



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“DISEÑO E IMPLEMENTACIÓN DE APLICACIÓN PARA
EMPAREJAMIENTO AUTOMÁTICO DE PACIENTES Y VOLUNTARIOS
CON PREFERENCIAS SIMILARES.”

INFORME DE PROYECTO INTEGRADOR

Previo a la obtención del Título de:

INGENIERO EN COMPUTACIÓN

JOYCE EVELYN BENÍTEZ PIZARRO

KEVIN ISMAEL FILELLA LOK

GUAYAQUIL – ECUADOR

AÑO: 2018

AGRADECIMIENTOS

Mis más sinceros agradecimientos a Dios por permitirme culminar mi carrera universitaria, a mis padres y familia por darme su apoyo incondicional y a mis profesores por compartir sus conocimientos con dedicación en cada una de las materias que he cursado en ESPOL.

Joyce Benítez

AGRADECIMIENTOS

Por el presente proyecto quisiera agradecer a mis padres, por darme todo el apoyo necesario para poder culminar mi carrera universitaria. Quisiera agradecer a la Escuela Superior Politécnica el Litoral por darme la oportunidad de estudiar y desenvolverme como estudiante. También quisiera agradecer a todos los profesores que formaron parte de mi carrera por brindarme sus conocimientos y ayudar a formarme como profesional.

Kevin Filella

DEDICATORIA

El presente proyecto lo dedico a Dios y a mis padres, ya que sin sus ayudas no me hubiera sido posible lograrlo.

Joyce Benítez

DEDICATORIA

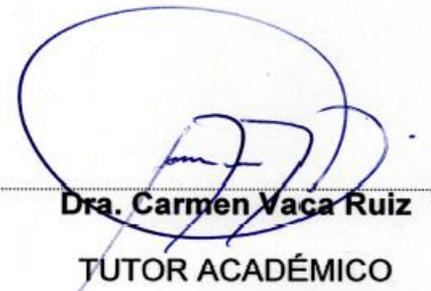
El presente proyecto se lo dedico a mis padres, por ser el pilar más importante y por brindarme siempre su incondicional apoyo y cariño.

Kevin Filella

TRIBUNAL DE EVALUACIÓN



Ph.D. Daniel Ochoa Donoso
PROFESOR DE MATERIA
INTEGRADORA



Dra. Carmen Vaca Ruiz
TUTOR ACADÉMICO

**"DISEÑO E IMPLEMENTACIÓN DE APLICACIÓN PARA EMPAREJAMIENTO AUTOMÁTICO
DE PACIENTES Y VOLUNTARIOS CON PREFERENCIAS SIMILARES"**

JOYCE EVELYN BENÍTEZ PIZARRO Mat. # 201304469

KEVIN ISMAEL FILELLA LOK Mat. # 200800084

II Término 2017

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"


Joyce Benítez


Kevin Filella

RESUMEN

En clínicas y hospitales de la ciudad de Guayaquil, existen fundaciones que se han afiliado para ofrecer el servicio de voluntariado. Los voluntarios visitan periódicamente los centros de salud para realizar actividades con los pacientes. Actualmente no existe un medio personalizado de interacción entre ellos y no hay forma de conocer la similitud de gustos que existen entre los pacientes y voluntarios, por esta razón se desea desarrollar un sistema que empareje voluntarios y pacientes basados en sus gustos similares.

Para la realización del emparejamiento entre voluntarios y pacientes basados en gustos similares, se implementó el algoritmo de matrimonio estable de Gale y Shapley adaptado a las necesidades del presente proyecto. Para implementarlo se utilizaron los gustos de los pacientes y voluntarios, teniendo los gustos calificados según su intensidad de agrado, se realizan los respectivos cálculos descritos en la sección de metodología para obtener el porcentaje de similitud que existe entre pacientes y voluntarios, estos porcentajes de similitud se utilizan para que el algoritmo dé como resultado parejas (paciente y voluntario) que se prefieren mutuamente.

Los resultados del proyecto fueron los siguientes:

- Aplicación móvil Android: Permite que los pacientes y voluntarios puedan registrar sus datos personales básicos y calificar sus gustos, estas calificaciones, las procesa el algoritmo de matrimonio estable de Gale y Shapley y da como resultado a la persona que tenga mayor similitud de gustos. Las otras funcionalidades que se pueden realizar en la aplicación móvil se encuentran en la sección de metodología.
- Aplicación web: Permite al administrador manejar los datos del sistema.

En conclusión, se implementó una aplicación móvil para que pacientes y voluntarios puedan interactuar y a su vez que los pacientes puedan conocer voluntarios que tenga mayor afinidad de gustos ya que se implementó el algoritmo de matrimonio estable de Gale y Shapley.

Palabras Clave: matrimonio estable, emparejamiento estable, voluntariado, gale shapley, hospitales, pacientes.

ABSTRACT

Several non-profit charity foundations offer volunteering services in clinics and hospitals in the city of Guayaquil, Ecuador. Volunteers occasionally visit these health centers to carry out recreational activities with patients. Currently, there are no personalized means of interaction between members of each group (patients and volunteers), which means there is no way for members of each group to know if they share hobbies with members of the opposite group. For this reason, we want to develop a system that facilitates pairing members of both groups based on these common interests.

In order to pair patients and volunteers, we have adapted the Gale-Shapley stable marriage algorithm to suit the needs of this project. As a requirement to run the algorithm, each member of each group must rate his or her interests based on a list of predefined options. The algorithm uses these ratings to calculate a percentage of similarity between members of each group. As a result, preference lists are generated for each member of a group with members of the opposite group; these lists are used as input for the aforementioned stable marriage algorithm.

The results of the project are the following:

- An Android mobile application that allows patients and volunteers to input their personal information and rate their interests. Using this information, the Gale-Shapley stable marriage algorithm is used to prompt users to pair up with a member of the opposite group which is deemed stable. Several other functionalities of this mobile application are described in the methodology section.*
- A web application that allows the administrators to manage and monitor system data.*

In conclusion, an Android mobile application was developed so that patients and volunteers can interact with one another and, in turn, patients are able to meet volunteers that share common interests.

Keywords: stable marriage, volunteering, stable matching, gale shapley, hospitals, patients

ÍNDICE GENERAL

RESUMEN.....	I
ABSTRACT.....	II
INDICE GENERAL.....	II
INDICE DE FIGURAS.....	V
ÍNDICE DE TABLAS.....	VI
CAPÍTULO 1.....	1
1. INTRODUCCIÓN.....	1
1.1 Antecedentes.....	1
1.2 Descripción del problema.....	1
1.3 Objetivos.....	1
1.3.1 Objetivo General.....	1
1.3.2 Objetivos Específicos.....	1
1.4 Alcance.....	2
1.5 Concepto.....	3
2. METODOLOGÍA.....	4
2.1 Metodología de desarrollo.....	4
2.2 Diagrama de la arquitectura del sistema.....	4
2.3 Diagrama de casos de uso.....	7
2.4 Algoritmo de emparejamiento de Gale y Shapley.....	8
2.4.1 Definición del algoritmo.....	9
2.5 Implementación del algoritmo de matrimonio estable de Gale y Shapley....	10
2.5.1 Entradas del algoritmo.....	10
2.5.2 Procedimiento.....	11
2.5.3 Salida del algoritmo.....	13
2.6 Pruebas de tiempo de ejecución del algoritmo de matrimonio estable de Gale y Shapley.....	13

2.7	Diagrama de base de datos.	14
CAPÍTULO 3		16
3.	IMPLEMENTACIÓN	16
3.1	Aplicación web de administración del sistema.....	16
3.2	Aplicación móvil Android	18
CAPÍTULO 4.....		23
4.	RESULTADOS.....	23
4.1	Perfil de los participantes de las pruebas	23
4.2	Historias de usuarios utilizadas.	23
4.3	Metas de usabilidad.	24
4.4	Resultados de las pruebas de los usuarios interactuando con la aplicación móvil. 27	
4.5	Respuesta de la encuesta de satisfacción.	34
CONCLUSIONES Y RECOMENDACIONES.....		37
	Conclusiones	37
	Recomendaciones	38
BIBLIOGRAFÍA.....		39
ANEXOS.....		40

ÍNDICE DE FIGURAS

Figura 2.1. Diagrama de despliegue.....	4
Figura 2.2. Diagrama de casos de uso.....	8
Figura 2.3. Gráfica tiempo de ejecución.....	14
Figura 2.4. Diagrama de base de datos.....	15
Figura 3.1. Página de inicio de la aplicación web.....	16
Figura 3.2. Diagrama del modelo-vista-controlador.....	17
Figura 3.3: Pantalla de inicio de la aplicación móvil.....	18
Figura 3.4. Consulta de voluntarios gustos similares.....	20
Figura 3.5. Función utilizada para validar que el matrimonio sea estable.....	22
Figura 4.1-4.11: Gráficos tiempos de realización de la actividad.....	27-34
Figura 4.12: Encuesta.....	35
Figura 4.13: Experiencia de los usuarios.....	35

ÍNDICE DE TABLAS

Tabla 2.1. Asignación de voluntarios a pacientes.....	12
Tabla 2.2. Asignación de pacientes a voluntarios.....	13
Tabla 4.1. Metas de usabilidad.....	25

CAPÍTULO 1

1. INTRODUCCIÓN

1.1 Antecedentes

En clínicas y hospitales de la ciudad de Guayaquil, existen fundaciones que se han afiliado para ofrecer el servicio social de voluntariado. Los voluntarios visitan periódicamente estos centros de salud para realizar diferentes actividades con los pacientes, cabe recalcar que cada paciente tiene sus propias actividades que le gusta hacer con el voluntario.

1.2 Descripción del problema

Actualmente no hay forma de saber que gustos tiene el paciente, ni que gustos tiene el voluntario, por lo tanto no se puede realizar algún tipo de emparejamiento basados en aquellos aspectos. Los voluntarios podrían tener un mayor impacto en sus pacientes si coinciden en sus habilidades o preferencias.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un sistema que empareje voluntarios y pacientes basados en sus gustos similares.

1.3.2 Objetivos Específicos

- Implementar el algoritmo de emparejamiento que permita escoger a los pacientes escoger los voluntarios adecuados mediante un emparejamiento de gustos.
- Desarrollar una aplicación web que permita administrar los datos asociados a pacientes, voluntarios, centros médicos, fundaciones e historial de citas.
- Desarrollar una aplicación móvil que permita a los pacientes y voluntarios tener un medio personalizado de interacción.

1.4 Alcance

El sistema estará compuesto de las siguientes aplicaciones:

- Aplicación móvil:

Usuario: Pacientes y voluntarios.

Sistema operativo: Android.

