ServiApp

Felipe Rojas, Fabian López, Víctor Barragán

No. de Equipo Trabajo: { *4* }

I. INTRODUCCIÓN

Este documento consiste en la presentación formal del proyecto en su 1ª entrega para la materia 'Estructuras de datos'. El proyecto consiste en el desarrollo de una página web denominada "ServiApp", se presentarán a continuación los avances en interfaz gráfica, software además de distintas pruebas de datos, funcionalidades, dificultades en el desarrollo, entre otros.

II. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

Existen distintas aplicaciones que se encargan de reunir servicios y sus proveedores con el fin de facilitar este proceso a los usuarios, en su mayoría son aplicaciones de domicilios enfocadas a la gastronomía y aplicaciones de transporte.
Sin embargo, no existe una aplicación que ofrezca servicios técnicos, especializados o poco frecuentes. Además, de unificar todos estos servicios en una aplicación para comodidad del usuario.
Otro aspecto para considerar es la valoración de los servicios ofrecidos, esta valoración promueve y permite brindar la mejor experiencia a los usuarios además de fortalecer la sana competencia de los proveedores, todo esto se puede realizar gracias al avance de la tecnología y resulta poco provechoso que los servicios y productos ofrecidos tradicionalmente aun no cuenten con la virtualización necesaria para aprovechar las tendencias tecnológicas.
Con lo anterior, ServiApp tiene como objetivo ofrecer servicios técnicos (plomería, eléctricos, entre otros) resaltando una contratación justa y libre para ambos perfiles (usuarios y proveedores) en donde se estipulen los términos entre ambas partes antes de que se realice el servicio.

III. USUARIOS DEL PRODUCTO DE SOFTWARE

ServiApp cuenta con 3 tipos de roles diferentes en cuanto a sus usuarios se refiere, estos son:

- Usuario informal: Es aquel usuario que ingresa a la página web sin una verificación de su cuenta (sin inicio de sesión), este usuario puede acceder a la información de contacto suministrada por los distintos proveedores de servicios y realizar una contratación por su cuenta.
- Usuario Formal: Este usuario posee y utiliza su cuenta en la aplicación para realizar contrataciones de un servicio y revisar en su perfil contrataciones pasadas.
- Usuario Proveedor: Son aquellos usuarios que desean registrarse en los distintos servicios como proveedores, deben suministrar información pertinente para la realización de los contratos y un posible contacto con los usuarios que deseen contratar sus servicios.

IV. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

Para la lectura de los requerimientos funcionales de este software hay que tener en cuenta que la definición de "página" no necesariamente refiere a la totalidad de la página como tal, también puede referir a una sección de la misma aludiendo a la forma como se ejecuta el código en cada sección y a la forma como funcionan las conexiones entre las distintas secciones a partir de "hipertextos" o más comúnmente conocidos como "enlaces" que en términos simples son las conexiones existentes entre elementos de la página (textos, botones, imágenes, etc) que redirigen al usuario ya sea a otra sección de la misma página web (refiérase a página web como a la totalidad del software) o a otra página web distinta a "ServiApp".

Registro de usuarios (Formales y proveedores)

- *Descripción:* Permite a un usuario informal (visitante de la página web) registrar sus datos en la base de datos de la página web de modo que su información quede guardada para poder hacer uso de las distintas funcionalidades de ServiApp que requieran del continuo almacenamiento de estos.

- *Acciones iniciadoras y comportamiento esperado*:
  1. Luego del acceso por parte del usuario en la página, este deberá hacer clic al botón de "registrarse" o "registrarse como proveedor."
  2. Para cualquiera de los dos casos, la página se redireccionará a otra dónde se encontrará un formulario el cual deberá diligenciar para su registro. (Ver anexo [1]).
  3. Una vez el usuario de clic en el botón ubicado al final del formulario se desencadenará un evento de tipo "submit" el cuál enviará los datos diligenciados en el

formulario como un objeto (clase Usuario o clase Proveedor respectivamente) al código fuente de la página.

4. A partir de aquí se ejecutará un método en donde el objeto creado se enviará y almacenará en una base de datos.

Requerimientos funcionales:

- Disponibilidad del enlace al formulario para los usuarios informales que visiten la página web.
    - Esto básicamente se resume en la disponibilidad de la página que contiene el formulario y que tanto esta como el enlace que redirecciona aquí no contengan errores.
- Conexión activa entre los archivos que contienen el código fuente de la página y los usados para la conexión con la base de datos (back-end).
    - Cabe agregar que para cada entrada del formulario se especifica un tipo de dato especifico y unas longitudes específicas que en dado de no cumplirse por parte del usuario se le indicará bajo la figura de un aviso en pantalla e imposibilitará el envió del formulario al deshabilitar el evento "submit" hasta que se corrijan los datos ingresados.
    - Si esta conexión no es posible, existirá una excepción para el posible error que ejecutará un aviso en pantalla que informará al usuario sobre el estado de la página web. Esto posiblemente desencadenará un aviso que le será enviado a los desarrolladores para corregir el error lo más pronto posible.
- Disponibilidad de la base de datos para recibir las peticiones por y ejecutar los métodos para almacenar estos datos.
    - Esto se resume en el hecho de que exista la conexión estable y vigente con la base de datos y que esta esté funcionando plenamente. Para ello se comprueba constantemente que los datos se envíen y se guarden adecuadamente.

