



Desarrollo frontend para la aplicación de Acceso a Palcos alojada en la intranet del INDER

Manuela Restrepo Cardona

Informe de práctica para optar al título de Ingeniero Electrónico

Asesor

Luis Germán García Morales

Universidad de Antioquia
Facultad de Ingeniería
Ingeniería Electrónica
Medellín, Antioquia, Colombia
2022

Cita	(Restrepo Cardona, M., 2022)
Referencia	Restrepo Cardona, M. (2022). <i>Desarrollo frontend para la aplicación de Acceso a Placos alojada en la intranet del INDER</i> Trabajo de grado profesional. Universidad de Antioquia, Medellín, Colombia.
Estilo APA 7 (2020)	



Biblioteca Carlos Gaviria Díaz

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.

Decano/director: Jesús Francisco Vargas Bonilla.

Jefe departamento: Augusto Enrique Salazar Jiménez.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Tabla de Contenido

1. Introducción.....	5
2. Objetivos.....	7
2.1 Objetivos Generales.....	7
2.2 Objetivos Específicos.....	7
3. Marco Teórico.....	8
3.1 Requisitos.....	8
3.2 Frontend.....	9
3.3 API REST.....	11
3.4 Backend.....	14
3.5 Black Box Testing.....	14
4. Metodología.....	15
4.1 Requisitos.....	15
4.1.1 Requisitos funcionales.....	15
4.1.2 Requisitos no funcionales.....	22
4.2 Frontend.....	23
4.3 API REST.....	38
5. Resultados.....	44
5.1 Mockup.....	44
5.2 Frontend.....	54
6. Conclusiones.....	68
Referencias	

Resumen

El estadio Atanasio Girardot siendo una de las sedes más importantes en la que se realizan todo tipo de eventos deportivos y culturales en la ciudad de Medellín, necesita de constantes intervenciones como mantenimientos y remodelaciones para brindar a los ciudadanos comodidad y buena experiencia durante ellos. Por tal razón, surgió la necesidad por parte de la alcaldía del municipio de hacer diversas remodelaciones, incluyendo la ampliación de los palcos, adecuación de la sala de prensa y cabinas de transmisión, mejoramiento de la iluminación y demás infraestructura del complejo deportivo.

Con la ampliación de los palcos surgió la necesidad de realizar una aplicación web para facilitar el acceso a ellos y realizarlo de forma más eficiente ya que anteriormente la reserva de asientos se llevaba a cabo de manera manual en el estadio. Durante las prácticas académicas realizadas en el Instituto de Deportes y Recreación (<https://www.inder.gov.co/>), se llevaron a cabo diferentes actividades siendo la principal el desarrollo frontend para la aplicación de 'Acceso a Palcos' utilizando React JS, en la que se puede, además de hacer reservas, cancelarlas o modificarlas, bloquear asientos, ver mapa de ocupación y administrar credenciales. Al finalizar la práctica, se entregó un desarrollo frontend funcional que cumple con los requerimientos establecidos por la empresa el cual puede ser integrado con el backend mediante la API REST creada y así obtener la aplicación de Reserva de Palcos completa.

1. Introducción

El Instituto de Deportes y Recreación de Medellín INDER, al ser un ente encargado de fomentar la actividad física y el aprovechamiento del tiempo libre, tiene a su disposición variadas ofertas de programas y espacios destinados al deporte y a la recreación de la ciudadanía. Entre estos espacios se encuentra el estadio Atanasio Girardot que, al igual que los demás escenarios, cuenta con diversos eventos tanto deportivos como culturales. Para dichos eventos, el estadio ofrece el servicio de palcos los cuales solo están a disposición de ciertos funcionarios de la alcaldía de Medellín y personas asociadas a ésta.

Anteriormente para realizar los procesos concernientes a los palcos se contaba con una aplicación web básica; sin embargo, debido a renovaciones realizadas en la estructura de los palcos y a que el framework con el que fue creada la aplicación estaba obsoleto la aplicación dejó de ser funcional. Por esta razón, se propuso realizar una aplicación nueva, de uso interno, desde la que un empleado del INDER pueda realizar las reservas a los palcos a los funcionarios de la alcaldía de forma más eficiente, rápida y que cumpla con los requisitos expuestos por la empresa, con el objetivo de que el estadio Atanasio Girardot pueda mejorar el servicio de acceso a los palcos. El desarrollo de la aplicación se dividió en dos partes, frontend y backend utilizando la arquitectura de microservicios (Dmitry, N., & Manfred, S. S, 2014), permitiendo que se puedan tratar y abordar los problemas de manera independiente. Este proyecto estuvo enfocado solamente en el desarrollo frontend de la aplicación, el cual consistió en la creación de cuatro páginas, una para la realización, modificación o cancelación de reservas, una para el bloqueo de asientos, una para la modificación de las credenciales de los usuarios y, por último, una que muestra el mapa de asientos de cada zona de palcos. La integración entre frontend y backend se llevó a cabo creando una API REST para poder obtener o entregar datos provenientes del backend. Una vez terminados el frontend y la API REST, se realizaron pruebas de caja negra para verificar

el cumplimiento completo de los requerimientos funcionales de la aplicación, resultandos exitosos cada uno de los casos expuestos.

2. Objetivos

2.2. General

Desarrollar el frontend de la aplicación de Acceso a Palcos del estadio Atanasio Girardot, mediante el uso de la biblioteca de Javascript React, HTML (HyperText Markup Language) y CSS (Cascading Style Sheets), que permita al usuario realizar la reserva y el bloqueo de asientos, administración de credenciales y tener una vista del mapa de reservas. Al igual que el desarrollo de una API REST para la integración entre el frontend y el backend utilizando NodeJS.

2.3. Específicos

- Establecer los requerimientos necesarios para la aplicación de Acceso a Palcos con el fin de definir su frontend.
- Diseñar un mockup de la aplicación de acuerdo con los requerimientos establecidos y, con base en esto, desarrollar la interfaz de usuario para la aplicación utilizando React JS, HTML y CSS.
- Definir una API REST para la integración del frontend con el backend de la aplicación y realizar su desarrollo mediante el lenguaje de programación JavaScript.
- Realizar pruebas de funcionalidad para la verificación de los requerimientos establecidos y llevar a cabo las correcciones a las que haya lugar.

3. Marco teórico

Este proyecto, conformado por el desarrollo del frontend de la aplicación de Acceso a Palcos así como la API REST para la integración con el backend se dividió en varias etapas para poder tener una mayor organización y control sobre él. Primero, junto con el equipo de tecnologías de la información se definieron los requerimientos funcionales y no funcionales que debía cumplir la aplicación. Segundo, se diseñó un mockup siguiendo las sugerencias dadas por el equipo de diseño gráfico de la empresa y a partir de este se desarrolló el frontend de la aplicación. Finalmente, se creó una API REST para la integración del frontend y el backend. Una vez terminadas estas etapas, se realizaron pruebas para verificar el correcto flujo de la aplicación. Para poder comprender cada una de estas etapas, se describirán a continuación los conceptos y términos relevantes para el proyecto.

3.1 Requisitos

3.1.1 Requisitos funcionales

Describen el comportamiento del sistema en condiciones específicas e incluyen las características y funciones del producto que los desarrolladores web deben dar solución. Dichos requerimientos deben ser precisos tanto para el equipo de desarrollo como para las partes interesadas. La lista de ejemplos de requerimientos funcionales incluye: reglas del negocio, correcciones, ajustes y cancelaciones de transacciones, funciones administrativas, autenticación, niveles de autorización, seguimiento de auditoría, interfaces externas, requerimientos de certificación, requerimientos de información e información histórica.
(Kompaniets, A, n.d)

3.1.2 Requisitos no funcionales

Se trata de atributos de calidad que describen las formas en que se debe comportar la aplicación. Algunos ejemplos de requerimientos básicos no funcionales se presentan a continuación:

- Usabilidad: es la facilidad con la que una persona promedio puede usar el software o el sitio web para lograr objetivos específicos. La usabilidad se compone de capacidad de aprendizaje, recordación, eficiencia, satisfacción y errores.
- Fiabilidad: para lograr una alta confiabilidad, el desarrollador debe eliminar todos los errores que puedan influir en la seguridad del código y, también, los problemas con los componentes del sistema.
- requerimientos legales: describen el cumplimiento de las leyes por parte de la aplicación y si se llegaran a violar estas regulaciones, puede resultaren multas o castigos legales.
- Rendimiento: indica cómo se comporta la aplicación final cuando el usuario final interactúa con ella en distintos escenarios. Un rendimiento deficiente puede generar una experiencia de usuario negativa y poner en peligro la seguridad del sistema. (Glinz, M., 2007)

3.2 Frontend

El desarrollo frontend, también conocido como el lado del cliente, es la parte de la aplicación con la que el usuario interactúa directamente a través de un navegador web, es decir, se encarga del diseño, composición e interactividad del sitio, incluyendo la ejecución de aspectos como menú de navegación, colores y estilos, tamaño de fuente,

botones e imágenes, entre otros (Frontend Masters, n.d.). Además, realiza la función de mostrar la información entregada por el backend al usuario y recolecta y gestiona información como clics, archivos y formularios. HTML, CSS y JavaScript son los lenguajes utilizados para el desarrollo de frontend, los cuales se analizan a continuación:

- HTML: Significa lenguaje de marcado de hipertexto. El hipertexto define el vínculo entre las páginas web. El lenguaje de marcado se utiliza para definir la documentación de texto dentro de la etiqueta que define la estructura de las páginas web (Musciano, C., & Kennedy, B, 2002).
- CSS: Las hojas de estilo en cascada son un lenguaje de diseño simple destinado a simplificar el proceso de hacer que las páginas web sean atractivas, adaptativas, con efectos visuales que brindan al usuario final una mejor experiencia de navegación (Duckett, J, 2011).
- JavaScript: Es un lenguaje de programación basado en texto que se usa tanto en el lado del cliente como en el lado del servidor y que permite hacer que las páginas web sean interactivas. La incorporación de este lenguaje mejora la experiencia de usuario de la página web al convertirla de una página estática a una interactiva. En resumen, JavaScript agrega comportamiento a las páginas web (Pérez, J. E, 2019).

Existen diversas librerías y frameworks que no solo ayudan a crear una buena experiencia de usuario y facilitan una futura escalabilidad sino también harán que la aplicación sea rápida y más estable. Entre las más populares se encuentra React JS (Fedosejev, A, 2015), una biblioteca de javascript para crear interfaces de usuario usando componentes reusables que permite una

comunicación en tiempo real facilitando así las interacciones con los usuarios. Por su popularidad, estabilidad y versatilidad, es la biblioteca que se usará para el desarrollo de la aplicación.

3.2.1 Mockup

Como se mencionó con anterioridad, antes de empezar con el desarrollo frontend de la aplicación se realizó en conjunto con los diseñadores gráficos de la empresa un mockup que cumple con los requerimientos de la empresa. Un mockup es un conjunto de imágenes de interfaz que tiene como finalidad mostrar el diseño general de cada página y las funcionalidades de cada elemento que la componen y además, permite la inserción de enlaces a páginas particulares para mostrar el flujo correcto que debe tener la aplicación final (WebDew, n.d.). Así, con la ayuda de un mockup, la empresa puede visualizar su proyecto y realizar cambios a conveniencia antes de comenzar con su desarrollo.

3.3 API REST

Para que sea posible una integración entre el frontend de la aplicación de Acceso a Palcos con su backend, es necesario crear una API REST. Una interfaz de programación de aplicaciones (API) para un sitio web, es un conjunto de códigos o funciones que permite que dos aplicaciones se comuniquen entre sí. REST es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP y permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que entienda HTTP (IBM Cloud Education, 2020). La información que entrega un cliente a un servidor o viceversa se puede entregar en cualquier formato, incluyendo Python, PHP o la notación

de objetos de javascript (JSON); además, las API REST pueden ser desarrolladas utilizando variados lenguajes de programación y admiten una gran diversidad de formatos de datos. La figura 1 muestra una representación de la comunicación cliente – servidor usando una API REST.

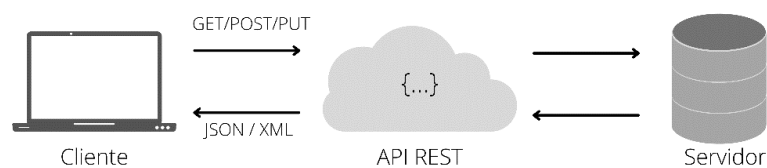


Figura 1. Arquitectura API REST.

3.3.1 HTTP

El Protocolo de transferencia de hipertexto se trata de un protocolo que trabaja en la capa de aplicación del modelo OSI (Alani, M. M., 2014), el cual está diseñado para intercambiar información en la web trabajando con el modelo cliente – servidor lo que significa que las peticiones las inicia el destinatario quien generalmente es el navegador web. Cada petición HTTP realizada lleva consigo una serie de datos codificados que transportan diferentes tipos de información y cada una contiene datos como el tipo de versión, una URL, un método y el encabezado de la solicitud (Berners-Lee, T., 1996). Los métodos HTTP indican la acción que la petición espera del servidor, los más usados son:

- GET: se usa para recuperar información del servidor. Las solicitudes que utilizan este método sólo pueden recuperar datos y no deben tener ningún otro efecto sobre ellos.

- POST: este método se usa para enviar datos al servidor, por ejemplo, información del cliente, carga de archivos, entre otros, mediante formularios HTML.
- PUT: se usa para solicitar al servidor que almacene la información en el URI que está enviando en la solicitud.

Después de que la petición sea procesada el cliente debe responder con el estado de la solicitud para mantener la comunicación, esta contiene un código de estado que identifica el tipo de respuesta el cual está compuesto por 3 dígitos. Los códigos se clasifican según su primer dígito, así:

- 1xx → Informativo
- 2xx → Éxito
- 3xx → Redirección
- 4xx → Error del cliente
- 5xx → Error del servidor

3.3.2 NodeJS

“Node.js es un entorno de ejecución de un solo hilo, de código abierto y multiplataforma para crear aplicaciones de red y del lado del servidor rápidas y escalables. Se ejecuta en el motor de ejecución de JavaScript V8, y utiliza una arquitectura de E/S basada en eventos y sin bloqueos, lo que la hace eficiente y adecuada para aplicaciones en tiempo real.” (OpenJS Foundation, n.d.)

3.4 Backend

A diferencia del frontend, el backend está del lado del servidor del sitio web, lo que significa que será el encargado de almacenar y organizar los datos provenientes de la interfaz de usuario. Esta información la combina y transforma de una forma que resulte útil para el usuario y luego la devuelve a través de la interfaz gráfica, garantizando así el buen funcionamiento del sitio (Interviewbit, 2022). Si bien este proyecto se basa en la creación del frontend de la aplicación, es importante conocer las funciones del backend para lograr la correcta integración entre ambas partes.

3.5 Black box testing

Black Box Testing es un método de prueba de software en el que las funcionalidades de las aplicaciones de software se prueban sin tener conocimiento de la estructura del código, los detalles de implementación ni las rutas internas. Este método se enfoca principalmente en las entradas y salidas de aplicaciones web y se basa completamente en los requisitos y especificaciones de la aplicación (Nidhra, S., & Dondeti, J., 2012). Para este proyecto se realizaron dos tipos de pruebas de caja negra:

- Prueba funcional: este tipo de prueba de caja negra está relacionada con los requisitos funcionales de un sistema.
- Pruebas no funcionales: este tipo de prueba de caja negra no está relacionada con la prueba de una funcionalidad específica, sino con los requisitos no funcionales, como el rendimiento, la escalabilidad y la facilidad de uso.

