

Introducción

En este documento se presenta la información del proyecto desarrollado hasta el momento:

1. Descripción inicial del sistema de información
2. Modelo del Dominio
3. Modelo de Casos de Uso
4. Flujos de Eventos
5. Diagrama de Secuencia del caso de uso "Set Availability"
6. Código fuente
7. Creación del fichero .jar.

1. Descripción inicial 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.

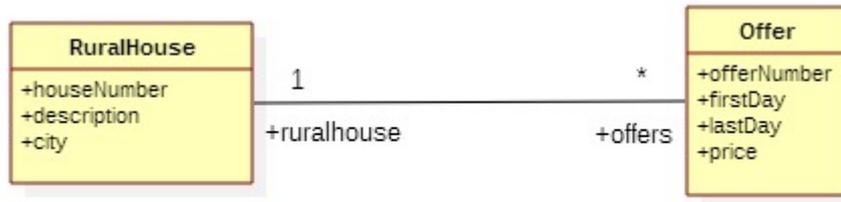
Para reservar una determinada casa habrá que indicar los siguientes datos: código de la casa, día de entrada y el número de noches que desea alquilar. El sistema comprobará si el alojamiento está libre en esas fechas y si lo está, se le mostrará un número de cuenta del propietario y una cantidad de dinero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

NOTAS:

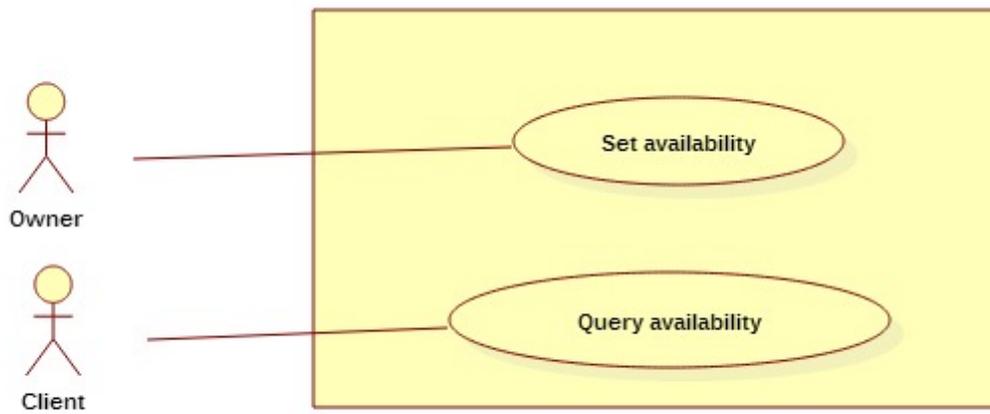
En las siguientes secciones se presentan los artefactos correspondientes al resultado de la iteración ya realizada: el modelo del dominio, el modelo de casos de uso, los de flujos de eventos de los 2 casos de uso desarrollados y el diagrama de secuencia para uno de esos casos de uso.

En esta primera iteración no se han realizado todos los casos de uso que corresponderían a la anterior descripción, ni todos los datos necesarios aparecen reflejados en el modelo del dominio.

2. Modelo del dominio



3. Modelo de Casos de Uso



4. Flujos de Eventos

Flujo de eventos del caso de uso: Set availability

Flujo de eventos normal

1. El *sistema* muestra todas las casas.
2. El *propietario* escoge una casa para la que quiere introducir un paquete (offer).
3. El *propietario* introduce el día de inicio, el número de días del paquete y el precio.
4. El *sistema* almacena 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 entonces no se permite introducirlo. El propietario podrá modificar los datos de día de inicio y el número de días para insertar otro paquete. Continuar.

