ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UNA API DE INTEGRACIÓN DE SERVICIOS DE REDES SOCIALES EN APLICACIONES MÓVILES

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

TEX EDUARDO ALBUJA ESTÉVEZ
eduardo.albuja.estevez@gmail.com

Director: ING. TITO ROLANDO ARMAS ANDRADE
rolandoarmas@gmail.com

Quito, Abril 2013

DECLARACIÓN

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

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

Tex Eduardo Albuja Estévez

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Tex Eduardo Albuja Estévez, bajo mi supervisión.

Ing. Tito Armas

DIRECTOR

AGRADECIMIENTOS

Agradezco a los docentes de la Escuela Politécnica Nacional, quienes nos han forjado no solo como buenos profesionales, sino como grandes personas. En especial al Ing. Tito Armas, mi director de tesis, quien me ha guiado en esta etapa importante de mi carrera.

Agradezco a mis compañeros de aula con quienes hemos conseguido una armonía grupal para enfrentar cada nuevo reto durante nuestra formación.

DEDICATORIA

A Dios quien me ha bendecido con bienestar, salud y consejo en cada paso.

A mi madre, a su eterno amor y aliento a sus hijos,
motivos suficientes para que seamos personas de bien.

A Ronny mi hermano, quien llena de alegría nuestro hogar.

A cada miembro de mi familia, por saberme apoyar con su ejemplo
y coraje para enfrentar cada escalón de su vida.

A Armando y Mishell, por abrirme las puertas en su familia.

A Andrea, la mujer que tomó mi mano y escribió mi historia.

A Benja, Marco, Javier y Cristian mis más cercanos
amigos del colegio por seguir incondicionalmente unidos.

Dedico finalmente a todos mis demás amigos,
que ninguno falte en mi corazón.

CONTENIDO

PRESENTACIÓN 5

RESUMEN 2

CAPÍTULO 1. PLANTEAMIENTO DEL PROBLEMA.....3

1.1. PLATAFORMA ANDROID.....6

1.1.1. PILA DE SOFTWARE ANDROID 8

1.1.2. PRINCIPALES PAQUETES JAVA DE ANDROID SDK. 9

1.2. METODOLOGÍA OPENUP.....10

1.2.1. PRINCIPIOS. 10

1.2.2. ÁREAS DE CONTENIDO. 11

1.2.3. ROLES.....12

1.2.4. ARTEFACTOS.....12

1.3. CONSIDERACIONES DEL CONSUMO DE API'S DE LAS REDES SOCIALES. 16

1.3.1. FACEBOOK. 16

1.3.2. TWITTER. 17

1.3.3. GOOGLE PLUS. 18

1.3.4. ANÁLISIS COMPARATIVO ENTRE API'S. 20

1.4. CONSIDERACIONES DE INTEGRACIÓN. 21

1.4.1. OPENAUTH. 21

1.5. CONSIDERACIONES DE ESTÁNDARES DE PROGRAMACIÓN DE ANDROID. 23

1.5.1. ESTÁNDARES DE PROGRAMACIÓN ANDROID. 24

1.6. ANÁLISIS DE LAS SOLUCIONES EXISTENTES. 26

1.6.1. SHAREKIT 26

1.6.2. SOCIALLIB 26

1.6.3. PINKELSTAR.....26

1.6.4. ANÁLISIS COMPARATIVO DE LAS SOLUCIONES EXISTENTES. 27

1.7. CARACTERÍSTICAS DE LA SOLUCIÓN PROPUESTA. 28

CAPÍTULO 2. DESARROLLO DE LA BIBLIOTECA DE CLASES. 29

2.1. FASE DE INICIO. 29

2.1.1. INICIAR EL PROYECTO 29

2.1.2. PLANIFICAR Y ADMINISTRAR LA ITERACIÓN. 40

2.1.3. IDENTIFICAR Y REFINAR LOS REQUERIMIENTOS.....43

2.1.4. APROBAR EL ENFOQUE TÉCNICO. 53

2.2. FASE DE ELABORACIÓN. 54

2.2.1. PLANIFICAR Y ADMINISTRAR LA ITERACIÓN. 54
2.2.2. DESARROLLO DE LA ARQUITECTURA. 55
2.2.3. DESARROLLO DE UN INCREMENTO DE LA SOLUCIÓN. 58
2.3. FASE DE CONSTRUCCIÓN. 61
2.3.1. PLANIFICAR Y ADMINISTRAR LA ITERACIÓN. 61
2.3.2. DESARROLLAR LA DOCUMENTACIÓN DEL PRODUCTO 62
2.3.3. IMPLEMENTAR LA SOLUCIÓN 63

CAPÍTULO 3. ELABORACIÓN DE UN PROTOTIPO COMO CASO DE ESTUDIO....65

3.1. DEFINICIÓN DE LOS REQUISITOS DEL PROTOTIPO. 65
3.1.1. MODELO DE CASOS DE USO DEL PROTOTIPO 66
3.1.2. DISEÑO DE LAS INTERFACES GRÁFICAS DEL PROTOTIPO....66
3.2. ELABORACIÓN DEL PROTOTIPO. 70
3.3. PRUEBAS DEL PROTOTIPO. 70
3.4. ANÁLISIS DE RESULTADOS. 77
3.4.1. MINIMIZAR EL COSTO Y TIEMPO DE DESARROLLO DE APLICACIONES MÓVILES. 77
3.4.2. REDUCIR COSTOS DE INFRAESTRUCTURA TI RELACIONADOS CON LA IMPLANTACIÓN Y MANTENIMIENTO DE SERVIDORES, Y POR ENDE REDUCIR LA CONTAMINACIÓN DEL MEDIO AMBIENTE. 78
3.4.3. REQUERIR MENOS CAPACITACIÓN PARA EL DESARROLLO DE APLICACIONES MÓVILES. 79
3.4.4. BRINDAR MÁS SERVICIOS Y CARACTERÍSTICAS A LAS APLICACIONES MÓVILES. 80

CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES. 81

4.1. CONCLUSIONES. 81
4.2. RECOMENDACIONES. 83

BIBLIOGRAFÍA 84

GLOSARIO....85

ANEXOS. 87

LISTA DE TABLAS

Tabla 1.1 Tabla de Distribución de Plataformas en Dispositivos Android. 6
Tabla 1.2 Estructura de Librerías de Android SDK 9
Tabla 1.3 Artefacto Vision 13
Tabla 1.4 Artefacto Plan de Proyecto 13
Tabla 1.5 Artefacto Lista de Riesgos.....14
Tabla 1.6 Artefacto Modelo de Casos de Uso 14
Tabla 1.7 Características de API's de Redes Sociales 20
Tabla 1.8 Análisis Comparativo de las Soluciones Existentes 27
Tabla 2.1 Lista de Riesgos. 35
Tabla 2.2 Estimación de costos (Recursos Humanos). 36
Tabla 2.3 Estimación de costos (Hardware). 37
Tabla 2.4 Estimación de costos (Software). 37
Tabla 2.5 Estimación de costos (Logística e Insumos). 38
Tabla 2.6 Estimación de costos (Resumen General).....38
Tabla 2.7 Estimación de costos (Resumen General).....39
Tabla 2.8 Definición de Requerimientos. 44
Tabla 3.1 Casos de Uso del Prototipo. 66
Tabla 3.2 Vista Principal. 67
Tabla 3.3 Tip. 67
Tabla 3.4 Agenda - Ingresar.....67
Tabla 3.5 Agenda - Explorar. 67
Tabla 3.6 Agenda - Invitar. 68
Tabla 3.7 Alimentación - Ingresar.....68
Tabla 3.8 Alimentación - Explorar. 68
Tabla 3.9 Alimentación - Compartir. 69
Tabla 3.10 Deporte - Ingresar. 69
Tabla 3.11 Deporte - Registro. 69
Tabla 3.12 Deporte - Compartir.....70
Tabla 3.13 Costo de Infraestructura TI. 79

LISTA DE FIGURAS

Figura 1.1 Diseño Corporativo de Aplicaciones Móviles. 4
Figura 1.2 Diseño Social de Aplicaciones Móviles 5
Figura 1.3 Distribución de Dispositivos Android por Sistema Operativo. 7
Figura 1.4 Pila detallada de Software Android SDK. 8
Figura 1.5 Organización del trabajo y el enfoque en el contenido en OpenUP. 11
Figura 1.6 Mapa Conceptual de la Metodología OpenUP. 15
Figura 1.7 Especificaciones de la API de Facebook. 17
Figura 1.8 Especificaciones de la API de Twitter.....18
Figura 1.9 Especificaciones de la API de Google Plus. 19
Figura 1.10 Flujo de Autenticación y Autorización de Contenido. 22
Figura 2.1 Estructura de Paquetes de SocialSphere 58
Figura 2.2 Estructura de Librerías Referenciadas en SocialSphere. 59
Figura 2.3 Estructura de Clases en el Paquete SocialSphere. 59
Figura 2.4 Estructura de Clases que implementan la definición de socialsphere.model. 60
Figura 2.1 Verificación de Requisitos de Software. 64
Figura 2.2 Instalación silenciosa de Android SDK. 64
Figura 3.1 Resultados de la Solución SocialSphere. 78

PRESENTACIÓN

La relevancia de las redes sociales en la población ha orientado a empresas y desarrolladores a pensar en estrategias específicas para aplicaciones las cuales son excelentes oportunidades para darse a conocer en el medio local y en la red.

La interacción con las redes sociales desde dispositivos móviles es una tendencia que requiere cada vez más interactuar y gestionar contenidos, transacciones y comunicaciones desde la palma de la mano.

En el Ecuador, según los resultados del reporte anual de estadísticas sobre tecnologías de la información (Instituto Nacional de Estadística y Censos, 2011) se indica que cerca del 80% de hogares poseen un teléfono celular, de los cuales más de 500 mil son teléfonos inteligentes con capacidad de navegación e interacción en internet y multimedia. Adicionalmente de la población que utiliza teléfonos inteligentes para redes sociales, el 76,3% corresponde a personas de 25 a 34 años, seguidos de las personas de 16 a 24 años con 71,9%.

Con este antecedente los desarrolladores de aplicaciones móviles se ven en la necesidad de incluir dentro de sus productos interacciones con redes sociales las cuales implican el consumo de recursos humanos y tecnológicos los cuales dilatan el costo final de un proyecto. El desarrollo de una librería que abstraiga la lógica de estas interacciones y las exponga de una manera consistente y estandarizada es la motivación del presente proyecto.

El diseño social de aplicaciones móviles además permite a los desarrolladores eliminar costos relacionados con la infraestructura TI, definición de procesos de mantenimiento entre otros. Siendo este un aspecto importante a considerar para la conservación del medio ambiente.

RESUMEN

En el presente proyecto de titulación se desarrollará un API de integración de servicios de redes sociales en aplicaciones móviles que proporcionará un conjunto de procedimientos de uso general para abstraer la lógica de las interacciones con las redes sociales.

En el primer capítulo se plantean los objetivos que abordará la API, las consideraciones de las redes sociales en las cuales se implementará, de igual manera las consideraciones en el desarrollo de librerías bajo la plataforma Android. Adicionalmente se precisarán los aspectos metodológicos que delinearán el proceso de desarrollo de la librería.

El segundo capítulo detalla el proceso de desarrollo de la librería aplicando los conceptos establecidos por la metodología de desarrollo de software OpenUP, además del seguimiento del proyecto mediante la planificación y gestión de los riesgos implicados en todo desarrollo.

Dentro del tercer capítulo se describe la construcción de un caso de estudio con el propósito de validar el diseño de la librería, dar seguimiento a los objetivos planteados y presentar a los desarrolladores un ejemplo de su funcionamiento.

Finalmente en el capítulo cuatro, se presentan las conclusiones y recomendaciones basadas en la experiencia adquirida dentro del ciclo de vida del proyecto.

CAPÍTULO 1. PLANTEAMIENTO DEL PROBLEMA

Los servicios que ofrecen las redes sociales se han tornado atractivas para los proyectos de aplicaciones móviles permitiendo a los usuarios mejorar su experiencia al sincronizar datos personales, amistades y contenidos. Desde el punto de vista de los desarrolladores el incluir estos servicios implica el consumo de diferentes API's, cada una con su propia documentación, protocolo, tipo de respuesta entre otros. Siendo la integración dentro de un mismo contexto una carga de trabajo que dilata el costo final de un proyecto móvil. La utilización de librerías que faciliten la abstracción de tales conceptos es de vital importancia para la optimización de recursos en el desarrollo siendo esta etapa la justificación para presentar al desarrollador una librería que brinde estas características.

En el presente trabajo se diseñará una API utilizando los procesos descritos en la metodología OpenUP y el lenguaje unificado de desarrollo UML la cuál será implementada dentro de la plataforma Android y consumirá e integrará servicios de las redes sociales Facebook, Google Plus y Twitter. Además se implementará en un caso de estudio denominado "Vida Verde" mediante el cual se evidenciará el diseño adecuado de esta API.

La principal justificación del proyecto se presenta a través de las características del diseño de aplicaciones móviles corporativas y sociales. Mientras en el diseño corporativo de aplicaciones (Figura 1.1) se observa una clara necesidad de infraestructura TI, personal con conocimiento de diferentes tecnologías y definición de procesos para su mantenimiento. En el diseño social de aplicaciones (Figura 1.2) se adaptan las ventajas tanto de procesamiento de un SmartClient, y el acceso a datos distribuidos de un ThinClient.

Sin embargo, surgen nuevos desafíos que comprenden la familiarización con diferentes API's, el procesamiento de las respuestas, sus formatos además del conocimiento de modelos de autorización y autenticación con proveedores de credenciales (OAuth, OpenID, entre otras). Desde esta perspectiva al ser tareas monótonas, pueden ser abstraídas en la librería objeto de nuestra solución la cual hace eficiente el desarrollo de aplicaciones, concentrándolas en el modelamiento del negocio, más que en la ejecución de estas labores.

Se presentan ventajas propias de este diseño como la deducción de costos de la infraestructura necesaria para soportar ambientes middleware, bases de datos entre otras que derivan en mantenimiento y soporte de esta inversión.

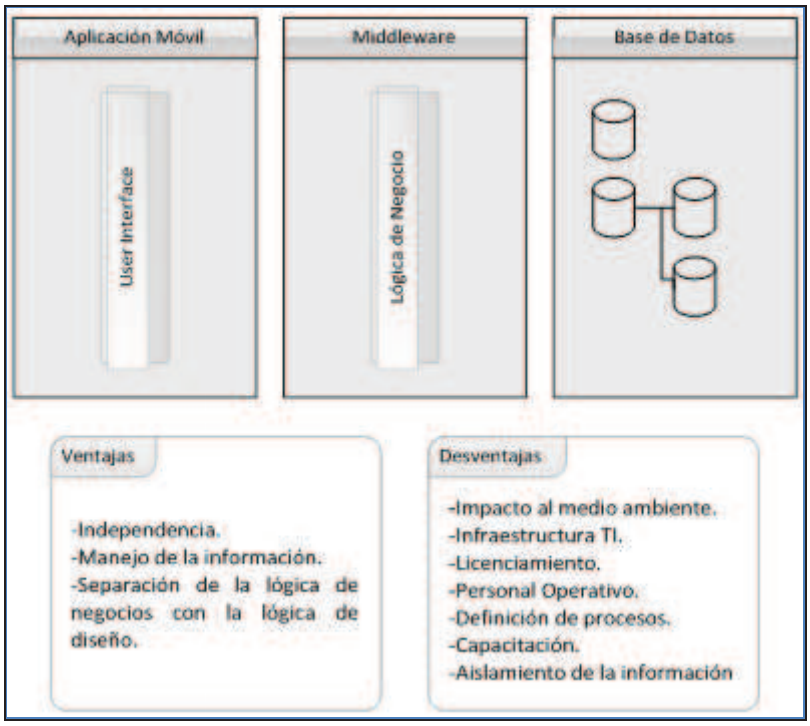


Figura 1.1 Diseño Corporativo de Aplicaciones Móviles.
Elaborado por: Tex Albuja

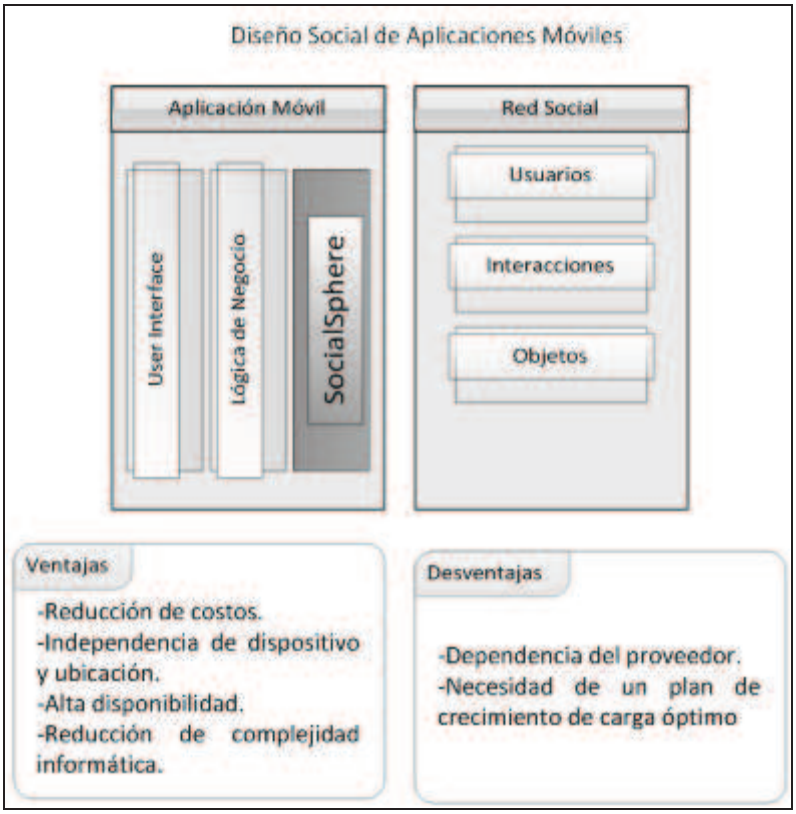


Figura 1.2 Diseño Social de Aplicaciones Móviles
Elaborado por: Tex Albuja

Como producto de esta tesis se desarrollará una librería bajo la plataforma Android la cual se denominará SocialSphere que pretende cumplir los siguientes objetivos.

- Minimizar el costo y tiempo de desarrollo de aplicaciones móviles.
- Reducir costos de infraestructura TI relacionados con la implantación y mantenimiento de servidores, y por ende reducir la contaminación del medio ambiente.
- Requerir menos capacitación para el desarrollo de aplicaciones móviles.
- Brindar más servicios y características a las aplicaciones móviles.

Una vez definidos la problemática y los objetivos de la solución en el siguiente capítulo se efectuará la descripción del estado del arte donde se hace un repaso de las técnicas, tecnologías y metodologías que sustentarán el proyecto.

1.1. PLATAFORMA ANDROID.

La evolución de los dispositivos dedicados como los teléfonos móviles y tablets han permitido que puedan contar con características de procesamiento, almacenamiento y comunicación autosuficientes las cuales han convocado a diferentes compañías al desarrollo de un sistema operativo que soporte la programación de dispositivos móviles con lenguajes de propósito general.

