



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL**

CARRERA: INGENIERÍA DE SISTEMAS

Tesis previa a la obtención del título de: INGENIERO DE SISTEMAS

TEMA:

**Aplicación de Donaciones para instituciones Benéficas mediante
Android**

AUTOR:

FREDDY VICENTE HAZ GARCÍA

DIRECTOR/A:

ING. MÁXIMO TANDAZO

Guayaquil, Abril de 2015

DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO DEL TRABAJO DE GRADO

Yo Freddy Vicente Haz García autorizo a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de grado y su reproducción sin fines de lucro.

Además declaro que los conceptos y análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad del/los /las autor/es/as.

Nombres y apellidos completos

CC 0927340422

AGRADECIMIENTO

A Dios por el regalo más hermoso que es la vida, llena de bendiciones y de personas que creen en mí, a mis padres por el gran esfuerzo que hacer día a día para darme lo mejor, por la confianza y apoyo incondicional que nos brindaron para poder terminar mis estudios; a la familia por su apoyo constante.

A mi director de tesis, Ing. Máximo Tandazo de manera especial, por su disposición para ayudarme, orientarme en el desarrollo de este tema.

A las fundaciones quienes depositaron su confianza en mí y facilitaron información para desarrollar de la mejor manera la investigación.

A la Universidad Politécnica Salesiana por darme la oportunidad de formarme y educarme como profesional, y tener la preparación para enfrentar cualquier reto.

ÍNDICE

1.	DISEÑO DE LA INVESTIGACIÓN.....	11
1.1.	Introducción.....	11
1.2	ANTECEDENTES DE LA INVESTIGACIÓN.....	12
1.2.1	PROBLEMA DE LA INVESTIGACION.....	12
1.2.2	PLANTEAMIENTO DEL PROBLEMA.....	12
1.2.3	TEMA.....	13
1.2.4	JUSTIFICACIÓN DE LA INVESTIGACIÓN.....	13
1.2.5	FACTORES JUSTIFICATIVOS DEL PROYECTO.....	14
2.	MARCO TEORICO15
2.1	CONCEPTO DE LAS ONG.....	15
2.1.2	OBJETIVO DE LAS ONG.....	16
2.1.3	AMBITO INTERNACIONAL Y NACIONAL DE UNA ONG.....	16
2.2	DONACIONES	17
2.2.1	PARA QUE EXISTEN.....	17
2.2.2	A QUIENES BENEFICIAN.....	17
2.2.2	INSTITUCIONES QUE PUEDEN PEDIR DONACIONES.....	17
2.3	ANDROID.....	17
2.3.1	FUNCIONALIDAD DE ANDROID.....	17
2.4	PYTHON.....	21
2.5	DJANGO.....	21
2.6	ORM.....	22
2.7	RESTFULL.....	23
2.8	CONCEPTO DE BASES DE DATOS.....	24
2.8.1	TIPOS DE BASES DE DATOS.....	25
2.8.2	COMPONENTES DE UN SISTEMA DE BASES DE DATOS.....	25
2.8.3	SISTEMAS MANEJADORES DE BASES DE DATOS.....	25
2.9	MySQL y SQLite.....	26
2.10	CONCEPTO DE SDK.....	27
2.10.1	RECURSOS DE UN SDK.....	27
2.11	Paypal SDK.....	27

2.12 SISTEMA OPERATIVO.....	28
2.13 UBUNTU.....	29
2.14 JSON.....	29
2.14.1 OAUTH	32
2.15 RUP	32
2.15.1 CARACTERISTICAS DE LA METODOLOGIA (RUP).....	33
2.15.1 CASOS DE USO.....	33
2.15.2 CENTRADO EN LA ARQUITECTURA.....	34
2.15.3 ITERATIVO E INCREMENTAL.....	34
2.15.4 LENGUAJE DE MODELADO UNIFICADO (UML).....	35
2.15.4.1. DIAGRAMA DE CASOS DE USO.....	36
2.15.4.2. DIAGRAMA DE CASOS DE USO DE ALTO NIVEL.....	37
2.15.4.3. DIAGRAMA DE CASOS DE USO DE EXPANDIDO.....	37
2.15.5 DIAGRAMA DE SECUENCIA.....	38
2.15.6 DIAGRAMA DE CLASES.....	38
2.16 BOOTSTRAP.....	39
2.17 FUNCIONAMIENTO DEL SISTEMA.....	39
2.18. HIPOTESIS.....	40
2.18.1 VARIABLES E INDICADORES.....	41
3. ANALISIS DEL SISTEMA.....	41
3.1 FASE DE PLANEACION.....	42
3.2.1 DESCRIPCION DE LOS ACTORES.....	42
3.2.4 DEFINICION DE ROLES DE LOS MODULOS.....	44
3.3. FUNCIONES BASICAS.....	45
3.4. FACTIBILIDAD TÉCNICA.....	45
3.4.1 VENTAJAS Y DESVENTAJAS DEL SOFTWARE.....	46
3.5. FACTIBILIDAD OPERACIONAL.....	47
3.6. FACTIBILIDAD ECONOMICA.....	48
3.7 DESCRIPCION CASOS DE USO DE ALTO NIVE	49
3.8 CASOS DE USO EXPANDIDO.....	51
3.9 DIAGRAMA DE SECUENCIA.....	57
3.10 DIAGRAMA DE CLASES.....	60
3.11 ESQUEMA DE BASE DE DATOS.....	62
3.12 ARQUITECTURA DEL SISTEMA.....	63

4. PROPUESTA.....	64
4.1.PROPUESTA DE APLICACIÓN.....	64
5 CONCLUSIONES Y RECOMENDACIONES.....	64
5.1.CONCLUSIONES.....	64
5.2 RECOMENDACIONES.....	64
6. BIBLIOGRAFIA.....	65
7. ANEXOS.....	66

ÍNDICE DE FIGURAS

Figura 1.1: Reporte de Crowdfunding Industry.....	11
Figura 2.1: Arquitectura de Android.	17
Figura 2.2: Logo de Python.....	21
Figura 2.3: Proceso de Peticion Http Django.....	21
Figura 2.3: Logo de Django.....	22
Figura 2.4. Arquitectura REST.....	22
Figura 2.5. Logo de Mysql.....	23
Figura 2.6. Logo SQLite.....	23
Figura 2.7. Logo Paypal	24
Figura 2.8. Estructura de un Objeto JSON.....	24
Figura 2.9. Estructura de un Arreglo JSON.....	26
Figura 2.10. Estructura de un Valor JSON.....	27
Figura 2.11. Estructura de un String JSON.....	28
Figura 2.12. Estructura de un Numero JSON.....	28
Figura 2.13. Funcionamiento de Twitter Outh.....	29
Figura 2.14 Esquema de la Metodología RUP.....	30
Figura 2.15: Proceso de un Caso de Uso.....	31
Figura 2.16: Proceso Iterativo/Incremental.....	32
Figura 2.17: Representación de un Actor.....	33
Figura 2.18: Representación de un Caso de Uso	34
Figura 2.19: Diagrama de caso de uso Expandido	35
Figura 2.20: Diagrama de Secuencia.....	36

Figura 2.21: Diagrama de Clases.....	36
Figura 3.1 Diagrama de Caso de Uso Principal de la Aplicación.....	38
Figura 3.2 Diagrama de Caso de Uso de Registro del Proyecto.....	41
Figura 3.3 Diagrama de Caso de Uso de Registro del Proyecto.....	42
Figura 3.4 Diagrama de Caso de Uso de Registro del Proyecto	44
Figura 3.5 Diagrama de Caso de Uso de Registro de Notificaciones a la Fundación	45
Figura 3.6 Diagrama de Caso de Uso de Registro de la Donación.....	47
Figura 3.7 Diagrama de Caso de Uso de Registro del Proyecto.....	48
Figura 3.8 Diagrama de Caso de Uso de Registro del Notificaciones al Donante....	48
Figura 3.9 Diagrama de Caso de Uso de Registro de Notificaciones a la Fundación.....	49
Figura 3.10 Diagrama de Clases del Proyecto	50
Figura 3.10 Esquema de bases de datos del Proyecto	51
Figura 3.11 Arquitectura de la Aplicación	52
Figura 3.12 Esquema de la Aplicación	53
Figura 4.1 Interfaz App en Android	60

ÍNDICE DE TABLAS

Tabla 2.1. Variables e indicadores.....	33
Tabla 3.0: Identificación de Casos de Uso.....	35
Tabla 3.1: Registro del Proyecto.....	36
Tabla 3.2: Registro de Notificaciones al Donante.....	36
Tabla 3.3: Registro de Donaciones.....	37
Tabla 3.4: Registro de Notificaciones a la fundación.....	37
Tabla 4.1: Caso de Uso Registro de Proyectos.....	39
Tabla 4.2: Caso de Uso Registro de Notificaciones al Donante.....	39
Tabla 4.3: Caso de Uso Registro de Donaciones.....	39
Tabla 4.4: Caso de Uso Registro de Notificaciones a la fundación.....	40
Tabla 4.4: Caso de Uso Extendido Registro de Donaciones.....	41
Tabla 4.5: Caso de Uso Extendido Registro de Proyectos.....	43
Tabla 4.6: Caso de Uso Extendido Registro de Notificaciones a los Donantes....	45
Tabla 4.7: Caso de Uso Extendido Registro de Notificaciones a la fundación....	46

ÍNDICE DE ANEXOS

Creación De Entornos Virtuales de la Aplicación en Django.....	61
Manual de Usuario Aplicación.....	66

RESUMEN

La fundación Clave del Sur es una institución que lleva 10 años formando a los jóvenes en la música, es la clave para sobrellevar las situaciones que se suscitan en sectores complicados como en el que se encuentra y por qué este concepto encierra también significaciones musicales. Debido a su gran crecimiento de alumnos se le hace difícil poder conseguir recursos para poder invertir en instrumentos musicales para la formación del alumnado.

Este proyecto tecnológico se presenta como una alternativa de solución a los diferentes problemas que la fundación Clave del Sur posee al momento de financiar sus proyectos musicales, a través de una aplicación móvil en Android que facilite seguir sus proyectos, donaciones y conseguir voluntarios.

ABSTRACT

The foundation Clave del Sur is an institution that has spent 10 years training young people in music , is the key to coping with situations that arise in complicated sectors where it is and why this concept also holds musical meanings. Because of its tremendous growth of students it`s difficult to raise funds to invest in musical instruments for their training.

This technological project is presented as an alternative solution to the several problems that the “Clave del Sur” foundation deals with at the moment of financing its musical projects, through a mobile app on Android that makes it easy to follow their projects, donations and recruits volunteers .

INTRODUCCIÓN

En la actualidad debido a la creciente demanda de dispositivos móviles (16,9%) y el acceso al internet (28,3%) según datos del INEC (INEC, 2013) notamos que cada vez la información debe estar al alcance de todos así como las opciones de poder fomentar campañas en las de las diferentes ONG`s (Organizaciones no Gubernamentales) en el internet para poder llegar a nuevas opciones de recaudar dinero para sus proyectos en diferentes comunidades.

Existen muchas ideas y plataformas que ayudan a las fundaciones como Une Option de Plus (Une Option de Plus, s.f.) en estas causas sociales en diversos países de Latinoamérica a pesar de que su enfoque es muy generalizado en varias ramas de la sociedad, se analizó la idea de hacer una propuesta similar en ayudas de recaudaciones de fondos para los diferentes proyectos que surgen de fundaciones en la ciudad como la que en este estudio analizamos la Fundación Clave del Sur que ayuda a las zonas vulnerables de la ciudad a integrarse a la educación mediante el aprendizaje de instrumentos musicales, es por esto que llegó a analizar los procesos que la fundación seguía para recaudar fondos de sus proyectos y también proponer una nueva forma de obtener financiamiento de los mismos.

La fundación Clave del Sur actualmente no tiene acceso a iniciativas que ayuden a recaudar fondos para sus proyectos sociales con presencia ante nuevos donantes que conozcan sus proyectos y se sientan identificados con la causa de la fundación, es por esto que se propuso el uso de una aplicación móvil en Android para difundir sus causas a los usuarios que usan Android.

La coordinación de la fundación para recibir ayudas en sus proyectos gestiona directamente con los donantes, los cuales son contactados personalmente o en funciones de la escuela de música que nació a través de los años con esta iniciativa social de aprendizaje en la música. (Clave del Sur, s.f.)

CAPÍTULO 1

DISEÑO DE LA INVESTIGACIÓN

1.1 Antecedentes de la investigación

1.1.1 Problema de la investigación

1.1.2 Planteamiento del problema

Según el estudio hecho por Massolution con el reporte “2013 Crowdfunding Industry Report” las diversas plataformas que se ubican en la internet está en un constante crecimiento. En el 2012 según este estudio las causas sociales fueron las más financiadas a través de este tipo de plataformas con un 27.4%.

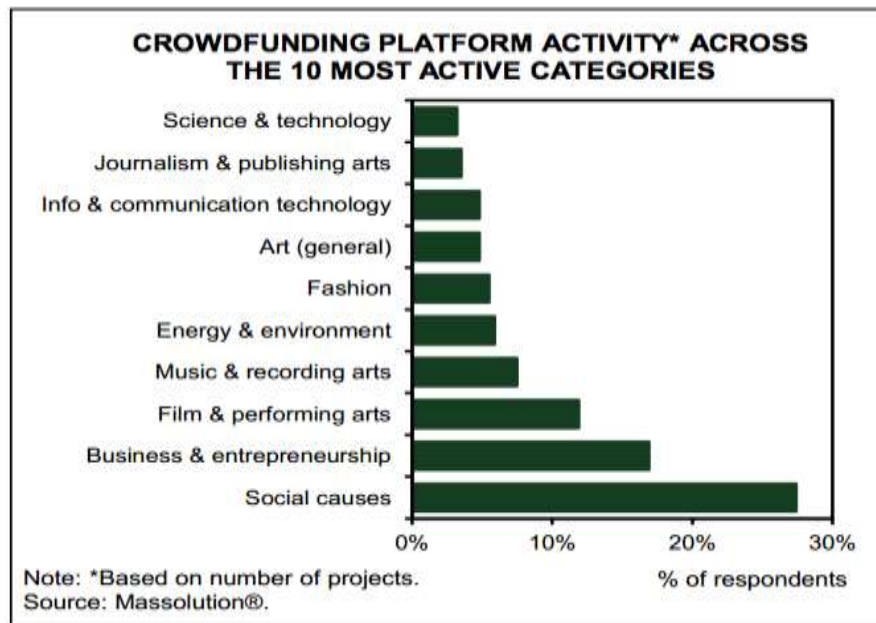


Figura 1.1: Reporte de Crowdfunding Industry

Fuente: (Mass, 2012)

Analizando las estadísticas de las 14 plataformas que recaudan dinero en Lationamerica (InfoCrowdsourcing, 2014), nos encontramos que las buenas

difusiones a través de muchos canales de comunicación como lo son los diversos medios en la web. Las ONG's, en la ciudad de Guayaquil no usan una buena estrategia para conseguir nuevos medios de financiamiento como lo son las plataformas de donaciones.

Debido a los diferentes canales de comunicación las ongs deben realizar acciones de comunicación online para ampliar la base social de colaboradores afianzando una relación basada en la confianza, para ello una de las formas eficientes y de menor costo es usar los medios sociales en la web.

Uno de los más grandes problemas es la difusión de estas pequeñas historias que no son más que proyectos que las diferentes ONGs disponen, pero hoy en día no tienen una amplia difusión en los diferentes medios escritos o televisivos, identificando este problema surge otro al no difundir estos proyectos se cae en una excesiva dependencia de financiaciones del sector público como privado, a estos dos problemas encontrados se ha propuesto compartir y conseguir financiamiento a estos proyectos ONGs.

1.1.3 Tema

Crear una aplicación móvil para recaudar donaciones para la ONG's Clave del Sur de la Ciudad de Guayaquil.

1.1.4 Justificación de la Investigación

Este proyecto busca fomentar una alternativa de difusión mediante una aplicación móvil al alcance de la población que representan las redes sociales que ayudara a facilitar un medio para donaciones en los proyectos de dichas ONG's.

