



Introducción

En este documento se presenta la información del proyecto que se proporciona.

1. Descripción del sistema de información
2. Modelo del Dominio
3. Modelo de Casos de Uso
4. Flujos de Eventos
5. Diagramas de Secuencia
6. Implementación e instrucciones para poner en marcha la aplicación.

1. Descripción del sistema de información

Se desea desarrollar una aplicación de gestión de reservas de casas rurales, que sirva tanto para los propietarios de las mismas como a sus posibles clientes.

Los propietarios de las casas rurales pueden darlas de alta en el sistema, proporcionando los siguientes datos: población donde se encuentra, número de dormitorios, baños, cocinas y comedores así como el número de plazas de garaje. Además, se puede incluir una descripción general de la casa. Las normas sobre casas rurales obligan a que por lo menos haya 1 cocina, 3 habitaciones y 2 baños. Por supuesto, los propietarios también podrán dar de baja en el sistema a las casas rurales.

Las casas rurales se alquilan por paquetes de días y cada propietario los organiza como quiere. Por ejemplo, en agosto se alquila por quincenas (del 1 al 15 y del 16 al 31), el puente del 1 de mayo entero (del 28 de abril al 1 de mayo), en julio por semanas, en noviembre por días, etc. Cada paquete tiene su precio.

Para no perder clientes potenciales, no es necesario estar registrado en el sistema para consultar por casas rurales o hacer reservas. Sin embargo, para insertar casas rurales sí que hay que estar registrado en el sistema.

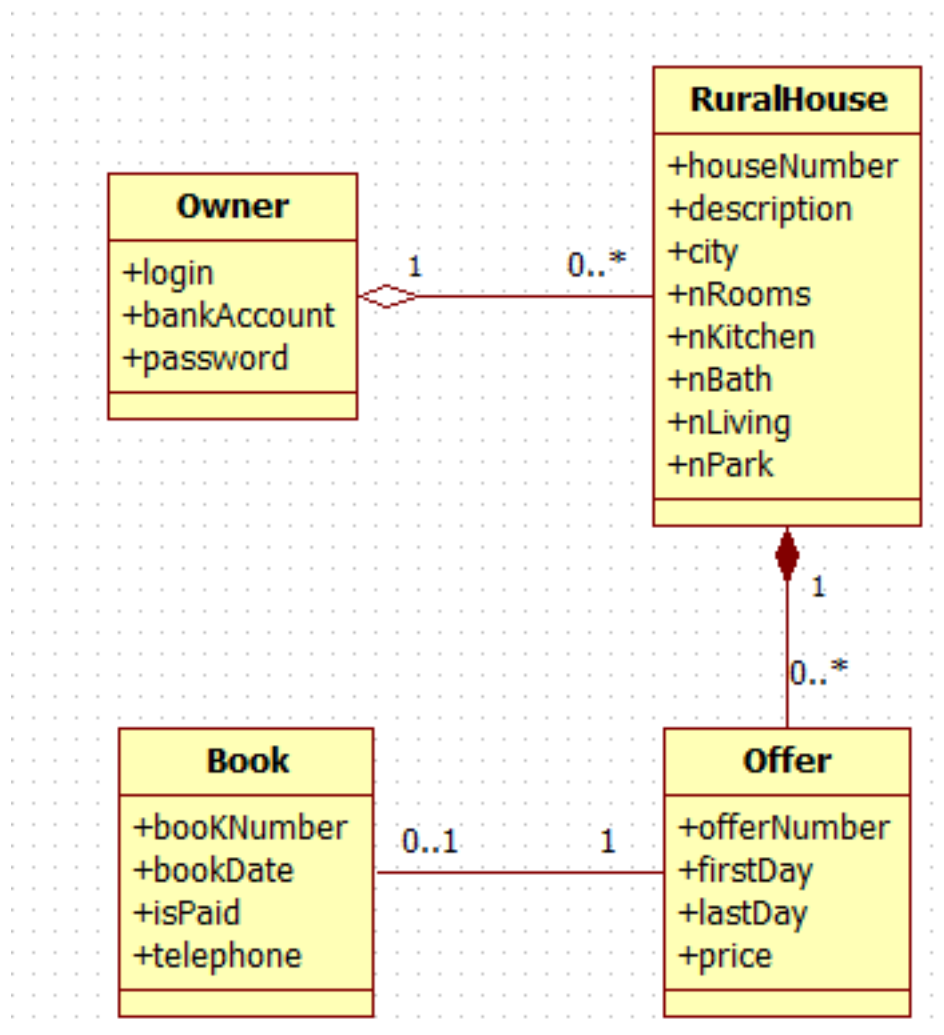
El sistema permite realizar búsquedas de las casas rurales por población. En ese caso se obtendrá una lista de las casas rurales de la población, donde podrá seleccionar una de ellas para ver las características de la casa. Si el usuario conoce el código de la casa rural, el sistema le dará la opción de ir directamente a conocer sus características.

