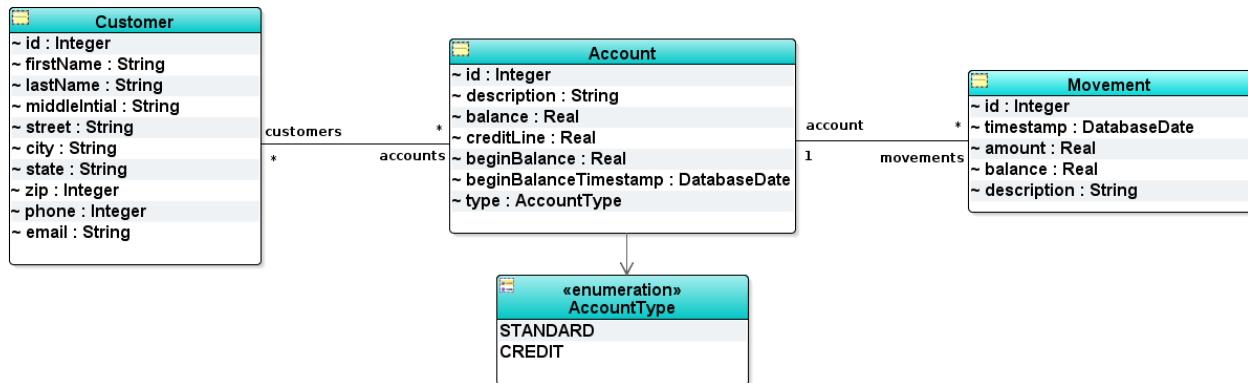


# EXAMEN 1ª FINAL DE DESARROLLO DE INTERFACES

## CURSO 2021/22

### Planteamiento

Una aplicación de gestión bancaria dispone de una aplicación web ya desplegada y accesible en la URL <http://hz059359:8080/CRUDBankServerSide/webresources/> que expone tres servicios RESTful accesibles en los paths **customer**, **account** y **movement**. Estos servicios proporcionan operaciones CRUD sobre el siguiente modelo de datos:



Se proporciona la aplicación web anterior empaquetada en el archivo *CRUDBankServerSide.war* que incluye el código fuente, para su consulta.

Se ha desarrollado así mismo parte de una aplicación cliente con UI de escritorio. Las partes desarrolladas de esta aplicación son el modelo de datos, la capa de lógica y el cliente que permiten explotar el servicio RESTful customer de la aplicación web del lado servidor. Dicho código se proporciona en el archivo *CRUDBankClientLibrary.jar* que se debe usar como librería.

**Se pide** desarrollar una aplicación Java FX que implemente actúe como interfaz de usuario (UI) mostrando la siguiente ventana titulada "Customer's accounts and balances":

The screenshot shows a Java FX window titled "Customer's accounts and balances". It features a search bar for "Customer ID" with the value "102263301" and a "Search" button. Below the search bar is a table titled "Accounts" with columns: "Account number", "Type", "Description", and "Balance". The table contains five rows of data, with the last row highlighted in blue. At the bottom of the window, there is a "Total Balance" label with the value "-75799.01" and an "Exit" button.

Account number	Type	Description	Balance
2074945725	STANDARD	Savings Account	100.0
5241087376	CREDIT	New Credit Account	9000.0
5255214522	CREDIT	Check Account with Credit Line	4099.0
1569874954	STANDARD	Savings Account	999.99
2654785441	STANDARD	Check Account	-89998.0

Total Balance: -75799.01

### Descripción:

Esta ventana muestra, para un cliente dado, un listado con sus cuentas y sus saldos. En un campo de saldo total muestra la suma de los saldos de todas sus cuentas

