

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA DE SISTEMAS**

**UNIDAD DE TITULACIÓN**

**DESARROLLO DE UNA APLICACIÓN WEB PARA OFERTA Y  
DEMANDA DE SERVICIOS DE PERSONAS CON ALGÚN GRADO DE  
DISCAPACIDAD VISUAL.**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

**MIGUEL CLEMENTE AGUILAR DÍAZ**

(miguel.aguilar01@epn.edu.ec)

**ENRIQUE JAVIER VÁSCONEZ GONZÁLEZ**

(enrique.vasconez@epn.edu.ec)

**Director**

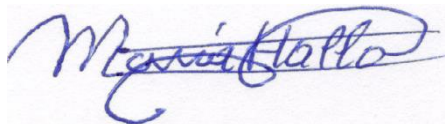
**María Asunción Hallo Carrasco**

(maria.hallo@epn.edu.ec)

**Quito, julio 2022**

## **APROBACIÓN DEL DIRECTOR**

Como director del trabajo de titulación Desarrollo de una aplicación web accesible a personas con discapacidad visual para oferta y demanda de servicios desarrollado por Miguel Clemente Aguilar Díaz y Enrique Javier Vásconez González, estudiantes de la carrera de Ingeniería de Sistemas Informáticos, habiendo supervisado la realización de este trabajo y realizado las correcciones correspondientes, doy por aprobada la redacción final del documento escrito para que prosiga con los trámites correspondientes a la sustentación de la Defensa oral.



---

**María Asunción Hallo Carrasco**

**DIRECTOR**

## DECLARACIÓN DE AUTORÍA

Yo, Miguel Clemente Aguilar Díaz, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



---

**Miguel Clemente Aguilar Díaz**

## DECLARACIÓN DE AUTORÍA

Yo, Enrique Javier Vásconez González, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



---

**Enrique Javier Vásconez González**

## ÍNDICE DE CONTENIDO

LISTA DE FIGURAS .....	i
LISTA DE TABLAS .....	iii
RESUMEN.....	iv
ABSTRACT .....	v
1.    Introducción.....	6
1.1.  Objetivo General .....	6
1.2.  Objetivos Específicos .....	6
1.3.  Marco Teórico.....	6
1.3.1.  Discapacidad.....	6
1.3.2.  Discapacidad Visual.....	8
1.3.3.  Técnicas de acceso a la información .....	9
1.3.4.  Sitio Web.....	11
1.3.5.  Aplicación <i>web</i> .....	11
1.3.6.  Diferencias entre sitio <i>web</i> y aplicación <i>web</i> . .....	12
1.3.7.  Estructura de una aplicación <i>web</i> .....	12
1.3.8.  Metodologías ágiles .....	14
1.3.8.1. <i>eXtreme Programming</i> (XP).....	15
1.3.8.2.  Scrum.....	15
1.3.9.  Accesibilidad web .....	16
1.3.9.1. <i>World Wide Web Consortium</i> (W3C).....	17
1.3.9.2.  Pautas de accesibilidad <i>web</i> .....	17
1.3.10.  Legislación ecuatoriana para el comercio electrónico.....	19
1.3.11.  Trabajos relacionados.....	20
2.    Metodología .....	22
2.1.  Herramientas y tecnologías utilizadas.....	24
2.2. <i>Scrum</i> .....	27
2.2.1.  Roles.....	27
2.2.2.  Eventos.....	28
2.2.3.  Artefactos.....	28

2.3.	Desarrollo de la aplicación .....	31
2.3.1.	<i>Sprint 1</i> .....	31
2.3.2.	<i>Sprint 2</i> .....	45
2.3.3.	<i>Sprint 3</i> .....	52
2.3.4.	<i>Sprint 4</i> .....	54
2.3.5.	<i>Sprint 5</i> .....	59
2.3.6.	<i>Sprint 6</i> .....	61
3.	Resultados .....	65
3.1.	Arquitectura en capas .....	66
3.2.	Esquema base de datos .....	67
3.3.	Nivel de accesibilidad de aplicación web " <i>Blind Service</i> ".....	67
3.4.	Comparativa de aplicación web <i>Blind Service</i> con aplicaciones similares. .	68
3.5.	Resultados de pruebas de accesibilidad con personas con algún grado de discapacidad visual.....	71
4.	Conclusiones y Recomendaciones.....	85
4.1.	Conclusiones.....	85
4.2.	Recomendaciones.....	86
5.	Anexos.....	87
6.	Bibliografía .....	88

## LISTA DE FIGURAS

Figura 1. Grado de discapacidad .....	9
Figura 2. Grupos etarios .....	9
Figura 3. Uso comercio electrónico en Ecuador .....	20
Figura 4. Arquitectura de la aplicación web.....	32
Figura 5. Esquema de base de datos .....	35
Figura 6. Página principal.....	36
Figura 7. Registro de usuario .....	37
Figura 8. Inicio de sesión .....	37
Figura 9. Página principal con usuario registrado.....	38
Figura 10. Formulario para publicar servicio .....	38
Figura 11. Detalle servicio .....	39
Figura 12. Perfil del usuario.....	40
Figura 13. Actualización de servicio.....	40
Figura 14. Solicitar servicios.....	41
Figura 15. Agregar servicios.....	42
Figura 16. Ver propuestas .....	42
Figura 17. Estructura básica de archivos React .....	43
Figura 18. Estructura de carpetas general.....	43
Figura 19. Conexión a base de datos .....	44
Figura 20. Pantalla de registro de usuario .....	46
Figura 21. Validación de errores en pantalla de registro .....	47
Figura 22. Pantalla de inicio de sesión de usuario .....	47
Figura 23. Validación de errores en pantalla de inicio de sesión .....	48
Figura 24. Página principal .....	48
Figura 25. Página de detalle de servicio .....	49
Figura 26. Pantalla de perfil de usuario.....	49
Figura 27. Pantalla para publicar un servicio.....	50
Figura 28. <i>localStorage</i> .....	51
Figura 29. <i>localStorage</i> en funcionamiento .....	51
Figura 30. Recursos de <i>backend</i> .....	53

Figura 31. Métodos del <i>backend</i> .....	53
Figura 32. Módulo de imágenes .....	54
Figura 33. Barra de búsqueda .....	56
Figura 34. Despliegue en <i>Heroku</i> .....	56
Figura 35. Despliegue automático.....	57
Figura 36. Código de conexión a <i>firebase</i> .....	57
Figura 37. Paginación.....	58
Figura 38. Total servicios por página.....	58
Figura 39. Métodos de <i>score</i> (calificación) .....	60
Figura 40. Sección para solicitar servicios.....	60
Figura 41. Sección de comentarios.....	61
Figura 42. Formato de registro para casos de prueba.....	64
Figura 43. Grado de accesibilidad <i>Lighthouse</i> .....	68
Figura 44. Cantidad de errores encontrados por <i>Lighthouse</i> .....	69
Figura 45. Cantidad de errores encontrados por <i>Wave</i> .....	70
Figura 46. Cantidad de errores encontrados por <i>ARC Toolkit</i> .....	70
Figura 47. Cumplimiento por Caso de Prueba .....	72
Figura 48. Dificultad por tarea y grado de discapacidad para CP1 .....	72
Figura 49. Dificultad por tarea y grado de discapacidad para CP2 .....	73
Figura 50. Dificultad por tarea y grado de discapacidad para CU3 .....	74
Figura 51. Dificultad por tarea y grado de discapacidad para CP4 .....	75
Figura 52. Dificultad por tarea y grado de discapacidad para CP5 .....	76
Figura 53. Dificultad por tarea y grado de discapacidad para CP6 .....	77
Figura 54. Dificultad por tarea y grado de discapacidad para CP7 .....	78
Figura 55 Dificultad por tarea y grado de discapacidad para CP8.....	79
Figura 56. Dificultad por tarea y grado de discapacidad para CP9 .....	80
Figura 57. Dificultad por tarea y grado de discapacidad para CP10 .....	81



## LISTA DE TABLAS

Tabla 1. Diferencias entre sitio <i>web</i> y aplicación <i>web</i> .....	12
Tabla 2. Diferencia y promedio del nivel de dificultad para CP1 .....	73
Tabla 3. Diferencia y promedio del nivel de dificultad para CP2.....	74
Tabla 4. Diferencia y promedio del nivel de dificultad para CP3.....	75
Tabla 5. Diferencia y promedio del nivel de dificultad para CP4.....	76
Tabla 6. Diferencia y promedio del nivel de dificultad para CP5.....	77
Tabla 7. Diferencia y promedio del nivel de dificultad para CP6.....	78
Tabla 8. Diferencia y promedio del nivel de dificultad para CP7.....	79
Tabla 9. Diferencia y promedio del nivel de dificultad para CP8.....	80
Tabla 10. Diferencia y promedio del nivel de dificultad para CP9.....	81
Tabla 11. Diferencia y promedio del nivel de dificultad para CP10 .....	82
Tabla 12. Variación porcentual total.....	82

## RESUMEN

En la actualidad existen personas con discapacidad visual que poseen cierto nivel de dificultad en acceder a páginas y aplicaciones web. Sumado a las dificultades que poseen las personas en adquirir servicios, existen páginas dedicadas a la oferta de los mismos, las cuales no aceptan solicitudes de servicios específicos por parte de los usuarios, como trabajos manuales o enseñanzas. Para resolver esta necesidad se desarrolló una aplicación web que permita a personas con discapacidad visual ofertar y demandar los servicios que deseen.

Esta aplicación fue desarrollada utilizando una adaptación del marco de trabajo *Scrum*. Todo el desarrollo estuvo enfocado en cubrir la mayor cantidad de problemas de accesibilidad que puedan presentarse en una aplicación, respaldando el correcto avance con reportes generados de aplicaciones encargadas de medir el nivel de accesibilidad, además de generalidades necesarias como el desarrollo de un *backend* con *frameworks* populares como *NestJS*.

*Blind Service*, como fue nombrada la aplicación web, fue probada con ayuda de voluntarios que poseen algún grado de discapacidad visual, los cuales completaron las pruebas propuestas. Se pudo validar el interés de instituciones como la Institución Educativa Especializada Mariana de Jesús y la Sociedad de Ciegos de Pichincha Luis Braille de impulsar este tipo de aplicaciones, ya que la demanda de un servicio como parte integral de una aplicación aporta un gran beneficio para este grupo de personas.

**Palabras clave:** Accesibilidad *web*, aplicación, *Scrum*, tecnologías

## ABSTRACT

Currently, there are people with visual disabilities who have a certain level of difficulty in accessing web pages and applications. In addition to the difficulties that people have in acquiring services, there are pages dedicated to the supply of services, which do not accept requests for specific services by users, such as manual labor or teaching. From this need, a web application was developed to allow visually impaired people to offer and request the services they want.

This app was developed using an adaptation of the Scrum framework. All the development was focused on covering as many accessibility issues that may arise in an application, supporting the correct progress with reports generated from applications responsible for measuring the level of accessibility, in addition to generalities such as the development of a backend with popular frameworks like NestJS.

Blind Service, as the web app was named, was tested with the help of volunteers with some degree of visual impairment, who completed the proposed tests. It was possible to validate the interest of institutions such as the Mariana de Jesus Specialized Educational Institution and the Luis Braille Society of the Blind of Pichincha in promoting this type of applications, since the demand for a service as an integral part of an application provides a great benefit for this group of people.

**Keywords:** App, web accessibility, Scrum, technologies

## **1. Introducción**

Esta aplicación web fue desarrollada con la finalidad de ofrecer accesibilidad a personas con algún grado de discapacidad visual, con la finalidad de ofertar y demandar servicios de una manera sencilla. En la actualidad existen dificultades en la adquisición de servicios, sea por desconocimiento de herramientas con características similares o por falta de habilidades en el manejo de dispositivos con acceso a internet.

El desarrollo de este trabajo de titulación beneficiará a personas con algún tipo de discapacidad visual, ya que la implementación de la aplicación seguirá las pautas de accesibilidad web para garantizar la accesibilidad. Como beneficio principal, las personas con discapacidad visual podrán hacer uso de esta aplicación web con ayuda de cualquier lector de pantalla con el que estén familiarizados.

### **1.1. Objetivo General**

Desarrollar una aplicación *web* para oferta y demanda de servicios para personas con algún grado de discapacidad visual.

### **1.2. Objetivos Específicos**

- Realizar un estudio de trabajos relacionados.
- Implementar una aplicación *web* accesible a personas con algún grado de discapacidad visual.
- Evaluar el nivel de accesibilidad de la aplicación desarrollada.

### **1.3. Marco Teórico**

En la presente sección se describen los principales conceptos relacionados con el desarrollo de este trabajo de titulación.

#### **1.3.1. Discapacidad**

Según la Organización Mundial de la Salud (OMS) [1] se puede definir la discapacidad como un término que abarca las deficiencias, las limitaciones de las actividades, y las restricciones de la participación de una persona; las deficiencias son los problemas que afectan a una estructura o función corporal, las limitaciones de actividades hacen referencia a las dificultades para ejecutar acciones o tareas y las restricciones de la participación son problemas para actuar en situaciones vitales. Además, la discapacidad se puede describir como un fenómeno complejo que refleja una

interacción entre las características del organismo humano y las características de la sociedad en la que vive.

La OMS [1] estima que el 15% de la población mundial experimentan algún tipo de discapacidad. Entre 110 y 190 millones de personas experimentan una discapacidad considerable. En general las personas con discapacidad tienen más probabilidad obtener resultados socioeconómicos adversos, como un nivel de educación inferior, peor acceso a salud y tasas más altas de pobreza.

Las discapacidades pueden afectar a las personas de diferentes maneras, aun cuando tengan el mismo tipo de discapacidad. Algunas discapacidades pueden estar ocultas o no se pueden notar con facilidad. La falta de inclusión social y económica plena de las personas con discapacidad desemboca en inaccesibilidad a los entornos físicos y de transporte, a la falta de acceso a dispositivos y tecnologías de apoyo, además de la ausencia de adaptación a medios de comunicación, prestación de servicios y prejuicios discriminatorios presentes en la sociedad [2]. Entre los principales tipos de discapacidad podemos encontrar:

- La discapacidad motriz, la cual implica una disminución de la movilidad total o parcial de uno o más miembros del cuerpo.
- La discapacidad auditiva, que es un déficit total o parcial en la percepción que se evalúa por el grado de pérdida de la audición en cada oído.
- La discapacidad visual tiene que ver con la disminución parcial o total de la vista.
- La discapacidad intelectual, que hace relación en el déficit que se ubica el desarrollo intelectual de una persona.
- La discapacidad visceral, hace referencia a aquellos individuos que, debido a alguna deficiencia en la función de órganos internos no pueden desarrollar su vida con total plenitud [2], [3].

El presente trabajo se enfocará en la discapacidad visual, además sobre el desarrollo de una propuesta a manera de aplicación *web* la cual sea accesible y útil a personas dentro de este grupo socioeconómico.

### 1.3.2. Discapacidad Visual

Se considera discapacidad visual a cualquier alteración del sentido de la vista, de acuerdo al grado de limitación de la visión; se suele distinguir entre personas ciegas y personas con disminución visual a aquellas que no obtienen información a través de su visión, con quienes sí pueden obtener información mediante su visión respectivamente [4].

Dentro del grupo de personas con discapacidad visual existen diferentes manifestaciones, las cuales se rigen de acuerdo al tipo de limitación, ya sea por su agudeza o campo visual. Según *Centers for Disease Control and Prevention* (CDC) o en español Centros para el Control y Prevención de Enfermedades y la OMS [5], la clasificación de la agudeza visual y de la debilitación incluye:

- La agudeza visual inferior significa una visión entre 20/70 y 20/400 con la corrección, o un campo de visión de 20 grados o menos.
- La ceguera se define como una agudeza visual peor de 20/400 con la corrección, o campo de visión de 10 grados o menos
- La ceguera en los Estados Unidos significa la agudeza visual de 20/200 o peor con la corrección, o un campo de visión de 20 grados o menos.

En el mundo existen aproximadamente 188,5 millones de personas que tienen una deficiencia visual moderada, 217 millones que tienen una deficiencia visual de moderada a grave y 36 millones son consideradas ciegas [6].

Según el Consejo Nacional para la Igualdad de Discapacidades (CONADIS) [7] en el Ecuador existen alrededor de 471.205 personas con algún tipo de discapacidad visual registradas.

En la figura 1 podemos observar la distribución de las personas con este tipo de discapacidad:

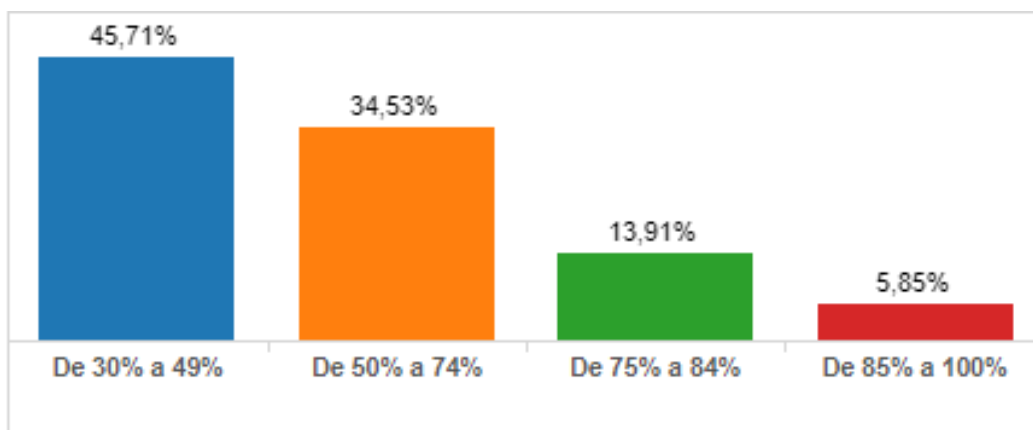


Figura 1. Grado de discapacidad [7]

El mayor grupo de personas con discapacidad visual tienen edades entre 36 a 64 años con un 40,72%, como se puede observar en la figura 2.

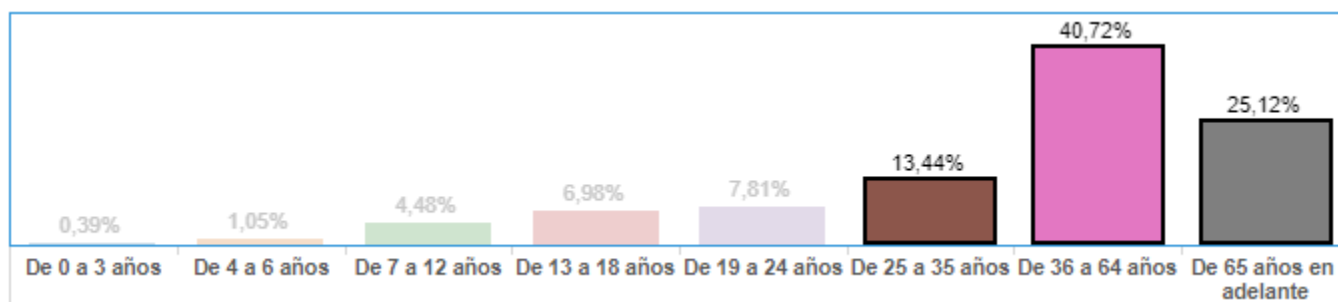


Figura 2. Grupos etarios [7]

Lo cual indica que en Ecuador existe una gran cantidad de personas que requieren atención de cualquier tipo para mejorar su calidad de vida, ya sea con herramientas tecnológicas que faciliten este propósito.

### 1.3.3. Técnicas de acceso a la información

No es nuevo que las personas con discapacidad visual tienen dificultades para acceder a mucha de la información presente en nuestro medio, ya sea texto en un letrero o una noticia en un periódico, las personas con discapacidad visual se encuentran en total desventaja con respecto a los demás en acceder al contenido de estos medios ya que en el mercado las tecnologías y los esfuerzos no han sido totales sobre estos grupos [8].

En el caso de la tecnología, más específicamente los computadores, las personas con discapacidad visual son incapaces de abstraer información que se encuentra en

un monitor, por ello no pueden interactuar con algún tipo de aplicación, o algún tipo de sitio/aplicación web que se esté visualizando en el computador. Por ello, un monitor y un *mouse* son indistintos para las personas con esta discapacidad, ya que, son incapaces de mover el ratón, no son capaces de visualizar la información representada por este dispositivo en el monitor [9].

Para ello se han desarrollado ayudas técnicas en temas de acceso a la información para personas con discapacidad visual, las cuales se las denominan productos tiflotécnicos<sup>1</sup> que han sido diseñados como productos especiales para ser utilizados por personas con discapacidad visual. Cuando se trata de ayudas técnicas para personas que poseen un resto funcional de la visión se habla de productos para baja visión, que son ayudas ópticas mecánicas o electrónicas que se utilizan para corregir, mejorar la capacidad visual de una persona [10]. A continuación, se describirán las principales ayudas para acceder a la información en una pantalla de un computador:

- **Programas de ampliación de caracteres:** Son programas o dispositivos de *hardware* que permiten visualizar la pantalla con un considerable aumento en el tamaño lo que facilita el uso y el acceso a la información de pantalla. Se pueden manejar con el teclado y con el ratón [11], [12].
- **Lectores de pantalla:** Son programas especializados que permiten la utilización de un sistema operativo y sus aplicaciones mediante un sintetizador de voz que lee y explica lo que se visualiza en pantalla [13]. Los lectores de pantalla fortalecen el aprendizaje de las personas con discapacidad visual severa al eliminar las barreras cognitivas-tecnológicas, generan mayores oportunidades de acceso a las diversas fuentes de información y fomentan el aprendizaje inclusivo [14].
- **Líneas braille:** Es un dispositivo de salida compuesto de un conjunto de celas que por medio de un mecanismo electromecánico que sube o baja un conjunto de 6 u 8 puntos por celda, lo que permite representar un carácter braille en cada una [15].

---

<sup>1</sup> Tiflotecnología: Conjunto de técnicas, conocimientos y recursos encaminados a procurar a las personas con ceguera o discapacidad visual grave los medios oportunos para la correcta utilización de la tecnología [76].



Los avances en accesibilidad se han logrado gracias a la tecnología, entre las cuales se pueden destacar el Internet, el cual permite que las personas discapacidad visual puedan acceder a sitios/aplicaciones web diseñados de acuerdo a pautas de un diseño universal [16].

#### **1.3.4. Sitio Web**

Un sitio *web* es un conjunto de páginas o documentos relacionados que se encuentran en internet y se identifican mediante el uso de nombres de dominio. Los sitios *web* pueden ser creados y mantenidos por una persona, un grupo, empresa u organización[17].

Existen una variedad de sitios *web* como sitios educativos, sitios de noticias, sitios de redes sociales, sitios de comercio electrónico, etc. Si bien no hay reglas que dicten como realizar un sitio *web* muchos sitios se rigen bajo un patrón estándar de una página de inicio que enlaza con las otras categorías y contenido del sitio *web*. La página de inicio, *home* o *index* representa la página principal del sitio que permite acceder a todas las demás páginas. Los sitios *web* están alojados en servidores y requieren un navegador web como *Chrome*, *Firefox*, *Microsoft Edge* para ser visitado ya sea por medio de un computador o un dispositivo móvil [18].

#### **1.3.5. Aplicación web**

Una aplicación *web* es un software de computadora que se almacena en un servidor remoto, al cual se accede por medio de un navegador de internet, a menudo se encuentra conectado a una base de datos para promocionar una experiencia interactiva específica a cada usuario. En las aplicaciones *web*, los datos se pueden referenciar, almacenar y acceder a través de una interfaz personalizada para la entrega de información [19]. A continuación, se enumeran unos cuantos ejemplos de aplicaciones *web* conocidas:

- *Twitter*
- *Facebook*
- *Amazon*
- *Google Docs*

### 1.3.6. Diferencias entre sitio *web* y aplicación *web*.

En la Tabla 1 se muestra las principales diferencias en un sitio *web* y una aplicación *web*.

Tabla 1. Diferencias entre sitio *web* y aplicación *web* [20], [21]

<b>Aplicación <i>web</i></b>	<b>Sitio <i>web</i></b>
Diseñada para interactuar con los usuarios finales	Su contenido es estático
El usuario puede manipular los datos	El usuario solo puede leer los datos, pero no manipularlos
Se debe compilar antes de su implementación	No es necesario precompilar para su implementación
La aplicación <i>web</i> es parte de un sitio <i>web</i> . La ampliación <i>web</i> no es un sitio <i>web</i> completo	Es un producto complejo el cual se accede por medio de un navegador.
Todos los cambios efectuados requieren que se vuelva a compilar e implementar todo el proyecto.	Los cambios pequeños no requieren volver a compilar solo se necesita actualizar el código <i>HTML</i>
Por lo general las aplicaciones <i>web</i> requieren autenticación ya que ofrecen opciones más amplias que un sitio <i>web</i>	La autenticación no es obligatoria para sitios <i>web</i> informativos.

### 1.3.7. Estructura de una aplicación *web*

No existe una regla escrita de que debe contener una aplicación *web*, pero se pueden resumir tres capas principales:

- 1. Interfaz gráfica:** Es la primera capa utilizada directamente por los usuarios, esta capa es ejecutada por un navegador web. Se implementan utilizando el lenguaje *HTML* o *XML*.
- 2. Servidor:** Se encarga de recibir todas las solicitudes realizadas por los usuarios, las interpreta, ejecuta y devuelve la información requerida al usuario. También se encarga de establecer conexión con la base de datos para introducir, actualizar, borrar o consular la información requerida por el usuario.

Para cumplir con sus funciones el servidor debe tener la capacidad de estar siempre encendido para evitar interrumpir el servicio que ofrece a los usuarios. Si el servidor falla o se apaga, los usuarios tendrán problemas al ingresar al sitio o aplicación *web*. [22].

Entre los servidores más utilizados se pueden destacar:

- *Apache*
- *Nginx*
- *LiteSpeed*
- *Microsoft IIS*

3. **Base de datos:** Es donde se almacena toda la información de la aplicación, la base de datos es un conjunto de datos estructurado y almacenado de forma sistemática con el objetivo de facilitar su posterior utilización [23].

Las herramientas de base de datos más utilizados son:

- *MySQL*
- *Oracle*
- *Postgre SQL*
- *Microsoft Access*
- *Microsoft SQL Server*

#### 1.3.7.1. Tipos de aplicaciones *web*

- **Aplicaciones *web* estáticas:** Es cualquier aplicación *web* que puede entregarse directamente al navegador del usuario final sin ningún cambio del contenido *HTML*, *CSS* o *JavaScript* del lado del servidor. Como ejemplos se puede nombrar un portafolio en línea, currículos digitales o una página de presentación de una empresa [24], [25].
- **Aplicaciones *web* dinámicas:** Son aplicaciones que generan los datos en tiempo real, según la solicitud del usuario. Utilizan bases de datos para cargar la información, el contenido se actualiza cada vez que el usuario accede a la aplicación [25]. Este tipo de aplicaciones cuentan con un *CMS* (Sistema de gestión de contenido) que permite a los administradores o usuarios actualizar o editar el contenido de la aplicación. Existen muchos lenguajes de

programación para desarrollar aplicaciones *web* dinámicas entre los más conocidos se encuentran *PHP* y *ASP* [26].

