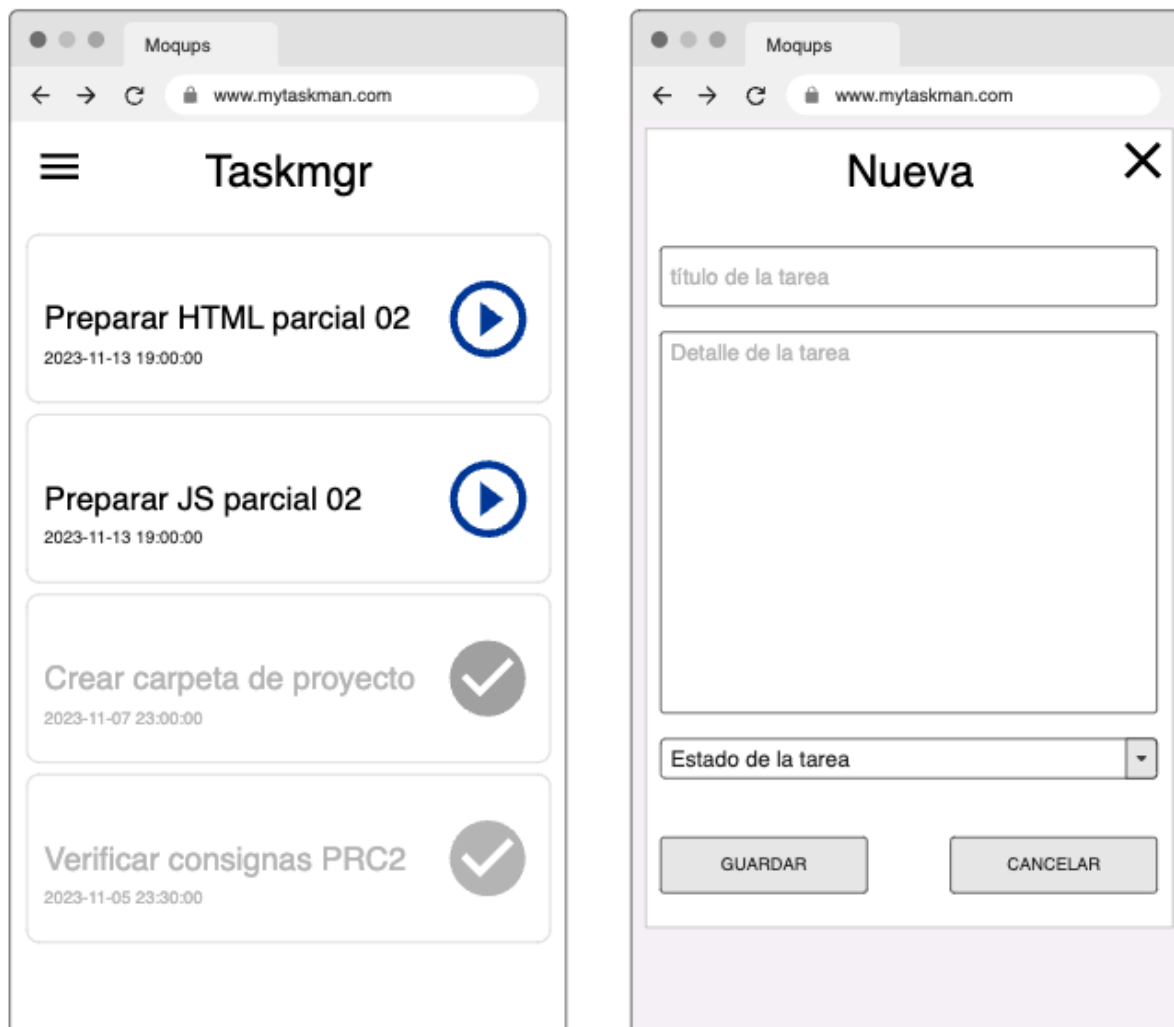


## Consigna para el Parcial 02



*Hay dos pantallas más, en el detalle del documento.*

**Fecha de entrega:** *hasta el día 19/11/2024 - 18 hs.*

Recuerden que tengo que revisar todos los proyectos y publicar las notas de uds, antes de las 00 hs del día siguiente, por lo tanto, estaré destinando el tiempo de la clase para realizar esta labor.

