

## Actividad de repaso Evaluación 2

UD 07-08

Duración: 3h

### Ejercicio de Repaso 1 - jQuery

#### Animación de un círculo con efectos de opacidad y desplazamiento

##### Descripción:

El HTML y el CSS viene dado y no se puede editar. Crea un JS y realiza el siguiente ejercicio haciendo uso solamente de jQuery y jQueryUI.

Se debe implementar una interfaz con los siguientes elementos:

- Un slider que permite seleccionar la opacidad del círculo (valores de 0.1 a 1, con paso de 0.1 y valor inicial en 1).
- Una lista ordenable con distintas posiciones de desplazamiento horizontal (en píxeles).
- Dos botones: "Iniciar" y "Parar".
- Un div circular de 100 píxeles de diámetro que comienza en la posición inicial (sin desplazamiento) y con opacidad 1.

### Ejercicio de Repaso - Animación de un Círculo

Opacidad: 1

Orden de desplazamiento


0 px

500 px

1000 px

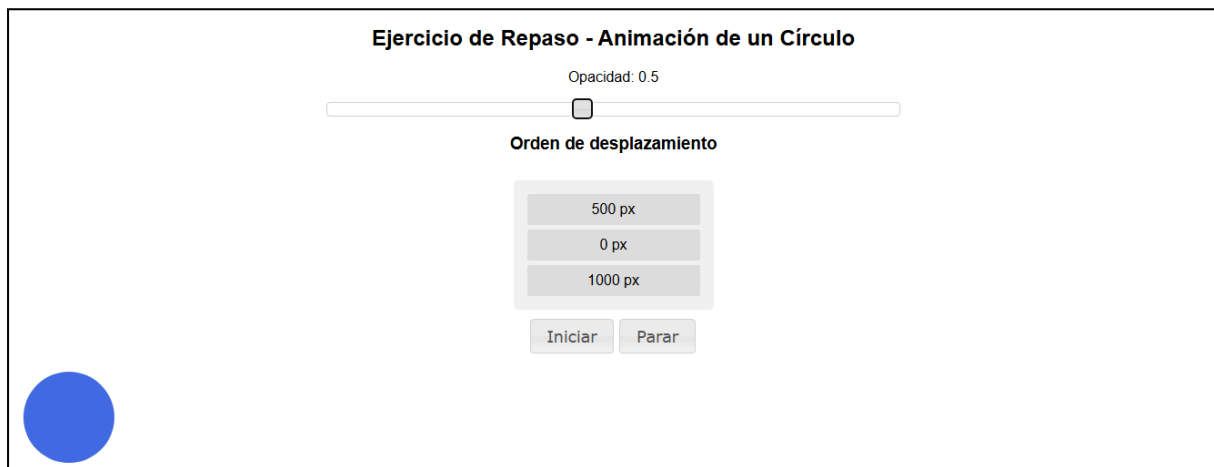
Iniciar

Parar

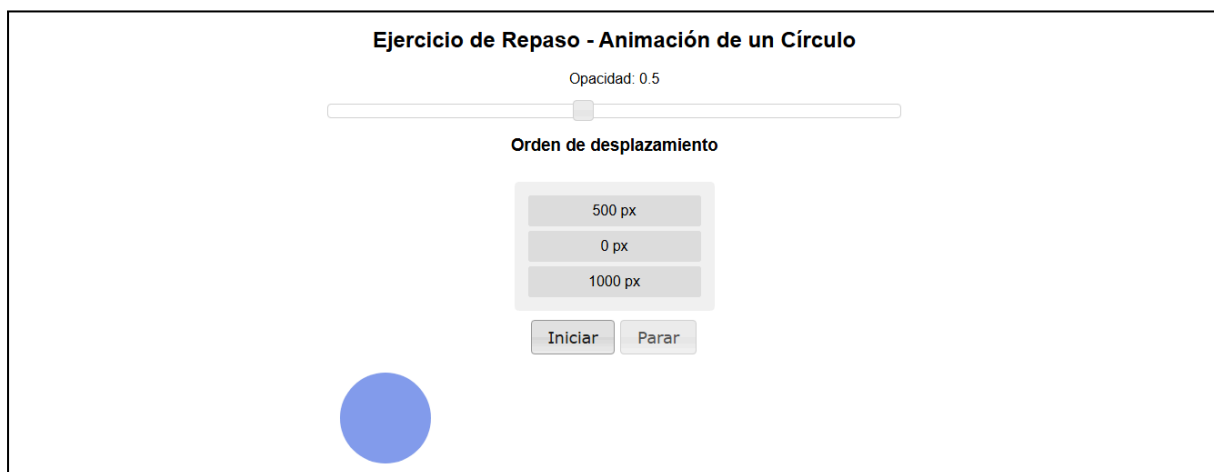


##### Funcionamiento:

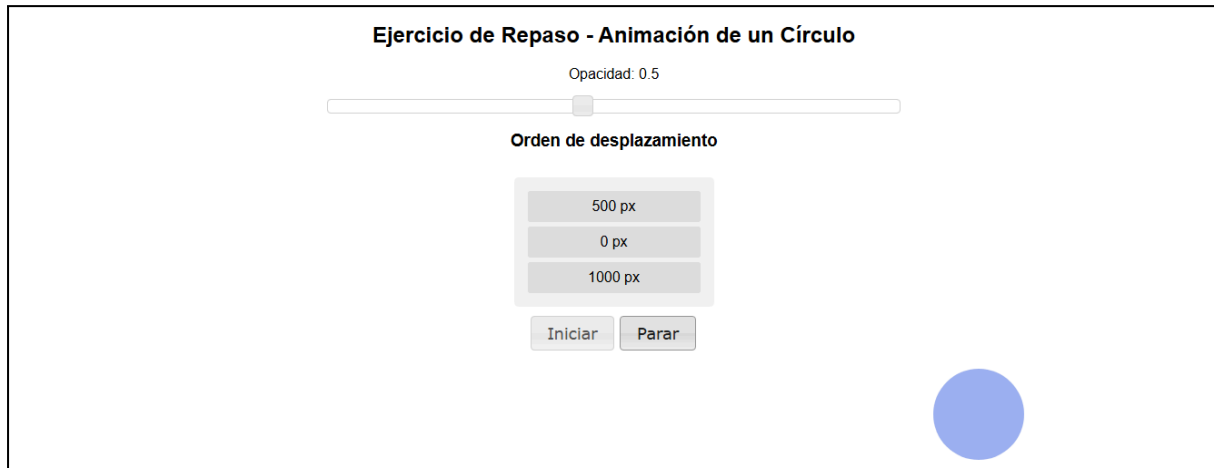
1. El slider permite ajustar la opacidad del círculo, mostrando su valor en un texto.
2. La lista de posiciones (por defecto: 0px, 500px, 1000px) es reordenable, para elegir el orden en que el círculo se desplazará en la animación.
3. Al hacer click en "Iniciar", se producen las siguientes animaciones secuencialmente, con una duración de 2 segundos cada una:
  - El círculo se mueve horizontalmente a la primera posición de la lista y cambia su opacidad al valor del slider.
  - Luego se mueve a la segunda posición de la lista.
  - Finalmente, se mueve a la tercera posición de la lista.



4. Si se pulsa "Iniciar" mientras la animación está en curso, esta se detiene y comienza un nuevo ciclo con los valores actuales.



5. El botón "Parar" detiene la animación en el punto en el que se encuentre. Si se pulsa "Iniciar" nuevamente, el ciclo comienza desde ese punto con los valores actuales.



## Ejercicio de Repaso 2 - Vue.js

### Organizador de Planes con Amigos en Vue.js

#### Descripción

Se debe desarrollar una aplicación web en **Vue.js 3 con Composition API y Pinia** que permita a los usuarios organizar planes con amigos, gestionar la asistencia y visualizar los planes de manera dinámica. La aplicación ofrecerá funcionalidades de creación, edición y eliminación de planes, así como la administración de asistentes.

---