- **Tienda virtual o comercio electrónico:** Modelo de negocios basado en transacciones de productos y servicios. Su proceso de desarrollo es más complejo ya que requiere una forma de realizar pagos de forma electrónica ya sea por tarjeta de crédito, *PayPal*, etc [27],[28].
- **Portal *web app*:** Es una plataforma que proporciona a los usuarios un único punto de acceso a la información a través de una única interfaz [28]. Los portales *web* se centran en los usuarios y cada usuario tiene su propio identificador para acceder a su contenido. Los portales tienen contenido dinámico, es decir, su contenido cambia cada vez que un usuario los visita [29].
- **Aplicación *web* con gestor de contenidos:** Es una aplicación *web* que permite a los usuarios crear, administrar, y modificar el contenido de un sitio web sin tener conocimientos especializados [30]. Cuentan con una interfaz que controla una o varias bases de datos donde se alojan los sitios *web* y permite manejarlos de manera independiente el contenido y el diseño [31].

El presente trabajo se enfocará en las aplicaciones de comercio electrónico que de forma más amplia se las puede definir como “el comercio de productos o servicios a través de redes informáticas, como el Internet. El comercio electrónico se basa en tecnologías como el comercio móvil, la transferencia electrónica de fondos, la gestión de la cadena de suministro, el marketing en Internet, el procesamiento de transacciones en línea, el intercambio electrónico de datos (EDI), los sistemas de gestión de inventario y los sistemas automatizados de recopilación de datos. El comercio electrónico moderno suele utilizar la *World Wide Web* durante al menos una parte del ciclo de vida de la transacción, aunque también puede utilizar otras tecnologías como el correo electrónico.” [32].

### 1.3.8. Metodologías ágiles

Las metodologías ágiles son una forma de trabajar, que consiste en dividir los proyectos de software en pequeñas partes, lo que permite adaptar sobre la marcha, complementar y resolver etapas en un periodo de tiempo corto de tiempo para aumentar la satisfacción del cliente [33], [34]. Dentro de este contexto podemos encontrar los marcos de trabajo como lo es Scrum.

### 1.3.8.1. *eXtreme Programming (XP)*

La metodología XP define cuatro variables para cualquier proyecto de software:

- costo
- tiempo
- calidad
- alcance

De estas cuatro variables, sólo tres de ellas pueden ser fijadas por actores externos al grupo de desarrolladores (clientes y jefes de proyecto). El valor de la variable restante podrá ser establecido por el equipo de desarrollo, en función de los valores de las otras tres [35].

Ciclo de vida de XP [36], [37] :

- **Fase de exploración:** Es la fase en la que se define el alcance general del proyecto. En esta fase, el cliente define lo que necesita mediante la redacción de historias de usuarios.
- **Fase de planificación:** La planificación es una fase corta, en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario y las entregas asociadas.
- **Fase de iteraciones:** Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración.
- **Fase de puesta en producción:** Si bien al final de cada iteración se entregan módulos funcionales y sin errores, puede ser deseable por parte del cliente no poner el sistema en producción hasta que no se tenga la funcionalidad completa.

### 1.3.8.2. **Scrum**

*Scrum* abarca varios conceptos a nivel de roles como a nivel de procesos. A breves rasgos se puede mencionar que en este marco de trabajo existen tres roles, los cuales son los siguientes:

- **Scrum Master:** Es conocido como el facilitador de proyectos, es la figura que lidera el equipo *Scrum*.

- **Product Owner:** Es el encargado de la gestión y optimización del *Product backlog*.
- **Development team:** Son las personas encargadas del desarrollo del producto, y son los responsables de los entregables que entren dentro de la definición de terminado. Es importante resaltar que en este tipo de equipos no existe ninguna jerarquía, en cambio, son equipos autogestionados.

Además, existen términos que deben ser especificados al inicio del proyecto, uno de ellos es la definición de terminado, el cual es definido como la percepción para el equipo Scrum de que un avance del producto está terminado y listo para su puesta en producción [38].

Las etapas de Scrum a seguir serán las siguientes:

- **Sprint Planning (Planificación del Sprint):** Se puede entender como la definición del objetivo para el próximo avance/entregable del proyecto.
- **Development Stage (Etapas de desarrollo):** Es el trabajo del sprint en curso, los encargados garantizarán el cumplimiento de los objetivos especificados.
- **Daily Scrum (Scrum Diario):** Es una reunión diaria de máximo 15 minutos que se realizará al final del día.
- **Sprint Review (Revisión del Sprint):** Consiste en una reunión al final de cada *sprint*, en la cual se analizarán y evaluarán los resultados obtenidos.
- **Sprint Retrospective (Retroalimentación del Sprint):** En esta reunión se revisará lo sucedido en el *sprint* y se contemplarán las posibles mejoras.

### 1.3.9. Accesibilidad web

La accesibilidad *web* es un concepto relativamente nuevo que en esencia busca integrar a las personas con algún tipo de discapacidad al uso de la *web*. El hablar de accesibilidad significa el desarrollo de diseños *web* que permitan a las personas el fácil uso de estas páginas/aplicaciones con la finalidad de que las personas con discapacidad puedan percibir, entender, interactuar y navegar con facilidad. Además, la accesibilidad web no se limita a atender a las personas con discapacidad, en cambio beneficia a otros sectores como son las personas de la tercera edad [39].

La accesibilidad *web*, busca atender a muchos tipos de discapacidades, sus principales son las más recurrentes, como lo son las personas con discapacidad visual, auditiva, física, del habla, etc. Ya que en la actualidad la mayoría de los sitios

*web* y *software web* presentan notablemente barreras de uso de las mismas, lo que dificulta de gran manera su uso [39].

La accesibilidad web beneficia en gran medida al sector de las personas con discapacidad, pero también beneficia directamente a las empresas y organizaciones, ya que posibilita el consumo de diferentes productos o servicios a un mayor número de personas [39].

#### **1.3.9.1. World Wide Web Consortium (W3C)**

El *World Wide Web Consortium (W3C)* es un consorcio internacional que busca generar estándares y recomendaciones que aseguren el crecimiento de la *web*. Fue creado en octubre de 1994 y es dirigido por el creador original de *URL*, *HTTP* y *HTML*, las cuales son las tecnologías principales sobre las que se basa todo el contenido de internet [40].

#### **1.3.9.2. Pautas de accesibilidad web**

Las pautas de accesibilidad *web* definen como crear un contenido más accesible en la *web* para personas con algún tipo de discapacidad. Existe actualmente muchos tipos de discapacidad, como lo es la visual, la auditiva, físicas, cognitivas, etc. Pero, a pesar de que estas pautas intentan cubrir muchas de estas discapacidades, no son capaces de hacerlo totalmente, pero en este esfuerzo se incluyen varios niveles que pueden guiar este desarrollo *web*, entre ellos podemos encontrar los principios fundamentales, las pautas generales, los criterios de éxito siempre y cuando sean verificables, además de una amplia colección de técnicas aconsejadas, todo esto documentado con ejemplos y recursos en código [41].

Existen 12 pautas agrupadas en 4 principios fundamentales, las cuales se mencionan a continuación:

**1. Perceptibilidad:** La información y los componentes de la interfaz del sitio web deben ser comprendidas por todos los usuarios [42].

1.1. Comprende alternativas textuales a todo el contenido *web* no textual.

1.2. Propone alternativas sincronizadas en contenidos del tipo multimedia.

1.3 Propone crear contenido presentable de diversas maneras, sean estos una composición más simple.

1.4 Buscar facilitar la comprensión de los usuarios a través de audios que separen el primer plano del fondo.

**2. Operabilidad:** Los componentes de la interfaz del sitio web y la navegación deben ser manejados por todos los usuarios [42].

2.1 Toda la funcionalidad deberá ser operable desde el teclado.

2.2 Proporcione el tiempo suficiente para leer y usar un contenido.

2.3 No se debe diseñar el contenido con la finalidad de causar ataques.

2.4 Proporcione localizadores de contenido, además se debe permitir al usuario determinar dónde se encuentran.

**3. Comprensibilidad:** La información del sitio web debe ser legible, comprensible y predecible para los usuarios [42].

3.1 El contenido textual debe ser legible y comprensible.

3.2 El comportamiento y el uso de la página web debe ser predecible.

3.3 Ayude a los usuarios a evitar y corregir errores.

**4. Robustez:** El sitio web debe ser compatible con la mayor cantidad de navegadores, sistemas operativos, dispositivos y aplicaciones de tecnologías de apoyo [43].

4.1 Maximice la compatibilidad con agentes de usuario, incluyendo tecnologías asistidas.

Además, estas pautas están asociadas a puntos de verificación, los cuales describen como se debe aplicar esa pauta. Cada punto de verificación tiene uno o tres niveles:

- Prioridad 1, si no se cumple algunos grupos de personas no serán capaces de acceder a la información.
- Prioridad 2, si no se cumple algunas personas encontrarán muchas dificultades de acceder a la información.
- Prioridad 3, si no se cumple algunas personas hallarán dificultades de acceder a la información.



Con la finalidad de facilitar la comprensión, los niveles de adecuación se han establecido de la siguiente manera:

- Nivel de adecuación A, si se cumple la prioridad 1.
- Nivel de adecuación AA, si se cumple la prioridad 1 y prioridad 2.
- Nivel de adecuación AAA, si se cumple la prioridad 1, prioridad 2 y prioridad 3.

#### **1.3.10. Legislación ecuatoriana para el comercio electrónico**

El comercio electrónico se encuentra contemplado dentro de la Ley No. 2002-67, ley de comercio electrónico, firmas electrónicas y mensajes de datos, la cual busca “Regular los mensajes de datos, la firma electrónica, los servicios de certificación, la contratación electrónica y telemática, la prestación de servicios electrónicos, a través de redes de información, incluido el comercio electrónico y la protección a los usuarios de estos sistemas” [44].

En base a esto, la Estrategia Nacional de Comercio Electrónico (ENCE) propone los siguientes componentes que buscan dinamizar el desarrollo del comercio electrónico en la actualidad:

1. Marco Legal, busca fortalecer la normativa vigente para el desarrollo satisfactorio de las actividades digitales [45].
2. Fomento electrónico de las MIPYMES (micro, pequeña y mediana empresa), con lo cual se busca capacitar, fortalecer y acompañar a los sectores de la sociedad para que adopten el comercio electrónico [45].
3. Sistema de pago electrónico, busca ayudar la adopción de las tecnologías de la información en transacciones, es decir en el pago de productos, bienes o servicios [45].
4. Logística en el proceso del comercio electrónico, busca impulsar el sector logístico y postal en un entorno seguro y confiable con ayuda de las TICs [45].

Para el caso de las personas con discapacidad, al igual que las personas sin ningún tipo de discapacidad, esta ley busca precautelar las transacciones en línea dando validez jurídica a toda aquella información explícita colocada en la web con finalidad de comercio.

Teniendo en cuenta esto, se puede observar que en Ecuador las compras por canales digitales han aumentado desde el primer cuartil del 2020, debido al distanciamiento

social, esto ha generado una oportunidad de apertura del comercio electrónico en el país, es decir, se han impulsado un 34% los usuarios de plataformas digitales como se muestra en la Figura 3, ahora las personas usan estas plataformas por evitar salir de casa a causa de la pandemia [45]

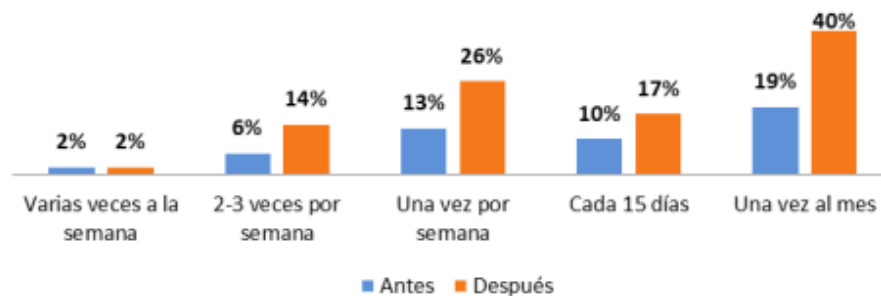


Figura 3. Uso comercio electrónico en Ecuador [45]

### 1.3.11. Trabajos relacionados

En la actualidad, existen pocas innovaciones que busquen facilitar la vida de las personas que poseen algún grado de discapacidad visual, esto puede ser evidenciado en la baja cantidad de modelos propuestos y aplicados, además de pocos productos y soluciones que enfocan a este colectivo como población principal [46]. Existen generalidades que se han puesto en marcha, como buenas prácticas en la generación de etiquetados en el cuerpo de una página web, como herramientas externas que buscan facilitar el acceso y adquisición de productos, pero que, a cierto nivel, no pueden solventar la gran demanda que se presentan en algunos países por estos servicios. Tal es el caso de una propuesta realizada en unión de Lazarillo con Sodimac Chile, a manera de aplicación web, la cual consiste en contactar a la persona con discapacidad visual, a través de videollamada, con un agente capacitado en la venta de mobiliarios para que sea el guía de la persona discapacitada en la compra de distintos tipos de artículos [47], esto a gran escala resulta en dificultades en la atención a personas con discapacidad visual ya que la cantidad de personas que pueden ser atendidas está estrechamente relacionada con la cantidad de agentes disponibles

Otro es el caso de Blind Bargains, una aplicación gratuita para dispositivos móviles que permite a personas con discapacidad visual adquirir productos como parlantes, impresoras braille, entre otros productos. Pero que, al parecer, por el poco apoyo que ha tenido en plataformas como Google Play y su limitado catálogo de productos, ha

ido desapareciendo del medio, dejando nada más que una idea de apoyo comunitario para este grupo social [48].

Además de aplicaciones que intentan cubrir el comercio electrónico existen propuestas orientadas a otros tipos de usos, como es el caso de una aplicación llamada Linguoo que nació principalmente como una aplicación de lectura de páginas web, como son las páginas de oferta de servicios y/o productos, pero que con el tiempo fue mutando hasta convertirse en una aplicación que permite generar listas de reproducción de artículos en voz y noticias [49]. Indirectamente relacionada al comercio podemos encontrar a KNFB Reader, una aplicación disponible en iOS, Android, y Windows 10 que tiene su propósito en ayudar a las personas con discapacidad visual a leer documentos escritos, como lo puede ser un catálogo de autos, un menú de comida, un catálogo de cortinas, etc [50].

No solo existen propuestas desarrolladas desde cero accesibles a personas con discapacidad visual, también existe la alternativa de mejorar el código con la finalidad de hacerlo accesible, esto aplica a cualquier tipo de aplicaciones web, sean blogs, de comercio electrónico o redes sociales, existen varios ejemplos como el trabajo de titulación “Desarrollo de aplicación web tipo Blog accesible para personas ciegas”, el cual toma código de una página básica HTML y es convertida en una aplicación web con accesibilidad con ayuda de WordPress, demostrando que la accesibilidad no se encuentra atada a tecnologías [51].

No solo pequeñas aplicaciones han tomado la posta en la adopción de la accesibilidad web, tal es el caso de Amazon, uno de los líderes en comercio electrónico, que, a finales del 2021, anuncio nuevas herramientas que buscan fortalecer la accesibilidad en sus productos, como lo es la página web de accesibilidad de Alexa, que con ayuda de la Asociación Mexicana para la atención de personas con discapacidad visual, están logrando reducir esta brecha existente en el acceso a los productos ofertados en sus plataformas [52].

La falta de aplicaciones webs que no consideran la accesibilidad web en el desarrollo de los mismos pierde una ventaja importante que es el potencial de ganancia financiera directas e indirectas por un mayor uso de la aplicación. La accesibilidad web puede hacer que sea más fácil para las personas encontrar un sitio web, acceder a él y utilizarlo con éxito [53].

Las aplicaciones y sitios web accesibles pueden tener mejores resultados de búsqueda, permiten reducir los costos de mantenimiento, aumentar el alcance de audiencia y demostrar la responsabilidad social corporativa (RSC). Por lo tanto, tener una aplicación web accesible y bien diseñado no solo hace que la aplicación esté disponible para personas con algún grado de discapacidad visual, sino que también permite mejorar significativamente la experiencia del usuario para todos los usuarios de su sitio [54].

Varios países poseen leyes y reglamentos que exigen la incorporación de pautas de accesibilidad web. Si esas leyes no existieran, la accesibilidad aún debería considerarse para el desarrollo de aplicaciones o sitios webs como ética moral, pero varias empresas solo priorizan a usuarios con algún grado de discapacidad para evitar consecuencias legales como posibles demandas [55].

En vista de todo esto, como equipo de trabajo, se busca desarrollar una aplicación web que mantenga en medida de lo posible el acceso para personas con discapacidad visual, que elimine la necesidad de terceros y permita una independencia total en la navegación para la adquisición y oferta de servicios en base a una aplicación web con secciones bien diferenciadas que apoyen el desarrollo de las actividades de este importante grupo.

## 2. Metodología

La metodología que se utilizará para el desarrollo de la aplicación *web* será XP junto con el marco de trabajo *Scrum* y los artefactos que este ofrece.

De XP se tomarán los siguientes pasos del ciclo de desarrollo [37]:

- **Fase de exploración:** Se definirán las historias de usuario y la estimación de tiempo de desarrollo, para este caso no se tomarán costos debido a que la aplicación se desarrolla sin fines de lucro.
- **Fase de planificación:** Se especificará el orden de implementación de las historias de usuario y sus entregables.
- **Fase de iteraciones:** Estará definida mayormente por los artefactos *Scrum* como *Sprint Planning*, *Sprint Review* y *Sprint Retrospective*, las cuales serán definidas más adelante en este apartado.
- **Fase de puesta en producción:** Será definida en base al criterio del cliente.

En Scrum existen tres roles que son:

- **Scrum Master:** Este rol será ejecutado por Enrique Vásconez.
- **Product Owner:** Este rol será ejecutado por un representante de una Sociedad De Ciegos De Pichincha.
- **Development Team:** Este rol será ejecutado por Miguel Aguilar y Enrique Vásconez.

Las etapas de Scrum a seguir serán las siguientes:

- **Sprint Planning:** Se realizará una reunión con el equipo Scrum donde se establecerán riesgos, plazos de entrega, funcionalidades, entre otros. Con fines prácticos las historias de usuario a ser ejecutadas serán escritas de la siguiente forma:

*Sprint n* [Campo] Nombre de la historia.

Donde:

- *Sprint n*, será el sprint en la que se ejecuta esta historia.
- [Campo], será el campo en el que se ejecutará la historia de usuario, estos campos pueden ser de infraestructura, base de datos, *frontend*, *backend*, etc.
- Nombre de la historia, será el nombre asignado a la historia de usuario.
- **Development Stage:** Es el trabajo del sprint en curso, los encargados garantizarán el cumplimiento de los objetivos especificados.
- **Daily Scrum:** Es una reunión diaria de máximo 15 minutos que se realizará al final del día.
- **Sprint Review:** En esta sección del *sprint* se realizarán comprobaciones de que el trabajo realizado cumpla con las pautas de accesibilidad.
- **Sprint Retrospective:** En esta reunión se revisará lo sucedido en el *sprint* y se contemplarán las posibles mejoras.

La duración estimada de cada *sprint* será de 30 días (un mes).

## 2.1. Herramientas y tecnologías utilizadas

### ***JavaScript***

Lenguaje de programación adaptado para el presente proyecto, está basado en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa [56].

### ***SQL (Structured Query Language)***

Lenguaje de programación que se utiliza para administrar base de datos relacionales y realizar diferentes operaciones con los datos que contienen. *SQL* permite la modificación de las estructuras de tablas de los índices de las bases de datos[57], este lenguaje es usado por TypeORM [58].

### ***HTML (HyperText Markup Language)***

Lenguaje de Marcas de Hipertexto es el componente más básico de una página *web*. Define el significado y la estructura del contenido *web*. [59]. *HTML* se escribe con elementos, estos elementos están constituidos por etiquetas, contenido y atributos [60] Fue utilizado dentro del *framework* de *ReactJS* para la generación de componentes.

### ***CSS (Cascading Style Sheets)***

Es el lenguaje de estilos utilizado para describir la presentación de documentos *HTML* o *XML* (incluyendo diferentes lenguajes basados en *XML* como *SVG*, *MathML* o *XHTML*). *CSS* es uno de los lenguajes base de la *Open Web* y posee una especificación estandarizada por parte del *W3C* [61]. Fue utilizado para la personalización de algunos elementos de la aplicación.

### ***Node.js***

Es un entorno de ejecución de *JavaScript* orientado a eventos asíncronos [62]. Permite la creación de aplicaciones de red rápidas, ya que es capaz de manejar una gran cantidad de conexiones simultáneas con un alto nivel de rendimiento, lo que equivale a una alta escalabilidad [63].

### **Visual Studio Code**

Es un editor de código fuente ligero pero potente que se ejecuta en el escritorio y está disponible para *Windows*, *macOS* y *Linux*. Viene con soporte incorporado para *JavaScript*, *TypeScript* y *Node.js* y tiene una gran variedad de extensiones para otros lenguajes como *C ++*, *C #*, *Java*, *Python*, *PHP*, *Go* y tiempos de ejecución como *.NET* y *Unity* [64]. Es el editor escogido para el desarrollo de la aplicación.

### **Bootstrap**

Es un *framework front-end* utilizado para desarrollar aplicaciones *web* y sitios *mobile* receptivos, es decir, con un *layout* que se adapta a la pantalla del dispositivo utilizado por el usuario. El *framework* combina *CSS* y *JavaScript* para estilizar los elementos de una página *HTML*. Fue utilizado para la personalización y adaptabilidad de los elementos de la aplicación a la pantalla de los dispositivos. [65].

### **Figma**

Es una aplicación de edición de gráficos y diseño de interfaz de usuario basada en la *web* *Figma* es diferente de otras herramientas de edición de gráficos. Principalmente porque funciona directamente en tu navegador [66]. Fue utilizado para diseñar los bosquejos del aplicativo.

### **DBeaver**

Es una herramienta de base de datos multiplataforma gratuita para desarrolladores, administradores de bases de datos, analistas y todas las personas que necesitan trabajar con bases de datos. Admite todas las bases de datos populares: *MySQL*, *PostgreSQL*, *SQLite*, *Oracle*, *DB2*, *SQL Server*, *Sybase*, *MS Access*, *Teradata*, *Firebird*, *Apache Hive*, *Phoenix*, *Presto*, etc [67]. Fue utilizado para la visualización y administración de la base de datos.

### **PowerDesigner**

Es una herramienta de modelización de datos. *PowerDesigner* permite a las empresas visualizar, analizar y manipular de manera más fácil los metadatos para tener una arquitectura de información de empresa eficaz. Fue utilizado para el diseño de la base de datos utilizada en la aplicación. [68].

## **Heroku**

*Heroku* es una plataforma en la nube como servicio (*PaaS*) basada en contenedores. Ofrece servicios de servidores y redes en donde se pueden alojar aplicaciones de diferentes lenguajes de programación como *Python*, *Java*, *PHP* y más [69]. Fue utilizado para desplegar la aplicación y realizar las pruebas de usuario de una forma más sencilla.

## **Swagger**

*Swagger* es un conjunto de reglas, especificaciones y herramientas de código abierto para desarrollar y describir *API RESTful*. El marco *Swagger* permite a los desarrolladores crear documentación *API* interactiva, legible por máquinas y por humanos [70]. Fue utilizado para la visualización de métodos y monitorización de datos presentes en el *backend*.

## **NestJS**

Es el marco usado para crear aplicaciones del lado del servidor *Node.js*. Utiliza *JavaScript* progresivo, está construido y es totalmente compatible con *TypeScript*, *FP* (Programación funcional) y *FRP* (Programación reactiva funcional) [71].

## **PostgreSQL**

Sistema de base de datos relacional usado, está orientado a objetos y de código abierto, que utiliza y amplía el lenguaje *SQL* combinado muchas características que almacenan y escalan de forma segura las cargas de trabajo de datos más complicadas [72].

## **Passport.js**

Es un middleware de autenticación para *Node.js*. Extremadamente flexible y modular, *Passport* se puede colocar discretamente en cualquier aplicación *web* basada en *Express* [73]. Fue utilizado para manejar la seguridad de la aplicación.

## **TypeORM**

Es un *ORM* (*Object-relational mapping* / Asignación objeto-relacional) que puede ejecutarse en las plataformas *NodeJS*, *Browser*, *Cordova*, *PhoneGap*, *Ionic*, *React Native*, *NativeScript*, *Expo* y *Electron* y puede usarse con *TypeScript* y *JavaScript*



(ES5, ES6, ES7, ES8). Su objetivo es admitir siempre las últimas funciones de *JavaScript* y proporcionar funciones adicionales que lo ayuden a desarrollar cualquier tipo de aplicación que utilice bases de datos [58]. Fue utilizado para el acceso a base de datos en backend.

### **React**

Es una biblioteca *JavaScript* para crear interfaces de usuario [74]. Fue utilizado para el desarrollo del *frontend* y la aplicación de los estándares de accesibilidad web,

### **JWT**

*JSON Web Token (JWT)* es un estándar abierto (*RFC 7519*) que define una forma compacta y autónoma de transmitir información de forma segura entre las partes como un objeto *JSON*. Esta información se puede verificar y confiar porque está firmada digitalmente [75] Fue utilizado para el manejo de contraseñas de usuario para el inicio de sesión en la aplicación.

## **2.2. Scrum**

A continuación, se detallarán como estarán compuestas para el presente proyecto las diferentes partes que comprenden el marco de trabajo *Scrum*, como lo son los roles, eventos y artefactos, además de la definición de cada uno de sus subelementos.

### **2.2.1. Roles**

#### **2.2.1.1. Product Owner**

Miguel Clemente Aguilar Díaz

#### **2.2.1.2. Scrum Master**

Enrique Javier Vásconez González

#### **2.2.1.3. Scrum Team**

- Enrique Javier Vásconez González
- Miguel Clemente Aguilar Díaz

## **2.2.2. Eventos**

### **2.2.2.1. *Sprint***

#### **2.2.2.1.1 *Sprint Planning***

La planificación del *sprint* se la llevará a cabo a inicio de cada mes, el tiempo estimado para esta será de dos horas máximo, en esta reunión se planificarán todas las historias de usuario que serán ejecutadas en el *sprint*, además de los *features* (características) y las tareas asociadas.

