución desarrollada cumple con todos los requisitos funcionales, y que cumple con otro requisito no funcional, la usabilidad de la aplicación.

Hay que resaltar que algunos de estos tipos de pruebas han sido aplicados de forma iterativa de acuerdo con el ciclo de vida iterativo e incremental que se ha seguido. Pruebas como las funcionales y las de aceptación, se han repetido al final de cada iteración, donde los usuarios han validado las nuevas funcionalidades desarrolladas en la dicha iteración antes de que se comenzara con la siguiente.

8. Implantación

En la fase de implantación hubo un plan inicial para implantar la aplicación de poli[Cita] en la Facultad de Bellas Artes a finales de marzo de 2020 como proyecto piloto de implantación de la aplicación en la UPV. Este plan de implantación contemplaba toda la funcionalidad de la aplicación salvo la *app*, que a fecha de hoy aún no ha sido desarrollada por temas de planificación de proyectos. La idea era implantar primero en esta facultad como prueba real de uso en explotación, para así detectar posibles errores de la aplicación o problemas de uso y corregirlos. Una vez asegurado el éxito de la implantación en esta facultad, el plan ya contemplaba la implantación al resto de escuelas y facultades de la UPV, así como a otras unidades de la universidad que necesitaran utilizar la aplicación poli[Cita].

Este plan de implantación fue truncado por la erupción a mediados de marzo de la pandemia del virus COVID-19, que a día de hoy todavía persiste en España y que nos ha obligado a estar confinados dos meses en nuestros hogares. No obstante, el proceso de desescalada del confinamiento que actualmente se está llevando a cabo ha requerido de nuevo la necesidad de la implantación de la aplicación poli[Cita] en la UPV. Esta implantación ha sido limitada en cuanto a la funcionalidad de la aplicación debido a las restricciones de presencialidad en la universidad por riesgos de contagio. Por ello, la implantación que actualmente ya se está llevando a cabo está siendo limitada, y sólo cubre la funcionalidad de solicitud de cita previa desde el portal del solicitante, y a una atención básica de las citas por parte de los gestores. Por ello, no han sido implantados los siguientes componentes de la aplicación:

- Panel informativo de llamadas. Este componente no ha sido implantado por la imposibilidad de la presencia física en la UPV para poder instalar el *hardware* que necesita: televisión, soporte, Raspberry Pi, etc. Este hecho ha provocado que la funcionalidad de llamada del gestor al solicitante desde la aplicación para atenderlo, y que luego aparezca la llamada en el panel informativo, haya sido también descartada.
- Módulo de solicitud de atención inmediata. Este componente no ha sido implantado tampoco por la imposibilidad de la presencia física en la UPV para poder instalar el *hardware* que necesita: RasPad, soporte, lector de tarjetas, etc. Además, la utilización de un dispositivo táctil para su uso queda desaconsejado a día de hoy por el peligro de contagio del virus.
- *App* de solicitud de cita previa. Como se ha comentado anteriormente, este componente de la aplicación aún no ha sido desarrollado por temas de planificación de proyectos en el ASIC.

De esta manera, y a día de hoy, estos son los tipos de citas previas que han sido implantados en la UPV:

- Gestión Defensa y Tesis, de la Escuela de doctorado.
- Relaciones Internacionales y con Empresa, de la Escuela de doctorado.
- Secretaria, de la Escuela de doctorado.
- Cita Registro, del Registro General.
- Oficina de Correos – Cita Oficina de Correos.
- Escuela Politécnica Superior de Alcoy – Secretaría.
- Escuela Politécnica Superior de Gandía – Secretaría.
- Escuela Técnica Superior de Ingeniería de Edificación – Secretaría.

A modo de ejemplo, en las figuras 26, 27 y 28 podemos apreciar el proceso de solicitud de la cita previa Gestión Defensa y Tesis que se ha implantado, accediendo desde la web de la Escuela de doctorado. Faltarían las figuras relativas a la pantalla del Justificante de Cita y a la Notificación de la cita por Email, pero no se han puesto para no tener que solicitar una cita real en explotación. Debido al mero carácter informativo de este tipo de cita y a las restricciones de presencialidad, este tipo de cita de momento se atiende de momento de forma telemática con la herramienta Teams.



Figura 26. Enlace de acceso a poli[Cita] desde la web de la Escuela de Doctorado.

