

Universidad Tecnológica Nacional
Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Computación III

Apellido:		Fecha:	21/11/2023
Nombre:		Docente ⁽²⁾ :	Christian Baus
División:	3° D	Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	<div style="display: flex; justify-content: space-around;"> <div>PP</div> <div></div> <div>RPP</div> <div></div> <div>SP</div> <div>X</div> <div>RSP</div> <div></div> <div>FIN</div> <div></div> </div>		

(1) Las instancias válidas son: 1^{er} Parcial (**PP**), Recuperatorio 1^{er} Parcial (**RPP**), 2^{do} Parcial (**SP**), Recuperatorio 2^{do} Parcial (**RSP**), Final (**FIN**) . Marque con una cruz.

(2) Campos a ser completados por el docente.

Tema 1

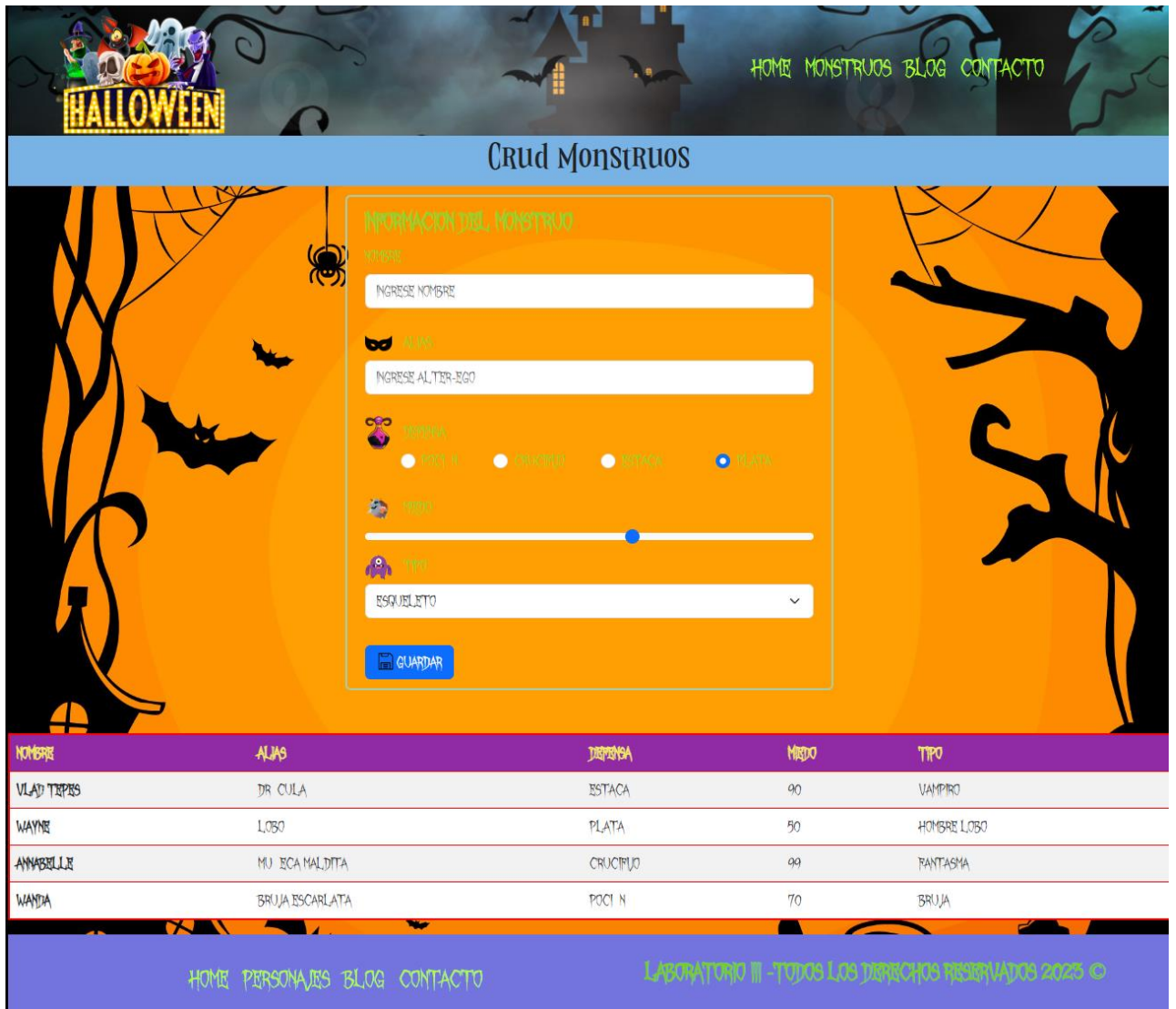
Bienvenidos a Halloween

"El Conde Drácula nos ha confiado la fascinante tarea de crear una aplicación SPA (Single Page Application) destinada a la administración y registro de monstruos de Halloween. En un mundo donde la magia, la oscuridad y la diversión se entrelazan, esta aplicación será esencial para mantener un control meticuloso de los seres aterradores y espeluznantes que pueblan nuestros terrores nocturnos.

