

UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE CIENCIAS



CURSO : PROGRAMACIÓN EN DISPOSITIVOS MÓVILES

TEMA : APLICACIÓN PARA DONAR A ORGANIZACIONES BENÉFICAS Y/O DONACIÓN DE SANGRE POR CERCANÍA.

DOCENTE : MANUEL CASTILLO

ALUMNOS : TOMAS J. CASAS RODRIGUEZ

DANNY JULCA CARHUAZ

KEVIN ZAMBRANO ROJAS

LEONEL CANAHUALPA

LIMA – PERÚ

2016

INDICE

1-INTRODUCCION	3
2-REQUERIMIENTOS	3
REQUERIMIENTOS FUNCIONALES:	
USABILIDAD	
CONFIABILIDAD	
RENDIMIENTO	
SOPORTE	
3-ANÁLISIS	4
CASOS DE USO	
ESPECIFICACION DE UC	
4- ARQUITECTURA	14
5-DISEÑO	15
PRUEBA	
DESPLIEGUE	
6-PLAN DE DESARROLLO DE SOFTWARE	16
INTRODUCCION8	
PROPOSITO	
ALCANCE	
RESUMEN	
PROPÓSITO DEL PROYECTO, ALCANCE Y OBJETIVO	
PLAN DEL PROYECTO	
PLAN DE FASE	
OBJETIVOS DE LAS ITERACIONES	
FASE DE CONCEPCIÓN	
FASE DE ELABORACIÓN	
FASE DE CONSTRUCCIÓN	
FASE DE TRANSICIÓN	
VERSIONES	
RECURSOS DEL PROYECTO	
PLAN DE STAFF	
PLAN DE CAPACITACIÓN	
PLAN DE CIERRE	
PLAN DEL PROCESO TÉCNICO	
CASO DE DESARROLLO	
MÉTODOS, HERRAMIENTAS Y TÉCNICAS	
PLAN DE SOPORTE DEL PROCESO	
PLAN DE EVALUACION	

APLICACIÓN PARA DONAR A ORGANIZACIONES BENÉFICAS Y/O DONACIÓN SANGRE POR CERCANÍA

1-INTRODUCCION

El objetivo de este documento es explicar las características generales y las especificaciones para el proyecto , para garantizar que todos los requerimientos funcionales, requerimientos del sistema, y otros elementos necesarios para proporcionar una descripción completa y comprensiva de los requerimientos para el software desarrollado.

Este documento también servirá de canal de comunicación entre las distintas partes implicadas en el desarrollo de la aplicación y los miembros definidos en el proyecto deberán tomar parte en su desarrollo.

Una vez aprobado este documento servirá de base al equipo de implementación para la construcción del software que se plantea.

2-REQUERIMIENTOS

Requerimientos funcionales:

- Disponibilidad de base de datos.
- Control de acceso a usuarios debidamente autorizados.
- El sistema deberá almacenar la información personal de cada usuario.
- El sistema deberá mostrar a los usuarios cercanos con el tipo de sangre deseado para poder ser vistos por usuarios registrados y no registrados .
- Podrá ver información detallada de cada usuario , así como su ubicación y disponibilidad de donación.

Ademas de un chat entre usuarios .

- El sistema deberá almacenar los distintos tipos de bancos de sangre, usuarios donantes ,pacientes solicitantes ,doctores y hospitales .
- Los usuarios ya sean registrados o invitados podrán acceder a precios de bancos de sangres , para cotizar la compra por tipo de sangre y cantidad .
- Se genera un documento informativo detallado de la cotización ;o del acuerdo entre usuarios donantes y solicitantes.

Usabilidad

El sistema presenta un listado de hospitales en donde donar , para la selección del hospital solo se necesita ser invitado.

Hay botones para registrarse , buscar donantes , logearse ,gestión de usuario, revisar compatibilidad de la sangre , conversar con otro donante , geolocalización de donante , actualización y mensajería .

Confiabilidad

El acceso a la aplicación se realiza mediante un login que pide usuario y password , donde el password es generado mediante un algoritmo de encriptacion AES , previo registro de usuarios .

El sistema garantiza que solo los usuarios que hayan sido registrados y se haya verificado que los datos de la cuenta ingresada sean verdaderas, puedan mandar a realizar una búsqueda vía web para el donante .

