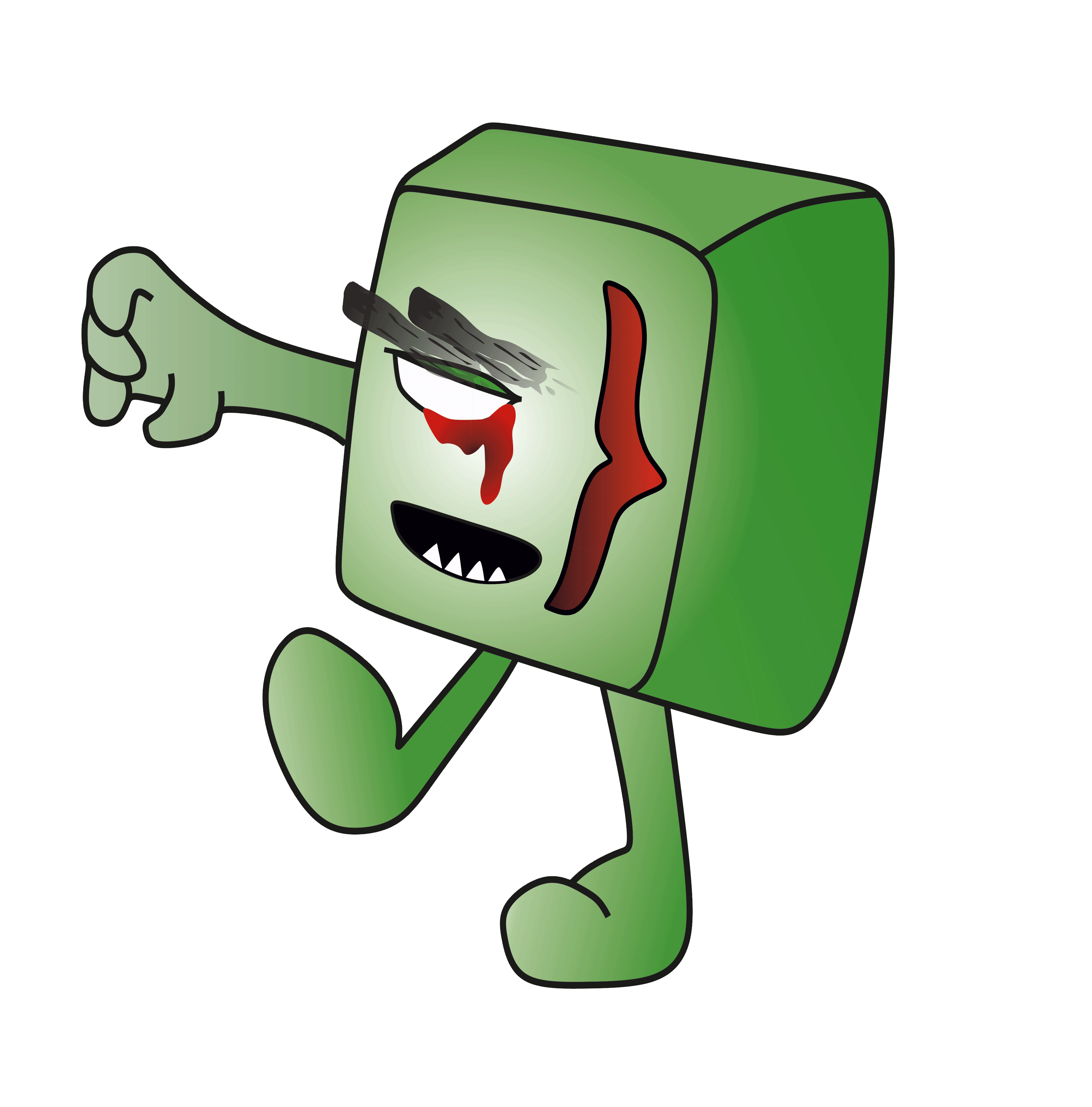
**SRS**



Alejandra Chacón

Alejandra Guerrero

María José Mendoza

Oscar Moreno

Juan Pablo Rodríguez

Miguel Urquijo

Camilo Zamora

**The**

**Walking**

**Dev**

# HISTORIAL DE CAMBIOS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VERSIÓN | FECHA | SECCIÓN DEL DOCUMENTO | DESCRIPCIÓN DEL CAMBIO | RESPONSABLES |
| 1.0 | 21/04/2016 | 2.1 | Se agregaron las secciones al documento | Scrum Master |
| 2.0 | 24/04/2016 | 1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7 | Se agregaron las secciones al documento | Scrum Master |
| 2.1 | 24/04/2016 | 4.1, 4.2 | Se agregaron las secciones al documento | Configuración y Calidad |
| 2.2 | 24/04/2016 | 2.5 | Se agregaron las secciones al documento | Scrum Master |
| 2.3 | 25/05/2016 | 4.3 | Se agregaron las secciones al documento | Configuración y Calidad |

# CONTENIDO

[HISTORIAL DE CAMBIOS 1](#_Toc449283518)

[CONTENIDO 2](#_Toc449283519)

[LISTA DE TABLAS 4](#_Toc449283520)

[LISTA DE ILUSTRACIONES 5](#_Toc449283521)

[1. INTRODUCCIÓN 6](#_Toc449283522)

[1.1 PROPÓSITO 6](#_Toc449283523)

[1.2 ALCANCE 7](#_Toc449283524)

[1.3 DEFINICIONES, ACRÓNIMOS Y ABREVACIONES 9](#_Toc449283525)

[1.4 REFERENCIAS 12](#_Toc449283526)

[1.5 APRECIACIÓN GLOBAL 15](#_Toc449283527)

[2. DESCRIPCIÓN GLOBAL 17](#_Toc449283528)

[2.1 PERSPECTIVA DEL PRODUCTO 17](#_Toc449283529)

[2.1.1 INTERFACES CON EL SISTEMA 17](#_Toc449283530)

[2.1.2 INTERFACES CON EL USUARIO 17](#_Toc449283531)

[2.1.3 INTERFACES CON EL HARDWARE 18](#_Toc449283532)

[2.1.4 INTERFACES CON EL SOFTWARE 0](#_Toc449283533)

[2.1.5 INTERFACES DE COMUNICACIÓN 0](#_Toc449283534)

[2.1.6 RESTRICCIONES DE MEMORIA 0](#_Toc449283535)

[2.1.7 OPERACIONES 0](#_Toc449283536)

[2.1.8 REQUERIMIENTOS DE ADAPTACIÓN DEL SITIO 1](#_Toc449283537)

[2.2 FUNCIONES DEL PRODUCTO 1](#_Toc449283538)

[2.3 CARACTERÍSTICAS DEL USUARIO 1](#_Toc449283539)

[2.4 RESTRICCIONES 4](#_Toc449283540)

[2.5 MODELO DEL DOMINIO 6](#_Toc449283541)

[2.6 SUPOSICIONES Y DEPENDENCIAS 7](#_Toc449283542)

[2.7 DISTRIBUCIÓN DE REQUERIMIENTOS 8](#_Toc449283543)

[3. REQUERIMIENTOS ESPECÍFICOS 11](#_Toc449283544)

[3.1 REQUERIMIENTOS DE INTERFACES EXTERNAS 11](#_Toc449283545)

[3.1.1 INTERFACES CON EL USUARIO 11](#_Toc449283546)

[3.1.2 INTERFACES CON EL HARDWARE 11](#_Toc449283547)

[3.1.3 INTERFACES CON EL SOFTWARE 11](#_Toc449283548)

[3.1.4 INTERFACES DE COMUNICACIONES 11](#_Toc449283549)

[3.2 CARACTERÍSTICAS DEL PRODUCTO DE SOFTWARE 11](#_Toc449283550)

[3.3 REQUERIMIENTOS DE DESEMPEÑO 11](#_Toc449283551)

