
Plan de Gestión de Proyecto

Proyecto: Gauchada
Revisión [0.01]

PGT Software

[Octubre]

Ficha del documento

Fecha	Revisión	Autor	Verificado
14/10/2016	0.01	PGT	

Documento validado por las partes en fecha: [Fecha]

Por el cliente	Por la empresa suministradora
Aclaración	Aclaración [Nombre]

Introducción

.1 Resumen del Proyecto

.1.1 Propósito, alcance y objetivos.

El propósito del PDS es proporcionar la información necesaria para controlar el proyecto. En él se describe el enfoque de desarrollo de software. Los usuarios del PDS son:

- La administración de PGT lo utilizara para organizar la agenda y necesidades de recursos, y para realizar su seguimiento.
- El área de desarrollo de PGT lo utilizara para entender lo qué deben hacer, cuándo deben hacerlo y qué otras actividades dependen de ello.

Este plan de desarrollo de software es una versión que permitirá la automatización del actual blog Una Gauchada, este documento provee una visión global del enfoque de desarrollo propuesto.

El proyecto ha sido ofertado por el cliente Ulises Zunetri y su esposa en fase única la cual corresponde al desarrollo e implementación del nuevo sistema.

El enfoque de desarrollo propuesto constituye una configuración del proceso Gauchada de acuerdo a las características del proyecto, seleccionando los roles de los participantes, las actividades a realizar y los entregables que serán generados. Este documento es parte de los documentos generados.

.1.2 Supuestos y restricciones

- Fecha de entrega: como estamos empleando una metodología de trabajo ágil (Scrum), el sistema no se entrega completo en una única fecha final, sino que se desarrolla un trabajo progresivo e incremental en el que se establecen distintas fechas, y en cada una se entrega el sistema al cliente con una nueva funcionalidad para que éste último la valide. Se nos han establecido tres fechas de entrega para el sistema.
- Presupuesto: se requerirá un monto de aproximadamente \$150000 para poner en marcha el proyecto y concluirlo exitosamente. Los motivos por los que establecemos este monto se describirán más adelante.
- Recursos:
 - Materiales: sólo las tres computadoras personales con las que ya cuentan y trabajan los desarrolladores, todas cuentan con el lenguaje Ruby on Rails y conexión a Internet para subir los documentos y el código del sistema.
 - Humanos: se considera que los tres desarrolladores socios y propietarios de la empresa desarrolladora son suficientes para las tareas que exige este sistema. También puede considerarse en este apartado a los dos propietarios de la empresa cliente, ya que es nuestra idea que ambos trabajen en el desarrollo del sistema con nosotros y formen parte activa y efectiva de nuestro grupo de trabajo.

.1.3 Entregables del proyecto

- 19/09/2016 (ya fue entregada): Entrega de tres documentos: uno que contuviera un cuestionario para realizarle a los clientes de la empresa cliente

sobre el sistema a desarrollar, otro en el que se ofrecieran detalles de la entrevista con los clientes y el último, una descripción sobre la fase de elicitación de requerimientos que sigue reglas estandarizadas.

- 17/10/2016: Entrega del Plan de Gestión del Proyecto (este documento) y de la pila de producto, una definición de las distintas historias de usuario para que la empresa cliente le asigne prioridades.
- Entrega del Diseño de Interfaces, el último de los documentos a presentar antes de la codificación del sistema.
- 9/11/2016: Primera demostración del sistema ante los clientes.
- 23/11/2016: Segunda demostración del sistema ante los clientes.
- 7/12/2016: Tercera y última demostración del sistema final ante los clientes.

