Seminario de Lenguajes

Opción: PHP, React y API Rest 2025

React

React es una librería de JavaScript que se utiliza para construir interfaces de usuario. Toda aplicación web React se construye a través de componentes reutilizables que conforman partes de la interfaz de usuario — podemos tener un componente distinto para nuestra barra de navegación, otro para el pie de página, otro para el contenido principal, etc.

Un componente es una pieza de UI (siglas en inglés de interfaz de usuario) que tiene su propia lógica y apariencia. Un componente puede ser tan pequeño como un botón, o tan grande como toda una página.

Arquitectura base

Este es un ejemplo de organización de carpetas una vez creado el proyecto de React, pueden tomarlo como base, agregar o modificar cosas para el desarrollo de esta entrega:

- /
    - package.json: Contiene las dependencias del proyecto y scripts de comandos.
    - README.md: Documentación del proyecto.
- public/: Contiene archivos estáticos que se sirven directamente, como index.html y el favicon.
- src/: Contiene todo el código fuente del proyecto.
    - assets/: Archivos estáticos como imágenes y hojas de estilo.
        - images/: Imágenes del proyecto.
        - styles/: Hojas de estilo CSS.
    - components/: Componentes reutilizables de la interfaz de usuario.
    - pages/: Componentes de página que representan vistas completas.
    - utils/: Funciones utilitarias y helpers.
    - App.jsx: Componente raíz de la aplicación.
    - main.jsx: Punto de entrada de la aplicación que renderiza el componente raíz en el DOM.

Segunda entrega

Esta entrega consistirá en el desarrollo de una aplicación React que consumirá los endpoints de la API creada en la entrega anterior. De ser necesario, se pueden crear nuevos endpoints si lo requieren.

1) Definición de Componentes Reutilizables

Dentro de la carpeta **src**, crear una carpeta llamada **components**. Esta carpeta contendrá todos los componentes reutilizables del proyecto.

Desde la cátedra, se les solicita obligatoriamente definir los siguientes tres componentes:

a) HeaderComponent: el cual debe contener un logo que identifique su página (puede ser una imagen de internet o creada) y un título.
b) FooterComponent el cual debe contener los nombres de los integrantes del grupo y el año en curso.
c) NavBarComponent: el cual debe permitir navegar las diferentes páginas.

Además de estos componentes obligatorios, **cada grupo debe crear otros componentes reutilizables que consideren necesarios** para mejorar la estructura y calidad de su proyecto

2) Estructura Mínima de las Páginas:

Cada sección o página del proyecto debe incluir, como mínimo, los siguientes componentes reutilizables:

a) HeaderComponent
b) FooterComponent
c) NavBarComponent

Consideraciones:

Al hacer click sobre el logo o título del Header, este debe redirigir a la página principal.

El Navbar deberá cambiar según el usuario esté logueado o no.

- Si no lo está, debe contener un botón que conduzca a “Registro de usuario” y otro a “Login”.
- Si está logueado debe incluir, una sección que diga “Hola –nombre del jugador–” un botón que lo lleve a “Mis mazos”, “Editar usuario” y “Logout”.

3) Módulos a desarrollar:

a) Listado de estadísticas

Crear un archivo dentro de la carpeta pages/stat llamado StatPage. Esta página deberá mostrarse como página por defecto. Dicha página debe contener:

- Un listado de todos los usuarios con su nombre público detallando en cuantas partidas participó, cuantas partidas tienen ganadas, perdidas, empatadas y un promedio de partidas ganadas.
- El listado debe permitir ordenar por mejor y peor performance.
- Quien tenga el mejor promedio debe aparecer primero con algún estilo que lo haga destacar.
- Puede agregar un paginado de 5 resultados por página.

El contenido de este listado no necesita que el usuario esté logueado.

b) Registro de usuario

Crear un archivo dentro de la carpeta pages/registro llamado RegistroPage. Dicha página debe contener un formulario con:

- Usuario (único)
- Nombre del usuario (será público)
- Password (type password)

El usuario podrá registrarse para luego iniciar la sesión en el sitio web. Tenga en cuenta que deberá validar que:

- El usuario tenga por lo menos 6 caracteres y no más de 20.
- Que sean únicamente alfanuméricos. Además, se debe verificar que ese usuario no esté en uso.
- El nombre de usuario no puede ser vacío y debe tener como máximo 30 caracteres.
- La password tenga por lo menos 8 caracteres y que tenga mayúsculas, minúsculas, números y caracteres especiales.