[3.4 RESTRICCIONES DE DISEÑO 11](#_Toc449283552)

[3.5 ATRIBUTOS DEL SISTEMA DE SOFTWARE (NO FUNCIONALES) 11](#_Toc449283553)

[3.6 REQUERIMIENTOS DE LA BASE DE DATOS 11](#_Toc449283554)

[4. PROCESO DE INGENIERÍA DE REQUERIMIENTOS 11](#_Toc449283555)

[4.1 PROCESOS DE ADMINISTRACIÓN Y CONTROL DE REQUERIMIENTOS 11](#_Toc449283556)

[4.1.1 ADQUISICIÓN Y ANÁLISIS DE REQUERIMIENTOS 11](#_Toc449283557)

[4.1.2 VALIDACIÓN DE REQUERIMIENTOS 14](#_Toc449283558)

[4.1.3 ADMINISTRACIÓN DE REQUERIMIENTOS 14](#_Toc449283559)

[4.2 MECANISMOS DE PRIORIZACIÓN Y TRAZABILIDAD 15](#_Toc449283560)

[4.2.1 PRIORIZACIÓN 15](#_Toc449283561)

[4.2.2 TRAZABILIDAD 17](#_Toc449283562)

[4.3 PROCESO DE CONSTRUCCIÓN DEL SRS 17](#_Toc449283563)

[5. PROCESO DE VERIFICACIÓN 18](#_Toc449283564)

[6. ANEXOS 18](#_Toc449283565)

# LISTA DE TABLAS

[Tabla 1. Acrónimos 10](#_Toc449283566)

[Tabla 2. Definiciones 11](#_Toc449283567)

[Tabla 3. Interfaces con el software 2](#_Toc449283568)

[Tabla 4. Usuarios 4](#_Toc449283569)

[Tabla 5. Documentación del modelo de dominio 7](#_Toc449283570)

# LISTA DE ILUSTRACIONES

[Ilustración 1. Propósito 7](#_Toc449283571)

[Ilustración 2. Descripción 8](#_Toc449283572)

[Ilustración 3. Apreciación Global [3] 16](#_Toc449283573)

[Ilustración 4. Interfaces con el usuario 17](#_Toc449283574)

[Ilustración 5. Actores 2](#_Toc449283575)

[Ilustración 6. Restricciones de usuario 4](#_Toc449283576)

[Ilustración 7. Distribución de requerimientos 9](file:///C:\Users\María\Documents\INGENIERIA%20DE%20SOFTWARE\SRS%20Linea%20Base%20v2.1.docx#_Toc449283577)

[Ilustración 8. Proceso de adquisición y análisis de requerimientos. 12](#_Toc449283578)

[Ilustración 9. Administración del cambio de requerimientos. 14](#_Toc449283579)

# INTRODUCCIÓN

## PROPÓSITO

El presente documento captura la descripción completa sobre como la aplicación web D-EVent, destinada a la planificación de eventos, se espera que funcione. Además, se presentarán las características que tendrá el sistema, describiendo la fase inicial del desarrollo de este y su diseño.

Es importante hacer la especificación de requerimientos ya que minimiza el tiempo y el esfuerzo requeridos por los desarrolladores para alcanzar las metas deseadas y también minimiza el coste de desarrollo. También es importante ya que define como la aplicación va a interactuar con el hardware del sistema, otros programas y los usuarios en una amplia variedad de situaciones del mundo real. Se evalúan parámetros como la velocidad de funcionamiento, el tiempo de respuesta, la disponibilidad, la portabilidad, la facilidad de mantenimiento, la huella, la seguridad y la velocidad de recuperación de los eventos adversos. [1]

Se describirá todo el sistema de una manera detallada y concisa. Esta especificación va dirigida a los miembros del grupo The Walking Dev y al cliente Miguel Eduardo Torres.

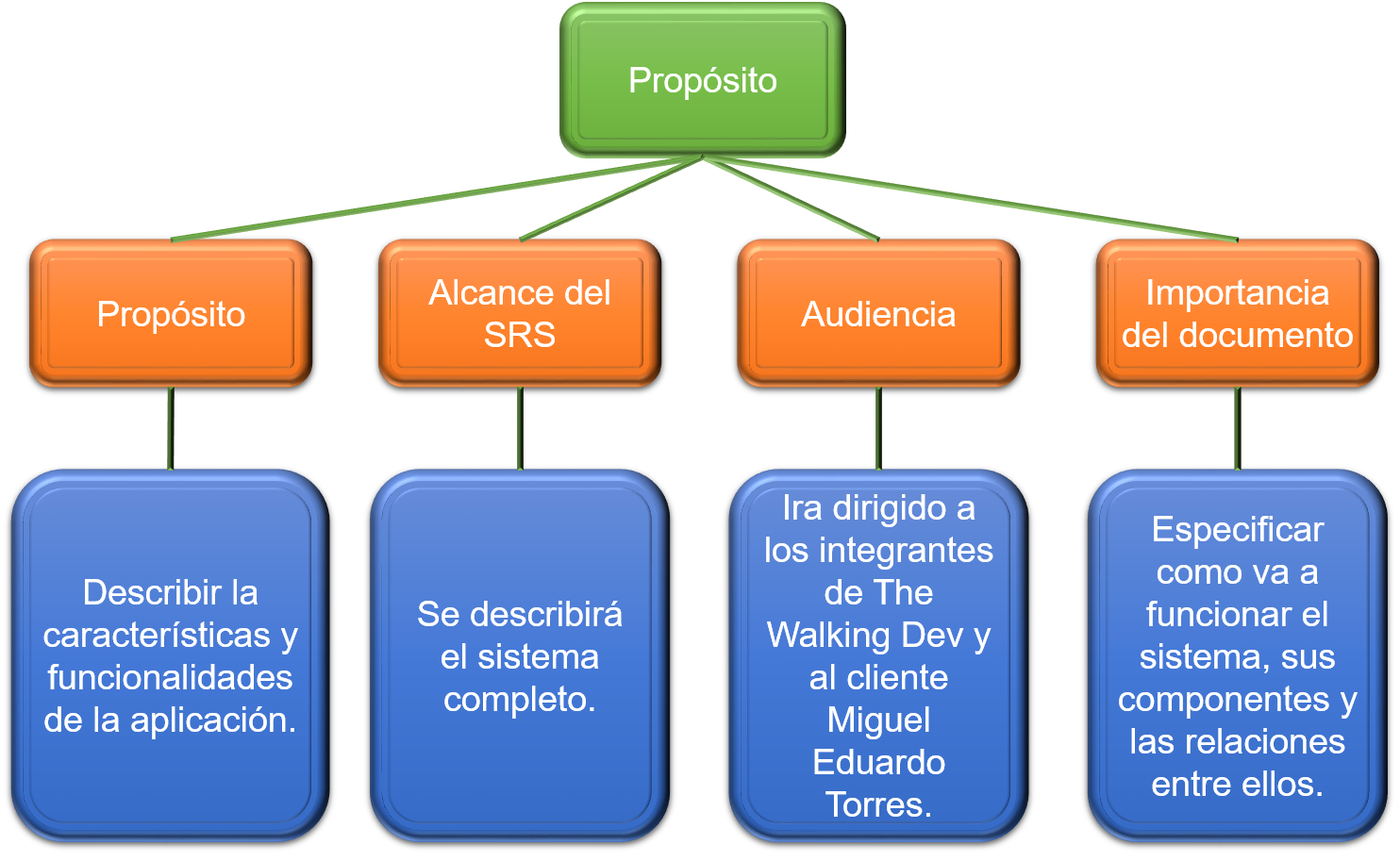


Ilustración 1. Propósito

## ALCANCE

* El producto a realizar se llama D-EVent y va dirigido a todas aquellas personas que quieren planificar un evento, para saber más sobre los usuarios [Ver Sección 2.3 Características del usuario](#_2.3_CARACTERÍSTICAS_DEL)

#### DESCRIPCIÓN

Ilustración 2. Descripción

#### BENEFICIOS

* Facilitar la planeación de eventos de escala pequeña y media.
* Tener guías predefinidas para cada tipo de evento de tal manera que el usuario pueda orientarse en la planificación del evento.
* Tener una lista de cotizaciones divididas en categorías para que el usuario pueda realizar una comparación entre los diferentes proveedores y escoger el mejor se adapte a su presupuesto y al tipo de evento.
* Manejo del presupuesto y balance de este.
* Permitir al usuario ubicar a los invitados en el plano del espacio del lugar del evento.