Eliminación de usuarios (Formales y proveedores)

- *Descripción:* Permite a un usuario formal o proveedor eliminar sus propios datos en la base de datos de la página de modo que su información quede eliminada para poder hacer uso de las distintas funcionalidades de ServiApp que requieran del continuo almacenamiento de estos.

- *Acciones iniciadoras y comportamiento esperado*:

Si el usuario desea eliminar sus datos de la página (es decir, eliminar su cuenta de ServiApp), encontrará está opción dentro de las opciones de su perfil y la seleccionará. Esto desencadenará un evento desde la página hasta el código en el cual se le enviará una petición a la base de datos para que ejecute un método que deshabilite la posibilidad de iniciar sesión y la posibilidad de que otro usuario pueda contratar sus servicios (en caso de ser un usuario proveedor). Cabe agregar que estos datos seguirán almacenados en la plataforma ya que seguirán siendo parte del historial e información de contratos de otros usuarios de la plataforma.

Requerimientos funcionales:

- Disponibilidad del enlace al perfil de la cuenta con sesión iniciada en el navegador desde cualquier otra página de la web.
    - Esto implica que el enlace alojado en la opción "Ver mi perfil" este funcionando correctamente.
- Conexión activa entre los archivos que contienen el código fuente de la página y los usados para la conexión con la base de datos (back-end).
    - Esto se resume en el hecho de que el evento provocado al hacer clic en la opción "eliminar cuenta" desde la página de perfil tenga una conexión vigente con el código alojado en el servidor (el computador donde se aloje el código que conecte a la base de datos).
- Disponibilidad de la base de datos para recibir las peticiones por y ejecutar los respectivos métodos respecto estos datos.
    - Esto se resume en el hecho de que exista la conexión estable y vigente con la base de datos y que esta esté funcionando plenamente. En dado caso de que el evento no pueda ser ejecutado correctamente por parte del usuario, se le informará a través de un aviso en pantalla y así mismo se le enviará un mensaje al equipo desarrollador para corregir este error lo más pronto posible.

Inicio de sesión por parte de usuarios registrados a la página web.

- *Descripción:* Permite a los usuarios (formales o proveedores) cuyos datos ya están almacenados en la base

de datos iniciar sesión en la página web para hacer uso de ciertas características que lo requieran.

- *Acciones iniciadoras y comportamiento esperado*:

El usuario visitante de la página web al hacer clic en el botón de "inicia sesión" será redirigido a otra página en donde se encuentra un formulario de tipo "login". Al completar este formulario se compararán los datos obtenidos en el formulario con los datos de alojados en la base de datos, y si estos coinciden con los datos de un objeto de tipo usuario o proveedor el usuario se encontrará identificado en la plataforma con sus respectivos datos (alojados en la base de datos) lo que le permitirá acceder a ciertas características tales como requerir un servicio (realizar contratos) o simplemente revisar su historial de servicios adquiridos y su información de perfil.

.

Requerimientos funcionales:

- Disponibilidad del formulario para los usuarios no formales:
  - Un usuario no formal es cualquier usuario no identificado al hacer uso de la página web, si el formulario es visible y disponible para él significa que documento html correspondiente a esta página funciona correctamente.
- Funcionamiento correcto del formulario y conexión entre el código fuente de la página y la base de datos:
  - Al momento de completar el formulario se tendrán en cuenta los tipos de datos proporcionados por el usuario en cada campo de entrada del formulario (estos están especificados y limitados en caracteres) de modo que el usuario no pueda enviar el formulario si esto ocurre (véase requerimientos funcionales de la función "Registro de usuarios"). Al finalizar el formulario el usuario hará clic en el botón "iniciar sesión" el cuál desencadenará un evento que funcionará como una petición para enviar un método a la base de datos el cuál comprobara si los datos ingresados coinciden con datos ya alojados en la base de datos, en caso de no ser así se le pedirá al usuario registrarse o comprobar sus datos. Y en caso de no poder completar la petición a la base de datos se le informará al usuario acerca del error y se les enviará un mensaje a los desarrolladores para corregir el error lo más pronto posible.
- Disponibilidad del enlace al perfil de la cuenta con sesión iniciada en el navegador desde cualquier otra página de la web:
  - El botón iniciar sesión deberá estar disponible y ser funcional desde cualquier ubicación dentro de las paginas contenidas en la web de ServiApp, esto quiere decir que no importa si el usuario se encuentre en la página de inicio o en la página correspondiente al listado de proveedores el botón deberá de funcionar correctamente.

