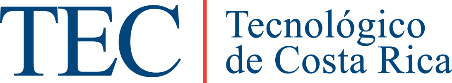
**Instituto Tecnológico de Costa Rica**

**Área Académica de Administración de Tecnología de Información**

**Programa de Licenciatura en Administración de Tecnología de Información**

**Curso: TI4601 Bases de Datos Avanzadas**



**Proyecto Programado 3**

**TRIPTEC**

**Realizado por:**

**Ronny Quesada Arias**

**Arturo Chinchilla Sánchez**

**Andrés Brais Chaves**

**Esteban Herrera Vargas**

**Profesora:**

**Ing. María José Artavia Jiménez**

**Fecha: Cartago, 21 de noviembre, 2018**

**Aspectos generales**

La mayoría de las personas tiene como meta viajar y conocer nuevos lugares turísticos, sin embargo, una vez que se tiene los recursos económicos y disponibilidad de tiempo se debe buscar información tal como hospedaje, tickets, reservaciones sobre los posibles lugares a viajar y en muchas ocasiones estos lugares no tienen página web, ni referencias de otros turistas que hayan visitado dicho lugar.

El objetivo de este proyecto es crear un sistema de reservación de lugares para la empresa TEC startup dicho sistema se denomina TRIPTEC, este sistema permitirá a la agencia gestionar la información de todos los sitios turísticos a los que puede llevar a un cliente, además permite gestionar la información de los clientes y los tours seleccionados por el cliente. Por lo que la primera vez que un usuario entra al sistema debe registrar algunos datos personales, luego de registrado, el cliente puede ingresar al sistema con un usuario y una contraseña. Los clientes pueden realizar reservaciones para cualquier número de personas y para cualquiera de los sitios turísticos registrados por los empleados, así como consultar en el sistema información sobre sitios de interés (restaurantes, cines, doctores, museos y bares) cercanos a una locación específica del sitio turístico. Es así como el cliente puede ver su historial de reservaciones y también agregar lugares a una lista de deseos.

Todo esto mediante una página web en el idioma inglés pues es un sistema global y sobre un mapa de Google, esto con el objetivo de que sea interactivo con el usuario.

**Herramientas utilizadas**

El sistema consta de dos bases de datos NoSQL y una página web, por lo tanto con el fin de crear este sistema se necesita un sistema administrador de base de datos, un servidor web y la de interfaz al usuario en las cuales se utilizaron varias herramientas para crear cada parte.

* **Administrador de base de datos**
* MongoDB: Es una base de datos NoSQL orientada a documentos de código abierto que proporciona alto rendimiento, alta disponibilidad y
* escalado automático. Un registro en MongoDB es un documento, que es una estructura de datos compuesta por pares de campos y valores. Los documentos de MongoDB son similares a los objetos JSON. [1]
* Neo4j: Es una base de datos orientada a grafos, de código abierto, NoSQL, que proporciona un backend transaccional compatible con ACID para sus aplicaciones [2].
* **Angular:** Es una framework que facilita la creación de aplicaciones con la web. Angular combina plantillas declarativas, inyección de dependencia, herramientas de extremo a extremo y mejores prácticas integradas para resolver los desafíos de desarrollo. Angular permite a los desarrolladores crear aplicaciones que viven en la web, el móvil o el escritorio [3].
* **Bootstrap:** Biblioteca de diseños y aplicaciones web, se utiliza con el fin de crear páginas web agradables a la vista y fácil de crear por medio de los componentes, además de hacer aplicaciones web responsive lo cual lo hace versátil para distintos tamaños de pantalla en los dispositivos móviles [4].
* **LoopBack**: Es un framework Node.js de código abierto muy extensible que le permite crear API REST dinámicas de extremo a extremo con poca o ninguna codificación. Además acceda a los datos de las principales bases de datos relacionales, tales como MongoDB [5].

