



UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS

FACULTAD DE INGENIERÍA

PROYECTO CURRICULAR DE INGENIERÍA DE SISTEMAS

Anteproyecto:

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE SNS ORIENTADO AL DEPORTE SOBRE TECNOLOGÍAS MÓVILES

Presentado por:

Nicolás Mauricio Garcia Garzon 20091020031

Luis Felipe Gonzalez Moreno 20091020035

Dirigida por:

Doctor Carlos Enrique Montenegro Marin

Índice general

Lista de figuras	7
Lista de tablas	9
Introducción	11
Glosario	13
1. Definición del problema	15
2. Justificación del problema	19
3. Objetivos	23
3.1. Objetivo General	23
3.2. Objetivos específicos	23
4. Marco conceptual	25
4.1. Comunicación	25
4.1.1. Evolución de la web	26
4.2. Réd social	26
4.3. Business Process Modeling Notation (BPMN)	28
4.3.1. Pertinencia con el proyecto	29
4.4. Service-Oriented Architecture (SOA)	29
4.4.1. Primeros conceptos	29

4.4.2.	Computación orientada a servicios	30
4.4.3.	Pertinencia con el proyecto	35
4.5.	Scrum	35
5.	Marco teórico	37
5.1.	Redes Sociales	37
5.1.1.	SNA: Social Network Analysis	37
5.1.2.	Análisis egocéntrico	37
5.1.3.	Grado de centralidad	38
5.1.4.	Grado de cercanía	40
5.1.5.	Factor distancia en la formación de las redes sociales	40
5.1.6.	Grado de intermediación	41
5.1.7.	Díadas	42
5.1.8.	Tríadas	43
5.1.9.	Análisis Triádico	45
5.2.	Business Process Modeling Notation (BPMN)	45
5.2.1.	Enlaces	45
5.2.2.	Colaboración	48
5.3.	Service-Oriented Architecture (SOA)	50
5.3.1.	Proceso de Análisis y diseño orientados a servicios	50
5.3.2.	Metas y beneficios de la computación orientada a servicios	51
5.3.3.	Principios de diseño SOA	52
5.4.	Scrum	53
5.4.1.	Equipo Scrum (Scrum Team)	53
5.4.1.1.	Product Owner (dueño del producto)	53
5.4.1.2.	Development Team (Equipo de desarrollo)	54
5.4.1.3.	Scrum Master	54

<i>ÍNDICE GENERAL</i>	5
5.4.2. Eventos	56
5.4.2.1. Sprint	56
5.4.2.2. Sprint Planning Meeting (Reunión de Planificación de Sprint)	56
5.4.2.3. Daily Scrum (Scrum Diario)	56
5.4.2.4. Sprint Review (Revisión de Sprint)	56
5.4.2.5. Sprint Retrospective (Retrospectiva de Sprint)	57
5.4.3. Artefactos	57
5.4.3.1. Sprint Goal (Objetivo del Sprint)	57
5.4.3.2. Product Backlog (Lista de Producto)	57
5.4.3.3. Sprint Backlog (Lista de Pendientes del Sprint)	57
5.5. Requerimientos no funcionales	58
5.5.1. Mantenibilidad	58
5.5.2. Reusabilidad	63
5.5.2.1. Medidas de reusabilidad	63
5.5.2.2. Blueprints y reusabilidad	63
5.5.2.3. Logic Centralization: Patrón de diseño pro-reusabilidad	64
5.5.2.4. Granularidad y reusabilidad	64
5.5.3. QiU: Quality in Use	65
5.5.3.1. Usabilidad	65
5.5.3.2. User acceptance (Aceptación de usuario)	65
5.5.3.3. Experiencia de usuario (User eXperience - UX)	66
5.5.3.4. Conceptos para medir QiU	67
5.5.4. Seguridad	68
5.6. Archimate 2.0	70
5.6.1. Capas de archimate	70
5.6.1.1. Capa de negocio	70

5.6.1.2.	Capa de aplicación	70
5.6.1.3.	Capa de infraestructura	72
5.6.2.	Relaciones	72
5.6.3.	Vistas archimate	75
5.7.	Estado del arte	75
6.	Marco legal	85
7.	Alcances y limitaciones	89
7.1.	Alcances y limitaciones	89
7.1.1.	Alcances	89
7.1.2.	Limitaciones	90
8.	Metodología	93
9.	Propuesta de arquitectura	97
10.	Cronograma	99
11.	Costos	101
11.1.	Recursos de Hardware	101
11.2.	Recursos de Software	101
11.3.	Recursos humanos	101
11.4.	Recursos misceláneos	101
11.5.	Costos totales	104

Índice de figuras

1.	Cambio de la utilización de internet en función de los servicios de conectividad social que son creados y soportados en ella . .	27
2.	La red mas simple.	27
3.	Ejemplo de una red asimétrica.	28
4.	Acercamiento a cómo se desarrollan los principios de diseño con los demás conceptos nombrados.	31
5.	Como extiende o soporta un patrón de diseño el diseño lógico de la solución de software.	32
6.	Componentes que hacen que el diseño lógico de la solución sea acorde al paradigma escogido.	33
7.	diferenciación entre tipos de servicio da lugar a la estructura en capas	33
8.	Interacción de los conceptos clave en la computación orientada a servicios	34
9.	Red de estrella.	39
10.	red corbatín.	42
11.	Tipos de diadas asimétricas.	43
12.	Tipos de triadas simétricas.	43
13.	Tipos de triadas asimétricas.	44
14.	Red social de Krackhardt kite.	46
15.	Ejemplo de uso del enlace exclusivo	46

16.	Ejemplo de uso del enlace paralelo	47
17.	Ejemplo de uso del enlace inclusivo	47
18.	Ejemplo de uso del enlace complejo	48
19.	Ejemplo de diagrama de colaboración	49
20.	Ejemplo de diagrama de colaboración con caja negra	49
21.	proceso de análisis y diseño orientado a servicios	50
22.	Cruce entre las subcaracterísticas atribuidas a la mantenibilidad del software en el estandar ISO/IEC 9126 y las propiedades a nivel de código. Cada cruce es indicado por una x	59
23.	correspondiente al nivel de riesgo en el software presentado por la complejidad ciclomática promedio del sistema	60
24.	correspondiente al nivel máximo relativo de LOC que podrían ser riesgosas al momento de mantener el software	61
25.	clasificación de duplicación de código	62
26.	Propuesta de arquitectura	97
27.	Distribución de las diferentes actividades a realizar	100

Índice de cuadros

1.	Artefactos de la capa de negocio	71
2.	Artefactos de la capa de aplicación	72
3.	Artefactos de la capa de infraestructura	73
4.	Relaciones estructurales	73
5.	Relaciones dinámicas	74
6.	Otras relaciones	74
7.	Punto de vista introductorio	75
8.	Punto de vista en capas	76
9.	Punto de vista de función	76
10.	Punto de vista de proceso	77
11.	Punto de vista de uso de aplicación	77
12.	Punto de vista de producto	78
13.	Comparacion de redes, parte 1	79
14.	Comparacion de redes, parte 2	80
15.	Comparacion de redes, parte 3	81
16.	Comparacion de redes, parte 4	82
17.	Comparacion de redes, parte 5	83
18.	Product backlog inicial	95
19.	Actividades generales a llevar a cabo	99

20. Recursos de hardware 102

21. Recursos de software 102

22. Recursos humanos 103

23. Recursos misceláneos 103

24. Costos totales 104

Introducción

El uso de los medios informáticos para la formación de comunidades deportivas en las que los deportistas puedan formar y gestionar sus redes sociales es restringido debido al modo de vida del deportista. Es usual que por medio de facebook y twitter los deportistas creen sus redes sociales. Sin embargo, facebook y twitter añaden información basura para los deportistas y no ofrecen servicios que han de ser propios de una red social deportiva.

En este documento se expone una propuesta para el desarrollo de una red social deportiva por medio de la teoría de redes sociales, SOA, BPMN y la investigación del estado del arte de las redes sociales deportivas. Lo que se pretende con el documento es exponer el problema existente que hay entre la utilización de las TIC y la comunicación (a modo de red social) entre las comunidades deportivas. Además, también se pretende, en el presente documento, dar una solución basándose en el análisis de la información existente en cuanto al problema formulado y el armado de un proceso de ingeniería para llevarla a cabo.

Primero, el lector encontrará un acercamiento al problema que se resolverá en la definición del problema, la justificación y los objetivos. Más adelante, el lector podrá echar un vistazo a la teoría que está detrás del problema a resolver y que es necesaria para su solución. Por último, se presenta al lector el marco a utilizar para la solución del problema en las secciones de metodología, cronograma y un estudio de presupuestos.

Glosario

- OSN : Online Social Network – Red Social En-línea, es una red social que crece dentro del ámbito web.
- SNS : Social Network Services – Servicios de Redes Sociales.
- Offline Social Network : Red social que crece en el ámbito real, no en el virtual como lo es en las OSN.
- Sistemas transversales : Sistemas que interactúan entre sí, hechos o no con tecnologías diferentes sobre paradigmas diferentes, con un fin común.
- API : Application Programming Interface – Interfaz de programas de aplicación, es el conjunto de funciones y procedimientos de un sistema que pueden ser utilizados en otro sistema. Es el puente de conexión entre ambos sistemas.
- Interoperabilidad : Capacidad de un sistema de software para trabajar con otro sistema con la característica de que esta cooperación sea hecha de la manera más transparente posible.
- SOA : Service-Oriented Architecture – Arquitectura Orientada a Servicios.
- TI : Tecnologías de la Información
- ROI : Return on investment – Retorno en inversión.
- Product Backlog : Productos que están pendientes por realizar para finalizar el proyecto.

- Sprint Goal : El objetivo, que de cumplirse, determina si el Sprint fue exitoso.
- Nodo : Punto de intersección de varios elementos
- Arista : Uniones entre nodos
- Stakeholders : Partes interesadas en un tema específico.
- UX (User eXperiencie): Se refiere a la percepción que adquiere el usuario de un dispositivo (software, hardware) con respecto a las tareas realizadas con este, basándose en las sensaciones/reacciones que este le provoca.
- LOC : En ingles Lines Of Code, se refiere a las líneas de código en una unidad (una función, un método, un servicio o cualquiera que sea la unidad según el paradigma usado).
- Stakeholder: Persona a quien compete o afecta un asunto en particular (proceso, proyecto, etc.).

Capítulo 1

Definición del problema

El hombre, en su estado de ser social, ha creado diversos medios de comunicación. Pasó del lenguaje a aparatos que constituyen un canal para difundir dicho lenguaje, tales como el teléfono, la radio, la televisión, la creación del internet y el computador personal y, en última instancia, la creación de los dispositivos móviles inteligentes (smartphones y tablets) . Hoy, gracias a la evolución de la web 1.0 estática en contenido a una web 2.0 dinámica en cuanto a interacción social, las redes sociales se han convertido en un medio de difusión con gran afluencia de información, de informantes y de informados. Redes sociales como facebook, twitter o youtube se han vuelto famosas para compartir **cualquier tipo de información**: Desde lo último ocurrido en la vida de un ciudadano del común hasta la difusión de los problemas por los que pasa el mundo en general. La gran cantidad de información diversa y dispersa sobre un solo medio hace que haya demasiado ruido para aquellas personas que usan las redes sociales con objetivos informativos particulares, ya sea de carácter científico, pasando por carácter artístico o, inclusive, de carácter deportivo.

Uno de los problemas con los que se ha encontrado uno de los autores del presente trabajo (Nicolás García, vicepresidente de la Asociación Colombiana de Lacrosse) en la integración del internet con temas de carácter deportivo, ha sido que los deportistas son bombardeados de información (consejos deportivos, nutricionales o de preparación física; historia de los deportes y sus reglas, etc) en diversos espacios de dicha red. Este problema no es nuevo: éste se ha presentado con la masificación de la utilización de internet desde que

éste fuera creado. Como dice en (Companies, 2014), hay algunos ítems que deberá cumplir una fuente para ser confiable (autor o patrocinador, información de contacto proveída, propósitos para los que fue creado el sitio donde se despliega el recurso, verificación de la información). Reducir la búsqueda de los deportistas a un solo lugar confiable (que de, de alguna manera, el soporte a las pautas sugeridas en (Companies, 2014)) con información deportiva desembocaría en la confiabilidad de la información que se encuentra allí y convertiría a dicho lugar en herramienta más corriente para búsquedas en temas de deporte.

Otro problema de la integración de internet y temas de carácter deportivo es la falta de utilización, de parte de los deportistas, de las redes sociales dedicadas al deporte. De nuevo, debido a la experiencia del vicepresidente de la Asociación Colombiana de Lacrosse, se ha visto que los deportistas utilizan medios como twitter, facebook o couchsurfing (que añaden ruido a la información que ellos necesitan) para difundir información deportiva sin pensar en las bondades que puede ofrecer un sitio dedicado a deportistas. Las estrategias que ha desplegado la Asociación (no solo en cabeza de él, sino que en conjunto con otras personas de las ciudades de Medellín, Tunja, Pereira y Bogotá) han sido sobre las redes sociales antes mencionadas y, además, sobre la red social Instagram. Un SNS deportivo tendrá como prioridad la gestión de una red social fuera de línea de carácter deportivo y, por tanto, debe proveer mecanismos que ayuden a facilitar tareas que se hagan en el mundo real (por ejemplo, búsqueda de personas interesadas en la participación de un nuevo deporte como lo es el lacrosse en Colombia).

Un fenómeno particular que se dio en el crecimiento de la Asociación Colombiana de Lacrosse fue la utilización de las redes sociales más conocidas como medio de comunicación y difusión del deporte, presentándose un problema: Las personas usuario de esas redes no buscaban por deportes nuevos, sino que tenían sus propios propósitos que no siempre se acoplaban a los deportivos. Aún con el esfuerzo de difusión por poco más de 2 años (la Asociación funciona desde el 15 de Julio del año 2012), el deporte aún es desconocido, teniendo personas (en su mayoría) extranjeras asociadas a cada red social (nuevamente, Facebook, Twitter e Instagram, e incluso Youtube o Google+).

Los problemas presentados aquí pueden ser resueltos por medio de la implementación de SNS. Ya que el componente Geolocalización en Tiempo

Real es importante debido a que la ubicación de los deportistas es de gran ayuda a otros deportistas para salir a interactuar con ellos, así como el hecho de tener a mano la información deseada en cualquier momento que se requiera (por ejemplo, consejos para tratar una lesión al instante), se puede abordar ambos problemas con la utilización de tecnologías móviles.

Así, con la evolución de la comunicación humana trasladándose a los espacios virtuales por medio de las redes sociales y la falta de aplicaciones, en el campo de los smartphones, que soporten interacciones sociales enfocadas a los deportes en general, en este trabajo se creará un servicio de red social centrado en los deportes sobre tecnologías móviles para la administración de las redes sociales de cada persona en un ámbito deportivo desde su dispositivo móvil.

Haciendo uso de las TIC, ¿cómo se puede facilitar la comunicación y el acceso a la información a quienes son parte de la comunidad deportiva?

Capítulo 2

Justificación del problema

Los humanos, desde siempre en su evolución, han necesitado de mecanismos para comunicarse con sus congéneres. En la actualidad, uno de los mecanismos es el uso de los SNS como facebook y twitter, cada uno de ellos modificando la forma de creación de redes sociales en la actualidad. (Sección 5.1)

De acuerdo al análisis egocéntrico de las redes sociales de cada individuo, se hace conveniente la utilización de SNS para gestionar las relaciones que un individuo mantiene con otros individuos (sean personas u organizaciones) en los diferentes círculos sociales en los que se mueve. (Sección 5.1.2)

El círculo social o comunidad escogida para el desarrollo propuesto es la comunidad deportiva debido a que hay mucha información dispersa alrededor de internet que es ambigua y a veces inclusive errónea. A su vez, debido a que gran cantidad de deportes no han tenido una acogida grande alrededor del mundo, las comunidades que se mueven sobre uno de esos deportes son más cerradas y, por ende, pequeñas y con poca información para un público que salga de las fronteras de dichas comunidades cerradas. Lo que se quiere con este trabajo es aportar al crecimiento de las redes sociales de las personas que practiquen deporte sin importar si lo hacen a nivel profesional o aficionado por medio de un SNS orientado a los deportes en general.

Un factor de utilización masiva de las SNS es que éstas estén orientadas a un público en particular y aumenten su cobertura dependiendo de su alcance de masa crítica sobre una red social definida (Tsvetovat y Kouznetsov, 2011). Al construir, en principio, la red social deportiva enfocada en dos deportes

en particular, la probabilidad de ganar la masa crítica es mayor y, por tanto, el SNS desarrollado puede volverse más útil con el tiempo.

La UX de los SNS (visto en la sección ??) es otro factor, debido a que juega un papel importante pues es esta la segunda carta de presentación de un SNS. Algunas de las características que evalúan los usuarios en cuanto a la UX no son suplidas por los SNS actuales – o al menos no parcialmente –, tres de ellas (fundamentales para la acogida de un nuevo SNS) son “curiosity, learning y completeness of the social network”. Así, habiendo analizado 19 SNS orientados al deporte (Tablas 13 a 17), se concluyó que fallaban en alguna de las tres características mencionadas.

Tener en cuenta la población a quien va dirigido el SNS a desarrollar es otro factor de éxito. Según (Elahi y Karlsen, 2012), entre los años de adolescencia y los 40 años de edad, las personas acuden con mayor interés al uso de los SNS; al ser la comunidad del deporte comprendida en su mayoría por personas entre la adolescencia y los 40 años, aumenta aún más la probabilidad de alcanzar la masa crítica y volver útil con el tiempo el SNS.

Un último factor, que se observó, afecta la creación de redes sociales (tanto fuera de línea como en línea) (Sección 5.1) es la distancia entre cada individual y el posible tipo de enlace que los uniría. Al ver la importancia de manejar SNS que ofrezcan servicios de geolocalización, se ha visto pertinente añadir dicho servicio a la creación del prototipo de SNS orientado a los deportes en general.