Pese a ser construido para dispositivos móviles, la plataforma Android presenta características similares a un framework para aplicaciones de escritorio. Esta plataforma está disponible para los programadores mediante su Software Development Kit (SDK) llamado Android SDK.

El panorama de distribución de los dispositivos Android permite decidir a los desarrolladores las funciones que su aplicación soportará, así como orientar su aplicación a un nivel de API determinado. La tabla 1.1 recogerá los datos generados hasta septiembre de 2012 (Android Developers, 2012), con respecto a los dispositivos que han utilizado los servicios de Google Play.

Plataforma	Nivel de API	%
4.x.x Jelly Bean	16	2,7%
4.0.x Ice Cream Sandwich	14-15	25,8%
3.x.x Honeycomb	12-13	1,8%
2.3.x Gingerbread	9-10	54,2%
2.2 Froyo	8	12%
2.1 Eclair	7	3,1%
1.6 Donut	4	0,3%
1.5 Cupcake	3	0,1%

Tabla 1.1 Tabla de Distribución de Plataformas en Dispositivos Android.
(Android Developers, 2012)

En la figura 1.3 se dispone la distribución de dispositivos con sistema Android en sus diferentes versiones, cada versión viene acompañada de un nivel de API (API Level). Un dispositivo que soporte un nivel de API mayor puede ejecutar aplicaciones que han sido desarrolladas sobre un nivel de API menor. En sentido contrario, una aplicación que requiera las características de un nuevo nivel de API no se ejecutará en un dispositivo con un nivel de API mayor.

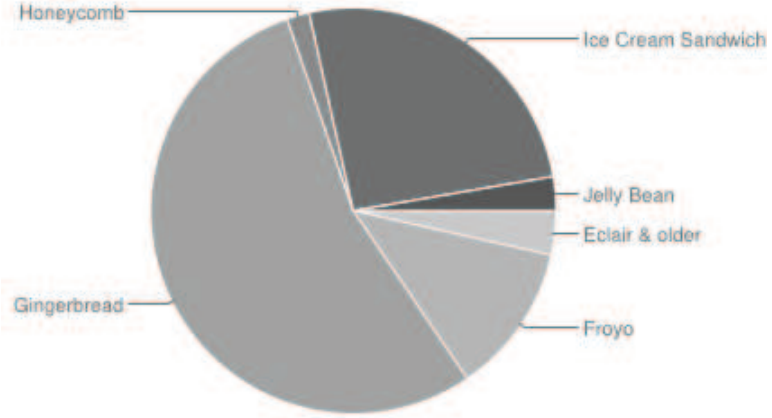


Figura 1.3 Distribución de Dispositivos Android por Sistema Operativo.
(Android Developers, 2012)

De esta información, el porcentaje de dispositivos a los que se procura soportar define el nivel mínimo de API en la que la aplicación se desarrollará. El nivel mínimo de API viene determinado por la menor versión posible de la plataforma en la que la aplicación se pueda compilar, el hardware que la aplicación manejará y las mejoras en la sintaxis para desarrollar.

1.1.1. PILA DE SOFTWARE ANDROID

Una visión general de la plataforma Android se obtiene mediante la comprensión de las capas de software que se pueden observar en la figura 1.4.

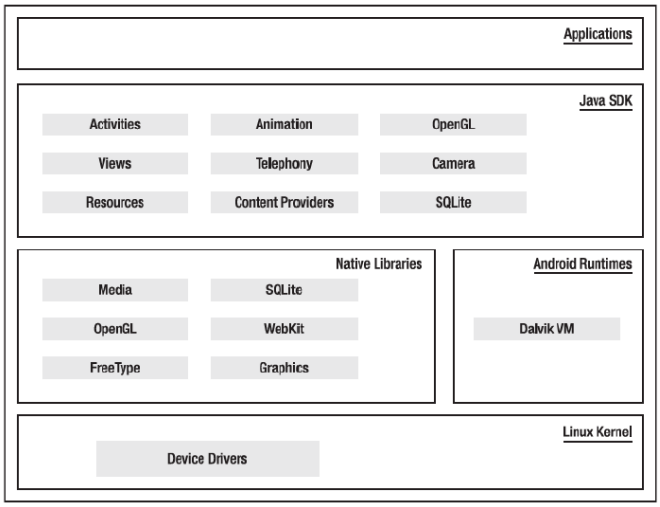


Figura 1.4 Pila detallada de Software Android SDK.
Referencia: Pro Android 3, 2011, pág. 7

El núcleo de la plataforma Android es un kernel Linux el cual tiene la responsabilidad de gestionar los controladores de los dispositivos, acceso a recursos, administración de la energía y otras tareas del Sistema Operativo. Dentro de los controladores se pueden incluir: Pantalla, Cámara, Teclado, WiFi, Memoria Flash, Audio e IPC. Las aplicaciones en un dispositivo Android son desarrolladas en el lenguaje Java y corren bajo la máquina virtual Dalvik.

Las librerías nativas que se encuentran en la parte superior son un conjunto de librerías C/C++ como OpenGL, WebKit, FreeType, SSL, SQLite, y Media. Que están afinadas para ser embebidas dentro de dispositivos basados en Linux.

La mayoría de frameworks de aplicaciones accesan a estas librerías a través de la máquina virtual Dalvik, la cual es el enlace a la plataforma Android. Dalvik está optimizado para correr múltiples instancias de las máquinas virtuales con el objetivo de dedicar una instancia de una máquina virtual a cada aplicación.

Las librerías principales de Android Java API's incluyen telefonía, recursos, lugares, Interfaces gráficas, proveedores de contenidos, administradores de paquetes entre otros que permiten desarrollar aplicaciones de usuario sobre esta API.

1.1.2. PRINCIPALES PAQUETES JAVA DE ANDROID SDK.

La tabla 1.1 identificará los paquetes importantes dentro de la plataforma Android SDK con el fin de abstraer los ejemplos comunes de librerías.

Paquete	Principales Clases	Características
android.bluetooth	BluetoothAdapter, BluetoothDevice, BluetoothSocket, BluetoothServerSocket, BluetoothClass	Controla el adaptador Bluetooth instalado habilitándolo, deshabilitándolo, proceso de descubrimiento, Las clases sockets se utilizan para establecer comunicación entre los dispositivos
android.content	Content	Implementa los conceptos de proveedores de contenidos para acceder a datos de almacenes de datos o URIs.
android.database	Cursor	Implementa la idea de una base de datos abstracta.
android.gesture	Gesture, GestureLibrary, GestureOverlayView, GestureStore, GestureStroke	Identifica los gestos de los gestos definidos por los usuarios y los vincula a una acción
android.sax	Element, RootElement, ElementListener	Contiene un eficiente parseador de XML

Tabla 1.2 Estructura de Librerías de Android SDK
Referencia : (Satya Komatineni, 2011)

El planteamiento de la librería de clases SocialSphere debe tener similar estructura en cuanto a documentación, estándares de programación, distribución y uso permitiendo abstraer los problemas de conectividad, manejo de sesión, procesamiento de respuestas desde una red social posicionándose fácilmente en un proyecto de desarrollo para aplicaciones móviles.

1.2. METODOLOGÍA OPENUP

OpenUp es un proceso ágil de desarrollo de software iterativo e incremental que promueve el uso del contenido fundamental dentro de un proyecto. Es extensible en el sentido de que puede utilizar otros artefactos que proporcionen orientación sobre temas que los proyectos pueden tratar, tales como situaciones contractuales, la seguridad o aplicaciones de misión crítica entre otros que se pueden agregar según sea necesario. Es completo ya que contiene un proceso para construir un sistema que se basa en casos de uso y escenarios, la gestión de riesgos y un enfoque centrado en la arquitectura para impulsar el desarrollo.

1.2.1. PRINCIPIOS.

Cuatro principios gobiernan al proceso con el fin de soportar los principios del manifiesto ágil.

- **Colaborar para alinear los intereses y compartir la comprensión**
  Promueve prácticas que fomenten un ambiente saludable de equipo que permita la colaboración y el desarrollo de entendimiento integral del proyecto priorizando individuos e interacciones sobre proceso y herramientas.
- **Balancear las prioridades para maximizar el valor a los stakeholders**
  Promueve prácticas que permitan desarrollar una solución que maximice los beneficios de los interesados y que sea compatible con las restricciones impuestas en el proyecto causando colaboración del cliente sobre negociaciones del contrato.
- **Enfoque en la arquitectura para minimizar los riesgos proactivamente y organizar el desarrollo**
  Promueve prácticas que permitan al equipo centrarse en la arquitectura para minimizar los riesgos y organizar el desarrollo brindando software de trabajo sobre documentación extensiva.
- **Evolución para obtener retroalimentación y mejoramiento continuo**
  Promueve prácticas para obtener retroalimentación temprana y continua demostrando el valor incremental a los interesados respondiendo al cambio sobre el seguimiento de un plan.

1.2.2. ÁREAS DE CONTENIDO.

En la figura 1.5 se muestra una perspectiva general de la metodología, se puede distinguir la organización del trabajo al nivel personal, del equipo y de los stakeholders.

En el nivel personal los miembros del equipo contribuyen con su trabajo en micro-incrementos, los cuales representan el fruto de pocas horas, o días de trabajo que son seguidos efectivamente cada día. Los demás miembros comparten abiertamente su progreso diario en micro incrementos, los cuales incrementan la visibilidad del trabajo, la confiabilidad y el trabajo en equipo.

Además se puede diferenciar las iteraciones (planificadas en intervalos de tiempo típicamente medidos en semanas). Las cuales apoyan al equipo apropiadamente enfocando su esfuerzo a lo largo del ciclo de vida de la iteración.

Se presenta también el ciclo de vida, el cual se divide en cuatro fases principales: Inicio, Elaboración, Construcción y Transición. Esta información provee a los stakeholders los mecanismos de dirección para controlar la financiación de proyectos, el alcance, la exposición al riesgo, el valor previsto, y otros aspectos del proceso.

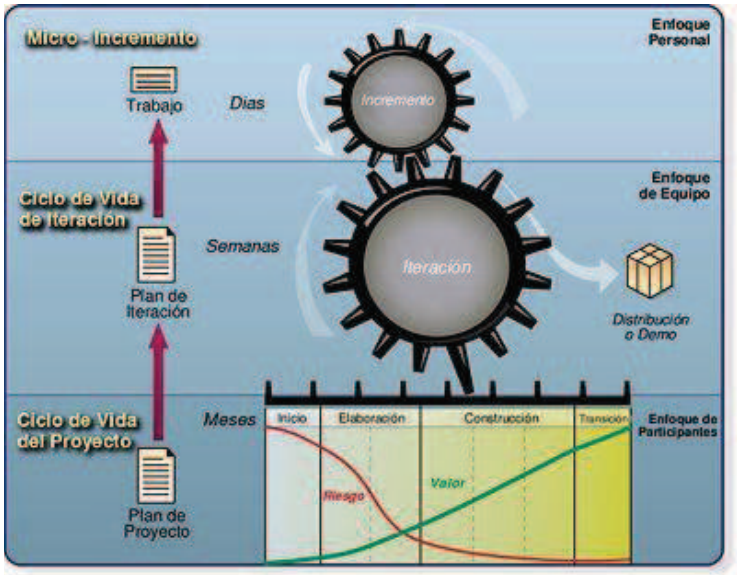


Figura 1.5 Organización del trabajo y el enfoque en el contenido en OpenUP.

Referencia: OpenUp Documentation.
(Eclipse Foundation, 2012)

1.2.3. ROLES

Las habilidades esenciales para equipos pequeños y situados en una misma ubicación son representados por los siguientes roles.

- **Stakeholder:** representa a los grupos de interés cuyas necesidades deben ser satisfechas por el proyecto. Es un papel que puede ser tomado por cualquier persona que es (o será potencialmente) afectado significativamente por el resultado del proyecto.
- **Analista:** representa las inquietudes del cliente y del usuario final mediante la recopilación de sus aportaciones para entender el problema, resolver, capturar y fijar las prioridades de los requisitos.
- **Arquitecto:** es responsable de diseñar la arquitectura de software, que incluye las decisiones técnicas clave que limitan el diseño y la implementación del proyecto.
- **Desarrollador:** es responsable del desarrollo de una parte del sistema, incluyendo el diseño que se ajuste a la arquitectura definida, la implementación, las pruebas unitarias e integrar los componentes que forman parte de la solución.
- **Tester:** es responsable de las tareas de pruebas tales como identificar, definir, implantar y llevar a cabo las pruebas necesarias, así como el registro de los resultados de pruebas y análisis de los resultados.
- **Gerente de Proyecto:** Planifica el proyecto en colaboración con los actores y el equipo, coordina las interacciones con las partes interesadas, y mantiene al equipo centrándolo en el cumplimiento del proyecto.

1.2.4. ARTEFACTOS.

La metodología propone los siguientes principios para la utilización de artefactos.

- Aplicar el mínimo proceso necesario que aporte valor
- Evitar el trabajo improductivo con la elaboración de artefactos.
- Utilizar un proceso que puede ser adaptado y ampliado para satisfacer las necesidades adicionales que puedan surgir durante el ciclo de vida de desarrollo de software.

Se utiliza entonces criterios que verifiquen la necesidad de la elaboración de los artefactos. Dentro de las tablas: 1.3, 1.4, 1.5, 1.6, se encuentran las principales razones por las cuales deben o no ser considerada la elaboración de cada artefacto a lo largo del proyecto Socialsphere.

Visión	
Item	**Descripción**
Impacto al omitirlo	Los interesados y el equipo de desarrollo pueden tener diferentes expectativas del proyecto. Puede desembocar en la cancelación del proyecto. Puede crear discrepancias en los interesados del Proyecto.
Razones para no necesitarlo	Es usado por todos los proyectos, excepto los triviales.
Opciones de representación	Documento, Herramienta de gestión de requisitos, Puede ser dividido.
Consideraciones Claves	Mantenerlo breve y entregarlo rápido a los stakeholders, que sea leíble y entendible. Incluir sólo las características más importantes y evitar los detalles de los requisitos.
Uso en SocialSphere	Aprobado

Tabla 1.3 Artefacto Vision
Elaborado por: Tex Albuja

Plan de Proyecto	
Item	**Descripción**
Impacto al omitirlo	Falta de límites claros del alcance del proyecto. Puede desembocar en sobreasignación y mal uso de recursos, planificación pobre del proyecto y dificultades en el control del proyecto.
Razones para no necesitarlo	Proyectos con un enfoque evolutivo, de mantenimiento, o soporte para el desarrollo.
Opciones de representación	Diagramas de Gantt, Calendarización de tareas.
Consideraciones Claves	Crear y actualizar el plan del proyecto en sesiones de seguimiento y planificación que involucran a todo el equipo y las partes interesadas pertinentes del proyecto con el fin de asegurarse de que todos están de acuerdo con él.
Uso en SocialSphere	Aprobado

Tabla 1.4 Artefacto Plan de Proyecto
Elaborado por: Tex Albuja

Lista de Riesgos	
Item	**Descripción**
Impacto al omitirlo	Si los riesgos no están identificados, entonces las estrategias de mitigación no se habrán establecido, y cualquier evento adverso que se produzca en un proyecto tendrá que ser tratado de una manera ad-hoc. Los riesgos indocumentados que se producen también son más propensos al impacto de la programación general y la calidad de la solución.
Razones para no necesitarlo	A menos que no existan riesgos, o existan pocos, no hay ninguna razón para no producir este artefacto.
Opciones de representación	Diagramas de Gantt, Calendarización de tareas.
Consideraciones Claves	Crear y actualizar el plan del proyecto en sesiones de seguimiento y planificación que involucran a todo el equipo y las partes interesadas pertinentes del proyecto con el fin de asegurarse de que todos están de acuerdo con él.
Uso en SocialSphere	Aprobado

Tabla 1.5 Artefacto Lista de Riesgos
Elaborado por: Tex Albuja

Modelo de Casos de Uso	
Item	**Descripción**
Impacto al omitirlo.	Si los casos de uso no están modelados, es difícil determinar todas las relaciones entre actores y casos de uso, además de las relaciones entre casos de uso.
Razones para no necesitarlo.	Si el conjunto total de casos de uso es pequeño y no tiene relaciones complejas. O si se cuenta con un enfoque alternativo para registrarlos.
Opciones de representación.	Reportes de los diagramas de UML, representaciones gráficas.
Consideraciones Claves.	Crear y actualizar el plan del proyecto en sesiones de seguimiento y planificación que involucran a todo el equipo y las partes interesadas pertinentes del proyecto con el fin de asegurarse de que todos están de acuerdo con él.
Uso en SocialSphere	Aprobado

Tabla 1.6 Artefacto Modelo de Casos de Uso
Elaborado por: Tex Albuja

En la figura 1.6 se presenta un mapa conceptual que sintetiza la metodología.

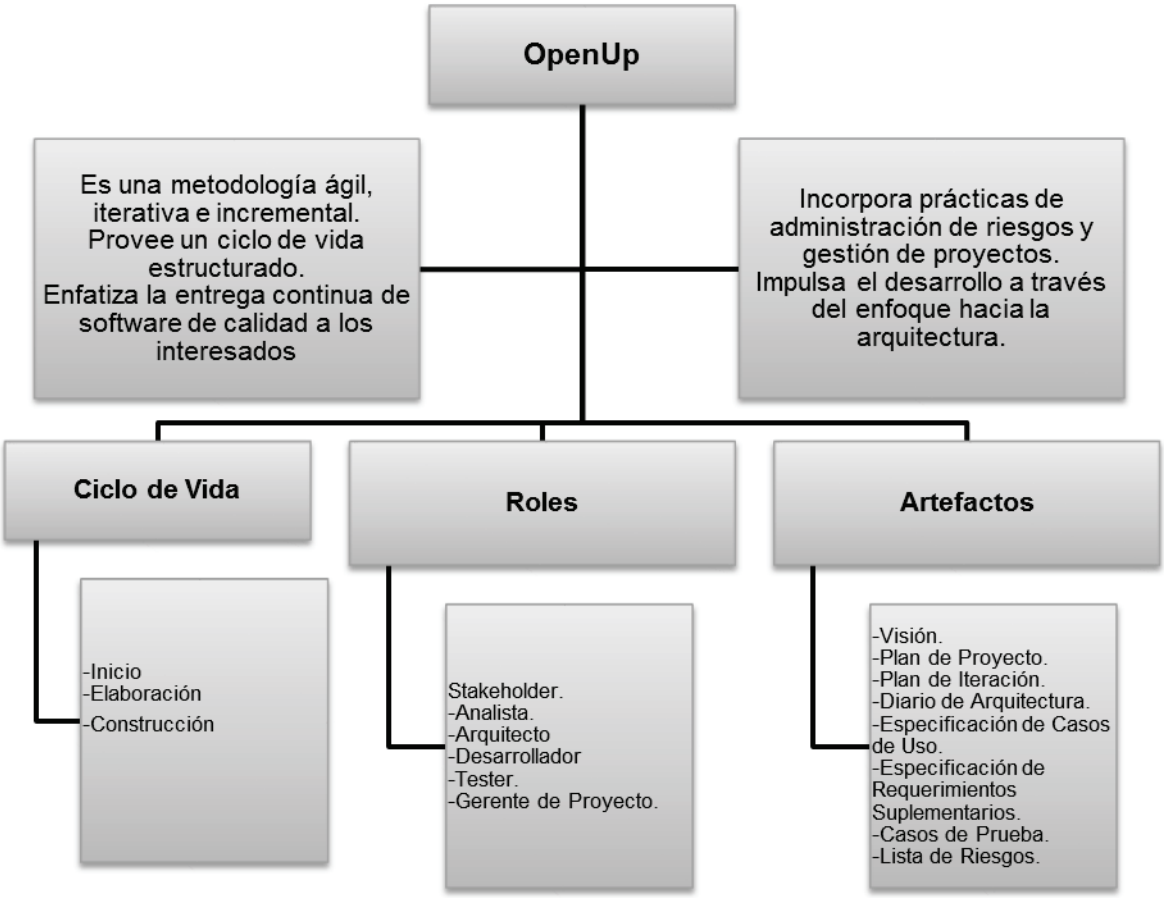


Figura 1.6 Mapa Conceptual de la Metodología OpenUP.
Elaborado por: Tex Albuja.