- Reclutar nuevos miembros / colaboradores / seguidores
- Interaccionar con miembros / seguidores existentes
- Reclutar donantes y aceptar donaciones del cualquier parte del mundo

- Forjar alianzas con otras organizaciones / grupos que tienen la misma misión que su Organización No Gubernamental
- Mejorar su visibilidad para que otras personas / organizaciones que tienen intereses parecidos puedan encontrar su organización y aprender de su trabajo
- Comunicar con otras organizaciones / personas haciendo trabajo parecido para compartir información / experiencias y aprender de ellas

1.1.5 Factores Justificativos del Proyecto

- Avances Tecnológicos. A medida que pasan los años es necesario implementar sistemas que vayan acorde al progreso e innovación creativa y tecnológica.
- Necesidad. Es una gran necesidad poder culturizar a la sociedad con nuevas fuentes de financiamiento a las instituciones benéficas mediante pequeños proyectos benéficos.
- Usabilidad. La aplicación podrá ser usada por los usuarios de smartphones Android y de la web, con el poder de apoyar a proyectos presentados en la app.
- Impacto Social. El impacto social de estos logros como nunca se ha hecho, compartiendo toda la información relativa a cada proyecto y sobretodo permitiendo seguir su evolución a lo largo de todo el proceso de recaudación y ejecución del proyecto.

1.1.6 Formulación del Problema

¿De qué manera se podrían desarrollar proyectos con sentido social con la ayuda de las aplicaciones móviles y difusión en redes sociales?

1.1.7 Sistematización del Problema

- A. ¿Cómo podría una institución benéfica desarrollar proyectos sociales sin depender de empresas e instituciones bancarias?
- B. ¿Qué impacto tendría sobre los donantes de la fundación la implementación de una aplicación benéfica que ayude a desarrollar estos proyectos?
- C. ¿Qué aporte tecnológico ofreceríamos a la comunidad implementando una aplicación que ayude a fomentar el sentido social de cada usuario en el internet?

1.2 Objetivos

1.2.1 Objetivo Principal

Crear una aplicación móvil que permita dar a conocer proyectos de las ONG'S que ayuden a fomentar el desarrollo educativo de las personas.

1.2.2 Objetivos Generales

- Desarrollar una aplicación que facilite a las instituciones benéficas conseguir recursos de las personas que usen smarthphones mediante dinero electrónico o artículos que necesiten para cualquier necesidad dichas instituciones.

1.2.3 Objetivos Específicos

- Ayudar a las organizaciones sin ánimo de lucro a comunicar el impacto social de sus proyectos centrándose en ayudar a las personas.
- Compartir proyectos con botones de ayuda para dar a conocer los proyectos en redes sociales.
- Permitir a las Instituciones benéficas poder contar las historias humanas detrás de cada proyecto. A partir de ahora se podrá disfrutar viendo como los proyectos se hacen realidad de principio a fin.

CAPÍTULO 2

MARCO TEÓRICO

2.1 Concepto de las Ong

Las ONG son entidades basadas en una ideología solidaria y altruista que preside todas sus actuaciones y en consecuencia su gestión. El término “*organización no gubernamental*” fue acuñado por las Naciones Unidas. “Las ONG se identifican con lo que se conoce como actividades del tercer sector. Se trata de organizaciones que nacen, se desarrollan y actúan en el ámbito privado, es decir, fuera de las organizaciones estatales.”

“La ONU define que una organización no gubernamental es cualquier grupo no lucrativo de ciudadanos voluntarios, que está organizado a nivel local, nacional o internacional. Con tareas orientadas y dirigidas por personas con un interés común. Las ONG realizan una variedad de servicios y funciones humanitarias, llevan los problemas de los ciudadanos a los gobiernos, supervisan las políticas y alientan la participación de la comunidad, proveen de análisis y experiencia, sirven como mecanismos de advertencia temprana y ayudan en la supervisión e implementación de acuerdos internacionales.” (Pablo, 2005)

2.1.2 Objetivo de las Ong

El objetivo de las ONG es contribuir en el desarrollo de los sectores más vulnerables de la población junto con el estado. Cumplen con la misión y los objetivos sociales que empresas privadas no pueden asumir por no ser rentables.

“Las ONG no tratan de reemplazar las acciones del Estado o de los organismos internacionales en sus correspondientes países, sino de cubrir y ayudar en aquellas áreas en las cuales no existen políticas sociales o económicas, o bien cuando estas políticas resultan insatisfactorias para algunos grupos de la sociedad o la sociedad en su conjunto.” (Rubén, 2011)

2.1.3 Ámbito Internacional y Nacional de una Ong

Cubren una gran variedad de temas y ámbitos que definen su trabajo y desarrollo. Dichos temas están relacionados con ayuda humanitaria, salud pública, investigación, desarrollo económico, cultura, derechos humanos, transferencia tecnológica, ecología ,etc.

Algunas ONG realizan acciones que promueven el desarrollo de las destrezas y habilidades de las personas para la superación de la pobreza. Su propósito es generar cambios en los planos individual, familiar y comunitario, convirtiéndolos en agentes autónomos de su propio desarrollo.

2.2 Donaciones

La Donación. Es el acto por el cual se realiza la transferencia gratuita de la propiedad de un bien que realiza una persona (Donante) a favor de otra (Donatario). (Código Civil)

2.2.1 Para Que Existen

Para que las instituciones ONG puedan desarrollar sus proyectos sociales en el ámbito de acción que se desempeñan.

2.2.2 A Quienes Benefician

A todas las instituciones ONG que estén debidamente registradas y aprobadas por la Secretaría Técnica de Cooperación Internacional en el país.

2.2.2 Instituciones que pueden pedir Donaciones

Las ONG internacionales operan a través de convenios suscritos con el Gobierno. En la actualidad existen 141 ONG's registradas en el país. (Internacional S. T., 2013)

2.3 Android

Android es un sistema operativo móvil basado en Linux, que junto con aplicaciones middleware está enfocado para ser utilizado en dispositivos móviles como smartphone, tablets, etc. Es desarrollado por la Open Handset Alliance² la cual es liderada por Google. (Rubén, 2011)

2.3.1 Funcionalidad De Android

El siguiente gráfico muestra la arquitectura de Android. Como se puede ver está formada por cuatro capas. Una de las características más importantes es que todas las capas están basadas en software libre.



Figura 2.1: Arquitectura de Android.

Fuente: (Gironés, 2010) El gran libro de Android

Núcleo Android .-El núcleo de Android está formado por el sistema operativo Linux versión 2.6. Esta capa proporciona servicios como la seguridad, el manejo de la memoria, el multiproceso, la pila de protocolos y el soporte de drivers para dispositivos.

Esta capa del modelo actúa como capa de abstracción entre el hardware y el resto de la pila. Por lo tanto, es la única que es dependiente del hardware.

Runtime .-Está basado en el concepto de máquina virtual utilizado en Java. Dado las limitaciones de los dispositivos donde ha de correr Android (poca memoria y procesador limitado) no fue posible utilizar una máquina virtual Java estándar. Google tomó la decisión de crear una nueva, la máquina virtual Dalvik, que respondiera mejor a estas limitaciones.

Algunas características de la máquina virtual Dalvik que facilitan esta optimización de recursos son: que ejecuta ficheros Dalvik ejecutables (.dex) - formato optimizado para ahorrar memoria. Además, está basada en registros. Cada aplicación corre en su propio proceso Linux con su propia instancia de la máquina virtual Dalvik. Delega al

kernel de Linux algunas funciones como threading y el manejo de la memoria a bajo nivel.

También se incluye en el Runfime de Android el "core libraries" con la mayoría de las librerías disponibles en el lenguaje Java.

Librerías Nativas.- Incluye un conjunto de librerías en C/C++ usadas en varios componentes de Android. Están compiladas en código nativo del procesador. Muchas de las librerías utilizan proyectos de código abierto. Algunas de estas librerías son:

- System C library: una derivación de la librería BSD de C estándar (libc), adaptada para dispositivos embebidos basados en Linux.
- Media Framework: librería basada en PacketVideo's OpenCORE; soporta codecs de reproducción y grabación de multitud de formatos de audio vídeo e imágenes MPEG4, H.264, MP3, AAC, AMR, JPG y PNG.
- Surface Manager: maneja el acceso al subsistema de representación gráfica en 2D y 3D.
- WebKit: soporta un moderno navegador web utilizado en el navegador Android y en la vista webview. Se trata de la misma librería que utiliza Google Chrome y Safari de Apple.
- SGL: motor de gráficos 2D.
- Librerías 3D: implementación basada en OpenGL ES 1.0 API. Las librerías utilizan el acelerador hardware 3D si está disponible, o el software altamente optimizado de proyección 3D.
- FreeType: fuentes en bitmap y renderizado vectorial.
- SQLite: potente y ligero motor de bases de datos relacionales disponible para todas las aplicaciones.
- SSL: proporciona servicios de encriptación Secure Sockef Layer.

Entorno de la Aplicación.- Proporciona una plataforma de desarrollo libre para aplicaciones con gran riqueza e innovaciones (sensores, localización, servicios, barra de notificaciones,). También se conoce como Java SDK.

La arquitectura ha sido diseñada para simplificar la reutilización de componentes. Las aplicaciones pueden publicar sus capacidades y otras pueden hacer uso de ellas (sujetas a las restricciones de seguridad). Este mismo mecanismo permite a los usuarios reemplazar componentes.

Una de las mayores fortalezas del entorno de aplicación de Android es que se aprovecha el lenguaje de programación Java. El SDK de Android no acaba de ofrecer todo lo disponible para su estándar del entorno de ejecución Java (JRE), pero es compatible con una fracción muy significativa de la misma.

Los servicios más importantes que incluye son:

Views: extenso conjunto de vistas, (parte visual de los componentes).

Resource Manager: proporciona acceso a recursos que no son en código.

Activity Manager: maneja el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellas.

Notification Manager: permite a las aplicaciones mostrar alertas personalizadas en la barra de estado

Content Providers: mecanismo sencillo para acceder a datos de otras aplicaciones (como los contactos). (Gironés, 2010) (Montoro, 2010)

Aplicaciones .- Este nivel está formado por el conjunto de aplicaciones instaladas en una máquina Android. Todas las aplicaciones han de correr en la máquina virtual Dalvik para garantizar la seguridad del sistema.

Normalmente las aplicaciones Android están escritas en Java. Para desarrollar aplicaciones en Java podemos utilizar el Android SDK. Existe otra opción consistente en desarrollar las aplicaciones utilizando C/C++. Para esta opción podemos utilizar el Android NDK (Native Development Kit).

2.4 PYTHON

Python es un lenguaje de programación de alto nivel, interpretado y multipropósito. Python puede ser utilizado en diversas plataformas y sistemas operativos, entre los que podemos destacar lo más populares, como Windows, Mac Os X y Linux. Pero además, también puede funcionar en smarthphones. Con este lenguaje podemos desarrollar software para aplicaciones científicas, para comunicaciones de red, para aplicaciones de escritorio con interfaz gráfica de usuario (GUI), para crear juegos, para smarthphones y por supuesto, para aplicaciones web. (Montoro, 2010)



Figura 2.2: Logo de Python.

Fuente: Python.org

2.5 Django

Es un framework de código libre para el desarrollo de aplicaciones web, escrito en Python, que sigue el patrón arquitectónico Modelo-Vista-Controlador.

El principal objetivo de Django es permitir la creación de sitios web complejos de forma sencilla, incentivando la reusabilidad y la conexión entre componentes. (Jiménez, 2010).

Los componentes básicos de este Framework son:

- Mapeador Objeto Relacional (ORM) `models.py`.- Contiene una descripción de la tabla de la base de datos, como una clase Python. Usando esta clase se pueden crear, buscar, actualizar y borrar entradas de tu base de datos usando código python.

- URL Dispatcher (Urls.py).- Especifica que vista llamada según el patrón URL.
- Vistas (views.py).- Contiene la Lógica de la página. A esta función se la llama Vista.
- Plantillas (HTML).-Es una plantilla HTML que describe el diseño de la página.

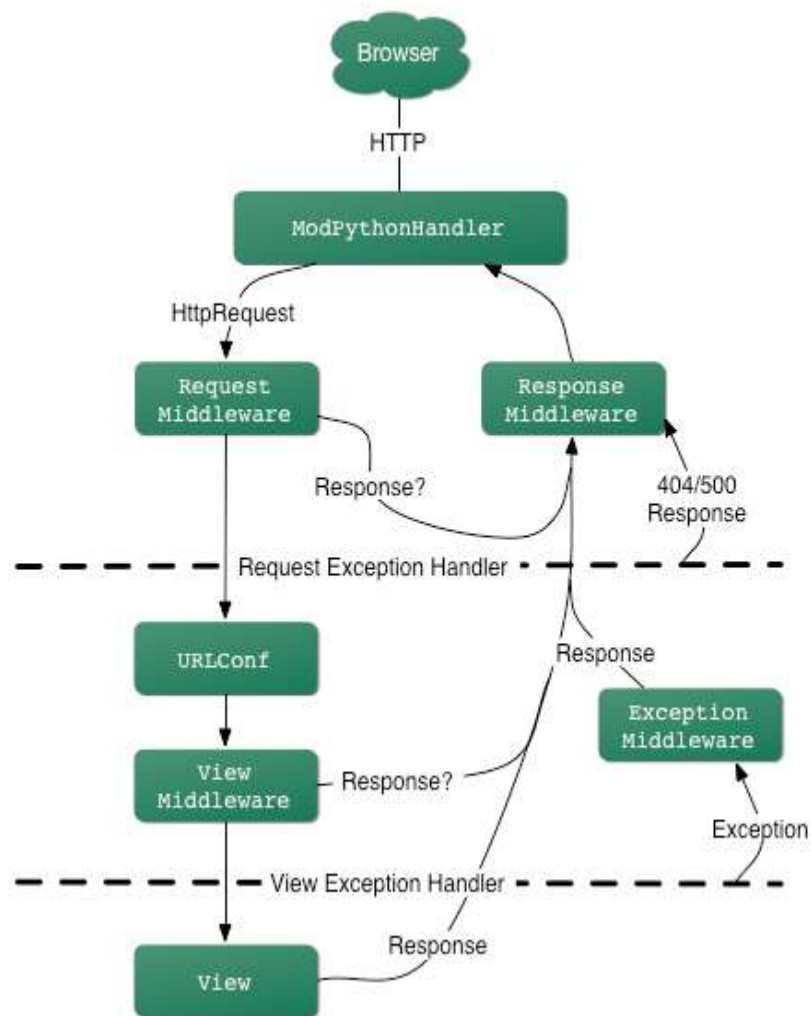


Figura 2.3: Proceso de Petición Http Django

Fuente: (Django, s.f.)



Figura 2.4: Logo de Django

Fuente: (Django P. , s.f.)

2.6 Orm

Es la persistencia (identifica a los objetos) automatizada y transparente de las tablas en una base de datos relacional, usando metadatos que definen el mapeo entre los objetos y la base de Datos. (Estudio de la Técnica ORM, 2008)

Es una tecnica que se usa para poder relacionar las bases de datos y los conceptos orientados a objetos creando entornos de bases de datos virtuales, es decir usada internamente por un framework (patron de desarrollo de una aplicación)

El ORM posee los siguientes componentes:

- Una interfaz de programación de aplicaciones (API) para realizar las operaciones de creación, actualización, consulta y eliminación de los objetos.
- Un lenguaje para especificar consultas sobre dichas clases.
- Una opción para especificar el mapeo de metadatos.
- Técnica de implementación del ORM para optimizar las búsquedas y funciones del framework.

2.7 Restfull

REST (Representational State Transfer) es un estilo de arquitectura de software para sistemas hipermedias distribuidos tales como la Web. Él termino fue introducido en la tesis doctoral de Roy Fielding en 2000, quien es uno de los principales autores de la especificación de HTTP.

El estilo de arquitectura subyacente a la Web es el modelo REST. Los objetivos de este estilo de arquitectura se listan a continuación:

Escalabilidad de la interacción con los componentes. La Web ha crecido exponencialmente sin degradar su rendimiento. Una prueba de ellos es la variedad de clientes que pueden acceder a través de la Web: estaciones de trabajo, sistemas industriales, dispositivos móviles.

Generalidad de interfaces. Gracias al protocolo HTTP, cualquier cliente puede interactuar con cualquier servidor HTTP sin ninguna configuración especial. Esto no es del todo cierto para otras alternativas, como SOAP para los Servicios Web.