También se encontró evidencia de poca utilización de los SNS que no estaban orientadas a móviles. Dichos SNS eran utilizados mucho más por personas que practican deportes que empiezan a tomar vuelo o deportes poco conocidos (un ejemplo de ello es el padel). El problema con dichos SNS es la naturaleza nómada de los deportistas. Una solución a la naturaleza nómada de los deportistas y el acercamiento de los últimos a las TICs y, en este caso, a los SNS deportivos es la aparición y utilización en masa de los smartphones.

En general, solo se encontraron dos redes sociales deportivas orientadas a cualquier deporte asociadas a aplicaciones para smartphones disponibles en el la tienda virtual de Android o en la tienda virtual de Apple (La red social de Fitivity y Huddlers) (Tablas 13 a 17). Además, hay una ventaja real en hacer una red social orientada a dispositivos móviles y es la capacidad de movilidad que ellos brindan mientras se está utilizando el servicio (Sapuppo, 2010). Dada la falta de aplicaciones móviles en el campo descrito y a su vez

la importancia que toman los dispositivos móviles por sus características, se ha decidido hacer el prototipo de SNS orientado al deporte sobre tecnologías móviles.

Capítulo 3

Objetivos

3.1. Objetivo General

Desarrollar un prototipo SNS (Social Networking Service) centrado en el deporte, sobre tecnologías Android, que permita al usuario el acceso a diferentes servicios propios de una red social con el fin de fortalecer a las comunidades que practican deporte, y a quienes quieren ser parte de ellas, facilitando la comunicación y el acceso a la información a los usuarios de la red.

3.2. Objetivos específicos

- Investigar acerca de la teoría de las redes sociales, la computación orientada a servicios y el estado del arte de las redes sociales, con el fin de que el prototipo a desarrollar esté cimentado sobre bases teóricas sólidas.
- Identificar, analizar y evaluar las necesidades de la comunidad deportiva concernientes al SNS a implementar y, posteriormente, postular requerimientos funcionales y no funcionales para el SNS.
- Formular una arquitectura híbrida bajo el paradigma SOA que cumpla los diferentes requerimientos del proyecto, con el fin de tener un marco de trabajo que soporte cada etapa posterior a la formulación de esta.

- Postular, escoger y diseñar los servicios que darán solución a los requerimientos y necesidades identificadas para la implementación de la SNS, con el fin de gestionar cada incremento del proyecto y escoger los servicios a diseñar.
- Implementar y probar los diferentes servicios con el fin de cumplir los requerimientos planteados en la fase de análisis.

Capítulo 4

Marco conceptual

A continuación, se definen algunos conceptos que intervienen con el desarrollo del presente proyecto.

4.1. Comunicación

Los científicos han estudiado el porqué de las relaciones complejas entre los humanos en comparación a la complejidad presentada en las relaciones entre otros animales. Una de las hipótesis, Social Brain Hypothesis (SBH) postula que el crecimiento cognitivo humano y sus intrincadas relaciones sociales se deben a “la necesidad de nuestros ancestros de mantener e incrementar el número de relaciones sociales con diferentes grupos para sobrevivir en las extremadamente desafiantes condiciones ambientales originadas durante la última era glacial”.(Arnaboldi, Conti, y Passarella, s.f.)

El hombre, en su continua evolución, ha utilizado el lenguaje como una herramienta creadora de conocimiento transferible a sus congéneres o cualquier otro ser que interactuase con él. Con esto, “los humanos han desarrollado el lenguaje como un instrumento ligero y conveniente para mantener sus relaciones” (Arnaboldi y cols., s.f.).

En la comunicación entre congéneres, el lenguaje puede ser dividido en dos funciones: función de transmisión de información (gossip) y función de entendimiento del estado interno (estado mental) del congénere (mentalisation) (Arnaboldi y cols., s.f.). Estas funciones de transmisión y entendimien-

to del otro han permitido que dos o varios humanos puedan asociarse entre sí formando redes sociales.

4.1.1. Evolución de la web

La evolución de los servicios proporcionados a través de internet ha sido drástica puesto que ha cambiado el modo de vida de las personas. En la figura 1 se evidencia que el crecimiento de internet (de los servicios que en ella se soportan) se da en función de los servicios de conectividad social que son creados y soportados en ella. La web 1.0 fue utilizada en mayor medida por científicos para el intercambio de información en formato hipertexto. No había una interacción fuerte entre cada científico sino que ellos acudían a internet para buscar o poner a disposición material científico. Con la venida de la web 2.0 y la introducción de la interacción del usuario con la web, generando contenido en tiempo real, fueron creados servicios de redes sociales en-línea (OSN en inglés: On-line Social Network), produciendo una partición en los tipos de redes sociales. Así, las redes sociales a las que pertenece el ser humano en la era digital se dividieron convenientemente en “redes sociales fuera de línea” y “redes sociales en línea” (Offline Social Network y Online Social Network) (Arnaboldi y cols., s.f.).

4.2. Réd social

La información contenida en la actual sección es tomada del libro *Social Network Analysis for Startups* (Tsvetovat y Kouznetsov, 2011)

Una red es un conjunto de relaciones. Mas específicamente, una red consiste en un conjunto de objetos (nodos) que están interconectados a través de relaciones (aristas). La red mas simple consiste en 2 nodos, N1 y N2, que están relacionados entre sí (Figura 2). Los nodos podrían representar personas, mientras la arista representa la relación que existe entre ellas (N1 y N2 son amigos, por ejemplo).

Las redes sociales fuera de línea son las redes sociales que se forman por comunicación tradicional (lenguaje oral y escrito en medios que difieran de aquellos que utilizan las telecomunicaciones). Las redes sociales en línea son aquellas redes sociales que están formadas por cibernautas y en las cuales la

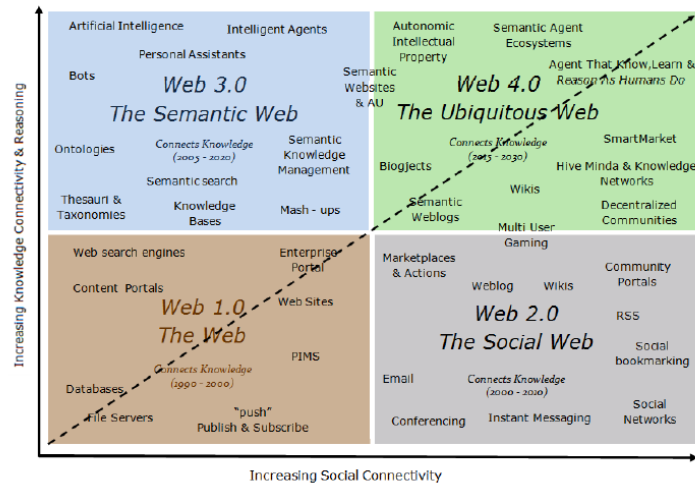


Figura 1: Cambio de la utilización de internet en función de los servicios de conectividad social que son creados y soportados en ella

(Lozada, 2014).

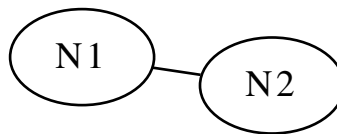


Figura 2: La red mas simple.

Fuente: Autores

comunicación se da por medio de los servicios de redes sociales. (Yang, Guo, y Ma, 2010)

Las relaciones pueden ser simétricas o asimétricas. Cuando se tiene una relación simétrica se dice que la relación no tiene dirección, es decir, la relación puede leerse en ambos sentidos. En el ejemplo anterior, significaría que N1 es amigo de N2 y que N2 es amigo de N1. Para que una relación se considere asimétrica, la relación debe poder leerse en un único sentido, es decir, la relación tiene una dirección determinada. En la figura 3 se puede

observar un ejemplo de una red asimétrica en donde el nodo (o persona) N1 sigue al nodo N2, pero el nodo N2 no sigue al nodo N1.

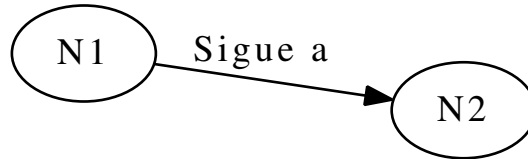


Figura 3: Ejemplo de una red asimétrica.

Fuente: Autores

Es posible que exista mas de una relación entre 2 nodos, en ese caso se dice que existe una *relación multiplex* (Kadushin, 2004, Cap.2)

4.3. Business Process Modeling Notation (BPMN)

La información contenida en la actual sección es tomada del libro *BPMN 2.0 Introduction to the Standard for Business Process Modeling* (Allweyer, 2010)

Al interior de una organización es importante documentar y especificar los diferentes procesos que se deben llevar a cabo. A menudo, se suelen utilizar diagramas de flujo o incluso descripciones textuales. Desafortunadamente, estas técnicas se quedan cortas a la hora de describir procesos mas complejos, y mas aún cuando cada organización define su propia notación, dificultando el entendimiento de los diferentes modelos que se realizan al interior de la organización. Es por esto que se hace necesario crear una notación estándar y especifica para describir este tipo de procesos.

BPMN, Business Process Modeling Notation, es precisamente el estándar que se viene adoptando a nivel masivo para la creación y descripción de los diferentes procesos que hacen parte del funcionamiento de las diferentes organizaciones.

Con el estándar BPMN, se tienen en cuenta diferentes elementos básicos que sirven como herramientas para crear y estructurar los diferentes modelos que se quieran hacer. Estos elementos solo describen una forma básica de

uso, pues BPMN permite personalizar los elementos a utilizar, siempre y cuando los cambios realizados no dificulten el proceso de entendimiento de los modelos.

4.3.1. Pertinencia con el proyecto

Se utilizará BPMN como ayuda para el modelamiento de los diferentes procesos de negocio que sean identificados en la fase de análisis. Esto facilitará el proceso de diseño e implementación de los diferentes componentes necesarios para dar solución a los diferentes requerimientos del negocio.

4.4. Service-Oriented Architecture (SOA)

La información contenida en la actual sección es tomada del libro *Soa principles of service design* (Erl, 2007)

4.4.1. Primeros conceptos

Los conceptos base del diseño de software deben ser expuestos para tener una mayor claridad en los temas siguientes. A continuación se expresan los conceptos base:

- **Características de diseño:** Son aquellos atributos que cumple un diseño y que pueden ser medidos.
- **Principio de diseño:** Es una guía o regla para solucionar un problema de acuerdo a las prácticas aceptadas por la comunidad de ingeniería de software.
- **Paradigma de diseño:** Es el compendio de principios de diseño que tienen un enfoque global común.
- **Patrones de diseño:** Son formas de resolver un problema de diseño que es repetitivo. Viene dado por 3 restricciones presentadas en el diseño de software:

- Restricciones impuestas por la tecnología existente
- Restricciones impuestas por las tecnologías usadas por sistemas transversales
- Restricciones de prioridades de proyectos

El patrón de diseño describe el problema y da la solución a modo de plantilla.

- **Lenguajes de patrones de diseño:** Es la configuración ordenada de patrones en un diseño lógico. La comunicación entre cada patrón se hace a través de dicho lenguaje.
- **Estándares de diseño:** En orden de ir acorde a las metas, prioridades, recursos y ambiente de la organización en la que se haga el diseño lógico de la solución, un estándar de diseño define convenciones para cada elemento utilizado en el diseño de acuerdo a las características de diseño definidas.
- **Buenas prácticas:** Es una técnica o acercamiento para resolver o prevenir problemas presentados en el desarrollo del diseño lógico de la solución de software. pg 34

La figura 4 presenta un acercamiento a cómo se desarrollan los principios de diseño con los demás conceptos nombrados.

La figura 5 presenta cómo extiende o soporta un patrón de diseño el diseño lógico de la solución de software.

La figura 6 presenta los componentes que hacen que el diseño lógico de la solución sea acorde al paradigma escogido.

4.4.2. Computación orientada a servicios

La computación orientada a servicios nace de la necesidad de desarrollar software integrado con otros construidos con diferentes arquitecturas y, por ende, diversas tecnologías. Este tipo de computación tiene como finalidad la construcción de inventarios de servicios. La computación orientada a servicios está compuesta por la interacción de la orientación a servicios (paradigma de diseño) y la arquitectura orientada a servicios, formando patrones de diseño

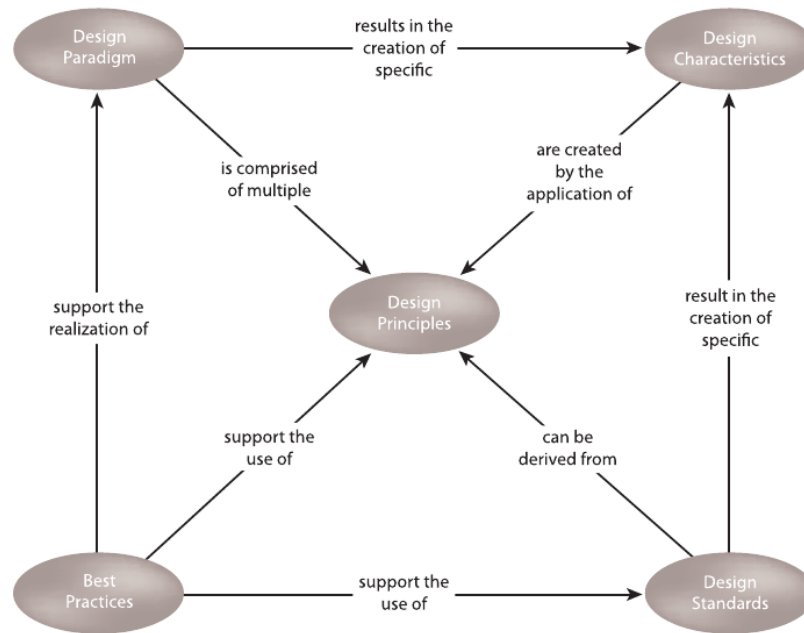


Figura 4: Acercamiento a cómo se desarrollan los principios de diseño con los demás conceptos nombrados.

Fuente: (Erl, 2007)

propios y estándares en cumplimiento de las características de diseño propias de la computación orientada a servicios.

Algunos de los conceptos clave llevados en la computación orientada a servicios son:

- Arquitectura Orientada a Servicios (SOA - Service Oriented Architecture): Comprende el compendio de tecnologías, APIs, infraestructura y repositorios enmarcados en el paradigma orientado a servicios y cuyo objetivo principal es el de trabajar sobre el "servicio" como el elemento más importante.
- Orientación a servicios: Es el paradigma manejado en la computación orientada a servicios en donde se acepta como unidad mínima y más importante el "servicio".

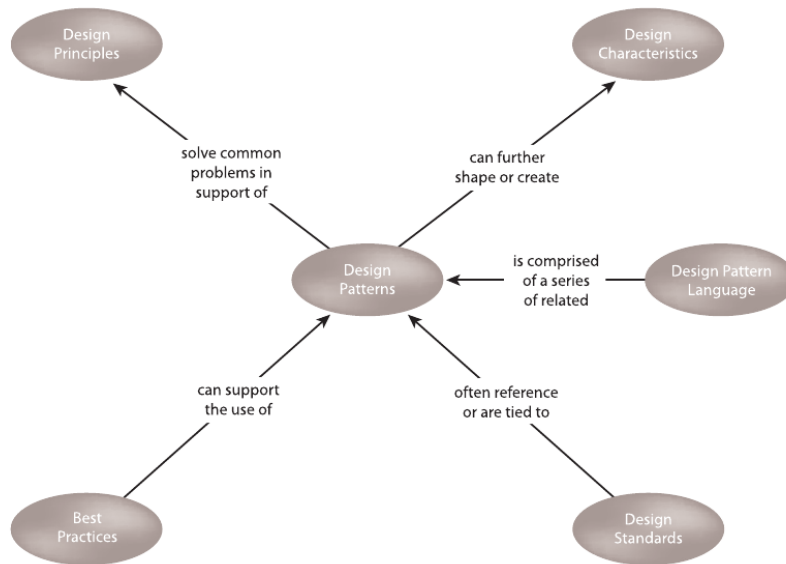


Figura 5: Como extiende o soporta un patrón de diseño el diseño lógico de la solución de software.

Fuente: (Erl, 2007)

- Servicio: Es un software independiente físicamente el cual tiene asignado un contexto de funcionalidades y que puede ser utilizado por otros servicios por medio del contrato del servicio (descripción del servicio en cuanto a funcionalidades, entradas requeridas y salidas). De acuerdo a su nivel de reúso, los servicios se dividen en 3 tipos y son:
 - Servicios entidad: Modela los servicios que se deben ofrecer respecto de las entidades del negocio (ej. empleados y clientes). Tienen un nivel de reúso alto y están centrados en el negocio.
 - Servicios tarea: Modela los servicios que deben cumplir tareas específicas del negocio (ej. generación de cortes de final de año). Tienen un nivel de reúso bajo. Estos servicios trabajan directamente con 1 o varios servicios entidad. Estos servicios están centrados en el negocio.
 - Servicios utilidad: Modela servicios que no están centrados en el negocio. Son los servicios con mayor reúso.

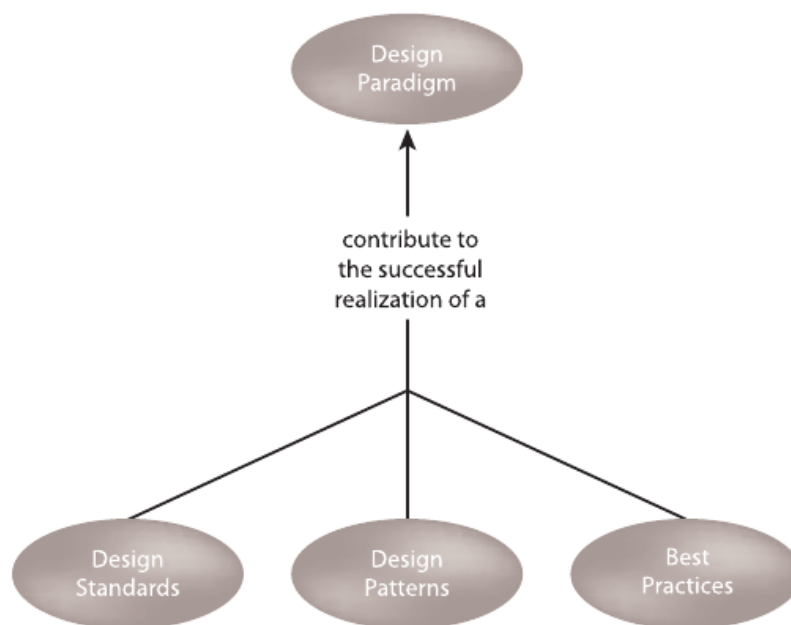


Figura 6: Componentes que hacen que el diseño lógico de la solución sea acorde al paradigma escogido.

