Diseño de la aplicación UCPark

Si bien, en el archivo de entrega se han adjuntado los diagramas correspondientes al diseño arquitectónico y al diseño detallado, se especifican a continuación algunas indicaciones que pueden resultar de cierta aclaración con respecto al diseño elaborado para la aplicación.

DISEÑO ARQUITECTÓNICO:

- Interfaces Presentación – Negocio: Se ha buscado realizar un diseño lo más reutilizable posible, para lo cual, se ha dividido en el mayor número de interfaces posibles las operaciones de interés, siguiendo un criterio en el que principalmente se ha observado lo que requeriría cada tipo de usuario/vista, obteniéndose las siguientes interfaces:
  En el ámbito de las denuncias:
  - IDenunciasAgentes
  - IDenunciasUsuarios
  - IConsultaDenuncias: En casos donde las operaciones pueden resultar de utilidad para ambos usuarios, se ha procedido a realizar este tipo de agrupamiento.

  En el ámbito de los estacionamientos:
  - IConsultaEstacionamientos: De nuevo compartida por agentes y usuarios.
  - IGestionEstacionamientos: En este caso enfocada prácticamente para los usuarios.

  En el ámbito de los vehículos:

  - IGestionVehiculos
  - IConsultaVehiculos
    Ambas diseñadas prácticamente para el usuario, pero separadas las operaciones en dos interfaces, para seguir un diseño relativamente estandarizado.

  Finalmente, en el ámbito del usuario:

  - IGestionUsuarios: Limitada en base a las especificaciones actuales, prácticamente a operaciones de registro y login.

- Interfaces Negocio – Persistencia: Las interfaces generadas se corresponden prácticamente con las DAO's a implementar, de forma que puedan especificarse los métodos necesarios para persistir o acceder a los datos persistidos. Es decir, existen las interfaces: IDenunciasDAO, IEstacionamientosDAO, IVehiculosDAO, IUsuariosDAO.

  Estas podrán ser empleadas por varios componentes al mismo tiempo, ya que algunas de las operaciones requieren el acceso a más de un tipo de dato.
  Pueden observarse las operaciones concretas en los diagramas adjuntos en la entrega.

En cuanto a los componentes, tal y como puede comprobarse en el diagrama, en el caso de la presentación existe uno por cada usuario principal (usuario conductor y agente), además de uno que se ha especificado aparte, con función para registros e inicios de sesión. Por último, de cara a la capa de negocio y a la de persistencia, la creación de los módulos está prácticamente ligada a la gestión de las principales entidades mencionadas anteriormente.

DISEÑO DETALLADO (Clases):

De igual forma que con la arquitectura, se adjunta el diagrama correspondiente en la entrega del proyecto. A continuación, se realizan algunas breves apreciaciones:

- En el caso de los usuarios y los vehículos, no se han añadido claves correspondientes a secuencias de números enteros para la identificación, pues teniendo en cuenta el contexto de la aplicación, el email de los usuarios y la matrícula de los vehículos se han considerado como algo suficiente para su identificación unívoca.

- Debido a la libertad que el enunciado y la práctica ofrecen, se ha construido la aplicación tomando como referencia que los vehículos pertenecen a un solo usuario. Esto puede variar en función de la interpretación que se realice.

- Se ha interpretado que todos los medios de pago futuros, además de las tarjetas, requerirán un id designado automáticamente con su registro en la aplicación.

- Con el fin de facilitar varias de las operaciones de negocio, se han garantizado repetidas relaciones bidireccionales entre las clases, con el fin de facilitar la futura programación y disminuir los accesos a las DAO's.

Plan de pruebas de la aplicación UCPark

Los niveles de prueba que se van a aplicar son los siguientes:

- Pruebas de aceptación. Las pruebas de aceptación se definirán siguiendo una estrategia basada en casos de uso y se ejecutarán de forma automatizada empleando Selenium IDE.
- Pruebas de integración. La estrategia para la definición del orden de las pruebas de integración será jerárquica. Se probará:
    - La integración entre la capa de negocio y la de persistencia. En este caso, para la definición de los casos de prueba se utilizarán técnica de métodos y caja negra y se utilizará JUnit.
    - La integración entre las tres capas. En este caso, para la definición de los casos de prueba se aplicarán técnicas de casos de uso y se utilizarán Junit y Selenium IDE.
- Pruebas unitarias. Se utilizará la técnica de prueba de métodos, usando técnicas de caja negra (partición equivalente y AVL) para la definición de los casos de prueba de cada método de cada clase o componente. Será necesaria la utilización de JUnit, Mockito y FEST.