Visualización del listado de proveedores y los servicios disponibles para contratar.

- *Descripción:* Permite a cualquier tipo de usuario (formales, proveedores o informales) acceder al listado de proveedores y servicios disponibles para ser contratados. Este listado dependerá del correcto funcionamiento de la base de datos y así mismo se tendrá en cuenta un sistema de calificación (disponible para los usuarios ya registrados únicamente) que servirá para organizar este listado de modo que se promocionen los mejores y más puntuados proveedores y servicios.

- *Acciones iniciadoras y comportamiento esperado*:

Dentro de la página de inicio de la web existirá un botón disponible para cualquier usuario que entre a la página principal en donde al hacer clic redirigirá al usuario a un listado completo y filtrable (por tipo de servicios o calificación) de los proveedores y servicios disponibles para su contratación. Esta información será proveniente directamente de la base de datos. Al momento de acceder a dicha página se ejecutará un script que además de hacer una solicitud para acceder a estos datos almacenados en la base de datos también se encargará de ordenarlos y mostrarlos en pantalla (haciendo uso de herramientas de tipo front-end, es decir, código Javascript, Html y Css ejecutado directamente en el navegador) esta organización se dará en base principalmente a un algoritmo de ordenamiento que priorice la calificación de cada servicio y proveedor y que así mismo sea capaz de filtrar las distintas opciones dependiendo del tipo de servicio ofrecido (filtros elegidos manualmente por el usuario) y/o horario de disponibilidad del proveedor de dicho servicio.

Requerimientos funcionales:

- Disponibilidad del enlace en la página de inicio.
- Conexión estable y vigente desde el código fuente de la página en cuestión con el código ejecutado en el servidor y la base de datos:
  - Dicha conexión se sustenta en el correcto funcionamiento del script responsable de realizar la solicitud a la base de datos y la respuesta correcta de este. Ello también implica que el script responsable de la visualización de la información funcione correctamente ya que es a partir de esto último que se comprueba el correcto funcionamiento de la página.
- Correcto funcionamiento del algoritmo de ordenamiento y así mismo del sistema de puntuación de proveedores y servicios:
  - Este punto es bastante importante ya que este sistema debe de ser inalterable por parte de

los proveedores (ellos mismo no pueden votar para sí mismos ni tampoco pueden manipular la calificación recibida) cosa que tiene que ser monitoreada y auditada durante el desarrollo y sobre todo durante y después del lanzamiento del proyecto.

- Correcta clasificación de cada servicio de modo que los filtros aplicables por el usuario para la visualización de la lista funcionen correctamente.

Selección y contratación de un servicio o proveedor por parte de un usuario formal.

- *Descripción:* Permite a un usuario formal elegir cualquiera de las opciones mostradas en la visualización de servicios y proveedores y solicitar un servicio (contrato) (esto sujeto a la disponibilidad dada por el proveedor).

- *Acciones iniciadoras y comportamiento esperado*:

Una vez el usuario este en la sección del listado de servicios y proveedores, si este se encuentra con la sesión iniciada podrá acceder a cualquiera de estos y elegir la opción "contratar", esto inicializará un evento en la página que desencadenará un método en donde: 1ro se verifique la disponibilidad del proveedor (en caso de no estar disponible el proceso de contratación quedará en estado de espera), esto abrirá un chat con el proveedor en donde se especificarán ciertos aspectos de conveniencia para ambos, una vez se llegue a un acuerdo existirá la opción de "confirmar contrato", al seleccionar esta opción se ejecutará otro método en la plataforma el cuál almacena en un objeto de la clase "contrato" variables como el precio (establecido previamente), el horario definido (cita), los actores de dicho contrato (nombres del usuario formal y proveedor como participantes del contrato) y el estado actual del contrato que tendrá 2 (por ahora) estados: En proceso (durante) y finalizado (después), este estado cambiará a su ultimo estado una vez el cliente califique al proveedor de dicho servicio (lo que hace que la calificación sea necesaria para continuar con el proceso de la plataforma y asegura que el sistema de puntos funcione de mejor manera.)

Requerimientos funcionales:

- Como requerimiento principal para el correcto funcionamiento de esta función se requiere que la función "Visualización del listado de proveedores y los servicios disponibles para contratar" funcione correctamente ya que esta función se desprende directamente del funcionamiento de esta otra.
- Disponibilidad de la base de datos:
    - Como bien se ha repetido, este es un requerimiento funcional clave para todas las funcionalidades importantes de este proyecto. Para este caso se requiere de esto debido al uso de datos tanto del usuario formal solicitante del servicio como los del usuario proveedor del respectivo servicio para la ejecución del proceso de contratación en todas sus fases.
- Disponibilidad de una herramienta de chat o de comunicación entre el usuario formal y el proveedor:
    - Se requiere del desarrollo de una herramienta de chat en tiempo real o en su defecto de la implementación de una herramienta que permita satisfacer esta necesidad clave dentro del proceso de acuerdo y contratación del servicio

---

Almacenamiento de los contratos finalizados en forma de historial.