Fuente: (Erl, 2007)

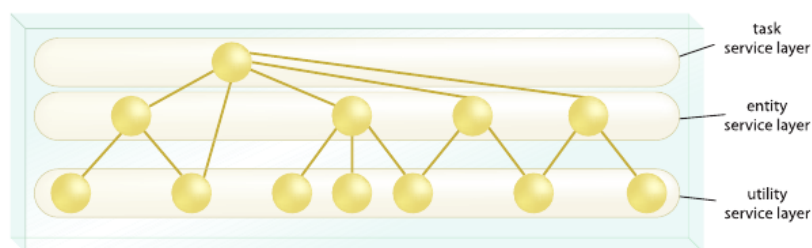


Figura 7: diferenciación entre tipos de servicio da lugar a la estructura en capas

Fuente: (Erl, 2007)

La diferenciación entre tipos de servicio da lugar a la estructura en capas

mostrada en la figura 7

- Composición de servicios: Es la agregación de servicios de manera ordenada.
- Inventario de servicios: Es la agrupación de varios servicios según un criterio definido por la organización. Cada inventario de servicios tiene su propio estándar de diseño e, inclusive, su propia configuración arquitectónica. El desarrollo de los inventarios de servicio es hecho a modo top-down, con la construcción de blueprints, también llamados modelo de servicios de negocio o modelos de inventario de servicios.

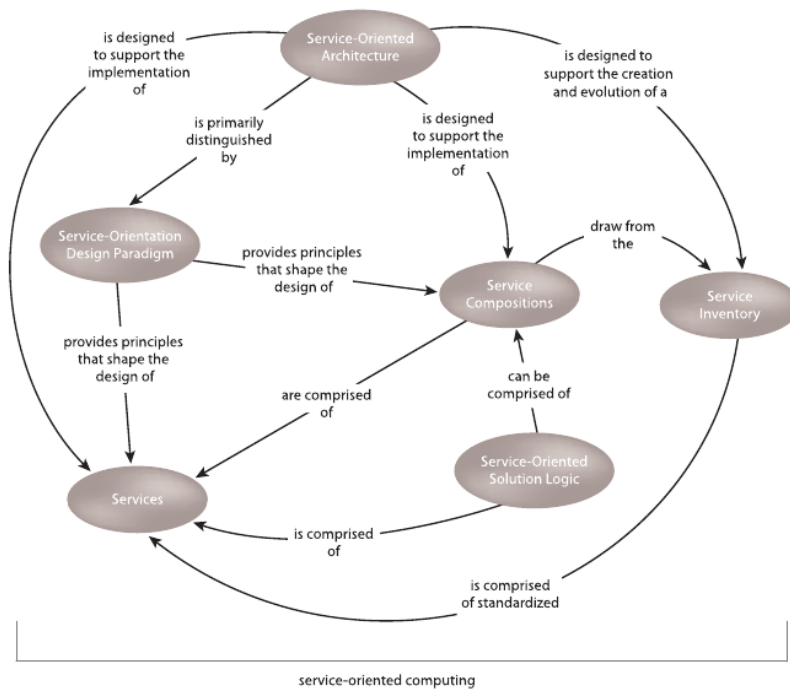


Figura 8: Interacción de los conceptos clave en la computación orientada a servicios

Fuente: (Erl, 2007)

4.4.3. Pertinencia con el proyecto

La arquitectura base para la ejecución del proyecto es SOA. Esto brinda diferentes beneficios a la hora del desarrollo del proyecto, como los son:

- Gracias al bajo acoplamiento conseguido con la arquitectura SOA, los componentes implementados son altamente reutilizables, lo que facilita y agiliza el proceso de desarrollo. Además,
- Es posible consumir servicios creados por terceros, limitando los servicios que deben ser creados a los que sean estrictamente de negocio.

4.5. Scrum

Sus creadores lo describen como “Un marco de trabajo por el cual las personas pueden acometer problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente” (Schwaber y Sutherland, 2013). Scrum se caracteriza por ser ágil, ligero, fácil de entender y difícil de llegar a dominar.

Scrum se basa en el empirismo, que dice que el conocimiento proviene de la experiencia, por lo que “utiliza un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo” (Schwaber y Sutherland, 2013).

Los tres grandes pilares de la teoría de scrum son:

- Transparencia: Todos los aspectos significativos deben ser visibles para sus stakeholders respectivos.
- Inspección: Los artefactos de scrum (y su progreso) deben ser inspeccionados frecuentemente para detectar variaciones. Estas inspecciones deben ser realizadas por un experto.
- Adaptación: Cuando una inspección detecta una variación que no cae en los límites permitidos, se deben hacer los reajustes necesarios para que el producto vuelva a rumbo deseado. Estos cambios deben hacerse lo más pronto posible, por esto las inspecciones deben ser realizadas de manera periódica a lo largo del desarrollo del proyecto.

Capítulo 5

Marco teórico

5.1. Redes Sociales

La información contenida en la actual sección es tomada del libro *Social Network Analysis for Startups* (Tsvetovat y Kouznetsov, 2011)

5.1.1. SNA: Social Network Analysis

La administración de una red social fuera de línea fue estudiada desde inicios del siglo XX (Arnaboldi y cols., s.f.) con un enfoque socio-matemático llamado “análisis de redes sociales” (SNA por sus siglas en inglés: Social Network Analysis). Sin embargo, era difícil el análisis del comportamiento humano según los designios de la SNA, puesto que la información debía ser recopilada por medio de entrevistas a las personas. Aun así, el enfoque SNA fue utilizado para analizar el comportamiento terrorista o inclusive el comportamiento de trabajadores en una empresa. (Tsvetovat y Kouznetsov, 2011)

5.1.2. Analisis egocéntrico

Los estudios basados en SNA pueden ser de tipo egocéntrico o sociocéntrico (Socievole y Marano, 2012). En los estudios egocéntricos de una red social, se analiza un individual dentro de una red social y todas las conexiones de

éste hacia otros individuales en la red social analizada. El individual analizado es llamado “ego” y los individuales que hacen conexión con él son llamados “alters”. Se han identificado 4 capas en el estudio de las redes egocéntricas, estas son:

- Support clique: En esta capa se identifican los alters con los que el ego hace más contacto por alguna razón de peso para él (e.g. para obtener soporte emocional). Esta capa tiene, en promedio, 5 alters.
- Sympathy group: En promedio a éste corresponden 15 alters.
- Affinity group: En promedio a éste corresponden 50 alters.
- Active network: En promedio a éste corresponden 150 alters.

Los números dados en las capas descritas en el análisis egocéntrico son congruentes con el número de Dunbar, el cual representa el umbral promedio de número de alters sobre la capa “active network” (150) según Robin Dunbar, argumentando que este límite se debe a la capacidad cognitiva del cerebro humano (Arnaboldi y cols., s.f., Pag. 3) (como más adelante será nombrado, los servicios de redes sociales ayudan al ser humano a gestionar su active network, proporcionando herramientas que, en teoría y de acuerdo a la brecha tecnológica, lo ayudarán a mantener sus lazos con los alters de su red ego).

El análisis egocéntrico permite conocer los factores que dirigen al ego a crear vínculos débiles o fuertes con potenciales alters, albergándolos en alguna de las cuatro capas o en ninguna.

Con la creación de las OSN y la gran cantidad de información que describe el comportamiento humano sobre este tipo de red social, ha sido más sencillo utilizar el enfoque de la SNA para estudiar que comportamientos tienen los humanos sobre una red social establecida.

5.1.3. Grado de centralidad

En todas las redes, sean virtuales o no, existen personas que son mas “importantes” que otras, más **populares**. Estas celebridades representan una parte muy pequeña de la red, pero debido a su gran influencia siempre es bueno identificarlos. Para esto se utiliza el **grado de centralidad**.

El grado de un nodo es la cantidad de conexiones que posee. En una red social, esto se representa por medio de las relaciones que cada nodo tenga, y ya que el significado de la relación varía en función de cada red, es necesario entender que significan las posibles relaciones existentes en una red para hacer el análisis correspondiente. Por ejemplo, en una red como Twitter en donde las relaciones son unidireccionales, puede existir un nodo con un grado de salida muy alto, esto es una persona que sigue a muchas otras. Aunque esta persona tenga un grado de centralidad muy alto, no representa una celebridad, sin embargo, un nodo que tenga un grado de entrada muy alto, que es seguido por muchas personas, si representa una persona que es muy popular en esta red.

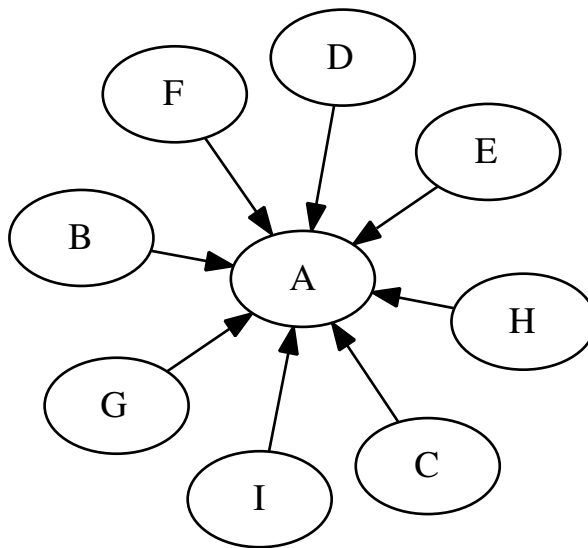


Figura 9: Red de estrella.

Fuente: Autores

En la figura 9 se puede ver un caso en el que el nodo A es una clara celebridad de la red. Este tipo de configuración, llamada red de estrella, es muy poco común en la vida real, pero sirve de ayuda visual para entender a simple vista el concepto de centralidad.

5.1.4. Grado de cercanía

A menudo se puede ver que personas que no tienen mayor influencia aparente en una red son capaces de difundir un mensaje en una gran parte de la red. Esto se debe a que tienen buenas conexiones en la red que les permiten llegar a mas personas, sin que ellos en si sean “importantes” en la red. Para medir que tan bien o mal posicionado esta un nodo en la red se utiliza el **grado de cercanía**. Este calculo es bastante caro computacionalmente ya que conlleva una gran cantidad de cálculos.

Los pasos para calcular el grado de cercanía de los nodos de una red son:

1. Calcular la ruta mas corta entre todos los pares de nodos posibles, utilizando el algoritmo de Dijkstra, y almacenar estos valores en una tabla.
2. Para cada nodo de la red:
 - a) Calcular la distancia promedio con todos los demás nodos.
 - b) Dividir el promedio por la distancia mas alta.
 - c) Calcular el inverso del valor anterior.
3. normalizar cada valor obtenido para obtener valores en el rango de 0-1.

Los nodos que tengan un valor mas cercano a 1 son los que tienen una distancia promedio menor con los nodos de la red, o los que tienen “*mejores contactos*”.

5.1.5. Factor distancia en la formación de las redes sociales

La formación de redes sociales (tanto fuera de línea como en línea) es afectada por la distancia entre cada individual y el posible tipo de enlace que los uniría. En (Allamanis, Scellato, y Moscolo, 2012) se hizo un estudio acerca de la formación de lazos, la formación de triadas entre individuales de una red social basada en la inscripción localizaciones recomendadas y frecuentadas por los usuarios, teniendo como parámetros “la edad” o tiempo de vinculación del individual a la red social, el grado de cada individual (número de

conexiones que tiene un individual a otro) y la localización de cada individual en la red social. También se analizó cómo afectaba la creación de nuevos lazos con la movilidad del usuario (el desplazamiento por lugares geográficos distintos). En conclusión, se verificó que la formación de lazos depende proporcionalmente de la edad y del grado del individual y es inversamente proporcional a la distancia que a cada individual y que la formación de lazos puede modelarse con solo dos de los tres factores (el grado y la distancia); en cuanto a la formación de triadas, se verificó que ésta depende de las características sociales de la red, tomando énfasis en los individuales compartidos entre los posibles individuales formadores de triadas. Además, en cuanto a la creación de nuevos lazos teniendo en cuenta los lugares visitados por cada usuario de la red social, se presenta un patrón: Los usuarios escogen un lugar popular para visitar y, posteriormente, dirimen con que usuario crear un lazo teniendo en cuenta su popularidad y que frecuente los mismos lugares siempre.

5.1.6. Grado de intermediación

En las redes sociales, suelen formarse grupos mas pequeños que comparten un interés común. Por ejemplo, es mas probable que dos personas que comparten el gusto por los videojuegos interactúen entre si que dos personas que no lo hagan, sin embargo hay casos en los que una persona comparte gustos con diferentes grupos, ayudando a que esta persona se pueda relacionar de manera efectiva con un grupo mas extenso de personas. Estas personas son conocidas como “puertas frontera” ya que, gracias a ellos, es posible que dos grupos que no tengan nada en común puedan relacionarse entre sí. La medida que ayuda a identificar estos elementos en una red es el **grado de intermediación**, y consiste en lo siguiente:

1. Calcular la ruta mas corta entre todos los pares de nodos posibles, utilizando el algoritmo de Dijkstra, y almacenar estos valores en una tabla.
2. Para cada nodo n de la red, contar las veces que el nodo n aparece en la lista de rutas mas cortas,
3. normalizar cada valor obtenido para obtener valores en el rango de 0-1.

Cabe notar que este algoritmo es bastante lento para redes que son muy grandes.

En la figura 10 se puede ver un claro ejemplo de este fenómeno. Esta red, comúnmente denominada la red corbatín”gracias a su forma similar a la de un corbatín, muestra como el nodo D se encuentra entre 2 grupos de nodos que, de otra manera, no podrían conectarse.

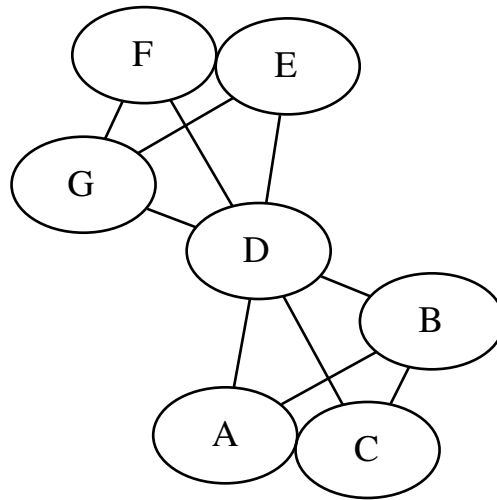


Figura 10: red corbatín.

Fuente: Autores

5.1.7. Díadas

Las díadas son la unidad básica de análisis una red social, ya que estas representan la relación entre una y otra persona, esto es, mis amigos, mis seguidores, mis suscriptores, etc. Existen 4 tipos de díadas, representadas en la figura 11, su uso varía en función del significado de la relación.

La díada 1 indica que ambos individuos existen en la red, pero todavía no existe ninguna relación entre ellos. Las díadas 2 y 3 muestran una relación unidireccional entre los dos individuos, la única diferencia es el sentido de esa relación. La díada 4, por su parte, es la de mayor interés de las cuatro ya que muestra una relación bidireccional entre los individuos, siendo esta la

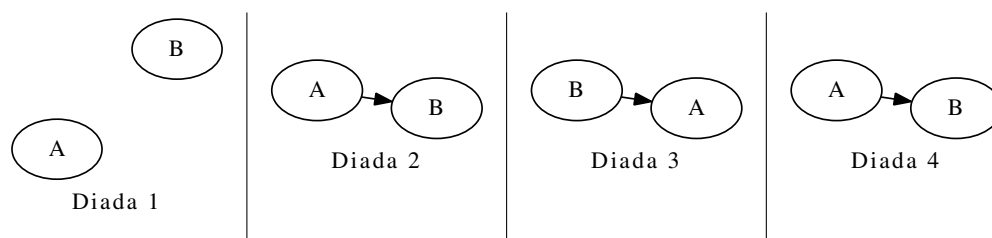


Figura 11: Tipos de diadas asimétricas.

Fuente: Autores

relación que mayor peso tiene en una red social dado que nos dice que existe un alto grado de reciprocidad en el intercambio de información entre ambos individuos.

5.1.8. Tríadas

Las triadas son básicamente 3 nodos conectados de alguna manera. Al igual que con las diadas, las triadas también pueden ser simétricas o asimétricas, dependiendo estrictamente del contexto en que son utilizadas. Existen 4 tipos de triadas simétricas, ilustradas en la figura 12.

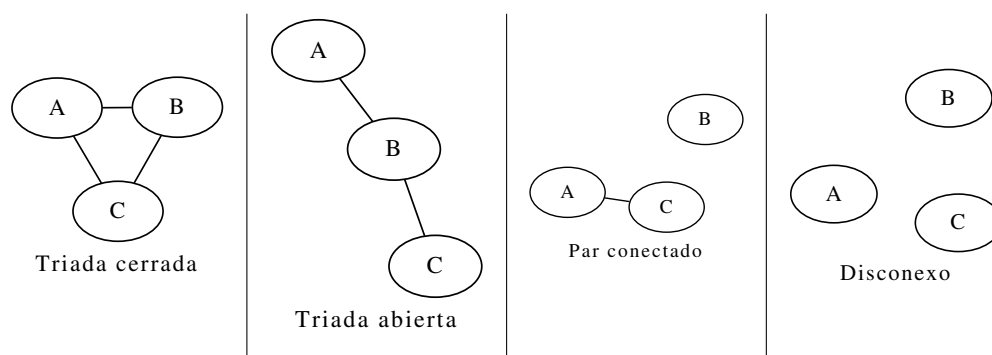


Figura 12: Tipos de triadas simétricas.

