

# La API JDBC

## Clases DAO (Data Access Object)

El patrón arquitectónico Data Access Object (DAO), permite separar la capa de de negocios de lógica de acceso a datos, de tal forma que el DAO encapsula toda la lógica de acceso de datos del resto de la aplicación.

Implementar la lógica de acceso a datos en la capa de lógica de negocio puede hacer el código complejo y no extensible. Se recomienda siempre usar DAO para abstraer y encapsular todos los accesos a los datos.

El DAO maneja la conexión con las fuentes de datos para obtener y almacenar datos.

# La API JDBC

## Clases DAO (Data Access Object)

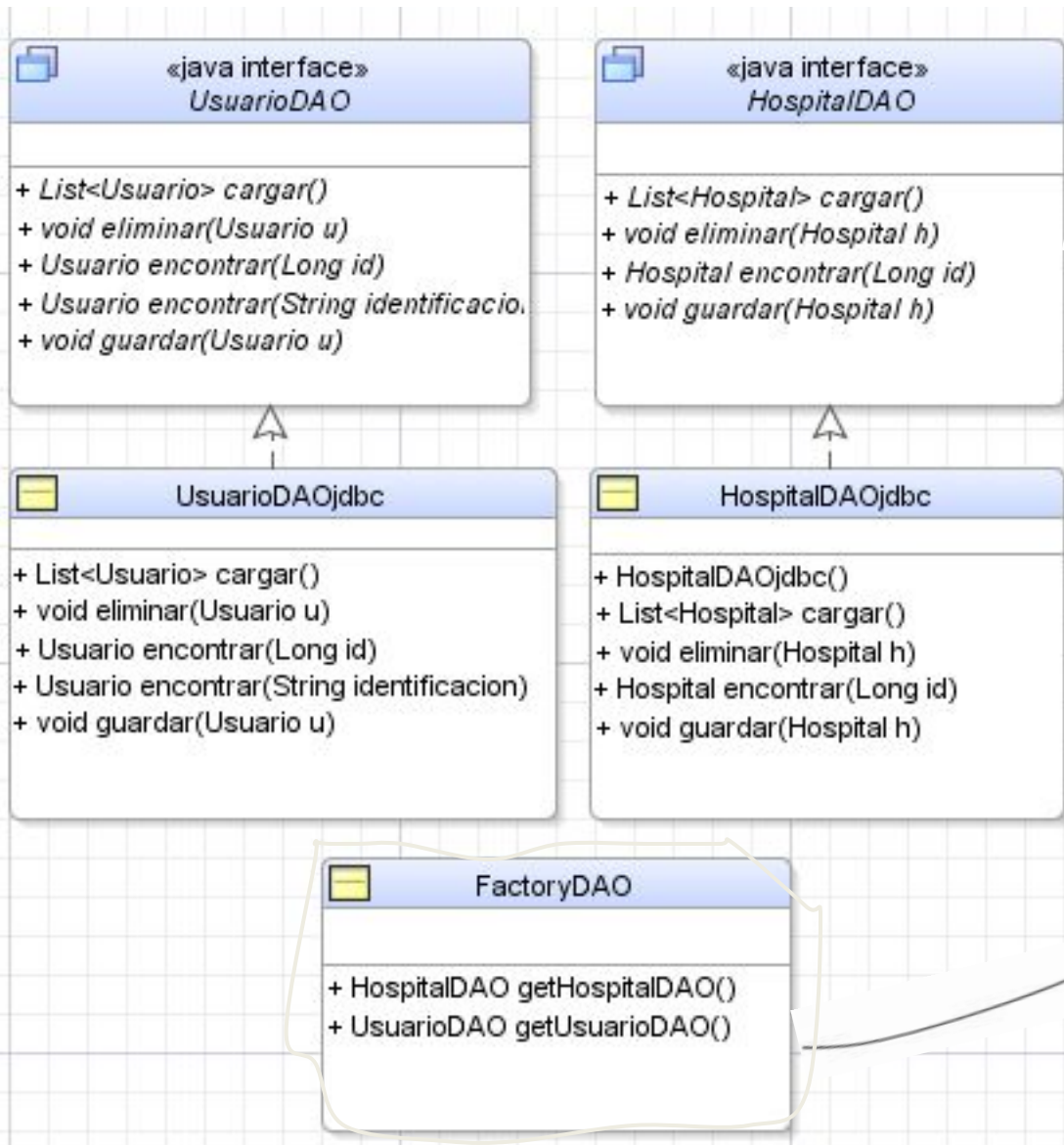
Antes de comenzar con los DAO veremos cómo ubicar el código que obtiene una conexión a la base de datos en un lugar único. Es una buena práctica usar el patrón Singleton.

```
import java.sql.*;
public class MiConnection {
    private static Connection con = null;
    static {
        try {
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/tokio2021");
        } catch (java.sql.SQLException e) {
            System.out.println("Error de SQL: "+e.getMessage());
        }
    }

    public static Connection getCon() {
        return con;
    }
}
```

