Universidad de Alcalá - Escuela Politécnica

Departamento de Ciencias de la Computación

PROGRAMACIÓN ORIENTADA A OBJETOS

(Laboratorio de Prácticas)

Titulaciones de Grado en Ingeniería Informática, Ingeniería en Sistemas de Información e Ingeniería de Computadores.

EJERCICIO LABORATORIO (PL-2)

Una empresa nos ha encargado el diseño y realización de una aplicación informática llamada "**JavaBnB**" que permita el alquiler de inmuebles vacacionales.

Los elementos a tener en cuenta en la aplicación son los siguientes:

Usuarios de la aplicación.

La aplicación puede ser utilizada por dos tipos de usuarios:

1. Administrador de la aplicación: Es el responsable de la aplicación y, utilizará los siguientes datos para acceder: correo "*admin@javabnb.com*" y clave "*admin*". Podrá gestionar tanto a los usuarios como a los inmuebles.

2. Clientes: Para poder acceder a la aplicación, los clientes deberán estar registrados como usuarios. En el momento de acceder, la aplicación les permitirá registrarse o entrar. En caso de que sean usuarios registrados entrarán con su correo y su clave.

   Tendremos dos tipos de clientes, por un lado, el cliente particular y por otro los anfitriones de los inmuebles.

   Los datos que tendremos que guardar de los clientes particulares son los siguientes:
   a. DNI.
   b. Nombre.
   c. Correo electrónico.
   d. Clave.
   e. Teléfono.
   f. Tarjeta de crédito (nombre del titular, número de 16 dígitos y fecha de caducidad).
   g. VIP (clientes con descuentos).

Los anfitriones tendrán los siguientes datos:

a. DNI.
b. Nombre.
c. Correo electrónico.
d. Clave.
e. Teléfono.
de la a a la e son comunes
f. Fecha de registro en la aplicación.
g. Superanfitrión (si la media de las reseñas de sus inmuebles es superior a 4).

Inmuebles.

La aplicación gestionará una serie de inmuebles, que son los que alquilan los anfitriones.

De cada inmueble que tenemos registrado en la aplicación hay que guardar los siguientes datos:

- Título (breve descripción del inmueble que aparece como encabezado, ejemplo: "La Casona de la Playa").
- Dirección (calle, número, código postal y ciudad).
- Datos del inmueble, compuesto por el número de:
  - Huéspedes (número de huéspedes máximos permitidos)
  - Habitaciones
  - Camas
  - Baños
- Tipo de propiedad: Casa o Apartamento.
- Precio por noche.
- Servicios (listado de los servicios ofrecidos, por ejemplo: Wifi, Cocina, Aire acondicionado, Zona de trabajo, Aparcamiento, Piscina, Lavadora, etc.)
- Fotografía.
- Calificación (media de las reseñas sobre 5).

Alquiler de los inmuebles

Para poder alquilar un inmueble, los usuarios podrán utilizar una herramienta de búsqueda. El usuario tiene que introducir la ciudad donde quiere alojarse, la fecha de entrada y la de salida.

Para presentar el listado de los inmuebles, el usuario puede seleccionar distintas opciones de ordenación:

- Por precio (ordenados de menor a mayor).
- Tipo: Casa o Apartamento.
- Por relevancia (se muestran los inmuebles con la mejor calificación).

A continuación, el cliente seleccionará un inmueble y se le presentará toda su información. Una vez seleccionado, si en las fechas seleccionadas está libre, se procederá al pago de la reserva, cargando en su tarjeta de crédito el importe total de la estancia (número de días * precio por noche). Los clientes VIP tendrán un 10% de descuento.

Una vez confirmado el pago, la aplicación guardará la reserva realizando automáticamente las siguientes operaciones:

- Generar una factura de reserva en un fichero de texto: Deberá incluir la fecha de la reserva, el importe, los datos del inmueble, los datos del cliente y la fecha de entrada y salida previstas.
- Almacenar los datos de la reserva (los que contiene la factura) en la aplicación para futuras consultas.

Funcionamiento de la aplicación

Se debe realizar una aplicación gráfica con Swing que permita, una vez seleccionado el acceso como administrador o cliente, realizar las siguientes operaciones:

1. En caso de acceder como administrador las opciones disponibles serán:
   1.1. Consulta de usuarios: Se mostrará un listado de los usuarios de la aplicación pudiendo visualizar su información y mostrando claramente si son clientes particulares o anfitriones.
   1.2. Consulta de inmuebles: Se mostrará un listado de los inmuebles pudiendo ver su información.
   1.3. Consulta de reservas realizadas: Se mostrarán las reservas ordenadas por fecha. *Como opción voluntaria se pueden seleccionar todas las reservas o solo a partir de una fecha dada (la fecha en la que se hizo la reserva, no las fechas de entrada y salida).*

2. En caso de acceder como cliente particular las opciones disponibles serán:
   2.1 Registro del cliente, si ya está registrado podrá hacer el login con su correo y clave.
   2.2 Búsqueda de inmuebles.
   2.3 Reserva de inmuebles.
   2.4 Reseñas sobre los inmuebles que ha alquilado, dando una nota del 1 al 5. *Como opción voluntaria se puede acompañar de un comentario.*
   2.5 Consulta de sus reservas.
   2.6 Modificación de sus datos personales.

3 En el caso de que el cliente sea un anfitrión:
   3.1 Crear nuevos inmuebles.
   3.2 Modificar los datos de sus inmuebles o darlos de baja.
   3.3 Consultar las reservas de sus inmuebles.
   3.4 Modificación de sus datos personales.

Cada vez que se arranque la aplicación, tendrá disponibles todos los datos almacenados desde la última vez que se utilizó. Del mismo modo, al terminar la sesión, deberán guardarse todos los datos actualizados. Se utilizarán para ello los archivos que sean precisos.

