



Universidad de Valladolid

E. U. DE INFORMÁTICA (SEGOVIA)

**Grado en Ingeniería Informática de Servicios y
Aplicaciones**

**eRural: Gestor de Reservas para
Casas Rurales**

Alumno: Miguel Ángel Navas Galán

Tutor: Jesús Cordobés Puertas

Trabajo de Fin de Grado

TÍTULO DEL TFG: eRural: Gestor de Reservas para Casas Rurales

TITULACIÓN: Grado en Ingeniería Informática de Servicios y Aplicaciones

AUTOR: Miguel Ángel Navas Galán

TUTOR: Jesús Cordobés Puertas

FECHA: 1 de septiembre de 2015

Índice

Contenido

Bloque I. Memoria del proyecto

Índice	3
Índice de figuras	5
Índice de tablas	8
Prólogo	10
1. Introducción	12
1.1. Identificación del proyecto.....	12
1.2. Organización de la memoria	12
1.3. Estructura del CD.....	13
2. Descripción general del proyecto.....	13
2.1. Planteamiento.....	13
2.2. Objetivos	13
2.3. Metodología de desarrollo.....	14
2.4. Entorno tecnológico	15
3. Descripción general del producto.....	17
4. Planificación y presupuesto.....	18
4.1. Estimación del esfuerzo	18
4.1.1. Modelo COCOMO.....	18
4.1.2. Modelo de estimación por analogía desde la experiencia	26
4.2. Estimación de los recursos	27
4.3. Presupuesto	28
4.4. Seguimiento del calendario	31
5. Análisis de requisitos del sistema.....	36
5.1. Introducción	36
5.2. Objetivos del sistema	36
5.3. Catálogos de requisitos del sistema.....	38
5.3.1. Requisitos de información.....	39
5.3.2. Requisitos funcionales.....	44
5.3.3. Requisitos no funcionales.....	58
5.4. Matriz de rastreabilidad objetivos / requisitos	60

5.5. Resumen	62
5.6. Estudio de viabilidad.....	63
6. Análisis del sistema.....	65
6.1. Modelo estático del sistema	65
6.1.1. Diagrama de tipos de objetos	65
6.2. Modelo de comportamiento del sistema.....	67
6.2.1. Diagramas de secuencia del sistema	67
6.2.2. Diagrama de actividad.....	74
7. Diseño del sistema.....	76
7.1. Arquitectura del Sistema	76
7.2. Diagrama de despliegue	77
7.3. Diseño de subsistemas y componentes.....	78
7.4. Modelo de datos	79
7.5. Diseño de clases	81
7.6. Diseño de Interfaces y perfiles de usuario.....	96
8. Implementación del sistema	100
8.1. Introducción	100
8.2. Implementación con el calendario de Google	101
8.3. Seguridad.....	102
8.4. Problemas surgidos en la fase de implementación.....	102
9. Pruebas del sistema	103
9.1. Introducción	103
9.2. Pruebas de Integración.	104
9.3. Pruebas de cumplimiento de requisitos	104
10. Conclusiones y líneas futuras	108
10.1. Conclusiones	108
10.2. Líneas futuras	108
11. Bibliografía.....	110

Bloque III. Manuales

Manual de usuario.....	113
Manual de instalación.....	125

Índice de figuras

Figura 1	Modelo en Cascada Realimentado	14
Figura 2	Arquitectura Android	16
Figura 3	Listado de tareas del proyecto	32
Figura 4	Listado de tareas del proyecto con diagrama Gantt detallado	32
Figura 5	Informe general del proyecto	33
Figura 6	Diagrama de casos de uso nivel 1	46
Figura 7	Diagrama de casos de uso para la gestión de la configuración de la aplicación.	47
Figura 8	Diagrama de casos de uso para la gestión de reservas de la aplicación	47
Figura 9	Diagrama de casos de uso para la visualización de clientes de la aplicación	48
Figura 10	Diagrama de casos de uso para la anulación de reservas de la aplicación	48
Figura 11	Diagrama de tipos de objetos general de la aplicación	65
Figura 12	Diagrama de tipos de objetos para la configuración de la aplicación	66
Figura 13	Diagrama de tipos de objetos para la acción de reservar de la aplicación	66
Figura 14	Diagrama de tipos de objetos para la acción de anular reserva de la aplicación	67
Figura 15	Diagrama de secuencia para el UC-1	68
Figura 16	Diagrama de secuencia para el UC-2	68
Figura 17	Diagrama de secuencia para el UC-3	69
Figura 18	Diagrama de secuencia para el UC-4	69
Figura 19	Diagrama de secuencia para el UC-5	70
Figura 20	Diagrama de secuencia para el UC-6	70
Figura 21	Diagrama de secuencia para el UC-7	71
Figura 22	Diagrama de secuencia para el UC-8	71
Figura 23	Diagrama de secuencia para el UC-9	72
Figura 24	Diagrama de secuencia para el UC-10	72
Figura 25	Diagrama de secuencia para el UC-11	73
Figura 26	Diagrama de secuencia para el UC-12	73
Figura 27	Diagrama de secuencia para el UC-13	74
Figura 28	Diagrama de actividad de la aplicación	75
Figura 29	Arquitectura lógica del sistema.	77
Figura 30	Diagrama de despliegue del sistema.	78
Figura 31	Subsistemas / Componentes.	78

Figura 32	Modelo de datos para la aplicación.	80
Figura 33	Diagrama de clases para el registro en la aplicación.	82
Figura 34	Diagrama de clases de la base de datos.	83
Figura 35	Diagrama de clases de eRuralGestorBD.	84
Figura 36	Diagrama de clases de Gestor.	84
Figura 37	Diagrama de clases de Casa.	84
Figura 38	Diagrama de clases de Habitaciones.	85
Figura 39	Diagrama de clases de Características.	85
Figura 40	Diagrama de clases de Reservas.	85
Figura 41	Diagrama de clases de Clientes.	86
Figura 42	Diagrama de clases de Pagos.	86
Figura 43	Diagrama de clases de la principal a la configuración.	87
Figura 44	Diagrama de clases de la configuración de la casa.	88
Figura 45	Diagrama de clases de la configuración de las habitaciones.	88
Figura 46	Diagrama de clases de la configuración de las características de las habitaciones.	89
Figura 47	Diagrama de clases de la configuración de la cuenta de correo.	90
Figura 48	Diagrama de clases de la configuración del pago.	91
Figura 49	Diagrama de clases de la principal a la reserva.	92
Figura 50	Diagrama de clases del subsistema reservar.	93
Figura 51	Diagrama de clases del subsistema anular reserva	94
Figura 52	Diagrama de clases del subsistema visualización clientes	95
Figura 53	Content Provider Google Calendar.	96
Figura 54	Diseño pantalla principal.	97
Figura 55	Diseño menú configuración.	98
Figura 56	Diseño pantalla de Casa.	98
Figura 57	Diseño pantalla de reserva.	99
Figura 58	Diseño pantalla de anulación de reserva.	100
Figura 59	Pantalla inicial de mensaje de registro de usuario.	113
Figura 60	Pantalla inicial de registro de usuario.	114
Figura 61	Pantalla inicial.	114
Figura 62	Pantalla inicial de la configuración de la aplicación.	115
Figura 63	Pantalla de la configuración de la casa.	115
Figura 64	Pantalla de la configuración de las habitaciones.	116

Figura 65	Pantalla de la configuración de las características comunes de las habitaciones.	116
Figura 66	Pantalla de la configuración de las características de las habitaciones.	117
Figura 67	Pantalla de la configuración de la cuenta de correo.	117
Figura 68	Pantalla de la configuración del pago.	118
Figura 69	Pantalla de reserva de una habitación.	118
Figura 70	Pantalla de selección de fechas.	119
Figura 71	Pantalla de registro de características comunes de la habitación.	119
Figura 72	Pantalla de confirmación de registro de cliente.	120
Figura 73	Pantalla de registro de cliente.	120
Figura 74	Pantalla de confirmación de envío de mail al cliente desde aplicación de correo.	121
Figura 75	Pantalla de clientes.	121
Figura 76	Pantalla de anular reservas	122
Figura 77	Pantalla de confirmación de anulación de reserva	122
Figura 78	Pantalla de configuración de cuenta Exchange.	123
Figura 79	Pantalla de configuración de cuenta POP3	124
Figura 80	Pantalla de la ubicación del archivo.	126
Figura 81	Pantalla de la seguridad.	127
Figura 82	Pantalla de instalar la aplicación.	127
Figura 83	Pantalla de permisos de la aplicación	128

Índice de tablas

Tabla 1	División del esfuerzo del proyecto - COCOMO	28
Tabla 2	Presupuesto desarrollo – COCOMO	29
Tabla 3	División del esfuerzo del proyecto - Analogía	29
Tabla 4	Presupuesto desarrollo – Analogía	29
Tabla 5	Presupuesto recursos – COCOMO	30
Tabla 6	Presupuesto recursos – Analogía	30
Tabla 7	Presupuesto total – COCOMO	31
Tabla 8	Presupuesto total – Analogía	31
Tabla 9	Presupuesto total	33
Tabla 10	OBJ-1 Gestión de la configuración de la aplicación	36
Tabla 11	OBJ-1.1 Configuración de parámetros	36
Tabla 12	OBJ-1.2 Configuración de componentes	37
Tabla 13	OBJ-1.3 Configuración del pago	37
Tabla 14	OBJ-2 Gestión de las reservas	37
Tabla 15	OBJ-2.1 Gestión de habitaciones	38
Tabla 16	OBJ-2.2 Gestión calendario	38
Tabla 17	OBJ-2.3 Gestión de clientes	38
Tabla 18	IRQ-1 Información usuario gestor	39
Tabla 19	IRQ-2 Información Casa Rural	39
Tabla 20	IRQ-3 Información habitaciones	40
Tabla 21	IRQ-3.1 Información características	40
Tabla 22	IRQ-4 Información de las reservas	41
Tabla 23	IRQ-5 Información del calendario	42
Tabla 24	IRQ-6 Información del cliente	42
Tabla 25	IRQ-7 Información del pago	43
Tabla 26	CRQ-1 Único usuario gestor	43
Tabla 27	CRQ-2 Reserva única en la mismas fechas	44
Tabla 28	CRQ-3 Número de tarjeta correcto	44
Tabla 29	REQ-1 El usuario configura la Casa Rural	44
Tabla 30	REQ-2 Reservar una habitación	45
Tabla 31	REQ-3 Anular reserva de una habitación	45
Tabla 32	REQ-4 Visualizar clientes	45
Tabla 33	ACT-1 Gestor	46
Tabla 34	UC-1 Identificar al usuario	48
Tabla 35	UC-2 Configuración de la casa	49
Tabla 36	UC-3 Configuración de las habitaciones	50
Tabla 37	UC-4 Configuración de las características	51
Tabla 38	UC-5 Configuración de las características comunes	51
Tabla 39	UC-6 Configuración de los componentes	52
Tabla 40	UC-7 Configuración del pago	53
Tabla 41	UC-8 Gestión de las habitaciones	54

Tabla 42	UC-9 Gestión del Calendario	54
Tabla 43	UC-10 Gestión de los clientes	55
Tabla 44	UC-11 Comunicación de la reserva	56
Tabla 45	UC-12 Gestión clientes grabados	56
Tabla 46	UC-13 Anular reserva calendario	57
Tabla 47	NFR-1 Rendimiento fluido	58
Tabla 48	NFR-2 Interfaz amigable, usable y accesible	58
Tabla 49	NFR-3 Compatibilidad entre versiones de Android	59
Tabla 50	NFR-4 Necesidad de cuenta de correo registrada	60
Tabla 51	NFR-5 Seguridad.	60
Tabla 52	Matriz de rastreabilidad.	60
Tabla 53	Resumen de requisitos.	62
Tabla 54	PCR-1A Usuario vacío.	104
Tabla 55	PCR-1B Contraseña vacía.	105
Tabla 56	PCR-1C Credenciales correctas.	105
Tabla 57	PCR-2 Nombre de la casa vacío.	105
Tabla 58	PCR-3 Nombre de la habitación vacío.	106
Tabla 59	PCR-4 Nombre de la característica vacío.	106
Tabla 60	PCR-5A Nombre del cliente vacío.	107
Tabla 61	PCR-5B Campo e-mail incorrecto.	107

Prólogo

El auge que viven los dispositivos móviles hoy en día no es ninguna noticia novedosa. Las personas ven los dispositivos —teléfonos inteligentes, tabletas multimedia, o los denominados **Android Wear** como gafas o relojes— que abren las puertas de Internet. Abren las puertas a todo tipo de información a tiempo real.

La agilización de las redes y el continuo abaratamiento de costes hardware ha provocado la aparición de nuevos y potentes dispositivos de muchos tipos, dando cabida a todo tipo de usuarios finales, incluyendo personas no habituadas en el manejo de la tecnología.

Android es un Sistema Operativo (SO) para diseñado principalmente para dispositivos móviles con pantalla táctil basado en el núcleo Linux. Android fue originado por un grupo de empresas conocido como **Open Handset Alliance**, liderada por **Google**. Hoy en día, muchas empresas - ambos miembros originales de la OHA y otros - han invertido mucho en **Android**. **Android**, al contrario que otros sistemas operativos para dispositivos móviles como **iOS** o **Windows Phone**, se desarrolla de forma abierta y se puede acceder al código fuente. **Android** es intencional y explícitamente un código abierto - a diferencia de un software libre.

El resultado es un producto con código fuente abierto para la personalización y portabilidad, perfecto para un amplio grupo de desarrolladores para esta plataforma. Hoy en día en **Google Play** (tienda de aplicaciones de **Google** para aplicaciones **Android**) se puede acceder a una infinidad de aplicaciones, se tiene la sensación de que siempre hay una aplicación para *eso*: una aplicación de software que normalmente se puede descargar en cuestión de minutos, comprar y pagar en línea y almacenar (o no) en el dispositivo móvil para su uso futuro.

De este mercado en continua evolución debido a la masiva confluencia de desarrolladores y clientes surge la idea de hacer una aplicación en **Android** para la gestión de casas rurales, una aplicación para la propia gestión personal de un negocio al alcance de la mano, sin engorros de calendarios ni cuadernos para hacer algo prácticamente obsoleto: llevar una gestión apuntada a mano.

Bloque I - Memoria del proyecto

1. Introducción

El presente documento recoge toda la información del desarrollo llevado a cabo para el trabajo final del grado. Este proyecto surge a partir de la motivación de desarrollar una aplicación para una plataforma móvil que permita al usuario poder gestionar las reservas de una casa rural y a su vez permitirle acceder a herramientas muy utilizadas como son el calendario y el correo electrónico.

El proyecto se basa en la creación de una aplicación para dispositivos Android, al ser una de las plataformas con mayor ratio de crecimiento del mercado y por permitir mediante interfaces, crear todas las funcionalidades que eran necesarias para el desarrollo de esta aplicación como se podrá ver a lo largo del documento.

1.1. *Identificación del proyecto*

Título: eRural: gestor de reservas para casas rurales

Autores: Miguel Ángel Navas Galán

Tutor: Jesús Cordobés Puertas

Departamento: Organización de Empresas y CIM

1.2. *Organización de la memoria*

Vamos a dividir la documentación en tres grandes apartados:

- Bloque I: Memoria del Proyecto.

Explicaremos los antecedentes que sirvieron para el desarrollo de este proyecto. También comentaremos las técnicas, metodologías y herramientas que se han utilizado, así como la planificación de trabajo, comparando el tiempo estimado con el tiempo real.

- Bloque II: Manual Técnico.

Explicaremos la aplicación desarrollada de una manera más técnica, especificando cada uno de los componentes a diseñar, tanto estructura de clases como de base de datos.

- Bloque III: Manuales.

Como su nombre indica, tendremos varios manuales de ayuda al usuario, una para la instalación de la aplicación y otro para la “navegación por la aplicación”, explicando cada una de las funcionalidades de la misma.

1.3. Estructura del CD

A continuación mostraremos los apartados que estarán incluidos en el CD, explicando brevemente su distribución:

- Carpeta Memoria del Proyecto: contendrá la Memoria del Proyecto en formato pdf, y los archivos utilizados en proyecto.
- Carpeta Instalación: contendrá el instalador y la carpeta Código Fuente que contendrá el código fuente de la aplicación.
- Carpeta Manuales: manual de instalación y manual de usuario.

2. Descripción general del proyecto

2.1. Planteamiento

La idea de la realización de este proyecto surgió del tutor del mismo, D. Jesús Cordobés Puertas, comentando en algunas reuniones la posibilidad de desarrollar un proyecto de gestión de reservas para casas rurales en un dispositivo móvil.

En este proyecto se van a tratar gran cantidad de aspectos de programación vistos durante estos años en la carrera y muchos otros que no se habían llegado a ver, entre ellos:

1.- Programación orientada a objetos:

1.1.- Desarrollo de herencia.

1.2.- Desarrollo de componentes Java.

1.3.- Desarrollo de componentes propios de Android.

2.- Manejo de Bases de Datos SQLite, característica de Android

2.1.- Sentencias SQL.

2.2. Objetivos

El objetivo principal del proyecto es desarrollar una aplicación Android para dispositivos móviles (smartphones y tablets) para la gestión de reservas de una casa rural.

Como subobjetivos del mismo:

- Definición de la casa rural: la definición de la casa rural como tal, con sus habitaciones y sus características.
- Gestionar reservas: registrar y anular en el calendario y en base de datos las reservas de las distintas habitaciones.
- Gestión de sus clientes: a raíz de la reserva de una habitación habilitar el registro del cliente.
- Comunicación vía correo electrónico: generación de un correo automático de comunicación con el cliente.

2.3. Metodología de desarrollo

Para realizar el proyecto se ha seguido el **modelo en cascada realimentado**, que consiste en repetir el modelo en cascada hasta la fase de pruebas y repetir el ciclo hasta que el software está totalmente desarrollado.

El modelo en cascada puro difícilmente se utiliza tal cual, pues esto implicaría un previo y absoluto conocimiento de los requisitos, la no volatilidad de los mismos (o rigidez) y etapas subsiguientes libres de errores; ello sólo podría ser aplicable a escasos y pequeños desarrollos de sistemas. Esto es utópico; ya que intrínsecamente el software es de carácter evolutivo, cambiante y difícilmente libre de errores, tanto durante su desarrollo como durante su vida operativa.

Siempre se produce alguna realimentación entre etapas, que no es completamente predecible ni rígida; esto da oportunidad al desarrollo de productos software en los cuales hay ciertas incertezas, cambios o evoluciones durante el ciclo de vida.

De esta manera se obtiene un **modelo en cascada realimentado**, que puede ser esquematizado como lo ilustra la siguiente figura y es el que utilizamos:

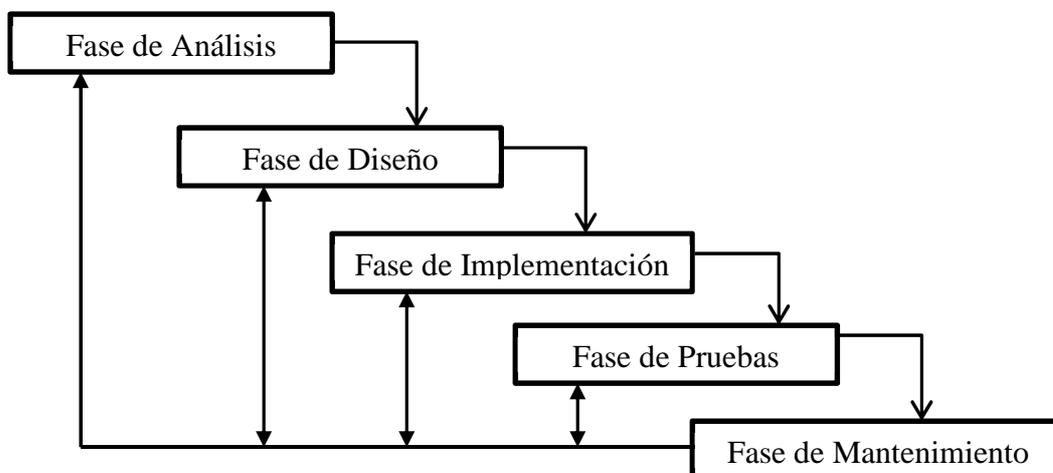


Figura 1. Modelo en Cascada Realimentado

Este ciclo de vida se divide en las etapas:

- Análisis: en esta fase se analizan las necesidades de los usuarios finales del software, obteniendo así los requerimientos y necesidades que el cliente quiere obtener una vez esté terminado. Además estudiaremos las tecnologías más adecuadas para llegar a la solución del problema de la manera más eficaz.
- Diseño: en esta fase se descompone el sistema en elementos que puedan ser desarrollados por separado. Es la descripción de la estructura relacional de todo el sistema y la especificación de lo que debe hacer cada una de sus partes. Como resultado se obtiene el Documento de Diseño del Sistema.
- Implementación: en esta fase construiremos el código de la aplicación, con las especificaciones obtenidas en las fases anteriores.
- Pruebas: en esta fase, una vez el software ya ensamblado se comprueba su correcto funcionamiento cumpliendo con los requisitos definidos inicialmente.
- Mantenimiento: esta fase, que tiene lugar después de la entrega, se mantiene el sistema para que siga funcionando.

El modelo utilizado para la programación es el de orientado a objetos. Este es el paradigma que utiliza objetos y sus interacciones para diseñar aplicaciones. Está basado en varias técnicas, entre las que se incluyen la herencia, la abstracción, el polimorfismo y el encapsulamientos, todas ellas utilizadas en el proyecto. El lenguaje utilizado será Java, orientado a objetos y el principal lenguaje en el que se realizan las aplicaciones Android.

En el aspecto de diseño, se ha utilizado el lenguaje de modelado UML, para especificar y describir objetos, métodos y procesos. Este lenguaje de modelado se utiliza para definir un sistema, para representar la realidad de una utilización de un requerimiento.

2.4. Entorno tecnológico

El proyecto se enmarca en el entorno tecnológico del software para dispositivos móviles inteligentes (smartphones o tablets).

Veamos un poco en perspectiva este entorno tecnológico actual: los teléfonos móviles han llegado a convertirse en una parte esencial en nuestras vidas, este entorno se ha visto favorecido por la gran expansión que ha sufrido el mercado y la demanda de este tipo de dispositivos. Esta expansión se ha visto favorecida, entre otras causas, por el avance tecnológico. El mercado ya no demanda móviles “básicos” que solo acepten llamadas o SMS, sino que demandan dispositivos que son capaces de funcionar con sistemas operativos que permiten la integración de software ofreciendo las mismas funcionalidades que un ordenador. Estos dispositivos permiten soportar aplicaciones totalmente personalizables aumentando sus funcionalidades hasta extremos casi

ilimitados.

Si se desea desarrollar para un dispositivo móvil es necesario conocer las plataformas de desarrollo existentes en el mercado. Actualmente las plataformas más populares en los terminales móviles son Symbian(condenada a la extinción, tendrá soporte hasta el año 2016), IOS, Android, RIM y Windows Mobile, aunque existen otras pero mucho menos utilizadas.

Dentro de este marco, reside Android, que es el sistema operativo base de este proyecto. Android es un sistema operativo basado en el núcleo de Linux, diseñado originariamente para dispositivos móviles, se ha expandido rápidamente su desarrollo pudiendo estar en reproductores mp3, netbooks, televisores, PCs, incluso en electrodomésticos, y en coches.

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones, sobre el núcleo de las bibliotecas (escritas en lenguaje C) en una máquina virtual Java propia denominada Dalvik con compilación en tiempo de ejecución. Esto se puede apreciar de manera más clara en la siguiente imagen que representa la arquitectura Android.

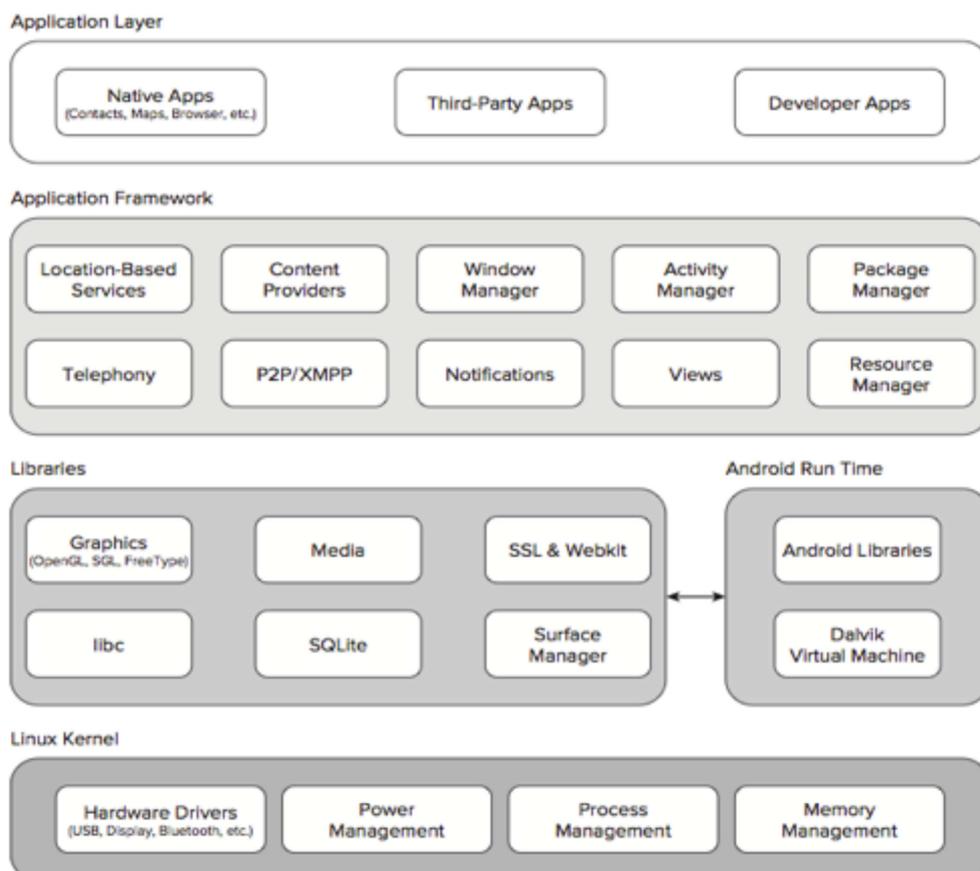


Figura 2. Arquitectura Android

Como plataforma para desarrollar aplicaciones, Android sigue un patrón Modelo Vista Controlador, donde las clases Java son el controlador, las vistas son archivos XML y el modelo vendría dado por los datos obtenidos de la base de datos o de servicios externos.

Como ya hemos mencionado anteriormente en la memoria, Android se desarrolla de forma abierta, y se puede acceder al código fuente, esto unido a que es altamente personalizable, dan lugar a uno de los principales problemas de esta plataforma: la **fragmentación**. Hay múltiples informes sobre la fragmentación de Android. Algunos lo llaman fragmentación y otros variedad de dispositivos, pero llámese como se llame, tiene sus beneficios y desventajas. Puede verse como un símbolo de **fortaleza del ecosistema Android y un exponente de la libertad que concede para desarrollar**. Desde nuestro punto de vista como desarrolladores de Android supone un dolor de cabeza, ya que hay múltiples versiones del propio sistema Android, más distintas personalizaciones del propio sistema operativo que hacen las propias marcas de los dispositivos, a eso hay que añadir los diversos tipos de tamaños y pantallas. El intentar que nuestra aplicación sea compatible con el mayor número de dispositivos posible puede suponer un mayor esfuerzo desde el punto de vista de desarrollo de la aplicación.

Ya hemos hablado de forma general de Android, como marco tecnológico en el que está contenida nuestra aplicación, la relación con ésta se basa en el desarrollo para gestionar una casa rural, integrándose con aplicaciones nativas de Android como son el Calendario y el Mail.

3. Descripción general del producto

Anteriormente se ha comentado en qué consiste el producto final que se pretende obtener de este proyecto y se ha puesto en el contexto en el que se define; en este apartado se dará una visión general del sistema, definiendo sus límites, funcionalidades básicas y los usuarios a los que va dirigido.

El producto se presenta como una aplicación informática típica de gestión, orientada al control de reservas de una casa rural, donde además de poder definir los componentes de la misma – la propia casa, habitaciones y sus características, así como los clientes que realizan reservas – puedes gestionar las reservas registrándolas tanto en base de datos como en el Calendario propio de Android con la versatilidad y seguridad que da la aplicación nativa de Calendario de Android, también cabe la posibilidad de la anulación de las reservas desde la aplicación, y en la parte de comunicación puedes ponerte en contacto cuando lo desees con un cliente para mandarlo un e-mail automatizado informándolo de su reserva.

Evidentemente, esta aplicación tiene sus limitaciones, desde el punto de vista de la comunicación con el cliente, solo se permite un mail automatizado a la hora de proceder

a la reserva pero no se permite en esta versión el poder parametrizar o poder modificar el correo a mandar. También tiene una limitación respecto al registro en el Calendario, ya que se queda registrado en el Calendario de Android, se podría acceder y modificar e incluso eliminar reservas desde el Calendario, al tener la información duplicada en la base de datos, se podría generar una sincronización periódica, pero podría haber momentos en los que no estuvieran sincronizados los datos de forma que llevaran a confusión al cliente.

El proyecto no puede dirigirse a un amplio espectro potencial de clientes, ya que está orientada a dueños gestores de casas rurales. Cualquier usuario que cumpla el anterior requisito que posea un dispositivo móvil compatible con la aplicación debería ser capaz de manejarla sin problemas, ya que el entorno es sencillo y amigable.

4. Planificación y presupuesto

Este apartado tiene un gran peso en la realización de todo proyecto, tiene gran importancia ya que el coste adecuado es una de las principales características que debe tener un proyecto, ya que de lo contrario es probable que el proyecto fracase.

La estimación del coste y el esfuerzo es muy complejo, no es una ciencia exacta, es una predicción que se basa en un modelo probabilista. Es complejo porque son demasiados los elementos variables que pueden afectar al coste del software y el esfuerzo para desarrollarlo. Aun así la estimación del coste del proyecto software proporciona una aproximación con un grado de riesgo aceptable.

Dividimos en dos apartados la estimación del coste del proyecto, por un lado la estimación del esfuerzo y por otro la estimación de los recursos.

Cabe destacar que en el apartado de estimación del esfuerzo tendremos dos tipos de cálculo del esfuerzo, una con el modelo COCOMO y otro desde un enfoque diferente, teniendo en cuenta el tamaño de componentes de la aplicación desde la experiencia, lo que se conoce como estimación por analogía con otros proyectos ya realizados sobre el mismo dominio de aplicación.

4.1. Estimación del esfuerzo

4.1.1. Modelo COCOMO

Para este punto vamos a utilizar el modelado algorítmico de costes denominado COCOMO (Modelo constructivo de Costes). Vamos a utilizar el COCOMO básico junto con el análisis de puntos de función, debido a que las funcionalidades del programa son concretas. El análisis del punto de función es utilizado como alternativa a

la medida del tamaño de una aplicación informática en líneas de código, independiente también de la tecnología empleada. Este sistema presenta una gran ventaja frente al sistema de líneas de código, ya que se estima el programa por lo que verdaderamente hace y no por el número de líneas de código que, al inicio del proyecto es muy difícil de estimar. Los puntos de función se determinan mediante la estimación del número de entradas, salidas, archivos de datos, consultas e interfaces, así como también por el uso de catorce valores de ajuste de complejidad que veremos más adelante a la hora de estimar el esfuerzo realizado en el proyecto.