**Estado del proyecto**

En la tabla 1 se muestran: las funcionalidades para el sistema, los requerimientos de cada funcionalidad obtenido mediante una lectura minuciosa del documento de especificación y el estado de cada requerimiento.

**Tabla 1**. Estado de los requerimientos del proyecto

|  |  |  |
| --- | --- | --- |
| **Funcionalidad** | **Requerimiento** | **Estado** |
| Gestionar sitios turísticos | Crud de sitios turísticos | Completado |
| Obtener del API de google la información del lugar |  |
| Consultar todos los sitios turísticos registrados en el sistema. | Completado |
| Registro de clientes | Clientes se registran la primera vez que ingresan al sistema | Completado |
| Acceso de clientes al sistema (Login) con un usuario y una contraseña | Completado |
| Realizar reservación | Realizar reservaciones a través del sitio web | Completado |
| Reservar servicios extras | Completado |
| Ver información de los sitios y guardarlos en una “lista de deseos” | Completado |
| Consultar en el sistema información sobre sitios de interés | Parcial |
| Agregar sitios como “posibles lugares a visitar” | Parcial |
| Ver historial de reservaciones | Un cliente ya registrado tendrá la opción de ver su historial de reservaciones | Completado |
| Migración de datos | Trasladar la información de los sitios, las reservaciones y los clientes a una base de datos de grafos. | Parcial |
| Consultas | Buscar un cliente en particular y mostrar todo su historial de reservaciones. | Completado |
| Ver todos los sitios en los que los clientes han hecho reservaciones. | Completado |
| Ver los 5 sitios para los que se han registrado más reservaciones. | Parcial |
| Dado un cliente en particular mostrar todos los demás clientes que hayan reservado al menos en un sitio en común con ese cliente. | Incompleto |
| Aplicar un algoritmo para obtener la ruta óptima para que el cliente pueda visitar los lugares cercanos al centro turístico, esta información se le debe enviar por medio de correo electrónico al usuario, en forma de recomendación. | Incompleto |

**Distribución del trabajo**

En la tabla 2 y 3 se muestran los informes enviados donde se especifican las tareas realizadas por cada estudiante en la semana del 02/11/2018 al 04/11/2018 y en las semanas Del 05/11/2018 al 16/11/2018 respectivamente. En la tabla 4 se muestran las tareas realizadas Del 16/11/2018 al 21/11/2018.

**Tabla 2**. Tareas realizadas por estudiante del 02/11/2018 al 04/11/2018

|  |  |
| --- | --- |
| **Semana** | Del **02/11/2018** al **04/11/2018** |
| **Actividades desarrolladas durante la semana** | |
| Ronny Quesada | * Creación del index.js del api server de NodeJs-Express (conexión con la base de datos de Mongo y configuración de las dependencias y routes). * Gestionar sitios turísticos: registrar la información de diferentes sitios turísticos que ofrece la compañía (latitud y longitud o la dirección exacta).   Estas funcionalidades se muestran en figura 1 y 2.    **Figura 1**. Obtener la dirección utilizando autocomplete de Google    **Figura 2**. Obtener la latitud y longitud a partir de una dirección |
| Esteban | * Se crean las primeras vistas de registro de datos. * Se utiliza angular 6 para darle funcionalidad a las páginas web en cuanto lo que es los botones y dejar las funciones listas para programar. * Se proponen las estructuras de las colecciones de la base de datos que se van a utilizar en mongodb. |
| Arturo | * Investigación sobre la herramienta LoopBack para del api de node * Se utiliza el “getting started” de la página oficial <https://loopback.io/getting-started/> como ejemplo. * Para la realización de este ejemplo ya se tenía instalado Node JS y NPM del proyecto anterior, así que se procedió a instalar el LoopBack y seguir las instrucciones presentadas en la página mencionada anteriormente. |
| Brais | * Investigación de la base de datos Neo4j * Se hacen los tutoriales usando el cliente web de la base de datos Neo4j. Esta base de datos está en la nube pero se va a trabajar con la base de datos local para el proyecto.      * Se hace planeamiento de la información que se va guardar en esta base. |
| **Actividades planeadas para la próxima semana** | |
| Ronny Quesada | * Obtener del API de google la información tipo de lugar (restaurante, cine, museo, etc), una imagen del sitio, teléfono internacional, el rating, del horario atención al público y el website. Guardar esta información en la base de datos de Mongo. * Terminar con la vista Empleado. |
| Esteban | * Se comenzará a programar la funcionalidad de las páginas web con los datos creados. * Probar todas las funciones del API, generada por loopback con postman para verificar su funcionamiento y ver si es lo que se espera. |
| Arturo | * Definición de las entidades y modelos que se van a utilizar en el proyecto (En conjunto con los compañeros). * Crear el API de la aplicación utilizando LoopBack (modelos de cada entidad dentro del proyecto). * Probar junto con Esteban los servicios/funciones proporcionadas por el API para su correcta lectura y escritura. |
| Brais | * Generación de la base de datos orientada a grafos con la estructura necesaria para el proyecto. * Pruebas de importación de información desde un lenguaje que externo al manejador de base de datos. * Comprobar el formato de respuesta de base de datos orientada a grafos y garantizar la lectura de estos en el lenguaje que recibe la información. * Analizar el uso de una base de datos Neo4j almacenada en la nube para facilidad de uso remoto para los integrantes del grupo. |