1.3. CONSIDERACIONES DEL CONSUMO DE API'S DE LAS REDES SOCIALES.

La arquitectura de software orientada a servicios (SOA) permite la creación de sistemas de información altamente escalables los cuales brindan una forma definida de exposición e invocación de servicios (comúnmente pero no exclusivamente servicios web), lo cual facilita la interacción entre diferentes sistemas propios o de terceros.

Esta arquitectura es ampliamente utilizada por las redes sociales y aplicaciones distribuidas en común, las cuales han definido su propia API para comunicarse tanto con otros sistemas, así como también para poder presentarse desde diferentes frentes como aplicaciones de escritorio, aplicaciones web, o móviles.

Las consideraciones que se deben resolver, estarán delineadas por la definición de la estructura de los mensajes, sus protocolos, los términos de uso. Se estudiará las características que presentan las API's de las redes sociales que han sido planteadas.

1.3.1. FACEBOOK.

La API de Facebook es una plataforma para crear aplicaciones que están disponibles para los miembros de la red social de Facebook. La API permite a las aplicaciones utilizar las conexiones sociales y la información del perfil para que las aplicaciones sean más envolventes, y para publicar las actividades de la fuente de noticias y páginas de perfil de Facebook, sujeta a los ajustes de privacidad de los usuarios. Con la API, los usuarios pueden añadir un contexto social a sus aplicaciones mediante la utilización de perfil, páginas de amigos, los datos del grupo, fotos y eventos. La API utiliza el protocolo REST y las respuestas se envían en formato XML o JSON.

Facebook: Specifications

Functionality	
Client Install Required	No
Service Endpoint	http://api.facebook.com/restserver.php
Signup and Licensing	
Signup Requirements	Facebook account and developer ID
Developer Key Required	Yes
Account Required	Yes
Commercial Licensing	Yes, within limits
Provider	facebook.com
Company	Facebook
Non-Commercial Lic.	
Data Licensing	
Usage Fees	
Program Fees	
Certification Program	
Usage Limits	100,000 requests 24 hour period
Terms of Service	
Security	
Authentication Model	Username/password, developer key, OAuth
SSL Support	
Read-only Without Login	
Support	
Vendor API Kits	PHP, Java
Community API Kits	
API Blog	Facebook Developer Blog
Site Blog	
API Forum	forum.developers.facebook.com/
Developer Support	forums
Console URL	developers.facebook.com/tools/explorer

Figura 1.7 Especificaciones de la API de Facebook.
(ProgrammableWeb, 2012)

1.3.2. TWITTER.

La API de Twitter es una plataforma para crear aplicaciones que están disponibles para los miembros del servicio de microblogging. La API permite a las aplicaciones utilizar las actualizaciones, los estados y la información del perfil del usuario. La API actualmente soporta el protocolo REST y las respuestas se envían en formato XML, JSON, RSS, Atom.

Twitter: Specifications

Functionality	
Client Install Required	No
Service Endpoint	http://twitter.com/statuses/
Signup and Licensing	
Signup Requirements	Account with service
Developer Key Required	No
Account Required	Yes
Commercial Licensing	
Provider	twitter.com
Non-Commercial Lic.	
Data Licensing	
Usage Fees	
Program Fees	
Certification Program	
Usage Limits	100 requests per 60 sixty minute time period
Terms of Service	
Security	
Authentication Model	HTTP Basic Auth, OAuth
SSL Support	No
Read-only Without Login	No
Support	
Vendor API Kits	ActionScript
Community API Kits	
API Blog	https://dev.twitter.com/blog
Site Blog	blog.twitter.com/
API Forum	groups.google.com/group/twitter-development-talk
Developer Support	
Console URL	https://dev.twitter.com/console

Figura 1.8 Especificaciones de la API de Twitter
(ProgrammableWeb, 2012)

1.3.3. GOOGLE PLUS.

La API de Google Plus permite a los desarrolladores acceder al contenido disponible en Google Plus, incluyendo la información del usuario y los elementos compartidos públicamente. La API actualmente soporta el protocolo REST y las respuestas se envían en formato JSON.

Google Plus: Specifications

Functionality	
Client Install Required	
Service Endpoint	https://www.googleapis.com/plus/v1/
Signup and Licensing	
Signup Requirements	API Key
Developer Key Required	Yes
Account Required	
Commercial Licensing	
Provider	google.com
Company	Google
Non-Commercial Lic.	
Data Licensing	
Usage Fees	
Program Fees	
Certification Program	
Usage Limits	1,000 queries/day
Terms of Service	https://developers.google.com/+/terms
Security	
Authentication Model	API Key, OAuth 2.0
SSL Support	Yes
Read-only Without Login	
Support	
Vendor API Kits	Java, GWT, Python, Ruby, PHP, .NET
Community API Kits	
API Blog	
Site Blog	googleplusplatform.blogspot.com/
API Forum	
Developer Support	developers.google.com/+/
Console URL	

Figura 1.9 Especificaciones de la API de Google Plus.
(ProgrammableWeb, 2012)

1.3.4. ANÁLISIS COMPARATIVO ENTRE API'S.

La integración entre estas API's requiere que se utilicen en lo posible tecnologías similares, así como tipos de respuesta y métodos de autenticación. Dentro de la tabla 1.6 se resumirán las especificaciones de cada una.

Red Social	Facebook	Twitter	GooglePlus
Funcionalidad			
Protocolo	REST	REST	REST
Formato de Datos	XML	XML, JSON	JSON
Registro y Licenciamiento			
Requerimientos de registro	Cuenta e ID de Desarrollador	Cuenta con servicio	API Key
Developer Key	Si	No	Si
Seguridad			
Modelo de Autenticación	OpenAuth, Developer Key, UserName/Password	HTTP Basic Auth, OpenAuth	API Key, OpenAuth 2.0
Soporte para SSL	No	No	Si
Soporte			
API Kit del Proveedor	PHP, Java	ActionScript	Java, GWT, Python, Ruby, PHP, .NET
Soporte para desarrollo	Foro	Foro	Site

Tabla 1.7 Características de API's de Redes Sociales
Elaborado por: Tex Albuja

1.4. CONSIDERACIONES DE INTEGRACIÓN.

El objetivo de esta sección es presentar tecnologías de uso común en las API's de las redes sociales que serán utilizados por la API de Integración y posteriormente implementadas por la librería SocialSphere.

1.4.1. OPENAUTH.

OpenAuth es utilizado por la gran mayoría de redes sociales para fines de autenticación y autorización del contenido al que un usuario requiera acceder (Hueniverse, 2011). Provee una API que ofrece una manera simple para publicar e interactuar con los datos protegidos y en el caso de almacenar credenciales la posibilidad de que los usuarios no las distribuyan por la web. Provee un esquema en donde se agrega un tercer actor al modelo cliente-servidor el dueño de los recursos. Para que un cliente obtenga acceso al recurso solicitado lo tiene que obtener desde el dueño de los recursos el cual es expresado mediante el token (Protege las credenciales) y el secreto compartido (Protege los recursos).

1.4.1.1. Flujo de Autenticación y Autorización de Contenido.

OAuth define tres roles dentro de cualquier transacción: Cliente, Servidor y Proveedor de recursos, este último rol es ejecutado por el cliente en algunos casos. A continuación se describe el flujo de autenticación en un ejemplo (SalesForce Developers, 2012).

1. El consumidor (Aplicación Cliente) pide un token de solicitud.
2. Salesforce verifica la petición y devuelve un token de solicitud.
3. El consumidor debe redirigir el usuario a Salesforce, donde se insta a iniciar sesión.
4. Salesforce autoriza al usuario y lo redirige a la aplicación.
5. El consumidor pide un token de acceso.
6. Salesforce verifica la petición y concede el token.
7. Una vez que se ha concedido el token, el consumidor accede a los datos ya sea a través de su aplicación o a través de la API de servicios Web de Force.com.
8. Salesforce verifica la petición y permite el acceso a los datos.

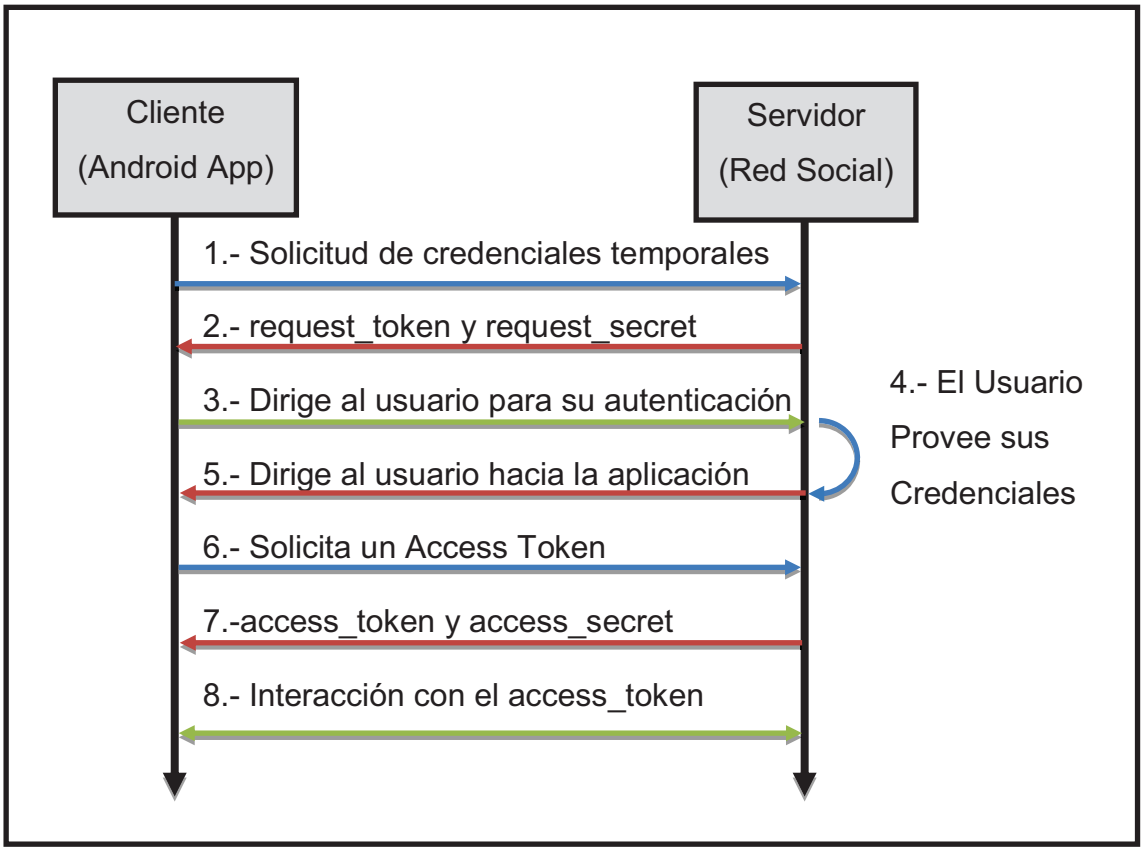


Figura 1.10 Flujo de Autenticación y Autorización de Contenido.
Elaborado por: Tex Albuja

El flujo descrito en la figura 1.10 muestra el proceso de autenticación a través de OAuth es conocido como 3-legged porque en él intervienen los tres roles descritos. Otro flujo que se suele utilizar es el llamado 2-legged en el que, debido a que no se acceden a los recursos del usuario, únicamente hay que autenticar a la aplicación.

1.5. CONSIDERACIONES DE ESTÁNDARES DE PROGRAMACIÓN DE ANDROID.

La importancia de seguir estándares de programación dentro de una librería nos permite ofrecer al desarrollador las siguientes ventajas.

- **Migración del código**: el código es de fácil adaptación a herramientas de migración para efectuar operaciones como la actualización a nuevas versiones del SDK o la adaptación a nuevas plataformas.
- **Escalabilidad de la librería:** si un nuevo miembro, o equipo desea potenciar las capacidades de la librería a través de sus aportes se puede adaptar fácilmente su conocimiento mediante la publicación de los estándares utilizados y que son comunes para los desarrolladores.
- **Independencia del proveedor de la librería:** brinda a los desarrolladores la capacidad de utilizar otros proveedores, o cambiarlas sin estar vinculado totalmente y comprometer la integridad de sus aplicaciones a la dinámica del mercado.
- **Curva de aprendizaje alta obteniendo un mejor costo-beneficio:** la abstracción de los elementos dentro de una librería que utiliza estándares es común a los desarrolladores ya que están utilizando librerías que se podrían comparar con las librerías del entorno de desarrollo.
- **Extensión del soporte:** se obtiene al evitar el empoderamiento del conocimiento por un cierto conjunto de personas o equipos de personas a sus estándares internos.

1.5.1. ESTÁNDARES DE PROGRAMACIÓN ANDROID.

1.5.1.1. Estándares de Programación en Clases.

- **No ignorar excepciones:** la aplicación no caerá en un determinado momento pero el usuario esperará una respuesta.

```
void setServerPort(String value) {
try {
serverPort = Integer.parseInt(value);
} catch (NumberFormatException e) { }
}
```

- **No envolver en una excepción genérica:** de igual manera la aplicación no caerá en un determinado momento sin embargo se pierde el tipo de excepción y el usuario no sabrá en qué momento la aplicación cayó.

```
try {
    someComplicatedIOFunction();  // may throw IOException
    someComplicatedParsingFunction();// may throw
ParsingException
    someComplicatedSecurityFunction(); // may throw
SecurityException
    // phew, made it all the way
} catch (Exception e) {                 // I'll just catch
all exceptions
    handleError();                      // with one generic
handler!
}
```

- **No utilizar finalizadores:** los finalizadores no garantizan que el código sea totalmente recorrido.

- **Importar librerías totalmente especificadas:** facilita la lectura del código para los nuevos desarrolladores además de minimizar el uso de recursos por la importación de paquetes no necesarios.

  Además de estas especificaciones definidas por Android es importante seguir los estándares definidos por el lenguaje Java como el uso de comentarios javadoc, comentarios TODO.

1.5.1.2. Estándares de Programación en Proyectos del tipo Librería.

A partir del plugin de Eclipse ADT 0.9.7, Android ya maneja la idea de proyectos del tipo librería. Un proyecto de este tipo es una colección de código java, y recursos que se ven como un proyecto regular con la diferencia que no finalizan en un archivo .apk por sí mismos. El código java, y los recursos son compilados en proyectos principales los cuáles hacen uso de las facilidades que ofrece esta librería.

Existen algunas consideraciones que se debe tener en cuenta en proyectos del tipo Librería.

- Una librería puede tener su propio nombre de paquete.
- Una librería no compilará en su propio archivo .apk. En cambio será absorbida en el archivo .apk del proyecto que usa a la librería como referencia.
- Una librería puede usar otros archivos JAR.
- Eclipse ADT combinará la(s) librería(s) referida(s) y las compilará juntos como parte de la compilación del proyecto principal.
- El acceso a los recursos son independientes entre la librería y el proyecto principal los cuales acceden a través de sus respectivos archivos R.java.
- Se puede usar prefijos como lib_ para los recursos de la librería.
- Un proyecto principal puede referenciar cualquier número de librerías.
- Se puede establecer prioridades de los proyectos para ver qué recursos son más importantes.
- Los componentes, como las actividades de una librería necesitan ser definidas en el archivo manifest. Cuando se ha hecho eso, el nombre del componente de la librería debe ser totalmente calificado con el nombre de la librería.

- No es necesario definir los componentes en el archivo manifest de la librería, pero puede ser una buena práctica conocer que componentes soporta.
- La creación de una librería comienza con la creación de un proyecto Android regular y se configura a través de la bandera *Is Library* en propiedades.

1.6. ANÁLISIS DE LAS SOLUCIONES EXISTENTES.

A continuación se presentan las soluciones presentes en el mercado actualmente, sus características, la plataforma para la que están orientados además de la documentación que presentan entre otras, con el fin de justificar el desarrollo de la solución SocialSphere.

1.6.1. SHAREKIT

ShareKit es una librería de código abierto que es distribuido bajo la licencia MIT Open Source License disponible para los desarrolladores de las aplicaciones sobre iOS la cual simplifica las tareas de seguimiento y mantención de las integraciones de cada servicio y de las funciones comunes entre ellas. Permite compartir contenido de tipo: URL, imágenes, texto y archivos dentro de los sitios: Delicious, Email, Facebook, Google Reader, Instapaper, Pinboard, Read It Later, Tumblr y Twitter. Ofrece además a los usuarios la posibilidad de compartir este contenido sin conexión a internet, enviando las peticiones pendientes cuando el usuario esté conectado.

1.6.2. SOCIALLIB

SocialLib es un proyecto open-source que es distribuido bajo la licencia GPLv3 el cual apoya al desarrollo de aplicaciones sociales elaboradas sobre la plataforma Android facilitando el consumo de los servicios de las redes sociales: Facebook, Twitter, Google Buzz, LinkedIn.

1.6.3. PINKELSTAR

PinkelStar es un servicio para los desarrolladores de aplicaciones móviles. Con la biblioteca nativa PinkelStar un desarrollador puede integrar redes sociales como Facebook y Twitter directamente en su aplicación. Con PinkelStar los usuarios van a llevar su aplicación a sus amigos, lo que acelera el reconocimiento de marca de su aplicación y distribución.

1.6.4. ANÁLISIS COMPARATIVO DE LAS SOLUCIONES EXISTENTES.

La tabla 1.7 muestra una síntesis de características referenciales de las soluciones existentes que han sido mencionadas.

Solución	ShareKit	SocialLib	PinkelStar
Plataforma			
Sistema Operativo	iOS 4.0	Android	iOSUI/AndroidUI
IDE	Xcode 3.2	Eclipse ADT	Eclipse ADT/XCode 4
Documentación			
Tutoriales	SI	Si	Si
Videos	Si	No	No
Registro y Licenciamiento			
Cuenta de desarrollo	No	No	SI
Licencia	MIT Open Source License	GNU LGPL v3.0	
Soporte a Redes Sociales			
Redes Sociales Soportadas	Facebook, Twitter, Delicious, Tumblr, Read It Later, Bit.ly	Facebook, Twitter, GoogleBuzz, LinkedIn	Facebook, Twitter
Servicios Ofrecidos	Sharing	Sharing	Share/Like Button
Integración y Sincronización	No	No	No
Extensibilidad a otras redes	No	No	No
Último Release Formal.			
Fecha	Enero, 2012	Sept, 2010	Oct, 2011
Actividad	Alta	Deprecada	Media

Tabla 1.8 Análisis Comparativo de las Soluciones Existentes
Elaborado por: Tex Albuja

Se puede inferir que, de las soluciones presentadas, únicamente SocialLib se ha desarrollado para la plataforma objetivo de esta solución, sin embargo no se encuentra en desarrollo activo. Además en las soluciones expuestas no se ofrecen características de integración entre redes sociales. Otro punto importante es la falta de documentación la cual se presenta en documentos como javadoc y FAQs, una solución adecuada debe proveer documentación de usuario, documentación del producto y documentación para la instalación como parte de la entrega de valor a los usuarios. Finalmente se provee también aplicaciones pequeñas que ejemplifican el uso de los productos.