Lo que nos ha dado la experiencia es que las estimaciones no siempre son correctas, no podemos asegurar que una estimación de la planificación del proyecto al inicio del mismo sea igual que la planificación real final.

Las métricas usadas para estimar el tamaño del proyecto software deben ser razonablemente fáciles de usar en etapas tempranas del proyecto y fácilmente medibles una vez terminado éste para que las comparaciones entre estas situaciones inicial y final nos provean de retroalimentación para hacer estimaciones más precisas en futuros proyectos.

Como hemos comentado anteriormente el método mediante análisis del punto de función se basa en una métrica que cuantifica la funcionalidad que se debe entregar al usuario al construir la aplicación. Vamos pues a identificar los parámetros y valores que sirven para evaluar dicha funcionalidad. Seguiremos los siguientes pasos:

1. Identificar las funciones (entradas, salidas, ficheros lógicos internos, ficheros lógicos externos, consultas externas) y clasificarlas en función de su grado de complejidad

Para poder determinar la complejidad relativa funcional de los procesos, salidas, consultas y entradas externas vamos a seguir los siguientes parámetros establecidos en el sistema de puntos de función:

Entradas y consultas externas	1 a 19 TED	20 a 50 TED	51 o más TED
0-1 TER accedidos	Bajo	Bajo	Medio
2 TER accedidos	Bajo	Medio	Alto
Más de 2 TER accedidos	Medio	Alto	Alto

Salidas externas	1 a 19 TED	20 a 50 TED	51 o más TED
0-1 TER accedidos	Bajo	Bajo	Medio
2-3 TER accedidos	Bajo	Medio	Alto
Más de 3 TER accedidos	Medio	Alto	Alto

Entradas externas:

- Almacenamiento registro usuario

- Datos elementales referenciados: 4
 - Registros lógicos referenciados: 2
 - Grado de complejidad: BAJA.
- Identificar al usuario
 - Datos elementales referenciados: 2
 - Registros lógicos referenciados: 3
 - Grado de complejidad: MEDIO.
- Almacenamiento definición casa
 - Datos elementales referenciados: 5
 - Registros lógicos referenciados: 2
 - Grado de complejidad: BAJA.
- Almacenamiento definición habitación
 - Datos elementales referenciados: 1
 - Registros lógicos referenciados: 2
 - Grado de complejidad: BAJA.
- Almacenamiento definición característica
 - Datos elementales referenciados: 3
 - Registros lógicos referenciados: 2
 - Grado de complejidad: BAJA.
- Almacenamiento configuración cuenta
 - Datos elementales referenciados: 1
 - Registros lógicos referenciados: 3
 - Grado de complejidad: MEDIO.
- Almacenamiento configuración pago
 - Datos elementales referenciados: 2
 - Registros lógicos referenciados: 1
 - Grado de complejidad: BAJA.
- Almacenamiento reserva
 - Datos elementales referenciados: 4
 - Registros lógicos referenciados: 3
 - Grado de complejidad: MEDIO.
- Anulación reserva
 - Datos elementales referenciados: 4
 - Registros lógicos referenciados: 2
 - Grado de complejidad: BAJA.
- Almacenamiento cliente
 - Datos elementales referenciados: 5
 - Registros lógicos referenciados: 2
 - Grado de complejidad: BAJA.

Salidas externas:

- Muestra la ventana registro en la aplicación

- Datos elementales referenciados: 4
 - Registros lógicos referenciados: 2
 - Grado de complejidad: BAJA.
- Muestra la ventana de cliente
 - Datos elementales referenciados: 5
 - Registros lógicos referenciados: 4
 - Grado de complejidad: MEDIA.
- Muestra la ventana de correo electrónico
 - Datos elementales referenciados: 1
 - Registros lógicos referenciados: 2
 - Grado de complejidad: BAJA.
- Fecha de inicio de reserva
 - Datos elementales referenciados: 1
 - Registros lógicos referenciados: 4
 - Grado de complejidad: MEDIA.
- Fecha de fin de reserva
 - Datos elementales referenciados: 1
 - Registros lógicos referenciados: 4
 - Grado de complejidad: MEDIA.
- Muestreo de errores
 - Datos elementales referenciados: 10
 - Registros lógicos referenciados: 1
 - Grado de complejidad: BAJA.
- Muestra listado habitaciones
 - Datos elementales referenciados: 3
 - Registros lógicos referenciados: 2
 - Grado de complejidad: BAJA.
- Muestra listado características
 - Datos elementales referenciados: 5
 - Registros lógicos referenciados: 2
 - Grado de complejidad: BAJA.
- Muestra listado configuraciones
 - Datos elementales referenciados: 3
 - Registros lógicos referenciados: 3
 - Grado de complejidad: BAJA.
- Muestra listado calendarios
 - Datos elementales referenciados: 2
 - Registros lógicos referenciados: 3
 - Grado de complejidad: BAJA.
- Muestra listado clientes
 - Datos elementales referenciados: 5
 - Registros lógicos referenciados: 2
 - Grado de complejidad: BAJA.

- Muestra listado anulación reservas
 - Datos elementales referenciados: 4
 - Registros lógicos referenciados: 2
 - Grado de complejidad: BAJA.

Consultas externas:

- Consulta Calendario
 - Datos elementales referenciados: 7
 - Registros lógicos referenciados: 4
 - Grado de complejidad: MEDIO.

Para poder determinar la complejidad relativa funcional de los ficheros lógicos vamos a usar los siguientes parámetros:

La asignación de complejidades a FLI (fichero lógico interno) se basa en el número de Tipos de Elementos de Datos (TED) y número de Tipos de Elementos de Registros (TER).

Un tipo de elemento de dato (TED) se define como un campo único, no recurrente y reconocible para el usuario en un FLI (fichero lógico interno).

Un tipo de elemento de registro (TER) se define como un subgrupo de elementos de datos reconocibles para el usuario dentro de un FLI (fichero lógico interno).

Una vez conocidos los tipos de elementos de datos y tipos de elementos de registros propios de cada fichero podemos establecer el nivel de complejidad apoyándonos en la siguiente tabla:

	1 a 19 TED	20 a 50 TED	51 o más TED
1 TER	Bajo	Bajo	Medio
2 a 5 TER	Bajo	Medio	Alto
6 ó más TER	Medio	Alto	Alto

Ficheros internos:

- Base de datos formada por 7 tablas con un total de 34 campos.
 - Datos elementales referenciados: 34
 - Registros lógicos referenciados: 7
 - Grado de complejidad: ALTA.
- Fichero lógico de configuración de la aplicación:
 - Datos elementales referenciados: 6
 - Registros lógicos referenciados: 3

- Grado de complejidad: BAJA.

2. Calcular puntos de función sin ajustar

Una vez identificadas y clasificadas las funciones vamos a proceder a calcular sus puntos de función sin ajustar:

Tipo de función	Complejidad	Total de la complejidad	Total de la función
Entrada externa	7 BAJA X 3	21	33
	3 MEDIA X 4	12	
	0 ALTA X 6	0	
Salida externa	9 BAJA X 4	36	51
	3 MEDIA X 5	15	
	0 ALTA X 7	0	
Consulta externa	0 BAJA X 3	0	4
	1 MEDIA X 4	4	
	0 ALTA X 7	0	
Fichero lógico interno	1 BAJA X 5	5	20
	0 MEDIA X 10	0	
	1 ALTA X 15	15	
Total de puntos de función no ajustados			108

3. Calcular el factor de ajuste

El cálculo del valor del factor de ajuste se basa en la cuantificación de 14 características generales del sistema de puntos de función ajustados. Cada una de estas características tiene un grado de influencia, según ese grado de influencia se le asigna un valor. Debido a las características de nuestro sistema los niveles de influencia serán los siguientes:

0- Sin influencia	3- Medio
1- Incidental	4- Significativo
2- Moderado	5- Esencial

Factores de complejidad (FC)	Nivel de influencia
Comunicación de datos	4
Rendimiento	3
Frecuencia de transacciones	2
Interface del usuario final	0
Procesos complejos	2
Facilidad de mantenimiento	0
Instalación en múltiples lugares	2
Procesamiento distribuido	0
Gran carga de trabajo	1

Entrada on-line de datos	3
Actualizaciones on-line	2
Utilización con otros sistemas, reusabilidad	2
Facilidad de operación	2
Facilidad de cambios	3
Nivel de influencia	26

El valor del **factor de ajuste** se calcula:

$$FA = 0,65 + (0,01 * \text{Nivel de influencia})$$

$$FA = 0,65 + (0,01 * 26) = \mathbf{0,91}$$

Puntos de función ajustados:

$$PF = PF \text{ no ajustados} * FA$$

$$PF = 108 * 0,91 = \mathbf{98,28}$$

Este número representa la cantidad de unidades de software de un proyecto, el cual nos sirve para realizar estimaciones de plazos, costes y recursos.

Modelo COCOMO BÁSICO

El modelo COCOMO permite la estimación del esfuerzo apoyándose en una estimación previa del tamaño del software en líneas de código que inferimos del calculo que acabamos de realizar de los puntos de función.

Este modelo se basa en la existencia de tres modos de desarrollo aplicables a las características del proyecto:

Modo	A	B	C	D
Orgánico	2,4	1,05	2,5	0,38
Semiacoplado	3	1,12	2,5	0,35
Integrado	3,6	1,2	2,5	0,32

Nuestro sistema se asemeja al modo Orgánico ya que es un proyecto desarrollado en un entorno estable, con poca innovación técnica con pocas presiones de tiempo y tamaño relativamente pequeño (menos de 50 KLDC), por lo tanto cogeremos los valores de las constantes del modo Orgánico.

Debido a que la aplicación se desarrollará en lenguaje JAVA y sabemos que para cada punto de función equivale a 53 líneas de código (LOC) obtenido de la tabla propuesta por Caper T. Jones:

$$KLOC = (53 * 92,28) / 1000 = 5,20884 \text{ KLOC (Kilo líneas de código)}$$

Para realizar el modelo necesitamos también el factor de complejidad total, que se obtiene mediante la multiplicación de los valores evaluados en 15 factores específicos

de coste que se observan en la siguiente tabla (se muestran sombreados los valores seleccionados para este proyecto):

FACTORES	Valor de los factores					
	Muy bajo	Bajo	Medio	Alto	Muy alto	Extra alto
Fiabilidad requerida	0,75	0,88	1,00	1,15	1,4	
Tamaño de la base de datos		0,94	1,00	1,08	1,16	
Complejidad del software	0,7	0,85	1,00	1,15	1,30	1,65
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria			1,00	1,06	1,21	1,56
Volatilidad del hardware		0,87	1,00	1,15	1,30	
Restricciones de tiempo de respuesta		0,87	1,00	1,07		
Calidad de los analistas	1,46	1,19	1,00	0,86	0,71	
Experiencia con el tipo de aplicación	1,29	1,13	1,00	0,91	0,82	
Experiencia con el hardware	1,21	1,10	1,00	0,90		
Exp. con el lenguaje de programación.	1,14	1,07	1,00	0,95		
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Técnicas modernas de programación	1,24	1,10	1,00	0,91	0,82	
Empleo de herramientas	1,24	1,10	1,00	0,91	0,83	
Restricciones a la duración del proyec.	1,23	1,08	1,00	1,04	1,10	

Así, para calcular el factor de ajuste del esfuerzo (FAE) se multiplican los valores sombreados distintos de 1:

$$\text{FAE} = \prod_{k=1}^n \text{Factor}_k = 0,94 * 0,90 = \mathbf{0,846}$$

Esfuerzo del desarrollo:

$$E = a * KLDC^b * \text{FAE}$$

$$E = 3,2 * 5,20884^{1,05} * 0,84 = \mathbf{15,21 \text{ personas/mes}}$$

*siendo a y b las constantes de la tabla de modos que corresponden al orgánico

Tiempo:

$$T = c * E^d$$

$$T = 2,5 * 15,21^{0,38} = \mathbf{7,03 \text{ meses}}$$

*siendo c y d las constantes de la tabla de modos que corresponden al orgánico

Productividad:

$$\text{PR} = \frac{\text{LDC}}{\text{Esfuerzo}} = \frac{5,208}{15,21} = \mathbf{342,41 \text{ LDC / personas mes}}$$

Personal promedio:

$$P = \frac{E}{T} = \frac{15,21}{7,03} = \mathbf{2,16 \text{ LDC/ personas mes}}$$

Atendiendo a estas cifras, para realizar el proyecto necesitaríamos 15 personas en un mes, pero como el proyecto solo lo realizará una sola persona, el tiempo se verá incrementado hasta los 15 meses aproximadamente. Puesto que un mes tiene 160 horas

laborables, el coste en horas sería de 2443 horas de trabajo.

4.1.2. Modelo de estimación por analogía desde la experiencia

Para este punto vamos a utilizar la experiencia como modelo de medida para estimar. Nuestro objetivo es estimar el número de ocurrencias de cada componente estándar y luego mediante datos de estadísticos históricos determinar el tamaño de entrega del componente estándar y así poder tener una idea aproximada de la estimación.

Debido a que las funcionalidades del programa son concretas, la medida del tamaño de una aplicación informática de la que conocemos la tecnología empleada debería ser factible poder hacer una relación estable entre una tarea y el tiempo que nos lleva hacerla.

Este sistema basado en la experiencia presenta una gran ventaja frente a cualquier otro modelo ya que se estima el programa por lo que verdaderamente hace y la experiencia dirá lo que se debe implementar para llegar a la solución funcional.

Como hemos comentado anteriormente, la experiencia nos revela que las estimaciones no siempre son correctas, no podemos asegurar que una estimación de la planificación del proyecto al inicio del mismo sea igual que la planificación real final del mismo, pero si es cierto que la experiencia es la mejor medida estadística para la estimación de un proyecto.

Para ello vamos a tomar como base la planificación real de este mismo proyecto. Como primer paso necesitamos identificar un patrón para englobar las previsiones de estimación; vamos a identificar las pantallas de la aplicación por dificultad, así podremos asemejar cualquier aplicación con pantallas de dificultad similar y en función de su funcionalidad o funcionalidades podremos estimar cuántas pantallas será necesario realizar y por tanto cuál será la estimación temporal del proyecto.

Vamos a realizar una tabla de conversión:

Tipo dificultad de pantalla	Descripción	Estimación
Fácil	Sin mucha complejidad de computación, típicas pantallas que muestran listas de datos o recogen datos de forma sencilla	32 horas
Media	Complejidad normal de computación, se necesita una mayor interacción con el usuario y un mayor procesado de datos por parte de la aplicación	64 horas

Difícil	Alta complejidad, en la que la aplicación tiene un alto coste computacional	128 horas
---------	---	-----------

Mediante esta relación podremos identificar el requisito del cliente, identificar cuál es la dificultad de lo que nos piden, y podremos, de manera aproximada, estimar cuántas pantallas vamos a necesitar y a partir de este punto poder hacer una estimación del proyecto.

En concreto, en este nuestro proyecto, hemos identificado:

Pantallas	Ocurrencias	Total de horas
Fáciles	9 x 32	288
Medias	6 x 64	384
Difíciles	1 x 128	128

Esto nos hace un total de 800 horas, y esto englobaría las partes de análisis, diseño e implementación de las pantallas, además habría que añadir un porcentaje de pruebas, de documentación y tener en cuenta un margen de sobrecarga, esta tabla final de división del esfuerzo vamos a reflejarla en el siguiente punto de presupuesto en el apartado de recursos humanos para compararla con el anterior modelo de estimación del COCOMO.

4.2. Estimación de los recursos

A continuación se detallan los recursos utilizados para la elaboración del proyecto, tanto humanos como materiales:

Recursos humanos:

- Un analista
- Un diseñador
- Un programador
- Un probador

Recursos hardware:

- Ordenadores:
 - Acer Aspire XC600 con SO Windows 8.
- Mviles:
 - HTC Desire
 - Huawei Ascend G6
- Otros:
 - Router Wi-Fi (conexión Internet)
 - Impresora HP psc 2210

Recursos software:

- Windows 8
- Microsoft Office 2010
- StarUML
- SDK y ADT de Android
- Gimp 2.8
- OpenProj
- Adobe Reader
- PDFCreator

4.3. Presupuesto

A partir de las estimaciones de esfuerzo y tiempo y de los recursos utilizados, vamos a proceder a calcular el presupuesto total del proyecto. Cabe destacar que el proyecto lo realizará 1 persona que alternará sus labores como programador, analista, diseñador.

También vamos a ver cómo quedarían nuestros presupuestos con las dos variantes de cálculo de la estimación del esfuerzo, dejando patente la gran diferencia que hay.

Recursos humanos

En este apartado vamos a diferenciar dos totales de horas.

Se estima la duración de cada tarea del proyecto en horas tomando como referencia la estimación anteriormente realizada con el modelo COCOMO. Dicha estimación daba como resultado 15 meses, con una media de 20 días laborables al mes y 8 horas al día, nos da un total de:

Estimación esfuerzo:

Análisis	10 %	} del proyecto completo
Diseño	20 %	
Programación	40 %	
Pruebas	15 %	
Sobrecarga	15 %	

TAREA	%	DURACION (HORAS)
Requisitos del sistema	5	120
Análisis del sistema	10	240
Diseño del sistema	15	360
Implementación	40	960
Pruebas	15	360
Documentación	5	120
Sobrecarga	10	240
TOTAL		2.400 horas

Tabla 1. División del esfuerzo del proyecto - COCOMO

Sueldos para los integrantes del proyecto:

	TIEMPO	COSTE
Analista / Diseñador	720	15€ / Hora
Programador	1200	10€ / Hora
Probador	360	10€ / Hora
Documentalista	120	10€ / Hora
TOTAL		27.600 €

Tabla 2. Presupuesto desarrollo – COCOMO

Se estima la duración de cada tarea del proyecto en horas tomando como referencia la estimación anteriormente realizada con el modelo de estimación por analogía. Dicha estimación nos da como resultado las horas totales de desarrollo, por lo que debemos añadir de manera porcentual los apartados de pruebas, documentación y tener en cuenta un margen de sobrecarga.

TAREA	%	DURACION (HORAS)
Requisitos del sistema	5	61,5
Análisis del sistema	10	123
Diseño del sistema	15	184,5
Implementación	40	492,3
Pruebas	15	184,5
Documentación	5	61,5
Sobrecarga	10	123
TOTAL		1.230,3 horas

Tabla 3. División del esfuerzo del proyecto - Analogía

Sueldos para los integrantes del proyecto:

	TIEMPO	COSTE
Analista / Diseñador	369	15€ / Hora
Programador	615,3	10€ / Hora
Probador	184,5	10€ / Hora
Documentalista	61,5	10€ / Hora
TOTAL		14.148 €

Tabla 4. Presupuesto desarrollo – Analogía

Recursos materiales

Se estiman los recursos materiales tomando como referencia la estimación anteriormente realizada con el modelo COCOMO, que indica 15 meses de desarrollo.

Recurso	Coste	% Coste aplicado	Coste Proyecto
HARDWARE			
Acer Aspire XC600	1000 €	22,2 %	222 €

HTC Desire	100 €	8 %	8 €
Huawei Ascend G6	180 €	8 %	14,4 €
Conexión Internet	24 €/mes	15 %	54 €
Impresora	60 €	15 %	3 €
SOFTWARE			
Windows 8	0	-	0
Microsoft Office 2010	270 €	10 %	27 €
StarUML	0	-	0
SDK de Android	0	-	0
Gimp 2.8	0	-	0
OpenProj	0	-	0
Adobe Reader	0	-	0
PDFCreator	0	-	0
OTROS			
Otros gastos	150 €	100 %	150 €
TOTAL			478,40 €

Tabla 5. Presupuesto recursos – COCOMO

Se estiman los recursos materiales tomando como referencia la estimación anteriormente realizada mediante el modelo de estimación por analogía, que indica casi 8 meses de desarrollo.

Recurso	Coste	% Coste aplicado	Coste Proyecto
HARDWARE			
Acer Aspire XC600	1000 €	12,15 %	121,50 €
HTC Desire	100 €	8 %	8 €
Huawei Ascend G6	180 €	8 %	14,4 €
Conexión Internet	24 €/mes	15 %	28,80 €
Impresora	60 €	15 %	3 €
SOFTWARE			
Windows 8	0	-	0
Microsoft Office 2010	270 €	10 %	27 €
StarUML	0	-	0
SDK de Android	0	-	0
Gimp 2.8	0	-	0
OpenProj	0	-	0
Adobe Reader	0	-	0
PDFCreator	0	-	0
OTROS			
Otros gastos	150 €	100 %	150 €
TOTAL			352,70 €

Tabla 6. Presupuesto recursos – Analogía

Presupuesto total

El Presupuesto total estará formado por la suma de los costes de los recursos humanos y los materiales:

Para el modelo COCOMO:

	COSTE
Recursos humanos	27.600 €
Recursos materiales	478,40 €
TOTAL	28.078,40 €

Tabla 7. Presupuesto total – COCOMO

Para el modelo de estimación por analogía:

	COSTE
Recursos humanos	14.148 €
Recursos materiales	352,70 €
TOTAL	14.500,70 €

Tabla 8. Presupuesto total – Analogía

4.4. Seguimiento del calendario

En este punto vamos a comentar el seguimiento real del calendario del proyecto, realizado con la herramienta de gestión OpenProj, veremos que los resultados distan mucho de la estimación inicial realizada con el método COCOMO, pero poco con la estimación que hemos hecho haciendo uso de la experiencia con este tipo de aplicaciones.

Antes del comienzo del mismo, después de hacer una estimación inicial con el método COCOMO, el proyecto parecía inabarcable, ya que 15 meses de proyecto por una sola persona eran inasumibles. Pero la experiencia realizando estimaciones nos dice que el método COCOMO suele estar sobreestimado, y haciendo la comparativa con la estimación por analogía ha quedado confirmado. Por suerte finalmente se ha podido reducir bastante esta estimación temporal del COCOMO.

También cabe recordar que seguimos el ciclo de vida en cascada realimentado, por lo que hasta que no haya finalizado una etapa no se inicia la siguiente.

Task Information

ID	Nombre	Duración	Inicio	Terminado	Predecesores	Nombres del Recurso
1	eRural	425 days?	2/06/14 8:00	31/07/15 17:00		
2	Planificación y estimación	414 days	2/06/14 8:00	21/07/15 8:00		
3	Planificación temporal	1 day	2/06/14 8:00	2/06/14 17:00		Analista-Diseñador
4	Estimación presupuestaria	1 day	3/06/14 8:00	3/06/14 17:00	3	Analista-Diseñador
5	Planificación terminada	0 days	21/07/15 8:00	21/07/15 8:00	4	
6	Requisitos	412 days	4/06/14 8:00	21/07/15 8:00	5	
7	Analizar requisitos	3 days	4/06/14 8:00	6/06/14 17:00	4	Analista-Diseñador
8	Documentar requisitos	5 days	7/06/14 8:00	11/06/14 17:00	7	Analista-Diseñador
9	Revisar requisitos	1 day	12/06/14 8:00	12/06/14 17:00	8	Analista-Diseñador
10	Requisitos completado	0 days	21/07/15 8:00	21/07/15 8:00	9	
11	Análisis	403 days	13/06/14 8:00	21/07/15 8:00	10	
12	Modelado analítico del sistema	15 days	13/06/14 8:00	27/06/14 17:00	10	Analista-Diseñador
13	Revisar modelo analítico	2 days	28/06/14 8:00	29/06/14 17:00	12	Analista-Diseñador
14	Análisis completado	0 days	21/07/15 8:00	21/07/15 8:00	13	
15	Diseño	386 days	30/06/14 8:00	21/07/15 8:00	14	
16	Diseño de clases	5 days	30/06/14 8:00	4/07/14 17:00	14	Analista-Diseñador
17	Diseño de interfaces	3 days	5/07/14 8:00	7/07/14 17:00	16	Analista-Diseñador
18	Diseño completado	0 days	21/07/15 8:00	21/07/15 8:00	17	Analista-Diseñador
19	Implementación	378 days	8/07/14 8:00	21/07/15 8:00	18	
20	Implementación controlador base	5 days	8/07/14 8:00	12/07/14 17:00	18	Programador
21	Implementación interface	25 days	13/07/14 8:00	6/08/14 17:00	20	Programador
22	Implementación calendario	10 days	7/08/14 8:00	16/08/14 17:00	21	Programador
23	Integración	4 days	17/08/14 8:00	20/08/14 17:00	22	Programador
24	Implementación complementaria	63 days	19/05/15 8:00	20/07/15 17:00	23	Programador
25	Implementación completado	0 days	21/07/15 8:00	21/07/15 8:00	24	Programador
26	Pruebas	4 days?	21/07/15 8:00	24/07/15 17:00	25	
27	Pruebas aplicación	2 days	21/07/15 8:00	22/07/15 17:00	25	Probador
28	Pruebas calendario	2 days?	23/07/15 8:00	24/07/15 17:00	27	Probador
29	Pruebas completado	0 days	24/07/15 17:00	24/07/15 17:00	28	Probador
30	Documentación	7 days?	25/07/15 8:00	31/07/15 17:00	29	

Figura 3. Listado de tareas del proyecto

En esta otra imagen vamos a apreciar el listado de tareas con el diagrama de Gantt, representando cada barra la tarea correspondiente, la longitud de la barra es proporcional a la duración de la tarea a la que está asociada.

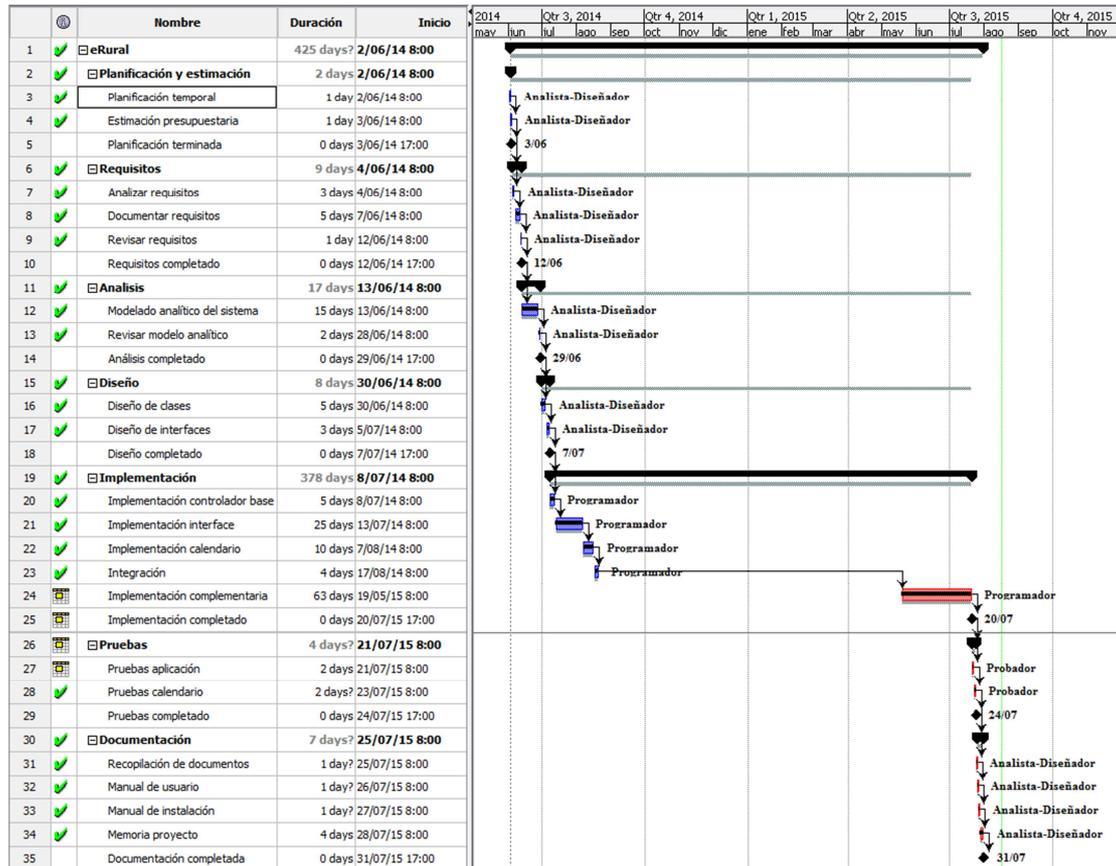


Figura 4. Listado de tareas del proyecto con diagrama Gantt detallado

Como se puede observar en la siguiente figura, aparece un resumen con todos los datos de la fecha de inicio y fin del proyecto, su duración en días laborables y horas totales de trabajo, y como dato más importante el coste total de los recursos humanos del proyecto, que suma un total de **14.040 euros**.

eRural TFG

Dates			
Start	2/06/14 8:00	Finish	31/07/15 17:00
Baseline Start	2/06/14 8:00	Baseline Finish	31/07/15 17:00
Actual Start	2/06/14 8:00	Actual Finish	31/07/15 17:00

Duration			
Scheduled	425 days	Remaining	425 days
Baseline	425 days	Actual	0 days
		Percent Complete	100%

Work			
Scheduled	1.232 horas	Remaining	0 horas
Baseline	2.224 horas	Actual	1.232 horas

Costs			
Scheduled	14040,00 €	Remaining	0,00 €
Baseline	14040,00 €	Actual	14040,00 €
		Variance	0,00 €

Figura 5. Informe general del proyecto

Presupuesto total

El presupuesto total final estará formado por la suma de los recursos humanos, que hemos detallado en el diagrama de Gantt, y los recursos materiales que hemos utilizado a lo largo del tiempo reflejado en tal diagrama:

	COSTE
Recursos humanos	14.040 €
Recursos materiales	352,70 €
TOTAL	14.392,70 €

Tabla 9. Presupuesto total

Viendo los resultados obtenidos, se puede sacar conclusiones a raíz de las estimaciones presupuestarias realizadas inicialmente, las cuales separamos en dos: la estimación según el modelo COCOMO y una estimación experimentada por analogía de este tipo de aplicaciones.

La estimación COCOMO está muy alejada de los datos obtenidos del seguimiento con la aplicación OpenProj, existiendo una **diferencia bastante grande de más de 13.000 € entre el presupuesto estimado y real de los recursos humanos**. Sería exactamente 13.560 € más el presupuesto total estimado en el anterior punto.

Queda plasmado que los tiempos que ha durado finalmente el proyecto es significativamente más pequeño que el tiempo estimado con el COCOMO, un total de 1.232 horas respecto a las 2.400 horas que estimaba el sistema COCOMO.

La conclusión más significativa es que ningún modelo de estimación es apropiado para todas las clases de software ni en todos los entornos de desarrollo, en este caso el modelado algorítmico de costes COCOMO no ha ido nada bien con este tipo de software, todo lo contrario que la estimación por analogía, que basándose en otros proyectos ya realizados sobre el mismo dominio de aplicación tiene una visión mucho más aproximada para poder hacer una estimación más correcta.

Bloque II – Documentación técnica

5. Análisis de requisitos del sistema

5.1. Introducción

El análisis de un sistema informático es una de las etapas de construcción más importante para cumplir los objetivos finales. Consiste en relevar la información actual y proponer los rasgos generales de la solución futura. Para llegar a la solución correcta, hay que extraer todos los requisitos del usuario, comprobarlos, asegurándonos de que todos esos requisitos estén contemplados en el diseño y que el diseño no difiera de los requisitos.