La búsqueda y geolocalizacion del donante es totalmente anónima , mediante nick de usuario .

La lista de hospitales donde se puede realizar donación. son legales y no centros de salud clandestinos .

En caso de usuarios o donantes malintencionados serán dados de baja .

Rendimiento

La aplicación se hará con la arquitectura Modelo -Vista-Controlador para una mejor eficiencia y ejecución .

El consumo de memoria de la aplicación es el menor posible , enfocándose en una interfaz gráfica sencilla pero amigable y realizando una programación optima.

El algoritmo de encriptamiento para las contraseñas de los usuarios asegura que su información quedará segura y libre de piratería por parte de los trabajadores en el sistema.

Se podrá usar apartir de android 4.4 y con memoria mínima de 512 mb .

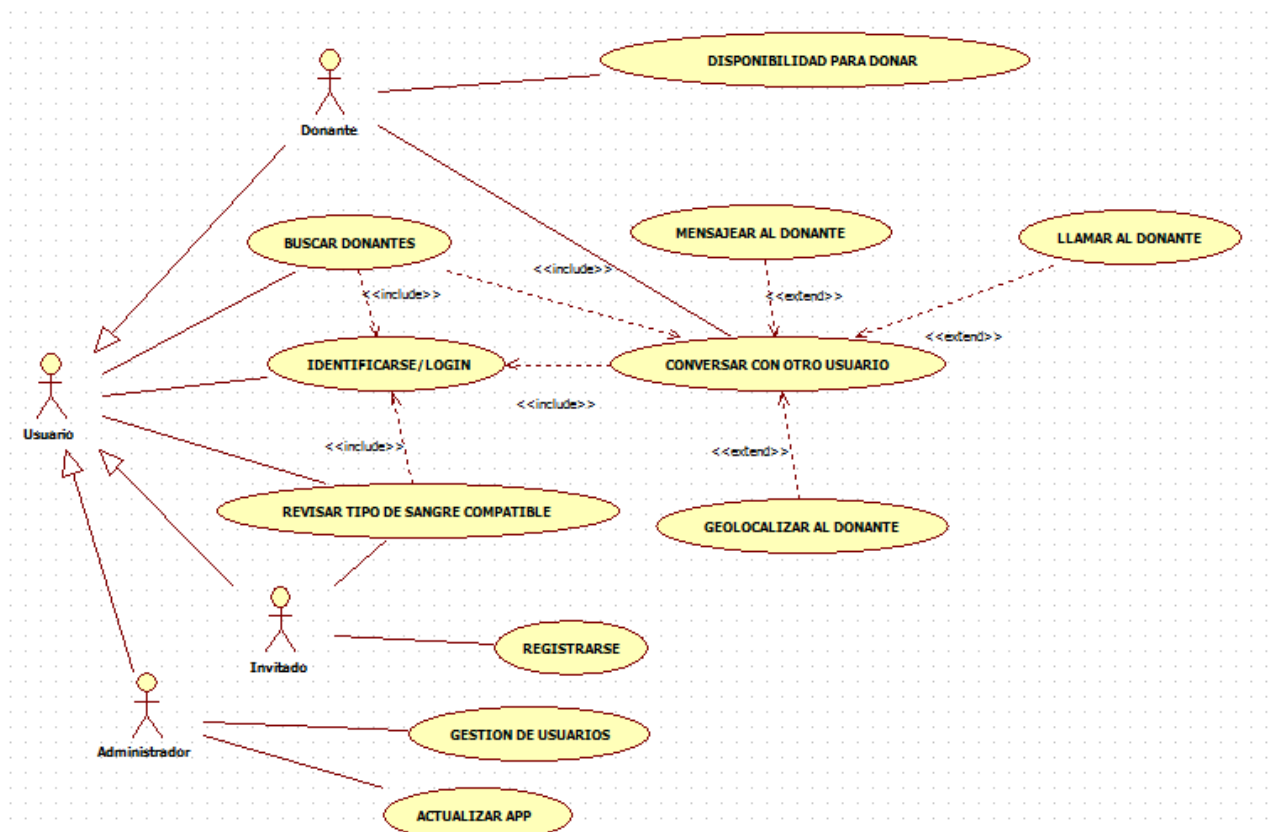
Soporte

El software deberá funcionar en cualquier plataforma móvil apartir de la API 17 ,que tenga un entorno de ejecución Java superior al 1.5 , que cuente con gps y que se puedan hacer llamadas telefónicas.

