



Casos prácticos

PROGRAMACIÓN CON JAVA ESTÁNDAR EDITION

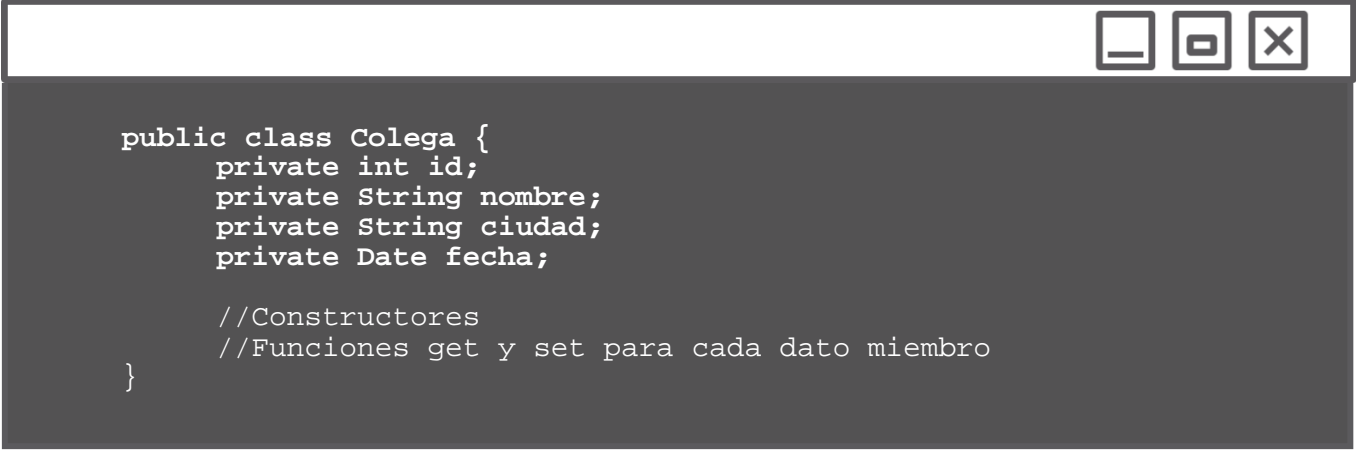
Telefonica
EDUCACIÓN DIGITAL

PRUEBA PRÁCTICA

Se tiene que escribir un programa que utilizando el patrón de diseño DAO, como en la última actividad del capítulo 4, realice operaciones de visualización, inserción, borrado y actualización de registros de una tabla **"colega"** de una base de datos MySQL **"colegajdbc"**.

Los pasos a seguir son los siguientes:

