Ejercicios 6C.2

Proyecto

Vamos a ver un ejemplo de cómo se trabajaría en un proyecto REAL. Considera que el texto que aquí tienes es la **documentación** del Proyecto Venta de Productos que acaba de empezar y que te acabas de incorporar al proyecto. Te van a asignar una parte que tienes que programar por tu cuenta, por supuesto siguiendo toda la información contenida en el presente documento.

Es un ejercicio pequeño, no habrá Documento de Objetivos super-detallados ni nada parecido por supuesto, pero deberías de saber manejarte con esto.

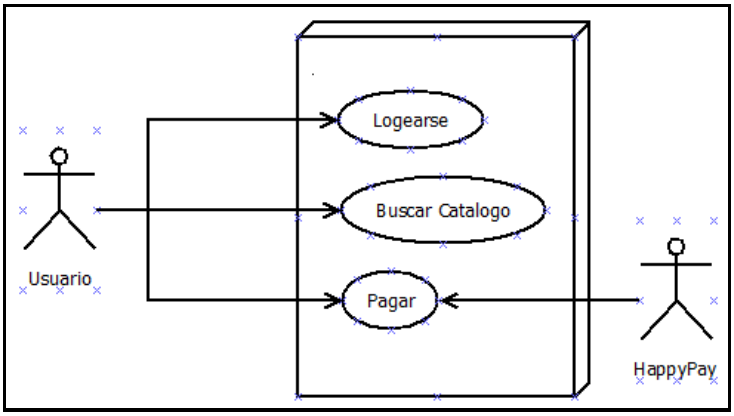
Contexto

La empresa **Ikea** ha decidido crear una web que permita a los clientes registrarse en la misma y acceder a un catálogo de productos. Los clientes podrán entonces consultar el catálogo y comprar los productos que deseen. Por supuesto una parte muy importante consiste en el pago, que se realizará a través de un sistema externo llamado *HappyPay* desarrollado por la propia Ikea y que ya funciona perfectamente.

Nuestra empresa ya ha presentado una maqueta de cómo sería la web y le ha encantado a Ikea, así que ahora nos toca a nosotros programarla. Nuestro **analista jefe** nos ha dicho que vamos a programar partiendo de esa maqueta ya hecha por lo que tan solo tenemos que ir añadiéndole cosas. Por supuesto, estamos hablando del ejercicio **Venta de Productos** hecho en clase.

Casos de Uso

Los analistas nos han remitido el siguiente esquema de Casos de Uso.



Casos de Uso

Junto con la siguiente documentación al respecto:

- **Actores**: Existen dos actores en nuestro proyecto.
    - **Usuario**: el actor principal. Cada uno de nuestros clientes.
    - **HappyPay**: el actor secundario, un sistema de pago de Ikea.
- **Escenario**: Se definen los siguientes escenarios.
    - **Logearse**: Este escenario comprende el proceso de login de un usuario en nuestro sistema. Un usuario deberá suministrarnos un usuario y un password, y si sus datos son correctos accederá a nuestro sitio.
    - **Buscar catálogo**: Este escenario corresponde a la búsqueda y muestra de todos los productos disponibles. Se definen dos sub-escenarios:
        - **Buscar todos**: muestra todos los productos
        - **Buscar por referencia**: búsqueda selectiva de un producto.
    - **Pagar**: Se contacta con HappyPay para realizar el pago. Nosotros no gestionamos esta parte, tan sólo damos un interfaz y utilizamos la librería HappyPay de Ikea. Se definen tres sub-escenarios para los métodos de pago.

AVISO: el Analista aún no ha definido los sub-casos

Diagramas de Actividad

Los analistas han establecido los siguientes Diagramas de Actividad para detallar las siguientes operaciones:

Login

El proceso de login…

AVISO: Pendiente de actualizar la documentación. El código ya está programado y funciona