1.7. CARACTERÍSTICAS DE LA SOLUCIÓN PROPUESTA.

Una vez analizados los tipos de respuestas, los métodos de autenticación, las soluciones existentes y los protocolos que cada API utiliza se proponen las siguientes características:

- Proveer la documentación del producto, instalación y de usuario.
- Proveer la documentación que ejemplifique con una aplicación su uso.
- Proveer un mecanismo de manejo de autenticación para las redes sociales a las que se quiere acceder.
- Proveer un mecanismo para registrar las credenciales de la aplicación.
- Proveer un mecanismo para la sincronización de datos del usuario dentro del ciclo de vida de la aplicación.
- Proveer mecanismos para efectuar las interacciones con las redes sociales, los cuales extiendan la funcionalidad de la aplicación y envuelvan al usuario en una mejor experiencia.
- Proveer un modelo que pueda ser implementado en el caso de que se requiera agregar más redes sociales a la librería.

CAPÍTULO 2. DESARROLLO DE LA BIBLIOTECA DE CLASES.

El capítulo presenta el ciclo de vida del desarrollo de la biblioteca de clases SocialSphere. Se realizará una breve descripción de la metodología OpenUp, aplicando sus principios y recomendaciones para efectuar cada una de las fases que la componen.

2.1. FASE DE INICIO.

Durante la Fase de Inicio, la atención se centra en obtener un acuerdo sobre el problema a resolver, la recopilación de necesidades de los interesados, y la captura de características de alto nivel del sistema.
Se desprenden los artefactos, visión, Plan de Proyecto, Plan de iteración, Especificación de Requerimientos Suplementarios, Diario de Arquitectura y se comprenden los costos y tiempos vinculados al proyecto.

2.1.1. INICIAR EL PROYECTO

Esta acción permite poner en marcha el proyecto obteniendo un acuerdo con las partes interesadas sobre el alcance del proyecto y un plan inicial para lograrlo.
Los roles que llevarán a cabo esta actividad serán:

- El Analista y el StakeHolder, los cuales trabajarán juntos para definir la visión del proyecto, capturando las necesidades y características del sistema en desarrollo.
- El Gerente de Proyecto, colaborará forjando acuerdos entre el equipo de desarrollo y los interesados. Adicionalmente, propone el plan de proyecto que incluye los hitos de las cuatro faces del proyecto.

Esta actividad involucra las tareas necesarias para definir la visión, un glosario y crear un plan de proyecto.

2.1.1.1. Desarrollo de la Visión.

La solución es propuesta para un problema en el que todos deben estar de acuerdo. Las partes interesadas colaboran con el equipo de desarrollo para expresar y documentar sus problemas, necesidades y características potenciales para el sistema con el fin de que el equipo del proyecto pueda comprender mejor lo que hay que hacer.

OpenUp define los siguientes pasos que permitirán desarrollar una visión correcta.

- **Identificar a los interesados**
- **Obtener acuerdos sobre el problema a ser resuelto**
- **Reunir los requerimientos de los interesados**
- **Definir las características del sistema**
- **Definir un vocabulario común**

La definición de cada una de las tareas, así como su elaboración dentro del proyecto SocialSphere se encuentran en el anexo 1.

Una vez completadas las tareas que propone la metodología, se dispone de la información suficiente para generar los artefactos planteados en esta actividad.

Documento de Visión.

Este artefacto permite la captura del punto de vista global de la solución, definiendo las necesidades, requisitos y características del sistema.

SocialSphere
Visión

1. Introducción

Los servicios que ofrecen las redes sociales se han tornado atractivas para los proyectos de aplicaciones móviles permitiendo a los usuarios mejorar su experiencia al sincronizar datos personales, datos de amigos, sitios y demás. Desde el punto de vista de los desarrolladores el incluir estos servicios implica el consumo de diferentes API's, cada una con su propia documentación, protocolo, tipo de respuesta entre otros. Siendo la integración dentro de un mismo contexto una carga de trabajo que dilata el costo final de un proyecto móvil. La utilización de librerías que faciliten la abstracción de tales conceptos es de vital importancia para la optimización de recursos en el desarrollo siendo esta etapa la justificación para presentar al desarrollador la librería SocialSphere que brinde estas características.

2. Posicionamiento

El desarrollo de una librería que haga transparente el consumo de diferentes API's y ofrezca a los desarrolladores de aplicaciones móviles la integración de características sociales permitirá el desarrollo eficiente, la mitigación de los errores comunes en el desarrollo minimizando la necesidad de pruebas y brindando al usuario aplicaciones funcionales en menor tiempo.

2.1 Planteamiento del Problema.

El problema de	El consumo de los servicios de redes sociales
Afecta a	Los desarrolladores de aplicaciones móviles
El impacto asociado es	El costo del proyecto se ve afectado en tiempo, recursos y necesidad de conocimientos.
Una solución exitosa sería	El consumo de librerías que abstraigan tales conceptos

2.2 Planteamiento de la posición del producto

Para	Desarrolladores de aplicaciones móviles sobre la plataforma Android.
Quienes	Desarrollan aplicaciones móviles que incluyen la integración con las redes sociales Facebook,Twitter, Google Plus.
SocialSphere	Es una librería.
Que	Minimiza el tiempo de desarrollo.
Difiere de	ShareKit, PinkelStar.
Nuestro producto.	Maneja estándares de programación Android.

3. Descripción de los Interesados.

3.1 Resumen de los interesados.

Nombre	Descripción	Responsabilidades
Desarrollador interesado.	Desarrolla Aplicaciones móviles de propósito general que integran características de redes sociales.	Asegura que la biblioteca sea fácil de mantener. Aprueba el diseño de las clases.
Usuario	Usuario de las Aplicaciones móviles desarrolladas mediante la integración de la librería SocialSphere.	Validan la interacción efectiva con sus redes sociales.
Arquitecto	Establece el marco donde la solución será desarrollada	Toma de decisiones técnicas. Diseño global de la solución.
Desarrollador de la biblioteca SocialSphere	Construye la solución.	Entrega valor al desarrollador interesado mediante la implementación efectiva de la solución

3.2 Entorno del Usuario.

El entorno de desarrollo de aplicaciones móviles tiene las siguientes características.

- Equipo reducido.
- Tareas de duración de mediano plazo.
- Plataformas de desarrollo dentro del IDE Eclipse mediante ADT
- Integración con otras plataformas mediante el consumo de servicios.

4. Características de la Solución.

4.1 Necesidades y características.

Necesidad	Prioridad	Características	Entrega Planificada
Manejo de Autenticación	Alta	Manejo del protocolo OpenAuth. Seguridad de las credenciales de los usuarios finales. Acceso a los datos del usuario.	Versión 1.0
Registro de las credenciales de la aplicación	Alta	Manejo de API Keys, Developers ID,Cuentas de la aplicación.	Versión 1.0
Sincronización de los datos del usurario.	Media	Integración entre redes sociales	Versión 1.0
Obtención de la información de amigos para extender la funcionalidad de la aplicación	Alta	Integración con las características de la aplicación que consume la librería	Versión 1.0
Optimización del rendimiento	Alta	Bajo consumo de red, espacio, memoria, procesamiento	Versión 1.0
Documentación de fácil abstracción	Alta	Documentación concreta, bien estructurada.	Versión 1.0

5. Otros requerimientos de la Solución.

Requerimiento	Prioridad	Entrega Planificada
Implementación de estándares de programación Android	Alta	Versión 1.0
Manuales de usuario	Alta	Versión 1.0
Ayuda en línea	Alta	Versión 2.0
Instalación	Alta	Versión 1.0

Una vez cumplido el objetivo de esta sección, se tiene los elementos necesarios para una planificación efectiva del proyecto.

2.1.1.2. Planificación del Proyecto.

Desarrollar el plan del proyecto provee una oportunidad al equipo al acordar el alcance del proyecto, objetivos, cronograma inicial, y los resultados. Se definen los criterios de éxito y las prácticas de trabajo a utilizar. La colaboración y el consenso de todos los participantes clave del proyecto es la meta.

OpenUp define los siguientes pasos que permitirán desarrollar una visión correcta del proyecto.

- **Identificar un equipo compacto.**
  Revisar la asignación de recursos para el proyecto identificando sus carencias y poniendo en marcha la contratación o reasignación de recursos en caso de que fuese necesario. Conversar con el equipo para definir sus roles y responsabilidades.
  Todos los roles y responsabilidades del proyecto SocialSphere son desarrolladas por Tex Albuja.

- **Evaluar los riesgos.**
  El equipo identifica los riesgos del proyecto, lleva a cabo un análisis de riesgos cualitativos para evaluar el orden de magnitud, y se actualiza el documento *Lista de Riesgos*. El gerente del proyecto facilita la decisión del equipo acerca de los riesgos que se deben asistir y qué riesgos deben tener en cuenta.

Lista de Riesgos.

Este artefacto es una lista ordenada de los riesgos conocidos al proyecto, ordenados por orden de importancia y se asocia con la mitigación o acciones específicas de contingencia, además son clasificado dependiendo de la disciplina OpenUp que los defina (Arquitectura AR, Despliegue DP, Desarrollo DV, Ambiente E, Administración del Proyecto PM, Requerimientos R, Pruebas T) la tabla 2.1 recoge la representación de los riesgos determinados en la solución SocialSphere.

SocialSphere
Lista de Riegos

Id	Título	Descripción	Tipo	Impacto	Probabilidad	Magnitud	Estrategia de Mitigación.
1	Tiempo y costo estimado de los mecanismos mal calculado.	Dilatación del tiempo de desarrollo debido a factores externos, o planificación no adecuada.	PM	6	65%	3.9	Optimizar los tiempos estimados, y establecer colchones de tiempo.
2	Cambios en las definiciones de las API's de Terceros.	Implementación considerada con definiciones anteriores.	AR	6	30%	1.8	Utilizar las definiciones más actualizadas de las API's a utilizar
3	Falta de conocimiento de la plataforma Android.	Dilatación del tiempo de desarrollo por desconocimiento de los procedimientos definidos por Android.	DV	5	65%	3.25	Capacitación previa en la plataforma.
5	Disponibilidad del personal.	No se cuenta con disponibilidad de tiempo dedicado de los recursos.	PM	7	70%	4.9	Planificación organizada de las actividades a realizar.
6	Ausencia de un diseño y especificación de requisitos.	No se cuenta con un diseño y especificación de requisitos	R	6	20%	1.2	Aplicación de una metodología de desarrollo.
7	Ausencia de documentación de las API's de las redes sociales.	No se cuenta con una documentación de las API's de las redes sociales.	E	5	20%	1.0	Selección de las redes sociales que posean mejor documentación.

Tabla 2.1 Lista de Riesgos.
Elaborado por: Tex Albuja

- **Establecer costos y los recursos necesarios.**

La información que ha sido recolectada permite establecer un presupuesto que permita dimensionar los diferentes elementos que intervienen en el desarrollo del proyecto. Se puede establecer un presupuesto referencial del proyecto a partir de los siguientes elementos.

Recursos Humanos:

Dentro de la Tabla 2.2 se estima el personal requerido, el tiempo que invertirán, las competencias que deben cumplir para asumir un rol dentro del proyecto definirán el valor que en este ítem se espera asignar.

Los roles han sido establecidos dentro de la metodología OpenUp.

Rol	Horas	Costo por Hora	Total
Analista	120	$5.00	$600
Arquitecto	80	$7.00	$560
Desarrollador	160	$6.00	$960
Tester	120	$5.00	$600
Gerente de Proyecto	80	$7.00	$560
Total RRHH.			$3280

Tabla 2.2 Estimación de costos (Recursos Humanos).
Elaborado por: Tex Albuja

Hardware:

La adquisición de dispositivos es una estrategia importante para el progreso del equipo de desarrollo. Este costo debe ser proyectado en función tiempo de vida de los dispositivos, dividido en el que se invertirá en el proyecto. En la tabla 2.3 se establece una estimación de costos destinados a este rubro.

Dispositivo	Tiempo en Proyecto (meses)	Tiempo de Vida (meses)	Cantidad (unidades)	Costo (dólares)	Total (dólares)
Laptop.	4	48	1	$870.00	$72.50
Teléfono Android.	4	24	1	$300.00	$50.00

Total Hardware.					$122.50

Tabla 2.3 Estimación de costos (Hardware).
Elaborado por: Tex Albuja

Software:

En el mismo sentido dentro de la tabla 2.4 será calculado en función del tiempo de uso en el proyecto respecto a su vida útil. Así como también el uso de productos de código abierto y distribución gratuita, optimizan los costos del proyecto.

Software	Tiempo en Proyecto (meses)	Tiempo de Vida. (meses)	Cantidad (unidades)	Costo de Licencia (dólares)	Total (dólares)
Microsoft Office 2010	4	36	1	$220.00	$24.44
Microsoft Project 2010	4	36	1	$300.00	$33.00
Eclipse IDE	4	24	1	$0.00	$0.00
Android SDK	4	12	1	$0.00	$0.00
Soap UI	4	24	1	$0.00	$0.00
Facebook API	4	36	1	$0.00	$0.00
Twitter API	4	36	1	$0.00	$0.00
Google plus API	4	36	1	$0.00	$0.00
Total Software					$57.44

Tabla 2.4 Estimación de costos (Software).
Elaborado por: Tex Albuja

Logística e insumos:

Existen otros productos y servicios necesarios para el desarrollo del proyecto, estos costos serán calculados en función del tiempo requerido. Estos costos se encuentran detallados en la tabla 2.5.

Item.	Tiempo en Proyecto (meses)	Valor Mensual	Total (dólares)
Energía Eléctrica	4	$35.00	$140.00
Internet Banda Ancha 1Mbps	4	$20.16	$80.64
Telefonía e Internet móvil	4	$15.00	$60.00
Movilización	4	$20.00	$80.00
Otros	4	60.00	$240.00

Total Logística e insumos			$600.64

Tabla 2.5 Estimación de costos (Logística e Insumos).
Elaborado por: Tex Albuja

La tabla 2.6 suma los rubros obtenidos, y se adhiere el cálculo de la utilidad del proyecto, este costo dependerá de cuánto se desea ganar porcentualmente frente al costo total.

Descripción.	Total (dólares)
Recursos Humanos	$3280.00
Hardware	$122.50
Software	$57.44
Logística e Insumos	$80.00
Total Costos	$3539.94
Utilidad (3539.94*25%)	$884.98
Precio Total del Proyecto	$4424,92

Tabla 2.6 Estimación de costos (Resumen General).
Elaborado por: Tex Albuja

- **Esbozar el ciclo de vida del proyecto.**

  Organizar las iteraciones en un conjunto de fases. Cada fase del ciclo de vida del proyecto terminará con un hito destinado a proporcionar a los interesados los mecanismos de supervisión y de dirección para controlar la financiación del proyecto, el alcance, la exposición al riesgo, el valor previsto, y otros aspectos del proceso.

  Dentro de la tabla 2.7 se establecen las faces e hitos que el proyecto perfilará.

Fase del Proyecto	Hitos
Inicio	1. Definir el proyecto. 2. Identificar y refinar los requerimientos.
Elaboración	1. Desarrollar la Arquitectura. 2. Desarrollar un Incremento en la solución. 3. Desarrollar pruebas de concepto.
Construcción	1. Refinar los requerimientos. 2. Desarrollar un Incremento en la solución. 3. Desarrollar pruebas de concepto.

Tabla 2.7 Estimación de costos (Resumen General).
Elaborado por: Tex Albuja

Una vez realizados los pasos propuestos por la metodología, se elaborará el artefacto Plan de Proyecto.

Plan de Proyecto

Este artefacto describe un acuerdo inicial sobre cómo el proyecto va a realizar sus objetivos. El plan del proyecto resultante proporciona un resumen general a nivel del proyecto.

SocialSphere
Plan de Proyecto

1. Introducción

En el siguiente plan del proyecto se especifica la organización del proyecto SocialSphere, sus miembros y roles en el ciclo de vida del proyecto. Introduce áreas de trabajo que serán asignados a los miembros del equipo, proyectos relacionados y canales de comunicación.

2. Organización del Proyecto

El equipo de desarrollo estará conformado por Tex Albuja como desarrollador, y el Ing. Tito Armas como conductor de la solución. El proyecto relacionado es el caso de estudio "Vida Verde" mediante el cual se evidenciará el correcto diseño de la API.

3. Prácticas y Medidas del Proyecto.

El proyecto se desarrollará con las prácticas de desarrollo iterativo y la integración continua. Se evaluará el avance del proyecto para el desarrollo iterativo mediante listas de verificación de los objetivos definidos en la iteración y el porcentaje de componentes integrados efectivamente. Obteniendo métricas del avance del proyecto.

4. Hitos y objetivos del Proyecto.

Fase del Proyecto	Iteración	Objetivos primarios	Duración
Inicio	I1	1. Iniciar el proyecto. 2. Identificar y Refinar los Requerimientos 3. Aprobar el Enfoque Técnico	30 Días.
Elaboración	I2	1. Desarrollar la Arquitectura. 2. Desarrollar un Incremento en la solución. 3. Desarrollar pruebas de concepto.	45 Días.
Construcción	I3	1. Desarrollar la Documentación del Producto.	30 Días.

		2. Implementar la solución.	

5. Despliegue.

Se entregará un instalador que incluya las herramientas necesarias para el desarrollo de aplicaciones Android.
La librería será distribuida mediante el formato JAR que incluye un archivo MANIFEST, con información de cómo se puede utilizar el paquete. Esta librería puede ser importada directamente por los desarrolladores en sus proyectos.
Se entregará la documentación del proyecto, de instalación y del usuario.

2.1.2. PLANIFICAR Y ADMINISTRAR LA ITERACIÓN.

Esta actividad da inicio a la iteración permitiendo al equipo ser agregado a tareas de desarrollo. Permite la monitorización y comunicación del estado del proyecto a los interesados externos. Además identifica y maneja las irregularidades y problemas.

Esta actividad se realiza durante todo el ciclo de vida del proyecto. El objetivo es identificar los riesgos y los problemas con suficiente antelación para que puedan ser mitigados, establecer los objetivos de la iteración, y para apoyar al equipo de desarrollo para alcanzar tales objetivos.

Plan de iteración.

A continuación se presenta una visión general de los planes de iteración, correspondientes al ciclo de vida del Proyecto que servirá de ejemplo para la explicación. Todos los artefactos resultantes de esta actividad serán expuestos en el anexo 2.

Plan de Iteración

Iteración I – Inicio del Proyecto

1. Hitos Claves

Hito
Iteración I – Inicio del Proyecto
Iniciar el Proyecto
Desarrollar la Visión.
Planificar el Proyecto.
Identificar y Refinar los Requerimientos
Detallar los Casos de Uso
Detallar los Requerimientos Suplementarios.
Crear Casos de Prueba
Aprobar el Enfoque Técnico
Visualizar la Arquitectura

Objetivos a alto nivel.

