

PRÁCTICA 2 MED

Buscador de Hoteles

Martin Quinteiro Santos
Francisco Pais Fondo

martin.quinteiro
francisco.pais1

Las historias de usuario definidas a continuación nos servirán de guía para calcular el tiempo de realización de las mismas. Comenzamos dividiendo los casos de uso identificados en la primera entrega en historias de usuario. A continuación describimos cada una de las historias y estimamos el tiempo que llevará implementarlas. Una vez hecho esto, consultamos con el cliente si la división es correcta y a continuación procedemos a jerarquizar las distintas historias según las prioridades indicadas por el cliente:

Historias de usuario jerarquizadas según el criterio del cliente

- Ver ofertas: 2 horas
 - El usuario podrá ver las ofertas disponibles a través de la sección Ofertas, así como los detalles de las mismas.-
- Buscar hotel
 - Buscar hotel por ciudad: ½ hora-
 - El usuario podrá buscar los hoteles que se encuentran en una determinada ciudad.
 - Buscar hotel por nombre: ½ hora-
 - El usuario podrá buscar los hoteles que tengan o contengan el nombre especificado.
 - Buscar hotel por calle: ½ hora-
 - El usuario podrá buscar los hoteles que estén ubicados en la calle especificada.
 - Buscar hotel por categoría: ½ hora-
 - El usuario podrá buscar el/los hotel(es) que tengan una categoría determinada.
- Ordenar hoteles
 - Ordenar hoteles por nombre: ½ hora-
 - Tras la búsqueda de los hoteles, el usuario podrá ordenarlos alfabéticamente por su nombre.
 - Ordenar hoteles por categoría: ½ hora-
 - Tras haber realizado la búsqueda de los hoteles, el usuario podrá ordenarlos por categoría.
 - Ordenar hoteles por precio: 1 hora-
 - Tras la búsqueda de los hoteles, el usuario podrá ordenarlos de forma ascendente por precio.
- Filtrar hoteles
 - Filtrar por disponibilidad: 1 hora-
 - Se podrán filtrar los hoteles encontrados por su disponibilidad en unas fechas determinadas.

- Filtrar por estrellas: ½ hora-
 - Se podrán filtrar los hoteles encontrados por su categoría, según el número de estrellas indicado.
- Filtrar por precio: 1 hora-
 - Se podrán filtrar los hoteles encontrados según el precio deseado de la reserva.
- Filtrar por servicios: 1 hora-
 - Se podrán filtrar los hoteles encontrados por los servicios que indica el usuario.
- Filtrar por número de habitaciones: 1 hora-
 - Se podrán filtrar los hoteles encontrados por el número de habitaciones que se desean reservar (según disponibilidad).
- Ver hotel
 - Ver servicios: 1 hora-
 - Se podrán consultar los servicios de los que dispone el hotel y de los cuales el cliente podrá disfrutar.
 - Ver habitaciones: 1 hora-
 - El usuario, previamente a reservar podrá ver las distintas habitaciones para motivar su decisión.
 - Consultar disponibilidad: 2 horas-
 - Previamente a reservar, el usuario podrá consultar si el hotel cuenta con habitaciones disponibles para su reserva.
 - Ver temporada: ½ hora-
 - Se podrá consultar si el hotel cuenta con oferta de temporada o no, la cual hará que su precio ordinario disminuya.
- Ver habitación (de un hotel)
 - Consultar información: ½ hora-
 - El usuario podrá visualizar la información detallada de la habitación indicada.
 - Consultar servicios: ½ hora-
 - El usuario podrá ver los servicios que tiene a disposición en dicha habitación.
 - Ver regímenes: ½ hora-
 - El usuario podrá ver los distintos regímenes que tiene la habitación, así como sus condiciones.
 - Ver tarifas: ½ hora-

- El usuario puede ver los distintos precios dentro del régimen indicado, así como los servicios que ofrece dicha tarifa.
- Reservar habitación (de un hotel): 1 hora
 - El usuario tendrá que rellenar un formulario para poder enviar la solicitud de reserva. Deberá aportar los siguientes datos:
 - Nombre
 - Apellidos
 - Email
 - Teléfono
 - Observación(opcional)
 - Número de tarjeta

Planificación semanal

La realización de las historias de usuario se dividirán en ciclos semanales, según la prioridad indicada por el cliente. En nuestro caso, los ciclos semanales comienzan los Martes, día en el que nos reunimos con el cliente para aclarar las historias a realizar en el siguiente ciclo semanal y para obtener el feedback a través de la utilización de las funcionalidades ya implementadas. El desarrollo de las historias y tests se realizan en pareja trabajando de forma alterna. Por las mañanas trabajamos en un mismo equipo aproximadamente 5 horas, momento que aprovechamos para discutir sobre la toma de decisiones y corregirnos mutuamente los errores que vayan surgiendo. Por la noche se trabaja de forma independiente (dividiendo el trabajo a realizar), haciendo uso de subversion para el control de versiones, durante 4 horas.