Puesta en funcionamiento independiente. Este hecho es una realidad que debe tratarse cuando se trabaja en Internet. Los clientes y servidores pueden ser puestas en funcionamiento durante años. Por tanto, los servidores antiguos deben ser capaces de entenderse con clientes actuales y viceversa. Diseñar un protocolo que permita este tipo de características resulta muy complicado. HTTP permite la extensibilidad mediante el uso de las cabeceras, a través de las URIs, a través de la habilidad para crear nuevos métodos y tipos de contenido.

Compatibilidad con componentes intermedios. Los más populares intermediarios son varios tipos de proxys para Web. Algunos de ellos, las caches, se utilizan para mejorar el rendimiento. Otros permiten reforzar las políticas de seguridad: firewalls. Y por último, otro tipo importante de intermediarios, gateway, permiten encapsular sistemas no propiamente Web. Por tanto, la compatibilidad con intermediarios nos permite reducir la latencia de interacción, reforzar la seguridad y encapsular otros sistemas. (Marset, 2006)

REST	
Características	Las operaciones se definen en los mensajes. Una dirección única para cada instancia del proceso. Cada objeto soporta las operaciones estándares definidas. Componentes débilmente acoplados.
Ventajas declaradas	Bajo consumo de recursos. Las instancias del proceso son creadas explícitamente. El cliente no necesita información de enrutamiento a partir de la URI inicial. Los clientes pueden tener una interfaz "listener" (escuchadora) genérica para las notificaciones. Generalmente fácil de construir y adoptar.
Posibles desventajas	Gran número de objetos. Manejar el espacio de nombres (URIs) puede ser engorroso. La descripción sintáctica/semántica muy informal (orientada al usuario). Pocas herramientas de desarrollo.

Figura 2.5. Arquitectura REST

Fuente: (Marset, 2006)

2.8 Concepto de Bases de Datos

Es una colección de archivos relacionados que permite manejar la información de cualquier ente que lo necesite. Estos archivos se pueden interpretar como una colección de registros y cada registro a su vez estará compuesto de una colección de campos.

Cada campo posee un registro que permite llevar la información de algún atributo de una entidad que se establezca.

2.8.1 Tipos de Bases de Datos

Existen 3 enfoques para el manejo de una base de datos

Enfoque Relacional: Este enfoque prioriza los datos como un conjunto de Tablas, donde cada tabla posee un conjunto de registros y cada registro consta de un campo. La relación entre tablas se establece por los nombres de los atributos que comparten en común.

Enfoque Jerarquico: Permite ver una base de datos compuesta por un conjunto de relaciones PADRE-HIJO. De esta forma hay una serie de nodos hijos de forma que puede haber mas de un hijo para el mismo padre.

Las entidades de este enfoque son llamados segmentos y los atributos campos.

Enfoque de Redes: Este enfoque de redes modela la informacion como conjuntos, donde hay dos tipos de conjuntos: Propietarios (es equivalente a un padre en el enfoque jerarquico) y Miembros (Equivalente a un hijo dentro del enfoque jerarquico). La redundancia en este enfoque se reduce al maximo, no se pierde informacion debido a que una parte no se suministre o a que un proveedor no suministre alguna parte.

2.8.2 Componentes de un Sistema de Bases de Datos

Estos componentes son :

Datos : Los datos estan integrados (Es la unificacion de varios archivos con redundancia parcial o totalmente eliminada) y son compartidos (pueden ser accedados concurrentemente por diferentes usuarios)

Hardware : Es una unidad de almacenamiento secundario, es decir discos duros etc.

Software: Es un manejador de base de datos (Sistemas Manejadores de Bases de Datos).

Usuarios: Eston son los programadores de aplicaciones, usuario final y el administrador de la base de datos.

2.8.3 Sistemas Manejadores de Bases de Datos

Es un conjunto de programas que se encargan de manejar la creacion y todos los accesos a la base de datos. Se componen de un lenguaje de definicion de datos (DDL), un lenguaje de manipulacion de datos (DML) y de un lenguaje de consulta (SQL).

2.9 Mysql y Sqlite

MySQL: Es un sistema de administración de bases de datos relacional (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización.

MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger y hacer volcado de datos.

Utiliza SQL se trata de un lenguaje utilizado por todas las bases relacionales. (Gilfillan)



Figura 2.6. Logo de Mysql

Fuente: (Mysql, s.f.)

SQLite.- Es otro sistema de gestión de base de datos relacional. La principal diferencia con otros sistemas similares, es independiente realizándose simplemente llamadas a subrutinas o funciones de las propias librerías de SQLite. (Fossati, 2007)



Figura 2.7. Logo SQLite

Fuente: (Sqlite, s.f.)

2.10 Concepto de Sdk

SDK es un kit de desarrollo de software en la cual se puede desarrollar aplicaciones y ejecutar emuladores de las diferentes plataformas web que gozamos hoy en día como por ejemplo: Amazon, Paypal, Foursquare entre otros. Es decir que las aplicaciones desarrolladas sobre el SDK estarán destinadas a algún sistema operativo, plataforma de hardware, consola de video juegos etc.

2.10.1 Recursos de un Sdk

Los recursos que maneja un SDK son:

Interfaz de Programación de aplicaciones (API): ES una abstracción del funcionamiento interno del entorno sobre el que se va a trabajar. Posee un conjunto de funciones, rutinas, estructura de datos, clases y variables que nos permiten manipular el mecanismo de la plataforma.

Entorno de Desarrollo Integrado (IDE): Nos facilita a escribir el código fuente del programa, posee opciones como: Debugger (Testear el programa en ejecución), Compilador (Traduce el código fuente a lenguaje de máquina). Emulador del Entorno(si se usa aplicaciones móviles procesará como será la vista para el usuario final).

2.11 Paypal Sdk

Los PayPal Mobile SDK permiten aplicaciones nativas para aceptar fácilmente PayPal y pagos con tarjeta de crédito.

Las bibliotecas nativas de los SDK de PayPal Mobile le permiten crear, aplicaciones de respuesta rápida para los siguientes casos de uso:

Pagos individuales. Para los pagos individuales, su aplicación puede aceptar PayPal y los pagos de tarjetas de crédito.

Los pagos futuros. Los clientes pueden dar su consentimiento para realizar futuros pagos de PayPal. Después de la autorización clientes, puede almacenar la información de pago, así que es más fácil de aceptar pagos en curso. Esta funcionalidad no está disponible todavía, para pagos con tarjeta de crédito. (Paypal, 2014)



Figura 2.8. Logo Paypal

Fuente: (Paypal, 2014)

2.12 Sistema Operativo

Un sistema operativo es un programa (software) encargado de poner en funcionamiento el ordenador, puesto que gestiona los procesos básicos del sistema. Así mismo se encarga de gestionar para el usuario el hardware.

El sistema operativo comienza a trabajar en cuanto se enciende el ordenador y es completamente fundamental para que el usuario trabaje con él. Los sistemas operativos realizan tareas básicas y, sin ellos, el ordenador no funcionaría. Así, por ejemplo, el sistema operativo reconoce la conexión del teclado, organiza y ordena los archivos, controla la impresora, la pantalla, etc. Es como un policía de tráfico pues, se encarga de que los programas no interfieran entre ellos.

Los sistemas operativos tienen numerosas funciones:

1. Gestionan la memoria RAM de los distintos procesos. Un proceso es simplemente, un programa en ejecución, es decir, una tarea que realiza el ordenador.
2. Gestiona el almacenamiento de información de forma permanente en unidades de disco (disco duro, disquetes, pen drives, etc.).
3. Gestiona el sistema de archivos que nos permite crear, eliminar y manipular archivos y carpetas (también llamadas directorios).
4. Crea mecanismos de protección para evitar el acceso de intrusos a recursos o servicios no autorizados.
5. Dispone de un intérprete de comandos. Es un recurso que permite al usuario comunicarse con el sistema operativo a través de órdenes o comandos que son escritos. También se llama consola o shell.

6. El Sistema Operativo gestiona los sistemas de entrada/salida, es decir, controla los diferentes dispositivos conectados al ordenador (monitor, impresora, etc.)

La parte más importante del Sistema Operativo se llama núcleo o kernel. Asigna tareas al procesador siguiendo un orden y administrando los tiempos que lleva cada tarea. (UNAM, 2015)

2.13 Ubuntu

Es un sistema operativo de software libre (Gratis) que permite al computador utilizar los diversos programas que posee un sistema operativo para su funcionamiento. Este S.O. puede usarse en interfaz grafica al usuario como también en modo de comandos.

Está orientado al usuario final con una facilidad de uso de sus programas que están licenciados libremente dentro de la distribución del sistema operativo, pertenece a la compañía Canonical que lo distribuye libremente.

2.14 Json

JSON (JavaScript Object Notation) es un formato utilizado en Javascript (JS) para serializar datos. Se lo suele usar en sitios web como formato de intercambio de datos "liviano"(en comparación con XML), y tiene una particularidad que los datos en este formato de datos son leídos de manera directa en JS. (Bassi)

Los tipos de datos que puede tomar el Value son:

Un **objeto** es un conjunto desordenado de pares nombre/valor. Un objeto comienza con { (llave de apertura) y termine con } (llave de cierre). Cada nombre es seguido por : (dos puntos) y los pares nombre/valor están separados por , (coma).

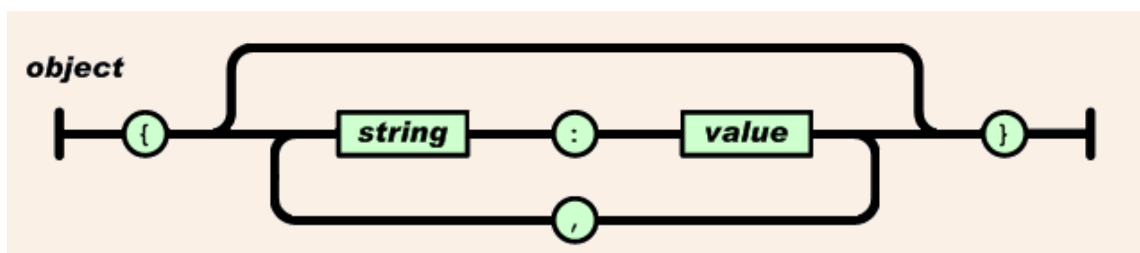


Figura 2.9. Estructura de un Objeto JSON

Fuente: (Ecma, 2015)

Un *arreglo* es una colección de valores. Un arreglo comienza con [(corchete izquierdo) y termina con] (corchete derecho). Los valores se separan por , (coma).

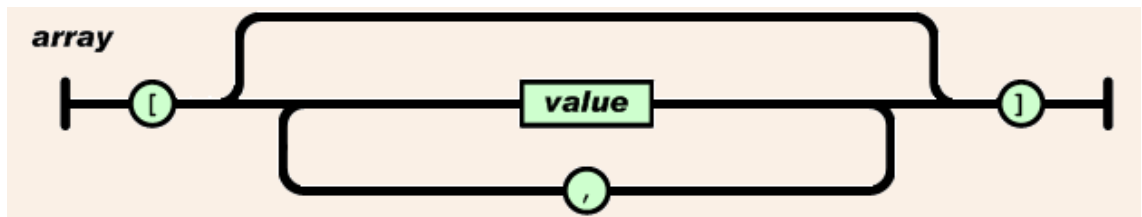


Figura 2.10. Estructura de un Arreglo JSON

Fuente: (Ecma, 2015)

Un *valor* puede ser una *cadena de caracteres* con comillas dobles, o un *número*, o true o false o null, o un *objeto* o un *arreglo*. Estas estructuras pueden anidarse.

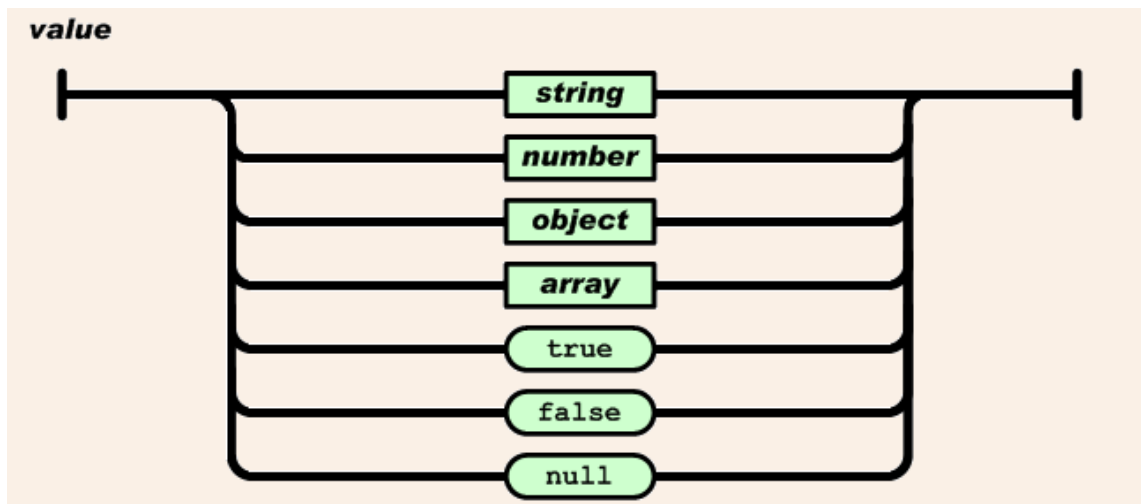


Figura 2.11. Estructura de un Valor JSON

Fuente: (Ecma, 2015)

Una *cadena de caracteres* es una colección de cero o más caracteres Unicode, encerrados entre comillas dobles, usando barras divisorias invertidas como escape. Un carácter está representado por una cadena de caracteres de un único carácter.

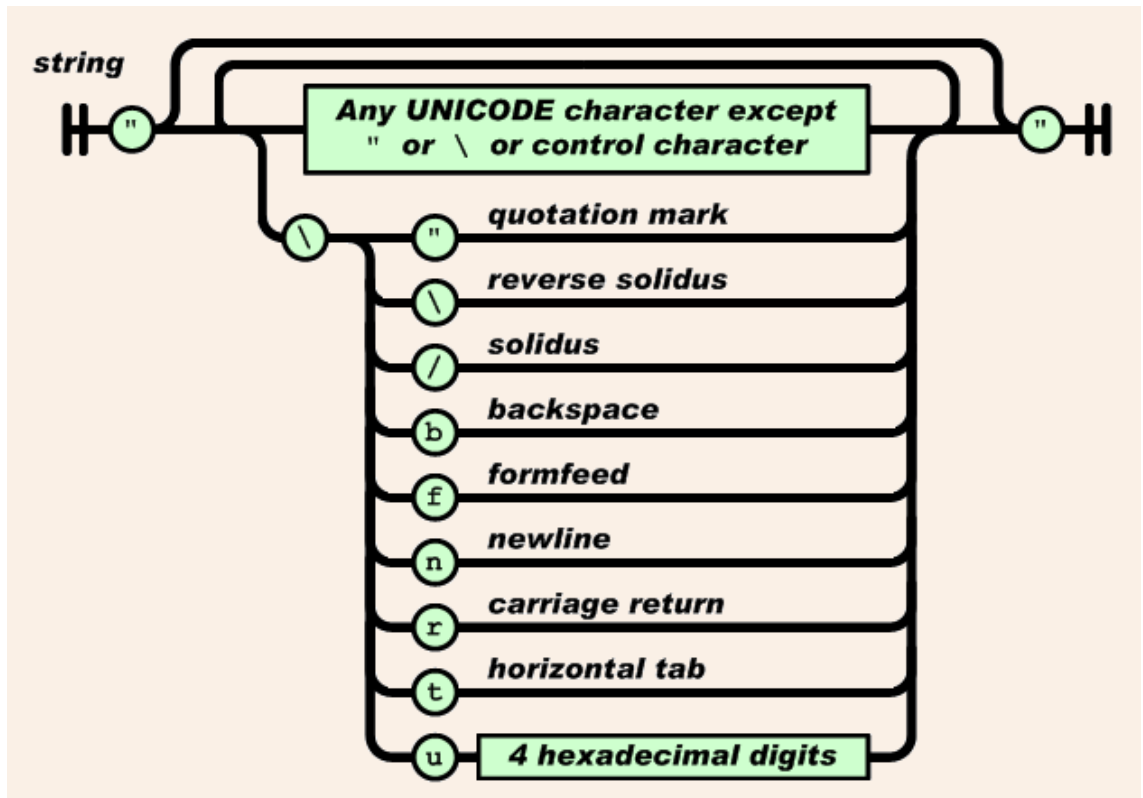


Figura 2.12. Estructura de un String JSON

Fuente: (Ecma, 2015)

Un *número* es similar a un número C o Java, excepto que no se usan los formatos octales y hexadecimales.