4. Metodología

La aplicación de Acceso a Palcos se encuentra dividida en dos partes principales, el frontend y el backend. Como se dijo anteriormente, este proyecto está enfocado en el desarrollo frontend de la aplicación al igual que en el desarrollo de la API REST encargada de la integración entre el frontend y el backend; y para esto, el trabajo se dividió en 3 partes principales que son: planteamiento de requerimientos, desarrollo frontend, creación de la API REST. A continuación, se describirán cada una de las partes que se llevaron a cabo para la culminación del proyecto.

4.1 Requisitos

4.1.1 Requisitos funcionales

A continuación, se presentarán los requerimientos funcionales de la aplicación web, además de la historia de usuario y los criterios de aceptación que debe cumplir cada página de la aplicación web al finalizar el proyecto.

Tabla 1*Inicio de Sesión*

Permite acceder al personal de la empresa encargado de los palcos, a la aplicación web “Acceso a Palcos”

Identificador	Nombre	Versión
FR 1	Inicio de sesión	1.0

Historia de usuario

Yo como usuario deseo iniciar sesión para hacer uso de la aplicación web

Criterios de aceptación

- El campo de usuario en el formulario de inicio de sesión no debe soportar el uso de símbolos especiales, por lo que todo caracter que no cumpla con lo establecido no deberá mostrarse.
- Si el usuario presiona el botón de ingreso y los campos del formulario se encuentran vacíos, se debe mostrar un mensaje informando al usuario que escriba usuario y contraseña.
- Si el usuario presiona el botón de ingreso, pero el usuario o la contraseña no coinciden con los datos de ingreso, se debe mostrar un mensaje informando al usuario que alguno de los datos escritos está erróneo.

Tabla 2*Ver reservas activas*

Permite al personal de la empresa encargado de los palcos, visualizar el mapa de ocupación de cada zona

Identificador	Nombre	Versión
FR 2	Ver reservas activas	1.0

Historia de usuario

Yo como usuario deseo saber la ocupación de cada zona de los palcos y así saber cuántos están disponibles para reserva

Criterios de aceptación

- La página debe desplegar un mapa de asientos por cada zona perteneciente a los palcos. Cada una debe mostrar un conjunto de estadísticas donde se enumera cuántos de esos asientos han sido reservados, bloqueados o se encuentran disponibles.
- La página debe desplegar en la parte superior un conjunto de estadísticas para el total de zonas pertenecientes a los palcos. Allí se deben enumerar el total de asientos que han sido reservados, bloqueados o se encuentran disponibles.
- Cada asiento debe mostrar a través de los colores elegidos, cuando están reservados, bloqueados o libres.

Tabla 3*Administración de credenciales*

Permite al personal de la empresa encargado de los palcos, agregar, eliminar o editar las credenciales de los usuarios que cuentan con acceso a los palcos del estadio Atanasio Girardot

Identificador	Nombre	Versión
FR 3	Administrador de credenciales	1.0

Historia de usuario

Yo como usuario deseo administrar las credenciales de las personas que cuentan con acceso a los palcos, dependiendo de las modificaciones que se pidan realizar desde la Alcaldía de Medellín.

Criterios de aceptación

- Si el usuario desea agregar una nueva persona, debe poder escribir el nombre, identificación y número de credencial de la persona en el formulario. Al dar clic en el botón agregar, se debe desplegar la nueva credencial al final de los datos existentes.
- Si el usuario desea agregar una nueva persona, pero ya existe alguien con la misma identificación, se debe mostrar un mensaje informando que esta persona ya se encuentra en la lista. Así mismo, si se desea agregar una nueva persona, pero ya existe alguien con el mismo número de credencial, se debe mostrar un mensaje informando que ese número de credencial ya se encuentra en la lista.
- Si el usuario excede el número de credenciales permitidas por el personal del estadio (100 credenciales en total), se debe mostrar un mensaje informando que se excedió el límite de credenciales.
- Se debe desplegar una tabla que contenga las credenciales de las personas que ya se encuentran agregadas en la base de datos. Al frente de cada credencial debe existir un botón de editar y borrar que permitan realizar las modificaciones pertinentes.

Tabla 4*Bloqueo de asientos*

Permite al personal de la empresa encargado de los palcos, bloquear asientos que no se consideren aptos para el uso de las personas con acceso a los palcos. Así como desbloquearlos una vez se encuentren listos para ser ofrecidos nuevamente al público.

Identificador	Nombre	Versión
FR 4	Bloqueo de asientos	1.0

Historia de usuario

Yo como usuario deseo poder bloquear o desbloquear los asientos que considere necesarios.

Criterios de aceptación

- Cuando el usuario acceda a la página esta debe mostrar el mapa de asientos por zona, mapa que debe coincidir en posición y numeración con el mapa real. Además de mostrar los asientos que ya fueron bloqueados o reservados a través de los colores elegidos.
- Cuando el usuario seleccione un asiento de cualquier zona que se encuentre libre, este debe tornarse del color elegido por el diseñador gráfico para un asiento bloqueado. Una vez seleccionados los asientos, al dar clic al botón de bloquear, se debe mostrar un mensaje informando que los asientos fueron bloqueados con éxito.
- Cuando el usuario seleccione un asiento de cualquier zona que ya se encuentre bloqueado, este debe tornarse del color elegido por el diseñador gráfico para un asiento libre. Una vez deseccionados los asientos, al dar clic al botón de desbloquear, se debe mostrar un mensaje informando que los asientos fueron desbloqueados con éxito.
- El usuario no puede elegir para bloquear un asiento que ya se encuentre en estado reservado.

Tabla 5*Reserva de asientos*

Permite al personal de la empresa encargado de los palcos, reservar el asiento que la persona elija, no sin antes realizar la validación de la credencial de la persona para verificar su existencia en la base de datos.

Identificador	Nombre	Versión
FR 5	Reserva de asientos	1.0

Historia de usuario

Yo como usuario deseo poder reservar el asiento que la persona de la alcaldía elija.

Criterios de aceptación

- Antes de acceder al mapa de asientos para realizar la reserva, se deben poder validar los datos de la credencial de la persona que desea reservar. Si los datos coinciden con los de la base de datos, al dar clic al botón de validar, el usuario puede acceder a la página de reservas. Si los datos no coinciden con los de la base de datos, se debe mostrar un mensaje informando que la credencial no es válida. Si el usuario da clic al botón cancelar, se deberán borrar los datos ingresados con anterioridad.
- Cuando el usuario pueda acceder a la página esta debe mostrar el mapa de asientos por zona, mapa que debe coincidir en posición y numeración con el mapa real. Además de mostrar los asientos que ya fueron bloqueados o reservados a través de los colores elegidos.
- El usuario puede reservar máximo dos asientos por credencial. Al tratar de elegir más, no se seleccionará el asiento hasta que otro sea deseleccionado.
- Cuando el usuario seleccione un asiento que se encuentre libre de cualquier zona, este debe tornarse del color elegido por el diseñador gráfico para un asiento reservado. Una vez seleccionados los asientos, al dar clic al botón de reservar, se debe mostrar un mensaje informando que los asientos fueron reservados con éxito.
- Cuando el usuario se encuentre viendo el mapa de reservas y presione el botón de volver, se debe dirigir a la página de validación de credencial.

Tabla 6*Modificación de reserva*

Permite al personal de la empresa encargado de los palcos, modificar una reserva realizada, no sin antes realizar la validación de la credencial de la persona para verificar su existencia en la base de datos.

Identificador	Nombre	Versión
FR 6	Modificación de reserva	1.0

Historia de usuario

Yo como usuario deseo poder modificar una reserva realizada anteriormente.

Criterios de aceptación

- Antes de acceder al formulario de modificación de reserva, se deben poder validar los datos de la credencial de la persona que desea modificar la reserva. Si los datos coinciden con los de la base de datos, al dar clic al botón de validar, el usuario podrá acceder a la página de reservas. Si los datos no coinciden con los de la base de datos, se debe mostrar un mensaje informando que la credencial no es válida. Si el usuario da clic al botón cancelar, se deben borrar los datos ingresados con anterioridad.
- Cuando el usuario pueda acceder a la página de modificación de reserva, se cargan en un formulario los datos de la persona con credencial. Los datos que deben ser mostrados son: nombre completo, identificación, número de credencial y el o los asientos que reservó con anterioridad.
- Si la persona con credencial desea modificar la reserva, el usuario debe dar clic al botón de modificar lo que lo dirigirá a la página de reserva de asientos sin necesidad de realizar la validación de credencial de la persona.
- Si la persona con credencial desea cancelar la reserva, el usuario debe dar clic al botón de cancelar y se debe mostrar un mensaje informando que la reserva fue cancelada con éxito

4.1.2 Requisitos no funcionales

A continuación, se presentarán los requerimientos no funcionales y su respectivo criterio conceptual para la aplicación web.

Tabla 7

Requerimiento no funcional 1

Identificador	Nombre	Versión
RNF 1	Mantenibilidad	1.0
Criterio conceptual		
El frontend de la aplicación web debe ser fácil de mantener de modo que corregir defectos o agregar nuevas funcionalidades sea una tarea sencilla.		

Tabla 8

Requerimiento no funcional 2

Identificador	Nombre	Versión
RNF 2	Usabilidad	1.0
Criterio conceptual		
El frontend de la aplicación web debe ser amigable con el usuario encargado. Se entiende por amigable a la facilidad de uso de un sitio web o una aplicación. Es un indicador que muestra si un sitio web es efectivo, fácil de navegar y eficiente desde el punto de vista de los usuarios.		

Tabla 9

Requerimiento no funcional 3

Identificador	Nombre	Versión
RNF 3	Adaptabilidad	1.0
Criterio conceptual		
El contenido de la aplicación web debe ser adaptable a cualquier tamaño de pantalla.		

Tabla 10

Requerimiento no funcional 4

Identificador	Nombre	Versión
RNF 4	Diseño gráfico	1.0

Criterio conceptual

El contenido de la aplicación web debe cumplir con las especificaciones de diseño presentadas en la siguiente lista:

- Colores asignados para asientos
 - Asiento libre #c4c4c4
 - Asiento reservado #39aa40
 - Asiento bloqueado #e9b418
- Fuentes sugeridas (El tamaño lo decide el desarrollador)
 - Metropolis regular
 - Metropolis bold
 - Metropolis semi - bold
- Los logos tanto del INDER como del estadio serán suministrados por el área de diseño gráfico.

4.2 Frontend

4.2.1 Mockup

El primer paso para la creación de la aplicación web fue la realización del mockup con ayuda de la herramienta online de generación de prototipos, Figma (<https://www.figma.com/>). Esta permitió, de una manera sencilla, crear un prototipo adecuado siguiendo los requerimientos exigidos por la empresa, el cual contiene la estructura, colores y funcionamiento de botones de cada página de la aplicación. Es importante destacar que el mockup creado muestra el funcionamiento de aquellos

botones destinados a redirigir al usuario a una página nueva o de los botones de validación como el de reservar de la página de Reservar o el de bloquear de la página de Bloquear Asientos. Los requerimientos establecidos para formularios o envío de formularios no se llevan a cabo debido a que la herramienta no permite realizar validaciones. A continuación, se expone una muestra del mockup. Los demás mockups se mostrarán en la sección de Resultados.

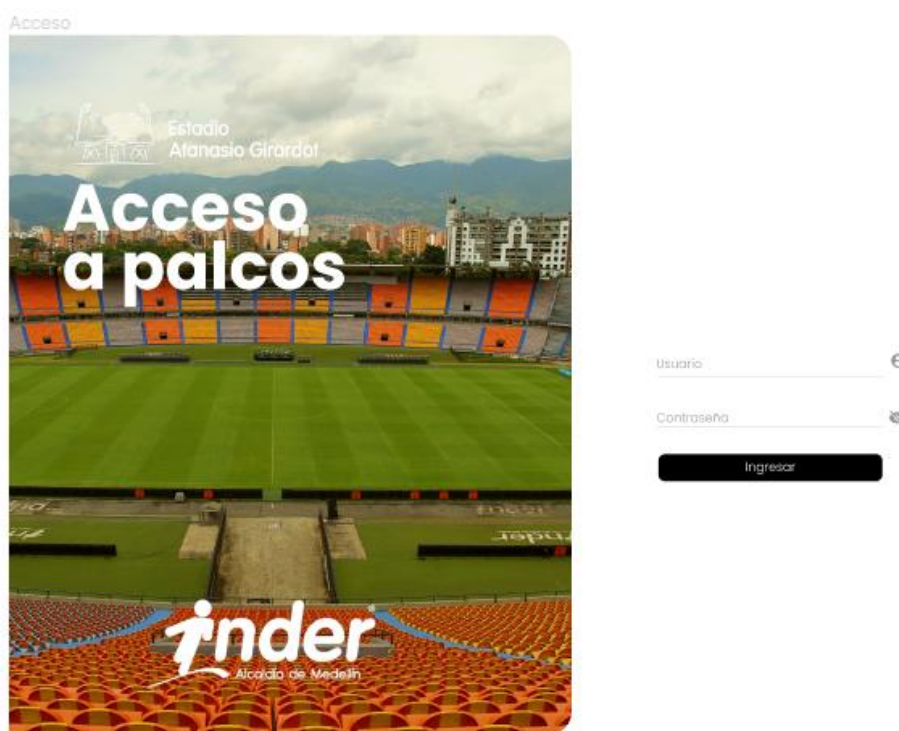


Figura 2. Mockup para inicio de sesión.

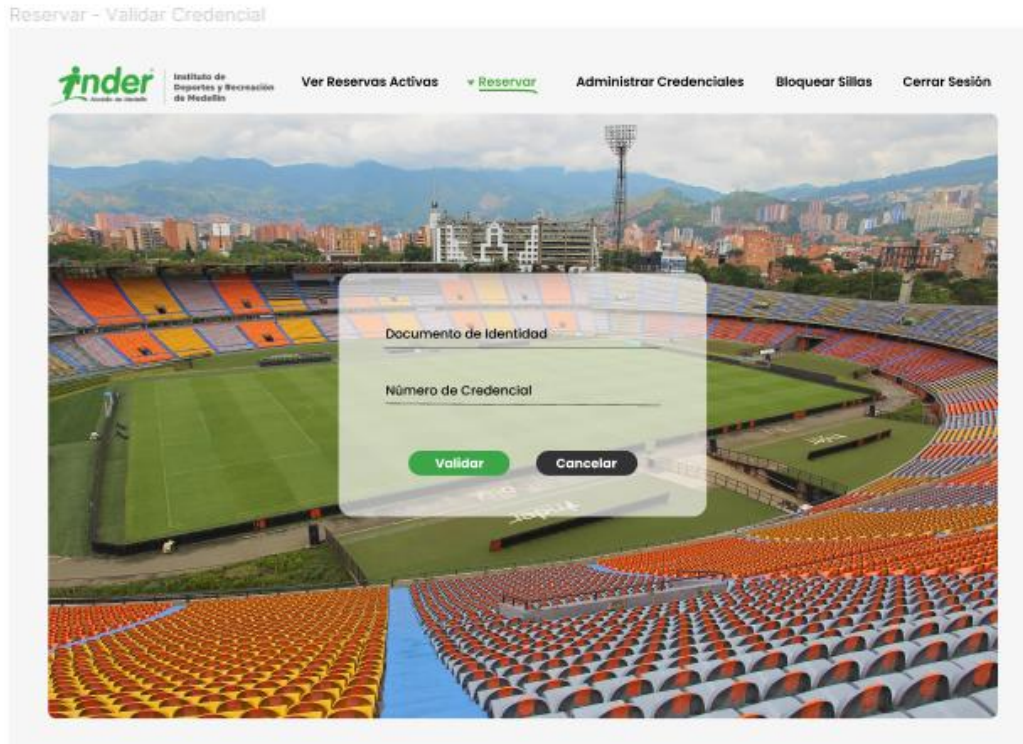


Figura 3. Mockup para validación de credencial.

