

# Trabajo Práctico N.º 3

## *Unidad 1*

### **Economía empresarial 2025**

Profesora: Boldrini Carolina

Alumno: Romeo Gago Lucia



## Análisis de conceptos en proyecto a desarrollar

### Proyecto:

En grupo realizar una aplicación para la administración de una lista de tareas que permite agregar, editar y eliminar tareas y donde cada elemento además del texto de la tarea tendrá un valor de prioridad (valor del 1 al 5), fecha de creación y fecha de modificación. Además, la aplicación deberá permitir la opción de ordenarlas ascendentemente por prioridad o fecha de modificación.

### Entregar informe que contenga:

- Descripción del proyecto: definir los objetivos, requisitos de la funcionalidad y alcance
- Planificación y diseño: estrategia utilizada y la justificación de las decisiones tomadas (especificar los recursos disponibles: tiempo, herramientas y personal).
- Implementación: proceso de desarrollo, retos encontrados y cómo fueron superados (documentar el proceso de desarrollo).
- Análisis:
  - Eficiencia: comparar el tiempo estimado y los recursos utilizados con los resultados obtenidos. ¿Fue óptimo su uso? ¿Podría haber sido más eficiente?
  - Parámetro de comparación: 2 hs, 2 programadores.
  - Eficacia: analizar si cumplieron los objetivos y como se logro
  - Productividad: medir la cantidad de funcionalidad desarrollada en relación con los recursos utilizados. (ej. Líneas de código x hora)
  - ¿Fue el tiempo y esfuerzo bien utilizado?
  - Compara con alternativas posibles

## Descripción del proyecto:

### Objetivos:

- Crear una aplicación web que funcione como lista para hacer tareas (To-Do List).
- La lista debe permanecer visible hasta que el usuario haya terminado con ella.
- El usuario debe poder tachar (completar) o eliminar tareas de la lista.
- Cada tarea debe ser introducida con su nivel de prioridad, de 1 a 5.
- Cada tarea debe almacenar su fecha de creación y fecha de modificación.
- La lista debe poder organizarse ascendentemente por prioridad o por fecha de modificación.

### Alcance:

La aplicación debe ser útil para cualquier persona con conocimientos básicos de informática, siendo intuitiva y transparente en su uso.

### Planificación y diseño:

- Estrategia utilizada:

- Se partió de un diseño funcional simple, priorizando la claridad y la facilidad de uso, y se fue evolucionando hacia una interfaz más profesional y moderna, como la mostrada en la imagen adjunta.
- Se decidió utilizar únicamente tecnologías web estándar: HTML, CSS y JavaScript puro
- Se implementó una estructura con componentes visuales:
  - Pestañas de tareas “Completadas” y “Sin Completar”.
  - Un botón verde (+) destacado y accesible que permita agregar la tarea.
  - Doble clic para editar tarea.
  - Botón (x) para eliminar tarea.
  - Se priorizo la experiencia del usuario, para que las acciones necesarias (agregar, modificar, completar, eliminar y ordenar) sean intuitivas y esten diferenciadas de manera correspondiente.

#### Recursos disponibles:

- Tiempo estimado: 2 horas.
- Equipo: un programador.
- Herramientas: editor de texto, navegador, recursos de diseño web.

#### Implementación

##### Proceso de desarrollo:

1. Estructura inicial:  
Se creó el HTML básico con campos para ingresar tareas y un botón para agregarlas.  
Luego, se agregó la necesidad de que cada tarea tuviera una prioridad.
2. Evolución de la estructura de datos:  
Las tareas pasaron de ser simples textos a objetos con propiedades: texto, prioridad, fechaCreacion, fechaModificacion, completada.  
Este cambio permitió manejar mejor la información y facilitar funcionalidades avanzadas.
3. Funcionalidades principales:
  - Agregar tareas con prioridad y fechas.
  - Editar tareas con doble clic (actualizando fecha de modificación).
  - Eliminar tareas con botón (x).
  - Completar tareas con checkbox circular.
  - Ordenar tareas por prioridad o fecha de modificación.
  - Separar tareas en pestañas según el estado, "Sin Completar" y "Completada".