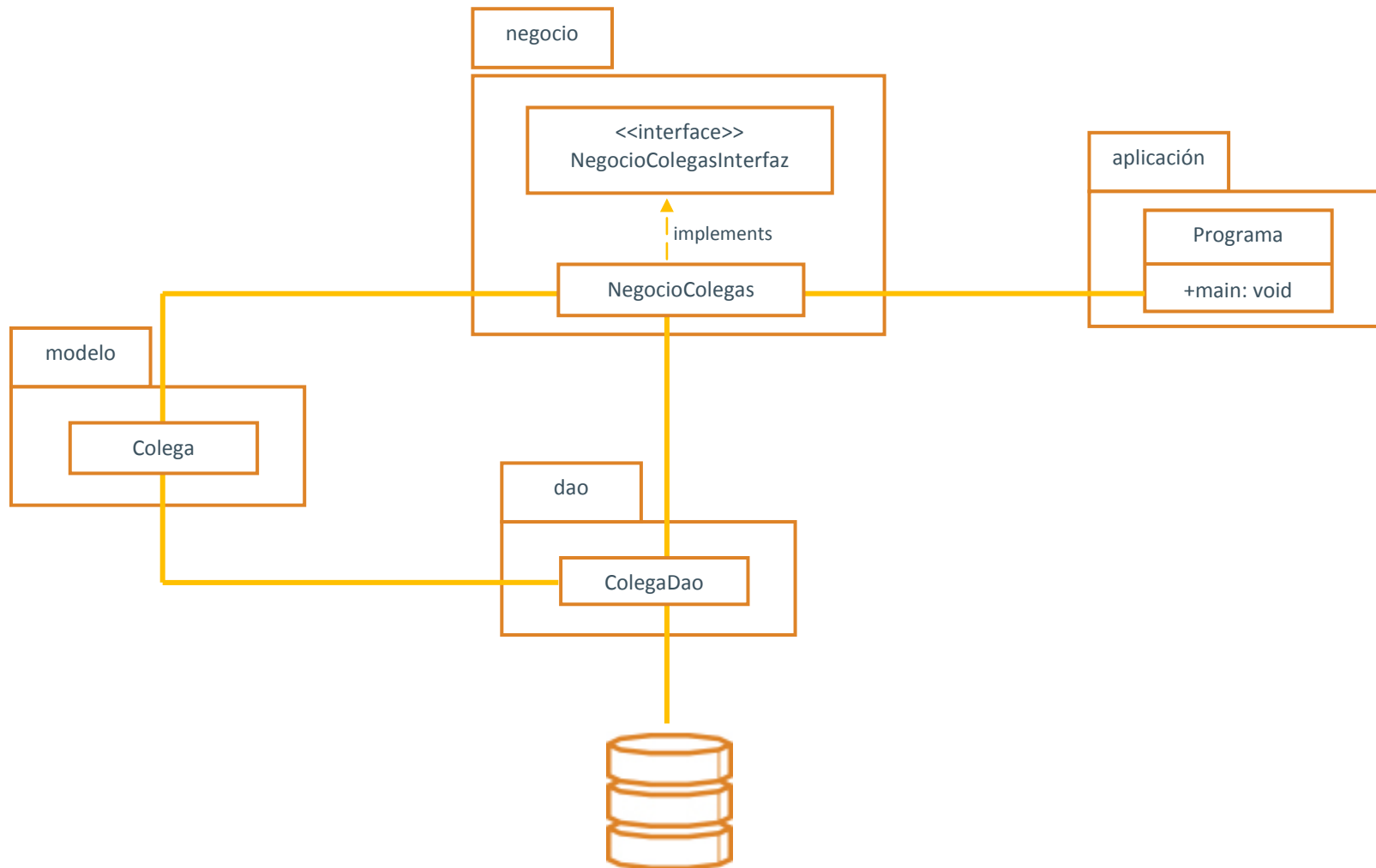
1. Crear la tabla **"colega"** en el esquema **"colegajdbc"** en MySQL. Dicha tabla tiene que corresponderse con la clase TDO Colega cuyos datos miembros son los que se indican a continuación:



```
public class Colega {  
    private int id;  
    private String nombre;  
    private String ciudad;  
    private Date fecha;  
  
    //Constructores  
    //Funciones get y set para cada dato miembro  
}
```

El campo "id" debe ser "Primary key" y generado su valor de forma automática por MySQL.

2. Introducir algunos registros en esta tabla.
3. En Eclipse crear un proyecto Java, con nombre **PruebaJSE**.
4. En proyecto **PruebaJSE**, implementar las clases de acuerdo con el diagrama del patrón DAO siguiente:



La estructura del proyecto es la siguiente:

