

Documentación Técnica: Sistema de Gestión de Tareas

1. Descripción del Desarrollo

La solución se desarrolló siguiendo una arquitectura de **N-Capas**, separando la lógica de presentación (Angular), la capa de servicios/transporte (Java Servlets) y la persistencia (Oracle PL/SQL).

Decisiones de Diseño:

- **Backend:** Se optó por **Jakarta Servlets** para manejar el ciclo de vida de las peticiones HTTP y **JDBC** para una comunicación directa y eficiente con Oracle.
 - **Persistencia:** Se implementó lógica en el lado de la base de datos mediante el paquete TASK_PKG, lo que permite centralizar las reglas de negocio y optimizar el rendimiento de las consultas.
 - **Frontend:** Se utilizó **Angular 17** con componentes *Standalone* y formularios reactivos para una experiencia de usuario fluida y validaciones en tiempo real.
-

2. Documentación de Métodos (Backend - Java & PL/SQL)

Paquete PL/SQL: TASK_PKG

- **GET_ALL_TASKS:** Retorna un SYS_REFCURSOR con el conjunto completo de registros de la tabla TASKS.
- **CREATE_TASK:** Recibe el título y descripción; inserta el registro y retorna el TASK_ID generado por la secuencia.
- **UPDATE_TASK:** Actualiza los campos TITLE, DESCRIPTION y COMPLETED basándose en el ID proporcionado.
- **DELETE_TASK:** Realiza el borrado físico del registro mediante su identificador único.

Servlet API: TaskServlet

- **doGet(req, resp):** Maneja la obtención de todas las tareas o una específica (/api/tasks/{id}) transformando el SYS_REFCURSOR en una lista de objetos Java.

- **doPost(req, resp):** Lee el cuerpo de la petición en formato JSON (vía Gson), instancia un POJO Task y llama al procedimiento de creación.
 - **doPut(req, resp):** Procesa actualizaciones de tareas existentes, enviando los datos al procedimiento UPDATE_TASK.
 - **doDelete(req, resp):** Extrae el ID del *path* de la URL y ejecuta la eliminación en la base de datos.
-

3. Documentación de Métodos (Frontend - Angular)

Servicio: TaskService

- **getAllTasks():** Utiliza HttpClient.get para obtener el array de tareas. Implementa catchError para gestionar fallos de red.
- **createTask(task: Task):** Envía un POST con el objeto tarea. El modelo utiliza taskId para mantener paridad con el backend.
- **updateTask(taskId, task):** Realiza una petición PUT enviando el identificador y los datos actualizados.
- **deleteTask(taskId):** Ejecuta un DELETE hacia el endpoint del servidor y retorna un observable para confirmar la operación.

Lógica de Componentes

- **TaskListComponent:** Implementa el refresco automático de la lista tras cada operación CRUD mediante suscripciones a observables.
 - **TaskFormComponent:** Gestiona el estado de edición/creación. Incluye validaciones reactivas donde el campo title es obligatorio para habilitar el guardado.
-

4. Desarrollo de la Prueba y Solución de Problemas

Durante el desarrollo se abordaron y solucionaron los siguientes desafíos técnicos:

1. **Configuración de CORS:** Se implementaron encabezados manuales en los Servlets (Access-Control-Allow-Origin) para permitir la comunicación segura entre el puerto 4200 (Angular) y 8080 (Tomcat).

2. **Sincronización de Modelos:** Se unificaron las interfaces de TypeScript con los POJOs de Java utilizando la propiedad `taskId`, asegurando que la serialización con Gson fuera transparente.
3. **Manejo de Tipos en Oracle:** Se realizó el mapeo manual del campo `COMPLETED` de `NUMBER(1,0)` en SQL a `boolean` en Java/Angular para mejorar la usabilidad del frontend.
4. **Imágenes del software**

img-1

The screenshot shows a web application titled "Gestión Inteligente de Tareas" with a subtitle "Organiza, edita y controla tus tareas de forma rápida y eficiente." The interface includes a blue header with "Task Manager" and "Tareas" links. A central modal form titled "Nueva Tarea" contains the following fields:

- Título ***: A text input field with the placeholder "Nombre de la tarea".
- Descripción**: A text area with the placeholder "Opcional...".
- Tarea Completada**: A radio button option.
- Buttons**: "Guardar" (with a save icon) and "Cancelar".