En esta misteriosa aventura de desarrollo, nos embarcaremos en la creación de un CRUD (Create, Read, Update, Delete) que permitirá a nuestro ilustre cliente mantener un registro completo de los monstruos que visitan su lujoso Hotel Transylvania. A medida que los monstruos de todas partes del mundo se congregan en este refugio de lo sobrenatural, la necesidad de gestionar sus datos se vuelve más apremiante que nunca.

Acompáñanos en esta emocionante odisea tecnológica mientras diseñamos una aplicación que hará que los corazones de los habitantes de Halloween Town latan con más fuerza. Desde la creación de nuevos monstruos hasta la visualización de aquellos ya registrados, estamos a punto de sumergirnos en un mundo de magia, misterio y horror, donde la información y el entretenimiento se unen en perfecta armonía. ¡Adelante, monstruosa audiencia, comencemos esta misión espeluznante!"

- Desarrollar una aplicación que cuente con una única página donde exista el listado de los monstruos dados de alta (mostrar un Spinner mientras cargan), así como un formulario para dar de alta a nuevos monstruos. Para esto deberá usar HTML y CSS para darle agradable "look and feel".
- La lista de monstruos deberá ser generada dinámicamente (con código JS) en base a la estructura del objeto utilizado.
- Clase Personaje: id, nombre, tipo
- Clase Monstruo que hereda de Personaje: alias, miedo, defensa.
- Además deberá contar con algún manejador de eventos, tal que al hacer click en algún elemento de la lista, se carguen los valores del monstruo seleccionado en el formulario. Se deberán habilitar los botones necesarios para permitirnos cancelar la edición o eliminar el monstruo. Para dar de alta un nuevo monstruo se utilizará el mismo botón, el cual debe cambiar el texto identificando la acción que desencadenará.



