

ESCOM
Escuela Superior de Cómputo

DESARROLLO DE SISTEMAS DISTRIBUIDOS

TAREA #17

DAO de Empresa de Envío

24 de Octubre del 2018

PROFESOR: ENRIQUE ZÁRATE JOSÉ ASUNCIÓN

ALUMNO: SALDAÑA AGUILAR ANDRÉS

GRUPO: 4CM3

Cómo funciona DAO?

Como dijimos antes, DAO encapsula el acceso a la base de datos. Por lo que cuando la capa de lógica de negocio necesite interactuar con la base de datos, va a hacerlo a través de la API que le ofrece DAO. Generalmente esta API consiste en métodos CRUD (Create, Read, Update y Delete). Entonces por ejemplo cuando la capa de lógica de negocio necesite guardar un dato en la base de datos, va a llamar a un método `create()`. Lo que haga este método, es problema de DAO y depende de como DAO implemente el método `create()`, puede que lo implemente de manera que los datos se almacenen en una base de datos relacional como puede que lo implemente de manera que los datos se almacenen en ficheros de texto. Lo importante es que la capa de lógica de negocio no tiene porque saberlo, lo único que sabe es que el método `create()` va a guardar los datos, así como el método `delete()` va a eliminarlos, el método `update()` actualizarlos, etc. Pero no tiene idea de como interactúa DAO con la base de datos.

En una aplicación, hay tantos DAOs como modelos. Es decir, en una base de datos relacional, por cada tabla, habría un DAO.

DAO consiste básicamente en una clase que es la que interactúa con la base de datos. Los métodos de esta clase dependen de la aplicación y de lo que queramos hacer. Pero generalmente se implementan los métodos CRUD para realizar las "4 operaciones básicas" de una base de datos.

Bien, nos falta comprender algo más para poder empezar a ver código. Los DTO (Data Transfer Object) o también denominados VO (Value Object). Son utilizados por DAO para transportar los datos desde la base de datos hacia la capa de lógica de negocio y viceversa. Por ejemplo, cuando la capa de lógica de negocio llama al método `create()`, ¿qué es lo que hace DAO? inserta un nuevo dato... ¿pero qué dato? el que la capa de lógica de negocio le pase como parámetro... ¿y cómo se lo pasa este dato? bueno, a través de un DTO.

Podría decirse que un DTO es un objeto común y corriente, que tiene como atributos los datos del modelo, con sus correspondientes `accessors` (getters y setters).

Por ejemplo, si tuviéramos una base de datos relacional con una tabla `employers`, con los campos `id`, `name` y `salary`. Entonces tendríamos que crear una clase `EmployerDTO`, con los atributos `id`, `name` y `salary`, que van a utilizar la capa de negocio y de persistencia para transportar los datos entre las dos capas.

Entonces cuando la capa de lógica de negocio quiera guardar un dato en la base de datos, va a crear un objeto `EmployerDTO`, a través de los `accessors` va a modificar los atributos, y después se lo va a pasar al método `create()` de DAO. Entonces DAO va a leer los datos del DTO, y los va a guardar en la base de datos. Lo mismo pasaría para eliminar datos. Y para actualizarlos además se le pasaría el ID, para saber que dato actualizar. Para buscar datos, sería parecido, ya que se le pasa al método `read()` el DTO para usarlo como patrón de búsqueda, pero con la diferencia de que este método tiene valor de retorno, ya que devuelve otro DTO con los datos del resultado de la búsqueda.

Desarrollo

En grandes rasgos podemos separar la aplicación en un cliente y servidor con sockets:

Cliente

El cliente se encarga de enviar todo tipo de acciones de altas, bajas, cambios y consulta de registros, por medio de una interfaz gráfica que con la cual el usuario enviará estas acciones sin que él tenga conciencia del trabajo de los sockets.

Servidor

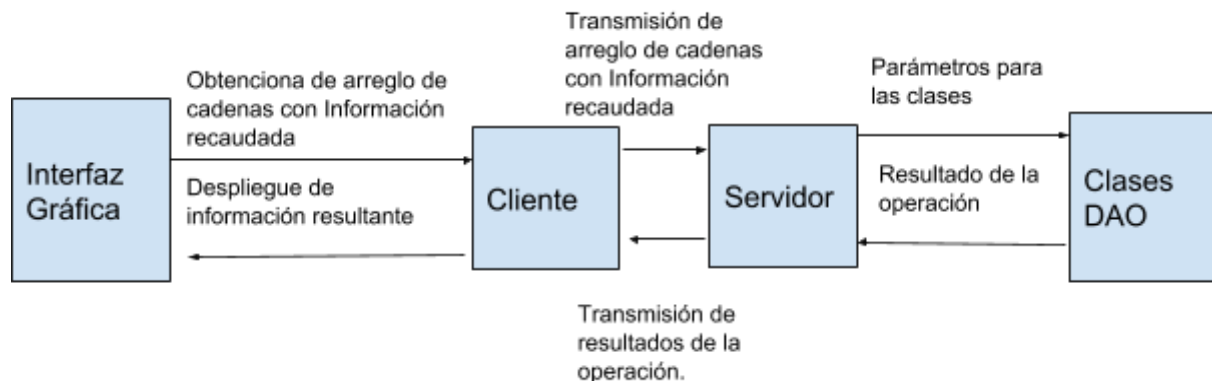
El servidor recibe siempre un arreglo de cadenas, donde se encarga de romperlas en tokens que son útiles como parámetros para las diferentes acciones que tiene implementadas.

