

Examen AJAX y Almacenamiento

Se evaluará:

- El grado de implementación de la funcionalidad que se solicita en cada ejercicio.
- Código claro y eficiente:
- Se resuelve de forma óptima
- Uso adecuado de los elementos dados en el tema.

Instalación de la API REST

Descomprime la carpeta “examenProyectos”.

Dentro de la carpeta ejecuta:

➤ npm i

➤ npm run dev

Instalación de colección de consultas en POSTMAN

- Abre el POSTMAN
- Importa (File->import)
“DSW2024-Ex6-Projects.postman_collection.json”

En la siguiente URL tienes documentación de la API:

<https://documenter.getpostman.com/view/2851145/2sAYQghoVL>

Instalación de archivos base

Descomprime la carpeta. En la carpeta public tienes la parte del cliente y donde debes hacer el examen. Será la carpeta que me tienes que entregar.

Realiza los siguientes apartados:

Examen Temas 5 y 6 | 2024-2025

Lista de enlaces

- [Empleados](#)
- [Proyectos](#)
- [Crear Tarea](#)
- [Costes](#)

En employee.html :

1. Se muestra el listado de empleados.
 - a. Se muestra con su nombre, cargo y coste.
2. Cuando se pulsa sobre un empleado, se debe marcar. Clase “marked”.
 - a. Si se pulsa uno marcado, se debe desmarcar.
3. Se debe almacenar en almacenamiento local los usuarios marcados mientras dura la sesión.
 - a. Si se vuelve a visitar la página, se deben visualizar los usuarios marcados anteriormente.

Lista de empleados

| |
|---|
| Alice Johnson Developer 30€/hora |
| Bob Smith Designer 25€/hora |
| Charlie Brown Tester 20€/hora |
| Diana Prince Project Manager 35€/hora |
| Eve Adams Developer 30€/hora |
| Frank Wright Designer 25€/hora |
| Grace Hopper Developer 30€/hora |
| Hank Pym Tester 20€/hora |
| Ivy Green Developer 30€/hora |
| Jack White Designer 25€/hora |

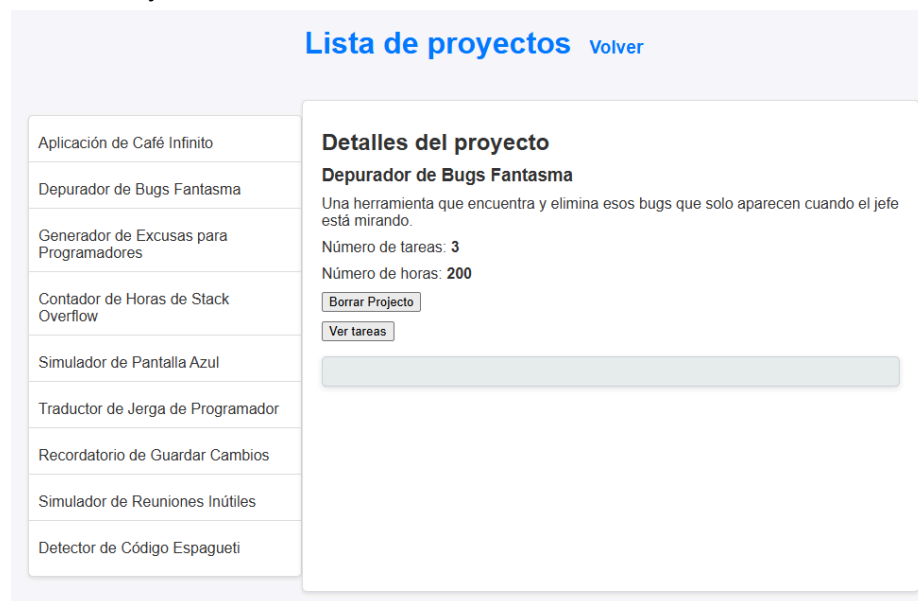
En project.html :

4. Listado de proyectos :
 - a. Muestra en la lista con id = “projectList” el listado de todos los proyectos.
 - b. Obten los proyectos mediante la consulta a la API Rest.
 - c. Muestra solo el nombre del proyecto.