- Módulo de registro del paciente: El paciente podrá registrar en la aplicación sus datos personales y las actividades que le gustaría realizar calificados según la intensidad de agrado, el puntaje será del 1 al 5 donde 1 es el puntaje más bajo y 5 el puntaje más alto.
- Módulo de registro del voluntario: El voluntario podrá registrar en la aplicación sus datos personales y las actividades que le gustaría realizar calificados según la intensidad de agrado, el puntaje será del 1 al 5 donde 1 es el puntaje más bajo y 5 el puntaje más alto.
- Módulo de emparejamiento global: El paciente o voluntario podrá visualizar una lista de las personas que tengan gustos afines, esta lista estará ordenada de mayor a menor similitud.
- Módulo de emparejamiento individual: El paciente o voluntario podrá visualizar diariamente su pareja estable que dio como resultado del algoritmo de matrimonio estable Gale y Shapley descrito en la sección 2.4.
- Módulo de solicitudes: El paciente o voluntario podrá enviar solicitudes de amistad o de citas para realizar cierta actividad, estas solicitudes podrán ser rechazadas o aceptadas.
- Módulo de actividades: El paciente o voluntario podrá visualizar las actividades que tienen agendadas, estas actividades podrán ser canceladas.
- Módulo de calificaciones: El paciente podrá calificar al voluntario para que los administradores puedan conocer la opinión del paciente acerca de su grado de satisfacción.
- Módulo de chat: Los pacientes y voluntarios emparejados podrán comunicarse en tiempo real entre ellos mediante

mensajes de texto o imágenes, de esta manera la interacción entre usuarios será mejor, ya que se tendrá un medio de comunicación propio de la aplicación.

- Aplicación web:

Usuario: Administrador del sistema.

- Módulo de solicitudes de registro de voluntarios: El usuario podrá aceptar o rechazar solicitudes de registro.
- Módulo de administración de pacientes: El usuario podrá realizar operaciones tales como crear, editar, consultar y eliminar sobre los pacientes.
- Módulo de administración de voluntarios: El usuario podrá realizar operaciones tales como crear, editar, consultar y eliminar sobre los voluntarios.
- Módulo de fundaciones: El usuario podrá realizar operaciones tales como crear, editar, consultar y eliminar sobre las fundaciones.
- Módulo de centros médicos: El usuario podrá realizar operaciones tales como crear, editar, consultar y eliminar sobre los centros médicos.
- Módulo de citas: El usuario podrá visualizar información de las citas que se han agendado entre pacientes y voluntarios.
- Módulo de calificaciones: El usuario podrá visualizar las calificaciones que le proporcionaron los pacientes a los voluntarios.

1.5 Concepto

La aplicación para emparejamiento automático de pacientes y voluntarios con preferencias similares está conformada por una aplicación móvil y web, donde en la aplicación móvil los pacientes pueden interactuar con voluntarios que tengan gustos, habilidades o preferencias en común, también está conformada por una aplicación web que permite a las fundaciones administrar a sus voluntarios y tener información de los pacientes asociados a ellos.

CAPÍTULO 2

2. METODOLOGÍA

2.1 Metodología de desarrollo.

Para el desarrollo de la aplicación, diseño e implementación de aplicación para emparejamiento automático de pacientes y voluntarios con preferencias similares se utilizó la metodología SCRUM.

SCRUM es una metodología ágil que facilita la administración de proyectos. La planificación del proyecto se la divide en varias metas llamadas sprints. Al finalizar un sprint, se entrega al cliente un avance funcional del proyecto. El cliente valida el entregable y si se requiere algún cambio o modificación de requerimientos en dicho entregable, estos cambios se los agrega como tareas al siguiente sprint. De esta forma se asegura que el cliente se encuentre satisfecho al finalizar su aplicación ya que se cumplieron con sus requerimientos.

2.2 Diagrama de la arquitectura del sistema

A continuación, se describen los componentes de la aplicación en un diagrama de despliegue:

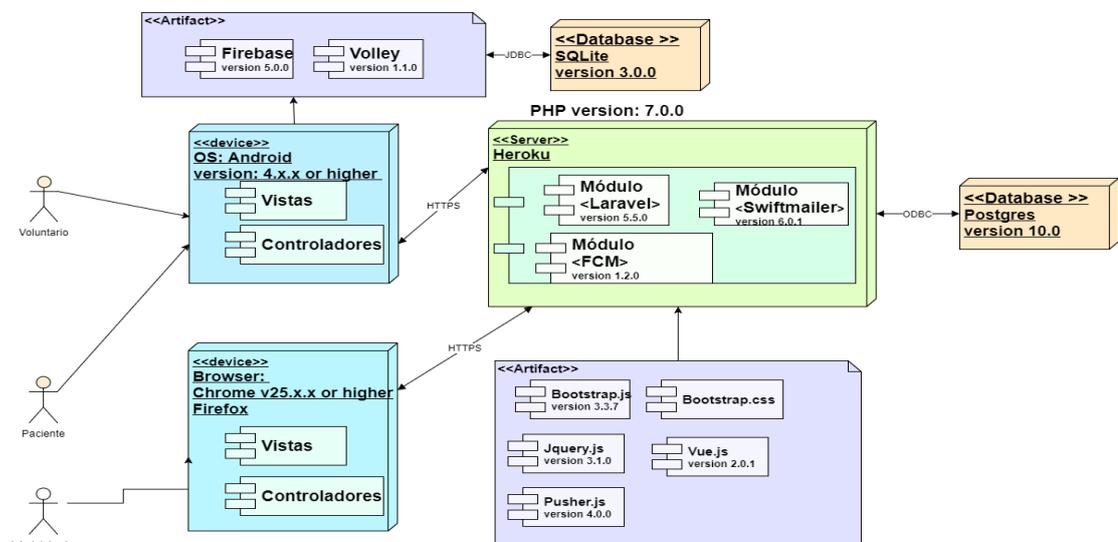


Figura 2.1 Diagrama de despliegue. Elaboración propia.

2.2.1 Descripción de los componentes de la aplicación móvil Android:

Sistemas operativo Android: Sistema operativo de dispositivos móviles con mayor importancia en el mercado según [1]. Las razones por la que

se escogió desarrollar la aplicación sobre este sistema operativo móvil son las siguientes:

- Los desarrolladores tienen acceso completo a las APIs del framework y su arquitectura está diseñada para simplificar la reutilización de componentes, lo que es útil para el desarrollo de la aplicación ya que se necesitará utilizar el manejador de imágenes o galería de fotos para que el paciente o voluntario escoja su foto de perfil.
- Android utiliza SQLite para el almacenamiento de datos, lo que va a permitir almacenar datos de la aplicación sin necesidad de instalar o configurar alguna otra base de datos.
- El IDE escogido para implementar la aplicación móvil Android es Android Studio, este IDE incluye emuladores de dispositivos móviles, herramientas para depuración de memoria y análisis del rendimiento del software, lo que facilita el desarrollo de la aplicación y el control de errores.

Los servicios que se escogieron para el desarrollo de la aplicación móvil Android son las siguientes:

Firestore: Firestore es una plataforma de desarrollo creada por Google. Los desarrolladores tienen acceso a una gama de herramientas orientadas a la creación de aplicaciones Android de alta calidad, las herramientas de Firestore que se escogieron se pueden utilizar gratuitamente hasta ciertas condiciones que se detallan en [5], se utilizaron las siguientes herramientas:

- Firestore Storage: Es una herramienta de almacenamiento de archivos en la nube. Se utilizó esta herramienta con el fin de almacenar la foto de perfil que sube el paciente y el voluntario cuando se registra en la aplicación móvil.
- Firestore Cloud Messaging: Es una herramienta que permite enviar notificaciones desde un servidor web a dispositivos móviles o de un dispositivo móvil a otro dispositivo móvil. Para enviar la notificación el servidor del sistema envía un requerimiento a Firestore, este requerimiento contiene el mensaje y la lista de los identificadores de

los usuarios a los cuales se les debe de enviar una notificación, Firebase valida que esos identificadores pertenecen a los dispositivos registrados en el sistema y procede a enviar las notificaciones a los mismos. Se utilizó esta herramienta para enviar notificaciones a los dispositivos móviles de pacientes y voluntarios cuando se envíen mensajes a través del chat de la misma.

- **Firestore Realtime Database:** Es una herramienta que almacena y sincroniza datos utilizando una base de datos NoSQL alojada en la nube. Los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente conectado y se mantienen disponibles aun cuando la aplicación móvil está sin conexión a internet. Se utilizó esta herramienta para que los pacientes y voluntarios emparejados puedan comunicarse de manera personal.

2.2.2 Descripción de los componentes de la aplicación web:

Laravel: Es un framework de código, se utilizó este framework para desarrollar la aplicación porque permite crear una API (interfaz de programación de aplicaciones), La API es consumida tanto por aplicación web del administrador y por la aplicación móvil de voluntarios y pacientes.

Laravel cuenta con todas las dependencias o librerías de código abierto necesarias para crear la aplicación. Las librerías que se escogieron son las siguientes:

- **Firebase Cloud Messaging PHP (FCM):** Es una librería implementada para enviar notificaciones desde el servidor web a los dispositivos móviles, su forma de enviar notificaciones es igual a la de Firebase Cloud Messaging, la cual se describe anteriormente. Se utilizó esta herramienta para enviar notificaciones de solicitudes de amistad y solicitudes de aceptación, rechazo y cancelación de citas.
- **SwiftMailer:** es una librería desarrollada para enviar mensajes de correo electrónico. Se utilizó esta herramienta en la aplicación web para que los pacientes y voluntarios reciban correos relativos a aceptación o rechazo de solicitudes de registro, activaciones de cuentas y cambios de contraseña.

- Pusher: Es un servicio en línea que encapsula la implementación de Websockets. Se utilizó esta herramienta en la aplicación web del administrador, ya que se tienen que enviar notificaciones en tiempo real a la misma cada vez que se registra un voluntario en la aplicación móvil.

2.2.3 Descripción de la base de datos del sistema

Base de datos PostgreSQL: Es un sistema de gestión de base de datos relacionales, se escogió esta base de datos porque está desarrollada bajo licencia de código abierto, es multiplataforma, e implementa el método de Control de Concurrencias Multiversión (o por sus siglas en inglés MVCC). MVCC ayuda a tener un mejor rendimiento en cuanto a velocidad, especialmente cuando la base de datos es grande y se tienen que hacer muchas operaciones en la base de datos para realizar una determinada transacción. Este gestor de base de datos permitirá que la aplicación se mantenga con un rendimiento normal a pesar de que los volúmenes de datos aumenten.

2.3 Diagrama de casos de uso

La Figura 2.8 representa las todas funcionalidades que los actores del sistema en este caso paciente, voluntario y administrador pueden realizar en el sistema.

Los casos de uso consultar emparejamiento individual y consultar emparejamiento global se describe a continuación:

- Consulta de emparejamiento individual: Se visualiza a la persona idónea para la persona que está consultado, esta persona idónea fue el resultado de la ejecución del algoritmo de Gale y Shapley (descrito en la sección 2.4) y se la recomienda mediante una notificación diaria que llega a la aplicación móvil del paciente o voluntario.
- Consulta de emparejamiento global: Se visualiza una lista de pacientes y voluntarios, esta lista se encuentra ordenada de mayor a menor porcentaje de preferencia, dicho porcentaje de preferencia se calcula con la ayuda de las calificaciones que el paciente o voluntario ingreso desde la aplicación móvil. El procedimiento se describe en la sección 2.5.2 en los pasos del 1 al 6.

matrimonio se prefiere mutuamente, es decir que ningún miembro de alguna pareja prefiere estar con algún otro miembro de otra pareja.

Las razones por la que se escogió este algoritmo son las siguientes:

- El resultado del algoritmo de matrimonio estable Gale y Shapley es el indicado para cumplir el objetivo del proyecto, ya que cada paciente tendrá como resultado al voluntario que tenga más afinidad con él, donde la afinidad se define como las similitudes de gustos que existen entre aquel paciente y aquel voluntario, los detalles del algoritmo se encuentra en la sección 2.5.
- La complejidad del tiempo de ejecución del algoritmo es de $O(n^2)$ pero las operaciones que realizan son comparaciones, por lo cual el algoritmo dará resultados en un tiempo corto incluso con una gran cantidad de usuarios, las pruebas de tiempo de ejecución se presentan en la sección 2.6.