- *Descripción: Le permite a un usuario (sea proveedor o formal) visualizar un listado de los contratos concluidos en orden cronológico, esta opción estará disponible para cualquier usuario con sesión iniciada en la sección de "perfil".*

V. DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR

Se realizó un diseño de interfaz gráfica preliminar utilizando HTML y una hoja de estilos CSS, se representan las distintas imágenes de la interfaz enfocada a las funcionalidades que requiere el software. Ver anexos [1]

VI. ENTORNOS DE DESARROLLO Y DE OPERACIÓN

El software está desarrollado en base al entorno de desarrollo node.js usando el lenguaje de programación JavaScript, así mismo para la interfaz se utilizaron herramientas de desarrollo web tales como HTML5 (usando el framework Bootstrap 4 en algunos aspectos) y CSS3. El sistema operativo en el cual fue desarrollado este prototipo es Windows y en cuanto a hardware usado para correr las principales pruebas con las estructuras de datos contiene las siguientes características:

- Fabricante: Acer
- Modelo: Nitro AN515-52
- Procesador: Intel®Core ™ i5-8300H CPU @ 2.30 GHz 2.30GHz
- Memoria Ram: 16.0 GB DDR4 SDRAM
- GPU: Gtx 1050 4gb
- Sistema operativo: Windows 10 - 64 Bits

VII. PROTOTIPO DE SOFTWARE INICIAL

El prototipo organizado de la manera solicitada y ubicado en una rama única para su entrega. se encuentra en el repositorio deGitHub: https://github.com/VbarraganP/ProyectoEstructuras

Además de esto se presenta el modelo relacional seguido en el desarrollo de la programación aclarando que está pensado para una entrega final de software donde las funcionalidades, clases que no es encuentren implementadas están involucradas con

otras estructuras de datos, diferentes operaciones de ordenamiento o simplemente un almacenamiento de datos que no se tiene en esta entrega. Ver Anexo [2].

VIII. PRUEBAS DEL PROTOTIPO

Para la realización de las pruebas se utilizaron 4 funcionalidades del software.

- Creación de Usuarios
- Eliminación de Proveedor
- Consulta total del Historial
- Consulta Total de proveedores por servicio

Se realiza un análisis de los resultados en la siguiente sección.

I. ANÁLISIS DE COMPARATIVO

Se presentan las tablas y gráficos correspondientes a las distintas funcionalidades. Ver anexos [3].

-Funcionalidad 1, Creación de Usuarios:
Se utilizó una librería de JavaScript denominada “Faker”, encargada de generar datos de prueba en todas las funcionalidades. Al medir la creación de ciertos objetos encontramos que el supuesto de la linealidad no es claro por el ajuste realizado (Valor R-Squared) y lo asociamos al funcionamiento de la librería donde la complejidad de esta funcionalidad parece ser cuadrática $O(N^2)$.

-Funcionalidad 2, Eliminación de Proveedor:
Luego de realizar la inserción en la lista doblemente enlazada que almacena los proveedores de ServiApp, se realizó la eliminación considerando el peor de los casos donde el proveedor a eliminar se encontraba en la posición N-1.
El resultado de las pruebas verificó el supuesto de la linealidad en la eliminación para una lista doblemente enlazada, donde la funcionalidad tiene una complejidad de O(N).

-Funcionalidad 3, Consulta total del historial:
Se realizó la creación de los diversos contratos que fueron insertados en la pila del Historial, se comprobó la linealidad de la operación de “recorrido” y acción (mostrar en consola) para una pila. La complejidad de la Consulta del historial es de O(N).

-Funcionalidad 4, Consulta Total de proveedor por Servicio:
Esta funcionalidad también realizaba una consulta total de datos pero en una lista doblemente enlazada, sin embargo se comprobó que es la funcionalidad más lineal con una complejidad de O(N).

II. ROLES Y ACTIVIDADES

Ver Anexo [4].

III. DIFICULTADES Y LECCIONES APRENDIDAS

1. Sobre la velocidad de procesamiento de las implementaciones:

- Realizar pruebas con un numero considerable de datos (10 millones de datos), resulta difícil de realizar porque en gran parte depende del hardware disponible para manejar esta cantidad en memoria.
- Medir el tiempo con un cronometro resulta demasiado inexacto porque existen muchos factores de error a considerar, para esto fue necesario buscar herramientas disponibles para medir el tiempo que fueran lo mas preciso posible, de lo contrario, de los resultados obtenidos se podría deducir erróneamente.

2. Hacer uso de una metodología ágil de desarrollo para un proyecto relativamente pequeño resulta útil, sin embargo, es necesario aprender nuevas metodologías con el fin de adquirir un mayor eficiencia en el desarrollo del proyecto.