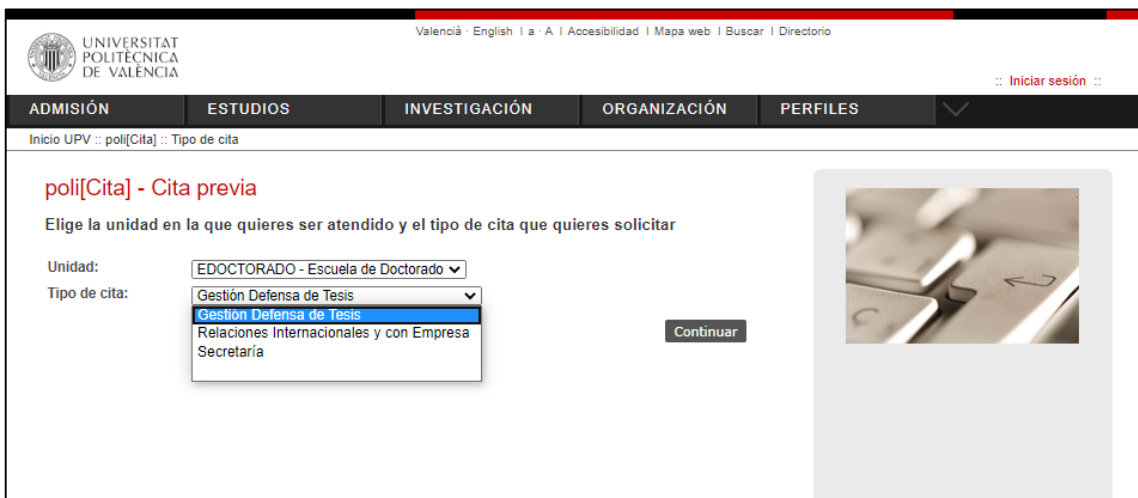


Figura 27. Selección del tipo de cita de la Escuela de doctorado.

Solicitud de cita previa

EDOCTORADO - Cita previa - Gestión Defensa de Tesis

Debido a la alerta sanitaria, esta cita no podrá tener carácter presencial en la Escuela de Doctorado. Las citas se atenderán por videoconferencia mediante la plataforma TEAMS (1).

Turno* UGR62

Nombre RAUTGAR - Utiel García, Raúl

Tema (2)* Evaluación y defensa de las tesis doctorales

Asunto* Proceso de evaluación de la tesis

Descripción* Me gustaría que me explicarais con detalle los criterios de evaluación de la tesis doctoral

Fecha de la cita*

Seleccione fecha y hora

Seleccione la fecha deseada en el calendario:

Puede desplazarse al mes siguiente y al anterior con las flechas de la parte superior del calendario.

Legenda:

No disponible Día seleccionado

Día libre Hoy

Seleccione la hora deseada en la lista:

Las horas deshabilitadas no están disponibles o ya han sido reservadas por otros usuarios.

Si no quedan horas libres marque el siguiente día libre en el calendario.

Junio 2020						
L	M	X	J	V	S	D
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Primera cita disponible

Reservas

Viernes 5 de Junio de 2020

- 11:00 - libre
- 11:20 - libre
- 11:40 - libre
- 12:00 - libre
- 12:20 - libre
- 12:40 - libre
- 13:00 - libre
- 13:20 - libre

Acceptar Cancelar

(1) Tu solicitud de cita se asignará a una persona de la Escuela de Doctorado en un plazo máximo de 1 día hábil. Una vez asignada, recibirás un correo electrónico con un enlace a la reunión programada. Si no recibes este correo en el plazo indicado contacta con nosotros (escueladoctorado@upv.es).

(2) Elige correctamente el tema para ser atendido por la persona más adecuada.

Figura 28. Formulario de alta de la cita Gestión Defensa de Tesis de la Escuela de Doctorado.

Por último, en breve serán implantados los tipos de citas Secretaria, y Prácticas e Intercambio Académico para todas las escuelas y facultades de la UPV.

9. Conclusiones

En este capítulo se recogen una serie de conclusiones finales tras la participación en este proyecto.

Tal y como se vio en el capítulo anterior, a día de hoy la aplicación no se ha podido implantar con funcionalidad plena en producción por causas de fuerzas mayor. No obstante, el desarrollo de estas funcionalidades ya fue validado por los usuarios finales, y se pueden implantar en el momento en que se requiera y se pueda. Por lo tanto, se puede afirmar que se ha conseguido el objetivo principal del proyecto, poder ofrecer un servicio de atención de cita previa y atención inmediata de calidad tanto a miembros de la universidad como a personas externas. A modo de ejemplo de estadística de uso, a día de hoy hay ya registradas 316 citas¹ del tipo de cita Cita Registro, del Registro General.