#### PROPÓSITO

*“Ayudar al mundo a reunirse al alcance de un clic”*

Ayudar aquellas personas que no tienen conocimiento de cómo organizar un evento, dotándolos de ciertas herramientas para lograr una buena planificación.

#### FUNCIONALIDAD

Para ver las funcionalidades que va a tener y las que no va a tener el producto Ver Anexo SPMP Sección 6.2 Alcance

#### RELACIÓN DEL CONTEXTO ORGANIZACIONAL

##### Metas corporativas

* Diseñar una aplicación web de calidad.
* Trabajar en equipo para lograr los objetivos propuestos.
* Desarrollar una aplicación que supla una necesidad real.
* Entender la importancia de seguir un plan de proyecto con el fin de cumplir las metas propuestas.

##### Estrategias de negocio

* Fomentar la buena comunicación entre el equipo de trabajo.
* Llevar a cabo las actividades planeadas en sus respectivos tiempos.
* Realizar un control de calidad riguroso.
* Usar las plantillas y los formatos establecidos de manera adecuada para estandarizar la forma en la que se consolida la información.

## DEFINICIONES, ACRÓNIMOS Y ABREVACIONES

|  |  |
| --- | --- |
| ACRONIMO | DEFINICIÓN |
| PHP | HYPERTECT PREPROCESSOR |
| SQL | STRUCTURED QUERY LANGUAGE |
| MVC | MODELO VISTA CONTROLADOR |
| CRUD | CREATE, READ, UPDATE AND DELETE |
| HTTP | HYPERTEXT TRANSFER PROTOCOL |
| TCP | TRANSMISSION CONTROL PROTOCOL |
| IP | INTERNET PROTOCOL |
| CSS | CASCADING STYLE SHEETS |
| HTML | HYPERTEXT MARKUP LANGUAGE |
| JS | JAVA SCRIPT |
| SAD | SOFTWARE ARCHITECTURE DOCUMENT |
| CU | CASOS DE USO |
| SRS | SOFTWARE REQUIREMENT SPECIFICATION |

Tabla 1. Acrónimos

|  |  |
| --- | --- |
| TÉRMINO | DEFINICIÓN |
| Aplicativo web | Herramienta que los usuarios pueden utilizar accediendo a un servidor web a través de Internet. Es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador. [2] |
| Balance de presupuesto | Resumen detallado del dinero que se planea invertir en el evento |
| Calificación | Es el valor que le asigna un usuario creador a un proveedor para expresar si es o no recomendable usar sus productos o sus servicios. |
| Carrito de compras | Es el conjunto de elementos que serán definitivamente adquiridos y consumidos por el usuario creador para la realización del evento. |
| Categoría | Conjunto de cotizaciones de un determinado tipo. Ejemplo: Alimentación (contiene cotizaciones de ese tipo). |
| Colaborador | [Ver Sección 2.3 Características del usuario](#_2.3_CARACTERÍSTICAS_DEL) |
| Comentario | Es la opinión u observación personal más detallada que un usuario creador puede hacer a un proveedor tras usar sus productos o servicios. |
| Cotización: | Es el valor que tendrá un producto o servicio de un proveedor si llegase a ser adquirido. |
| Creador | [Ver Sección 2.3 Características del usuario](#_2.3_CARACTERÍSTICAS_DEL) |
| Cuenta | Es el conjunto compuesto por el perfil de usuario, sus permisos, y eventos. |
| Estado de presupuesto | Dado un presupuesto, es el valor que este tiene tras incurrir en gastos que se evidencian en el carrito de compras. |
| Evento | Acontecimiento de importancia que se encuentra programado. |
| Gestor de contenidos | [Ver Sección 2.3 Características del usuario](#_2.3_CARACTERÍSTICAS_DEL) |
| Historial de eventos | Conjunto de eventos que se finalizaron hace menos de 3 años. |
| Invitado | [Ver Sección 2.3 Características del usuario](#_2.3_CARACTERÍSTICAS_DEL) |
| Lista de cotizaciones | Conjunto de categorías que contiene el total de cotizaciones realizadas. |
| Lista de eventos a colaborar | Conjunto de eventos no finalizados en los que el usuario cumple el rol de colaborador. |
| Lista de eventos creados | Conjunto de eventos no finalizados en los que el usuario cumple el rol de creador. |
| Lista de invitados | Conjunto de personas que se espera que asistan o participen en el evento. |
| Lista de regalos/aportes | Conjunto de elementos que se espera que lleven los invitados el día del evento. |
| Nombre de usuario | Nombre corto y único que se da a un usuario al registrarse. |
| Perfil (de usuario): | Conjunto de rasgos o atributos que caracterizan a un usuario y que se diligencian cuando este se registra. |
| Perfil del evento | Conjunto de rasgos o atributos que caracterizan a un evento y que se diligencian cuando este es creado. |
| Plano del espacio | Representación gráfica del lugar donde se llevará a cabo el evento, facilita la ubicación de elementos como mesas, sillas, decoraciones, etc. |
| Plantilla | Patrón que sirve como modelo para hacer otros eventos. Varían con la naturaleza del evento (tipo de evento, cantidad de personas, presupuesto). |
| Presupuesto | Cantidad de dinero que se dispone para un evento. |
| Recordatorio | Mensaje breve en que se recuerda la fecha y lugar de un evento. |
| Recordatorio automático | Aquel que es generado por el sistema de forma simple y que se programa para enviarse una semana antes de la fecha del evento. |
| Recordatorio personalizado | Aquel que es realizado por un usuario y que puede contener un mensaje o postal personalizada |
| Regalo/aporte | Elemento que se espera que lleven los invitados el día del evento. |
| Tarjeta | Nombre con que se designa a una actividad o tarea a realizar y que hace parte del To-Do List. |
| Tipo de evento | Los tipos de eventos que principalmente se tendrán en cuenta para el aplicativo son: Asados, cumpleaños, 15 años, fiestas infantiles, aniversarios, matrimonios, bautizos, primeras comuniones y un tipo general denominado Otros. |
| To-Do List | Conjunto de tarjetas o actividades a llevar a cabo para la planeación y organización del evento. |
| Usuario general | Usuario a punto de registrarse, registrado y que no necesariamente tiene un evento en el que cumple el rol de creador o colaborador. Esta definición solo aplica para requerimientos. |