4.2.2 Desarrollo frontend

El frontend fue construido a partir del mockup creado con anterioridad, utilizando múltiples componentes de React. Los componentes son pequeños pedazos de código independientes entre sí que poseen su propia lógica y controles, devuelven código HTML y se pueden reutilizar cuando se requiera, haciendo que los errores de implementación en el código sean más simples de hallar y que la aplicación sea más fácil de desarrollar y mantener. Las páginas hechas para este proyecto y sus respectivos componentes se listan a continuación.

4.2.2.1 Iniciar sesión

La página de inicio de sesión es la primera página con la que se encuentra el usuario una vez abre la aplicación. Contiene un componente que alberga un formulario donde se solicitan los datos de usuario único para dar acceso a la aplicación. El componente consulta al servidor mediante la API y devuelve alguna de las siguientes opciones:

- Si los datos ingresados coinciden con los albergados en la base de datos, el usuario será redirigido al componente de Ver Reservas Activas.
- Si los datos ingresados no coinciden con los albergados en la base de datos, el componente devolverá un mensaje de error diciendo “Usuario o contraseña incorrectos” al presionar el botón de ingresar.
- Si no se ingresan los datos y se presiona el botón de ingresar, el componente devuelve un mensaje de error diciendo “Por favor, ingrese Usuario y Contraseña”.

En la Figura 4 se muestra el diagrama de bloques perteneciente a la página de Inicio de Sesión.

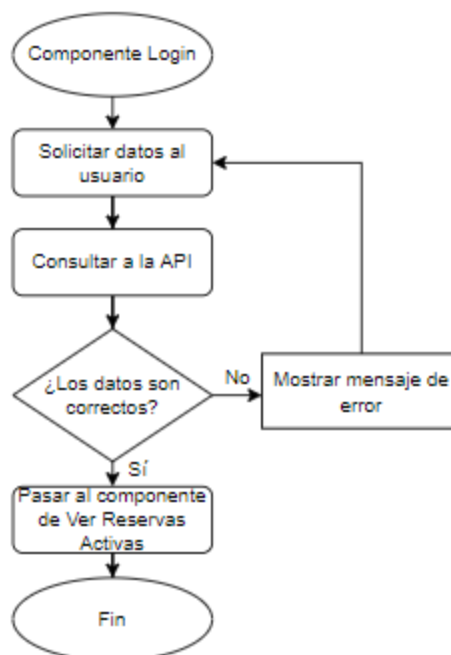


Figura 4. Diagrama de flujo para componente de Inicio de Sesión.

4.2.2.2 Ver reservas activas

Para esta página se creó un componente que consulta al servidor para saber el estado de cada asiento y devuelve un carrusel el cual contiene el mapa de asientos y estadísticas de ocupación por cada zona y así saber cuáles están libres, reservados o bloqueados. De igual forma, muestra una estadística general por el total de asientos donde se visualizan los datos mencionados anteriormente, ubicada en la parte superior de la página. En la Figura 4 se muestra el diagrama de bloques perteneciente a la página de Ver Reservas Activas.

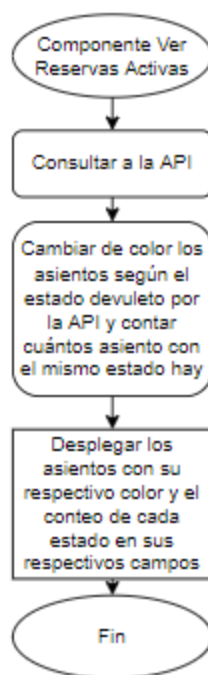


Figura 5. Diagrama de flujo para componente de Ver Reservas Activas.

4.2.2.3 Reservar

La página de Reservar está dividida en dos partes, una para la reserva de asientos y otra para modificar o cancelar una reserva creada con anterioridad. Para realizar cualquiera de las acciones mencionadas primero se debe verificar la validez de la credencial del usuario, esto se hace por medio del componente Validación donde, a través de un formulario que solicita el documento de identidad y número de credencial del usuario, se consulta al servidor mediante la API y se valida si la persona sí se encuentra en ella. De no ser así, el componente devuelve un mensaje de error de validación. Por el contrario, si la validación es correcta, continúa el proceso de reserva o modificación de reserva según sea el caso. El diagrama para el componente Validación se muestra en la Figura 6

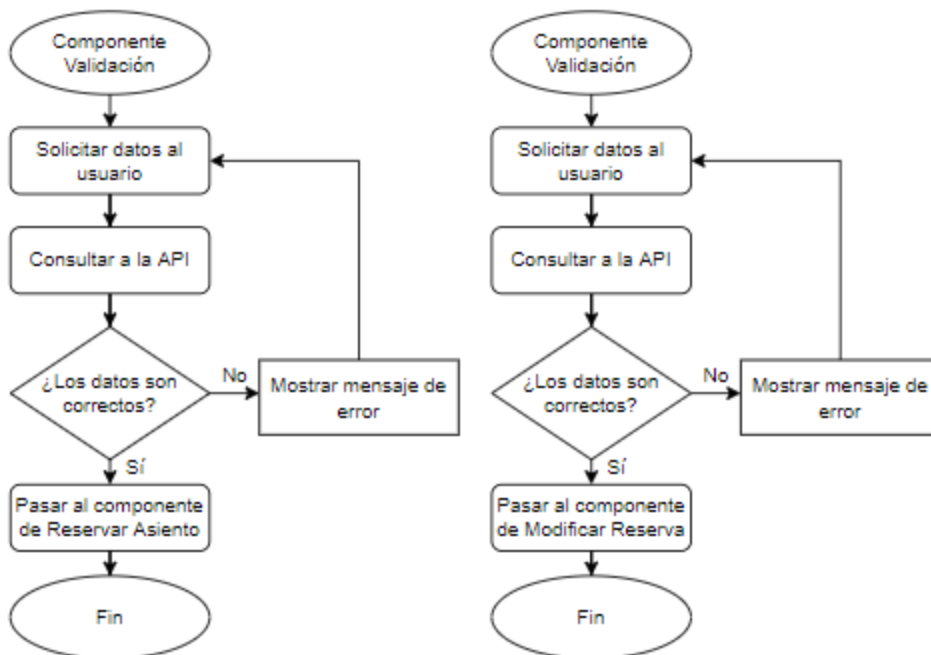


Figura 6. Diagrama de flujo para componente de Validación de Credencial.

Luego de la validación, si el usuario desea realizar una reserva, el componente Reservar Asiento se encarga de consultar al servidor mediante la API para conocer el estado de los asientos y luego devolver un carrusel con el mapa y las vistas de cada palco para que el usuario seleccione el lugar que desea escoger la persona que cuenta con la credencial. Una vez escogido el asiento, al presionar el botón de reservar se despliega un modal para la confirmación o cancelación del proceso; este modal se encuentra albergado dentro del componente Modal. Al dar clic en el botón confirmar, los asientos reservados cambian de estado en la base de datos. El diagrama para los componentes de Reservar Asiento y Modal se muestran en las Figuras 7 y 8, respectivamente

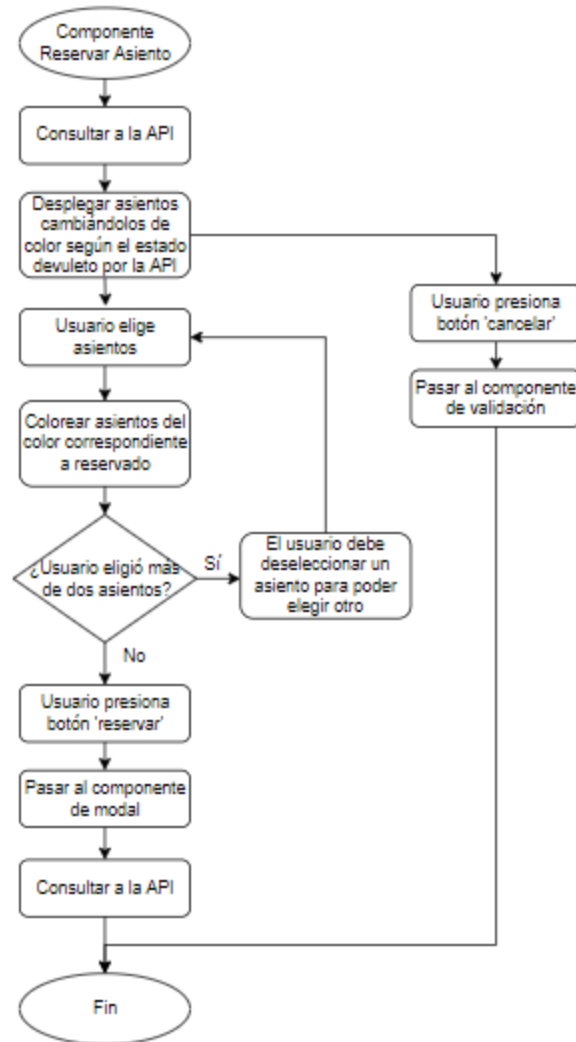


Figura 7. Diagrama de flujo para componente de Reservar Asiento.

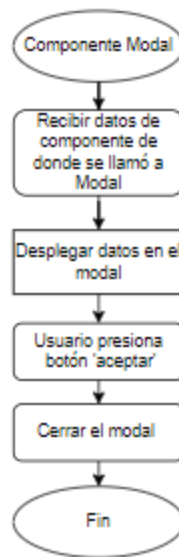


Figura 8. Diagrama de flujo para componente Modal

En cambio, si la reserva ya fue realizada y el usuario desea cancelarla o modificarla, el componente de Modificar Reserva consulta al servidor y despliega un formulario con los datos de la reserva y dos botones que permiten modificarla o cancelarla. Si se presiona el botón de modificar, el componente se redirige a la página de reservas, si se elige eliminar, el componente envía la solicitud al backend mediante la API y se elimina la reserva cambiando también el estado del asiento. El diagrama para el componente Modificar Reserva se muestra en la Figura 9

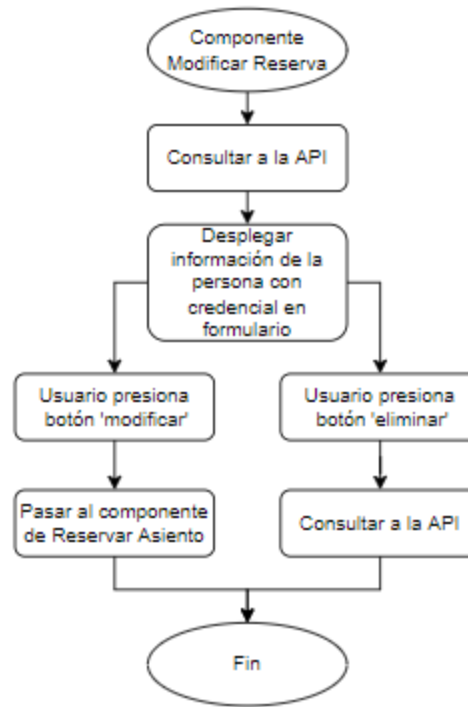


Figura 9. Diagrama de flujo para componente Modificar Reserva.

4.2.2.4 Administrar credenciales

El componente principal de esta página hace uso de componentes externos que brindan funcionalidad completa a la sección. El componente principal, Administrar Credenciales, consulta al servidor y llama al componente Solo Lectura el cual escribe en las filas de una tabla los datos de personas que cuentan con credencial y al lado de cada persona agrega dos botones que ofrecen diferentes funcionalidades de las que se hablarán más adelante: el botón borrar permite eliminar de la base de datos la persona y el botón editar permite modificar los datos de la persona. Una vez termina, se devuelve al componente Administrar Credenciales.

El componente de Administrar Credenciales permite agregar una persona nueva a la tabla. Para esto, el componente despliega un formulario que recibe los datos de la persona que se desea agregar. Al ingresar los datos y presionar el botón de agregar en el formulario, los datos de la persona nueva se envían al servidor para ser agregado a la base de datos que contiene la lista y se le adiciona un código de identificación único generado aleatoriamente. Si al intentar agregar un usuario se excede el número permitido, el componente despliega un mensaje dando a entender que el límite se ha superado y que no es posible completar la acción.

La acción de borrar una credencial también se hace posible desde el componente de Administrar Credenciales. Para ello, cuando se da clic al botón de borrar, el componente se encarga de tomar el id de la persona que se desea eliminar y lo envía al servidor quién se encarga de filtrar la base de datos y borrar los datos correspondientes al id del usuario. Al final, devuelve una nueva lista en la tabla de administración de credenciales. El diagrama tanto del componente de Administrar Credenciales como el de Solo Lectura se pueden observar en las Figuras 10 y 11, respectivamente.

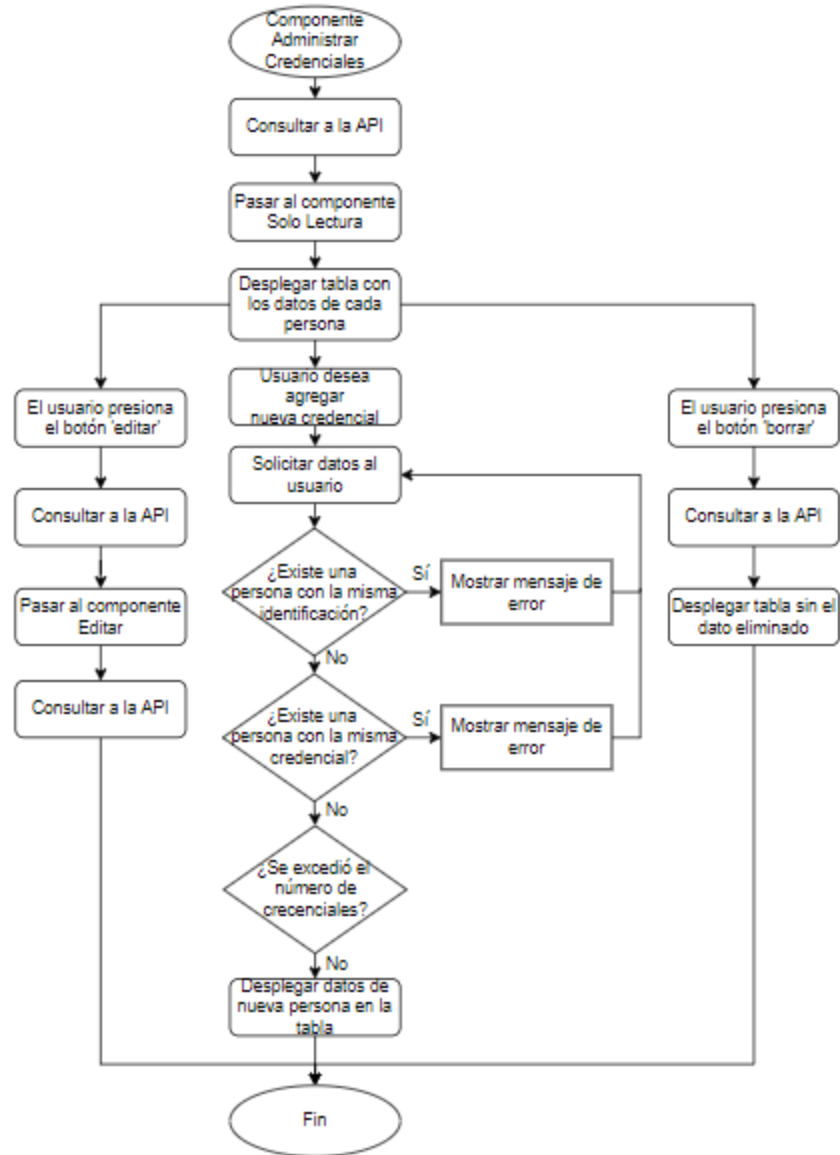


Figura 10. Diagrama de flujo para componente Administrar Credenciales.