No obstante, han sido varias las dificultades encontradas a lo largo del camino, alguna de índole burocrática y la mayor parte de ellas de índole técnica. Respecto a las dificultades burocráticas, éstas han sido principalmente debidas a la composición de un equipo grande de trabajo para la ejecución de diferentes tareas del proyecto. Esto ha originado que en algunos casos se hayan generado retrasos por dependencias entre tareas que no se podían solapar en el tiempo. También podemos mencionar los retrasos originados por las autorizaciones de compra del *hardware* necesario para hacer pruebas de integración (televisión, RasPad, etc.), y por el tiempo de llegada del mismo.

Respecto a las dificultades de índole técnica, la mayoría han sido debidas a temas de integración entre distintos componentes de la aplicación. La falta de experiencia lidiando con este tipo de problemas en proyectos previos ha producido en muchos casos más dificultad y retraso del deseado a la hora de solucionarlos. Se han encontrado las siguientes dificultades técnicas:

- Navegador de la televisión para visualizar el panel informativo de llamadas. El modelo de televisión que se adquirió llevaba un navegador propio, y no era posible instalar otro en caso de que hubiera algún contratiempo relacionado con el mismo a la hora de mostrar el panel informativo de llamadas. En este sentido surgieron los siguientes problemas.
 - Durante el desarrollo del panel informativo de llamadas se simuló en el navegador las posibles resoluciones que pudiera tener la televisión que fuera a mostrarlo (4K, FHD, y HD), y que aún estaba pendiente de elegirse y adquirir su modelo. A pesar de que la televisión que finalmente se adquirió tenía una de esas tres resoluciones, la 4K, cuando el panel informativo de llamadas se visualizaba en ella, no se ajustaban exactamente las proporciones de todos los elementos tal y como se veía en el navegador utilizado en el desarrollo. Esto obligó a hacer ciertos ajustes a medida de los ficheros CSS para que se viera tal y como se esperaba.
 - Por otro lado, el navegador de la televisión tenía problemas a la hora de reproducir el aviso sonoro de las nuevas llamadas. A la tercera o cuarta nueva llamada dejaba de escucharse el aviso. Esto era debido a que su *garbage collector* no era lo suficientemente rápido a la hora de liberar de memoria los objetos AJAX que se creaban cada segundo para comprobar si había llamadas nuevas. El código se optimizó al máximo posible para intentar solventar este problema, pero después de ello, lo único que se logró fue que se pudiera aumentar el nº de reproducción de avisos a ocho o nueve. Finalmente se optó por conectar a la televisión una Raspberry Pi con Chromium, un navegador más estándar que no da ese tipo de

¹ El tipo de cita Cita Registro, del Registro General fue implantado el 01/06/2020. El dato estadístico de uso se obtuvo el 26/06/2020.

problemas. Aunque eso solventó dicho problema, también se forzó la reproducción de la televisión a resolución FHD para que no hubiera saltos en la reproducción del video del panel informativo de llamadas, ya que el procesador gráfico de la Raspberry Pi no era lo suficientemente potente para reproducción de video en 4K.

Una experiencia previa en un proyecto similar, hubiera servido para darse cuenta de que comprar un dispositivo al que no le puedes instalar un navegador distinto al suyo propio es un gran error. Si se hubiera adquirido una televisión a la que se le hubiera podido haber instalado otro navegador distinto (como una televisión con Android TV), se hubiera evitado directamente todos esos problemas.

- Políticas restrictivas del navegador para poder reproducir automáticamente sonido y audio. Chromium y otros navegadores, como Chrome o Mozilla, presentan desde hace un tiempo una restricción a la hora de reproducir video y audio. Esta restricción consiste en que es necesario que el usuario interactúe previamente con el navegador¹ para que se reproduzca el video o audio. Esto obligaba a que cada vez que se arrancara el monitor hubiera que hacer al menos un clic en pantalla, con los inconvenientes que esto podía suponer, como p. ej. que se olvidara hacerlo y no se reprodujeran los avisos sonoros de nuevas llamadas que hicieran los gestores. La solución se encontró cambiando la política `AutoplayAllowed`² del navegador.
- Problemas con la lectura de los carnets universitarios. Para poder leer los carnets universitarios con el lector de tarjetas conectado a la RasPad en la solicitud de atención inmediata, se utilizó como mecanismo la implementación de los servicios `Readusb.service` y `Webservice.service`, tal y como se explicó en el capítulo de implementación de la solución. Este mecanismo, que *a priori* puede no parecer presentar dificultades, originó los siguientes problemas:
 - Problema de CORS (Cross-origin resource sharing) [5]. Los problemas de CORS se producen cuando se realizan peticiones AJAX desde una página web a un servicio web que se encuentra en un dominio distinto al de la propia página web, en cuyo caso dichas peticiones son bloqueadas por el navegador. En este caso el problema de CORS se presentó cuando desde la página web en la cual se hace la lectura del carnet, con dominio `intranet.upv.es`, se hacía la petición AJAX al servicio web `Webservice.service` de la RasPad, con dominio `localhost`. Para evitar esto se hizo una pasarela, de tal forma que la petición AJAX ya no se hace directamente al servicio web de la RasPad, sino a un servicio web con dominio `intranet.upv.es` (el mismo que el de la página) que hace de pasarela, y que es el que finalmente llama al servicio web `Webservice.service` a través de su IP externa.
 - Problema de sincronización de servicios. Que el servicio web `Webservice.service` tuviera que consultar a su vez a otro servicio web para obtener los datos del propietario del carnet, hacía que en ocasiones estos datos todavía no estuvieran disponibles y se devolvieran vacíos. La solución a esto pasa por que se sincronicen ambos servicios, es decir, asegurarse que el servicio `Webservice.service` no devuelva los datos del propietario del carnet hasta que realmente los haya podido obtener del otro servicio web que invoca.

