

Especificación de Requerimientos de Software



PROYECTO: GAUCHADA

REVISIÓN: 02



MASTERBRANCH



Ficha del documento

Fecha	Revisión	Autor	Verificado
19-Sep-2016	01	MasterBranch	
18-Oct-16	02	MasterBranch	

Documento validado por las partes en fecha: 22-Oct-2016

Por el cliente	Por la empresa suministradora
	
Aclaración: <i>Nancy Díaz</i>	Aclaración: <i>Master Branch</i>

Contenidos

1	Introducción	4
1.1	Propósito	4
1.2	Alcance	4
1.3	Definiciones, acrónimos y abreviaturas	4
1.4	Referencias	4
1.5	Resumen	4
2	Descripción general	5
2.1	Perspectiva del producto	5
2.2	Funcionalidad del producto	5
2.3	Características de los usuarios	5
2.4	Restricciones	5
2.5	Suposiciones y dependencias	6
2.6	Evolución previsible del sistema	6
3	Requisitos específicos	6
3.1	Requisitos comunes de los interfaces	6
3.1.1	Interfaces de usuario	6
3.1.2	Interfaces de hardware	6
3.1.3	Interfaces de software	6
3.2	Requisitos funcionales	6
3.3	Requisitos no funcionales	6
3.3.1	Requisitos de rendimiento	6
3.3.2	Seguridad	7
3.3.3	Fiabilidad	7
3.3.4	Disponibilidad	7
3.3.5	Mantenibilidad	7
3.3.6	Portabilidad	7
3.4	Otros requisitos	7
4	Apéndices	7

1 Introducción

Teniendo en cuenta las pautas e indicaciones dadas, el presente informe abarca las necesidades y requerimientos que fueron solicitados.

1.1 Propósito

- Tiene como propósito dar a conocer el funcionamiento general y registrar la planeación de actividades a realizar durante éste proyecto de desarrollo de software.
- Este informe está dirigido a las personas involucrado en el desarrollo del sistema o «stakeholder».

1.2 Alcance

- El producto a desarrollar se denominada «Gauchada».
- *Gauchada*, es una página web en la cual se puede publicar favores.
- Dirigida a usuarios dentro del rango de Argentina.

1.3 Definiciones, acrónimos y abreviaturas

- Los *usuarios registrados* en esta aplicación se clasifican en tres: por un lado el *Favorecido*, el cual pide un favor en la aplicación y está a la espera de que algún usuario se ofrezca a realizarla; este último se lo denomina *Voluntario*. Luego de que el Favorecido elija quién de los voluntarios va a realizar el favor, el elegido se convierte ahora en *Ayudante*.
- El patrón de arquitectura *MVC* (Modelo Vista Controlador) es un patrón que define la organización independiente del *Modelo* (Objetos de Negocio), la *Vista* (interfaz con el usuario u otro sistema) y el *Controlador* (controlador del workflow de la aplicación).
- *Ruby on Rails*, también conocido como *RoR* o *Rails*, es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el paradigma del patrón Modelo Vista Controlador (MVC).
- Scrum: Es una metodología ágil. Un proceso en el que se aplican, de manera regular, un conjunto de mejores prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto. En Scrum se realizan entregas parciales y regulares del resultado final del proyecto, priorizadas por el beneficio que aportan al receptor del proyecto.

1.4 Referencias

Referencia	Título	Fecha	Autor
421	Entrevista de elicitación	19-Sep-2016	Master Branch
32	Cuestionario	15-Oct-2016	Master Branch
542	Estándar <i>IEEE 830</i> para Especificación de Requerimientos de Software	20-Oct-1998	<i>Institute of Electrical and Electronics Engineers</i>

1.5 Resumen

En el presente informe se menciona las características de este proyecto con el propósito de desarrollar *Gauchada*.

En sección 2 se describe aspectos necesarios para entender el mecanismo de pagina web como su funcionamiento específico, los involucrados en la interacción del sistema y las pautas que limitan el desarrollo, entre otros.

En la sección 3 se hace hincapié en las cláusulas que determinan la estructura del sistema requeridos, cuestiones como: el diseño, seguridad, mantenimiento, entre otras.

2 Descripción general

2.1 Perspectiva del producto

Es un producto independiente, que debido al crecimiento de los usuarios se debe migrar del blog a una plataforma web apropiada para el correcto funcionamiento. A su vez, no es autónomo ya que se piensa incluir funcionalidades que interactúan con otros sistemas como pueden ser entidades bancarias para validar tarjetas de crédito.

2.2 Funcionalidad del producto

Publicar la necesidad o favores en la página (para que la gente pueda ayudarlo). Solicitar y realizar el favor. Preguntar y responder sobre alguna duda o inquietud sobre el favor. Luego de cumplido el favor, se puede calificar si la ayuda aportada fue buena o mala. Esta calificación restará o sumará puntos, como corresponda, a la persona que entrega su ayuda. El Favorecido podrá dejar un comentario en el caso de que no le hayan hecho bien el favor.