Especificaciones

Header

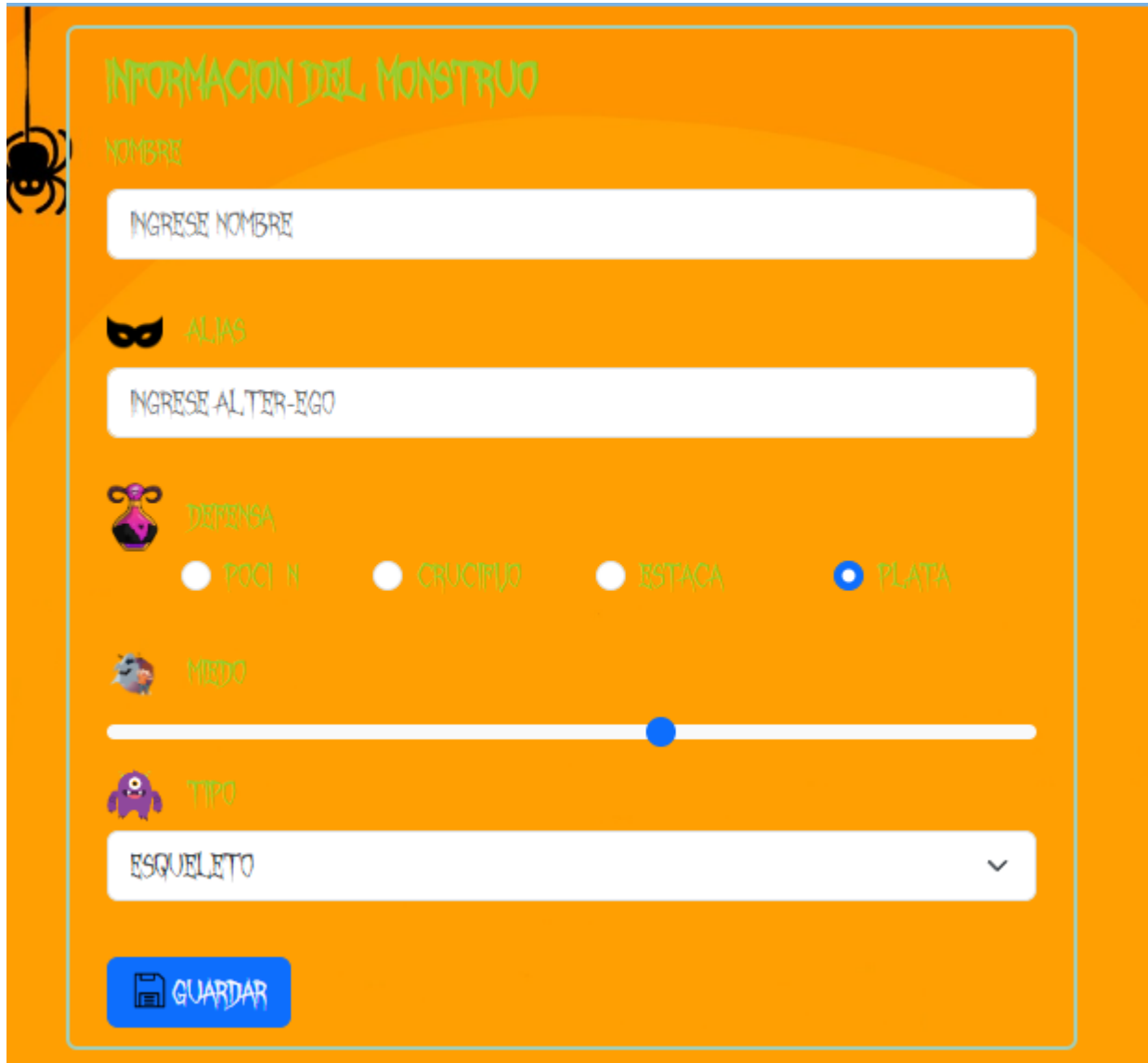


1) Crear un Logo de la marca

2) Agregar un navbar

3) A partir de cierto ancho de pantalla, cambiar (a gusto, usando media queries) la manera en la que se muestran las opciones de navegación. Ejemplo:

```
const armas = ["Esqueleto", "Zombie", "Vampiro", "Fantasma", "Bruja", "Hombre Lobo"];
```



INFORMACION DEL MONSTRUO

NOMBRE


ALIAS

DEFENSA

☐ POCIÓN
 ☐ CRUCIFIXO
 ☐ ESTACA
 ☒ PLATA

MIEDO

TIPO

 **GUARDAR**

- 1)Respetar los controles de la imagen (el posicionamiento es a gusto).
- 2)Agregar iconos a los input de características del superhéroe
- 3)Agregar icono al botón de “Guardar”.

tabla

NOMBRE	ALIAS	DEFENSA	MIEDO	TIPO
VLAD TEPES	DR CULA	ESTACA	90	VAMPIRO
WAYNE	LOBO	PLATA	50	HOMBRE LOBO
ANNABELLE	MUJER MALDITA	CRUCIFIXO	99	PANTASMA
WANDA	BRUJA ESCARLATA	POCIÓN	70	BRUJA

- 1)pintar por css el encabezado
- 2)centrar la tabla en la pantalla
- 3)No mostrar el id por pantalla.

footer

HOME PERSONAJES BLOG CONTACTO

LABORATORIO III - TODOS LOS DERECHOS RESERVADOS 2023 ©

1) Respetar posicionamiento

spinner



1) Agregar un spinner acorde al tema.

Javascript

1) Utilizar una clase "Personaje" que contenga los atributos:

- id
- nombre
- fuerza

2) Utilizar una clase "Monstruo" que herede de "Personaje" con el resto de los atributos necesarios

- Alias
- Defensa
- Miedo

Backend

- El backend será en localStorage. Agregar delay de 2 segundos en cada operación, de manera de poder visualizar el spinner.