Pagar

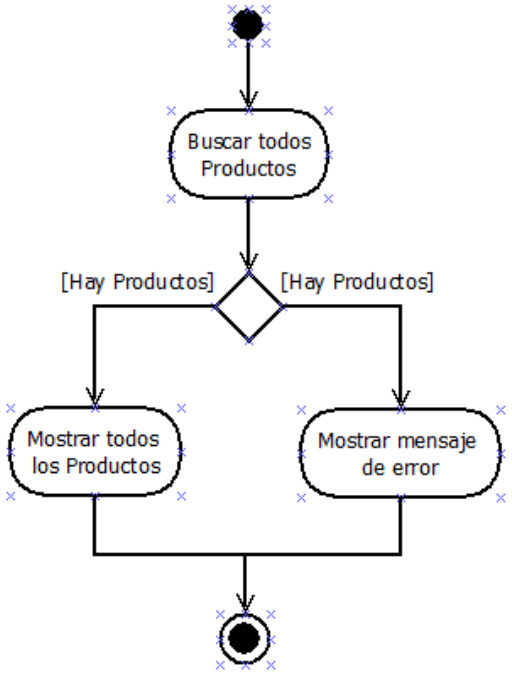
El proceso de Pago.

AVISO: Pendiente de especificar. Ikea tiene que pasarnos toda la información pertinente a la librería HappyPay_2.306.jar

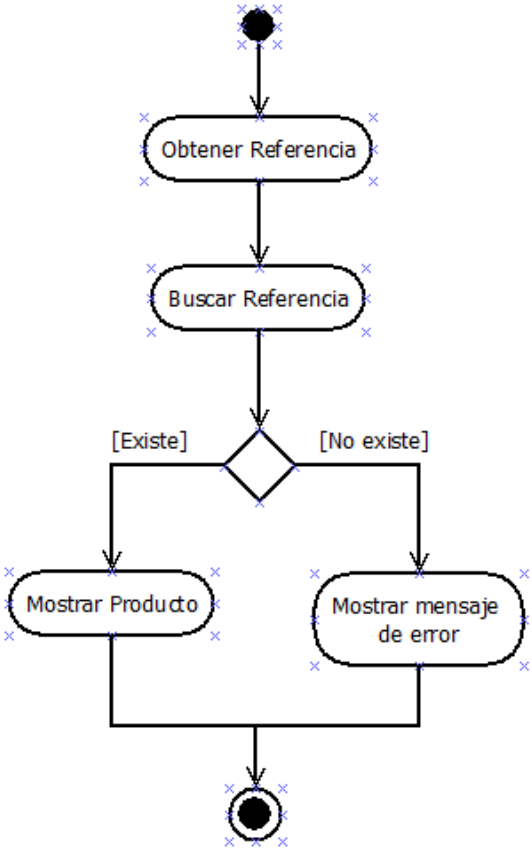
Buscar catálogo

Existen diagramas de actividad que especifican cada uno de los dos sub casos.

Buscar todos



Buscar referencia



Diagramas de paquetes

La estructura de los paquetes del proyecto debe de ser la siguiente:

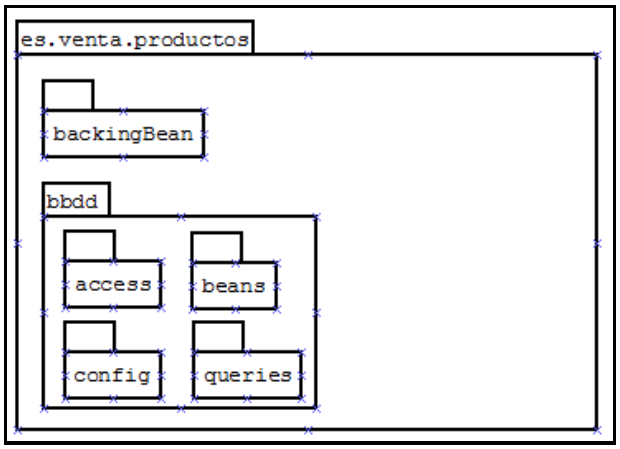


Diagrama de Paquetes

Descripción de los paquetes:

- **backingBean**: las clases del CONTROLADOR relacionadas con la toma de decisiones se encuentran aquí. Existirá una clase por cada operación pertinente. Todas las clases tienen el siguiente formato: **nombreOperación + BackingBean**. Ejemplo: *LoginBackingBean.java*.

- **bbdd**: las clases del MODELO relacionadas con el tratamiento de la información de Base de Datos están organizadas dentro de este paquete.

    - **access**: contiene las clases que poseen los métodos de acceso a BBDD. Cada clase de acceso corresponderá con una única tabla de BBDD, con el siguiente formato: **nombreTabla + Access**. Ejemplo: *UserAccess.java*.