Por favor, lean todo el documento completo para conocer bien el objetivo del proyecto.

## Objetivo

**Desarrollar una aplicación para gestionar tareas.** En las imágenes está el nombre sugerido, pero por supuesto cada uno puede innovar y ponerle un nombre diferente.

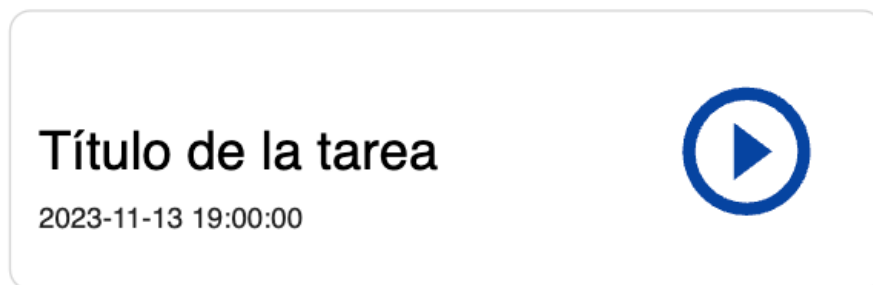
La aplicación web integrará la funcionalidad de la plataforma **MOCKAPI.IO** utilizada anteriormente para el proyecto del primer parcial. Debemos tomar la siguiente estructura de datos JSON que se almacenará.

```
{
  id: "1",
  fechacreacion: "2023-10-10 15:21:00",
  fechaconclusion: "",
  titulo: "Título de la tarea",
  descripcion: "Descripción más extensa
                de la tarea a realizar",
  estado: "pendiente"
}
```

## Descripción de las funcionalidades esperadas

### Pantalla principal

Al ingresar a la aplicación se visualizarán todas las tareas cargadas, **ordenadas por fecha de forma descendente**. Primero las no finalizadas luego las del estado tipo **finalizado**. Cada tarea estará representada en una Card horizontal, con el estilo similar al que se visualiza a continuación:



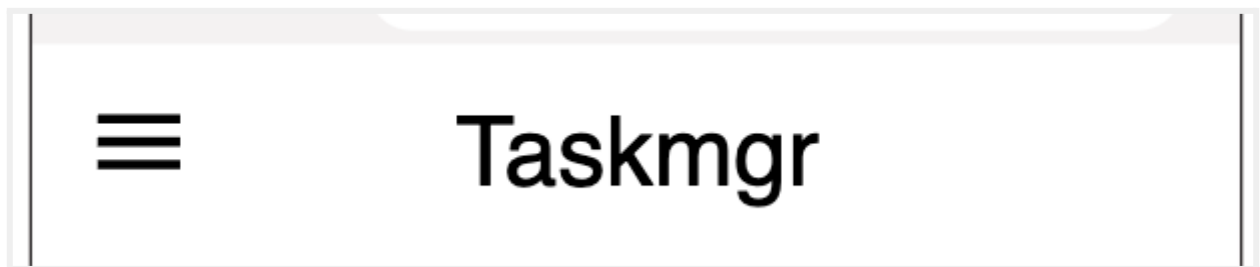
Se debe visualizar el título de la tarea y la fecha y hora de creación. Del lado derecho se verá un botón del tipo Play, el cual, permitirá reproducir en formato audio el detalle de la tarea cargada. La funcionalidad del botón play se agregará al final de este documento en el transcurso de la semana.

Sugerimos que, si se visualiza la aplicación en una ventana de navegador web desktop, elijan una configuración de Card del tipo múltiples tarjetas en formato horizontal, luego vertical, aplicando así el modelo responsivo que consideren más apropiado.