2.3 Características de los usuarios

Tipo de usuario	Administrador
Formación	Saber como agregar, eliminar, modificar y ver logros; conocer la organización y cualquier actividad que es posible realizar en pagina.
Actividades	Ver un ranking de usuarios con respecto a su puntaje; obtener reportes donde se indique la ganancia (por compra de puntos) obtenida entre dos fechas; cargar o agregar al sitio los diferentes tipos de logros. Además, tiene la capacidad de hacer todas las actividades de un <i>usuario registrado</i> (exceptuando eliminar cuenta).

Tipo de usuario	Usuario registrado
Formación	Básico
Actividades	Autenticarse o «loguearse» y «desloguearse»; buscar favores con algún criterio; ver la lista y los detalles de favores publicar, calificar un favor; aceptar a un voluntario que haga el favor; postularse, preguntar y responder en un favor; registrar una tarjeta; comprar puntos; modificar su perfil; eliminar cuenta.

Tipo de usuario	Usuario no registrado
Formación	Básico
Actividades	Ver la pagina sin ninguna función habilitada; registrarse al sistema.

2.4 Restricciones

Las limitaciones a tener en cuenta a la hora de diseñar y desarrollar el sistema van a estar dadas por:

- Metodologías de desarrollo:
 - Patrón MVC (Model View Controller).
 - Utilización de la metodología ágil *Scrum*.
- Lenguaje de programación: Particularmente la utilización del framework «Ruby on Rails».
 - *Ruby* v2.3.1.

- Restricciones de Hardware: N/A.
- Restricciones de Sistema Operativo: Preferentemente utilizar navegador web actualizado.

2.5 Suposiciones y dependencias

Se asume que los desarrolladores del proyecto posean una PC con las restricciones anteriormente mencionadas.

2.6 Evolución previsible del sistema

A futuro está pensado crear un chat e implementar una versión para móviles (smartphone o tablet).

3 Requisitos específicos

3.1 Requisitos comunes de los interfaces

3.1.1 Interfaces de usuario

Se conservará el logo del antiguo logo del blog¹. Para el diseño de la página se respetarán los colores del logo: tonos anaranjados.

En la página principal se verá un listado de todos los favores, donde para cada uno se verá logo o imagen en el caso de que se haya subido una.

3.1.2 Interfaces de hardware

No aplica.

3.1.3 Interfaces de software

Se utiliza el sistema del banco para la validación de tarjetas de crédito.

3.2 Requisitos funcionales

Serán descritos como Historias de Usuario. Las mismas serán utilizadas para la especificación de requisitos a través de la plataforma *Pivotal Tracker*.

3.3 Requisitos no funcionales

3.3.1 Requisitos de rendimiento

- El 90 % de los posteos de favores deben realizarse en menos de un segundo.
- Se espera un máximo de 114 (ciento catorce) usuarios conectados simultáneamente. Y se prevé un incremento de usuarios usando la pagina por día.

Estos requisitos van a estar relacionados con el hardware que posea cada usuario a la hora de entrar en la página, que no debería de haber problema alguno o lentitud con cualquier hardware a la venta en la actualidad.

Así mismo, la conexión a Internet establecida por el usuario con su proveedor afectará en la rapidez de respuesta de la página pero no llegando a ser un problema funcional para el usuario.

¹Link: <http://unagauchada.tumblr.com>. Contraseña para ingresar: abracadabra.

3.3.2 Seguridad

Para mantener la información segura de todos los usuarios:

- Única visibilidad a la información personal de cada usuario sólo la va a tener el administrador de la pagina web.
- Solamente se podrá ver la información de contacto entre usuarios una vez acuerden realizar un favor (entre Favorecido y Ayudante).
- Verificación de las cuentas de usuario mediante el mail.

3.3.3 Fiabilidad

Se asegura una cantidad de incidentes de seguridad menor o igual a uno por mes durante los primeros tres meses de funcionamiento del sistema. Se asegura una cantidad de incidentes de caída del sistema menor o igual a tres por mes durante los primeros tres meses de funcionamiento del sistema.

3.3.4 Disponibilidad

La disponibilidad de la plataforma será de 24 (veinticuatro) hs. Las únicas excepciones de esta disponibilidad son para mantenimiento o diversas acciones de los desarrolladores/administrador de la página con previo aviso al no estar disponible.

3.3.5 Mantenibilidad

El mantenimiento será para solucionar los errores y conflictos que puede tener el sistema en el momento de usar la plataforma, como también incrementar la funcionalidad del mismo. En la entrevista se pactó seis meses de mantenimiento y que pasado ese periodo de tiempo, si es necesario, se decidirá si es conveniente seguir manteniendo.

3.3.6 Portabilidad

Al ser un producto con plataforma web de por sí conlleva una portabilidad general en la cual se puede acceder mediante distintos dispositivos a la web *Gauchada*.

3.4 Otros requisitos

Se puede pedir cualquier favor mientras esté en el marco legal.

4 Apéndices

Url del proyecto de la plataforma *Pivotal Tracker*:

<https://www.pivotaltracker.com/n/projects/1874477>.