



Patrones de diseño

Sesión Conceptual 2





Inicio

{desafío}
latam_



15 minutos

- Profundizar en los conceptos de JSP, aplicando sus funcionalidades en una estructura con arquitectura MVC.
- Aproximar al trabajo con bases de datos desde una aplicación web mediante la API de JDBC y se dará forma a un sistema.

Objetivo



Desarrollo

{desafío}
latam_



150 minutos

**/* Codificación MVC de
la capa de presentación */**

Construcción Inscripción Servlet

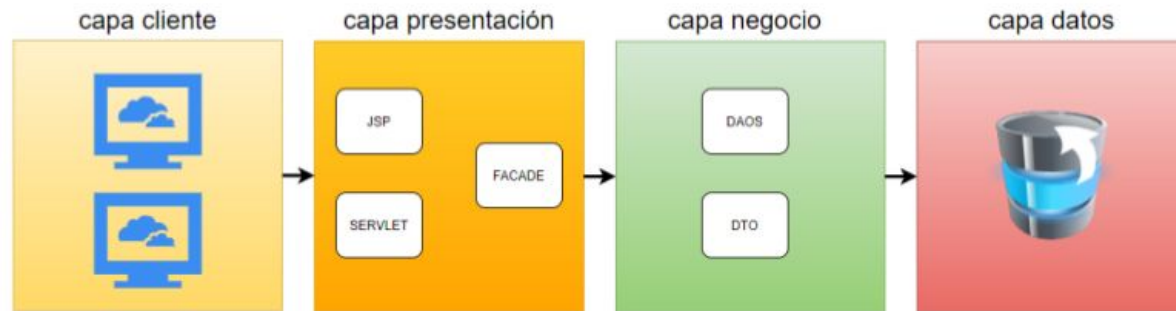
Permite al usuario
ingresar su
nombre y un
número de celular

Muestra dos
listas
desplegable

Una que
despliega los
cursos
disponibles

Otra que
muestra los
distintos tipos
de pagos

Presentación del entorno



com.desafiolatam.daos

CursoDAO

```
listaDeCursos: List<CursoDTO>  
consultaSql: String  
+ obtieneCursos(): List
```

FormaDePagoDAO

```
+ formaDePago: List<FormaDePagoDTO>  
+ consultaSql: String  
+ obtieneFormasDePago(): List<FormaDePagoDTO>
```

InscripcionDAO

```
max: int  
consultaProximoid: String  
insertarInscripcion: String  
+ insertarInscripcion(InscripcionDTO dto): int
```

com.desafiolatam.entidades

CursoDTO

```
- idCurso: int  
- descripcion: String  
- precio: double  
+ get()  
+ set()
```

FormaDePagoDTO

```
- idFormaDePago: int  
- descripcion: String  
- recargo: double  
+ get()  
+ set()
```

InscripcionDTO

```
- idInsc: int  
- nombre: String  
- celular: String  
- idCurso: int  
- idFormaDePago: int  
+ get()  
+ set()
```

com.desafiolatam.facade

Facade

```
+ field: Type  
+ registrarInscripcion(): int  
+ obtenerCursos(): List<CursoDTO>  
+ obtenerFormasDePago(): List<FormaDePagoDTO>
```

com.desafiolatam.servlets

PosInscripcion

```
+ field: Type  
+ doGet(): void
```

PreConfirmacion

```
+ field: Type  
+ doGet(): void
```

PreInscripcion

```
+ field: Type  
+ doGet(): void
```

Diagrama de clases: Servlets

☒ Autocommit Rows  

select * from inscripcion

Results Explain Describe Saved SQL History

ID_INSCRIPCION	NOMBRE	TELEFONO	ID_CURSO	ID_FORMA_PAGO
3	superman	98097989	4	2
4	marcelo	765786	2	2
1	jorge	45525225	1	1
2	goku	23837	1	1

4 rows returned in 0.00 seconds [Download](#)

Revisando la
base de datos

/* Concepto de ORM */

Descripción de la tecnología

En el principio de los tiempos, era necesario mezclar códigos y conceptos distintos entre sí.