## Menú principal

Se debe contar con un menú. El mismo puede ser en formato menú sandwich que despliega un panel lateral, un menú estilo navigation bar (*horizontal superior*), o un menú footer bar en el pie de página del documento.

**Este punto queda a criterio de c/u de cual es la mejor forma de aplicarlo.**



Este menú debe dar acceso a:

- crear una nueva tarea
- cambiar el estado de una tarea existente
- configurar las bases de la aplicación

Cada sección puede estar creada en un documento HTML dedicado o en un elemento HTML del tipo **<modal>**. Este punto también es a libre elección de c/u.

## Crear una nueva tarea

El documento HTML / Panel emergente para crear una nueva tarea, debe permitir cargar:

- un título
- un detalle de la tarea
- un estado inicial predeterminado
  - los estados los define c/u
    - pueden ser 2 estados, 3 estados, o más

Al guardar la tarea, se debe almacenar también la fecha y hora de creación.

```
{
  id: "1",
  fechacreacion: "2023-10-10 15:21:00",
  fechaconclusion: "",
  titulo: "Título de la tarea",
  descripcion: "Descripción más extensa
               de la tarea a realizar",
  estado: "pendiente"
}
```

Moqups

← → ↻

🔒 www.mytaskman.com

Cambiar estado

✕

Preparar HTML parcial 02

Analizar el documento funcional compartido en la plataforma para comenzar a construir el código HTML y CSS base del proyecto correspondiente al segundo parcial.

2023-11-13 19:00:00

Por hacer

▼

GUARDAR

CANCELAR

Moqups

← → ↻

🔒 www.mytaskman.com

Configuración

✕

Seleccionar idioma

Paulina - es-MX

▼

Velocidad de audio

-112

## Cambiar estado de una tarea

Se debe permitir cambiar el estado de una tarea por cualquier otro estado posible que hayan definido. Uno de los estados a definir, debe corresponder a la tarea concluída/finalizada/o como gusten llamarlo.

Cuando se seleccione dicho estado, se debe mostrar algún indicador sobre la Card correspondiente a la tarea como, por ejemplo, un ícono del tipo checked como se muestra de ejemplo a continuación, o un texto como alternativa en forma de componente 'chip'. Si van por esta última, sugiero manejen colores.



A toda tarea marcada como **concluída**, se le puede aplicar el estilo que consideren apropiado. Por ejemplo:

- Card con un estilo grisado en los textos
- Check en formato ícono, o texto del tipo componente 'chip'

Junto con el cambio de estado, también se debe guardar fecha y hora si la tarea se pasa al estado concluído.

```
{
  id: "1",
  fechacreacion: "2023-10-10 15:21:00",
  fechaconclusion: "2023-10-30 21:21:01",
  titulo: "Título de la tarea",
  descripcion: "Descripción más extensa
                de la tarea a realizar.",
  estado: "completada"
}
```

Es importante que toda tarea marcada como concluída, se visualice de forma ordenada, descendente, luego de las tareas que no han sido concluídas.

El diseño de la pantalla para cambiar el estado de una tarea, es tentativo, y cómo llegar a la tarea a la cual le quieren cambiar el estado, también es a elección de cada uno.

## Configuración

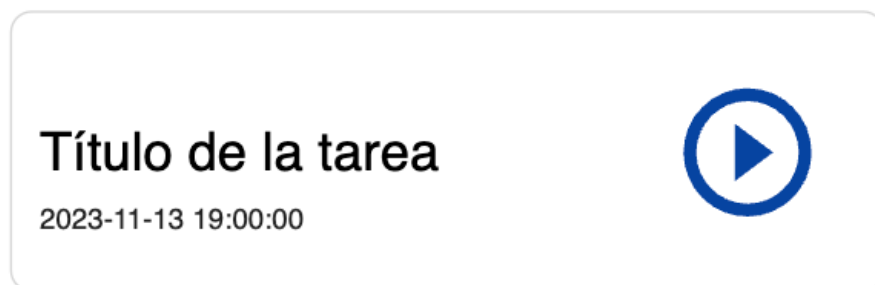
Se sugiere crear un panel emergente tipo **<modal>** o un documento HTML dedicado donde se pueda establecer una configuración para la aplicación. Este panel deberá tener un combo **<select>** donde se listarán datos que vendrán desde una API JS.