¹ El detalle del cambio de la política de reproducción automática en Chrome puede verse en <https://developers.google.com/web/updates/2017/09/autoplay-policy-changes#chrome-enterprise-policies>

² La política `AutoplayAllowed` de Chrome puede cambiarse tal como se describe en <https://cloud.google.com/docs/chrome-enterprise/policies/?policy=AutoplayAllowed>

Por último, creo que se podría haber puesto en práctica una programación más modular para los principales componentes de la aplicación. En el capítulo de implementación, se vio que el paquete SIC_POLICITA agrupaba cuatro grandes componentes de la aplicación: el portal del solicitante, el portal del gestor, el panel informativo de llamadas, y la solicitud de atención inmediata. En el caso de que se hubiera separado cada componente en un paquete distinto, se hubiera podido aislar la parte del solicitante de la parte del gestor. De esta manera, si p.ej. ocurriera un error puntual en el paquete del portal del solicitante y éste se descompilara, los solicitantes no podrían pedir temporalmente citas a través del portal web hasta que se resolviera dicho error. En cambio, sí que podrían seguir pidiendo citas de atención inmediata, y, además, los gestores podrían seguir llamando a solicitantes de citas previas ya existentes o de citas de atención inmediata nuevas, mostrándose todas ellas en el panel informativo de llamadas. De la manera actual, estando todos los componentes en el mismo paquete, si éste se descompilara, dejarían de funcionar todos ellos.

10. Trabajos futuros

En este capítulo se va a detallar funcionalidades que se tiene previsto que tenga la aplicación, así como nuevas sugerencias de mejora.

Por un lado, está previsto el desarrollo de una *app* para los solicitantes. Este es un requisito recogido en la fase de análisis (ReqNF4), pero por temas de planificación se ha decidido abordar su desarrollo e implantación en una fase posterior al resto de la aplicación. Esta *app* ofrecerá la funcionalidad equivalente al portal del solicitante para la solicitud y consulta de citas previas por parte de los solicitantes.

Por otro lado, hay varias sugerencias de mejora contempladas para potenciar la funcionalidad de la aplicación. Estas funcionalidades nuevas sugeridas se desarrollarán o no a futuro en función de diversos factores, como pueden ser su demanda o necesidad futura por parte de los usuarios, o su complejidad y coste. Las sugerencias de mejora que se han recogido son:

- Modificación de la fecha de la cita previa por parte de los solicitantes. Actualmente para poder cambiar la fecha de la cita, el solicitante tiene que eliminarla o anularla, y después solicitarla de nuevo con la nueva fecha deseada. Poder modificar la fecha sin tener que hacer todo esto mejoraría la usabilidad, ya que el usuario podría utilizar esta funcionalidad de una manera más rápida y eficiente
- Elección del gestor que atenderá la cita por parte del solicitante. En tipos de citas de trato más personal, como el servicio médico p. ej., podría tener sentido esta utilidad de cara al nivel de satisfacción del solicitante al usar el servicio.
- Impresión de un tique con el identificador en la solicitud de atención inmediata. Actualmente cuando se solicita atención inmediata, el identificador con el que se llamará al solicitante se muestra por pantalla al final del proceso de solicitud y se envía por correo electrónico al solicitante. Esto se ha decidido así por la política de “no papel” de la UPV. No obstante, habrá que ver si a futuro esto no genera inconvenientes, como p. ej. que la persona no lleve un teléfono móvil encima donde poder consultar el mail con el identificador en caso de que se le olvide.
- Mecanismo de sanción a los solicitantes de citas previas que no se presenten. Esta funcionalidad ya estaba presente en la aplicación ya existente de cita previa que se decidió adaptar como solución. Con este mecanismo se puede configurar para un tipo de cita un número de días de sanción en los cuales el solicitante no podrá volver a solicitar una cita previa de ese tipo, en caso de que se marque una cita suya de ese tipo como no presentado. Esta funcionalidad no es un requisito de la aplicación, y no se han configurado días de sanción para los diferentes tipos de citas. No obstante, es posible que a futuro se requiera este mecanismo de sanciones para algunos tipos de citas y haya que configurarlo.
- Configuración de rangos horarios de atención por gestor. Actualmente los rangos horarios de atención se configuran por tipo de cita, y es el mismo para todos los gestores que atienden el tipo de cita. Este modelo de configuración de la atención parece más que suficiente, pero es posible que en algún caso se quiera personalizar los horarios de atención de determinados gestores. Hay que tener en cuenta que esta configuración tendría una lógica mucha más compleja a nivel de implementación, y por tanto tendría un impacto importante en la aplicación.
- Utilidades de informes y estadísticas. Esto no se ha definido como requisito de la aplicación y no se ha implementado, pero en caso de que a futuro se pidan informes o estadísticas, habrá que exportar la información directamente desde la base de datos. Esto