### Retos encontrados y soluciones:

- Cambio de estructura de tarea:  
El paso de manejar tareas como texto simple a objetos completos requirió refactorizar la lógica de renderizado y manipulación de datos.  
Se modificó la mayoría del código javascript para manejar propiedades adicionales y actualizar fechas.

### Conflictos de eventos:

- Inicialmente, el clic en la tarea podría marcarla como completada o interferir con la edición por doble clic.  
Es decir, si un usuario quería modificar la tarea, el primer click interferiría, haciendo que la tarea fuera chequeada como completada, cuando no era esa su consecuencia deseada.
- Se asignó el cambio de estado solo al checkbox, la edición solo al doble clic, y la eliminación solo al botón cerrar, evitando interferencias.

### Ordenamiento y filtrado:

- Implementar el ordenamiento sin mezclar tareas sin completar y completadas requirió lógica adicional para filtrar y ordenar por pestaña.

¿Qué tipo de lógica requirió?

Explicación:

1. La estructura del dato: Cada tarea es un objeto con la propiedad completada (booleano) que indica si la tarea está completada o no.

```
{
  texto: "...",
  prioridad: 3,
  fechaCreacion: "...",
  fechaModificacion: "...",
  completada: false // o true
}
```

2. Renderizado separado por pestañas: se creó la función renderizarTareas() donde se filtran las tareas y se agregan a la lista correspondiente. Se separa claramente la visualización según el estado completada:

```
tareas.forEach((tarea, idx) => {
  // ... crear elemento li ...
  if (tarea.completada) {
    listaCerrado.appendChild(li);
  } else {
    listaAbierto.appendChild(li);
  }
});
```

3. Función de ordenamiento: función ordenarLista(criterio) ordena el array completo tareas según el criterio:

```
function ordenarLista(criterio) {
  if (criterio === 'prioridad') {
    tareas.sort((a, b) => a.prioridad - b.prioridad);
  } else if (criterio === 'fechaModificacion') {
    tareas.sort((a, b) => new Date(a.fechaModificacion) - new Date(b.fechaModificacion));
  }
}
```

```

    }
    renderizarTareas();
  }

```

La lógica para no mezclar tareas activas y completadas se implementa en el renderizado, no en el ordenamiento. El array tareas está ordenado globalmente, pero en `renderizarTareas()`, las tareas se distribuyen en dos listas separadas según completada. Esto significa que aunque el array está ordenado globalmente, la separación visual en pestañas mantiene las tareas sin mezclar.

Diseño responsivo:

- Adaptar el diseño para que se vea bien en dispositivos móviles y escritorio implicó el uso de flexbox y media queries. Esto fue contemplado desde un principio, debido a la utilidad que ha sido utilizada por el programador anteriormente en otros proyectos.

Análisis

- Eficiencia
  - Tiempo estimado: 2 horas.
  - Tiempo real: El desarrollo fue de 4 horas al tiempo planificado.
  - Recursos estimados: 2 programadores, herramientas estándar.
  - Recursos reales: 1 programador, herramientas estándar.

Comparación: Si bien se duplicó el tiempo estimado, el trabajo fue realizado con recursos limitados. El uso de frameworks podría haber acelerado el desarrollo, pero no era necesario para el alcance.

- Eficacia
  - Cumplimiento de objetivos:  
Todas las funcionalidades requeridas están implementadas y accesibles desde una interfaz moderna y clara.
  - El usuario puede agregar, editar, eliminar, completar y ordenar tareas fácilmente, y navegar entre tareas abiertas y cerradas.

Cómo se logró:

Mediante:

- Iteraciones rápidas.
- Pruebas constantes.
- Separación clara de responsabilidades en el código y la interfaz.

Productividad

- Líneas de código por hora:
  - Aproximadamente 200 líneas de código (HTML, CSS y JS) en 4 horas, incluyendo comentarios y refactorizaciones.

- Funcionalidad desarrollada:
  - Todas las funciones principales y el diseño visual (añadido en la carpeta de github) fueron completados dentro del tiempo y recursos asignados.
- Comparación con alternativas:
  - Usar librerías como React o Bootstrap podría haber facilitado algunos aspectos, pero habría aumentado la complejidad y el tamaño del proyecto.