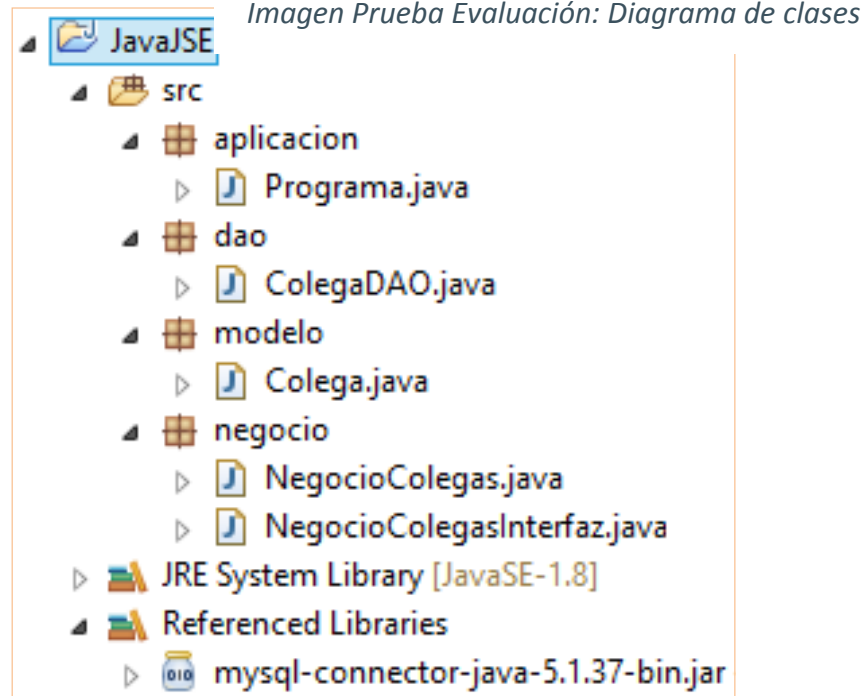


Imagen Prueba Evaluación: Estructura del proyecto

5. Implementar la clase DTO **Colega**, de acuerdo con las siguientes características:

- Un dato miembro para cada campo de la tabla, constructores.
- El dato fecha será de tipo `java.sql.Date`.

- Dos constructores:

```
public Colega(int id, String nombre, String ciudad, Date fecha)
public Colega(String nombre, String ciudad, Date fecha)
```

- Funciones get y set para cada dato miembro.

6. Implementar la clase **NegocioColegas** con todas las funciones de la interface **NegocioColegasInterfaz**:

```
public interface NegocioColegasInterfaz {
    //
    public Colega unColega();
    public void listarTodosLosColegas() ;
    public int idMayorColegas();
    public int numeroColegas();
    public void insertarUnColega();
    public void insertarVariosColegas();
    public void buscarColega();
}
```

- La función **unColega()**, devuelve una referencia a un objeto de la clase Colega creado con de esta forma Colega c = new Colega(nombre, ciudad, fechaSql).

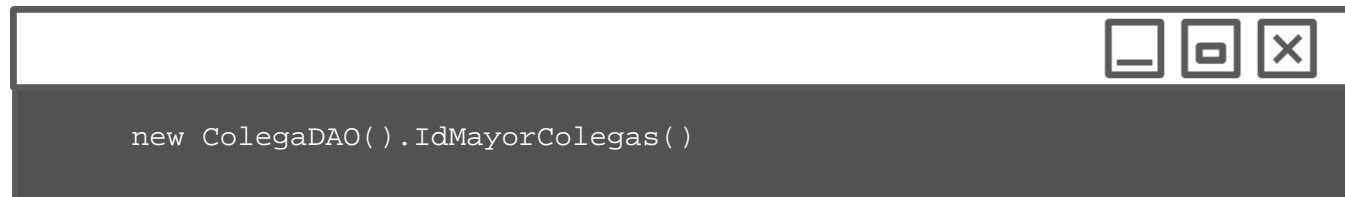
Los valores para el nombre, la ciudad y la fechaSql son valores aleatorios, obtenidos apoyándose en la clase Random.

- La función **listarTodosLosColegas()**, visualizara por pantalla los valores de todos los campos de todos los registros de la tabla "colega", obtenidos de la forma siguiente:

A screenshot of a Java IDE window. The window has a title bar with standard minimize, maximize, and close buttons. The main area is dark-themed and contains the following Java code:

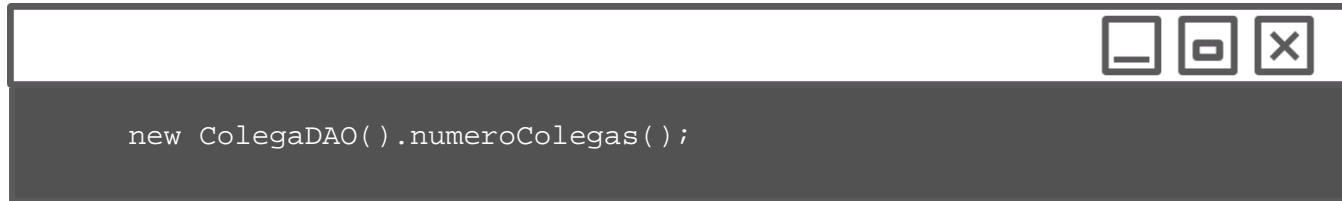
```
List<Colega> listacolegas = new ColegaDAO().mostrarTodos();
```

- La función **idMayorColegas()**, visualizara por pantalla el valor del campo "id" de la tabla que tenga mayor valor, obtenido de la forma siguiente:

A screenshot of a Java IDE window. The window has a title bar with standard minimize, maximize, and close buttons. The main area is dark-themed and contains the following Java code:

```
new ColegaDAO().IdMayorColegas()
```