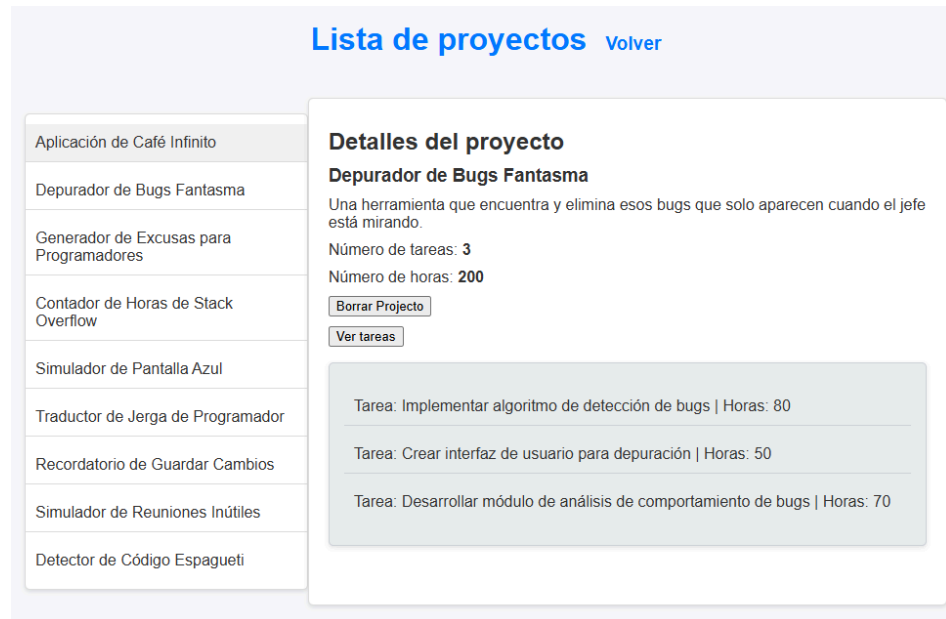
5. Detalles del proyecto:

- Cuando un usuario pulsa en un proyecto se debe mostrar en la div con id="detailProject" los detalles del proyecto. Esto incluye: name, description, taskCount y totalHours.



6. Tareas:

- Además de los detalles del proyecto se debe mostrar un listado de las tareas cuando se pulsa el botón ver tareas.
- El botón debe desaparecer si no hay tareas.
- Se debe mostrar el nombre de la tarea y la duración de la tarea.
- Si se elige otro proyecto, asegúrate de borrar el listado anterior.



7. Borrar proyecto:
 - a. Cuando se pulsa el botón borrar. Se debe borrar el proyecto.
 - b. Si el proyecto tiene tareas, la API devolverá un error.
 - i. *El proyecto 7 - Generador de Nombre de Variables no tiene tareas.*
 - c. Mostrar un mensaje de éxito o error, mediante un alert.
 - d. Si ha tenido éxito, se debe borrar del listado y ocultar el panel de detalle (clase hidden).

En create_tarea.html :

8. El formulario de creación de tareas debe permitir al usuario ingresar el concepto y la duración de la tarea, y seleccionar un proyecto.
 - a. Al cargar la página, se debe realizar una solicitud a la API para obtener el listado de proyectos.
 - b. Los proyectos deben mostrarse en el campo de selección del formulario de creación de tareas.
9. Crear tarea
 - a. Al enviar el formulario, se debe realizar una solicitud a la API para crear una nueva tarea.
 - b. Si la tarea se crea correctamente, mostrar el mensaje: "Tarea creada con éxito." y borrar los datos del formulario.
 - c. Si ocurre un error, mostrar el mensaje: "Error en la petición."

Crear Tarea

[Volver](#)

Concepto:

Duración:

Proyecto:

Aplicación de Café Infinito
▼

Crear Tarea

Tarea creada con éxito

En cost.html : (voluntario - nota extra)

10. La página muestra los costes totales de los empleados marcados
 - a. Se muestran los empleados marcados anteriormente en la página empleados.
 - b. Se muestran todas las tareas asignadas a cada empleado.
 - c. Se calcula el coste de cada empleado.
 - d. Se calcula el coste total.

| Costes Volver | | | |
|---|--|---|------------------|
| Alice Johnson | Charlie Brown | Hank Pym | Total: |
| Cargo: Developer Coste: 30€/hora | Cargo: Tester Coste: 20€/hora | Cargo: Tester Coste: 20€/hora | 33300.00€ |
| <ul style="list-style-type: none"> Implementar notificaciones push: 20 horas Optimizar consumo de batería: 25 horas Implementar algoritmo de detección de bugs: 80 horas Desarrollar módulo de análisis de comportamiento de bugs: 70 horas Optimizar rendimiento del generador de excusas: 25 horas Desarrollar diccionario de jerga: 700 horas Optimizar rendimiento del traductor: 60 horas Analizar código para detectar espaguetis: 50 horas Refactorizar módulos con nombres graciosos: 40 horas | <ul style="list-style-type: none"> Desarrollar generador de excusas: 15 horas Realizar pruebas de usabilidad: 40 horas | <ul style="list-style-type: none"> Simular reunión de 3 horas sobre nada: 3 horas Crear agenda de reuniones inútiles: 2 horas | |
| Coste empleado: 32100.00€ | Coste empleado: 1100.00€ | Coste empleado: 100.00€ | |