2.4.1 Definición del algoritmo

En 1962, David Gale y Lloyd Shapley definieron su algoritmo de matrimonio estable como [2]:

Dado n , número de hombres y mujeres, siempre es posible resolver el problema del matrimonio estable y realizar un matrimonio estable, donde n es igual a la cantidad de hombres y mujeres.

Terminología:

Conjunto A: Personas que proponen matrimonio a las personas que conforman el conjunto B.

Conjunto B: Personas que reciben propuestas de matrimonio de las personas que conforman el conjunto A.

X: Elemento del conjunto A.

Y: Elemento del conjunto B.

El algoritmo cumple con las siguientes condiciones:

Dadas las listas ordenadas de preferencias de cada elemento de un conjunto sobre los elementos del otro conjunto. Un matrimonio se

considera inestable si se cumple la siguiente condición para cualquier par de elementos (X, Y) :

- a. Existe un elemento X , perteneciente a A , que prefiere a un elemento Y , perteneciente a B , sobre el elemento actualmente emparejado con X .

El matrimonio se considera estable cuando no existe ningún par de elementos (X, Y) , tales que X y Y se prefieran mutuamente sobre sus parejas actuales.

El algoritmo es el siguiente:

- Durante la primera ronda, cada elemento del primer conjunto A , le propone matrimonio a su mayor preferencia en el conjunto B . Cada elemento del conjunto B responde cada propuesta de matrimonio, siempre quedándose con el elemento del conjunto A que prefiere más. Al finalizar la primera ronda, algunos elementos del conjunto A estarán tentativamente emparejados con elementos del conjunto B .
- En las rondas que siguen, cada elemento del conjunto A que aún no está tentativamente emparejado le propone matrimonio al elemento del conjunto B de mayor preferencia, sin importar si dicho elemento ya está tentativamente emparejado con un elemento del conjunto A . Si el elemento del conjunto B prefiere a su pareja actual, rechaza la propuesta de matrimonio, caso contrario, se genera un nuevo emparejamiento tentativo y el elemento anterior queda sin pareja.
- El proceso se repite hasta que todos los elementos estén emparejados.

2.5 Implementación del algoritmo de matrimonio estable de Gale y Shapley.

A continuación, se describe la adaptación del algoritmo para emparejar pacientes y voluntarios según la similitud de sus gustos.

2.5.1 Entradas del algoritmo.

Las entradas del algoritmo fueron las siguientes:

- Conjunto de voluntarios registrados.
- Conjunto de pacientes registrados.
- Listas de preferencias calificadas para cada conjunto.

Cabe recalcar, que una de las condiciones del algoritmo es que los conjuntos de pacientes y voluntarios a analizarse deben ser del mismo tamaño, para cumplir aquella condición se limitó el tamaño de los conjuntos, aquel tamaño de cada conjunto se lo definió por el tamaño del conjunto con menor cantidad de usuarios (voluntarios o pacientes) registrados.

La alternativa que se da a los elementos que fueron excluidos por haber formado parte del conjunto con mayor cantidad de usuarios se describe en la sección 2.5.3.

2.5.2 Procedimiento.

Paso 1: Limitar la longitud de los conjuntos donde la nueva longitud del arreglo será igual a la longitud del conjunto con menor cantidad de elementos

Paso 2: Escoger un elemento del conjunto de pacientes o de voluntarios.

Paso 3: Escoger un elemento de un conjunto diferente al escogido en el paso anterior.

Paso 4: Obtener la lista de gustos en los que coinciden cada paciente con cada voluntario.

Paso 5: Obtener las calificaciones de la lista de gustos generada del paso 1, tales calificaciones son del paciente y voluntario. Estas calificaciones son del 1 del 5, siendo 1 la calificación mínima (total desagrado) y 5 la calificación máxima (total agrado).

Paso 6: Obtener el porcentaje de similitud utilizando las diferencias absolutas de los gustos calificados en el paso anterior.

Repetir los pasos del 2 al 6 hasta que se haya hecho un recorrido de todos contra todos.

Paso 7: Asignar a cada paciente los voluntarios de mayor a menor similitud.

Paso 8: Asignar a cada voluntario los pacientes de mayor a menor similitud.

En la Tabla 1.1 se presenta una ilustración de la asignación de la lista de los voluntarios ordenados por orden de preferencia a cada paciente, donde P1, P2, ..., P10 representa un paciente y V1, V2, ..., V10 representa un voluntario.

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
V1	V1	V1	V1	V6	V1	V7	V2	V9	V3
V6	V9	V6	V2	V1	V9	V8	V8	V1	V9
V10	V4	V2	V3	V2	V3	V2	V9	V2	V2
V7	V6	V4	V6	V3	V6	V9	V4	V7	V1
V5	V8	V5	V5	V4	V2	V4	V6	V6	V7
V2	V3	V7	V9	V7	V7	V5	V1	V4	V6
V9	V2	V9	V10	V9	V4	V1	V5	V8	V4
V3	V5	V3	V4	V5	V8	V3	V3	V10	V5
V8	V10	V10	V7	V8	V10	V6	V10	V3	V8
V4	V7	V8	V8	V10	V5	V10	V7	V5	V10

Tabla 2.1 Ejemplo de listas de preferencias de los pacientes. Elaboración propia.

En la Tabla 1.2 se presenta una ilustración de la asignación de la lista de los pacientes ordenados por orden de preferencia a cada voluntario, donde P1, P2, ..., P10 representa un paciente y V1, V2, ..., V10 representa un voluntario.

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
P6	P8	P10	P3	P4	P5	P1	P8	P6	P1
P4	P5	P5	P6	P6	P6	P5	P6	P10	P3
P1	P2	P6	P5	P2	P10	P8	P7	P8	P2
P9	P10	P3	P2	P3	P1	P7	P2	P9	P8
P3	P1	P8	P7	P7	P4	P10	P3	P5	P9
P5	P3	P2	P8	P9	P3	P4	P5	P2	P7
P8	P4	P4	P1	P8	P8	P3	P4	P4	P6
P10	P6	P9	P10	P5	P9	P9	P10	P1	P10
P2	P7	P1	P9	P1	P7	P6	P1	P3	P5
P7	P9	P7	P4	P10	P2	P2	P9	P7	P4

Tabla 2.2 Ejemplos de listas de preferencias de los voluntarios. Elaboración propia.

Paso 9: Aplicar algoritmo de Gale y Shapley, utilizando las matrices obtenidas en el paso 4 y 5.

2.5.3 Salida del algoritmo.

Dado n número de pacientes y voluntarios, se obtiene como resultado n parejas (paciente y voluntario) donde se produjo un matrimonio estable. Las parejas ideales que dieron como resultado de la ejecución del algoritmo de Gale y Shapley se dan a conocer mediante una notificación que llega a la aplicación móvil del paciente y voluntario, cabe recalcar que los elementos que fueron excluidos, ya que formaron parte del conjunto con mayor cantidad de elementos (pacientes o voluntarios) se los incluye en la siguiente ejecución del algoritmo.

2.6 Pruebas de tiempo de ejecución del algoritmo de matrimonio estable de Gale y Shapley.

Se realizaron pruebas para medir el tiempo de ejecución del algoritmo, estas pruebas se realizaron en el servidor local donde se encuentra alojada la implementación del algoritmo en lenguaje PHP. El algoritmo se ejecuta cada vez que llegue un requerimiento al servidor web.

Las pruebas se plasmaron sobre en un gráfico que relaciona el tiempo de ejecución del algoritmo versus cantidad de pacientes y voluntarios.

Resultados de la prueba:

Terminología del gráfico:

n: Representa la cantidad de pacientes y voluntarios con intervalo de 1 hasta 100.

tiempo (ms): Representa el tiempo de ejecución del algoritmo en milisegundos.

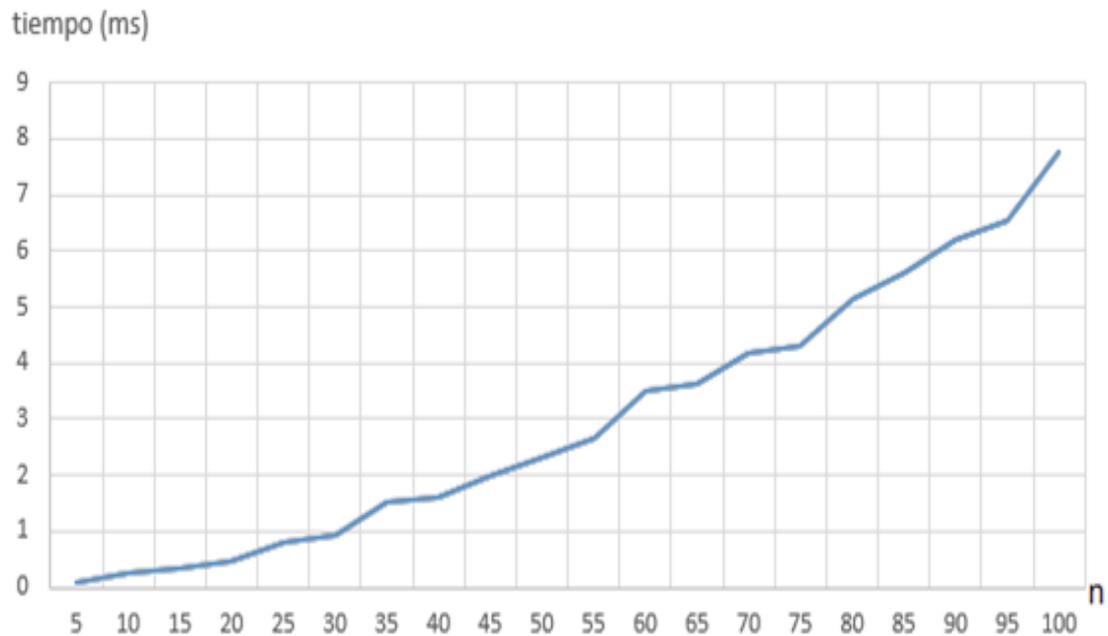


Figura 2.3 Gráfica tiempo (ms) vs n. Elaboración propia.

Según la Figura 2.6 se puede inferir que el algoritmo tiene un comportamiento lineal y para $n = 100$ se demora 7,5 milisegundos, es decir menos de 1 segundo. Los resultados del tiempo de ejecución variaran dependiendo de los recursos y capacidad de procesamiento del servidor.

2.7 Diagrama de base de datos.

La Figura 2.7 representa la base de datos del sistema, esta base de datos consta de 13 donde cada tabla representa un modelo de datos, cada modelo almacena los datos o campos que le corresponden.

Los modelos de datos principales son:

- Pacientes
- Voluntarios
- Fundaciones
- Parejas
- Gustos
- Categorías
- Centros

- Fundaciones

Los datos almacenados en la base de datos sirven para que el administrador pueda consultarlos, editarlos, borrarlos o crearlos desde la aplicación web, a su vez los datos almacenados de los pacientes y voluntarios en el modelo gustos se utilizan en el algoritmo de matrimonio de Gale y Shapley para generar emparejamientos entre ellos.