A continuación, se muestra una especificación detallada de los casos de prueba a aplicar en cada nivel mencionado anteriormente.

PRUEBAS DE ACEPTACIÓN

En base a los casos de uso se identifican los siguientes escenarios:

A1. CU: Registrarse
- a. Registro válido (nuevo usuario)
- b. Registro no válido (usuario ya registrado con el email indicado)

A2. CU: Registrar vehículo
- a. Registro válido (nuevo vehículo)
- b. Registro no válido (vehículo ya registrado con la matrícula indicada)

A3. CU: Eliminar vehículo
- a. Eliminación válida

A4. CU: Consultar
- a. Consulta denuncia
- b. Consulta estacionamiento en vigor
- c. Consulta histórico de estacionamientos

A5. CU: Nuevo estacionamiento
- a. Creación válida:

  a1. El usuario realiza una pulsación de selección (un click) sobre la opción “Nuevo estacionamiento”.

a2. La aplicación muestra una nueva ventana, con un formulario compuesto de un desplegable y un campo de texto, en los que poder especificar el vehículo y la duración del estacionamiento.

a3. El usuario realiza una selección (un click) sobre el desplegable correspondiente a sus vehículos registrados.

a4. La aplicación muestra una lista con las matrículas registradas a nombre del usuario.

a5. El usuario selecciona el vehículo de interés para la creación del estacionamiento.

a6. La aplicación recoge la lista, mostrando la selección del usuario.

a7. El usuario introduce una duración de estacionamiento, por debajo de los 120 minutos.

a8. El usuario realiza una selección (un click) sobre "Crear estacionamiento".

a9. Se verifica que el vehículo no posee un estacionamiento en vigor.

a10. Se verifica que la duración introducida no supera los 120 minutos. Si supera el valor o es negativo, avisa al usuario para que modifique el valor.

a11. La aplicación realiza la transacción bancaria correspondiente al cobro del estacionamiento.

a12. Se verifica que los datos del estacionamiento han sido registrados.

a13. La aplicación muestra los datos del nuevo estacionamiento al usuario.

b. Creación no válida (estacionamiento vigente para vehículo seleccionado):

b1. El usuario realiza una pulsación de selección (un click) sobre la opción "Nuevo estacionamiento".

b2. La aplicación muestra una nueva ventana, con un formulario compuesto de un desplegable y un campo de texto, en los que poder especificar el vehículo y la duración del estacionamiento.

b3. El usuario realiza una selección (un click) sobre el desplegable correspondiente a sus vehículos registrados.

b4. La aplicación muestra una lista con las matrículas registradas a nombre del usuario.

b5. El usuario selecciona el vehículo de interés para la creación del estacionamiento, el cual ya posee un estacionamiento en vigor.

b6. La aplicación recoge la lista, mostrando la selección del usuario.

b7. El usuario introduce una duración de estacionamiento, por debajo de los 120 minutos.

b8. El usuario realiza una selección (un click) sobre "Crear estacionamiento".

b9. Se verifica que el vehículo ya dispone de un estacionamiento en vigor.

b10. La aplicación muestra una notificación de error al usuario, indicando que el vehículo seleccionado ya dispone de un estacionamiento vigente.

c. Creación no válida (no se puede realizar el cobro):

c1. El usuario realiza una pulsación de selección (un click) sobre la opción "Crear un nuevo estacionamiento".

c2. La aplicación muestra una nueva ventana, con un formulario compuesto de un desplegable y un campo de texto, en los que poder especificar el vehículo y la duración del estacionamiento.

c3. El usuario realiza una selección (un click) sobre el desplegable correspondiente a sus vehículos registrados.

c4. La aplicación muestra una lista con las matrículas registradas a nombre del usuario.
c5. El usuario selecciona el vehículo de interés para la creación del estacionamiento.
c6. La aplicación recoge la lista, mostrando la selección del usuario.
c7. El usuario introduce una duración de estacionamiento, por debajo de los 120 minutos.
c8. El usuario realiza una selección (un click) sobre "Crear estacionamiento".
c9. Se verifica que el vehículo no posee un estacionamiento en vigor.
c10. Se verifica que la duración introducida no supera los 120 minutos.
c11. La aplicación intenta realizar la transacción bancaria correspondiente al cobro del estacionamiento, sin éxito.
c12. La aplicación muestra una notificación de error al usuario, indicando que ha ocurrido un error a la hora de realizar el pago, manteniendo cargado el formulario con los datos introducidos por el usuario.