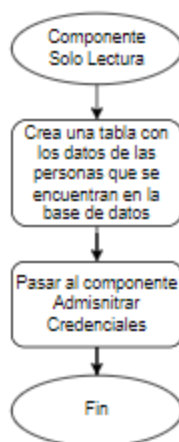


Figura 11. Diagrama de flujo para componente Solo Lectura.

Para la acción de editar, el componente Administrar Credenciales llama al componente Editar. Este componente se encarga de cambiar los elementos de sólo lectura de la tabla por formularios editables que cargan la información del usuario, cambiando también los botones de agregar y borrar por los botones de guardar y cancelar. De esta forma se pueden editar los datos deseados y el servidor puede guardar los cambios en la en la base de datos o cancelar la edición en caso ya no requerirse; al presionar cualquiera de los botones guardar o cancelar, el componente regresa la tabla a datos de solo lectura, con sus respectivos botones de agregar y borrar. El diagrama de flujo del componente Editar se muestra en la Figura 12.

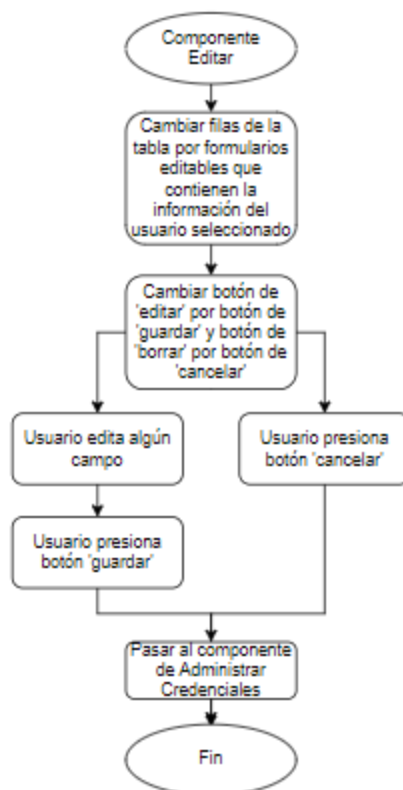


Figura 12. Diagrama de flujo para componente Editar.

4.2.2.5 Bloquear asientos

Si el usuario desea bloquear asientos, el componente Bloquear Asientos se encarga de consultar al servidor el estado de los asientos para luego devolver un carrusel con el mapa de asientos y las vistas de cada palco. El usuario procede a elegir los asientos que desea bloquear y el componente se encarga de colorear los asientos del color correspondiente a bloqueo; luego al presionar el botón de bloquear, el componente Bloquear Asientos llama al componente Modal para desplegar un mensaje informando que los asientos fueron bloqueados con éxito. Ahora, si el usuario desea desbloquear asientos, simplemente debe deseleccionar los asientos que ya se encuentran bloqueados y luego presionar el botón de desbloquear, así el componente Bloquear Asientos se encarga de llamar al

componente Modal para desplegar un mensaje informando que los asientos fueron desbloqueados con éxito y de colorear los asientos del color correspondiente a libre. El diagrama de flujo del componente Bloquear Asientos se muestra en la Figura 13.

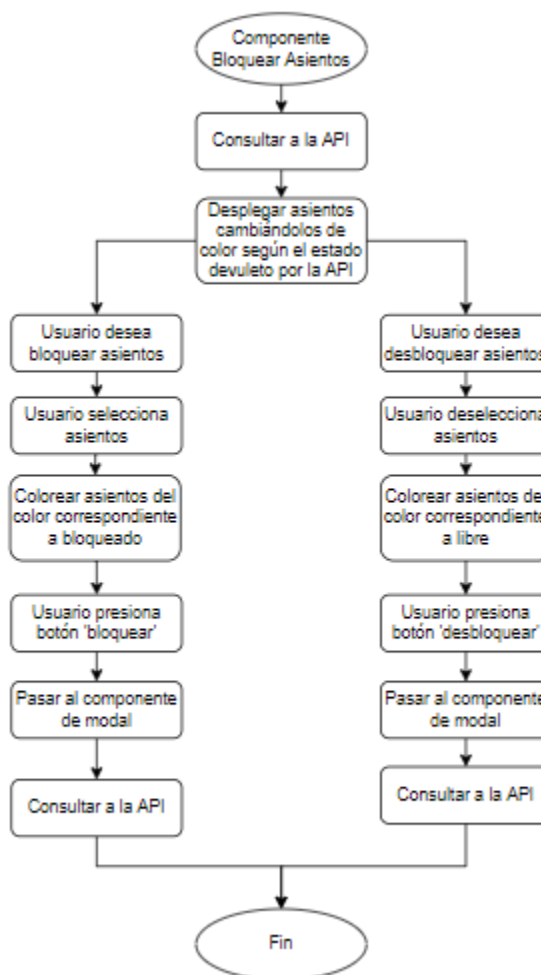


Figura 13. Diagrama de flujo para el componente Bloquear Asientos.

4.3 API REST

La API REST fue desarrollada utilizando Node JS, framework escrito en Javascript. Esta realiza una petición y devuelve un valor o un grupo de valores dependiendo de la solicitud realizada; los diferentes endpoints creados para la API REST al igual que su diagrama de flujo se exponen a continuación.

4.3.1 /users/signIn

Este Endpoint permite a través de una petición POST verificar las credenciales del usuario en la página de inicio de sesión. Si el usuario, por equivocación, da clic en ingresar sin haber ingresado los datos de acceso, el servidor devolverá un error HTTP 400 seguido por el mensaje de error “Por favor, ingrese usuario y contraseña.”. Si el usuario ingresa algún dato incorrecto, el servidor devolverá un error 401 indicando que la petición no ha sido ejecutada porque carece de credenciales válidas de autenticación para el recurso solicitado, seguido por el mensaje de error “Usuario o contraseña incorrectos.”. En caso de que el usuario ingrese los datos correctos, el servidor retornará un token de acceso al dar clic en ingresar dando así acceso a la aplicación. El diagrama de flujo para este endpoint se muestra en la Figura 14.

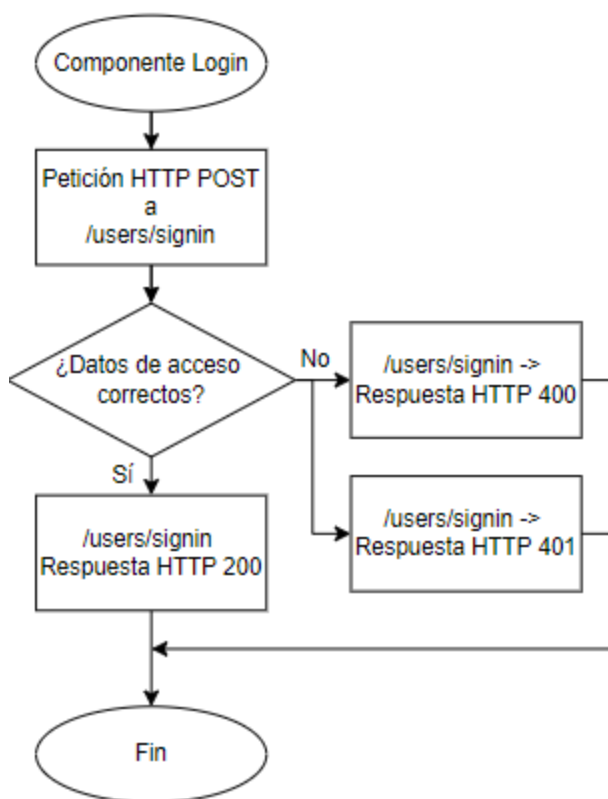


Figura 14. Diagrama de flujo para el EndPoint /users/signin.

4.3.2 /credentials

Este Endpoint permite a través de cuatro peticiones, GET, POST, PUT y DELETE modificar de diversas formas las credenciales de los usuarios desde la página de administración de credenciales. Primero, realizando una petición GET al servidor se solicita la lista con los datos de cada usuario y este retorna un arreglo con nombre, documento de identidad y número de credencial los cuales se visualizan en forma de tabla en la página. Para agregar una credencial utilizando el formulario ubicado en la parte superior de la página se realiza una petición POST al servidor. Éste se encarga de almacenar la nueva credencial en la base de datos y envía como respuesta la misma credencial enviada indicando que el proceso se realizó correctamente. Si el formulario se

envía con campos vacíos el servidor devuelve un error HTTP 400 (bad request) indicando que no puede procesar la petición debido a que percibió un error del cliente.

Para editar una credencial se realiza una petición PUT al servidor con el id de la credencial elegida donde el cuerpo de la petición lleva la nueva información suministrada y la actualiza en la base de datos. Como respuesta, el servidor devuelve la misma credencial indicando que el proceso se realizó correctamente y se ve reflejada en la tabla de la página.

Por último, para eliminar una credencial se realiza una petición DELETE al servidor con el id de la credencial elegida. Éste se encarga de eliminar de la base de datos al usuario que corresponde al id. Si la credencial no existe, el servidor devuelve un error HTTP 400. El diagrama de flujo para este endpoint se muestra en la Figura 15.

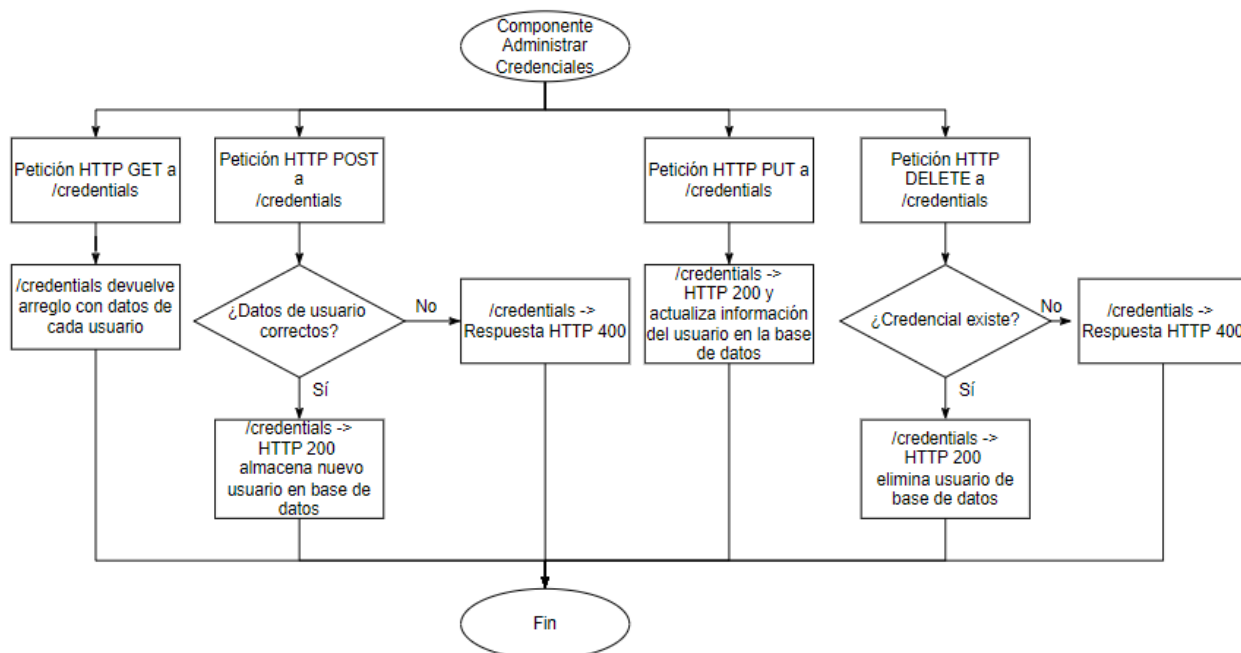


Figura 15. Diagrama de flujo para el EndPoint /credenciales.

4.3.3 /credentials/id

Endpoint llamado desde el componente de validación de credenciales utilizado en la página de reservas, el cual permite consultar la validez de una credencial antes de realizar la reserva de un asiento, esto se logra mediante una petición GET con la que a través del id del usuario el servidor consulta si la credencial existe o no en la base de datos. Si existe permite el acceso a los mapas de palcos para continuar con el proceso de reserva, de lo contrario, devuelve el mensaje de error “Credencial no válida”. El diagrama de flujo para este endpoint se muestra en la Figura 16.

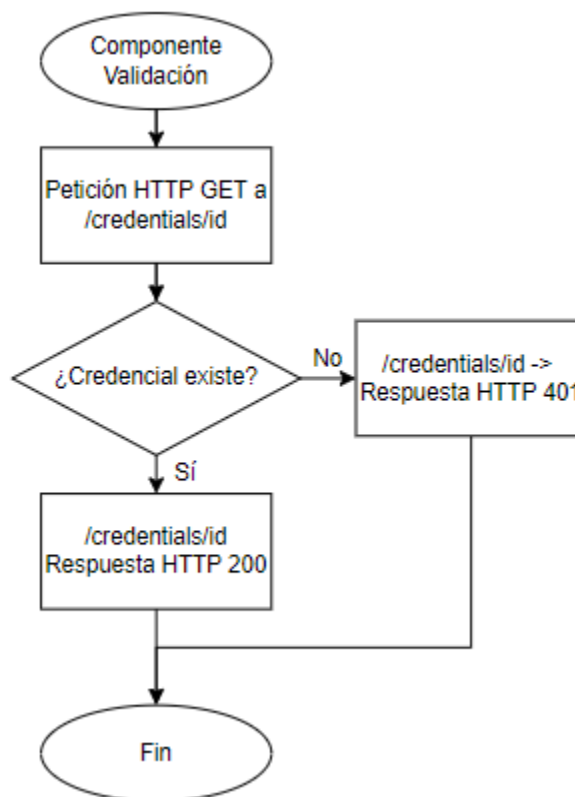


Figura 16. Diagrama de flujo para el EndPoint /credentials/id.

4.3.4 /bookings

Endpoint que permite a través de una petición GET traer el estatus de los asientos, sea libre, reservado o bloqueado, para así asignarles el color correspondiente. Este Endpoint es llamado desde cada página que utilice un mapa de asientos, reservas, bloqueo de asientos y ver reservas activas. El diagrama de flujo para este endpoint se muestra en la Figura 17.

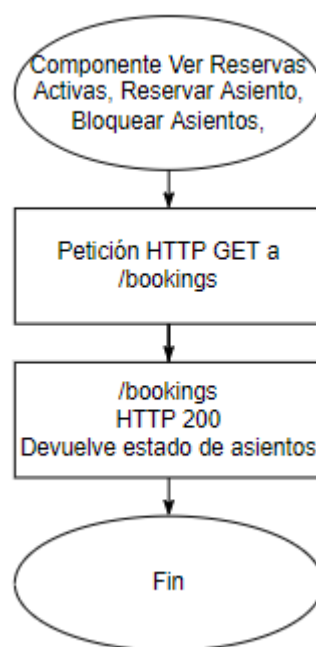


Figura 17. Diagrama de flujo para el EndPoint /bookings.

4.3.5 /bookings/id

Endpoint llamado desde las páginas de Reservar y Bloqueo de Asientos que permite hacer ambas peticiones PUT y GET para realizar diferentes acciones.

Desde el componente de Reservar, al elegir el asiento y completar el proceso de reserva, se envía una petición PUT al servidor el cual se encarga de modificar el estatus

del asiento a “reservado” y asignarlo al id de la persona con la que se realizó el proceso de validación de credencial.