Ya en estos niveles de programación hablamos de capas.

Mediante código incrustado en string de java generamos las querys correspondientes.

ORM.



Por qué un ORM

- Disminuir los riesgos.
- Facilita el diseño de la capa de datos mediante el mapeo de las mismas clases del sistema.

Mapeo Relacional

- Genera “espejos” de las tablas del modelo de datos.
- Crea clases con la misma estructura y atributos.

Algunos métodos:

- `buscarUsuarios();`
- `buscarUsuariosPorId(idUsuario);`
- `eliminarUsuario(idUsuario);`

- Biblioteca de mapeo relacional de objetos de código abierto para java.

Diferencias con el estándar JDBC:

Hibernate se conecta con la base de datos y usa HQL para ejecutar todas las consultas.

Se configura mediante xml facilitando su gestión.

Mejor manejo de la conexión ya que trabaja mediante la sesión de la aplicación.

Administra la conexión como si de un singleton se tratara.

Maapeo de objetos

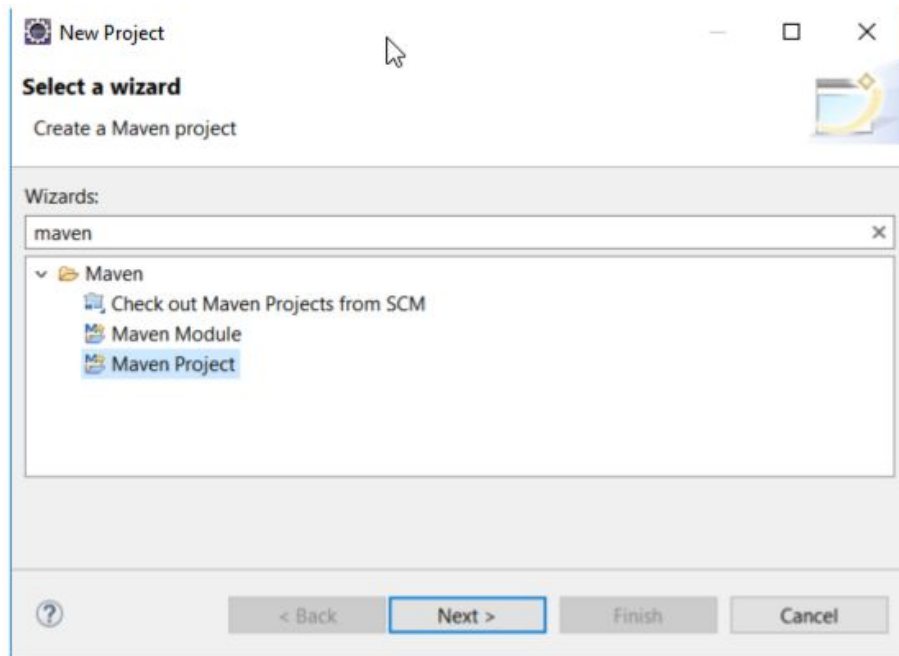
Hibernate asigna directamente las clases de java a las tablas de la base de datos utilizando xml o anotaciones.

