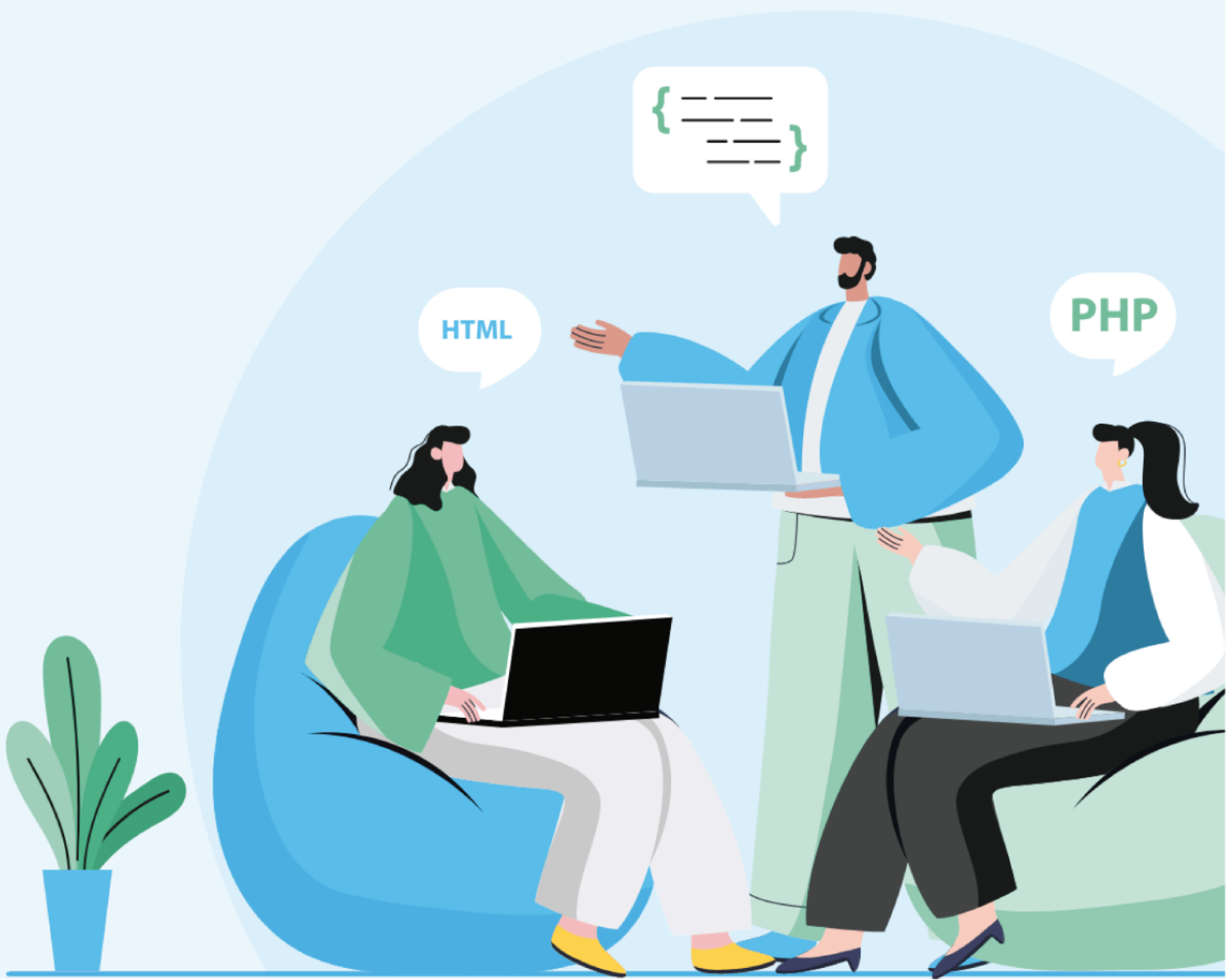




Skill Up Lab - React

Unidad 2





Unidad 2: Armado del componente Login

Antes de adentrarse en esta Unidad

Para que puedas aprovechar mucho más el contenido presente en esta unidad, te recomendamos que antes de avanzar tengas en claro lo siguiente:

- cómo instalar una aplicación de React con el comando `create-react-app`
- tener una relativa familiaridad con el entorno de archivos de una aplicación de react
- entender la diferencia entre el archivo `index.js` y el componente `App.js`
- cómo ejecutar la aplicación de React desde la terminal

Objetivos de la unidad

Con el estudio de esta unidad buscamos que:

- Puedas armar el componente Login, que permitirá a las personas que usan tu aplicación, loguearse en ella
- Logres validar el formulario de Logueo para evitar que una persona con credenciales erradas pueda hacer uso de la aplicación
- Logres enviar los datos suministrados por la persona a una API para que de esta última puedas obtener el token de autenticación
- Implementes el manejo de mensajes de error con la librería Sweet Alert
- Persistas el token obtenido de la API, en el almacenamiento local del navegador
- Generes un redireccionamiento al componentes "Listado" una vez las credenciales de usuario estén validadas y la API haya entregado el respectivo token de autenticación



Clase 1: Armado del componente "Login"

¿Qué es el componente "Login"?