puede ser algo no viable a largo plazo en caso de que se solicite mucho, ya que consume tiempo de los responsables de la aplicación para cada petición. En este sentido, es muy probable que se acabe desarrollando alguna utilidad de informes y estadísticas para ser utilizada directamente por los usuarios que lo necesiten. Estas utilidades podrían ser:

- Una pantalla de consulta con diferentes filtros para visualizar las citas que se han generado y que sean exportables a otros formatos.
- Diversos informes ya configurados, y que sean generables por el usuario de acuerdo a diferentes filtros.
- Gráficos que totalicen estadísticas de uso de acuerdo a diferentes criterios.

11. Referencias

- [1] «Software propietario,» Wikipedia, la enciclopedia libre, [En línea]. Disponible en: https://es.wikipedia.org/wiki/Software_propietario. [Último acceso: Junio 2020].
- [2] «Software libre,» Wikipedia, la enciclopedia libre, Junio 2020. [En línea]. Disponible en: https://es.wikipedia.org/wiki/Software_libre.
- [3] «Software como servicio,» Wikipedia, la enciclopedia libre, Junio 2020. [En línea]. Disponible en: https://es.wikipedia.org/wiki/Software_como_servicio.
- [4] «Unified Modeling Language,» Object Management Group, Diciembre 2017. [En línea]. Disponible en: <https://www.omg.org/spec/UML/2.5.1/PDF>. [Último acceso: Abril 2020].
- [5] «Modelo de estimación de proyectos,» Área de Sistemas de Información y Comunicaciones -UPV, [En línea]. Disponible en: <https://wiki.upv.es/confluence/download/attachments/101089343/Anexo%20B%20bis%20Modelo%20estimaci%C3%B3n%20de%20proyectos%20vf.docx?version=1&modificationDate=1484727376000&api=v2>. [Último acceso: Mayo 2020].
- [6] «Developing PL/SQL Web Applications,» Oracle and/or its affiliates, [En línea]. Disponible en: https://docs.oracle.com/database/121/ADFNS/adfn_web.htm#ADFNS1101. [Último acceso: Abril 2020].
- [7] J. Zuriaga, «Estándar Sistemas Multivista,» Área de Sistemas de Información y Comunicaciones - UPV, 30 Enero 2007. [En línea]. Disponible en: https://wiki.upv.es/confluence/download/attachments/1706483/manual_e3.pdf?version=1&modificationDate=1426237632000&api=v2. [Último acceso: Marzo 2020].
- [8] «Estándar Bases de Datos,» Área de Sistemas de Información y Comunicaciones - UPV, [En línea]. Disponible en: https://wiki.upv.es/confluence/download/attachments/1706444/estandar_basedatos.pdf?version=1&modificationDatehttps://wiki.upv.es/confluence/download/attachments/1706444/estandar_basedatos.pdf?version=1&modificationDate=1426237582000&api=v2. [Último acceso: Mayo 2020].
- [9] L. C., «Control de acceso HTTP (CORS),» Mozilla and individual contributors, 2 Octubre 2015. [En línea]. Disponible en: https://developer.mozilla.org/es/docs/Web/HTTP/Access_control_CORS. [Último acceso: Junio 2020].

Para acceder a los contenidos del Área de Sistemas de Información y Comunicaciones es necesario estar autenticado en la intranet de la UPV y pedir permiso de acceso a rmengod@cc.upv.es.

12. Anexos

En este capítulo se detallan y explican los anexos referenciados a lo largo del texto del documento.