Desde Modificar Reserva se realiza una petición GET al servidor para traer los datos de la persona al formulario. Para cancelar una reserva se envía una petición PUT al servidor para que asigne al asiento un estatus disponible y elimine la asignación del asiento al usuario.

Para el bloqueo de asientos, luego de seleccionar los asientos deseados y dar clic en bloquear, al igual que en la página de reservas, se realiza una petición PUT al servidor que se encarga de modificar el estatus del asiento a “bloqueado”. El diagrama de flujo para este endpoint se muestra en la Figura 18.

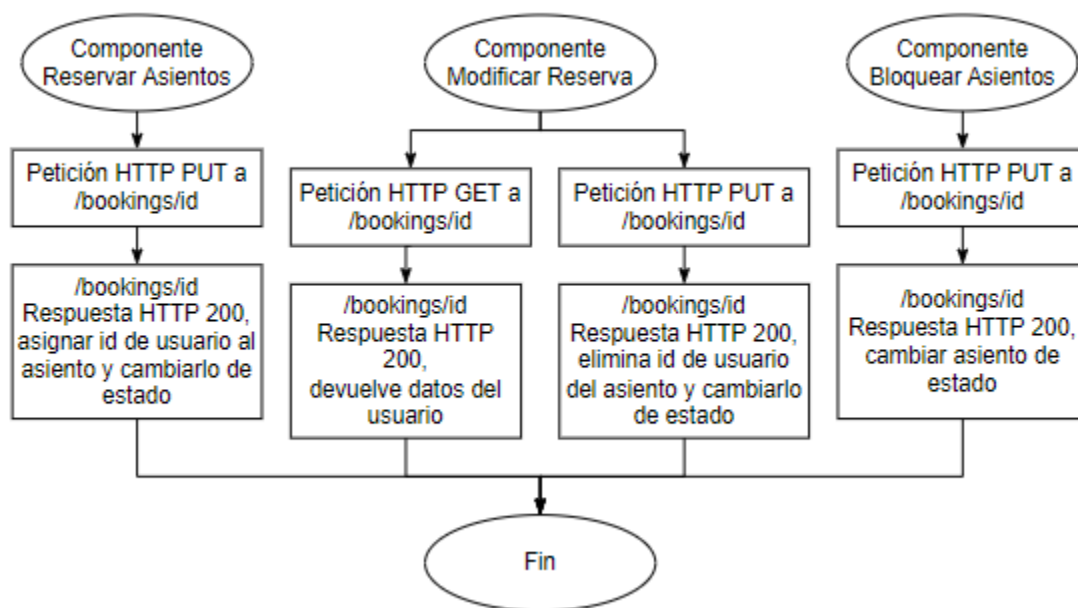


Figura 18. Diagramas de flujo para el EndPoint /bookings/id.

5. Resultados

El resultado de este trabajo es un desarrollo frontend completamente funcional, capaz de establecer comunicación con el backend a través de la API REST creada, cumpliendo así con los requisitos establecidos con la empresa.

5.1 Mockup

A continuación, se presenta el resultado completo, página a página, de la creación del Mockup usando la herramienta Figma. Este diseño se creó en conjunto con el área de diseño gráfico del INDER Medellín, se fijaron colores, tamaños y tipos de letra y posición de los asientos, botones y formularios de modo que la aplicación resultara amigable con el empleado de la empresa encargado de utilizar la aplicación quién sería el usuario final.

El mockup para la página de inicio de sesión se muestra en la Figura 19



Figura 19. Mockup para inicio de sesión.

El mockup para la página que permite la validación de credencial para reservar asiento o modificar reservas se muestra en la Figura 20.

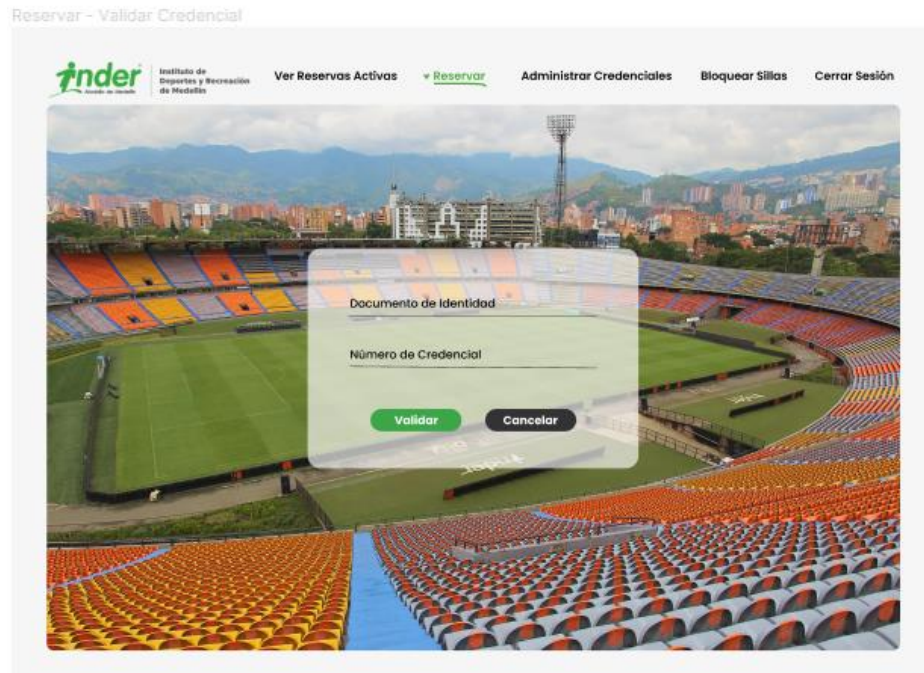


Figura 20. Mockup para validación de credencial.

El mockup creado para la página de reservar asientos se muestra desde las Figura 21 hasta la 25.

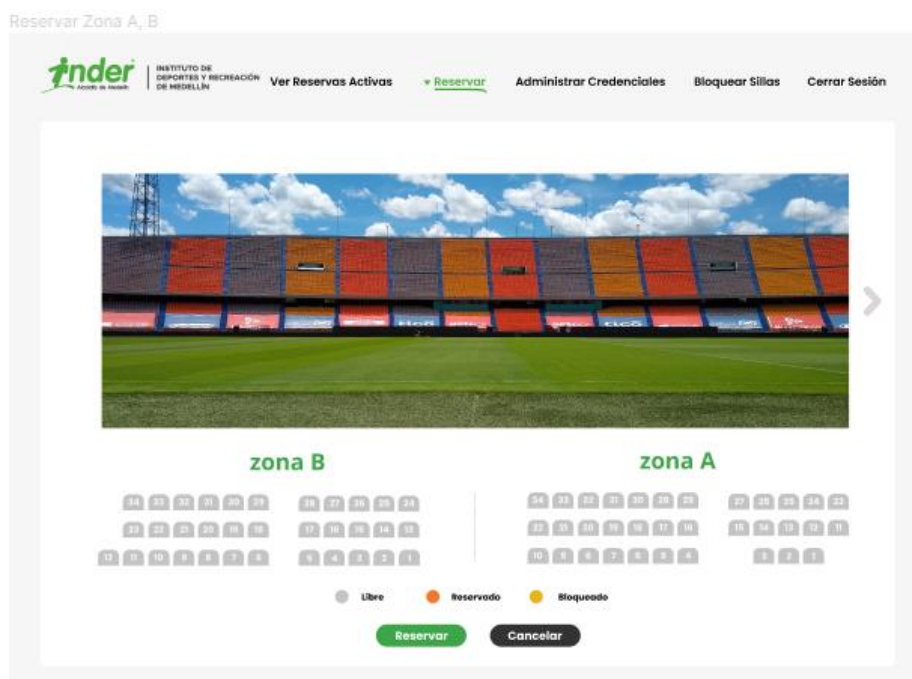


Figura 21. Mockup para reservar, zonas A y B.

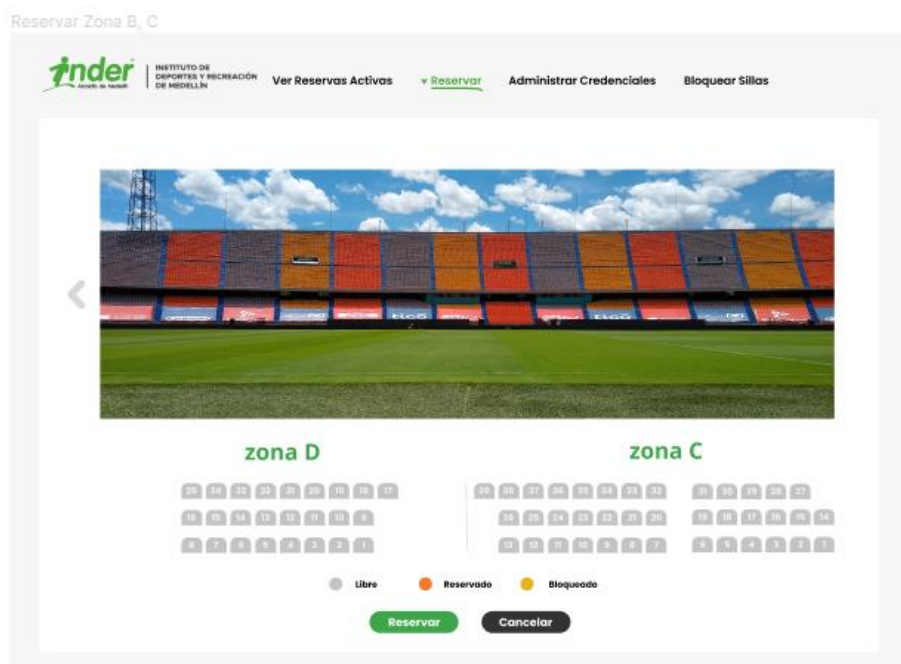


Figura 22. Mockup para reservar, zonas A y B.

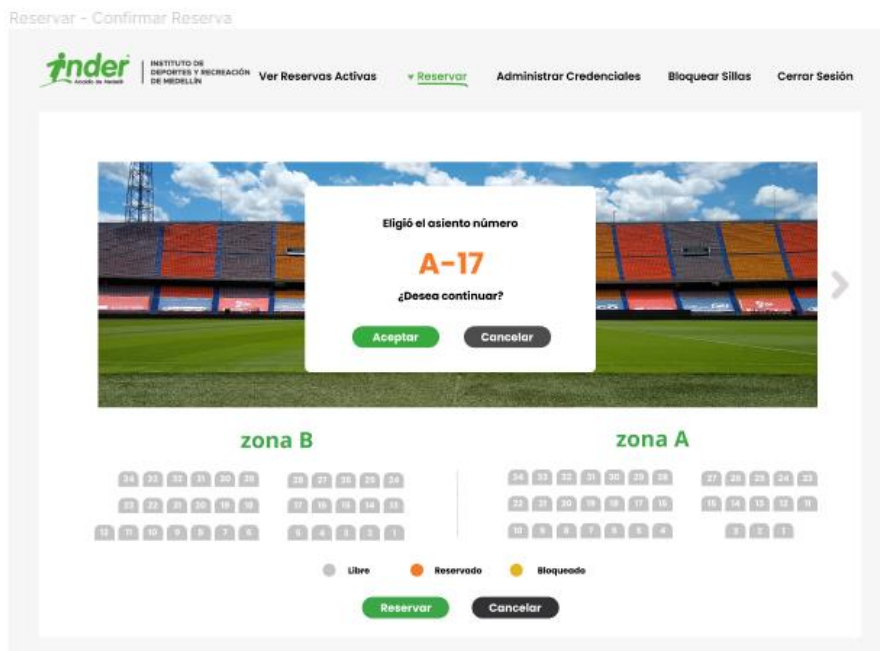


Figura 23. Mockup para reservar, zonas C y D.

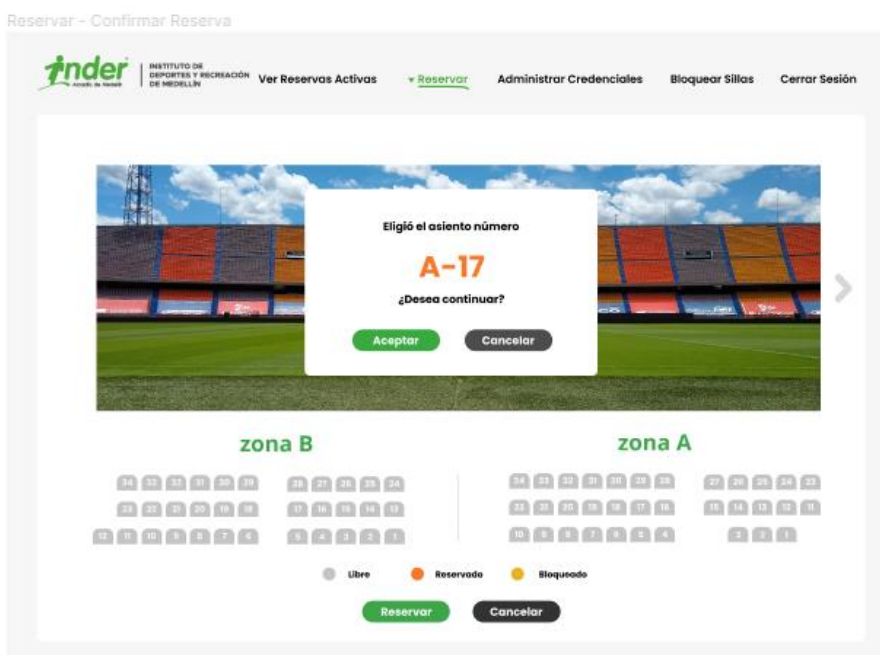


Figura 24. Mockup para ventana emergente de confirmación.

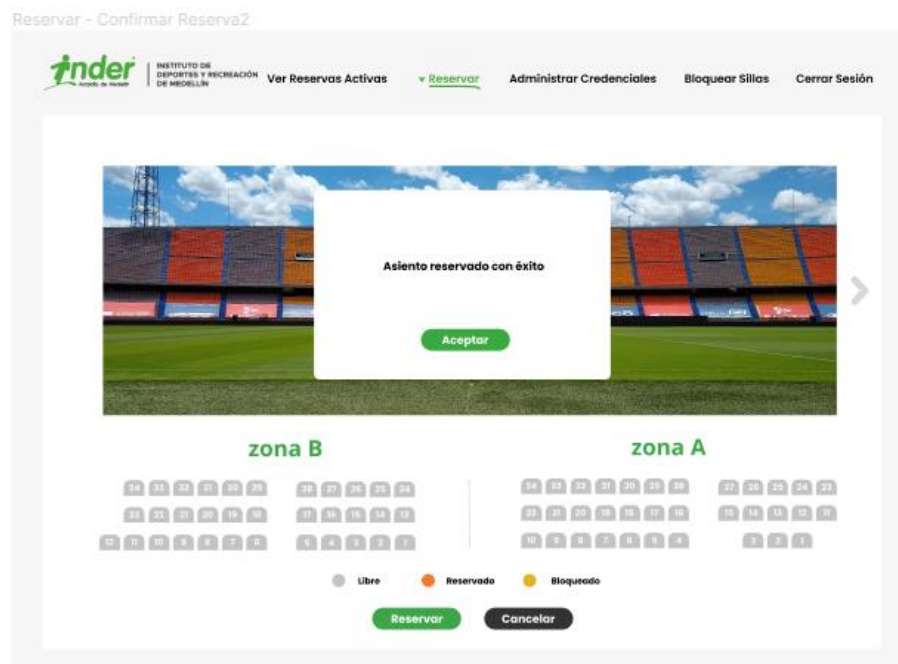


Figura 25. Mockup para ventana emergente de reserva exitosa.

El mockup para la página de modificar reserva se muestra en las Figuras 26 y 27.