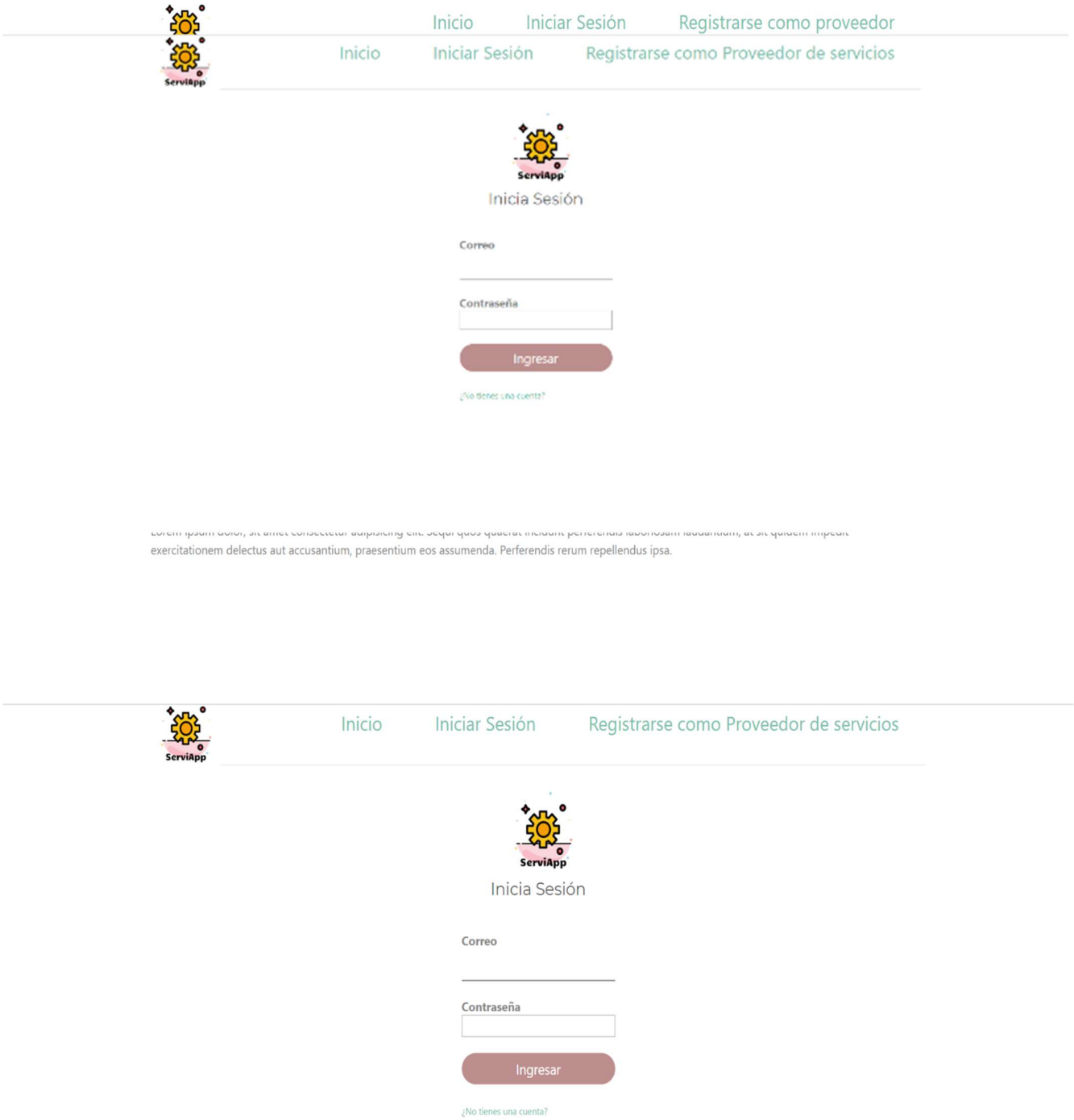
3. Es difícil depender solo de un entorno Front-End para implementar estructuras de datos y almacenar una gran magnitud de datos como los requeridos para las pruebas sobre este prototipo.