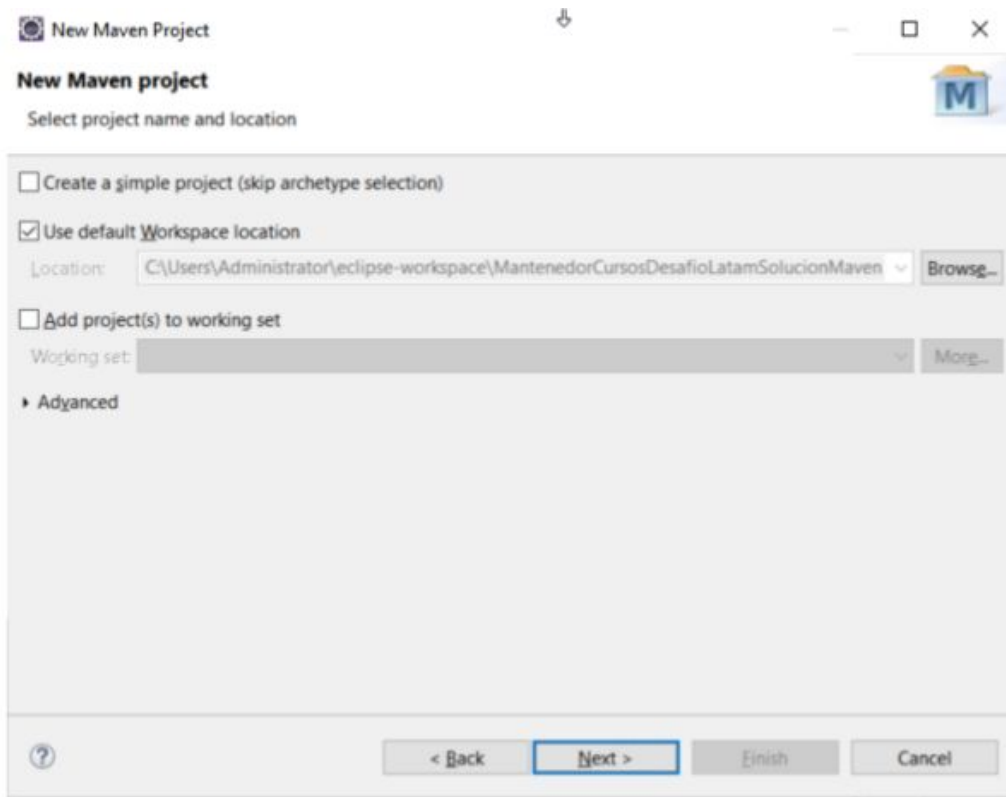
Implementación con JDBC.

```
while(resultado.next()) {  
    CursoDTO cursoDto = new CursoDTO();  
  
    cursoDto.setIdCurso(resultado.getInt("id_curso"));  
  
    cursoDto.setDescripcion(resultado.getString("descripcion"));  
  
    cursoDto.setPrecio(resultado.getDouble("precio"));  
    listaDeCursos.add(cursoDto);  
}
```


Cómo funciona un ORM con un ejemplo utilizando Hibernate con anotaciones

File-New Project-Maven project.





Solo marcar las
opciones
señaladas

New Maven Project

New Maven project

Configure project

Artifact

Group Id:

Artifact Id:

Version:

Packaging:

Name:

Description:

Parent Project

Group Id:

Artifact Id:

Version:










Browse... Clear

Advanced

< Back Next > Finish Cancel

@

Los campos 'Name' y 'Description' son opcionales

- ▼  cursos
 - >  src/main/java
 - >  src/main/resources
 - >  src/test/java
 - >  src/test/resources
 - >  JRE System Library [J2SE-1.5]
 - >  src
 - >  target
 -  pom.xml

Generación del proyecto maven

Anotaciones

@Entity	Convierte la clase que la acompaña en un bean de entidad.
@Table	Hace referencia a la tabla de base de datos que la clase representa.
@Id	Identifica a la primary key de la tabla.
@Basic	Indicar que son atributos normales de la base de datos.

/* Java Beans */

Los JavaBean y las bibliotecas de clases

- Los beans.
 - son apropiados para crear componentes de software visuales que son manipulados y personalizados para cumplir con requisitos específicos.
- Las bibliotecas de clases.
 - Son pojos que proporcionan funcionalidades para los programadores.

Ventajas	Desventajas
Son persistentes, y se pueden recuperar.	Los beneficios de los objetos inmutables están ausentes.
Controlamos que propiedades, métodos y eventos deben exponerse a la aplicación que desea usarla.	Un objeto sea instanciado en un estado inválido.
Generar eventos.	Contienen mucho código repetitivo.

Reglas para los JavaBean

A diagram consisting of several overlapping circles. Three large orange circles contain text about JavaBean rules. They are surrounded by several smaller green circles of varying sizes. The background is white.