A continuación vamos a ir viendo la fase de análisis de la aplicación.

5.2. Objetivos del sistema

En este apartado vamos a ir viendo la lista de objetivos que deberán cumplirse cuando la aplicación esté en funcionamiento.

OBJ-1	Gestión de la configuración de la aplicación
Versión	1.0
Autores	Miguel Ángel Navas Galán
Fuentes	Especificación de requisitos
Descripción	El sistema pedirá los datos obligatorios de configuración de la aplicación.
Subobjetivos	OBJ-1.1 Configuración de parámetros OBJ-1.2 Configuración de componentes
Importancia	Alta.
Urgencia	Alta.
Estado	Validado.
Estabilidad	Alta.
Comentarios	Ninguno.

Tabla 10. OBJ-1 Gestión de la configuración de la aplicación

OBJ-1.1	Configuración de parámetros
Versión	1.0
Autores	Miguel Ángel Navas Galán
Fuentes	Especificación de requisitos
Descripción	El sistema pedirá los parámetros necesarios para registrar la casa rural.
Importancia	Alta.
Urgencia	Alta.
Estado	Validado.
Estabilidad	Alta.

Comentarios	Englobamos dentro de la configuración de parámetros los datos de la casa, habitaciones, características y clientes que quedarán registrados.
--------------------	--

Tabla 11. OBJ-1.1 Configuración de parámetros

OBJ-1.2	Configuración de componentes
Versión	1.0
Autores	Miguel Ángel Navas Galán
Fuentes	Especificación de requisitos
Descripción	El sistema debe gestionar el componente Calendario y los componentes del correo para su comunicación.
Importancia	Alta.
Urgencia	Alta.
Estado	Validado.
Estabilidad	Alta.

Tabla 12. OBJ-1.2 Configuración de componentes

OBJ-1.3	Configuración del pago
Versión	1.0
Autores	Miguel Ángel Navas Galán
Fuentes	Especificación de requisitos
Descripción	El sistema debe gestionar las configuraciones de pago.
Importancia	Alta.
Urgencia	Alta.
Estado	Validado.
Estabilidad	Alta.

Tabla 13. OBJ-1.3 Configuración del pago

OBJ-2	Gestión de las reservas
Versión	1.0
Autores	Miguel Ángel Navas Galán
Fuentes	Especificación de requisitos
Descripción	El sistema deberá definir el comportamiento general de las reservas de las habitaciones.
Subobjetivos	OBJ-2.1 Gestión de habitaciones OBJ-2.2 Gestión calendario OBJ-2.3 Gestión clientes
Importancia	Alta.
Urgencia	Alta.
Estado	Validado.
Estabilidad	Alta.
Comentarios	Ninguno.

Tabla 14. OBJ-2 Gestión de las reservas

OBJ-2.1	Gestión de habitaciones
Versión	1.0
Autores	Miguel Ángel Navas Galán
Fuentes	Especificación de requisitos
Descripción	El sistema debe gestionar las habitaciones registradas en la aplicación.
Importancia	Alta.
Urgencia	Alta.
Estado	Validado.
Estabilidad	Alta.

Tabla 15. OBJ-2.1 Gestión de habitaciones

OBJ-2.2	Gestión calendario
Versión	1.0
Autores	Miguel Ángel Navas Galán
Fuentes	Especificación de requisitos
Descripción	El sistema debe gestionar el calendario en el que hemos registrado para grabar y anular las reservas.
Importancia	Alta.
Urgencia	Alta.
Estado	Validado.
Estabilidad	Alta.

Tabla 16. OBJ-2.2 Gestión calendario

OBJ-2.3	Gestión de clientes
Versión	1.0
Autores	Miguel Ángel Navas Galán
Fuentes	Especificación de requisitos
Descripción	El sistema debe gestionar los clientes registrados en la aplicación.
Importancia	Alta.
Urgencia	Alta.
Estado	Validado.
Estabilidad	Alta.

Tabla 17. OBJ-2.3 Gestión de clientes

5.3. Catálogos de requisitos del sistema

El objetivo de la especificación es definir en forma clara, precisa, completa y verificable todas las funcionalidades y restricciones del sistema que se desea construir.

Esta sección se divide en las siguientes subsecciones en las que se describen los

requisitos del sistema.

5.3.1. Requisitos de información

Esta subsección debe contener la lista de requisitos de almacenamiento y de restricciones de información que se hayan identificado.

IRQ-1	Información usuario gestor	
Versión	1.0	
Autores	Miguel Ángel Navas Galán	
Fuentes	Especificación de requisitos	
Objetivos asociados	OBJ-1 Gestión de la configuración de la aplicación OBJ-1.1 Configuración de parámetros	
Requisitos asociados	UC-1 Identificar al usuario	
Descripción	El sistema almacenará la información necesaria para el registro de la Casa Rural.	
Datos específicos	<ul style="list-style-type: none"> - Nombre del gestor - Apellidos - Alias / usuario - Contraseña 	
Tiempo de vida	Medio	Máximo
	Indeterminado.	Indeterminado.
Ocurrencias simultaneas	Medio	Máximo
	1	1
Importancia	Alta.	
Urgencia	Alta.	
Estado	Validado.	
Estabilidad	Alta.	
Comentarios	Ninguno.	

Tabla 18. IRQ-1 Información usuario gestor

IRQ-2	Información Casa Rural	
Versión	1.0	
Autores	Miguel Ángel Navas Galán	
Fuentes	Especificación de requisitos	
Objetivos asociados	OBJ-1 Gestión de la configuración de la aplicación OBJ-1.1 Configuración de parámetros	
Requisitos asociados	UC-2 Configuración de la Casa	
Descripción	El sistema almacenará la información necesaria para el registro de la Casa Rural.	

Datos específicos	<ul style="list-style-type: none"> - Nombre de la casa - Dirección de la casa - Teléfono de la casa. - Comentarios 	
Tiempo de vida	Medio	Máximo
	Indeterminado.	Indeterminado.
Ocurrencias simultaneas	Medio	Máximo
	1	1
Importancia	Alta.	
Urgencia	Alta.	
Estado	Validado.	
Estabilidad	Alta.	
Comentarios	Ninguno.	

Tabla 19. IRQ-2 Información Casa Rural

IRQ-3	Información habitaciones	
Versión	1.0	
Autores	Miguel Ángel Navas Galán	
Fuentes	Especificación de requisitos	
Objetivos asociados	OBJ-1 Gestión de la configuración de la aplicación OBJ-1.1 Configuración de parámetros OBJ-2 Gestión de las reservas OBJ-2.1 Gestión de habitaciones	
Requisitos asociados	UC-3 Configuración de las habitaciones	
Descripción	El sistema almacenará la información necesaria para el registro de las distintas habitaciones.	
Datos específicos	- Nombre de la habitación	
Tiempo de vida	Medio	Máximo
	Indeterminado.	Indeterminado.
Ocurrencias simultaneas	Medio	Máximo
	1	1
Importancia	Alta.	
Urgencia	Alta.	
Estado	Validado.	
Estabilidad	Alta.	
Comentarios	Ninguno.	

Tabla 20. IRQ-3 Información habitaciones

IRQ-3.1	Información características	
Versión	1.0	
Autores	Miguel Ángel Navas Galán	
Fuentes	Especificación de requisitos	

Objetivos asociados	OBJ-1 Gestión de la configuración de la aplicación OBJ-1.1 Configuración de parámetros OBJ-2 Gestión de las reservas OBJ-2.1 Gestión de habitaciones	
Requisitos asociados	IRQ-3 Información Habitaciones UC-4 Configuración de las características	
Descripción	El sistema almacenará la información necesaria para el registro de las características tanto comunes como las propias de las habitaciones.	
Datos específicos	<ul style="list-style-type: none"> - Nombre de la característica - Descripción - Precio 	
Tiempo de vida	Medio	Máximo
	Indeterminado.	Indeterminado.
Ocurrencias simultaneas	Medio	Máximo
	1	1
Importancia	Alta.	
Urgencia	Alta.	
Estado	Validado.	
Estabilidad	Alta.	
Comentarios	Ninguno.	

Tabla 21. IRQ-3.1 Información características

IRQ-4	Información de las reservas	
Versión	1.0	
Autores	Miguel Ángel Navas Galán	
Fuentes	Especificación de requisitos	
Objetivos asociados	OBJ-2 Gestión de las reservas	
Requisitos asociados	UC-8 Gestión de las habitaciones UC-9 Gestión del Calendario UC-10 Gestión de los clientes UC-11 Comunicación de la reserva	
Descripción	El sistema almacenará la información necesaria para el registro de las reservas de habitación.	
Datos específicos	- Nombre de la habitación	
Tiempo de vida	Medio	Máximo
	Indeterminado.	Indeterminado.
Ocurrencias simultaneas	Medio	Máximo
	1	Indeterminado.
Importancia	Alta.	
Urgencia	Alta.	
Estado	Validado.	
Estabilidad	Alta.	
Comentarios	Ninguno.	

Tabla 22. IRQ-4 Información de las reservas

IRQ-5	Información del calendario	
Versión	1.0	
Autores	Miguel Ángel Navas Galán	
Fuentes	Especificación de requisitos	
Objetivos asociados	OBJ-1.2 Configuración de componentes OBJ-2 Gestión de las reservas OBJ-2.2 Gestión calendario	
Requisitos asociados	UC-6 Configuración de los componentes UC-9 Gestión del Calendario	
Descripción	El sistema almacenará la información necesaria para el registro en el calendario.	
Datos específicos	<ul style="list-style-type: none"> - Id calendario - Nombre de la cuenta asociada al calendario 	
Tiempo de vida	Medio	Máximo
	Indeterminado.	Indeterminado.
Ocurrencias simultaneas	Medio	Máximo
	1	1
Importancia	Alta.	
Urgencia	Alta.	
Estado	Validado.	
Estabilidad	Alta.	
Comentarios	Ninguno.	

Tabla 23. IRQ-5 Información del calendario

IRQ-6	Información del cliente	
Versión	1.0	
Autores	Miguel Ángel Navas Galán	
Fuentes	Especificación de requisitos	
Objetivos asociados	OBJ-2 Gestión de las reservas OBJ-2.3 Gestión de clientes	
Requisitos asociados	UC-10 Gestión de los clientes	
Descripción	El sistema almacenará la información necesaria para el registro de un cliente.	
Datos específicos	<ul style="list-style-type: none"> - Nombre - Apellidos - E-Mail - Comentarios - Forma de pago - Información de pago 	
Tiempo de vida	Medio	Máximo
	Indeterminado.	Indeterminado.
Ocurrencias simultaneas	Medio	Máximo
	1	1
Importancia	Alta.	

Urgencia	Alta.
Estado	Validado.
Estabilidad	Alta.
Comentarios	Ninguno.

Tabla 24. IRQ-6 Información del cliente

IRQ-7	Información del pago	
Versión	1.0	
Autores	Miguel Ángel Navas Galán	
Fuentes	Especificación de requisitos	
Objetivos asociados	OBJ-1.3 Configuración del pago OBJ-2 Gestión de las reservas OBJ-2.3 Gestión de clientes	
Requisitos asociados	UC-7 Configuración del pago	
Descripción	El sistema almacenará la información necesaria para el pago de un cliente.	
Datos específicos	<ul style="list-style-type: none"> - Cuenta bancaria - Porcentaje de reserva 	
Tiempo de vida	Medio	Máximo
	Indeterminado.	Indeterminado.
Ocurrencias simultaneas	Medio	Máximo
	1	1
Importancia	Alta.	
Urgencia	Alta.	
Estado	Validado.	
Estabilidad	Alta.	
Comentarios	Ninguno.	

Tabla 25. IRQ-7 Información del pago

CRQ-1	Único usuario gestor
Versión	1.0
Autores	Miguel Ángel Navas Galán
Fuentes	Especificación de requisitos
Objetivos asociados	OBJ-1 Gestión de la configuración de la aplicación OBJ-1.1 Configuración de parámetros
Requisitos asociados	IRQ-1 Información usuario gestor
Descripción	El sistema solo aceptará un usuario gestor para la casa rural.
Importancia	Alta.
Urgencia	Alta.
Estado	Validado.
Estabilidad	Alta.

Comentarios	Solo podrá ser un usuario gestor de la casa rural, al validar siempre el usuario y contraseña al entrar damos seguridad a la acción de reservar habitaciones.
--------------------	---

Tabla 26. CRQ-1 Único usuario gestor

CRQ-2	Reserva única en la mismas fechas
Versión	1.0
Autores	Miguel Ángel Navas Galán
Fuentes	Especificación de requisitos
Objetivos asociados	OBJ-2 Gestión de las reservas
Requisitos asociados	IRQ-4 Información de las reservas
Descripción	El sistema solo aceptará una reserva para la misma franja de fechas de la misma habitación.
Importancia	Alta.
Urgencia	Alta.
Estado	Validado.
Estabilidad	Alta.
Comentarios	Ninguno.

Tabla 27. CRQ-2 Reserva única en la mismas fechas

CRQ-3	Número de tarjeta correcto
Versión	1.0
Autores	Miguel Ángel Navas Galán
Fuentes	Especificación de requisitos
Objetivos asociados	OBJ-2 Gestión de las reservas OBJ-2.3 Gestión de clientes
Requisitos asociados	IRQ-4 Información de las reservas IRQ-6 Información del cliente
Descripción	El sistema solo aceptará una tarjeta que cumpla el algoritmo de Luhn para la validación de números de tarjeta.
Importancia	Alta.
Urgencia	Alta.
Estado	Validado.
Estabilidad	Alta.
Comentarios	Ninguno.

Tabla 28. CRQ-3 Número de tarjeta correcto

5.3.2. Requisitos funcionales

Esta sección contiene la lista de requisitos funcionales que se han identificado.

Identificador	REQ-1
Definición	Configurar la Casa Rural

Autores	Miguel Ángel Navas Galán
Descripción	El usuario de la aplicación puede conectarse y configurar todos los objetos y elementos de la Casa Rural.
Importancia	Alta.
Comentarios	Ninguno.

Tabla 29. REQ-1 El usuario configura la Casa Rural

Identificador	REQ-2
Definición	Reservar una habitación
Autores	Miguel Ángel Navas Galán
Descripción	El usuario de la aplicación puede reservar una habitación.
Importancia	Alta.
Comentarios	Ninguno.

Tabla 30. REQ-2 Reservar una habitación

Identificador	REQ-3
Definición	Anular la reserva de una habitación
Autores	Miguel Ángel Navas Galán
Descripción	El usuario de la aplicación puede anular una reserva de habitación.
Importancia	Alta.
Comentarios	Ninguno.

Tabla 31. REQ-3 Anular reserva de una habitación

Identificador	REQ-4
Definición	Visualizar clientes
Autores	Miguel Ángel Navas Galán
Descripción	El usuario de la aplicación podrá ver los datos de reserva de los clientes.
Importancia	Alta.
Comentarios	Ninguno.

Tabla 32. REQ-4 Visualizar clientes

Definición de actores

Los actores son un conjunto coherente de roles que juegan los usuarios cuando interactúan con los casos de uso y además estimulan el sistema con eventos.

A continuación mostraremos la definición de los actores que hemos detectado que tiene nuestra aplicación:

ACT-1	Gestor
Versión	1.0
Autores	Miguel Ángel Navas Galán
Descripción	Este actor representa el usuario gestor de la Casa Rural.
Comentarios	Este usuario será único por dispositivo móvil.

Tabla 33. ACT-1 Gestor

Diagramas de casos de uso

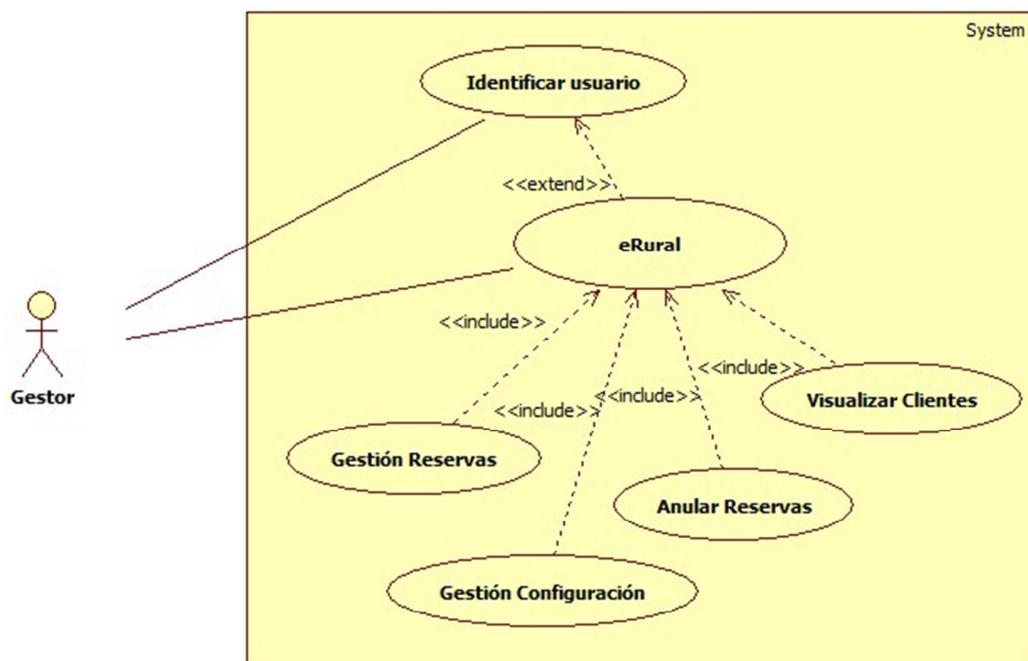


Figura 6. Diagrama de casos de uso nivel 1

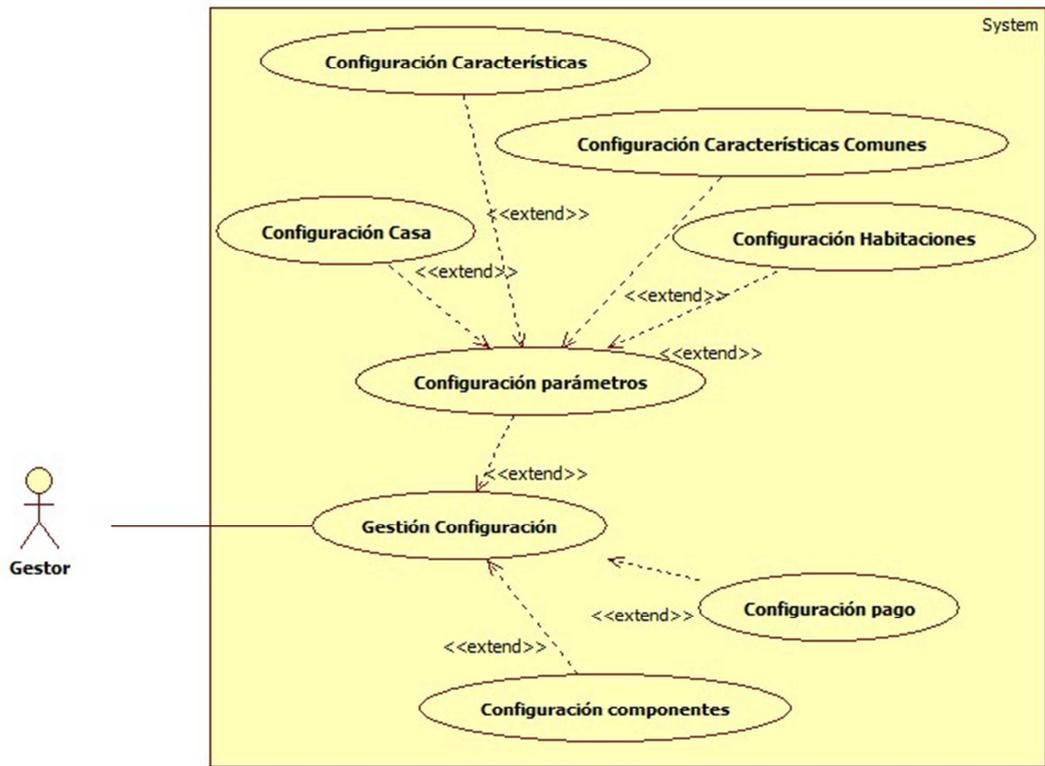


Figura 7. Diagrama de casos de uso para la gestión de la configuración de la aplicación.

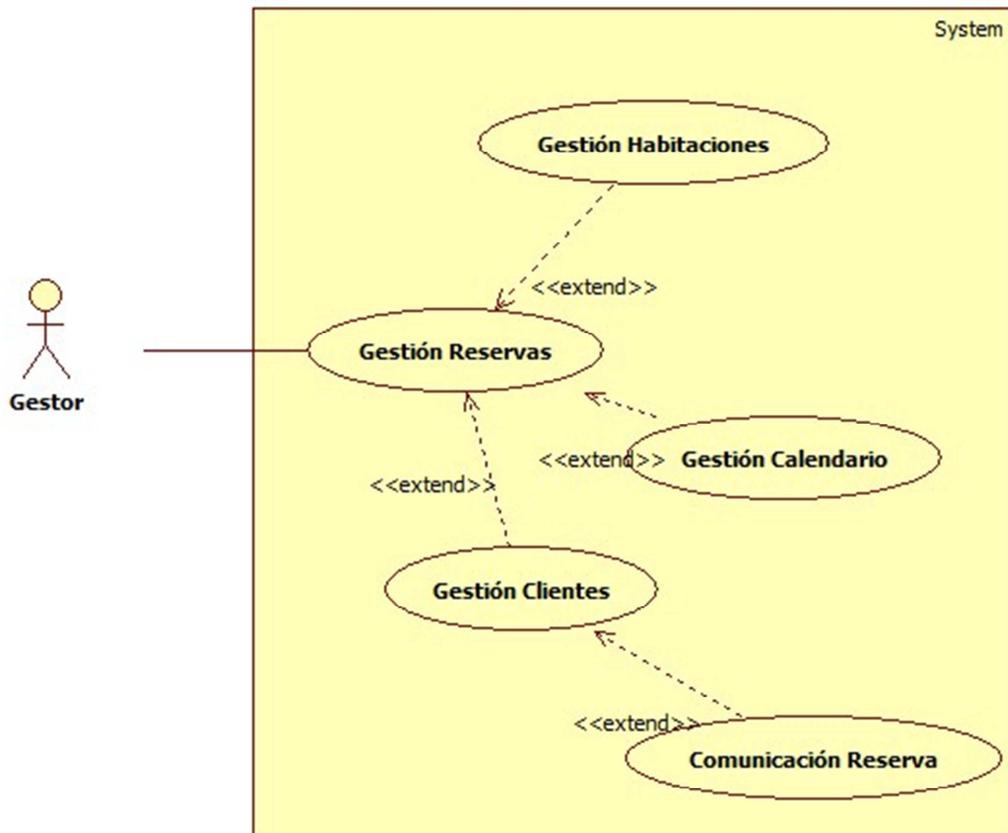


Figura 8. Diagrama de casos de uso para la gestión de reservas de la aplicación

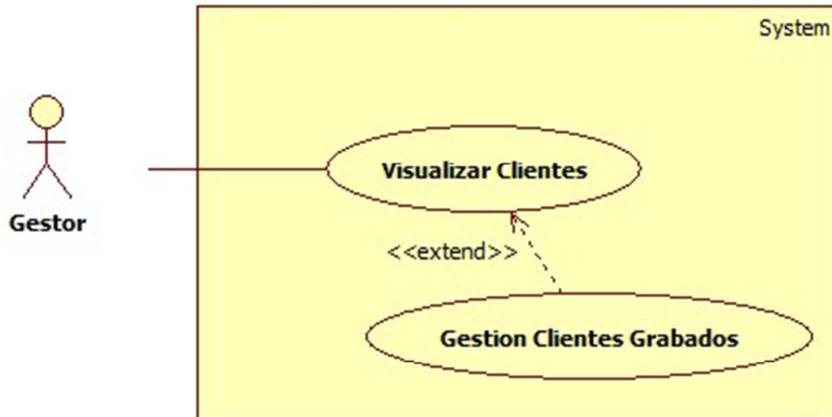


Figura 9. Diagrama de casos de uso para la visualización de clientes de la aplicación

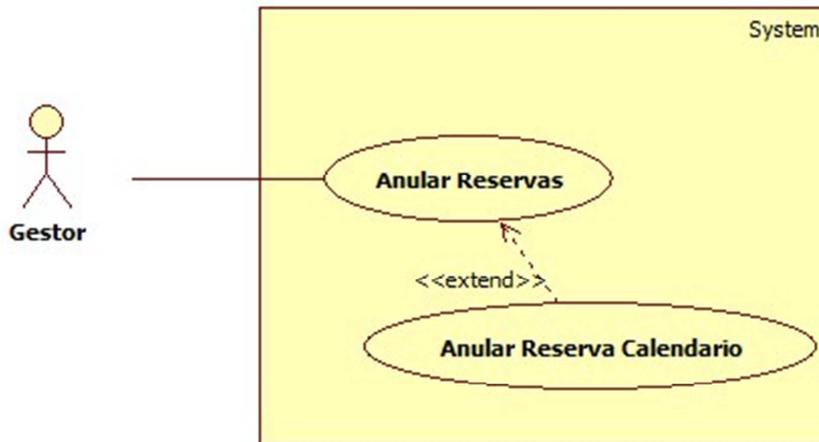


Figura 10. Diagrama de casos de uso para la anulación de reservas de la aplicación

Especificación de casos de uso

A continuación iremos viendo todos los casos de uso que hemos desarrollado para ésta aplicación.

UC-1	Identificar al usuario
Versión	1.0
Autores	Miguel Ángel Navas Galán
Fuentes	Especificación de requisitos
Objetivos asociados	OBJ-1 Gestión de la configuración de la aplicación OBJ-1.1 Configuración de parámetros

Requisitos asociados	IRQ-1 Información usuario gestor	
Descripción	El sistema te pedirá que te identifiques solamente la primera vez que abras la aplicación y se almacenará el usuario en el dispositivo móvil.	
Precondición	Sea un usuario existente.	
Secuencia normal	Paso	Acción
	p1	Al iniciar la aplicación la primera vez nos pedirá que nos identifiquemos con nuestro usuario.
	p2	Accederá a la base de datos para validar el usuario.
	p3	Si el usuario es correcto entrará al menú principal de la aplicación.
Postcondición	Identificado durante toda la vida de la aplicación en el dispositivo, es decir, hasta que se desinstale.	
Excepciones	Paso	Acción
	p2	Usuario incorrecto.
Frecuencia	1 vez cuando se inicie la aplicación.	
Importancia	Alta.	
Urgencia	Alta.	
Estado	Validado.	
Estabilidad	Alta.	
Comentarios	Será la manera de identificar al gestor de una forma segura.	

Tabla 34. UC-1 Identificar al usuario

UC-2	Configuración de la Casa	
Versión	1.0	
Autores	Miguel Ángel Navas Galán	
Fuentes	Especificación de requisitos	
Objetivos asociados	OBJ-1 Gestión de la configuración de la aplicación OBJ-1.1 Configuración de parámetros	
Requisitos asociados	IRQ-2 Información Casa Rural	
Descripción	El sistema te pedirá, solamente la primera vez que inicies la aplicación, que procedas a la configuración la Casa Rural	
Precondición	No haya registrada ya casa.	
Secuencia normal	Paso	Acción
	p1	El usuario introduce los datos en el formulario.
	p2	El sistema valida que los datos introducidos sean correctos

	p3	El sistema almacena la información en base de datos, siempre que los datos sean válidos.
Postcondición	Se puede modificar, pero solo habrá una por aplicación.	
Excepciones	Paso	Acción
	p2	Datos incorrectos.
Frecuencia	1 vez cuando se inicie la aplicación.	
Importancia	Alta.	
Urgencia	Alta.	
Estado	Validado.	
Estabilidad	Alta.	
Comentarios	No se podrá reservar hasta que no se haya creado.	

Tabla 35. UC-2 Configuración de la casa

UC-3	Configuración de las habitaciones	
Versión	1.0	
Autores	Miguel Ángel Navas Galán	
Fuentes	Especificación de requisitos	
Objetivos asociados	OBJ-1 Gestión de la configuración de la aplicación OBJ-1.1 Configuración de parámetros OBJ-2 Gestión de las reservas OBJ-2.1 Gestión de habitaciones	
Requisitos asociados	IRQ-3 Información habitaciones	
Descripción	El sistema te pedirá, solamente la primera vez que inicies la aplicación, que procedas a la configuración de las habitaciones	
Precondición	No haya una habitación con el mismo nombre.	
Secuencia normal	Paso	Acción
	p1	El usuario introduce los datos en el formulario.
	p2	El sistema valida que los datos introducidos sean correctos
	p3	El sistema almacena la información en base de datos, siempre que los datos sean válidos.
Postcondición	Se pueden eliminar.	
Excepciones	Paso	Acción
	p2	Datos incorrectos.
Frecuencia	1 ó N veces / Depende de las habitaciones	
Importancia	Alta.	
Urgencia	Alta.	
Estado	Validado.	
Estabilidad	Alta.	

Comentarios	No se podrá reservar hasta que no se haya creado.
--------------------	---

Tabla 36. UC-3 Configuración de las habitaciones

UC-4	Configuración de las características	
Versión	1.0	
Autores	Miguel Ángel Navas Galán	
Fuentes	Especificación de requisitos	
Objetivos asociados	OBJ-1 Gestión de la configuración de la aplicación OBJ-1.1 Configuración de parámetros	
Requisitos asociados	IRQ-3.1 Información características	
Descripción	El sistema está preparado para albergar características de las habitaciones.	
Precondición	No haya una característica con el mismo nombre para la misma habitación.	
Secuencia normal	Paso	Acción
	p1	El usuario introduce los datos en el formulario.
	p2	El sistema valida que los datos introducidos sean correctos
	p3	El sistema almacena la información en base de datos, siempre que los datos sean válidos.
Postcondición	Se pueden eliminar.	
Excepciones	Paso	Acción
	p2	Datos incorrectos.
Frecuencia	0 ó N veces / Depende de las características	
Importancia	Media.	
Urgencia	Media.	
Estado	Validado.	
Estabilidad	Alta.	
Comentarios	Deben tener características las habitaciones para poder ser reservadas.	

Tabla 37. UC-4 Configuración de las características

UC-5	Configuración de las características comunes	
Versión	1.0	
Autores	Miguel Ángel Navas Galán	
Fuentes	Especificación de requisitos	
Objetivos asociados	OBJ-1 Gestión de la configuración de la aplicación OBJ-1.1 Configuración de parámetros	
Requisitos asociados	IRQ-3.1 Información características	

Descripción	El sistema está preparado para albergar características comunes para todas las habitaciones.	
Precondición	No haya una característica con el mismo nombre.	
Secuencia normal	Paso	Acción
	p1	El usuario introduce los datos en el formulario.
	p2	El sistema valida que los datos introducidos sean correctos
	p3	El sistema almacena la información en base de datos, siempre que los datos sean válidos.
Postcondición	Se pueden eliminar.	
Excepciones	Paso	Acción
	p2	Datos incorrectos.
Frecuencia	0 ó N veces / Depende de las características	
Importancia	Media.	
Urgencia	Media.	
Estado	Validado.	
Estabilidad	Alta.	
Comentarios	Pueden no tener características comunes las habitaciones para poder ser reservadas.	