**Tabla 3**. Tareas realizadas por estudiante del 05/11/2018 al 16/11/2018

|  |  |
| --- | --- |
| **Semana** | Del **05/11/2018** al **16/11/2018** |
| **Actividades desarrolladas durante la semana** | |
| Ronny Quesada | * Registro de clientes: En la figura 1 se muestra el formulario que el cliente debe completar para registrarse en el sistema, tal como se establece en la especificación del proyecto se almacena la cédula, el nombre, el teléfono, correo electrónico, la fecha de nacimiento, nombre de usuario y clave de acceso.     **Figura 1.** Formulario web para registrar un cliente  La figura 2 muestra cómo se almacena dicha información en la base de datos de MongoDB, como se observa la contraseña tiene un encriptación.    **Figura 2.** Datos almacenados en MongoDB  La figura 3 muestra el inicio de sesión del sistema tanto para cliente como para administrador y empleado (Funcionario).    **Figura 3**. Pagina web para iniciar sesión |
| Esteban | * Debido a que estamos en semana 17 con muchos exámenes finales y proyectos no se ha logrado trabajar en las vistas de la aplicación, además se pensaba reutilizar código ya creado con anterioridad como lo es el del login y otras vistas que pueden ser muy similares para este proyecto. * Por último, ya se leyó bien el documento pero se necesitan realizar los cambios necesarios a las vistas ya realizadas en los proyectos anteriores para que este realice las funcionalidades descritas en el documento. |
| Arturo | * Debido a las implicaciones que tiene el final de semestre, la cantidad de exámenes y proyectos finales, en esta semana no se logra cumplir con los objetivos y actividades planificadas. * Se definieron preliminarmente algunos modelos de datos a utilizar en el API como:   + cliente (cedula, nombre, teléfono, fecha de nacimiento, correo electrónico y contraseña).   + empleado (cedula, nombre, teléfono, fecha de nacimiento, correo electrónico y contraseña).   + sitio (nombre, latitud, longitud, dirección, descripción)   + reservación (id cliente, id sitio, cantidad de personas, fecha de llegada, fecha de salida, necesidad en particular (opcional), servicios extras) |
| Brais | * Esta semana por la naturaleza de la semana 17 no se concretó completamente con los objetivos proyectados en el informe anterior. Se determinaron los límites de la plataforma en línea de Neo4j por lo tanto se va a utilizar la plataforma local. Se encontraron maneras, sin comprobar, de conectarse desde Node.js a la base orientada a grafos. |
| **Actividades planeadas para la próxima semana** | |
| Ronny Quesada | * Obtener del API de google la información tipo de lugar (restaurante, cine, museo, etc), una imagen del sitio, teléfono internacional, el rating, del horario atención al público y el website. Guardar esta información en la base de datos de Mongo. * Terminar el diseño y funcionalidad de la vista Empleado. |
| Esteban | * Creación de las vistas finales y junto con Arturo verificar la funcionalidad de estas mismas con los datos esperados además de unificar todo el proyecto. * Hacer pruebas relevantes en la aplicación para corroborar que no se encuentra ningún error en ella, de haberlo se tratara de corregir. |
| Arturo | * Definición final de las entidades y modelos que se van a utilizar en el proyecto (En conjunto con los compañeros). * Crear el API de la aplicación utilizando LoopBack (modelos de cada entidad dentro del proyecto). * Probar junto con Esteban los servicios/funciones proporcionadas por el API para su correcta lectura y escritura. |
| Brais | * Para la próxima semana se planea terminar con la conexión de la base de datos orientada a grafos, con el diseño de las relaciones para obtener información relevante del grafo. * Conexión y obtención de información desde la base orientada a documentos Mongo y su importación a la base de datos Neo4j. * Análisis de la información y maneras de obtención para la aplicación como tal. * Apoyo en vistas faltantes. |