- Definir el proyecto.
- Identificar y Refinar los requerimientos.

2. Asignaciones de Ítems de Trabajo

Nombre / Descripción	Prioridad	Tamaño Estimado	Recurso	Esfuerzo
Iteración 1				
Identificar a los interesados	Alta	4	TAL	10
Definir las características de la solución	Media	6	TAL	10
Planificar el Proyecto.	Media	3	TAL	12
Identificar y Perfilar los Requerimientos	Alta	1	TAL	16
Detallar los Casos de Uso	Media	3	TAL	20
Detallar los Requerimientos Suplementarios.	Media	4	TAL	20
Crear Casos de Prueba	Media	2	TAL	16
Aprobar el Enfoque Técnico	Alta	1	TAL	16
Visualizar la Arquitectura	Alta	3	TAL	20

2.1.2.1. Preparación del Ambiente.

Esta actividad permite al equipo estandarizar la manera en la que se instala un ambiente de desarrollo, con el fin de difundir el conocimiento dentro del equipo, y manejar consistencia entre los diferentes ambientes de desarrollo.

El desarrollo de las siguientes tareas se puede evidenciar en el anexo 3.

- **Adaptación del proceso**
- **Configuración de herramientas**
- **Comprobar la herramienta de configuración e instalación**
- **Implementar el proceso**

2.1.2.2. Administrar la iteración.

Esta tarea permite al equipo a cumplir los objetivos de iteración y mantener el proyecto en marcha. Maneja las expectativas de las partes interesadas como los descubrimientos técnicos y prácticos que se realizan durante el proyecto. Una administración efectiva es realizada a través de los objetivos y no por el reporte del tiempo de los recursos. Consiste además del monitoreo del progreso y del trabajo restante.

El involucramiento de las partes interesadas en la aprobación de cambios que sean necesarios para que el proyecto no se dilate, es un paso importante que debe ser resuelto por el gerente de proyecto.

OpenUp propone los siguientes pasos en esta actividad:

- **Seguimiento del progreso de la iteración actual**
- **Capturar y comunicar el estado del proyecto**
- **Tratar las excepciones y problemas**
- **Identificar y gestionar los riesgos**
- **Administrar objetivos**

2.1.2.3. Evaluar los resultados.

Esta tarea permite coordinar la evaluación y discutir con el equipo cómo los resultados de la iteración se presentarán a los interesados para que puedan aprender todo acerca de la solución como sea posible. Escuchar lo que el equipo tiene que decir acerca de lo que salió mal (y lo que salió bien) durante la iteración.

Este conocimiento ayudará a todos a tomar decisiones informadas sobre la planificación de la siguiente iteración, y determinar el mejor curso de acción para el proyecto. Esta tarea se lleva a cabo al final de cada iteración hasta el final del proyecto.

2.1.3. IDENTIFICAR Y REFINAR LOS REQUERIMIENTOS.

Esta actividad describe las tareas que se realizan para reunir, especificar, analizar y validar un subconjunto de los requisitos del sistema antes de la implementación y verificación. Esto no implica que todos los requisitos se detallen antes de comenzar su ejecución. Por el contrario, esta actividad se realiza durante todo el ciclo de vida con los actores y el equipo de desarrollo colaborando para asegurar que un conjunto claro, consistente, correcto, verificable y viable de los requisitos esté disponible, cuando sea necesario, para orientar la aplicación y verificación.

Es necesario realizar las siguientes precisiones que aplicarán para la solución SocialSphere.

- Debido a que la solución se incluye como un componente interno de una aplicación, ningún usuario puede utilizar o consumir directamente las funcionalidades que ofrece la solución.
- Los mecanismos que la solución reúne no representan acciones de valor para los usuarios, ya que se enfocan en proveer a los desarrolladores elementos que incorporen estos mecanismos en sus aplicaciones.

2.1.3.1. Identificar y Describir los requerimientos.

Esta tarea describe cómo identificar y delinear los requisitos para el sistema de modo que el alcance del trabajo se pueda determinar.

- **Recopilar Información**
- **Identificar y capturar los términos del dominio**
- **Identificar los tipos de requisitos que afecten al sistema.**

Colaborar con los interesados para identificar los tipos de requisitos que afecten a su sistema. Esto ayudará a que se evalúe la integridad del conjunto de requisitos. La tabla 2.8 presenta la definición de requerimientos para el proyecto.

Requerimiento	Descripción.
ProvideAPIKey	Proveer las credenciales de la aplicación a la red social.
RegisterUser	Vincular a un usuario a la red social.
GetCurrentUser	Obtener las propiedades del usuario de la red social.
GetContent	Solicitar un contenido a la red social.
GetRegisteredUsers	Obtener la lista de usuarios que han sido registrados en la aplicación.
ShareContent	Compartir un contenido a la red social.
SendAppRequest	Enviar una solicitud de la aplicación a la red social.
SendAppInvitation	Enviar una invitación de uso de la aplicación.

Tabla 2.8 Definición de Requerimientos.
Elaborado por: Tex Albuja

2.1.3.2. Detallar los escenarios de casos de Uso.

El objetivo de esta tarea es describir los escenarios de casos de uso con suficiente detalle para validar la comprensión de los requisitos, garantizando la coincidencia con las expectativas de las partes interesadas, y para permitir que inicie el desarrollo de software.

Modelo de Casos de Uso.

A continuación se presenta una vista general de las funcionalidades que presentará la solución representados como casos de uso.

SocialSphere

Modelo de Casos de Uso

1. Introducción.

El presente modelo de casos de uso identifica los actores y los casos de uso presentes en la solución SocialSphere.

2. Perspectiva General.

SocialSphere es una librería que sirve como capa de abstracción de la lógica que permite conectar a redes sociales.

3. Actores.

3.1 Aplicación Android. Aplicación desarrollada bajo la plataforma ADT que requiere consumir diversos servicios de redes sociales.

3.2 Red Social. Proveedor de servicios de redes sociales.

3.3 Librería SocialSphere. Librería desarrollada para minimizar el costo de desarrollo en un proyecto Android.

3.4 Usuario de la Red Social. Usuario que hace uso de la red social, y que se ha interesado en utilizar la aplicación.

4. Casos de Uso.

4.1 ProvideAPIKey.

Este caso de uso describe cómo una aplicación Android proveerá sus credenciales a la red social con la cual interactuará.

4.2 RegisterUser.

Este caso de uso describe cómo una aplicación Android vinculará a un usuario autenticado en una red social.

4.3 GetCurrentUser

Este caso de uso describe cómo una aplicación Android obtendrá las propiedades del usuario que se ha autenticado en una red social.

4.4 GetContent

Este caso de uso describe cómo una aplicación Android solicitará un contenido como fotos, actualizaciones de estado, videos o lista de amigos.

4.5 GetRegisteredUsers

Este caso de uso describe cómo una aplicación Android obtendrá una lista de usuarios que son amigos del usuario actual y que además han sido registrados en el ambiente de la aplicación.

4.6 ShareContent

Este caso de uso describe cómo una aplicación Android compartirá contenido como fotos,

actualizaciones de estado, videos o lista de amigos.

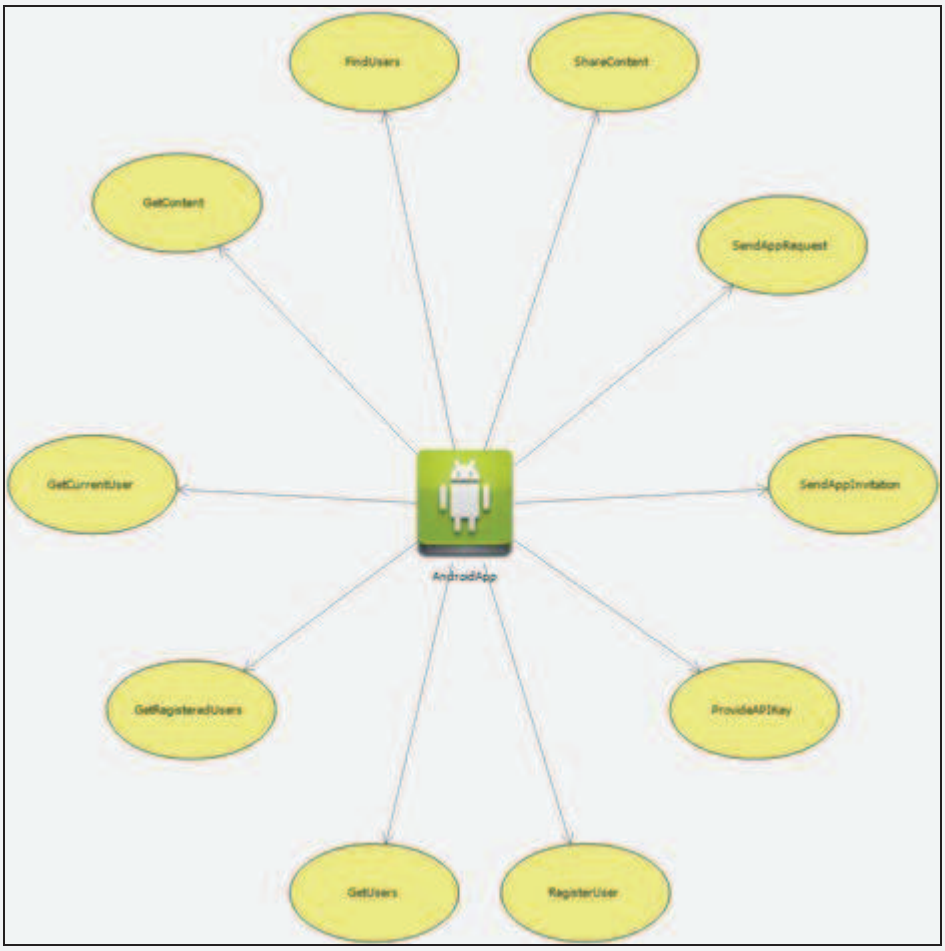
4.7 SendAppRequest

Este caso de uso describe cómo una aplicación Android enviará a un amigo del usuario actual, una petición de la aplicación.

4.8 SendAppInvitation

Este caso de uso describe cómo una aplicación Android enviará una invitación amigo del usuario actual para que se integre al ambiente de la aplicación.

5. Diagrama de Casos de Uso.



Especificación de Casos de Uso.

A continuación y a manera de ejemplo se presenta la especificación del caso de uso *ProvideAPIKey*.

SocialSphere

Caso de uso: ProvideAPIKey

1. Descripción Breve.

Este caso de uso describe cómo una aplicación Android proveerá sus credenciales a la red social con la cual interactuará.

2. Descripción breve de los actores.

a. Aplicación Android. Aplicación desarrollada bajo la plataforma ADT que requiere consumir diversos servicios de redes sociales.

b. Red Social. Proveedor de servicios de redes sociales.

c. Librería SocialSphere. Librería desarrollada para minimizar el costo de desarrollo en un proyecto Android.

d. Usuario de la Red Social. Usuario que hace uso de la red social, y que se ha interesado en utilizar la aplicación.

3. Precondiciones

La aplicación se encuentra registrada en la Red Social.

API Key.

4. Flujo Básico de eventos.

1. El caso de uso comienza cuando la *Aplicación Android* es lanzada.
2. La *Aplicación Android* provee sus credenciales a la *librería SocialSphere*.
3. *La librería SocialSphere* autentica la aplicación a la *Red Social* vinculada.
4. El caso de uso termina satisfactoriamente almacenando el Access Token asociado.

5. Flujo de Excepción.

a. Sin respuesta del proveedor.

Si en el paso 3 del flujo básico de eventos la *librería SocialSphere* no ha recibido una respuesta entonces

4. *La librería SocialSphere* despliega un diálogo con el error “No existe respuesta desde el proveedor”.
5. *La librería SocialSphere* agrega a un log la información del error.
6. El caso de uso termina cuando *La librería SocialSphere* ha propagado la excepción.

b. Permiso denegado a red.

Si en el paso 3 del flujo básico de eventos el conector no ha recibido una respuesta entonces

4. *La librería SocialSphere* despliega un diálogo con el error “No se han brindado los permisos de

uso de red".

5. *La librería SocialSphere* agrega a un log la información del error.

6. El caso de uso termina cuando *La librería SocialSphere* ha propagado la excepción.

c. API Key y la firma de la aplicación no concuerdan.

Si en el paso 3 del flujo básico de eventos *La librería SocialSphere* ha recibido como respuesta : invalid_key

4. La librería SocialSphere despliega un mensaje con el error "La aplicación no ha podido ser registrada con el proveedor".
5. La librería SocialSphere agrega a un log la información del error específico.
6. El caso de uso termina cuando La librería SocialSphere ha propagado la excepción.

6. Post-condiciones.

a. Post-condición de éxito.

- El conector almacena el Access Token.

b. Post-condición de excepción.

- El conector almacena el log.

7. Requerimientos Especiales.

[SpReq: PAK-1]: La librería proveerá un mecanismo adecuado de logging.

[SpReq: PAK-2]: La librería debe proveer un mecanismo adecuado de almacenamiento del Access Token.

[SpReq: PAK-3]: La librería debe proveer un mecanismo de despliegue de diálogos.

2.1.3.3. Especificar los Requerimientos Suplementarios.

En esta actividad se describirán los requerimientos suplementarios que resolverán las necesidades que se han encontrado en el proyecto, sin expresarlos como casos de uso y especificando las cualidades que el sistema debe proveer.

Especificación de Requerimientos Suplementarios.

El objetivo de este artefacto es capturar los requerimientos funcionales que no son expresados como casos de usos. Brindar asesoramiento del tamaño, costo y viabilidad del sistema propuesto. Comprender los requerimientos de niveles de servicio para la administración operacional de la solución.

A continuación se expone el artefacto como resultado del desarrollo de los pasos propuestos por la metodología.

SocialSphere
Especificación de Requerimientos Suplementarios.

1. Introducción

El presente documento está dirigido a los desarrolladores y los interesados con el objetivo de especificar los requerimientos que constituirán el punto de partida, estos requerimientos no serán expresados como casos de uso.

2. Cualidades del Sistema.

2.1 Usabilidad

Dentro de los requerimientos de usabilidad de la librería se incluyen.
Librería de fácil uso.
Librería de fácil aprendizaje.
Librería basada en estándares de programación.
Librería extensible para cualquier red social.
Librería de fácil adquisición.

2.2 Confiabilidad.

La librería debe proveer mecanismos para garantizar:
Disponibilidad en repositorios de código de fácil acceso.
Manejo adecuado de excepciones.

2.3 Rendimiento.

SocialSphere debe satisfacer el uso mínimo de recursos de red, procesamiento, memoria y garantizar el tiempo de respuesta.

2.4 Soporte.

Para mejorar el soporte y mantenimiento de la librería se proveerá la documentación del proceso de desarrollo, entregables, código fuente mediante repositorios de código colaborativos.

3. Interfaces del Sistema.

El sistema será entregado en formato .jar que podrá ser consumido por las aplicaciones mediante la codificación a través de sus clases.

3.1 Interfaces de Usuario.

Debido a la naturaleza que posee la librería las interfaces gráficas serán presentadas mediante el caso de estudio Vida Verde.

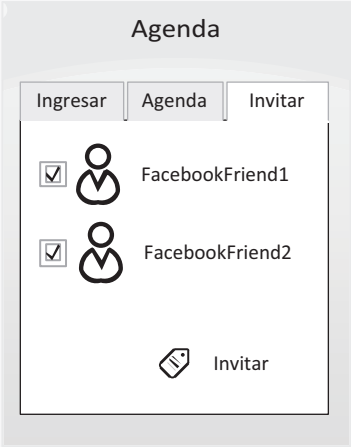
3.1.1 Apariencia.

Iniciar Sesión

Obtener Tip Verde

Organizar Evento

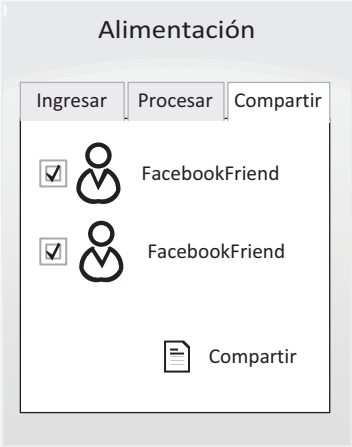
Revisar Agenda



Agregar Invitados

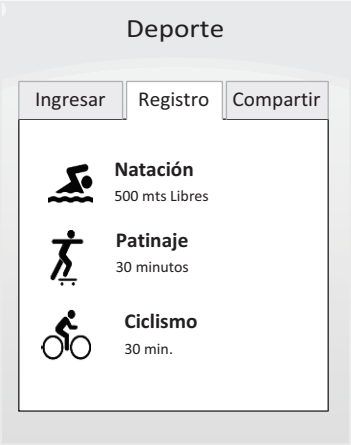
Ingresar Receta

Explorar Recetas



Compartir Receta

Ingresar Deporte



Registrar Deporte

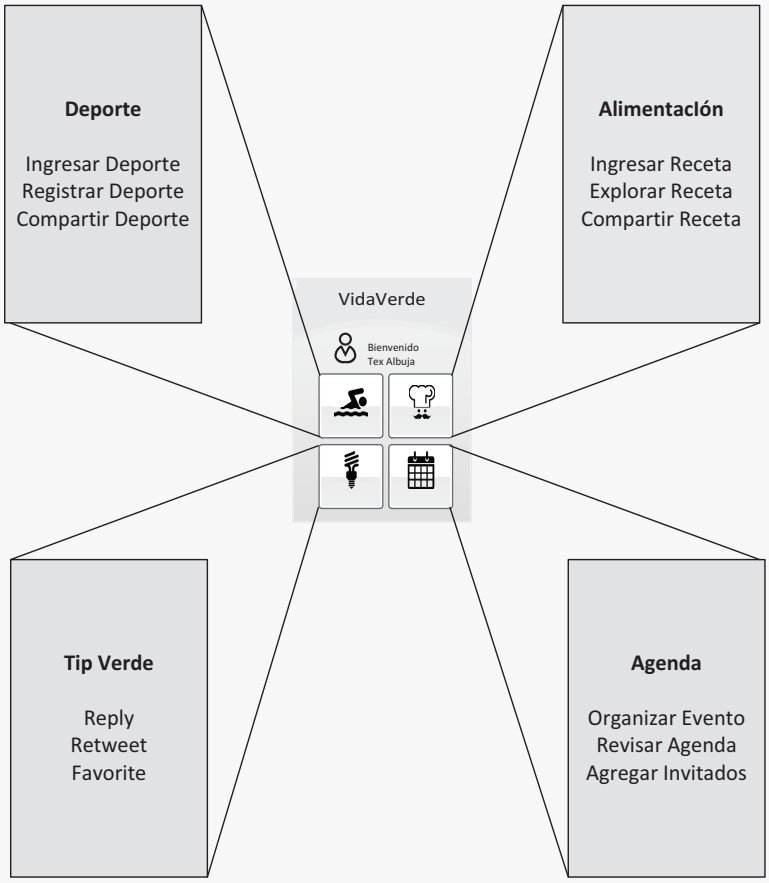
Compartir Deporte

3.1.2 Requerimientos de Navegabilidad.

Los requerimientos de navegabilidad se presentan bajo el almacén de las credenciales durante el ciclo de vida de la aplicación.