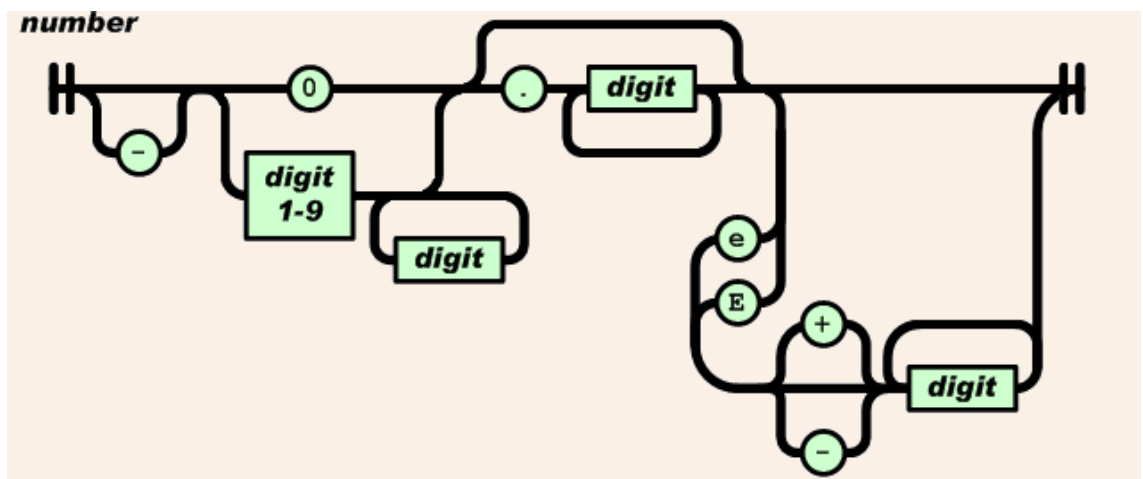


Figura 2.13. Estructura de un Numero JSON

Fuente: (Ecma, 2015)

2.14.1 OAuth

OAuth es un protocolo abierto que permite autorización segura mediante una API que gestiona el acceso para todo tipo de aplicaciones.

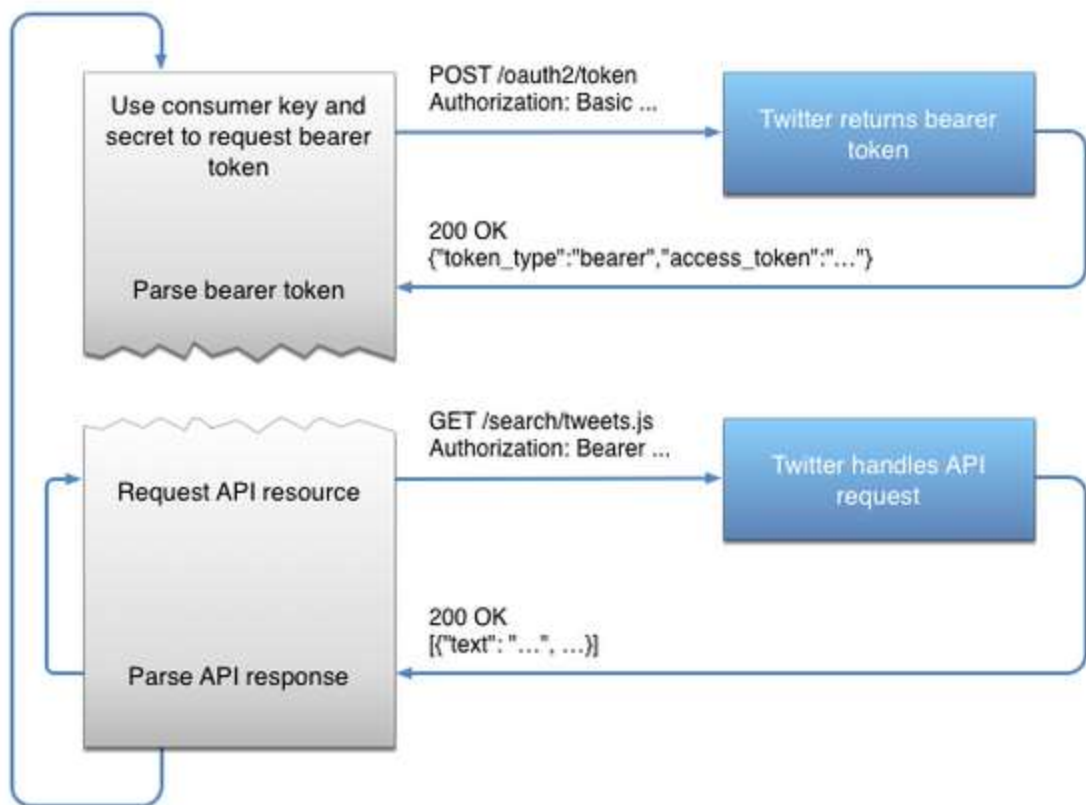


Figura 2.14. Funcionamiento de Twitter Outh

Fuente: (Developers, 2015)

2.15 Rup

Rational Unified Process (RUP) es una metodología de desarrollo de software orientado a objeto que establece las bases, plantillas, y ejemplos para todos los aspectos y fases de desarrollo del software, combinan los aspectos del proceso de

desarrollo (como fases definidas, técnicas, y prácticas) ayudado de otros aspectos en el desarrollo de software (como documentos, modelos, manuales, código fuente, etc.) dentro de un framework unificado.

RUP consta de los siguientes componentes como se muestra en la figura.

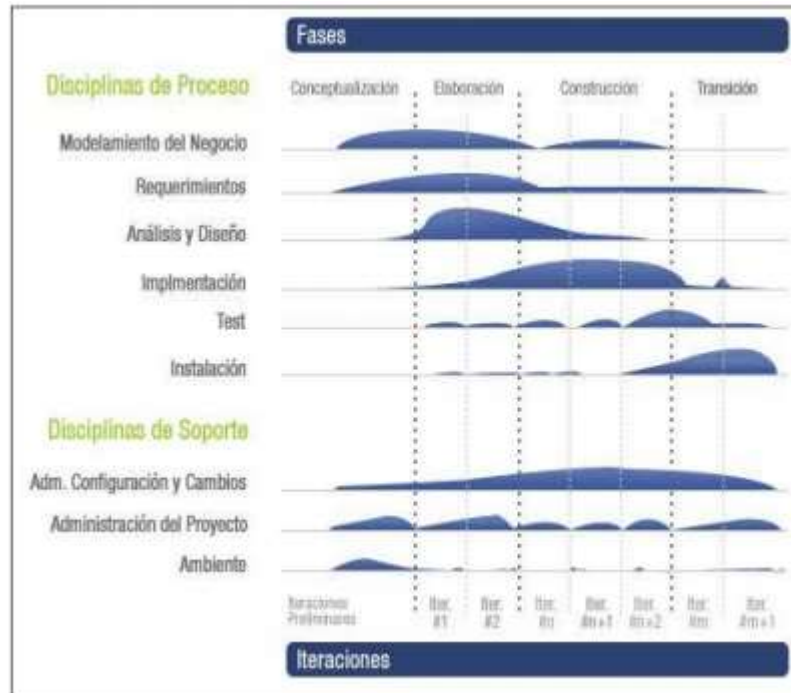


Figura 2.15 Esquema de la Metodología RUP

Fuente: (Jacobson, 2000)

2.15.1 Características de la Metodología (Rup)

Esta metodología se caracteriza de 3 maneras:

- Casos de Uso
- Centrado en la Arquitectura
- Iterativo e Incremental.

2.15.1 Casos De Uso

Los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario.

La descripción obtenida de los requerimientos debe ser comprendida por casos de uso que nos ayudan a recopilar la información acerca de la interacción que tiene los

usuarios en este caso actores con el sistema. Un caso de uso es una secuencia, reacciones que el sistema lleva a cabo para ofrecer un resultado de valor a algún actor, que sirven para realizar pruebas sobre los componentes desarrollados (ver Figura. 2.15). Los casos de uso enlazan los flujos de trabajo fundamentales.

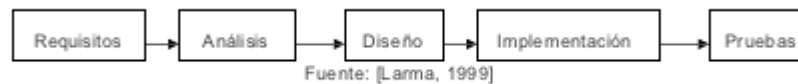


Figura 2.16: Proceso de un Caso de Uso.

Fuente: (Larman, 1999)

2.15.2 Centrado en la Arquitectura

Este proceso en la arquitectura del software compromete aspectos estáticos y dinámicos más importantes del sistema, la cual nace de las necesidades de la empresa y como son interpretadas por el usuario final descrita en los casos de uso.

La importancia de tener una arquitectura radica en poder entender todo lo que un sistema va a realizar y funcionar, ya que así las partes involucradas en su desarrollo logren un correcto funcionamiento del mismo.

Al tener un conocimiento claro del funcionamiento de lo que se vaya a desarrollar en el software y con qué herramientas poder emplear una solución a dicho problema se podrá establecer los componentes y requisitos que se podrán realizar en el sistema.

2.15.3 Iterativo E Incremental

Este proceso unificado Iterativo e incremental proporciona el desarrollo del producto de software en pequeños pasos manejables para su propósito.

Estos pasos se resumen en los siguientes según (Jacobson, 2000):

- Planificar un poco

- Especificar, diseñar e implementar un poco
- Integrar, probar y ejecutar un poco cada iteración.

Las iteraciones son pasos en el flujo de trabajo, mientras que los incrementos son el crecimiento del producto que logran ser más efectivos al momento del desarrollo del sistema.

Las ventajas de usar el proceso iterativo e incremental es la retroalimentación de cada ciclo por lo cual hace que el sistema se vaya perfeccionando a medida de los incrementos de estas iteraciones.

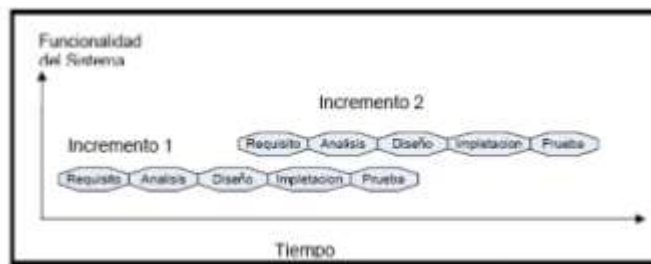


Figura 2.17: Proceso Iterativo/Incremental

Fuente: (Xavier Ferre Grau, 2005)

2.15.4 Lenguaje de Modelado Unificado (Uml)

El lenguaje unificado de diagrama o notación (UML) sirve como herramienta para especificar, visualizar esquemas de software orientado a objeto. El UML ayuda a tener una correcta visualización a la hora del diseño de esta manera lo hace más accesible para otros.

El UML está diseñado para un software orientado a objetos y se compone de muchos elementos de esquematización, las cuales dan forma a diferentes partes del sistema.

Los elementos se clasifican en los siguientes tipos de diagramas:

- Diagrama de casos de uso
- Diagrama de clases
- Diagrama de secuencia
- Diagrama de colaboración
- Diagrama de estado
- Diagrama de actividad
- Diagrama de componentes
- Diagrama de implementación
- Diagrama de relaciones de entidad

2.15.4.1. Diagrama de Casos de Uso

Describen y muestran los actores y procesos que se producen cuando utilizan en el sistema y son muy útiles al momento de desarrollar un software.

- Actores

Un actor es una entidad externa (fuera del sistema) que interacciona con el software participando en un caso de uso, tenemos por ejemplo como actores a los usuarios que usan el sistema, ordenadores, etc.

Estos actores representan un rol en específico dentro del sistema a desarrollarse. Es decir cuando una persona interactúa con el sistema de varias formas.



Figura 2.18: Representación de un Actor

Fuente: (Larman, 1999)

- Caso de Uso

El caso de uso es un documento narrativo que describe la secuencia de eventos de un actor (agente externo) que utiliza un sistema para completar un proceso. (Jacobson, 2000).

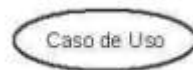


Figura 2.19: Representación de un Caso de Uso

Fuente: (Larman, 1999)

2.15.4.2. Diagrama de Casos de Uso de Alto Nivel

Son descripciones de un proceso determinado del sistema.

CASO DE USO:	Nombre del caso de uso
ACTORES:	Lista de actores
TIPO:	Primario, Secundario u opcional, Esencial o real
DESCRIPCIÓN:	Explica brevemente el proceso que realizan los actores en el caso de uso

Figura 2.20: Diagrama de caso de uso de alto nivel

Fuente: (Larman, 1999)

2.15.4.3. Diagrama de Casos de Uso de Expandido

Caso de Uso:	Registro de entrada de medicamentos
Actores:	Lista de actores
Propósito:	Intención del caso de Uso
RESUMEN:	Explica brevemente el proceso que realiza los actores en el caso de uso
Tipo:	Primario secundario u opcional
Referencias cruzadas:	Casos de uso y funciones también relacionadas del sistema
Curso normal de Eventos	
Acción de los actores	Respuesta de Sistema
Acciones numeradas de los actores	Descripciones numeradas de las respuestas del sistema

Figura 2.21: Diagrama de caso de uso Expandido

Fuente: (Larman, 1999)

2.15.5 Diagrama de Secuencia

Estos diagramas muestran el intercambio de mensajes dentro del sistema, estos ponen un énfasis especial en el orden y el momento en que se envían los mensajes a los objetos.

Los diagramas están representados como lo muestra la siguiente figura (véase)

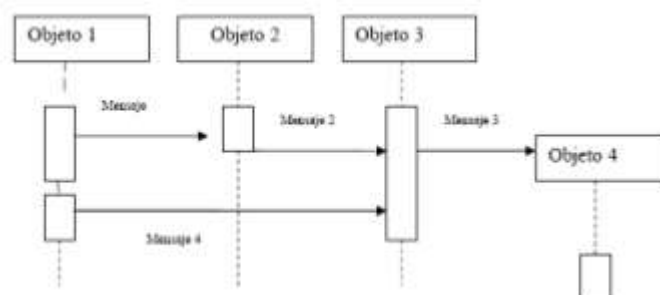


Figura 2.22: Diagrama de Secuencia

Fuente: (Jacobson, 2000)

2.15.6 Diagrama de Clases

Los diagramas de clases muestran las diferentes clases que componen el sistema y como están relacionadas. Estos diagramas muestran la estructura y el comportamiento de cada uno de los objetos del sistema.

Estos diagramas de clases según su tipo de relación son:

- La Asociación que representan un conjunto de enlaces entre objetos o instancias de clases.
- Herencia que indica que una subclase hereda los métodos y atributos especificados por una superclase hereda los métodos por ende la subclase, además de poseer sus propios métodos y atributos poseerá las características y atributos visibles de la superclase.
- La agregación que es un tipo de relación jerárquica entre un objeto que representa la totalidad de ese objeto y las partes que la componen.
- Permite el agrupamiento físico de estructuras relacionadas lógicamente.