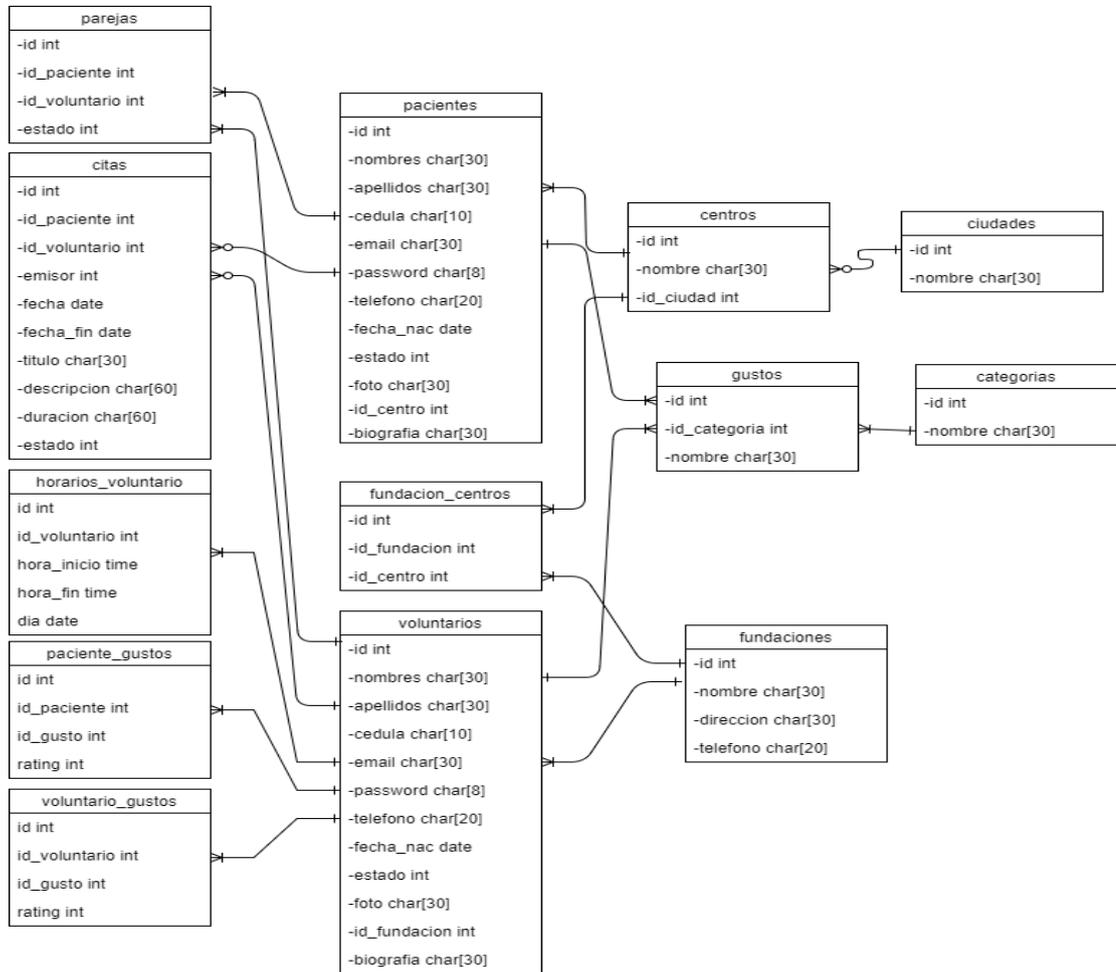


Figura 2.4 Diagrama de la base de datos. Elaboración propia.

CAPÍTULO 3

3. IMPLEMENTACIÓN

3.1 Aplicación web de administración del sistema

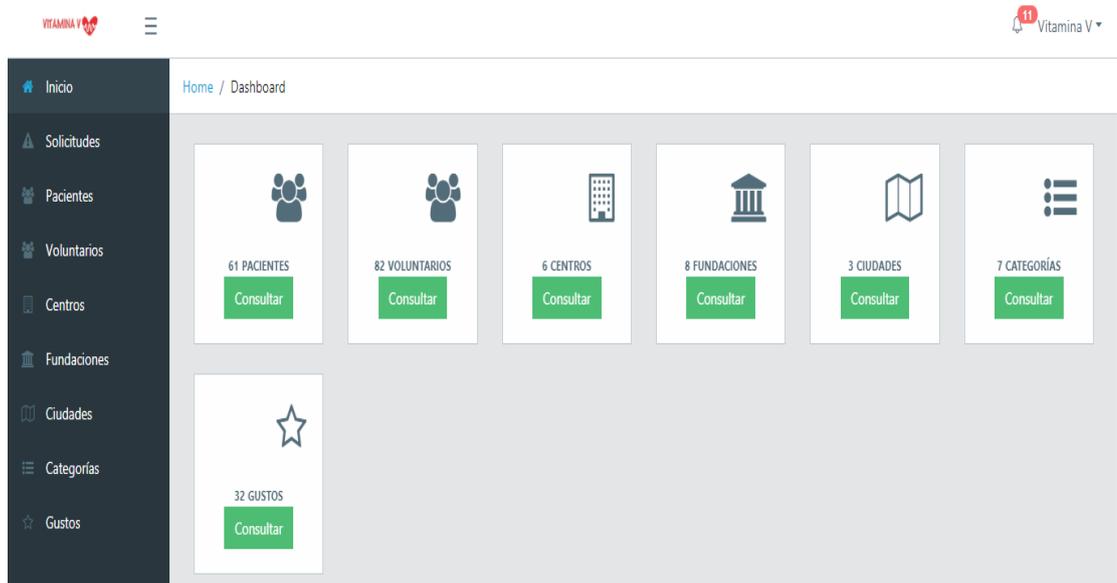


Figura 3.1 Página de inicio de la aplicación web.

La aplicación web se la desarrolló siguiendo el patrón modelo - vista – controlador. Los componentes de este modelo se describen a continuación:

- La capa del modelo: Se encarga de definir los componentes que representan la información.
- La capa de la vista: Es la interfaz de usuario.
- La capa del controlador: Es el medio de interacción entre la vista y el modelo, ya que es el encargado de gestionar las peticiones de los usuarios y de responder la información solicitada con la ayuda tanto del modelo como de la vista [3].



Figura 3.2 Diagrama del modelo-vista-controlador [4].

Se desarrollaron los siguientes módulos:

- Módulo de solicitudes de registro de voluntarios:

Este módulo se lo implementó utilizando la librería Pusher con notificaciones de registros de voluntarios en tiempo real, ya que cada vez que un voluntario se registre, le debe de llegar una notificación al administrador para que este se entere de la solicitud de registro lo más pronto posible, y el administrador pueda aceptar o rechazar dicha solicitud. También, este módulo se lo implemento con utilizando la librería de mensajería Adicionalmente, se implementaron notificaciones vía correo electrónico usando la librería Swiftmailer. Por cada solicitud rechazada o aceptada, se envía un mensaje a la dirección de correo electrónico del voluntario indicando cual fue la respuesta a su solicitud. Si esta fue aceptada le llega un mensaje de activación de cuenta y si fue rechazada le llega un mensaje indicando cual fue el motivo del rechazo. Las capturas de las pantallas se encuentran en la sección de Anexos, Figura A1.1, A1.2, A1.3.
- Módulo de administración de pacientes:

Este módulo fue implementando para que el administrador pueda crear, modificar, consultar y eliminar datos personales de los pacientes, adicionalmente, se puede consultar el historial de citas y sus voluntarios emparejados. Las capturas de las pantallas se encuentran en la sección de Anexos Figura A1.4, A1.5, A1.6.

- **Módulo de administración de voluntarios:**
Este módulo fue implementado para que el administrador pueda crear, modificar, consultar y eliminar datos personales de los voluntarios, adicionalmente, se puede consultar el historial de citas y los voluntarios emparejados. Las capturas de las pantallas se encuentran en la sección de Anexos, Figura A1.7, A1.8, A1.9.
- **Módulo de administración de centros, fundaciones, ciudades, categorías y gustos:**
Estos módulos fueron implementados para que el administrador pueda crear, modificar, consultar y eliminar información asociada a los centros, fundaciones, ciudades, categorías y gustos, las vistas de listas y formularios para estos módulos son similares. Las capturas de las pantallas se encuentran en la sección de Anexos, Figuras A1.10, A1.11.

3.2 Aplicación móvil Android



Figura 3.3 Pantalla de inicio de la aplicación móvil. Elaboración propia.

La aplicación móvil consta de los siguientes módulos:

- **Módulo de inicio de sesión:**
Este módulo permite que el voluntario o paciente se autentifique con las credenciales que fueron ingresadas en el módulo de registro, en este

módulo se utilizó SQLite para almacenar ciertos datos del usuario registrado en la aplicación móvil.

- **Módulo de registro de paciente:**
Este módulo permite que el paciente ingrese sus datos personales incluida una foto de perfil, en este módulo se utilizó Firebase Storage, esta es una herramienta de almacenamiento de archivos en la nube, y se la utilizó con el fin de almacenar la foto de perfil del paciente.
- **Módulo de registro de voluntario:**
Este módulo permite que el voluntario ingrese sus datos personales incluida una foto de perfil, en este módulo se utilizó Firebase Storage, esta es una herramienta de almacenamiento de archivos en la nube, y se la utilizó con el fin de almacenar la foto de perfil del voluntario. A diferencia del paciente, el voluntario tiene que pasar por un proceso de aceptación en el sistema, este proceso fue descrito en el módulo de solicitudes de registro de la aplicación móvil.
- **Módulo de emparejamiento global:**
Este módulo permite que el paciente o voluntario visualice una lista de las personas que tengan gustos afines, los pasos para conseguir el emparejamiento se lo detalla en la sección 2.5.2 en los pasos del 1 al 6.

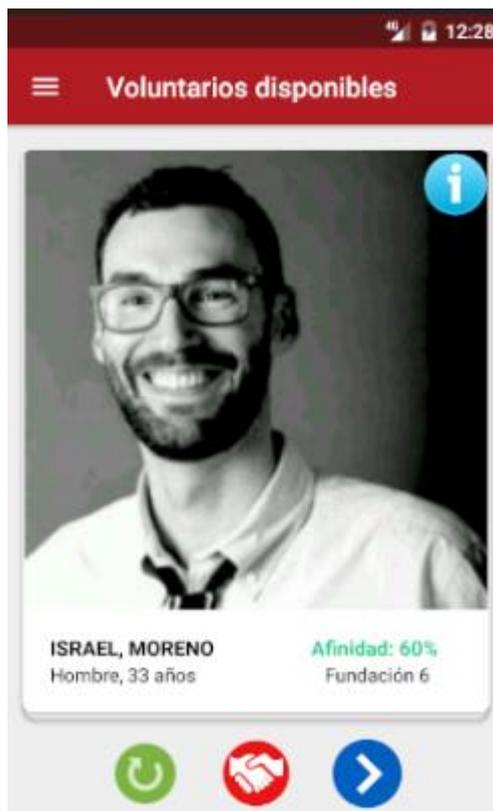


Figura 3.4 Consulta de voluntarios gustos similares. Elaboración propia.

- Módulo de emparejamiento individual:
Este módulo permite que el paciente o voluntario visualice diariamente mediante una notificación alguna recomendación que contiene a la persona ideal, en este módulo se utilizó el algoritmo de matrimonio estable de Gale y Shapley para que recomiende a la persona con más afinidad en gustos, este algoritmo se lo detalla en la sección 2.4.