Fuente: Autores

Por otra parte, existen 16 tipos de triadas asimétricas, numeradas del

1-16. Su uso es mas frecuente ya que de ellas se puede hacer un análisis mas complejo en comparación a las díadas. Cada una de estas triadas recibe un nombre específico para facilitar su identificación, a continuación se explica como debe leerse ese nombre:

- El primer numero representa la cantidad de vértices bidireccionales
- El segundo numero representa la cantidad de vértices simples
- El tercer numero representa la cantidad de vértices inexistentes
- Si una triada se repite, se utiliza una letra extra para determinar que variante es:
 - U - Arriba (Up)
 - D - Abajo (Down)
 - C - Circulo (Circle)
 - T - Transitiva (Transitive)

En la figura 13 se muestran todas las triadas posibles con su respectivo código asociado.

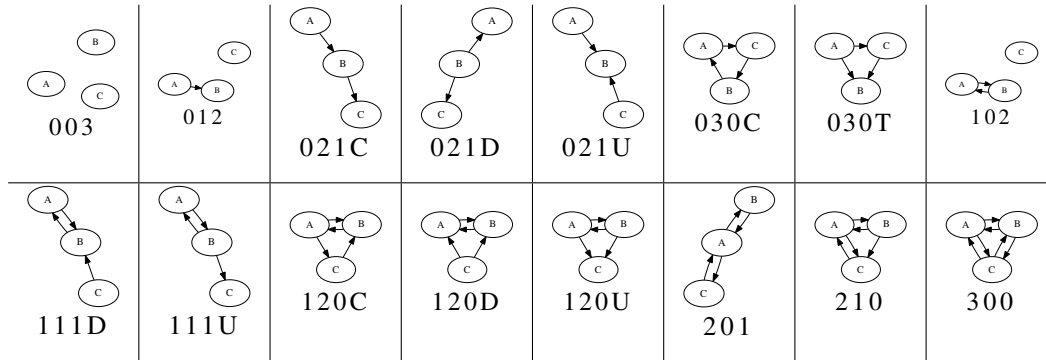


Figura 13: Tipos de triadas asimétricas.

Fuente: Autores

Gracias a esta discriminación topológica, se puede hacer un análisis mas completo de una red. Este análisis recibe el nombre de **Análisis Triadico**.

5.1.9. Análisis Triádico

Este proceso, que también recibe el nombre de **Censo Triádico**, consiste en contar la ocurrencia de cada uno de los tipos de triada para cada nodo, y de esa forma determinar el rol que desempeña este nodo en la red. Por ejemplo, un nodo que presente en mayoría triadas del tipo 4, 7 y 11 es un nodo que **genera contenido**, mientras que si la mayoría de sus triadas son del tipo 5 y/o 10, es un nodo que recibe o **consume contenidos**.

Adicionalmente se puede hacer el mismo análisis a la red en general, para tener un punto de vista global de la red. En la figura 14 se puede ver una de las redes mas utilizadas en la teoría de redes sociales, la red de Krackhardt-kite. En esta red se pueden ver muchas características de una red social, facilitando el estudio de las mismas. Al hacer el censo a esta red, nos damos cuenta que presenta una gran cantidad de nodos tipo 201, que representan un agujero estructural, y nodos tipo 300, que representan triadas cerradas, esto nos indica que en esta red existen zonas que tienen una gran concentración de nodos interconectados, mientras hay zonas que no se encuentran muy pobladas. Todo esto se puede ver a simple vista en esta red, pero para redes mas grandes puede que represente un problema mayor.

5.2. Business Process Modeling Notation (BPMN)

La información contenida en la actual sección es tomada del libro *BPMN 2.0 Introduction to the Standard for Business Process Modeling* (Allweyer, 2010)

A continuación, se enuncian algunos conceptos básicos de BPMN.

5.2.1. Enlaces

Los enlaces sirven para unir o bifurcar el flujo de secuencia de un modelo. Son utilizadas cuando es necesario tomar algún tipo de decisión que lleve a tomar uno o varios caminos alternos. Existen varios tipos de enlaces, como lo son los enlaces exclusivos (XOR), paralelos (AND), inclusivos (OR) y complejos. A continuación, se muestran ejemplos de aplicación de cada uno de estos enlaces.

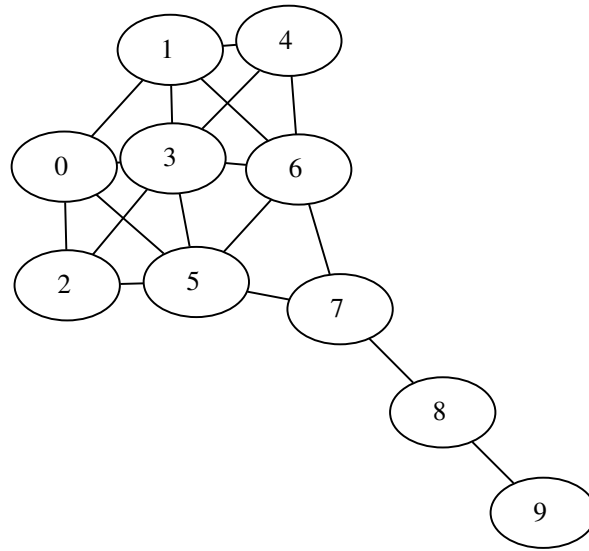


Figura 14: Red social de Krackhardt kite.

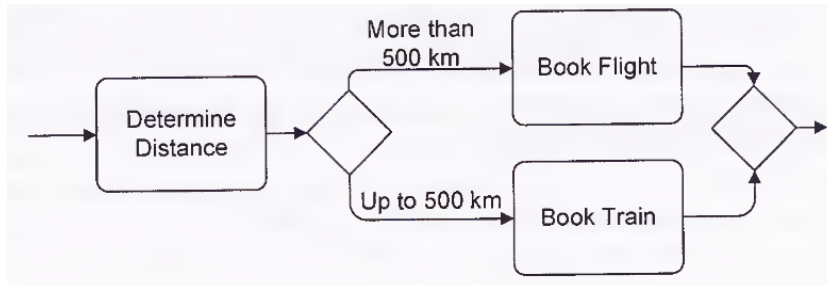
Fuente: Autores

Figura 15: Ejemplo de uso del enlace exclusivo

Fuente: (Allweyer, 2010)

En este caso (Figura 15), se realiza la actividad *Determinar distancia* y se decide que tipo de medio de transporte se debe utilizar. Nótese que luego de realizar la reserva, se vuelve a unir el flujo de secuencia del modelo, ya que no se sabe en la práctica cual de las 2 alternativas será la seleccionada.

En la figura 16 se ve un ejemplo de uso de este enlace. Aquí, luego de

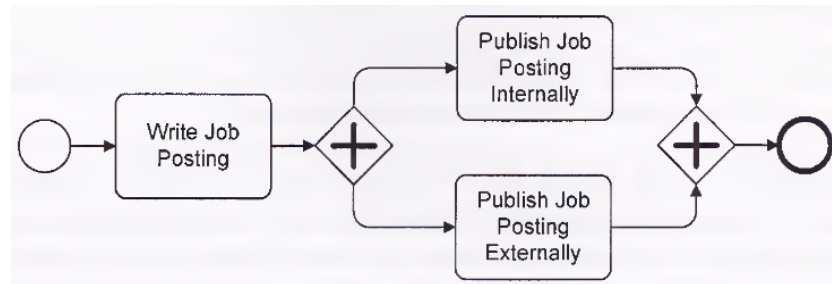


Figura 16: Ejemplo de uso del enlace paralelo

Fuente: (Allweyer, 2010)

que se redacta la oferta de trabajo, le procede a publicarla, tanto interna como externamente. En este caso, se utiliza el enlace paralelo para agilizar el proceso de publicación.

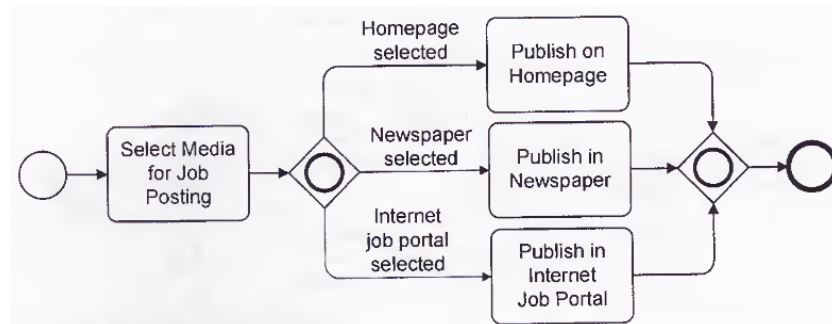


Figura 17: Ejemplo de uso del enlace inclusivo

Fuente: (Allweyer, 2010)

En la figura 17 se ve un ejemplo de uso en donde a partir de la actividad “Seleccionar medio para publicar oferta de trabajo” se pueden seleccionar una o varias opciones. Cualquier combinación de opciones, que al menos contenga una opción, es válida.

En el ejemplo mostrado en la figura 18, se requiere la referencia de 2 empleadores previos y de la universidad. En realidad, solo son necesarias dos referencias, pero para estar seguros se piden tres, por lo que tan pronto como

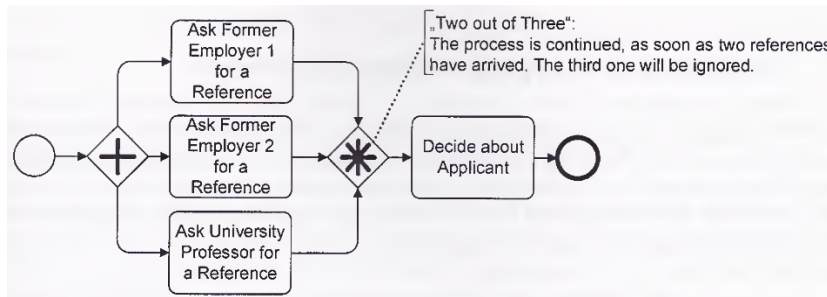


Figura 18: Ejemplo de uso del enlace complejo

Fuente: (Allweyer, 2010)

llegan las primeras 2 referencias, la tercera puede ser ignorada sin mayor problema.

5.2.2. Colaboración

A menudo, en un proceso intervienen diferentes partes interesadas (*Stakeholders*) y es necesario ver el proceso de manera global, de manera que el paso de mensajes entre las partes implicadas sea mas claro. A este tipo de diagramas se les llama **diagrama de colaboración**

La figura 19 muestra como es la interacción entre un aspirante y una empresa en el proceso de acceder a una oferta de empleo. Puede verse como, entre las actividades que ejecuta cada una de las partes, existe un paso de mensaje que une ambos procesos. Esta unión se representa por una flecha con línea punteada, donde un extremo tiene un círculo y el otro una flecha vacía que indica la dirección del mensaje.

Por ejemplo, la actividad “Recibir solicitud” recibe un mensaje de la actividad “Redactar solicitud de empleo”, por lo cual esta actividad (recibir solicitud) no puede iniciar si el aspirante no envía su solicitud. Esto significa que los mensajes deben ser respetados y una actividad no puede realizarse si le falta algún mensaje de entrada.

Sin embargo, en estos casos solo se conoce el proceso que sigue la empresa, por lo que es común representar a las partes externas como una caja negra (figura 20).

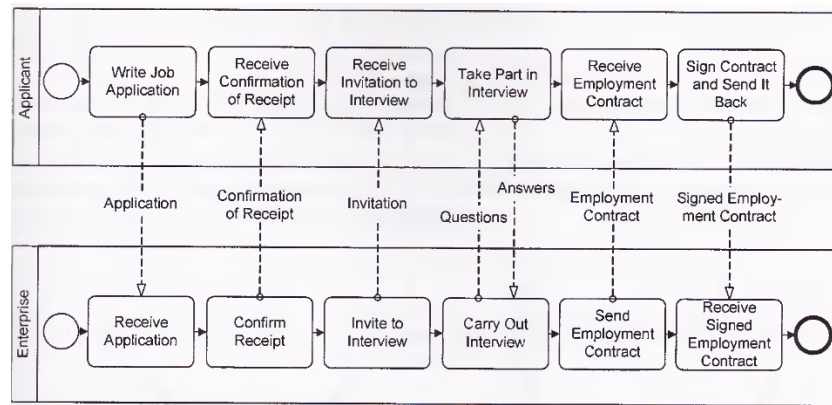


Figura 19: Ejemplo de diagrama de colaboración

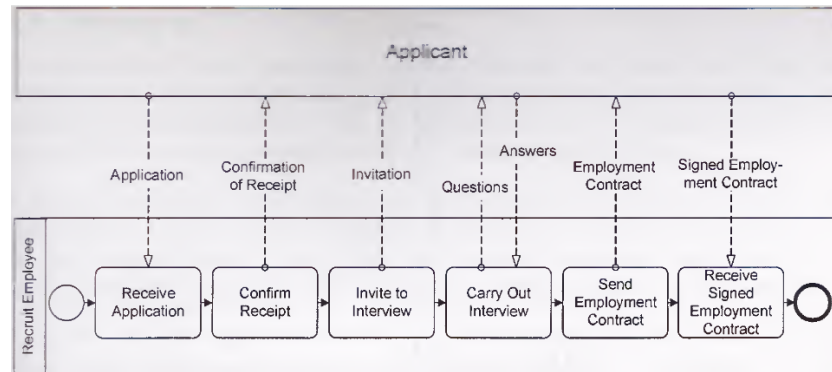
Fuente: (Allweyer, 2010)

Figura 20: Ejemplo de diagrama de colaboración con caja negra

Fuente: (Allweyer, 2010)

5.3. Service-Oriented Architecture (SOA)

5.3.1. Proceso de Análisis y diseño orientados a servicios

En la etapa de análisis y diseño, el arquitecto de software se reúne con un analista de negocio, el cual diseñará unos servicios candidatos que después se convertirán en los servicios incluidos en los blueprints. Luego de tener los planos, el arquitecto escoge un subconjunto de dichos candidatos a servicios para ser implementados físicamente, dotándolos de algún método para realizar la composición de servicios. En la figura 21 se muestra el proceso de análisis y diseño orientado a servicios.

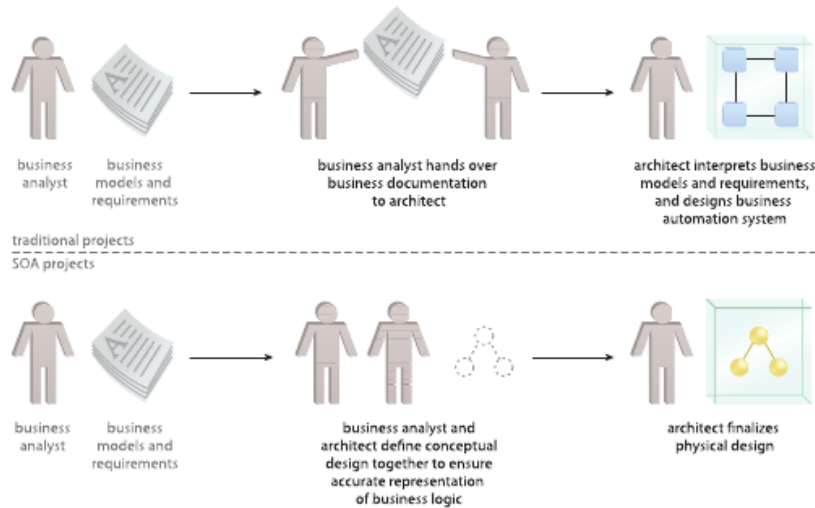


Figura 21: proceso de análisis y diseño orientado a servicios

Fuente: (Erl, 2007)

5.3.2. Metas y beneficios de la computación orientada a servicios

Las metas y los beneficios que trae la implementación de la computación orientada a servicios en una organización son:

- **Incremento de la interoperabilidad intrínseca:** Es una meta de la computación orientada a servicios ya que la composición de servicios requiere que, sin necesidad de integrar un servicio con el otro, estos puedan intercambiar información para desarrollar la funcionalidad a la que han sido inscritos. Aplicando principios de diseño orientados a servicios, así como también estándares de diseño orientados a servicios, se logra el incremento de la interoperabilidad intrínseca.
- **Federación incrementada:** Un entorno de TI federado es aquel en el cual los recursos y aplicaciones se manejan y gobiernan con autonomía y por sí mismos. En el ámbito de la orientación a servicios, cada servicio puede tener su propia implementación independiente y aun así comunicarse. Esto se logra por medio de una especial atención a los estándares de diseño.
- **Incremento de opciones en la escogencia de proveedores de servicios:** Como la federación en la computación orientada a servicios es una meta, la escogencia de proveedores de servicios diferentes es un beneficio ya que la composición de servicios puede ser lograda sin importar que proveedor provea de un servicio específico.
- **Incremento de la alineación entre el negocio y las tecnologías:** Ya que en el proceso de diseño actúan tanto el analista de negocio como el arquitecto de software, la alineación entre negocio y tecnologías es incrementada. La reconfiguración en la composición de servicios, además, provee alineación extra al poder cambiar el proceso de negocio y la composición de los servicios al mismo tiempo.
- **ROI:** Al hacerse inventarios de servicios y, además, ser los servicios reutilizables a lo largo del tiempo y compuestos de diferente forma gracias a la composición de servicios, se da una relación costo/beneficio más baja que con otros paradigmas usados.

- **Agilidad organizacional incrementada:** Debido a la orientación a servicios, se hace una composición rápida de los servicios que se tengan y se crean los que se necesiten para agilizar el proceso del departamento de TI y así agilizar los procesos subyacentes.
- **Reducción de cargas al departamento TI:** Debido a la agilidad organizacional cuando es aplicada la orientación a servicios, son reducidos costos operacionales (tiempo u otros recursos) y el departamento de TI adquiere un papel activo en el sector estratégico.

5.3.3. Principios de diseño SOA

Según (Erl, 2007), hay ocho principios de diseño sobre la computación orientada a servicios. A continuación se dará una definición de cada uno de ellos.

- **Standardized Service Contract (Contrato de servicio estandarizado):** Exposición de capacidades de los servicios por medio de los contratos de servicio.
- **Service Loose Coupling (Bajo acoplamiento de servicios):** Este principio busca el desacoplamiento de la implementación de los servicios y los consumidores de dichos servicios (sus capacidades), así como también del contrato de servicio. Este principio garantiza la interoperabilidad servidor – contrato – consumidor, sin necesidad de tener un alto acoplamiento entre los tres.
- **Service Abstraction (Abstracción de servicio):** Este principio enfatiza en mostrar las capacidades de servicio que sean necesarias, escondiendo aquellas que no deben publicarse en los contratos de servicio. El manejo de metadata es imprescindible en este principio debido a que, por la aplicación de este, los servicios tenderán a ser mayormente agnósticos y podrán ser utilizados en diversas composiciones de servicios, así como también en la publicación de los mismos (para que sean descubribles).
- **Service reusability (Reusabilidad de servicio):** Enfatiza en la necesidad de la creación de servicios agnósticos para que estos puedan

ser utilizados en gran variedad de escenarios y tengan una vida útil más larga.