At the bottom of the page, the text "Prueba Técnica Sysman" is visible.

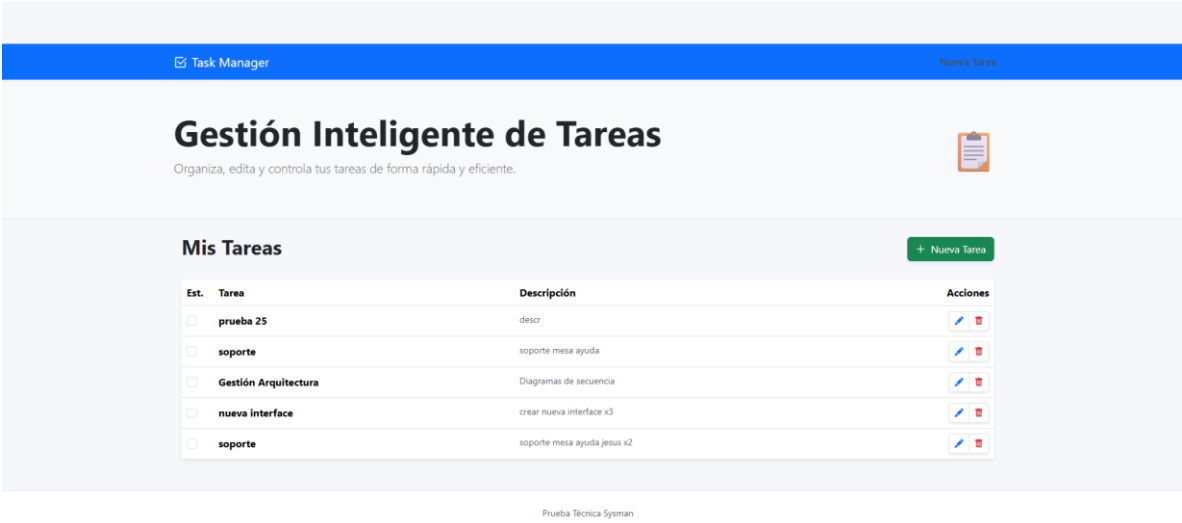
Img-2

This screenshot shows the same "Gestión Inteligente de Tareas" application, but the "Nueva Tarea" form is populated with sample data:

- Título ***: "Gestión Arquitectura"
- Descripción**: "Diagramas de secuencia"
- Tarea Completada**: The radio button is selected.
- Buttons**: "Guardar" and "Cancelar" are visible.

The page footer also displays "Prueba Técnica Sysman".