Este componente es el que va a permitir que las personas que usan la aplicación se logueen en la misma. El mismo estará compuesto por un breve formulario con dos campos: correo electrónico y contraseña, los cuales funcionarán como un mecanismo de entrada de información para el proceso de autenticación de usuarios.

¿Para qué sirve el componente "Login"?

Este componente tiene como objetivo principal servir como un filtro de usuarios, para que, solamente aquellas personas que se loguean con las credenciales válidas puedan usar el completo de la aplicación.

¿Cómo se implementa el proceso de Login en un proyecto real?

Esto es algo bastante habitual, pues muchas aplicaciones para que puedan ser usadas, necesitan autenticar al usuario para así proveer de toda su funcionalidad. Por ello es importante conocer cuales son los pasos a dar para poder llevar adelante esta implementación, la cual, posteriormente seguramente usaremos en muchas ocasiones en nuestra vida diaria.

Clase 2: Validación del formulario de Login

¿Qué es la validación de formularios?

La validación es el proceso que nos permite constatar (cómo developers) que la información suministrada de parte del usuario y que requiere nuestro sistema para funcionar, sea la correcta.

¿Para qué sirve la validación de formularios?

La validación de formularios nos permite corroborar que la información que estamos recibiendo y que posteriormente enviaremos al servidor, sea la correcta y no presente errores, pues en el desarrollo de aplicaciones siempre estaremos en la búsqueda constante de la optimización y no sería lógico ni performante enviar al servidor información errada, pues lo estaríamos sobrecargando con responsabilidades y tareas que podrías resolver fácilmente del lado del cliente. Para esto es para lo que sirve puntualmente



la validación de formularios, para resolver en el entorno del cliente los posibles problemas que se puedan llegar a presentar.

¿Qué ejemplos existen sobre la validación de formularios?

El más común lo encontramos en los formularios de registro. Aquel en donde al momento de poner nuestra dirección de correo electrónico, se verifica que la misma coincida con el formato habitual: "user@server.ext" y de no coincidir nuestra información ingresada con este formato, es ahí entonces donde obtendremos algún mensaje alertándonos de este error.

¿Cómo se implementa la validación de formularios en un proyecto real?

De la misma manera que lo haremos en esta clase. Pues, siempre que deseemos verificar que la información ingresada por el usuario es la correcta, es entonces allí en donde necesitaremos de estos procesos de validación.

Clase 3: Envío del formulario (HTTP I)

¿Qué es el envío de formularios vía HTTP?

Los formularios en HTML no son nada más que presentacionales y para poder usar la información suministrada por el usuario, tendremos que enviar la misma a algún lugar. Tradicionalmente a una API. Y para poder generar el envío de dicha información necesitaremos implementar algún método HTTP que nos permita hacer esto.

¿Para qué sirve el envío de información vía HTTP?

Tal como lo mencionamos anteriormente, un formulario HTML no es funcional si no envíamos su información a algún lugar. Y por este motivo es que los métodos HTTP tales como GET, POST o PUT nos van a ser de gran utilidad, pues con los mismos tendremos la posibilidad de hacerle llegar al servidor aquellos datos que el usuario ha dejado en nuestra aplicación.

¿Dónde se pone en uso el envío de información vía HTTP?

Generalmente, siempre que diligencies un formulario (como el de login, registro, contacto, etc) y des click en el botón "ENVIAR", allí mismo estás disparando un envío de la información vía HTTP, para que sea el servidor



quien procese dicha información y pueda darte una respuesta dependiendo de la información suministrada.

¿Cómo se implementa el envío de información vía HTTP en un proyecto real?

En los formularios de contacto, de registro, de logueo o en cualquier otra petición que obtenga o envíe datos desde y hacia el servidor se implementan este tipo de solicitudes HTTP.

Clase 4: Manejo de errores

¿Qué es el manejo de errores?

El manejo de errores es la técnica a través de la cual como desarrolladores podemos manejar las distintas respuestas obtenidas después de enviar información a una API o incluso cuando nos encontramos validando la información de un formulario.

