



## PRÁCTICA DE LABORATORIO / TALLER Nro. 3

Carrera Computación

A. DATOS INFORMATIVOS		
<b>Asignatura:</b> Programación Orientada a Objetos	<b>Ciclo / Semestre:</b> Segundo	<b>Paralelo:</b> A - B
<b>Docente:</b> Wilman Chamba Zaragocín	<b>Período Académico:</b> Marzo 2025 – Agosto 2025	
<b>Integrantes:</b> Yober Gaona		

B. INFORMACIÓN GENERAL	
<b>Unidad:</b> 3. Clases y objetos	
<b>Resultado de aprendizaje:</b> R2. Usa subclases para diseñar jerarquías de clases simples que permite crear código reutilizable para distintas subclases, bajo los principios de solidaridad, transparencia, responsabilidad y honestidad.	
<b>Tema:</b> Desarrollo de una Interfaz Gráfica de Usuario (IGU) utilizando Jakarta Faces (JSF) en un proyecto autónomo propuesto por el equipo	
<b>Fecha:</b> Loja,	<b>Nro. horas:</b> 3 – 4
<b>Objetivos:</b> <ul style="list-style-type: none"><li>● Diseñar e implementar una interfaz web funcional utilizando Jakarta Faces.</li><li>● Integrar componentes IGU (como formularios, tablas y validaciones) en el proyecto propuesto.</li><li>● Aplicar el patrón de diseño MVC (Modelo-Vista-Controlador) en la arquitectura del proyecto.</li><li>● Fomentar el trabajo colaborativo mediante la distribución de tareas en el equipo..</li></ul>	
<b>Recursos y/o materiales:</b>	
<b>Software:</b> <ul style="list-style-type: none"><li>Entorno de desarrollo Java SE 17+.</li><li>☐ IDE (NetBeans, IntelliJ IDEA).</li><li>☐ Servidor de aplicaciones: Jakarta EE (OpenLiberty, MicroPayara o TomEE).</li><li>☐ Dependencias Maven para Jakarta Faces y PrimeFaces.</li><li>☐ Documentación oficial de Jakarta Faces y PrimeFaces.</li><li>☐ Proyecto autónomo previamente propuesto por el equipo.</li><li>☐</li></ul>	
<b>Recursos:</b> <ul style="list-style-type: none"><li>☐ Recursos proporcionados en clase</li><li>☐ Documentación del API de Java.</li><li>☐ Plantilla de UML.</li><li>☐ <a href="#">SOLID Cheat Sheet</a>.</li></ul>	

C. DESARROLLO
<b>Problemática del Proyecto Autónomo:</b>
Título de proyecto: Parqueo Facil Repositorio en GitHub: <a href="https://github.com/Gaona-Yober/Practica_03">https://github.com/Gaona-Yober/Practica_03</a>
Descripción de módulos del proyecto autónomo a desarrollar
<b>Procedimiento:</b>
<b>1. Organización del equipo (30 min):</b> <ul style="list-style-type: none"><li>● Definir roles:</li></ul>



- Diseñador de interfaz: Encargado de maquetar las vistas (.xhtml).
- Lógica de vista (Controlador): Implementa los managed beans (@Named o @ManagedBean).
- Coordinador de pruebas: Valida el flujo y la usabilidad.
- Revisar los requisitos del proyecto propuesto y priorizar las pantallas a desarrollar.

## 2. Implementación técnica (2 horas):

- Crear vistas con Facelets (ej: registro.xhtml) usando componentes como:
 

```
<h:form>
  <h:inputText value="#{usuarioBean.nombre}" required="true" />
  <h:commandButton value="Guardar" action="#{usuarioBean.guardar()}" />
</h:form>
```
- Configurar navegación entre páginas.
- Implementar validadores personalizados o usar los incorporados (<f:validateBean>).

## 3. Integración con el proyecto existente (1 hora):

- Conectar los managed beans con las vistas.
- Asegurar que los datos fluyan correctamente entre la IGU y la lógica del sistema.

## 4. Pruebas y depuración (30 min):

- Verificar que los formularios respondan a eventos correctamente.
- Validar mensajes de error y retroalimentación al usuario.