# La API JDBC

## Clases DAO



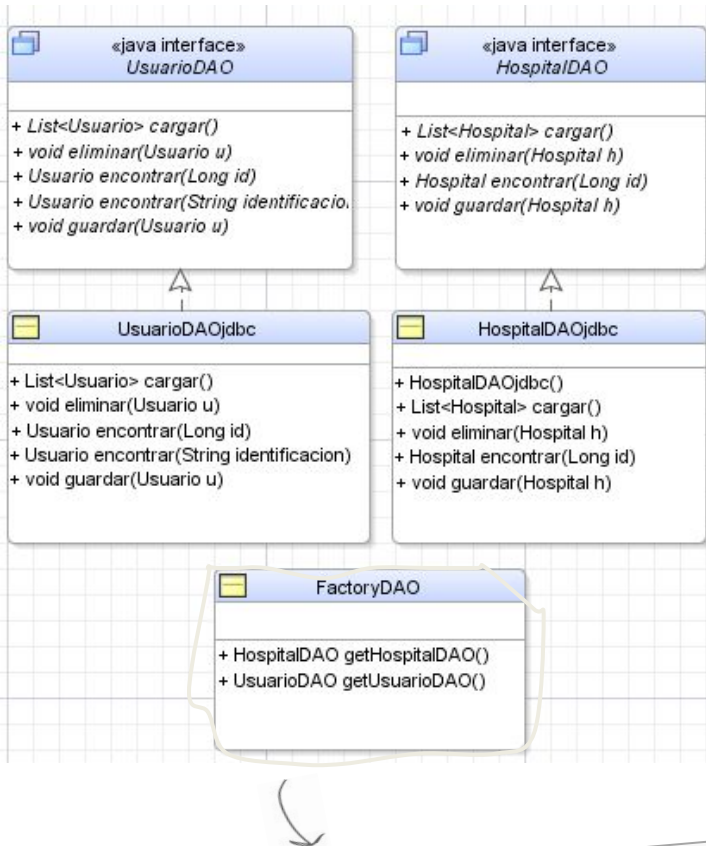
Las interfaces xxxDAO tienen operaciones comunes de acceso a datos.

Implementaciones de las interfaces xxxDAO usando JDBC

Esta clase crea objetos xxxDAO. Nos provee de objetos que implementan las distintas interfaces xxxDAO. Estos objetos son usados para acceder a la capa de datos.

# La API JDBC

## Clases DAO



```

public class FactoryDAO {
    public static UsuarioDAO
    getUsuarioDAO() {
        return new UsuarioDAOjdbc();
    }
    . . .
}
    
```

```

package dao.implJDBC;

public class UsuarioDAOjdbc implements UsuarioDAO {

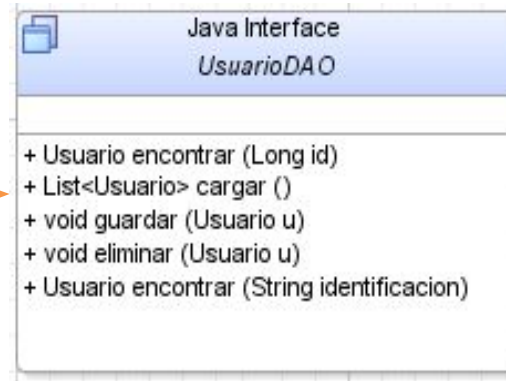
    public Usuario encontrar(String identificacion) {
        Usuario usuario = null;
        try{
            Connection con = MiConnection.getCon();
            Statement st = con.createStatement();
            ResultSet rs= st.executeQuery("Select * from Usuarios
                                         where u.identificacion='"+identificacion+"'");
            if (rs.next()==true) {
                usuario = new Usuario();
                usuario.setMatricula(rs.getInt(1));
                usuario.setApeynom(rs.getString(2));
                // más setters
            }
            rs.close();
            st.close();
            con.close();
        } catch (java.sql.SQLException e) {
            System.out.println("Error de SQL: "+e.getMessage());
        }
        return usuario;
    }

    public List<Usuario> cargar() {...}
    public void eliminar(Usuario u) {...}
    public Usuario encontrar(Long id) {...}
    public void guardar(Usuario u) {...}
}
    
```

# La API JDBC

## Clases DAO

Cuando se implemente la aplicación, se invocará al DAO correspondiente para acceder a los datos.



```
Usuario usr = new Usuario();  
UsuarioDAO uDAO = FactoryDAO.getUsuarioDAO();  
Usuario u = uDAO.guardar(usr);
```