**Modelo de datos implementado**

**Diagrama de la base de datos orientada a documentos**

A continuación la base de datos no relacional orientada a documentos se muestra las entidades que van a ser utilizadas y almacenadas para el funcionamiento de la a aplicación. En la siguiente imagen se muestra la el diagrama de la base.



Se muestra cada una de las entidades de la base de datos.

|  |  |
| --- | --- |
| **Nombre de la entidad** | **Descripción** |
| Reservation | Contiene la información relevante a la reservación y necesaria para la gestión |
| Client | Información del cliente para poder hacer reservaciones |
| Employee | Información del empleado necesaria para agregar sitios y ver la de los clientes |
| Site | Información de los lugares disponibles para ir a vacacionar |
| Wishlist | Lista de los lugares a los que un cliente quiere ir en el futuro |

**Modelo de la base de datos orientada a grafos**

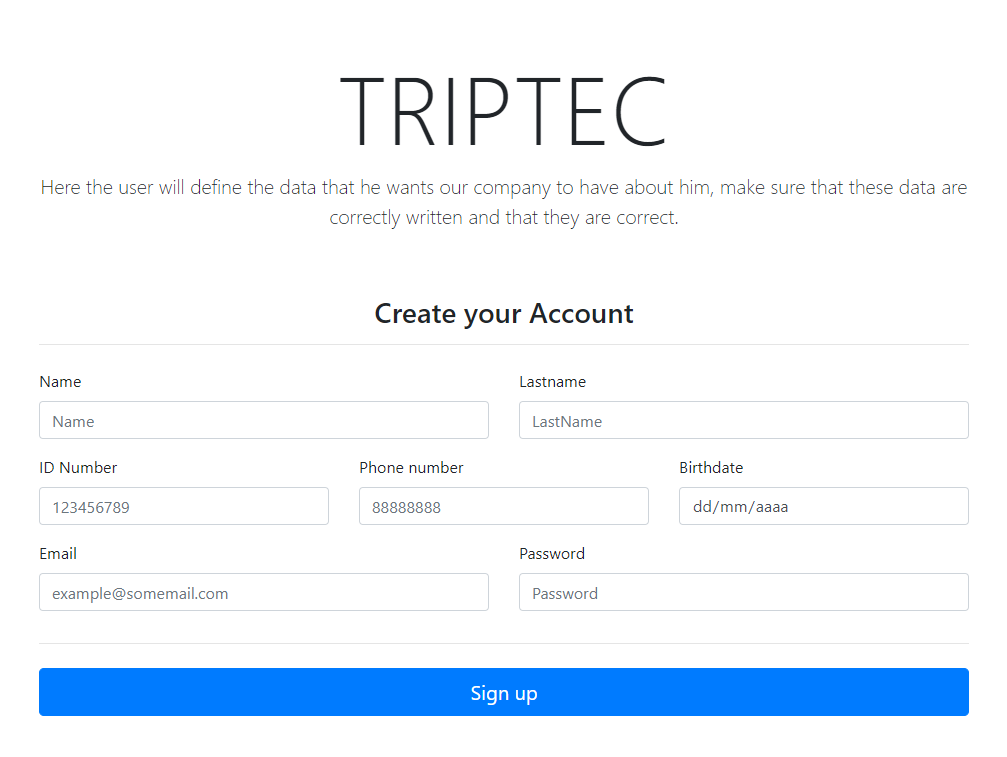
Por la naturaleza de las consultas requeridas en la base de datos orientada a grafos solo se necesitan dos nodos y una relación. En la siguiente tabla se va a describir el funcionamiento de la descripción de las entidades y las relaciones.