La implementación se la realizó en lado del servidor, en lenguaje PHP.

Las estructuras de datos utilizadas fueron arreglos y tablas hash. PHP provee diversos métodos que son convenientes al momento de trabajar con arreglos. Los principales métodos que se utilizaron en la implementación fueron:

- `array_combine`: Crea un arreglo con datos clave/valor
`array array_combine (array $keys , array $values)`

Parámetros:

`array $keys`: Arreglo donde sus valores son las claves.

`array $values`: Arreglo donde sus valores son los valores correspondientes a las claves.

- `array_slice`: Extrae una parte de un arreglo.

`array array_slice (array $array , int $offset)`

Parámetros:

`array $array`: El array de entrada.

`int $offset`: Índice en donde se empezará a extraer la posición del arreglo.

- `array_search`: Busca un valor determinado en un arreglo y devuelve la primera clave correspondiente en caso de éxito.

`array_search (mixed $needle , array $haystack)`

Parámetros:

mixed `$needle`: El valor a buscar.

`array $haystack`: El arreglo con el valor a buscar.

En la Figura 3.1a se visualiza como se utilizaron los métodos descritos.

```
public function check ($engaged, $guyprefers, $galprefers)
{
    $inverseengaged = array_combine(array_values($engaged), array_keys($engaged));
    foreach ($engaged as $she => $he)
    {
        $shelikes = $galprefers[$she];
        $shelikesbetter = array_slice($shelikes, 0, array_search($he, $shelikes));
        $helikes = $guyprefers[$he];
        $helikesbetter = array_slice($helikes, 0, array_search($she, $helikes));

        foreach ($shelikesbetter as $guy)
        {
            $guysgirl = $inverseengaged[$guy];
            $guylikes = $guyprefers[$guy];
            if (array_search($guysgirl, $guylikes) > array_search($she, $guylikes))
            {
                printf("<br><b>%%s y %%s se prefieren mas que sus actuales parejas: %%s y %%s, respectivamente.</b><br>",
                    $she, $guy, $he, $guysgirl);
                return false;
            }
        }
        foreach ($helikesbetter as $gal)
        {
            $girlsguy = $engaged[$gal];
            $gallikes = $galprefers[$gal];
            if (array_search($girlsguy, $gallikes) > array_search($he, $gallikes))
            {
                printf("<br><b>%%s y %%s se prefieren mas que sus actuales parejas: %%s y %%s, respectivamente.</b><br>",
                    $he, $gal, $she, $girlsguy);
                return false;
            }
        }
    }
    return true;
}
```

Figura 3.5 Función utilizada para validar que el matrimonio sea estable. Elaboración propia.

En la implementación inicial del algoritmo de matrimonio estable de Gale y Shapley, los recorridos de arreglos o búsquedas se las realizaba de manera lineal e incluso se usaban ciclos anidados, lo que ocasionaba que la ejecución del algoritmo sea lenta, para optimizar el tiempo de ejecución del algoritmo se cambiaron los arreglos por tablas hash.

El tiempo de ejecución de la búsqueda hash es de $O(1)$ (constante), este tiempo promedio de búsqueda es constante sin importar el número de elementos que contenga la tabla hash. Realizada esta optimización el tiempo de ejecución del algoritmo es de aproximadamente de 7,5 milisegundos para 100 pacientes y voluntarios y antes de la optimización era de 10 segundos.

CAPÍTULO 4

4. RESULTADOS.

Las pruebas de experiencia de usuario se las realizó para:

- Validar con los participantes de la prueba si la aplicación móvil cumple con los objetivos propuestos en la sección 1.3.
- Conocer el grado de satisfacción de los participantes de la prueba al interactuar con la aplicación.

4.1 Perfil de los participantes de las pruebas

10 personas con un rango de edad de 15 a 65 años de edad.

4.2 Historias de usuarios utilizadas.

Título: Crear cuenta.

Descripción: Como paciente/voluntario puedo crear una cuenta para poder ingresar a la aplicación.

Título: Consultar personas con gustos similares.

Descripción: Como paciente/voluntario puedo consultar personas con gustos similares para poder enviar solicitudes de amistad.

Título: Enviar solicitud de amistad.

Descripción: Como paciente/voluntario puedo enviar una solicitud de amistad a una persona para entablar una amistad.

Título: Aceptar solicitudes de amistad.

Descripción: Como paciente/voluntario puedo aceptar una solicitud de amistad para comenzar a realizar actividades con esa persona.

Título: Rechazar solicitudes de amistad.

Descripción: Como paciente/voluntario puedo rechazar una solicitud de amistad para no tener contacto con esa persona.

Título: Chatear con amistades.

Descripción: Como paciente/voluntario puedo chatear con mis amistades para tener una mejor una comunicación.

Título: Enviar solicitud de cita.

Descripción: Como paciente/voluntario puedo enviar una solicitud de cita para realizar alguna actividad.

Título: Aceptar solicitudes de cita.

Descripción: Como paciente/voluntario puedo aceptar una solicitud de cita para realizar alguna actividad con esa persona.

Título: Rechazar solicitudes de cita.

Descripción: Como paciente/voluntario puedo rechazar una solicitud de cita.

Título: Consultar citas aceptadas.

Descripción: Como paciente/voluntario puedo consultar mis citas agendadas para conocer que citas están próximas a realizarse.

Título: Cancelar cita aceptada.

Descripción: Como paciente/voluntario puedo cancelar una cita para que la otra persona se entere de que ya no puedo realizar cita.

Título: Consultar perfil.

Descripción: Como paciente/voluntario puedo consultar mi perfil para poder revisar mis datos personales.

Título: Editar perfil.

Descripción: Como paciente/voluntario puedo editar mi perfil para poder actualizar algún dato.

4.3 Metas de usabilidad.

Las metas de usabilidad representan las tareas o actividades que se pueden realizar en la aplicación, las cuales se van a evaluar con los participantes.

Meta	Valor a medir	Criterio de usabilidad	Instrumento de medición	Peor valor aceptable	Valor objetivo	Mejor valor aceptable
Crear cuenta.	Tiempo en realizar la tarea	Eficiencia	Cronómetro	30s	20s	10s
Consultar personas con gustos similares.	Tiempo en realizar la tarea	Eficiencia	Cronómetro	30 s	20s	10s
Enviar solicitud de amistad	Tiempo en realizar la tarea	Eficiencia	Cronómetro	30 s	20s	10s
Aceptar o rechazar solicitudes de amistad	Tiempo en realizar la tarea	Eficiencia	Cronómetro	50 s	40s	30s
Chatear con mis amistades	Tiempo en realizar la tarea	Eficiencia	Cronómetro	30 s	20s	10s
Enviar solicitud de cita	Tiempo en realizar la tarea	Eficiencia	Cronómetro	30 s	20s	10s
Aceptar/rechazar solicitudes de cita	Tiempo en realizar la tarea	Eficiencia	Cronómetro	50 s	40s	30s
Consultar citas aceptadas	Tiempo en realizar la tarea	Eficiencia	Cronómetro	30 s	20s	10s

Cancelar cita aceptada	Tiempo en realizar la tarea	Eficiencia	Cronómetro	30 s	20s	10s
Consultar mi perfil	Tiempo en realizar la tarea	Eficiencia	Cronómetro	30 s	20s	10s
Editar mi perfil	Tiempo en realizar la tarea	Eficiencia	Cronómetro	30 s	20s	10s

Tabla 4.1 Metas de usabilidad. Elaboración propia.

El instrumento de medición es el cronómetro ya que se debe tomar el tiempo que ocupa el participante en realizar la tarea o actividad, los rangos de tiempos establecidos en la Tabla 4.3 fueron definidos por experiencias anteriores en cuanto a lo que una persona con conocimientos escasos de tecnología puede demorarse en realizar una determinada actividad en alguna aplicación móvil, a continuación se presenta la tabla de metas de usabilidad.

4.4 Resultados de las pruebas de los usuarios interactuando con la aplicación móvil.

A continuación se presentan en un diagrama de barras los resultados por cada meta o actividad que los participantes de la prueba realizaron. Las fotos de los participantes interactuando con la aplicación se encuentran en la sección Anexos A2.

Meta 1: Crear cuenta.

Criterio medido: Eficiencia.

Resultados:

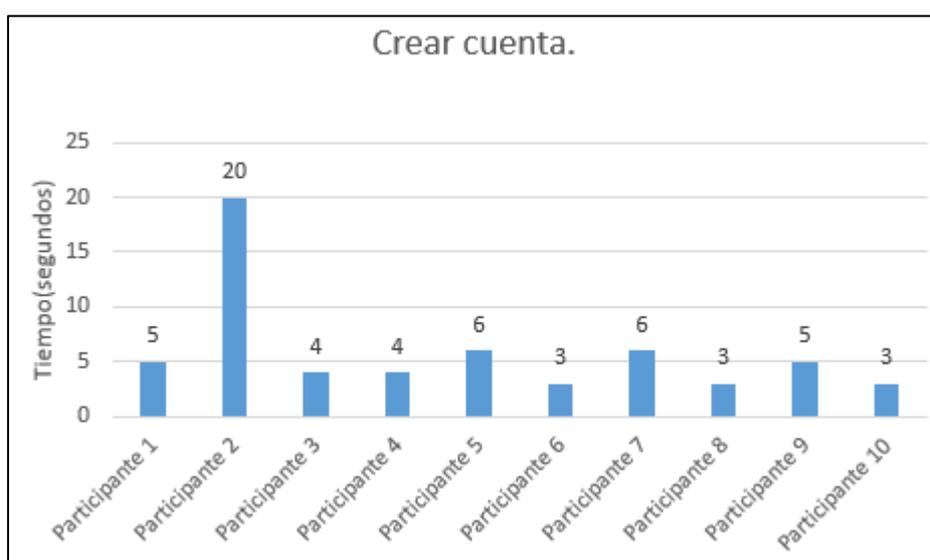


Figura 4.1: Tiempos de realización de la actividad. Elaboración propia.

Peor valor aceptable: 30 segundos.

Mejor valor aceptable: 10 segundos.

Tiempo objetivo: 20 segundos.

Tiempo promedio obtenido: 5.9 segundos.

Meta 2: Consultar personas con gustos similares.

Criterio medido: Eficiencia.

Resultados:

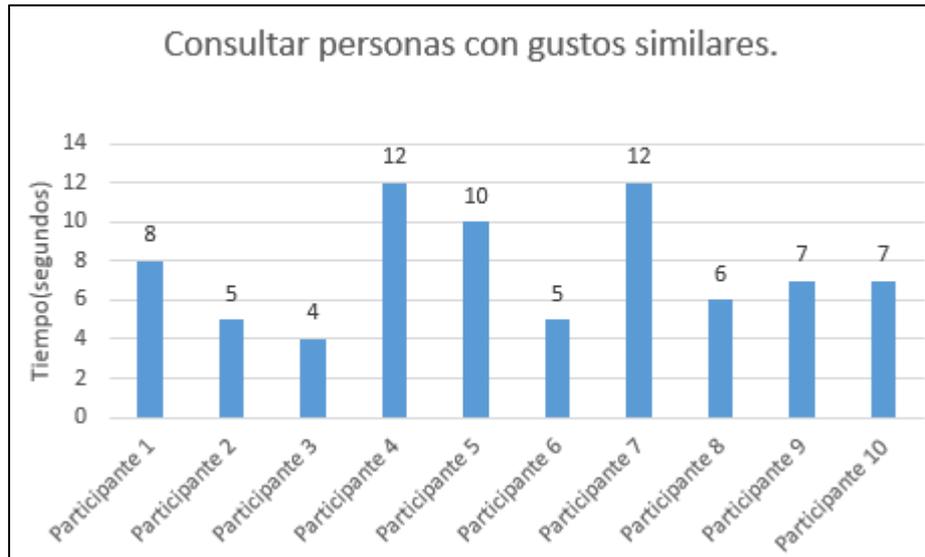


Figura 4.2: Tiempos de realización de la actividad. Elaboración propia.