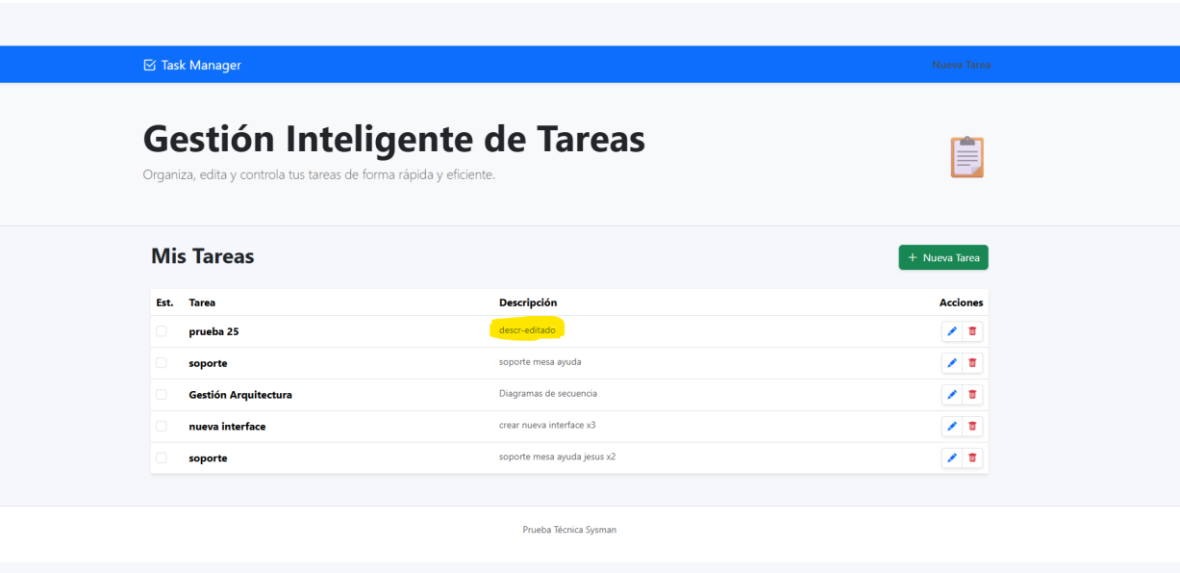
Img-3



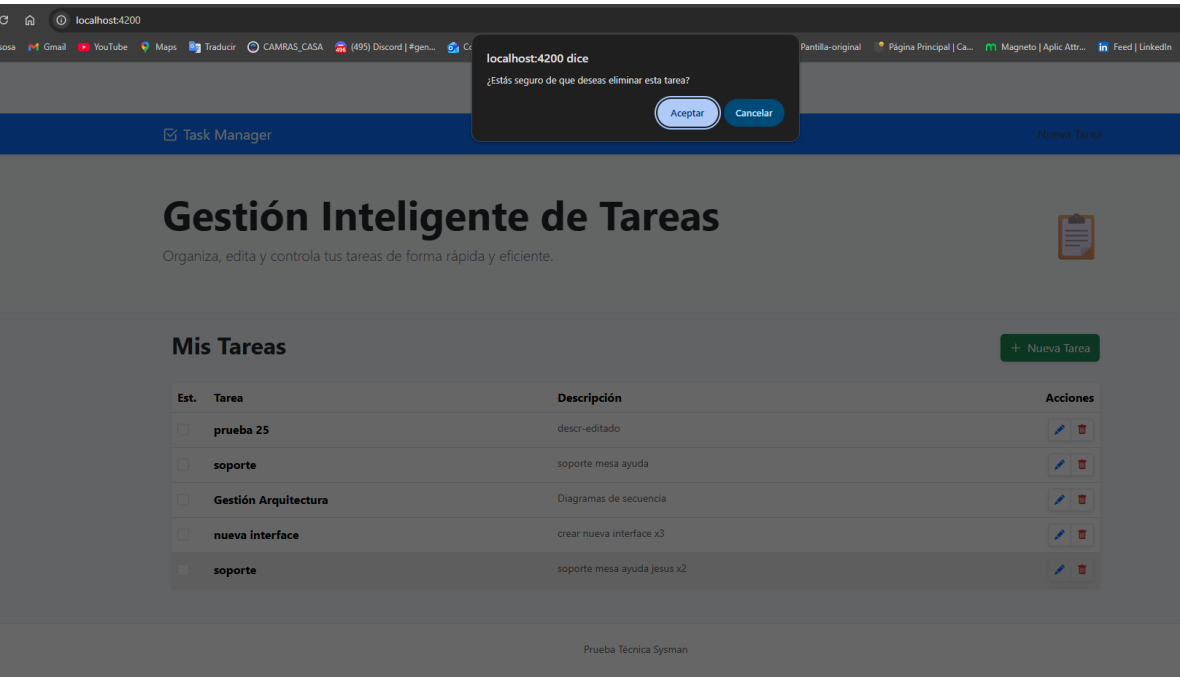
Img-4



Img-5



img-6




img-7

Task Manager

Nueva Tarea

Gestión Inteligente de Tareas

Organiza, edita y controla tus tareas de forma rápida y eficiente.



Mis Tareas

+ Nueva Tarea

Est.	Tarea	Descripción	Acciones
<input type="checkbox"/>	prueba 25	descr-editado	<div><div></div><div></div></div>
<input type="checkbox"/>	soporte	soporte mesa ayuda	<div><div></div><div></div></div>
<input type="checkbox"/>	Gestión Arquitectura	Diagramas de secuencia	<div><div></div><div></div></div>
<input type="checkbox"/>	nueva interface	crear nueva interface x3	<div><div></div><div></div></div>

Prueba Técnica Sysman