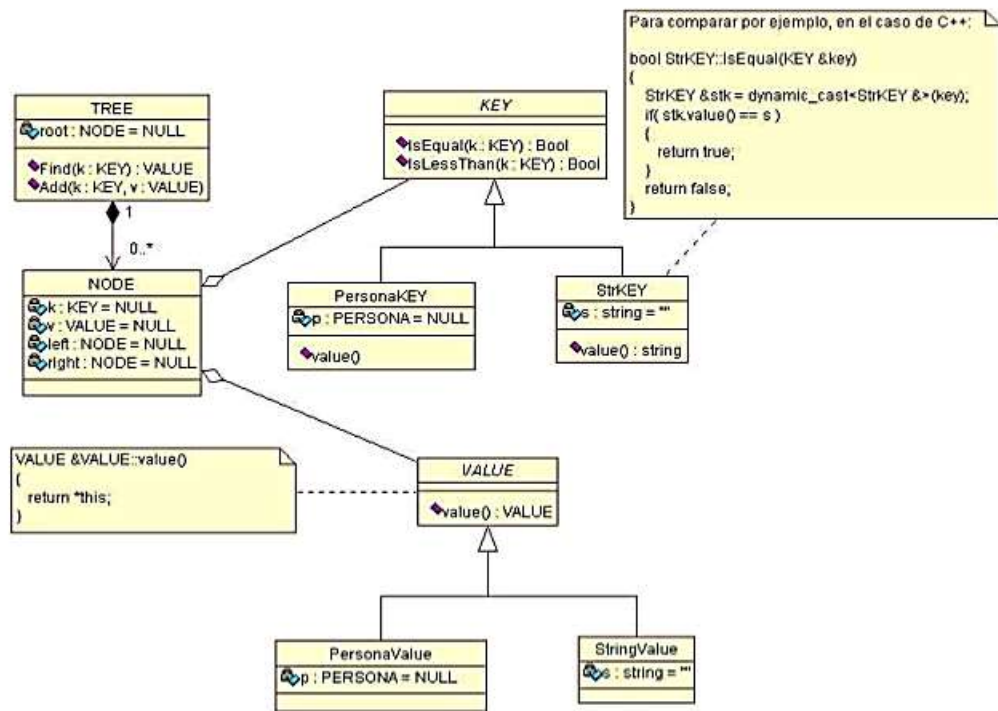


Figura 2.23: Diagrama de Clases
Fuente: (Xavier Ferre Grau, 2005)

2.16 Bootstrap

Es un Framework CSS para diseño de sitios y aplicaciones en la web esta contiene elementos CSS y HTML:

- Plantillas
- Formularios
- Botones
- Menús de Navegación

2.17 Funcionamiento del Sistema

- Consta de 4 Fases que son Requerimiento del proyecto, Presupuesto, Seguimiento y por último los Donantes que se han unido a la causa
- La entidad benéfica que da a conocer el proyecto podrá establecer el requerimiento del proyecto con los sigtes puntos: Resumen del Proyecto, Antecedentes y localización Objetivos del Proyecto y Beneficiarios.
- La entidad tiene la obligación de mostrar un presupuesto detallado del costo total del proyecto a realizarse.
- Se mostrara un Timeline con fotos, videos del desarrollo del proyecto.
- Se podrá visualizar a los donantes de la causa los mismos podrán seguir las actualizaciones del proyecto, compartir mediante publicaciones en redes sociales y además de poder dar su aportación mediante una plataforma de dinero electrónico o donaciones de víveres no perecibles.

El funcionamiento de dicha aplicación móvil estará llevada por una web la cual receptara los formularios de todas las instituciones benéficas que desean ser ayudadas en el proceso de recolección de fondos para sus diversos proyectos, los cuales tendrán un tiempo prudencial para poder financiar la totalidad de su proyecto, sino lo logra no se ejecutaran las aportaciones de los donantes y se les devolverá su aportación.

Para esta propuesta se llevara una plataforma compuesta por un framework en Django (python) la cual ofrecerá un servicio web que interactuara con Android para los donantes. En el caso de Android el donante podrá acceder mediante sus cuentas de redes sociales en este caso se usara Twitter para poder interactuar con los proyectos.

El método para donar monetariamente será de PayPal en caso de que las instituciones busquen ese tipo de financiamiento, si es en el caso no monetario el donante pondrá a

su disposición de donación de alimentos no perecibles los cuales serán receptados por la entidad benéfica en este caso.

2.18. Hipótesis

Hipótesis principal: La aplicación brindara un mejor mecanismo de captación de fondos para los proyectos de las ONG.

Hipótesis segunda: Transmitir la importancia de los proyectos a los donantes mediante el uso de la aplicación.

2.18.1 Variables E Indicadores

Tabla 2.1. Variables e indicadores

VARIABLES	DEFINICIÓN	DIMENSIONES	INDICADORES
Acceso a recursos económicos	Posibilidad de generar recursos económicos	Tiempo límite de obtener dinero	Tiempo que tarda cada institución en obtener los recursos para sus proyectos o ayudas a situaciones apremiantes
Difusión	Facilidad de difundir proyectos o ayudas emergentes de las instituciones benéficas	Agilizar y propagar proyectos de cada institución benéfica	Conocimiento de los proyectos benéficos en las redes sociales.

Fuente: Autor

CAPÍTULO 3

ANÁLISIS DEL SISTEMA

Dentro del estudio actual la fundación no maneja un proceso informático para la recaudación de las donaciones, la captación de las donaciones son dejadas en sus instalaciones o de ser posible recogidas donde el donante indique el momento de la entrega.

3.1 Fase de Planeación

3.1.1 Descripción de los Actores

Son los usuarios que interactúan, alimentan y sacan información del sistema de acuerdo a sus necesidades.

A continuación se da una lista de los actores a interactuar con el sistema:

Administrador: Es el encargado de analizar, validar y verificar los proyectos a publicarse dentro de la aplicación sus funciones son:

- Aceptar los proyectos a publicarse.
- Verificar la información del proyecto.
- Mantenimiento a la infraestructura.
- Reportes de proyectos terminados y no terminados.

Donante: Es la persona que sigue las actualizaciones y comparte los proyectos para dar su donación a las necesidades publicadas de la fundación.

- Compartir por redes sociales los proyectos.
- Donar.
- Ver proyecto.

3.1.2 Identificación de los Casos de Uso

La aplicación de donaciones en android está constituida en los siguientes casos de uso:

Tabla 3.1: Identificación de Casos de Uso.

Actor	Caso de Uso
Administrador	<ul style="list-style-type: none"> • Aprueba los Proyectos. • Mantenimiento de la Aplicación. • Notificaciones de los Proyectos.
Representante Fundacion	<ul style="list-style-type: none"> • Registro de Proyectos. • Actualiza el proyecto con Avances de lo recaudado.
Donante	<ul style="list-style-type: none"> • Registra Donaciones a los proyectos. • Comparte los proyectos a redes sociales.

	<ul style="list-style-type: none">• Sigue a los proyectos que aporó.
--	--

Fuente: Autor

3.1.3. Requerimientos del Sistema

- Creación de proyectos de la fundación.
- Aprobación de proyectos de la fundación.
- Captar donaciones de proyectos.
- Publicar actualizaciones de los proyectos en ejecución.
- Reporte de proyectos en ejecución y Finalizados.

3.1.4 Definición de Roles de los Módulos

Las funciones de la aplicación son lo que esta podrá hacer, estos requerimientos se detallaran a continuación con la categoría evidente y oculta.

- Evidente: Función que debe realizarse, y el usuario debería saber que se ha realizado.
- Oculta: Debe realizarse, pero no es visible para el usuario.
- Superflua: Es opcional no repercute en el funcionamiento del sistema.

Tabla 3.2: Registro del Proyecto.

#	Función	Categoría
R.1.1	Llenar el Formulario del Proyecto	Evidente
R.1.2	Envió de mail con los datos del proyecto	Oculto
R.1.3	Se Acepta o Deniega el proyecto	Evidente
R.1.4	Se publica el proyecto cuando es aceptado	Evidente

Fuente: Autor

Tabla 3.3: Registro de Notificaciones al Donante.

#	Función	Categoría
R.2.1	Publica actualizaciones del proyecto	Evidente
R.2.2	Se envía una notificación a los donantes del proyecto	Oculto
R.2.3	Verifica los montos obtenidos del Proyecto	Evidente

Fuente: Autor

Tabla 3.4: Registro de Donaciones.

#	Función	Categoría
R.3.1	Elegir el proyecto	Evidente
R.3.2	Monto a donar	Evidente
R.3.3	Se incrementa el monto al presupuesto del proyecto	Oculto

Fuente: Autor

Tabla 3.5: Registro de Notificaciones a la fundación.

#	Función	Categoría
R.4.1	Verifica actualizaciones con el proyecto.	Evidente
R.4.2	Genera reporte de los ingresos donados.	Oculto
R.4.3	Se envía una notificación por los montos donados.	Oculto
R.4.4	Se envía una notificación cuando termine el proyecto.	Evidente

Fuente: Autor

3.2. Funciones Básicas

A continuación se detallaran los diferentes casos de uso de alto nivel que usara la aplicación para explicar los requerimientos de los actores.

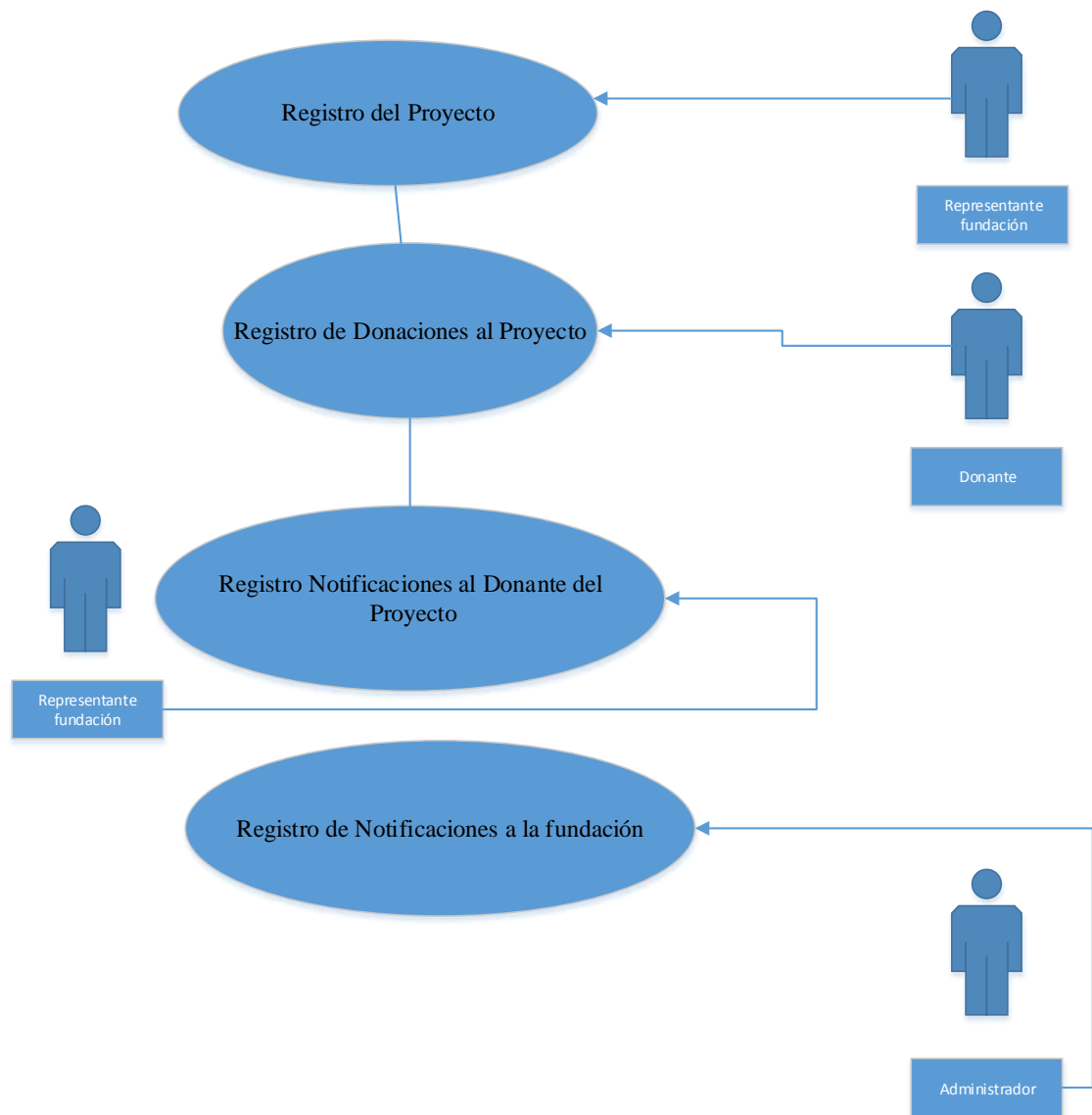


Figura 3.1 Diagrama de Caso de Uso Principal de la Aplicación

Fuente: Autor

3.3. Factibilidad Técnica

La tecnología que se ha propuesto funcionara de una manera correcta una vez puesta en produccion, a su vez el ciclo de vida del mismo. Estara a sujeta a cambios que no influiran en la estructura principal del programa.

Para el funcionamiento de la aplicación se requieren los siguientes componentes:

Una PC clon que sera usada como servidor.

- Intel Core I3

- 16 GB de RAM
- 1 TB de Disco Duro
- Ubuntu Server

Base de Datos

MySQL del lado del Servidor combinada con SQLite.

Herramientas de Programación

Python

Framework Django

Android

3.3.1 Ventajas y Desventajas del Software

Django

Ventajas

Altamente adaptable (Multiplataforma)

Desarrollo Agil

Facil de Mantener

Optimiza los tiempos de consulta dentro de la aplicación.

Diseños de Plantillas Amigables.

Maneja Modelo, Vista, Controlador

Desventajas

Curva de aprendizaje del framework es muy amplio.

No es tan simple de iniciar los servicios.

Android

Ventajas

Es de Código Abierto.

Funciona en varios dispositivos Móviles.

Buena gestion de las aplicaciones.

Personalizable por el usuario.

Desventajas

Es multitarea (esto implica que la gestión de la duración de la batería se vea afectada)

Android es totalmente fragmentado y eso provoca problemas de incompatibilidad con algunas aplicaciones debido a las versiones del S.O.

Las actualizaciones dependen del fabricante del smartphone.

Paypal

Ventajas

Es cómodo para hacer los pagos.

Funciona como método de pago en varios sitios web.

Fácil de usar.

Es seguro y sus transacciones encriptadas.

Desventajas

Cargos a terceros al momento de pagar, es decir obtiene un porcentaje por la transacción.

Mysql

Ventajas

Es open source.

Veloz al realizar las operaciones que el usuario requiere en las consultas.

Multiplataforma y de bajo costo en requerimientos de hardware.

Es seguro y sus transacciones encriptadas.

Desventajas

No es intuitivo.

Ciertas utilidades de Mysql no están documentadas, lo que hace un poco difícil manejar ciertas herramientas dentro del mismo.

Ubuntu

Ventajas

Es software libre y gratuito.

Es seguro para el usuario, por esto es escaso la cantidad de virus.

Eficiente al momento de cargar los programas.

Sus actualizaciones son gratuitas.

Desventajas

Compatibilidad con algunos componentes de hardware.

Instalacion de los programas mediante consola.

Compatibilidad con programas comunes con otros sistemas operativos.

3.4. Factibilidad Operacional

La aplicación funcionara bajo los criterios expuestos en el diseño y no dara problemas tecnicos.

3.5. Factibilidad Económica

La aplicación año a año sera sometida a mantenimiento, por la cual se establece un presupuesto del mismo.

Analisis Costo-Beneficio del Proyecto

Software	Costo
Django	\$0
Mysql	\$0
Ubuntu Server	\$0
Android	\$0
Total	\$0

Otros Gastos	Costo
Mantenimiento	\$500
Transporte	\$100
Soporte Técnico	\$200
Implementación	\$700
Total	\$1500

Total de Gastos del Sistema

Software Utilizado	\$ 0
Gastos operacionales	\$1500
Total	\$1500

3.6 Descripción Casos de Uso de Alto Nivel

A continuación se describirán los diferentes de casos de uso:

Tabla 3.6: Caso de Uso Registro de Proyectos.

Caso de Uso	REGISTRO DE PROYECTOS
Actores:	Administrador, Representante Fundacion
Tipo :	Primario
Descripción:	El Representante de la fundación realiza el registro del proyecto en la aplicación, el administrador aprueba o deniega el proyecto y publica el proyecto cuando es aprobado.

Fuente: Autor

Tabla 3.7: Caso de Uso Registro de Notificaciones al Donante.

Caso de Uso	REGISTRO DE NOTIFICACIONES AL DONANTE
Actores:	Representante de la Fundación
Tipo :	Primario
Descripción:	El Representante de la fundación actualiza el contenido del proyecto con lo recaudado o avances de la realización del mismo.

Fuente: Autor

Tabla 3.8: Caso de Uso Registro de Donaciones.

Caso de Uso	REGISTRO DE DONACIONES
Actores:	Donante
Tipo :	Primario
Descripción:	El Donante realiza la donación al proyecto en la aplicación, Comparte el proyecto en las redes sociales.

Fuente: Autor

Tabla 3.9: Caso de Uso Registro de Notificaciones a la fundación.

Caso de Uso	REGISTRO DE NOTIFICACIONES A LA FUNDACION
Actores:	Administrador
Tipo :	Primario
Descripción:	El Administrador de la aplicación notifica novedades del proyecto publicado y cuando el mismo finaliza.

