

Ejercicio Spring - 1

Se desea diseñar una aplicación para **gestionar los ingresos y gastos** de una serie de usuarios.

Para el proceso vamos a emplear las tablas **Usuarios, Conceptos y Movimientos** que tienen la siguiente estructura (la tabla Usuarios es la de siempre):

```
create table Conceptos (
    id integer identity(1,1) primary key,
    tipo char,
    descripcion varchar(60)
);

create table Operaciones(
    id integer identity(1,1) primary key,
    idConcepto integer references Conceptos,
    idCliente integer references Clientes,
    fecha date,
    importe numeric(8,2),
    comentarios varchar(255)
);

create table Clientes(
    id integer identity(1,1) primay key,
    nombre varchar(100),
    password varchar(100)
);
```

Se deberá emplear el modelo por **capas** e **Hibernate**. En la capa **DAO** se definirán los siguientes métodos (podéis añadir otros si lo consideráis necesario):

- **Cliente getCliente(String nombre)** : Localizará el cliente con el nombre indicado. Si **no existe**, se retornará un valor **null**
- **List<Concepto> getConceptos (String tipo)** : retornará los conceptos cuyo campo **tipo** coincida con el indicado **ordenados alfabéticamente**
- **Concepto getConcepto(int id)** : Retornará el concepto con el id indicado o un valor **null** si no existe dicho concepto.
- **void nuevoConcepto(Concepto c)** : Añadirá el concepto a la base de datos
- **void nuevaOperacion(Operación o)** : Añadirá la operación indicada a la BBDD
- **void eliminarOperacion(int id)** : Eliminará de la BBDD la operación con ese id

En la capa de servicio, además de los métodos indicados, se deberá definir un método **login(nombre, password)** que retornará el **Cliente** con ese nombre y contraseña o un valor **null** si el cliente no existe o su password no es correcta.

Cuando la aplicación se inicie deberá ir a la siguiente página (sin y con errores). NOTA: En el caso de tener errores, el **usuario debe permanecer y la password se pondrá en blanco**.

Introduzca sus datos

Usuario :	<input type="text"/>
Password :	<input type="password"/>
<input type="button" value="Entrar"/>	

Recorte rectangular

Introduzca sus datos

Usuario :	<input type="text" value="egomez"/>
Password :	<input type="password"/>
<input type="button" value="Entrar"/>	

Nombre o Password erróneos

Si el usuario es válido, se mostrará la siguiente página (sin y con operaciones).

NOTA: En el campo **fecha** se deberá pasar la fecha del día en formato **yyyy-MM-dd** para que se muestre correctamente en un campo HTML de tipo **date**

NOTA: Las operaciones deben aparecer ordenadas en **descendente** por fecha de operación.

Bienvenido egomez

No hay operaciones

Nueva Operación

Concepto [Nuevo Concepto](#)

Fecha

Importe

Comentarios

Bienvenido egomez

Listado de Operaciones

Fecha	Concepto	Tipo	Importe	Comentarios	
20-feb-2017	Comida	Gasto	Comida en Etxanobe	120.00 €	Eliminar
20-feb-2017	Electricidad	Gasto		85.00 €	Eliminar
20-feb-2017	Nómina / Pensión	Ingreso		1.750.00 €	Eliminar
17-feb-2017	Intereses Financieros	Ingreso	Intereses Fondo Maxx	120.00 €	Eliminar
Total				1.665.00 €	

Nueva Operación

Concepto [Nuevo Concepto](#)

Fecha

Importe

Comentarios

El botón **Guardar** deberá comprobar que el **importe exista y sea MAYOR QUE 0** y dar un mensaje de error de no ser así (emplear **validaciones**). Si el importe cumple, se almacenará la operación en la base de datos y se volverá a la misma página (el listado de operaciones deberá mostrar la nueva operación).

El botón **Volver** volverá a la página de login.

El botón **Eliminar** eliminará la operación seleccionada y volverá a la misma página actualizada,

El botón **Nuevo Concepto** irá a la siguiente página:

Introduzca los datos

Descripción	<input type="text"/>
Tipo	<input type="text" value="Ingreso"/>
<input type="button" value="Guardar"/> <input type="button" value="Volver"/>	

Si no se especifica la **descripción** se deberá dar un mensaje de error. Si se ha indicado, se **añadirá** el concepto a la BBDD y se retornará a la página anterior que deberá mostrar la información actualizada.

El botón **Volver** simplemente retornará a la página anterior.

```

create table Clientes (
    id integer identity(1,1) primary key,
    nombre varchar(100),
    password varchar(100)
);

insert into Clientes(nombre,password) values('jmiguel','jmiguel');
insert into Clientes(nombre,password) values('egomez','egomez');
insert into Clientes(nombre,password) values('cperez','cperez');

create table Conceptos (
    id integer identity(1,1) primary key,
    tipo char,
    descripcion varchar(60)
);

insert into Conceptos(tipo,descripcion) values('I','Nómina / Pensión');
insert into Conceptos(tipo,descripcion) values('I','Intereses Financieros');
insert into Conceptos(tipo,descripcion) values('G','Hipoteca');
insert into Conceptos(tipo,descripcion) values('G','Gas Natural');
insert into Conceptos(tipo,descripcion) values('G','Electricidad');
insert into Conceptos(tipo,descripcion) values('G','Comida');
insert into Conceptos(tipo,descripcion) values('G','Ocio y Tiempo Libre');
insert into Conceptos(tipo,descripcion) values('G','Tarjeta');

create table Operaciones(
    id integer identity(1,1) primary key,
    idConcepto integer references Conceptos,
    idCliente integer references Clientes,
    fecha date,
    importe numeric(8,2),
    comentarios varchar(255)
);

```