#### **2.2.2.1.2 *Daily Scrum***

Los *Scrum* diarios se llevarán a cabo cada día de la semana laboral. La duración de los *Scrum* diarios serán de máximo 15 minutos y se tratarán temas relacionados a al trabajo/avances realizados el día laborable anterior.

#### **2.2.2.1. *Sprint Review***

La revisión del *sprint* será llevada a cabo al final de cada *sprint*, en esta reunión se tratarán temas relacionados a las tareas completadas y pendientes del *sprint* que se tengan. Esta reunión tendrá un tiempo estimado de 30 minutos.

#### **2.2.2.5. *Sprint Retrospective***

La revisión del *sprint* será llevada a cabo al final de cada *sprint*, en esta reunión se tratarán temas relacionados a los logros alcanzados y aspectos de mejora para el equipo. Esta reunión tendrá un tiempo estimado de 30 minutos.

## **2.2.3. Artefactos.**

### **2.2.3.1. Historias de usuario.**

Los siguientes puntos corresponden a las historias de usuario obtenidas para la creación de la aplicación web. Esta sección corresponde a la fase de exploración y planificación de XP. Para facilitar la lectura se decidió omitir la introducción "Yo como usuario, deseo/quiero" de la siguiente lista:

- Desarrollar una aplicación que pueda ser usada por cualquier persona, sea que posea o no una discapacidad visual, para que no se restrinja el acceso a personas con necesidades de servicios.

- Publicar, solicitar y agregar servicios de todo tipo para que puedan los usuarios encontrar los servicios que necesiten, solicitarlos o agregarlos en respuesta a una necesidad.
- Visualizar propuestas de servicios a una necesidad para facilitar la adquisición de servicios.
- Calificar y comentar los servicios publicados en la aplicación para establecer que tan bueno es el servicio.
- Clasificar los servicios por categorías comprensibles para cualquier persona.
- Buscar los servicios publicados por cualquier persona para facilitar el acceso a los mismos.
- Conservar la información publicada en el servicio para que esté disponible y pueda ser consultada por cualquier usuario.
- Publicar requerimientos de servicios para que puedan ser respondidos por otros usuarios con servicios que puedan solventar el requerimiento.
- Poseer un espacio que cuente con información básica del usuario para que pueda ser manipulable y poder consultar acerca de su actividad realizada en la aplicación web.

El tiempo estimado de implementación será aproximadamente de seis meses correspondiente a seis *sprints* en el marco de trabajo *Scrum*. Además, en este punto se definió el orden de las historias de usuario que serán ejecutadas.

### **2.2.3.2. Lista de producto (Product Backlog)**

Los siguientes puntos corresponden al levantamiento de requerimientos necesarios para la aplicación web.

- Generar esquema de base de datos relacional.
- Realizar investigación de tecnologías existentes para aplicaciones *web*.
- Modificar *Readme* de presentación del proyecto en *github*.
- Normalizar carpetas de repositorio.
- Realizar bosquejos de funcionamiento de la aplicación *web*.
- Generar métodos de conexión a base de datos.
- Generar métodos de validación de datos antes de *crud* para la tabla usuario.
- Exponer métodos de conexión a base de datos a través de un api en *React*.

- Crear métodos de creación de tabla desde api para role.
- Crear métodos de creación de tabla desde api para servicio.
- Controlar acceso en base de estrategias a métodos http de la *api*.
- Levantar *api-backend* de conexión a base de datos.
- Crear pantalla de registro de usuario.
- Crear pantalla de inicio de sesión.
- Crear página principal.
- Crear pantalla de detalle de servicio.
- Crear pantalla de perfil de usuario.
- Crear pantalla para publicar un servicio.
- Crear *local storage* para almacenamiento de datos en navegador.
- Crear métodos de creación de tabla desde *api* para servicios (*service*).
- Crear métodos de creación de tabla desde *api* para role.
- Crear clase RQRP para procesar solicitudes y respuestas del *api* (*fetch*).
- Controlar acceso en base de estrategias a métodos *http* del *api*.
- Crear métodos de creación de tabla desde *api* para servicio.
- Crear módulo de imágenes.
- Agregar campos de servicio.
- Desplegar en *Heroku* la aplicación *web*.
- Clase de conexión para subir imágenes a *firebase*.
- Devolver valores necesarios para paginación.
- Agregar ciudad y categoría.
- Agregar calificación y promedios de *calificación* en cada servicio.
- Corregir edición de usuario.
- Añadir métodos de acceso a *calificación*.
- Agregar sección de comentarios.
- Agregar sección de solicitar servicios.
- Diseño de pruebas de caso de prueba a usuarios reales.
- Definir métodos para la medición de la accesibilidad de la aplicación.
- Definir el método de comparación de la accesibilidad de la aplicación con otras aplicaciones *web* similares.

## 2.3. Desarrollo de la aplicación

En esta sección se detallará las actividades realizadas en cada uno de los *sprints* con la finalidad de que el proceso de desarrollo de la aplicación *web* sea trazable y transparente.

Los *sprints* corresponden a las iteraciones de XP, donde cada uno de ellos fueron analizados y negociados con ayuda del cliente, priorizando historias de usuario y entregables.

### 2.3.1. Sprint 1

#### 2.3.1.1. Objetivos del Sprint

Generar el esquema de base de datos en base a los requerimientos de usuario y su api con sus respectivas conexiones en *React*.

#### 2.3.1.2. Lista de producto

- *Sprint 1* [Infraestructura] Realizar arquitectura de aplicación *web*.
- *Sprint 1* [Base de datos] Generar esquema de base de datos Relacional.
- *Sprint 1* [Requerimientos] Definir lista de requerimientos.
- *Sprint 1* [PoC] Tecnologías existentes.
- *Sprint 1* [Infraestructura] Modificar *Readme* de presentación del proyecto en *github*.
- *Sprint 1* [Infraestructura] Normalizar carpetas de repositorio.
- *Sprint 1* [Infraestructura] Realizar *bosquejos* de funcionamiento de la aplicación *web*.

#### 2.3.1.3. Planificación del Sprint

Este sprint tiene una duración de 30 días, correspondientes al mes de noviembre del 2021. Las historias que se ejecutarán en este sprint serán las siguientes:

- *Sprint 1* [Infraestructura] Realizar arquitectura de aplicación *web*
- *Sprint 1* [Base de datos] Generar esquema de base de datos Relacional.
- *Sprint 1* [Requerimientos] Definir lista de requerimientos.
- *Sprint 1* [PoC] Realizar investigación de tecnologías existentes para aplicaciones *web*.

- *Sprint 1* [Infraestructura] Modificar *Readme* de presentación del proyecto en *github*.
- *Sprint 1* [Infraestructura] Normalizar carpetas de repositorio.
- *Sprint 1* [Infraestructura] Realizar *bosquejos* de funcionamiento de la aplicación *web*.

## 2.3.1.4. Implementación

### 2.3.1.4.1. Realizar arquitectura de aplicación web

La figura 4 muestra la arquitectura de la aplicación web.

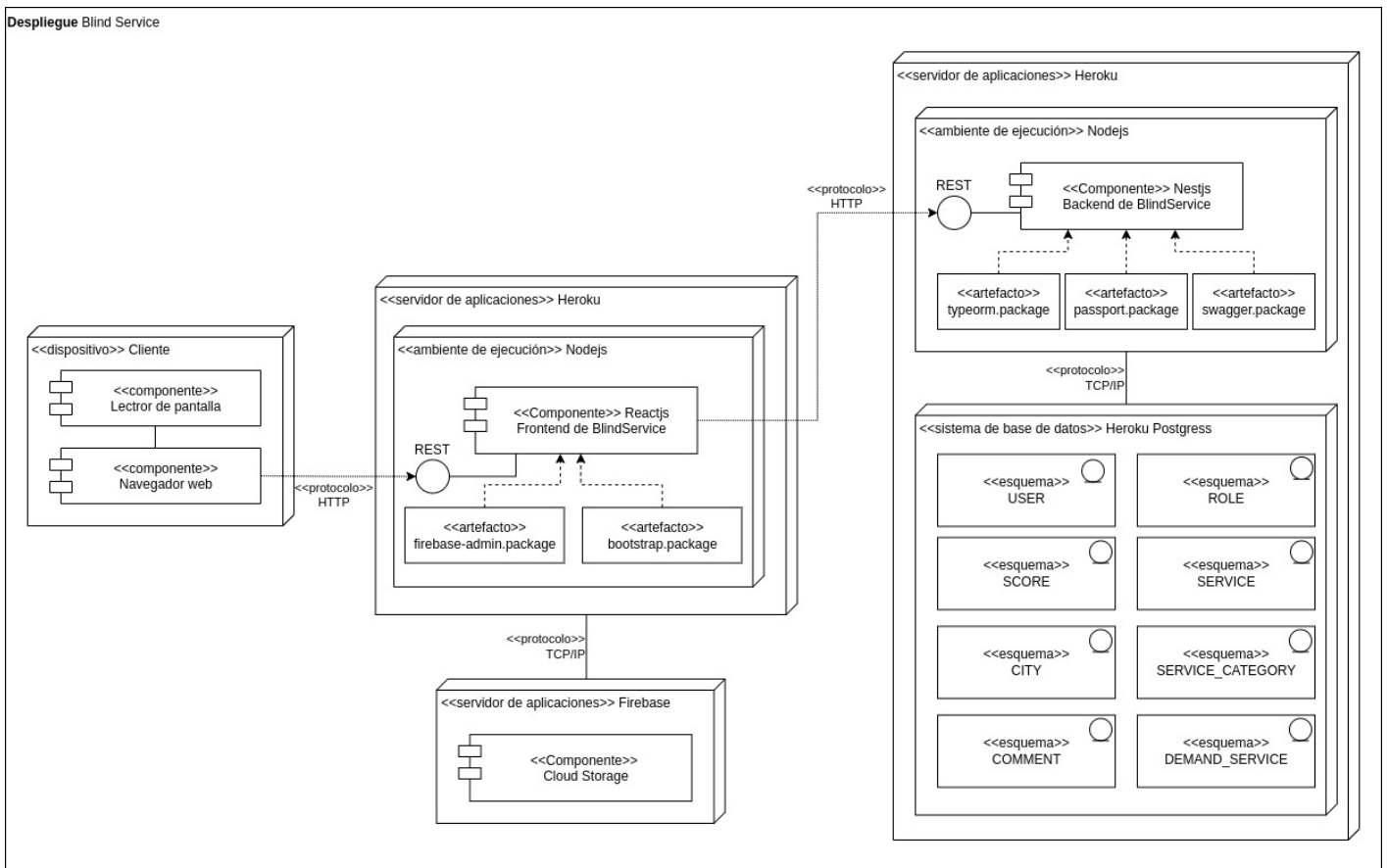


Figura 4. Arquitectura de la aplicación web

Se definieron las siguientes tecnologías para el desarrollo de la aplicación además del uso que se le dará a cada una:

- Toda la aplicación web será desplegada en *Heroku*.
- La aplicación web será desarrollada con *Node* y los *frameworks* de *NestJS* y *ReactJS*.
- La tecnología usada para la base de datos será *SQL*.
- El *framework* para acceso a la base de datos será un *ORM*.

- La seguridad podrá ser manejada por *Passportjs*.
- Como *framework* de estilo se utilizará *Bootstrap 5*.
- Para el almacenamiento de archivos se utilizará *Firebase*.
- Para el acceso del *backend* y *frontend* será *Heroku* como plataforma, *Swagger* como visualización de métodos y datos presentes en el *backend*, y *ReactJS* con su respectiva interfaz de usuario como *frontend*.
- El usuario final podrá acceder a la aplicación a través de cualquier dispositivo.

#### **2.3.1.4.2. Definir la lista de requerimientos**

A continuación, se describen los requerimientos obtenidos de reuniones con personas con discapacidad visual realizadas en el mes julio del 2021:

- La aplicación permitirá al usuario registrarse completando un formulario con los campos Nombre, Correo, Teléfono, Contraseña y Reingrese contraseña.
- La aplicación permitirá ingresar al usuario al formulario de registro haciendo clic en el botón Registrarse.
- La aplicación permitirá al usuario iniciar una sesión completando un formulario con los campos Correo y Contraseña.
- La aplicación permitirá al usuario ingresar al formulario de iniciar sesión haciendo clic en el botón Iniciar Sesión.
- La aplicación permitirá a un usuario publicar servicios completando un formulario con los siguientes campos:
  - Imagen
  - Nombre del servicio
  - Tipo de servicio
  - Precio
  - Provincia
  - Ciudad
  - Descripción del servicio
- Los campos Tipo de servicio, Provincia y Ciudad en el registro para publicar un servicio, se deben presentar en forma de lista mostrando las opciones disponibles al usuario.
- El formulario para publicar un servicio debe mostrar el teléfono ingresado por el usuario en el formulario de registro y no puede ser modificado.

- La aplicación permitirá filtrar servicios completando los campos Categoría, Ciudad, Calificación, Precio mínimo, Precio máximo y haciendo clic en el botón Aplicar.
- Los campos Categoría y Ciudad en la sección filtrar, deben ser presentados en forma de lista mostrando las opciones disponibles al usuario.
- La aplicación permitirá limpiar los filtros al usuario haciendo clic en el botón Limpiar.
- La aplicación permitirá buscar servicios al usuario ingresando información en la barra de búsqueda y haciendo clic en el botón Buscar.
- La aplicación permitirá al usuario comentar un servicio en la sección de comentarios ubicada al final de cada servicio, mostrando su nombre y la descripción del comentario.
- La aplicación permitirá al usuario calificar un servicio mediante un sistema de estrellas.
- La aplicación permitirá al usuario solicitar un servicio mediante un formulario ingresando el tipo de servicio y una descripción.
- La aplicación permitirá al usuario observar otras solicitudes de servicio realizadas por otros usuarios.
- La aplicación deberá tener una sección de paginación en la página principal que permitirá mostrar un máximo de 12 servicios por página.
- La aplicación deberá mostrar una breve descripción de los servicios publicados que conste de la imagen, título del servicio, Precio, Categoría, Descripción, Fecha de creación y Calificación.
- La aplicación permitirá ingresar al usuario a la información completa de los servicios haciendo clic en los servicios mostrados en la página principal.
- La aplicación contará con un botón Ver usuario que permitirá al usuario acceder a su perfil y servicios publicados.
- El usuario podrá modificar los campos Nombre, Correo y teléfono al hacer clic en el botón Editar perfil.
- El usuario podrá modificar su contraseña al hacer clic en el botón Editar contraseña.
- El usuario podrá modificar un servicio publicado al hacer clic en el icono en forma de lápiz situado en cada uno de los servicios publicados.



- La aplicación permitirá cerrar sesión al usuario al hacer clic en el botón cerrar sesión.
- Agregar calificación y promedios de calificación en cada servicio.
- Sección para solicitar servicios.
- Corregir edición de usuario.
- Añadir métodos de acceso a calificación.
- Agregar sección de comentarios.

### 2.3.1.4.3. Generar esquema de base de datos Relacional.

La figura 5 corresponde al esquema de la base de datos:

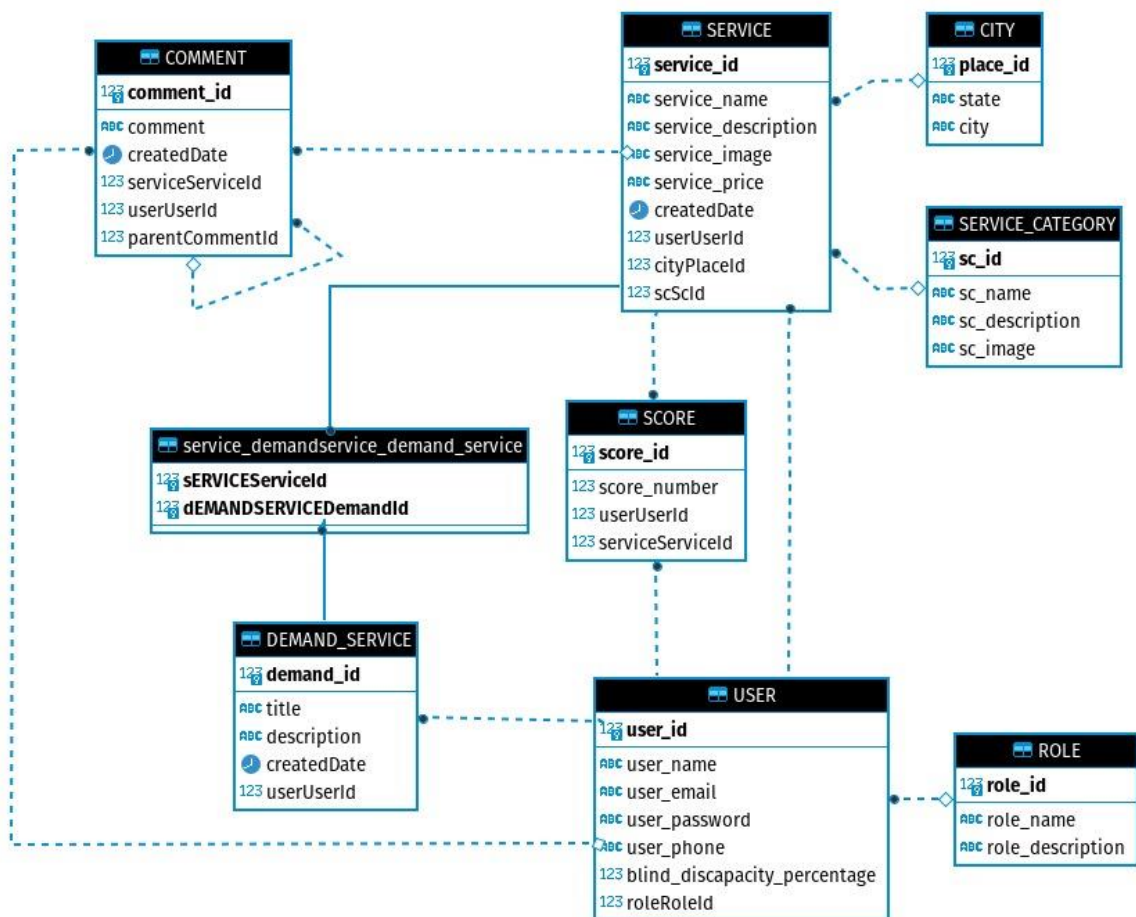


Figura 5. Esquema de base de datos

En la figura 5:

- La tabla USER almacenará los datos del usuario.
- La tabla ROLE almacenará los roles que puede tener el usuario.

- La tabla SCORE almacenará las calificaciones que los usuarios den a los servicios.
- La tabla SERVICE almacenará los datos de los servicios publicados por los usuarios.
- La tabla COMMENT almacenará los comentarios realizados por los usuarios a los servicios.
- La tabla SERVICE\_CATEGORY almacenará las categorías disponibles para los servicios.
- La tabla CITY almacenará la provincia y ciudad disponibles.
- La tabla DEMAND\_SERVICE almacenará los servicios demandados por los usuarios.

#### 2.3.1.4.4. Realizar bosquejos de funcionamiento de la aplicación web.

Los bosquejos fueron realizados en *Figma* y corresponden a las siguientes figuras:

La figura 6 muestra el bosquejo de la página principal de la aplicación.

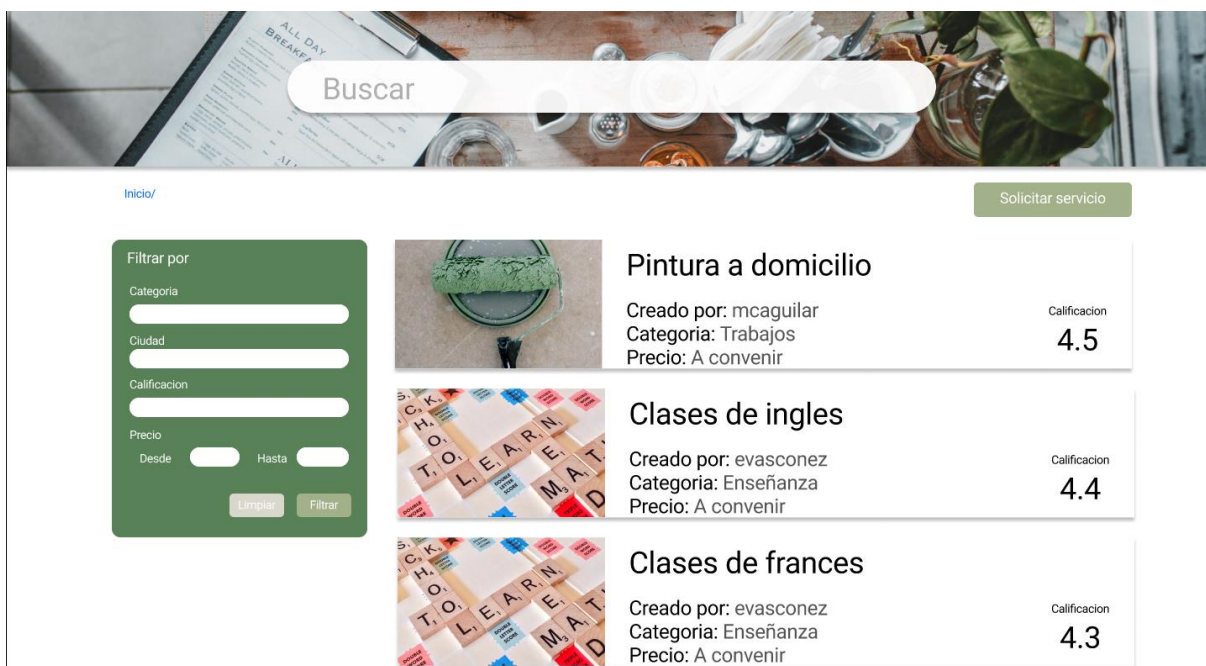


Figura 6. Página principal

La figura 7 muestra el bosquejo del registro de usuario.

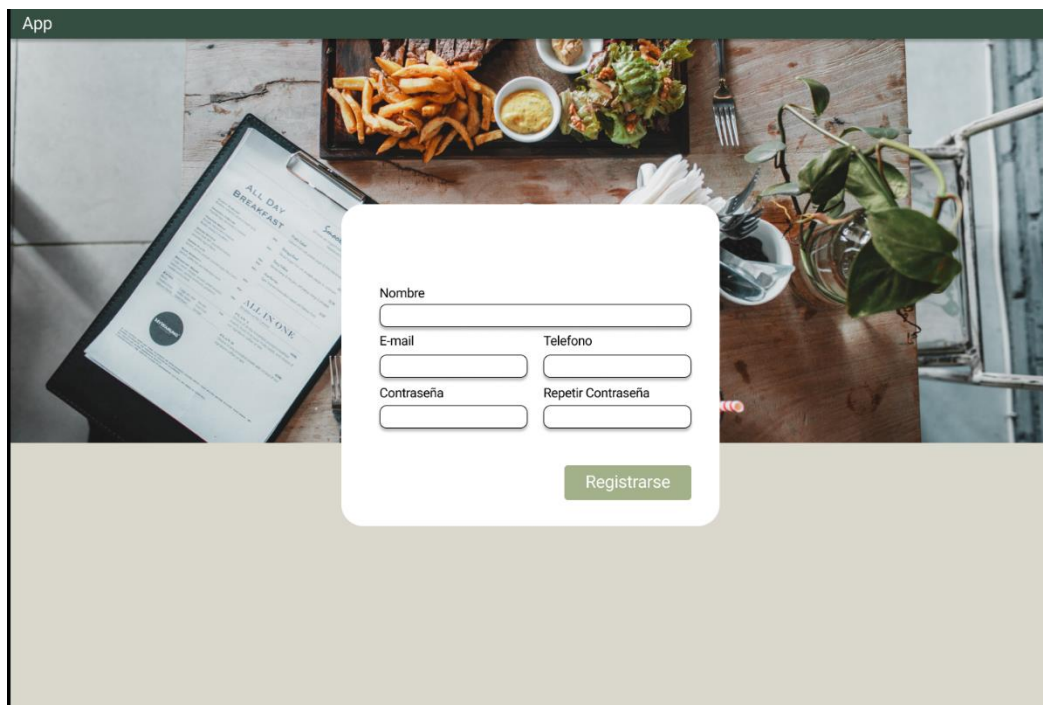


Figura 7. Registro de usuario

La figura 8 muestra el bosquejo del formulario de inicio de sesión de usuario.