También un elemento HTML del tipo **input type range**, o slider, donde se pueda especificar:

- valor mínimo: -0.9
- valor predeterminado: 1
- valor máximo: 1.2

## Reproducir tarea

El botón Play que figura en cada una de las tareas, permitirá reproducir el detalle de la misma en formato audio.



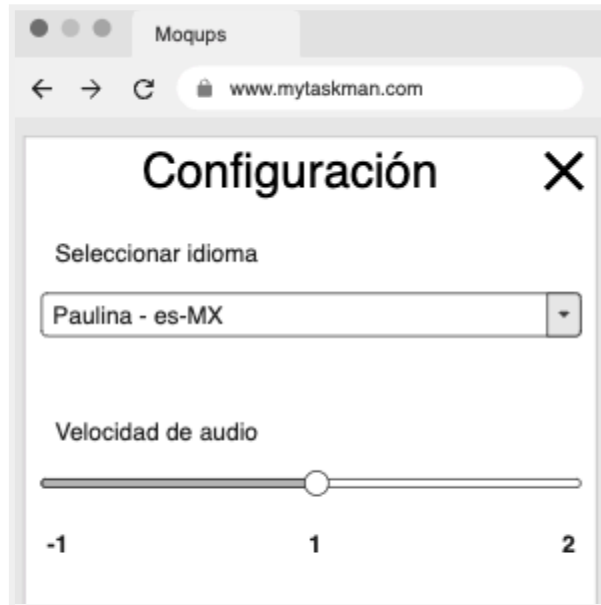
La aplicación de tareas debe utilizar la API de **Speech Synthesis** a través de la librería JS que vimos en clase, detectando cuál es la tarea agendada que el usuario desea escuchar.

Al pulsar el botón **Play** se debe reproducir en formato audible el detalle de la tarea elegida.

- Puede reproducirse el título y el detalle de la tarea, o solamente el detalle. Queda a elección de c/u.
- También es opcional identificar mediante un ícono, color, o cualquier otro estilo visual la tarea que se está reproduciendo en formato audio.
- No se requiere poder pausar o detener la reproducción en curso.

Los parámetros para la funcionalidad de reproducción de audio se deben tomar desde la configuración establecida en la pestaña o pantalla **“Configuración”** de la aplicación web: (*idioma del audio, velocidad de reproducción*).

La configuración establecida se puede guardar y recuperar en/desde un endpoint de MOCKAPI o localmente, utilizando para este último caso, LocalStorage.



## Para validar el cumplimiento de las consignas

- ✓ **Librería de Speech Synthesis** sugerida por el profe
- ✓ **Manejo de DOM y Eventos JS** para el armado de pantallas dinámicas
- ✓ **Uso de Fetch (GET + POST + PUT)** mostrar, crear, modificar tareas contra endpoint de MOCKAPI
- ✓ **Endpoint en MOCKAPI.IO + Estructura de datos compartida** (*pueden mejorarla si lo desean*)
- ✓ **Uso de módulos JS** (*un sólo archivo JS en el documento*)
- ✓ **Un o más archivos HTML** (*a gusto de ustedes*)

Si quieren resolver todo en un HTML, pueden usar el tag **<dialog>** para las ventanas:

1. Configuración
2. Nueva tarea
3. Modificar tarea (*esta parte es opcional, si llegan a hacerla, genial*)

- ✓ **Configuración de Speech Synthesis guardada en LocalStorage** (*guardar y recuperar desde LocalStorage*)