Debe ser serializable.

Las propiedades privadas de acceso programático deben ser accesibles a través de un método de obtención y establecimiento público.

Debe tener un constructor público sin argumentos.

Características Importantes

Propiedades

Atributos con nombre asociados con un bean.

Métodos

Métodos POJO, el bean puede elegir exponer un subconjunto de sus tipos públicos.

Eventos

Notifique a otros componentes si ocurre algo interesante.

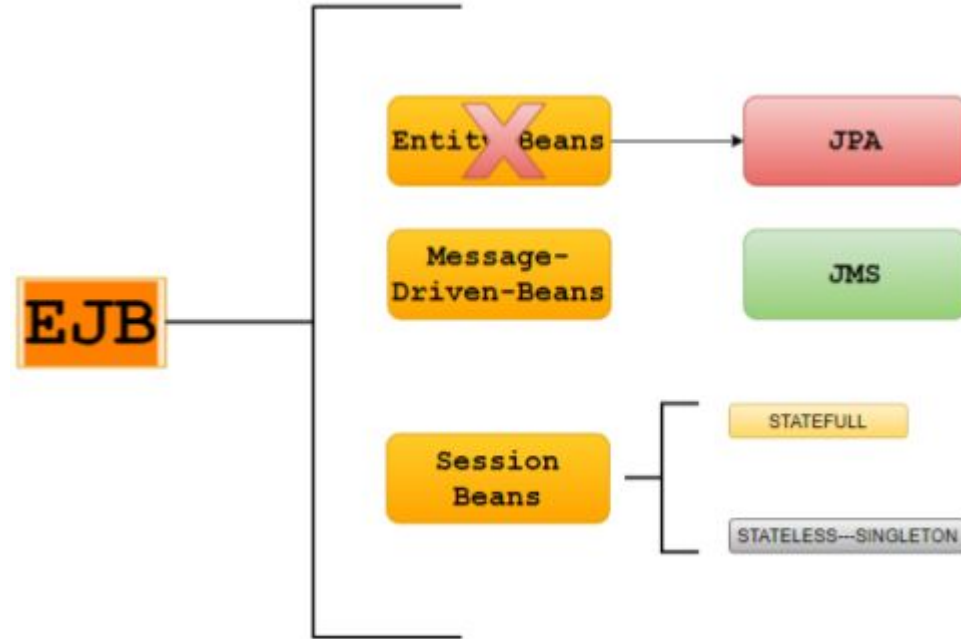
Cuando usar los EJB

- Considerar el uso de los EJB si tu aplicación tiene los siguientes requerimientos.

La aplicación debe ser escalable	Distribuir los componentes de la aplicación.
Las transacciones deben asegurar la integridad de los datos	Mecanismo que administra el acceso concurrente de objetos compartidos.
La aplicación tendrá una gran variedad de clientes	Cientes remotos pueden fácilmente localizar EJB.

Tipos de EJB

1. Entity Bean: herederos de las versiones viejas del framework.
2. Message-driven beans: son listener que esperan ser notificados por el container.
3. Session Beans: se dividen en dos categorías, stateful y stateless.



Primer EJB

Para crear un EJB:

```
package com.ejb.sesionbean;

import javax.ejb.LocalBean;
import javax.ejb.Stateless;

/**
 * Session Bean implementation class
 * CalculadoraBean
 */
@Stateless
@LocalBean
public class CalculadoraBean {

    public double sumar(double a, double b){
        return a+b;
    }

}
```

- Cualquier proceso local o remoto que necesite de los servicios de este EJB.
 - **Cientes Web:** Un servlet
 - **Cientes EJB:** Un EJB puede solicitar los servicios de otro EJB
 - **Cientes StandAlone:** Un proyecto java clásico con método main.

```
@EJB
private CalculadoraBean ejb;

@WebServlet("/TestEJB")
public class TestEJB extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException,
        IOException {
        PrintWriter out = response.getWriter();
        out.print("<html><body>");

        int a = Integer.parseInt(request.getParameter("a"));
        int b = Integer.parseInt(request.getParameter("b"));

        out.print(ejb.sumar(a,b));

        out.print("</body></html>");
        out.close();
    }
}
```



