
Plan de Gestión de Proyecto

Proyecto: BestnidWeb
Revisión 1.0



Octubre 2014

Ficha del documento

Fecha	Revisión	Autor	Verificado
10/10/2014	1.0	Dead Programmers Society	

Documento validado por las partes en fecha: 14/10/2014

Por el cliente	Por la empresa suministradora
Aclaración Bestnid	Aclaración Dead Programmers Society



1 Introducción

1.1 Resumen del Proyecto

1.1.1 Propósito, alcance y objetivos.

El propósito del software a desarrollar es funcionar como una plataforma de subastas online en la cual los usuarios puedan tanto subastar como adquirir productos de manera totalmente anónima mientras dure la subasta. Permitiendo de manera fácil, intuitiva y segura navegar por la plataforma.

El proyecto tiene como finalidad entregar el producto incrementalmente, añadiendo funcionalidad en cada entrega de manera de construir el software con calidad y evitando la mayor cantidad de inconvenientes durante el desarrollo de manera tal de llegar a la última fecha de entrega en tiempo y forma.

Como desarrolladores nuestro objetivo es entregar el producto con las características óptimas de funcionamiento y sobre todo cumplir con la fecha prevista para la entrega final.

1.1.2 Supuestos y restricciones

Se supone que el software debe ser entregado a mitad de diciembre de 2014 como fecha límite. Las características funcionales y cualquier documento de relevancia del sistema a desarrollar serán facilitadas por el Cliente mientras que las técnicas serán provistas por Dead Programmers Society.

1.1.3 Entregables del proyecto

Demo N° 1: 14/11/2014

Demo N° 2: 28/11/2014

Demo N° 3: 12/12/2014

1.1.4 Calendario y resumen del presupuesto

El proyecto será realizado en un lapso de 12 semanas, desde la obtención de los requerimientos del software hasta la última entrega. El presupuesto total será de AR\$15.000 (pesos, quince mil).

2 Documentos referenciados

Referencia	Título	Fecha	Autor
E1	Entrevista N° 1	19/09/2014	Dead Programmers Society
C1	Cuestionario N° 1	30/09/2014	Dead Programmers Society
STD	Especificación de requerimientos (versión 1)	30/09/2014	Dead Programmers Society
A1	Planilla de Administración	19/09/2014	Bestnid



3 Definiciones y acrónimos

Demo: las demos o versiones demo son prototipos, versiones incompletas o de evaluación de un determinado programa informático con el fin de mostrar la idea de funcionamiento y demostrar sus funcionalidades.

Wiki: es el nombre que recibe un sitio web cuyas páginas pueden ser editadas directamente desde el navegador, donde los usuarios crean, modifican o eliminan contenidos que, generalmente, comparten.

SGBD: es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos.

Hosting: es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web.

Heroku: es una plataforma como servicio de computación en la Nube que soporta distintos lenguajes de programación

4 Organización del proyecto

4.1 Interfaces externas

Para desarrollar el sistema, se utilizará la técnica de Scrum. Para ello se definirá un Scrum Master, que es quien se encargará, junto con el grupo de trabajo, de que los objetivos se cumplan utilizando la técnica adecuadamente. El Scrum Master variará a lo largo del proyecto.

4.2 Estructura interna

El grupo de desarrollo estará compuesto por el Scrum Master y un equipo de trabajo formado por tres personas. Como se mencionó anteriormente el rol de Scrum Master lo tomará cada miembro del equipo de trabajo a lo largo del proyecto.

Por otro lado se mantendrá contacto frecuente con los dueños del producto (Product Owner).

4.3 Roles y responsabilidades

Product Owner: genera y hace saber al grupo de desarrollo sobre las necesidades que tiene que cubrir el software (historias de usuario).

Scrum Master: actúa como una protección entre el equipo y cualquier influencia que le distraiga. El Scrum Master se asegura de que el proceso Scrum se utiliza como es debido. Es quien hace que las reglas se cumplan y ayuda al equipo ante cualquier inconveniente que pueda llegar a surgir.

Equipo de desarrollo: grupo de tres personas con las habilidades necesarias para realizar el trabajo (análisis, diseño, desarrollo, pruebas, documentación, etc).

5 Planes de administración del proceso

5.1 Plan inicial

5.1.1 Plan del personal

Se dispondrá de cuatro personas en total para el grupo, las cuales estarán a disposición del proyecto hasta la fecha de entrega.