Para ver los paquetes de días disponibles de una casa, el usuario debe proporcionar el código de la casa rural, el día de entrada y la cantidad de noches. Dichos paquetes no se pueden partir, esto es, la casa se podría reservar el conjunto de días completo y no un subconjunto de los mismos. Si no hay un paquete exacto para los días solicitados por el usuario, el sistema devolverá ofertas que cumplen parcialmente la petición del usuario. Por ejemplo, si los paquetes posibles son: 5/5-7/5, 10/5-14/5 y 17/5-20/5, y el usuario quisiera reservar 15 días a partir del 3/5, el sistema devolvería los siguientes paquetes: 5/5-7/5, 10/5-14/5.

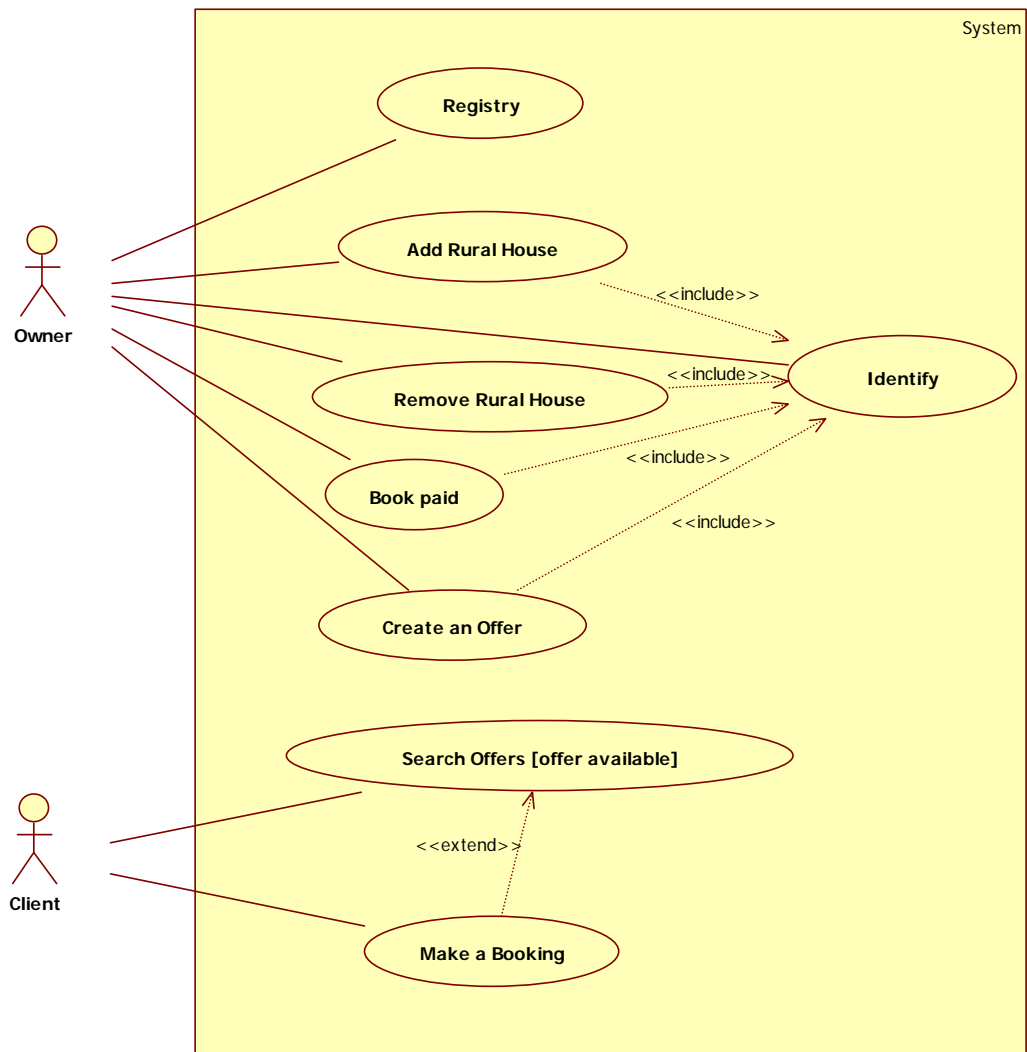
Para reservar una determinada casa habrá que indicar los siguientes datos: código de la casa, día de entrada y el número de noches que desea alquilar. El sistema comprobará si el alojamiento está libre en esas fechas y si lo está, se le mostrará un

