

Proyecto primera evaluación DIN semipresencial

Crear una aplicación y documentar el proyecto.

Fecha límite de presentación **4/12/2024** a las **23:59**.

En caso de no presentar o no superar el proyecto se realizará el examen de la primera evaluación en la fecha establecida.

Repositorio de *GitHub Classroom*: <https://classroom.github.com/a/NNH2WQk5>

Crear una aplicación utilizando Electron y React

Se creará una aplicación de escritorio que permita una gestión básica de una lista de tareas simples, utilizando Electron y React, con las siguientes características:

Vista principal:

- **Título de la ventana:** Listado de tareas.
- Listado de tareas creadas.
- Elemento para ordenar el listado por "título", "estado" o "fecha".
- Elemento que permita mostrar/ocultar tareas "Completadas" y / o "Canceladas" - Cada ítem de la lista tendrá los siguientes elementos:
 - o Título.
 - o Estado. Pulsando permitirá cambiar el estado, entre "Pendiente", "En proceso", "Completada", "Cancelada".
 - o Fecha límite.
 - o Botón editar. Abre la vista de edición.
 - o Botón borrar. Muestra un dialogo de confirmación, en caso afirmativo, borra la tarea y actualiza la lista.
- Botón para crear una nueva tarea.
- Abrir la vista de detalle de una tarea.

Vista de creación:

- **Título de la ventana:** Nueva tarea
- Se mostrará una vista con un formulario que permita crear una nueva tarea. El formulario tendrá los siguientes elementos:
 - o **Título:** Requerido.
 - o **Descripción:** Opcional. Bloque de texto.
 - o **Estado:** Requerido. Seleccionar entre "Pendiente", "En proceso", "Completada", "Cancelada"
 - o **Fecha límite:** Opcional. Por defecto, siete días desde la fecha actual.
 - o Botón **Guardar**. Almacena la tarea y vuelve al listado.
 - o Botón **Descartar**. Muestra un dialogo de confirmación, en caso afirmativo, vuelve al listado sin guardar cambios.

Vista de edición:

- **Título de la ventana:** Edición de {título de la tarea}
 - Donde {título de la tarea} es el título que tiene la tarea que se está editando.
- Se mostrará un formulario **similar a la vista de creación**, con los datos rellenos. Tendrá las siguientes características:
 - No aparece el botón descartar.
 - Botón **Borrar**. Muestra un dialogo de confirmación, en caso afirmativo, borra la tarea y vuelve al listado.
 - Botón **Guardar**. Se activa si se han hecho cambios.
 - Botón **Volver**. Si hay cambios sin guardar, mostrará un dialogo de confirmación indicando si queremos guardar los cambios, descartarlos o cancelar la acción.
 - **Guardar**, guarda cambios y vuelve al listado.
 - **Descartar**, no guarda cambios y vuelve al listado.
 - **Cancelar**. Cierra el dialogo.

Documentación

Se creará un fichero de texto **Readme.md** en la raíz del proyecto con la siguiente información:

- Nombre del proyecto.
- Nombre y apellidos.
- Detalles de implementación, decisiones y problemas encontrados.

Seguimiento

El repositorio de *GitHub Classroom* debe reflejar el progreso, para ello:

- Se deben realizar al menos 10 commits.
- Los commits deben ser concisos e informativos.
- Durante el desarrollo del proyecto, se realizarán reuniones de seguimiento para comprobar el progreso.

Vídeo demo de la aplicación

Se recomienda hacer vídeo demostrativo del correcto funcionamiento de la aplicación, así como con las explicaciones oportunas de los detalles de implementación, decisiones y problemas encontrados.

Este vídeo tendrá entre 2 y 5 minutos de duración.

Puntos a valorar:

- **Diseño** de las vistas (3 puntos):
 - Se diseñan todas las vistas. ○ Se incluyen todos los elementos. ○ Se seleccionan elementos apropiados. ○ Se utilizan colores con contrastes adecuados que permiten leer los textos. ○ Se presenta *feedback* en elementos necesarios (botones, inputs, etc.). ○ Se controlan los errores en el formulario.
 - Se hace un buen uso de librerías o CSS.
- **Componentes** (3 puntos):
 - Se crean al menos tres componentes personalizados.
 - Se reutilizan adecuadamente los componentes creados.
- **Funcionalidad** (2 puntos):
 - Se gestiona el estado de la aplicación. ○ Se implementan las funcionalidades.
 - Se implementa la persistencia de datos.
- **Implementación** (2 puntos) ○ El código está bien estructurado, tiene buena legibilidad y permite realizar cambios.
 - Se han hecho al menos 10 *commits*, en el repositorio que demuestren la progresión del proyecto.
 - Se indican mensajes claros y concisos de los *commits* realizados. ○ Se aporta una buena documentación.

En caso de detectar alguna irregularidad, se os convocará una defensa del proyecto de manera presencial.