.1.4 Calendario y resumen del presupuesto

Actividad	Esfuerzo (hs)	Fecha de inicio	Fecha de finalización	Presupuesto Subtotal (\$)
Elicitación de requerimientos	60	14/9/2016	17/9/2016	6000
Planificación	90	7/10/2016	12/10/2016	9000
Diseño	111	15/10/2016	22/10/2016	11100
Codificación Demo 1	315	23/10/2016	9/11/2016	31500
Pruebas Demo 1	40	10/11/2016	14/11/2016	4000
Codificación Demo 2	150	15/11/2016	23/11/2016	15000
Pruebas Demo 2	40	23/11/2016	28/11/2016	4000
Codificación Demo 3	150	29/11/2016	7/12/2016	15000
Pruebas Demo 3	40	8/12/2016	12/12/2016	4000
Documentación	504	14/9/2016	16/12/2016	50400

	Esfuerzo Total (hs)	Fecha de inicio	Fecha de finalización	Presupuesto Total (\$)
Proyecto	1500	10/9/2016	1/3/2017	150000

Documentos referenciados

Referencia	Título	Fecha	Autor
	Cuestionario	2/9/2016	PGT
	Entrevista	7/9/2016	PGT
	Entrevista2	14/9/2016	PGT
	Especificación de requisitos de software	18/10/2016	PGT
	IEEE Std 1058-1998	1998	IEEE

Definiciones y acrónimos

Nombre	Descripción
Gauchada	Aplicación web, que provee a los usuarios pedir y solicitar favores.
Favor	Actividad a realizar por un usuario hacia otro
Solicitante	Usuario que solicita / pide un favor
Proveedor	Usuario que cumple un favor a otro
Usuario	Persona que utilizará el sistema para comunicarse
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional
ERS	Especificación de Requisito de Software
Visitante	Usuario no registrado en la aplicación
PDS	Plan de Desarrollo de Software
PGP	Plan de Gestión de Proyecto

Organización del proyecto

.2 Interfaces externas

Como en este proyecto tanto la empresa cliente como la desarrolladora son de dimensiones muy pequeñas (dos y tres personas respectivamente), no es necesario establecer fronteras administrativas y gerenciales entre el proyecto y otras entidades ya que dichos límites no existen: todos los individuos de ambas empresas intervienen en el proyecto con roles similares.

.3 Estructura interna

La estructura interna de la empresa desarrolladora está conformada únicamente por sus tres socios fundadores: Federico Tubaro, Gurmendi Alan, Pereira Luciano, que en conjunto toman todas las decisiones administrativas y desarrollan el software encargado por la empresa cliente.

.4 Roles y responsabilidades

- Scrum Master: actualmente Tubaro Federico es el Scrum Master, sin embargo este rol va a rotar luego de cada Sprint entre los desarrolladores de la empresa. Será la persona que asegura el seguimiento de la metodología guiando las reuniones y ayudando al equipo ante cualquier problema que pueda aparecer y cuya responsabilidad es entre otras, la de hacer de paraguas ante las presiones externas.
- Scrum Team: actualmente Gurmendi Alan, Pereira Luciano conforman el Scrum Team, es decir, el equipo de desarrollo del proyecto, responsable de implementar la funcionalidad o funcionalidades elegidas por el Product Owner durante la planificación del próximo sprint. Los integrantes cambian continuamente, ya que para cada entrega uno de ellos se convierte en Scrum Master, volviendo el anterior a formar parte del Scrum Team.
- Product Owner: Ulises Zunetri y su esposa, ambos propietarios de la empresa cliente y, por lo tanto, de nuestro producto de software, que conocen y marcan las prioridades del proyecto. Son los dueños del producto de software. Éstos no cambian nunca

Planes de administración del proceso

.5 Plan inicial

.5.1 Plan del personal

No se requerirá ningún personal adicional además de los tres desarrolladores propietarios de la empresa desarrolladora, que serán necesarios hasta la finalización del proyecto que, según nuestra estimación, sería el primero de Enero del año 2017.

Cantidad y tipos del personal requerido. Duración para la cual son necesarios. Se requiere de al menos tres programadores competentes, con conocimiento en el Framework a utilizar (RubyOnRails), además de conocimientos básicos en HTML y CSS durante todo el desarrollo del proyecto, estimado en 2 meses.