Peor valor aceptable: 30 segundos.

Mejor valor aceptable: 10 segundos.

Tiempo objetivo: 20 segundos.

Tiempo obtenido: 7.6 segundos.

Meta 3: Enviar solicitud de amistad.

Criterio medido: Eficiencia.

Resultados:

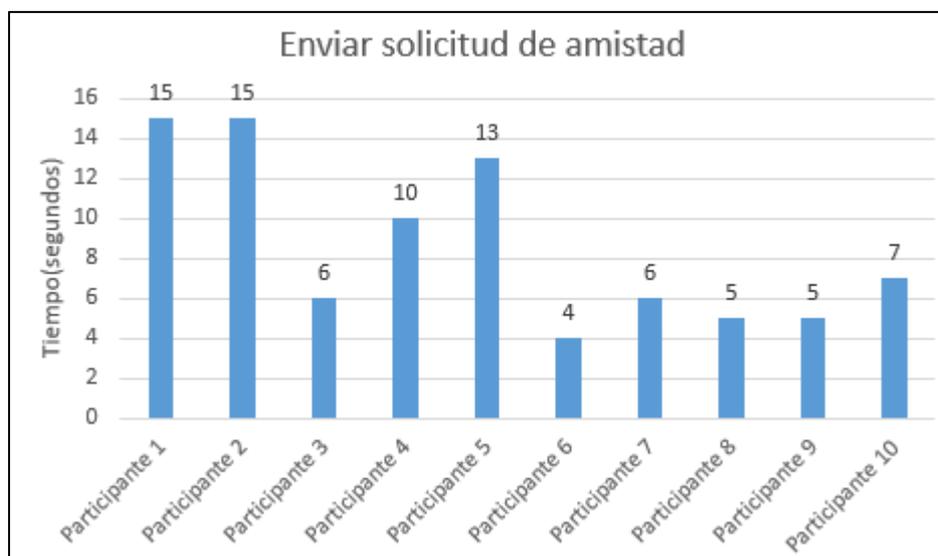


Figura 4.3 Tiempos de realización de la actividad. Elaboración propia.

Peor valor aceptable: 30 segundos.

Mejor valor aceptable: 10 segundos.

Tiempo objetivo: 20 segundos.

Tiempo obtenido: 8.6 segundos.

Meta 4: Aceptar o rechazar solicitud de amistad.

Criterio medido: Eficiencia.

Resultados:

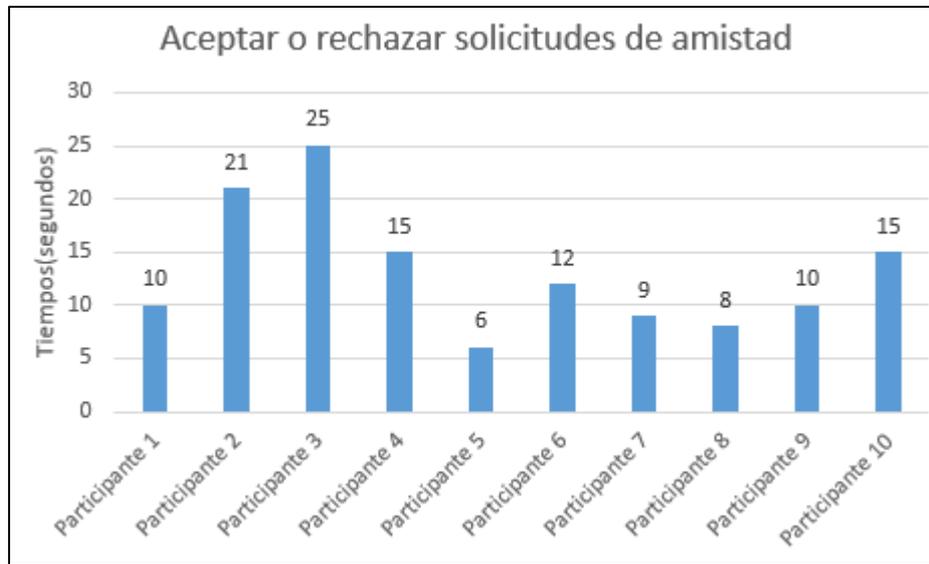


Figura 4.4 Tiempos de realización de la actividad. Elaboración propia.

Peor valor aceptable: 50 segundos.

Mejor valor aceptable: 30 segundos.

Tiempo objetivo: 40 segundos.

Tiempo obtenido: 13,1 segundos.

Meta 5: Chatear con amistades.

Criterio medido: Eficiencia.

Resultados:

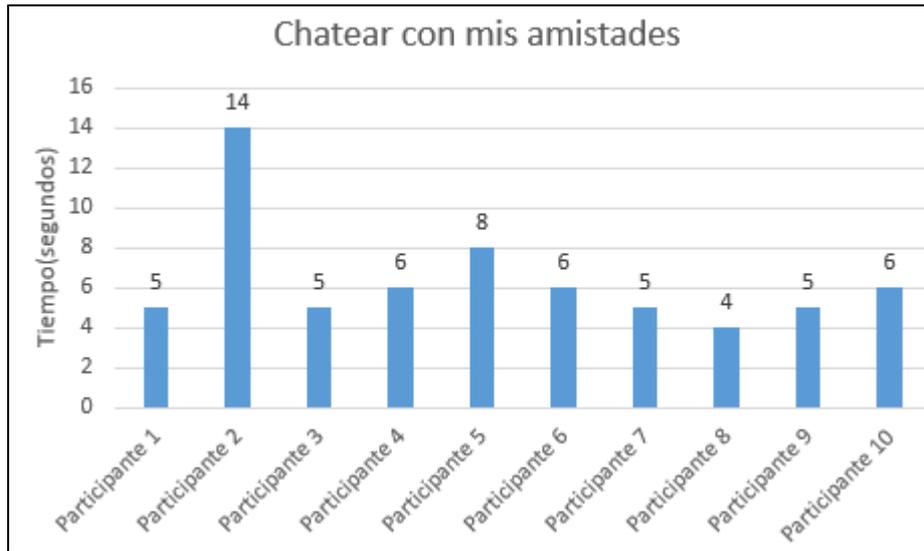


Figura 4.5 Tiempos de realización de la actividad. Elaboración propia.

Peor valor aceptable: 30 segundos.

Mejor valor aceptable: 10 segundos.

Tiempo objetivo: 20 segundos.

Tiempo obtenido: 6,4 segundos.

Meta 6: Enviar solicitud de cita.

Criterio medido: Eficiencia.

Resultados:

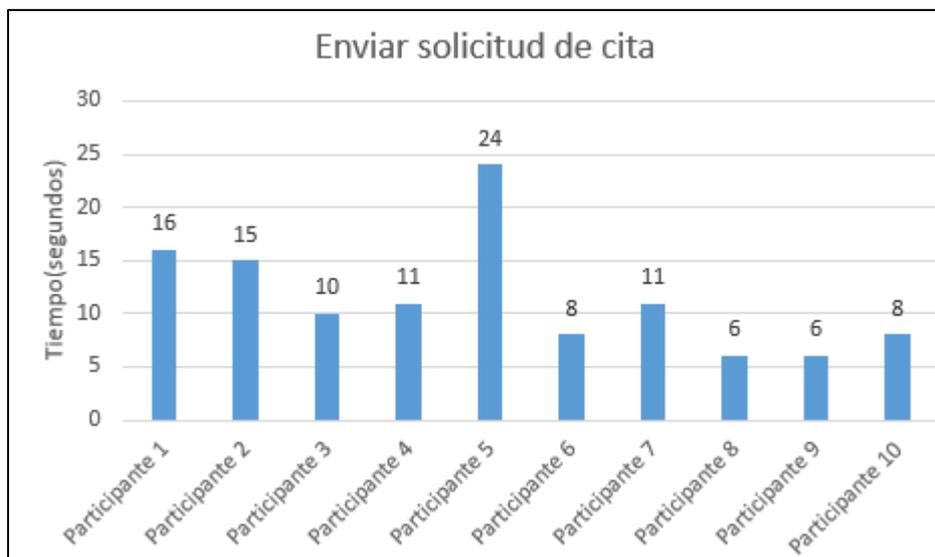


Figura 4.6 Tiempos de realización de la actividad. Elaboración propia.

Peor valor aceptable: 30 segundos.

Mejor valor aceptable: 10 segundos.

Tiempo objetivo: 20 segundos.

Tiempo obtenido: 11,5 segundos.

Meta 7: Aceptar o rechazar solicitud de cita.

Criterio medido: Eficiencia.

Resultados:

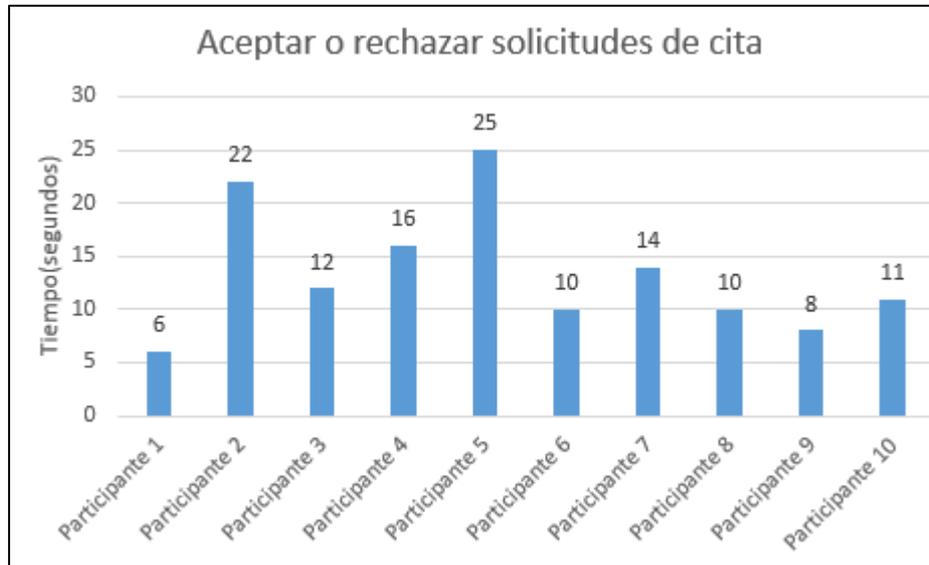


Figura 4.7 Tiempos de realización de la actividad. Elaboración propia.

Peor valor aceptable: 50 segundos.

Mejor valor aceptable: 30 segundos.

Tiempo objetivo: 40 segundos.

Tiempo obtenido: 13,4 segundos.

Meta 8: Consultar citas aceptadas.

Criterio medido: Eficiencia.