Tabla 2. Definiciones

## REFERENCIAS

|  |  |
| --- | --- |
| [1] | M. Rouse, «TechTarget,» [En línea]. Available: http://searchsoftwarequality.techtarget.com/definition/software-requirements-specification. |
| [2] | Wikipedia, «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Aplicaci%C3%B3n\_web. |
| [3] | Iron Works, «Plantilla SRS Iron Works,» [En línea]. Available: http://uvirtual.javeriana.edu.co/bbcswebdav/pid-382553-dt-content-rid-1584302\_1/xid-1584302\_1. |
| [4] | Networking and emerging optimization, «Herramientas web para la enseñanza de protocolos de comunicación,» [En línea]. Available: http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/http.html. |
| [5] | M. A. J. N. O. Y. N. T. G. Enríquez Toledo Alma, «Grid Morelos,» [En línea]. Available: http://www.gridmorelos.uaem.mx/~mcruz/cursos/miic/MySQL.pdf . |
| [6] | «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Microsoft\_Windows. |
| [7] | L. Alegsa, «Alegsa,» [En línea]. Available: http://www.alegsa.com.ar/Dic/google%20chrome.php. |
| [8] | Wikipedia, «Laravel,» [En línea]. Available: https://es.wikipedia.org/wiki/Laravel. |
| [9] | Wikipedia, «Ubuntu,» [En línea]. Available: https://es.wikipedia.org/wiki/Ubuntu. |
| [10] | M. A. Alvarez, «Desarrollo Web,» 17 Octubre 2014. [En línea]. Available: http://www.desarrolloweb.com/articulos/composer-gestor-dependencias-para-php.html. |
| [11] | D. Orozco, «Concepto Definición,» 21 Junio 2014. [En línea]. Available: http://conceptodefinicion.de/google-chrome/. |
| [12] | «EcuRed,» [En línea]. Available: http://www.ecured.cu/Ubuntu. |
| [13] | Information Sciences Institute, «RFC - PROTOCOLO DE CONTROL DE TRANSMISIÓN,» Septiembre 1981. [En línea]. Available: https://www.rfc-es.org/rfc/rfc0793-es.txt. |
| [14] | Information Sciences Institute, «RFC - INTERNET PROTOCOL,» Septiembre 1981. [En línea]. Available: https://www.rfc-es.org/rfc/rfc0791-es.txt. |
| [15] | Digital Ocean, «Digital Ocean,» [En línea]. Available: https://www.digitalocean.com/. |
| [16] | IEEE, Systems and software engineering - Life cycle processes - Requirements engineering, 2011. |
| [17] | M. Sebely, «Qué es un web master,» 23 Junio 2001. [En línea]. Available: http://www.desarrolloweb.com/articulos/465.php. [Último acceso: 20 Abril 2016]. |
| [18] | C. Larma, UML y Patrones, Madrid: Pearson, 2003. |
| [19] | IEEE, 1233 IEEE guide for developing system requirements specifications, New York, 1996. |
| [20] | I. Sommerville, Ingeniería de Software, Naucalpan de Juárez, Estado de México: Pearson, 2011. |
| [21] | N. Carod Martinez, «Priorización de Requerimientos de Software utilizando una estrategia cognitiva,» [En línea]. Available: http://sedici.unlp.edu.ar/bitstream/handle/10915/20793/Documento\_completo.pdf?sequence=1. |
| [22] | M. Cohn, Agile Estimating and Planning, Pearson. |
| [23] | B. Berenbach, D. J. Paulish, J. Kazmeier y A. Rudorfer, Software And Systems Requirements Engineering in Practice. |

## APRECIACIÓN GLOBAL

Este documento contiene la descripción y especificación de los requerimientos y características que la aplicación D-Event tendrá, de esta manera tanto nuestro cliente como el grupo de trabajo de The Walking Dev podrán conocer el funcionamiento del sistema desarrollado.

Para lo anterior, este documento se divide en secciones las cuales tienen una finalidad específica, así, a lo largo del SRS (Software Requirements Specification) se describe una serie de procesos y planes mediante los cuales se realizó la recolección, análisis, verificación, organización y administración de requerimientos, información esencial para que el documento general cumpla su propósito.

A continuación, se describe cada una de las secciones del presente documento:

Ilustración 3. Apreciación Global [3]

# DESCRIPCIÓN GLOBAL

## PERSPECTIVA DEL PRODUCTO

D-EVent es un producto totalmente nuevo. Los usuarios se verán beneficiados con este producto ya que esta les facilitará la planeación de sus eventos sin tener que salir de sus casas ni tener que acudir a personas externas.

### INTERFACES CON EL SISTEMA

Aunque D-EVent es un producto totalmente nuevo está usará el API de Google Maps para Web, el cual nos permitirá personalizar los mapas y la información en ellos. Se usará para que el usuario pueda visualizar mediante un mapa la ubicación de los diferentes lugares que muestra la aplicación, como lo son los sitios donde se harán los eventos.

### INTERFACES CON EL USUARIO

Para que el usuario pueda interactuar con el sistema, necesitará las siguientes interfaces:

Ilustración 4. Interfaces con el usuario

### INTERFACES CON EL HARDWARE

Protocolo HTTP: El Protocolo de Transferencia de HiperTexto (Hypertext Transfer Protocol) es un sencillo protocolo cliente-servidor que nos permitirá el intercambio de información entre los usuarios de la aplicación y los servidores HTTP. [4]

Protocolo TCP/IP: debido a que el protocolo HTTP está soportado sobre los servicios de conexión TCP/IP, esté entra a ser parte de nuestras interfaces con el hardware y funciona de la siguiente manera un proceso servidor escucha en un puerto de comunicaciones TCP (por defecto, el 80), y espera las solicitudes de conexión de los clientes Web. Una vez que se establece la conexión, el protocolo TCP se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores. [4]

### INTERFACES CON EL SOFTWARE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PRODUCTO DE SOFTWARE | MYSQL | WINDOWS | GOOGLE CHROME | LARAVEL | LINUX UBUNTU | COMPOSER |
| DESCRIPCIÓN | Es un sistema de gestión de bases de datos relacional. Es un software de código abierto, licenciado bajo la GPL de la GNU. El lenguaje de programación que utiliza MySQL es Structured Query Language (SQL) [5] | Es el nombre de una familia de distribuciones desoftware para PC, smartphone, servidores y sistemas empotrados, desarrollados y vendidos por Microsoft, y disponibles para múltiples arquitecturas [6] | Google Chrome es un navegador web de código abierto desarrollado por Google. Este navegador está constituido por componentes de otros navegadores de código abierto como WebKit y Mozilla. [7] | Framework de código abierto para desarrollar aplicaciones y servicios web con PHP. [8] | Ubuntu es un sistema operativo basado en GNU/Linux y que se distribuye como software libre, el cual incluye su propio entorno de escritorio denominado Unity. [9] | Composer es un gestor de dependencias en proyectos, para programación en PHP. Eso quiere decir que nos permite gestionar (declarar, descargar y mantener actualizados) los paquetes de software en los que se basa nuestro proyecto PHP. [10] |
| PROPÓSITO DE USO | Utilizar una base de datos que sea rápida al momento de realizar operaciones, con bajo costo en requerimientos para la elaboración de bases de datos, facilidad de configuración e instalación, adémas soporta gran variedad de sistemas operativos, es baja la probabilidad de corromper los datos y tiene la ventaja de asegurar la integridad de la información [5] | Ya que este sistema operativo es el mas usado actualmente, aseguramos que llegue a la mayoría de usuarios finales | Usar un navegador rápido, seguro, practico, estable y con un sentido minimalista único, que le brinda al usuario la mayor comodidad a la hora de navegar por la web. [11] | Permite manejo sencillo de bases de datos. Además de permite el uso de una sintaxis elegante y expresiva para crear código de forma sencilla [8] | Proporciona un sistema operativo actualizado y estable para el usuario promedio, con un fuerte enfoque en la facilidad de uso y de instalación del sistema. [12] | Laravel utiliza Composer para gestionar sus dependencias |
| VERSIÓN | Versión 5.7 | Versión 8.1 a versión 10 | Versión 49.0.2623.110 m | Versión 5.2 | Versión 14.04 | Versión 1.0.0-beta2 |
| FUENTE | Documentación: http://dev.mysql.com/doc/ Descarga: http://dev.mysql.com/downloads/installer/ | Ayuda: http://windows.microsoft.com/es-es/windows/support#1TC=windows-10 | Centro de ayuda: https://support.google.com/chrome/?hl=en#topic=3227046 Descarga: https://www.google.es/chrome/browser/desktop/ | Documentación: https://laravel.com/docs/5.2 Manual de instalación: https://laravel.com/docs/4.2 | Documentación: https://help.ubuntu.com/ Descarga: http://www.ubuntu.com/download | Documentación: https://getcomposer.org/doc/ Descarga: https://getcomposer.org/download/ |