- La función **numeroColegas()**, visualizara por pantalla el número de registros de la tabla "colega", obtenido de la forma siguiente:



- La función **insertarUnColega()**, inserta en la tabla "colega" un registro, cuyos valores son los de un objeto TDO Colega que devuelve como referencia la función `unColega()`, y utiliza la función `altaColega()` de la clase `ColegaDAO`.
- La función **insertarVariosColegas()**, inserta un número aleatorio de registros en la tabla "colega", apoyándose en la función anterior.
- La función **buscarColega()**, busca en la tabla "colega" un registro cuyo "id" sea el valor que de forma aleatoria se obtiene, en el rango de valores de dicho campo "id", el valor mínimo del rango será desde el "id" menor y al "id" mayor. Si el registro existe se muestra y si no se informa.

La búsqueda se realiza utilizando la función `buscarColega(int índice)` de la clase `DAO`.

7. Implementar la clase **ColegaDAO**, que facilita las funcionalidades necesitadas por **NegocioColegas**. La clase **ColegaDAO** es la encargada del acceso a los datos. Las funciones que hay que implementar en esta clase son las que se muestran en la imagen siguiente.

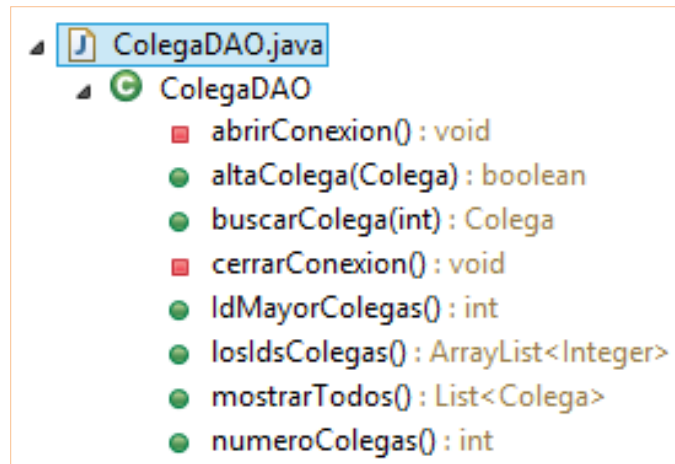
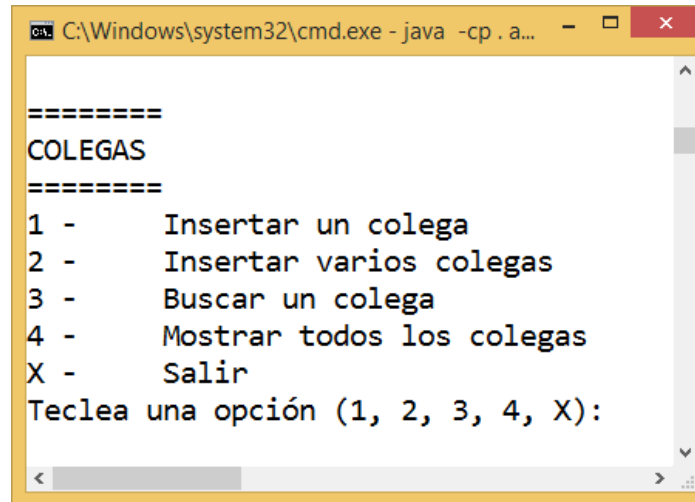


Imagen Prueba Evaluación: Clase ColegaDAO

8. Implementar la clase **Programa**, con función main que a través de un menú como el mostrado en la imagen, permita ejecutar las funcionalidades de la clase **NegocioColegas**.



```
C:\Windows\system32\cmd.exe - java -cp . a...  
  
=====  
COLEGAS  
=====  
1 -      Insertar un colega  
2 -      Insertar varios colegas  
3 -      Buscar un colega  
4 -      Mostrar todos los colegas  
X -      Salir  
Teclea una opción (1, 2, 3, 4, X):
```

Imagen Prueba Evaluación: Ejecución del programa

9. Se recomienda primero se implemente una función en la clase **ColegaDAO** y probando sus funciones, para después añadirlas en la clase **NegocioDAO** y a continuación implementar la función main de la clase Programa.

Telefónica

EDUCACIÓN DIGITAL