Tabla 38. UC-5 Configuración de las características comunes

UC-6	Configuración de los componentes	
Versión	1.0	
Autores	Miguel Ángel Navas Galán	
Fuentes	Especificación de requisitos	
Objetivos asociados	OBJ-1 Gestión de la configuración de la aplicación OBJ-1.2 Configuración de componentes	
Requisitos asociados	IRQ-5 Información del calendario	
Descripción	El sistema está preparado para leer las cuentas de calendarios que hay y registrar sobre la que se va a actuar.	
Precondición	Debe haber cuentas de calendario.	
Secuencia normal	Paso	Acción
	p1	El usuario selecciona el calendario de una lista.
	p2	El sistema valida que los datos introducidos sean correctos
	p3	El sistema almacena la información en el archivo de configuración, siempre que los datos sean válidos.
Postcondición	Se pueden eliminar.	
Excepciones	Paso	Acción

	p2	Datos incorrectos.
Frecuencia	1	
Importancia	Alta.	
Urgencia	Alta.	
Estado	Validado.	
Estabilidad	Alta.	
Comentarios	Solo se podrá asociar a la cuenta de un calendario y correo electrónico, porque esta cuenta de correo electrónico va a ser también la utilizada para mandar los e-mails de confirmación de reserva.	

Tabla 39. UC-6 Configuración de los componentes

UC-7	Configuración del pago	
Versión	1.0	
Autores	Miguel Ángel Navas Galán	
Fuentes	Especificación de requisitos	
Objetivos asociados	OBJ-1 Gestión de la configuración de la aplicación OBJ-1.3 Configuración del pago	
Requisitos asociados	IRQ-7 Información del pago	
Descripción	El sistema está preparado para leer la configuración del pago y según sea comunicárselo al cliente vía mail.	
Precondición	Ninguna.	
Secuencia normal	Paso	Acción
	p1	El usuario introduce los datos en el formulario.
	p2	El sistema valida que los datos introducidos sean correctos
	p3	El sistema almacena la información en el archivo de configuración, siempre que los datos sean válidos.
Postcondición	Se pueden eliminar.	
Excepciones	Paso	Acción
	p2	Datos incorrectos.
Frecuencia	1	
Importancia	Alta.	
Urgencia	Alta.	
Estado	Validado.	
Estabilidad	Alta.	
Comentarios	Ninguno.	

Tabla 40. UC-7 Configuración del pago

UC-8	Gestión de las habitaciones	
Versión	1.0	
Autores	Miguel Ángel Navas Galán	
Fuentes	Especificación de requisitos	
Objetivos asociados	OBJ-2 Gestión de las reservas OBJ-2.1 Gestión de habitaciones	
Requisitos asociados	IRQ-4 Información de las reservas	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando proceda a la reserva de una habitación.	
Precondición	Debe haber cuentas de calendario.	
Secuencia normal	Paso	Acción
	p1	El usuario selecciona la habitación de una lista.
	p2	El sistema procede al siguiente paso de la reserva.
Postcondición	Se pueden eliminar.	
Excepciones	Ninguna.	
Frecuencia	1	
Importancia	Alta.	
Urgencia	Alta.	
Estado	Validado.	
Estabilidad	Alta.	
Comentarios	Solo se podrá escoger habitaciones para proceder la reserva.	

Tabla 41. UC-8 Gestión de las habitaciones

UC-9	Gestión del Calendario	
Versión	1.0	
Autores	Miguel Ángel Navas Galán	
Fuentes	Especificación de requisitos	
Objetivos asociados	OBJ-2 Gestión de las reservas OBJ-2.2 Gestión calendario	
Requisitos asociados	IRQ-4 Información de las reservas IRQ-5 Información del calendario	
Descripción	El sistema está preparado para gestionar la reserva leyendo el calendario previamente elegido para el registro de una reserva.	
Precondición	Debe haber cuenta de calendario y habitación seleccionada.	
Secuencia normal	Paso	Acción
	p1	El usuario unas fechas.
	p2	El sistema valida que los datos introducidos sean correctos, si se puede reservar

	p3	El sistema almacena la información en el calendario y en la base de datos, siempre que los datos sean válidos.
Postcondición	Se pueden eliminar.	
Excepciones	Paso	Acción
	p2	Fechas incorrectas.
Frecuencia	N	
Importancia	Alta.	
Urgencia	Alta.	
Estado	Validado.	
Estabilidad	Alta.	
Comentarios	No se podrá reservar una misma habitación mientras se solapen en horas una reserva y otra, se puede reservar el mismo día si una termina y otra comienza.	

Tabla 42. UC-9 Gestión del Calendario

UC-10	Gestión de los clientes	
Versión	1.0	
Autores	Miguel Ángel Navas Galán	
Fuentes	Especificación de requisitos	
Objetivos asociados	OBJ-2 Gestión de las reservas OBJ-2.3 Gestión de clientes	
Requisitos asociados	IRQ-4 Información de las reservas IRQ-6 Información del cliente	
Descripción	El sistema está preparado para gestionar los clientes asociados a cada reserva.	
Precondición	Debe haberse podido realizar la reserva.	
Secuencia normal	Paso	Acción
	p1	El usuario rellena el formulario de datos del cliente.
	p2	El sistema valida que los datos introducidos sean correctos
	p3	El sistema almacena la información en el calendario y en la base de datos, siempre que los datos sean válidos.
	p4	Si el usuario ha informado el mail correctamente se le comunica el estado de la reserva vía e-mail.
Postcondición	Ninguna.	
Excepciones	Paso	Acción
	p2	Dirección de correo incorrecto. Nombre obligatorio.
Frecuencia	N	
Importancia	Media.	
Urgencia	Media.	
Estado	Validado.	

Estabilidad	Alta.
Comentarios	Una vez confirmada la posible reserva, se solicita si se quiere grabar cliente y si se desea contactar con él, al hacerlo se graba la reserva en el calendario y en base de datos.

Tabla 43. UC-10 Gestión de los clientes

UC-11	Comunicación de la reserva	
Versión	1.0	
Autores	Miguel Ángel Navas Galán	
Fuentes	Especificación de requisitos	
Objetivos asociados	OBJ-1.2 Configuración de componentes OBJ-2 Gestión de las reservas OBJ-2.3 Gestión de clientes	
Requisitos asociados	IRQ-4 Información de las reservas	
Descripción	El sistema está preparado para gestionar la comunicación con clientes asociados a cada reserva.	
Precondición	Debe haberse podido realizar la reserva y haberse registrado correctamente el cliente.	
Secuencia normal	Paso	Acción
	p1	El usuario confirma que desea mandar la comunicación al cliente.
	p4	El sistema se comunica con el servicio de mail del dispositivo y se envía el e-mail.
Postcondición	Ninguna.	
Excepciones	Paso	Acción
	p2	Dirección de correo incorrecto. Nombre obligatorio.
Frecuencia	N	
Importancia	Media.	
Urgencia	Media.	
Estado	Validado.	
Estabilidad	Alta.	
Comentarios	Una vez grabada la reserva, se le manda el correo.	

Tabla 44. UC-11 Comunicación de la reserva

UC-12	Gestión Clientes Grabados
Versión	1.0
Autores	Miguel Ángel Navas Galán
Fuentes	Especificación de requisitos
Objetivos asociados	OBJ-2 Gestión de las reservas OBJ-2.3 Gestión de clientes

Requisitos asociados	IRQ-4 Información de las reservas IRQ-6 Información del cliente	
Descripción	El sistema está preparado para gestionar la lista de clientes asociados a cada reserva.	
Precondición	Debe haberse podido realizar la reserva y haberse registrado correctamente el cliente.	
Secuencia normal	Paso	Acción
	p1	El usuario lista los clientes.
	p2	El sistema se comunica con la base de datos y lista todos los clientes.
Postcondición	Ninguna.	
Excepciones	Ninguna.	
Frecuencia	N	
Importancia	Media.	
Urgencia	Media.	
Estado	Validado.	
Estabilidad	Alta.	
Comentarios	Serán listados por orden descendente para su consulta.	

Tabla 45. UC-12 Gestión clientes grabados

UC-13	Anular Reserva del Calendario	
Versión	1.0	
Autores	Miguel Ángel Navas Galán	
Fuentes	Especificación de requisitos	
Objetivos asociados	OBJ-2 Gestión de las reservas OBJ-2.2 Gestión calendario	
Requisitos asociados	IRQ-4 Información de las reservas IRQ-5 Información del calendario IRQ-6 Información del cliente	
Descripción	El sistema está preparado para anular una reserva.	
Precondición	Debe haberse podido realizar la reserva y haberse registrado correctamente el cliente.	
Secuencia normal	Paso	Acción
	p1	El usuario lista las reservas con sus respectivos clientes.
	p2	El usuario elegirá la reserva a anular y el sistema procederá a su eliminación del calendario.
Postcondición	Ninguna.	
Excepciones	Ninguna.	
Frecuencia	N	
Importancia	Media.	
Urgencia	Media.	

Estado	Validado.
Estabilidad	Alta.
Comentarios	Ninguno.

Tabla 46. UC-13 Anular reserva calendario

5.3.3. Requisitos no funcionales

Un requisito no funcional es una característica requerida del sistema, del proceso de desarrollo, del servicio prestado o de cualquier otro aspecto del desarrollo, que señala una restricción del mismo. Vamos a exponerlos según su tipo de restricción:

Rendimiento

NFR-1	Rendimiento fluido
Versión	1.0
Autores	Miguel Ángel Navas Galán
Objetivos asociados	OBJ-1 Gestión de la configuración de la aplicación OBJ-1.1 Configuración de parámetros OBJ-1.2 Configuración de componentes OBJ-1.3 Configuración del pago OBJ-2 Gestión de las reservas OBJ-2.1 Gestión de habitaciones OBJ-2.2 Gestión calendario OBJ-2.3 Gestión de clientes
Descripción	El sistema debe interactuar con la base de datos y con los calendarios con cierta ligereza.
Importancia	Alta.
Urgencia	Media.
Estado	Validado.
Estabilidad	Alta.
Comentarios	Ninguno.

Tabla 47. NFR-1 Rendimiento fluido

Interfaz gráfica de usuario

NFR-2	Interfaz amigable, usable y accesible
Versión	1.0
Autores	Miguel Ángel Navas Galán

Objetivos asociados	OBJ-1 Gestión de la configuración de la aplicación OBJ-1.1 Configuración de parámetros OBJ-1.2 Configuración de componentes OBJ-1.3 Configuración del pago OBJ-2 Gestión de las reservas OBJ-2.1 Gestión de habitaciones OBJ-2.2 Gestión calendario OBJ-2.3 Gestión de clientes
Descripción	El sistema deberá tener una interfaz amigable para el usuario, cuyo funcionamiento sea fácil e intuitivo.
Importancia	Alta.
Urgencia	Media.
Estado	Validado.
Estabilidad	Alta.
Comentarios	Se ha mantenido componentes Android utilizados que son comunes en otras muchas aplicaciones del sistema operativo.

Tabla 48. NFR-2 Interfaz amigable, usable y accesible

Compatibilidad

NFR-3	Compatibilidad entre versiones de Android
Versión	1.0
Autores	Miguel Ángel Navas Galán
Objetivos asociados	OBJ-1 Gestión de la configuración de la aplicación OBJ-1.1 Configuración de parámetros OBJ-1.2 Configuración de componentes OBJ-1.3 Configuración del pago OBJ-2 Gestión de las reservas OBJ-2.1 Gestión de habitaciones OBJ-2.2 Gestión calendario OBJ-2.3 Gestión de clientes
Descripción	El sistema está diseñado para las versiones 4.1 y superiores.
Importancia	Alta.
Urgencia	Media.
Estado	Validado.
Estabilidad	Alta.
Comentarios	El 95% de los dispositivos de Android están entre las versiones 4.0 y 5.1 actualmente.

Tabla 49. NFR-3 Compatibilidad entre versiones de Android

Interoperabilidad

NFR-4	Necesidad de cuenta de correo registrada
Versión	1.0
Autores	Miguel Ángel Navas Galán
Objetivos asociados	OBJ-1.2 Configuración de componentes OBJ-2 Gestión de las reservas OBJ-2.1 Gestión de habitaciones OBJ-2.2 Gestión calendario OBJ-2.3 Gestión de clientes
Descripción	El sistema debe tener una cuenta asociada al dispositivo móvil y a su vez al Calendario Google.
Importancia	Alta.
Urgencia	Alta.
Estado	Validado.
Estabilidad	Alta.
Comentarios	Ninguno.

Tabla 50. NFR-4 Necesidad de cuenta de correo registrada

Seguridad

NFR-5	Seguridad
Versión	1.0
Autores	Miguel Ángel Navas Galán
Objetivos asociados	OBJ-1 Gestión de la configuración de la aplicación
Descripción	El sistema debe autenticar al usuario gestor.
Importancia	Alta.
Urgencia	Alta.
Estado	Validado.
Estabilidad	Alta.
Comentarios	El sistema debe manejar esta información encriptada, para mayor seguridad.

Tabla 51. NFR-5 Seguridad.

5.4. Matriz de rastreabilidad objetivos / requisitos

De esta forma podemos comprobar con facilidad las relaciones entre objetivos y requisitos:

	OBJ-1	OBJ-1.1	OBJ-1.2	OBJ-1.3	OBJ-2	OBJ-2.1	OBJ-2.2	OBJ-2.3
IRQ-1	*	*						
IRQ-2	*	*						
IRQ-3	*	*			*	*		

IRQ-3.1	*	*			*	*		
IRQ-4					*			
IRQ-5			*		*		*	
IRQ-6					*			*
IRQ-7				*	*			*
CRQ-1	*	*						
CRQ-2					*			
CRQ-3					*			*
UC-1	*	*						
UC-2	*	*						
UC-3	*	*			*	*		
UC-4	*	*						
UC-5	*	*						
UC-6	*		*					
UC-7	*			*				
UC-8					*	*		
UC-9					*		*	
UC-10					*			*
UC-11			*		*			*
UC-12					*			*
UC-13					*		*	
NFR-1	*	*	*	*	*	*	*	*
NFR-2	*	*	*	*	*	*	*	*
NFR-3	*	*	*	*	*	*	*	*
NFR-4			*		*	*	*	*
NFR-5	*							

Tabla 52. Matriz de rastreabilidad.

5.5. Resumen

TIPO	ID	Descripción
ACTORES	ACT-1	Gestor
OBJETIVOS	OBJ-1	Gestión de la configuración de la aplicación
	OBJ-1.1	Configuración de parámetros
	OBJ-1.2	Configuración de componentes
	OBJ-1.3	Configuración del pago
	OBJ-2	Gestión de las reservas
	OBJ-2.1	Gestión de habitaciones
	OBJ-2.2	Gestión calendario
	OBJ-2.3	Gestión de clientes
REQUISITOS GENERALES	REQ-1	Configurar la Casa Rural
	REQ-2	Reservar una habitación
	REQ-3	Anular reserva
	REQ-4	Visualizar clientes
REQUISITOS INFORMACION	IRQ-1	Información usuario gestor
	IRQ-2	Información Casa Rural
	IRQ-3	Información habitaciones
	IRQ-3.1	Información características
	IRQ-4	Información de las reservas
	IRQ-5	Información del calendario
	IRQ-6	Información del cliente
IRQ-7	Información del pago	
REQUISITOS	CRQ-1	Único usuario gestor

RESTRICCIÓN	CRQ-2	Reserva única en la mismas fechas
	CRQ-3	Número de tarjeta correcto
REQUISITOS FUNCIONALES	UC-1	Identificar al usuario
	UC-2	Configuración de la Casa
	UC-3	Configuración de las habitaciones
	UC-4	Configuración de las características
	UC-5	Configuración de las características comunes
	UC-6	Configuración de los componentes
	UC-7	Configuración del pago
	UC-8	Gestión de las habitaciones
	UC-9	Gestión del Calendario
	UC-10	Gestión de los clientes
	UC-11	Comunicación de la reserva
	UC-12	Gestión clientes grabados
	UC-13	Anular reserva del Calendario
REQUISITOS NO FUNCIONALES	NFR-1	Rendimiento fluido
	NFR-2	Interfaz amigable, usable y accesible
	NFR-3	Compatibilidad entre versiones de Android
	NFR-4	Necesidad de cuenta de correo registrada
	NFR-5	Seguridad

Tabla 53. Resumen de requisitos.

5.6. Estudio de viabilidad

Para que un proyecto pueda llegar a ser una realidad es necesario realizar una serie de estudios previos que garanticen la viabilidad del proyecto. A continuación explicaremos estos estudios (viabilidad técnica, económica y legal) para poder saber si el proyecto será finalmente viable.

Viabilidad técnica u operativa

Dentro de este tipo de viabilidad se valorará cual es la situación de partida desde el

punto de vista técnico u operacional y cuáles serán los pasos necesarios a seguir para solucionar las posibles complicaciones con las que podemos encontrarnos.

Contaba con ordenador, Internet, libros, propios dispositivos móviles y un lugar adecuado para poder trabajar sobre el proyecto. A su vez, el estudio de numerosos manuales y tutoriales aportaron información sobre el lenguaje Java y sobre todo de Android, concretamente del SDK de Android en su web para desarrolladores.

Teniendo en cuenta todos los soportes técnicos con los que cuento y formándome en conocimientos útiles, se concluye que existe viabilidad técnica para poder llevar a cabo este trabajo.

Viabilidad económica

El apartado relacionado con la viabilidad económica es uno de los más complejos. Una vez obtenidos los resultados se ha de analizar si, en base a ellos, el proyecto es rentable o no.

Según la estimación de esfuerzo realizada anteriormente en el apartado del “Planificación y presupuesto” se concluía que el presupuesto es de 14.392,70 €.

Como podemos observar, se concluye que la aplicación debería ser vendida por una cantidad superior a dicha estimación, sin olvidar que el cliente podría haber propuesto unos requisitos económicos. Pero en este caso se trata de una aplicación genérica, con el objetivo de ser vendida a un cliente por el momento. Se podría vender la aplicación por un coste inferior en caso de contar con un número suficiente de clientes, aunque previamente debemos estimar los clientes potenciales y dar comienzo negociaciones o propuestas con éstos para poder estudiar su aceptación en el mercado, para que finalmente la aplicación sea rentable.

Viabilidad legal

En relación a la viabilidad legal se han de tener en cuenta que el único aspecto en que el proyecto puede infringir alguna ley vigente es en el de la ley orgánica de protección de datos (LOPD). Ésta establece las obligaciones que los responsables de las aplicaciones informáticas han de garantizar el derecho a la protección de los datos de carácter personal.

De esta manera se puede concluir que para que el software sea viable son necesarias unas medidas de seguridad que permitan al cliente utilizar la aplicación sin vulnerar dicha ley. Que en esta aplicación estamos cumpliendo al ser accesible solo por nuestra aplicación la base de datos.

6. Análisis del sistema

En este documento se explicará el proyecto desde un punto de vista de análisis. Para empezar analizaremos el modelo estático del sistema. Después pasaremos al modelo de comportamiento del sistema, en el que examinaremos el comportamiento de los casos de uso explicados en el anterior apartado. Para ello, empezaremos con los diagramas de secuencia y de actividad más importantes para la comprensión de la aplicación.

6.1. Modelo estático del sistema

Esta sección describe el modelo estático del sistema, formado por la descripción de los diferentes tipos de objetos y asociaciones obtenidos durante la fase de análisis de requisitos.

6.1.1. Diagrama de tipos de objetos

En este apartado se presentan los diagramas UML de los subsistemas y tipos de objetos resultantes de la fase de análisis de requisitos previa.

Para la facilidad de análisis del diagrama de clases y evitar que sea demasiado grande, se dividirá en varias partes. Los diagramas mostrarán las clases e interfaces del sistema.

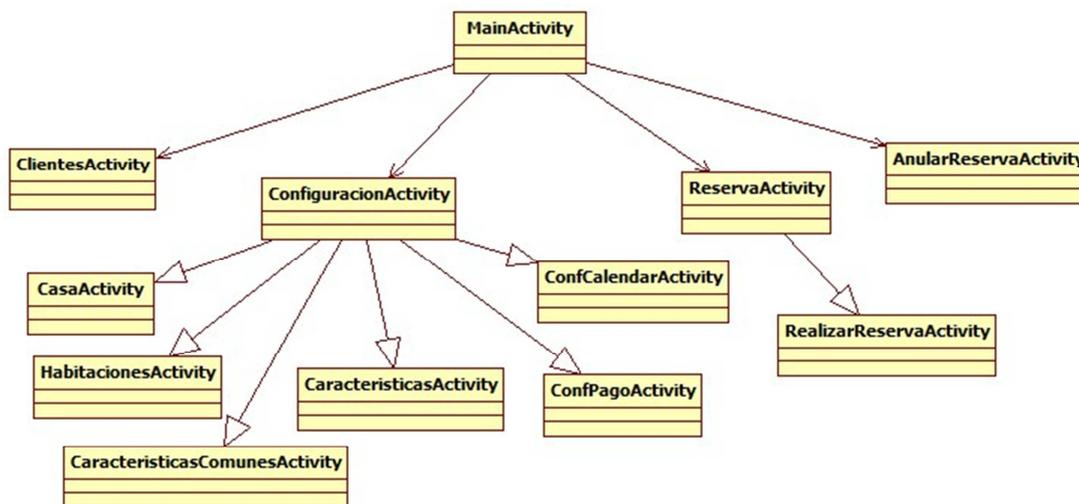


Figura 11. Diagrama de tipos de objetos general de la aplicación

En esta figura queremos dar una visión general de la aplicación, vamos a ver los siguientes diagramas en los que se visualiza mejor sus cometidos: el configurar la

aplicación, el de reservar y el de anular reservas.

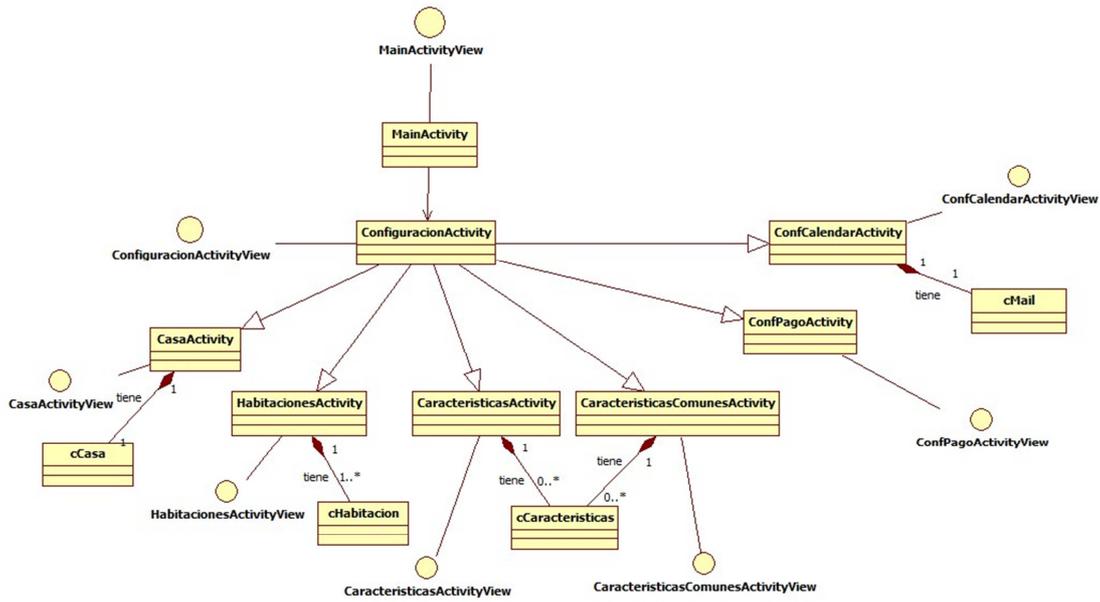


Figura 12. Diagrama de tipos de objetos para la configuración de la aplicación

Esta parte sería la parte inicial en la que el gestor de la aplicación configurará la casa rural según sean sus características, y también configurará la cuenta a la que irá asociada el calendario de reservas de la aplicación.

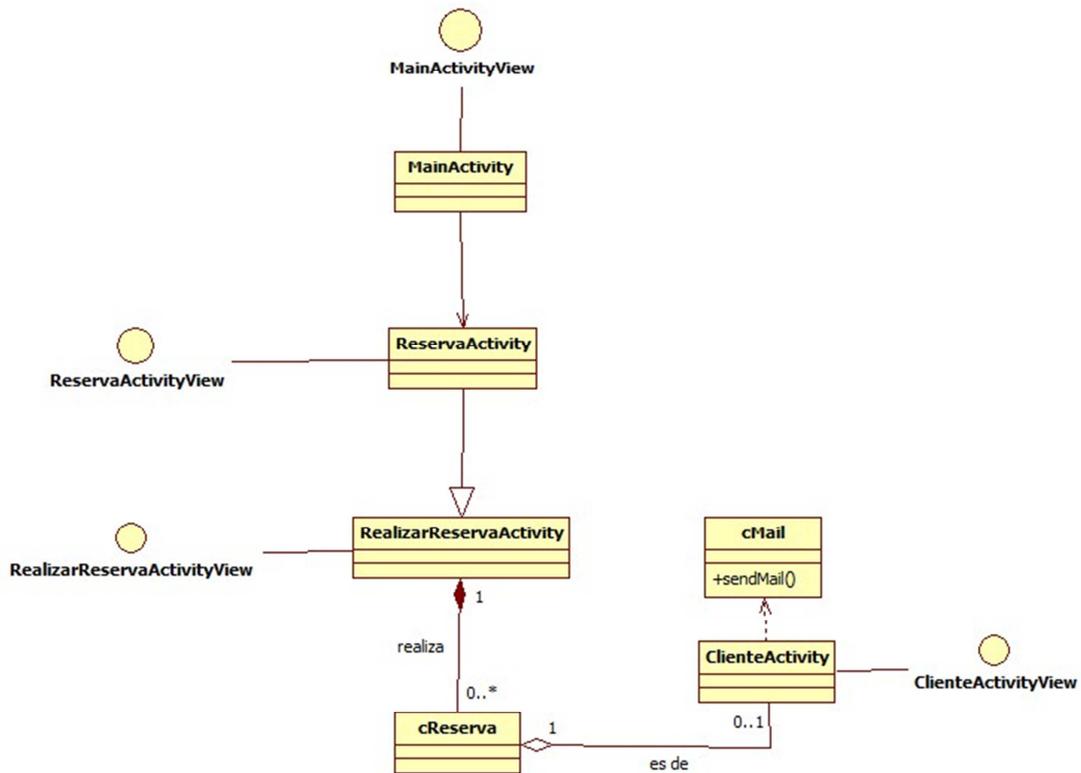


Figura 13. Diagrama de tipos de objetos para la acción de reservar de la aplicación

En este diagrama se ve con claridad que es la parte central de la aplicación, el acto de reservar habitaciones y llevar su gestión en nuestro dispositivo móvil.

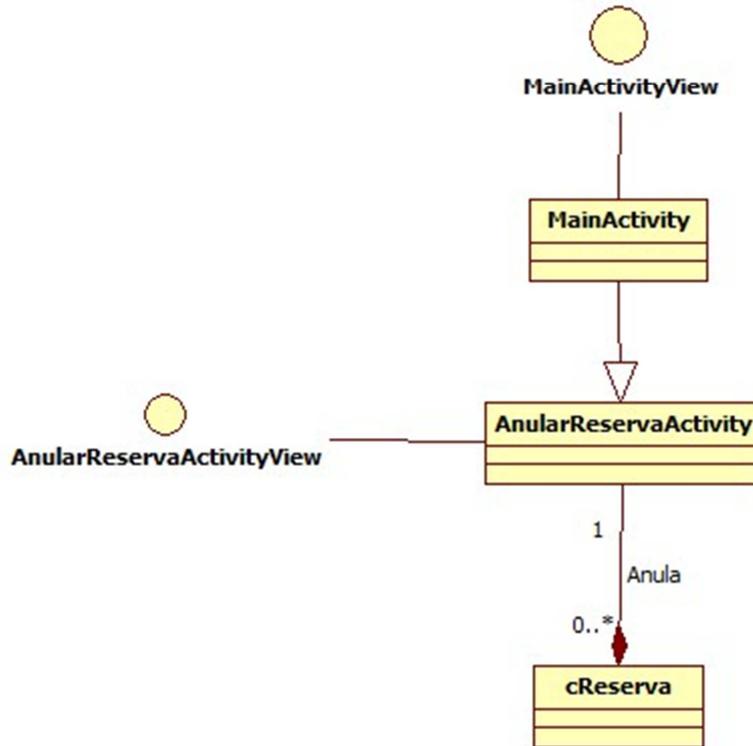


Figura 14. Diagrama de tipos de objetos para la acción de anular reserva de la aplicación

En este diagrama se ve con claridad el acto de anular reservas de habitaciones actualizando nuestro calendario asociado.

Después de ver los diagramas de clases, pasaremos a ver detalladamente cada una de las clases que forman la aplicación en el siguiente apartado de diseño del sistema.

6.2. Modelo de comportamiento del sistema

6.2.1. Diagramas de secuencia del sistema

Diagramas de secuencia del subsistema de configuración de la aplicación.