Reservar - Modificar Reserva

nder INSTITUTO DE DEPORTES Y RECREACIÓN DE MEDELLÍN [Ver Reservas Activas](#) [Modificar Reserva](#) [Administrar Credenciales](#) [Bloquear Sillas](#) [Cerrar Sesión](#)

Datos de Reserva

Nombre	_____	Identificación	_____
Credencial	_____	Asiento reservado	_____

[modificar](#) [cancelar](#)

Figura 26. Mockup para formulario de modificación o cancelación de reserva.

Reservar - Modificar Reserva

nder INSTITUTO DE DEPORTES Y RECREACIÓN DE MEDELLÍN [Ver Reservas Activas](#) [Modificar Reserva](#) [Administrar Credenciales](#) [Bloquear Sillas](#) [Cerrar Sesión](#)

Datos de Reserva

Reserva cancelada con éxito

[Aceptar](#)

[modificar](#) [cancelar](#)

Figura 27. Mockup para ventana emergente de cancelación de reserva exitosa.

El mockup para la página de Administración de credenciales se muestra en la Figura 28.

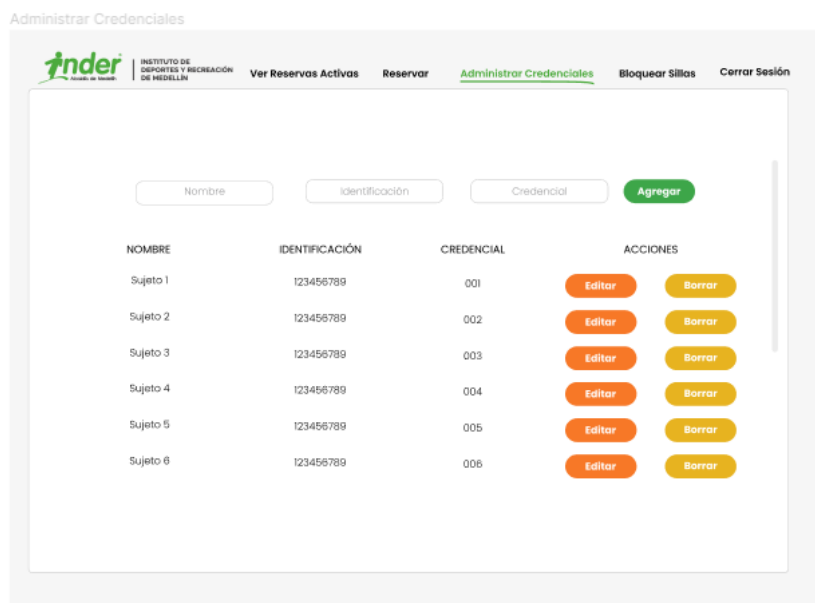


Figura 28. Mockup para Administración de Credenciales.

El mockup para la página de Bloqueo de asientos se muestra en las Figuras 29, 30 y 31

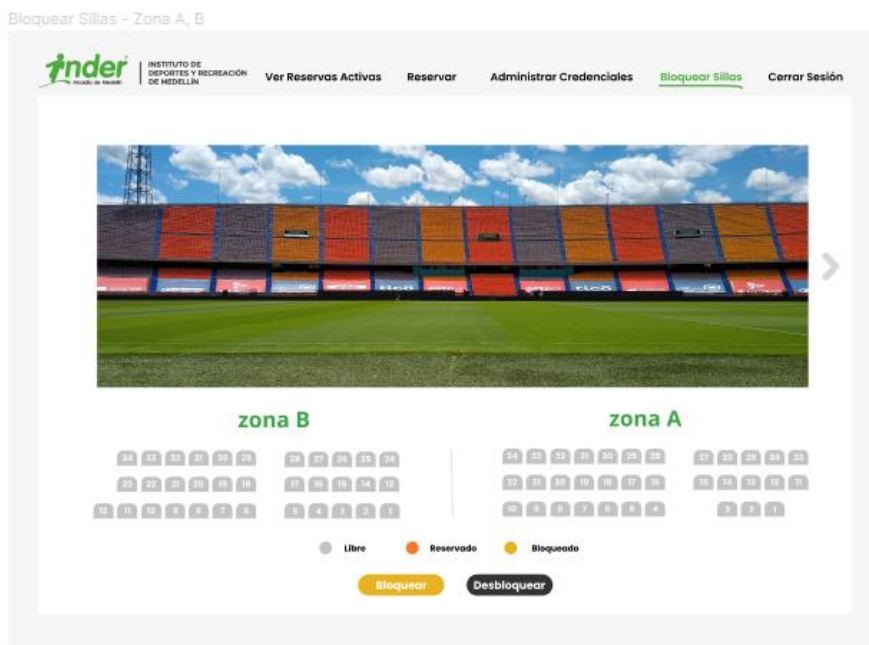


Figura 29. Mockup para Bloqueo de Asientos, zonas A y B.

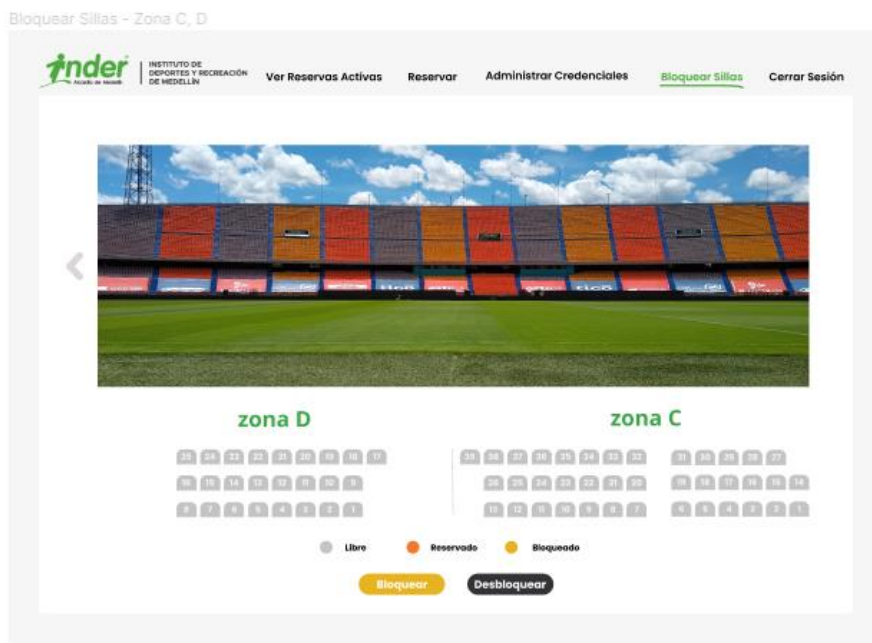


Figura 30. Mockup para Bloqueo de Asientos, zonas C y D.

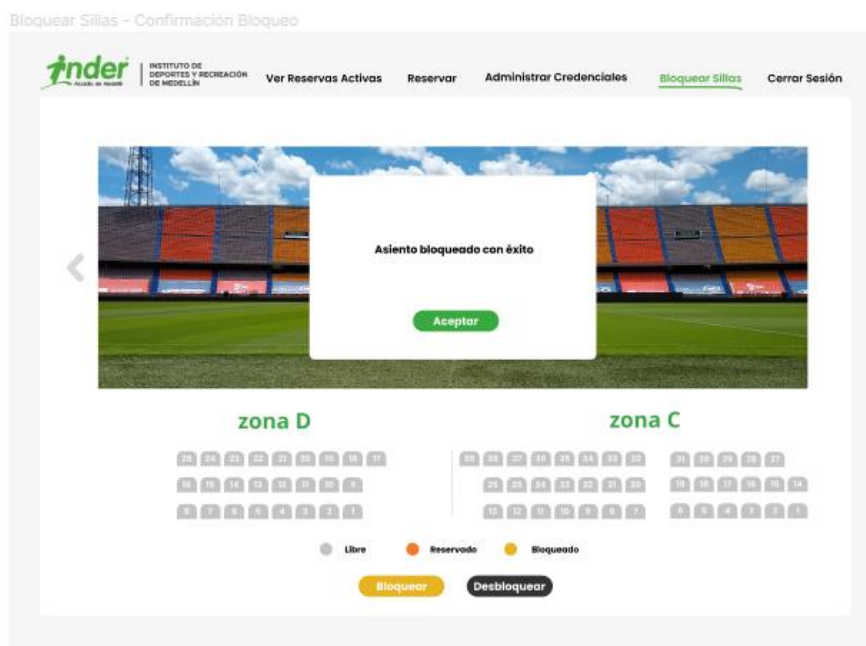


Figura 31. Mockup para ventana emergente de cancelación de bloqueo exitoso.

Por último, el mockup para la página de Reservas Activas se muestra en las Figuras 32 y 33.

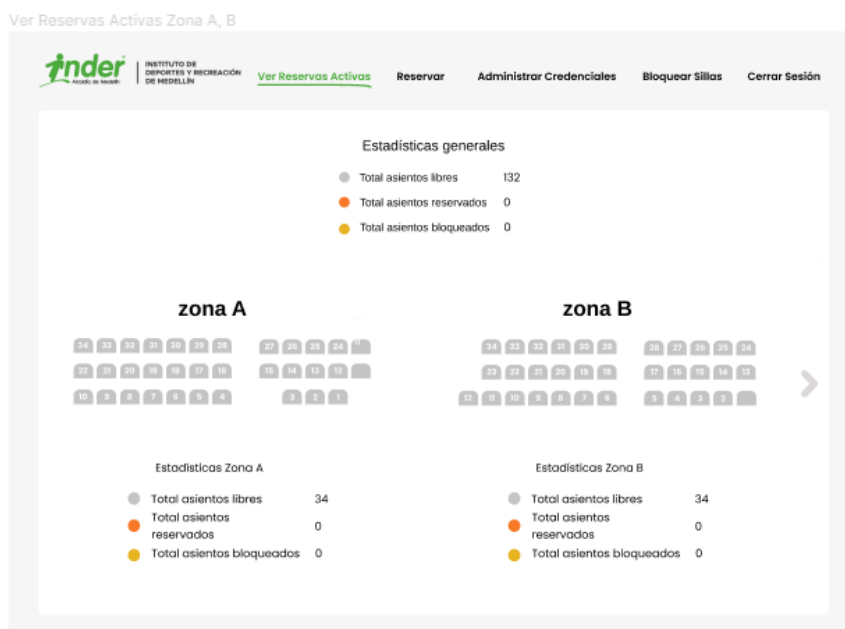


Figura 32. Mockup para Ver Reservas Activas, zona A y B.

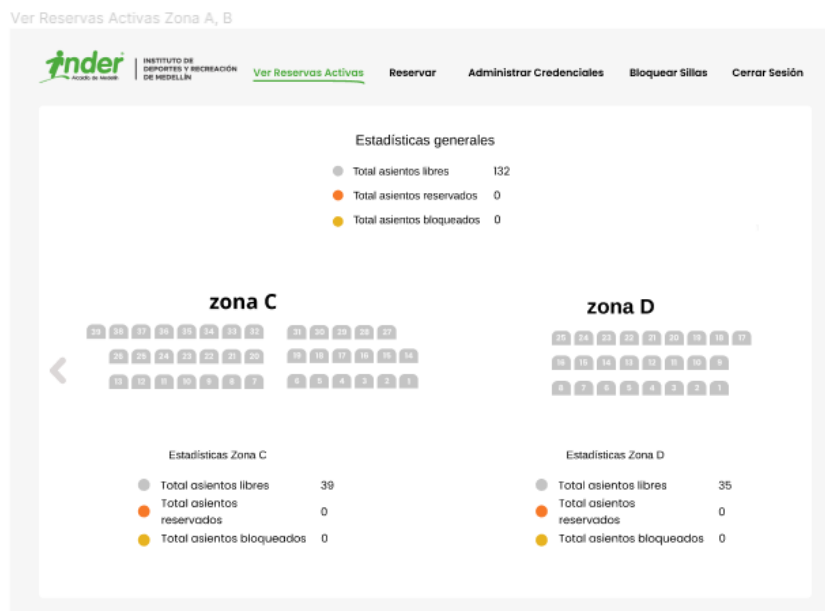


Figura 33. Mockup para Ver Reservas Activas, zona C y D.

5.2 Frontend

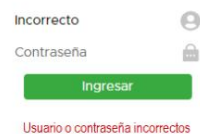
A continuación, se mostrarán los resultados para los diferentes procesos que se pueden realizar usando la aplicación.

5.2.1 Página Inicio de sesión

Para el inicio de sesión, el servidor devuelve diferentes respuestas a los posibles errores que puede cometer el usuario. La Figura 34a) muestra la respuesta del servidor cuando no se ingresa ningún campo del formulario de acceso; mientras que en la Figura 34b) se puede ver la respuesta para cuando algún dato, usuario o contraseña, se ingresa de forma errónea. Si los datos se ingresan de forma correcta, se permite el acceso a la aplicación llevando al usuario a la página principal Ver Reservas Activas.



a)




b)

Figura 34. a) Error en inicio de sesión por campos vacíos, b) Error en inicio de sesión por datos incorrectos.

5.2.2 Página Administración de credenciales

Al ingresar a la página de administración de credenciales se tiene una lista con los datos de usuarios que en el momento se encuentran en la base de datos como se muestra en la Figura 35a. Si se desea agregar un usuario nuevo se utiliza el formulario de la parte superior de la página ingresando los datos de la persona como se muestra en la Figura 35b; al dar clic al botón agregar, el servidor almacena al usuario en la base de datos y se despliega en la parte inferior de la lista como se puede observar en la Figura 35c.

 Instituto de Deportes y Recreación de Medellín


Ver Reservas Activas Reservar Administrar Credenciales Bloquear Asientos Cerrar Sesión

Agregar Persona

Nombre Cédula Credencial **Agregar**

NOMBRE	IDENTIFICACIÓN	CREDECIAL	ACCIONES
David	123456	001	Editar Borrar
Manuela	123456	002	Editar Borrar
Gerardo	123456	003	Editar Borrar
Antonio	8326765	004	Editar Borrar
Adele	876551	005	Editar Borrar
Patrick	9826551	006	Editar Borrar
León	9876544	007	Editar Borrar
Hernán	123245	008	Editar Borrar
Atenea	09213645	009	Editar Borrar
Ernesto	983625637	010	Editar Borrar

a)

 Instituto de Deportes y Recreación de Medellín

Ver Reservas Activas Reservar Administrar Credenciales Bloquear Asientos Cerrar Sesión

Agregar Persona

Jesús 8123131 011 **Agregar**

NOMBRE	IDENTIFICACIÓN	CREDECIAL	ACCIONES
David	123456	001	Editar Borrar
Manuela	123456	002	Editar Borrar
Gerardo	123456	003	Editar Borrar
Antonio	8326765	004	Editar Borrar
Adele	876551	005	Editar Borrar
Patrick	9826551	006	Editar Borrar
León	9876544	007	Editar Borrar
Hernán	123245	008	Editar Borrar
Atenea	09213645	009	Editar Borrar
Ernesto	983625637	010	Editar Borrar

b)

Agregar Persona


Nombre Cédula Credencial **Agregar**

NOMBRE	IDENTIFICACIÓN	CREDENCIAL	ACCIONES
David	123456	001	Editar Borrar
Manuela	123456	002	Editar Borrar
Gerardo	123456	003	Editar Borrar
Antonio	8326765	004	Editar Borrar
Adele	876551	005	Editar Borrar
Patrick	9826551	006	Editar Borrar
León	9876544	007	Editar Borrar
Hernán	123245	008	Editar Borrar
Atenea	09213645	009	Editar Borrar
Ernesto	983625637	010	Editar Borrar
Jesús	8123131	011	Editar Borrar

c)

Figura 35. a) Lista de credenciales actuales, b) Datos de nuevo usuario antes de ser agregado, c) Usuario nuevo agregado.