|  |  |
| --- | --- |
| Entidad | Descripción |
| Client | De la misma manera que en al anterior contiene la información para identificar al cliente |
| Site | De igual forma contiene la información para describir los sitios turísticos. |
| Reservation | Es una relación entre cliente y sitio, para determinar cuál cliente le ha hecho una reservación a un sitio. |

**Pruebas de funcionalidad**

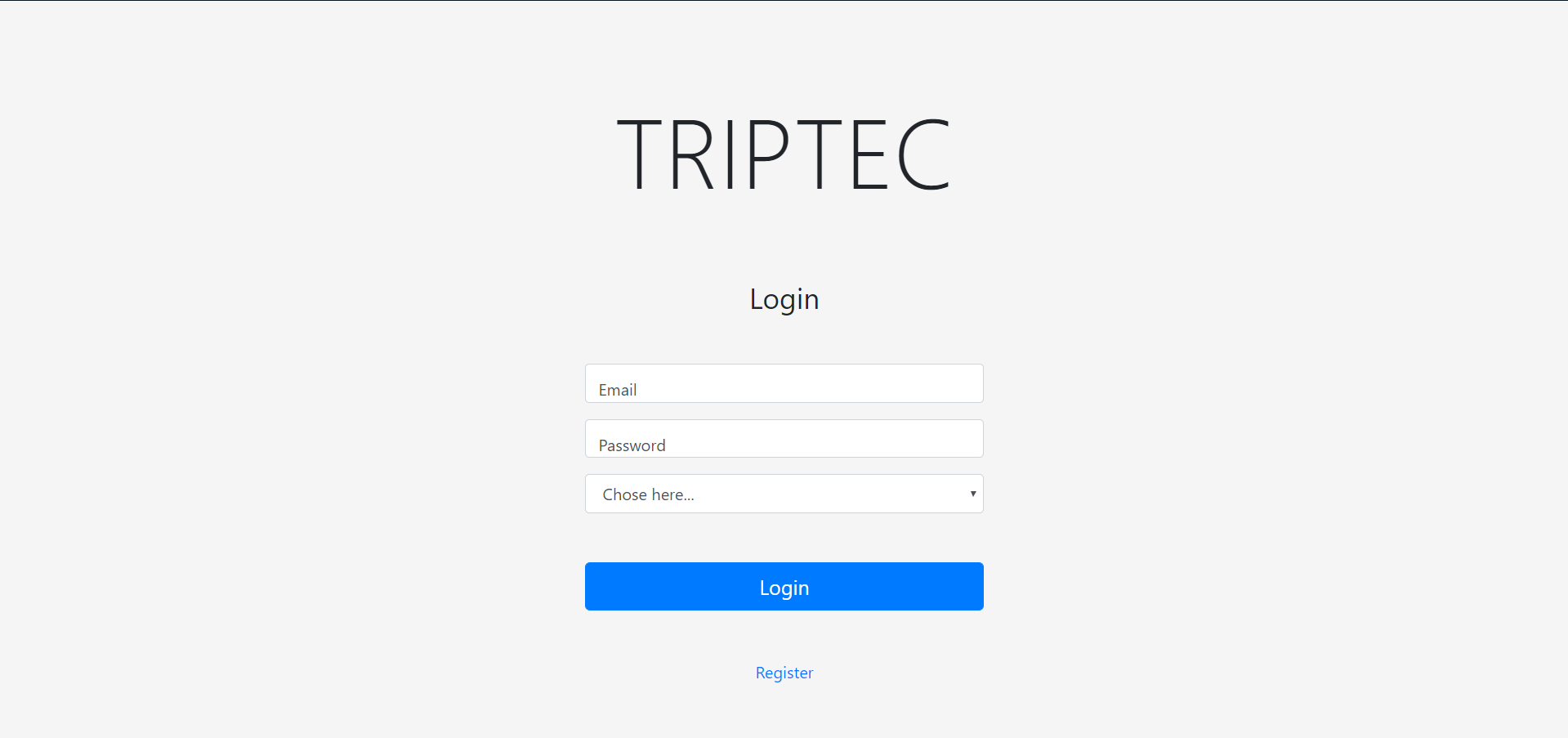
**Create Account**

Para un cliente que no se ha registrado, ésta página se encarga de recoger la información necesaria para la creación de la cuenta del cliente.