3-ANÁLISIS

CASOS DE USO

Acá presentaremos los casos de uso para una rápida comprensión de las facilidades y actividades que tendrá que realizar un usuario para llevar a cabo algún proceso, y se mostrará la interacción entre el usuario con el sistema para cada acción que desee realizar.



ESPECIFICACION DE UC

Nombre:	CU.1 Flujo Principal.	
Actor:	Usuario – Sistema.	
Descripción:	Un usuario accede al sistema de donación de sangre.	
Pre condición:	Usuario nuevo no registrado observa la lista de hospitales donde donar.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El usuario ingresa al menú principal.	
2		Se muestran las opciones: -Registrarse -Hospitales donde donar -Iniciar sesión -Compatibilidad de la sangre
3	El usuario pulsa “Iniciar Sesión”, se pasa al flujo <u>Login</u> . El usuario pulsa “Registrarse”, se pasa al flujo <u>Registro</u> . El usuario pulsa “Compatibilidad de la sangre” se pasa al flujo <u>Compatibilidad de la sangre</u> . El usuario pulsa “ver hospitales donde donar”, se pasa al flujo <u>ver hospitales</u> .	

4		<p>Se verificó el acceso del usuario (usuario/donante) , el sistema además ofrece <u>buscar donantes</u> , <u>disponibilidad para donar</u> , <u>conversar con usuarios</u>, <u>mensajear a donante</u> , <u>llamar a donante</u> , <u>geolocalizar a donante</u> .</p> <p>Se verificó el acceso del usuario – Administrador, el sistema ofrece acceso a todas las gestiones : gestión <u>de usuarios</u> , gestión <u>hospitales</u> , gestión <u>donantes</u>.</p>
5	El usuario finaliza su sesión. El usuario - administrador finaliza su sesión.	
		El sistema guarda los datos que se hallan ingresado por el usuario en su sesión en la base de datos de la aplicación .

Nombre:	CU.2 Login	
Actor:	Usuario – Sistema.	
Descripción:	Sin condición.	
Pre condición:	Ser un usuario registrado.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El usuario ingresa nombre de usuario y contraseña.	
2		<p>Se valida el usuario:</p> <p>-Se muestra un mensaje de existo.</p> <p>En caso de error se pide otra vez nombre de usuario y contraseña y además se muestra la opción de <u>Registro</u>.</p>

Nombre:	CU.3 Registro	
Actor:	Usuario – Sistema.	
Descripción:	Un usuario ingresa los datos necesario para su registro.	
Pre condición:	Sin condición.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El usuario pide registrarse en el sistema.	
2		Se muestran los campos obligatorios (*) y los complementarios.

3	El usuario llena sus datos.	
4		Se verifica los campo obligatorio: -Se muestra un mensaje de existo. En caso de error se pide corregir los datos.

Nombre:	CU.4 Compatibilidad de sangre	
Actor:	Usuario – Sistema.	
Descripción:	Se muestra los hospitales que necesiten el tipo de sangre.	
Pre condición:	Sin condición.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El usuario ingresa su tipo de sangre.	
2		Se muestran una lista con los hospitales que requieren el tipo de sangre señalado.

Nombre:	CU.5 Ver hospitales	
Actor:	Usuario – Sistema.	
Descripción:	Se muestra los hospitales que usan el sistema.	
Pre condición:	Sin condición.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El usuario pide ver la lista de hospitales en el sistema.	
2		Se muestran una lista con los hospitales registrados en el sistema.

Nombre:	CU.6 Buscar donantes.	
Actor:	Usuario – Sistema.	
Descripción:	Se muestra los donantes del tipo de sangre buscado.	
Pre condición:	Ser usuario registrado.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El usuario busca donantes con un el tipo de sangre y en cercano a un hospital.	
2		Se manda una alerta a los usuarios con el igual tipo de sangre. Se muestra una lista con los donantes mas cercanos al hospital señalado.

Nombre:	CU.7 Disponibilidad para donar	
Actor:	Usuario – Sistema.	
Descripción:	El usuario cambia su disponibilidad a donar.	
Pre condición:	Ser usuario registrado.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El usuario pide cambiar su disponibilidad a donar.	
2		-Se muestran su estado actual o sin estado en caso de ser un nuevo registro. -Se pide indicar su nuevo estado.
3	El usuario elige su nuevo estado.	
4		-Se valida en nuevo estado. -Se muestra un mensaje de éxito. En caso contrario se pide elegir otra vez un nuevo estado.