Ahora, para realizar la edición de alguna credencial, se da clic al botón de editar ubicado al frente de la persona. Para la demostración se elige el primer usuario de la tabla; al presionar el botón de editar los campos de la tabla se vuelven editables y los botones cambian a guardar y cancelar como lo muestra la Figura 36a. Una vez se hace el cambio que se requiera se presiona el botón de guardar, volviendo así al formato que se tenía anteriormente pero ahora con los datos nuevos guardados por el servidor en la base de datos como se muestra en la Figura 36b.



 Instituto de Deportes y Recreación de Medellín

[Ver Reservas Activas](#)
[Reservar](#)
[Administrar Credenciales](#)
[Bloquear Asientos](#)
[Cerrar Sesión](#)

Agregar Persona

NOMBRE	IDENTIFICACIÓN	CREDECIAL	ACCIONES
David	123456	001	<input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>
Manuela	123456	002	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Gerardo	123456	003	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Antonio	8326765	004	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Adele	876551	005	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Patrick	9826551	006	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
León	9876544	007	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Hernán	123245	008	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Atenea	09213645	009	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Ernesto	983625637	010	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Jesús	8123131	011	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>

a)


 Instituto de Deportes y Recreación de Medellín

[Ver Reservas Activas](#)
[Reservar](#)
[Administrar Credenciales](#)
[Bloquear Asientos](#)
[Cerrar Sesión](#)

Agregar Persona

NOMBRE	IDENTIFICACIÓN	CREDECIAL	ACCIONES
David Henao	123456	001	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Manuela	123456	002	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Gerardo	123456	003	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Antonio	8326765	004	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Adele	876551	005	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Patrick	9826551	006	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
León	9876544	007	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Hernán	123245	008	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Atenea	09213645	009	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Ernesto	983625637	010	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Jesús	8123131	011	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>

b)

Figura 36. a) Proceso de edición de una credencial, b) Edición guardada por el servidor.

Para borrar una credencial basta con dar clic al botón de borrar ubicado al frente de la credencial y una vez presionado el servidor se encarga de eliminar los datos del usuario de la base de datos. La Figura 37 muestra la tabla sin el primer usuario el cual fue borrado.

The screenshot shows the 'Agregar Persona' interface. At the top, there is a navigation bar with the iNder logo and the text 'Instituto de Deportes y Recreación de Medellín'. The navigation menu includes 'Ver Reservas Activas', 'Reservar', 'Administrar Credenciales', 'Bloquear Asientos', and 'Cerrar Sesión'. Below the navigation bar is a form titled 'Agregar Persona' with three input fields: 'Nombre', 'Cédula', and 'Credencial', and a green 'Agregar' button. Below the form is a table with the following data:

NOMBRE	IDENTIFICACIÓN	CREDECIAL	ACCIONES
Manuela	123456	002	Editar Borrar
Gerardo	123456	003	Editar Borrar
Antonio	8326765	004	Editar Borrar
Adele	876551	005	Editar Borrar
Patrick	9826551	006	Editar Borrar
León	9876544	007	Editar Borrar
Hernán	123245	008	Editar Borrar
Atenea	09213645	009	Editar Borrar
Ernesto	983625637	010	Editar Borrar
Jesús	8123131	011	Editar Borrar

Figura 37. Lista de usuarios con la primera credencial eliminada.

5.2.3 Página Reservar

Para realizar una reserva primero se debe validar la credencial del usuario. La Figura 38 muestra el resultado de ingresar datos incorrectos. Si se ingresaron datos correctos, al dar clic al botón validar este se dirige a la página que contiene los mapas de cada zona de palcos para que el usuario elija el asiento que desea reservar.

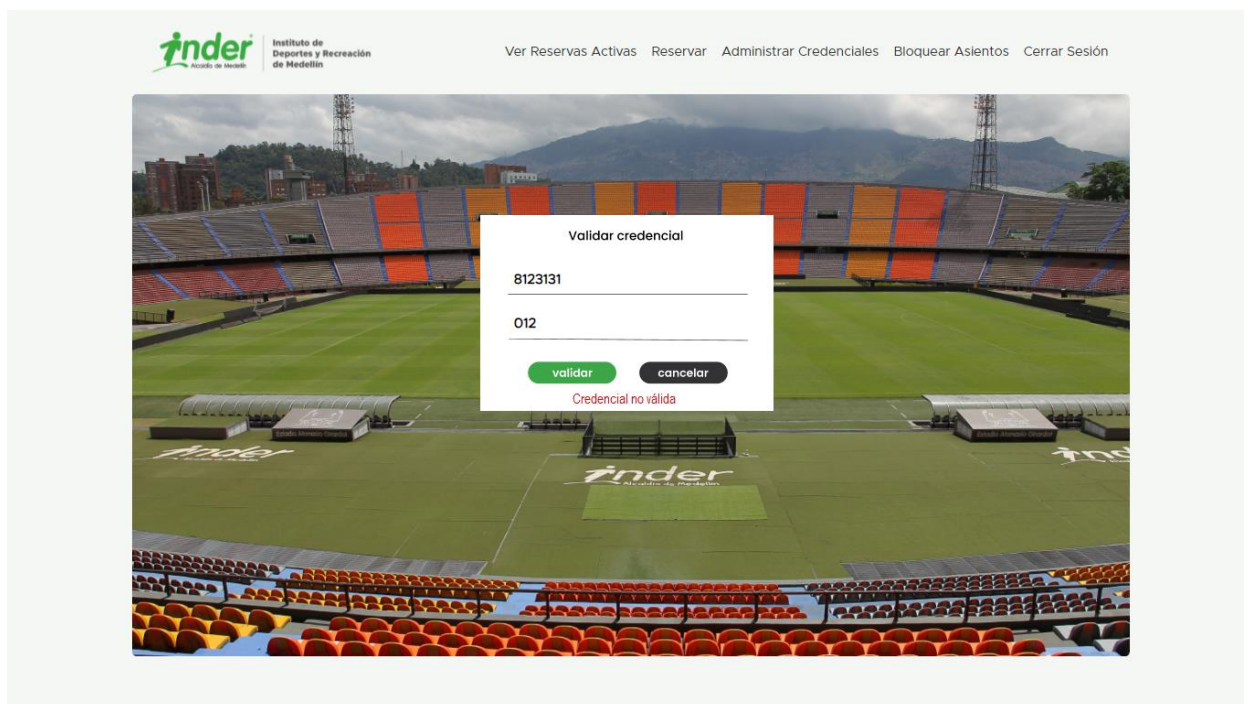




Figura 38. Validación fallida de credencial para reservas.

Con el mapa de asientos disponible, se elige un asiento aleatorio para continuar con la reserva. Para esta ocasión se eligió el asiento 32 de la zona B; como se puede observar en la Figura 39a, este se marca de color naranja indicando ha sido seleccionado. Por último, para completar el proceso, se da clic al botón de reservar el cual despliega un modal para confirmar la elección realizada como se puede ver en la Figura 39b y al dar clic en aceptar se despliega otro modal indicando que la reserva se realizó exitosamente como se observa en la Figura 39c.

 Instituto de Deportes y Recreación de Medellín

[Ver Reservas Activas](#)
[Reservar](#)
[Administrar Credenciales](#)
[Bloquear Asientos](#)
[Cerrar Sesión](#)



zona B

37	36	35	34	33	32	31	30	29	28	27	26	
25	24	23	22	21	20	19	18	17	16	15	14	
13	12	11	10	9	8	7	6	5	4	3	2	1


zona A

36	35	34	33	32	31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16	15	14	13	12	11
10	9	8	7	6	5	4	3	2	1			

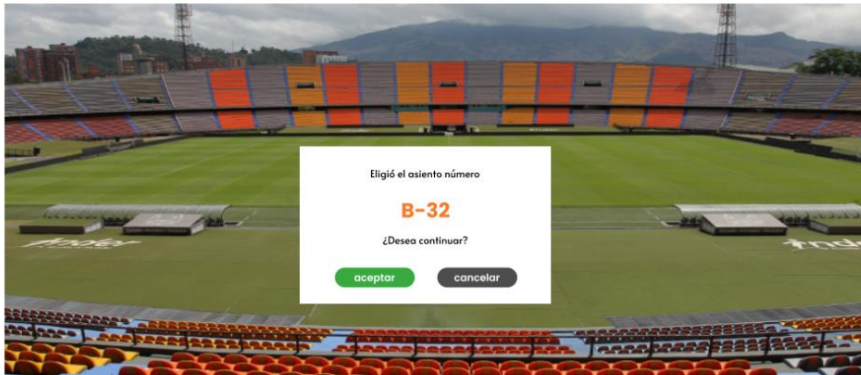
libre
 reservado
 bloqueado

reservar
volver

a)

 Instituto de Deportes y Recreación de Medellín

[Ver Reservas Activas](#)
[Reservar](#)
[Administrar Credenciales](#)
[Bloquear Asientos](#)
[Cerrar Sesión](#)



zona B

37	36	35	34	33	32	31	30	29	28	27	26	
25	24	23	22	21	20	19	18	17	16	15	14	
13	12	11	10	9	8	7	6	5	4	3	2	1

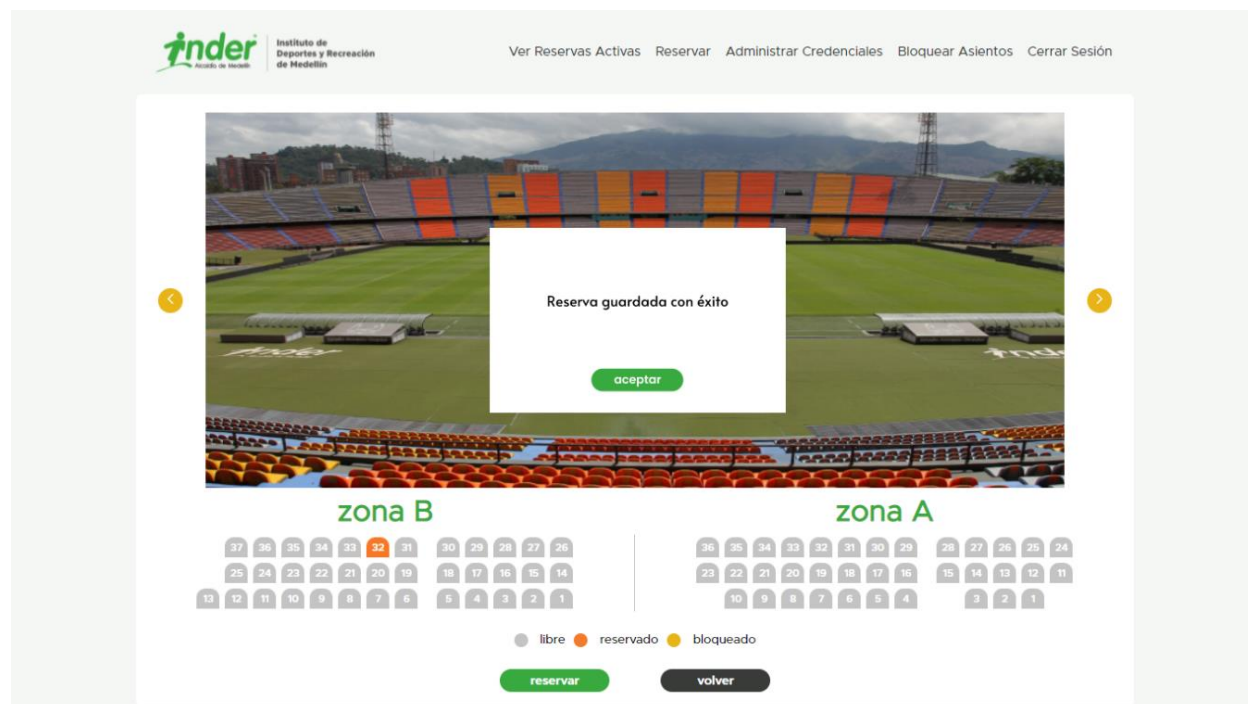
zona A

36	35	34	33	32	31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16	15	14	13	12	11
10	9	8	7	6	5	4	3	2	1			

libre
 reservado
 bloqueado

reservar
volver

b)




c)


Figura 39. a) Selección de asiento, b) Despliegue de modal de confirmación de elección de asiento, c) Confirmación de reserva exitosa.

5.2.4 Página de Bloquear Asientos

Para acceder a bloquear asientos no se necesita de validación previa; así, solo es necesario seleccionar los asientos que se deseen bloquear y a continuación dar clic al botón “bloquear” con el que aparecerá un mensaje de confirmación de bloqueo de asiento exitoso. Los asientos seleccionados para la prueba se muestran en la Figura 40a y 40b; cuando un asiento se selecciona se cambia a un color amarillo indicando su selección. Para desbloquear los asientos, solo basta con presionar de nuevo el asiento y a continuación dar clic en el botón “desbloquear”.

 Instituto de Deportes y Recreación de Medellín

[Ver Reservas Activas](#)
[Reservar](#)
[Administrar Credenciales](#)
[Bloquear Asientos](#)
[Cerrar Sesión](#)



zona B

37	36	35	34	33	32	31	30	29	28	27	26	
25	24	23	22	21	20	19	18	17	16	15	14	
13	12	11	10	9	8	7	6	5	4	3	2	1


zona A

36	35	34	33	32	31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16	15	14	13	12	11
10	9	8	7	6	5	4	3	2	1			


libre
 reservado
 bloqueado

bloquear
desbloquear

a)

 Instituto de Deportes y Recreación de Medellín

[Ver Reservas Activas](#)
[Reservar](#)
[Administrar Credenciales](#)
[Bloquear Asientos](#)
[Cerrar Sesión](#)



zona D

25	24	23	22	21	20	19	18	17
16	15	14	13	12	11	10	9	
8	7	6	5	4	3	2	1	

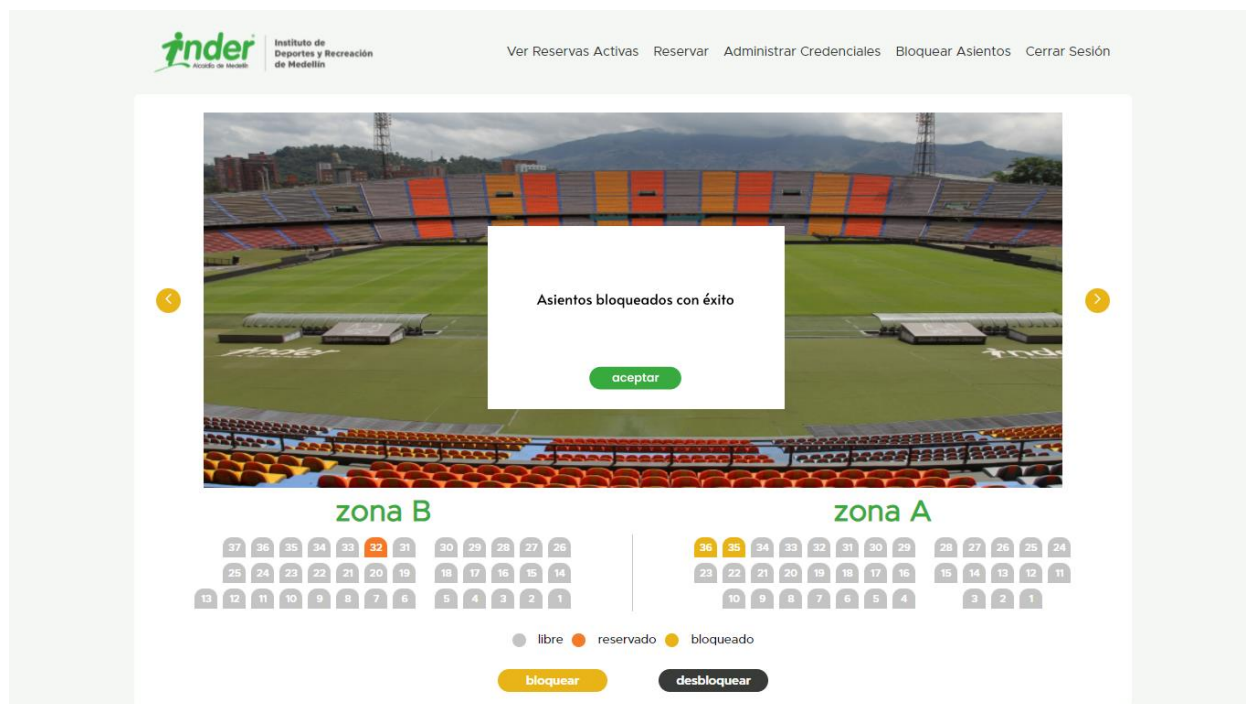
zona C

39	38	37	36	35	34	33	32	31	30	29	28	27
26	25	24	23	22	21	20	19	18	17	16	15	14
13	12	11	10	9	8	7	6	5	4	3	2	1

libre
 reservado
 bloqueado

bloquear
desbloquear

b)