12.1. Anexo I. Ejemplo de uso del estándar multivista.

En este anexo podemos ver un ejemplo de uso del estándar multivista [3] para implementar la pantalla inicial del portal del solicitante de poli[Cita]. La parte de código que construye la estructura de la página es casi idéntica al de la plantilla de ejemplo del estándar multivista¹, que son las líneas correspondientes a las llamadas al paquete sic_upve del estándar. El resto de código adicional tiene el propósito de implementar la funcionalidad y contenido particular de la propia página.

¹ La plantilla de ejemplo del estándar multivista y su código se puede ver en http://www.upv.es/pls/doalu/stdes.MV_Plantilla?P_IDIOMA=c&P_VISTA=


```

1  PROCEDURE Bienvenida (P_IDIOMA in varchar2 default null, P_VISTA in varchar2 default
2  null, P_ORIGEN in varchar2 default null) IS
3  --
4  l_vista varchar2(20) default 'normal';
5  l_id varchar2(1) default g_id;
6  -- Variables control acceso
7  l_nip          varchar2(10) default null;
8  l_tickets      varchar2(4000) default null;
9  l_autenticado  boolean default true;
10 -- Uso temporal
11 l_msg_error    varchar2(500);
12 l_texto        varchar2(4000);
13 l_texto_movil  varchar2(4000);
14 l_text_anuncio cp_conf_citaprevia.valor_c&type
15 l_text_bienvenida cp_conf_citaprevia.valor_c&type
16 l_enlace_manual cp_conf_citaprevia.valor_c&type
17 l_text_notas_legal cp_conf_citaprevia.valor_c&type
18 l_text_enlace   cp_conf_citaprevia.valor_c&type
19 l_url_solicitar  varchar2(1000);
20 l_url_consultar  varchar2(1000);
21 l_url_gestionar  varchar2(1000);
22 l_url           varchar2(1000);
23 l_acceso_activo boolean:= acceso_activo;
24 BEGIN
25     if P_VISTA is not null then
26         l_vista:=P_VISTA;
27     end if;
28     --
29     if P_IDIOMA is not null then
30         l_id:=lower(P_IDIOMA);
31     end if;
32     --
33     if not autenticado(l_nip) then
34         l_autenticado:= false;
35     end if;
36     --
37     --validaciones previas para los gestores
38     --
39     if P_ORIGEN = g_P_ORIGEN_GESTOR then
40         if not l_autenticado then --validamos que esté autenticado
41             sic upve.error(MSG_ERROR=>est upv.traduce('Per a gestionar les cites de la
42             teua unitat has de connectar-te en la Intranet de la UPV.', 'Para gestionar
43             las citas de tu unidad debes conectarte en la Intranet de la UPV.',l_id,
44             'To manage the dates of your unit
45             have to connect you in the UPV
46             Intranet.'),
47             P_VISTA=>l_vista, P_IDIOMA=>l_id);
48         end if;
49     end if;
50     --
51     --Implementamos la pantalla
52     --
53     sic upve.InicializaVista(P_VISTA=>l_vista, config=>g_config);
54     --Documento
55     sic upve.Cab Documento(P_IDIOMA=>l_id,
56     config=>g_config,
57     P_TITULO=>est upv.traduce('Accés a poli[Cita]','Acceso a
58     poli[Cita]',l_id,'poli[Cita] access'));
59     Y_Inicializame(P_IDIOMA=>l_id, P_VISTA=>l_vista);
60     --
61     Nav_Cita_Previa(l_vista, l_id, P_ORIGEN);
62     sic upve.Cab Pagina(P_IDIOMA=>l_id,
63     P_VISTA=>l_vista,
64     P_TITULO=>est upv.traduce('poli[Cita]','poli[Cita]',l_id,'poli[C
65     ita]'));
66     sic upve.Cab Container(P_SUBTITULO=>null, P_H2CLASS=>null);
67     --
68     --obtenemos los textos de los campos configurables Anuncio, Texto Bienvenida,
69     Manual de usuario, Soporte y Nota Legal
70     --