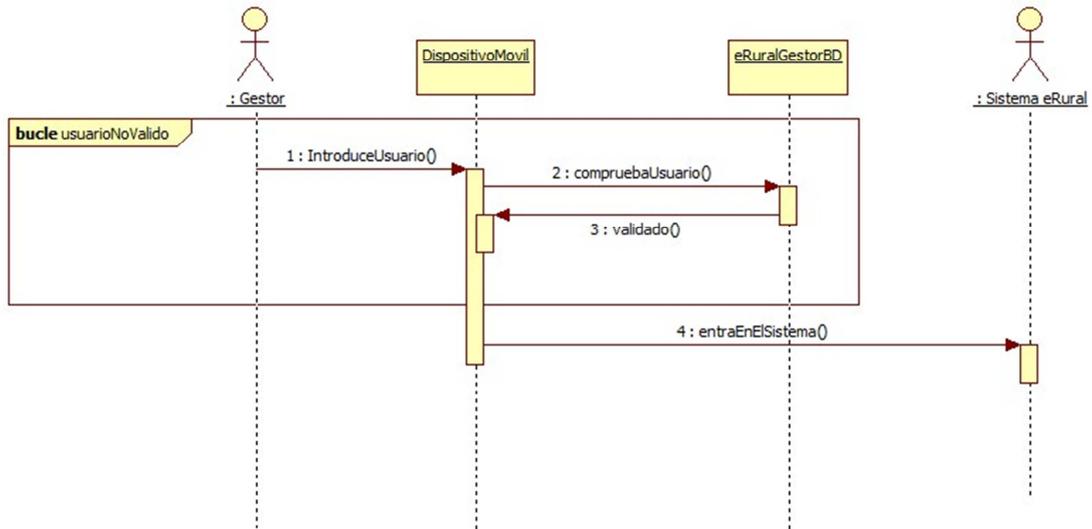


Figura 15. Diagrama de secuencia para el UC-1

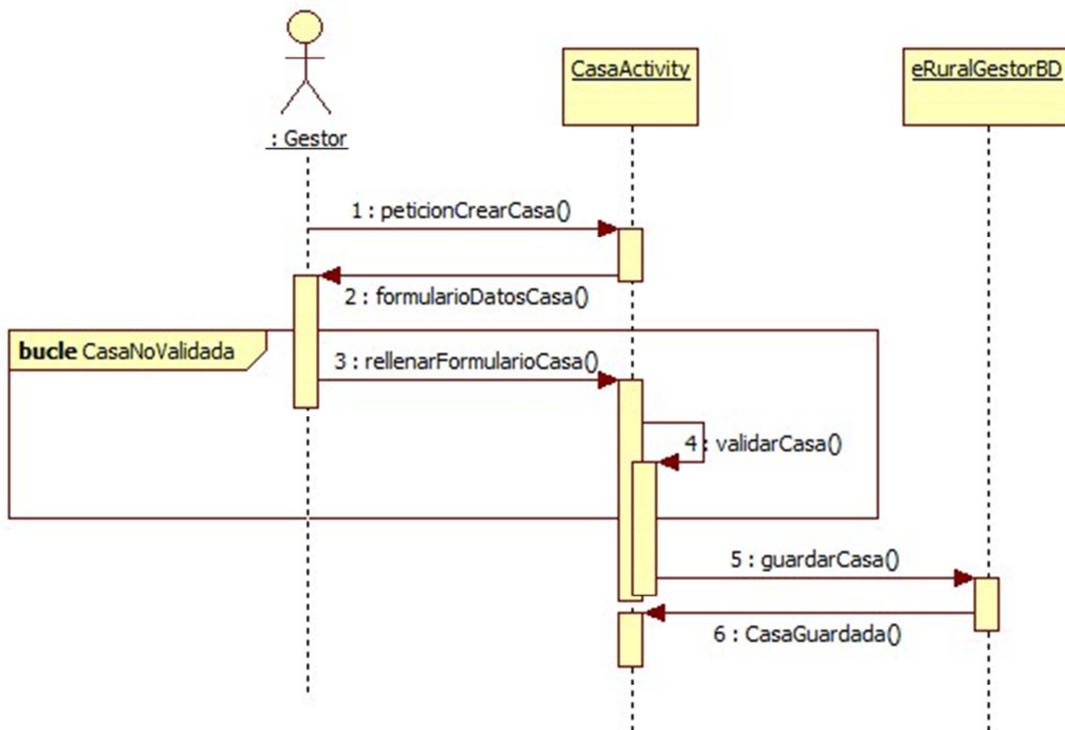


Figura 16. Diagrama de secuencia para el UC-2

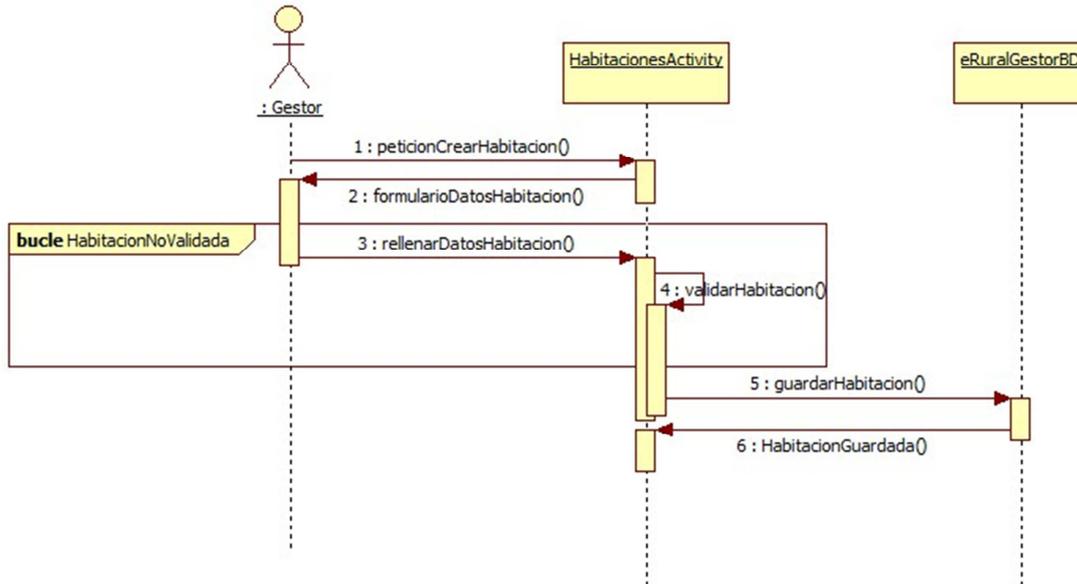


Figura 17. Diagrama de secuencia para el UC-3

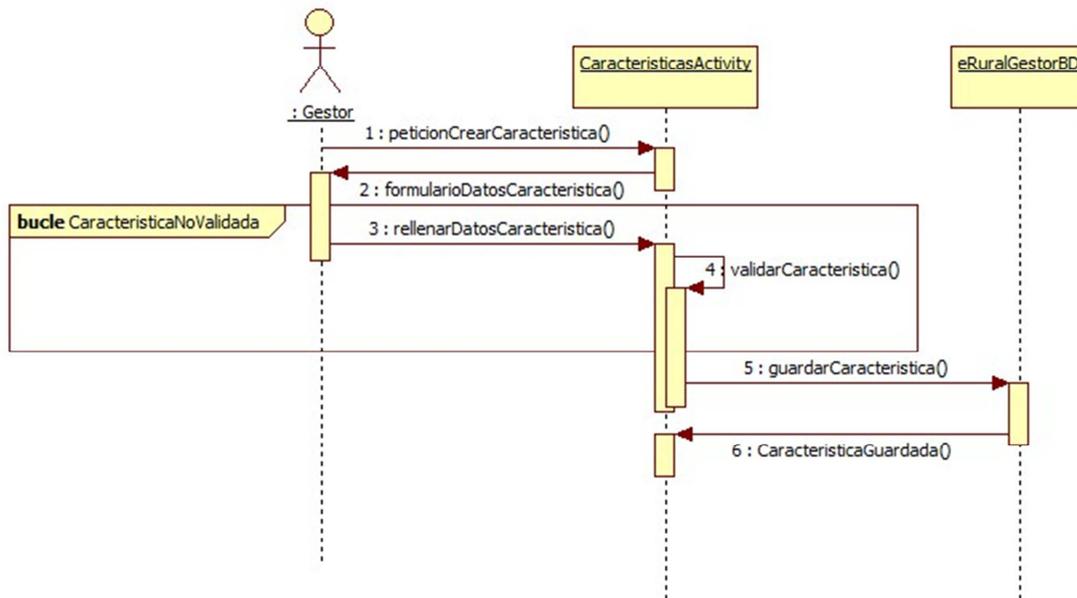


Figura 18. Diagrama de secuencia para el UC-4

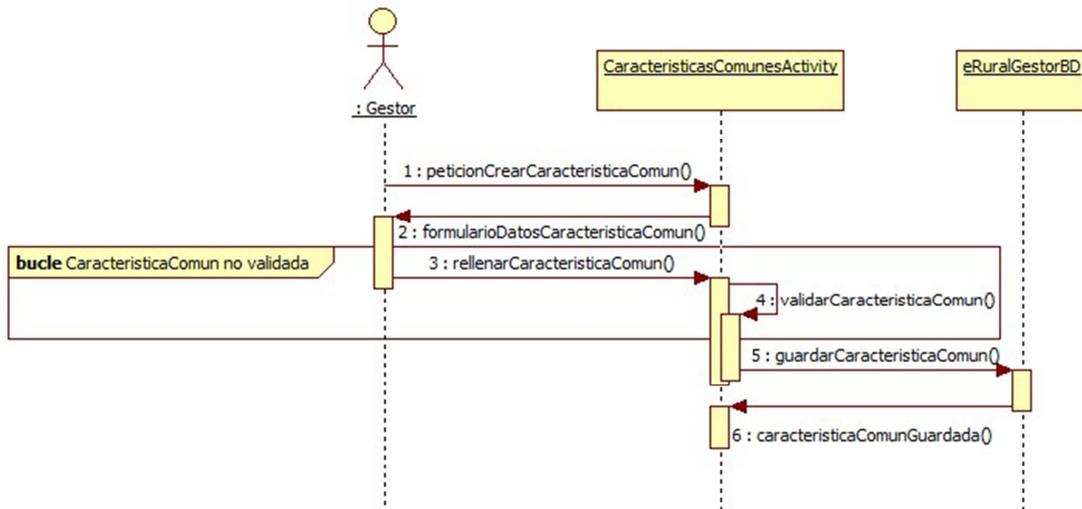


Figura 19. Diagrama de secuencia para el UC-5

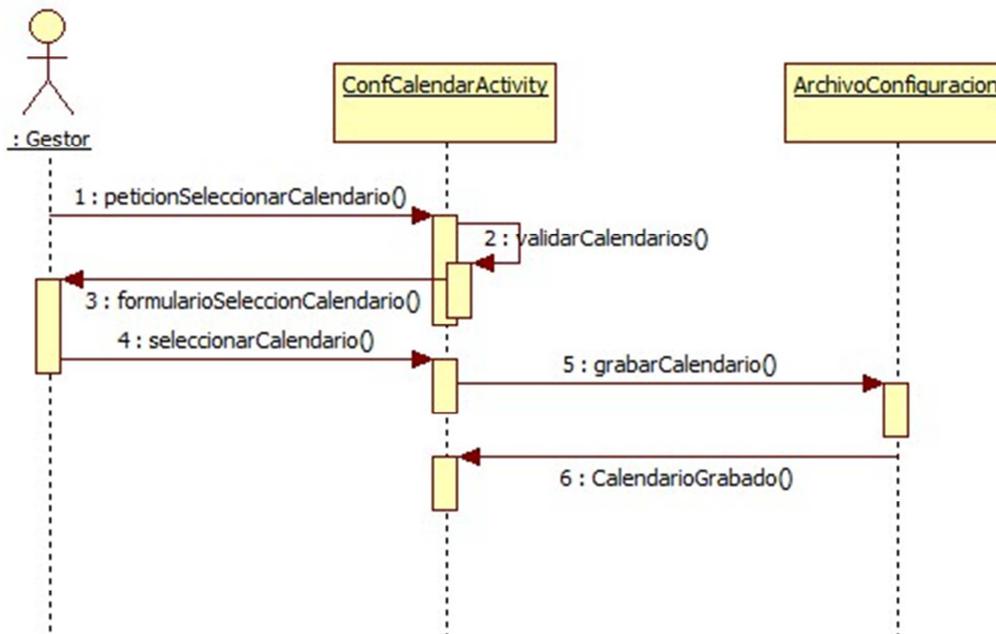


Figura 20. Diagrama de secuencia para el UC-6

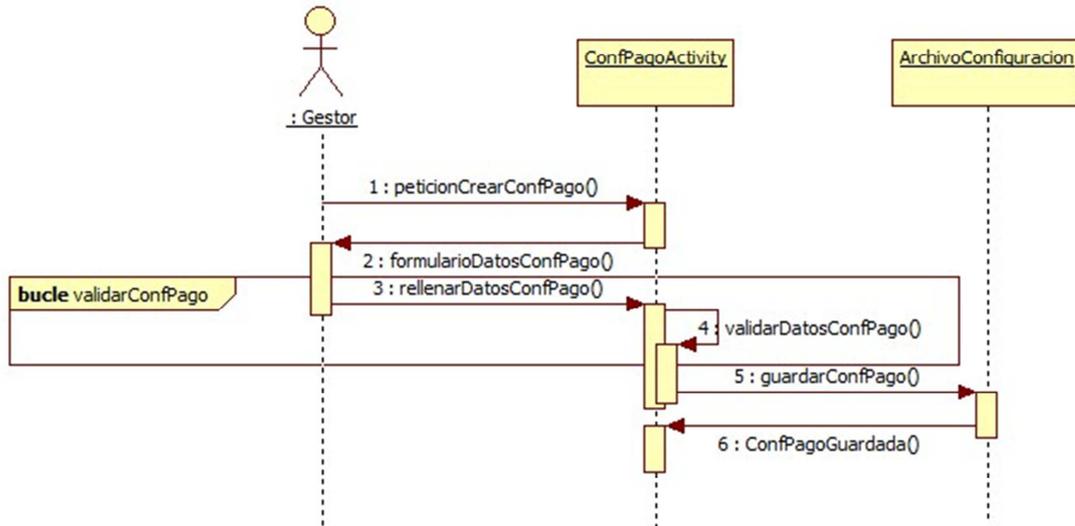


Figura 21. Diagrama de secuencia para el UC-7

Diagramas de secuencia del subsistema de gestión de reservas de la aplicación.

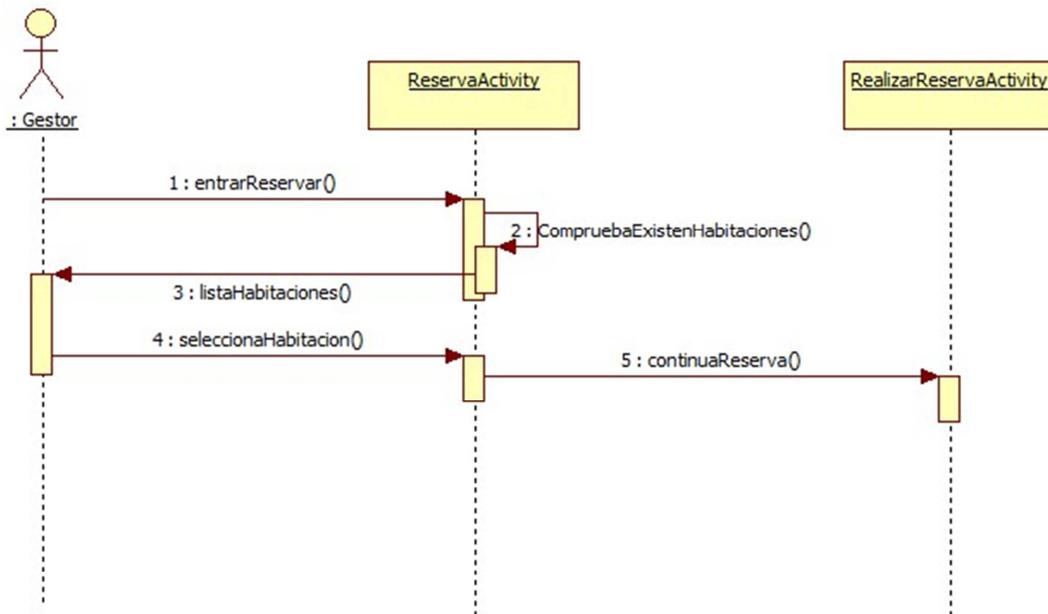


Figura 22. Diagrama de secuencia para el UC-8

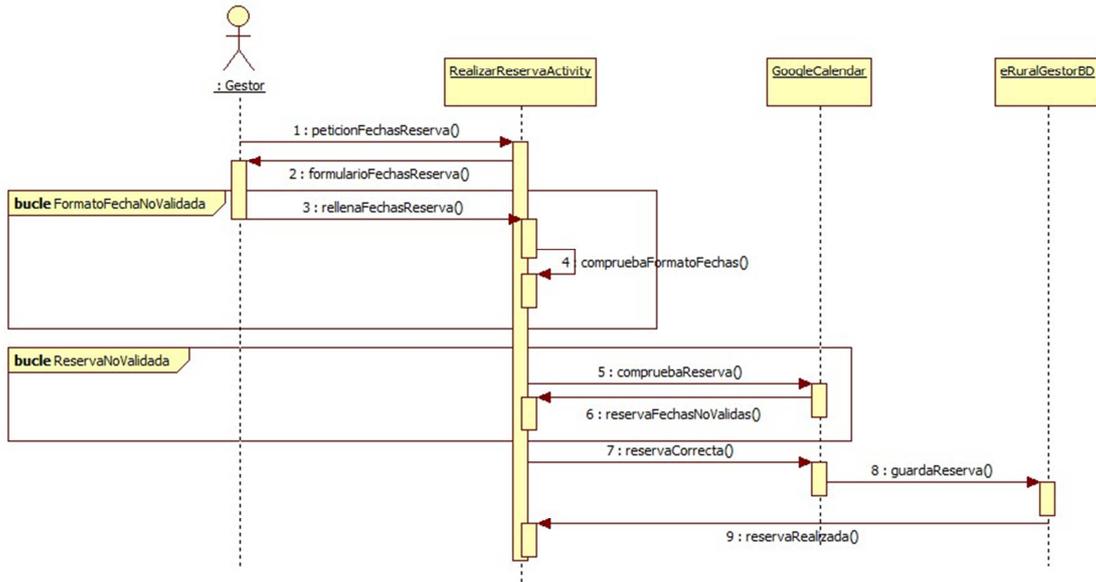


Figura 23. Diagrama de secuencia para el UC-9

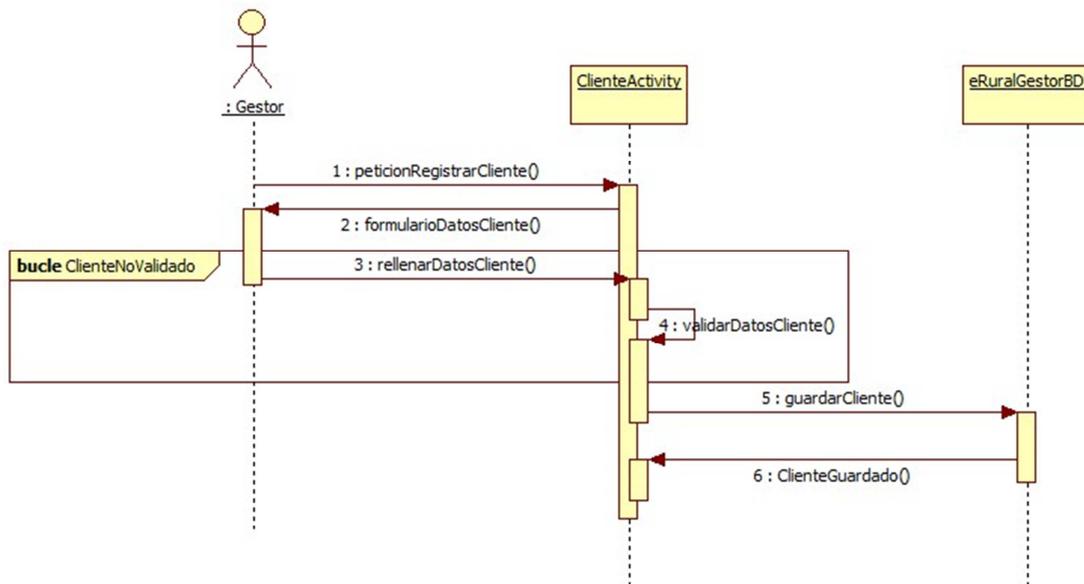


Figura 24. Diagrama de secuencia para el UC-10

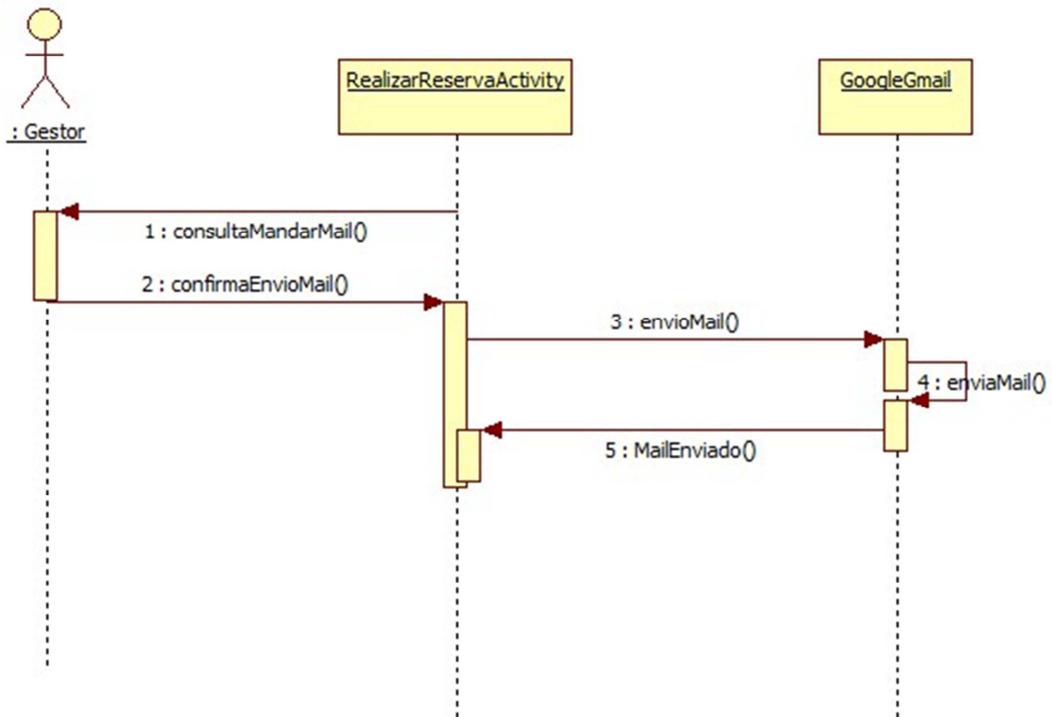


Figura 25. Diagrama de secuencia para el UC-11

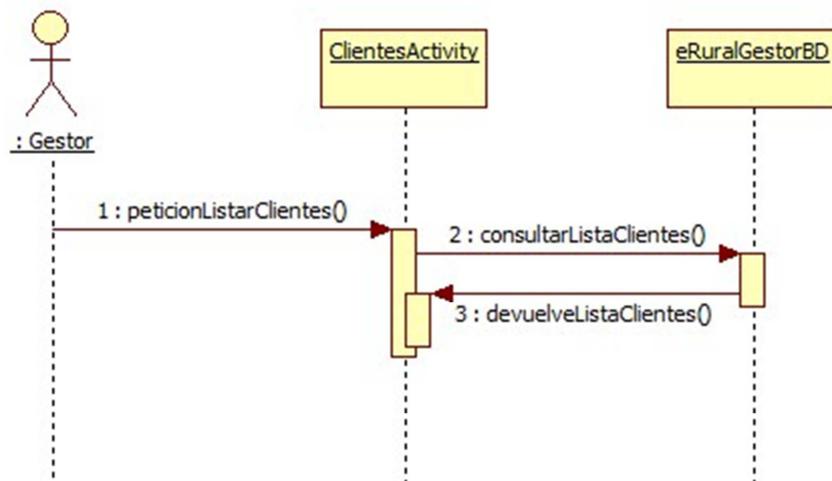


Figura 26. Diagrama de secuencia para el UC-12

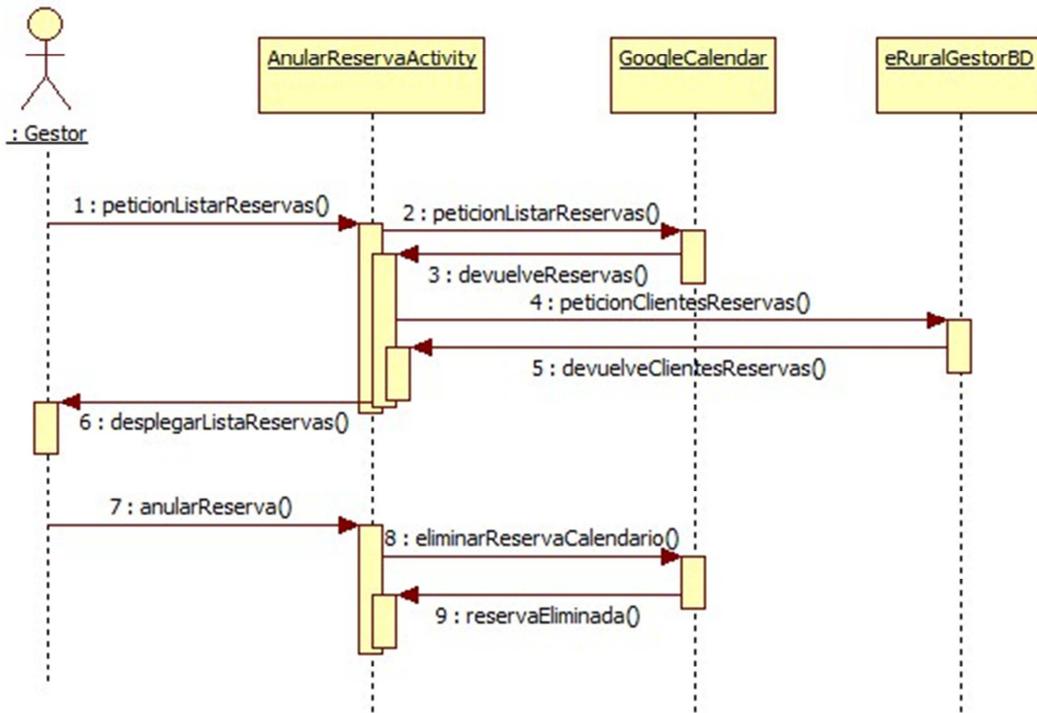


Figura 27. Diagrama de secuencia para el UC-13

6.2.2. Diagrama de actividad

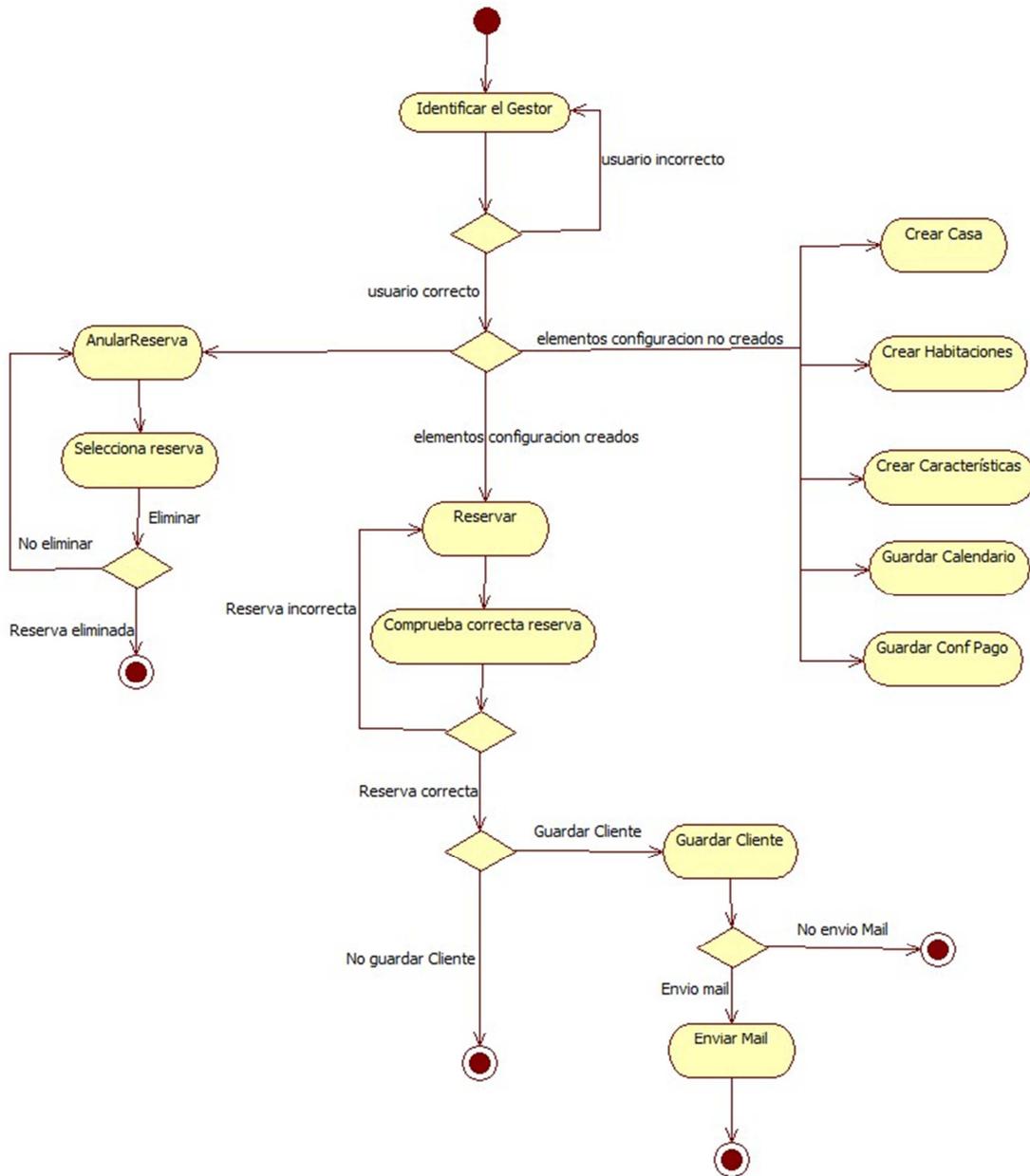


Figura 28. Diagrama de actividad de la aplicación

7. Diseño del sistema

Después de haber descrito los requisitos del proyecto y realizado el análisis, vamos a pasar a detallar el diseño del sistema. Procederemos a describir la organización del sistema a nivel lógico.

Inicialmente se explicará la arquitectura del sistema a desarrollar. Después pasaremos a diseñar el sistema propiamente dicho.

7.1. Arquitectura del Sistema

La arquitectura es la organización fundamental de un sistema descrita en sus componentes, la relación entre esos componentes y el ambiente, y en los principios que guían su diseño y evolución.

Vamos a centrarnos en el diseño de la arquitectura lógica específica de la aplicación. En lo referente a la arquitectura lógica de la aplicación individual, se tiene una separación entre la lógica del juego y la parte de la presentación, propio del Modelo Vista Controlador típico de Android. Esto implica contar con diferentes capas, que son: capa de presentación, capa de lógica de negocio y capa de acceso a datos.

Inicialmente, la capa de acceso a datos para nuestro sistema es relativamente sencilla, todos los componentes de la aplicación tendrán registro en ella, más adelante desgranaremos esta capa de datos.

En cuanto a la capa de lógica de negocio, en ella principalmente se define la lógica de la gestión de las reservas de la casa rural.

La capa de presentación representa dónde se muestran los Componentes de interfaz de usuario (pantallas, escenas, etc.) y Componentes de Procesos de interfaz de usuario (principalmente encargados de controlar el flujo de operaciones con el usuario).

Además de todo esto, hay una serie de componentes de servicios comunes, cuyo elemento más importante es el referido a las comunicaciones.