Nombre:	CU.8 Conversar con usuarios	
Actor:	Usuario – Sistema.	
Descripción:	El usuario pide número de celular de otro usuario.	
Pre condición:	Ser usuario registrado.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El usuario("usuario1") pide el número de celular para conversar con otro usuario registrado "usuario2".	
2		-El sistema manda un mensaje al "usuario2" indicando que el "usuario1" requiere su número de celular para iniciar una conversación. -Si se acepta, se muestra el número. -Si se rechaza se manda un mensaje al "usuario1" .

Nombre:	CU.9 Mensajes a donante	
Actor:	Usuario – Sistema.	
Descripción:	El usuario pide iniciar chat con otro usuario.	
Pre condición:	Ser usuario registrado.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El usuario("usuario1") pide iniciar chat con otro usuario registrado	

	“usuario2”.	
2		-El sistema manda un mensaje al “usuario2” indicando que el “usuario1” quiere iniciar una conversación. -Si se acepta, se establece el chat. -Si se rechaza se manda un mensaje al “usuario1”.

Nombre:	CU.10 Geolocalizar a donante.	
Actor:	Usuario – Sistema.	
Descripción:	El usuario pide ubicación de un donante.	
Pre condición:	Ser usuario registrado.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El usuario(“usuario1”) pide ubicación del usuario registrado “usuario2”.	
2		-El sistema manda un mensaje al “usuario2” indicando que el “usuario1” pide su ubicación. -Si se acepta, se establece manda las coordenadas. -Si se rechaza se manda un mensaje al “usuario1”.

Nombre:	CU.11 Gestión de usuarios.	
Actor:	Usuario – Sistema.	
Descripción:	Administrar los usuario.	
Pre condición:	Ser administrador.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El administrador ingresa al menú de administrador.	
2		Se muestra la opción: -Agregar usuario. Se muestra una lista de usuarios con las opciones de: -Editar usuario. -Eliminar usuario.
3	El administrador pulsa “Agregar usuario” pasa al flujo <u>Agregar usuario</u> . El usuario pulsa “Editar usuario” pasa al flujo de <u>Editar usuario</u> . El usuario pulsa “Eliminar usuario”	

	pasa al flujo de <u>Eliminar usuario</u> .	
--	--	--

Nombre:	CU.12 Agregar usuario	
Actor:	Usuario – Sistema.	
Descripción:	Agregar nuevo usuario.	
Pre condición:	Ser administrador.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El administrador pide agregar usuario.	
2		Se piden los datos del nuevo usuario.
3	Se ingresa los datos del usuario.	
4		Se registra el nuevo usuario.

Nombre:	CU.13 Editar usuario	
Actor:	Usuario – Sistema.	
Descripción:	Editar usuario.	
Pre condición:	Ser administrador.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El administrador pide editar usuario.	
2		Se piden los nuevos datos del usuario.
3	Se ingresa los datos del usuario.	
4		Se registra el usuario.

Nombre:	CU.14 Eliminar usuario	
Actor:	Usuario – Sistema.	
Descripción:	Eliminar usuario.	
Pre condición:	Ser administrador.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El administrador pide eliminar usuario.	
2		Se piden confirmación de la eliminación.
3	Se confirma la eliminación.	
4		Se elimina usuario.

Nombre:	CU.15 Gestión de donante.	
Actor:	Usuario – Sistema.	
Descripción:	Administrar de donante.	
Pre condición:	Ser administrador.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El administrador ingresa al menú de administración de donante.	
2		Se muestra la opción: -Agregar donante. Se muestra una lista de donantes con las opciones de: -Editar donante. -Eliminar donante.
3	El administrador pulsa “Agregar donante” pasa al flujo <u>Agregar donante</u> . El usuario pulsa “Editar donante” pasa al flujo de <u>Editar donante</u> . El usuario pulsa “Eliminar usuario” pasa al flujo de <u>Eliminar donante</u> .	

Nombre:	CU.16 Agregar donante	
Actor:	Usuario – Sistema.	
Descripción:	Agregar nuevo donante.	
Pre condición:	Ser administrador.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El administrador pide agregar donante.	
2		Se piden los datos del nuevo donante.
3	Se ingresa los datos del donante.	
4		Se registra el nuevo donante.