Una vez conseguidos los parámetros, se busca la opción correspondiente y llamamos a la clase DAO que nos retorna verdadero o falso en caso de una alta, baja o cambio y los registros si se trata de una Consulta.

DAO

Es una serie de clases que representan las entidades de la base de datos como objetos con métodos CRUD.

En un diagrama a bloques lo podemos ver de la siguiente forma:

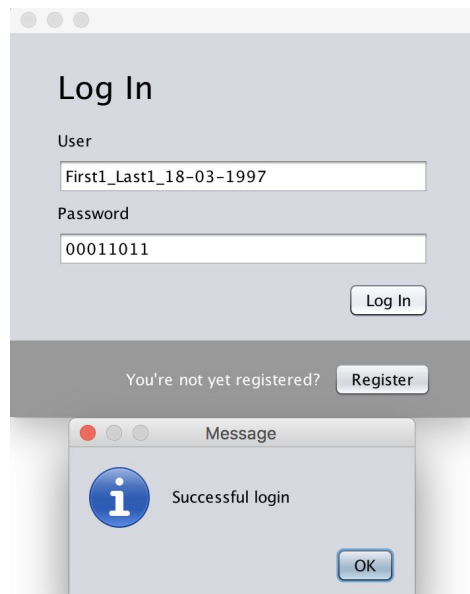


Resultados

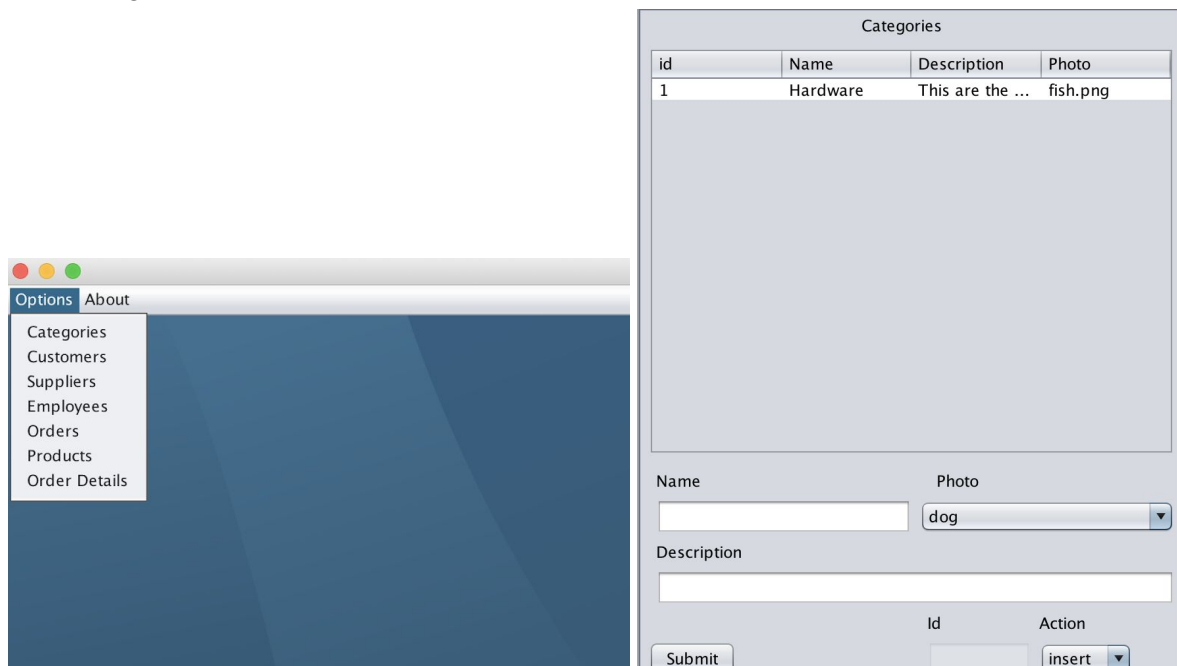
Corremos el servidor de la aplicación:

Servidor esperando por conexión...

Corremos la aplicación cliente, que es una pantalla “splash screen” donde podemos iniciar sesión o podemos registrarnos, si tenemos cuenta solamente debemos meter nuestro usuario y contraseña, si todo sale bien podremos ver un mensaje de logeo exitoso.