```

```

obtener_texto_campos_conf(P_ORIGEN,l_id,l_text_anuncio,l_text_bienvenida,l_enlace_ma
65 nual,l_text_notas_legal);
66 --
67 --texto anuncio y texto de bienvenida
68 --
69 sic_upve.Separador;
70 --
71 if (l_text_anuncio <> ' ') then
72 sic_upve.Cab_ContainerObj(P_TIPO=>'upv_listadesc');
73 if l_acceso_activo then
74 sic_upve.LisDesc_Elemento(P_DESC=>'<div
75 class="anuncio">'||l_text_anuncio||'</div>');
76 else
77 sic_upve.LisDesc_Elemento(P_DESC=>'<div
78 class="anuncio">'||l_text_anuncio||'</div>',
79 P_IMAGEN=>'');
81 end if;
82 sic_upve.Pie_ContainerObj(P_TIPO=>'upv_listadesc');
83 end if;
84 --
85 sic_upve.Cab_ContainerObj(P_TIPO=>'upv_listadesc');
86 sic_upve.LisDesc_Elemento(P_DESC=>l_text_bienvenida,
87 P_IMAGEN=>'');
89 sic_upve.Pie_ContainerObj(P_TIPO=>'upv_listadesc');
90 --
91 sic_upve.Separador;
92 --
93 --botones de acceso a poli[Cita]
94 --
95 if l_acceso_activo then
96 sic_upve.Cab_ContainerObj(P_TIPO=>'upv_formulario');
97 sic_upve.Frm_CabBotonera;
98 --
99 if P_ORIGEN=g_P_ORIGEN_SOLICITANTE then
100 if l_autenticado then
101 l_url_consultar:='sic frm v2.ConsultaSSP?P_AGRUPADOR=poliCita&P_VISTA='||l_vista
102 ||'&P_IDIOMA='||l_id;
103 else l_url_consultar:=
104 'sic frm v2.FrmSSP?P_VISTA='||l_vista||'&P_IDIOMA='||l_id;
105 end if;
106 --
107 l_texto:= est_upv.traduce('Sol·licitar cita','Solicitar cita',l_id,'Request an
108 appointment');
109 l_url_solicitar:=
110 'SIC_POLICITA.Seleccion Cita?P_VISTA='||l_vista||'&P_IDIOMA='||l_id;
111 http.p('<a href="||l_url_solicitar||'" class="upv_btlink"
112 title="||l_texto||">'||l_texto||'</a>');
113 l_texto:= est_upv.traduce('Consultar les meues cites','Consultar mis
114 citas',l_id,'View my appointments');
115 http.p('<a href="||l_url_consultar||'" class="upv_btlink"
116 title="||l_texto||">'||l_texto||'</a>');
117 elsif P_ORIGEN=g_P_ORIGEN_GESTOR then
118 l_texto:= est_upv.traduce('Gestionar cites','Gestionar citas',l_id,'Manage
119 appointments');
120 l_url_gestionar:=
121 'SIC_POLICITA.Seleccion Cita_Gestor?P_VISTA='||l_vista||'&P_IDIOMA='||l_id;
122 http.p('<a href="||l_url_gestionar||'" class="upv_btlink"
123 title="||l_texto||">'||l_texto||'</a>');
124 end if;
125 --
126 sic_upve.Frm_PieBotonera;
127 sic_upve.Pie_ContainerObj(P_TIPO=>'upv_formulario');
128 end if;
129 --
130 --enlaces manual de usuario y soporte, y nota legal
131 --
132 sic_upve.Cab_ContainerObj(P_TIPO=>'upv_texto');
133 sic_upve.Separador;
134 sic_upve.Separador;
135 http.p('<br><br><br><br><br><br>'||l_enlace_manual);
136 sic_upve.Pie_ContainerObj(P_TIPO=>'upv_texto');

```



```

121  --
122  l_texto:= est_upv.traduce('amb l''aplicació poli[Cita]', 'con la aplicación
poli[Cita]', l_id, 'with the poli[Cita] application');
123  l_text_enlace:= est_upv.traduce('Incidències tècniques', 'Incidencias
técnicas', l_id, 'Technical issues');
124  l_url:= obtener_param_config('INCIDENCIA');
125  l_texto_movil:= l_text_enlace||' '||l_texto;
126  l_texto:= '<a href=""||l_url||"' target=" blank"
title=""||l_text_enlace||"'>'||l_text_enlace||'</a> '||l_texto;
127  --
128  sic_upve.Cab_ContainerObj(P_TIPO=>'upv_texto');
129  if P_VISTA = 'movil' then htp.p(l_texto_movil);
130  else htp.p(l_texto);
131  end if;
132  sic_upve.Pie_ContainerObj(P_TIPO=>'upv_texto');
133  --
134  if P_ORIGEN = g_P_ORIGEN_SOLICITANTE then
135    sic_upve.ContainerNota(P_TXT_NOTA=>l_text_noto_legal);
136  end if;
137  --
138  sic_upve.Pie_Container;
139  sic_upve.Pie_Pagina(P_IDIOMA=>l_id, P_VISTA=>P_VISTA);
140  sic_upve.Pie_Documento(P_VISTA=>P_VISTA);
141  exception
142  when errTicket then
143    --raise errTicket; -- Si soy un procedimiento que se puede llamar desde URL
comentar esta linea
return;
144  when others then
145    l_msg_error:= est_upv.traduce('S''ha produït una excepció no controlada.', 'Se ha
producido una excepción no controlada.', l_id)||':
'||sqlexrm||'<br/><br/>'||DBMS_UTILITY.format_error_backtrace;
147    sic_upve.error(MSG_ERROR=>l_msg_error, P_VISTA=>l_vista, P_IDIOMA=>l_id);
148    return;
149  END Bienvenida;

```

12.2. Anexo II. Ejemplo de casos de pruebas funcionales.

En este anexo se muestra una tabla con el conjunto de casos de pruebas funcionales que se definieron para validar el correcto funcionamiento del módulo de solicitud de atención inmediata. Los resultados de las pruebas han sido todos satisfactorios tras algunas revisiones del código.