A6. CU: Ampliar tiempo estacionamiento
a. Ampliación válida
b. Ampliación no válida (excede 120 minutos globales)
c. Ampliación no válida (no se puede realizar el cobro)

A7. CU: Finalizar estacionamiento
a. Finalización válida

A8. CU: Comprobar estacionamiento
a. Comprobación válida

A9. CU: Denunciar
a. Denuncia válida

A10. CU: Anular denuncia
a. Anulación válida
b. Anulación no válida (no se puede realizar el cobro)

Los casos de prueba definidos para cada uno de estos escenarios, partiendo de un sistema sin información registrada, son los que se muestran en la Tabla 1. Los casos de prueba deberían ser ejecutados en el orden indicado, para que el estado final del sistema sea igual que al comienzo.

Dado que solo se solicita la especificación de las pruebas correspondientes a la creación de un estacionamiento, se indican únicamente los datos necesarios para llevar a cabo dichas pruebas (casos A1 y A2).

Tabla 1. Casos de prueba de aceptación

Identificador	Entrada	Resultado
A1.a	Email: mmp819@email.com Contraseña: 1234 Tarjeta: 0000111122223333, 123, Mario M.	Registro válido

A2.a	Usuario: mmp819@email.com Vehículo: 1234ABC, Renault, Megane	Registro válido
A5.c	Usuario: mmp819@email.com Vehículo: 1234ABC Duración: 60 minutos *Sin conexión a internet*	Creación no válida (Error al cobrar)
A5.a	Usuario: mmp819@email.com Vehículo: 1234ABC Duración: 60 minutos	Creación válida
A5.b	Usuario: mmp819@email.com Vehículo: 1234ABC Duración: 60 minutos *Antes de que pasen 60 minutos del caso de aceptación A5.a*	Creación no válida (Estacionamiento vigente)

PRUEBAS DE INTEGRACIÓN

El orden de las pruebas y los casos de prueba a realizar serían los siguientes:

1. GestionEstacionamientos con UsuariosDAO, VehiculosDAO y EstacionamientosDAO. Se usarían los mismos casos de prueba definidos como UGIC.x en la sección de pruebas unitarias, aquí renombrados como IGIC.x.
2. VistaUsuarioLoggeado con GestionEstacionamientos, UsuariosDAO, VehiculosDAO y EstacionamientosDAO. Estas pruebas coincidirían con las pruebas de aceptación, aunque en este caso se automatizarían utilizando Selenium IDE.
   Podría hacerse una prueba previa de integración VistaUsuarioLoggeado y GestionEstacionamientos usando objetos Mock de los componentes DAO, pero no se considera necesario.

PRUEBAS UNITARIAS

Pruebas unitarias de la capa de persistencia (componente: DatosEstacionamientos)

Se aplica prueba de métodos, siendo los casos de prueba definidos para cada método los que se exponen a continuación y seguidos en el orden especificado. Los casos expuestos para cada método suponen como punto de partida un sistema en el que teóricamente existe ya un vehículo registrado con matrícula 5678DEF y un estacionamiento con las mismas características que el especificado para la prueba UCD.1a, variando únicamente la hora de inicio por la fijada en su creación, el vehículo al que se asigna, que es el especificado previamente y su ID (1).

- Método creaEstacionamiento

Identificador	Entrada	Valor esperado
UCD.1a	Vehículo: 1234ABC Duración: 60 minutos Importe: 60 * 0.10 (Precio minuto)	Vehículo: 1234ABC Duración: 60 minutos Id: 2 (autogenerado)

	(Estacionamiento nuevo) Hora Inicio: LocalDateTime.now()	Importe: 60 * 0.10 Hora Inicio: La establecida en la creación
UCD.1b	Vehículo: 5678DEF Duración: 60 minutos Importe: 60 * 0.10 (Precio minuto) (Estacionamiento nuevo) Hora Inicio: LocalDateTime.now() Id: 1	Error, estacionamiento ya registrado.