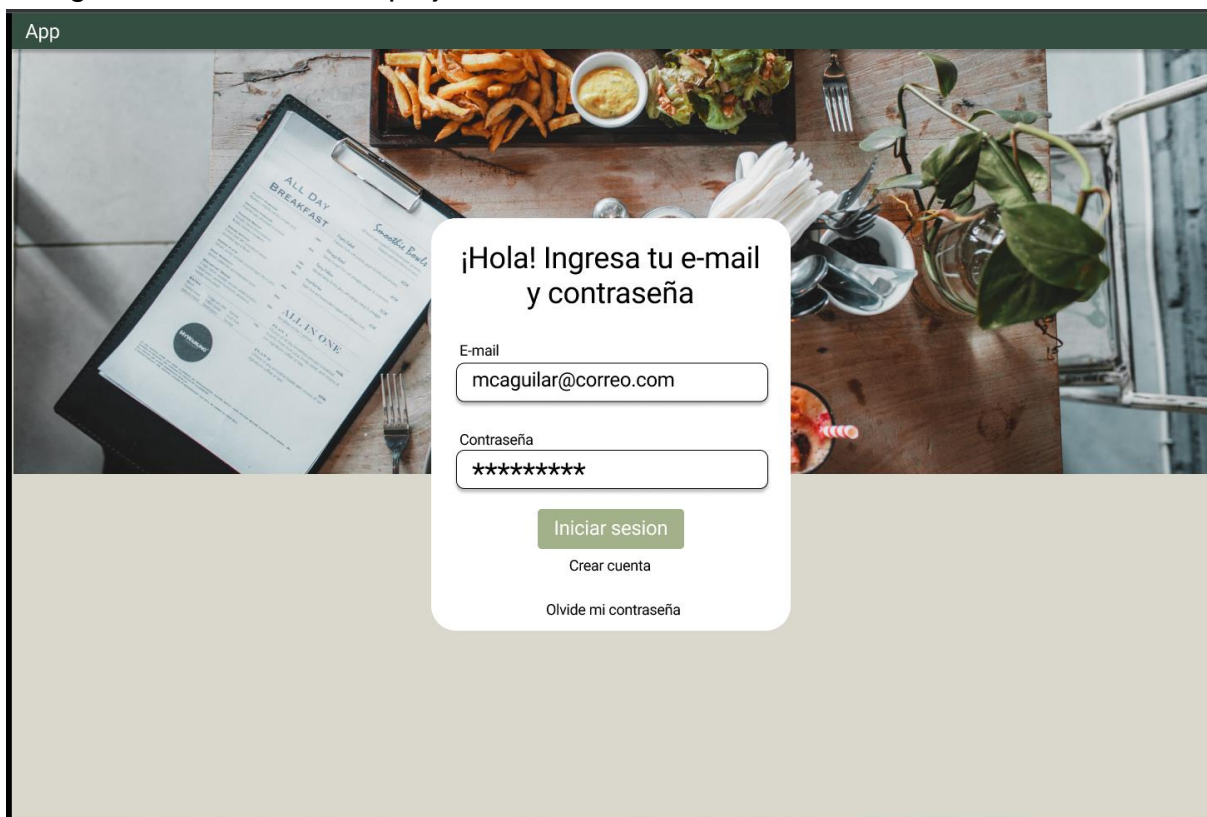


Figura 8. Inicio de sesión

La figura 9 muestra el bosquejo de la página principal con un usuario

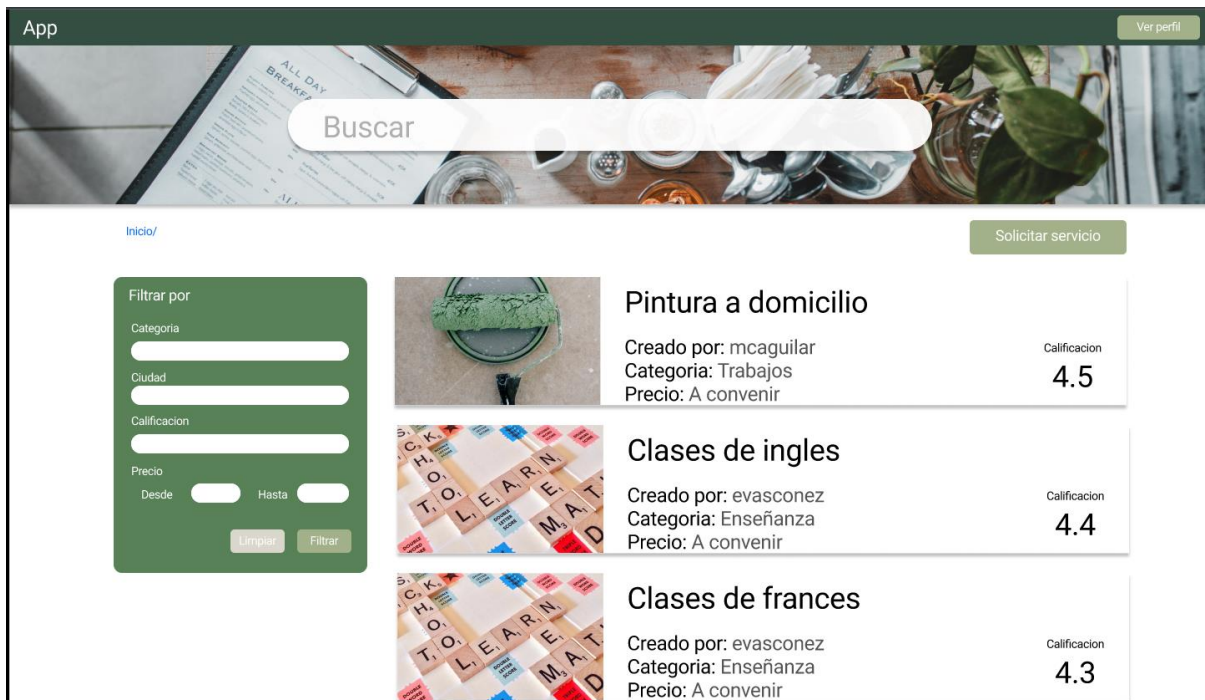


Figura 9. Página principal con usuario registrado

La figura 10 muestra el bosquejo del formulario a llenar para publicar un servicio.

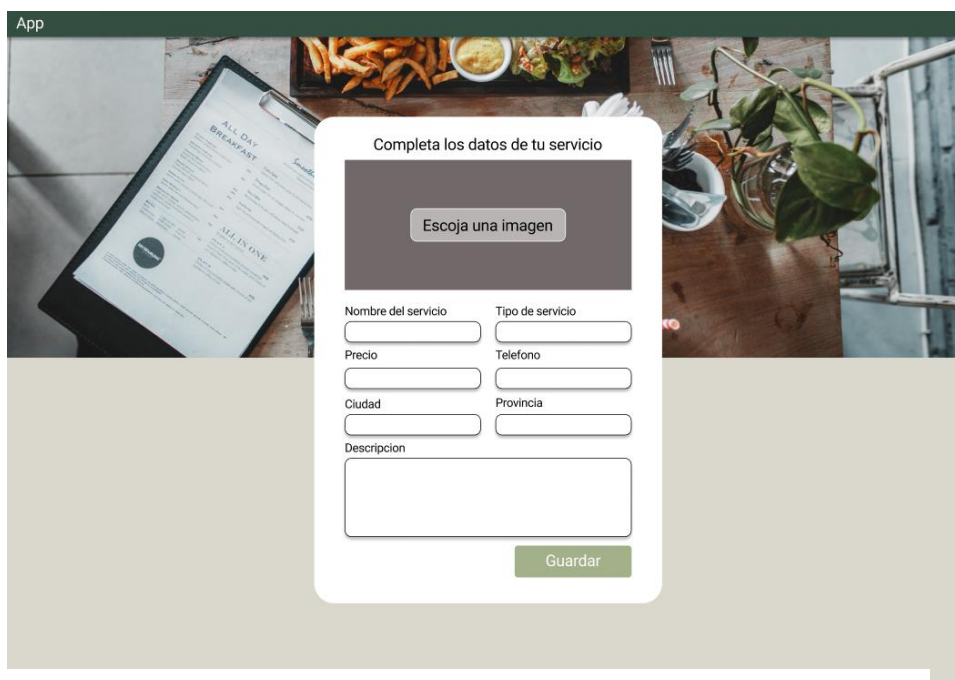


Figura 10. Formulario para publicar servicio

La figura 11 muestra el bosquejo de la página para poder observar un servicio disponible.

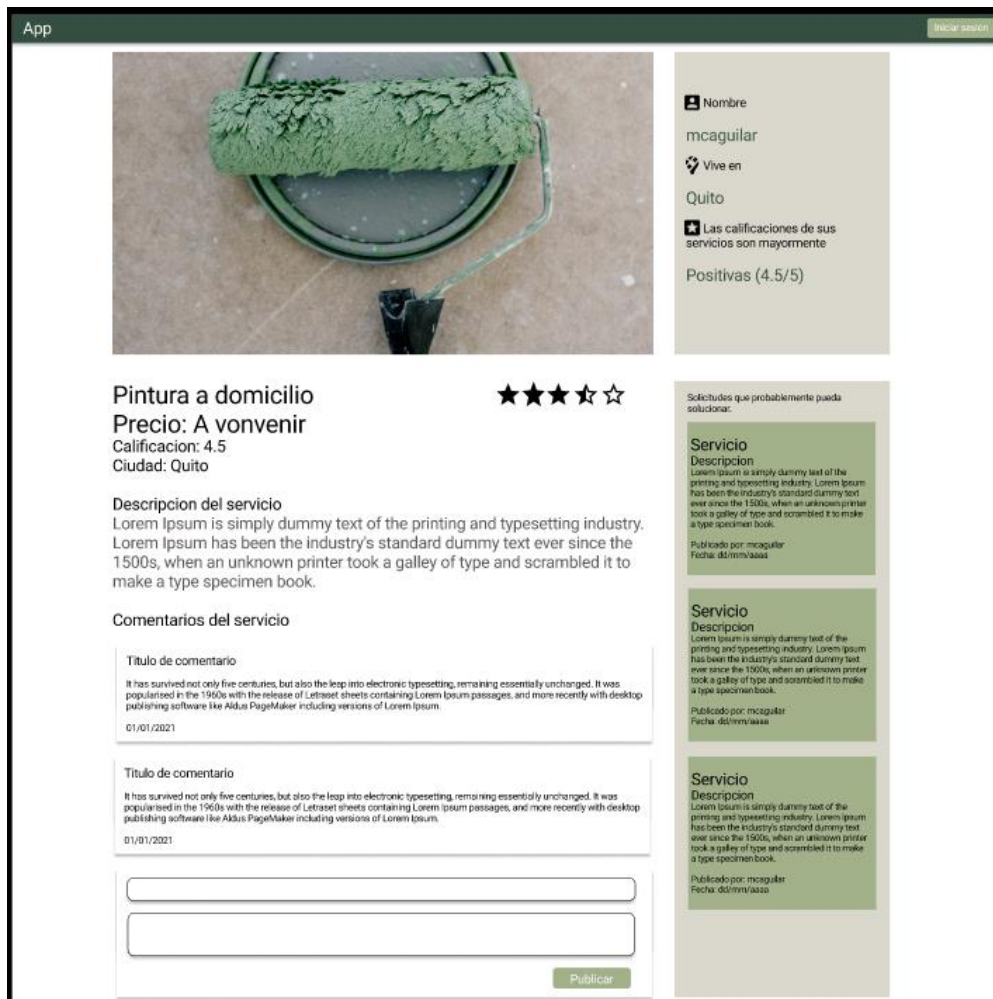


Figura 11. Detalle servicio

La figura 12 muestra el bosquejo del perfil del usuario donde se podrá actualizar los datos.

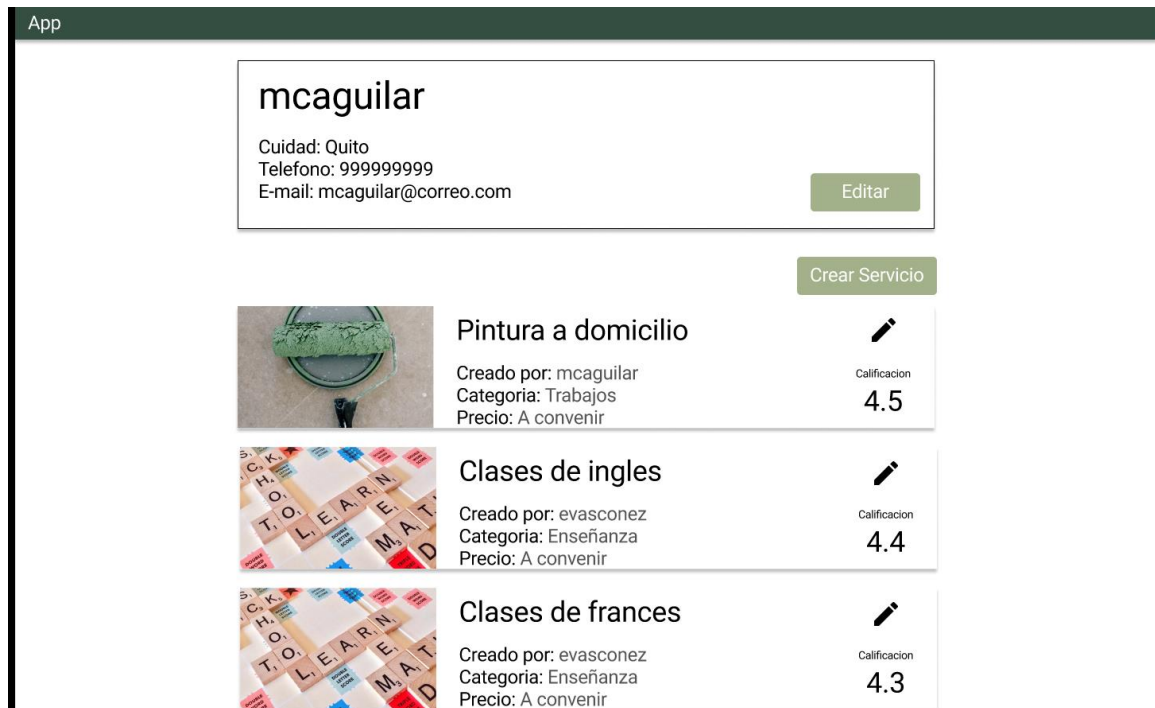


Figura 12. Perfil del usuario

La figura 13 muestra el bosquejo del formulario para la actualización de un servicio.

Edita los datos de tu servicio

Escoja una imagen

Nombre del servicio

Tipo de servicio

Precio

Telefono

Provincia

Descripcion

Guardar

Figura 13. Actualización de servicio

La figura 14 muestra el bosquejo para solicitar y observar servicios.

The image shows a web application interface with a dark green header. On the left, it says 'App' and on the right, 'Inicio/Logout'. Below the header is a search bar with the text 'Buscar'. The main content area is divided into two sections: 'Solicitar servicio' and 'Servicios solicitados'. The 'Solicitar servicio' section contains a form with three input fields: 'Nombre del servicio', 'Descripcion', and 'Categoria', followed by a 'Publicar' button. The 'Servicios solicitados' section displays two identical service cards. Each card has a title 'Servicio', a 'Descripcion' field, a paragraph of text, and a 'Publicado por: mcagalar' field. At the bottom of each card are two buttons: 'Ver propuestas' and 'Agregar this servicio'.

Figura 14. Solicitar servicios

La figura 15 muestra el bosquejo para agregar servicios.

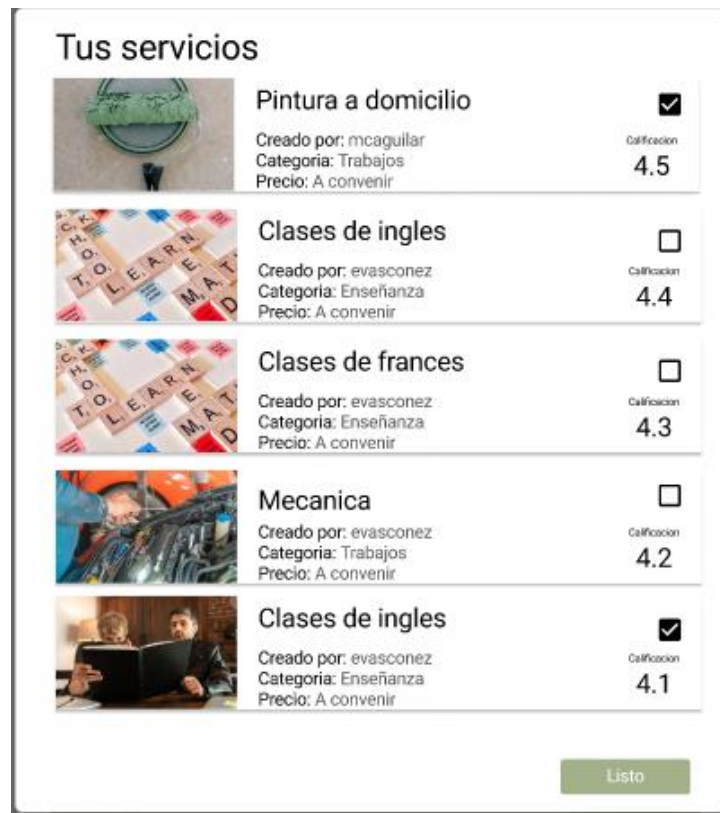


Figura 15. Agregar servicios

La figura 16 muestra el bosquejo para ver propuestas de servicios.

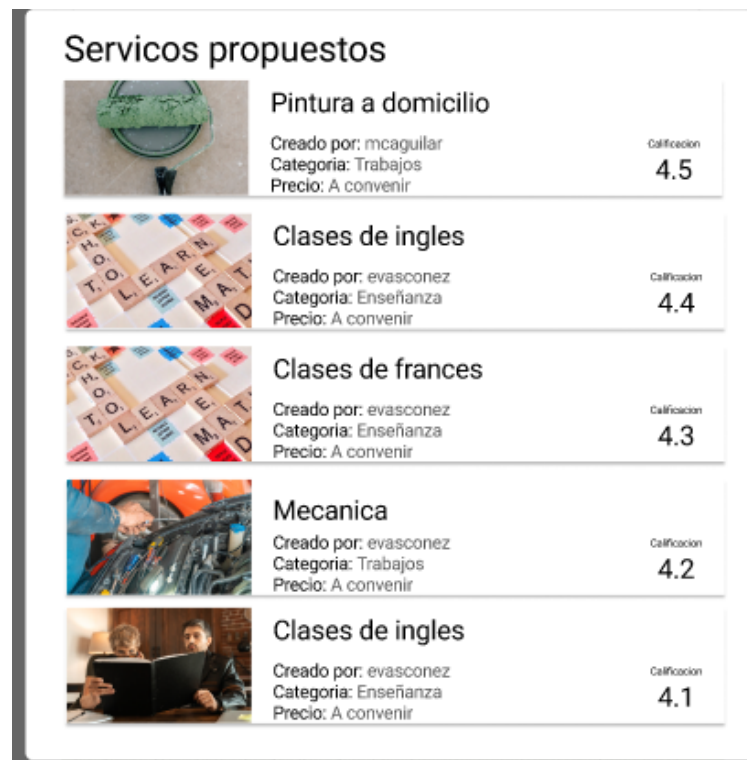


Figura 16. Ver propuestas



#### 2.3.1.4.5. Normalizar carpetas de repositorio

La figura 17 muestra la estructura básica de archivos *React* para el proyecto, esta se realizó utilizando el comando `npx create-app-react`.

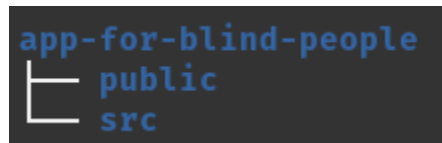


Figura 17. Estructura básica de archivos React

La figura 18 muestra la estructura de carpetas general que se usará para el presente proyecto:



Figura 18. Estructura de carpetas general

En la figura 18:

- *App-for-blind-people* corresponde a la aplicación desarrollada en *React*.
- *Database* corresponde a los scripts de base de datos que serán usados en el proyecto.
- *Server* corresponde al código de *backend* que se usará para realizar las respectivas conexiones entre el *front* y la base de datos.

#### 2.3.1.4.6. Realizar conexión a base de datos (odbc)

La figura 19 muestra el código que se utilizó para establecer la conexión a la base de datos usando el driver *PostgreSQL* disponible para *TypeORM*, se usó *dotenv* para establecer variables de entorno en el ambiente, a través de un archivo nombrado *.env*, Además se muestra el código utilizado para conectar a la base de datos.

```
TypeOrmModule.forRoot({
  name: 'default',
  type: 'postgres',
  dropSchema: false,
  autoLoadEntities: true,
  synchronize: true,
  logging: false,
  url: process.env.DATABASE_URL,
  ssl: true,
  extra: {
    ssl: {
      rejectUnauthorized: false,
    },
  },
},
),
```

Figura 19. Conexión a base de datos

### 2.3.1.5. Revisión de *Sprint*.

El *sprint* número uno, realizado en el mes de noviembre del 2021 se completó satisfactoriamente con un cien por ciento de cumplimiento en historias de usuario.

De este *sprint* se obtuvo lo siguiente:

- Se definió la lista de requerimientos con los cuales se tomarán en cuenta para el desarrollo de futuros *sprints*.
- Se seleccionaron las tecnologías con las cuales se desarrollará la aplicación.
- Se posee una arquitectura sobre la cual se trabajará en el desarrollo de la aplicación web.
- Se realizó un esquema de base de datos relacional que solventa la necesidad de almacenamiento de datos de la aplicación a realizarse.
- Se realizó un bosquejo del funcionamiento de la aplicación a través de mockups realizados en Figma para definir todas las características de la aplicación.

De este *sprint* se tienen los siguientes entregables:

- Arquitectura de despliegue de la aplicación
- Esquema de la base de datos
- Bosquejo de las pantallas de la aplicación con enlaces funcionales

## 2.3.2. *Sprint 2*

### 2.3.2.1. **Objetivos del *Sprint***

Generar la estructura de archivos *React* para la elaboración del proyecto.

### 2.3.2.2. **Lista de producto**

- *Sprint 2 [Backend]* Generar métodos de conexión a base de datos.
- *Sprint 2 [Backend]* Generar métodos de validación de datos antes de *crud* para la tabla usuario.
- *Sprint 2 [Backend]* Exponer métodos de conexión a base de datos a través de un *api* en *react*.
- *Sprint 2 [Backend]* Levantar *api-backend* de conexión a base de datos.
- *Sprint 2 [Frontend]* Crear pantalla de registro de usuario.
- *Sprint 2 [Frontend]* Crear pantalla de inicio de sesión.
- *Sprint 2 [Frontend]* Crear página principal.
- *Sprint 2 [Frontend]* Crear pantalla de detalle de servicio.
- *Sprint 2 [Frontend]* Crear pantalla de perfil de usuario.
- *Sprint 2 [Frontend]* Crear local storage para almacenamiento de datos en navegador.

### 2.3.2.3. **Planificación del *Sprint***

Este *sprint* tiene una duración de 31 días, correspondientes al mes de diciembre del 2021. Las historias que se ejecutarán en este *sprint* serán las siguientes:

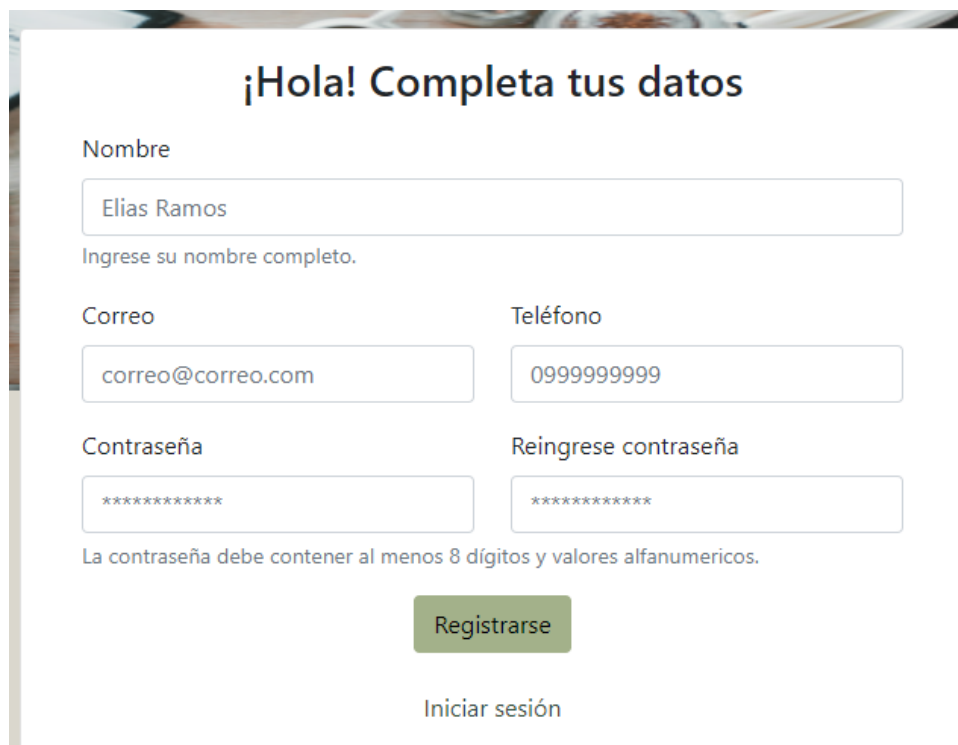
- *Sprint 2 [Backend]* Generar métodos de conexión a base de datos.
- *Sprint 2 [Backend]* Generar métodos de validación de datos antes de *crud* para la tabla usuario.
- *Sprint 2 [Backend]* Exponer métodos de conexión a base de datos a través de un *api* en *react*.
- *Sprint 2 [Backend]* Levantar *api-backend* de conexión a base de datos.
- *Sprint 2 [Frontend]* Crear pantalla de registro de usuario.
- *Sprint 2 [Frontend]* Crear pantalla de inicio de sesión.
- *Sprint 2 [Frontend]* Crear página principal.
- *Sprint 2 [Frontend]* Crear pantalla de detalle de servicio.

- *Sprint 2 [Frontend]* Crear pantalla de perfil de usuario.
- *Sprint 2 [Frontend]* Crear local storage para almacenamiento de datos en navegador.

#### 2.3.2.4. Implementación

##### 2.3.2.4.1. Crear pantalla de registro de usuario

Para la creación de la pantalla de registro de usuario se utilizó la biblioteca *Bootstrap* 5 para facilitar su desarrollo, la figura 20 muestra el aspecto del registro de usuario.



¡Hola! Completa tus datos

Nombre  
Elias Ramos  
Ingrese su nombre completo.

Correo  
correo@correo.com

Teléfono  
0999999999

Contraseña  
\*\*\*\*\*

Reingrese contraseña  
\*\*\*\*\*

La contraseña debe contener al menos 8 dígitos y valores alfanumericos.

Registrarse

Iniciar sesión

Figura 20. Pantalla de registro de usuario

La pantalla de registro cuenta con la validación de errores antes de ser enviados al *backend* para su registro en la base de datos, la figura 21 muestra la validación de errores cuando los campos están vacíos.

## ¡Hola! Completa tus datos

Nombre

Ingrese su nombre completo

Correo  Teléfono

Contraseña  Reingrese contraseña

La contraseña debe contener al menos 8 dígitos y valores alfanumericos.

[Registrarse](#)

[Iniciar sesión](#)

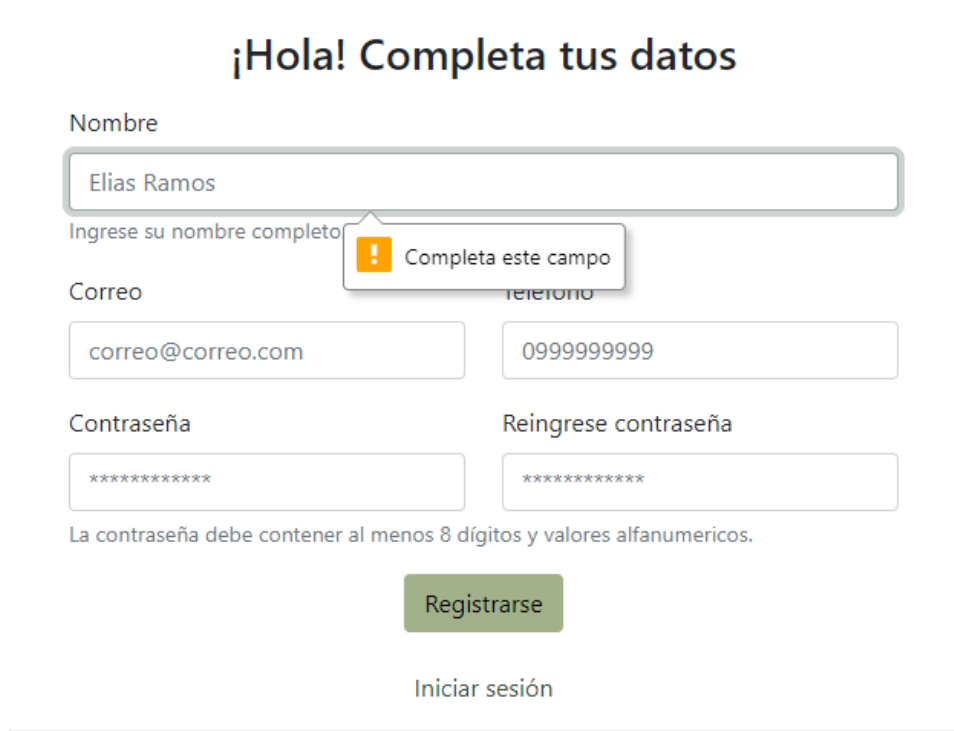


Figura 21. Validación de errores en pantalla de registro

#### 2.3.2.4.2. Crear pantalla de inicio de sesión de usuario

Para la creación de la pantalla de inicio de sesión de usuario se utilizó la biblioteca *Bootstrap 5* para facilitar su desarrollo, la figura 22 muestra el aspecto del inicio de sesión de usuario.