### Comportamiento:

CONTROL	ACCIÓN	RESPUESTA
Vista	Inicializar (initStage)	<ul style="list-style-type: none"><li>• Establecer el título de la ventana con el valor <b>"Customer's accounts and balances"</b>.</li><li>• La ventana no debe ser redimensionable.</li><li>• Los valores de las celdas de las columnas de la tabla se corresponderán respectivamente con los campos <b>id</b>, <b>type</b>, <b>description</b> y <b>balance</b> de la entidad <b>Account</b> del modelo de datos.</li><li>• El valor de las columnas <b>Account number</b>, <b>balance</b> y del campo <b>Total Balance</b> debe estar alineado a la derecha.</li><li>• El campo <b>Total Balance</b> estará deshabilitado.</li><li>• El foco debe estar en el campo <b>Customer ID</b>.</li><li>• Mostrar la ventana.</li></ul>
Campo de texto <i>CustomerID</i>	Cambiar texto (textPropertyChange)	<ul style="list-style-type: none"><li>• Vaciar el contenido de la tabla de cuentas y del campo <b>Total Balance</b>.</li></ul>
Botón <i>Search</i>	Pulsar (ActionEvent)	<ul style="list-style-type: none"><li>• Validar que el campo <i>CustomerID</i> contiene un valor <b>java.lang.Long</b> positivo :<ul style="list-style-type: none"><li>◦ Si no es válido crear y lanzar una excepción con el mensaje de error correspondiente.</li><li>◦ Si es válido, llamar al método <b>getCustomerAccountsFullInfo</b> de la interfaz <b>CustomerManager</b> para obtener un objeto <b>Customer</b> con los datos del cliente. Dicho objeto contendrá una lista con los datos de las cuentas del cliente. Si dicha lista es nula, lanzar una excepción con un mensaje que indique que el cliente no tiene cuentas. Si dicha lista no es nula, cargar la tabla con los datos de las cuentas del usuario y mostrar en el campo <b>Total Balance</b> la suma de los saldos de todas las cuentas del cliente.</li><li>◦ Mostrar los datos del cliente en los campos correspondientes de la ventana.</li></ul></li><li>• Si se produce alguna excepción atraparla y mostrar un mensaje de alerta con el mensaje de la excepción y enfocar el campo <i>CustomerID</i>.</li></ul>

CONTROL	ACCIÓN	RESPUESTA
Botón <i>Exit</i>	Pulsar	<ul style="list-style-type: none"> <li>• Pedir confirmación al usuario para salir: <ul style="list-style-type: none"> <li>◦ Si el usuario confirma, salir de la aplicación.</li> <li>◦ Si no confirma, mantenerse en la ventana.</li> </ul> </li> </ul>
Ventana	Cerrar	

### **Clase Aplicación**

La clase aplicación deberá realizar como mínimo las siguientes acciones (requisitos):

- Obtener mediante la factoría ***clientside.controller.CustomerManagerFactory*** un objeto que implemente la interfaz ***clientside.controller.CustomerManager***.
- Establecer el nombre de servidor que usará su aplicación al valor ***hz059359.tartangalh.eus*** . Este nombre se leerá de un archivo de propiedades y se establecerá con el método ***setServerName*** de la interfaz ***CustomerManager***.
- Cargar la vista de la ventana “Customer’s accounts and balances”.
- Pasar al controlador de la vista los datos y objetos necesarios para su funcionamiento.
- Llamar al método de inicialización del controlador de la vista.

### **Entrega**

Debe entregar el archivo .jar que contiene la aplicación que ha desarrollado y que demuestre su funcionalidad. Este archivo debe contener el código fuente de todas las clases desarrolladas. **Si el archivo no contiene el código fuente el examen se calificará con 0 puntos.**

### **Criterios de calificación**

- Generación de la vista XML de acuerdo a especificación= **2 puntos**.
- Clase aplicación funcional de acuerdo a requisitos = **1 punto**
  - por cada punto no implementado **-0.5 puntos**
- Ventana “Customer’s accounts and balances” funcional de acuerdo a requisitos de aspecto y comportamiento= **5 puntos**
  - por cada accion-respuesta no implementada **-2 puntos**
  - por cada punto de una respuesta no implementado **-0.5 puntos**
- Si la aplicación cliente implementada no está integrada con el lado servidor **-5 puntos**.
- Si la aplicación cliente implementada es funcional e integrada con el lado servidor:
  - La UI implementada incorpora la gestión de excepciones apropiada **1 punto**.
  - La UI hace uso apropiado de un log o bitacora: **1 punto**.

**NOTA MÍNIMA: 0 puntos.** **NOTA MÁXIMA: 10 puntos.** **NOTA PARA APROBAR: 5 puntos.**