Dentro de la aplicación nos mostrará como opciones las diferentes entidades en del sistema, donde cada una de ellas nos permite insertar, borrar, actualizar y ver el listado de registros de cada uno de ellos, en la figura siguiente podemos ver en la parte izquierda, el escritorio de la aplicación con todas sus opciones, y del lado derecho el panel de gestión para categorías.



En la siguiente figura podemos ver el panel de gestión de los customers.

Customers

id	Compa...	Contac...	Contac...	Address	City	Region	Postal ...	Country	Phone	Fax
custm1	Microsoft	Stephani	Recruiter	Linden ...	California	East C...	1803	USA	52098...	No fax

Company Name

ContactName

Contact Title

Address

City

Region

Postal Code

Country

Phone (Ex. 521298786509)

Fax

Action

insert

ID

Submit

En la siguiente figura podemos ver el panel de gestión de los suppliers.

Suppliers

id	Compa...	Contact...	Contact...	Address	City	Region	Postal ...	Country	Phone	Fax	HomeP...
1	CardBo...	Aaron	Prime ...	Sepulve...	California	East Co...	1805	USA	52129...	No fax	www.its...

Company Name

ContactName

Contact Title

Address

City

Region

Postal Code

Country

Phone (Ex. 557623091827)

Fax

HomePage (www.example.com)

Action

insert

ID

Submit

En la siguiente figura podemos ver el panel de gestión de los products.

Products

id	Name	Quantity ...	Unit Price	Units/Sto...	Units on ...	Reorder ...	Disconti...	Supplier	Category
1	SSD 250...	1	100	200	75	3	4	1	1

ProductName

Units on Order (Numerical)

Supplier

QuantityPerUnit

Reorder Level (Numerical)

Category

Unit Price (Numerical)

Discontinued (Numerical)

Action

Units in Stock (Numerical)

ID

Submit

En la siguiente figura podemos ver el panel de gestión de los employees.

Employees

id	Pri...	La...	Fir...	Title	Tit...	Bir...	Hi...	Ad...	City	Re...	Po...	Co...	Ph...	Ex...	Ph...	No...	Re...
1	00...	0	La...	Fir...	En...	Sis...	18...	12...	Av...	Ca...	Ea...	18...	USA	52...	432	cat...	Ne...
2	00...	1	La...	Fir...	En...	Sis...	18...	12...	AV...	Ca...	Ea...	18...	USA	52...	432	cat...	Ne...

Last Name

BirthDate (dd-mm-yyyy)

Region

Extension

First Name

HireDate (dd-mm-yyyy)

Postal Code

Photo

Title

Address

Country

Notes

Title of courtesy

City

Phone (Ex. 5219891821)

Reports To

Password

Privilege

ID

Action

Submit

En la siguiente figura podemos ver el panel de gestión de los Orders.

id	Orde...	Requ...	S. Date	S. Via	Freight	S. Na...	S. Ad...	S. City	S. Re...	S. CP	S. Co...	Empl...	Custo...
1	10-0...	20-0...	11-0...	1	1	Andrew	Hunti...	Califo...	East ...	1802	USA	1	custm1

OrderDate (dd-mm-yyyy)	Freight (Numerical)	ShipCity	Employee
<input type="text"/>	<input type="text"/>	<input type="text"/>	1_First1 Last1
Required (dd-mm-yyyy)	ShipName	ShipRegion	Customer
<input type="text"/>	<input type="text"/>	<input type="text"/>	custm1_Stephani Micro...
Shipped Date (dd-mm-yyyy)	ShipAddress	Ship Postal Code	Action
<input type="text"/>	<input type="text"/>	<input type="text"/>	insert
ShippedVia (Numerical)	ShipCountry	ID	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Submit"/>

En la siguiente figura podemos ver el panel de gestión de los Order Details.

id	UnityPrice	Quantity	Discount	OrderID	ProductID
1	100	74	0.34	1	1

UnityPrice (Numerical)	OrderID
<input type="text"/>	1_10-09-2018
Quantity (Numerical)	ProductID
<input type="text"/>	1_SSD 250GB Ex...
Discount (Numerical)	Action
<input type="text"/>	insert
ID	
<input type="text"/>	<input type="button" value="Submit"/>

En la siguiente imagen mostramos las diferentes acciones que realiza el sistema.

UnityPrice (Numerical)

Quantity (Numerical)

Discount (Numerical)

OrderID
1_10-09-2018

ProductID
1_SSD 250GB Ex...

Action
insert
delete
update
consult

En la siguiente imagen, mostramos el panel de registro de un nuevo usuario

Register

Last Name

First Name

Title

Title of Courtesy

Password

BirthDate

HireDate

Address

City

Privilege
1 Basic User Usage

Region

Postal Code

Country

Phone

Extension

Photo
cat

Notes

Reports To

Register

Conclusiones

Este ejercicio me ayudo a practicar mis habilidades para el desarrollo de aplicaciones en red, además de que use un patrón de diseño llamado DAO, que ya despues evoluciono a cosas más avanzadas como entidades en hibernate o en modelos en el paradigma de modelo-vista-controlador.