## ¡Hola! Ingresa tu correo y contraseña

Correo

Contraseña

La contraseña debe contener al menos 8 digitos y valores alfanumericos.

[Iniciar sesión](#)

[Crear cuenta](#)

[Olvide mi contraseña](#)



Figura 22. Pantalla de inicio de sesión de usuario



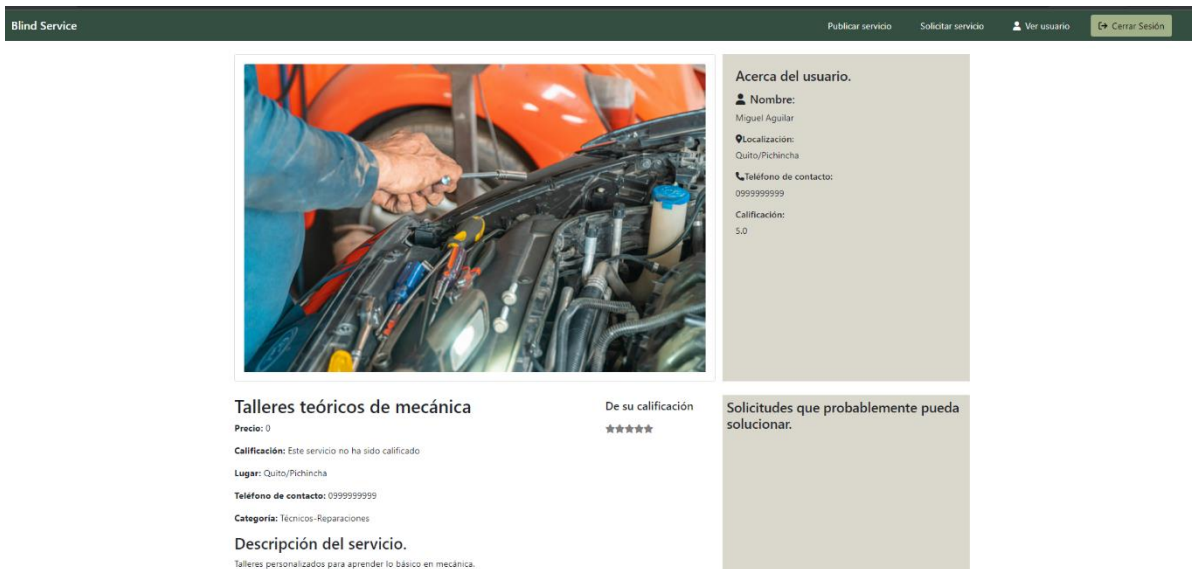


Figura 25. Página de detalle de servicio

### 2.3.2.4.5. Crear pantalla de perfil de usuario

Para la creación de la pantalla de perfil de usuario se utilizó la biblioteca *Bootstrap 5* para facilitar su desarrollo, la figura 26 muestra el aspecto de la página perfil de usuario.

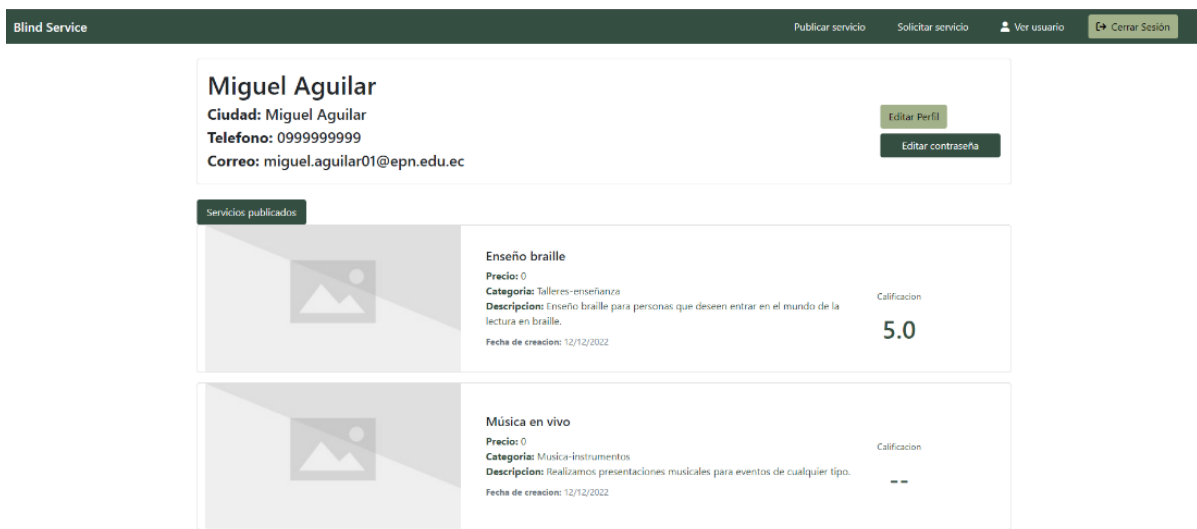
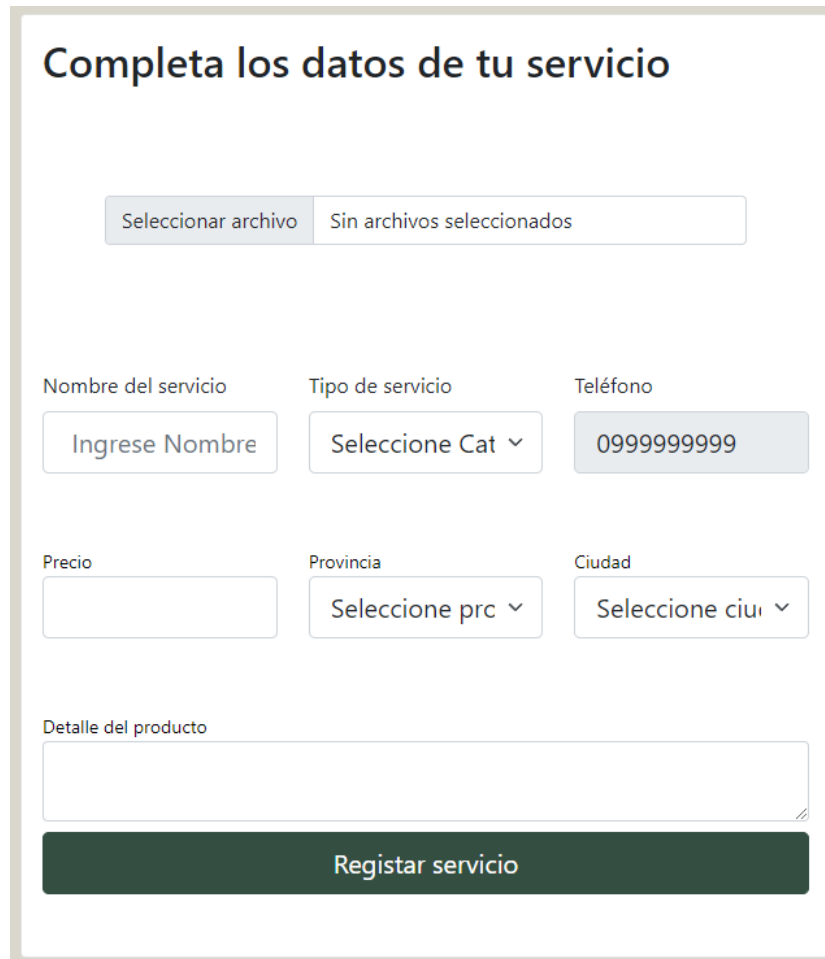


Figura 26. Pantalla de perfil de usuario

#### 2.3.2.4.6. Crear pantalla para publicar un servicio

Para la creación de la pantalla para publicar un servicio se utilizó la biblioteca *Bootstrap* 5 para facilitar su desarrollo, la figura 27 muestra el aspecto de la página para publicar un servicio.



The screenshot shows a web form titled "Completa los datos de tu servicio". At the top, there is a file selection area with a button labeled "Seleccionar archivo" and a text box containing "Sin archivos seleccionados". Below this, the form is organized into several sections:

- Nombre del servicio:** A text input field with the placeholder "Ingrese Nombre".
- Tipo de servicio:** A dropdown menu with the placeholder "Seleccione Cat".
- Teléfono:** A text input field with the placeholder "0999999999".
- Precio:** A text input field.
- Provincia:** A dropdown menu with the placeholder "Seleccione prc".
- Ciudad:** A dropdown menu with the placeholder "Seleccione ciui".
- Detalle del producto:** A large text area for entering product details.

At the bottom of the form is a prominent dark green button labeled "Registrar servicio".

Figura 27. Pantalla para publicar un servicio

#### 2.3.2.4.7. Crear *localStorage* para almacenamiento de datos en navegador.

Para mantener la sesión del usuario se utilizó la propiedad *localStorage*, que mantiene un área de almacenamiento separada para cada origen que se mantiene incluso cuando el navegador se cierre y se abra. La figura 28 muestra la sección de código utilizada para realizar el almacenamiento del usuario.



```

.then(
  (result) => {
    localStorage.setItem("user-info", JSON.stringify(result));
    this.setState("error", "status", false);
    window.location.href = "/";
  }
)

```

Figura 28. *localStorage*

La figura 29 muestra el almacenamiento de un usuario de prueba en el *localStorage* (Almacenamiento Local) utilizando el navegador *Edge*.

Key	Value
user-info	{"user_id":16,"user_name":"Miguel Aguilar","u...

Figura 29. *localStorage* en funcionamiento

### 2.3.2.5. Revisión de *Sprint*

El *sprint* número dos, realizado en el mes de diciembre del 2021, se completó satisfactoriamente con un cien por ciento de cumplimiento en historias de usuario.

- Se generaron los métodos de conexión a base de datos.
- Se generaron los métodos de validación para la tabla de usuario.
- Se creó el proyecto en *Heroku* en *backend* con su respectiva conexión a base de datos respectiva.
- Se crearon la pantalla de registro de usuario, inicio de sesión, página principal, página de detalle de servicio y página de perfil de usuario en *frontend*.
- Se añadió el funcionamiento de almacenamiento por medio de *local storage* para los datos de inicio de sesión del usuario.

De este *sprint* se tiene el siguiente entregable:

- Las pantallas de registro de usuario, inicio de sesión, página principal, página de detalle de servicio y página de perfil de usuario con *mock* de datos; todas las pantallas cuentan con accesibilidad web.

### 2.3.3. *Sprint* 3

#### 2.3.3.1. Objetivos del *Sprint*

- Validar la posibilidad de manejar errores en clase global.
- Desarrollar métodos de acceso a las entidades de servicio (*service*) y rol (*role*).
- Controlar acceso al *backend* en base a estrategias de autenticación.
- Crear métodos de carga y lectura de imágenes.

#### 2.3.3.2. Lista de producto

- *Sprint* 3 [*Backend*] Crear métodos de creación de tabla desde api para servicios (*service*).
- *Sprint* 3 [*Backend*] Crear métodos de creación de tabla desde api para role.
- *Sprint* 3 [*Frontend*] Crear clase *RQRP* para procesar solicitudes y respuestas del api (*fetch*).
- *Sprint* 3 [*Backend*] Controlar acceso en base de estrategias a métodos *http* del api.
- *Sprint* 3 [*Backend*] Crear métodos de creación de tabla desde api para servicio.
- *Sprint* 3 [*Backend*] Crear módulo de imágenes.
- *Sprint* 3 [*Frontend*] Validar la posibilidad de manejar errores en clase global.

#### 2.3.3.3. Planificación del *Sprint*

Este *sprint* tiene una duración de 31 días, correspondientes al mes de enero del 2022. Las historias que se ejecutarán en este *sprint* serán las siguientes:

- *Sprint* 3 [*Backend*] Crear métodos de creación de tabla desde api para servicios (*service*)
- *Sprint* 3 [*Backend*] Crear métodos de creación de tabla desde api para role.
- *Sprint* 3 [*Frontend*] Crear clase *RQRP* para procesar solicitudes y respuestas del api (*fetch*).
- *Sprint* 3 [*Backend*] Controlar acceso en base de estrategias a métodos *http* del api.
- *Sprint* 3 [*Backend*] Crear métodos de creación de tabla desde api para servicio.
- *Sprint* 3 [*Backend*] Crear módulo de imágenes.
- *Sprint* 3 [*Frontend*] Validar la posibilidad de manejar errores en clase global.

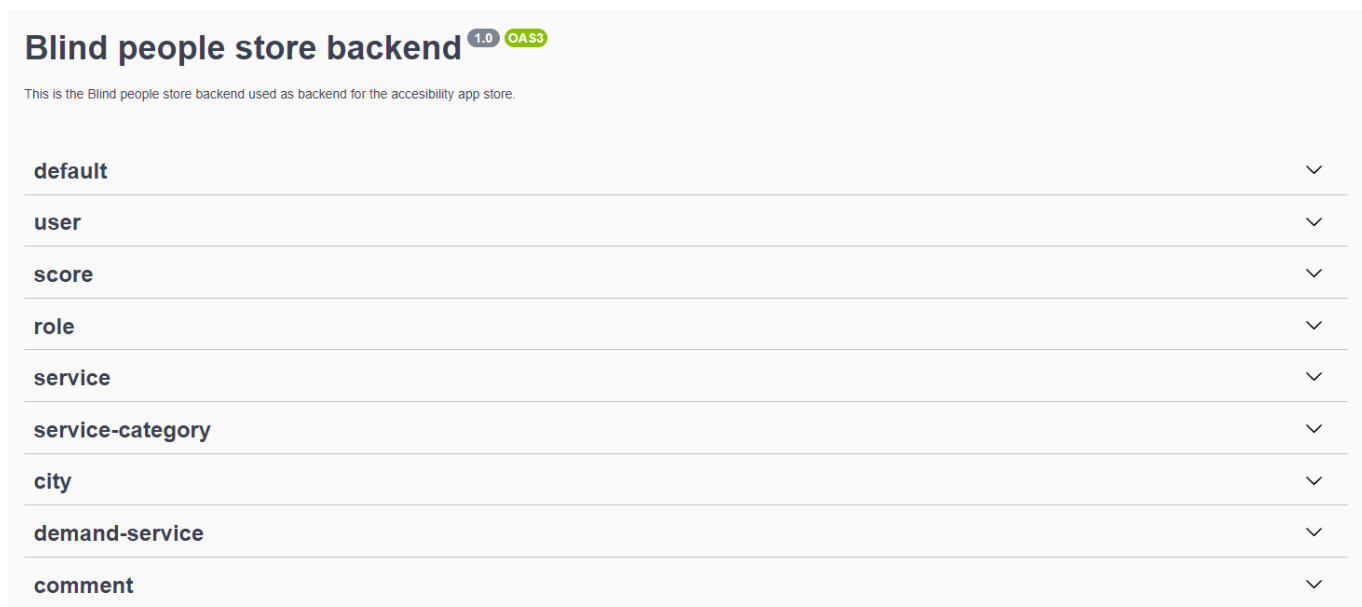
## 2.3.3.4. Implementación

### 2.3.3.4.1. Crear clase *RQRP*

Se creó una clase para *request - response* o traducido del inglés solicitud – respuesta la cual responde a la necesidad presente en el proyecto, manejar las solicitudes al *backend*.

### 2.3.3.4.2. Controlar acceso en base de estrategias a métodos *http* del *api*

En base al estándar de *OpenAPI* se crearon los siguientes recursos: *user*, *score*, *role*, *service*, *service-category*, *city*, *demand-service*, *comment*, como se muestra en la figura 30.

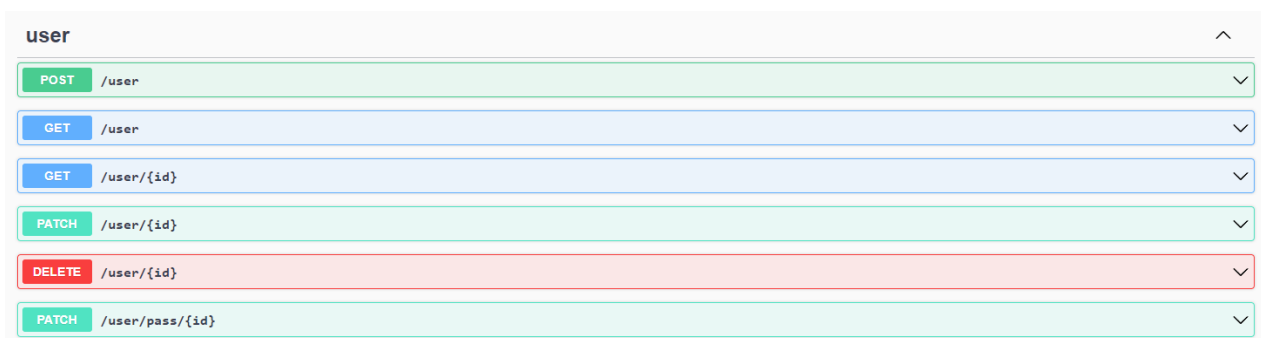


The screenshot shows the OpenAPI specification for 'Blind people store backend' (version 1.0, OAS3). It lists the following resources:

Resource	Expand
default	▼
user	▼
score	▼
role	▼
service	▼
service-category	▼
city	▼
demand-service	▼
comment	▼

Figura 30. Recursos de *backend*

Además, para cada recurso se crearon métodos similares a los que se muestran en la figura 31.



The screenshot shows the OpenAPI specification for the 'user' resource with the following methods:

Method	Endpoint	Expand
POST	/user	▼
GET	/user	▼
GET	/user/{id}	▼
PATCH	/user/{id}	▼
DELETE	/user/{id}	▼
PATCH	/user/pass/{id}	▼

Figura 31. Métodos del *backend*

### 2.3.3.4.3. Crear módulo de imágenes

El módulo de imágenes corresponde a una implementación de *firebase*, para ello se creó un proyecto como se muestra en la figura 32 y se lo conectó a través del SDK disponible para *JavaScript*.

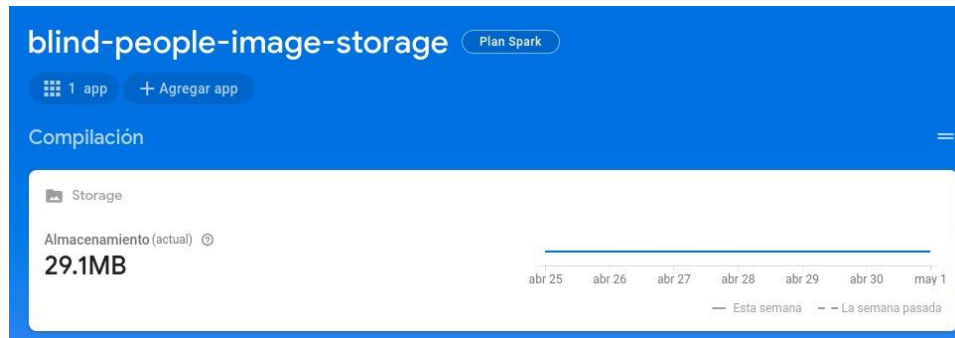


Figura 32. Módulo de imágenes

### 2.3.3.5. Revisión de *Sprint*

El *sprint* número tres, realizado en el mes de septiembre, se completó satisfactoriamente con un cien por ciento de cumplimiento en historias de usuario.

De este *sprint* se obtuvo lo siguiente:

- Se generaron las entidades necesarias para la conexión con *TypeORM* a la base de datos para las tablas *role* y *servicio*.
- Se generaron métodos de acceso *http* los cuales fueron publicados en *Swagger* en *backend*.
- Se creó módulo para manejo de imágenes en el *backend*.
- Se realizó la validación de manejo de errores en de forma global en el *frontend*.
- Se entró que el campo teléfono no funcionaba correctamente en el *frontend*.

De este *sprint* se tienen los siguientes entregables:

- Las pantallas de registro de usuario, inicio de sesión, página principal, página de detalle de servicio y página de perfil de usuario con las conexiones establecidas.

### 2.3.4. *Sprint* 4

En este *sprint* se empezará a realizar despliegues en *Heroku* ya que con el cliente se pudo validar que la aplicación ya puede encontrarse en producción para puede ser utiliza por el público general

#### 2.3.4.1. Objetivos del *Sprint*

- Realizar correcciones en la aplicación *web*.
- Desplegar primera versión de la aplicación *web* en *Heroku*.

#### 2.3.4.2. Lista de producto

- *Sprint* 4 [*Frontend*] Corregir error del campo teléfono al agregarlo al campo de la etiqueta HTML *value*.
- *Sprint* 4 [*Frontend*] Agregar campos de servicio.
- *Sprint* 4 [*Frontend*] Cambiar barra de búsqueda en el *dom*.
- *Sprint* 4 [*Frontend*] Desplegar en *Heroku* la aplicación *web*.
- *Sprint* 4 [*Frontend*] Clase de conexión para subir imágenes a *firebase*
- *Sprint* 4 [*Backend*] Devolver valores necesarios para paginación
- *Sprint* 4 [*Backend*] Agregar ciudad y categoría

#### 2.3.4.3. Planificación del *Sprint*

Este *sprint* tiene una duración de 31 días, correspondientes al mes de octubre. Las historias que se ejecutarán en este *sprint* serán las siguientes:

- *Sprint* 4 [*Frontend*] Corregir error del campo teléfono al agregarlo al campo de la etiqueta HTML *value*.
- *Sprint* 4 [*Frontend*] Agregar campos de servicio.
- *Sprint* 4 [*Frontend*] Cambiar barra de búsqueda en el *dom*.
- *Sprint* 4 [*Frontend*] Desplegar en *Heroku* la aplicación *web*.
- *Sprint* 4 [*Frontend*] Clase de conexión para subir imágenes a *firebase*
- *Sprint* 4 [*Backend*] Devolver valores necesarios para paginación
- *Sprint* 4 [*Backend*] Agregar ciudad y categoría

#### 2.3.4.4. Implementación

##### 2.3.4.4.1. Agregar campos de servicio

A falta de implementación se agregaron los campos: Imagen, Nombre del servicio, Precio y Detalle del producto.

### 2.3.4.4.2. Cambiar barra de búsqueda en el *dom*

Debido a conversaciones se decidió estilizar la barra de búsqueda como se muestra en la figura 33.



Figura 33. Barra de búsqueda

### 2.3.4.4.3. Desplegar en *Heroku* la aplicación *web*

Se decidió desplegar la aplicación web en *Heroku* como se muestra la figura 34, para ello es necesario que todos los cambios sean expuestos a través de la rama *master* que en el caso del proyecto se manejaba a través de la rama *develop* que hacía una unión a la rama *master*.

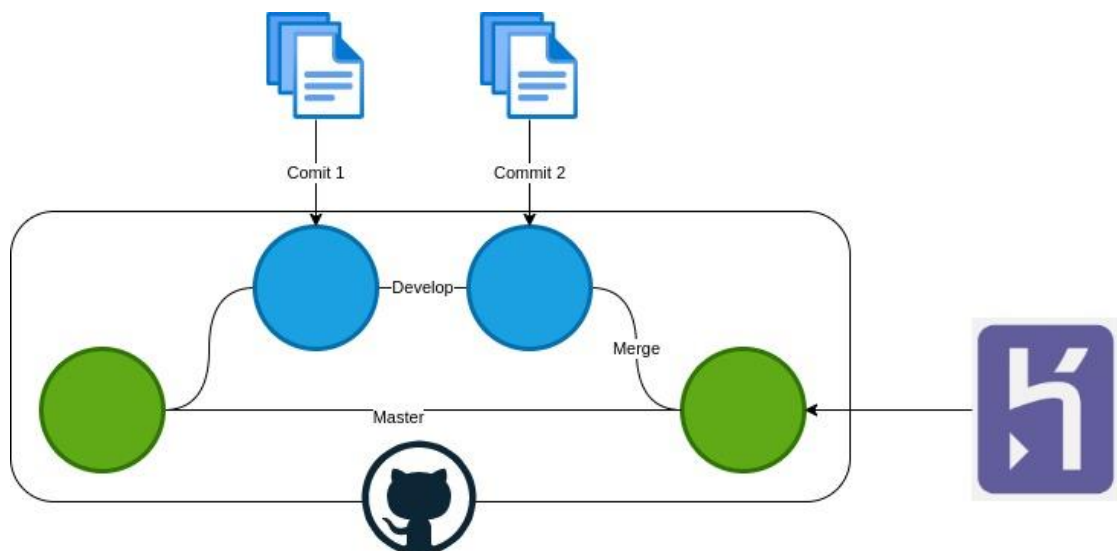


Figura 34. Despliegue en *Heroku*

Heroku nota estos cambios y realiza el despliegue de manera automática como se muestra en la figura 35.