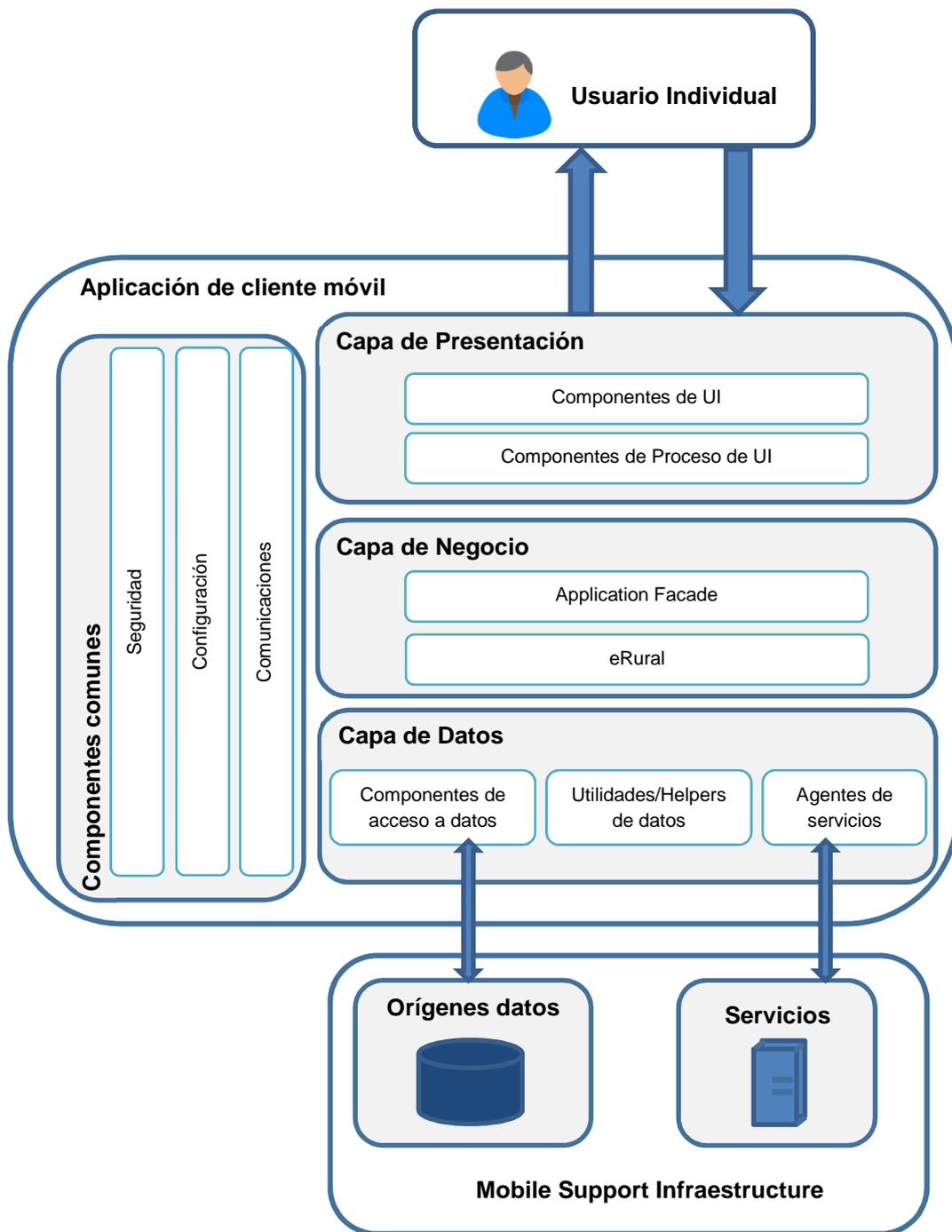


Figura 29. Arquitectura lógica del sistema.

7.2. Diagrama de despliegue

En este diagrama de despliegue se muestra la arquitectura del sistema desde el punto de vista de la distribución de los artefactos del software en los destinos de despliegue.

Los artefactos representan elementos concretos en el mundo físico que son el resultado de un proceso de desarrollo. Ejemplos de artefactos son archivos ejecutables, bibliotecas, archivos, esquemas de bases de datos, archivos de configuración, etc.

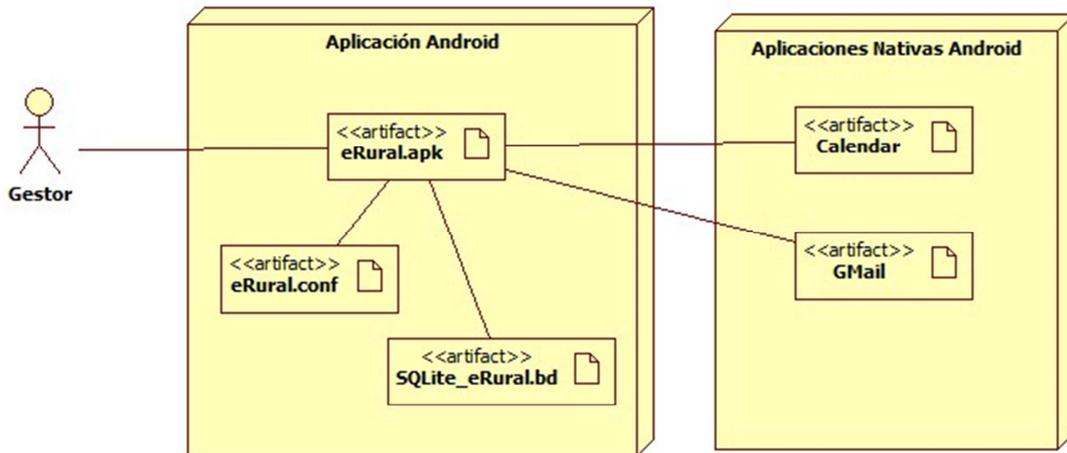


Figura 30. Diagrama de despliegue del sistema.

Podemos observar los componentes que contiene nuestro proyecto, por un lado nuestra aplicación en comunicación directa con nuestra base de datos SQLite y el archivo de configuración de la aplicación.

7.3. Diseño de subsistemas y componentes

Podemos definir este punto como la división de requisitos software en conjuntos coherentes de elementos reutilizables.

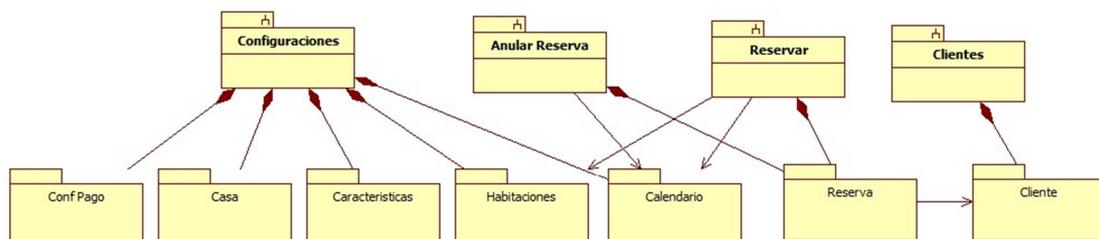


Figura 31. Subsistemas / Componentes.

Como se ha visto antes, el sistema se divide varios subsistemas de los que derivan el resto de componentes.

7.4. Modelo de datos

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.

Existen programas denominados sistemas gestores de bases de datos, abreviado SGBD, que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos SGBD, así como su utilización y administración, se estudian dentro del ámbito de la informática. En nuestro caso hemos utilizado SQLite, ya que es el soporte de bases de datos del que viene provisto el sistema operativo Android.

Nuestro sistema está compuesto por una base de datos implementada con el sistema gestor de bases de datos SQLite, utilizada para almacenar los datos de los componentes de la casa rural como son la casa, las habitaciones, las características de las mismas, el gestor, y en lo correspondiente al subsistema de reservas como el cliente, el histórico de reservas y forma de pago.

Debemos tener en cuenta que este sistema gestor de base de datos se está volviendo muy popular en la actualidad por ofrecer características tan interesantes como su pequeño tamaño, no necesitar servidor, precisar poca configuración, ser transaccional y por supuesto ser de código libre. Utiliza tipado dinámico (**Manifest Typing**). SQLite permite almacenar cualquier valor de cualquier tipo en cualquier registro de una tabla de la base de datos, independientemente del tipo declarado al crear la tabla. También fomenta que las claves primarias sean enteros incrementales, por lo que nuestras tablas siguen ese estándar, aunque no sea obligatorio. Otra característica interesante es que una base de datos es privada a la aplicación que la crea.

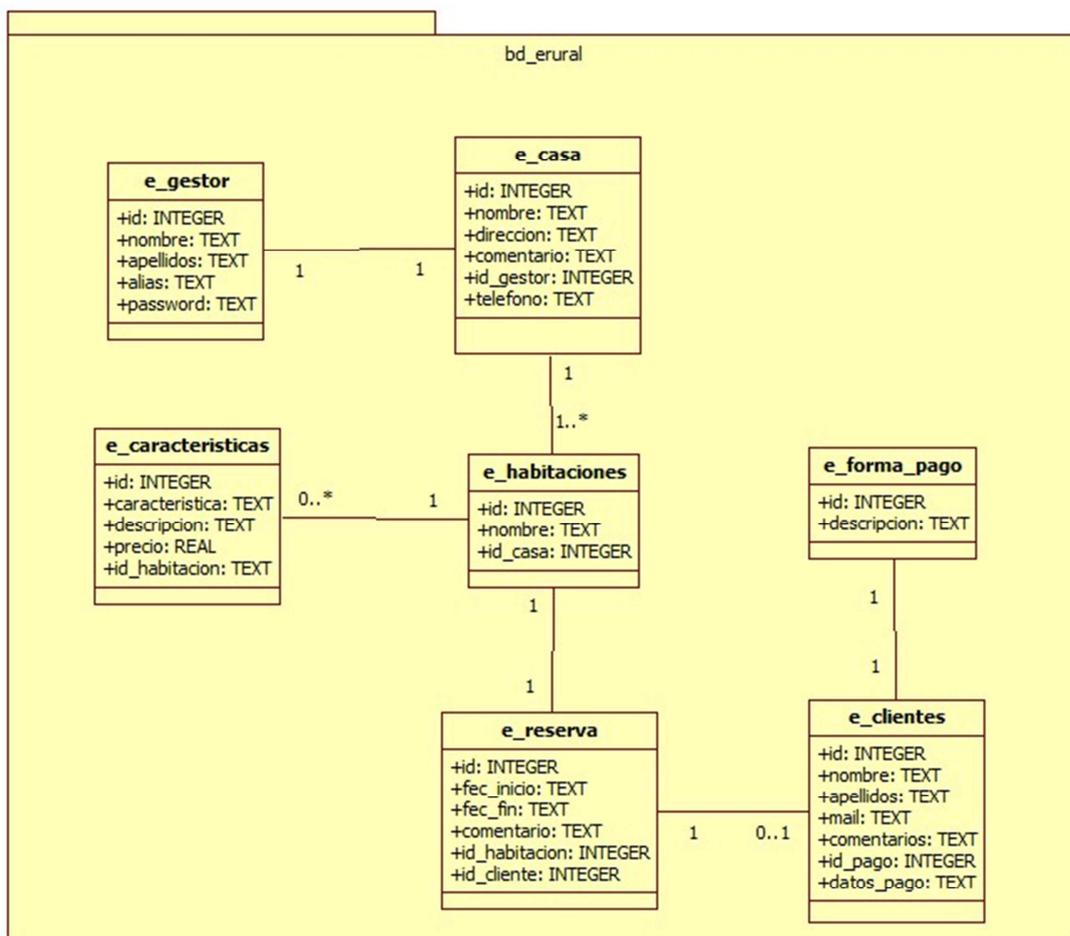


Figura 32. Modelo de datos para la aplicación.

Ahora procederemos a explicar la utilización de cada una de las tablas del modelo de datos de la figura anterior:

- **e_casa**: Contiene la información relativa a la casa como tal. Como vemos sus parámetros son: id, nombre, dirección, comentario, teléfono, id_gestor, referenciando al gestor de la aplicación como clave ajena.
- **e_gestor**: Almacena la información relativa al usuario gestor de la aplicación. Como vemos sus campos son: id, nombre, apellidos, alias y password, reseñar que éste último se almacena encriptada para mayor seguridad.
- **e_habitaciones**: Contiene la información relativa a las habitaciones de la casa. Sus campos son id, nombre y id_casa referenciando a la casa rural a la que pertenecen.
- **e_caracteristicas**: Almacena la información de las características o extras que puede tener una habitación. Se compone de id, característica, descripción, precio e id_habitacion para referenciar a la habitación a la que pertenece la característica en cuestión.

- e_reserva: Se comporta a modo histórico, almacena las reservas, almacenándose en los campos: id, fec_inicio, fec_fin, comentario, id_habitacion e id_cliente, para saber a qué habitación y a qué cliente está referenciando esta reserva.
- e_clientes: Contiene la información relativa a los clientes registrados en sistema. Se compone de id, nombre, apellidos, mail, comentarios, datos_pago, e id_pago, que referencia el modo de pago elegido por el cliente.
- e_forma_pago: Tabla auxiliar que almacena los tipos de pago existentes. Se compone de id, y descripción de la forma de pago.

7.5. Diseño de clases

El diseño detallado es una descripción del sistema muy cercana a la codificación (por ejemplo, describir no sólo las clases en abstracto, sino también sus atributos y los métodos con sus tipos).

A continuación, se explica el funcionamiento general del proyecto, resaltando las clases deducidas del mismo, para después mostrar todas ellas en el diagrama de clases general y también en diagramas de clases separados por subsistemas.

Comencemos por la parte de registro, y primeras clases para las primeras acciones del usuario, vamos a poner la imagen de las clases que nos dan acceso al registro y las acciones principales de nuestra aplicación.

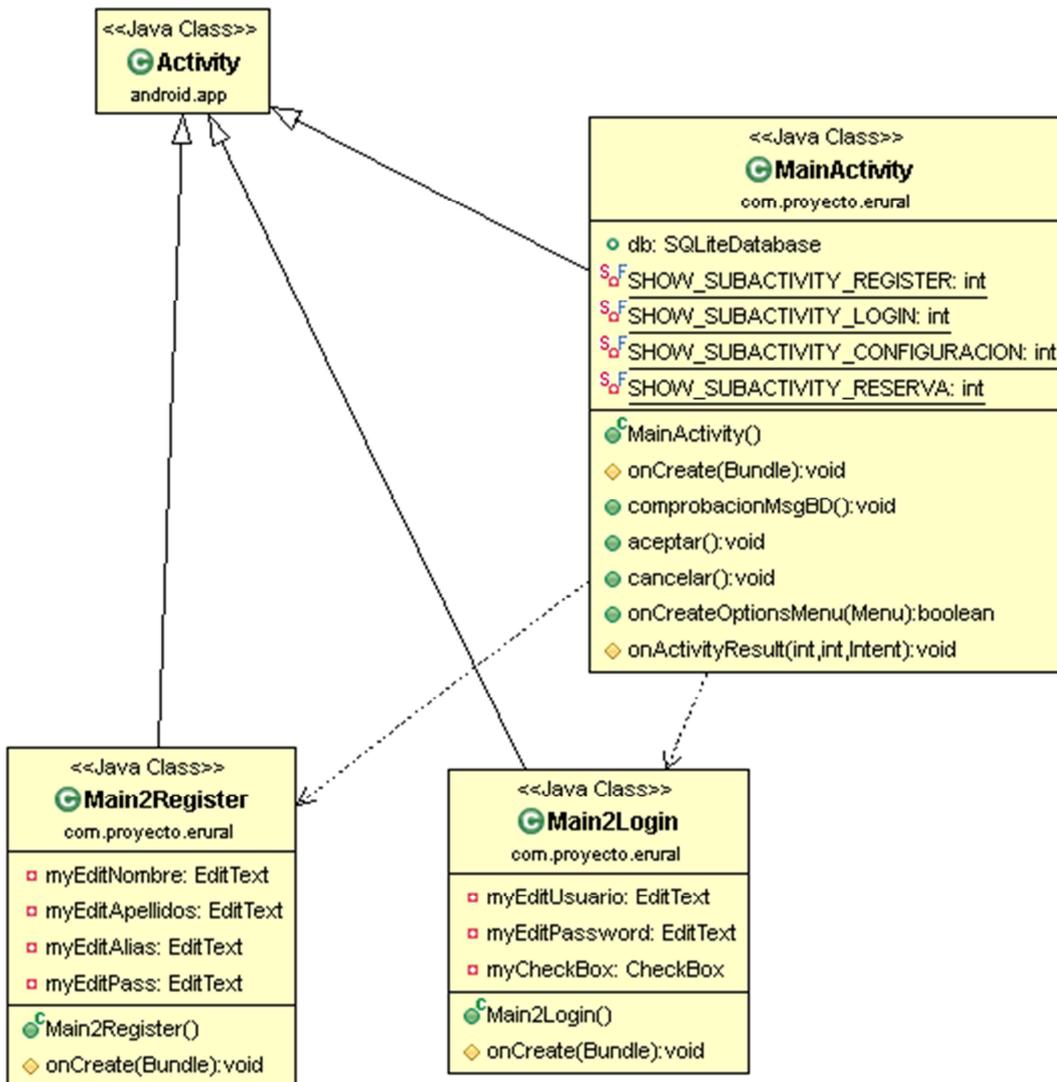


Figura 33. Diagrama de clases para el registro en la aplicación.

Vemos como en esta figura la clase genérica de Activity no extendemos todos sus atributos y métodos porque nos quedaríamos sin espacio para poder describirla en profundidad.

Otra jerarquía de clases que habría que describir antes de meternos con la configuración y con la reserva de la aplicación sería con las clases utilizadas para comunicarse con la base de datos y registrar todos los componentes de nuestra aplicación. La clase que se ha utilizado es la clase eRuralGestorBD, que es la encargada de gestionar todas las comunicaciones con la base de datos. Toda la gestión de las distintas acciones que pueden ser necesarias con la base de datos están centralizadas en esta clase.

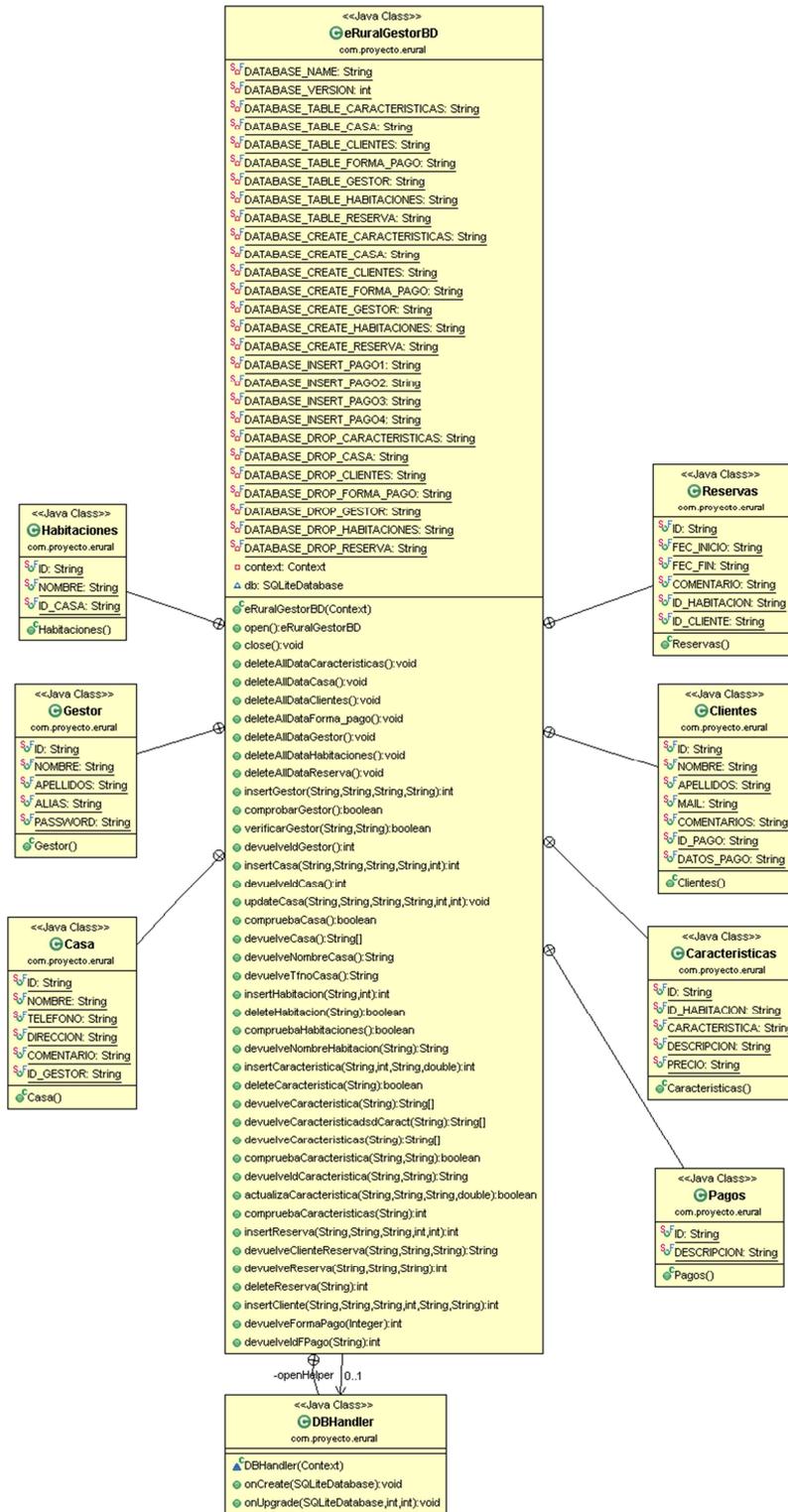


Figura 34. Diagrama de clases de la base de datos.

Vamos a visualizar las clases de este diagrama por separado para que queden más claras, ya que son piedra angular de los componentes de nuestra aplicación.

Clase eRuralGestorBD: se encarga de centralizar todas las acciones requeridas por la comunicación de la aplicación y la base de datos. Se puede observar con claridad que

esta clase utiliza un **helper** del tipo **DBHandler**, ésta nos permitirá crear y eliminar tablas, registros, hacer actualizaciones, etc.

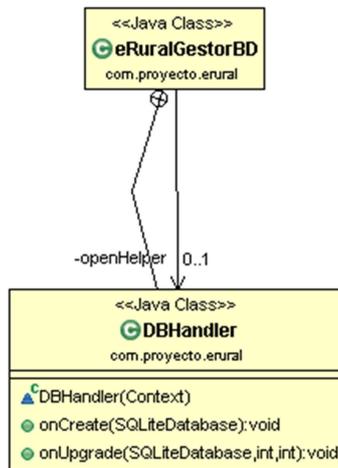


Figura 35. Diagrama de clases de eRuralGestorBD.

Clase Gestor: es la clase que representa el usuario gestor de la aplicación.



Figura 36. Diagrama de clases de Gestor.

Clase Casa: es la clase que representa a la entidad casa rural dentro de la aplicación.

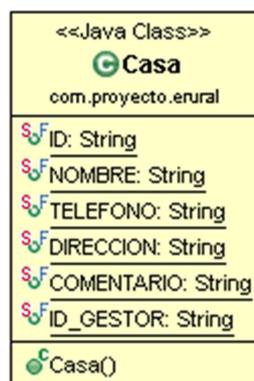


Figura 37. Diagrama de clases de Casa.

Clase Habitaciones: es la clase que representa a la entidad habitación dentro de la aplicación.



Figura 38. Diagrama de clases de Habitaciones.

Clase Características: es la clase que representa a la entidad característica de una habitación dentro de la aplicación.



Figura 39. Diagrama de clases de Características.

Clase Reservas: es la clase que representa a la entidad reserva como tal dentro de la aplicación, la clase reserva se comporta como histórico, a modo registro de la reserva, almacenando también habitación y cliente.



Figura 40. Diagrama de clases de Reservas.

Clase Clientes: es la clase que representa a la entidad cliente dentro de la aplicación.

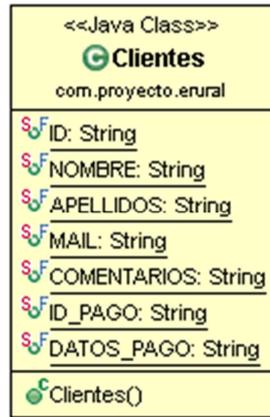


Figura 41. Diagrama de clases de Clientes.

Clase Pagos: es la clase que representa a la entidad forma de pago dentro de la aplicación.



Figura 42. Diagrama de clases de Pagos.

Sigamos por la parte de configuraciones, ya se ha comentado en varias ocasiones que la configuración se realiza en un listado en forma de menú, es por ello por lo que se tienen una clase para cada uno de dichos elementos. Hay que decir que se trata de clases públicas y genéricas. Son públicas porque definen un comportamiento común y queremos que sean accesibles desde cualquier clase. Son una extensión de la clase genérica Android Activity, que es la que se puede definir como conjunto de acciones (tocar la pantalla para apretar un botón, para escribir con el teclado, etc) que son una iteración directa con el usuario y que afectan a una parte de la aplicación.



Figura 43. Diagrama de clases de la principal a la configuración.

Vamos a ver las clases que derivan de la de configuración, de manera pormenorizada. Esta que visualizamos a continuación es la de la configuración de la casa, en la que el gestor podrá registrar la entidad Casa, necesario para el resto de acciones que tiene la aplicación, entre otras reservar.

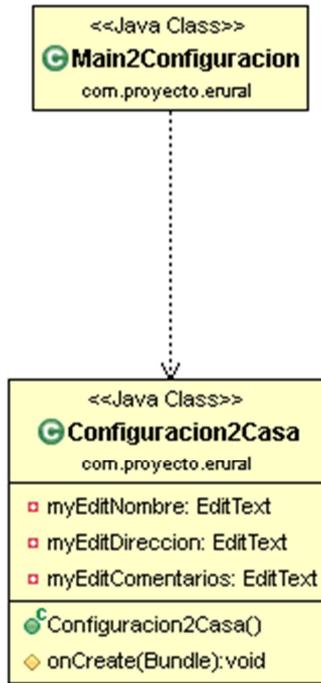


Figura 44. Diagrama de clases de la configuración de la casa.

En esta siguiente vemos el diseño de la configuración de las habitaciones dentro de la aplicación.

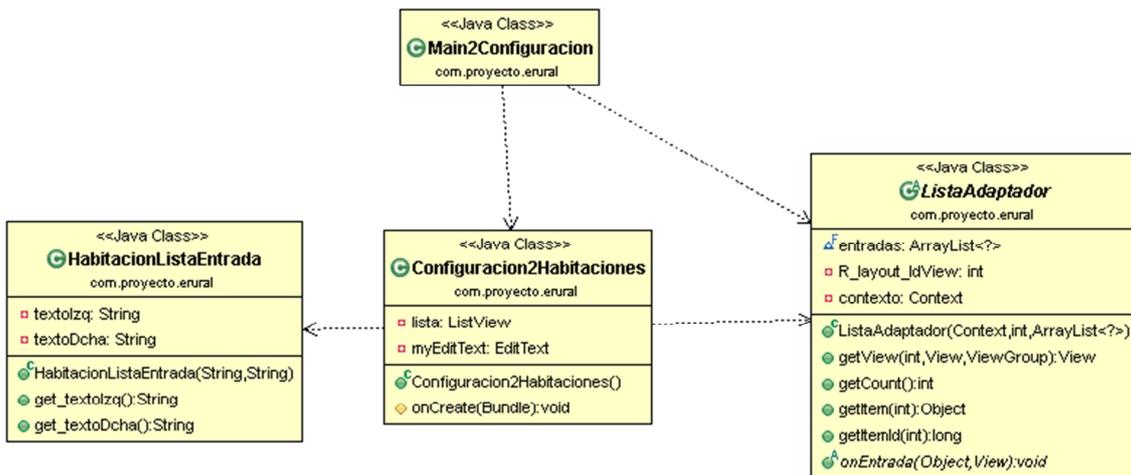


Figura 45. Diagrama de clases de la configuración de las habitaciones.

En la siguiente figura vamos a ver la parte de configuración de las características de las habitaciones, vamos a ver que también reutilizamos la clase ListaAdaptador, una clase base utilizada para la visualización de todas las listas de los componentes ListView propios de las interfaces de Android. La configuración de las características comunes es similar, por lo que nos hacemos una idea.

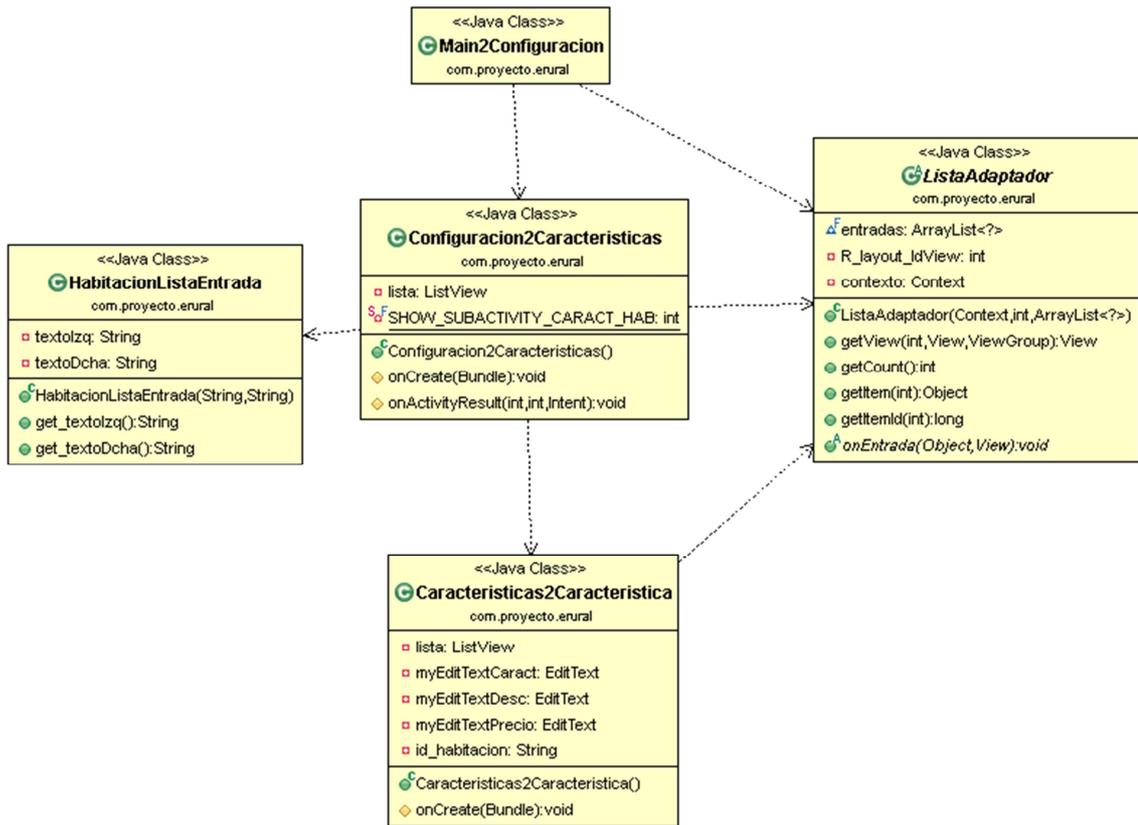


Figura 46. Diagrama de clases de la configuración de las características de las habitaciones.

Vamos a seguir con la configuración de la cuenta a la que irá asociada el calendario y el envío de correo electrónico. En esta imagen vemos que aquí tenemos otra clase de lista no genérica como la que vimos en la anterior imagen, esta es propia del listado checkeable de las distintas cuentas de correo que puede tener el dispositivo móvil, y en concreto el Calendario asociadas a él. Ya que todos los eventos que se informan en Android van asociados a la cuenta del Calendario que el usuario quiera.