A continuación intentamos plasmar toda la información importante que vaya surgiendo durante el transcurso de cada ciclo semanal:

- Semana 1 (06/10/2015 - 13/10/2015)
 - Al hablar con el cliente, este nos ha indicado que sus prioridades eran los métodos de búsqueda y ordenación con lo que nuestra propuesta para la primera semana de trabajo será implementar el servicio REST con todos los métodos de ordenación y búsqueda definidos en las historias de usuario. Además de esto, también se creará una base de datos para su realización basándonos en el esquema entidad-relación realizado para la primera entrega.
 - Crearemos un nuevo proyecto web al que le añadiremos Jersey para gestionar las invocaciones al servicio (<http://www.vogella.com/tutorials/REST/article.html>) e Hibernate para la gestión de la persistencia.

- El desarrollo de las historias de usuario se hará siguiendo el desarrollo dirigido por los test, es decir, para cada historia de usuario definiremos su test y a continuación procedemos a la implementación de la funcionalidad que lo cumpla. Intentamos realizar los test utilizando Jersey Test Framework pero debido a un error de compatibilidad entre distintas versiones de JAX-RS no pudimos hacerlo. El error es el siguiente "java.lang.NoSuchMethodError javax.ws.rs.core.Application.getProperties()Ljava". Tras buscar en Google el error y consultar varias páginas (sirva de ejemplo la siguiente) (<http://stackoverflow.com/questions/19225618/java-restfull-webservice-jax-rs-i-mplementation-with-jersey-2-3-1-libraries>), no conseguimos solucionar el error ni encontrar el desencadenante del mismo, por lo que perdimos bastante tiempo a pesar de seguir paso a paso el tutorial de Jersey Test Framework (<https://jersey.java.net/documentation/latest/test-framework.html>). Por este motivo y por falta de tiempo, hemos decidido realizar los test sin utilizar Jersey Test Framework utilizando únicamente JUnit.
- A la hora de implementar el test de la historia "buscar por lugar", nos entró la duda de si el cliente deseaba buscar los hoteles por ciudad o dirección. Al consultarlo con él, nos dice que en principio, lo más relevante será buscar los hoteles por ciudad. En el modelado entidad-relación inicial habíamos contemplado la ciudad de un Hotel como parte de la dirección del mismo, lo que no permitiría distinguir si la búsqueda se realiza por dirección o por ciudad (contenida en la dirección) del hotel. Por lo tanto, decidimos añadir los atributos "ciudad" y "calle" que sustituirán al atributo "dirección" inicial.
- Previo a la implementación de las distintas historias de búsqueda, para gestionar la persistencia de los datos en la base de datos, hemos decidido añadir hibernate a nuestro proyecto. Hibernate actúa como mapeador objeto relacional que nos permite mapear automáticamente los objetos con su correspondiente entidad en base de datos, agilizando mucho el proceso de desarrollo. Para incluir hibernate, ha sido necesario realizar diversas configuraciones como la de incluir más librerías al proyecto, modificar el archivo web.xml, crear los archivos de configuración de Hibernate (hibernate.cfg.xml y Hotel.hbm.xml) y además añadir los DAO's y los servlets correspondientes para poder procesar las peticiones.
- Semana 2 (13/10/2015 - 16/10/2015)
 - Una vez obtenido el feedback por parte del cliente, este nos ha comunicado que debíamos cambiar ciertos aspectos en el código que hasta ahora teníamos implementado. Uno de estos cambios sería añadir el precio de la mejor oferta de cada hotel en el momento en el que se ordenan los hoteles por precio. Además de ese cambio, otra modificación que nos ha comunicado es la de utilizar JSON o XML como formato estándar de salida por pantalla.

- El cliente prioriza la realización de las historias consultar disponibilidad, ver habitación y reservar habitación que procedemos a implementar, dejando las demás historias en un plano secundario, pero que se retomarán si hay margen de tiempo.
- Eliminamos id_hotel de Reserva, pues a través de Habitación y TipoHabitación se puede averiguar el hotel al que hace referencia dicha reserva. También decidimos incluir el atributo enTemporada en Hotel para poder ver si un hotel está en temporada o no.
- Hemos realizado una sobreestimación del tiempo de implementación de las historias de usuario, pues la mayoría (aunque no todas) refleja un tiempo de realización inferior al real.
- También cabe destacar que la mayor parte del tiempo se ha empleado en la búsqueda acerca de cómo configurar las distintas utilidades sobre las que nos apoyamos para implementar el servicio Rest (Hibernate, Jersey, dependencias, ...) así como en la búsqueda de los distintos errores obtenidos durante la realización del proyecto. Como prueba de ello podemos mencionar la mañana del día 14 como la menos productiva de todas, pues la empleamos en buscar un error de Hibernate (que no mostraba información de ayuda de ningún tipo ni tan siquiera ejecutando el proyecto en Tomcat se podían ver las trazas del flujo de ejecución) que se debía a un fallo en el nombre de un atributo de una de las clases Java. Como consecuencia de todos estos errores, del tiempo empleado en las búsquedas en Google y del tiempo de configuración de las utilidades empleadas, el tiempo empleado en el desarrollo del servicio Rest ha sido mayor de lo esperado, pues la mayor parte del tiempo no ha sido empleado en la realización de las historias de usuario.
- Dado que implementamos las historias prioritarias para el cliente, procedemos a la implementación de las demás historias que son menos prioritarias.