- Método estacionamientos

Identificador	Entrada	Valor esperado
UCD.2a		Listado con 2 estacionamientos.
UCD.2b		Lista vacía (partiendo de un sistema sin estacionamientos)

- Método modificaEstacionamiento

Identificador	Entrada	Valor esperado
UCD.2ª	Vehículo: 1234ABC Duración: 90 minutos Importe: 90 * 0.10 (Precio minuto) (Estacionamiento nuevo) Hora Inicio: Establecida previamente Id: 2	Vehículo: 1234ABC Duración: 90 minutos Importe: 90 * 0.10 (Precio minuto) Hora Inicio: Establecida previamente Id: 2
UCD.2b	Id: 3 (Estacionamiento no existe)	null

- Método estacionamientoPorId

Identificador	Entrada	Valor esperado
UCD.3a	Id: 2	Vehículo: 1234ABC Duración: 90 minutos Importe: 90 * 0.10 (Precio minuto) Hora Inicio: Establecida previamente Id: 2
UCD.3b	Id: 3 (Estacionamiento no existe)	null

- Método eliminaEstacionamiento

Identificador	Entrada	Valor esperado
UCD.4a	Id: 2	Vehículo: 1234ABC Duración: 90 minutos Importe: 90 * 0.10 (Precio minuto) Hora Inicio: Establecida previamente

		Id: 2
UCD.4b	Id: 3 (Estacionamiento no existe)	null

(Similares casos de prueba para los métodos de la interfaz IEstacionamientosDAO).

Pruebas unitarias de la capa de negocio (componente: GestionEstacionamientos)

Para poder llevar a cabo estas pruebas, será necesario el uso de objetos Mock para las interfaces IVehiculosDAO, IEstacionamientosDAO. Se aplica prueba de métodos, siendo los casos de prueba definidos para cada método los siguientes y debiendo ser seguidos en orden.

* Siguiendo la especificación del enunciado, se han elaborado pruebas únicamente para el módulo *GestionEstacionamientos* y su interfaz IGestionEstacionamientos, de entre las dos interfaces que implementa (*IGestionEstacionamientos e IConsultaEstacionamientos)*. *

- Método creaEstacionamiento: Conceptualmente se trata de los mismos casos identificados para el caso de uso 5.

Identificador	Entrada	Valor esperado
UGIC.1a	Vehículo: 1234ABC Minutos: 60	Vehículo: 1234ABC Duración: 60 minutos Id: 1 (autogenerado) Importe: 60 * 0.10 Hora Inicio: LocalDateTime.now
UGIC.1b	Vehículo: 1234ABC Minutos: 60 *Después de insertar el anterior estacionamiento*	Vehículo con estacionamiento en vigor.
UGIC.1c	Inserción con error en acceso a pago (No implementable)	Error en cobro.

- Método ampliaEstacionamiento: Conceptualmente se trata de los mismos casos identificados para el caso de uso 6.

Identificador	Entrada	Valor esperado
UGIC.2a	Vehículo: 1234ABC Minutos: 30	Vehículo: 1234ABC Duración: 90 minutos Id: 1 Importe: 90 * 0.10 Hora Inicio: Establecida previamente
UGIC.2b	Vehículo: 1234ABC Minutos: 90	Error al sobrepasar los 120 minutos de tiempo global.
UGIC.2c	Ampliación con error en acceso a pago (No implementable)	Error en cobro.

Informe de pruebas – Pruebas implementadas

Componente: GestionEstacionamientos

A continuación, se adjuntan los errores encontrados a lo largo del desarrollo y ejecución de las pruebas relativas al método ampliaEstacionamiento.

- **ampliaEstacionamiento (Estacionamiento estacionamiento, int minutos):**
  - Error en la actualización del importe:
    La implementación original solo tenía en cuenta el incremento del tiempo de estacionamiento, pero no el incremento del importe total del mismo.

Componente: DatosEstacionamientos

A continuación, se adjuntan los errores encontrados a lo largo del desarrollo y ejecución de las pruebas relativas a los siguientes métodos:

- **creaEstacionamiento (Estacionamiento estacionamiento):**
  No se han encontrado errores.

- **modificaEstacionamiento (Estacionamiento estacionamiento):**
  No se han encontrado errores.

Componente: VistaUsuarioLogueado

Se ha detectado que faltaba la implementación de un método para poder cargar las matrículas de los vehículos registrados a nombre de un usuario, con el fin de poder otorgar datos al desplegable de matrículas.

Mario Martín Pérez