Resultados:



Figura 4.8 Tiempos de realización de la actividad. Elaboración propia.

Peor valor aceptable: 30 segundos.

Mejor valor aceptable: 10 segundos.

Tiempo objetivo: 20 segundos.

Tiempo obtenido: 15,2 segundos.

Meta 9: Cancelar cita aceptada.

Criterio medido: Eficiencia

Resultados:

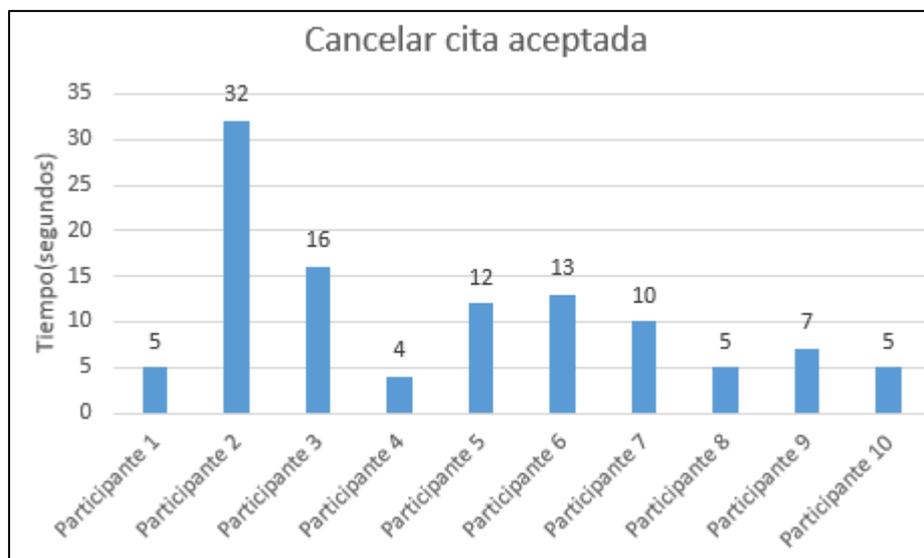


Figura 4.9 Tiempos de realización de la actividad. Elaboración propia.

Peor valor aceptable: 30 segundos.

Mejor valor aceptable: 10 segundos.

Tiempo objetivo: 20 segundos.

Tiempo obtenido: 10,9 segundos.

Meta 10: Consultar perfil.

Criterio medido: Eficiencia.

Resultados:

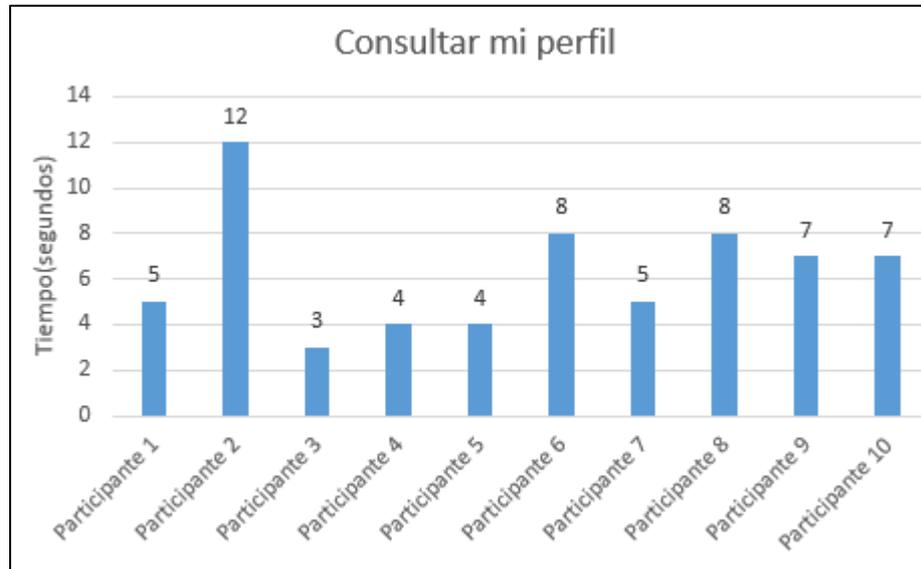


Figura 4.10 Tiempos de realización de la actividad. Elaboración propia.

Peor valor aceptable: 30 segundos.

Mejor valor aceptable: 10 segundos.

Tiempo objetivo: 20 segundos.

Tiempo obtenido: 6,3 segundos.

Meta 11: Editar perfil.

Criterio medido: Eficiencia.

Resultados:



Figura 4.11 Tiempos de realización de la actividad. Elaboración propia.

Peor valor aceptable: 30 segundos.

Mejor valor aceptable: 10 segundos.

Tiempo objetivo: 20 segundos.

Tiempo obtenido: 10,3 segundos.

4.5 Respuesta de la encuesta de satisfacción.

Se realizaron encuestas a los participantes de las pruebas después de realizar las pruebas de interacción con la aplicación móvil, esta encuesta se la realizó por los siguientes motivos:

- Conocer las fortalezas y debilidades de la aplicación.
- Conocer sugerencias del usuario.
- Conocer el grado de satisfacción que tuvo el usuario al interactuar con la aplicación móvil.

Las preguntas de las cuales estaba conformada la encuesta se las presenta en la Figura 4.5a:

¿Qué tal fue su experiencia en general con la aplicación?

1 2 3 4 5

Mala Excelente

¿Qué tarea se le hizo mas difícil de realizar y por que?

¿Qué tarea se le hizo mas fácil de realizar y por que?

Sugerencias para mejorar la aplicación

Figura 4.12 Encuesta. Elaboración propia.

Pregunta 1: ¿Qué tal fue su experiencia en general con la aplicación?

Resultados:

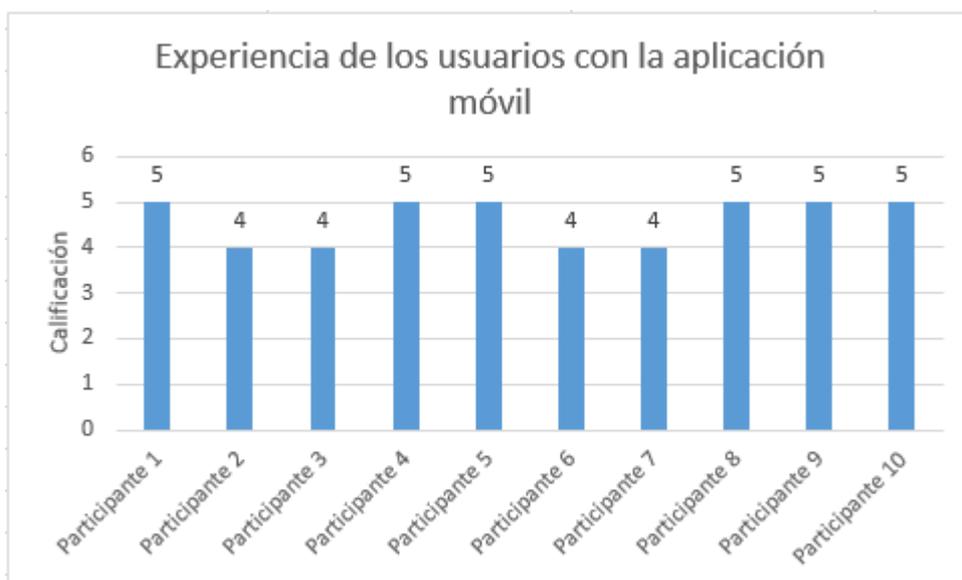


Figura 4.13 Experiencia de los usuarios. Elaboración propia.

Figura 4.5b: Experiencia general de los usuarios de la aplicación.

El 60% de los participantes calificaron a la aplicación con 5/5.

El 40% de los participantes calificaron a la aplicación con 4/5.

Pregunta 2: ¿Qué tarea se le hizo más difícil de realizar y por qué?

Resumen de respuestas:

- Editar perfil, porque el ícono no estaba visible.
- Consultar citas, porque se tomó un poco más de tiempo en encontrarlo.

Pregunta 3: ¿Qué tarea se le hizo más fácil de realizar y por qué?

Resumen de respuestas:

- Ir a perfil, debido a que el ícono del menú es el adecuado.
- Chatear con mis amistades, porque se lo encuentra fácilmente en el menú.

Pregunta 4: Sugerencias para mejorar la aplicación.

Resumen de respuestas:

- Agregar video llamadas y si el mensaje ha sido o no ha sido leído.
- Permitir tomar fotos con la cámara del dispositivo móvil para subirla como foto de perfil en el registro.

CONCLUSIONES Y RECOMENDACIONES

Con el presente proyecto desarrollado, los pacientes y los voluntarios podrán interactuar de mejor manera, ya que estos comparten gustos y por ende el voluntario tendrá un mayor impacto anímico en el paciente, ya que haciendo realizando actividades que disfruten ambos, la estadía del paciente en el centro médico será un poco más llevadera o tolerable.

Como trabajo futuro, en cuanto al resultado del algoritmo de matrimonio estable de Gale y Shapley, este se puede mejorar para que el algoritmo no solo utilice los gustos del usuario sino que también utilice otra fuente de datos del usuario, por ejemplo, los registros de las conversaciones que realiza el usuario en el módulo del chat que provee la aplicación móvil, estos registros pueden ser procesados y analizados de tal manera que se obtenga información vital que sirva para mejorar el resultado del algoritmo.

Conclusiones

Se diseñó e implementó una aplicación móvil donde los pacientes de un centro médico y voluntarios de una fundación pueden realizar un emparejamiento con personas que tengan gustos similares debido a que se implementó en el lado del servidor el algoritmo de matrimonio estable de Gale y Shapley adaptado para la selección de parejas (paciente-voluntario) de acuerdo a los gustos de cada uno. En cuanto a las características del algoritmo implementado se destaca su tiempo de ejecución, debido a que se usaron las estructuras de datos adecuadas (tablas hash). Aplicando tablas hash se logró implementar un algoritmo que se ejecuta en 7,5 milisegundos para 100 pacientes y 100 voluntarios registrados en la base de datos del sistema, cabe mencionar que la implementación con arreglo, el tiempo de ejecución del algoritmo era de 10 segundos para 100 pacientes y 100 voluntarios registrados en la base de datos. Las pruebas de tiempo de ejecución del algoritmo se las realizaron en el servidor local y que el tiempo de ejecución del algoritmo dependerá de los recursos y capacidad de procesamiento del servidor.

Entre los módulos esenciales de la aplicación móvil se tiene el módulo de emparejamiento global e individual, el módulo envío y recepción de solicitudes de citas y el módulo de chat, con la ayuda de estos módulos los usuarios de la aplicación

tendrán una interacción personalizada y de esta manera los voluntarios tendrán un mayor impacto en sus pacientes ya que tienen determinados gustos en común.

También se diseñó e implementó una aplicación web donde el administrador del sistema puede manejar la información registrada en la base de datos.

En cuanto a las pruebas de interacción entre el usuario y la aplicación móvil, estas dieron resultados que todas las actividades o tareas propuestas a los usuarios están dentro del rango entre el valor objetivo y el mejor valor aceptable lo que quiere decir que la aplicación logra transmitir al usuario de manera adecuada y sencilla lo que él puede realizar en la aplicación.

Como trabajo futuro se puede implementar las funcionalidades sugeridas por los usuarios en la encuesta que se describe en la sección 4.6, entre las más relevantes para la aplicación está el desarrollo de video llamadas para que las citas no sólo sean presenciales sino también virtuales, también en la parte de registro además de que la aplicación permita escoger una foto de perfil de las existentes en el dispositivo móvil que también permita tomar fotos con la cámara del mismo.