Flujo de eventos del caso de uso: Query availability

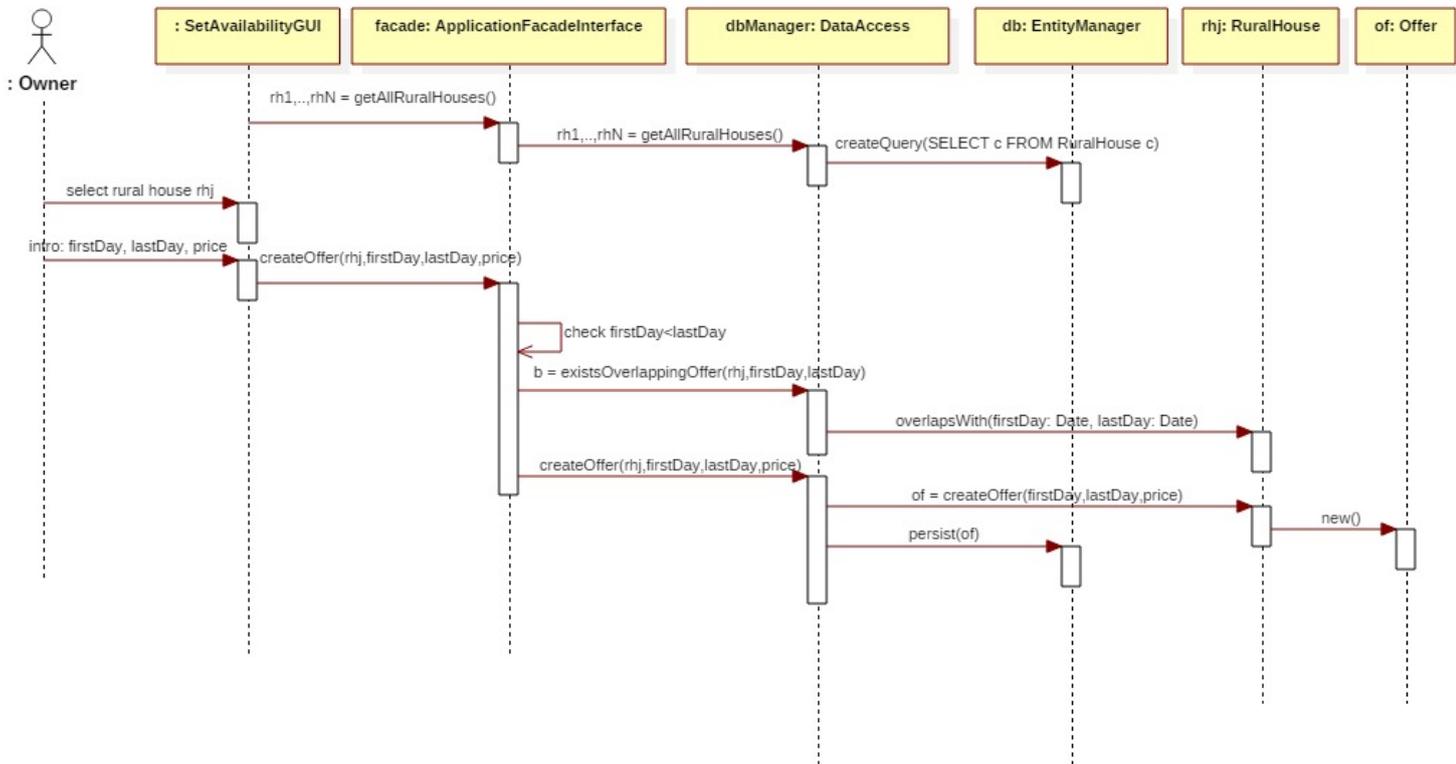
Flujo de eventos normal

1. El *sistema* muestra todas las casas al *cliente*.
2. El *cliente* escoge una casa.
3. El *cliente* introduce la fecha de entrada y el número de días de estancia.
4. El *sistema* devuelve y muestra las ofertas cuyas fechas de entrada y salida están incluidas completamente en las fechas establecidas por el cliente.

Flujo de eventos alternativo

1. Si una oferta tiene algún día fuera del intervalo indicado por el cliente, entonces no se muestra.

5. Diagrama de secuencia del caso de uso “Set availability”



6. Código fuente.

El código fuente de la aplicación de casas rurales, resultado de la iteración 0 se encuentra disponible en la sección “Proyecto” de curso OCW (fichero WSRuralHouse.zip)

Para usar el proyecto en el entorno de desarrollo Eclipse hay que hacer lo siguiente:

- 1) Descomprimir el fichero en el disco duro. Se creará una carpeta llamada WSRuralHouse que contiene el código fuente Java, las librerías que trabajan con calendarios JCalendar y con bases de datos objectDB, y algunos ficheros que se encuentran en el directorio raíz del proyecto. Se trata del fichero donde se almacenará la base de datos (ruralHouses.odb) y un fichero XML con datos de configuración para la aplicación de casas rurales.

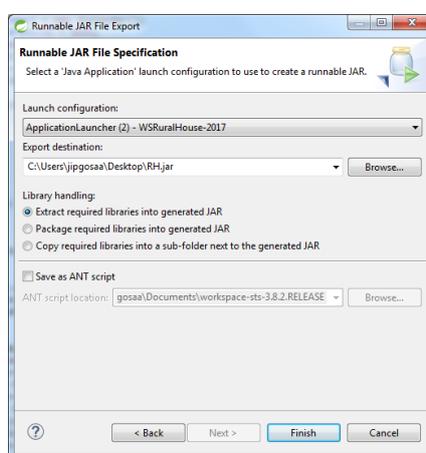
- 2) Importar el proyecto desde Eclipse-

File=>Import => General => “Existing projects into workspace” y seleccionar el directorio RuralHousesProject que se ha descomprimido

7. Creación del fichero .jar utilizado en el primer laboratorio de la asignatura.

Creación del fichero RH.jar: Hacer click derecho sobre el proyecto RuralHousesProject en el explorador de paquetes (Package Explorer) de Eclipse => Export => Java => Runnable Jar File => Escoger la configuración correspondiente a ejecutar “ApplicationLauncher” en el proyecto RuralHouses y dar un nombre (RH.jar en este caso) y destino. En este caso, se ha seleccionado la opción para que las librerías usadas se incluyan en el .jar también.

NOTA: si al ejecutar el .jar se necesita algún recurso (por ejemplo la base de datos resources/ruralHouses.odt), entonces donde se encuentra el .jar tendría que situarse la carpeta “resources” que contenga el fichero ruralHouses.odt



Creación del fichero RH.zip: En un directorio RH, incluiremos el fichero anterior RH.jar, un directorio llamado “resources” que contenga el fichero config.xml, y se comprimirá el directorio RH para obtener el fichero comprimido RH.zip.

NOTA: si se quisieran añadir los ficheros fuente .java en el jar ejecutable

La opción anterior no incluye los fuentes Java, pero como el .jar es un fichero comprimido (como un .zip, .rar, etc.) se pueden incluir abriendo el fichero jar con un programa descompresor (winzip, winrar, 7-zip) y agregando los ficheros fuente. En la figura siguiente se puede ver el resultado de haber arrastrado desde el proyecto el directorio “src” que contiene todos los fuentes java en el fichero RH.jar abierto con el programa 7-zip