### Resolución o Resultados esperados:

- Pantallas funcionales integradas al proyecto autónomo.
- Código fuente documentado en el repositorio del equipo (enlace).  
[https://github.com/Gaona-Yober/Practica\\_03](https://github.com/Gaona-Yober/Practica_03)
- Demostración de la navegación entre vistas (Puede ser enlace a un video).  
<https://drive.google.com/drive/folders/1GNIEGkxVAjNBDrho7L1hWFQLqwC8OYLc?usp=sharing>

### Preguntas de retroalimentación:

- **¿Qué ventajas identificaron al usar JSF frente a otros frameworks web como Spring MVC?**  
JSF es ideal para desarrollos donde se busca rapidez, integración con Jakarta EE, y una interfaz basada en componentes visuales, mientras que Spring MVC se prefiere en entornos más modernos, flexibles o basados en microservicios.
- **¿Cómo manejaron la comunicación entre los managed beans y las entidades del proyecto?**
  1. Managed Beans reciben datos del formulario.
  2. Los datos se vinculan a una entidad JPA (@Entity).
  3. Se llama a un servicio o DAO para persistir, actualizar o consultar los datos.
  4. El flujo es:  
JSF Form → ManagedBean → Servicio → Entidad (JPA) → Base de datos
- **¿Qué desafíos técnicos enfrentaron al integrar Jakarta Faces con su proyecto existente?**

Aunque JSF ofrece muchos beneficios, su integración exigió cuidado con:

- La configuración del entorno,
- La comprensión del ciclo de vida, y
- El manejo preciso del modelo de datos y la vista.
- **¿Qué mejoras sugerirían para la interfaz desarrollada?**
  - Adaptabilidad para dispositivos móviles
  - Mejorar el diseño visual y la navegación
  - Mejorar la selección de horarios y espacios
  -

### Conclusiones:



1. JSF facilitó la creación de una interfaz basada en componentes que simplificó el desarrollo y mantenimiento del sistema, permitiendo una interacción más natural con las entidades y la lógica del negocio.
2. La integración de Jakarta Faces con JPA y la capa de servicios permitió gestionar eficientemente la persistencia de datos, garantizando que las reservas se almacenaran y consultaran correctamente en la base de datos.
3. A pesar de los desafíos técnicos iniciales, la aplicación logró cumplir con los objetivos funcionales, proporcionando una solución práctica y escalable para la gestión de espacios de parqueo, con posibilidades claras de mejora en usabilidad y diseño.

Estudiante(s):

Firma

Yober Gaona

1105256315

#### Rúbrica de Evaluación (10 pts)

Criterio	Cumple (2 pts)	Parcialmente (1 pt)	No cumple (0 pts)	Nota
Pantallas funcionales	Las vistas (.html) funcionan correctamente, con navegación y validaciones.	Funcionan, pero con errores menores o falta de pulido.	No funcionan o faltan.	
Integración con el proyecto	Los managed beans se conectan correctamente con la lógica del sistema.	Hay conexión, pero con fallos o datos inconsistentes.	Sin integración.	
Diseño básico y usabilidad	Interfaz clara, con componentes JSF bien utilizados (formularios, tablas, etc.).	Diseño aceptable pero poco intuitivo o desorganizado.	Diseño confuso o incompleto.	
Trabajo en equipo	Todos contribuyeron (evidenciado en commits o roles definidos).	Participación desigual o falta de coordinación.	No hubo colaboración.	
Presentación/Demo	Muestra el flujo completo, explica el código y responde preguntas.	Demo básica, con explicaciones limitadas o errores.	No se presentó.	



Universidad  
Nacional  
de Loja

FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y  
LOS RECURSOS NATURALES NO RENOVABLES  
*Carrera de Ingeniería en Sistemas / Carrera Computación*

**TOTAL:**

**Notas Adicionales de la Rúbrica:**

- Si no cumple el formato de la práctica, no poseerá calificación.
- Si no existe enlace al repositorio de código github o gitlab y con acceso, no poseerá calificación.
- El enlace al repositorio de código github o gitlab debe constar tanto en el informe oficial como en los comentarios de la actividad.
- Si no existe enlace al video o demostración de Vistas funcionando, no poseerá calificación.
- Bonus (+2): Uso de AJAX o características avanzadas de JSF.