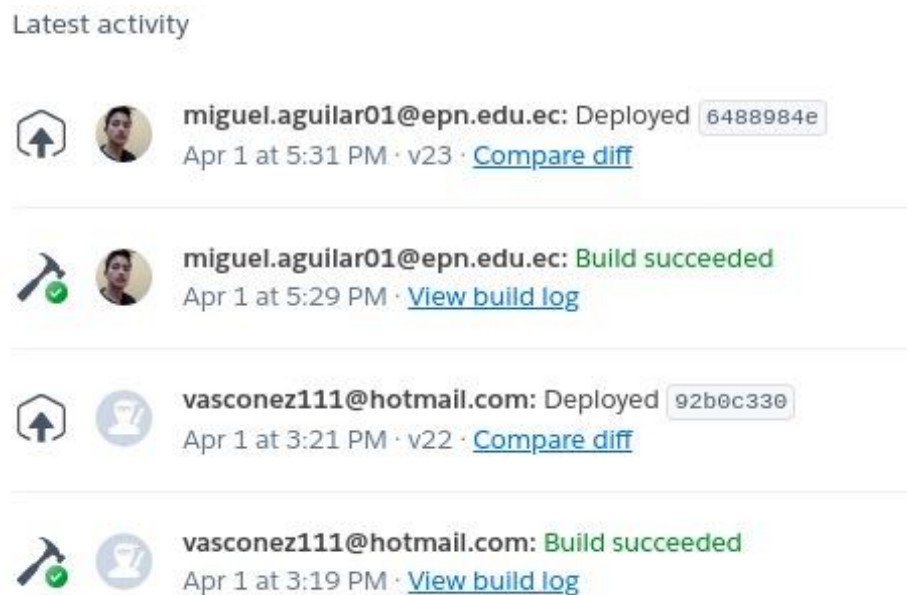


Figura 35. Despliegue automático

#### 2.3.4.4.4. Clase de conexión para subir imágenes a *firebase*

Para subir imágenes se creó una conexión a *firebase*, una sección del código se muestra en la figura 36.

```
import { initializeApp } from "firebase/app";
import { getStorage, ref, uploadBytes, getDownloadURL } from "firebase/storage";

export class FirebaseCnn {
  constructor() { }

  init = () => {
    const firebaseConfig = {
      apiKey: process.env.REACT_APP_API_KEY,
      authDomain: process.env.REACT_APP_AUTH_DOMAIN,
      projectId: process.env.REACT_APP_PROJECT_ID,
      storageBucket: process.env.REACT_APP_STORAGE_BUCKET,
      messagingSenderId: process.env.REACT_APP_MESSAGING_SENDER_ID,
      appId: process.env.REACT_APP_APP_ID
    };
    const app = initializeApp(firebaseConfig);
    return app;
  }
}
```

Figura 36. Código de conexión a *firebase*

#### 2.3.4.4.5. Devolver valores necesarios para paginación

La figura 37 muestra un ejemplo de valores para la paginación.

```
Curl
curl -X 'GET' \
  'https://blind-people-app-backend.herokuapp.com/service?order=desc&take=12' \
  -H 'accept: */*' \
  -H 'x-api-key: 420f77de-2cea-4e13-841a-b43ca729a7a9'

Request URL
https://blind-people-app-backend.herokuapp.com/service?order=desc&take=12
```

Figura 37. Paginación

La figura 38 muestra el máximo de servicios que se muestran por página.

```
}
],
"count": 7
}
```

Figura 38. Total servicios por página

#### 2.3.4.5. Revisión de *Sprint*

El *sprint* número cuatro, realizado en el mes de enero del 2022, se completó satisfactoriamente con un cien por ciento de cumplimiento en historias de usuario.

De este sprint se obtuvo lo siguiente:

- Se corrigió el error del campo teléfono al agregarlo al campo de la etiqueta HTML *value*.
- Se agregaron todos los campos en la sección de servicio en el *frontend*.
- Se cambió la posición de la barra de búsqueda en el dom.
- Se desplegó la aplicación en *Heroku*.
- Se creó la clase de conexión para subir imágenes a *firebase*.
- Se agregó los campos ciudad y categoría al *backend*.
- Se agregaron las cabezeras necesarias para la paginación de resultados en *backend*.
- Se encontró un fallo en la función de editar el perfil del usuario que será resuelta en el siguiente *sprint*.



De este sprint se tiene el siguiente entregable:

- Aplicación disponible en la web a través de Heroku

### **2.3.5. *Sprint* 5**

#### **2.3.5.1. *Objetivos del Sprint***

- Añadir características a aplicación *web*.

#### **2.3.5.2. *Lista de producto***

- *Sprint* 5 [*Frontend*] Agregar calificación y promedios de calificación en cada servicio.
- *Sprint* 5 [*Frontend*] Agregar sección para solicitar servicios.
- *Sprint* 5 [*Frontend*] Corregir edición de usuario.
- *Sprint* 5 [*Backend*] Añadir métodos de acceso a calificación.
- *Sprint* 5 [*Backend*] Agregar sección de comentarios.
- *Sprint* 5 [*Backend*] Sección para solicitar servicios.

#### **2.3.5.3. *Planificación del Sprint***

Este *sprint* tiene una duración de 28 días, correspondientes al mes de febrero del 2022. Las historias que se ejecutarán en este sprint serán las siguientes:

- *Sprint* 5 [*Frontend*] Agregar calificación y promedios de calificación en cada servicio.
- *Sprint* 5 [*Frontend*] Agregar sección para solicitar servicios.
- *Sprint* 5 [*Frontend*] Corregir edición de usuario.
- *Sprint* 5 [*Frontend*] Agregar sección de comentarios por servicio.
- *Sprint* 5 [*Backend*] Añadir métodos de acceso a *calificación*.
- *Sprint* 5 [*Backend*] Agregar sección de comentarios.
- *Sprint* 5 [*Backend*] Sección para solicitar servicios.

## 2.3.5.4. Implementación

### 2.3.5.4.1. Agregar calificación y promedios de calificación en cada servicio

Se crearon dos métodos que extraen la calificación por cliente y por servicio como se muestra en la figura 39.

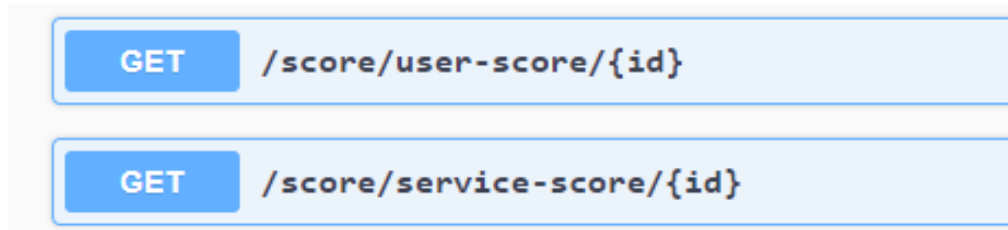


Figura 39. Métodos de *score* (calificación)

### 2.3.5.4.2. Agregar sección para solicitar servicios

La figura 40 muestra el formulario para la solicitud de un servicio por parte de un usuario.

### Solicitar servicio

Título

Ingrese el nombre del servicio que desea solicitar.

Descripción

Ingrese una descripción del servicio que desea solicitar.

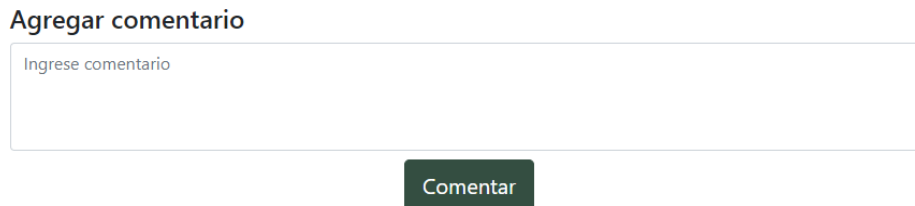
Figura 40. Sección para solicitar servicios

### 2.3.5.4.3. Corregir edición de usuario

Se realizó una corrección en el estado del perfil del usuario con la finalidad que sea presentado correctamente.

#### 2.3.5.4.4. Agregar sección de comentarios por servicio

La figura 41 muestra el formulario para agregar un comentario por parte de un usuario.



El formulario para agregar un comentario consta de un encabezado que dice "Agregar comentario". Debajo de este encabezado hay un campo de texto rectangular con el texto "Ingrese comentario" en un color gris claro. Debajo del campo de texto hay un botón rectangular de color verde oscuro con el texto "Comentar" en blanco.

Figura 41. Sección de comentarios

#### 2.3.5.5. Revisión de *Sprint*

El *sprint* número cinco, realizado en el mes de febrero del 2022, se completó satisfactoriamente con un cien por ciento de cumplimiento en historias de usuario.

De este *sprint* obtuvimos lo siguiente:

- Se agregó un método para calificar cada servicio y un campo para mostrar el promedio de la calificación total de cada servicio en el *frontend*.
- Se agregó la sección para solicitar servicios en el *frontend*.
- Se corrigió la edición de usuario que ocasionaba que los usuarios no pudieron editar su información.
- Se agregaron las secciones de comentarios, calificación y solicitar servicios al backend.

De este *sprint* se tienen los siguientes entregables:

- Función para calificar cada servicio dentro de la pantalla detalle de servicio
- Sección para solicitar servicios

#### 2.3.6. *Sprint* 6

##### 2.3.6.1. Objetivos del *Sprint*

- Diseño de pruebas con usuarios reales.
- Definir métodos para la medición de la accesibilidad de la aplicación.
- Definir el método de comparación de la accesibilidad de la aplicación con otras aplicaciones web similares.

### 2.3.6.2. Lista de producto

- *Sprint 6 [Test]* Diseño de pruebas de caso de prueba a usuarios reales.
- *Sprint 6 [Test]* Definir métodos para la medición de la accesibilidad de la aplicación.
- *Sprint 6 [Test]* Definir el método de comparación de la accesibilidad de la aplicación con otras aplicaciones web similares.
- *Sprint 6 [Correcciones]* Correcciones y mejoras en la aplicación.

### 2.3.6.3. Planificación del Sprint

El *sprint* número seis, tiene una duración de 31 días, correspondientes al mes de marzo del 2022. Las historias que se ejecutarán en este *sprint* serán las siguientes:

- *Sprint 6 [Test]* Diseño de pruebas con usuarios reales.
- *Sprint 6 [Test]* Definir métodos para la medición de la accesibilidad de la aplicación.
- *Sprint 6 [Test]* Definir el método de comparación de la accesibilidad de la aplicación con otras aplicaciones *web* similares.
- *Sprint 6 [Correcciones]* Correcciones y mejoras en la aplicación.

### 2.3.6.4. Implementación

#### 2.3.6.4.1. Diseño de pruebas con usuarios reales.

Para el diseño de pruebas con usuarios reales se estimaron inicialmente casos de prueba que fueron levantados con voluntarios. El proceso para la realización de pruebas será el siguiente:

- Se realizará una lista de casos de prueba con los casos considerados más importantes.
- Se seleccionará 10 personas con diferentes grados de discapacidad para la realización de las pruebas.
- El registro se lo llevará a cabo a través de una tabla de constará de las siguientes columnas:
  - Identificador del caso de prueba
  - Nombre del caso de prueba
  - Logrado:
    - Si

- No
  - Nivel de dificultad
  - Observaciones

Se tomará especial atención en el nivel de dificultad y las observaciones registradas, con la finalidad de ofrecer mejoras en la aplicación y el servicio.

Los casos de prueba para las pruebas fueron definidos y evidenciados en el Anexo 2 del presente documento, a continuación, se los describen:

- Registro del usuario ingresado su nombre completo, correo, teléfono, contraseña y reingrese contraseña.
- Inicio de sesión del usuario ingresado su correo electrónico y contraseña.
- Búsqueda de servicios ingresando información en la barra de búsqueda para encontrar servicios relacionados con dicha información.
- Publicar servicio completando el formulario correspondiente.
- Solicitar un servicio en el foro disponible en la aplicación.
- Filtrar servicios de acuerdo a los filtros seleccionados.
- Comentar un servicio publicado en la aplicación.
- Editar perfil del usuario actualizando los campos “Nombre”, “Correo” y “Teléfono”.
- Agregar servicios en una solicitud realizada por otro usuario utilizando el botón “Agregar mis servicios”.
- Ver propuestas visualizando la lista de servicios agregados a una solicitud por medio del botón “Ver Propuestas”.

Para referencias de los registros tomados de los usuarios se muestra la figura 42.

Escuela Politécnica Nacional

Responsable de la prueba: \_\_\_\_\_  
 Nivel de discapacidad del tester: \_\_\_\_\_

Identificador del caso de prueba	Nombre del caso de prueba	Logrado		Nivel de dificultad
		Sí	No	
CP1	Registro de usuario			
CP2	Inicio de sesión de usuario			
CP3	Búsqueda de servicios			
CP4	Publicar servicio			
CP5	Solicitar servicio			
CP6	Filtrar Servicios			
CP7	Comentar un servicio			
CP8	Editar Perfil			
CP9	Agregar servicios			
CP10	Ver propuestas			

Total de aciertos: \_\_\_\_\_

Figura 42. Formato de registro para casos de prueba

#### 2.3.6.4.2. Definir métodos para la medición de la accesibilidad de la aplicación

Para la medición de la accesibilidad de la aplicación se usaron las siguientes herramientas de medición de accesibilidad *web*:

- *Wave*
- *Lighthouse*
- *Arc Toolkit*

Estas herramientas servirán de referencia para realizar correcciones y mejorar la accesibilidad de la aplicación.

#### 2.3.6.4.3. Comparar la accesibilidad de la aplicación con otras aplicaciones *web* similares.

Para la medición de la accesibilidad de las aplicaciones *web* se usarán las siguientes herramientas de medición de accesibilidad *web*:

- *Wave*
- *Lighthouse*
- *Arc Toolkit*

Las aplicaciones *web* seleccionadas son las siguientes:

- *Amazon*
- Mercado Libre
- *Ebay*
- *Newegg*

Las páginas que serán probadas serán las siguientes:

- Página principal
- Ver un servicio/producto
- Inicio de sesión
- Registrase

Lo que se busca es determinar el nivel de accesibilidad en aplicaciones web similares, para así verificar la viabilidad de la aplicación, además de realizar respectivas optimizaciones

#### **2.3.6.5. Revisión de Sprint**

El *sprint* número seis, realizado en el mes de marzo del 2022, se completó satisfactoriamente con un cien por ciento de cumplimiento en historias de usuario.

De este sprint se obtuvo lo siguiente:

- Se diseñaron las pruebas de caso de prueba para los voluntarios.
- Se definieron los métodos para la medición de la accesibilidad de la aplicación.
- Se definió el método para realizar la comparación de nivel de accesibilidad de la aplicación con otras aplicaciones web similares
- Se realizaron correcciones y mejoras en la aplicación.

De este sprint se tienen los siguientes entregables:

- Versión final de la aplicación con todas las correcciones indicadas por el *product owner*.
- Casos de prueba

### **3. Resultados**

A continuación, se detallarán los resultados obtenidos del desarrollo y pruebas de la aplicación, además de comparativas con aplicaciones similares con el propósito de validar la calidad de la aplicación *web*.

La aplicación web consta de los siguientes componentes:

- Una página de inicio la cual posee una barra de búsqueda, un cuadro de filtros, y la presentación de los servicios disponibles para la aplicación.
- Una página de registro para que el usuario pueda crear una cuenta en la aplicación.
- Una página de inicio de sesión para que el usuario pueda acceder a todas las funcionalidades de la aplicación con sus credenciales,
- Una página de oferta de servicios para que el cliente publique un servicio.
- Una página de demanda de servicios para que los usuarios soliciten servicios que no estén disponibles en la aplicación.
- Un apartado de usuario donde se encuentra la información personal y los servicios publicados,
- Una página de detalle de servicios donde se encuentra toda la información relacionada al servicio seleccionado.

Para mayor información de las pantallas y usos consulte el anexo 3 correspondiente al manual de usuario.

### 3.1. Arquitectura en capas

La arquitectura en capas usada para el presente proyecto se lo puede visualizar en la Figura 4 presente en el sprint 1. Para el desarrollo de la aplicación web se utilizaron las siguientes tecnologías:

- Esta ampliación web se encuentra desplegada en *Heroku*.
- La aplicación web está desarrollada con *Node* y los *frameworks* de *NestJS* y *ReactJS*.
- La tecnología usada para la base de datos es *PostgreSQL*.
- El *framework* para acceso a la base de datos es *TypeORM*.
- La seguridad es manejada por *Passportjs*.
- Como *framework* de estilo se utiliza *Bootstrap 5*.
- Para el almacenamiento de archivos se utiliza *Firebase*.
- Para el acceso del *backend* y *frontend* *Heroku* es usado como plataforma, *Swagger* como visualización de métodos y monitorización de datos presentes en el *backend*, y *ReactJS* con su respectiva interfaz de usuario como *frontend*.



- El usuario final puede acceder a la aplicación a través de cualquier dispositivo.

### **3.2. Esquema base de datos**

La base de datos usada para el presente proyecto se la puede visualizar en la Figura 5 presente en el *sprint* 1, donde:

- La tabla USER almacenará los datos del usuario
- La tabla ROLE almacenará los roles que puede tener el usuario.
- La tabla SCORE almacenará las calificaciones que los usuarios den a los servicios.
- La tabla SERVICE almacenará los datos de los servicios publicados por los usuarios.
- La tabla COMMENT almacenará los comentarios realizados por los usuarios a los servicios.
- La tabla SERVICE\_CATEGORY almacenará las categorías disponibles para los servicios.
- La tabla CITY almacenará la provincia y ciudad disponibles.
- La tabla DEMAND\_SERVICE almacenará los servicios demandados por los usuarios.

### **3.3. Nivel de accesibilidad de aplicación web “Blind Service”.**

A continuación, en la figura 43, se muestran los resultados de cada página de la aplicación web obtenidos con la herramienta *Lighthouse*.

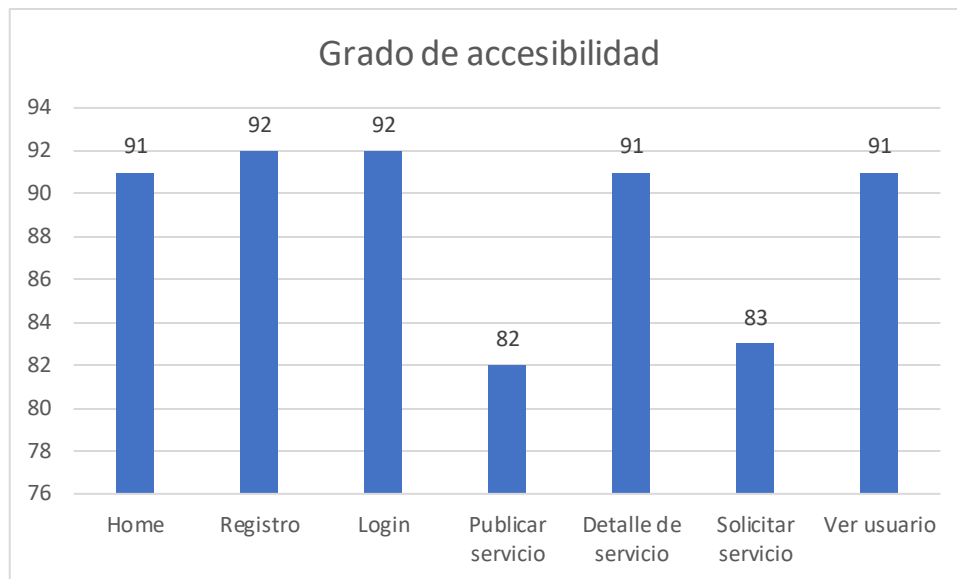


Figura 43. Grado de accesibilidad *Lighthouse*

Las barras representadas en el gráfico corresponden a la calificación dada por la aplicación *Lighthouse*, esto quiere decir que, mientras más alto sea la calificación, mayor accesible se considerará la aplicación, por lo tanto, para el paso de la aplicación "*Blind service*" puede ser considerada una aplicación altamente accesible para el público general.

### 3.4. Comparativa de aplicación web *Blind Service* con aplicaciones similares.

Para la medición de la accesibilidad de las aplicaciones web se usaron las siguientes herramientas de medición de accesibilidad, esto con la finalidad de proveer un punto de comparación con la aplicación desarrollada y similares:

- *Wave*
- *Lighthouse*
- *ARC Toolkit*

Las aplicaciones se seleccionaron en base a su giro de negocio, es decir que estas aplicaciones están orientadas a la oferta y demanda de productos/servicios:

- *Amazon*
- *Mercado Libre*
- *Ebay*
- *Newegg*

Las páginas seleccionadas de cada aplicación serán las siguientes, estas páginas fueron escogidas ya que cada una de ellas son normalmente las más utilizadas por los usuarios:

- Página principal
- Ver un servicio/producto
- Inicio de sesión
- Registrarse

La figura 44 muestra la cantidad de problemas de accesibilidad encontrados por la herramienta *Lighthouse* en las diferentes aplicaciones.

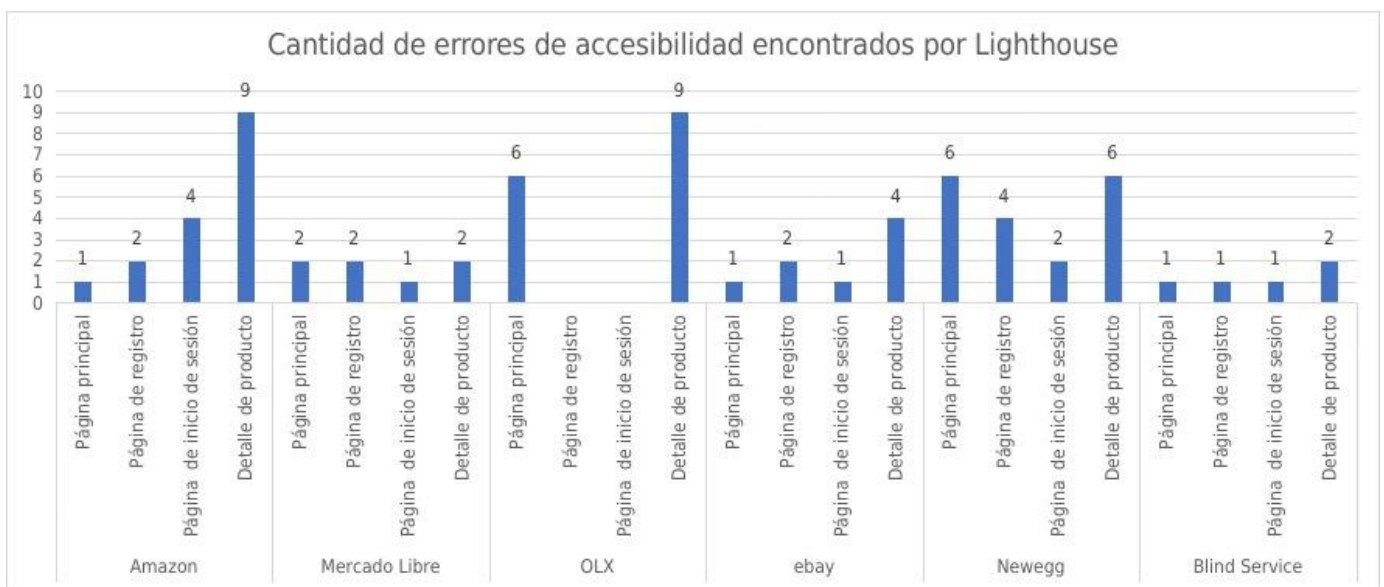


Figura 44. Cantidad de errores encontrados por *Lighthouse*

La figura 45 muestra la cantidad de problemas de accesibilidad encontrados por Wave en cada página seleccionada de las diferentes aplicaciones.

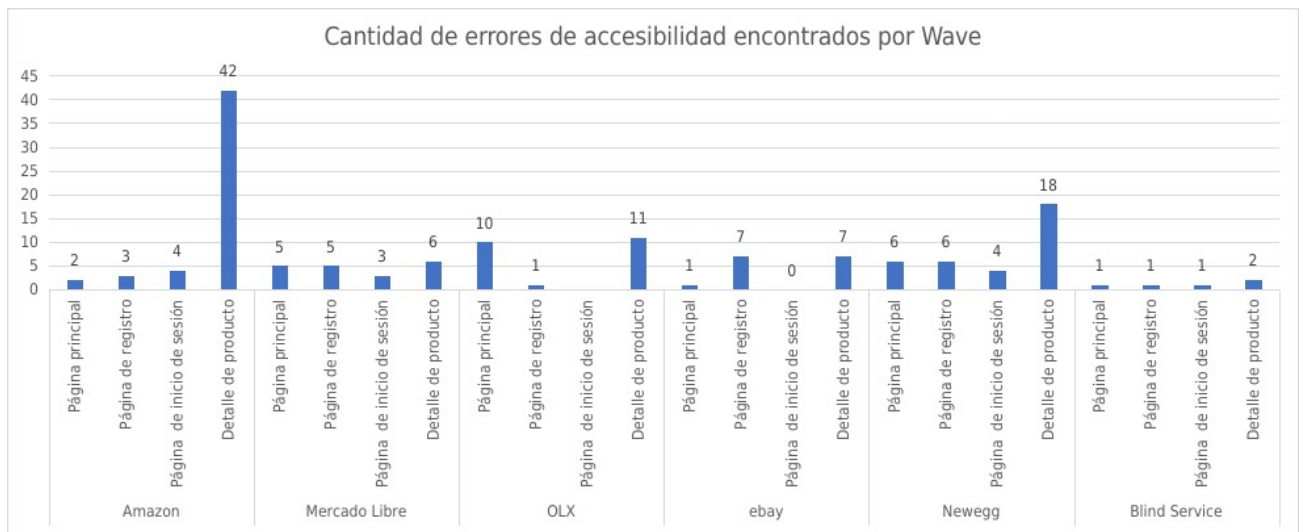


Figura 45. Cantidad de errores encontrados por Wave

La figura 46 muestra la cantidad de problemas de accesibilidad encontrados por la herramienta ARC Toolkit en cada página seleccionada de las diferentes aplicaciones.

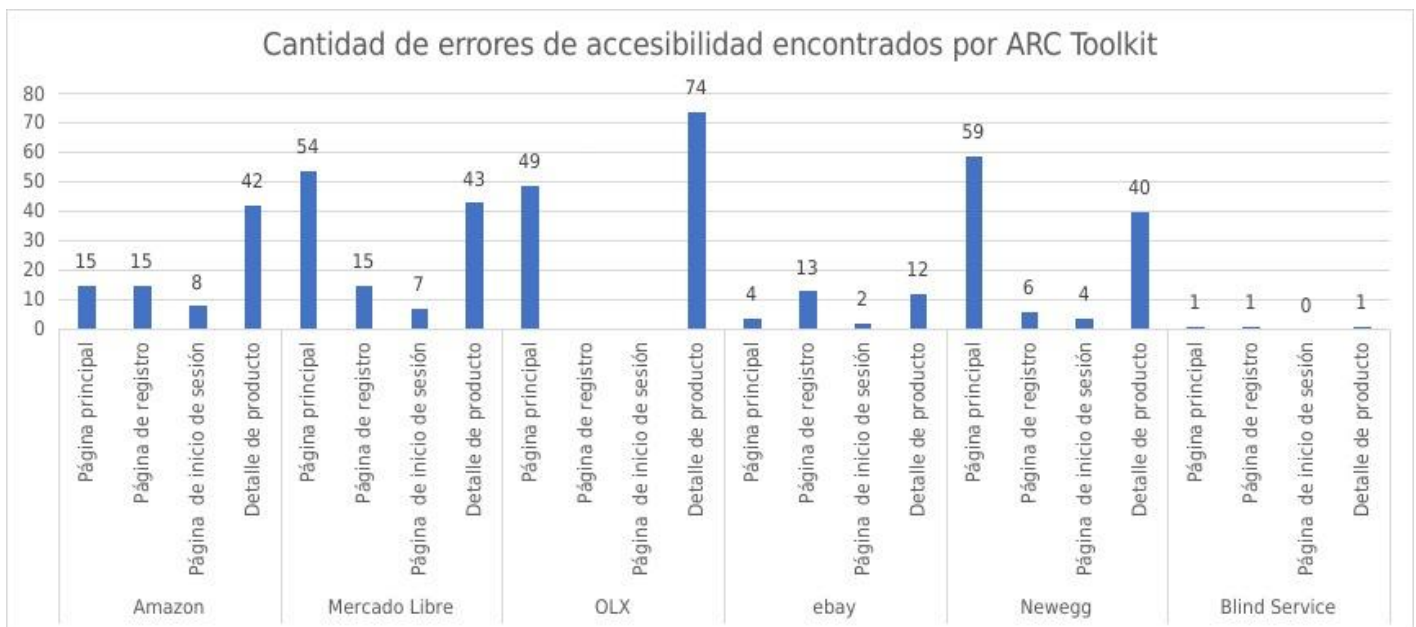


Figura 46. Cantidad de errores encontrados por ARC Toolkit

Como puede ser observado, la aplicación desarrollada tiene la menor cantidad de errores de accesibilidad presentada por las diferentes aplicaciones, esto es debido a que se buscó apegarse de mejor manera a los estándares de accesibilidad establecidos por la W3C.