- **Service Autonomy (Autonomía de servicio):** Se enfoca en la necesidad de un servicio de controlar su ambiente (entorno y recursos propios) en orden de volver este un servicio confiable y predecible.
- **Service Statelessness (Servicios sin estado):** Se enfoca en la construcción de servicios que, en lo posible, prescinden de utilizar estados (guardar información dentro de ellos mismos para utilizarla en un futuro). Si la cantidad de estados es excesiva, el servicio perderá la capacidad de ser escalable y disponible.
- **Service Discoverability (Capacidad de descubrimiento de servicios):** Se centra en el diseño de servicios entendibles y fácilmente identificados, haciendo posible el aumento del ROI (Return On Investment).
- **Service Composability (Capacidad de composición de servicios):** El más complejo de los principios, que se enfoca en la composición de servicios, en la variedad de configuraciones con que estos pueden hacerse una vez la creación de los servicios sea lograda con la utilización de los principios anteriormente enunciados.

5.4. Scrum

A continuación, se explican los diferentes roles y artefactos que existen en el marco de trabajo de Scrum.

5.4.1. Equipo Scrum (Scrum Team)

5.4.1.1. Product Owner (dueño del producto)

Es el responsable de gestionar el product backlog y el trabajo del equipo de desarrollo. Entre sus funciones se encuentran:

- Expresar los elementos del product backlog.

- Ordenar de la mejor manera posible los elementos del product backlog para lograr el objetivo final.
- Asegurarse de que el equipo de desarrollo entiende los items del product backlog.

5.4.1.2. Development Team (Equipo de desarrollo)

Son los encargados de llevar a cabo el incremento al producto en cada iteración o sprint. El equipo de desarrollo es el encargado de organizar y gestionar su propio trabajo.

El equipo de desarrollo se caracteriza por:

- Es autoorganizado: Se le dice al equipo que debe hacer, pero el es libre de decidir como lo hace.
- Son multifuncionales: Se tienen integrantes que manejan diferentes áreas de experticia para ayudar a realizar el incremento necesario.
- No se reconocen los sub-equipos que se puedan formar, la responsabilidad de lo que se haga recae en el equipo de desarrollo como un todo.

5.4.1.3. Scrum Master

Es el encargado de que se cumpla la teoría de scrum a lo largo de todo el proyecto. El Scrum Master ayuda a las personas externas al Equipo Scrum a entender qué interacciones con el Equipo Scrum pueden ser de ayuda y cuáles no (Schwaber y Sutherland, 2013).

Servicios que ofrece al Product Owner

- Encontrar técnicas para gestionar la Lista de Producto de manera efectiva
- Ayudar al Equipo Scrum a entender la necesidad de contar con elementos de Lista de Producto claros y concisos
- Entender la planificación del producto en un entorno empírico

- Asegurar que el Dueño de Producto conozca cómo ordenar la Lista de Producto para maximizar el valor
- Entender y practicar la agilidad; y,
- Facilitar los eventos de Scrum según se requiera o necesite.

(Schwaber y Sutherland, 2013)

Servicios que ofrece al Development Team

- Guiar al Equipo de Desarrollo en ser autoorganizado y multifuncional;
- Ayudar al Equipo de Desarrollo a crear productos de alto valor;
- Eliminar impedimentos para el progreso del Equipo de Desarrollo;
- Facilitar los eventos de Scrum según se requiera o necesite; y,
- Guiar al Equipo de Desarrollo en el entorno de organizaciones en las que Scrum aún no ha sido adoptado y entendido por completo.

(Schwaber y Sutherland, 2013)

Servicios que ofrece a la organización

- Liderar y guiar a la organización en la adopción de Scrum;
- Planificar las implementaciones de Scrum en la organización;
- Ayudar a los empleados e interesados a entender y llevar a cabo Scrum y el desarrollo empírico de producto;
- Motivar cambios que incrementen la productividad del Equipo Scrum; y,
- Trabajar con otros Scrum Masters para incrementar la efectividad de la aplicación de Scrum en la organización.

(Schwaber y Sutherland, 2013)

5.4.2. Eventos

5.4.2.1. Sprint

El Sprint representa un espacio de tiempo, no mayor a un mes, en el que se trabaja para crear un incremento en el desarrollo del proyecto. Es conveniente que los sprint tengan una duración consistente a lo largo del proyecto, y un nuevo sprint inicia tan pronto el actual termina.

Cada sprint debe tener un objetivo definido (Sprint Goal), un plan flexible y concepto de “terminado” claro.

5.4.2.2. Sprint Planning Meeting (Reunión de Planificación de Sprint)

Esta reunion se lleva a cabo al inicio de cada Sprint y no tiene una duración mayor a 8 horas. En esta reunión, que se lleva a cabo en presencia de todo el equipo scrum, se crea un plan para el sprint que inicia. En este plan se responden dos preguntas fundamentales, ¿Qué puede entregarse en el Incremento resultante del Sprint que comienza? y ¿Cómo se conseguirá hacer el trabajo necesario para entregar el Incremento?

5.4.2.3. Daily Scrum (Scrum Diario)

Esta reunion, que no debe durar mas de 15 minutos, se realiza a diario entre los miembros del development team. El objetivo de esta reunion es socializar lo que se hizo en las ultimas 24 horas y planear que hacer en las proximas 24 horas. Se evalua si se esta haciendo lo necesario para cumplir el sprint goal y, si es necesario, se puede adaptar o redefinir el trabajo del resto del sprint.

5.4.2.4. Sprint Review (Revisión de Sprint)

Al final de cada sprint se realiza esta reunion cuyo objetivo es el de socializar lo que se hizo en el presente sprint. En esta reunion se discuten cosas como qué fue bien durante el Sprint, qué problemas aparecieron y cómo fueron resueltos esos problemas. Al final de la revisión se debe generar

un product backlog actualizado con los elementos que se proponen para el siguiente sprint. Esta reunion tiene una duracion no mayor a 4 horas.

5.4.2.5. Sprint Retrospective (Retrospectiva de Sprint)

Esta reunión es similar al sprint review, pero en lugar de tratar el Qué se hizo, se trata el Cómo se hizo. Al final de esta reunión se genera un plan para mejorar el desempeño del equipo de scrum para que los sprint posteriores sean de mayor provecho para el proyecto.

5.4.3. Artefactos

5.4.3.1. Sprint Goal (Objetivo del Sprint)

El sprint goal es una meta que se plantea al inicio de cada sprint que puede ser alcanzada mediante el incremento en el proyecto. Este sprint goal “Proporciona una guía al Equipo de Desarrollo acerca de por qué está construyendo el incremento” (Schwaber y Sutherland, 2013). Es importante y necesario que el objetivo sea claro, coherente y sea entendido por todos los integrantes del equipo de scrum.

5.4.3.2. Product Backlog (Lista de Producto)

Esta lista representa todos los requisitos que tenga el proyecto o producto que son conocidos y entendidos en un momento determinado del desarrollo. Debido a la naturaleza cambiante y dinámica del entorno, la lista nunca está vacía. A medida que el producto evoluciona, la retroalimentación que se obtiene ayuda a completar la lista y a refinar el producto final.

5.4.3.3. Sprint Backlog (Lista de Pendientes del Sprint)

Esta lista esta compuesta por los diferentes items seleccionados del product backlog que van a ser tratados en cada sprint. Adicionalmente, se incluye un plan que ayude a conseguir el objetivo del sprint.

La Lista de Pendientes del Sprint es una predicción hecha por el Equipo de Desarrollo acerca de qué funcionalidad formará parte del próximo Incremento

y del trabajo necesario para entregar esa funcionalidad en un Incremento (Schwaber y Sutherland, 2013).

5.5. Requerimientos no funcionales

5.5.1. Mantenibilidad

En (?, ?), los autores hicieron una valoración de las métricas utilizadas para evaluar la mantenibilidad del software, según el estandar ISO/IEC 9126, encontrando deficiencias en la métrica más usa, el Índice de Mantenibilidad, o MI por sus siglas en ingles. Encontradas las deficiencias en claridad, crearon nuevas métricas basados en las subcaracterísticas usadas para medir la mantenibilidad de un software. Las subcaracterísticas atadas a la Mantenibilidad según el estandar son:

- **Analizabilidad (Analisability):** Se refiere a la dificultad de saber en que parte del software se debe hacer un cambio y, a su vez, cuales tienen deficiencias.
- **Cambiabilidad (Changeability):** Se refiere al nivel de dificultad para hacer adaptaciones al software.
- **Estabilidad (Stability):** Compete a este el análisis de cuan estable es el software mientras este está sufriendo algún cambio
- **Probabilidad (Testability):** Responde a la pregunta, ¿cuán difícil es probar el sistema luego de haberle hecho una modificación?
- **Conformidad de mantenibilidad (Maintenability conformance):** Éste se refiere al análisis del cumplimiento de estándares en cuanto a mantenibilidad se refiere.

Los autores presentan propiedades del código fuente que serán cruzadas con las subcaracterísticas del estandar ISO/IEC 9126 y, luego, dan las medidas creadas. Ellas son:

- **Volumen:** Cantidad de código del sistema. Mientras más grande es, más difícil es de analizar.

- **Complejidad por unidad:** Se refiere a la complejidad requerida por una unidad, esto es, “la pieza de código más pequeña que puede ser ejecutada y probada individualmente” (? , ?), para cumplir su uso.
- **Duplicación:** Se refiere al código clonado” (? , ?), esto es, código repetido muchas veces.
- **Tamaño de unidad:** Se refiere al tamaño de las unidades en un software.
- **Prueba de unidad:** Son unidades creadas especialmente para probar otras.

A continuación se presenta el cruce propiedades/subcaracterísticas:

		source code properties				
		volume	complexity per unit	duplication	unit size	unit testing
ISO 9126 maintainability	analysability	x		x	x	x
	changeability		x	x		
	stability					x
	testability		x		x	x

Figura 22: Cruce entre las subcaracterísticas atribuidas a la mantenibilidad del software en el estandar ISO/IEC 9126 y las propiedades a nivel de código. Cada cruce es indicado por una *x*

Fuente: (? , ?)

Así, en el presente proyecto, se ha decidido utilizar las siguientes medidas creadas por los autores:

- **Nivel de complejidad por unidad:** El nivel de complejidad es calculado, en primera instancia, calculando la complejidad ciclomática de cada unidad del software, la cual es definida como

$$cc = e - n + 2$$

donde n representa a los nodos del grafo de control de flujo y e el número de aristas de dicho grafo(*****paginaweb). Luego de esto, será necesario ubicar cada unidad en un nivel de riesgo. En el siguiente cuadro se podrá observar cómo son valorados los niveles de riesgo:

CC	Risk evaluation
1-10	simple, without much risk
11-20	more complex, moderate risk
21-50	complex, high risk
> 50	untestable, very high risk

Figura 23: correspondiente al nivel de riesgo en el software presentado por la complejidad ciclomática promedio del sistema

Fuente: (?, ?)

Luego de tener dichas cuentas y, además, por cada unidad tener los LOC asociados, se ubicarán las sumas de los LOC ubicados en cada nivel de riesgo, esto es, $\sum LOC_{ij}$, donde i representa el cambio en los niveles de riesgo y j representa el cambio en las unidades que pertenecen a dicho nivel i , dentro de un nivel maximo relativo de LOC que podrían ser riesgosas al momento de mantener el software.

- **Duplicación:** Se utiliza una medida que, aunque básica, debido a los estandares definidos para la realización de este proyecto, se hace más precisa. La técnica, descrita en (?, ?), consiste en contar los duplicados como un bloque de 6 líneas de código exáctamente iguales en una o varias partes del software. Luego, se harán medidas relativas al dividir la cantidad de líneas de código duplicadas por la cantidad total del

	maximum relative LOC		
rank	moderate	high	very high
++	25%	0%	0%
+	30%	5%	0%
o	40%	10%	0%
-	50%	15%	5%
--	-	-	-

Figura 24: correspondiente al nivel máximo relativo de LOC que podrían ser riesgosas al momento de mantener el software

Fuente: (?, ?)

software, clasificando el resultado en una de las categorías mostradas en la figura ??.

■ **Volumen por unidad:** FALTANTE

En lo que concierne a la mantenibilidad de web services específicamente, se tomó de (?, ?) aquellas medidas que pueden ayudar a medir la mantenibilidad en un sistema SOA hecho con web services.

En (?, ?), basados en las definiciones de los documentos WSDL, los autores crean medidas de complejidad basados en la complejidad de los tipos de datos manejados, haciendo de estos un árbol formado por estructuras de datos simples y complejas. Cabe anotar que los tipos de datos complejos creados en el WSDL y puestos dentro de otro tipo de dato complejo hacen un sub-árbol, haciendo más complejo el tipo de dato que compone. La complejidad de un tipo de dato se mide por la profundidad del árbol que lo compone. Así, a continuación se explicarán las medidas utilizadas:

- **Métrica de complejidad basada en mensaje:** Esta métrica busca medir la complejidad de los mensajes pasados de un servicio a otro

rank	duplication
++	0-3%
+	3-5%
o	5-10%
-	10-20%
--	20-100%

Figura 25: clasificación de duplicación de código

Fuente: (?, ?)

haciendo una sumatoria de las complejidades de los tipos de datos utilizados en los mensajes, es decir:

$$COM_k = \sum_{i=1}^n ci$$

donde k es el k -ésimo mensaje, i es el tipo evaluado y n el número de tipos que utiliza el k -ésimo mensaje.

- **Métrica de complejidad basada en operación:** En esta métrica se busca medir la complejidad de una operación definida en el documento WSDL. Para ello se ha de tener en cuenta que las operaciones trabajan con mensajes de salida y de entrada. Por tanto, la métrica será la media de la complejidad de todos los mensajes de salida y entrada que

estén definidos en la operación, esto es:

$$CBO = \frac{\sum_{i=1}^p COM_i + \sum_{j=1}^q COM_o}{m}$$

donde p representa la cantidad de mensajes de entrada y q representa la cantidad de mensajes de salida de la operación. Asimismo, m representa la cantidad total de mensajes.

5.5.2. Reusabilidad

5.5.2.1. Medidas de reusabilidad

En (Erl, 2007), se propone una calificación para las medidas de reuso planeadas, estas son:

- **Reusabilidad táctica:** Es la reusabilidad llevada a cabo sólo con las capacidades de los servicios que son requeridas inmediatamente.
- **Reusabilidad de destino:** Es la reusabilidad llevada a cabo con capacidades que serán más propensas a ser reusadas.
- **Reusabilidad completa:** Es la reusabilidad que equipa a un servicio con todas las características que éste puede cumplir.

Para llevar a cabo una medida del correcto reuso de los servicios, será necesario tener medidas de las capacidades de dichos servicios que son, realmente, utilizadas, tales como la cantidad de consumidores y la frecuencia con que son usadas.

5.5.2.2. Blueprints y reusabilidad

Además, en orden de hacer reusables los servicios, se debe lograr adicionar el estatus de *servicio agnóstico* a la mayor cantidad de servicios posible, esto es, separando a los servicios construidos de cualquier vínculo irreparable con el negocio o con las diversas plataformas.

También, los arquitectos de información y los analistas de negocio deberán separar cuidadosamente cada servicio en cada blueprint de servicios, con motivo de agrupar los servicios por dominios de aplicación en la organización. Con esta metodología será posible una mayor separación de necesidades y, por ende, mayor desacoplamiento entre servicios.

5.5.2.3. Logic Centralization: Patrón de diseño pro-reusabilidad

Cuando se habla de reusabilidad en SOA, se habla también de un patrón de diseño llamado Logic Centralization. Este patrón es aquel que garantiza la reutilización de un mismo servicio cuando un mismo requerimiento quiera ser utilizado por un servicio consumidor. A manera de ejemplo, uno podría tener dos servicios con logica duplicada de facturación y el servicio consumidor. Esto genera un costo a largo plazo debido al re-desarrollo de capacidades ya suplidas por desarrollos anteriores. Se debe, pues, centralizar la lógica a un servicio agnóstico y utilizar este siempre que se necesite. Si este último no llegase a tener las capacidades que se requieren, será necesario entonces adicionarlas.

A su vez, el patrón de diseño Contract centralization debe ser utilizado en conjunto con Logic Centralization para poder centralizar puntos de acceso a una misma capacidad requerida por un servicio consumidor. Contract centralization requiere que los servicios consumidores se dirijan siempre al contrato del servicio como punto de acceso al mismo y nunca a otra fuente externa.

5.5.2.4. Granularidad y reusabilidad

En cuanto a granularidad, en (Erl, 2007), se identifica un tratamiento a esta a nivel de servicio, de capacidades de servicio y de datos. En general, el tratamiento a nivel de servicio tiende a ser hecho con grano fino debido a la filosofía de SOA, la cual lleva a la producción de servicios agnósticos primordialmente. En cuanto a las capacidades, la tendencia es a hacer versiones de grano fino y de grano grueso de cada una. En cuanto a los datos, se tiene a trabajar con grano grueso en ellos.

5.5.3. QiU: Quality in Use

En (?, ?), antes de hablar de la usabilidad del software, los autores nombran la existencia del concepto QiU según ISO/IEC 9126-4, el cual lo define como **La capacidad del producto de software para permitir a usuarios específicos lograr metas específicas con eficiencia, productividad, seguridad y satisfacción en contextos de uso específicos**. En QiU se define a la usabilidad como una de las cualidades que realiza la capacidad del software para ser de calidad.

5.5.3.1. Usabilidad

En (?, ?), se nombran algunas de las definiciones de usabilidad. Una de ellas, la definición propuesta en la ISO/IEC 9126-4, dice: **La extensión por la cual un producto puede ser usado por usuarios específicos para lograr metas específicas con efectividad, eficiencia y satisfacción en un contexto específico de uso**.