La navegabilidad de la aplicación viene definida en el siguiente esquema.

Se tiene una pantalla principal, de donde se desprenden pantallas que poseen pestañas cada una de las cuales ha sido identificada anteriormente.



3.1.3 Consistencia

Las clases deben guardar consistencia en cómo se implementan a lo largo de la aplicación y en lo posible en el portafolio de redes sociales que soportará. Los mecanismos que provea la librería deben ser de fácil presentación y terminología.

3.1.4 Requerimientos de personalización.

La librería debe ser extensible, es decir que a partir del código fuente se puedan agregar o quitar redes sociales dependiendo de los requerimientos de la aplicación que la va a utilizar.

3.2 Interfaces a sistemas externos.

La librería interactuará con las API's de las Redes Sociales mediante el consumo de Web-Services. Los formatos que se utilizarán en el envío y recepción de peticiones y respuestas serán mediante el protocolo REST y los formatos XML o JSON.

3.2.1 Interfaces de Software.

La librería proveerá mecanismos para el óptimo consumo de recursos propios de la aplicación.

3.2.2 Interfaces de Hardware.

La librería será definida para dispositivos móviles que puedan soportar el sistema operativo Android 2.2 con un nivel mínimo de versión de API en 8.

3.2.3 Interfaces de Comunicación.

La librería consumirá los servicios comunicación del sistema operativo Android, siendo la administración de estos adaptadores independientes de los desarrolladores.

2.1.4. APROBAR EL ENFOQUE TÉCNICO.

El objetivo de esta actividad es definir un enfoque técnico para el sistema que soporte adecuadamente los requisitos del proyecto, dentro de las limitaciones que el contexto del sistema presenta.

Las diferentes acciones que componen esta actividad son la principal razón por la que ha sido escogida la metodología OpenUp.

2.1.4.1. Visualizar la Arquitectura.

Esta tarea por su complejidad desprende los siguientes pasos que se describirán con profundidad en el anexo 4.

- **Identificar los objetivos arquitectónicos**
- **Identificar los requisitos de gran importancia arquitectónica**
- **Identificar las limitaciones en la arquitectura**
- **Identificar las abstracciones clave**
- **Identificar las oportunidades de reutilización**
- **Definir el enfoque para dividir el sistema**
- **Definir el enfoque para la implementación del sistema**
- **Identificar los mecanismos de arquitectura**
- **Identificar las interfaces con sistemas externos**
- **Verificar la coherencia arquitectónica**
- **Capturar y comunicar las decisiones arquitectónicas**

Todas estas tareas permitirán elaborar el Diario de Arquitectura que será expuesto en la sección 2.2.2.

2.2. FASE DE ELABORACIÓN.

Durante la elaboración, la atención se traslada a la definición de la solución, esta consiste en encontrar los requisitos que tienen el mayor valor para los stakeholders, cuáles son particularmente difíciles o de riesgo, o que son de gran importancia arquitectónica. A continuación se describen los requisitos con suficiente detalle para validar la comprensión del equipo de desarrollo garantizando la aceptación de las partes interesadas, y para comenzar el desarrollo. Para cada uno de estos requisitos, el definir los casos asociados a prueba asegura que los requerimientos sean verificables.

2.2.1. PLANIFICAR Y ADMINISTRAR LA ITERACIÓN.

En el mismo sentido que en punto 2.1.2, Se planificará y administrará la iteración concentrando a entender con más detalle los requisitos, diseñar, implementar, validar y fundamentar una arquitectura, mitigar los riesgos esenciales.

Plan de Iteración

Iteración II – Elaboración del Proyecto

1. Hitos Claves

Hito
Iteración II – Elaboración del Proyecto
Desarrollar la Arquitectura
Visualizar la Arquitectura
Desarrollar un Incremento de la Solución
Diseñar la Solución
Implementar Test de Desarrollo
Implementar la Solución
Realizar Pruebas Unitarias
Integrar y Crear una Compilación

2. Objetivos a alto nivel.

- Diseñar la solución.
- Implementar un incremento de la solución.
- Generar el primer entregable.

3. Asignaciones de Ítems de Trabajo

Nombre / Descripción	Prioridad	Tamaño Estimado	Recurso	Esfuerzo
Iteración 2				
Desarrollar la Arquitectura	Alta	8	TAL	10
Elaborar el Diario de Arquitectura	Media	8	TAL	10
Desarrollar un Incremento de la Solución	Media	10	TAL	110
Diseñar la Solución	Alta	2	TAL	20
Implementar Test de Desarrollo	Media	3	TAL	30
Implementar la Solución	Media	3	TAL	30
Realizar Pruebas Unitarias	Media	1	TAL	20
Integrar y Crear una Compilación	Alta	1	TAL	10

2.2.2. DESARROLLO DE LA ARQUITECTURA.

La arquitectura evoluciona a medida que la solución se diseña e implementa, y la documentación de arquitectura se actualiza para reflejar los cambios realizados durante el desarrollo.

Diario de Arquitectura.

El objetivo principal del diario de arquitectura es capturar y formalizar las decisiones para poder exponerlas a los desarrolladores. Además informa a los miembros del equipo de como está particionado el sistema y su organización, para que el equipo lo pueda adaptar a las necesidades del sistema.

SocialSphere
Diario de Arquitectura

1. Propósito

El propósito de este documento es describir la filosofía, decisiones, limitaciones, justificaciones y elementos y cualquier otro aspecto que puede delinear el diseño y la implementación del sistema.

2. Objetivos y filosofía de la Arquitectura.

- La librería debe responder y adaptarse a los cambios en la definición de los servicios prestados por los servicios de redes sociales escogidos.
- Los cambios que se produzcan en las diferentes API's deben ser soportadas mediante la actualización de la librería, sin necesidad de realizar cambios en la programación de la aplicación.

3. Suposiciones y Dependencias.

- Se asume que el equipo posee conocimientos altos del lenguaje de programación Java.
- Se asume que se probará en dispositivos basados el sistema operativo Android 2.2 en adelante.
- La solución depende de las definiciones descritas en las API's de cada proveedor de servicios de redes sociales.

4. Requisitos de gran importancia arquitectónica.

Debido a que la solución que se propone, será utilizada por aplicaciones de propósito general, se han elegido los requisitos de la aplicación VidaVerde como trascendentales.

5. Decisiones, limitaciones, y justificaciones.

- La solución debe minimizar el consumo de recursos de red del sistema.
- La solución debe proveer documentación de cada mecanismo en forma de talleres.
- La solución deberá implementarse en versiones superiores del SDK de Android.
- La solución deberá minimizar el consumo de memoria física al agregarse a la aplicación Android.

6. Mecanismos de arquitectura.

Los mecanismos arquitectónicos son soluciones comunes a los problemas comunes que pueden ser utilizados durante el desarrollo para minimizar la complejidad.

6.1 Sesión.

La solución proveerá acceso a las variables propias de la sesión iniciada por el usuario a través de la exposición de estas propiedades a capas superiores.

6.2 Acceso a contenido.

La solución manejará el acceso a contenido, abstrayendo los permisos que tenga el usuario sobre los recursos de la red social.

6.3 Sincronización de contenido.

La solución sincronizará el contenido dentro del ciclo de vida de la aplicación, en el caso que se lo requiera a través de las firmas de los métodos que se van a utilizar.

6.4 Manejo de Excepciones.

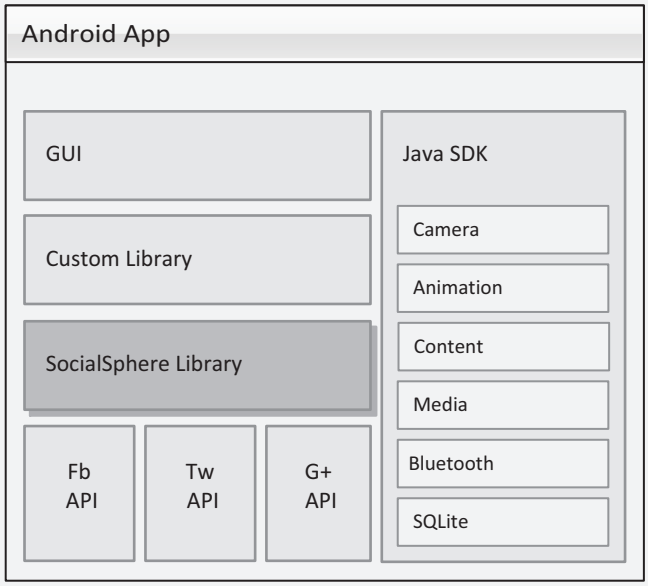
La solución abstraerá las excepciones generadas en el consumo de API's evitando que se necesiten manejar en la aplicación.

6.5 Notificaciones.

La solución presentará al usuario final notificaciones sobre las actividades que sean agregadas en tiempo de ejecución.

7. Capas y Framework arquitectónico.

En el desarrollo de aplicaciones móviles típicamente se posee una capa de interfaz de usuario (GUI), seguida por las librerías que se generan como resultado de la abstracción de la lógica del negocio (Custom Library). La librería SocialSphere (SocialSphere Library), se ubica debajo de esta capa esencialmente para proveer mecanismos de abstracción que harán transparente al desarrollador las operaciones necesarias para la comunicación con API's de redes sociales (Facebook API, Twitter API, GooglePlus API). Finalmente y a nivel transversal se utilizan las librerías que están presentes en el kit de desarrollo de Android (Java SDK).



8. Vistas de la Arquitectura.

Se utilizarán la vista lógica de la arquitectura la cual describe la estructura y comportamiento de porciones significantes para el sistema. Puede incluir además la estructura de paquetes, interfaces, clases y subsistemas críticas y sus relaciones.

8.1 Vista lógica de la Arquitectura.

2.2.3. DESARROLLO DE UN INCREMENTO DE LA SOLUCIÓN.

La atención se enfoca en llevar a cabo los elementos de trabajo o tareas descritos para el proyecto. El diseño de la solución debe encontrarse completo y estable para cada uno de los componentes involucrados.

2.2.3.1. Estructura de paquetes.

La librería SocialSphere cuenta con una estructura de paquetes perfilada en la figura 2.1. En esta distribución se encuentra el paquete socialsphere.model, el cual define los objetos y las operaciones comunes a las redes sociales que se implementarán, con el objetivo de que sea un paquete de uso común.

Este paquete además provee de elementos que facilitarán la implementación de nuevas redes sociales a la librería, proporcionando extensibilidad

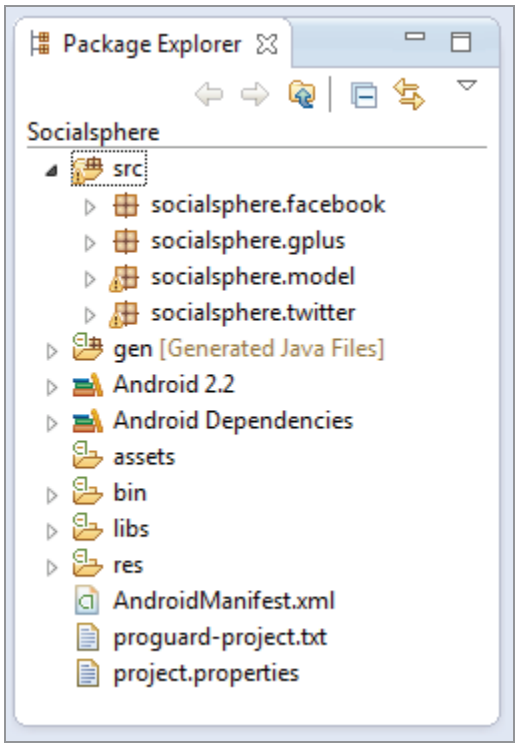


Figura 2.1 Estructura de Paquetes de SocialSphere

2.2.3.2. Estructura de librerías.

SocialSphere, a su vez consume librerías para abstraer diferentes aspectos del desarrollo. Las librerías se encuentran en la carpeta libs, la cuál ha sido especificada en los estándares de desarrollo de Android ya que facilita el manejo de referencias.

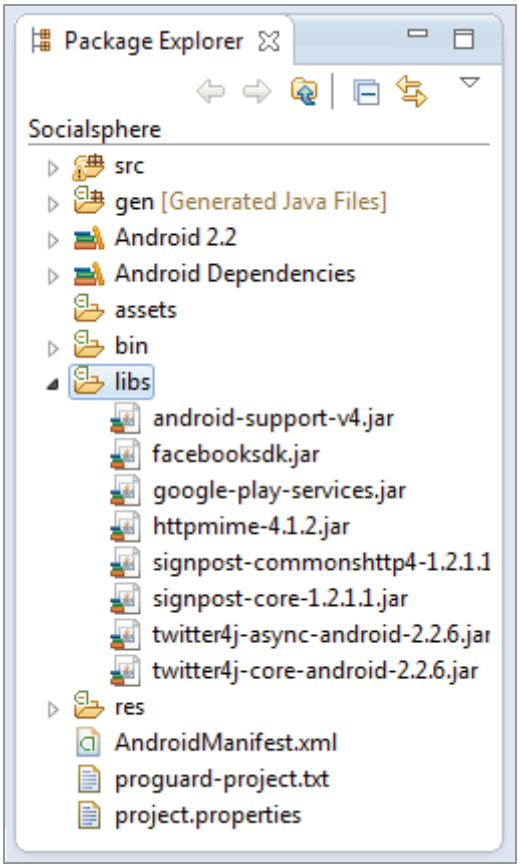


Figura 2.2 Estructura de Librerías Referenciadas en SocialSphere.

2.2.3.3. Estructura de clases.

La estructura de clases está diseñada en el siguiente esquema.

Se compone del paquete socialsphere.model que contiene las definiciones de los elementos comunes para todas las redes sociales, la implementación de estas definiciones vienen dada en cada paquete que representa la conectividad a una red social.

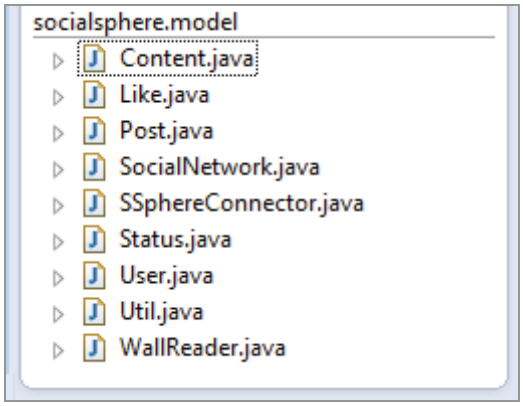


Figura 2.3 Estructura de Clases en el Paquete SocialSphere.

Seguidamente se tienen los paquetes socialsphere.facebook, socialsphere.gplus, socialsphere.twitter, los cuales contienen las clases que permitirán abstraer la lógica necesaria para cumplir con las definiciones de socialsphere.model.

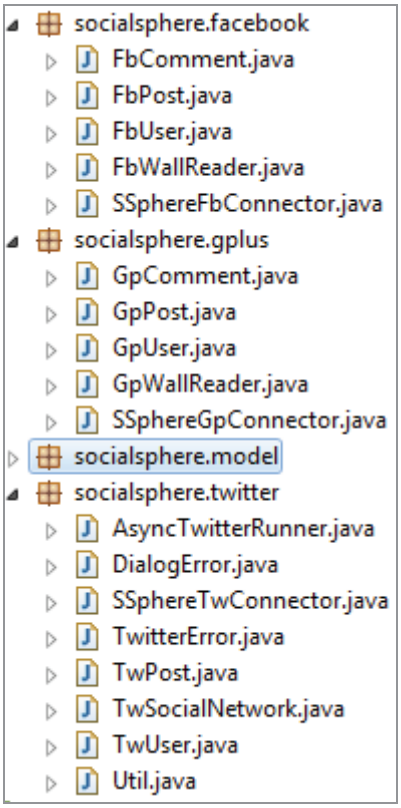


Figura 2.4 Estructura de Clases que implementan la definición de socialsphere.model.

Con la solución definida, se tienen l en la fase de construcción de la librería en donde se desarrollarán la documentación y los entregables necesarios.

2.3. FASE DE CONSTRUCCIÓN.

Durante la construcción, la atención se desplaza a refinar la definición del sistema. Esta consiste en detallar el resto de requisitos y casos asociados prueba si es necesario para impulsar la aplicación y verificación, y la gestión de los requerimientos cambian. Además se desarrollarán la documentación y los entregables de la solución.

2.3.1. PLANIFICAR Y ADMINISTRAR LA ITERACIÓN.

En el mismo sentido que en punto 2.1.2, Se planificará y administrará esta iteración enfocando a la implementación de la solución generando los entregables para el cliente y la documentación necesaria.

Plan de Iteración

Iteración III – Construcción del Proyecto

1. Hitos Claves

Hito
Iteración III – Construcción del Proyecto
Desarrollar la Documentación del Producto
Desarrollar la Documentación del Producto
Desarrollar la Documentación de Usuario
Desarrollar la Documentación de Soporte
Implementar la Solución
Generar los Instaladores.
Validar la Instalación.

2. Objetivos a alto nivel.

- Generar la documentación necesaria para proveer en los entregables.
- Minimizar el costo de soporte al usuario.
- Generar un conjunto de instaladores.

3. Asignaciones de Ítems de Trabajo

Nombre / Descripción	Prioridad	Tamaño Estimado	Recurso	Esfuerzo
Iteración 3				
Desarrollar la Documentación del Producto	**Media**	20	TAL	60
Desarrollar la Documentación del Producto	Media	10	TAL	20
Desarrollar la Documentación de Usuario	Media	6	TAL	20
Desarrollar la Documentación de Soporte	Media	4	TAL	20
Implementar la Solución	Media	12	TAL	20
Generar los Instaladores.	Media	8	TAL	10
Validar la Instalación.	Media	4	TAL	10

2.3.2. DESARROLLAR LA DOCUMENTACIÓN DEL PRODUCTO

2.3.2.1. Desarrollar la Documentación del Producto.

SocialSphere es una librería Android que le permite centrarse en su problema de negocio en lugar de tener el conocimiento que conecta su aplicación móvil con las redes sociales de mayor auge a nivel mundial lo que acelera el reconocimiento y difusión de su marca.

Tanto desarrolladores expertos como principiantes pueden incrementar su productividad y eficiencia utilizando las características que ofrece. Olvide complejos algoritmos de autenticación, manejo de respuestas y otros aspectos

técnicos involucrados en el desarrollo. Ahora es fácil construir, ejecutar y administrar sus modernas aplicaciones Android.

SocialSphere, al utilizar la infraestructura TI implementada en las redes sociales, le ofrecerá un gran ahorro de recursos TI, los cuáles derivarán en la deducción de costos de mantenimiento entre otros.

2.3.2.2. Desarrollar la Documentación de Usuario.

La documentación de usuario se presentará en un formato wiki, el cual podrá ser accedido en la página web http://socialshpere.wikispaces.com

2.3.2.3. Desarrollar la Documentación de Soporte.

La documentación de soporte será presentada mediante los artefactos javadoc, y faq. Esta documentación brindará soporte de primer nivel, adicionalmente se pueden generar requerimientos de soporte en la página sourceforge.net en donde estará alojada la librería.