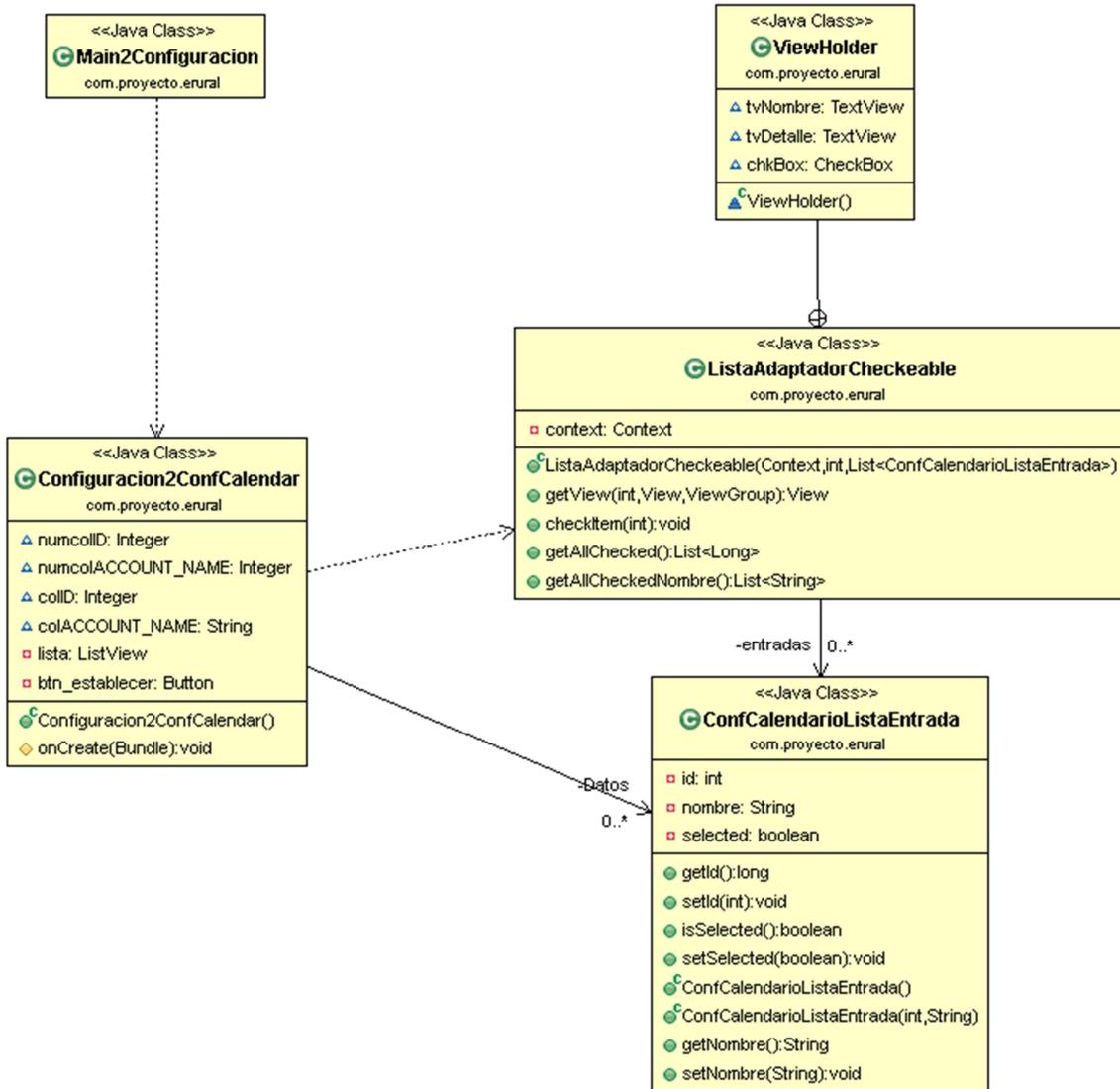


Figura 47. Diagrama de clases de la configuración de la cuenta de correo.

Por último, vemos el diagrama de clases de la configuración del pago de las reservas.

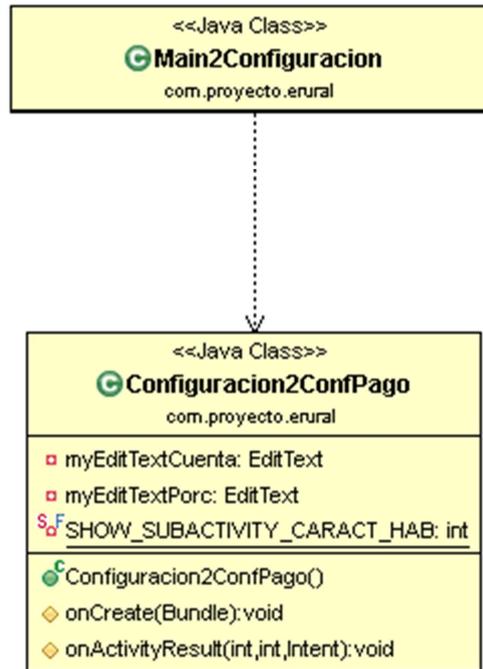


Figura 48. Diagrama de clases de la configuración del pago.

Por otra parte, se tiene el subsistema de reservas, en el cual representamos con la imagen siguiente, el cual es una interfaz en la que podemos seleccionar la habitación a reservar, y previamente consultar la disponibilidad en forma de calendario, para posteriormente pasar a la pantalla de realizar la reserva.



Figura 49. Diagrama de clases de la principal a la reserva.

Una vez elegida la habitación a reservar vamos a ver la imagen completa de todas las clases que componen este subsistema de reservar del que hemos hablado con anterioridad. Vamos a detenernos un poco en esta imagen, y comentar varias cosas de ésta. La clase **Reserva2Reserva** es la que centralizará el proceso propiamente dicho de reservar, es la que tendrá el hilo principal de la reserva en todo momento. Como se puede apreciar con la imagen, ésta clase de la que estamos hablando dispondrá del formulario para informar los parámetros de la reserva, siendo esta clase la que compruebe si disponibilidad dentro del calendario, y derive el flujo a la clase **Reserva2Cliente** para que pueda registrar el cliente, y de vuelta a nuestra clase

Reserva2Reserva podemos volver a comunicarnos con el cliente de correo propio de Android y así poder enviar el mail que hemos creado al cliente recién creado también.

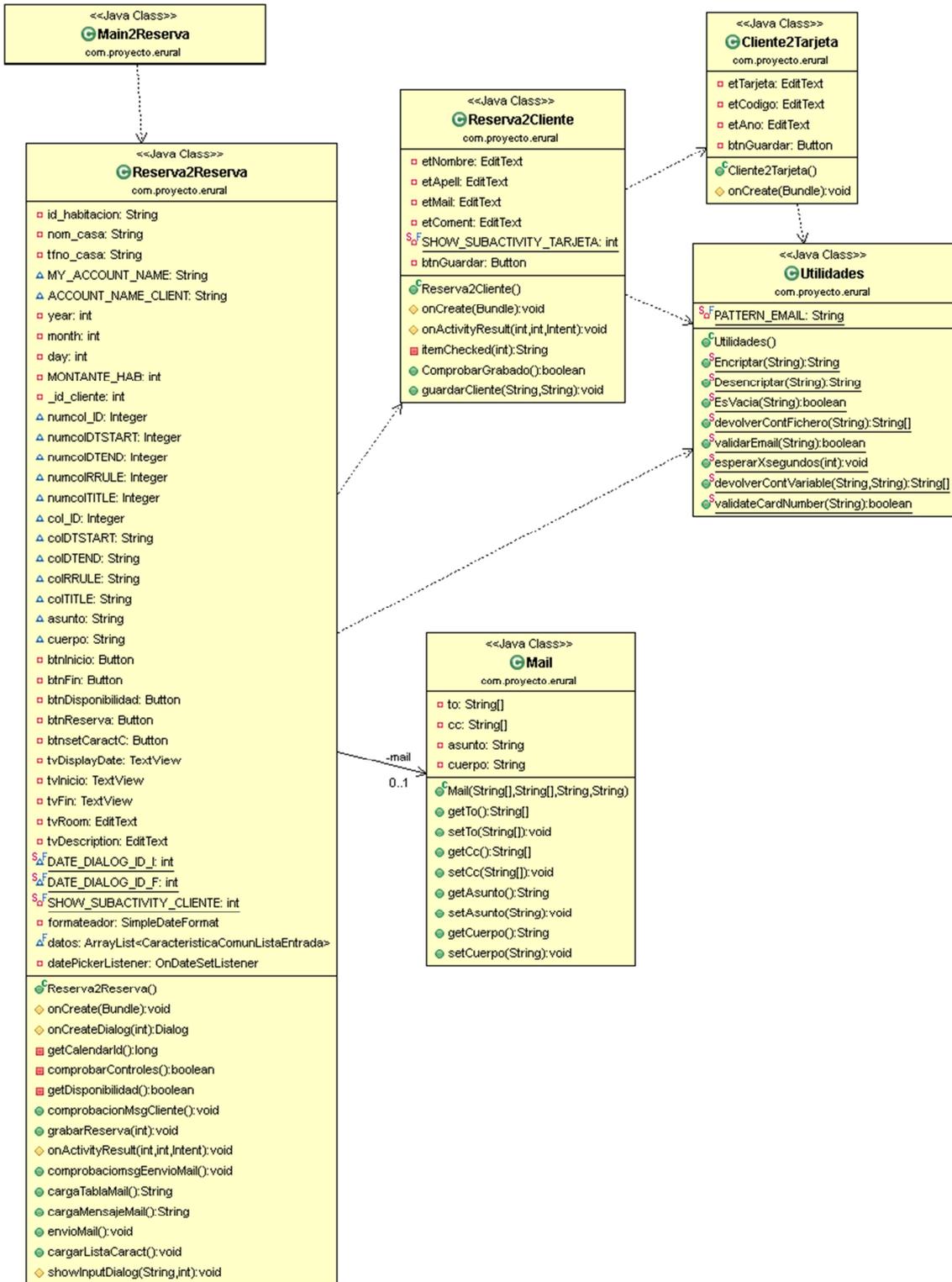


Figura 50. Diagrama de clases del subsistema reservar.

Una vez realizadas las reservas podemos necesitar realizar una anulación, vamos a

ver la imagen completa de todas las clases que componen este subsistema de anular una reserva:

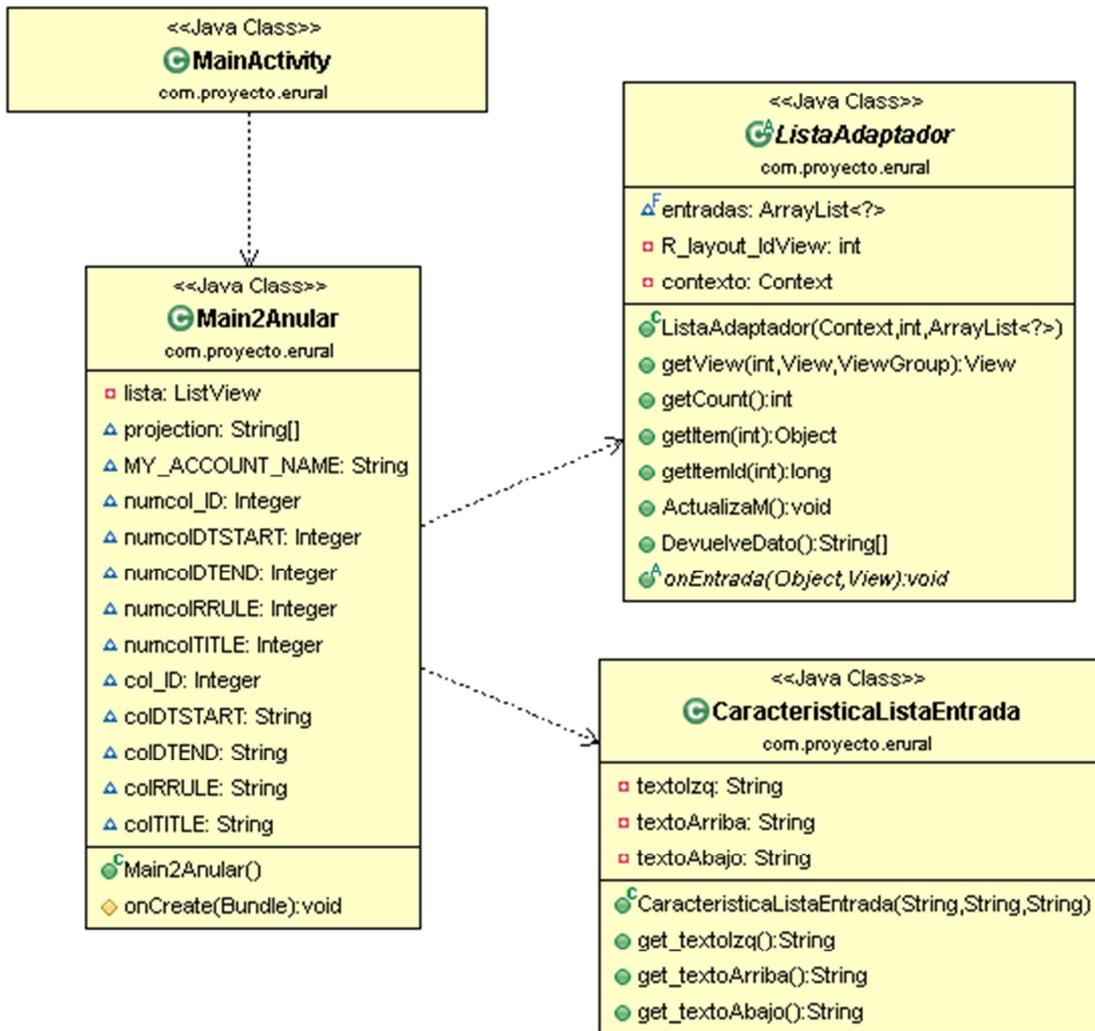


Figura 51. Diagrama de clases del subsistema anular reserva.

Para finalizar con el diseño de clases de nuestra aplicación vamos a mostrar el subsistema de visualización de clientes:

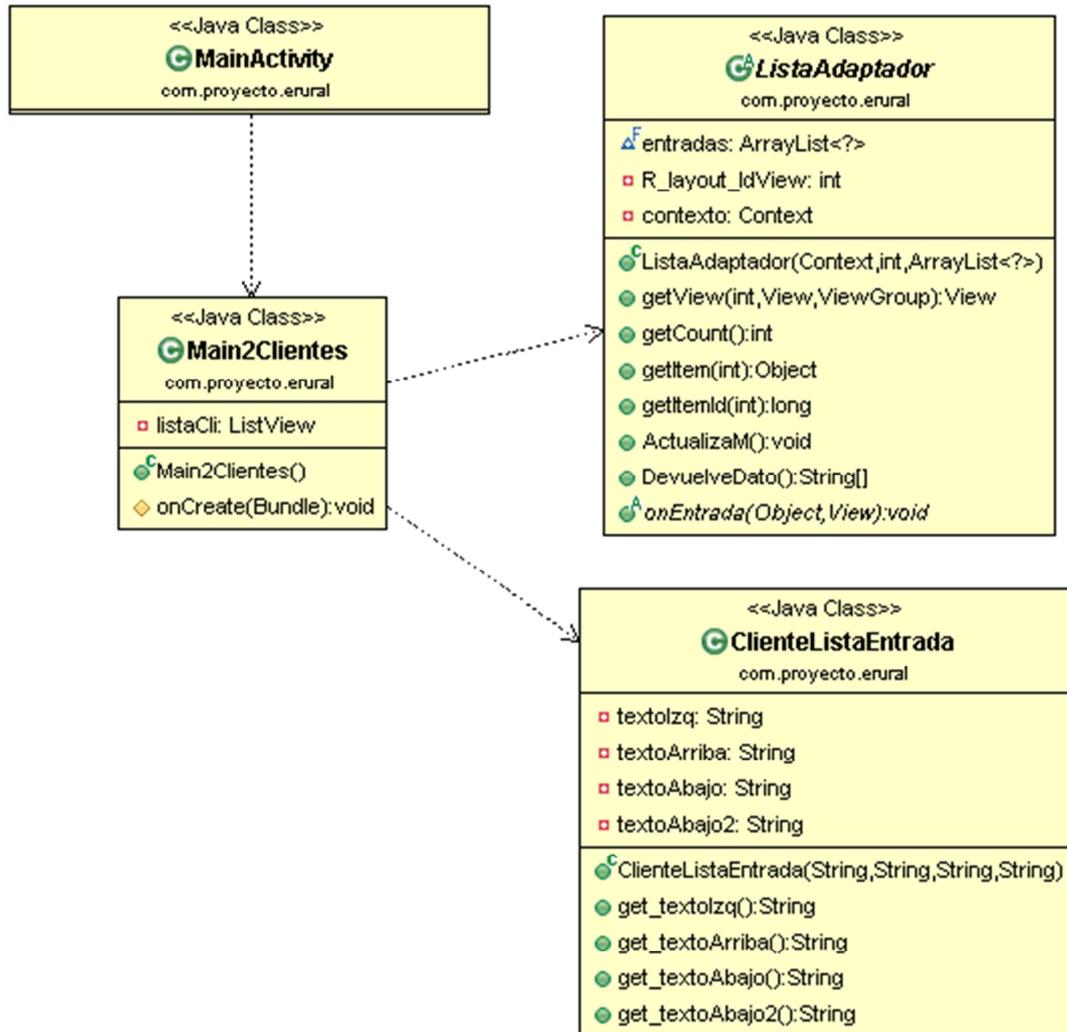


Figura 52. Diagrama de clases del subsistema visualización clientes.

Para finalizar querríamos resaltar una característica fundamental en Android y que es piedra angular del funcionamiento de nuestra aplicación: **Content Providers**. Vamos a definir un poco este término, como vimos en la figura 2 la arquitectura de Android, tenemos una capa denominada **Application Framework**, la cual está compuesta por distintas aplicaciones y librerías para el manejo de la interfaz, notificaciones, telefonía, localización, acceso a datos (como de calendarios), etc. Este **framework** nos proporciona las clases para crear aplicaciones en Android.

Pues en concreto, los **Content Providers** son almacenes de datos compartidos. Se utilizan para intercambiar información entre procesos, se comunican mediante **Intents**, que son **framework** de paso de mensajes entre aplicaciones o componentes de una misma aplicación y se utilizan para pedir acciones a otras aplicaciones o componentes.

Por lo tanto, en nuestra aplicación, la comunicación con el Calendario y Gmail, se realiza mediante **Intents** a los **Content Providers** propios de las aplicaciones nativas

Android como son el **Calendario** y **Gmail**.

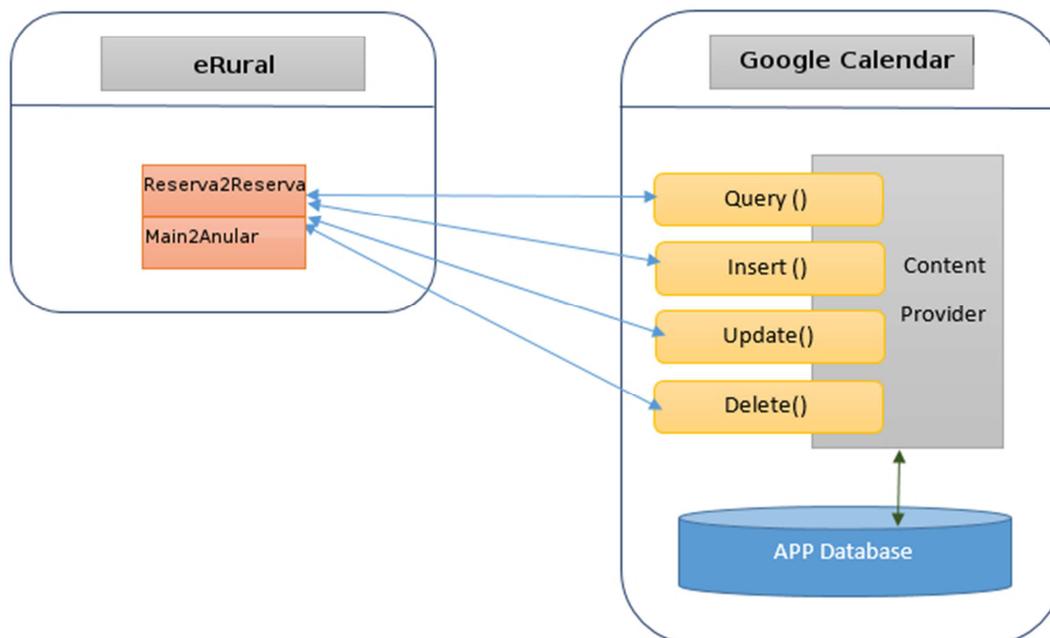


Figura 53. Content Provider Google Calendar.

7.6. Diseño de Interfaces y perfiles de usuario

El diseño se realizará según estas dos características: Accesibilidad y Usabilidad. La primera característica es importante, ya que se trata de que cualquier tipo de usuario pueda acceder a la aplicación, sin que éste requiera un alto nivel informático. La segunda característica es la facilidad de uso de la aplicación. Se diseñará una interfaz lo más cómoda e intuitiva (amigable) para que el usuario no tenga dudas y puede acceder a cualquier función de la aplicación rápidamente. Lo consideramos tan importante que está registrado como uno de nuestros requisitos no funcionales.

Respecto a los perfiles de usuario, solo hay uno, el usuario normal que se comporta como gestor de la Casa Rural, será el encargado de la gestión de los componentes, de la reserva de las habitaciones, y de la anulación de reservas.

A continuación se muestra el diseño de pantalla (sólo de las pantallas más significativas) de los diferentes elementos del sistema a partir de unas imágenes de referencia:

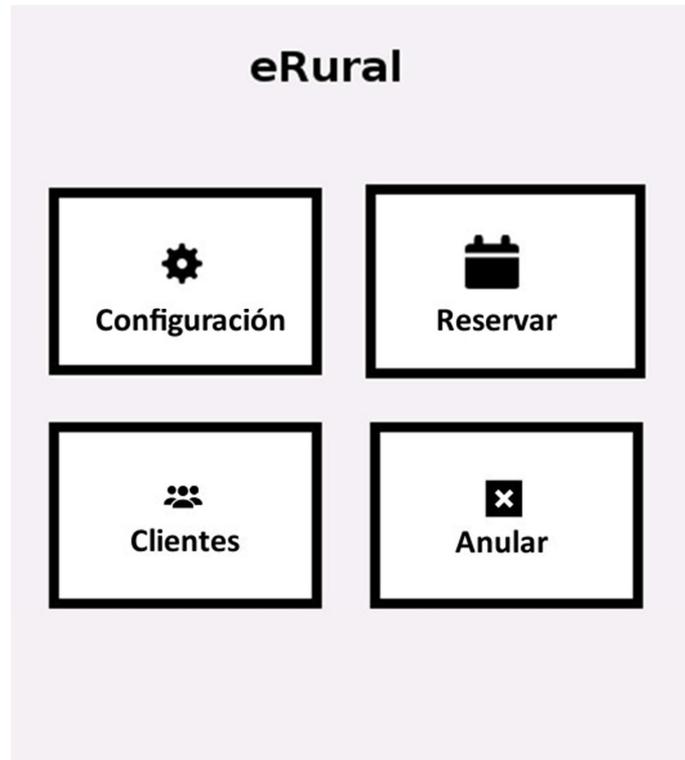


Figura 54. Diseño pantalla principal.

La aplicación nos permite ejecutar los subsistemas diferentes, que son la configuración de los componentes, la reserva de habitaciones, la visualización de clientes y la posibilidad de anular reservas. Por lo tanto existen cuatro botones diferenciados para cada uno.

Para el primer subsistema de configuración, necesitábamos informar cuatro áreas fundamentalmente: la casa, las habitaciones, las características de éstas, la configuración del pago y la configuración de la cuenta del calendario.

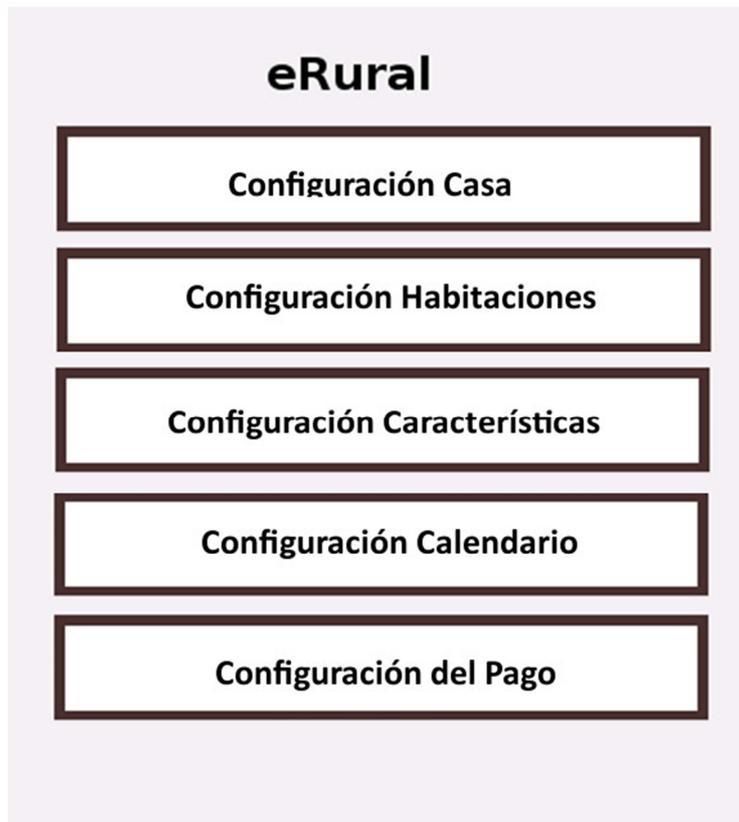


Figura 55. Diseño menú configuración.

Luego cualquier pantalla para registrar cualquiera de estos será el típico formulario en el que podremos informar los campos necesarios para registrarlo en base de datos, así por ejemplo para la casa diseñaremos una interfaz en la que podamos informar de todos los campos necesarios.

Este diagrama muestra un formulario de registro para una casa en el sistema eRural. El formulario está encabezado por el título "eRural" en un recuadro superior. Debajo del título, se encuentran cuatro campos de entrada de texto, cada uno con un recuadro rectangular y un borde oscuro: "Nombre", "Teléfono", "Dirección" y "Comentarios". Debajo de los campos de entrada, hay un botón rectangular con el texto "Guardar".

Figura 56. Diseño pantalla de Casa.

Centrándonos en el acto de reservar, diseñamos algo sencillo como podía ser una típica interfaz de reserva con un botón que levantara una modal para informar de las fechas, y varias entradas para poner el título y descripción de la reserva.

The wireframe shows a reservation form for 'eRural'. At the top is the title 'eRural'. Below it are two rows of date selection: 'Inicio' (Start) and 'Fin reserva' (End reservation), each with a text input field and a calendar icon. Below these are two larger text input fields labeled 'Título de la reserva' (Reservation title) and 'Comentarios de la reserva' (Reservation comments). At the bottom is a large button labeled 'Reservar' (Reserve).

Figura 57. Diseño pantalla de reserva.

Tanto para la visualización de clientes como para la anulación de reserva pensábamos en algo como una lista de datos.



Figura 58. Diseño pantalla de anulación de reserva.

8. Implementación del sistema

8.1. Introducción

En este capítulo se ha intentado reflejar cómo se ha dado el paso desde el diseño a la implementación, así como los problemas más representativos que se han presentado.

Ante todo, comentar que la herramienta que se ha utilizado es **Eclipse** integrado con **Android Developer Tools**. **Eclipse** es un entorno de desarrollo integrado de sobra conocido en la comunidad de desarrolladores Java. Y **Android Developer Tools** es un **plugin** para Eclipse que ofrece un entorno de desarrollo de nivel profesional para la creación de aplicaciones de Android.

La implementación de la aplicación se ha llevado a cabo mediante el lenguaje de programación orientado a objetos Java aplicado a Android, que es el lenguaje propio con el que se desarrollan la gran mayoría de aplicaciones para este sistema operativo,

conjugándolo con archivos XML que conforman la parte visual de la aplicación.

Se ha hablado a lo largo de todo el proyecto (y especialmente en el proceso de diseño) de dos partes diferenciadas (la referida a la configuración y a la reserva) que conforman el conjunto del proyecto.

8.2. Implementación con el calendario de Google

Como se va a trabajar con el servicio de Google Calendar, es recomendable conocer el funcionamiento del protocolo de Google. Para ello se aconseja la lectura de la documentación del API de GoogleCalendar dónde se explica de manera muy completa dicho protocolo.

El hecho de que estemos utilizando el entorno IDE Eclipse con el plugin ADT que proporciona Android nos proporciona una serie de herramientas que facilitan el desarrollo.

Como ya se ha explicado en otras secciones, desarrollar para Android implica utilizar el lenguaje de programación Java, lo cual facilita la tarea al ser un lenguaje ya conocido y al disponer de numerosa documentación y librerías de fácil acceso.

Al querer interactuar con el servicio del calendario de Google, es necesaria la importación de varias librerías de componentes del calendario que proporciona Google para Java.

El primer paso de la aplicación es obtener los credenciales del usuario para acceder al servicio Google relacionado con su cuenta. En nuestra aplicación vamos a utilizar: **Método de autenticación mediante Android Account Manager.**

En la versión del SDK de Android de 2.0 aparece por primera vez el administrador de cuentas AccountManager de Android. Su función es administrar todas las autenticaciones y autorizaciones en un sistema central de cuentas.

Este método de autenticación consiste en utilizar la autenticación que provee este administrador seleccionado la cuenta desde la que se desea acceder al servidor y utilizar el token que devuelve para poder obtener los servicios. El requisito indispensable es que debe existir dicha cuenta en el panel de control de Cuentas y Sincronizaciones de la configuración de Android.

Una de las principales funcionalidades que hace que el calendario de Google sea muy utilizado por los usuarios es poder añadir elementos en fechas específicas de tal manera que permita recordar al usuario cualquier tipo de eventos que éste pueda tener.

Añadir un evento a un calendario desde una aplicación externa a Google Calendar es algo más complejo ya que conlleva lo primero de todo, conocer y obtener el calendario

en el que se va crear el evento, seguidamente se necesitará crear un mensaje que represente al evento y por último utilizar el token de autenticación obtenido en el proceso de autenticación de la aplicación para enviar la petición de creación.

Crear un evento requiere construir un mensaje que contenga los atributos necesarios para poder añadir el evento en el calendario.

Debido a estas importantes funcionalidades de Google Calendar la hacen piedra angular de nuestra aplicación, ya que todas las reservas estarán registradas como eventos en el calendario, con su correspondiente aviso con antelación, un recordatorio que saltará en cualquier dispositivo móvil y será fácil para nuestro usuario Gestor de visualizar gracias a la gran incidencia en nuestros días de la aplicación Calendario de Google.

8.3. Seguridad

En nuestra aplicación uno de nuestros requisitos no funcionales es la seguridad, y para ello tenemos un proceso de identificación de usuario, para que la aplicación no caiga en manos de usuarios no deseados. Por eso tenemos la necesidad de tener la dupla usuario y contraseña registrada. Por ello, y para mayor seguridad al tener almacenada nuestra contraseña en el sistema, vamos a hacer lo que se denomina **encriptar**.

La encriptación de datos corresponde a una tecnología que permite la transmisión segura de información, al codificar los datos transmitidos usando una fórmula matemática que "desmenuza" los datos. Asegurar que la Información viaje segura, manteniendo su autenticidad, integridad, confidencialidad y el no repudio de la misma entre otros aspectos.

Cabe destacar entonces, que hemos implementado una clase denominada Utilidades, la cual se encarga del encriptado y desencriptado de la contraseña, en una y otra dirección para que nuestra aplicación cumpla con los estándares de seguridad.

