|  |
| --- |
|  |

**Plan de Gestión de Proyecto**

**Proyecto: Bestnid**

|  |  |  |
| --- | --- | --- |
| lemon-dev_logo.png |  |  |

**Ficha del documento**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Revisión** | **Autor** | **Verificado** |
|  |  |  |  |

Documento validado por las partes en fecha:

|  |  |
| --- | --- |
| Por el cliente | Por la empresa suministradora |
|  |  |
| Aclaración | Aclaración |

1. **Introducción**
   1. **Resumen del Proyecto**
      1. **Propósito, alcance y objetivos.**

El propósito de la aplicacion es facilitar del negocio, haciendo más dinámico el proceso de subastas, y proporcionarle al mismo más escalabilidad.

Alcance:

El sistema Bestnid será un sistema para gestionar de manera automática la subasta de una gran diversidad de productos, creará perfiles de usuario para permitirle a los mismo mantener un seguimiento de sus operaciones y permitirá a los administradores ( ya sea los dueños o un tercero)  tener acceso a funciones específicas relacionadas con la administración del negocio.

Los objetivos del negocio son principalmente incrementar la cantidad de clientes, independizarse en la mayor parte de procesos manuales y tener un registro estadístico más preciso y accesible de las operaciones del mismo.

* + 1. **Supuestos y restricciones**

Garantizar la confidencialidad de los datos suministrados por los usuarios, así como garantizar la integridad y la disponibilidad de los datos almacenados.

El producto en su totalidad será entregado por partes, entregando en cada una, una nueva funcionalidad que se acoplará a la aplicación actual. Esto responde a la implementación de la metodología ágil de desarrollo llamada Scrum.

El sistema deberá ser fiel a los procesos y reglas que el negocio Bestnid ya tiene.

Al ser una aplicación Web será necesario utilizar varios lenguajes de programación, como HTML5, CSS3, Javascript y PHP, el cuál tendrá un rol muy importante para poder desarrollar esta aplicación, de tipo market place.

La fecha de entrega estipulada es el Sábado 18/7/2015.

Se supone un presupuesto de 4000 pesos argentinos para amortizar los gastos de capacitación, alquileres, infraestructura y adquisición de recursos para el proyecto. Esto será especificado en el resumen del presupuesto.

* + 1. **Entregables del proyecto**

-Demo 1 del Sistema - Sábado 06/6.

-Demo 2 del Sistema - Sábado 27/6.

-Demo 3 del Sistema - Sábado 18/7.

* + 1. **Calendario y resumen del presupuesto**

El proyecto durará en total 18 semanas con un presupuesto final de 31520 pesos argentinos.

1. **Documentos referenciados**

|  |  |  |  |
| --- | --- | --- | --- |
| **Referencia** | **Titulo** | **Fecha** | **Autor** |
|  | IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements | 1998 | IEEE |
|  | Entrevista ID:01 - Bestnid | 17/03/15 | Lemon-dev |
|  | Entrevista ID:02 - Bestnid | 31/03/15 | Lemon-dev |
|  | STD-830 Especificación de requisitos de software | 07/04/15 | Lemon-dev |