¿Fue el tiempo y esfuerzo bien utilizado?

Sí, el tiempo fue bien invertido, logrando un producto funcional, moderno y fácil de mantener, sin dependencias externas.

## Conclusión

El proyecto cumple con todos los objetivos funcionales y de diseño planteados. La aplicación es intuitiva, profesional y eficiente, tanto en su desarrollo como en su uso. El cambio de estructura de datos permitió implementar funcionalidades avanzadas y el diseño final es visualmente atractivo, como muestra la imagen de referencia. Los retos de usabilidad y ordenamiento se resolvieron con una clara separación de eventos y lógica adicional para filtrar y ordenar tareas según estado.

## Resultados

- Página de tareas sin completar.

The screenshot displays a web application interface for task management. At the top, there are two tabs: 'Sin Completar' (active) and 'Completadas'. A green '+' button is located to the right of the 'Completadas' tab. Below the tabs, there is a search bar labeled 'Nueva tarea...', a dropdown menu for 'Prioridad', and two buttons: 'Ordenar por Prioridad' and 'Ordenar por Fecha'. The main content area shows a list of five tasks, each with a radio button, the task description, the priority level, and a red 'x' icon. The tasks are: 'Comprar recursos (tecnología) (Prioridad: 2)', 'Pagar a empleados (Prioridad: 4)', 'Modificar el calendario (Prioridad: 3)', 'Reunión con cliente nuevo (Prioridad: 5)', and 'Revisar nueva característica del software (Prioridad: 1)'. At the bottom, a note states: 'Para modificar una tarea, haga doble clic sobre ella.'

- Lista de tareas sin completar organizadas por su prioridad, de forma ascendente.

The screenshot shows a task management interface with two tabs: 'Sin Completar' (Active) and 'Completadas' (Completed). A green '+' button is in the top right. Below the tabs are input fields for 'Nueva tarea...' and 'Prioridad'. Two buttons, 'Ordenar por Prioridad' and 'Ordenar por Fecha', are present. A list of five tasks follows, each with a radio button, a description, a priority level, and a delete icon (red 'x').

Radio Button	Task Description	Priority	Delete Icon
<input type="radio"/>	Comprar recursos (tecnología)	Prioridad: 2	✗
<input type="radio"/>	Modificar el calendario	Prioridad: 3	✗
<input type="radio"/>	Pagar a empleados	Prioridad: 4	✗
<input type="radio"/>	Revisar nueva característica del software	Prioridad: 4	✗
<input type="radio"/>	Reunión con cliente nuevo	Prioridad: 5	✗

Para modificar una tarea, haga **doblo clic** sobre ella.

- Lista de tareas sin completar organizadas por su fecha de creación/modificación, de forma ascendente.

**Sin Completar** **Completadas** +

Nueva tarea...

Prioridad ▼

**Ordenar por Prioridad**

**Ordenar por Fecha**

- ☐ Comprar recursos (tecnología) (Prioridad: 2) ✕
- ☐ Pagar a empleados (Prioridad: 4) ✕
- ☐ Modificar el calendario (Prioridad: 3) ✕
- ☐ Reunión con cliente nuevo (Prioridad: 5) ✕
- ☐ Revisar nueva característica del software (Prioridad: 5) ✕

Para modificar una tarea, haga **doble clic** sobre ella.

- Lista de tareas completadas.

**Sin Completar** **Completadas** +

Nueva tarea...

Prioridad ▼

**Ordenar por Prioridad**

**Ordenar por Fecha**

- ☒ Pagar a empleados (Prioridad: 4) ✕
- ☒ Reunión con cliente nuevo (Prioridad: 5) ✕

Para modificar una tarea, haga **doble clic** sobre ella.

- Lista de tareas sin completar, la visual de modificar tareas, al hacer doble click sobre ella.



Esta página dice

Editar tarea:

Comprar recursos (tecnología)

Aceptar

Cancelar

Ordenar por Prioridad

Ordenar por Fecha

☐

Comprar recursos (tecnología) (Prioridad: 2)

x

☐

Modificar el calendario (Prioridad: 3)

x

☐

Revisar nueva característica del software (Priori..x)

x

Para modificar una tarea, haga **doblo clic** sobre ella.