### **3.5. Resultados de pruebas de accesibilidad con personas con algún grado de discapacidad visual.**

Los casos de prueba fueron probados con diez voluntarios, las pruebas fueron realizadas de la siguiente manera:

- Cada caso de prueba fue registrado en base a su totalidad de cumplimiento, es decir, si el usuario pudo o no realizar la tarea.
- Cada caso de prueba tuvo una calificación que refleja el grado de dificultad de la tarea, esta calificación se encuentra parametrizada entre 0 a 10, siendo 0 “nada difícil” y 10 “extremadamente difícil”.
- Cada caso de prueba tiene una sección de comentarios que el usuario pudo llenar libremente.
- Los cálculos presentados corresponden a las siguientes formulas:

$$\text{Diferencia} = \max(n_1, n_2, \dots) - \min(n_1, n_2, \dots)$$

$$\text{Promedio}(\bar{x}) = \frac{\sum_{i=1}^n x_i}{n}$$

$$\text{Variación Porcentual (Reducción de la dificultad)} = \frac{\bar{x}_1 - \bar{x}_0}{\bar{x}_0} * 100$$

A continuación, se muestran los resultados de estas pruebas realizadas a voluntarios con distinto grado de discapacidad visual, la figura 47 muestra el porcentaje de usuarios que completaron y que no completaron cada caso de prueba evaluado.

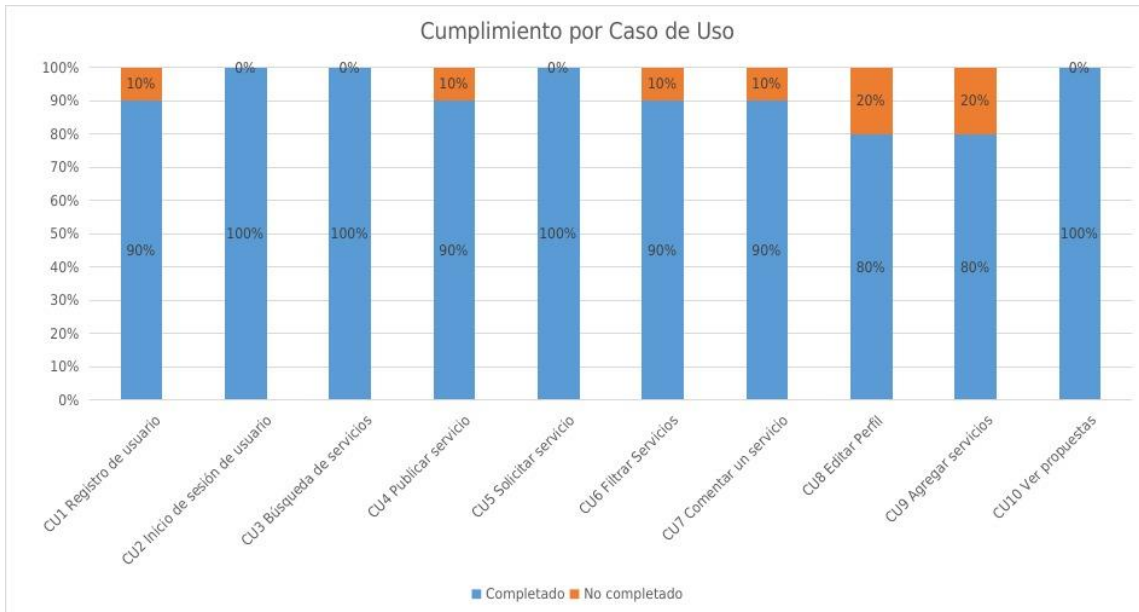


Figura 47. Cumplimiento por Caso de Prueba

La mayor parte se obtuvo un buen cumplimiento de las pruebas realizadas, además se pudo observar que en muchos de los casos las tareas resultaron ser muy sencillas para los usuarios.

La figura 48 muestra el grado de dificultad para completar el caso de prueba CP1.

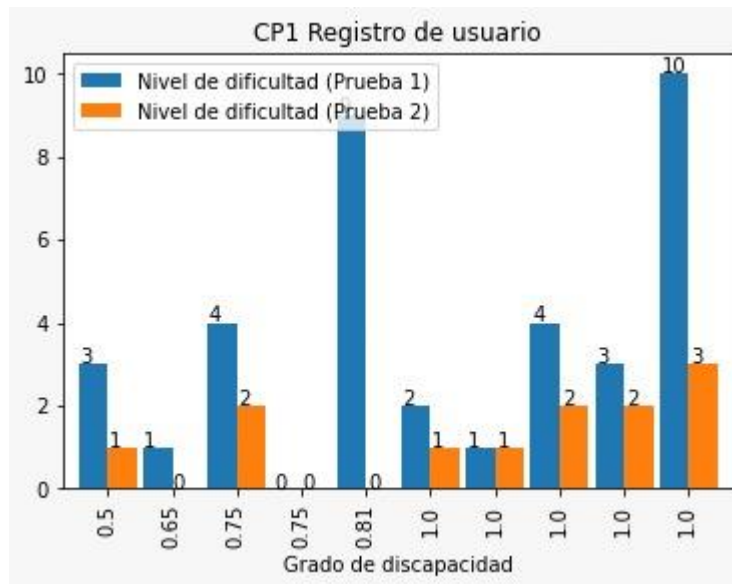


Figura 48. Dificultad por tarea y grado de discapacidad para CP1

La Tabla 2 muestra la diferencia y el promedio del nivel de dificultad en el caso de prueba CP1.

Tabla 2. Diferencia y promedio del nivel de dificultad para CP1

	Nivel de dificultad (Prueba 1)	Nivel de dificultad (Prueba 2)
Diferencia	10	3
Promedio	3,7	1,2

La variación porcentual (Reducción de la dificultad) fue del 67,57%

La figura 49 muestra el grado de dificultad para completar el caso de prueba CP2.

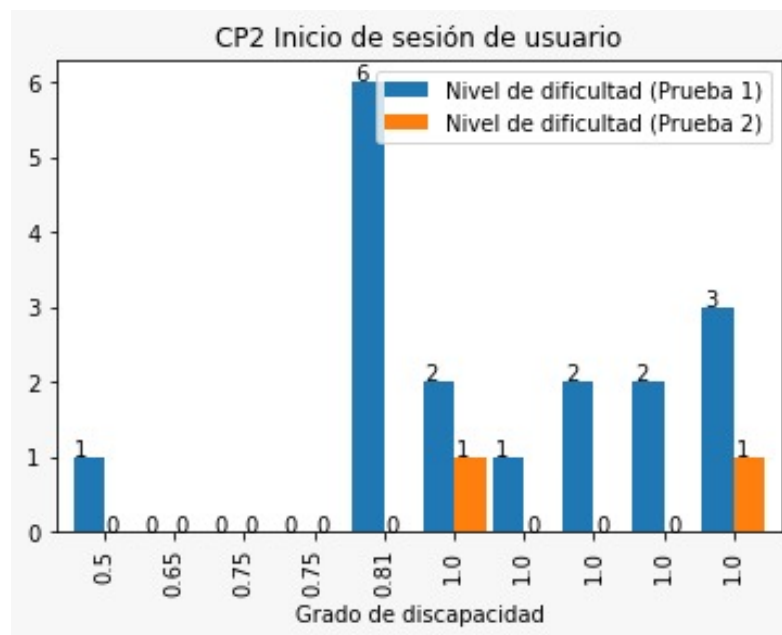


Figura 49. Dificultad por tarea y grado de discapacidad para CP2

La Tabla 3 muestra la diferencia y el promedio del nivel de dificultad en el caso de prueba CP2.

Tabla 3. Diferencia y promedio del nivel de dificultad para CP2

	Nivel de dificultad (Prueba 1)	Nivel de dificultad (Prueba 2)
Diferencia	6	1
Promedio	1,7	0,2

La variación porcentual (Reducción de la dificultad) fue del 88,24%

La figura 50 muestra el grado de dificultad para completar el caso de prueba CP3.

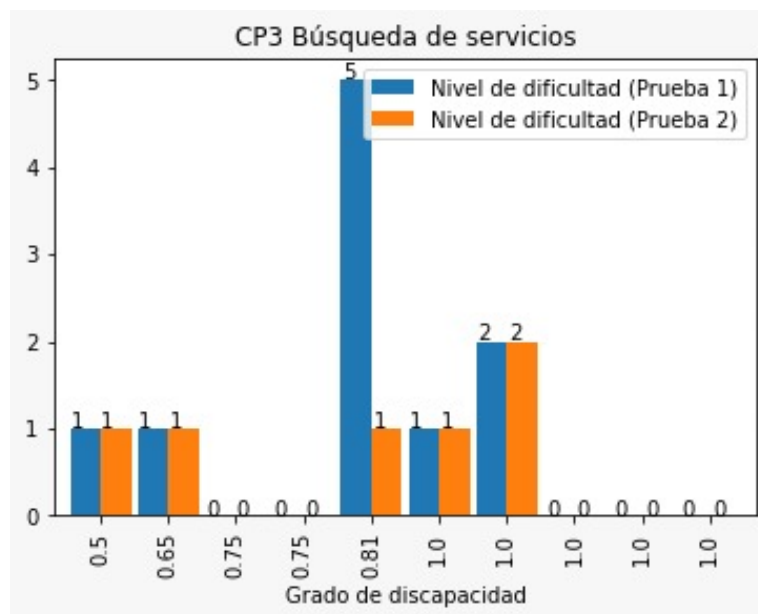


Figura 50. Dificultad por tarea y grado de discapacidad para CU3

La Tabla 4 muestra la diferencia y el promedio del nivel de dificultad en el caso de prueba CP3.



Tabla 4. Diferencia y promedio del nivel de dificultad para CP3

	Nivel de dificultad (Prueba 1)	Nivel de dificultad (Prueba 2)
Diferencia	5	2
Promedio	1	0,6

La variación porcentual (Reducción de la dificultad) fue del 40,00%

La figura 51 muestra el grado de dificultad para completar el caso de prueba CP4.

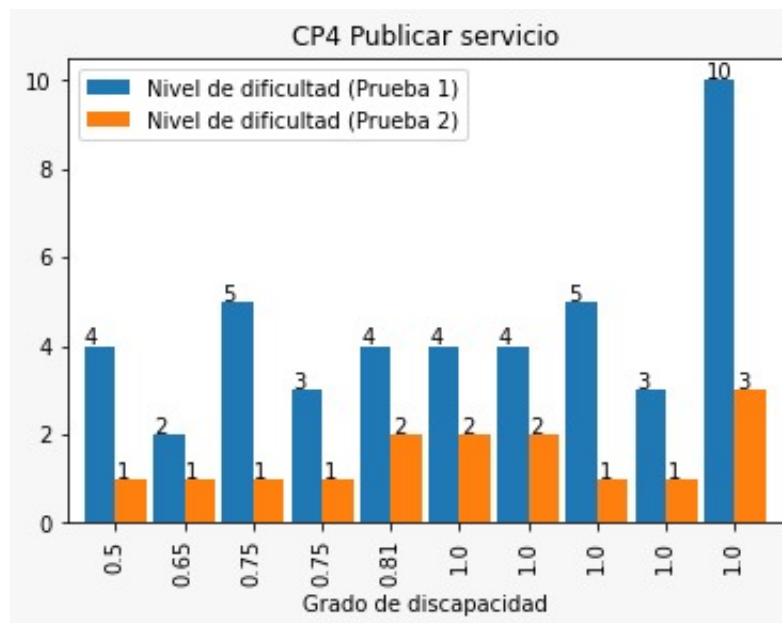


Figura 51. Dificultad por tarea y grado de discapacidad para CP4

La Tabla 5 muestra la diferencia y el promedio del nivel de dificultad en el caso de prueba CP4.

Tabla 5. Diferencia y promedio del nivel de dificultad para CP4

	Nivel de dificultad (Prueba 1)	Nivel de dificultad (Prueba 2)
Diferencia	8	2
Promedio	4,4	1,5

La variación porcentual (Reducción de la dificultad) fue del 65,91%

La figura 52 muestra el grado de dificultad para completar el caso de prueba CP5.

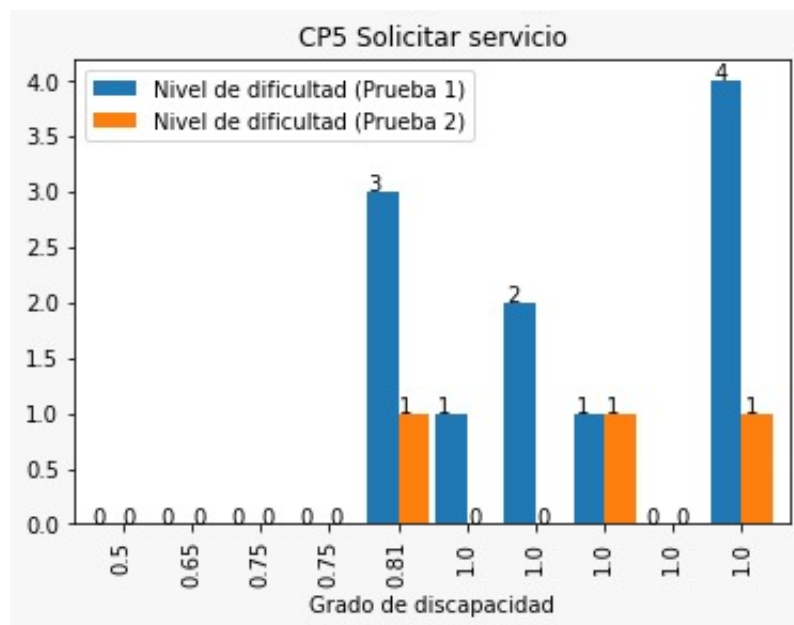


Figura 52. Dificultad por tarea y grado de discapacidad para CP5

La Tabla 6 muestra la diferencia y el promedio del nivel de dificultad en el caso de prueba CP5.

Tabla 6. Diferencia y promedio del nivel de dificultad para CP5

	Nivel de dificultad (Prueba 1)	Nivel de dificultad (Prueba 2)
Diferencia	4	1
Promedio	1,1	0,3

La variación porcentual (Reducción de la dificultad) fue del 72,73%

La figura 53 muestra el grado de dificultad para completar el caso de prueba CP6.

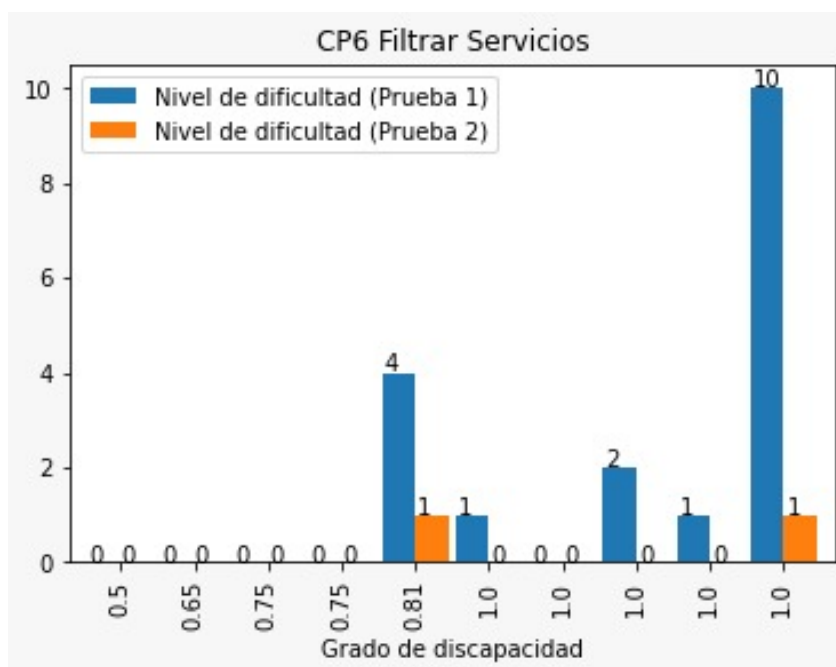


Figura 53. Dificultad por tarea y grado de discapacidad para CP6

La Tabla 7 muestra la diferencia y el promedio del nivel de dificultad en el caso de prueba CP6.

Tabla 7. Diferencia y promedio del nivel de dificultad para CP6

	Nivel de dificultad (Prueba 1)	Nivel de dificultad (Prueba 2)
Diferencia	10	1
Promedio	1,8	0,2

La variación porcentual (Reducción de la dificultad) fue del 88,89%

La figura 54 muestra el grado de dificultad para completar el caso de prueba CP7.

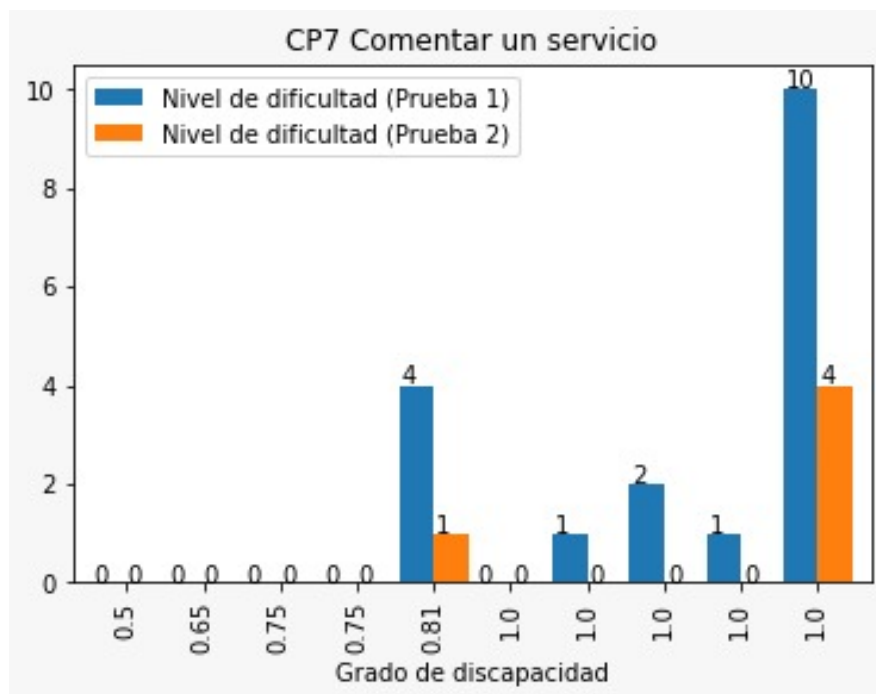


Figura 54. Dificultad por tarea y grado de discapacidad para CP7

La Tabla 8 muestra la diferencia y el promedio del nivel de dificultad en el caso de prueba CP7.

Tabla 8. Diferencia y promedio del nivel de dificultad para CP7

	Nivel de dificultad (Prueba 1)	Nivel de dificultad (Prueba 2)
Diferencia	10	4
Promedio	1,8	0,5

La variación porcentual (Reducción de la dificultad) fue del 72,22%

La figura 55 muestra el grado de dificultad para completar el caso de prueba CP8.

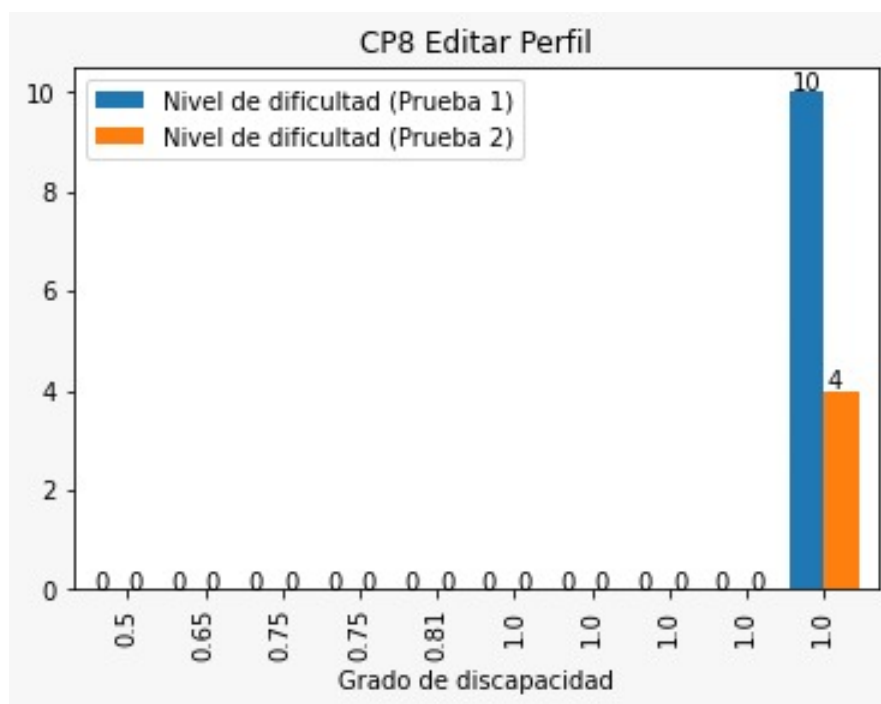


Figura 55 Dificultad por tarea y grado de discapacidad para CP8

La Tabla 9. muestra la diferencia y el promedio del nivel de dificultad en el caso de prueba CP8.

Tabla 9. Diferencia y promedio del nivel de dificultad para CP8

	Nivel de dificultad (Prueba 1)	Nivel de dificultad (Prueba 2)
Diferencia	10	4
Promedio	1	0,4

La variación porcentual (Reducción de la dificultad) fue del 60,00%

La figura 56 muestra el grado de dificultad para completar el caso de prueba CP9.

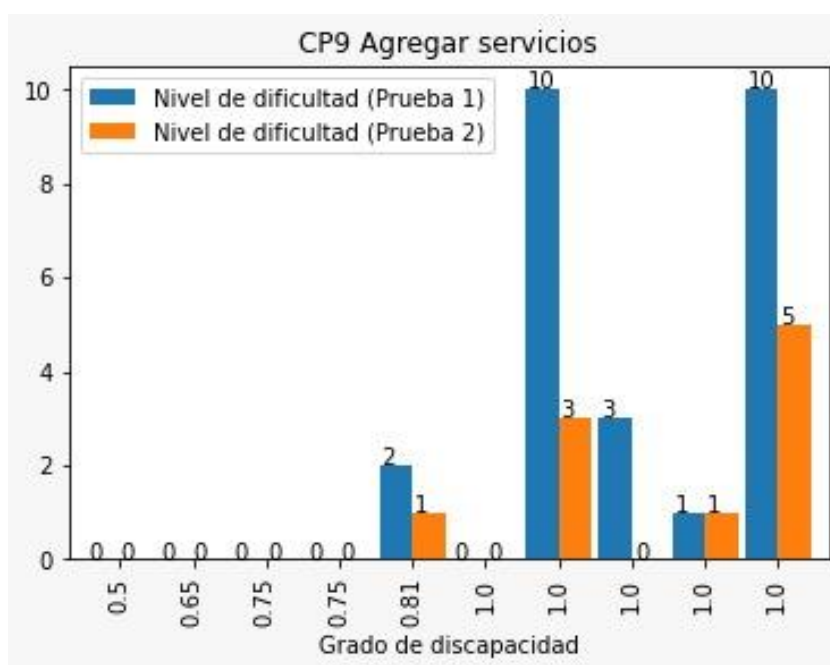


Figura 56. Dificultad por tarea y grado de discapacidad para CP9

La Tabla 10 muestra la diferencia y el promedio del nivel de dificultad en el caso de prueba CP9.

Tabla 10. Diferencia y promedio del nivel de dificultad para CP9

	Nivel de dificultad (Prueba 1)	Nivel de dificultad (Prueba 2)
Diferencia	10	5
Promedio	2,6	1

La variación porcentual (Reducción de la dificultad) fue del 61,54%

La figura 57 muestra el grado de dificultad para completar el caso de prueba CP10.

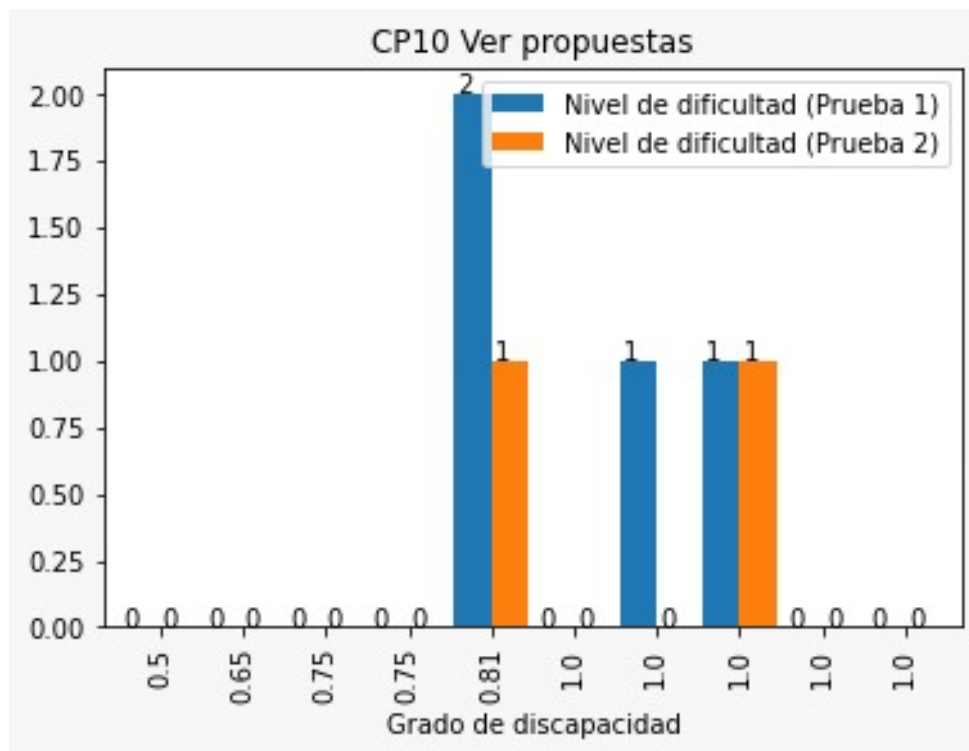


Figura 57. Dificultad por tarea y grado de discapacidad para CP10

La Tabla 11 muestra la diferencia y el promedio del nivel de dificultad en el caso de prueba CU10.