5.1.2 Plan de adquisición de recursos

Un servidor que mantenga el sistema en línea, en este caso se utilizara “Heroku” como plataforma de hosting online. Esta plataforma gratuita será utilizada para presentar las demos, una vez que el sistema deba entrar en producción definitivamente habrá que elegir un plan de pago que de adecue a las necesidades del proyecto.

5.1.3 Plan de entrenamiento del personal del Proyecto

El equipo realizo un curso de capacitación sobre las tecnologías a utilizar en el proyecto, por ello, el equipo se encuentra óptimamente entrenado para llevar a cabo el proyecto.

5.2 Plan de trabajo

5.2.1 Principales actividades del proyecto

Se especifican las actividades, detallando cada tarea, si es apropiado.

- Elicitación de requerimientos
- Diseño
- Codificación
- Testing
- Mantenimiento

5.2.2 Asignación de esfuerzo

Actividad	Cantidad	Esfuerzo (hs) Unitario	Esfuerzo (hs) Subtotal
Elicitación de requerimientos	4	2	8
Diseño	4	8	32
Codificación	4	15	60
Testing	2	4	20
Mantenimiento	1	3	3

5.2.3 Asignación de presupuesto

Valores en AR\$:

Elicitación de requerimientos = \$800

Diseño = \$3.200

Codificación = \$6.000

Testing = \$2.000

Mantenimiento = \$300 (pesos, dieciséis mil).

Gastos varios = \$2.700

Presupuesto total del proyecto = AR\$15.000 (pesos, quince mil).



5.3 Plan de control

5.3.1 Plan de control de requerimientos

No Aplica

5.3.2 Plan de control de calendario

No Aplica

5.3.3 Plan de control de presupuesto

No Aplica

5.3.4 Plan de control de calidad

No Aplica

5.3.5 Plan de informe

No Aplica

5.3.6 Plan de recolección de métricas

No Aplica

5.4 Plan de administración de riesgos

Id Riesgo	Nombre	Probabilidad	Impacto	Responsable	Estado
01	Adición o eliminación de requisitos del software	90%	3	Cliente o empresa	Sin suceder.
02	Cambios en los plazos de entrega	30%	4	Cliente	Sin suceder
03	Conflictos con la tecnología utilizada	35%	4	Empresa	Sin suceder
04	Conflictos internos del equipo	25%	2	Empresa	Sin suceder
05	Caída del servidor	15%	4	Empresa	Sin suceder
06	Disconformidad del cliente respecto a la interfaz grafica	40%	1	Empresa	Sin suceder
07	Recorte de presupuesto	2%	5	Empresa	Sin suceder
08	Cancelación del proyecto	2%	8	Cliente	Sin suceder



Id Riesgo: 01	Nombre: Adición o eliminación de requisitos del software	
	Fecha : 10/10/2014	
	Descripción: El cliente decide agregar o quitar funcionalidades al sistema.	
Probabilidad: 90%		
Impacto: 3		
Responsable: Cliente o Empresa		Clase: Proyecto y Producto
Estrategia de Mitigación (Anulación/Minimización): Comunicación fluida con el cliente para la detección temprana de los nuevos requisitos asegurando el entendimiento de los mismos y permitiendo su implementación en el menor tiempo posible.		
Plan de Contingencia : Implementar los nuevos requisitos y posteriormente mostrar los resultados al cliente. De ser necesario se acordara con el cliente una nueva fecha de entrega.		

Id Riesgo: 02	Nombre: Cambios en los plazos de entrega	
	Fecha : 10/10/2014	
	Descripción: El cliente solicita entregas antes de las fechas acordadas.	
Probabilidad: 30%		
Impacto: 4		
Responsable: Cliente		Clase: Proyecto
Estrategia de Mitigación (Anulación/Minimización): Aceleración del ritmo de desarrollo agregando horas extra de trabajo.		
Plan de Contingencia: El Scrum Master avisará al equipo que los trabajos deberán realizarse en tiempos más cortos, y se acordará realizar sprints más frecuentes.		

Id Riesgo: 03	Nombre: Conflictos con la tecnología utilizada	
	Fecha : 10/10/2014	
	Descripción: El equipo de desarrollo se encuentra con dificultades a la hora de realizar la implementación del sistema.	
Probabilidad: 35%		
Impacto: 4		
Responsable: Empresa		Clase: Proyecto y Producto
Estrategia de Mitigación (Anulación/Minimización): Capacitación y consultas sobre los servicios utilizados.		
Plan de Contingencia: Consultar por distintos medios a desarrolladores experimentados y foros de soporte de las tecnologías utilizadas.		