Fuente: Autor

3.7 Casos de Uso Expandido

A continuación se detallaran los casos de uso expandido.

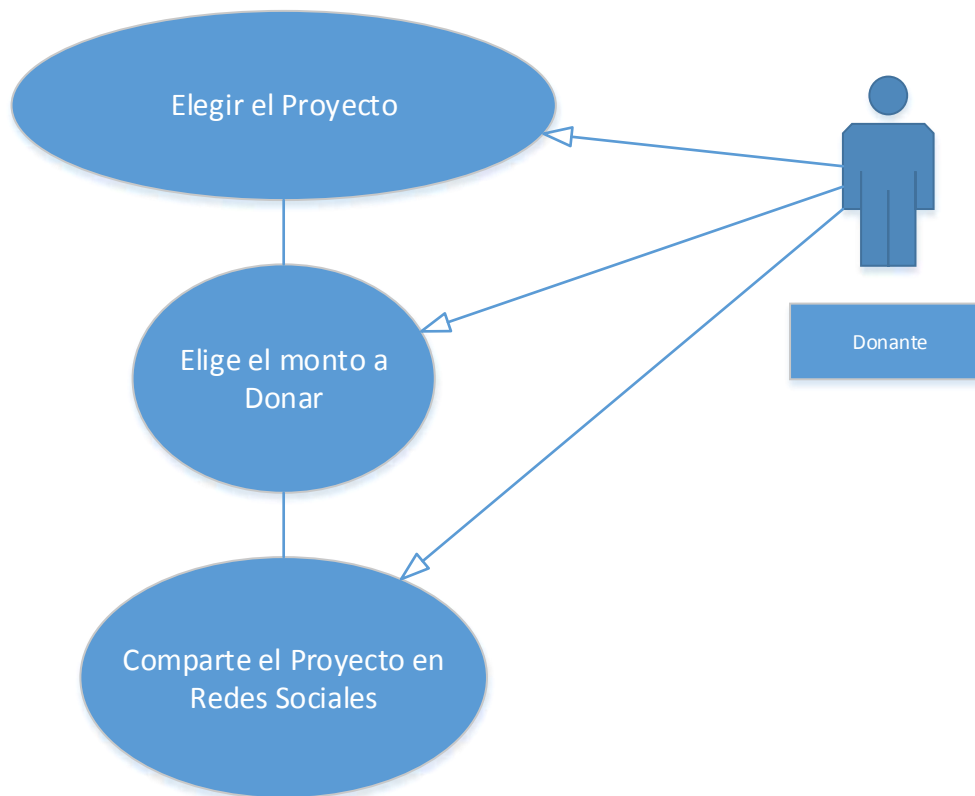


Figura 3.2 Diagrama de Caso de Uso de Registro del Proyecto

Fuente: Autor

Tabla 3.10: Caso de Uso Extendido Registro de Donaciones.

Caso de Uso	REGISTRO DE DONACIONES
Actores:	Donante
Propósito:	Registra las donaciones de los proyectos
Resumen:	El Donante procede hacer la donación, el monto va directo al proyecto y comparte el proyecto en las redes sociales
Tipo:	Primario y Esencial
Referencias Cruzadas:	R2.1,R2.2,R.2.3,R.2.4
Curso Normal de Eventos	
Acción de Actores	Respuesta del Sistema

<p>1.- El Donante procede hacer la donación al proyecto</p> <ul style="list-style-type: none"> • Elije el Proyecto • Fija el monto a Donar • Presiona el Botón Compartir <p>2.- El actor pone Guardar</p>	<p>3.- Se incrementa el Valor donado al monto del Proyecto.</p>
--	---

Fuente: Autor

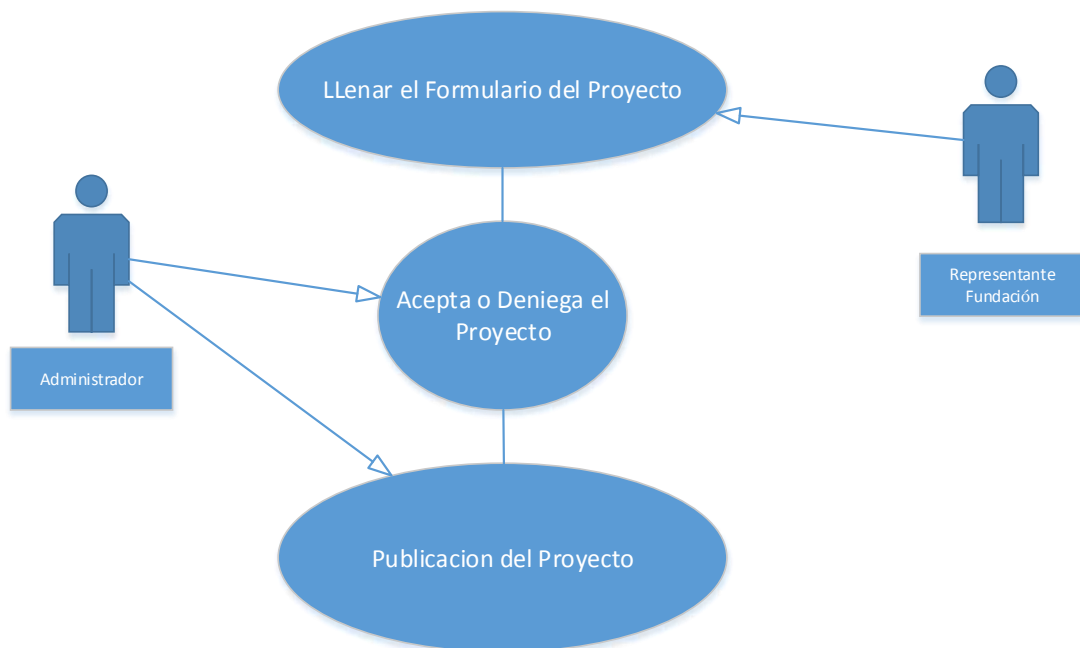


Figura 3.3 Diagrama de Caso de Uso de Registro del Proyecto

Fuente: Autor

Tabla 3.11: Caso de Uso Extendido Registro de Proyectos.

Caso de Uso	REGISTRO DE PROYECTOS
Actores:	Representante Fundacion
Propósito:	Registra los proyectos en la aplicación
Resumen:	El Representante de la fundación ingresa el proyecto, el administrador aprueba y publica el proyecto.
Tipo:	Primario y Esencial
Referencias Cruzadas:	R1.1,R1.2,R.1.3,R.1.4
Curso Normal de Eventos	
Acción de Actores	Respuesta del Sistema
<p>1.- El Representante de la Fundacion ingresa el proyecto en la aplicación</p> <ul style="list-style-type: none"> • Título del Proyecto • Descripción • Tipo • Fotos • Monto <p>2.- El actor pone Guardar</p>	<p>3.- La aplicación envía un mail al representante que el proyecto está pendiente de aprobación.</p> <p>4.- El administrador aprueba o deniega el proyecto.</p> <p>5.- Publica el proyecto en la aplicación.</p>

Fuente: Autor

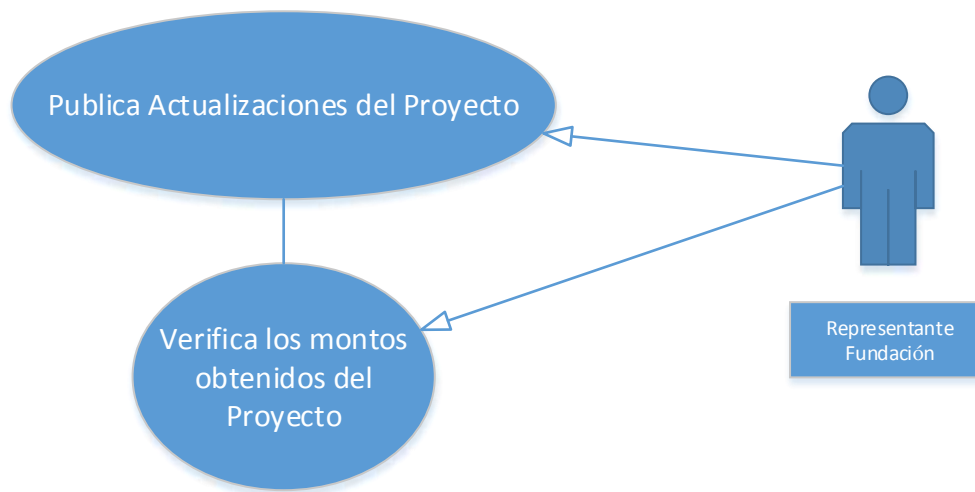


Figura 3.4 Diagrama de Caso de Uso de Registro del Proyecto

Fuente: Autor

Tabla 3.12: Caso de Uso Extendido Registro de Notificaciones a los Donantes.

Caso de Uso	REGISTRO NOTIFICACIONES A LOS DONANTES
Actores:	Representante Fundación
Propósito:	Publica las actualizaciones de los proyectos en la aplicación
Resumen:	El Representante de la fundación publica nuevas actualizaciones y avances del

	proyecto.
Tipo:	Primario y Esencial
Referencias Cruzadas:	R3.1,R3.2, R3.3
Curso Normal de Eventos	
Acción de Actores	Respuesta del Sistema
<p>1.- El Representante de la Fundacion ingresa al proyecto avances acerca del mismo.</p> <ul style="list-style-type: none"> • Publica avances del proyecto • Verifica los montos Obtenidos hasta el momento de recaudación <p>2.- El actor pone Guardar</p>	<p>3.- Graba las actualizaciones hechas por el actor.</p> <p>4.- Publica las actualizaciones hechas.</p>

Fuente: Autor

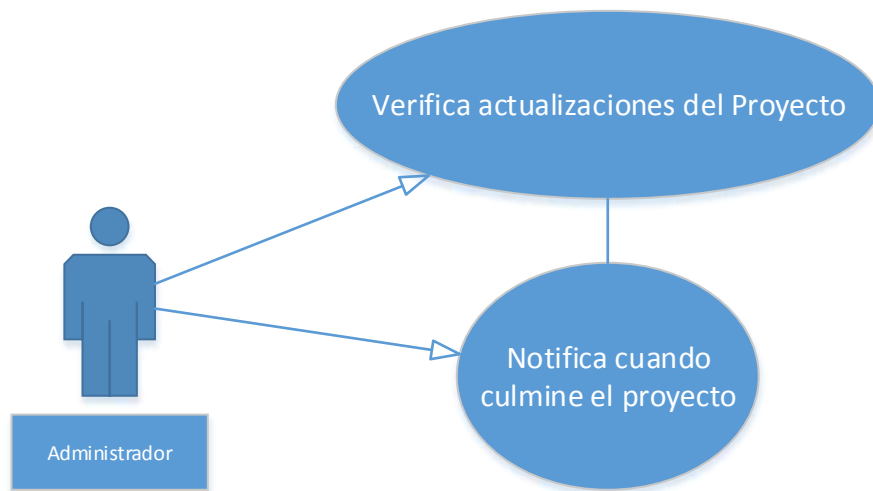


Figura 3.5 Diagrama de Caso de Uso de Registro de Notificaciones a la Fundación

Fuente: Autor

Tabla 3.13: Caso de Uso Extendido Registro de Notificaciones a la fundación.

Caso de Uso	REGISTRO DE NOTIFICACIONES A LA FUNDACION
Actores:	Administrador.
Propósito:	Verificar novedades de los proyectos publicados.
Resumen:	El Administrador de la aplicación revisa las actualizaciones de los proyectos publicados.
Tipo:	Primario y Esencial
Referencias Cruzadas:	R14.1,R4.2,R.4.3,R.4.4
Curso Normal de Eventos	
Acción de Actores	Respuesta del Sistema
1.- El administrador de la aplicación	

<p>verifica las actualizaciones de los proyectos en ejecución.</p> <ul style="list-style-type: none"> • Revisa el tiempo límite de los proyectos. • Notifica alguna novedad con los proyectos en ejecución. • Da por finalizado los proyectos al momento de que se cumple el monto. <p>2.- El actor pone Guardar</p>	<p>3.- La aplicación envía un mail al representante con las novedades del proyecto.</p> <p>4.- La aplicación cierra el proyecto cuando esta cumple el tiempo de recaudación de fondos.</p>
---	--

Fuente: Autor

3.8 Diagrama de Secuencia

El diagrama de secuencia representa la forma en cómo se logran comunicar los objetos al transcurrir el tiempo en orden de las llamadas o eventos del sistema. Un evento del sistema es una entrada externa que se origina como una operación del sistema como respuesta a un evento, representado secuencialmente.