1. **Definiciones y acrónimos**

|  |  |
| --- | --- |
| **Término** | **Definición** |
| Usuario administrador | Usuario especial que puede acceder a funciones de gestión del sistema, tales como visualizar reportes estadísticos así como agregar o eliminar otros usuarios administradores |
| Usuario registrado | Usuario que ya ha ingresado sus datos al sistema y que puede acceder a funciones especificas. |
| Perfiles de usuario | Se refiere al “ lugar” o al espacio que tendrá cada usuario registrado dentro del sistema. |
| HTML5, CSS3, Javascript y PHP | Lenguajes de programación que serán utilizados para crear el sistema. |
| Tester | Miembro del equipo desarrollo del sistema  que se encarga de probar que las distintas funcionalidades del sistema trabajen correctamente. |
| Desarrollador | Miembro del equipo a que se encarga de la programacion y codificacion, para que el sistema coincida con los requisitos ya estipulados. |
| Aplicaciones web dinámicas | Las páginas web dinámicas son aquellas en las que la información presentada es generada dinámicamente por los distintos usuarios del sitio |
| Aplicaciones tipo market place | Aplicaciones en que su objetivo principal se basan en la compra y venta de productos y/o servicios. |
| Backup | Se refiere a una copia de los datos originales que se realiza con el fin de disponer de un medio para recuperarlos en caso de su pérdida |
| Base de datos | Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso |
| Algoritmo de cifrado | Es una tecnica informatica que asegura que los datos solo sean visibles para las usuarios que poseen los permisos necesarios para acceder a dicha información. |
| Cloud hosting | También conocido como almacenamiento en la nube, servicio de internet que permite a cualquier usuario, guardar, acceder, editar, a los datos desde cualquier dispositivo brindando tambien asi una forma de backup ya que dichos datos se encuentran alojados en lo que comúnmente se denomina “ la nube de internet” |
| Software Requirements Specification (SRS) | Un documento que describe por completo todas las funciones de un sistema que se propone y las limitaciones con las que debe funcionar . |
| IEEE Std 830-1998 | Estándar de especificación de requisitos de software (ERS), este documento describe las características de una adecuada ERS. |
| DER | Es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades. |
| Plan de Gestión de Proyecto (PGP) | Un documento que describe en detalle la forma en la cual será llevado a cabo el proyecto, dando información específica de aspectos tales como la división del trabajo, recursos que serán necesarios para llevar a cabo dicho proyecto , el modelo de desarrollo que será utilizado , etc |

1. **Organización del proyecto**
   1. **Interfaces externas**

El proyecto consta de tres integrantes, reduciendo la interacción interna de la empresa a la comunicación entre los mismos, siendo los integrantes del equipo: Santiago Jose Figueiras, Juan Cruz Almazán y Rodrigo Palacios Rusillo, como Desarrollador, Tester y Scrum Master respectivamente.

Y como contacto con la organización cliente, el contacto es directo con los Product Owner (ambos socios totalitarios del negocio).

* 1. **Estructura interna**

La estructura interna es organizada siguiendo el organigrama de control centralizado controlado y presenta la ventaja de que las decisiones son tomadas por una persona que esta a cargo, lo cual acorta el tiempo que se necesitaria en el caso de ser un esquema descentralizado democrático, en votar la desicion a tomar. Este organigrama se adecúa a las caracteristicas del proyecto ya que le da mayor velocidad.

* 1. **Roles y responsabilidades**

-Scrum master: Persona que lidera al equipo guiandolo para que cumpla las reglas y procesos de la metodología. Gestiona la reducción de impedimentos del proyecto y trabaja con el Product Owner para maximizar el ROI.

-Product owner (PO): Representante de los accionistas y clientes que usan el software. Se focaliza en la parte de negocio y el es responsable del ROI del proyecto (entregar un valor superior al dinero invertido). Traslada la visión del proyecto al equipo, formaliza las prestaciones en historias a incorporar en el Product Backlog y las reprioriza de forma regular.

-Team: Grupo de profesionales con los conocimientos técnicos necesarios y que desarrollan el proyecto de manera conjunta llevando a cabo las historias a las que se comprometen al inicio de cada sprint.

1. **Planes de administración del proceso**
   1. **Plan inicial**
      1. **Plan del personal**

Scrum Master Rodrigo Palacios Rusillo   10 hrs/semana

Desarrollador Santiago Jose Figueiras   10 hrs/semana

Tester   Juan Cruz Almazán 10 hrs/semana

* + 1. **Plan de adquisición de recursos**

- Compra de un único servidor Apache en principio - $2000.

- Alquiler oficina equipada para proyectos de desarrollo - 2 meses a $6000 x mes.

- Capacitación para desarrollo Web en HTML, CSS3 y Javascript - 1 mes 800 c/u.

- Capacitación para desarrollo de aplicaciones Web Dinámicas con PHP y MySQL. 1 mes 800 c/u.

* + 1. **Plan de entrenamiento del personal del Proyecto**

Los siguientes cursos fueron tomados por parte del equipo de desarrollo para fortalecer los conocimientos sobre estructuración y diseño de páginas web:

- Curso interactivo de Desarrollo Web de Acamica (HTML, CSS y Javascript) de [www.acamica.com](http://www.acamica.com/).

- Curso de HTML, CCS3 y Javascript del Centro de Estudiantes de la Facultad de Infórmatica de la Universidad Nacional de La Plata, dada por Milton Sosa.