Tabla 3. Interfaces con el software

### INTERFACES DE COMUNICACIÓN

TCP: TCP es un protocolo orientado a la conexión, fiable y entre dos extremos, diseñado para encajar en una jerarquía en capas de protocolos que soportan aplicaciones sobre múltiples redes. TCP proporciona mecanismos para la comunicación fiable entre pares de procesos en computadoras 'host' ancladas en redes de comunicación de computadoras distintas, pero interconectadas. TCP presenta interfaz por un lado con el usuario o los procesos de aplicación y por el otro con un protocolo de más bajo nivel como es el protocolo de internet (IP). [13]

IP: El Protocolo Internet está diseñado para su uso en sistemas interconectados de redes de comunicación de ordenadores por intercambio de paquetes. El protocolo internet proporciona los medios necesarios para la transmisión de bloques de datos llamados datagramas desde el origen al destino, donde origen y destino son hosts identificados por direcciones de longitud fija. [14]

### RESTRICCIONES DE MEMORIA

Servidor:

El servidor que se usará es Digital Ocean. Tiene de memoria RAM 512 MB, disco de 20GB y tiene 1 core de procesador. [15]

Para que la aplicación web funcione correctamente, se necesita de mínimo 200MB de memoria RAM en el computador donde se quiera usar la aplicación.

### OPERACIONES

Se tendrán dos modos de operación:

* Administrador: quien es el encargado de la configuración general de la aplicación.
* Usuario: este será el modo para el usuario final de la aplicación.

El sistema estará activo para los procesos de administración y mantenimientos el primer lunes de cada mes desde la 1:00 am hasta las 3:00 am y los demás estará inactivo. Mientras se haga el debido mantenimiento el usuario no podrá hacer uso de la aplicación.

Si el servidor en el cual está alojada la aplicación se cae, está quedará inaccesible hasta que el servidor vuelva a funcionar correctamente y los datos que no se alcanzaron a ser agregados a la base de datos se perderán.

Se harán copias de seguridad de la base de datos cada semana para tener un medio de recuperación en caso de su pérdida. Está se guardará en el disco duro de uno de los integrantes del equipo de desarrollo.

Si la base de datos se llena no se aceptarán las siguientes transacciones y se deberá adquirir más espacio de alojamiento de datos. Y si la base de datos se bloquea la aplicación quedará inaccesible hasta que responda de nuevo la base de datos.

### REQUERIMIENTOS DE ADAPTACIÓN DEL SITIO

Cualquier usuario que tenga una versión de Windows mayor o igual a la 8.1 hasta la versión 10, y que tenga el navegador Google Chrome con versión 49.0.2623.110 m, podrá hacer uso de la aplicación web D-EVent.

## FUNCIONES DEL PRODUCTO

Esta sección provee un resumen de las funciones principales que el software desarrollará [16]. A partir del Planning Póker hecho en el Sprint 3, se filtraron los veinte casos de uso considerados más relevantes por el equipo, debido a su nivel de dificultad y su importancia para el cliente, las funcionalidades primordiales del producto son:

* La creación de eventos.
* La cotización de los recursos a utilizar en el evento con distintos proveedores.
* La ubicación de los invitados dentro de un diagrama.
* Todo el manejo del presupuesto.
* Lista de invitados y de regalos/aportes.

D-EVent tiene como factor diferenciador, la integración de varias funcionalidades o servicios que ofrece la competencia, con el fin de atrapar al usuario en un aplicativo por el cual pueda guardar y revisar todo lo necesario para la organización de eventos sin tener cuentas complementarias en otras aplicaciones. Además, da la oportunidad de involucrar a otros usuarios como colaboradores de tu evento. (Ver SPMP Sección 6.1 Visión)

## CARACTERÍSTICAS DEL USUARIO

Para el desarrollo del producto se definieron varios actores con los que el sistema debe interactuar, como se muestra en el diagrama a continuación, sin embargo, por la restricción de tiempo, se ha decidido darle mayor importancia en cuanto al modelado de los usuarios, al público establecido en la visión del producto a los cuales se les ofrecen las funcionalidades principales del mismo destacados con una estrella en el diagrama.

Ilustración 5. Actores

Cabe destacar que, a corto plazo, los usuarios son personas entre 18 y 70 años, que sepan leer y escribir en español, que no posean ningún tipo de discapacidad, interesados en la organización de eventos. A futuro se espera también incluir a personas jurídicas que deseen ofrecer servicios para la organización de evento.

Además, la aplicación debe tener a un Gestor de contenidos, el cual tiene como tarea principal administrar y actualizar toda la información de D-Event, verificando que el contenido sea válido y vigente. El Web Master será el encargado de actualizar toda la estructura del HTML, agregar widgets, mantener los dominios que utiliza el aplicativo y crear formas de contacto entre los usuarios y los proveedores. [17]

El invitado es un actor que claramente es importante en la organización de un evento dado a que son ellos los que participan en el “producto final” que es el evento en sí más no tienen que estar involucrado en todo el proceso de planeación por lo que su interacción con el aplicativo no es muy alta, limitándose a su confirmación de asistencia y de regalo/aporte.

En la siguiente tabla se especifican los privilegios, roles, el nivel de estudio o experiencia técnica y la frecuencia de uso de cada uno de los actores. La frecuencia de uso se estima a partir del uso que le da el usuario a la aplicación cuando hace parte de un evento que está activo. Alto: todos los días, Medio: de 2 a 4 días a la semana, Baja: 1 día a la semana.

Los roles están definidos de la siguiente manera [18]:

* Actor principal: tiene objetivos de usuario que satisfacen mediante el uso de los servicios.
* Actor de Apoyo: proporciona un servicio. Normalmente se trata de un sistema informático, pero podría ser una organización o persona
* Actor pasivo: está interesado en el comportamiento del caso de uso, pero no es principal ni de apoyo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USUARIO | PRIVILEGIOS | ROLES | NIVEL DE ESTUDIOS O EXPERIENCIA TÉCNICA | FRECUENCIA DE USO |
| Creador | - Manejo del presupuesto - Enviar invitación - Crear y cancelar evento - Aprobación de proveedores (Añadir al carrito de compras) - Cambiar perfil del evento \* Adicionalmente posee los privilegios del Colaborador | Actor principal | - Conocimiento básico en manejo de computadores e internet | ALTA |
| Colaborador | - Añadir invitado - Añadir proveedores a la lista de cotizaciones - Añadir elemento a la lista de regalos/aportes - Modificar la plantilla de ubicación de invitados  - Registrar, iniciar y cerrar sesión  - Añadir, modificar y eliminar elemento del To-Do List - Ver calendario | Actor principal | - Conocimiento básico en manejo de computadores e internet | ALTA |
| Invitado | - Confirmar asistencia - Escoger regalo/aporte | Actor pasivo | - Conocimiento básico en manejo de computadores e internet - Manejo de correo electrónico | BAJA |
| Gestor de contenidos | - Administrar las plantillas por evento - Administrar la información de los proveedores - Configurar la base de datos | Actor de apoyo | - Conocimiento de bases de datos | MEDIA |

Tabla 4. Usuarios

## RESTRICCIONES

Las restricciones son requerimientos que se imponen a la solución debido a ciertas circunstancias, fuerzas o coacciones; estas impiden el libre uso de herramientas y soluciones por parte del diseñador mediante la proposición de límites inamovibles [19]. A continuación, se muestra el diagrama de las restricciones de D-Event.

**RESTRICCIONES DE USUARIO**

Ilustración 6. Restricciones de usuario

**RESTRICCIONES DEL PRODUCTO**

## MODELO DEL DOMINIO

Un modelo de dominio se utiliza con frecuencia como fuente de inspiración para el diseño de los objetos de software. Es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés; es el artefacto más importante que se crea durante el análisis orientado a objetos. Una clase conceptual es una idea, cosa u objeto [18]. Para el modelo de dominio de D-EVent Ver Anexo Diagrama Modelo de Domino.