Comprobación del funcionamiento de las historias a través del navegador

A continuación aportamos el formato de las URL's para ejecutar cada una de las historias de usuario realizadas.

- Ver ofertas: 2 horas
 - http://localhost:8080/Hotels/Rest/Hotels/1/Habitaciones/{id_Habitacion}/Tarifas

- Buscar hotel
 - Buscar hotel por ciudad:
 - [http://localhost:8080/Hotels/Rest/Hotels?ciudad="ciudad"](http://localhost:8080/Hotels/Rest/Hotels?ciudad=)
 - Buscar hotel por nombre:
 - [http://localhost:8080/Hotels/Rest/Hotels?nombre="nombre"](http://localhost:8080/Hotels/Rest/Hotels?nombre=)
 - Buscar hotel por calle:
 - [http://localhost:8080/Hotels/Rest/Hotels?calle="calle"](http://localhost:8080/Hotels/Rest/Hotels?calle=)
 - Buscar hotel por categoria:
 - <http://localhost:8080/Hotels/Rest/Hotels?categoria=categoria>
- Ordenar hoteles
 - Ordenar hoteles por nombre:
 - <http://localhost:8080/Hotels/Rest/Hotels?orderby=nombre>
 - Ordenar hoteles por categoría:
 - <http://localhost:8080/Hotels/Rest/Hotels?categoria=categoria>
 - Ordenar hoteles por precio:
 - <http://localhost:8080/Hotels/Rest/Hotels?precio=precio>
- Filtrar hoteles
 - Filtrar por disponibilidad: 1 hora-
 - <http://localhost:8080/Hotels/Rest/Hotels?fechaEntrada=fecha&fechaSalida=fecha>
 - Filtrar por estrellas:
 - <http://localhost:8080/Hotels/Rest/Hotels?categoria=categoria>
 - Filtrar por precio:
 - <http://localhost:8080/Hotels/Rest/Hotels?preciominimo=precio&precioMaximo=precio>
 - Filtrar por servicios:
 - <http://localhost:8080/Hotels/Rest/Hotels?servicios=servicios>
 - Filtrar por número de habitaciones:
 - <http://localhost:8080/Hotels/Rest/Hotels?numeroHabitaciones=numero>
- Ver hotel
 - Ver servicios:
 - http://localhost:8080/Hotels/Rest/Hotels/{id_hotel}/Servicios

- Ver habitaciones:
 - http://localhost:8080/Hotels/Rest/Hotels/{id_hotel}/Habitaciones
- Consultar disponibilidad:
 - http://localhost:8080/Hotels/Rest/Hotels/{id_hotel}/Habitaciones?fechaEntrada=fecha&fechaSalida=fecha
- Ver temporada:
 - http://localhost:8080/Hotels/Rest/Hotels/{id_hotel}/
- Ver habitación (de un hotel)
 - Consultar información:
 - http://localhost:8080/Hotels/Rest/Hotels/1/Habitaciones/{id_Habitacion}
 - Consultar servicios:
 - http://localhost:8080/Hotels/Rest/Hotels/1/Habitaciones/{id_Habitacion}/Servicios
 - Ver régimenes:
 - http://localhost:8080/Hotels/Rest/Hotels/1/Habitaciones/{id_Habitacion}/Regimenes
 - Ver tarifas:
 - http://localhost:8080/Hotels/Rest/Hotels/1/Habitaciones/{id_Habitacion}/Tarifas
- Reservar habitación (de un hotel):
 - <http://localhost:8080/Hotels/Rest/Hotels/1/Habitaciones/1/Tarifas/1/Reservar?nombre=nombre&apellido1=apellido1&apellido2=apellido2&email=email&telefono=telefono&numCuenta=numerocuenta&observacion=observacion&fechaEntrada=fecha&fechaSalida=fecha>

Nota: Los archivos CreacionTablas e InsercionDatos que se adjuntan con el proyecto en la carpeta sql, contienen las sentencias necesarias para la creación de las tablas e inserción de los datos utilizados para la realización de los test.