Nombre:	CU.17 Editar donante	
Actor:	Usuario – Sistema.	
Descripción:	Editar donante.	
Pre condición:	Ser administrador.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El administrador pide editar donante.	
2		Se piden los nuevos datos del donante.
3	Se ingresa los datos del donante.	

4		Se registra el donante.

Nombre:	CU.18 Eliminar donante	
Actor:	Usuario – Sistema.	
Descripción:	Eliminar donante.	
Pre condición:	Ser administrador.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El administrador pide eliminar donante.	
2		Se piden confirmación de la eliminación.
3	Se confirma la eliminación.	
4		Se elimina donante.

Nombre:	CU.19 Gestión de hospitales.	
Actor:	Usuario – Sistema.	
Descripción:	Administrar de hospitales.	
Pre condición:	Ser administrador.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El administrador ingresa al menú de administración de hospitales.	
2		Se muestra la opción: -Agregar hospitales. Se muestra una lista de hospitales con las opciones de: -Editar hospitales. -Eliminar hospitales.
3	El administrador pulsa “Agregar hospitales” pasa al flujo <u>Agregar hospital</u> . El usuario pulsa “Editar hospitales” pasa al flujo de <u>Editar hospital</u> . El usuario pulsa “Eliminar hospitales” pasa al flujo de <u>Eliminar hospital</u> .	

Nombre:	CU.20 Agregar hospitales	
Actor:	Usuario – Sistema.	
Descripción:	Agregar nuevo hospitales.	

Pre condición:	Ser administrador.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El administrador pide agregar hospitales.	
2		Se piden los datos del nuevo hospitales.
3	Se ingresa los datos del hospitales.	
4		Se registra el nuevo hospitales.

Nombre:	CU.21 Editar hospitales	
Actor:	Usuario – Sistema.	
Descripción:	Editar hospitales.	
Pre condición:	Ser administrador.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El administrador pide editar hospitales.	
2		Se piden los nuevos datos del hospitales.
3	Se ingresa los datos del hospitales.	
4		Se registra el hospitales.

Nombre:	CU.22 Eliminar hospitales	
Actor:	Usuario – Sistema.	
Descripción:	Eliminar hospitales.	
Pre condición:	Ser administrador.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
1	El administrador pide eliminar hospitales.	
2		Se piden confirmación de la eliminación.
3	Se confirma la eliminación.	
4		Se elimina donante.

4-ARQUITECTURA

En este trabajo se utilizó como arquitectura base, la arquitectura modelo -vista-controlador (MVC) en 3 capas, este patrón es una de las técnicas más comunes, pues divide sistemas de software complicados.

Para hacer una analogía el sistema basado en capas están ubicados de la misma forma que las capas de un pastel, donde cada capa descansa sobre la inferior. En este patrón la capa más alta utiliza varios servicios definidos por la inferior.

Además, normalmente cada capa oculta las capas inferiores de las siguientes superiores a esta.

Beneficios:

- Se puede considerar una capa como un todo, sin considerar a las otras.
- Las capas se pueden sustituir con implementaciones alternativas de los mismos servicios básicos.
- Se minimizan dependencias entre capas.
- Las capas posibilitan la estandarización de servicios.
- Luego de tener una capa construida, puede ser utilizada por muchos servicios de mayor nivel.

A continuación se describen las tres capas principales de un patrón de arquitectura por capas:

1. Capa de Vista: Referente a la interacción entre el usuario y el software.

Puede ser tan simple como un menú basado en líneas de comando o tan complejo como una aplicación basada en formas.

Su principal responsabilidad es mostrar información al usuario, interpretar los comandos de éste y realizar algunas validaciones simples de los datos ingresados.

2. Capa de Reglas de Negocio: También denominada Lógica de Dominio, esta capa contiene la funcionalidad que implementa la aplicación. Involucra cálculos basados en la información dada

por el usuario y datos almacenados y validaciones.

Controla la ejecución de la capa de acceso a datos y servicios externos. Se puede diseñar la lógica de la capa de negocios para uso directo por parte de componentes de presentación o

su encapsulamiento como servicio y llamada a través de una interfaz de servicios que coordina la conversación con los clientes del servicio o invoca cualquier flujo o componente de negocio.