También cabe resaltar una vez más, que una de las peculiaridades del sistema gestor de base de datos SQLite es que las bases de datos solo son accesibles por las propias aplicaciones que las crean.

8.4. Problemas surgidos en la fase de implementación

Los principales problemas encontrados a la hora de implementar las distintas partes del proyecto han sido debidos, en su mayoría, a problemas de inexperiencia; y es que antes de sumergirme en este proyecto no tenía ningún tipo de conocimiento previo en desarrollo de aplicaciones de la plataforma Android y muy poca experiencia en el lenguaje Java, por lo que fue necesario un esfuerzo previo de formación. Pese a este

inicial periodo de formación, fue insuficiente, como ha quedado patente en los resultados de la planificación. También hay que decir que esa formación también ha estado presente a lo largo de todo el proceso de desarrollo, de hecho la mejor formación es la práctica.

En cuanto a problemas concretos, podríamos decir que una de mis mayores dificultades vino inicialmente con la clase gestora de la base de datos, pues partía de una inexperiencia con Java y con SQLite, el gestor de base de datos propio de Android. Una vez entendida la filosofía de visibilidades y comportamientos de esta clase gestora de base de datos me fue más sencillo insertar registros desde cualquier clase del proyecto, incluyendo subactividades.

Otra de las dificultades fueron los diálogos que se abren de fechas, ya que sus estilos me trajeron buenos quebraderos de cabeza, ya que el salirse del estándar de los estilos y temas propios de las aplicaciones Android es un tanto confuso, no hay mucha documentación al respecto, o por lo menos no hay documentación concreta a cambiar estilos solo de clases **datepicker dialog**.

En contra de lo que pudiera parecer, la comunicación con el calendario de google ha sido relativamente sencilla, ya que se disfruta de mucha y buena documentación en la red, no solo de la página oficial de desarrolladores de Android, sino multitudes de blogs y comunidades de desarrolladores libres.

Por fortuna, todos estos problemas han podido ser solventados, aunque han tenido su incidencia evidentemente en la planificación del desarrollo de la aplicación.

9. Pruebas del sistema

9.1. Introducción

La fase de pruebas es una etapa más del ciclo de vida de un programa. Sin embargo, en esta etapa se ha invertido una gran cantidad de tiempo y esfuerzo para poder subsanar los errores y pequeños detalles que surgieron en dicha etapa. Por esta razón se ha convertido en una fase clave en cuanto al funcionamiento de la aplicación se refiere.

Si se tiene en cuenta el hecho de que todos los programas tienen errores, la fase de pruebas es una etapa fundamental para localizarlos.

Que el programa funcione bien fue el objetivo de las anteriores fases. Sólo cuando se ha cumplido este primer objetivo se puede pasar al diseño de los casos de prueba. No obstante, el objetivo último de todo proceso de desarrollo de sistemas informáticos es hacer que su funcionamiento sea el correcto. Sin embargo cada fase tiene su propio

objetivo, y el de la fase de pruebas no es otro que el de encontrar el mayor número de fallos posibles, porque el objetivo final es la validación o comprobación de la aplicación.

9.2. Pruebas de Integración.

Las pruebas de integración se centraron en probar la coherencia semántica, tanto estática como dinámica, entre los diferentes módulos. La coherencia estática consistió en probar si se llamaba correctamente a los procedimientos que proporcionaba cada uno de los módulos. En cuanto a la coherencia dinámica, las pruebas realizadas fueron encaminadas a comprobar si un módulo recibía de otro módulo lo que se esperaba en un primer momento. Este tipo de pruebas se fueron realizando por etapas, englobando en cada etapa sucesiva más y más módulos hasta lograr así, el poder completar el funcionamiento de todas las unidades que forman el programa.

Hay que señalar que en este punto se realizó un diseño de pruebas ascendente. De esta manera se fueron probando en un primer momento los módulos base para terminar probando los módulos más generales.

9.3. Pruebas de cumplimiento de requisitos

También conocidas como las pruebas de validación, son las que se consiguen cuando el software funciona de acuerdo a los requisitos del sistema.

Por cada requisito se diseña y ejecuta al menos una prueba que garantice su cumplimiento, pudiendo ser necesarias más de una prueba para garantizar totalmente el correcto desempeño del requisito.

Requisito	UC-1 Identificar al usuario
Descripción prueba	Comprobación de que usuario y contraseña no vayan vacíos.
Resultado esperado	Mensaje de error.

Prueba	PCR-1A Usuario vacío
Requisito asociado	UC-1 Identificar al usuario.
Prerrequisito	Ninguno.
Procedimiento	1. Entrar en la aplicación 2. Pedirá identificarse y dejar el usuario vacío
Resultado	Mensaje de error informando que debe informar el usuario.
Comentario	Ninguno.

Tabla 54. PCR-1A Usuario vacío.

Prueba	PCR-1B Contraseña vacía
Requisito asociado	UC-1 Identificar al usuario.
Prerrequisito	Ninguno.
Procedimiento	<ol style="list-style-type: none"> 1. Entrar en la aplicación 2. Pedirá identificarse y dejar la contraseña vacía
Resultado	Mensaje de error informando que debe informar el usuario.
Comentario	Ninguno.

Tabla 55. PCR-1B Contraseña vacía.

Prueba	PCR-1C Credenciales correctas
Requisito asociado	UC-1 Identificar al usuario.
Prerrequisito	Ninguno.
Procedimiento	<ol style="list-style-type: none"> 1. Entrar en la aplicación 2. Pedirá identificarse y hacerlo de manera incorrecta, con usuario y/o contraseña incorrectos
Resultado	Mensaje de error de credenciales.
Comentario	Ninguno.

Tabla 56. PCR-1C Credenciales correctas.

Requisito	UC-2 Configuración de la casa
Descripción prueba	Comprobación de que al menos se informa el nombre de la casa, que es el único campo no nulo en base de datos.
Resultado esperado	Mensaje de error.

Prueba	PCR-2 Nombre de la casa vacío
Requisito asociado	UC-2 Configuración de la casa.
Prerrequisito	Ninguno.
Procedimiento	<ol style="list-style-type: none"> 1. Entrar en la aplicación 2. Identificarse 3. Entrar a la configuración de la casa 4. Informar vacío el campo del nombre de la casa
Resultado	Mensaje de error de que está vacío el campo nombre.
Comentario	Ninguno.

Tabla 57. PCR-2 Nombre de la casa vacío.

Requisito	UC-3 Configuración de las habitaciones
------------------	---

Descripción prueba	Comprobación de que se informa el nombre de la habitación, que es el único campo no nulo en base de datos.
Resultado esperado	Mensaje de error.

Prueba	PCR-3 Nombre de la habitación vacío
Requisito asociado	UC-3 Configuración de la habitación.
Prerrequisito	Ninguno.
Procedimiento	<ol style="list-style-type: none"> 1. Entrar en la aplicación 2. Identificarse 3. Entrar a la configuración de las habitaciones 4. Informar vacío el campo del nombre de la habitación
Resultado	Mensaje de error de que está vacío el campo nombre.
Comentario	Ninguno.

Tabla 58. PCR-3 Nombre de la habitación vacío.

Requisito	UC-4 Configuración de las características
Descripción prueba	Comprobación de que se informa el nombre de la característica, que es el único campo no nulo en base de datos.
Resultado esperado	Mensaje de error.

Prueba	PCR-4 Nombre de la característica vacío
Requisito asociado	UC-4 Configuración de las características.
Prerrequisito	Ninguno.
Procedimiento	<ol style="list-style-type: none"> 1. Entrar en la aplicación 2. Identificarse 3. Entrar a la configuración de las características 4. Informar vacío el campo del nombre de la habitación
Resultado	Mensaje de error de que está vacío el campo nombre.
Comentario	Ninguno.

Tabla 59. PCR-4 Nombre de la característica vacío.

Requisito	UC-10 Gestión de los clientes
Descripción prueba	Comprobación de que se informa el nombre del cliente, que es el único campo no nulo en base de datos. Y si se quiere informar por correo electrónico también deberá cumplir con el patrón de una cuenta de dirección de correo electrónico.
Resultado esperado	Mensaje de error.

Prueba	PCR-5A Nombre del cliente vacío
Requisito asociado	UC-10 Gestión de los clientes.
Prerrequisito	Ninguno.
Procedimiento	<ol style="list-style-type: none"> 1. Entrar en la aplicación 2. Identificarse 3. Realizar una reserva 4. Seleccionar que quieres informar el cliente
Resultado	Mensaje de error de que está vacío el campo nombre.
Comentario	Ninguno.

Tabla 60. PCR-5A Nombre del cliente vacío.

Prueba	PCR-5B Campo e-mail incorrecto
Requisito asociado	UC-10 Gestión de los clientes.
Prerrequisito	Ninguno.
Procedimiento	<ol style="list-style-type: none"> 1. Entrar en la aplicación 2. Identificarse 3. Realizar una reserva 4. Seleccionar que quieres informar el cliente 5. Informar los campos de nombre de cliente y el e-mail del cliente.
Resultado	Mensaje de error de que el campo e-mail no cumple con el patrón de cuenta de correo electrónico.
Comentario	Si no se informa el campo de e-mail no pasa nada, no se enviará correo electrónico, pero si se informa debe cumplir con el estándar de una dirección de correo electrónico.

Tabla 61. PCR-5B Campo e-mail incorrecto.

10. Conclusiones y líneas futuras

10.1. Conclusiones

En todos estos años de carrera, hemos tenido numerosas asignaturas en las que nos han enseñado muchas cosas. Pero uno se da cuenta que con la experiencia de hacer un proyecto, de todos los conocimientos adquiridos y también de todos los que quedan por adquirir.

Lo más importante a la hora de enfrentarte con un proyecto, es hacer una buena extracción de requerimientos y a continuación una exhaustiva fase de diseño. Es preferible tardar más en estas fases iniciales, que son las cruciales para que la aplicación salga adelante de la manera más eficaz y en el menor tiempo posible.

Este proyecto ha sido finalizado cumpliendo con los objetivos iniciales y los requisitos especificados al comienzo del mismo, que eran crear una aplicación para un dispositivo móvil que permitiera la gestión de una casa rural. Aunque no hemos llegado a conseguir realizar todos los objetivos que nos habíamos marcado inicialmente.

Aunque el proyecto cumpla con las especificaciones expuestas en un principio, es un proyecto con grandes posibilidades de ampliaciones y mejoras, que servirán para convertirlo poco a poco en una aplicación más completa ampliando el mercado de clientes en potencia.

Personalmente la realización de este proyecto me ha ayudado mucho a completar y ampliar los conocimientos adquiridos anteriormente en el día a día de mi desempeño profesional en el sector de la informática. También tengo que destacar que ha sido bastante duro el compaginar mi desempeño profesional como informático y realizar este proyecto fin de grado, ya que mi preparación ha sido de una manera autodidacta y fuera del horario de mi empresa.

10.2. Líneas futuras

En este apartado vamos a hacer referencia a posibles ampliaciones de la aplicación, en primer lugar las utilidades que no nos han dado tiempo a introducir en la aplicación para Android son:

- Configuración del mail automático que se envía: en un futuro podríamos tener una opción más con la que podríamos dejar al usuario que parametrizara todos los componentes del correo, como su asunto y el texto automático que se generaría previo al envío del correo electrónico.

- Otra ampliación que se barajó como posibilidad en función del tiempo del que dispusiéramos fue la exportación a un archivo configurable por el usuario. El exportar las reservas por fechas a una hoja de cálculo.
- Otra ampliación interesante sería recordar los datos del cliente, como se hace a modo de búsqueda predictiva en los navegadores.
- Otra ampliación podría ser una mejora gráfica y dar la posibilidad de configurar una gama de colores para personalizar la aplicación.

11. Bibliografía

Para la realización de este proyecto se ha utilizado los siguientes elementos bibliográficos:

- Apuntes de asignaturas cursadas a lo largo de la carrera de Ingeniero Técnico en Informática de Gestión de la Escuela Universitaria de Informática de Segovia:

- Bases de Datos.
- Calidad del Software.
- Ingeniería del Software.

- Apuntes de asignaturas cursadas a lo largo de la carrera de Grado de Ingeniería Informática de Servicios y Aplicaciones de la Escuela Universitaria de Informática de Segovia.

- Gestión de Proyectos Basados en las Tecnologías de la Información.
- Plataformas de Software Empresariales.

- Sznajdleder, P. de (2011): “*Libro Java a fondo*”. Ed Ra-Ma. Madrid

- Sitios web de consulta:

- Android developers. Página oficial de Google para desarrolladores Android: <http://developer.android.com> (31/05/2015)
- Blog Todo lo que querías saber sobre Android y no te atreviste a preguntar: <http://androidesverdesfritos.wordpress.com/> (01/09/2014)
- Pensamientos varios sobre programación, Android, .NET y Java: <http://www.sgoliver.net/> (30/07/2014)
- SaduceLabs Blog. Blog sobre Android: <http://blog.saducelabs.com/> (01/09/2014)
- Stack Overflow. Foro de resolución de dudas referidas a todo tipo de lenguajes de programación: <http://stackoverflow.com/> (17/05/2015)

Bloque III – Manuales

Manual de usuario

1. INTRODUCCIÓN

En este apartado explicaremos a los usuarios el funcionamiento de nuestra aplicación de una manera visual y sencilla, para que cualquier usuario pueda sacar el máximo partido de la misma.

El manual comienza explicando los objetivos del proyecto y construcción de la aplicación para que se pueda entender la solución adoptada. Posteriormente, se comenta dicha solución. Una vez establecida esta base, se pasa a desgranar toda la aplicación.

2. OBJETIVOS DEL PROYECTO

El objetivo principal de esta aplicación para dispositivos móviles Android (smartphones y tablets) es dar soporte a usuarios gestores proporcionando una herramienta ágil y fácil de usar para la gestión de reservas de su casa rural.

También se tienen como subobjetivos alcanzados: la configuración de componentes de la casa rural, y gestión de las reservas realizadas, pudiendo realizar y anular reservas, como la completa visualización de clientes de la aplicación.

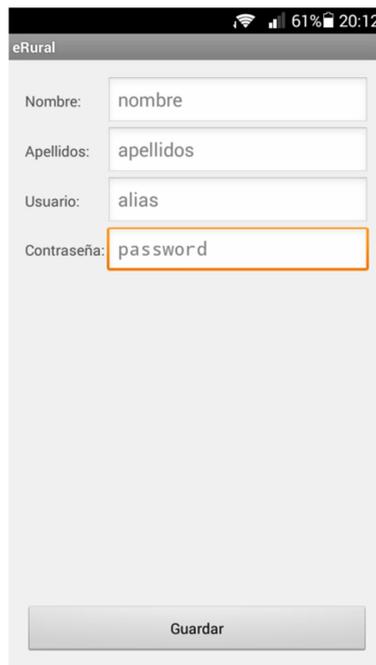
3. ESTRUCTURA DE NAVEGACIÓN DE LA APLICACIÓN

Teniendo en cuenta que existen varios subsistemas de la aplicación, el de la configuración, el de las reservas, el de la visualización de clientes y el de anulación de reservas, vamos a ver una serie de imágenes para reconstruir el correcto flujo de acción de la aplicación.

Al iniciar nuestra aplicación la primera vez, detectará que no hay un usuario gestor de la misma, por lo que pedirá al usuario registrarse, para que quede almacenada.



Figura 59. Pantalla inicial de mensaje de registro de usuario.



The screenshot shows the registration screen of the eRural application. At the top, the status bar displays the time as 20:12 and battery level at 61%. The app title 'eRural' is visible in the top left corner. Below the title, there are four input fields: 'Nombre:' with the placeholder 'nombre', 'Apellidos:' with 'apellidos', 'Usuario:' with 'alias', and 'Contraseña:' with 'password'. The 'Contraseña' field is highlighted with an orange border. At the bottom of the screen, there is a 'Guardar' button.

Figura 60. Pantalla inicial de registro de usuario.

Una vez que ya nos hemos registrado, accedemos a la pantalla principal de nuestra aplicación:



Figura 61. Pantalla inicial.

Si es la primera vez que accedemos a la aplicación, debemos configurarla, por lo que entraremos al menú de Configuración:



Figura 62. Pantalla inicial de la configuración de la aplicación.

Configuraremos por orden de primero a último, ya que si intentamos registrar por ejemplo habitaciones no nos dejará al no haber creado antes la casa.

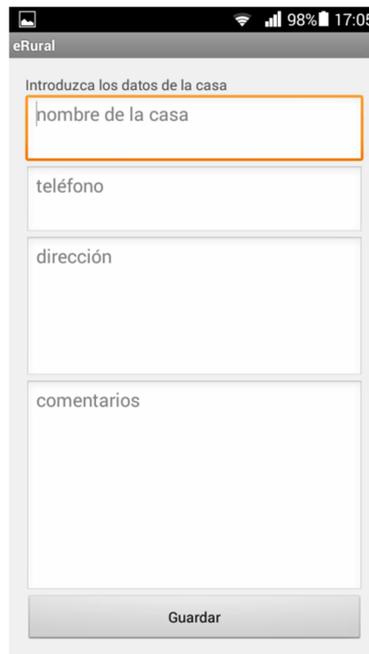


Figura 63. Pantalla de la configuración de la casa.



Figura 64. Pantalla de la configuración de las habitaciones.



Figura 65. Pantalla de la configuración de las características comunes de las habitaciones.



Figura 66. Pantalla de la configuración de las características de las habitaciones.

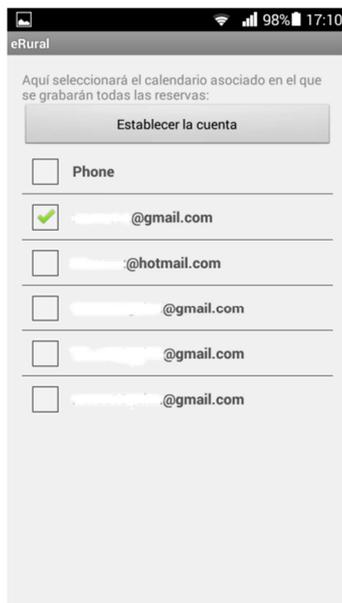


Figura 67. Pantalla de la configuración de la cuenta de correo.



eRural

Número Cuenta:

EEEE O000 DD NNNNNNNNNN

Porcentaje Reserva: %

Guardar

Figura 68. Pantalla de la configuración del pago.

Una vez que hemos hecho esta configuración inicial, no tendremos que realizarla de nuevo, a menos que queramos cambiar el nombre a alguna habitación o cambiar una característica o cambiarla de precio simplemente, pero este paso de configuración solo se dará una única vez inicial.

Por lo que ahora vamos a ver la confirmación de una reserva, su secuencia de pasos sería la siguiente:



eRural

Fecha de la reserva

Entrada: dd/mm/aaaa Inicio

Salida: dd/mm/aaaa Fin

Titulo: RHab. La1

Descripción:

Rellenar características

Consultar disponibilidad

Reservar

Figura 69. Pantalla de reserva de una habitación.

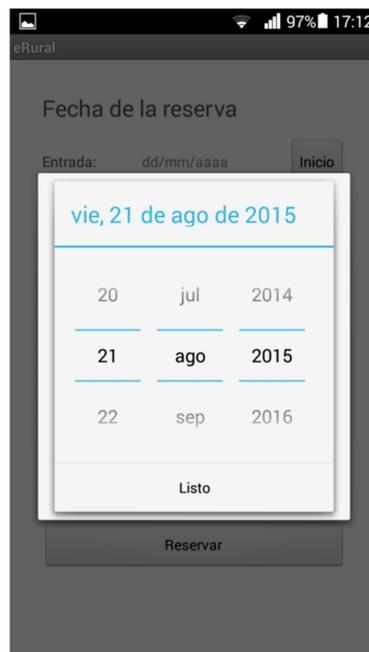


Figura 70. Pantalla de selección de fechas.



Figura 71. Pantalla de registro de características comunes de la habitación.

A partir de aquí, si le deja la aplicación reservar, es que habría disponibilidad sino le indica lo contrario, y podría guardar el cliente o no, si decide guardarlo, podrá dejarlo registrado en base de datos, y si también quiere se le confirmará mediante un correo electrónico la reserva que ha realizado.

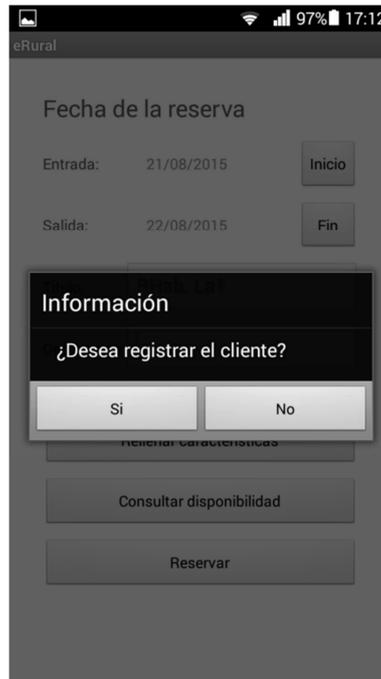


Figura 72. Pantalla de confirmación de registro de cliente.

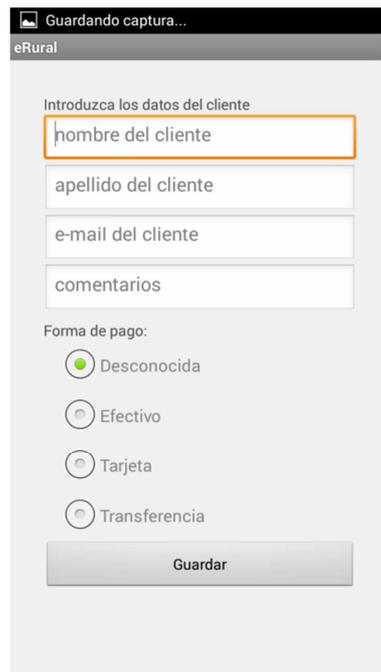


Figura 73. Pantalla de registro de cliente.

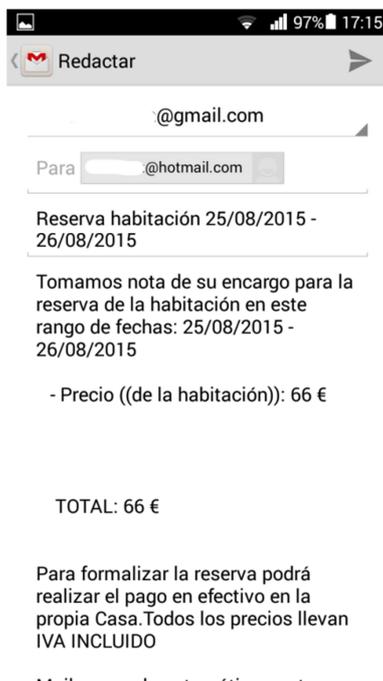


Figura 74. Pantalla de confirmación de envío de mail al cliente desde aplicación de correo.

Por lo tanto este sería el flujo normal de la configuración de nuestra aplicación y de una reserva completa de una habitación.

A continuación detallamos la vista de los clientes de la aplicación, para conocer la forma de pago de los clientes y su número de tarjeta en el caso de que paguen con tarjeta:



Figura 75. Pantalla de clientes.

Finalizamos con uno de los subobjetivos de la aplicación que es la anulación de las

reservas, en la que solamente habremos de seleccionar la reserva a anular y la aplicación nos pedirá confirmación.



Figura 76. Pantalla de anular reservas.



Figura 77. Pantalla de confirmación de anulación de reserva.

Por lo tanto este sería el flujo completo de nuestra aplicación.

4. ANEXO DE CONFIGURACIÓN DE CUENTA DE CORREO DE LA APLICACIÓN

Teniendo en cuenta que por defecto en los dispositivos Android tendrán asociados a cuentas de correo de gmail, existirán casos en los que los usuarios querrán modificar la cuenta de correo asociada al calendario y poner la de otro dominio.

Vamos a desglosar los pasos a seguir para poder adjuntar nuestra cuenta a nuestro calendario:

1. Seleccione **Configuración > Cuentas > Agregar cuenta > Correo electrónico**. Puede que esta aplicación se llame **Correo** en otras versiones de Android.
2. Para ello tenemos varias opciones, como cuenta Exchange, como cuenta POP3 o IMAP. Por ejemplo podemos configurar la cuenta Outlook.com como tipo Exchange o como cuenta POP3, otros dominios como por ejemplo yahoo se configura tipo POP3. Vamos a ver cómo configuraríamos una cuenta Outlook para nuestro terminal:

- a. Configuración Exchange:

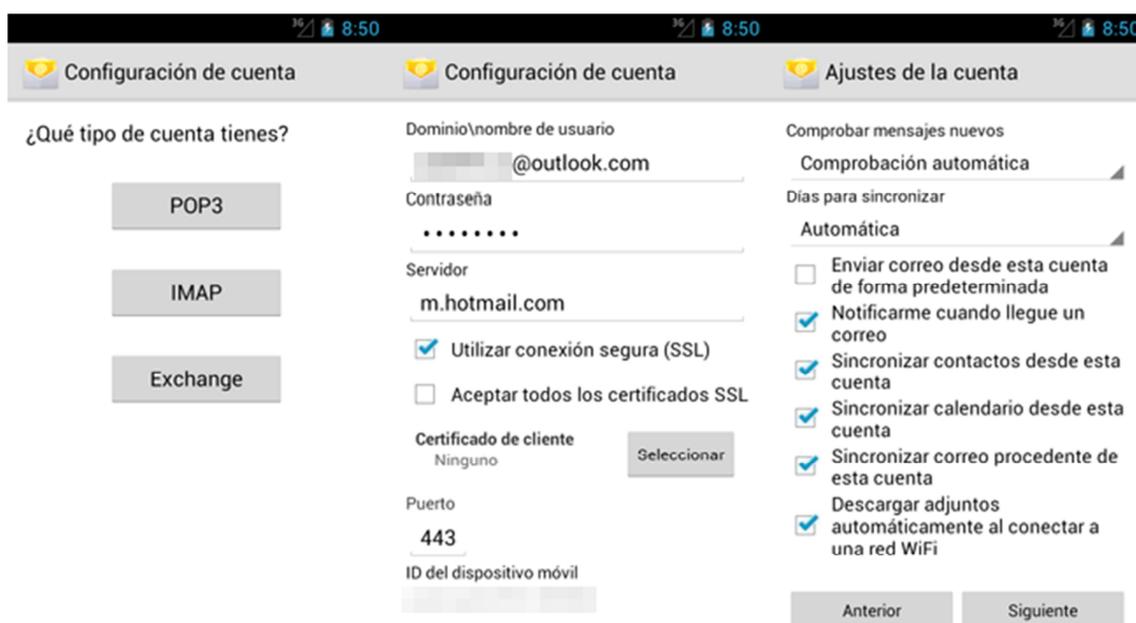


Figura 78. Pantalla de configuración de cuenta Exchange.

- b. Configuración POP3:



Figura 79. Pantalla de configuración de cuenta POP3.

Manual de instalación

1. INTRODUCCIÓN

En este documento se recoge el procedimiento a seguir para realizar la instalación de la aplicación eRural en un dispositivo Android.

Para empezar, se explicarán los requisitos que debe cumplir todo dispositivo móvil en el que se quiera instalar la aplicación para su correcto funcionamiento. Seguidamente, se explicará el proceso completo de instalación de la aplicación.

2. REQUISITOS DE INSTALACIÓN

La aplicación ha sido desarrollada para funcionar únicamente en dispositivos con sistema operativo Android, es por ello que el primer requisito y más importante es que el dispositivo en el que se quiera instalar la aplicación tenga dicho sistema operativo.

Como en todos los sistemas operativos, Android dispone de varias versiones. Cada versión contiene nuevas funcionalidades respecto a la anterior y necesitará un hardware más potente. En el caso concerniente, eRural ha sido diseñada para funcionar a partir de la versión 4.0 de Android (Ice Cream Sandwich).

Además de lo anterior, para realizar la instalación se necesita espacio libre, ya sea interno o externo (tarjeta de memoria compatible con el terminal), donde copiar el archivo de instalación de la aplicación.

3. INSTALACIÓN

El proceso de instalación se realiza en varios pasos:

Copiar el archivo de instalación en el dispositivo móvil, consiste en ubicar el archivo de instalación en una ruta visible para poder ejecutarlo.



Figura 80. Pantalla de la ubicación del archivo.

Una vez ubicado, procedemos ejecutar el archivo que anteriormente se ha copiado.

Antes de realizar la instalación será necesario activar la opción de “orígenes desconocidos” en el dispositivo Android. Para ello hay que ir a Ajustes / Seguridad (o Aplicaciones dependiendo de la versión de Android) / Orígenes desconocidos (ver imágenes siguientes).

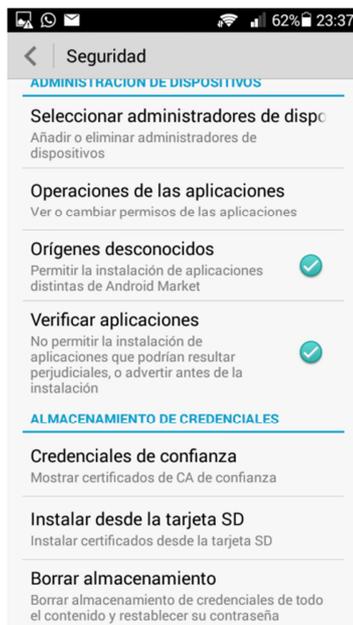


Figura 81. Pantalla de la seguridad.

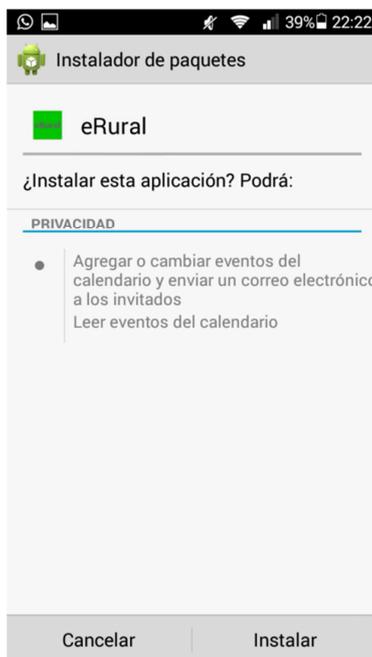


Figura 82. Pantalla de instalar la aplicación.

Importante también es dar permisos a la aplicación:

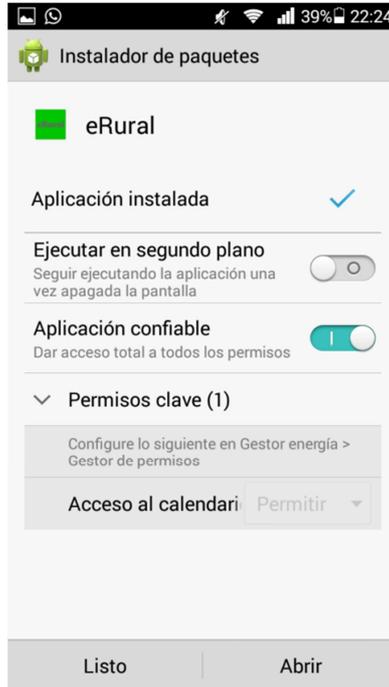


Figura 83. Pantalla de permisos de la aplicación.

Y ya se puede Abrir la aplicación instalada correctamente en nuestro dispositivo Android.