número de cuenta del propietario y una cantidad de dinero para que el usuario ingrese como fianza, y se le pedirá al usuario que introduzca un número de teléfono. Cuando el propietario compruebe más adelante que se ha realizado el pago, se pondrá la reserva como pagada.

2. Modelo del dominio



3. Modelo de Casos de Uso



4. Flujos de Eventos

CASO DE USO: Identify

1. El propietario introduce el nombre y la clave
2. El sistema, si existe un propietario con ese nombre y clave, le deja entrar

Flujo de eventos alternativo:

- 1.- Si no hay propietario con ese nombre. Fin.
- 2.- Si no coincide la clave para ese nombre. Fin.

CASO DE USO: Registry

1. El propietario introduce el número de cuenta, el nombre y la clave.
2. El sistema crea un propietario con dichos datos.
3. El sistema comunica al propietario que ha creado la cuenta.

CASO DE USO: Add Rural House

1. El propietario proporciona la población, número de habitaciones, baños, cocinas, comedores y aparcamientos, y si lo desea, una descripción de la casa.
2. El sistema crea una casa rural nueva y se la asigna el propietario.

Flujo de eventos alternativo:

- 1.- Si el número de cocinas es mejor que 1. Fin.
- 2.- Si el número de habitaciones es mejor que 3. Fin.
- 3.- Si el número de baños es mejor que 2. Fin.

CASO DE USO: Remove Rural House

1. El sistema muestra todas las casas rurales del propietario.
2. El propietario escoge una casa.
3. El sistema borra todas las reservas y paquetes de esa casa. Si hubiera una reserva, se le comunica al propietario.
4. El sistema borra la casa rural.



CASO DE USO: Book paid

1. El sistema presenta todas las casas rurales del propietario.
2. El propietario escoge una casa.
3. El sistema presenta al propietario todas las reservas sin pagar para esa casa rural.
4. El propietario escoge una reserva.
5. El sistema pone el estado "pagado" a dicha reserva

Flujo de eventos alternativo:

- 1.- Si existe una reserva caducada, se pregunta al propietario si desea eliminarla, indicándole el número de teléfono de quien hizo la reserva. Continuar.

CASO DE USO: Create an Offer

1. El sistema presenta todas las casas rurales del propietario.
2. El propietario escoge una casa.
3. El propietario introduce el día de inicio y número de días del paquete (offer)
4. El sistema crea el paquete para la casa rural escogida.

Flujo de eventos alternativo:

- 1.- Si el paquete que se está insertando colisiona en fechas con alguno existente, da la opción de borrarlo (pero sólo si está libre), o permite modificar los días de inicio y número de días del nuevo paquete. Continuar.

CASO DE USO: Search Offers

1. El cliente proporciona un nombre de población.
2. El sistema busca las casas de esa población y devuelve una lista de las mismas.
3. El cliente escoge una casa rural.
4. El cliente proporciona la fecha de entrada y número de días.
5. El sistema devuelve los paquetes que cumplen esas condiciones y ofrece la posibilidad de realizar la reserva. Si no hubiera paquete apropiado, se le hace saber al cliente y se le muestran paquetes de días cercanos.