Los elementos que componen cada objeto son:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *ID* | Identificador unico del objeto | | *Elemento del Dominio* | Nombre del objeto |
| *Descripción* | Explicación detallada del objeto | | | |
| *Atributos:* Valor de datos lógico de un objeto | | | | |
| *Nombre* | | *Descripción* | | *Tipo de Dato* |
| Nombre del atributo | | Explicación breve del atributo del objeto | | Tipo de dato que se le asigna al elemento |
| *Objetivo* | Descripción global del objeto documentado donde se explica la funcionalidad que tiene este en el sistema | | | |

Tabla 5. Documentación del modelo de dominio

Para ver la documentación completa del modelo de dominio para D-EVent Ver Anexo Documentación Modelo de Dominio

## 2.6 SUPOSICIONES Y DEPENDENCIAS

Una suposición es una condición que se toma como verdadera [16] y una dependencia es una condición de la que no se tiene control, en este contexto aquellas que pueden llegar a afectar los requerimientos de D-Event son:

* D-Event correrá usando la conexión inalámbrica a internet de la Pontificia Universidad Javeriana
* Se requiere de una previa instalación de la base de datos en el servidor y su respectiva configuración
* Es necesario tener una conexión de red estable para el buen funcionamiento de la aplicación.
* Si la red de internet se cae, se debe iniciar sesión nuevamente en la página
* La aplicación funciona en todo explorador de computador de escritorio
* Se debe tener mínimo 200MB de memoria RAM para la ejecución del aplicativo web.
* Los usuarios se familiarizan rápidamente con la aplicación web y sus funcionalidades.
* Los computadores de las salas de la Facultad de Ingeniería se encuentran disponibles para el desarrollo, ejecución y pruebas del aplicativo web
* El cliente hará parte del proceso de recolección y especificación de requerimientos
* El cliente no puede solicitar cambios en los requerimientos una vez finalizado el proceso de recolección de los mismos y estén aprobados en el SRS
* El cliente no puede solicitar funcionalidades extra en esta etapa del proceso
* D-Event en su versión final será utilizado únicamente con fines académicos y no será comercializado ni distribuido por la universidad o los miembros de The Walking Dev con fines lucrativos
* Velocidad de red suficiente para tener los tiempos de respuesta de la aplicación descritos en los requerimientos.
* El servidor empleado funciona correctamente

## 2.7 DISTRIBUCIÓN DE REQUERIMIENTOS

En la siguiente ilustración se puede observar más detalladamente la distribución de los requerimientos funcionales, no funcionales e infraestructura, posteriormente encontrará su explicación.

Ilustración 7. Distribución de requerimientos

**NO FUNCIONALES**

DEL PRODUCTO

DE LA ORGANIZACIÓN

EXTERNOS

**FUNCIONALES**

**INFRAESTRUCTURA**

NEGOCIO

USUARIO

**CREADOR**

**COLABORADOR**

HARDWARE

SOFTWARE

**REQUERIMIENTOS**

Para la construcción del diagrama anterior nos basamos en las definiciones de Ingeniería de requerimientos de Ian Sommerville [20]; como se puede evidenciar, a las tres categorías principales de requerimientos se les asignó un color que se relacionará con el listado de requerimientos de D-Event.

#### 2.7.1 REQUERIMIENTOS NO FUNCIONALES:

Son requerimientos que no se relacionan directamente con los servicios específicos que el sistema entrega a sus usuarios, por el contrario, son limitaciones sobre esas funciones o servicios. Contemplan tres ítems:

* Requerimientos del producto

Especifican o restringen el comportamiento del software. Tiene en cuenta atributos como:

* + Seguridad
  + Rendimiento (velocidad de ejecución o memoria requerida)
  + Fiabilidad
  + Usabilidad
  + Persistencia
  + Mantenibilidad
  + Portabilidad
  + Comunicación
  + Interfaz
  + Hardware
* Requerimientos de la organización

Derivados de políticas y procedimientos en la organización del cliente y del desarrollador. Tiene en cuenta atributos como:

* + Implementación
  + Documentación
* Requerimientos externos

Derivados de factores externos al sistema y su proceso de desarrollo. Incluye requerimientos regulatorios, legislativos y éticos.

#### 2.7.2 REQUERIMIENTOS FUNCIONALES

Describen lo que el sistema debe hacer, cómo debe reaccionar a entradas o situaciones específicas. Comprende dos categorías: requerimientos de usuario (enunciados a cerca de los servicios que esperan los usuarios finales del sistema) y requerimientos funcionales de negocio.

Cada una de las categorías se dividirán en dos grupos según el usuario: Creador y Colaborador. Está forma de organización permite crear, modificar o eliminar requerimientos fácilmente sin afectar requerimientos de otra categoría.

#### 2.7.3 INFRAESTRUCTURA

Se compone de hardware y software. Las especificaciones de estos ítems no están explícitas en el listado de requerimientos, sino que están en secciones anteriores de este documento: [2.4 Restricciones](#_RESTRICCIONES) y [2.1.8 Requerimientos de adaptación del sitio](#_REQUERIMIENTOS_DE_ADAPTACIÓN)

# REQUERIMIENTOS ESPECÍFICOS

## 3.1 REQUERIMIENTOS DE INTERFACES EXTERNAS

### 3.1.1 INTERFACES CON EL USUARIO

### 3.1.2 INTERFACES CON EL HARDWARE

### 3.1.3 INTERFACES CON EL SOFTWARE

### 3.1.4 INTERFACES DE COMUNICACIONES

## 3.2 CARACTERÍSTICAS DEL PRODUCTO DE SOFTWARE

## 3.3 REQUERIMIENTOS DE DESEMPEÑO

## 3.4 RESTRICCIONES DE DISEÑO

## 3.5 ATRIBUTOS DEL SISTEMA DE SOFTWARE (NO FUNCIONALES)

## 3.6 REQUERIMIENTOS DE LA BASE DE DATOS

# PROCESO DE INGENIERÍA DE REQUERIMIENTOS

## 4.1 PROCESOS DE ADMINISTRACIÓN Y CONTROL DE REQUERIMIENTOS

La gestión de los requerimientos nos permitirá descubrir, analizar, documentar y verificar los servicios y restricciones que debe ofrecer nuestro sistema. De esta manera podemos definir una serie de pasos que nos guiaran por todo el proceso de administración y control de requerimientos [20]:

* Adquisición y análisis de requerimientos
* Validación de requerimientos
* Administración de requerimientos

### 4.1.1 ADQUISICIÓN Y ANÁLISIS DE REQUERIMIENTOS

Esta es la primera actividad del proceso de ingeniería de requerimientos y su desarrollo se llevará a cabo en tres pasos: descubrimiento de requerimientos, priorización de requerimientos y especificación de requerimientos:

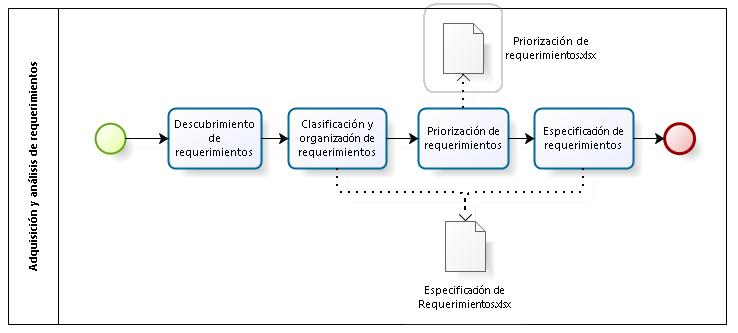


Ilustración 8. Proceso de adquisición y análisis de requerimientos.