3. Capa de Datos: Esta capa contiene la lógica de comunicación con otros sistemas que llevan a cabo tareas por la aplicación. Estos pueden ser monitores transaccionales, otras aplicaciones, sistemas de mensajerías, etc.

Para el caso de aplicaciones empresariales, generalmente está representado por una

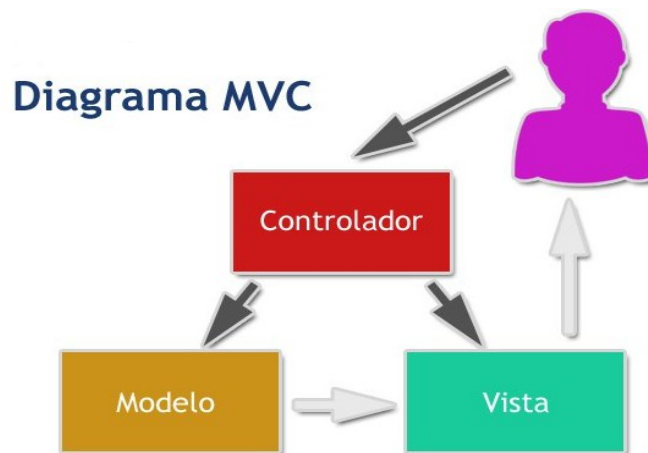
base de datos, que es responsable por el almacenamiento persistente de información. Esta capa debe abstraer completamente a las capas superiores (negocio) de la forma utilizada para comunicarse con los repositorios de datos, en este caso, MYSQL.

Ventajas:

- Una de las ventajas es que se logra encapsular funcionalidades, para su futuro reuso.
- Otra de las ventajas que tiene la separación, es que reduce el impacto ante un cambio tecnológico.

Por ejemplo, se podría sacar la capa lógica que se ocupa de los datos en una Base de Datos SQL y cambiarlo por otra que persista los datos en una Base ORACLE, al hacer esto no tendría impacto en las otras capas.

Gracias a esto se realiza una arquitectura escalable.



5-DISEÑO

Pruebas

Las pruebas se realizarán en máquinas virtuales de Android Studio con API 17, un 1 giga de memoria Ram, procesador de 2 núcleos de 1.2 ghz.

Despliegue

Para el uso de esta aplicación se debe tener:

- Debe tener instalada como mínimo API 17.
- Android Studio.
- Xampp
- Driver mysql-connector-java-5.0.8-bin.jar
- Importar la base de datos.

6-PLAN DE DESARROLLO DE SOFTWARE

Propósito

Identificar los objetivos requeridos de cada iteración para la completitud y éxito del proyecto. Establecer las actividades de desarrollo de cada fase e iteración necesarias para implementar la aplicación teniendo en cuenta las estimaciones del proyecto y el tiempo disponible.

Asimismo, proponer las medidas y formas de trabajo a tomar en cuenta durante el desarrollo del sistema por parte del equipo de desarrollo.

Alcance

El presente Plan de Desarrollo de Software describe, de modo genérico, el plan a ser llevado a cabo por el equipo de desarrollo para la implementación de la aplicación, perteneciente al proyecto final del curso de aplicaciones móviles.

Resumen

El presente Plan de Desarrollo de Software contiene la siguiente información:

- Sinopsis del Proyecto – Descripción del propósito del proyecto, su alcance y objetivos. Además, define entre otros, los entregables que se esperan del proyecto.
- Organización del Proyecto – Describe la estructura organizacional del equipo del proyecto.
- Proceso de Administración – Presenta las estimaciones del proyecto, define las fases e hitos del proyecto y describe cómo será monitoreado el proyecto.
- Planificación del Proceso Técnico – Metodología, herramientas y técnicas que serán utilizadas y el plan de aceptación del producto.

Propósito del proyecto, alcance y objetivo :

La aplicación para donar a organizaciones benéficas y/o donación de sangre por cercanía tiene como propósito proponer una solución valiosa para la necesidad de donantes de sangre en caso de emergencia como accidentes, operaciones de riesgo, transfusiones y estudio científico ya sea en hospitales, clínicas o centros de investigación.

La solución debe respaldar el alcance de los objetivos enfocados en su proceso de negocio .