Tabla 11. Diferencia y promedio del nivel de dificultad para CP10

	Nivel de dificultad (Prueba 1)	Nivel de dificultad (Prueba 2)
Diferencia	2	1
Promedio	0,4	0,2

La variación porcentual (Reducción de la dificultad) fue del 50,00%

La Tabla 12 muestra la variación porcentual total de la aplicación

Tabla 12. Variación porcentual total

Variación porcentual total (Reducción de la dificultad total de la aplicación)	66,71%
--------------------------------------------------------------------------------	--------

En síntesis, en los resultados obtenidos de los casos de prueba, se puede observar que:

- Al analizar el CP1 Registro de usuario, se encontró que el 90% de voluntarios pudieron completarlo exitosamente a comparación de un 10% que no lo logró.
- Al analizar el CP2 Registro de usuario, se encontró que el 100% de voluntarios pudieron completarlo.
- Al analizar el CP3 Búsqueda de servicios, se encontró que el 100% de voluntarios pudieron completarlo.
- Al analizar el CP4 Publicar servicio, se encontró que el 90% de voluntarios pudieron completarlo exitosamente a comparación de un 10% que no lo logró.
- Al analizar el CP5 Solicitar servicio, se encontró que el 100% de voluntarios pudieron completarlo exitosamente a comparación de un 0% que no lo logró.
- Al analizar el CP6 Filtrar Servicios, se encontró que el 90% de voluntarios pudieron completarlos exitosamente a comparación de un 10% que no lo logró.
- Al analizar el CP7 Comentar un servicio, se encontró que el 90% de voluntarios pudieron completarlo exitosamente a comparación de un 10% que no lo logró.



- Al analizar el CP8 Editar Perfil, se encontró que el 80% de voluntarios pudieron completarlo exitosamente a comparación de un 20% que no lo logró.
- Al analizar el CP9 Agregar servicios, se encontró que el 80% de voluntarios pudieron completarlo exitosamente a comparación de un 20% que no lo logró.
- Al analizar el CP10 Ver propuestas, se encontró que el 100% de voluntarios pudieron completarlo exitosamente.

Al analizar los resultados obtenidos de la dificultad de los casos de prueba en la prueba 1, se puede observar que:

- Al analizar el CP1 Registro de usuario, se encontró que los voluntarios tuvieron en promedio una dificultad de 3.7 puntos, con lo que se realizaron correcciones respectivas, para así obtener en una segunda prueba una reducción del 67,57% en la dificultad.
- Al analizar el CP2 Registro de usuario, se encontró que los voluntarios tuvieron en promedio una dificultad de 1.7 puntos, con lo que se realizaron correcciones respectivas, para así obtener en una segunda prueba una reducción del 88,24% en la dificultad.
- Al analizar el CP3 Búsqueda de servicios, se encontró que los voluntarios tuvieron en promedio una dificultad de 1 punto, con lo que se realizaron correcciones respectivas, para así obtener en una segunda prueba una reducción del 40,00 % en la dificultad.
- Al analizar el CP4 Publicar servicio, se encontró que los voluntarios tuvieron en promedio una dificultad de 4.4 puntos, con lo que se realizaron correcciones respectivas, para así obtener en una segunda prueba una reducción del 65,91% en la dificultad.
- Al analizar el CP5 Solicitar servicio, se encontró que los voluntarios tuvieron en promedio una dificultad de 1.1 puntos, con lo que se realizaron correcciones respectivas, para así obtener en una segunda prueba una reducción del 72,73% en la dificultad.
- Al analizar el CP6 Filtrar Servicios, se encontró que los voluntarios tuvieron en promedio una dificultad de 1.8 puntos, con lo que se realizaron correcciones respectivas, para así obtener en una segunda prueba una reducción del 88,89% en la dificultad.

- Al analizar el CP7 Comentar un servicio, se encontró que los voluntarios tuvieron en promedio una dificultad de 1.8 puntos, con lo que se realizaron correcciones respectivas, para así obtener en una segunda prueba una reducción del 72,22% en la dificultad.
- Al analizar el CP8 Editar Perfil, se encontró que los voluntarios tuvieron en promedio una dificultad de 1 punto, con lo que se realizaron correcciones respectivas, para así obtener en una segunda prueba una reducción del 60,00% en la dificultad.
- Al analizar el CP9 Agregar servicios, se encontró que los voluntarios tuvieron en promedio una dificultad de 2.6 puntos, con lo que se realizaron correcciones respectivas, para así obtener en una segunda prueba una reducción del 61,54% en la dificultad.
- Al analizar el CP10 Ver propuestas, se encontró que los voluntarios tuvieron en promedio una dificultad de 0.4 puntos, con lo que se realizaron correcciones respectivas, para así obtener en una segunda prueba una reducción del 50,00% en la dificultad.

## 4. Conclusiones y Recomendaciones

### 4.1. Conclusiones

- Se realizó una investigación acerca de trabajos relacionados a accesibilidad web y comercio electrónico, de lo que se puede resaltar que existen aplicaciones e implementaciones de las mismas que buscan facilitar la vida de las personas con discapacidad visual, además de que se pudo constatar que los aportes no se encuentran solamente en el ámbito del comercio electrónico, sino que abarcan otros temas como lo son las aplicaciones de juegos, geolocalización, etc. (sección 1.3.11).
- Se desarrolló una aplicación web probada con herramientas que miden la accesibilidad web, de las cuales se obtuvieron resultados que reflejan una cantidad errores baja con respecto a otras aplicaciones similares en mismo giro de negocio (sección 3.3)
- Las pruebas con usuarios reales realimentan mejoras a la aplicación con la finalidad de incrementar la accesibilidad y dar un mejor servicio a los usuarios.
- En la prueba piloto (prueba 1 en la sección 3.5) se pudo validar varios tipos de dificultades en el uso, de la cual se obtuvieron retroalimentaciones que ayudaron a mejorar la aplicación, permitiendo una reducción del 66,71% en la dificultad al momento de realizar una segunda prueba (sección 3.5).
- Los resultados obtenidos de diferentes herramientas que miden la accesibilidad *web* pueden variar dependiendo de las pautas que tomen, pero son un buen referente al momento de desarrollar aplicaciones web accesibles a personas con discapacidad visual (sección 3.3 y sección 3.4).
- El uso de *ReactJS* para el desarrollo de la aplicación, permitió la reutilización de una gran cantidad de código en comparación al desarrollo tradicional con *JavaScript* puro (sección 2.3.5).
- El uso de *Bootstrap 5* para el desarrollo de la aplicación permitió la realización de interfaces de usuario con mayor facilidad en comparación de sus competencias como puede ser *Bulma* o *Foundation* (sección 2.3.5).
- El uso de *Swagger* permite un acceso rápido a los recursos y métodos del *Backend* además que fomenta buenas prácticas de desarrollo ya que es una manera visual de ver la aplicación de estándares *OpenApi 3.0* (sección 2.3.3)

## 4.2. Recomendaciones

- Utilizar tecnologías como *Heroku* para tener una versión web de la aplicación y poder realizar las diferentes pruebas a los voluntarios de forma remota, más no de forma local.
- Es importante continuar con el desarrollo de esta aplicación, o similares, ya que se pudo constatar un alto nivel de interés por entidades como lo son la Institución Educativa Especializada Mariana de Jesús y la Sociedad de Ciegos de Pichincha Luis Braille, que a pesar de no tener recursos suficientes para administrar las operaciones inherentes a la aplicación podrían darle otros tipos de uso como lo es la capacitación en dispositivos con acceso a internet.

## **5. Anexos**

### **Anexo I Requerimientos de usuario**

El documento con los requerimientos de usuario puede ser descargado a través del siguiente link: [https://epnecuador-my.sharepoint.com/:b:/g/personal/enrique\\_vasconez\\_epn\\_edu\\_ec/EVgLOUGJBXIKgAatKctbU5wBt4z7ni2AK2OO4RHDjlZMDg?e=xLKIsi](https://epnecuador-my.sharepoint.com/:b:/g/personal/enrique_vasconez_epn_edu_ec/EVgLOUGJBXIKgAatKctbU5wBt4z7ni2AK2OO4RHDjlZMDg?e=xLKIsi)

### **Anexo II Casos de prueba**

El documento de casos de prueba puede ser descargado a través del siguiente link: [https://epnecuador-my.sharepoint.com/:b:/g/personal/enrique\\_vasconez\\_epn\\_edu\\_ec/EQMMZnjXuL5PiV48DkkYhqoBox5YkJ9OMhjEDLT2WwKZMQ?e=l2ckZJ](https://epnecuador-my.sharepoint.com/:b:/g/personal/enrique_vasconez_epn_edu_ec/EQMMZnjXuL5PiV48DkkYhqoBox5YkJ9OMhjEDLT2WwKZMQ?e=l2ckZJ)

### **Anexo III Manual de usuario**

El manual de usuario de la aplicación puede ser descargado a través del siguiente link: [https://epnecuador-my.sharepoint.com/:b:/g/personal/enrique\\_vasconez\\_epn\\_edu\\_ec/EV-V5ECgtt5Mjxq2GsdQEQ4BBvy\\_P7xlzTWBAekAqHwN5Q?e=UviVk7](https://epnecuador-my.sharepoint.com/:b:/g/personal/enrique_vasconez_epn_edu_ec/EV-V5ECgtt5Mjxq2GsdQEQ4BBvy_P7xlzTWBAekAqHwN5Q?e=UviVk7)

## 6. Bibliografía

- [1] OMS, “Discapacidad y salud.” <https://www.who.int/es/news-room/fact-sheets/detail/disability-and-health> (accessed Aug. 20, 2022).
- [2] Incluyeme, “¿Que tipos de discapacidad existen?” <https://www.incluyeme.com/que-tipos-de-discapacidad-existen/> (accessed Jun. 04, 2021).
- [3] O. Castellero, “Los 6 tipos de discapacidad y sus características,” *Psicología y Mente*, p. 01, 2018, Accessed: Jun. 04, 2021. [Online]. Available: <https://psicologiaymente.com/salud/tipos-de-discapacidad>
- [4] Fundación Once, “Características de la discapacidad visual y ceguera,” *ONCE*, 2020. <https://www.once.es/dejanos-ayudarte/la-discapacidad-visual> (accessed Jun. 04, 2021).
- [5] A. Mandal, “Tipos de discapacidad visual,” pp. 1–2, 2019, Accessed: Jun. 04, 2021. [Online]. Available: [https://www.news-medical.net/health/Types-of-visual-impairment-\(Spanish\).aspx](https://www.news-medical.net/health/Types-of-visual-impairment-(Spanish).aspx)
- [6] OMS, “Ceguera y discapacidad visual,” *Nota descriptiva N° 282*, p. 2, 2013, Accessed: Aug. 23, 2020. [Online]. Available: <https://www.who.int/es/news-room/fact-sheets/detail/blindness-and-visual-impairment>
- [7] CONADIS, “Estadísticas de Discapacidad – Consejo Nacional para la Igualdad de Discapacidades,” *Ministerio de salud publica*, 2020. <https://www.consejodiscapacidades.gob.ec/estadisticas-de-discapacidad/#> (accessed Feb. 01, 2022).
- [8] S. López, “Ciegos en internet: discapacidad visual y acceso a la informacion,” 1998. <http://www.sidar.org/publica/press/recortes/25ciegos.htm> (accessed Jun. 04, 2021).
- [9] S. Luján-Mora, “Accesibilidad Web: Ceguera,” 2018. <http://accesibilidadweb.dlsi.ua.es/?menu=deficit-visual-ceguera> (accessed Jun. 04, 2021).

- [10] J. Antonio Muñoz Sevilla, "Las TIC y la discapacidad visual," 2012. Accessed: Jun. 04, 2021. [Online]. Available: [https://ruc.udc.es/dspace/bitstream/handle/2183/13227/CC-127\\_art\\_18.pdf](https://ruc.udc.es/dspace/bitstream/handle/2183/13227/CC-127_art_18.pdf)
- [11] S. Luján-Mora, "Accesibilidad Web: Magnificadores de pantalla." <http://accesibilidadweb.dlsi.ua.es/?menu=magnificadores> (accessed Jun. 04, 2021).
- [12] M. Carrio, "Herramientas Tiflotécnicas y su función en la escuela. | Observatorio Tecnológico," 2006. <http://recursostic.educacion.es/observatorio/web/fr/cajon-de-sastre/38-cajon-de-sastre/317-herramientas-tiflotecnicas-y-su-funcion-en-la-escuela> (accessed Jun. 04, 2021).
- [13] S. Luján-Mora, "Accesibilidad Web: Lectores de pantalla," 2019, Accessed: Jun. 04, 2021. [Online]. Available: <http://accesibilidadweb.dlsi.ua.es/?menu=lectores>
- [14] A. Basantes, F. Guerra, M. Naranjo, and D. Ibadango, "Los Lectores de Pantalla: Herramientas Tecnológicas para la Inclusión Educativa de Personas no Videntes," *Información tecnológica*, vol. 29, no. 5, p. 82, 2018, doi: 10.4067/s0718-07642018000500081.
- [15] S. Luján-Mora, "Accesibilidad Web: Líneas braille." <http://accesibilidadweb.dlsi.ua.es/?menu=lin-braille> (accessed Jun. 04, 2021).
- [16] A. Martins, "A new survey highlights a 'digital divide' between the blind and access to the internet," 2019. <https://www.businessnewsdaily.com/15264-is-your-website-inaccessible-to-the-visually-impaired.html> (accessed Jun. 04, 2021).
- [17] S. Lab, "¿Qué es un sitio web y cómo diferenciarlo de una página web?," 2017. <https://softwarelab.org/es/sitio-web/> (accessed Jun. 04, 2021).
- [18] Techopedia, "WHAT IS A WEBSITE? - DEFINITION FROM TECHOPEDIA," 2021. <https://www.techopedia.com/definition/5411/website> (accessed Jun. 04, 2021).
- [19] M. MacPherson, "Website vs Web App: What's the Difference? | by Essential Designs | Medium," 2019. <https://medium.com/@essentialdesign/website-vs-web-app-whats-the-difference-e499b18b60b4> (accessed Jun. 04, 2021).

- [20] Sugandha, "Difference Between Web application and Website," *geeksforgeeks.org*, 2020. <https://www.geeksforgeeks.org/difference-between-web-application-and-website/> (accessed Jun. 04, 2021).
- [21] Guru99, "Difference between Website and Web Application." <https://www.guru99.com/difference-web-application-website.html> (accessed Jun. 04, 2021).
- [22] I. de Souza, "¿Qué es un servidor web y cuáles son sus características?," 2019. <https://rockcontent.com/es/blog/que-es-un-servidor/> (accessed Jun. 04, 2021).
- [23] V. Olaya, "Sistemas de Información Geográfica," in *Sistemas de Información Geográfica*, New York, 2020, p. 159.
- [24] STATIC APPS, "Defining Static Web Apps," 2014. <https://www.staticapps.org/articles/defining-static-web-apps/> (accessed Jun. 04, 2021).
- [25] YeePLY Editor, "Desarrollo de aplicaciones web: 5 tipos que debes conocer," *YEEPLY*, 2020. <https://www.yeeply.com/blog/6-tipos-desarrollo-de-aplicaciones-web/#tienda> (accessed Jun. 04, 2021).
- [26] H. Darwin, "What are the different types of web hosting? | TechRadar," 2019. <https://hellodarwin.com/blog/different-web-apps> (accessed Jun. 04, 2021).
- [27] E. Higuerey, "¿Qué es el comercio electrónico y cuáles son sus ventajas?," *rockcontent*, 2019. <https://rockcontent.com/es/blog/comercio-electronico/> (accessed Jun. 04, 2021).
- [28] Codelessplatforms, "What is a Web Portal? | Quick Start Guide: Web Portal Project," 2017. <https://www.codelessplatforms.com/blog/what-is-a-web-portal/> (accessed Jun. 04, 2021).
- [29] J. Bhaskar, "Key Differences Between Websites and Web Portals," *Customer Think*, 2020. <https://customerthink.com/key-differences-between-websites-and-web-portals/> (accessed Jun. 04, 2021).
- [30] Kinsta, "What Is a Content Management System (CMS)?," *Kinsta*, 2021. <https://kinsta.com/knowledgebase/content-management-system/> (accessed Jun. 04, 2021).



- [31] innovamedia, “¿Qué es un gestor de contenido o CMS?,” 2020. <https://www.innovamediaconsultores.com/blog/gestor-contenido-cms/> (accessed Jun. 04, 2021).
- [32] M. Kurtz, “Introduction to E-commerce: Combining Business and Information Technology,” in *BookBoon*, 1st ed., 2016, p. 16. [Online]. Available: <https://irp-cdn.multiscreensite.com/1c74f035/files/uploaded/introduction-to-e-commerce.pdf>
- [33] SALESFORCE LATINOAMÉRICA, “Qué son las metodologías ágiles y cómo pueden ayudar | Salesforce - Blog de Salesforce.” <https://www.salesforce.com/mx/blog/2021/12/que-son-metodologias-agiles-y-como-pueden-ayudar-a-tus-equipos-de-trabajo.html> (accessed Aug. 16, 2022).
- [34] REDHAT, “¿Qué es la metodología ágil?” <https://www.redhat.com/es/devops/what-is-agile-methodology> (accessed Aug. 16, 2022).
- [35] K Beck and D Cleal, “Optional Scope Contracts,” 1999.
- [36] J Joskowicz, “Reglas y Prácticas en eXtreme Programming,” 2008.
- [37] E Bahit, “Scrum y eXtreme Programming para Programadores,” 2012.
- [38] K. Schwaber and J. Sutherland, “La Guía de Scrum TM,” 2017. Accessed: Aug. 21, 2020. [Online]. Available: <http://creativecommons.org/licenses/by-sa/4.0/legalcodeandalsodescribedinsummaryformathttp://creativecommons.org/licenses/by-sa/4.0/.Byutilizing>
- [39] W3C, “Introducción a la Accesibilidad Web | Web Accessibility Initiative (WAI) | W3C.” <https://www.w3.org/WAI/fundamentals/accessibility-intro/es> (accessed Aug. 20, 2022).
- [40] W3C, “World Wide Web Consortium,” W3C, 2019. <https://www.w3c.es/> (accessed Jun. 06, 2021).
- [41] Agencia Nacional de Discapacidad de Argentina, “Pautas de Accesibilidad para el Contenido de la Web (WCAG),” 2016.

- [42] Universidad de Alicante, “Accesibilidad Web: Principios y directrices de WCAG 2.0.” <http://accesibilidadweb.dlsi.ua.es/?menu=principios-2.0> (accessed Jun. 28, 2022).
- [43] Id4you, “Los 4 principios básicos de una web accesible - ID4YOU.” <https://www.id4you.com/blog/tendencias/los-4-principios-basicos-de-una-web-accesible/> (accessed Jun. 28, 2022).
- [44] Portal único de trámites ciudadanos -Gobierno nacional - Ecuador, “Ley de Comercio Electrónico, Firmas y Mensajes de Datos | Ecuador - Guía Oficial de Trámites y Servicios,” 2002. <https://www.gob.ec/regulaciones/2002-67-ley-comercio-electronico-firmas-mensajes-datos> (accessed Jun. 06, 2021).
- [45] MINTEL, “Estrategia Nacional de Comercio Electrónico,” Ecuador, 2020. Accessed: Jun. 06, 2021. [Online]. Available: <https://aportecivico.gobiernoelectronico.gob.ec/system/documents/attachments/000/000/011/original/58b9ab393399dc479d2fb43c7a305ff0de62ec96.PDF>
- [46] J. Villar Cloquell, “PROPUESTA METODOLÓGICA PARA IMPULSAR EL DESARROLLO DE NUEVAS APLICACIONES PARA LAS PERSONAS CIEGAS O CON VISIÓN REDUCIDA A TRAVÉS DEL ANÁLISIS DE LA VIGILANCIA TECNOLÓGICA EN EL SECTOR DE LA TIFLOTECNOLOGÍA.” <https://riunet.upv.es/handle/10251/53241> (accessed Jun. 24, 2022).
- [47] LAZARILLO, “Comercio Digital: Una alternativa accesible para personas ciegas en el mundo - Lazarillo.” <https://lazarillo.app/es/2020/11/19/comercio-digital-una-alternativa-accesible-para-personas-ciegas-en-el-mundo/> (accessed Jun. 24, 2022).
- [48] American Academy of Ophthalmology, “30 aplicaciones, dispositivos y tecnologías para personas con discapacidad visual - American Academy of Ophthalmology.” <https://www.aaopt.org/salud-ocular/consejos/30-aplicaciones-dispositivos-y-tecnolog%C3%ADas-para-pe> (accessed Jun. 24, 2022).
- [49] E. Pérez, “Accesibilidad en Android: 21 apps para personas ciegas o con problemas de visión.” <https://www.xatakandroid.com/aplicaciones-android/accesibilidad-en-android-21-aplicaciones-para-personas-invidentes> (accessed Jun. 26, 2022).

- [50] Compartolid, “KNFB Reader.” <https://www.compartolid.es/knfb-reader/> (accessed Jun. 26, 2022).
- [51] J. Sáenz De La Joya, “Desarrollo de aplicación web tipo Blog accesible para personas ciegas,” 2017.
- [52] V. Díaz, “Amazon impulsa la inclusión a través de un nuevo website de Accesibilidad de Alexa | El Economista.” <https://www.eleconomista.com.mx/empresas/Amazon-impulsa-la-inclusion-a-traves-de-un-nuevo-website-de-Accesibilidad-de-Alexa-20211005-0044.html> (accessed Jun. 26, 2022).
- [53] G. Sabrina, A. Sánchez, D. Belén, E. Sanz Cecilia, and G. Gorga, “Accesibilidad web. Aplicación a un estudio de caso”, Accessed: Jun. 24, 2022. [Online]. Available: [http://sedici.unlp.edu.ar/bitstream/handle/10915/118598/Documento\\_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y](http://sedici.unlp.edu.ar/bitstream/handle/10915/118598/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y)
- [54] Monsido, “What is Web Accessibility & Why Is It So Important?,” *monsido*. <https://monsido.com/web-accessibility> (accessed Jun. 24, 2022).
- [55] Bureau of Internet Accessibility, “Why Web Accessibility is Important: 4 Reasons to Create Accessible Content.” <https://www.boia.org/blog/why-web-accessibility-is-important-4-reasons-to-create-accessible-content> (accessed Jun. 26, 2022).
- [56] MDN web docs, “JavaScript - Речник | MDN,” 2021. <https://developer.mozilla.org/es/docs/Web/JavaScript> (accessed Jul. 09, 2021).
- [57] J. Sirkin, “¿Qué es SQL o lenguaje de consultas estructuradas? - Definición en Whatls.com,” 2021. <https://searchdatacenter.techtarget.com/es/definicion/SQL-o-lenguaje-de-consultas-estructuradas> (accessed Jul. 09, 2021).
- [58] TypeORM, “TypeORM,” *TypeORM*. <https://typeorm.io/> (accessed May 03, 2022).
- [59] MDN Web Docs, “HTML: Lenguaje de etiquetas de hipertexto | MDN,” *HTML: Lenguaje de etiquetas de hipertexto*, 2021. <https://developer.mozilla.org/es/docs/Web/HTML> (accessed Jul. 09, 2021).

- [60] Juan José Pino Reyes, "QUÉ ES HTML," *01 de enero 2001*, 2018. <https://devcode.la/blog/que-es-html/> (accessed Jul. 09, 2021).
- [61] MDN web docs, "CSS," 2021. <https://developer.mozilla.org/es/docs/Web/CSS> (accessed Jul. 09, 2021).
- [62] OpenJS, "Acerca | Node.js," 2021. <https://nodejs.org/es/about/> (accessed Jul. 09, 2021).
- [63] J. Lucas, "Qué es NodeJS y para qué sirve | OpenWebinars," 2019. <https://openwebinars.net/blog/que-es-nodejs/> (accessed Jul. 09, 2021).
- [64] Visual Studio Code, "Documentation for Visual Studio Code," *Visual Studio Code Documentation*, 2015. <https://code.visualstudio.com/docs> (accessed Jul. 09, 2021).
- [65] P. Guajardo, "Bootstrap: ¿qué es, para qué sirve y cómo instalarlo?," 2020. <https://rockcontent.com/es/blog/bootstrap/> (accessed Jul. 09, 2021).
- [66] Theme Junkie, "What is Figma? (And How to Use Figma for Beginners)," Sep. 17, 2020. <https://www.theme-junkie.com/what-is-figma/> (accessed May 03, 2022).
- [67] DBeaver Community, "DBeaver Community: Free Universal Database Tool," 2020. <https://dbeaver.io/> (accessed Jul. 09, 2021).
- [68] PowerDesigner, "Funcionalidades principales de PowerDesigner," *PowerDesigner*, 2020. <https://www.powerdesigner.biz/ES/powerdesigner/powerdesigner-features.html> (accessed Jul. 09, 2021).
- [69] Heroku, "About Heroku | Heroku," 2019. <https://www.heroku.com/about> (accessed Nov. 21, 2021).
- [70] "¿Qué es Swagger? Definición de Whatls.com." <https://searcharchitecture.techtarget.com/definition/Swagger> (accessed Nov. 21, 2021).

- [71] COGNOS - Capacitación, “Desarrollo de Aplicaciones Móviles Nativas con Nestjs.” <https://cognos-capacitacion.com/Sistema/Cursos/NESTJS-100> (accessed May 25, 2022).
- [72] PostgreSQL, “PostgreSQL: About,” 2022. <https://www.postgresql.org/about/> (accessed May 03, 2022).
- [73] Passport.js, “Passport.js,” *Passport.js*. <https://www.passportjs.org/> (accessed May 03, 2022).
- [74] ReactJS, “Empezando – React.” <https://es.reactjs.org/docs/getting-started.html> (accessed May 03, 2022).
- [75] “JSON Web Token Introduction - jwt.io.” <https://jwt.io/introduction> (accessed May 03, 2022).
- [76] Grupo Social ONCE, “Tiflotecnología, ¿qué es y dónde solicitarla?” <https://www.once.es/servicios-sociales/tecnologiayrecursosadaptados/tiflotecnologia> (accessed May 25, 2022).