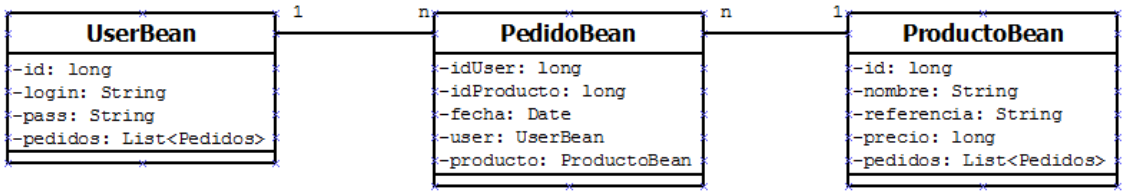
    - **beans**: contiene las clases que poseen los mapeos de las tablas de BBDD. Cada Bean se corresponderá con una única tabla, con el formato: **nombreTabla + Bean**. Ejemplo: *UserBean.java*.

    - **config**: contiene las clases de configuración de BBDD.

    - **queries**: contiene las clases con las sentencias SQL.

Diagramas de Clases

La estructura de clases del proyecto para la BBDD es la siguiente:



Diseño: Buscar Catálogo

Este documento explica los distintos elementos concernientes a los sub casos Buscar Catálogo.

VISTA

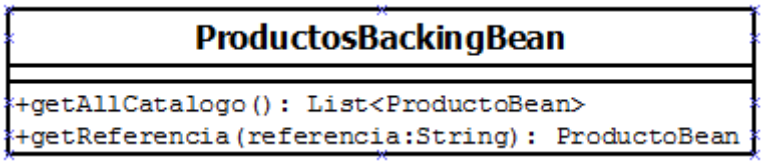
La vista costa de dos JSP localizados en la carpeta /jsp del proyecto. El nombre de los jsp son:

- mostrarCatalogo.jsp: para el sub caso **Buscar todos.**
- mostrarReferencia.jsp: para el sub caso Buscar por Referencia.

AVISO: Diseño preliminar e incompleto de la maqueta inicial. Revisar.

CONTROLADOR

La clase ProductosBackingBean.java se encargará de gestionar las búsquedas de productos. La clase dispondrá del siguiente esquema:

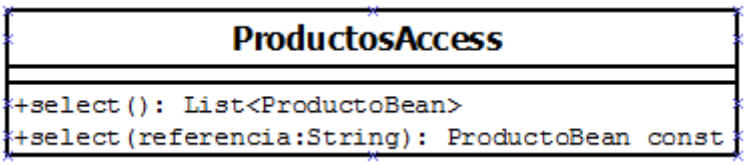


Descripción de los métodos:

- getAllCatalogo(): Retorna el catálogo de productos completo, o un nulo si no hay ningún producto en la BBDD.
- getReferencia(): Retorna el producto especificado o null si no existe.

MODELO

La clase ProductosAccess accede a la tabla correspondiente de Productos de BBDD.



- select(): Retorna el catálogo de productos completo, o un nulo si no hay ningún producto en la BBDD.

- select(referencia: String): Retorna el producto especificado o null si no existe.

En cuanto a los Beans de BBDD...

AVISO: Pendiente de completar. Consulta el apartado Diagrama de Clases.

¿Qué hay que hacer?

El alumno va a tener que realizar las siguientes actividades:

1) Añade el Javadoc a todas las clases y métodos del proyecto tan y como se ha comentado en clase.

2) Añade la funcionalidad log4J para la generación de trazas. Los criterios son los siguientes:

    a. Una traza ERROR para capturar los mensajes de error de las excepciones

    b. Una traza INFO al inicio y fin de cada método, así como cada vez que los métodos hagan 'algo' relevante.

3) Completa el proyecto para los dos sub casos de búsqueda de catálogo. Ten en cuenta qué

    a. Los JSP están completos, sólo les falta la parte java que va entre las etiquetas <% %> y es posible que las importaciones de <%@page import...

    b. Crea las clases ProductosAccess y ProductosBackingBean.

    c. Crea la estructura de los Beans de BBDD (completa).

4) Los métodos de ProductosAccess hazlos como quieras. Si devuelve una List, pues carga una List con lo que quieras etc...

5) Haz que funcione el proyecto, por supuesto