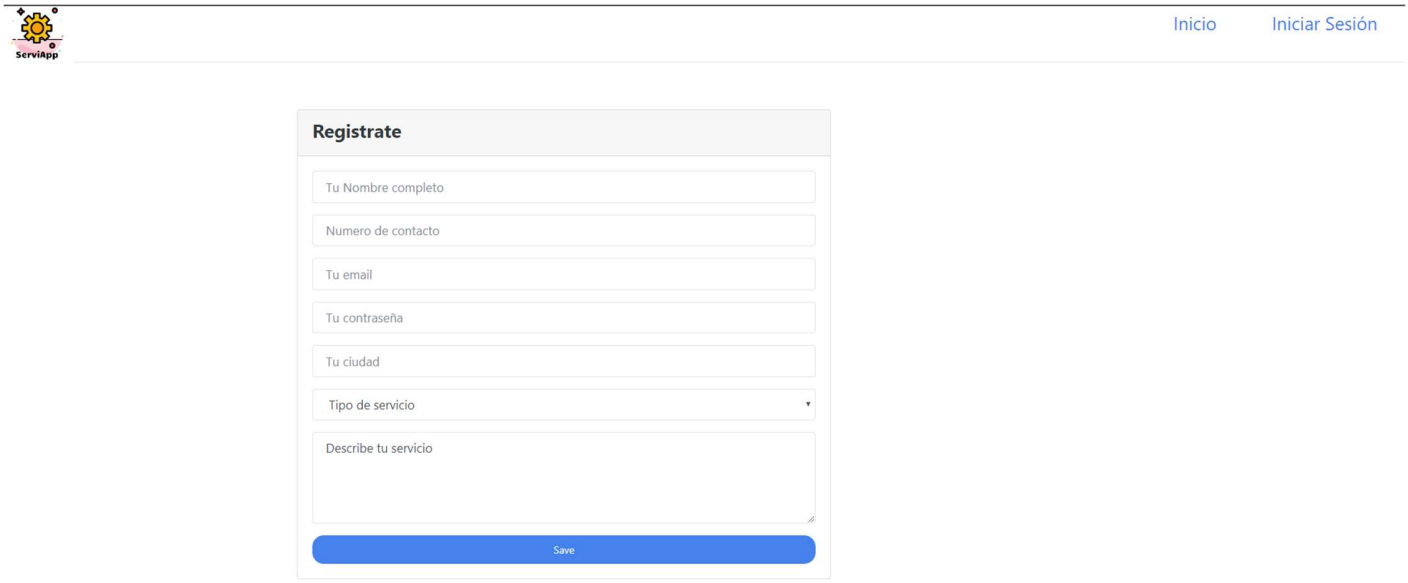
4.Una lección aprendida fue la gran utilidad y versatilidad que implica el conocer JavaScript, ya que a pesar de dedicarle tiempo a su aprendizaje con solo la idea de su funcionamiento en un entorno netamente del lado del navegador, resultó que es también una herramienta verdaderamente útil a la hora de hacer desarrollos “back-end” esto gracias a la existencia de la herramienta node.js

5.Lo anterior nos lleva a este punto ya que nos dimos cuenta de la importancia y del porqué de la existencia de un “Back-end” y un ”front-end” en el mundo del desarrollo web, ya que nos fue necesario recurrir a herramientas de back-end para ejecutar las implementaciones de las estructuras en la memoria de la computadora directamente y solucionar el problema de la imposibilidad de hacer esto sobre la estructura del navegador.