Id Riesgo: 04	Nombre: Conflictos internos del equipo.	
	Fecha : 10/10/2014	
	Descripción: Desacuerdo en el equipo de desarrollo a la hora de diseñar y/o implementar el sistema.	
Probabilidad: 25%		
Impacto: 2		
Responsable: Empresa		Clase: Proyecto, Producto y Negocio
Estrategia de Mitigación (Anulación/Minimización): Se intentará llegar a un acuerdo en el equipo.		
Plan de Contingencia: El Scrum master actuará como mediador entre las partes en desacuerdo y en el caso de no llegar a un acuerdo, decidirá cómo debe seguir el desarrollo.		

Id Riesgo: 05	Nombre: Caída del servidor	
	Fecha : 10/10/2014	
	Descripción: La empresa contratada para realizar el alojamiento web de la plataforma desarrollada tiene problemas técnicos que afectan a sistema a desarrollar.	
Probabilidad: 15%		
Impacto: 4		
Responsable: Empresa		Clase: Producto
Estrategia de Mitigación (Anulación/Minimización): Se elegirá una empresa de hosting que tenga una rápida respuesta ante fallos.		
Plan de Contingencia: Se recurrirá al soporte de la empresa que provee el hosting al sistema, para indicar los problemas del servidor y solicitar su resolución.		

Id Riesgo: 06	Nombre: Disconformidad del cliente respecto de la interfaz gráfica.	
	Fecha : 10/10/2014	
	Descripción: El Cliente encuentra la interfaz gráfica implementada poco amigable o en desacuerdo con sus expectativas o las de los usuarios.	
Probabilidad: 40%		
Impacto: 1		
Responsable: Empresa		Clase: Proyecto y Producto
Estrategia de Mitigación (Anulación/Minimización): Se tratará de resolver la interfaz gráfica del sistema lo más acorde a los requerimientos del cliente, y se consultará asiduamente al cliente sobre ésta.		
Plan de Contingencia: Reorganizar la gráfica, cambiando el código, pero sin alterar ninguna funcionalidad del producto ni la eficiencia del mismo.		



Id Riesgo: 07	Nombre: Recorte de presupuesto.	
	Fecha : 10/10/2014	
	Descripción: El cliente no puede costear el proyecto en su totalidad.	
Probabilidad: 2%		
Impacto: 5		
Responsable: Empresa		Clase: Proyecto, Producto y Negocio
Estrategia de Mitigación (Anulación/Minimización): Acordar con el cliente las condiciones de pago y verificar que la escala del proyecto se corresponda con posibilidades económicas.		
Plan de Contingencia: Consensuar con el cliente las características del proyecto que pasarán a ser prescindibles y el nuevo presupuesto. Mantener el ritmo de trabajo normal, pero reduciendo el tamaño del proyecto.		

Id Riesgo: 08	Nombre: Cancelación del Proyecto	
	Fecha : 10/10/2014	
	Descripción: El cliente decide cancelar el proyecto por algún motivo.	
Probabilidad: 2%		
Impacto: 8		
Responsable: Cliente		Clase: Proyecto, Producto y Negocio
Estrategia de Mitigación (Anulación/Minimización): Averiguar las razones por las cuales se llegaría a esta medida e intentar resolverlas. Como última opción aceptar la cancelación.		
Plan de Contingencia: Organizar una reunión de urgencia entre el grupo completo y los clientes para intentar convencer a los últimos para que el proyecto siga adelante, buscando resolver cualquier problema que se les haya presentado.		

5.5 Plan de liberación de proyecto

Se pondrá el software en producción haciéndolo disponible a los usuarios mediante Heroku, se entregará junto con el software la documentación producida durante el desarrollo que permita la administración (manual de usuario) y facilite los cambios futuros. Asimismo, se le proveerá al cliente los datos de contacto de la empresa desarrolladora por cualquier inconveniente que surja y para coordinar el mantenimiento del programa cuando sea necesario.



6 Planes de procesos técnicos

6.1 Modelo de proceso

Se seguirá un modelo de proceso de desarrollo de software incremental e iterativo, gracias al cual se realizarán entregas parciales y regulares del resultado final del proyecto, priorizadas por el beneficio que aportan al cliente. Con este modelo las fases de desarrollo se solapan de manera que las tendremos todas juntas y se ejecutarán una y otra vez, hasta que se crea suficiente.