Estas validaciones son independientes de las que realice el backend. En caso de error se debe informar cual o cuales validaciones fallaron para que el usuario pueda solucionarlo (si está dentro de sus posibilidades, por ejemplo, formato de la contraseña, el usuario ya existe, etc).

c) Login

Dicha página debe contener un formulario con:

- Usuario
- Password (type password)

Un usuario previamente registrado podrá enviar sus credenciales para hacer el login. En caso de que sea exitoso se guardará el token y el nombre de usuario. En caso de error, debe informarlo.

d) Editar usuario

Dicha página debe contener un formulario con:

- Nombre del usuario
- Password (type password)

- Repetir Password (type password)

Se debe validar que:

- El nombre de usuario no puede ser vacío y debe tener como máximo 30 caracteres.
- La password tenga por lo menos 8 caracteres y que tenga mayúsculas, minúsculas, números y caracteres especiales.

Estas validaciones son independientes de las que realice el backend. En caso de error se debe informar cual o cuales validaciones fallaron para que el usuario pueda solucionarlo (si está dentro de sus posibilidades, por ejemplo, formato de la contraseña, etc).

e) Mis Mazos

Un usuario logueado podrá ver sus mazos creados. Para eso se listará el nombre del mazo y 4 opciones:

- Ver Mazo: se muestra en un modal las cartas que lo componen.
- Eliminar: da de baja el mazo seleccionado (siempre que no haya sido previamente usado en una partida).
- Editar: Muestra un pequeño formulario junto a la fila para cambiar el nombre del mazo (puede ser junto a la fila o junto al nombre del mazo).
- Jugar: Enlace a “Jugar” con el mazo seleccionado.

La forma en que se muestra la información anterior puede modificarse por una más óptima si así lo considera pero debe permitirle al usuario realizar las acciones descriptas.

En la parte inferior o superior debe haber un botón o link para ir al “Alta de nuevo mazo”. Si el usuario ya tiene 3 mazos creados, este botón debe estar deshabilitado.

f) Alta de un mazo

Para crear un nuevo mazo el usuario deberá completar un formulario con:

- nombre del mazo (máximo de 20 caracteres).
- Listado de todas las cartas disponibles para seleccionar.
    - Este listado debe contener las características que den una vista relevante de la carta (nombre, puntos de ataque, atributo, etc).
    - La manera en que se muestra el listado, la información de las cartas y su modo de selección queda a su criterio. Por ejemplo, una imagen de la carta junto a un checkbox para seleccionarla.
    - El listado podrá filtrarse para reducir la cantidad de cartas que se ven. En la parte superior, debe haber un filtro/buscador con 2 criterios:
        - “Atributo” de la carta,
        - nombre de la carta, e
        - incluir un botón para “limpiar” los filtros.

En caso que el usuario ya tenga un máximo de 3 mazos creados, se debe devolver el error correspondiente cuando este quiera guardar el mazo. Idem si ocurre otro error al intentar guardar el mazo.

g) Jugar

Al iniciar una partida, se mostrará un tablero como imagen de fondo o algo que lo represente como tal.

- Las cartas pertenecientes al mazo del usuario se mostrarán en fila en la parte inferior (pueden agregarle una imagen).
- Las cartas del servidor se mostrarán en la parte superior, no serán visibles para el usuario, estarán dadas vuelta pero si se verá su atributo.
- De las cartas del usuario se podrá ver: nombre, atributo, ataque y una foto (opcional)
- Para jugar una carta se debe hacer doble click sobre la misma o arrastrarla al centro del tablero (elegir solo un método).
- Al jugar una carta, el centro del tablero se actualizará mostrando la carta que jugó el usuario, la carta del servidor y quien ganó la jugada.
- Al jugar las 5 cartas se debe mostrar quien ganó la jugada, la partida y un botón de “Jugar otra vez?”

A medida que avance la partida, las cartas jugadas deben ir desapareciendo y los atributos de las cartas disponibles del servidor ir actualizandose.

Se puede proponer una manera más óptima de mostrar el avance de la partida pero siempre teniendo en cuenta los requerimientos mínimos antemencionados.

Requisitos obligatorios

- Usar las últimas versiones estables de nodejs y ReactJS.
- El nombre del proyecto debe ser **pokebattle**.
- Los estilos NO pueden estar embebidos en los tags del código. Generar un file .css, .scss,etc. donde se definan cada id o class para utilizar.
- Validar todos los formularios antes de realizar los llamados al backend.
- Si instalan librerías externas dejarlas documentadas en el archivo README.md.
- Si modifican o agregan algunos endpoints, documentarlos en el README.md especificando ruta del servicio y definir por qué se realizó la modificación.
- Para la creación de componentes se debe utilizar componentes de funciones (Function Components).

Metodología para la entrega en IDEAS

- Hacer un archivo .zip o .rar con todos los archivos de ambos proyectos (excepto la carpetas vendor —API— y node_modules —React—).
- Subir al repositorio del grupo. Fecha límite de entrega: 1/7