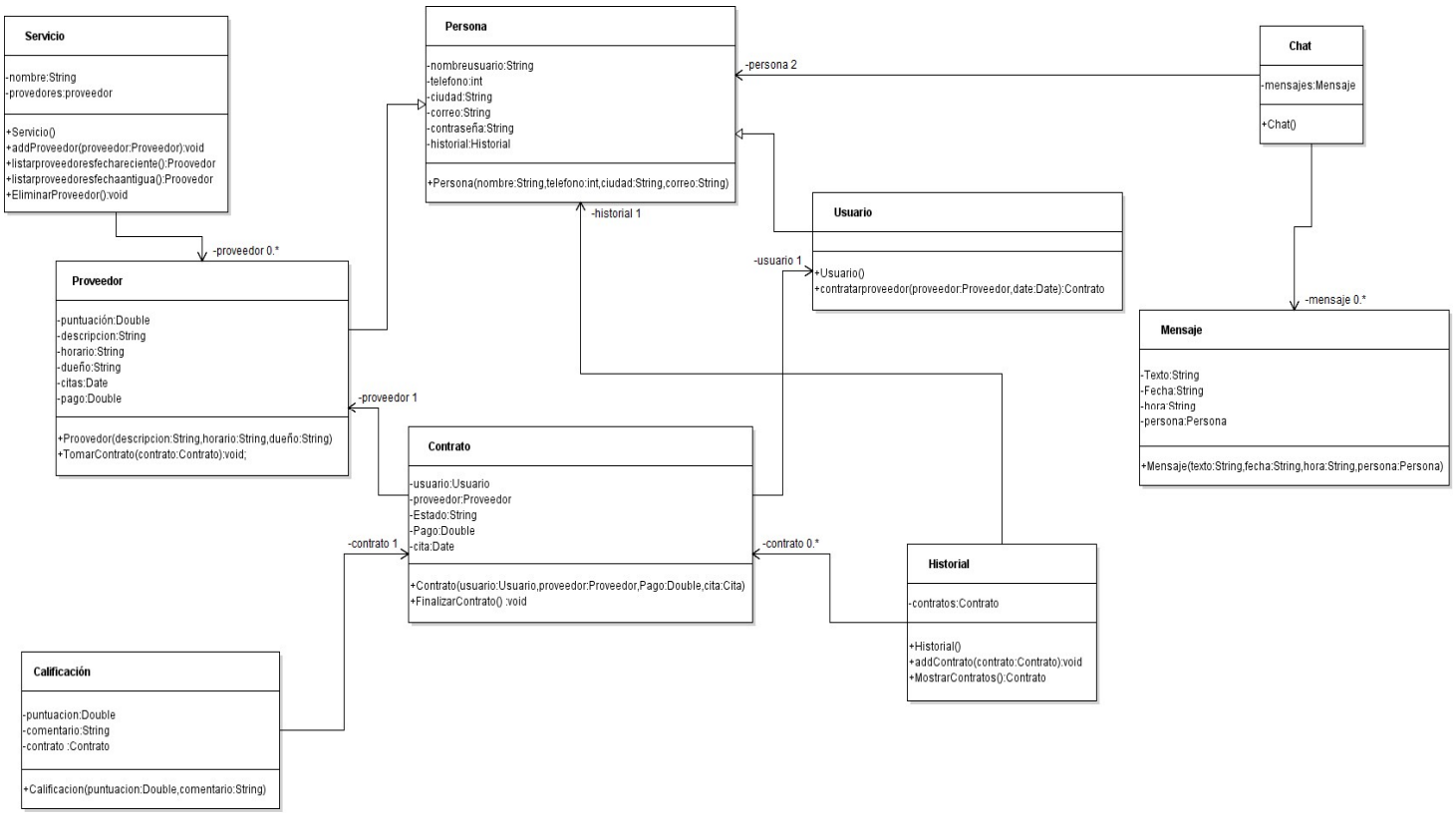
IV. ANEXOS

[1] Diseños correspondientes a la aplicación Web.





[2] Modelo Relacional del Proyecto:

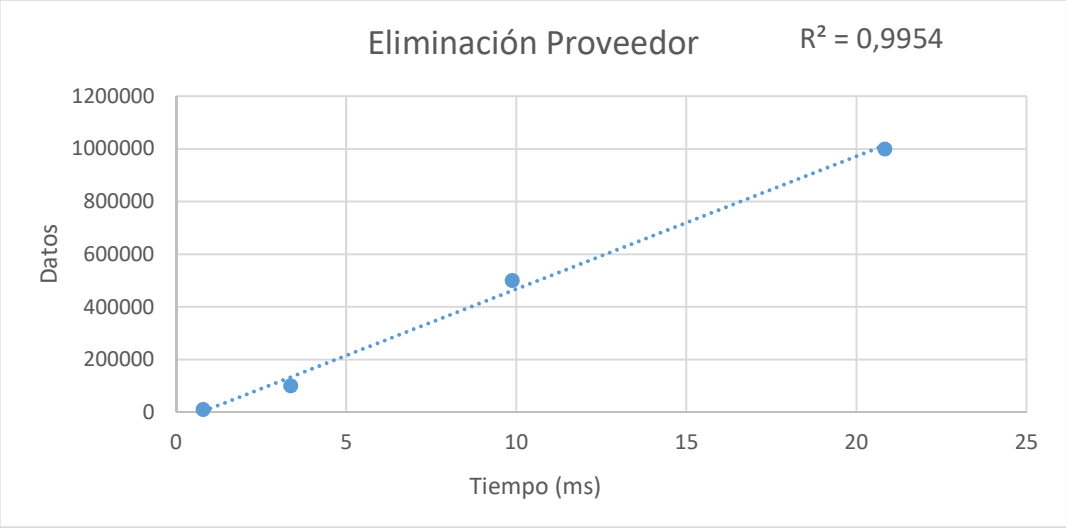


[3] Tabla de datos y Gráficos de complejidad
1) Funcionalidad: Creación de Usuarios

Tiempo (ms)	Datos
76.412501	10000
496.115601	100000
2433.7192	500000
3982.1804	1000000
7667.0175	10000000