1. Descubrimiento de requerimientos: Éste es el proceso de interactuar con los posibles usuarios del sistema y nuestro cliente final, para así poder descubrir sus requerimientos [20]. Para la obtención de los requerimientos funcionales de nuestro sistema nos basamos en los casos de uso ya que estos representan el flujo de acciones de las funcionalidades de nuestro aplicativo web. Por otro lado, los requerimientos no funcionales fueron obtenidos mediante reuniones de trabajo en las cuales los integrantes de The Walking Dev proponían candidatos a este tipo de requerimientos y entre todos se decidía cuales se tomaban y cuáles no.
2. Clasificación y organización de requerimientos: En esta parte del proceso se organizan los requerimientos en categorías para facilitar su identificación y su trazabilidad, estas categorías están especificadas en la sección [2.7 Distribución de Requerimientos](#_2.7_DISTRIBUCIÓN_DE). La organización de los requerimientos se lleva a cabo en el documento de Especificación de Requerimientos

Ver Anexo: Especificación de Requerimientos.xlsx

1. Priorización de requerimientos: Esta actividad se realiza con el fin de evitar que los requerimientos entren en conflicto debido a que intervienen en ellos diversos participantes [20]. Se priorizarán los requerimientos mediante el uso de los mecanismos especificados en la sección [4.2 Mecanismos de Priorización y Trazabilidad.](#_4.2_Mecanismos_de)
2. Especificación de requerimientos: Es esta actividad los requerimientos se documentan detalladamente mediante el uso del formato de Especificación de Requerimientos tomada de la Plantilla SRS de Iron Works [3].

Ver Anexo: Especificación de Requerimientos.xlsx

El formato de especificación de requerimientos contendrá entre otros los siguientes campos para la especificación de cada uno de los requerimientos:

* **Id:** Identificador único del requerimiento.
* **Autor:** La persona o grupo dentro de The Walking Dev autor del requerimiento.
* **Origen:** Indica de donde surgió el requerimiento, de un caso de uso, de una entrevista, etc.
* **Razón:** Justificación de porque se contempló ese requerimiento.
* **Prioridad:** Indica que tan importante es el requerimiento para el funcionamiento del sistema.
* **Casos de uso asociados:** Son los casos de usos cuyo funcionamiento está asociando al requerimiento.
* **Nombre:** Nombre que describe el requerimiento
* **Versión:** Indica la versión actual de requerimiento, esta comienza en 1 y va aumentando en su parte decimal (1.1,1.2,1.3, etc.) de acuerdo al número de cambios que se le hayan realizado.
* **Descripción:** Contiene la descripción del requerimiento en lenguaje natural.
* **Estado:** Estado de desarrollo del requerimiento: propuesto, especificado, implementado, rechazado, diseñado, en prueba, finalizado.
* **Tipo:** Indica el tipo o categoría a la que el requerimiento pertenece, estas están definidas en la sección [2.7 Distribución de Requerimientos.](#_2.7_DISTRIBUCIÓN_DE)
* **Requerimientos asociados:** Son los requerimientos que están asociados al requerimiento que se está especificando debido a que son necesarios para que este se cumpla a cabalidad.
* **Verificación:** Describe las condiciones o características que se deben cumplir el requerimiento para que se pueda tomar como aceptado o completo.
* **Restricciones:** Es el conjunto de limitaciones dentro del requerimiento.
* **Comentarios Adicionales:** Espacio en cual se agregará información referente al requerimiento la cual no se puede categorizar en ninguno de los anteriores campos.

### 4.1.2 VALIDACIÓN DE REQUERIMIENTOS

La validación de requerimientos es el proceso de verificar que los requerimientos definan realmente el sistema que en verdad quiere el cliente. Esta validación de requerimientos es importante porque los errores en un documento de requerimientos nos pueden conducir a grandes costos por tener que rehacer [20]. Por ello se debe especificar procesos para la validación y verificación de requerimientos, estos se desarrollan y explican en la sección [5. Proceso de Verificación](#_PROCESO_DE_VERIFICACIÓN).

### 4.1.3 ADMINISTRACIÓN DE REQUERIMIENTOS

Durante el proceso de desarrollo de software, la comprensión que los participantes tienen de los problemas cambia constantemente. Entonces, los requerimientos del sistema también deben evolucionar para reflejar esa visión cambiante del problema [20]. Por lo anterior es necesario el definir un proceso para la administración de requerimientos donde se especifique como se llevará a cabo los cambios realizados a estos, especificando quien, como y cuando se hará

La administración del cambio en los requerimientos por lo tanto debe aplicarse a todos los cambios propuestos a los requerimientos de nuestro sistema, a continuación, se describirá el proceso de administración de requerimientos:

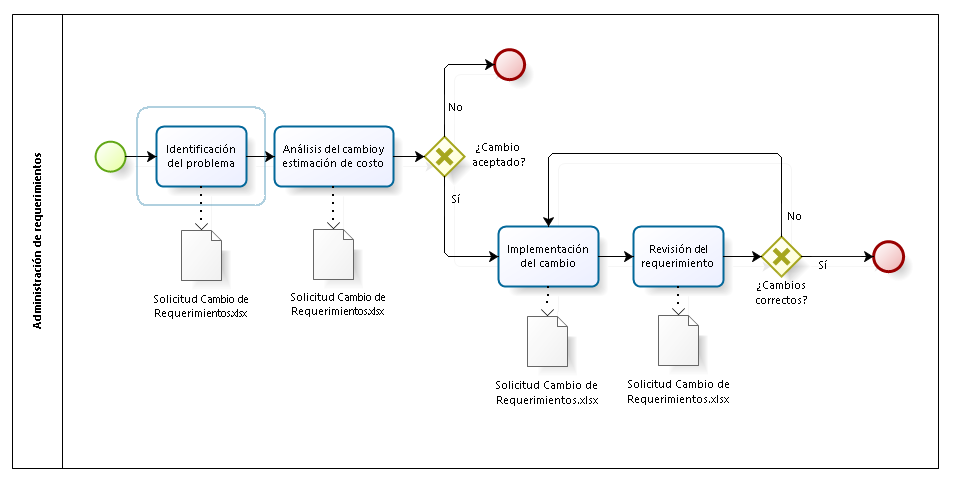


Ilustración 9. Administración del cambio de requerimientos.

1. Identificación del problema: El proceso de administración de requerimientos comienza con la identificación de un problema o, en ocasiones, con una propuesta de cambio específica [20]. De esta manera, al realizarse el proceso de verificación de requerimientos especificado en la [sección 5 Proceso de Verificación](#_PROCESO_DE_VERIFICACIÓN), el integrante del equipo que identifique un problema en la especificación de los requerimientos deberá agregar la petición del cambio al formato de Solicitud Cambio de Requerimientos con la información correspondiente, llenando el campo de estado del requerimiento con el valor de “En análisis”.

Ver Anexo: Solicitud Cambio de Requerimientos

1. Análisis del cambio y estimación de costo: Durante esta etapa, el problema o la propuesta de cambio se analizan para comprobar que es válida. Para ello se deberá tener en cuenta el costo que tendrá realizar el cambio, este será estimado en términos de modificaciones al documento de requerimientos [20]. Una vez completado este análisis, llevado a cabo en los sprint semanales, se toma una decisión acerca de si se procede o no con el cambio de requerimientos. Si la decisión es la de realizar el cambio, el estado de este pasará a “Aceptado” y se continuará con el proceso, sino el estado del cambio pasará a “Rechazado” y el proceso de administración de requerimientos finalizará.
2. Implementación del cambio: Se modifican el documento de requerimientos, en nuestro caso el documento Especificación de Requerimientos, el diseño y la implementación del sistema en donde sea necesario [20]. Además, en el formato de Solicitud Cambio de Requerimientos se deberá cambiar el estado de la petición del cambio a “Realizado”.