¿Para qué sirve el manejo de errores?

Sirve para advertir a la persona que usa nuestra aplicación de los posibles errores que se encontraron en el procesamiento de la información, sea porque suministró información errada o quizás porque aún habiendo suministrada información correcta, el servidor no pudo procesar la misma de manera satisfactoria.

¿Cómo se implementa el manejo de errores en un proyecto real?

Existen diversas maneras de hacerlo, pero una muy común es hacerlo usando librerías que nos permitan disparar cajas modales que alerten e informen al usuario del error resultante de la operación. Puntualmente en esta clase, implementaremos la librería "Sweet alert" para poder mostrar estos mensajes de una manera mucho más amigable para el usuario.



Clase 5: Persistencia del token

¿Qué es un token y que es la persistencia del mismo?

Un token es un identificador único obtenido tras un proceso de autenticación el cual le permite saber al servidor, que la persona que usa la aplicación se ha logueado satisfactoriamente. Por otro lado, persistir un token significa: mantenerlo guardado en algún lugar para posteriormente usar su información. Para que así de esta manera, evitemos solicitar al usuario en cada momento sus credenciales de acceso.

¿Para qué sirve la persistencia del token?

Cómo ya lo dijimos con anterioridad, sirve para evitar hacer múltiples solicitudes de autenticación al servidor, cada vez que el usuario de la aplicación desea llevar a cabo una determinada actividad. Por este motivo, el token es tan importante ya que él mismo lo obtendremos en el primer llamado de autenticación y de ahí en adelante usaremos el mismo cada vez que sea necesario.

¿Cómo se implementa la persistencia del token en un proyecto real?

Existen un par de caminos para lograr este resultado, pero de momento el más sencillo y cómodo para nosotros será guardar el mismo en el almacenamiento local del navegador, pues de esta manera aprovecharemos las múltiples ventajas de este tipo de almacenamiento.

Clase 6: Redireccionamiento al componente "Listado"

¿Qué es un redireccionamiento?

El redireccionamiento es un proceso mediante el cual, después de ejecutar una acción determinada, llevamos a la persona que usa nuestra aplicación a una ubicación específica. Usualmente es muy común de implementar en el proceso de logueo. Aquel en donde después de validar las credenciales de una persona, le daremos acceso completo a la aplicación. Vale la pena aclarar que una redirección también se puede generar cuando nos damos cuenta que sin tener los permisos correspondientes, una persona desea ingresar a una sección a la que no tiene permiso.



¿Para qué sirve la redirección?

Sirve para enviar a la persona que usa nuestro sistema, a un determinado lugar después de que ejecuta una acción. Sirve puntualmente para proteger secciones de la aplicación que requieren que la persona usuaria tenga permisos específicos.

¿Qué ejemplos existen sobre el redireccionamiento?

El más sencillo y el que puedes probar por tu cuenta es el siguiente: estando logueado en tu cuenta de correo, dirígete a la bandeja de entrada, allí copia la dirección presente en la barra de direcciones. Ahora, ve a una ventana de incógnito en tu navegador y pega la dirección que copiaste anteriormente. ¿Pudiste entrar a la bandeja de entrada de tu correo electrónico? Estamos seguros que no. Y que lo que sucedió fue que el sistema te redireccionó a la sección en donde debes loguearte en tu cuenta de correo. Si esto fue lo que sucedió, ahí tienes un ejemplo claro de lo que es una redirección en todo su esplendor.

Enlaces y documentación

- [Visual Studio Code](#): Desde este enlace podrás descargar el editor de texto más utilizado por todo el mundo desde hace ya un par de años. Excelente para escribir código en cualquier lenguaje de programación.
- [NodeJS](#): Descarga este entorno de ejecución para JavaScript, que te permitirá construir cualquier tipo de aplicación usando JS.
- [ReactJS](#): Documentación oficial de esta espectacular librería para el manejo de interfaces en el Frontend. Siempre que tengas una duda, seguramente vendrás a este enlace.
- [Axios](#): Descubre la documentación oficial de la librería más utilizada en el ambiente del desarrollo para hacer peticiones asíncronas y aprende un poco más sobre la misma.



- [Sweet Alert](#): Mostrar mensajes de error o de éxito será aburrido, a menos que uses esta genial librería. Descubre todo el poder de las cajas modales gracias a Sweet Alert.
- [React router dom](#): Si estás buscando construir una SPA (single page application) con React esta librería es infaltable. Ya que la misma te permite gestionar todo el sistema de ruteo de la misma. Con esta sencilla pero eficiente documentación, cualquier duda que tengas, será resuelta.