#### Funcionalidades Principales

- **Gestión de Planes:** El usuario podrá agregar y eliminar planes.
  - **Control de Asistencia:** Se podrá cambiar la asistencia del usuario y de los asistentes de cada plan.
  - **Lista de Asistentes:** Se podrán añadir amigos a los planes y gestionar su asistencia.
  - **Interfaz Dinámica:** La aplicación reflejará visualmente los estados de los planes y asistentes, así como del total de planes a los que acudirá.
  - **Modularidad con Componentes Vue y Pinia:** Se estructurará en componentes reutilizables y un store centralizado.
- 

#### Pasos a Seguir

##### 1. Configuración del Proyecto

- a. Inicializar un proyecto en Vue 3 con Vite.
- b. Instalar **Pinia** para la gestión del estado.
- c. Configurar la estructura de carpetas para componentes y stores.

##### 2. Implementación del Store (Pinia)

- a. Crear un **store** para gestionar los planes ([planStore.js](#)).

- b. Incluir funciones para agregar, eliminar y modificar planes.
- c. Implementar la lógica para manejar la asistencia del usuario y de los amigos.

### 3. Creación de Componentes Vue

- a. **AddPlan.vue** → Input y botón para añadir nuevos planes.
- b. **PlanItem.vue** → Mostrar los detalles de un plan con opciones de gestión.

### 4. Integración de Todos los Componentes

- a. Estructurar la aplicación en **App.vue**, importando y organizando los componentes.

### 5. Funcionalidad de los Planes en **planStore.js**

- a. Permitir añadir planes con validaciones para evitar duplicados.
- b. Permitir cambiar el estado de asistencia de un plan.
- c. Posibilidad de eliminar un plan existente.
- d. Cálculo a tiempo real del total de planes a los que el usuario confirma asistencia.

### 6. Gestión de Asistentes en **planStore.js**

- a. Implementar un sistema para añadir amigos a un plan.
- b. Mostrar los asistentes en la interfaz con su estado de asistencia.
- c. Incluir opciones para cambiar la asistencia de un amigo.

### 7. Actualización del DOM y Eventos

- a. Implementar reactividad para que los cambios en los planes y asistentes se reflejen automáticamente.
- b. Manejar eventos como **click** y **keyup** para interacciones.

### 8. Asegurar que todas las funcionalidades se ejecuten correctamente.

## Hola! Aquí están tus planes:

### Cena de fin de curso

- Gorka
- Xabi
- Leire

### Concierto de Me fritos and the gimme cheetos

- Jon
- Xabi

### Festival GameFest

- Ainhoa
- Miren

**Tus planes totales: 2**