2.3.3. IMPLEMENTAR LA SOLUCIÓN

2.3.3.1. Generar los Instaladores.

Se ha desarrollado la solución con un único instalador, el cual verifica si el usuario tiene en su computador los requisitos de software necesarios, y en el caso de no poseerlos los instala. En la figura 2.1 y 2.2 se presentan las interfaces gráficas del instalador.

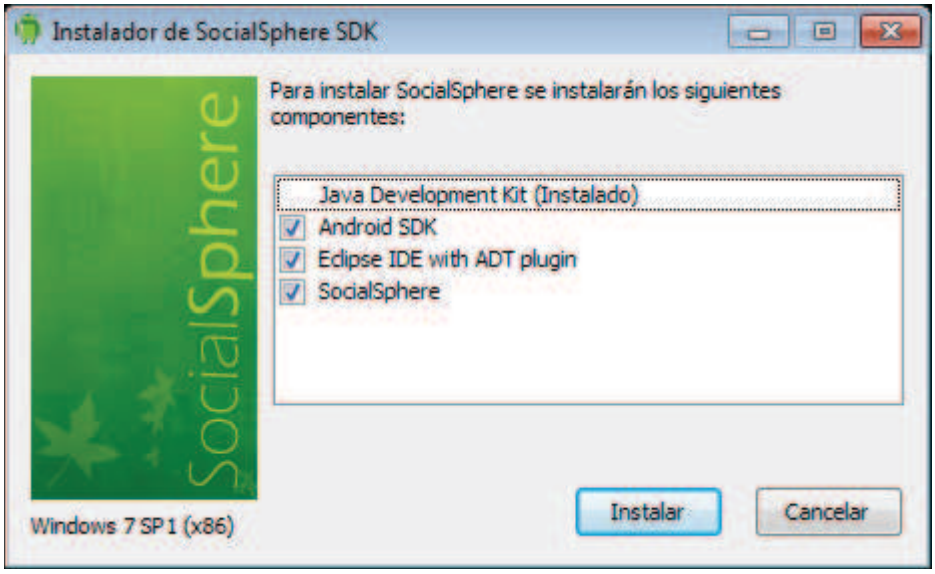


Figura 2.5 Verificación de Requisitos de Software.

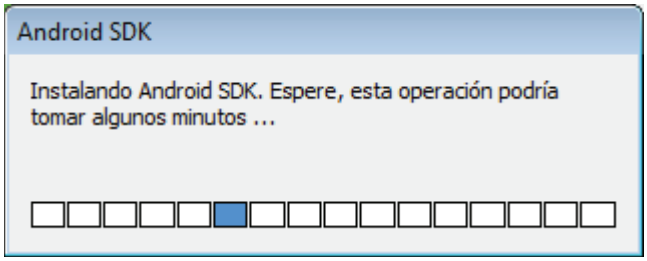


Figura 2.6 Instalación silenciosa de Android SDK.

Gracias a la abstracción de diferentes aspectos como la arquitectura, el sistema operativo y otros componentes que varían entre instalaciones, hacen eficiente el proceso de instalación.

2.3.3.2. Validar la Instalación.

Al finalizar el instalador, se ejecutan automáticamente validadores de que todos los componentes han sido instalados, además se provee un mecanismos

La prueba de la solución será presentada mediante el siguiente capítulo en la que se validará el correcto diseño a través de la elaboración de la aplicación prototipo VidaVerde. Esta aplicación guiará a los desarrolladores de la librería a través de la naturaleza de cada mecanismo de arquitectura que ha sido propuesto.

CAPÍTULO 3. ELABORACIÓN DE UN PROTOTIPO COMO CASO DE ESTUDIO.

Este capítulo presentará el desarrollo de la aplicación VidaVerde la cual consumirá la librería SocialSphere siendo parte de la definición de sus componentes, delineando su funcionalidad y verificando su diseño.

Debido a que el desarrollo del prototipo es un elemento de apoyo para el desarrollo de la librería, este validará la solución a través del cumplimiento de sus casos de uso así como el cumplimiento de los objetivos de la librería SocialSphere.

El caso de estudio VidaVerde, presenta a los desarrolladores una forma de ejemplificar el funcionamiento de la librería SocialSphere.

3.1. DEFINICIÓN DE LOS REQUISITOS DEL PROTOTIPO.

El prototipo permitirá observar el funcionamiento de los componentes definidos como parte de la solución SocialSphere. Como usuario final se observa la concientización de las personas a través de la difusión de consejos para el cuidado del medio ambiente, actividades deportivas, recetas y agenda de eventos verdes.

- VidaVerde es una aplicación Android que validará los componentes desarrollados como parte de la solución SocialSphere.
- Utilizará las redes sociales como medio para la propagación de estos consejos.
- Permitirá el envío de solicitudes de la aplicación a usuarios de otras redes sociales.
- Permitirá el despliegue de consejos desde cuentas de Twitter.
- Se podrá conectar con cuentas de Facebook y GooglePlus.
- Compartirá contenido en las redes sociales.

3.1.1. MODELO DE CASOS DE USO DEL PROTOTIPO

A partir de la definición de funcionalidades que deben ser incorporadas en el prototipo, se puede establecer los casos de uso que se encuentran enumerados en la Tabla 3.1

Clave	Caso de Uso	Caso(s) de Uso de la librería utilizados.
VidaVerde001	Iniciar Sesión	ProvideAPIKey, GetCurrentUser.
VidaVerde002	Obtener Tip Verde	GetContent.
VidaVerde003	Organizar Evento	SendAppRequest.
VidaVerde004	Revisar Agenda	GetContent.
VidaVerde005	Agregar Invitados	SendAppInvitation, GetUsers
VidaVerde006	Ingresar Receta	GetCurrentUser
VidaVerde007	Explorar Recetas	GetContent.
VidaVerde008	Compartir Receta	Like

Tabla 3.1 Casos de Uso del Prototipo.
Elaborado por: Tex Albuja

3.1.2. DISEÑO DE LAS INTERFACES GRÁFICAS DEL PROTOTIPO.

Con el objetivo de delinear los casos de uso descritos, se ha diseñado un conjunto de vistas asociadas a cada Caso de Uso.

Iniciar Sesión.		
	Descripción funcional. Permite tener un acceso fácil a las funcionalidades que provee la aplicación. Se presenta un cuadro de diálogo para iniciar sesión en cada una de las redes sociales. **Descripción en la librería.** Proveer los Id's de la aplicación a las redes Sociales. Propagar los Access Tokens, como resultado de la autenticación. Almacenar las preferencias del usuario en el teléfono Android. **Observaciones.** Se debe manejar las excepciones generadas por flujos de excepción en la autenticación.	
Número : 1	**Riesgo :Alto**	**Red Social : Fb, Tw, G+**

Tabla 3.2 Vista Principal.
Elaborado por: Tex Albuja

Tip.		
	Descripción funcional. Obtener tips para ayudar al planeta, permitir responder a este aporte, difundirlo en su perfil de twitter, o marcarlo como favorito. **Descripción en la librería.** Consumir feeds de usuarios específicos de la red social. Proveer la funcionalidad de un pequeño cliente de twitter. **Observaciones.** Se debe proveer un mecanismo adecuado de actualización de feeds.	
Número : 1	**Riesgo :Bajo**	**Red Social: Tw.**

Tabla 3.3 Tip.
Elaborado por: Tex Albuja

Organizar Evento		
	Descripción funcional. Organizar un evento que esté relacionado con temas ambientales para el desarrollo sustentable del planeta. **Descripción en la librería.** Manejar la funcionalidad de eventos relacionada con la red social Facebook. **Observaciones.**	
Número : 1	**Riesgo :Medio**	**Red Social : Fb.**

Tabla 3.4 Agenda - Ingresar.
Elaborado por: Tex Albuja

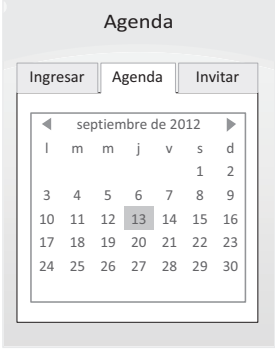
Revisar Agenda		
	**Descripción funcional.** Ver los eventos próximos en un calendario. **Descripción en la librería.** Manejar la funcionalidad de eventos relacionada con la red social Facebook. **Observaciones.**	
Número : 1	**Riesgo : Medio**	**Red Social: Fb.**

Tabla 3.5 Agenda - Explorar.
Elaborado por: Tex Albuja

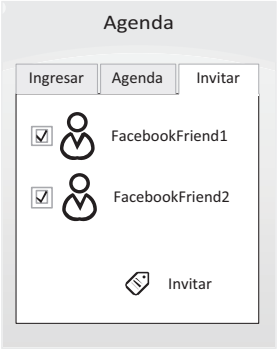
Agregar Invitados		
	**Descripción funcional.** Agregar amigos a los eventos relacionados con temas ambientales. **Descripción en la librería.** Obtener la lista de amigos del usuario conectado a la red social Facebook. Manejar el estado de las invitaciones realizadas. **Observaciones.**	
Número : 1	**Riesgo : Medio**	**Red Social: Fb.**

Tabla 3.6 Agenda - Invitar.
Elaborado por: Tex Albuja

Ingresar Receta.		
	Descripción funcional. Ingresar una receta que sea saludable, y poder compartirla en su red social. **Descripción en la librería.** Manejar la funcionalidad de post en el muro, con una actividad diferente que permita extender la experiencia de usuario. **Observaciones.**	
Número : 1	**Riesgo :Medio**	**Red Social: Fb.**

Tabla 3.7 Alimentación - Ingresar.
Elaborado por: Tex Albuja

Explorar Recetas.		
	Descripción funcional. Explorar las recetas ingresadas por sus amistades en la red social. **Descripción en la librería.** Manejar consultas sobre las amistades de una actividad relacionada con la aplicación. **Observaciones.**	
Número : 1	**Riesgo :Medio**	**Red Social : Fb.**

Tabla 3.8 Alimentación - Explorar.
Elaborado por: Tex Albuja

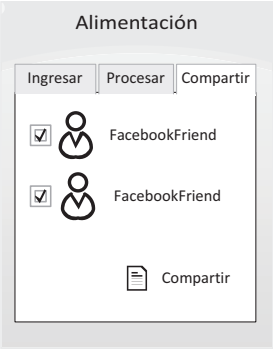
CompartirReceta.		
	**Descripción funcional.** Compartir una receta en el muro de uno o varios amigos, con una descripción. **Descripción en la librería.** Compartir contenido en el muro de uno o varios amigos. **Observaciones.**	
Número : 1	**Riesgo :**	**Red Social : Fb.**

Tabla 3.9 Alimentación - Compartir.
Elaborado por: Tex Albuja

Ingresar Deporte		
	Descripción funcional. Ingresar una actividad física realizada, con una descripción del trabajo hecho. **Descripción en la librería.** Ingresar una actividad realizada por la aplicación, y agregarla en el muro de la persona. **Observaciones.**	
Número : 1	**Riesgo :Medio**	**Red Social : G+**

Tabla 3.10 Deporte - Ingresar.
Elaborado por: Tex Albuja

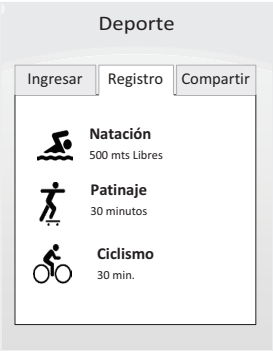
Registrar Deporte		
	**Descripción funcional.** Manejar un registro de la actividad deportiva realizada. **Descripción en la librería.** Permite analizar la actividad de una aplicación relacionada con un usuario de la red social G+ **Observaciones.**	
Número : 1	**Riesgo :**	**Red Social : G+**

Tabla 3.11 Deporte - Registro.
Elaborado por: Tex Albuja

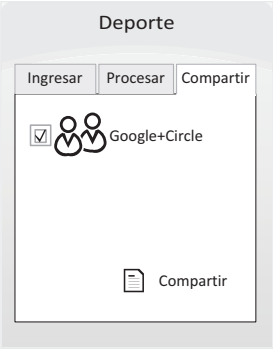
Compartir Deporte.		
	**Descripción funcional.** Compartir la actividad física con sus círculos en Google+ **Descripción en la librería.** Permitir abstraer grupos de usuarios en la red social Google+ **Observaciones.**	
Número : 1	**Riesgo :**	**Red Social: G+**

Tabla 3.12 Deporte - Compartir.
Elaborado por: Tex Albuja

3.2. ELABORACIÓN DEL PROTOTIPO.

Durante la elaboración del prototipo se bosquejó la aplicación siguiendo el diseño de capas arquitectónicas definidas en el cuaderno de arquitectura.

Se creó las interfaces de usuario necesarias para soportar los casos de uso, para cubrir la capa GUI. Además de la capa de la lógica Custom Library la cual es responsable de invocar a las funciones expuestas en la librería SocialSphere.

3.3. PRUEBAS DEL PROTOTIPO.

El objetivo de esta actividad es lograr una comprensión compartida de las condiciones específicas que la solución debe satisfacer.

La metodología OpenUp propone los siguientes pasos para definir las pruebas que serán validadas.

- **Revise los requisitos para ser probados**
- **Identificar casos de prueba pertinentes**
- **Describir los casos de prueba**
- **Identificar las necesidades de datos de prueba**
- **Compartir y evaluar los casos de prueba**

El desarrollo de estos pasos deriva en la documentación de Casos de Prueba.

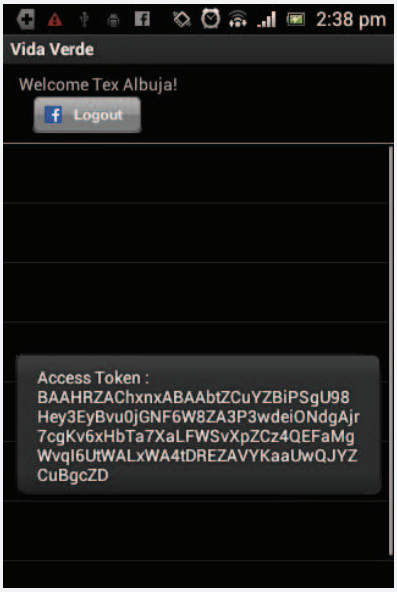
SocialSphere

Casos de Prueba.

1. ProvideAPIKey.

Ssp01. Obtener Access Token.

Obtener Access Token.



Descripción.

Verificar que se ha obtenido exitosamente un Access Token una vez que la aplicación se ha autenticado exitosamente.

Pre-condiciones.

La aplicación ha sido registrada en la red social exitosamente.

Post-condiciones.

La aplicación ha sido registrada en la red social exitosamente.

Datos requeridos.

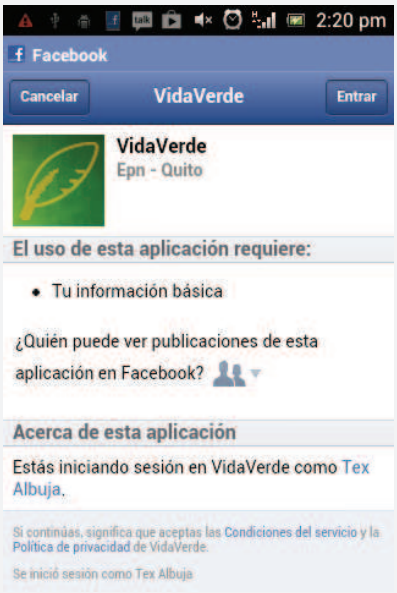
La aplicación ha sido registrada en la red social exitosamente.

Resultado : Satisfactorio	**Código:**	**Red Social : Fb.**

2. RegisterUser.

Ssp02. Registrar Usuario.

Registrar el usuario.



Descripción.

Verificar que al momento de iniciar la aplicación y el usuario no ha sido registrado se solicite acceso a los permisos definidos por la aplicación.

Pre-condiciones.

La aplicación ha sido registrada en la red social exitosamente.

La aplicación ha provisto su API Key a la librería.

La aplicación ha provisto su Key Hash a la red social.

Post-condiciones.

El usuario ha sido registrado en la aplicación de la red social.

Datos requeridos.

API Key, Key Hash.

Resultado : Satisfactorio	**Código:**	**Red Social : Fb.**

Ssp03. Obtener el usuario Actual.

Registrar el usuario.		
	Descripción. Verificar que al momento de iniciar la aplicación y el usuario no ha sido registrado se solicite acceso a los permisos definidos por la aplicación. **Pre-condiciones.** La aplicación ha sido registrada en la red social exitosamente. La aplicación ha provisto su API Key a la librería. La aplicación ha provisto su Key Hash a la red social. **Post-condiciones.** El usuario ha sido registrado en la aplicación de la red social. **Datos requeridos.** API Key, Key Hash.	
Resultado : Satisfactorio	**Código:**	**Red Social : Tw.**

3. GetCurrentUser.

Ssp04. Obtener el usuario Actual.

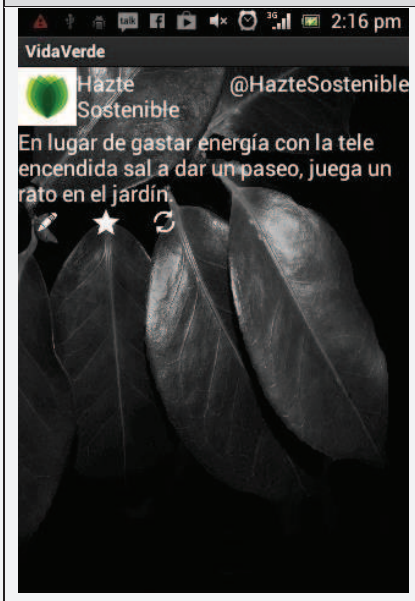
Obtener el usuario Actual.		
	Descripción. Verificar que se ha obtenido exitosamente un Access Token una vez que la aplicación se ha autenticado exitosamente. **Pre-condiciones.** La aplicación ha sido registrada en la red social exitosamente. **Post-condiciones.** La aplicación ha sido registrada en la red social exitosamente. **Datos requeridos.** La aplicación ha sido registrada en la red social exitosamente.	
Resultado : Satisfactorio	**Código:**	**Red Social : Fb.**

4. GetContent.

Ssp05. Obtener los datos del usuario.

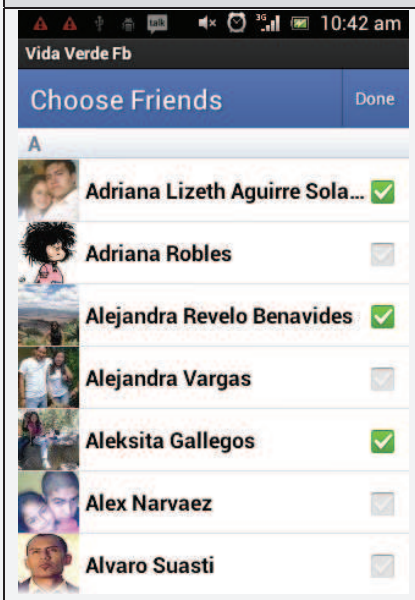
Obtener los datos del usuario.		
	Descripción. Verificar que se ha obtenido exitosamente un Access Token una vez que la aplicación se ha autenticado exitosamente. **Pre-condiciones.** La aplicación ha sido registrada en la red social exitosamente. **Post-condiciones.** La aplicación ha sido registrada en la red social exitosamente. **Datos requeridos.** La aplicación ha sido registrada en la red social exitosamente.	
Resultado : Satisfactorio	**Código:**	**Red Social : Fb.**

Ssp06. Obtener los tweets de un usuario.