Requisitos

La aplicación estará escrita en lenguaje Java, compatible con Java SE. No deberán utilizarse clases o métodos obsoletos ("*deprecated*")[1].

La aplicación deberá realizar un correcto tratamiento de las **excepciones** (por ejemplo, tratar de comprar una cantidad negativa de comidas) y presentar una **interfaz gráfica** de usuario, intuitiva y fácil de utilizar.

Las clases del dominio de la aplicación deberán estar en un **paquete diferente** a las de la gestión de la interfaz de usuario. **El código de la interfaz de usuario y de las clases del modelo debe estar estrictamente separado**, de manera que las clases del modelo puedan reutilizarse en el futuro en otros contextos, por ejemplo, si se quiere hacer una versión Web o móvil de la interfaz.

Es obligatoria la utilización de clases, estructuras de datos, persistencia y serialización en ficheros para el almacenamiento de los datos de la aplicación, estando prohibido el uso de bases de datos u otras formas de almacenamiento.

En todas las ventanas deberán visualizarse el anagrama "JavaBnB" (se admiten ideas para su diseño).

Para poder comprobar el buen funcionamiento de la aplicación, se deberá poder arrancar con un mínimo de **4 usuarios (al menos 2 clientes particulares y 2 anfitriones), 6 inmuebles y varias reservas** ya dadas de alta, con toda su información asociada.

Los programas fuente deberán ser fáciles de mantener por cualquier programador experto en Java. Deberán estar bien estructurados y contendrán los comentarios necesarios para un fácil seguimiento del código.

Mediante técnicas de ingeniería inversa UML deberá obtenerse el diagrama de clases de la aplicación que se incluirá en la documentación.

Los requerimientos recogidos en el presente documento deben entenderse como los mínimos que debe cumplir el sistema, pudiendo ser presentadas cuantas mejoras se crean convenientes.

[1] https://docs.oracle.com/en/java/javase/21/docs/api/deprecated-list.html

Entregables para Blackboard

Documento explicativo

La documentación de la aplicación debe redactarse para ser leída por un hipotético técnico informático que tuviese que evaluar la calidad de la aplicación desarrollada. No incluirá código fuente. Los apartados que debe incluir son:

- **Portada.** En la portada de la memoria, deberá aparecer:
  Grado en _____". Curso 2023/24
  Práctica de POO 2024.
  Autor: DNI – Apellidos, Nombre (o autores)

- **Especificación de requisitos.** Mediante *diagramas de casos de uso* se deben describir las acciones principales que los usuarios de la aplicación pueden realizar.

- **Análisis y Diseño de la aplicación.** Descripción del análisis y diseño de cómo se ha desarrollado la aplicación y qué elementos principales intervienen en la misma. Para realizarlo se construye un *Diagrama de Clases* sencillo, identificando las clases de la lógica de negocio.

  Generalmente las herramientas de modelado UML permiten seleccionar los elementos y copiarlos y pegarlos directamente.
  - Para hacer diagramas UML se puede usar *Enterprise Architect Free Trial* (https://sparxsystems.com/products/ea/trial/request.html) es tan sencillo como seleccionar el menú | *Publish* - *Save*.
  - También se puede utilizar Modelio: https://www.modelio.org/
  - Para hacer diagramas UML online se puede usar: https://app.diagrams.net/

- **Implementación (JavaDoc[2]).** Para completar este apartado se debe haber realizado una correcta documentación de la aplicación donde se han detallado sus clases principales (clases de lógica de negocio) y su funcionalidad y las estructuras de datos utilizadas para almacenar la información que maneja la aplicación. Esto se puede realizar mediante los comentarios del código fuente para la generación de la documentación en *JavaDoc*.

- **Manual de Usuario (descripción del funcionamiento).** En este apartado se explicará cómo interactúa un usuario con la aplicación. Es conveniente capturar las pantallas de la aplicación (se hace con las teclas *Alt + Impr Pant* y luego se pegan en el documento) y mostrar el funcionamiento de las distintas opciones, mensajes de error, etc.

[2] https://docs.oracle.com/en/java/javase/21/docs/api/index.html

Proyecto Maven

Acompañando a la memoria del proyecto, se incluirá un proyecto NetBeans (o de otro entorno de desarrollo) con Maven listo para arrancar la aplicación, para comprobar que se cumplen las especificaciones formuladas. Se comprimirá en un **fichero** indicando el **primer apellido de cada miembro del grupo y el curso 23_24**.
Formato del fichero: PL2_Apellido1_Apellido2_23_24.rar

Consideraciones

- La aplicación se puede hacer en **grupo de dos alumnos o de manera individual**.
- La evaluación de la práctica se llevará a cabo mediante una **entrevista** en la que el alumno deberá mostrar el funcionamiento de su aplicación y someterse a las preguntas que se consideren necesarias. Las pruebas se podrán realizar en el ordenador portátil del alumno.
- Con el objetivo de verificar la autoría de la práctica, el profesor podrá solicitar la inclusión de algún elemento de código no recogido en el presente enunciado o la modificación de algún aspecto de la aplicación. Los alumnos que no sepan introducir adecuadamente las modificaciones solicitadas serán evaluados como SUSPENSO, independientemente del contenido y calidad de la práctica entregada.

La entrega de la práctica consistirá en el proyecto Maven junto con la memoria explicativa en un fichero comprimido. Se realizará a través de la sección Trabajos de Blackboard. En el caso de que la práctica se realice en grupos de dos alumnos, solo es necesario que lo envíe uno, indicando los nombres de los integrantes del grupo.

La fecha límite de entrega será el 28 de mayo. A partir de esta fecha cada profesor especificará la fecha para la defensa de la práctica.