****

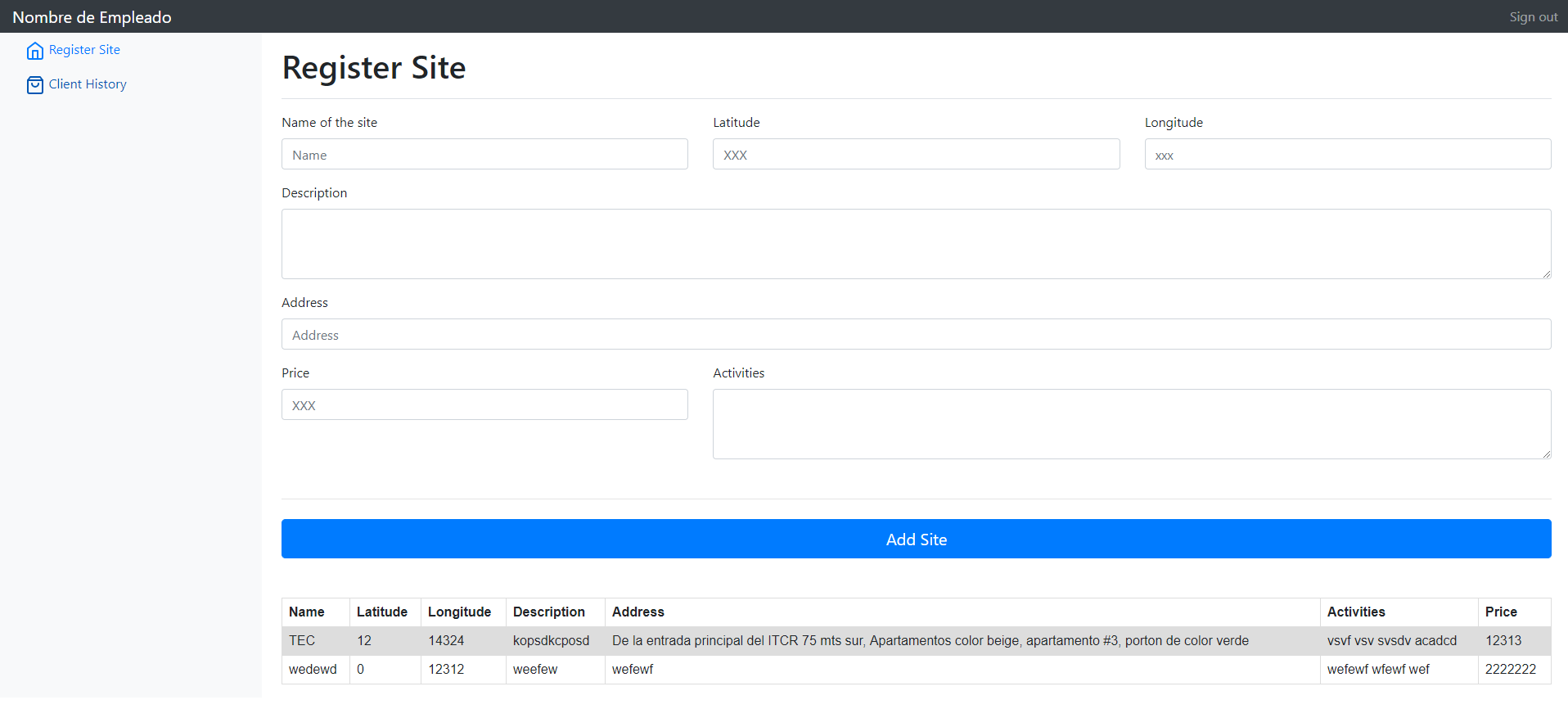
**Login**

Vista de login para empleados y clientes.