El siguiente curso fue tomado por el equipo de desarrollo para conseguir un dinámico de la aplicacion web mediante la comunicación de la misma con la base de datos MySQL.

       -Curso interactivo de Introducción a PHP5 en [www.codeacademy.com](http://www.codeacademy.com/).

* 1. **Plan de trabajo** 
     1. **Principales actividades del proyecto**

Actividades de elicitacion de requerimientos:

1. Entrevistas: documentar las primeras dos entrevistas con el cliente.
2. SRS (Software Requirements Specification): documento detallado sobre la especificación del producto a desarrollar.
3. Pila de producto - Historias de usuario, en representación de las funcionalidades acordadas para la aplicación:
4. Registrarse: permitir que un usuario pueda registrarse y tener una cuenta para luego poder acceder a funcionalidades principales de la aplicación, como subastar u ofertar.
5. Iniciar sesión: que una vez que un usuario este registrado, pueda acceder a su cuenta.
6. Cerrar sesión: una vez que un usuario inicia sesión, debe poder cerrarla.
7. Eliminar cuenta: permitir que un usuario registrado elimine su cuenta.
8. Crear categoría: es necesario desarrollar esta funcionalidad con anticipación a otras como “Subastar”, debido a que es necesario tener las categorías definidas a la hora de permitir realizar la subasta de un producto.
9. Modificar categoria: si se permite crear una categoria, se debe permitir modificarlo.
10. Eliminar categoría: debe ser posible eliminar una categoría creada.
11. Publicar subasta: un usuario que ya esta registrado, debe poder (habiendo iniciado su sesión), publicar una subasta sobre un producto.
12. Eliminar subasta: un usuario registrado debe tener la posibilidad de eliminar una subasta propia por el motivo que sea, siempre que sea antes de la fecha de finalización.
13. Ordenar subastas: Como usuario, quiero poder ver todas las subastas activas del sitio para estar al tanto mismas.
14. Buscar por categoría: un usuario debe poder buscar subastas por categoría.
15. Consultar al subastador: un usuario registrado debe poder realizarle consultas al subastador sobre la subasta.
16. Responder a una consulta: un usuario registrado debe poder responder consultas sobre su subasta.
17. Ofertar: un usuario registrado debe poder ofertar en una subasta.
18. Eliminar oferta: un usuario registrado debe poder eliminar una oferta realizada, por el motivo que quiera, siempre y cuando sea antes de la fecha de finalización de la subasta.
19. Seleccionar oferta ganadora: un usuario registrado debe poder seleccionar una oferta ganadora entre las disponibles.
20. Ver mis ofertas pendientes: un usuario registrado debe poder ver en su cuenta las ofertas pendientes para su subasta.
21. Ver mis ofertas realizadas: un usuario registrado debe poder ver en su cuenta las ofertas que a realizado.
22. Reportar usuarios registrados entre dos fechas: un usuario administrador debe poder ver la cantidad de usuarios que se registraron entre dos fechas elegidas.
23. Reportar subastas exitosas entre dos fechas: un usuario administrador debe poder ver la cantidad de subastas que fueron realizadas con éxito entre dos fechas elegidas.

d. DER (Diagrama de Entidad de Relacion): Diagrama que especifica la forma en que se representarán los datos para la aplicacion.

e. Planificacion: el presente documento.

f. Planificación de riesgos: especificación sobre que riesgos se van a tratar, como se va a reducir la probabilidad de que ocurran, y como se tratarán en caso de que ocurran.

g. Diseño de interfaz: descripción del diseño de la interfaz de usuario, tipo, control y gestión de errores, ayudas, etc.

* + 1. **Asignación de esfuerzo**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actividad** | **Cantidad** | **Esfuerzo (hs)**  **Unitario** | **Esfuerzo (hs)**  **Subtotal** |
| a | 2 | 3 | 6 |
| b | 2 | 3 | 6 |
| c.1 | 2 | 2 | 4 |
| c.2 | 1 | 3 | 3 |
| c.3 | 1 | 3 | 3 |
| c.4 | 3 | 2 | 6 |
| c.5 | 3 | 3 | 9 |
| c.6 | 2 | 4 | 8 |
| c.7 | 1 | 2 | 3 |
| c.8 | 3 | 4 | 12 |
| c.9 | 2 | 2 | 4 |
| c.10 | 3 | 5 | 15 |
| c.11 | 3 | 4 | 12 |
| c.12 | 3 | 3 | 9 |
| c.13 | 2 | 3 | 6 |
| c.14 | 3 | 4 | 12 |
| c.15 | 2 | 2 | 4 |
| c.16 | 2 | 3 | 6 |
| c.17 | 2 | 4 | 8 |
| c.18 | 2 | 4 | 8 |
| c.19 | 3 | 4 | 12 |
| c.20 | 3 | 4 | 12 |
| d | 1 | 6 | 6 |
| e | 3 | 6 | 12 |
| f | 2 | 6 | 18 |
| g | 2 | 4 | 8 |