CASO DE USO: Make a Booking

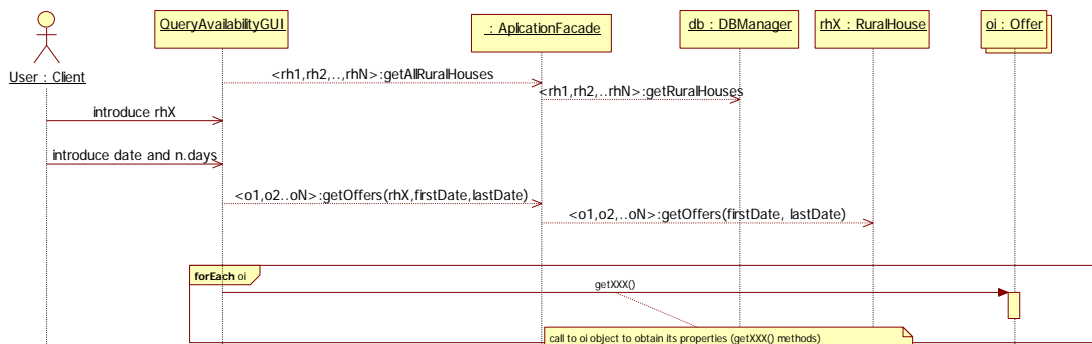
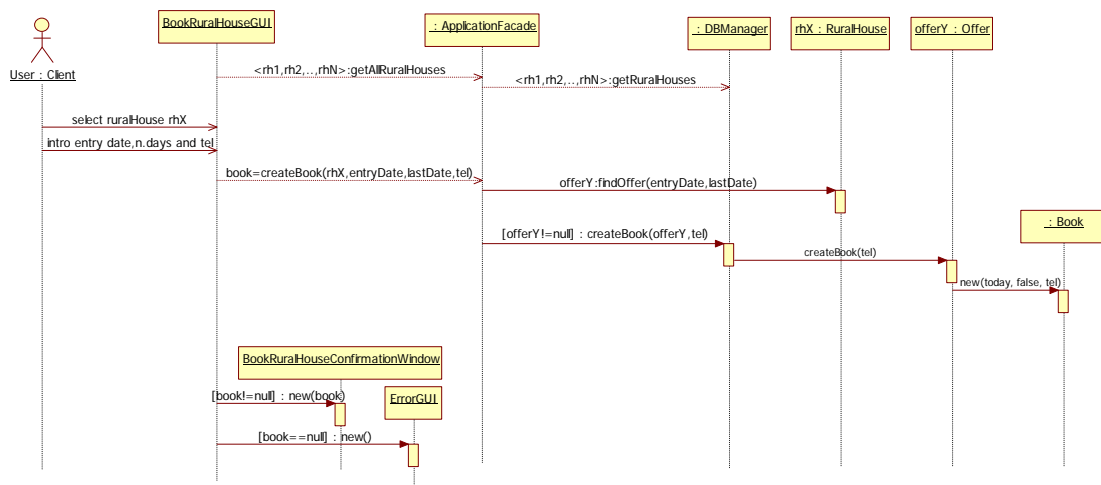
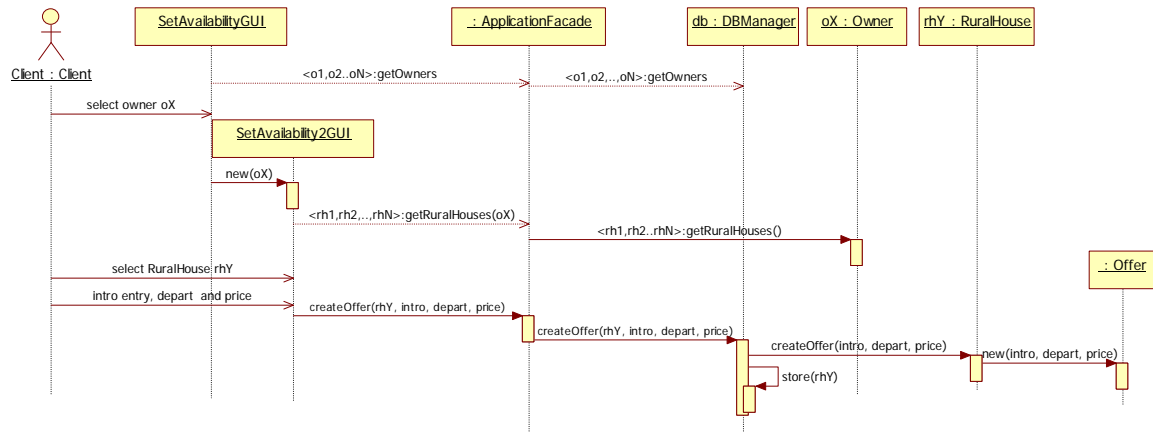
1. El cliente proporciona el código de la casa, el día de entrada y cantidad de días.
2. El sistema comprueba que hay paquetes libres para esos días.
3. Si hay paquete libre para esas fechas, el sistema muestra el número de cuenta corriente del propietario, solicita un número de teléfono al cliente y crea la reserva con esas datos.
4. Si no hay reserva libre, se lo comunica al cliente.