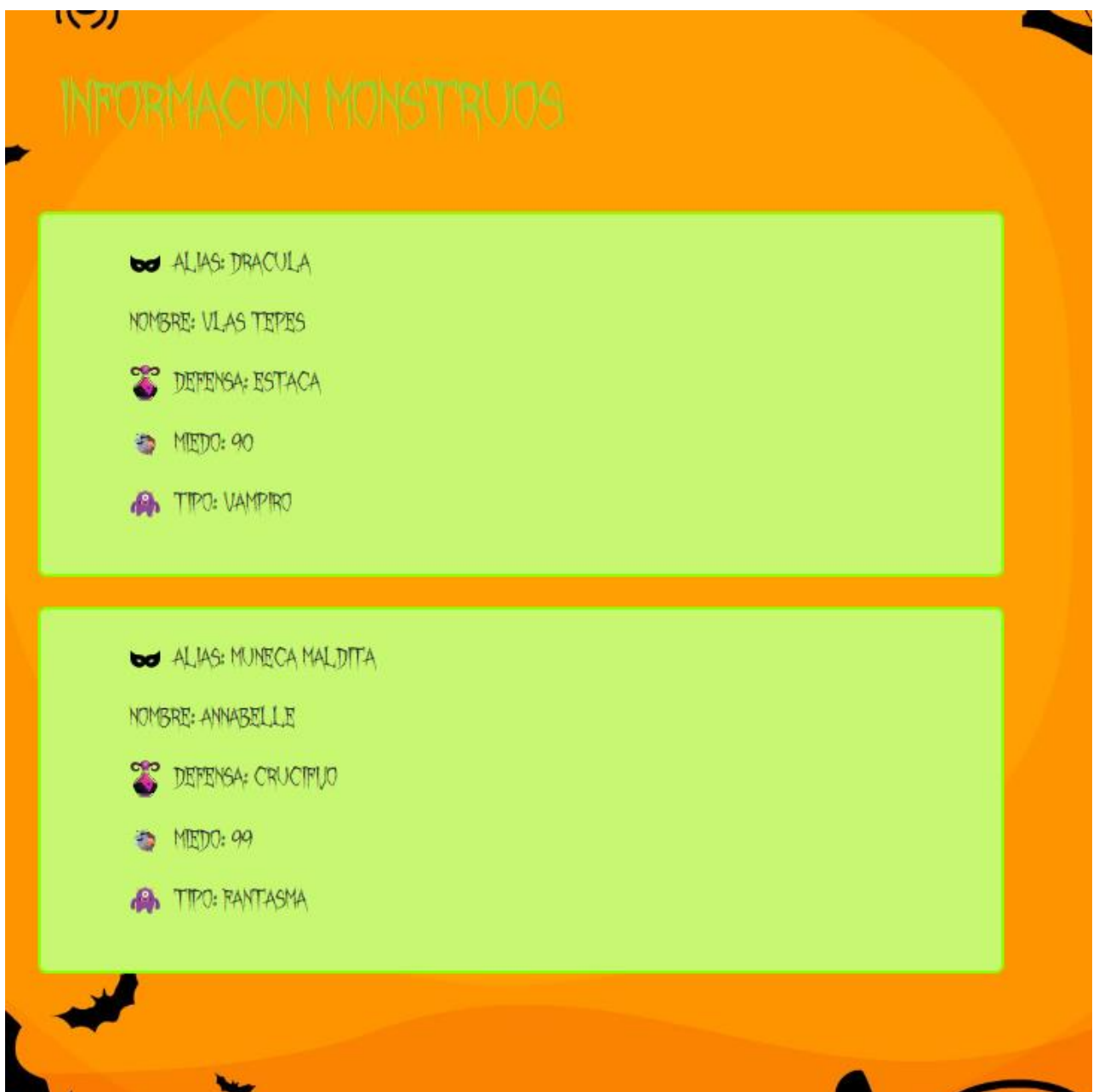
PAGINA PRINCIPAL

Como ya tenemos nuestro CRUD funcionando, ahora se deben mostrar los elementos almacenados en la página principal de la aplicación.

Crear una página principal que consuma los datos del localStorage y genere dinámicamente las cards de Monstruos.



- Agregar una imagen de background que ocupe el 100% del viewport.
- Mantener el mismo navbar que en la página de administración.



- Para cada elemento almacenado en localStorage, crear un artículo en la sección principal de la página.


- El diseño es a gusto.
- Se pueden utilizar cursores personalizados



Segundo Parcial

Map/Reduce/Filter

- Agregar una sección que permita filtrar la tabla que se muestra por pantalla por tipo de monstruo cuyas opciones son Todos/Vampiro/Hombre Lobo/Fantasma/Esqueleto/Bruja/Zombie. Por defecto debe ser todos
- Mostar en un input de solo lectura el promedio de nivel de miedo de los monstruos representados en la tabla.
- Agregar controles que permitan seleccionar las columnas que deben aparecer en la tabla de monstruos. Por defecto todas las columnas deben ser visibles
- Los monstruos que figuran en la tabla deben estar ordenados de manera decreciente por fuerza

				
FILTRAR POR TIPO		PROMEDIO MIEDO		
<input type="text" value="TODOS"/>		<input type="text" value="66.25"/>		
<input checked="" type="checkbox"/> NOMBRE	<input checked="" type="checkbox"/> ALIAS	<input checked="" type="checkbox"/> DEBILIDAD	<input checked="" type="checkbox"/> MIEDO	<input checked="" type="checkbox"/> TIPO
NOMBRE	ALIAS	DEBILIDAD	MIEDO	TIPO
VILAD TRIPES	DRACULA	ESTACA	90	VAMPIRO

Bootstrap

- Utilizar la biblioteca Bootstrap y su esquema de columnas donde sea conveniente para lograr un comportamiento “responsive”
- Utilizar entre 5 y 10 clases de Bootstrap (Además de las ya utilizadas en el punto anterior).

Backend

- Reemplazar la capa de acceso a datos por llamadas Ajax al servidor.
- Utilizar **XMLHttpRequest** en al menos dos de las cuatro operaciones.
- Utilizar fetch para el “get” de la página principal.
- Utilizar axios para el “delete” de monstruo.
- El spinner debe ser visible mientras se realizan dichas operaciones.
- Utilizar como servidor el paquete **json-server**