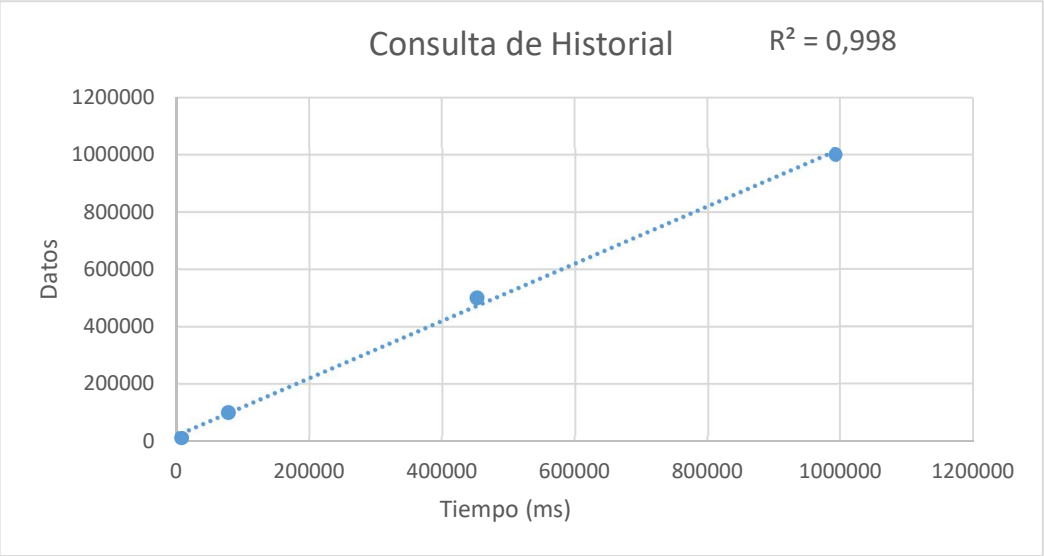
2) Funcionalidad: Eliminación de Proveedor

Tiempo	Datos
0.794099	10000
3.3673	100000
9.8837	500000
20.837	1000000



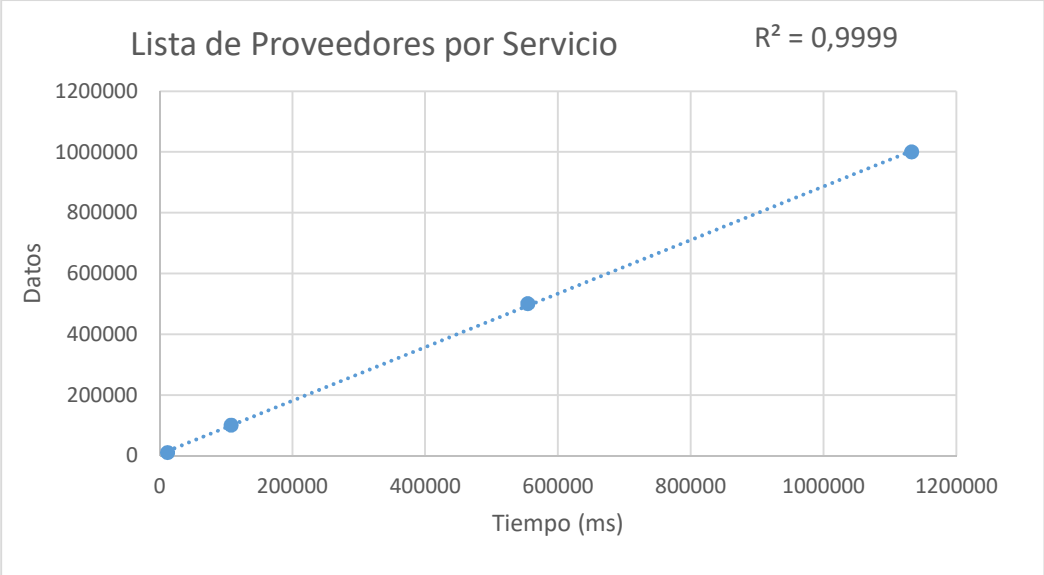
3) Funcionalidad: Consulta total de Historial

Tiempo (ms)	Datos
8148.6185	10000
78357.8260	100000
452288.05	500000
992146.597	1000000



4) Listar Proveedores por Servicio

Tiempo (ms)	Datos
11731.1835	10000
107532.437	100000
554141.86	500000
1132589.76	1000000



[4] Roles: (Tambien se puede encontrar en el repositorio del proyecto)

Integrante	Rol(es)	Actividades realizadas
Juan Felipe Rojas cendales	Coordinador	Programación de citas con el equipo de trabajo para resolver aspectos técnicos del problema.
	Técnico	Resolución de problemas para los códigos implementados en JavaScript
	Investigador	Investigación en el lenguaje de HTML y CSS para la interfaz , JavaScript en el programa
Fabian Leando Lopez Gomez	Observador	Mayor disponibilidad para la resolución de problemas de ultimo momento y asesoramiento en el uso de herramientas como Git
	Animador	Disponibilidad para solventar cuestiones (emocionales o de disponibilidad) del equipo por fuera del ambito tecnico bajo el marco del reto del confinaminamiento obligatorio
	Investigador	Investigación respecto las funciones y metodos disponibles en las herramientas trabajadas para facilitar el trabajo tecnico grupal e investigación acerca del uso del entorno de desarrollo
Victor Alfredo Barragan Paez	Lider	Consulta constante al equipo con el fin de mantener la constancia de la informacion y plan de trabajo.
	Investigador	Investigacion de lenguajes y frameworks de utilidad para el desarrollo del proyecto.
	Secretario	Documentacion de enlaces, instrucciones y tareas a realizar para facilitar la comunicacion del equipo