A su vez, en (?, ?), enuncian tres acercamientos al concepto de usabilidad:

- **Usabilidad de sistema:** Data del conocimiento de las metas de la organización aplicadas al sistema construido.
- **Experiencia de usuario (UX):** Consideración de las metas hedónicas y pragmáticas de los usuarios al momento de construir el software.
- **Usabilidad de producto:** Soporte de UX y usabilidad del sistema al software construido.

5.5.3.2. User acceptance (Aceptación de usuario)

Se refiere, en (?, ?), a la acogida de un producto software por parte del cliente cuando éste cumple características específicas que *atrapan* a este. Dichas características son:

- **Utilidad percibida:** Esta se refiere a cuanto el software puede cumplir con las necesidades pragmáticas del usuario.

- **Facilidad de uso percibida:** Esta se refiere a la premisa de que el usuario busca siempre aquello que le tome menos tiempo aprender pero que, a su vez, le siga siendo útil.

5.5.3.3. Experiencia de usuario (User eXperience - UX)

En (?, ?) y según la ISO 9241-210 define UX como **Las percepciones y respuestas de una persona del uso o anticipado uso de un producto, sistema o servicio**. El análisis de UX es motivado por el cómo las personas se sienten al utilizar cierto software, su nivel de satisfacción al momento de usarlo y después de ello.

Los servicios de redes sociales (SNS por sus siglas en ingles: Social Network Services) como Facebook, LinkedIn, Twitter, SportTracker o Xportia, ofrecen servicios para la gestión de la OSN de cada usuario que acceda a estas aplicaciones. Según un estudio hecho para medir la experiencia de usuario (Elahi y Karlsen, 2012) (UX por sus siglas en ingles: User eXperience) en los SNS, se encontraron 8 categorías que son críticas a la hora de diseñar una SNS y son:

1. Self-expresion: Capacidad que tengan las OSN de compartir contenido relacionado a la vida real de los usuarios tal como lo pueden ser las fotos, los videos, los comentarios o las comunicaciones directas.
2. Reciprocity: Interacción bilateral en tiempo real, es decir, interacción instantánea con uno o varios individuales en la OSN (por ejemplo, por medio de los servicios de mensajería instantánea).
3. Learning: La información recibida por medio de la OSN debe poder ser utilizada en pro del desarrollo cognitivo del individual; debe existir información útil al individual que usa la OSN.
4. Curiosity: El contenido de la OSN debe ser interesante para quien la utiliza.
5. Suitability of functionality: Se refiere a cuán “utilizable” es una funcionalidad.
6. Suitability of content: La calidad y exactitud de la información que en la OSN reside debe ser suficiente para el individual perteneciente a ella.

7. Completeness of the user network: Los individuales deben querer pertenecer a la red social y buscar eficientemente a otros individuales para poder formar lazos con ellos y hacer crecer su red social.
8. Trust and privacy: Confianza en los servicios de las OSN, así como también la capacidad que tiene el usuario de gestionar la privacidad del contenido que comparte en dicha OSN. (Väänänen-Vainio-Mattila, Wäljas, Ojala¹, y Segerståhl, 2010)

Cada uno de las categorías nombradas hace parte de los factores que impulsan la utilización de los SNS para la gestión de las OSN de las personas.

5.5.3.4. Conceptos para medir QiU

Para medir las características de QiU en el software desarrollado, se utilizaron los conceptos retratados en (?), donde los autores crearon un framework para evaluar, tanto UX como usabilidad en sistemas móviles. Se analizaron los siguientes aspectos:

- **Eficiencia:** Capacidad del producto de software para ofrecer rendimiento apropiado, visto desde la cantidad de recursos usados.
- **Eficacia:** Capacidad del producto de software para permitir al usuario ejecutar tareas de manera precisa, según las necesidades de éste.
- **Satisfacción:** Valoración de la actitud del usuario frente al software utilizado
- **Productividad:** Capacidad del software para permitir al usuario ser efectivo (cumplir el objetivo sin gastar cantidades grandes de recursos valiosos para él).
- **Learn-ability:** Puede definirse como la capacidad que tiene el software para darse a conocer al usuario.
- **Seguridad:** Capacidad del software de no propiciar ningún daño sobre la propiedad del usuario (su empresa, su ser, etc.).
- **Entendibilidad:** Capacidad que tiene el software para dar a entender al usuario para que está hecho y él pueda decidir si puede cumplir con las tareas que requiere o no.

5.5.4. Seguridad

En (? , ?), la seguridad es definida como un concepto que guía a la protección de las posesiones, ya sean tangibles o intangibles. Para hablar de seguridad, primero definen conceptos clave de seguridad. Estos últimos serán listados a continuación:

- **Asset (Posesión):** Es definido como el tangible o intangible a proteger.
- **Threat (Amenaza):** Es un escenario en el cual es posible causar daño a una posesión. Esta requiere de la existencia de una vulnerabilidad para ser llevada a cabo con éxito.
- **Vulnerability (vulnerabilidad):** Es una debilidad que hace una amenaza posible.
- **Attack (Ataque):** Es la acción de llevar a cabo una amenaza.

Otros elementos de la seguridad, enfocada más al aspecto informático, son:

- **Autenticación:** Es la presentación de los participantes de la conexión. Este concepto responde a la pregunta *¿quien eres?*.
- **Autorización:** Es la posibilidad de un usuario (humano o no) de realizar una operación u obtener algún dato. Este concepto responde a la pregunta *¿que puedes hacer?*.
- **Auditoría:** *****
- **Confidencialidad:** Este concepto establece la necesidad de que la información compartida entre diferentes usuarios sea vista solo por ellos.
- **Integridad:** Este concepto es logrado cuando se logra proteger los datos de modificaciones indeseadas antes de que el receptor reciba un mensaje de un emisor.
- **Disponibilidad:** Es la seguridad de que una aplicación estará siempre disponible para usuarios legítimos.

En (? , ?), se exponen principios de seguridad en Web Services, los cuales son utilizados para definir el requerimiento no funcional seguridad. A continuación se exponen aquellos nombrados:

- **Aplicar defensa en profundidad:** Uso de varios *gatekeeper* (en español, portero), es decir, utilizar múltiples capas de seguridad para mantener las posesiones seguras.
- **Chequear en la puerta:** Utilizar la autenticación y la autorización como la primera capa de seguridad.
- **Compartimentar:** Limitar el rango de acción de un atacante aislando posibles vulnerabilidades.
- **Crear *por defecto* seguros:** Este principio se refiere a la necesidad de crear usuarios por defecto que sean manejados de forma segura, esto es, limitando sus privilegios sobre recursos (datos y operaciones) que sean sensibles y, quizás, deshabilitando dicha cuenta desde un principio.
- **No confiar en las entradas del usuario:** Debido a que el ataque principal es hecho por medio de entradas al sistema, será necesario aplicar el principio de defensa en profundidad con dichas entradas, nunca confiando en que ellas no sean potencialmente malignas.
- **Establecer límites de confianza:** Este principio se refiere a la decisión de ser permisivo o no con cierto flujo de datos o entradas, con el fin de poder analizar más en profundidad que amenazas pueden presentarse en el sistema según los límites que se den a dichos datos y entradas.
- **Fallar siendo seguro:** Este principio dice que, cuando ocurra un error, se deben retornar mensajes amigables al usuario y que no revelen información que pueda ser usada en contra para explotar vulnerabilidades de nuestro sistema.
- **Reducir el área en riesgo de ataque:** Este principio se refiere al control de recursos disponibles a usuarios. Si un recurso no se usa, será mejor deshabilitarlo o eliminarlo, reduciendo así el área vulnerable.

- **Asegurar el enlace más débil:** Se necesita estar atento de todas las posibles partes del sistema que puedan tener más vulnerabilidades y, por lo tanto, ser utilizadas para atacar el sistema.
- **Usar menos privilegios:** Al restringir privilegios a usuarios, es menos probable que un atacante pueda explotar las posibles vulnerabilidades de un sistema, no podrá utilizar ciertos recursos para atacarlo.

*** PENDIENTE ENCRIPCIÓN EN LOS PRINCIPIOS

5.6. Archimate 2.0

Según (?), archimate es un lenguaje estandar utilizado por los arquitectos de software para modelar las necesidades de cada *stakeholder*, dividiendo el contexto en el que se desenvuelve el software a desarrollar en tres capas: capa de negocio, capa de aplicación y capa de infraestructura.

5.6.1. Capas de archimate

A continuación serán descritas las tres capas definidas en el estandar archimate.

5.6.1.1. Capa de negocio

Esta capa envuelve todos los conceptos relacionados con la organización sobre la cual se aplicará el software a desarrollar, esto es, los roles, los servicios ofrecidos, los procesos, los productos y demás conceptos aplicados sobre su estructura y dinámica. Los artefactos de esta capa utilizados en el proyecto se enuncian en el cuadro 1.

5.6.1.2. Capa de aplicación

Esta capa cubre los conceptos relacionados con el modelamiento del sistema de información que soporta el negocio (la organización), previamente modelado en la capa de negocio. Los artefactos de esta capa utilizados en el proyecto se enuncian en el cuadro 2.











Cuadro 1: Artefactos de la capa de negocio

Fuente: (? , ?)

Artefacto	Descripción	Notación
<i>Actor de negocio (Business actor)</i>	Ente (persona) organizacional quien cumple tareas en la organización	<div>Business actor</div>
<i>Colaboración de negocio (Business collaboration)</i>	Rol que surge de la combinación de dos o más roles	<div>Business collaboration</div>
<i>Rol de negocio (Business role)</i>	Es un conjunto de responsabilidades que tiene asignado un actor. Dicho conjunto está definido por un solo concepto (nombre)	<div>Business role</div>
<i>Proceso de negocio (Business process)</i>	Agrupación de comportamientos que se ejecutan como una serie de pasos	<div>Business process</div>
<i>Función de negocio (Business function)</i>	Agrupación de comportamientos que requieren una serie de competencias o recursos de negocio	<div>Business function</div>
<i>Interacción de negocio (Business interaction)</i>	Comportamientos realizados por colaboraciones de negocio	<div>Business interaction</div>
<i>Evento de negocio (Business event)</i>	Hecho que dispara un comportamiento (o grupo de comportamientos) en la organización	<div>Business event</div>
<i>Servicio de negocio (Business service)</i>	Servicio que cumple una necesidad del usuario	<div>Business service</div>
<i>Producto (Product)</i>	Agrupación de servicios que, definidos por un contrato, serán compartidos a clientes (tanto roles dentro de la organización como para clientes externos)	<div>Product</div>

Cuadro 2: Artefactos de la capa de aplicación

Fuente: (?, ?)

Artefacto	Descripción	Notación
<i>Componente de aplicación (Application component)</i>	Unidad de software	 
<i>Colaboración de aplicación (Application collaboration)</i>	Colaboración entre dos o más componentes de aplicación para realizar una tarea que necesita de cada uno de ellos	 
<i>Función de aplicación (Application function)</i>	Agrupar comportamientos que pueden ser automatizados por un componente de aplicación	 
<i>Interacción de aplicación (Application interaction)</i>	Agrupar comportamientos que pueden ser automatizados por una colaboración de aplicación	 
<i>Servicio de aplicación (Application service)</i>	Un servicio que expone las funcionalidades automatizadas ofrecidas	 

5.6.1.3. Capa de infraestructura

Capa que modela el posicionamiento físico del software a utilizar, así como también los requerimientos que este tiene para su funcionamiento a nivel físico (servidores, redes, nodos, etc.). Los artefactos de esta capa utilizados en el proyecto se enuncian en el cuadro 3.

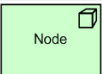

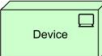



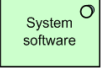

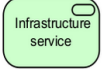

5.6.2. Relaciones

Las relaciones, en archimate, son aquellas conexiones existentes entre dos artefactos. Hay tres tipos de relaciones en archimate: relaciones estructurales, relaciones dinámicas y otras que no caben en las dos últimas.

- **Relaciones estructurales:** Estas relaciones modelan la coherencia estructural generada por la unión estructural de todos los artefactos de la arquitectura. En el cuadro 4 se hace un sumario de las relaciones estructurales utilizadas para unir los artefactos utilizados en el presente proyecto:
- **Relaciones dinámicas:** Esta relación expresa una unión temporal entre dos artefactos donde, posiblemente, uno de los artefactos use a otro







Cuadro 3: Artefactos de la capa de infraestructura

Fuente: (? , ?)

Artefacto	Descripción	Notación
Nodo (Node)	Recurso computacional que agrupa artefactos almacenables o desplegables para ser ejecutados	 
Dispositivo (Device)	Hardware que contiene elementos software para ser ejecutados	 
Red (Network)	Medio de comunicación entre dos o más nodos o dispositivos	 
Sistema de software (System software)	Software sobre el cual se realizan (o representan) los artefactos (componentes de aplicación) desplegables	 
Servicio de infraestructura (Infrastructure service)	Servicios que agrupan funcionalidades prestadas por nodos	 



Cuadro 4: Relaciones estructurales

Fuente: (? , ?)

Relación	Descripción	Notación
Asociación (Association)	Cumple la función de asociar dos artefactos que no tienen una relación más específica	
Usado por (Used by)	Modela el acceso a servicios o interfaces. El acceso a interfaces solo puede ser realizado por artefactos estructurales y a los servicios solo los artefactos comportamentales	
Realización (Realization)	Une un artefacto abstracto con otro más concreto que lo realiza	
Asignación (Assignment)	Une artefactos con aquellos que, por obligación, deben utilizar otros (por ejemplo, un rol con una función de negocio)	
Agregación (Aggregation)	Es la composición de un artefacto por otros. Esta composición no es destructiva, es decir, si el artefacto que es compuesto deja de existir, los otros seguirán existiendo	
Composición (Composition)	Es la composición de un artefacto por otros. Esta composición es destructiva, es decir, si el artefacto que es compuesto deja de existir, los otros también	

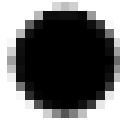
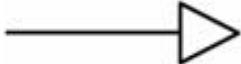
Cuadro 5: Relaciones dinámicas

Fuente: (?, ?)

Relación	Descripción	Notación
<i>Flujo (Flow)</i>	Describe intercambios de información entre artefactos comportamentales	
<i>Disparador (Triggering)</i>	Describe la relación temporal o factual entre 2 artefactos comportamentales	

Cuadro 6: Otras relaciones

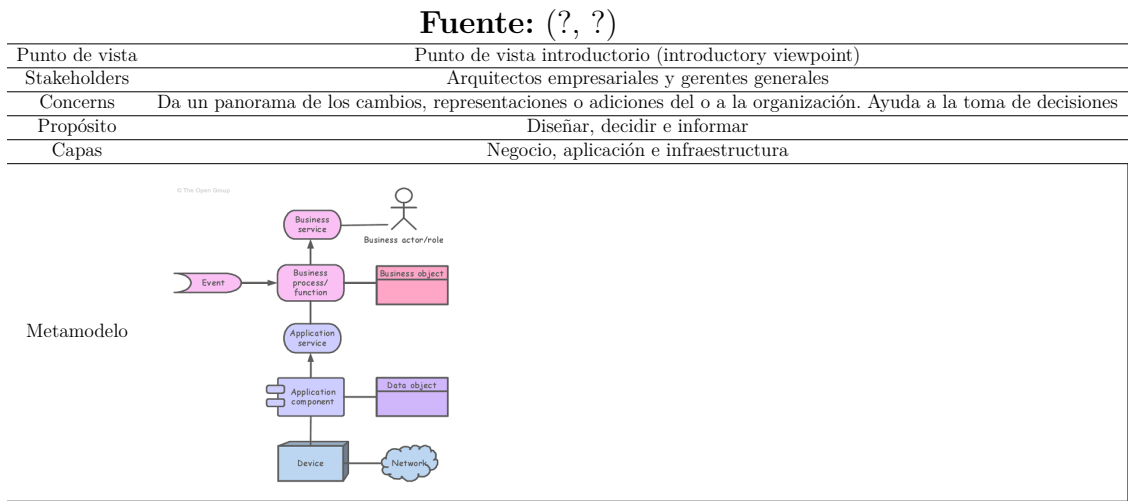
Fuente: (?, ?)

Relación	Descripción	Notación
<i>Unión (Junction)</i>	Cumple la función de asociar dos artefactos que no tienen una relación más específica	
<i>Especialización (Specialization)</i>	Indica la especialización de un artefacto tomando otro como referencia	

para cumplir un fin específico. En el cuadro 5 se hace un sumario de las relaciones dinámicas utilizadas para unir los artefactos utilizados en el presente proyecto.

- **Otras relaciones:** En el cuadro 6 se hace un sumario de las relaciones que no pueden ser incluidas en las dinámicas o en las estructurales y que son utilizadas para unir los artefactos utilizados en el presente proyecto.

Cuadro 7: Punto de vista introductorio



5.6.3. Vistas archimate

Debido a la existencia de diferentes stakeholders en el desarrollo de software, se hace necesario mostrar aquella parte de la arquitectura que cierto stakeholder necesita para tener una vista entera del negocio, el sistema de información o la infraestructura, o bien una combinación de ellos.

Archimate utiliza el concepto de “vistas” como la segmentación de la arquitectura en vistas que conciernen al stakeholder que quiera echar un vistazo a la arquitectura, escondiendo detalles de la arquitectura que no le interesan a este.

En los cuadros 7 a 12 se pueden ver las vistas utilizadas, con su descripción y su metamodelo.

5.7. Estado del arte

Se hizo una búsqueda de redes sociales basadas en deporte que existen actualmente en la red. Una vez encontradas, se eligieron exactamente 16


Cuadro 8: Punto de vista en capas

Fuente: (?, ?)

Punto de vista	Punto de vista en capas (layered viewpoint)
Stakeholders	Arquitectos de dominio, infraestructura, procesos, aplicación y negocio
Concerns	Impacto del cambio, flexibilidad, reducción de la complejidad y consistencia
Propósito	Diseñar, decidir e informar
Capas	Negocio, aplicación e infraestructura
Metamodelo	Se usan los artefactos y relaciones de todas las capas, según se considere pertinente

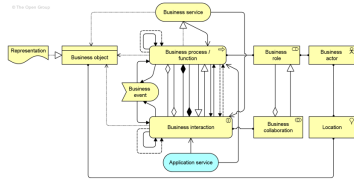
Cuadro 9: Punto de vista de función

Fuente: (?, ?)