.5.2 Plan de adquisición de recursos

- Hardware: aunque ya contamos con el hardware necesario para llevar adelante el proyecto (computadoras personales), tenemos pensado actualizar nuestros dispositivos de red ya que una buena conexión a Internet es fundamental para desarrollar correctamente el proyecto. Consultaríamos a distintos proveedores disponibles en el mercado y se los compraríamos al que los ofrezca a un precio más bajo.

- Software:
 - Descargaremos el lenguaje Ruby on Rails en forma gratuita de la página railsinstaller.org para instalarlo en Windows y poder programar el sistema.
 - Abriremos una cuenta de usuario para cada integrante del grupo en Github en forma gratuita para publicar toda la documentación del proyecto para que los clientes la revisen y nos realicen correcciones sobre ella.
 - Abriremos una cuenta de usuario para cada integrante del grupo en Pivotal Tracker en forma gratuita para publicar los casos de usuario a partir de los requerimientos de los clientes y usuarios, para, junto con los clientes, asignarles puntajes y ubicarlas en distintos sprints para ser presentados en diferentes fechas.
 - Abriremos una cuenta de usuario para cada integrante del grupo en Heroku en forma gratuita para desplegar nuestro sistema en la plataforma para que nuestro cliente pueda utilizarlo y probarlo durante las distintas demostraciones de los casos de usuario asignados al sprint actual.
- Servicios administrativos: en un principio creemos que somos capaces de realizar todas las tareas administrativas del proyecto sin ningún inconveniente. Si tenemos problemas, contrataríamos a algún estudiante de Ciencias Económicas para que se una a nuestro equipo como administrador.
- Contratos de servicio: tenemos planeado contraer un contrato particular de Internet de alta velocidad y fiabilidad para no tener ningún inconveniente a la hora de acceder a las distintas plataformas online necesarias para desarrollar el proyecto. Analizaríamos a los distintos proveedores (Cablevisión y Telefónica) y compararíamos sus ofertas, eligiendo la que consideremos más conveniente, priorizando en este caso la calidad del servicio por encima de su costo.

.5.3 Plan de entrenamiento del personal del Proyecto

Actualmente los desarrolladores asisten dos días por semana a clases prácticas en la Facultad de Informática de la UNLP donde se les enseña el lenguaje Ruby on Rails, requerido por la empresa cliente como medio para programar el código de la solución y del cual no tenían ningún conocimiento previo.

.6 Plan de trabajo

.6.1 Principales actividades del proyecto

- Elicitación de requerimientos: Es el proceso de adquirir todo el conocimiento relevante necesario para producir un modelo de los requerimientos del dominio del problema.
- Planificación: El objetivo de la planificación del proyecto de software es proporcionar un marco de trabajo que nos permita hacer estimaciones razonables de recursos, costos y esfuerzo a lo largo de la duración prevista del proyecto.
- Diseño: Se basa en la especificación de requisitos producido por la elicitación de requerimientos. El diseño define cómo estos requisitos se cumplirán, la estructura que debe darse al sistema de software para que se haga realidad. El diseño sigue siendo una fase separada de la programación o codificación.
- Codificación: Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior. Esta tarea la realiza el programador, siguiendo por completo los lineamientos impuestos en el diseño y en consideración siempre a los requisitos funcionales y no funcionales (ERS) especificados en la primera etapa.

- Prueba: Entre las diversas pruebas que se le efectúan al software se pueden distinguir principalmente:
 - Pruebas unitarias: Consisten en probar o testear piezas de software pequeñas; a nivel de secciones, procedimientos, funciones y módulos; aquellas que tengan funcionalidades específicas. Dichas pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código, mucho más reducidas que el conjunto, y que tienen funciones concretas con cierto grado de independencia.
 - Pruebas de integración: Se realizan una vez que las pruebas unitarias fueron concluidas exitosamente; con éstas se intenta asegurar que el sistema completo, incluso los subsistemas que componen las piezas individuales grandes del software funcionen correctamente al operar en conjunto.
- Documentación: Es todo lo concerniente a la documentación del propio desarrollo del software y de la gestión del proyecto, diagramas de casos de uso, pruebas, manuales de usuario, manuales técnicos, etc; todo con el propósito de eventuales correcciones, usabilidad y ampliaciones al sistema.