Flujo de eventos alternativo:

- 1.- Si no hay una casa rural con ese código. Fin.

5. Diagramas de secuencia de 3 casos de uso

La aplicación está diseñada utilizando una arquitectura en 3 niveles. Se utiliza AWT/SWING para definir las interfaces gráficas, el acceso a la lógica del negocio situada en otro ordenador remoto se realiza por medio de RMI, y para el nivel de datos, se utiliza la base de datos orientada a objetos db4o. Para agrupar todas las operaciones de la lógica del negocio se ha aplicado un patrón de diseño GRASP, y por ello se ha definido una clase CONTROLADOR, con todas las operaciones. A continuación presentamos los diagramas de secuencia de 3 casos de uso.





6. Implementación e instrucciones para poner en marcha la aplicación.

En el fichero (Proyecto.zip) se encuentran los paquetes necesarios para poner en marcha la aplicación. La aplicación está organizada en 6 paquetes:

gui: clases con las interfaces gráficas
businessLogic: clases con la lógica del negocio
domain: clases que definen el modelo del dominio

dataAccess: clases para proporcionar la persistencia

configuration: paquete que guarda los parámetros de configuración

exception: paquete donde se encuentran las excepciones

Nota: en la implementación proporcionada no se ha tenido en cuenta el flujo alternativo correspondiente al caso de uso "Create an Offer". Si alguien se anima a implementarlo, que lo haga y lo añada a la documentación presentada.

Hay que seguir los siguientes pasos para ejecutar dicha aplicación.

1. Descargar el fichero Proyecto.rar del sitio Moodle de la asignatura y descomprimirlo en C:\. Esto hará que el código y los ficheros necesarios se copien en C:\Proyecto. Si no se hace así, en los siguientes pasos, habrá que asegurarse de que los ficheros con la política de seguridad, la base de datos DB4o y las librerías DB4o tienen los nombres correctos.
2. Importar el proyecto Java (File->Import->General-> "Existing projects into workspace" y seleccionar el directorio C:\Proyecto.
3. Comprobar que las librerías necesarias están accesibles. Para ello posicionarnos encima del nombre del proyecto y con el botón derecho del ratón seleccionar "Properties". Seleccionar "Java Build Path" en el menú de la izquierda que aparece en la pantalla y en la opción "Libraries" a la derecha de la pantalla pulsar "Add external Jar". Las librerías que tenéis que cargar las podéis encontrar en el fichero proporcionado:
 - a) Librería JCalendar, en la carpeta AdditionalLibraries.
 - b) Todas las clases de la librería Db4o, esto es, todos los ficheros jar que se encuentran en el directorio ObjectManager-7.14/lib.
4. Configurar la aplicación. Hay que configurar las variables de la clase Config que se encuentra en el paquete configuration:
 1. El atributo javaPolicyPath, con la dirección en la que se encuentra el fichero java.policy (proporcionado en Proyecto.rar).
 2. El atributo db4oFilename, con el nombre del fichero y carpeta donde se dejará la base de datos. La carpeta que se proporcione debe existir (o crearse) en el sistema.
 3. El atributo dataBaseOpenMode, para definir el modo en que queremos que se abra la base de datos. Hay dos posibilidades: (1) *initialize*, para que se inicialice la base de datos con unos valores por defecto, y (2) *open*, para que se abra con los valores que contenga la base de datos. Usando este último modo, si la base de datos no existe, se crea una nueva.



5. Lanzar el servidor RMI. "Run" la clase `businessLogic.RemoteServer`.
6. Lanzar la interfaz gráfica. Run la clase `gui.StartWindow`. En este clase se puede configurar el atributo "*isLocal*". Si se configura "*isLocal*" a `true`, la lógica del negocio y la interfaz gráfica están en la misma máquina, y se accederá como una clase local. Sin embargo, si "*isLocal*" es `false`, entonces quiere decir que la lógica del negocio y la interfaz gráfica de presentación se encuentran en distintas máquinas y que se accederá por medio de RMI.