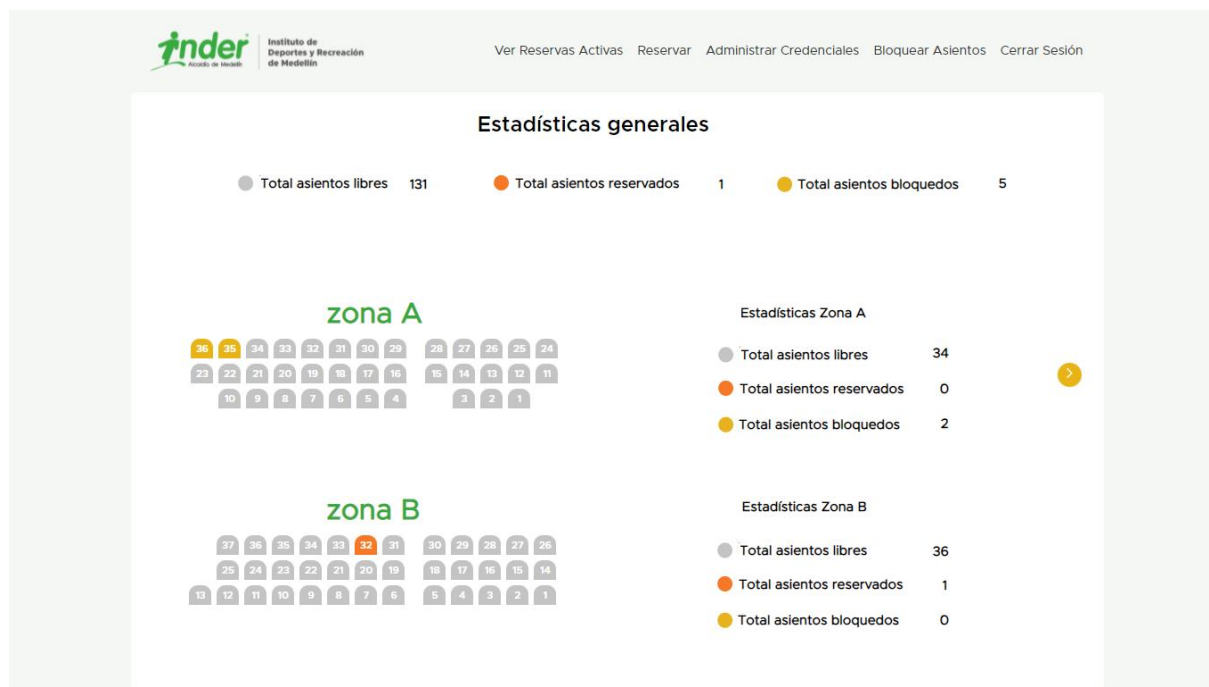


c)

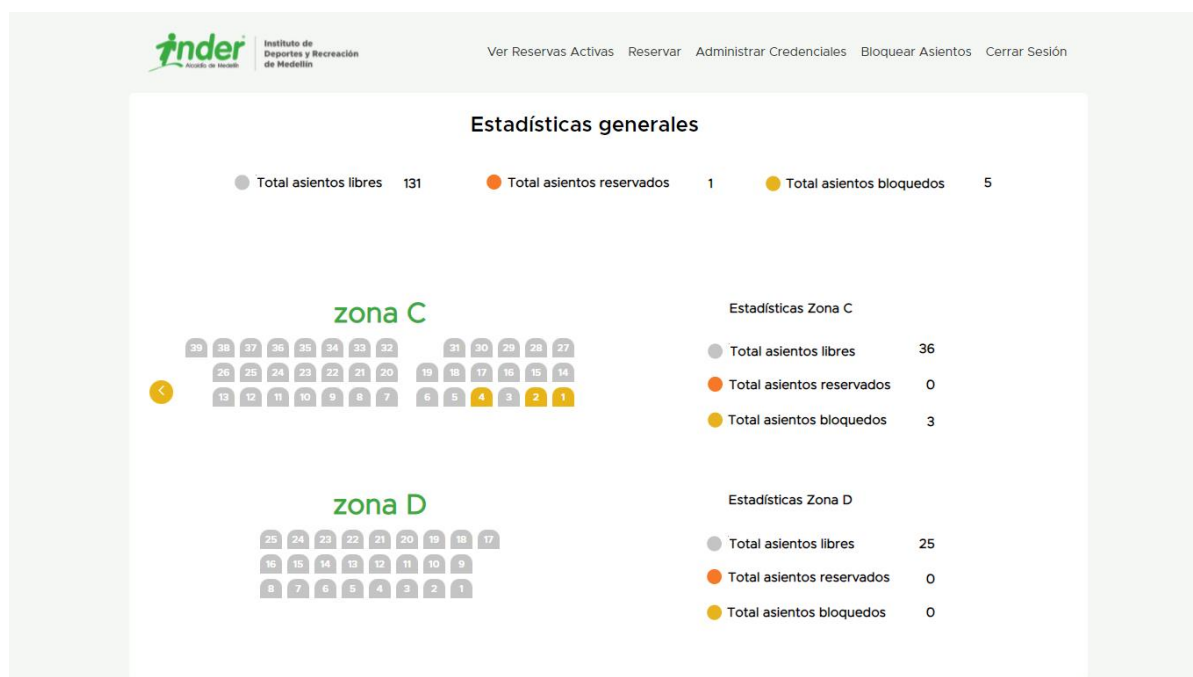
Figura 40. a) Selección de asientos a bloquear en zona A, b) Selección de asientos a bloquear en Zona B, c) Confirmación de bloqueo exitoso.

5.2.5 Página Ver Reservas Activas

La página de ver reservas activas es la primera página con la que el usuario se encuentra luego de iniciar sesión. Esta muestra los resultados de la reserva y bloqueo de asientos realizados anteriormente; al frente de cada mapa de asientos por zona se encuentran estadísticas de ocupación y en la parte superior estadísticas generales de todos los palcos.



a)

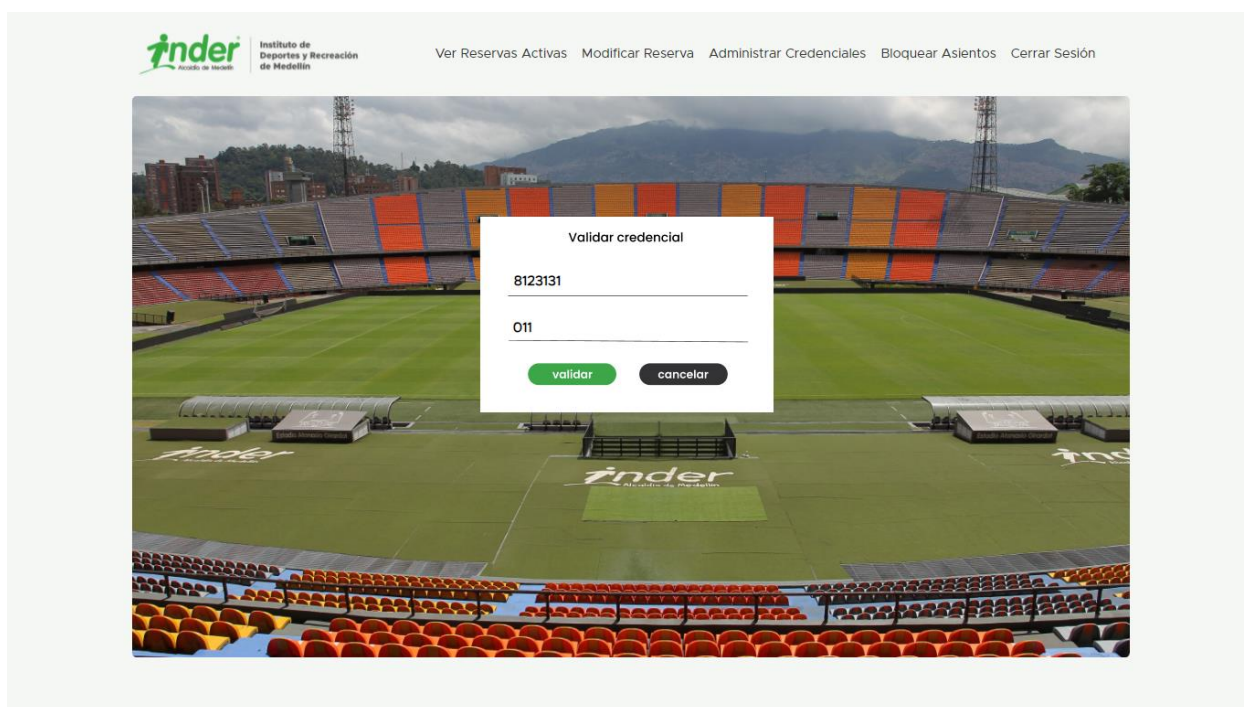


b)

Figura 41. a) Mapa de ocupación de primera zona, b) Mapa de ocupación de segunda zona.

5.2.6 Página de Modificar Reserva

Esta página permite modificar o cancelar una reserva hecha con anterioridad. Para ello se deben validar los datos del usuario como se hace en la Figura 42a y al dar clic en validar se dirigirá a otra página que contiene un formulario sencillo que despliega la información de la persona como se observa en la Figura 42b. Al dar clic en modificar se dirigirá a la página de Reservas donde el usuario podrá elegir un nuevo asiento; si se hace clic en cancelar, la reserva será eliminada de la base de datos.



a)

The screenshot displays the 'Datos de reserva' (Reservation Data) page. At the top left, the logo for 'inder' (Instituto de Deportes y Recreación de Medellín) is visible. The top navigation bar includes links for 'Ver Reservas Activas', 'Modificar Reserva', 'Administrar Credenciales', 'Bloquear Asientos', and 'Cerrar Sesión'. The main content area is titled 'Datos de reserva' and contains the following information:

Nombre	Jesús
Credencial	011
Identificación	8123131
Asiento reservado	B - 32

At the bottom of the form, there are two buttons: a green 'modificar' button and a dark grey 'volver' button.

b)

Figura 42. a) Validación de credencial para modificación de reserva, b) Datos de reserva de usuario con credencial validada.

6. Conclusiones

Se desarrolló el frontend de un aplicativo web para el INDER Medellín, el cual le permitirá a la empresa integrar su programa de reserva de palcos con una interfaz gráfica sencilla y fácil de usar para el usuario. La comunicación por medio de APIs permite a la aplicación tener información en tiempo real sin tiempos de carga demorados, permitiendo dar una respuesta rápida al usuario. Previo a la entrega de la aplicación se presentó un mockup a la empresa la cual permitió que los directivos del área de tecnología pudieran realizar comentarios sobre estilos y funcionalidades de la aplicación, así como mencionar aspectos relevantes.

El desarrollo frontend realizado en este trabajo, permitió al INDER Medellín mejorar el proceso de reserva de palcos al acortar el tiempo que tomaba realizar una reserva o modificarla, debido a la disminución de pasos en el proceso con respecto a la aplicación que se utilizaba anteriormente. También, al brindar a los encargados del manejo de ella una mejor experiencia, ya que se agregaron estadísticas de reserva generales y por cada palco y se mejoró la forma de modificar las credenciales de cada persona con acceso a los palcos de la alcaldía de Medellín.

La integración entre frontend y backend usando API REST facilitó que los desarrollos de las diferentes partes del proyecto se pudieran dar de manera independiente, permitiendo así que la aplicación final sea más escalable y flexible. Además, como la API REST se adapta en todo momento a cualquier lenguaje de programación, brindó la posibilidad de escoger libremente los frameworks para la realización del proyecto que más se adaptaran a las necesidades de la empresa.

Como un trabajo a futuro es necesario seguir aumentando la funcionalidad de la aplicación; para ello podrían crearse 2 perfiles de usuario diferentes, el administrador o súper - usuario el cual podrá tener acceso a todas las funciones de la aplicación, incluyendo reservas y

bloqueo de asientos y administrador de credenciales; y el usuario correspondiente a cada persona con credencial, que deberá registrarse previamente en la plataforma y que podrá realizar el proceso de reserva desde cualquier dispositivo con acceso a internet sin tener que desplazarse hasta las instalaciones en el estadio. También, se podría crear un sistema de ingreso mediante el uso de QR el cual se enviaría a cada persona al momento de reservar un asiento donde muestre información como nombre de la persona, zona y asiento elegidos.

Referencias

- A complete guide to website design mockup. (n.d.). WebDew.
<https://www.webdew.com/blog/website-design-mockup>
- Alani, M. M. (2014). OSI model. In *Guide to OSI and TCP/IP Models* (pp. 5-17). Springer, Cham.
- Backend developer skills you must have. (2022, March 29). Interviewbit.
<https://www.interviewbit.com/blog/backend-developer-skills/>
- Berners-Lee, T., Fielding, R., & Frystyk, H. (1996). Hypertext transfer protocol--HTTP/1.0.
- Dmitry, N., & Manfred, S. S. (2014). On micro-services architecture. *International Journal of Open Information Technologies*, 2(9), 24-27
- Duckett, J. (2011). *HTML & CSS: design and build websites* (Vol. 15). Indianapolis, IN: Wiley.
- Fedosejev, A. (2015). *React.js essentials*. Packt Publishing Ltd.
- Glinz, M. (2007, October). *On non-functional requirements*. In *15th IEEE international requirements engineering conference (RE 2007)* (pp. 21-26). IEEE.
- IBM Cloud Education. (2020, August 19). *Application programming interface (API)*. IBM.
<https://www.ibm.com/cloud/learn/api>
- Kompaniets, A. (n.d.). *Functional Vs. Non-Functional requirements: Why are both important?* UpTech. From <https://www.uptech.team/blog/functional-vs-non-functional-requirements>
- Musciano, C., & Kennedy, B. (2002). *HTML & XHTML: The Definitive Guide: The Definitive Guide*. "O'Reilly Media, Inc."
- Nidhra, S., & Dondeti, J. (2012). Black box and white box testing techniques-a literature review. *International Journal of Embedded Systems and Applications (IJESA)*, 2(2), 29-5
- OpenJS Foundation. (n.d.). *Acerca de Node.js*. Node JS. <https://nodejs.org/es/about/>
- Pérez, J. E. (2019). *Introducción a JavaScript*
- What is a Front-End developer? (n.d.). Frontend Masters.
<https://frontendmasters.com/guides/front-end-handbook/2018/what-is-a-FD.html>