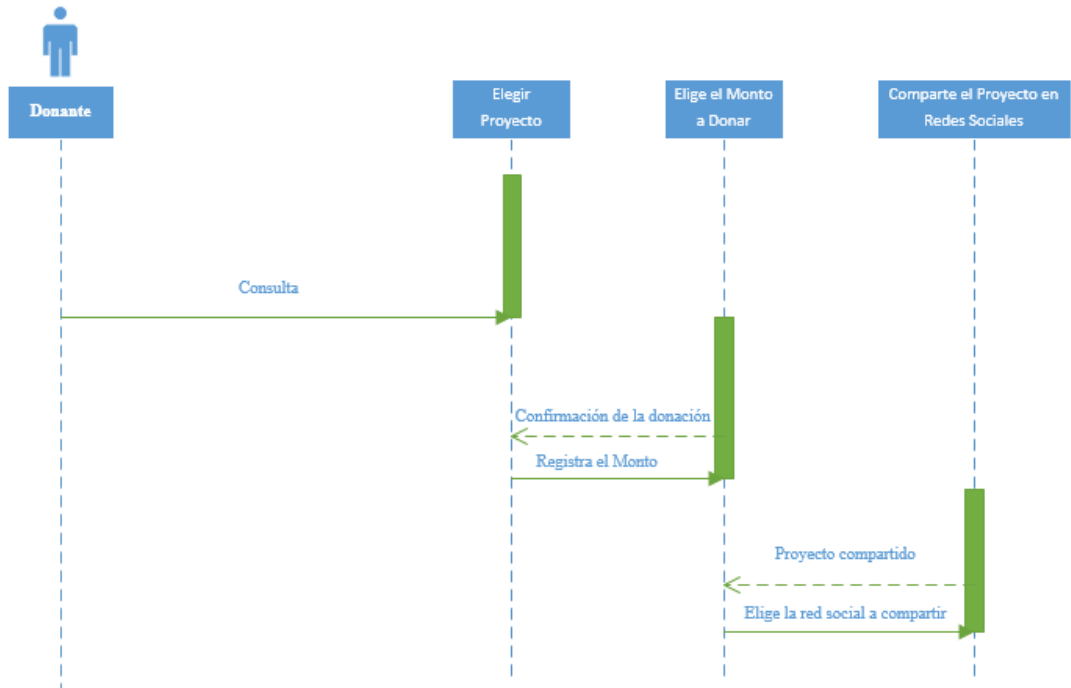


Figura 3.6 Diagrama de Caso de Uso de Registro de la Donación

Fuente: Autor

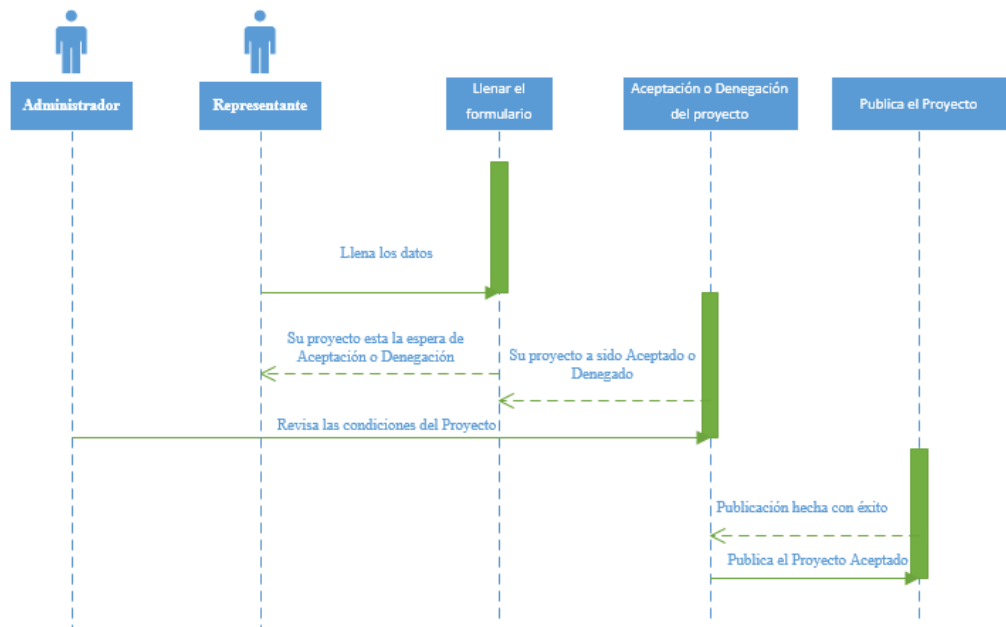


Figura 3.7 Diagrama de Caso de Uso de Registro del Proyecto

Fuente: Autor

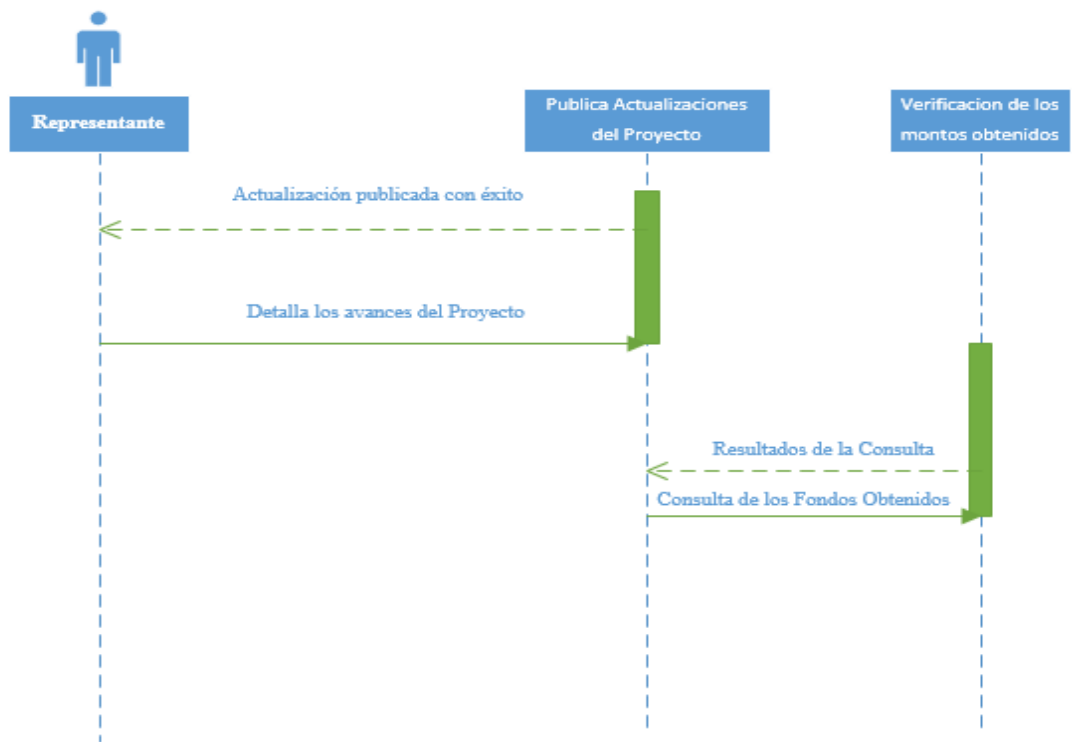


Figura 3.8 Diagrama de Caso de Uso de Registro del Notificaciones al Donante

Fuente: Autor

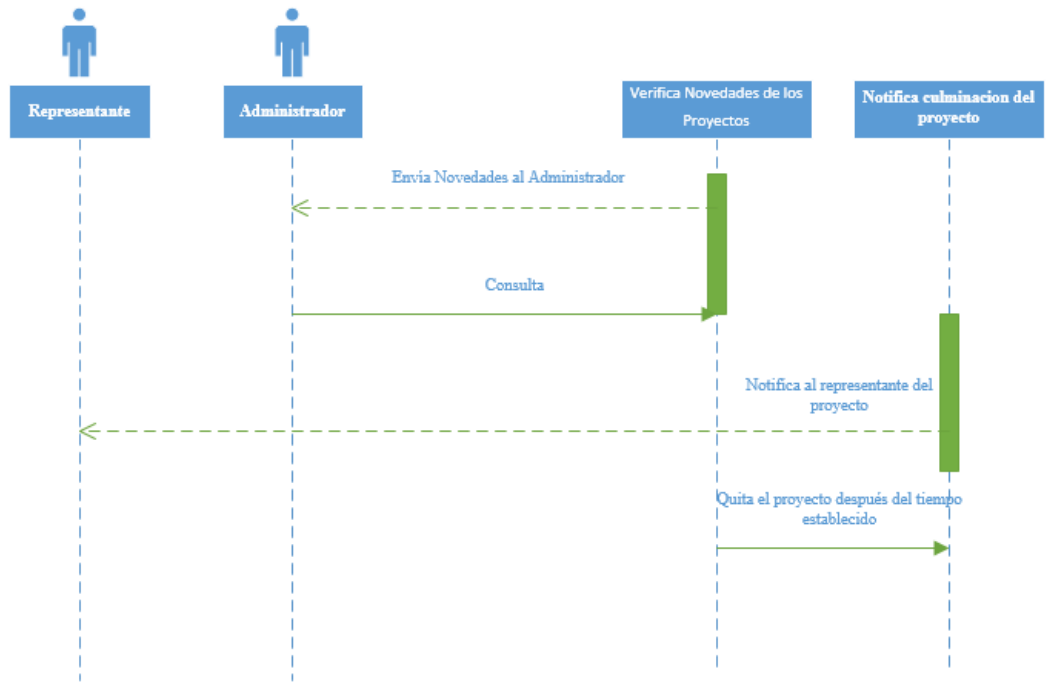


Figura 3.9 Diagrama de Caso de Uso de Registro de Notificaciones a la Fundación

Fuente: Autor

3.9 Diagrama de Clases

Este diagrama describe gráficamente la estructura de un sistema mostrando, clases y atributos y las relaciones entre ellos.

Se define a una clase como categoría o grupo de cosas que tiene atributos o acciones similares.

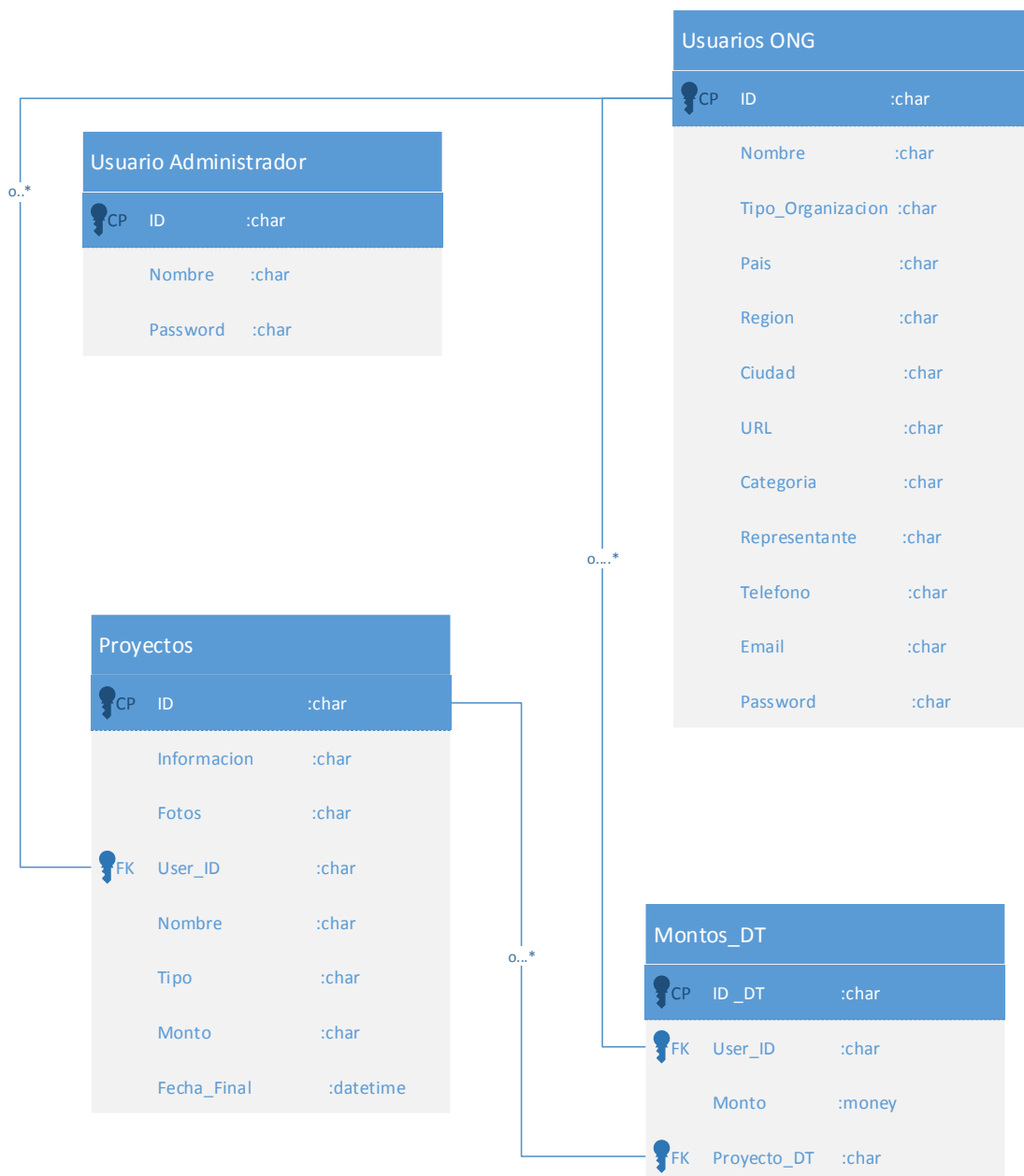


Figura 3.10 Diagrama de Clases del Proyecto

Fuente: Autor

3.10 Esquema de Base de Datos

A partir del diagrama de clases representamos el esquema de base de datos de manera gráfica en el figura 4.10



Figura 3.11 Diagrama Entidad-Relación

Fuente: Autor

3.11 Arquitectura del Sistema

La arquitectura de la aplicación está constituida con la información fundamental y el manejo de la aplicación.

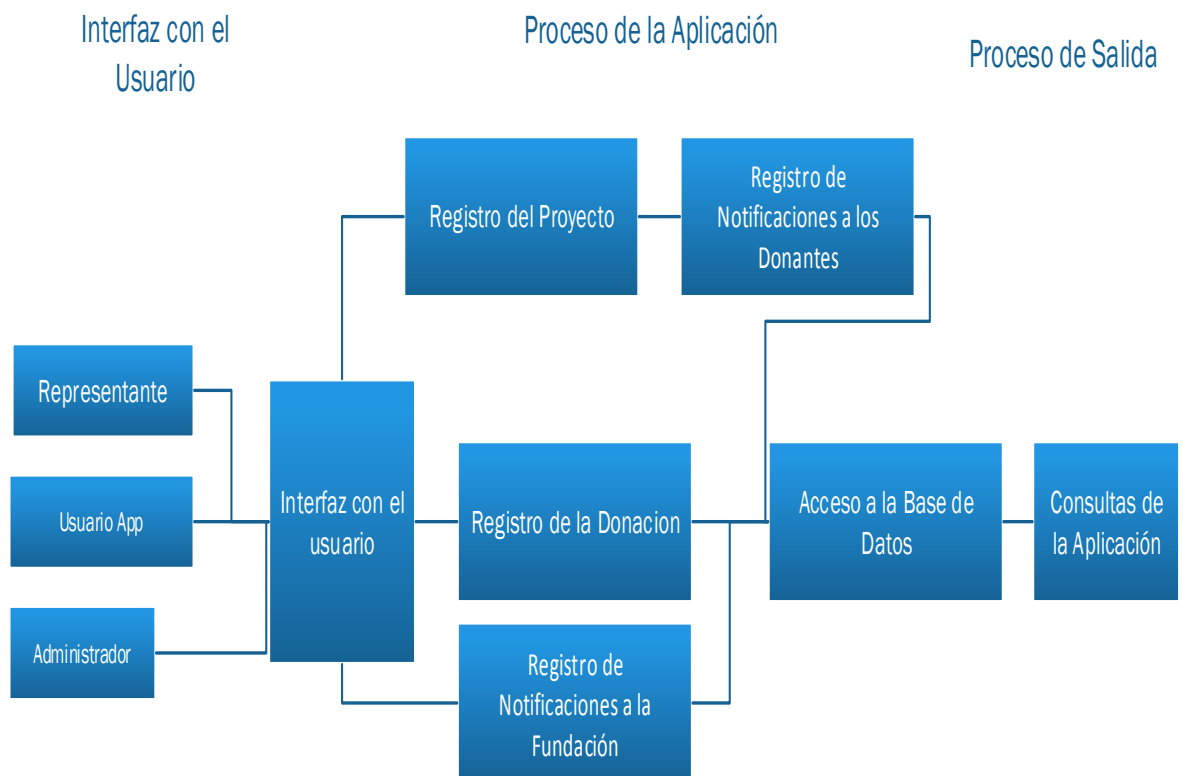


Figura 3.12 Arquitectura de la Aplicación

Fuente: Autor

A continuación se muestra un esquema de arquitectura de la aplicación en la figura 3.12 con dispositivos con los cuales se comunicara.

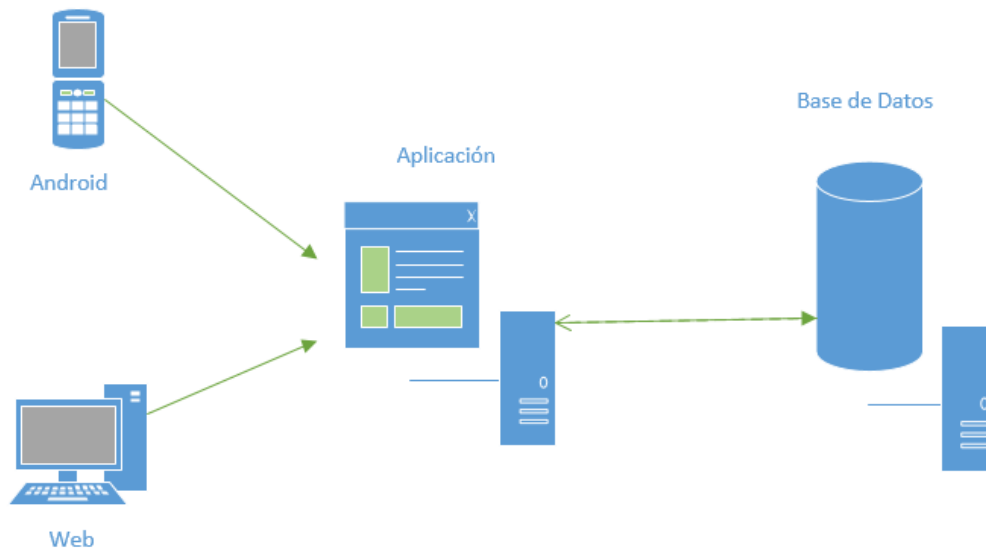


Figura 3.13 Esquema de la Aplicación

Fuente: Autor

Como se puede ver en la figura 3.13 el esquema de la aplicación están dadas por los siguientes componentes:

Servidor de la Aplicación que interactúa con la base de datos, aplicación móvil y la web.

La aplicación móvil que presentará los diferentes proyectos almacenados en la base de datos, los mismos que son ingresados mediante la web por los responsables de la fundación y son aprobados por el administrador de la aplicación.

CAPÍTULO 4

PROPUESTA

4.1 Propuesta de la Aplicación

La propuesta de aplicación para la fundación estara realizada con el Sistema Operativo Android,a la misma que usara los siguientes componentes en interfaz gráfica:

- Una línea de tiempo de Imágenes que presentara los avances del proyecto en todas sus fases, la cual ayudara a los donantes a seguir el proyecto en la cual han hecho su contribución.
- Botones de Donar y Compartir dichos botones ayudarán a concretar dos de los objetivos del proyecto que son la de poder acercar al donante a contribuir con la causa presentada y difundiendo el proyecto por redes sociales.
- Una breve descripción del proyecto publicado.
- Con un video del proyecto en la cual dará a conocer con profundidad los objetivos, avances, agradecimientos por el apoyo al mismo.