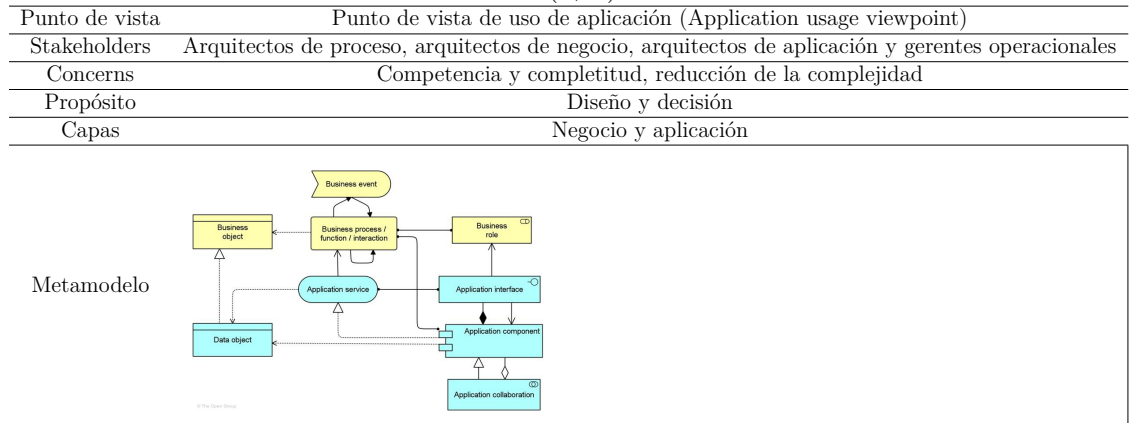
Punto de vista	Punto de vista de función (business function viewpoint)
Stakeholders	Organización, arquitectos de proceso y de dominio
Concerns	Identificación de competencias, identificación de actividades principales y reducción de complejidad
Propósito	Diseño
Capas	Negocio
Metamodelo	

Cuadro 10: Punto de vista de proceso

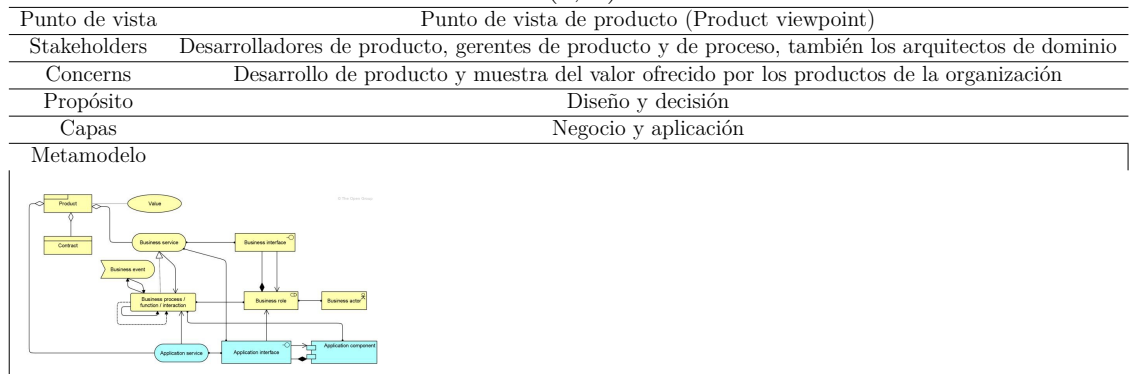
Fuente: (?, ?)

Punto de vista	Punto de vista de proceso (business process viewpoint)
Stakeholders	Gerentes operacionales, arquitectos de dominio y de proceso
Concerns	Estructura del proceso de negocio, consistencia y completitud así como también responsabilidades
Propósito	Diseño
Capas	Negocio y aplicación
Metamodelo	

Cuadro 11: Punto de vista de uso de aplicación

Fuente: (?, ?)

Cuadro 12: Punto de vista de producto

Fuente: (?, ?)

SNS deportivas que ofrecían, en conjunto, las funcionalidades que se observaban en las demás redes sociales que no fueron escogidas (Cuadros 13 a 17). Luego de la elección de la muestra de SNS, se reunieron aspectos de cada una hasta formar un grueso de sus funcionalidades y se realizó un cuadro de Funcionalidades vs SNS en donde se expresa con detalle cómo se presenta cada funcionalidad con respecto a cada SNS (en caso de no haber una conexión funcionalidad – SNS, entonces la casilla se dejó en blanco). En los cuadros 13 a 17 se da evidencia del análisis Funcionalidad vs SNS realizado.

Cada una de las funcionalidades que fueron descubiertas en otros SNS deportivos ya creados son el primer paso, pues, para conocer las necesidades de los usuarios de los SNS deportivos. El análisis de estos SNS será, entonces, el punto de partida para definir los requerimientos funcionales del SNS que plantearemos desde el punto de vista funcional.

Cuadro 13: Comparacion de redes, parte 1

Fuente: Autores.

Fun\Red social	Sportfactor	Deportesreunidos	Mybestplay	Subetudeporte
Gestión de foros		Si		Si
Gestión de encuentros deportivos		- Organización de eventos -Encuentros deportivos informales		
Creación de grupos		Si		
Manejo de torneos		- Organización y difusión		
Difusión info. Deportiva	-RSS de noticias	- Blog propio	-Difusión de eventos -Blog propio	- Gestión de blogs
Serv. self-expression		-Difusión de multimedia	-Difusión de multimedia	-Difusión de multimedia
Sistema estadístico	-Medición de avance en estadísticas del deporte practicado	- Sistemas de estadísticas para cada servicio		
Gestión de transversales	-Trainner personales -Guías de nutrición - Catalogo de lesiones y fisioterapia			
Servicios deportivos	-Guía deportiva (shops, restaurantes, etc.)			
Soporte multi-deporte	Si	Si	Solo deportes en equipo	Si
Gestión de tipos de usu.		- Equipos - Clubes -Centros deportivos	Si	
Gestión de sponsors			Si	
Gestión del conocimiento				
Gestión de geolocaliza.				
Soporte móvil				
Conexión con otros SNS				

Cuadro 14: Comparacion de redes, parte 2

Fuente: Autores.

Fun\Red social	Sporttia	Amateur	Fitivity
Gestión de foros			
Gestión de encuentros deportivos	- Organización de eventos en centros deportivos - Gestión de jugadores -Gestión de características del partido	- Publicación o búsqueda de eventos deportivos	-Basado en geolocalización
Creación de grupos			
Manejo de torneos			
Difusión info. Deportiva			
Serv. self-expression		-Difusión de multimedia	
Sistema estadístico			
Gestión de transversales			
Servicios deportivos	-Alquiler de centros deportivos	- Servicios de compra y venta de artículos deportivos	
Soporte multi-deporte	Si	Si	Si
Gestión de tipos de usu.	-Deportista -Centro deportivo	-Deportista -Equipo -Organización	
Gestión de sponsors		-Promoción como deportista, equipo u organización	
Gestión del conocimiento	- Clases virtuales		
Gestión de geolocaliza.		Si	Si
Soporte móvil			-Android -IOS
Conexión con otros SNS			

Cuadro 15: Comparacion de redes, parte 3

Fuente: Autores.

Fun\Red social	Bkool	Deportmeet	Sportsnak
Gestión de foros			- Foros con profesionales (managers, coaches, teams) - Ofrece posibilidad al usuario de ser moderador de foros
Gestión de encuentros deportivos	- Creación de eventos deportivos (solo o con amigos) - Gestión de “retos”	- Gestión de eventos deportivos	- Manejo de eventos deportivos
Gestión de grupos	Si		
Manejo de torneos			
Difusión info. Deportiva	- Gestión de información de ligas	- Artículos de profesionales	-Asociación con blogs deportivos - Manejo de “live scores”
Serv. self-expression	-Subida de texto plano -Difusión de multimedia	-Difusión de multimedia	- Manejo contenido plano y multimedia - Uso de mensajería instantánea
Sistema estadístico	- Estadísticas de deportista	- Gestión del nivel del deportista -Manejo de perfiles de usuario	
Gestión de transversales		-Foros de nutrición	
Servicios deportivos		- Venta de artículos deportivos	- Módulos para negociantes en temas de deporte - Manejo de ofertas en ofrecimiento de instalaciones deportivas - Herramientas para hacer “boost” a negociantes (bussiness member)
Soporte multi-deporte	Deportes de ruta	Si	Si
Gestión de tipos de usu.			-Public member -Club member -Bussiness member
Gestión de sponsors			- Manejo de “sponsorship”
Gestión del conocimiento			
Gestión de geolocaliza.	- Posibilidad de grabar trazados (deportes de ruta)	- Localización de eventos	- Geolocalización de actividad deportiva cercana a un punto
Soporte móvil	-Android -IOS		
Conexión con otros SNS	-Facebook		

Cuadro 16: Comparacion de redes, parte 4

Fuente: Autores.

Fun\Red social	Huddlers	Yoyde	Timpik
Gestión de foros			
Gestión de encuentros deportivos	- Organización de eventos deportivos	- Manejo de eventos deportivos	- Manejo de eventos deportivos
Gestión de grupos			
Manejo de torneos		Si	
Difusión info. Deportiva		- Manejo de blogs	
Serv. self-expression		-Manejo de “muro”	- Manejo de “muro” -Gestión de mensajería
Sistema estadístico			
Gestión de transversales			
Servicios deportivos			
Soporte multi-deporte	Si	Si	Si
Gestión de tipos de usu.		-Club deportivo -Deportista	- Manejo de perfil deportivo
Gestión de sponsors			
Gestión del conocimiento			
Gestión de geocaliza.	- Funcionalidad “jugando en”	- Manejo de escenarios deportivos - Manejo de “rutas”	
Soporte móvil	-IOS		-Android
Conexión con otros SNS			

Cuadro 17: Comparacion de redes, parte 5

Fuente: Autores.

Fun\Red social	Socialsports	Strava	Ineftos
Gestión de foros			Si
Gestión de encuentros deportivos	- Organizador de eventos deportivos	- Manejo de desafíos (challenges)	- Organización de eventos
Gestión de grupos			Si
Manejo de torneos			
Difusión info. Deportiva			- Manejo de blogs para estudiantes
Serv. self-expression	- Manejo de multimedia		- Manejo de mensajería - Manejo de “muro” - Manejo de multimedia
Sistema estadístico		- Gestión de estadísticas del atleta - Gestión de “follows” a otros deportistas para comparación de estadísticas (competencia)	- Utiliza mecanismo de encuestas para autorregularse - Gestión de foros: Estadísticas de foro
Gestión de transversales			
Servicios deportivos	- Evaluación de la comunidad sobre los prestadores de servicio		
Soporte multi-deporte	Si	Monodeporte (ciclomontañismo)	Si
Gestión de tipos de usu.	- Manejo de perfil de deportista (deportes practicados, lugares frecuentados, horarios frecuentados) - Manejo de usuarios (prestadores de servicio y deportistas)		- Manejo de usuarios (profesores, alumnos, entidades sin ánimo de lucro)
Gestión de sponsors			
Gestión del conocimiento		- Encuentro de consejos deportivos	- “Social learning”
Gestión de geolocaliza.		- Gestión de trazados logrados - Gestión de trazados	
Soporte móvil		-Android	
Conexión con otros SNS			

Capítulo 6

Marco legal

En cuanto al trabajo con datos, las leyes creadas en Colombia para la protección y manejo de estos son:

- Constitución Nacional, en el artículo 15, del derecho a la intimidad y al buen nombre. Debido al carácter que tiene un SNS, es posible que una persona pueda degradar el buen nombre de otra, así como violar su intimidad.
- Código penal, artículos 220 y 221, acerca de la injuria y la calumnia. Similar al anterior punto, casos de injuria y calumnia corresponden a una mala utilización de un SNS, en aras de provocar daño a otra persona.
- Ley 23 de 1982, de los derechos de autor. Debido a los espacios de difusión de información que se presentan en un SNS, esta ley entra a escena. Además, las normas de derechos de autor también califican dentro de cualquier trabajo de software que se haga.
- Ley 527 de 1999, la cual reglamenta el manejo de mercancías en el comercio electrónico, la utilización de firmas digitales, la reglamentación para certificados expedidos de forma electrónica con firma digital, el manejo de los mensajes de datos y las disposiciones de la Superintendencia de Industria y Comercio. Debido a que este SNS puede llegar a ser utilizado con carácter comercial, esta ley aplica en la creación del SNS.

- Ley 663 de 2000, artículo 91, de las obligaciones de presentar registro mercantil para los sitios web que tengan carácter comercial. De igual manera que el anterior, algunas de las funcionalidades que se pudieran llegar a implementar, pudiesen llegar a ser con un carácter netamente comercial, por lo que es necesario estar al tanto de esta ley y artículo en particular.
- Ley 1266 de 2008, el cual reglamenta el tratamiento de datos personales en bases de datos personales, haciendo énfasis en las financieras y comerciales. Debido a que los SNS manejan datos personales de sus usuarios y, a su vez, puede llegar a incluir funcionalidades con carácter comercial, es debido tener en cuenta la ley.
- Ley 1273 de 2009, la cual reglamenta el uso de la información y los sistemas de información en contra de la violación de la confidencialidad, la integridad y la disponibilidad de los datos y los sistemas de información, así como también hurtos informáticos. Debido a que ésta ley reglamenta la utilización de la información que tengan en su poder los manejadores de un sistema informático, es menester conocer dicha ley.
- Ley 1480 de 2011, la cual reglamenta los derechos y deberes tanto de consumidores como de productores en todos los sectores económicos, aplicándose ésta a los productos tanto importados como nacionales. Debido a que el SNS puede tener funcionalidades de carácter comercial, es necesario saber que leyes aplican a consumidores y productores o prestadores de servicio que la pudieren utilizar.
- Resolución 3066 de 2011, la cual busca proteger los derechos de los usuarios de servicios de comunicaciones en los cuales se establece también los derechos sobre los servicios adquiridos en telecomunicaciones. Debido a que el usuario del SNS pudiera llegar a obtener servicios en telecomunicaciones (por ejemplo, la conexión del SNS con la línea telefónica para algún tipo de acción), es necesario conocerla para saber cuando si y cuando no se debe juzgar una posible violación de los derechos de los usuarios de parte de los creadores y de aquellos que pudieran llegar a mantener el SNS.
- Ley 1581 de 2012, de disposiciones acerca del tratamiento de datos personales, haciendo énfasis en las políticas de manejo y en las audi-

torías. Debido a que el SNS tratará con algunos datos personales de sus usuarios, se debe prestar atención a que políticas de manejo de datos personales existen, una de ellas, en esta ley.

- Decreto 1377 de 2013, el cual dictamina las políticas de protección y tratamiento de datos personales. Igual que el anterior, se hace necesario conocer que decretos dictaminan la utilización de datos personales por parte de los usuarios del SNS.

Lo anterior es tomado de *Legislación en internet* (Balanta, 2014).

Además, se deben tener en cuenta las condiciones de servicio que Google ha impuesto para las aplicaciones desarrolladas para Android, así como también las condiciones aplicadas a la utilización de dichas aplicaciones. Cada una de estas políticas desarrolladas para Android involucra el análisis del entorno legal que enmarca al SNS, debido a que la tecnología a utilizar será principalmente basa en Android. Entonces, se han de tener en cuenta las siguientes condiciones de servicio:

- Google Play Terms of Service, el cual dictamina las pautas de uso de Google Play por parte del usuario final, así como también las facultades que tiene Google sobre la información y las aplicaciones instaladas en el dispositivo de un usuario.
- Developer Distribution Agreement, el cual reglamenta el uso que el desarrollador o distribuidor de aplicaciones hace de Google Play. Habla acerca del licenciamiento, el manejo de precios y pagos, el manejo de marcas y publicidad y la dada de baja de las aplicaciones de Google Play.
- Google Play Business and Program Policies, el cual reglamenta cómo deben ser utilizadas las aplicaciones en cuanto a la información publicada en las mismas y además quien puede utilizar Google Play. Además, reglamenta la devolución, compra, descarga y soporte de productos (aplicaciones) en Google Play.
- Developer Content Policy, el cual establece las políticas de contenido y publicidad que puede poner un desarrollador en sus aplicaciones.

Capítulo 7

Alcances y limitaciones

7.1. Alcances y limitaciones

7.1.1. Alcances

Este proyecto pretende diseñar e implementar un prototipo de SNS orientado al deporte bajo dispositivos móviles ANDROID, utilizando una arquitectura orientada a servicios que facilite el desarrollo y la interoperabilidad con diferentes sistemas que existan actualmente en el mercado. Para esto, se utilizará un entorno de desarrollo que brinda Android a los desarrolladores en conjunto con los diferentes dispositivos disponibles para el desarrollo del proyecto.

Debido a la escogencia de tecnologías móviles para el desarrollo del trabajo, se ha decidido incluir funcionalidades de geolocalización y demás de las que dependa ésta. Las funcionalidades de que utilizan el componente de geolocalización serán:

- Ubicación de lugares deportivos por parte de usuarios del SNS
- Cercanía a eventos deportivos por parte de un usuario del SNS
- Cercanía entre usuarios del SNS que compartan una relación (sea simétrica o asimétrica)

Otras funcionalidad que se hace interesante (y que será implementada) a la hora de revisar los hallazgos en otras redes sociales, son los reportes estadísticos sobre densidad de población ubicada en cierto espacio deportivo en cada hora del día.

Una última funcionalidad que, para un “usuario deportista” de la red social deportiva, sería muy atractiva es aquella que maneje contenidos de salud y una base de conocimiento de los deportes a implementar sobre la base de datos.

En cuanto a los deportes, se ha decidido realizar (en la etapa de análisis), encuestas a deportistas para averiguar que deportes pueden ser los candidatos a implementar sobre el SNS a desarrollar, teniendo como pauta la siguiente aseveración: Los deportes, resultado de la encuesta, elegidos, serán aquellos que en su participación sean los de menor población practicante.