****

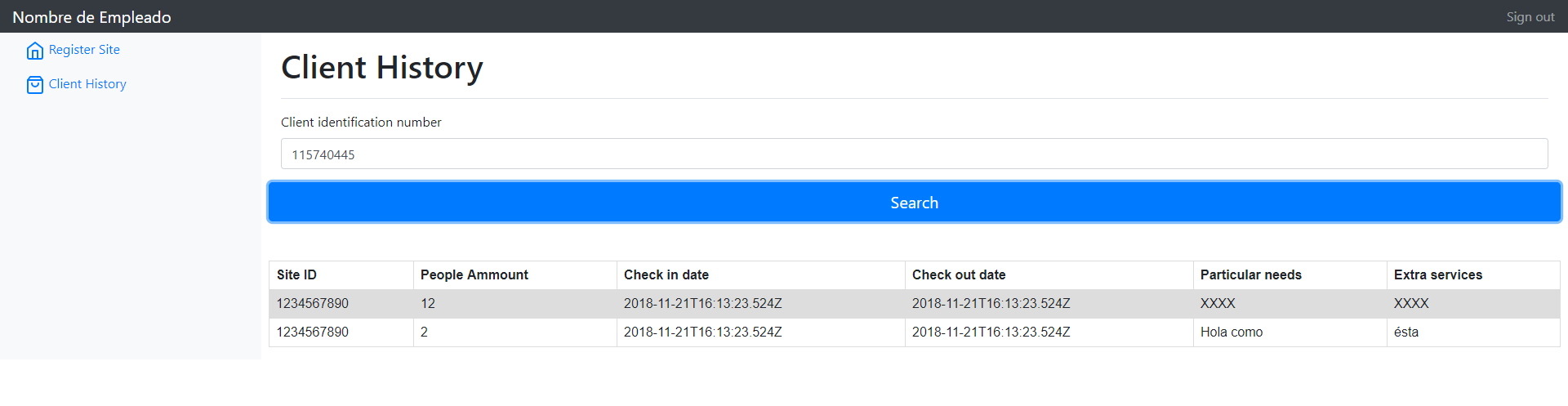
**Register Site**

Vista del empleado para agregar un sitio turístico.