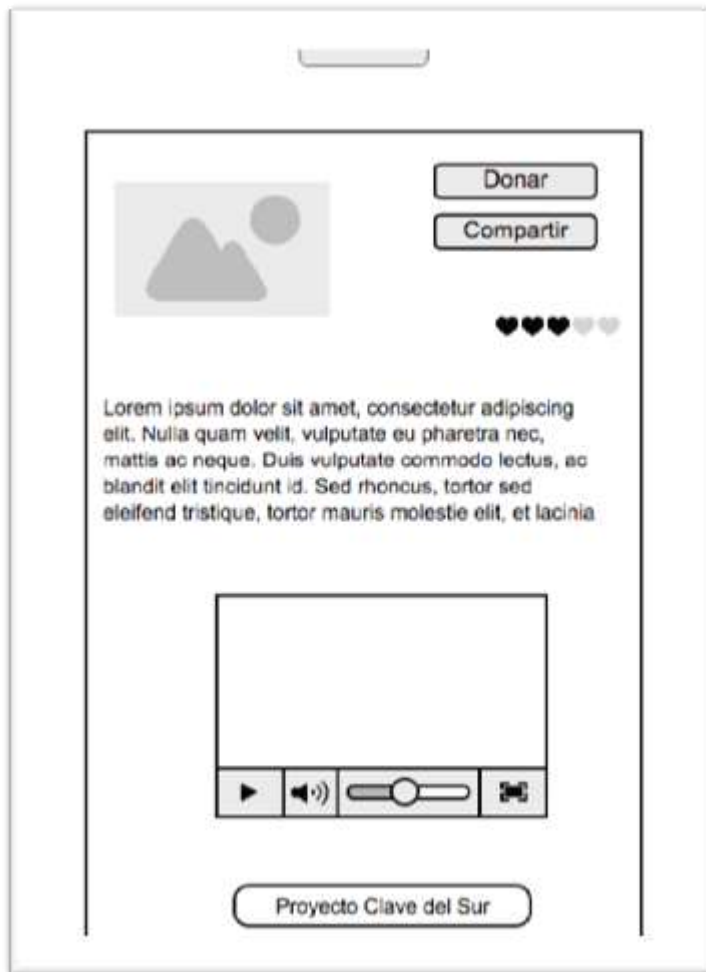


Figura 4.1 Interfaz de la Aplicación en Android

Fuente: Autor

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- En la actualidad, la fundación “Clave del Sur” recepta sus donaciones de instituciones privadas mediante acuerdos previamente conversados durante las diferentes presentaciones musicales de la escuela que promociona la fundación.
- Ayudar a organizaciones no gubernamentales como la fundación “Clave del Sur” a dar a conocer el impacto social de sus proyectos que están centrados en apoyar proyectos dentro de la comunidad con el aprendizaje de instrumentos musicales.
- Agilizar el tiempo de recaudación de los proyectos sociales que requiere la fundación compartiendo dichos proyectos mediante una interacción con aplicación en Android, la cual dará a conocer con ayuda de una interfaz de botones con la redes sociales.
- Obtener nuevos donantes mediante una aplicación móvil que facilitara las donaciones monetarias o lo que la fundación requiera para su recaudación, de esta manera fomentando una nueva herramienta de donaciones.

5.2 Recomendaciones

Con la finalización de este proyecto surgieron las siguientes recomendaciones:

- Utilizar las redes sociales como herramienta de difusión de causas sociales.
- Usar con responsabilidad la aplicación gestionando un correcto uso del usuario.
- Establecer con otras fundaciones para que usen la aplicación que se ha propuesto.

Trabajos Citados

- Bassi, S. (s.f.). *Python en 8 clases: Aprendiendo a programar con Python*.
- Chávez, M. A. (2008). *Concepto de Bases de datos*.
- Clave del Sur. (s.f.). Obtenido de Clave del Sur: www.clavedelsur.org
- Código Civil, art. 531-7.
- Developers, T. (2015). *Twitter Developers*. Obtenido de dev.twitter.com/oauth
- Django. (s.f.). *El libro de Django 1.0*. Obtenido de <http://django-book.mkkaufmann.com.ar/>
- Django, P. (s.f.). *django project.com*.
- Dodino, F. (2007). *Mapeo Objeto/Relacional*.
- Ecma. (2015). *Introducción a JSON*. Obtenido de Introducción a JSON: <http://www.json.org/json-es.html>
- Estudio de la Técnica ORM (2008).
- Facundo, B. (n.d.). *Organizaciones no Gubernamentales*.
- Fossati, M. (2007). Todo sobre MySQL. En M. Fossati, *Todo sobre MySQL* (pág. 5).
- Gilfillan, I. (s.f.). *Biblia de MySQL*. Amaya.
- Gironés, J. T. (2010). *El gran Libro de Android*.
- INEC. (2013). *Tecnologías de la información y comunicaciones (TICS) 2013*.
- InfoCrowdsourcing. (2014). *InfoCrowdsourcing*. Obtenido de InfoCrowdsourcing: <http://www.infocrowdsourcing.com/crowdfunding-espana-2013-19-millones/>
- Internacional, S. T. (2013). *Convenios ONG en Ecuador*.
- Jacobson. (2000). *El Proceso Unificado de Desarrollo de Software*.
- Jiménez, E. Á. (2010). Mejoras a detrás. En E. Á. Jiménez, *Mejoras a detrás: Describiendo la actividad humana frente al ordenador* (pág. 22).
- Larman. (1999). *UML y Patrones*.
- Marset, R. N. (2006). Diseño e Implementación de Servicios Web. En R. N. Marset, *Modelado, Diseño e Implementación de Servicios Web* (págs. 4-5).
- Mass. (2012). *Crowdfunding Industry Report*. Massolutions.
- Montoro, A. F. (2010). *Python al descubierto*.
- Mysql. (s.f.). *Mysql.com*.
- Pablo, M. (2005). *El Rol de las ONG en la reducción de la pobreza en América Latina*. Paris.
- Paypal. (2014). *Paypal*. Obtenido de Paypal: <https://developer.paypal.com/docs/integration/mobile/mobile-sdk-overview/>

Rubén, G. P. (2011). *Aplicación de Supermercados*, Tesis de Titulación en Ingeniería Técnica de Gestión. Barcelona, Cataluña, España.

Sqlite. (s.f.). *Sqlite.com*.

UNAM. (2015). *Fundamentos de Sistemas Operativos*.

Une Option de Plus. (s.f.). Obtenido de <http://www.uneoptiondeplus.org/?es/inicio>

Xavier Ferre Grau, M. I. (2005). *Desarrollo Orientado a Objetos con UML*.

Anexos

Manual de Creación de Entorno Virtual con Django

Paso 1: Se crea un entorno virtual el cual creara un directorio con el nombre proyecto.

```
sh-3.2# virtualenv 'app_donaciones'  
New python executable in app_donaciones/bin/python  
Installing setuptools, pip...done.
```

Paso2: Se instala la versión de Python para el proyecto en este caso se uso la versión 2.7

```
sh-3.2# easy_install virtualenv  
Searching for virtualenv  
Reading https://pypi.python.org/simple/virtualenv/  
Best match: virtualenv 12.0.7  
Downloading https://pypi.python.org/packages/source/v/virtualenv/virtualenv-12.0.7.tar.gz#md5=e08796f79d112f3bfa6653cc10840114  
Processing virtualenv-12.0.7.tar.gz  
Writing /tmp/easy_install-ZLHhp3/virtualenv-12.0.7/setup.cfg  
Running virtualenv-12.0.7/setup.py -q bdist_egg --dist-dir /tmp/easy_install-ZLHhp3/virtualenv-12.0.7/egg-dist-tmp-zvr04g  
warning: no previously-included files matching '**' found under directory 'docs/_templates'  
warning: no previously-included files matching '**' found under directory 'docs/build'  
Adding virtualenv 12.0.7 to easy-install.pth file  
Installing virtualenv script to /usr/local/bin  
Installing virtualenv-2.7 script to /usr/local/bin  
  
Installed /Library/Python/2.7/site-packages/virtualenv-12.0.7-py2.7.egg  
Processing dependencies for virtualenv  
Finished processing dependencies for virtualenv
```

Paso 3: Se activa el proyecto con el comando Source bin/activate para de esta manera trabajar dentro del directorio de la aplicación en la consola de Ubuntu.

```
sh-3.2# cd app_donaciones  
sh-3.2# source bin/activate  
(app_donaciones)sh-3.2# █
```

Paso 4: Con el comando `./manage.py runserver` se inicia el servidor Django en ubuntu.

```
(app_donaciones)sh-3.2# ./manage.py runserver
Performing system checks...

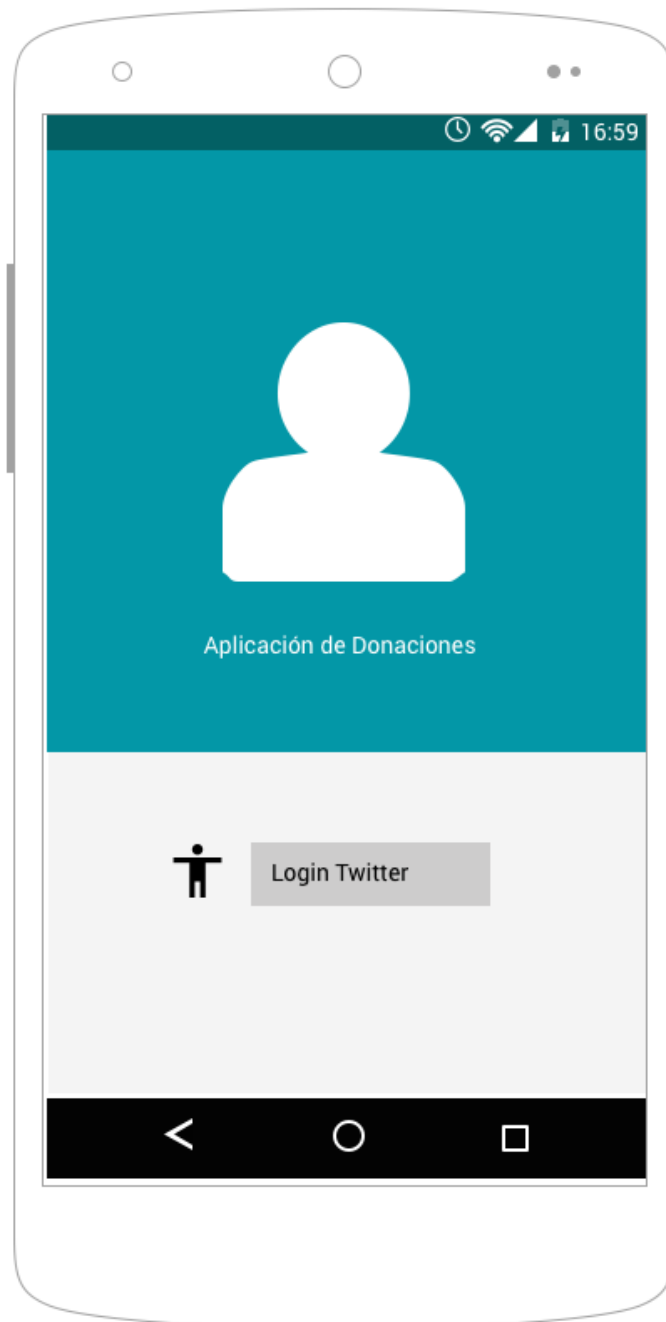
System check identified no issues (0 silenced).

You have unapplied migrations; your app may not work properly until they are applied.
Run 'python manage.py migrate' to apply them.

February 25, 2015 - 04:38:32
Django version 1.7.4, using settings 'app_donaciones.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Manual de Usuario

1. Se ingresa con el usuario del twitter para acceder a la aplicación de Donaciones.



Pantalla de Acceso a la Aplicación

1. El usuario debe autorizar el uso del inicio de sesión mediante twitter al momento que pone sus datos.

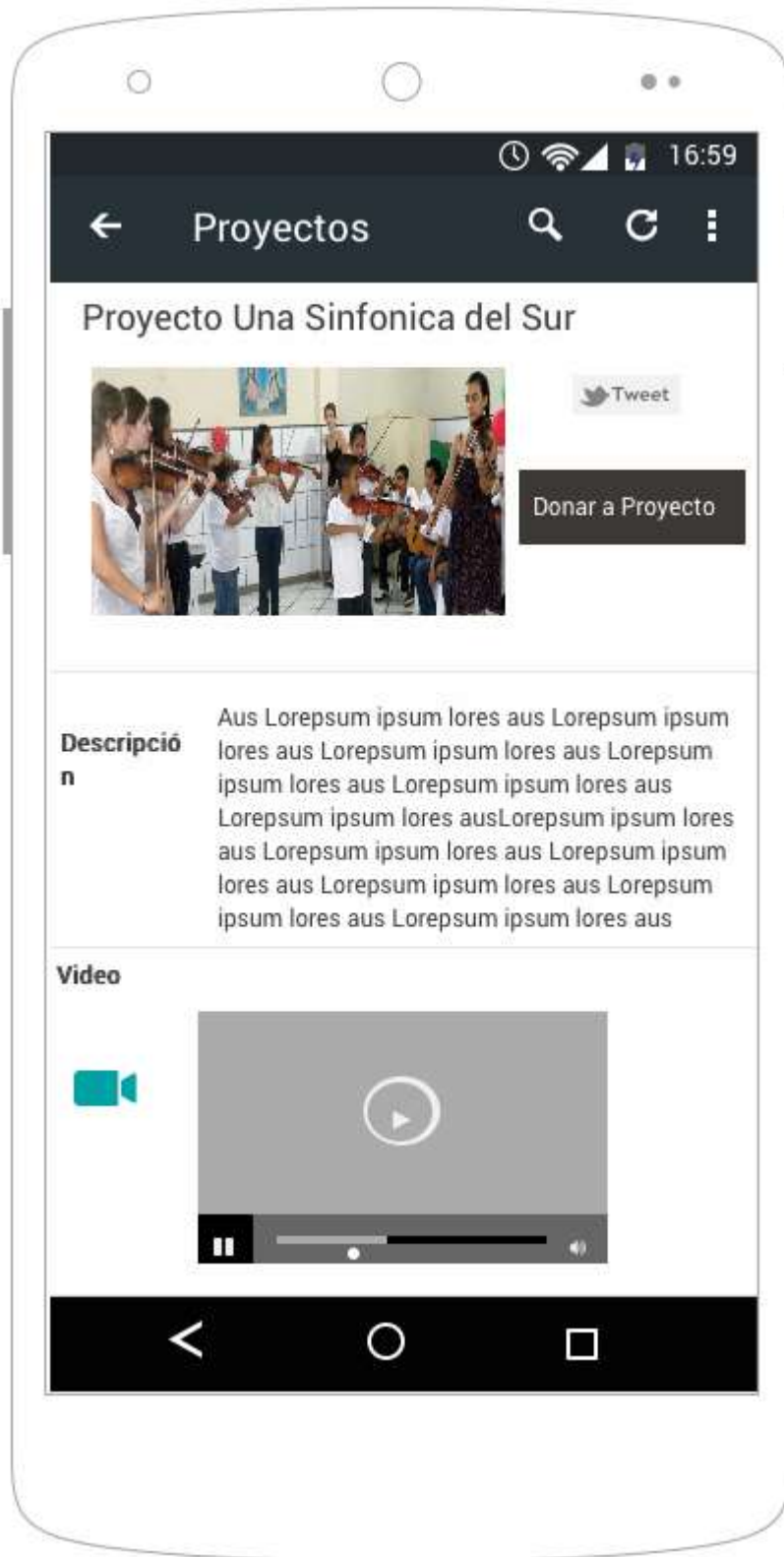


Pantalla de Autorización de la Aplicación Twitter.

3. Después de haber ingresado a la aplicación el usuario elige el proyecto que está a disposición en la aplicación.



4. Una vez elegido el proyecto el usuario accederá a la información del mismo dándole la posibilidad de donar y compartir el proyecto.



5. Después de donar y haber compartido el proyecto la aplicación le dará un agradecimiento por la colaboración realizada.