Objetivos de las iteraciones

Fase de Concepción – Iteración Única

- Junto con los integrantes del proyecto identificar y comprender los requerimientos del sistema
- Tener identificados los principales casos de uso del sistema (Priorización)
- Lograr obtener, refinar o actualizar los entregables.

Fase de Elaboración – Iteración Única

- Al finalizar la fase de elaboración los casos de uso (priorizados) habrán completado su fase de análisis y diseño.
- Se logra elaborar la arquitectura del sistema.
- Lograr obtener, refinar o actualizar los entregables .
- Probar satisfactoriamente la arquitectura del sistema.
- Mitigar los siguientes riesgos:
 - Pérdida de la información entre uno y otro formulario . Tablas temporales .
 - Problemas con las licencias del software de desarrollo (Disponibilidad de herramientas de desarrollo)
 - Cambios en el alcance del proyecto - (Alcance no estable),
 - No disponibilidad de algunos integrantes del proyecto para consultas - (Requerimientos no entendidos completamente)
 - Dificultad de la comunicación del equipo de desarrollo con los demás miembros del equipo .
 - Problemas relacionados con funcionalidades que dependen de otros formularios . (Nuevas tecnologías para el equipo)

Fase de Construcción

- Objetivo común para las iteraciones de construcción la constituye la mitigación de los siguientes riesgos:
 - Pérdida de la información de los pedidos almacenada temporalmente .
 - Problemas relacionados con funcionalidades que dependen de otros formularios .
 - No disponibilidad de algunos miembros para consultas -

(Requerimientos no entendidos completamente)

- Poco o ningún dominio de android studio , java ee y programación web .

VERSIONES

Fecha entrega	Versión	Descripción	Fase
16/09/2016	0.1	-Diseño del cascaron de la aplicación. -Diseño e implementación de la base de datos .	Elaboración
20/09/2016	0.2	.Construcción de los siguientes casos de uso y web services : -flujo principal -login -registro -compatibilidad de la sangre -ver hospitales	Construcción 1
25/09/2016	0.3	Construcción de los siguientes casos de uso y web services : -buscar donantes -disponibilidad para donar -conversar con usuarios -mensajear a donante -geolocalizar a donante	Construcción 2
30/09/2016	0.7	.Construcción de los siguientes casos de uso y web services : -gestión de usuarios -gestión donantes -gestión hospitales	Construcción 3

05/10/2016	1.0	-Contiene todos los casos de uso y web services .	Transición 1
------------	-----	---	--------------

Recursos del Proyecto

Plan de Staff

No se requiere de un plan de staffing, pues los recursos disponibles los aportan los alumnos y la universidad.

Se pretende que el personal se mantenga a lo largo del auto aprendizaje en los que se desarrollará el proyecto.

Plan de capacitación

El equipo de desarrollo deberá recibir una capacitación adecuada en el uso de la herramienta Android Studio, dominio del programa de desarrollo especializado para Android. para eso se dispone de un DVD que contiene videotutoriales elaborados por desarrolladores de experiencias , este sera repartido para su uso a los integrantes del proyecto y capacitación directa.

Plan de Cierre

Se presentarán los entregables . Además se realizará un documento de informe final

del Proyecto en el que se indicarán las lecciones aprendidas como aporte para futuros proyectos siguiendo la metodología RUP.

Plan del Proceso Técnico

Caso de Desarrollo

La arquitectura planteada para cada sistema es de 3 de capas . Se cuentan con bases de datos para el sistema , se permite que las bases de datos se encuentren en un servidor .

Métodos, herramientas y técnicas

Se usarán las siguientes herramientas:

- Rational Unified Process – Guía para el proceso de desarrollo.
- Se seguirá la notación UML.

- StarUml
- Microsoft Word -Para la edición de los entregables .

Plan de soporte del proceso

Plan de Evaluación

La evaluación del trabajo realizado (administración del proyecto, entregables, etc) será realizado por el profesor del curso .

,Las cuales deberán presentar al equipo de desarrollo sus comentarios y correcciones que encuentren pertinentes con claridad y por escrito.

El grupo de desarrollo deberá tomar las correcciones y los comentarios en cuenta para la mejora del trabajo o presentar las objeciones o dudas que crea necesarias a fin de llegar a un acuerdo sobre el resultado de la evaluación.

Estas evaluaciones serán realizadas los días acordados al inicio de cada ciclo con el grupo de desarrollo.