Prueba	Condiciones / Acciones	Resultado esperado	Resultado prueba
Selector de idioma	Nos encontramos en cualquier pantalla salvo la de error.	El selector de muestra los iconos de los idiomas castellano, valenciano e inglés. Al pulsar sobre ellos la pantalla se traduce al idioma correspondiente.	Correcto
Pantalla de inicio.	Existen citas configuradas en la unidad.	La pantalla debe tener un aspecto similar a la del panel informativo de llamadas en cuanto a fuentes, colores, etc. El contenido de los botones de los tipos de citas debe distribuirse de forma simétrica y proporcional en la pantalla.	Correcto
Pantalla de inicio.	No existe ninguna cita configurada en la unidad.	No aparece ningún botón de tipo de cita.	Correcto

Pantalla de inicio.	Existen cuatro o menos citas configuradas en la unidad.	Aparece una única columna de botones de los tipos de citas.	Correcto
Pantalla de inicio.	Existen más de cuatro citas configuradas en la unidad.	Aparecen dos columnas de botones de los tipos de citas.	Correcto
Pantalla de inicio.	Hay más de ocho tipos de citas configuradas.	A partir del noveno tipo de cita inclusive, los tipos de citas se descartan y no se muestran sus botones correspondientes.	Correcto
Pantalla de inicio.	Hay citas configuradas con rango horario que no incluye la hora actual.	Las citas con rango horario de atención que no incluyen la hora actual no se muestran en pantalla.	Correcto
Pantalla de inicio.	Hay tipos de citas no activos.	Los tipos de citas no activos no se muestran en pantalla.	Correcto
Pantalla de inicio.	Se pulsa el botón de un tipo de cita.	Se muestra la pantalla de confirmación de cita correspondiente al tipo de cita del botón pulsado.	Correcto
Pantalla de confirmación de tipo de cita.		Muestra correctamente el número de personas por delante a ser atendidas para el grupo de ese tipo de cita.	Correcto
Pantalla de confirmación de tipo de cita.	Se pasa el carnet universitario por el lector de tarjetas. Después se pulsa el botón de tomar turno.	Al pasar el carnet se muestra el nombre del propietario del mismo en el campo de nombre. Al pulsar el botón de tomar turno se genera la cita, se envía <i>email</i> de notificación de la misma a la dirección de correo electrónico del propietario de la tarjeta, y se muestra la pantalla de confirmación de envío de <i>email</i> .	Correcto
Pantalla de confirmación de tipo de cita.	No se pasa el carnet universitario por el lector de tarjetas, y se introduce manualmente el nombre en el campo. Después se pulsa el botón de tomar turno.	Al pulsar el botón de tomar turno se genera la cita, y se muestra la pantalla de asignación de turno.	Correcto
Pantalla de asignación de turno.		Se muestra el identificador de turno asignado para la cita generada. Muestra correctamente el número de personas por delante a ser atendidas para el grupo de ese tipo de cita.	Correcto
Pantalla de asignación de turno.	Se introduce el <i>email</i> en el campo <i>email</i> y después se pulsa el botón enviar.	Se envía <i>email</i> de notificación de la cita a la dirección de correo introducida, y se muestra la pantalla de confirmación de envío de <i>email</i> .	Correcto
Pantalla de confirmación		Se muestra la hora de llegada del solicitante de la cita. Se muestra el mensaje de confirmación de envío de	Correcto

de envío de <i>email</i> .		<i>email</i> a la dirección de correo electrónico del solicitante con su identificador de turno asignado para la cita.	
Pantalla mensaje de error.	Se produce algún error de validación de lógica o de excepción no controlada.	Se presenta un texto con una descripción del error producido que es comprensible para el usuario.	Correcto
Notificaciones por correo electrónico.		Se generan en el idioma en el que se mostrara previamente la pantalla, y con el texto y formato correcto.	Correcto
Tiempo de inactividad.	Nos encontramos en cualquier pantalla salvo en la de inicio. Se supera el tiempo máximo de inactividad en la pantalla.	La aplicación vuelve automáticamente a la pantalla de inicio.	Correcto