****

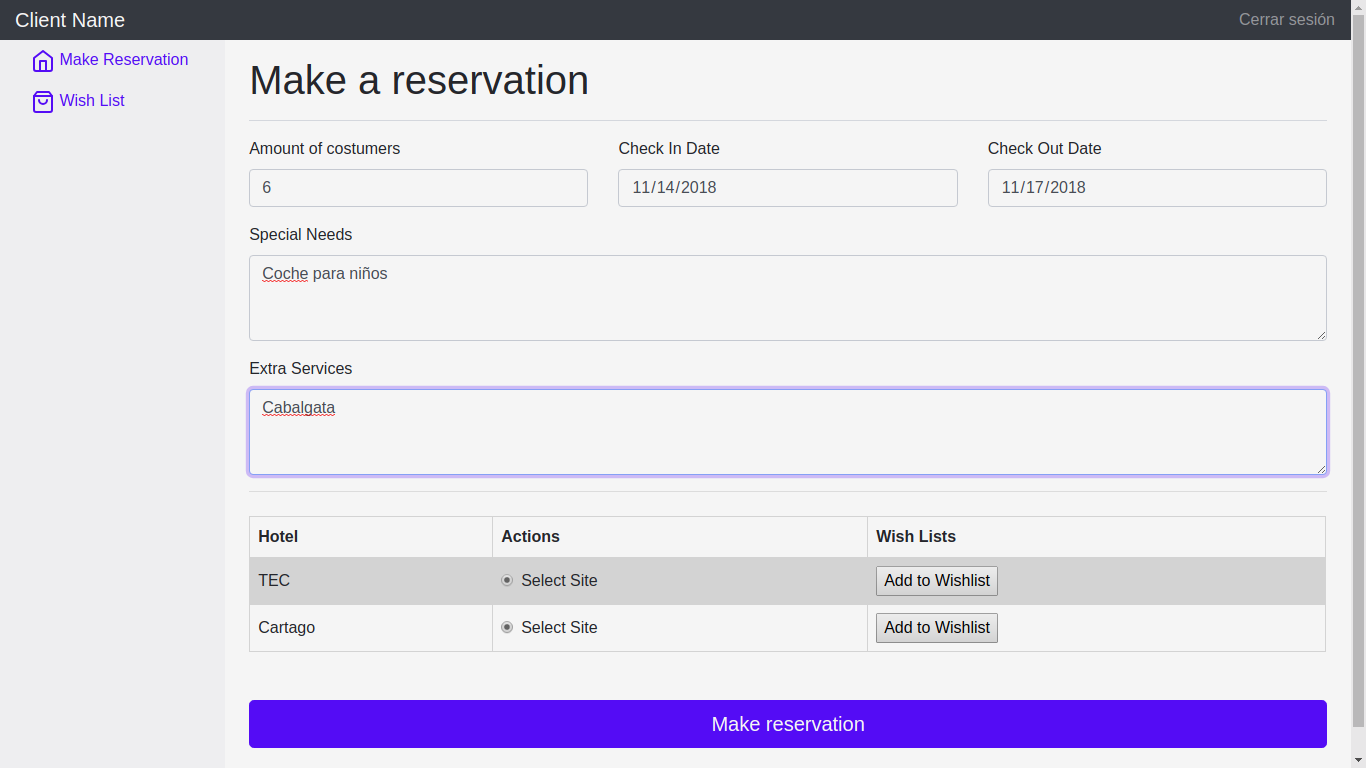
**Client History**

Muestra las reservaciones anteriores de un cliente ingresando su número de identificación.

****

**Realizar una reservación**

Realiza una reservación con la información requerida.

****

**Referencias**

[1]Docs.mongodb.com. (2018). *Introduction to MongoDB — MongoDB Manual*. [online] Disponible en: <https://docs.mongodb.com/manual/introduction>

[2] Neo4j Graph Database Platform. (2018). *What Is a Graph Database and Property Graph | Neo4j*. [online] Disponible en: <https://neo4j.com/developer/graph-database>

[3] Angular.io. (2018). *Angular Docs*. [online] Disponible en: <https://angular.io/docs>

[4] Mark Otto, a. (2018). *Bootstrap*. [online] Getbootstrap.com. Disponible en: <https://getbootstrap.com>

[5] Loopback.io. (2018). LoopBack Documentation | LoopBack Documentation. [online] Disponible en: <https://loopback.io/doc>