Ver Anexo: Especificación de Requerimientos.xlsx

1. Revisión del requerimiento: Finalmente se llevará a cabo una revisión de la especificación del requerimiento cambiado, esta revisión será llevada a cabo por el equipo de configuración y calidad. Si el cambio fue el correcto el proceso finaliza y el estado del cambio pasa a ser “Realizado/Correcto”, sino el estado pasará a “Realizado/Incorrecto” y la persona encargada de realizar el cambio deberá tener en cuenta los comentarios dados para corregir el cambio.

## 4.2 MECANISMOS DE PRIORIZACIÓN Y TRAZABILIDAD

### 4.2.1 PRIORIZACIÓN

Generalmente los desarrolladores no pueden implementar todos los requerimientos por cuestiones de limitaciones de tiempo, dinero, recursos, etc. Es así como se vuelve imprescindible detectar los “verdaderos requerimientos” para implementarlos [21]. Este proceso involucra claramente la priorización de requerimientos para así poder seleccionar cuales se van a satisfacer con mayor prioridad.

Para la priorización de requerimientos de nuestro proyecto usaremos Planning Poker ya que este método combina las opiniones de todos los integrantes del proyecto en un enfoque agradable mediante el cual estimar resultados de una manera rápida y segura [22].

Al inicio del Planning Poker, cada integrante del equipo de The Walking Dev deberá tener instalado en su dispositivo móvil la aplicación “Planning Poker” de Android especificada en la sección 8.2 Lenguajes y Herramientas del SPMP. La aplicación consiste en un grupo de tarjetas con un determinado valor, para nuestra priorización utilizaremos los valores 2, 8, 20, 40, 80 y 100; siendo 2 con menos prioridad y 100 con más prioridad. Teniendo cada participante la aplicación se procede con los siguientes pasos [22]:

1. Se elige una persona quien será el moderador de la actividad.
2. El moderador lee la descripción de cada uno de los requerimientos de manera ordenada, teniendo en cuenta la forma en la que se organizaron al hacer la recolección y análisis de estos.
3. Al ir leyendo cada uno de los requerimientos el moderador pregunta si hay alguna duda acerca estos, si es así cualquiera de los integrantes del equipo puede intervenir para aclarar a especificación del requerimiento.
4. Después de que las preguntas son resueltas todas las personas seleccionan en la aplicación móvil la tarjeta con el valor que ellos creen es el adecuado para estimar la prioridad del requerimiento teniendo en cuenta que el criterio que utilizaremos para la estimación será la dificultad de las dependencias y la importancia en el proyecto de cada requerimiento. De esta manera todos muestran al tiempo el valor elegido frente a los demás.
5. Ahora se observa los diferentes valores seleccionados, si la mayoría de valores son cercanos se procede a ingresar estos en el formato Especificación de Requerimientos en la pestaña “Priorización”, sino, quienes hayan dado la estimación más alta y más baja procederán a explicarle a los demás el porqué de su elección y así poder cambiar su opinión.

Ver Anexo: Especificación de Requerimientos.xlsx

1. Luego de que las personas con la estimación más alta y más baja hayan dado las razones de su elección, se vuelve a realizar la votación secreta en la cual se espera que los valores obtenidos sean cercanos para poder calcular la prioridad del requerimiento.
2. Si lo anterior no sucede se vuelve a repetir el paso en el cual se explica el porqué de la elección del valor más altos y más bajo, para después realizar la votación nuevamente. Esto se repite hasta que los valores sean lo más acorde posibles.

Al finalizar la actividad, la prioridad de cada requerimiento será traspasada a la especificación de cada uno de ellos en el documento de Especificación de Requerimientos.

Ver Anexo: Especificación de Requerimientos.xlsx

### 4.2.2 TRAZABILIDAD

La trazabilidad de requerimientos es la habilidad de describir y seguir la vida de un requerimiento tanto hacia adelante como hacia atrás, desde su origen, a través de su desarrollo y especificación, hasta su posterior despliegue y uso [23]. De esta manera, la trazabilidad es clave para poder solucionar muchos de los problemas surgidos en etapas antes y después de la creación y especificación de requerimientos.

Para poder seguirla vida de cada uno de nuestros requerimientos, y así cumplir con que estos sean trazables contaremos con los siguientes ítems:

* La especificación de requerimientos en donde se puede encontrar su priorización, especificación y organización para de esta manera observar su relación tanto con otros requerimientos como con casos de uso.

Ver Anexo: Especificación de Requerimientos.xlsx

* Para poder visualizar mejor la relación entre los requerimientos funcionales y los casos de uso, se creó una matriz en la cual se relaciona cada requerimiento con el caso de uso que lo origino.

Ver Anexo: Requerimientos contra Casos de Uso.xlsx

* Se creó un grafo de trazabilidad de requerimientos en el cual se puede observar de manera gráfica la relación que existe entre cada uno de los requerimientos.

Ver Anexo: Grafo de trazabilidad.jpg

## PROCESO DE CONSTRUCCIÓN DEL SRS

Inicialmente hemos de decir que para la construcción del presente documento seguimos la plantilla dada por nuestro profesor y cliente Miguel Eduardo Torres [3], esta nos sirvió de guía y así pudimos realizar el desarrollo del segundo avance de nuestro proyecto.

Primero que todo, se describieron los factores generales que afectarían a nuestro producto y requerimientos [(Ver sección 2. Descripción Global)](#_DESCRIPCIÓN_GLOBAL), especificando las características de los usuarios que van a interactuar con nuestro sistema, las restricciones que este tendrá, las suposiciones y dependencias tomadas en cuenta para su desarrollo y algo muy importante como lo es la realización del modelo de dominio que describiría el vocabulario y los conceptos clave del dominio de nuestro sistema.

Al finalizar la descripción global de nuestro producto continuamos con lo que respecta a los requerimientos del sistema los cuales describirían las características y funcionamiento de este, así, lo primero que se hizo es esta parte fue la recolección de los requerimientos [(Ver sección 4.1.1 Adquisición y Análisis de Requerimientos)](#_4.1.1_ADQUISICIÓN_Y), tanto funcionales como no funcionales. Para la obtención de estos nos basamos inicialmente en los casos de uso, de esta manera, mediante su análisis surgieron los requerimientos funcionales, luego, para la recolección de los requerimientos no funcionales usamos dos métodos. El primero consistió en reuniones semanales con todo el equipo de The Walking Dev en las cuales discutimos y aportábamos ideas para dichos requerimientos, de esta manera al tener una cantidad razonable decidíamos igualmente entre todos cuales se iban a tomar y cuáles no. Por otro lado, el segundo método se basó en analizar las restricciones para la realización de nuestro proyecto dadas por el cliente Miguel Eduardo Torres y de esta manera encontrar más requerimientos no funcionales.

El proceso de construcción continuo entonces con lo relacionado al análisis, organización, priorización y especificación de los requerimientos obtenidos, de esta manera, se definió la forma en que estos se iban a clasificar según determinados criterios para facilitar su manejabilidad y tener una buena organización sobre lo que se iba a trabajar [(Ver sección 2.7 Distribución de Requerimientos)](#_2.7_DISTRIBUCIÓN_DE), esto para después realizar el proceso de priorización el cual se especifica en la sección [4.2 Mecanismos de Priorización y trazabilidad](#_4.2_Mecanismos_de), finalizando con el proceso de especificación de cada uno de los requerimientos. Además de definir mecanismos de priorización se establecieron otros para asegurar la trazabilidad y así tener una manera de identificar más fácilmente el origen de muchos problemas que pudieran surgir.

Luego de lo comentado anteriormente y por último se defino el proceso de verificación y validación en la sección [5. Proceso de Verificación](#_PROCESO_DE_VERIFICACIÓN) con el cual se esperaba saber si tanto los requerimientos como su especificación cumplían con las características necesarias para pertenecer a nuestro sistema.

# PROCESO DE VERIFICACIÓN

# ANEXOS