.6.2 Asignación de esfuerzo

Actividad	Cantidad	Esfuerzo (hs) Unitario	Esfuerzo (hs) Subtotal
Elicitación de requerimientos	3	20	60
Planificación	3	30	90
Diseño	3	37	111
Codificación	3	205	615
Prueba	3	40	120
Documentación	3	168	504
Esfuerzo Total = 1500			

.6.3 Asignación de presupuesto

Actividad	Esfuerzo (hs) Subtotal	Costo por Hora (\$)	Presupuesto Subtotal (\$)
Elicitación de requerimientos	60	100	6000
Planificación	90		9000
Diseño	111		11100
Codificación	615		61500
Prueba	120		12000
Documentación	504		50400
Presupuesto Total = \$150000			

.7 Plan de control

.7.1 Plan de control de requerimientos

No aplica

.7.2 Plan de control de calendario*No aplica***.7.3 Plan de control de presupuesto***No aplica***.7.4 Plan de control de calidad***No aplica***.7.5 Plan de informe***No aplica***.7.6 Plan de recolección de métricas***No aplica***.8 Plan de administración de riesgos***Quedará pendiente para una entrega posterior***.9 Plan de liberación de proyecto***Acciones que se deben tomar una vez que el proyecto esté terminado.***Planes de procesos técnicos****.10 Modelo de proceso**

Como nuestra metodología de trabajo, Scrum, es iterativa e incremental, consideramos que el modelo del ciclo de vida más adecuado para el proyecto es el de desarrollo por fases de tipo incremental, en el cual el sistema es particionado en subsistemas de acuerdo con su funcionalidad. Cada entrega agrega un subsistema.

.11 Métodos, herramientas y técnicas

- Metodología de desarrollo: Scrum, 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. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos. En Scrum se realizan entregas parciales y regulares del resultado final del proyecto, priorizadas por el beneficio que aportan al receptor del proyecto.
- Lenguaje de programación: Ruby on Rails, también conocido como RoR o Rails, un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby (interpretado, reflexivo y orientado a objetos, software libre), siguiendo el paradigma de la arquitectura Modelo Vista Controlador (MVC). Trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones del mundo real escribiendo menos código que con otros frameworks y con un mínimo de configuración. El lenguaje de programación Ruby permite la metaprogramación, de la cual Rails hace uso, lo que resulta en una sintaxis que muchos de sus usuarios encuentran muy legible. Rails se distribuye a través de RubyGems, que es el formato oficial de paquete y canal de distribución de bibliotecas y aplicaciones Ruby.
- Plataforma como servicio de computación en la Nube: Heroku es propiedad de Salesforce.com. Heroku, es una de las primeras plataformas de computación en la nube, que fue desarrollada desde junio de 2007, con el objetivo de soportar solamente el

lenguaje de programación Ruby, pero posteriormente se ha extendido el soporte a Java, Node.js, Scala, Clojure y Python y (no documentado) PHP.

- Plataforma como servicio (en inglés platform as a service, PaaS), es la encapsulación de una abstracción de un ambiente de desarrollo y el empaquetamiento de una serie de módulos o complementos que proporcionan, normalmente, una funcionalidad horizontal (persistencia de datos, autenticación, mensajería, etc.). De esta forma, un arquetipo de plataforma como servicio podría consistir en un entorno conteniendo una pila básica de sistemas, componentes o APIs preconfiguradas y listas para integrarse sobre una tecnología concreta de desarrollo (por ejemplo, un sistema Linux, un servidor web, y un ambiente de programación como Perl o Ruby).