Obtener los tweets de un usuario.		
	**Descripción.** Verificar que se ha obtenido el contenido de un tweet dado el usuario. **Pre-condiciones.** La aplicación ha registrado el nombre del usuario del que quiere obtener sus tweets. **Post-condiciones.** La librería despliega el nombre del usuario, su foto de perfil y sus tweets. **Datos requeridos.** Nombre de usuario de la red social.	
Resultado : Satisfactorio	**Código:**	**Red Social : Tw.**

5. GetRegisteredUsers.

Ssp07. Obtener los usuarios registrados.

Obtener los usuarios registrados.		
	**Descripción.** Obtener una colección de usuarios de la red social **Pre-condiciones.** El usuario ha sido autorizado exitosamente en la aplicación. **Post-condiciones.** Se almacena una lista de usuarios seleccionados por el usuario. **Datos requeridos.** Acces_token.	
Resultado : Satisfactorio	**Código:**	**Red Social : Fb.**

6. ShareContent.

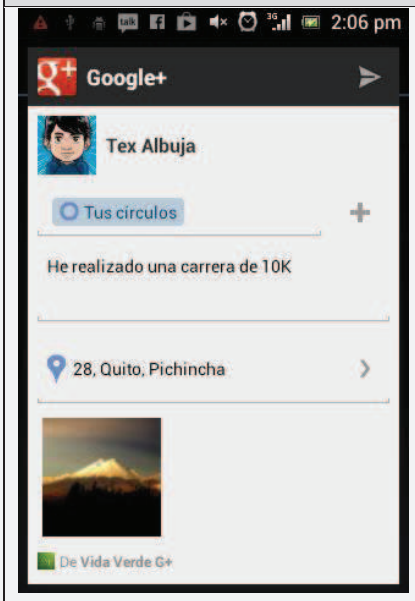
Ssp08. Compartir del contenido en el muro.

Compartir el contenido en el muro (Facebook).		
	Descripción. Verificar que se pueda compartir contenido en el muro del usuario registrado. **Pre-condiciones.** La aplicación ha provisto el contenido del post a la librería. **Post-condiciones.** El contenido del post ha sido publicado exitosamente. **Datos requeridos.** Access Token. Permisos de publicación.	
Resultado : Satisfactorio	**Código:**	**Red Social : Fb.**

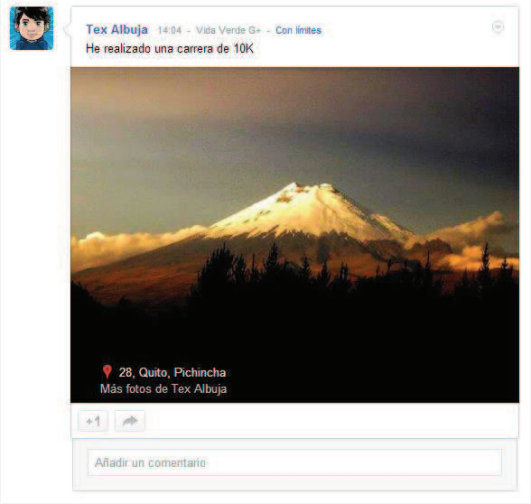
Ssp09. Visualizar el contenido de un Post en el Muro.

Visualizar el contenido de un Post en el muro (Facebook).		
	Descripción. Verificar que desde la aplicación web, se pueda visualizar el contenido del post en el muro. **Pre-condiciones.** El usuario ha compartido contenido desde su aplicación móvil. **Post-condiciones.** El usuario puede visualizar el contenido del post desde el browser de su aplicación. **Datos requeridos.** Access Token	
Resultado : Satisfactorio	**Código:**	**Red Social : Fb.**

Ssp10. Compartir del contenido en el muro.

Compartir el contenido en el muro (GooglePlus).		
	**Descripción.** Verificar que se dispare la pantalla para publicar en la red social. **Pre-condiciones.** La aplicación ha sido registrada y autorizada exitosamente en la red social, el usuario se encuentra autenticado en la aplicación **Post-condiciones.** La aplicación ha compartido en el muro del usuario satisfactoriamente. **Datos requeridos.** La aplicación ha sido registrada en la red social exitosamente.	
Resultado : Satisfactorio	**Código:**	**Red Social : Fb.**

Ssp11. Visualizar el contenido de un Post en el Muro.

Visualizar el contenido de un Post en el muro (GooglePlus).		
	**Descripción.** Verificar que desde la aplicación web, se pueda visualizar el contenido del post en el muro. **Pre-condiciones.** El usuario ha compartido contenido desde su aplicación móvil. **Post-condiciones.** El usuario puede visualizar el contenido del post desde el browser de su aplicación. **Datos requeridos.**	
Resultado : Satisfactorio	**Código:**	**Red Social : Fb.**

7. SendAppRequest.

Ssp12. Visualizar la petición de la aplicación.

Visualizar la petición de la aplicación.		
	Descripción. Verificar que se dispara la pantalla de envío de solicitudes. **Pre-condiciones.** La aplicación ha sido registrada en la red social exitosamente. **Post-condiciones.** La aplicación envía exitosamente una solicitud. **Datos requeridos.** La aplicación requiere que se haya autenticado el usuario previamente.	
Resultado : Satisfactorio	**Código:**	**Red Social : Fb.**

3.4. ANÁLISIS DE RESULTADOS.

Una vez validada la solución desde el punto de vista funcional, se realizará un estudio de los objetivos planteados en la sección de planteamiento del problema de este documento los cuales justificarían su uso.

Se medirán a través de la comparación entre los diseños corporativos y sociales de aplicaciones presentados los mismos que servirán de referencia para obtener un punto de vista objetivo.

Los objetivos que la librería cumplirá se describen a continuación.

3.4.1. MINIMIZAR EL COSTO Y TIEMPO DE DESARROLLO DE APLICACIONES MÓVILES.

Se abordará esta sección mediante la herramienta de medición de código Metrics, ha sido elegida ya que es especializada en el IDE Eclipse, y puntualmente en el lenguaje de programación Java.

- **Diseño corporativo de aplicaciones móviles.**
  Android Application (369 líneas de código).
  Web-Services. (519 líneas de código).
  Scripts de Base de Datos (400 líneas de código).
- **Diseño social de aplicaciones móviles.**
  Android Application (195 líneas de código).

Los resultados obtenidos se los puede mostrar en la figura 3.1.

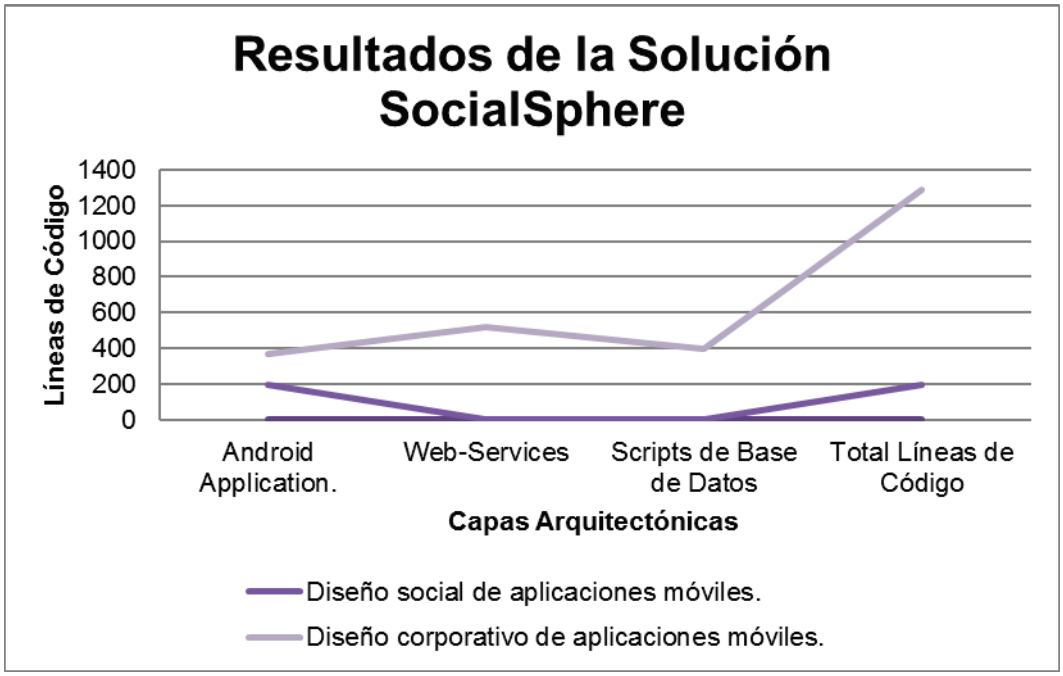


Figura 3.1 Resultados de la Solución SocialSphere.

Elaborado por: Tex Albuja

Se puede evidenciar que una vez desarrollada la solución, la implementación en una aplicación móvil requiere menor número de líneas de código.

3.4.2. REDUCIR COSTOS DE INFRAESTRUCTURA TI RELACIONADOS CON LA IMPLANTACIÓN Y MANTENIMIENTO DE SERVIDORES, Y POR ENDE REDUCIR LA CONTAMINACIÓN DEL MEDIO AMBIENTE.

Los costos de infraestructura pueden ser medidos a través de los requerimientos que presenta la aplicación. Estos costos pueden ser establecidos mediante la contratación de estos recursos como un servicio.

- **Diseño corporativo de aplicaciones móviles.**

  Se ha calculado el costo de mantenimiento de middleware que soporte la comunicación con aplicaciones móviles alojados en amazon web services con el propósito de abstraer todos los gastos en los que se podría derivar.

Amazon EC2 Service (US-East)	
Compute:	$ 380,64
Regional Transfer:	$ 0,20
EBS Volumes:	$ 150,00
EBS IOPS:	$ -
EBS Snapshots:	$ 206,25
Reserved Instances (One-time Fee):	$ 500,00
Elastic IPs:	$ 14,64
Elastic LBs:	$ -
Data Processed by Elastic LBs:	$ -
AWS Data Transfer In	$ -
US East & US Standard Region (Virginia)	$ -
AWS Data Transfer Out	$ 5,88
US East & US Standard Region (Virginia)	$ 5,88
AWS Support (Business)	$ 125,28
Support for all AWS services:	$ 75,28
Support for Reserved Instances (One-time Fee):	$ 50,00
Free Tier Discount:	$ (4,81)
Total One-Time Payment:	$ 550,00
Total Monthly Payment:	**$ 828,09**

Tabla 3.13 Costo de Infraestructura TI.
(Amazon Web Services, 2012)

- **Diseño social de aplicaciones móviles.**

  Infraestructura.

  Este costo ha sido suprimido por la librería al apoyarse en la infraestructura presente en las redes sociales.

3.4.3. REQUERIR MENOR CAPACITACIÓN PARA EL DESARROLLO DE APLICACIONES MÓVILES.

La capacitación para el desarrollo de aplicaciones móviles se concentra en la plataforma Android SDK, y el conocimiento de la librería SocialSphere, mientras que en el diseño de aplicaciones móviles corporativas es necesario tener conocimientos de otras tecnologías como bases de datos, capas de persistencia, sesión, autorización entre otras que requieren un conocimiento amplio de tecnologías. Estos conceptos son abstraídos por las API's de las redes sociales, por lo que no es necesario asimilarlos, el programador no requiere explorar estas áreas de difícil abordaje.

3.4.4. BRINDAR MÁS SERVICIOS Y CARACTERÍSTICAS A LAS APLICACIONES MÓVILES.

Los servicios que puede brindar una aplicación que utilice redes sociales serán similares a los servicios que brindan las API's de los proveedores. En general se presentan servicios de geo localización, fotos, difusión de canales, trending topics, seguidores entre otros lo que eventualmente hacen atractiva una aplicación móvil.

CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES.

Como parte del desarrollo de la librería SocialSphere, y la aplicación VidaVerde, se desprenderán las conclusiones y recomendaciones que expongo en el siguiente apartado.

4.1. CONCLUSIONES.

- El diseño de la API de integración de servicios de redes sociales y el desarrollo de la biblioteca SocialSphere como capa de abstracción de diferentes tecnologías de comunicación, ha permitido implementar de manera efectiva su uso a través de la aplicación móvil Vida Verde.
- La aplicación móvil Vida Verde ha permitido puntualizar objetivos planteados tales como: integración de servicios, abstracción de tecnologías de comunicación.
- La plataforma Android debido a su filosofía de código abierto, es un excelente material para realizar tanto desarrollos sobre sus plataformas, como librerías que aporten la funcionalidad tanto para el usuario, como para los desarrolladores.
- La metodología OpenUP es una metodología que debido a su orientación al afinamiento de la arquitectura, es un componente que facilita la captura de requerimientos técnicos en el desarrollo de soluciones que no contemplan requerimientos del negocio.
- El desarrollo de librerías tiene como objetivo minimizar el tiempo de desarrollo de aplicaciones de negocio, haciendo transparente al desarrollador distintos mecanismos de propósito general.
- El consumo de API's de arquitecturas orientadas a servicios, admiten el soporte desde diferentes plataformas (ThinClient, SmartClient, Móviles) dando como resultado aplicaciones con mejor experiencia de usuario y prestaciones.
- Los niveles de granularidad presentados en el artefacto Plan de Proyecto de OpenUP, colaboran a la estimación de tiempos y costos que un proyecto puede tomar.

4.2. RECOMENDACIONES.

- La librería SocialSphere puede evolucionar tanto en funcionalidad para cada red social, así como extender el soporte a varias redes sociales. En este sentido, los componentes agregados permitirían tener un producto robusto y más útil. Las redes que se podrían incluir podrían ser las siguientes:
    - Linked In.
    - Picassa.
    - Instagram.
    - Flickr.
- Con el fin de determinar si el diseño de una biblioteca de clases de propósito general es el apropiado, el desarrollo de una aplicación de estudio permite direccionar las clases y funcionalidades. Adicionalmente el impacto asociado de los eventuales cambios en la definición de la biblioteca de clases es menor.
- Las herramientas disponibles para control de versiones es una práctica que promueve la colaboración de miembros del equipo y el control de cambios.
- La publicación en repositorios de dominio público de este tipo de librerías apoyan tanto la colaboración con proyectos que necesitan las funcionalidades presentadas, así como el mantenimiento y la evolución del proyecto como tal, enriqueciendo su funcionalidad y su difusión.

BIBLIOGRAFÍA

Amazon Web Services. (2012, Noviembre 20). *Amazon*. Retrieved from Simple Monthly Calculator: http://calculator.s3.amazonaws.com

Android Developers. (2012, September 4). *Android Developers*. Retrieved from Platform Versions - Current Distribution: http://developer.android.com/about/dashboards/index.html

Eclipse Foundation. (2012, Junio 1). *Introduction to OpenUP*. Retrieved from OpenUp Web Site: http://epf.eclipse.org/wikis/openup

Hueniverse. (2011, July 15). *OAuth*. Retrieved from The OAuth 1.0 Guide: http://hueniverse.com/oauth/guide/

Instituto Nacional de Estadística y Censos. (2011). *REPORTE ANUAL DE ESTADISTICAS SOBRE TECNOLOGÍAS DE LA INFORMACIÓN.* Quito: INEC.

ProgrammableWeb. (2012, September 12). *ProgrammableWeb API Directory*. Retrieved from ProgrammableWeb : http://www.programmableweb.com/api

SalesForce Developers. (2012, Octubre 30). *Flujo de autenticación de OAuth 1.0.A*. Retrieved from SalesForce: http://help.salesforce.com/apex/HTViewHelpDoc?id=remoteaccess_oauth_1_flows.htm&language=es

Satya Komatineni, D. M. (2011). *Pro Android 3.* New York: Apress.

Twitter. (2012). *Twitter Developers*. Retrieved from Twitter Dev: https://dev.twitter.com/

GLOSARIO.

A

Android

Sistema operativo diseñado para dispositivos portátiles basado en GNU/LINUX., 7, 8, 9, 11, 1, 4, 5, 6, 7, 8, 9, 26, 27, 28, 30, 36, 39, 41, 42, 45, 51, 52, 54, 57, 62, 63, 65, 69, 70, 71, 73, 79, 80, 81, 1, 2

Atom

Formato de redifusión para sindicar contenidos. Suele ser utilizado por la comunidad de blogs para compartir las entradas recientes, titulares, textos completos., 20

B

Bluetooth

Bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPANs) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz., 8

C

casos de uso

Técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software., 10, 15, 50, 51, 55, 56, 71, 72

D

Dalvik

Máquina virtual que utiliza la plataforma Java para dispositivos móviles Android., 8

F

FAQs

Recopilación de respuestas a las dudas más frecuentes de los usuarios de cualquier servicio de Internet., 30

fases

Estructura del ciclo de vida de un proyecto., 11, 33, 44

I

Instagram

Instagram es una aplicación gratuita para compartir fotos con la que los usuarios pueden aplicar efectos fotográficos como filtros, marcos y colores retro y vintage y compartir las fotografías en diferentes redes sociales como Facebook, Twitter, Tumblr y Flickr., 1

J

javadoc

Estándar de la industria para documentar clases escritas en Java. La mayoría de los IDEs los generan automáticamente., 30

K

Kernel

Núcleo del sistema operativo. Es el que se encarga de las labores de más bajo nivel (el nivel más cercano al hardware) tales como gestión de memoria, de entrada/salida de dispositivos, etc., 7

M

microblogging

Servicio que permite a sus usuarios enviar y publicar mensajes breves (alrededor de 140 caracteres)., 19

O

OAuth

Protocolo abierto, propuesto por Blaine Cook y Chris Messina, que permite autorización segura de un API de modo estándar y simple para aplicaciones de escritorio, móviles, y web., 2, 23, 1

OpenGL

Especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D., 8

P

plugin

Un plugin es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy especifica., 27

R

REST

(Representational State Transfer) o REST es una técnica de arquitectura software para sistemas hipermedia distribuidos., 18

RSS

Es una familia de formatos para documentos utilizados para publicar contenidos que se actualizan frecuentemente en el Internet, por ejemplo cambios recientes en el wikcionario, entradas en un blog o encabezados de noticias., 19

S

SmartClient

Aplicación de Escritorio que ofrece gran capacidad de procesamiento., 2, 81

T

ThinClient

Aplicación entregada sobre internet que posee limitadas características de procesamiento., 2

ANEXOS.

Los anexos que se enumeran a continuación se encuentran en el disco compacto que asiste el presente documento.

Anexo 1. Elaboración del artefacto Visión.
Anexo 2. Elaboración del artefacto Diario de Arquitectura.
Anexo 3. Elaboración del artefacto Elaboración del Plan de Proyecto.
Anexo 4. Elaboración del artefacto Elaboración del Plan de Iteración.
Anexo 5. Elaboración del artefacto Preparación de Ambiente.
Anexo 6. Elaboración del artefacto Planes de Iteración.
Anexo 7. Elaboración del artefacto Modelo de Casos de Uso.
Anexo 8. Elaboración del artefacto Desarrollo del Incremento de la Solución.
Anexo 9. Elaboración del artefacto Especificación de Casos de Uso.
Anexo 10. Elaboración del artefacto Gestión de Riesgos.
Anexo 11. Elaboración del artefacto Glosario.