6.2 Métodos, herramientas y técnicas

Se seguirá la metodología ágil de desarrollo de software Scrum para completar el proyecto. Se realizarán “sprints” cada 2 semanas. Al principio de cada sprint se analizarán cuáles son las prioridades del cliente para determinar que funcionalidad debe desarrollarse durante esa iteración y se implementará de manera tal de tener un demo (ver Acrónimos) al final de cada sprint.

Una vez terminado el sprint se mostrará al cliente el avance del proyecto mediante estas, hasta el último sprint en el cual se entregará el software completo. En esta última entrega, el cliente decidirá si quiere agregar, quitar o modificar alguna funcionalidad o algún elemento de la parte gráfica.

El software será desarrollado utilizando Ruby on Rails, un framework de aplicaciones web escrito en el lenguaje de programación Ruby, siguiendo el paradigma de arquitectura Modelo Vista Controlador.

La edición del código será realizada mediante el editor de texto que cada integrante del equipo considere conveniente, preferentemente Sublime Text.

Las demos serán presentadas utilizando el servicio de computación en la nube Heroku, y utilizando PostgreSQL como SGBD.

El gerenciamiento del desarrollo del proyecto será utilizada la plataforma Pivotal Tracker. Asimismo utilizaremos la plataforma Github para realizar la gestión del código y el control de versiones.

6.3 Plan de infraestructura

Cada integrante del equipo dispondrá de una computadora en la que pueda escribir, compilar y ejecutar el código que escriba, luego su avance será sincronizado con el repositorio central del proyecto en la plataforma Github.

6.4 Plan de aceptación del producto

El cliente tendrá a su disposición el producto completo alojado en un servidor web y se asegurará que cumpla con los requisitos acordados previamente. Básicamente, el software deberá:

- Ser de interfaz intuitiva y ágil para que sea de fácil entendimiento.
- Permitir el alta de usuarios nuevos y su baja en la plataforma web (registro).
- Permitir el alta y baja de productos por parte de los usuarios registrados (inicio y fin de subasta).
- Contar con un sistema para comunicar a los usuarios involucrados una vez terminada una subasta.
- Permitir la navegación de las distintas subastas por parte de cualquier usuario.
- Permitir a cada usuario registrado enviar mensajes a los dueños del sitio (Cliente).
- Permitir que los usuarios registrados realicen consultas anónimas sobre las subastas de otros usuarios.



7 Plan de procesos de apoyo

7.1 Plan de administración de configuración

No Aplica.

7.2 Plan de pruebas

No Aplica.

7.3 Plan de documentación

Durante la duración del proyecto se realizarán tareas de documentación tanto de las características internas del software desarrollado como del proceso de construcción del sistema, desde su concepción hasta su puesta en producción definitiva. Entre estos documentos se incluyen:

- E1: entrevista realizada al Cliente durante la etapa de obtención de requerimientos.
- C1: cuestionario dirigido a los usuarios de la plataforma web desarrollada con el objetivo de conocer al usuario promedio y mejorar el sistema en base a sus opiniones.
- STD: Especificación de Requerimientos de Software escrito por Dead Programmers Society que contiene la propuesta inicial de la desarrolladora y será usado posteriormente como documento de referencia tanto por el Cliente como por la desarrolladora sobre las funcionalidades con las que debe contar el sistema.
- A1: planilla de administración provista por el Cliente que contiene datos de los usuarios y subastas que contiene el sistema actual y que deben ser migradas al nuevo.
- Manual de Usuario: será escrito por la Empresa desarrolladora para ser entregado al Cliente con el objetivo de facilitar la administración del sistema una vez se haya puesto en producción.
- Wiki: será el material de referencia para cualquier interesado en el proyecto, estará disponible en la página del repositorio de este en Github y será actualizada por el equipo desarrollador tanto con las funcionalidades esperadas del sistema, como con las actuales; asimismo contará con cualquier información relevante respecto del estado proyecto y el equipo de trabajo.
- PGP: este mismo documento servirá como guía durante todo el proceso de desarrollo permitiendo que cualquier integrante del equipo pueda consultar sus dudas.

7.4 Plan de aseguramiento de calidad

No Aplica.

7.5 Plan de revisiones y auditorías

No Aplica.

7.6 Plan de resolución de problemas

No Aplica.

7.7 Plan de administración de terceros

No Aplica.



7.8 Plan de mejoras en el proceso

No Aplica.

8 Planes adicionales

No Aplica.