

EVALUACIÓN PARCIAL PRIMER MOMENTO DE DESARROLLO DE SOFTWARE WEB BACKEND EN JAVA CON SERVLETS

Problema a Solucionar: Sistema de Gestión de Reservas para un Coworking

Tu tarea es desarrollar una aplicación Web Backend utilizando **Java Servlets** que permita gestionar las reservas de espacios en un coworking. El sistema debe tener las siguientes funcionalidades:

Requerimientos Funcionales:

1. Página de Inicio:

- Al acceder a la página principal (`index.html`), debe mostrar un formulario donde los usuarios pueden seleccionar:
 - o Nombre del usuario.
 - o Fecha de reserva (calendario).
 - o Espacio de trabajo (escritorio, sala de reuniones, oficina privada).
 - o Duración de la reserva (en horas).

2. Servlet para Procesar Reservas:

- El formulario enviado debe ser procesado por un **Servlet** llamado `ReservaServlet` .
- El servlet debe:
 - o Validar que todos los campos han sido completados.
 - o Guardar la reserva (no se requiere una base de datos real, se puede usar una **lista en memoria** para almacenar las reservas).
 - o Enviar una respuesta al cliente confirmando la reserva con los detalles ingresados.

3. Lista de Reservas:

- Crear un **Servlet** adicional llamado `ListaReservasServlet` que muestre la lista de todas las reservas almacenadas.
- Cada reserva debe mostrar:

- Nombre del usuario.
- Fecha de la reserva.
- Espacio reservado.
- Duración de la reserva.

4. Eliminación de Reservas:

- Desde la página de la lista de reservas, cada reserva debe tener un botón para eliminarla.
- Al hacer clic en el botón, se debe enviar una petición a un **Servlet** llamado `EliminarReservaServlet` que eliminará la reserva correspondiente de la lista.

5. Manejo de Errores:

- Si algún campo está vacío o la fecha es inválida, el sistema debe mostrar un mensaje de error amigable al usuario y permitirle corregir los datos.

Restricciones y Condiciones:

- Debes utilizar **Java Servlets** para manejar las peticiones HTTP.
- No se permite el uso de frameworks como Spring o Hibernate, pero puedes usar **JSP o HTML** puro para las vistas.
- El sistema debe ser desplegado en un servidor **Apache Tomcat**.
- Se permite el uso de listas en memoria (en lugar de bases de datos reales) para almacenar las reservas durante el tiempo de ejecución de la aplicación si así les parece más manejable.

Estructura del Proyecto

El proyecto debe tener la siguiente estructura:

```
bash

/src
  /webapp
    /index.html
    /reservas.jsp
  /java
    /servlets
      ReservaServlet.java
      ListaReservasServlet.java
      EliminarReservaServlet.java
```

Instrucciones Adicionales:

- El proyecto debe ser subido a un servidor **Tomcat** para que los usuarios puedan acceder a él desde un navegador.
- Asegúrate de que el formulario de reserva se procese correctamente y que los datos ingresados se muestren en la lista de reservas.
- Documenta tu código para que sea legible y fácil de entender.

Evaluación:

Criterio	Puntaje Máximo
Implementación de Servlets	10 puntos
Formulario de reserva y procesamiento	10 puntos
Lista de reservas con eliminación	10 puntos
Validación de campos y manejo de errores	7.5 puntos
Correcto despliegue en Tomcat	5 puntos
Estructura del código y comentarios	5 puntos
Uso adecuado de sesiones o manejo en memoria	2.5 puntos
Total	50 puntos

Recomendaciones para los Estudiantes:

1. Comienza configurando correctamente el entorno de desarrollo y asegúrate de que el proyecto funcione en Tomcat antes de implementar la lógica.
2. Ve paso a paso, empezando por el formulario y su procesamiento en el servlet.
3. No olvides hacer validaciones y manejar errores, esto es importante para evitar fallos durante la ejecución.
4. Asegúrate de probar bien tu aplicación con diferentes entradas.
5. Documenta bien tu código y sigue una estructura clara.

ENTREGA DE LA EVALUACION

Para asegurar una entrega clara y organizada de la evaluación, aquí están los pasos y las instrucciones detalladas que los estudiantes deben seguir al momento de entregar su trabajo:

1. Estructura del Proyecto

El proyecto debe entregarse siguiendo la estructura de carpetas requerida. Esto facilitará la corrección y el despliegue del proyecto. Asegúrate de que todas las carpetas y archivos estén correctamente organizados.

2. Instrucciones de Despliegue

El estudiante debe proporcionar un archivo README.md o README.txt con las **instrucciones de despliegue** del proyecto. Esto debe incluir:

- **Requisitos previos:** Indicar la versión de **Java** y **Tomcat** que se utilizó para el proyecto.
- **Pasos para el despliegue:** Explicar cómo clonar el proyecto y desplegarlo en un servidor Tomcat, ya sea en un entorno local o en una nube de desarrollo (si aplica).
- **Acceso a la aplicación:** URL predeterminada para acceder al formulario de reservas (por ejemplo, <http://localhost:8080/ProyectoReservaCoworking/index.html>).

3. Subida del Proyecto

El estudiante puede entregar el proyecto de las siguientes maneras:

- **Repositorio GitHub o GitLab:**

- El proyecto debe subirse a un repositorio de GitHub o GitLab, con acceso público o privado (si es privado, debe invitarse al profesor/corrector).
- El repositorio debe incluir el código fuente completo, archivos de configuración y el archivo README.md con las instrucciones de despliegue.
- El link del repositorio debe entregarse en el sistema de evaluación o por correo electrónico según las instrucciones del curso.

Ejemplo de URL a entregar: <https://github.com/usuario/ProyectoReservaCoworking>

- **Archivo ZIP:**

- El proyecto completo debe comprimirse en un archivo ZIP con el nombre del estudiante y el nombre del proyecto.
- Ejemplo de nombre del archivo ZIP:
NombreApellido_ProyectoReservaCoworking.zip
- El archivo ZIP debe incluir toda la estructura del proyecto junto con el archivo README.md con las instrucciones de despliegue.

4. Documentación

El proyecto debe incluir una **documentación mínima**, con comentarios en el código y la explicación de cómo se implementaron las principales funcionalidades. Esta documentación puede ser incluida como comentarios en el código o en el archivo README.md.

5. Criterios de Evaluación

La evaluación será calificada de acuerdo con los siguientes criterios:

- **Funcionamiento de las funcionalidades requeridas:**
 - Proceso de reserva.
 - Lista de reservas.
 - Eliminación de reservas.
- **Validación y manejo de errores:** Comprobación de campos vacíos y manejo de excepciones.
- **Uso adecuado de Java Servlets:** Correcta implementación de la lógica de negocio en los servlets.

- **Despliegue en Tomcat:** El proyecto debe desplegarse y ejecutarse correctamente en el servidor Tomcat.
- **Estructura y limpieza del código:** Código bien organizado, comentado y fácil de entender.
- **Documentación:** Incluyendo el archivo README.md y comentarios en el código.

6. Entrega Final

El estudiante debe entregar:

1. El enlace al repositorio GitHub/GitLab o el archivo ZIP comprimido.
2. El archivo README.md con las instrucciones de despliegue y funcionalidad.
3. El proyecto bien estructurado, con el código fuente y configuración para el servidor Tomcat.
4. Un breve informe sobre cómo se resolvió el problema y cualquier desafío encontrado durante el desarrollo.

Plazo de Entrega:

- Fecha límite: Miércoles 12 de marzo de 2025.
- Tiempo máximo: 4 horas desde el inicio de la evaluación.