- La computación en la nube, conocido también como servicios en la nube, informática en la nube, nube de cómputo o nube de conceptos, (del inglés cloud computing), es un paradigma que permite ofrecer servicios de computación a través de Internet.

- Control de versiones: GitHub es una forja (plataforma de desarrollo colaborativo de software que se enfoca hacia la cooperación entre desarrolladores para la difusión de software y el soporte al usuario) para alojar proyectos utilizando el sistema de control de versiones Git (software de control de versiones con funcionalidad plena, diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente). Utiliza el framework Ruby on Rails por GitHub, Inc. (anteriormente conocida como Logical Awesome). El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

- Gestión de proyectos ágiles: Esta es una herramienta de gestión de proyectos ágiles, por ejemplo, para proyectos con Scrum, que es la metodología de trabajo que nosotros implementamos para nuestro proyecto. De esta herramienta podemos destacar frente a otras:

- Está en la nube. Con lo que no tenemos que instalar nada en nuestros ordenadores, todo lo haremos a través del navegador.

- Se centra en algo muy interesante que es la gestión de la pila de producto, y métricas como la velocidad.

- Por supuesto se puede usar para trabajo con equipos descentralizados, puesto que está en internet. Además se puede crear varios perfiles de acceso por lo que podría haber usuarios que, por ejemplo, sólo tienen permiso para ver las cosas, pero no para modificarlas.

- Guarda histórico de nuestras acciones. Esto es bastante importante si tenemos, por ejemplo, alguna certificación de calidad, como puede ser ISO, que siempre nos exigen trazabilidad de nuestro trabajo.

.12 Plan de infraestructura

Nuestro lugar de encuentro es frecuentemente la Facultad de Informática de la UNLP, un lugar público, libre y gratuito, y ocasionalmente el hogar de alguno de los integrantes del grupo. Ambas instalaciones cuentan con la infraestructura y los recursos necesarios para desarrollar de forma amena y cómoda el proyecto. No contamos aún con una sede propia porque no poseemos los recursos para adquirirla. Desde luego que esto cambiará una vez que desarrollemos el sistema, el mismo resulte exitoso y empecemos a percibir el monto acordado con los clientes por él.

Con respecto a los equipos, podemos mencionar que ya contamos con todos los recursos necesarios para desarrollar el proyecto, ya que, como se mencionó, sólo se necesita conexión a Internet para subir los cambios a la Nube. No requiere una elevada

capacidad de procesamiento, incluso se podría manejarlo desde un dispositivo móvil. Además de nuestras computadoras y celulares personales, la misma Facultad de Informática en la que nos reunimos cuenta en su Biblioteca con computadoras de alto rendimiento con acceso a Internet que pueden ser utilizadas libre y gratuitamente por cualquier estudiante.

.13 Plan de aceptación del producto

Los clientes aceptarán el producto sólo si el mismo posee y cumple con las siguientes funcionalidades (historias de usuario):