Recomendaciones

Se recomienda no quedarse con las versiones gratuitas de las herramientas sino comprar versiones de pago para que a futuro, cuando la base de datos crezca, no se de ningún tipo de problema.

BIBLIOGRAFÍA

- [1] M. A., J. Parvez, "Why Android: A Case Study of Smartphone Operating Systems", Department of Computer Science, University of Kashmir. [En línea]. Disponible en: [http://beta.nielit.gov.in/sites/default/files/CPT_7_Muneer%20\(1\).pdf](http://beta.nielit.gov.in/sites/default/files/CPT_7_Muneer%20(1).pdf) [Accedido: Enero 13, 2018]
- [2] D. Gusfield and R. Irving, "The Stable marriage problem - Structure and Algorithms", The MIT, 1989. [En línea]. Disponible en: <https://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15251-f10/Site/Materials/Lectures/Lecture21/lecture21.pdf> [Accedido: Enero 13, 2018]
- [3] "Entendiendo el Modelo - Vista - Controlador - 2.x", *Book.cakephp.org*, 2017. [Online]. Disponible en: <https://book.cakephp.org/2.0/es/cakephp-overview/understanding-model-view-controller.html>.
- [4]"Modelo Vista Controlador", *Rodrigogr.com*, 2017. [online]. Disponible en: http://rodrigogr.com/blog/wp-content/uploads/2015/11/1111115_0034_ModeloVista1.png. [Accesado: 17- Dec- 2017].
- [5]"Firebase", *Firebase*, 2018. [online]. Disponible en: <https://firebase.google.com/pricing/?hl=es-419>. [Accesado: 16- Feb- 2018].

ANEXOS

Anexo A1

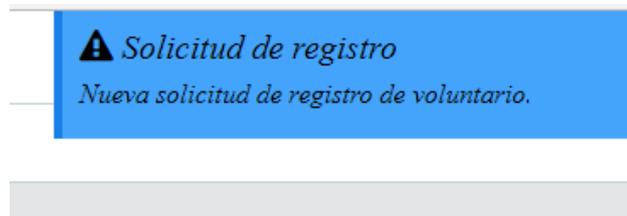


Figura A1.1 Notificación de solicitud de registro. Elaboración propia.

voluntario



Nombres: Joyce	Apellidos: Benitez
Cédula: 0909069213	Correo: evelynbenitez263@gmail.com
Teléfono: 0953653417	Fecha Nac: 1972-12-14
Género: Mujer	Fundación: Fundación 3

Figura A1.2 Formulario para consultar la solicitud de registro. Elaboración propia.

Alerta - Aceptar solicitud de registro x

¿Está seguro que desea aceptar la solicitud de: Joyce Benitez?

Nota: Al aceptar la solicitud de registro, el voluntario podrá iniciar sesión en la aplicación móvil.

Alerta - Rechazar solicitud de registro x

¿Está seguro que desea rechazar la solicitud de registro de: Joyce Benitez?

Nota: Se borrarán todos los datos asociados a este voluntario. Esta acción es irreversible.

Motivo de rechazo:

Figura A1.3 Cuadros de dialogo al aceptar/rechazar una solicitud de registro. Elaboración propia.

[+ Registrar paciente](#)

Pacientes

	Nombres	Apellidos	Correo	Centro	Género	
	<input type="text" value="Filter Nombres"/>	<input type="text" value="Filter Apellidos"/>	<input type="text" value="Filter Correo"/>	<input type="text" value="Filter Centro"/>	<input type="text" value="Filter Género"/>	
1	ANDREA	CEDENO	andrea@gmail.com	Centro 2	F	  
2	ANDREA	JIMENEZ	nqwebpnn@gmail.com	Centro 4	F	  
3	ANDREA	LOOR	lzh8vfh5@gmail.com	Centro 5	F	  

Figura A1.4 Lista de pacientes registrados en el sistema. Elaboración propia.

Paciente



Nombres: ANDREA	Apellidos: CEDENO
Cédula: 1134960006	Correo: andrea@gmail.com
Teléfono: 6733436461	Fecha Nac: 1963-12-23
Género: Mujer	Centro: Centro 2

[Cancelar](#)
[Editar](#)

Figura A1.5 Consulta de un paciente. Elaboración propia.

Emparejamientos

	Voluntario	Fundación	Estado
	<input type="text" value="Filter Voluntario"/>	<input type="text" value="Filter Fundación"/>	<input type="text" value="Filter Estado"/>
1	MARIA GARCIA	Fundación 2	Pendiente
2	DANIELA NUÑEZ	Fundación 2	Pendiente
3	JOYCE GONZALEZ	Fundación 2	Aceptado

Rows per page: 10 ▾ ◀ Prev 1 - 3 of 3 Next ▶

Historial de citas

	Voluntario	Fundación	Actividad	Fecha	Duración (horas)	Estado	Calificación (sobre 5)
	<input type="text" value="Filter Voluntar"/>	<input type="text" value="Filter Fundack"/>	<input type="text" value="Filter Actividad"/>	<input type="text" value="Filter Fecha"/>	<input type="text" value="Filter Duración"/>	<input type="text" value="Filter Estado"/>	<input type="text" value="Filter Calificac"/>
1	JOYCE GONZALEZ	Fundación 2	Cantar	2017-12-24 07:00:00	1	Pendiente	No tiene

Rows per page: 10 ▾ ◀ Prev 1 - 1 of 1 Next ▶

Figura A1.6 Consulta los emparejamientos e historial de citas de un paciente. Elaboración propia.

[+ Registrar voluntario](#)

Voluntarios

	Nombres	Apellidos	Correo	Fundación	Género	
	<input type="text" value="Filter Nombres"/>	<input type="text" value="Filter Apellidos"/>	<input type="text" value="Filter Correo"/>	<input type="text" value="Filter Fundación"/>	<input type="text" value="Filter Género"/>	
1	ADRIANO	FERNANDEZ	c3zsg5po@gmail.com	Fundación 2	M	  
2	ALBERTO	MORALES	6kwug0gl@gmail.com	Fundación 3	M	  
3	ALFONSO	LOOR	y67crigl@gmail.com	Fundación 6	M	  

Figura A1.7 Lista de voluntarios registrados en el sistema. Elaboración propia.

Voluntario



Nombres: ADRIANO	Apellidos: FERNANDEZ
Cédula: 9353393943	Correo: c3zsg5po@gmail.com
Teléfono: 6361518190	Fecha Nac: 1966-12-23
Género: Hombre	Fundación: Fundación 2

[Cancelar](#)
[Editar](#)

Figura A¡Error! No hay texto con el estilo especificado en el documento..8 **Consulta de un voluntario. Elaboración propia.**

Emparejamientos						
Paciente	Centro	Estado				
<input type="text" value="Filter Paciente"/>	<input type="text" value="Filter Centro"/>	<input type="text" value="Filter Estado"/>				
1	FAUSTO FERNANDEZ	Centro 2	Aceptado			
2	MATEO DELGADO	Centro 2	Pendiente			
Rows per page: 10			◀ Prev 1 - 2 of 2 Next ▶			

Historial de citas							
				★★★★★ No tiene calificaciones			
Paciente	Centro	Actividad	Fecha	Duración (horas)	Estado	Calificación (sobre 5)	
<input type="text" value="Filter Paciente"/>	<input type="text" value="Filter Centro"/>	<input type="text" value="Filter Actividad"/>	<input type="text" value="Filter Fecha"/>	<input type="text" value="Filter Duración"/>	<input type="text" value="Filter Estado"/>	<input type="text" value="Filter Calificac"/>	
1	FAUSTO FERNANDEZ	Centro 2	Cantar	2017-12-23 11:00:00	1	Pendiente	0

Figura A Error! No hay texto con el estilo especificado en el documento..9 **Consulta los emparejamientos e historial de citas junto con el promedio de calificaciones dadas por los pacientes. Elaboración propia.**

Centros						
Nombre	Ciudad	Pacientes				
<input type="text" value="Filter Nombre"/>	<input type="text" value="Filter Ciudad"/>	<input type="text" value="Filter Pacientes"/>				
1	Centro 1	Guayaquil	11			
2	Centro 2	Guayaquil	10			
3	Centro 3	Quito	10			

Figura A1.10 Lista de centros registrados en el sistema. **Elaboración propia.**

Centro				
Nombre:	Centro 1		Ciudad:	Guayaquil
<input type="button" value="Cancelar"/>		<input type="button" value="Editar"/>		
Pacientes de Centro 1				
Nombres	Apellidos	Correo	Cédula	
<input type="text" value="Filter Nombres"/>	<input type="text" value="Filter Apellidos"/>	<input type="text" value="Filter Correo"/>	<input type="text" value="Filter Cédula"/>	
1	JOSUE	ROMERO	jkslf9ro@gmail.com	2412385701
2	FAUSTO	VACA	7ta6ezix@gmail.com	1883996125
3	EDDA	VAZQUEZ	sphkfqns@gmail.com	467348557

Figura A1.11 Consulta de un centro y con sus respectivos pacientes. **Elaboración propia.**

Anexo A2



Figura A2.1 Participante de la prueba interactuando con la aplicación.

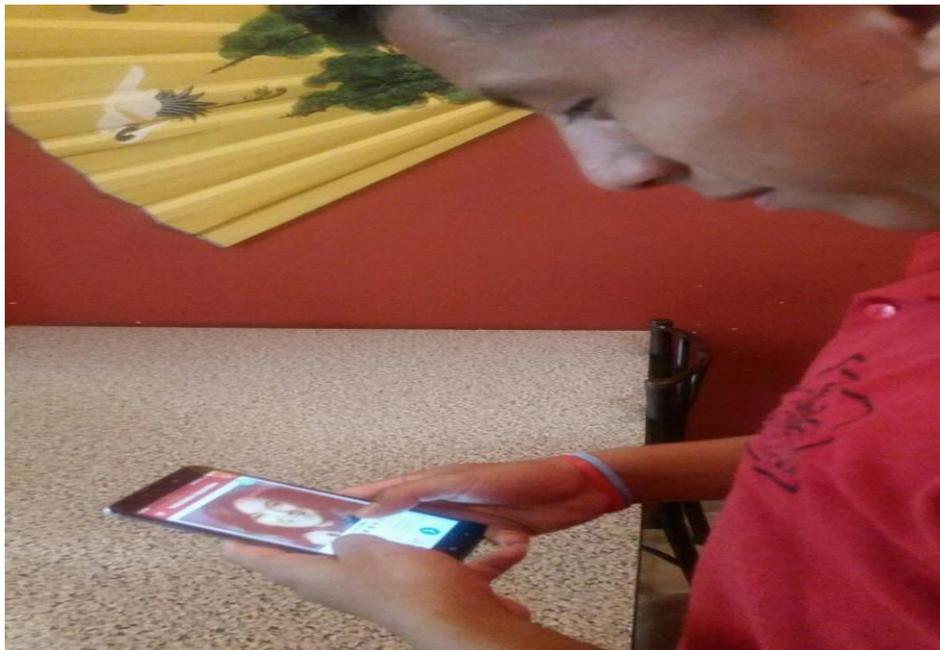


Figura A2.2 Participante de la prueba interactuando con la aplicación.