7.1.2. Limitaciones

Entre las diferentes limitaciones que se pueden encontrar en el desarrollo del actual proyecto, se encuentran las siguientes:

- **Disponibilidad de dispositivos de prueba:** Ya que en el mercado existe una gran cantidad de dispositivos móviles, todos con diferentes especificaciones, es imposible garantizar que la aplicación a diseñar sea soportada por todos los dispositivos del mercado. Sin embargo, se tienen diferentes dispositivos, entre tablets y celulares, en donde se pueden realizar las pruebas (referenciados en los recursos de hardware, capítulo 11), limitando los dispositivos soportados oficialmente por el prototipo.
- **Disponibilidad de equipos a usar como servidores:** Ya que el proyecto se basa en la creación de un prototipo, se utilizarán los computadores personales disponibles para proveer los servidores que se necesiten, limitando el rendimiento que de los mismos.
- **Recolección de información:** La búsqueda de información se hará sobre la ciudad de Bogotá, haciendo énfasis en la comunidad universitaria.
- **Utilización de software libre y con fines académicos:** Será utilizado, en su mayoría, software libre para la realización del proyecto,

así como también software que preste licencia con fines académicos, debido a que no se cuenta con el presupuesto necesario para probar herramientas privativas (a parte de versiones de prueba) que pudieran llegar a ser mejores que sus homólogos libres.

- **Etapas del ciclo de vida del software no contempladas:** No se llevará acabo una etapa de implantación del software debido a que éste prototipo, aunque funcional, no estará direccionado de inmediato al mercado próximo ya que, debido a las limitaciones de tiempo de los autores, no será posible implementar todos los requerimientos no funcionales que se llegaran a dar al SNS. Por supuesto, al no haber una etapa de implantación, para este trabajo tampoco será presentada la etapa de mantenimiento.

Capítulo 8

Metodología

Equipo SCRUM:

- Product Owner
Doctor Carlos Enrique Montenegro
- Developement Team
 - Nicolás Mauricio García Garzón
 - Luis Felipe Gonzalez Moreno
- Scrum Master
Profesor Alejandro Daza

Actividades de cada Sprint (15 días máximo):

- Sprint Planning (4 horas máximo)
Se planea una reunión con el Scrum Master al inicio de cada iteración. En esta reunión se discutirá el desempeño del Sprint anterior, los servicios/funcionalidades que hagan falta para completar el prototipo, y que servicios deben ser realizados como parte de la actual iteración.
 - ¿Qué puede hacerse en este sprint?
Con base a los servicios y casos de uso de negocio que han sido encontrado, se determina que puede ser desarrollado por el developement team en la próxima iteración del prototipo.

- ¿Como se llevará a cabo este trabajo?
Se dividen los servicios propuestos como parte del Sprint entre los desarrolladores y se comparten las expectativas que debe cumplir cada servicio para que sea aceptado en el desarrollo.
- Daily Scrum (15 minutos máximo)[Solo participa el development team]
Partiendo de los items asignados a cada desarrollador, se dividen en tareas aún mas pequeñas que, de cumplirse satisfactoriamente, sirven como ruta para cumplir con el Sprint Goal. Se responden las siguientes preguntas para llevar un seguimiento continuo del desarrollo del prototipo:
 - ¿Qué se hizo ayer que ayudó al development team para cumplir el Sprint goal?
 - ¿Qué se va a hacer hoy para ayudar al development team a cumplir el Sprint goal?
 - ¿Hay algún impedimento para que el development team cumpla el Sprint goal?
- Sprint Review (2 horas máximo)
Se realiza al final de cada Sprint. La idea de esta actividad es mostrar que se hizo en el sprint con respecto a las tareas propuestas desde el inicio del mismo. Se determina si se alcanzó el Sprint goal y se discute que servicios/tareas proponer para el siguiente sprint. Las actividades básicas son:
 - Se socializa la experiencia en el sprint, que problemas ocurrieron y cómo se solucionaron.
 - Se exponen los diferentes elementos que fueron construidos y se resuelven preguntas acerca de los mismos.
 - Se propone que puede hacerse en el siguiente Sprint basados en la experiencia del actual.
 - Se revisa como el cambio en el entorno puede cambiar las prioridades en el trabajo del equipo.
- Sprint Retrospective
 - Se toman las experiencias del actual sprint para formular sugerencias que ayuden a mejorar los sprint futuros

A continuación, se presenta un product backlog inicial y tentativo para iniciar el proyecto.

Cuadro 18: Product backlog inicial

Tarea	Días	Condición de aprobación
Levantamiento de requerimientos	2	Satisfacción de todos los requerimientos para la red social
Definición de requerimientos funcionales y no funcionales	2	Modularización y descripción total de todos los requerimientos
Investigación de tecnologías existentes	6	Escogencia de las tecnologías a utilizar para implementar la red social
Diseño de casos de uso	5	Cubrimiento de todos los requerimientos identificados
Refinamiento de requerimientos	1	Trazabilidad entre casos de uso y requerimientos
Identificación de servicios candidatos	1	Cubrimiento de todos los requerimientos identificados
Diseño de blueprints de servicios	4	Cubrimiento de todos los servicios candidatos
Escogencia de servicios a ser implementados	1	Viabilidad de un prototipo funcional
Composición estática de servicios	5	Concordancia entre los casos de uso y la composición de servicios
Diseño de diagrama de clases	5	Estructuración completa de los servicios a ser implementados y concordancia con requerimientos no funcionales
Diseño de base de datos	6	Diseño que cubra los servicios a ser implementados y requerimientos no funcionales
Diseño de interfaz gráfica de usuario	4	Cubrimiento de los servicios y casos de uso a ser implementados a ser implementados
Refinamiento de casos de uso	1	Trazabilidad
Refinamiento de blueprints de servicios	1	Trazabilidad
Refinamiento de servicios a ser implementados	1	Trazabilidad
Refinamiento de composición estática de servicios	1	Trazabilidad
Construcción de la interfaz de usuario y refinamiento de interfaz de usuario	20	Navegabilidad sobre las funcionalidades a ser implementadas
Construcción de los servicios a ser implementados y refinamiento de modelos	35	Construcción de prototipo funcional sin fallas detectadas en tiempo de desarrollo
Pruebas del prototipo por parte del equipo de desarrollo	2	Prototipo sin fallas detectables en sus funcionalidades
Pruebas del prototipo por parte del usuario final	7	Prototipo aceptado por el usuario final en al menos un 70 %

Capítulo 9

Propuesta de arquitectura

La arquitectura propuesta inicialmente para el desarrollo del proyecto, representada en la figura 26, está basada en una arquitectura cliente-servidor, en la que los clientes (dispositivos android) se comunican con un servidor central, encargado de direccionar la petición del usuario al servidor que ofrezca el servicio solicitado. A su vez, se tiene una capa de persistencia donde se llevará registro de la información del sistema y de los usuarios.

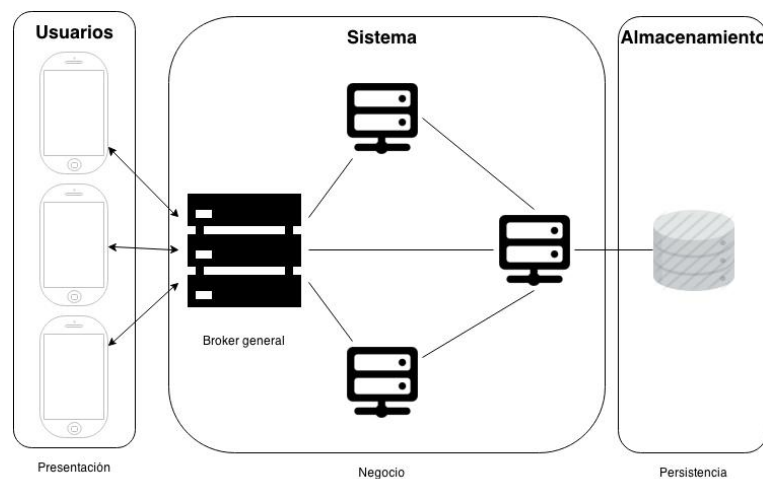


Figura 26: Propuesta de arquitectura

Fuente: Autores.

Capítulo 10

Cronograma

A continuación, se presenta un estimado de las diferentes actividades a desarrollar en el proyecto, con su respectiva duración estimada.

Cuadro 19: Actividades generales a llevar a cabo

Tarea	Duración	Inicia	Termina
Formalización de la investigación	11 Días	2 de junio de 2014	14 de junio de 2014
Sprint inicial	12 Días	16 de junio de 2014	1 de julio de 2014
Primer Sprint intermedio	10 Días	2 de julio de 2014	15 de julio de 2014
Segundo Sprint intermedio	13 Días	16 de julio de 2014	1 de agosto de 2014
Tercer Sprint intermedio	11 Días	2 de agosto de 2014	15 de agosto de 2014
Cuarto Sprint intermedio	12 Días	16 de agosto de 2014	1 de septiembre de 2014
Quinto Sprint intermedio	10 Días	2 de septiembre de 2014	15 de septiembre de 2014
Sexto Sprint intermedio	12 Días	16 de septiembre de 2014	1 de octubre de 2014
Séptimo Sprint intermedio	10 Días	2 de octubre de 2014	15 de octubre de 2014
Octavo Sprint intermedio	13 Días	16 de octubre de 2014	1 de noviembre de 2014
Noveno Sprint intermedio	12 Días	2 de noviembre de 2014	15 de noviembre de 2014
Sprint final	12 Días	17 de noviembre de 2014	2 de diciembre de 2014

Cabe resaltar que al inicio de cada Sprint realiza un Sprint Planning, y al final un Sprint review.

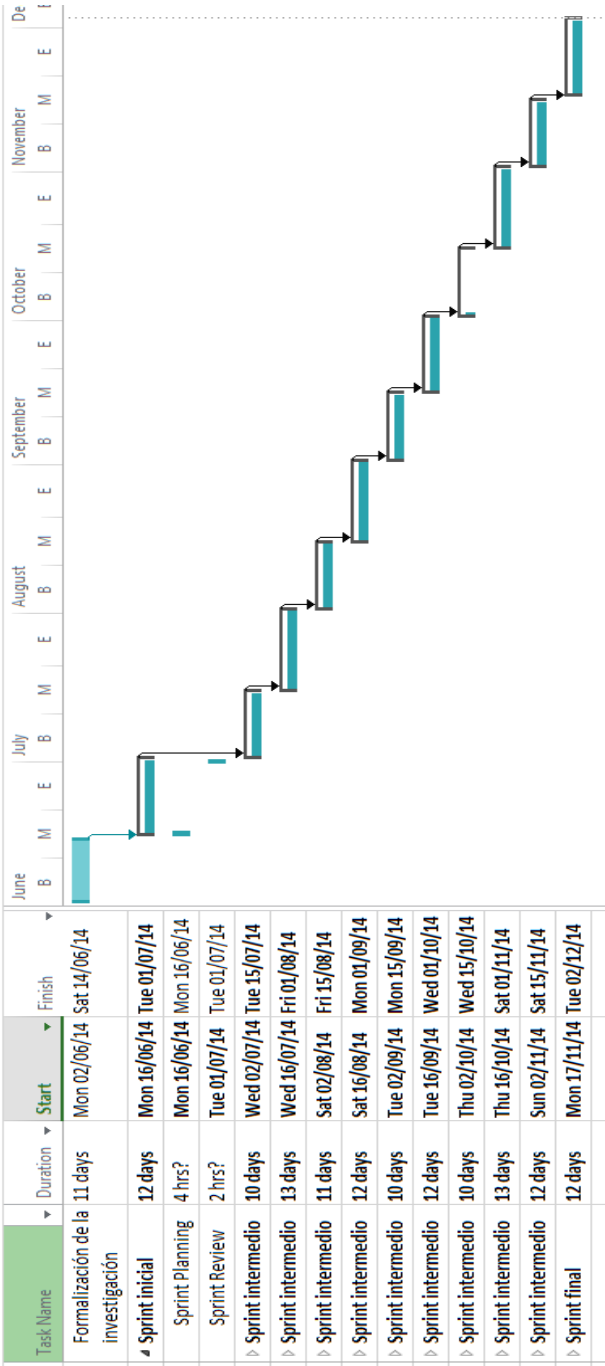


Figura 27: Distribución de las diferentes actividades a realizar

Capítulo 11

Costos

11.1. Recursos de Hardware

En la tabla 20 se muestran los costos estimados en los que se incurrirá para el desarrollo del proyecto con respecto a recursos de hardware.

11.2. Recursos de Software

En la tabla 21 se muestran los costos estimados en los que se incurrirá para el desarrollo del proyecto con respecto a recursos de software.

11.3. Recursos humanos

En la tabla 22 se muestran los costos estimados en los que se incurrirá para el desarrollo del proyecto con respecto a recursos humanos.

11.4. Recursos misceláneos

En la tabla 23 se muestran los costos estimados en los que se incurrirá para el desarrollo del proyecto con respecto a recursos de misceláneos.

Cuadro 20: Recursos de hardware

Recurso	Descripción	Cantidad	Costo Unitario	Total
Computador	Core i7 2670QM, 1TB de Disco Duro, 10GB de memoria RAM	1	\$145.000/mes	\$870.000
Tablet	Google Nexus 10	1	\$75.000/mes	\$450.000
Smartphone	Samsung Galaxy S3 Mini	1	\$40.000/mes	\$240.000
Smartphone	Huawei Y300-0151	1	\$30.000/mes	\$180.000
Tablet	QBEX S7916E, procesador 1.0GHz 1 núcleo, 512MB memoria interna, 16GB sd-card, 512MB RAM	1	\$55.000/mes	\$330.000
Tablet	Imitación Galaxy Tab GT-P1000, procesador 1.0GHz, 512MB memoria interna, 2GB sd card, 512MB de memoria RAM	1	\$55.000/mes	\$330.000
Computador	Intel Pentium G2020 @ 2.90GHz 2 núcleos, 320GB de disco duro, 4GB de memoria RAM	1	\$120.000/mes	\$720.000

Fuente: Cotización con empresa **RentaSistemas** - www.rentasistemas.com

Cuadro 21: Recursos de software

Recurso	Descripción
Debian Versión 7.4 (wheezy), 32-bit	Sistema operativo en el que se realizará el desarrollo.
Eclipse IDE	Entorno de desarrollo de código abierto
Android Development Tools (ADT)	Plugin para el IDE Eclipse que brinda un entorno de desarrollo para construir aplicaciones Android
Archi	Herramienta libre y gratuita para crear modelos en el estándar Archimate

Nota: Los costos incurridos en instalación, configuración o capacitaciones están cubiertos en el salario del development team.

Cuadro 22: Recursos humanos

Persona	Cargo	Salario mensual	Total
Nicolás Mauricio García Garzon	Ingeniero miembro del development team	\$1'848.000/mes	\$11'088.000
Luis Felipe Gonzalez Moreno	Ingeniero miembro del development team	\$1'848.000/mes	\$11'088.000
Doctor Carlos Enrique Montenegro	Director de tesis*	\$2'464.000/mes	\$14'784.000
Profesor Alejandro Paolo Daza	Co-director de tesis*	\$2'464.000/mes	\$14'784.000

(ACIEM, 2008)

*No tienen dedicación completa para el desarrollo del proyecto, por lo que se contabiliza el 50 % del salario únicamente.

Cuadro 23: Recursos misceláneos

Concepto	Cantidad	Valor mensual	Valor total
Papelería	N/A	\$10.000	\$60.000
Servicios Públicos	3	\$180.000	\$1'080.000
Transporte	2 personas	\$144.000	\$864.000

11.5. Costos totales

Cuadro 24: Costos totales

Concepto	Financiación	Valor
Hardware	Propia	\$3'120.000
Humanos (Director/co-Director)	Universidad Distrital	\$29'568.000
Humanos (Developement team)	Propia	\$22'176.000
Misceláneos	Propia	\$2'004.000
Sub-Total		\$55'064.400
Otros (20 %)	Propia	\$11'012.880
Total		\$66'077.280

Referencias

ACIEM. (2008). *Manual de referencia de tarifas aciem (asociacion colombiana de ingenieros)*. Descargado de http://issuu.com/comisionesaciem/docs/doc_manual_ref_tarifas/8?e=1622024/3914555

Allamanis, M., Scellato, S., y Moscolo, C. (2012). Evolution of a location-based online social network: Analysis and models. *IMC 12*.

Allweyer, T. (2010). *Bpmn 2.0 introduction to the standard for business process modeling*. BoD.

Arnaboldi, V., Conti, M., y Passarella, A. (s.f.). Dynamics of personal social relationships in online social networks: a study on twitter.

Balanta, H. (2014). *Legislación en internet*. Descargado de http://www.unipymes.com/diplomado-gerencia-marketing/presentaciones/modulo5_presentacion4.pdf

Companies, T. M.-H. (2014). *How to judge the reliability of internet information*. Descargado de <http://www.mhhe.com/mayfieldpub/webtutor/judging.htm>

Elahi, N., y Karlsen, R. (2012). User behavior in online social networks and its implications: A user study. *WIMS'12*.

Erl, T. (2007). *Soa principles of services design*. Prentice Hall.

Kadushin, C. (2004). *Introduction to social network theory*.

Lozada, P. (2014). *Evolución de la web*. Descargado de <http://profesores.elo.utfsm.cl/~tarredondo/info/networks/Evolucion.Web.pdf>

- Sapuppo, A. (2010). Spiderweb: A social mobile network. *2010 European Wireless Conference*.
- Schwaber, K., y Sutherland, J. (2013). *The scrum guide*. Descargado de <https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide.pdf>
- Socievole, A., y Marano, S. (2012). Exploring user sociocentric and egocentric behaviors in online and detected social networks. *2012 2nd Baltic Congress on Future Internet Communications*.
- Tsvetovat, M., y Kouznetsov, A. (2011). *Social network analysis for startups*. O'Reilly.
- Väänänen-Vainio-Mattila, K., Wäljas, M., Ojala1, J., y Segerståhl, K. (2010). Identifying drivers and hindrances of social user experience in web services. *CHI 2010: Finding Your Mojo and Doing Some Good*.
- Yang, Y., Guo, Y., y Ma, Y. (2010). Analysis on community characteristics of online social network. *2010 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*.