- Como usuario registrado quiero poder avisar que una gauchada propia no fue realizado por el prestador.
- Como usuario registrado quiero ver el "Acerca de"
- Como usuario registrado quiero poder recuperar mi contraseña para poder iniciar sesión.
- Como Usuario NO Registrado quiero poder registrarme al sistema.
- Como Usuario NO Registrado quiero poder ver el listado de gauchadas.
- Como Usuario Administrador quiero poder ver las ganancias obtenidas entre dos fechas determinadas.
- Como Usuario Administrador quiero poder ver un ranking de usuarios con mayor cantidad de eslabones.
- Como Usuario Administrador quiero poder modificar un logro
- Como Usuario Administrador quiero poder dar de baja un logro
- Como Usuario Administrador quiero poder dar de crear un logro
- Como Usuario Administrador quiero poder modificar el estado de un usuario.
- Como Usuario Registrado quiero poder ver el perfil de otro usuario.
- Como Usuario Registrado quiero poder comprar eslabones para poder realizar publicaciones.
- Como Usuario Registrado quiero poder buscar las publicaciones por ciudad.
- Como Usuario Registrado quiero poder editar mi perfil.
- Como Usuario Registrado quiero poder ver mi perfil. (*)
- Como Usuario Registrado quiero poder ver mi listado de publicaciones.
- Como Usuario Registrado quiero poder avisar que una gauchada propia fue realizada.
- Como Usuario Registrado quiero poder elegir quien deberá cumplir mi favor solicitado
- Como Usuario Registrado quiero poder ver los comentarios de mi publicación.
- Como Usuario Registrado quiero poder eliminar un comentario en una publicación.
- Como Usuario Registrado quiero poder realizar un comentario en una publicación para ofrecerme para realizar la gauchada.
- Como Usuario Registrado quiero poder modificar mis publicaciones.
- Como Usuario Registrado quiero poder eliminar mis publicaciones de favores.
- Como Usuario Registrado quiero poder realizar una publicación para pedir un favor.
- Como Usuario Registrado quiero poder cerrar sesión, para salir del sistema.
- Como Usuario Registrado quiero poder iniciar sesión, para utilizar el sistema.

Plan de procesos de apoyo

.14 Plan de administración de configuración

Se describe la administración de los componentes de la gestión de la configuración.

.15 Plan de pruebas

Se describen los planes de prueba que se utilizan a lo largo del desarrollo.

.16 Plan de documentación

- Cuestionario: Documento conformado por preguntas que permiten a los desarrolladores del sistema recabar hechos, información y opiniones generalizadas de todos los futuros usuarios.
- Entrevista: Técnica de exploración mediante la cual los desarrolladores recolectamos información de los clientes a través de la interacción cara a cara. En particular, el documento está basado en un formato de preguntas y respuestas que permiten conocer opiniones y sentimientos personales de los clientes.
- Especificación de requerimientos de software (Standard 830): Documento que sigue una reglamentación estándar para reflejar toda la información referente a la fase de elicitación de requerimientos.
- Plan de Gestión del Proyecto (Standard 1058): Es este documento, que sigue una reglamentación estándar para reflejar toda la información referente a la fase de planificación del proyecto.
- Diseño de Interfaz: Es el diseño enfocado en la experiencia de usuario y la interacción. Su objetivo es que las aplicaciones o los objetos sean más atractivos y además, hacer que la interacción con el usuario sea lo más intuitiva posible, conocido como el diseño centrado en el usuario. Existe una norma estándar al respecto, la ISO 9241.

.17 Plan de aseguramiento de calidad

Se describen todos los aspectos de aseguramiento de calidad, incluyendo pruebas, estándares y revisiones.

.18 Plan de revisiones y auditorías

Se describe como se realizarán las revisiones y auditorías.

.19 Plan de resolución de problemas

Se describen los pasos a seguir para la resolución de los problemas que ocurran a lo largo del desarrollo.

.20 Plan de administración de terceros

Se describe como se seleccionaran y administraran las empresas contratadas en caso que algún producto del proyecto sea realizado por otra empresa.

.21 Plan de mejoras en el proceso

Se describen las estrategias de mejoras de procesos que se utilizarán en el proyecto.

Planes adicionales

Una vez que el producto se encuentre finalizado, se tendrán en cuenta los siguientes planes:

- Plan de backup: Cada tres meses se realizará un backup de la información almacenada en las bases de datos. Este servicio será brindado de forma personal a fin de que la tarea sea realizada por el equipo de desarrollo y no por un software.
- Plan de instalación: El equipo de desarrollo se encargará de probar el software en las

maquinas del cliente a modo de verificación del funcionamiento del mismo.

- Plan de mantenimiento: Cada seis meses se realizará el mantenimiento del código fuente del software. Este mantenimiento involucra una refactorización de dicho código para aprovechar las potenciales mejoras del lenguaje que puedan volver al sistema más eficiente y seguro.