Quiz

{desafío}
latam_



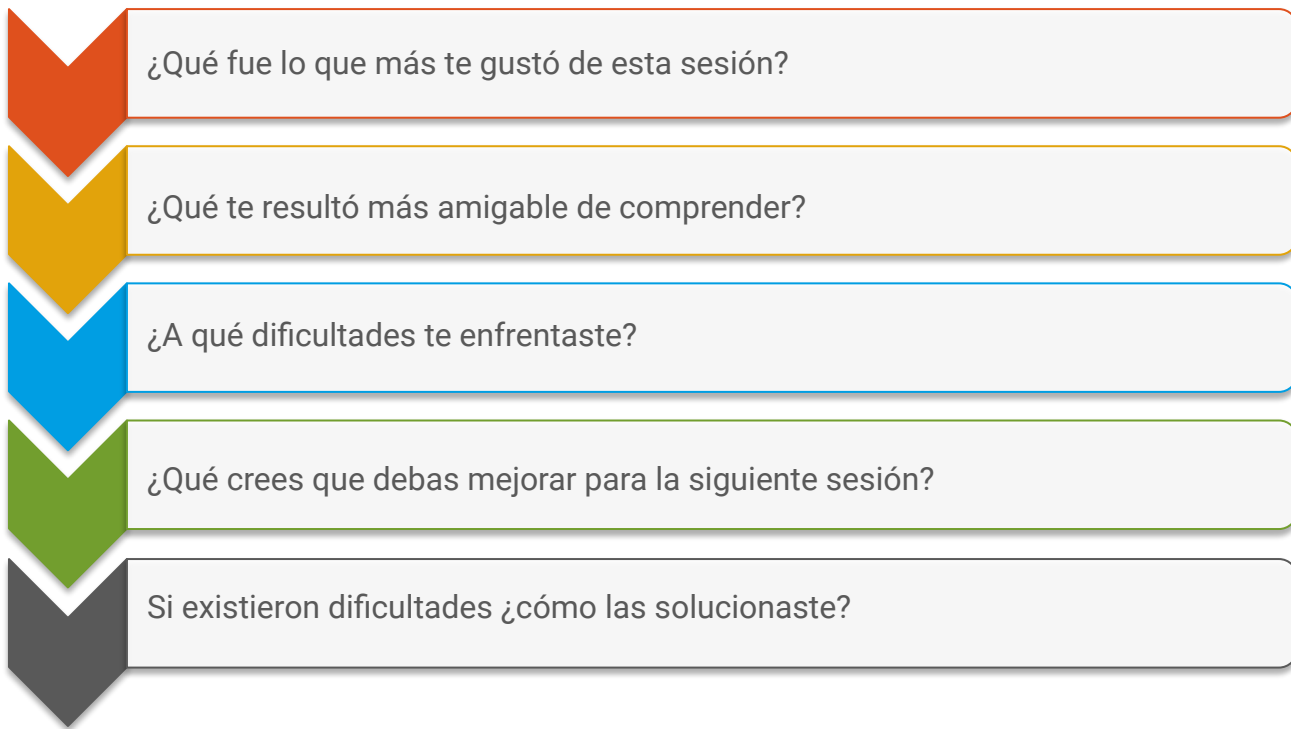


Cierre

{desafío}
latam_



15 minutos



¿Qué fue lo que más te gustó de esta sesión?

¿Qué te resultó más amigable de comprender?

¿A qué dificultades te enfrentaste?

¿Qué crees que debas mejorar para la siguiente sesión?

Si existieron dificultades ¿cómo las solucionaste?



*Academia de
talentos digitales*

www.desafiolatam.com



/DesafioLatam



/DesafioLatam



/DesafioLatam



/DesafioLatam