* + 1. **Asignación de presupuesto**

A una tarifa de 60 pesos la hora, con un total de 212 horas de desarrollo y una asignación de recursos de 18800 pesos argentinos, el presupuesto final sería de 31520 pesos argentinos.

* 1. **Plan de control**
     1. **Plan de control de requerimientos**

*No aplica*

* + 1. **Plan de control de calendario**

*No aplica.*

* + 1. **Plan de control de presupuesto**

*No aplica.*

* + 1. **Plan de control de calidad**

*No aplica.*

* + 1. **Plan de informe**

*No aplica.*

* + 1. **Plan de recolección de métricas**

*No aplica.*

* 1. **Plan de administración de riesgos**

*Entrega en posterior documento.*

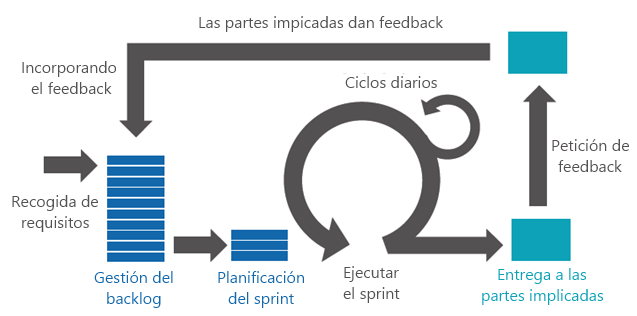
* 1. **Plan de liberación de proyecto**

*No aplica.*

1. **Planes de procesos técnicos**
   1. **Modelo de proceso**

La metodología a utilizar será el modelo de SCRUM. Este es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, 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.

El desarrollo se realiza de forma iterativa e incremental. Cada iteración, denominada Sprint, tiene una duración preestablecida de entre 2 y 4 semanas, obteniendo como resultado una versión del software con nuevas prestaciones listas para ser usadas. En cada nuevo Sprint, se va ajustando la funcionalidad ya construida y se añaden nuevas prestaciones priorizando siempre aquellas que aporten mayor valor de negocio.



* 1. **Métodos, herramientas y técnicas**

La metodología de desarrollo elegida es la metodología de Scrum, la cual es explicada en el punto 6.1, una metodología de desarrollo ágil muy utilizada en la actualidad para proyectos de corta duración y alta calidad.

Los lenguajes de programación en los que se implementará la aplicación son principalmente HTML5 (HyperText Markup Language 5) , que representará la estructura de los documentos de las páginas del sitio, un lenguaje utilizado y aceptado ampliamente para el desarrollo web. CCS3 (Cascading Style Sheets 3) el cuál representará la presentación y estilo del sitio. Javascript, utilizado para operaciones del lado del cliente, como animaciones en la página. Y PHP5, el lenguaje que le dará “vida” a la aplicación, el cuál será utilizado para realizar la comunicación entre la aplicación en los dispositivos, el servidor, y la base de datos. permitiendo así la persistencia de cuentas o sesiones así como de otros datos a lo largo del tiempo, y que la aplicacion sea dinámica.

La base de datos será una base de datos relacional, la elegida es MySQL ya que tiene gran soporte para su utilización con PHP5.

* 1. **Plan de infraestructura**

El lugar de trabajo es una oficina alquilada por la empresa Lemon-dev con la distribución perfecta para llevarse a cabo la metodología de SCRUM facilitando así el trabajo en equipo y el intercambio de ideas.

La oficina es un cuarto sencillo de solo 4 paredes de 30 metros cuadrados.Esta cuenta con una mesa al medio del espacio con 5 laptops listas para programar y testear el sistema, los desarrolladores se encuentran enfrentados mirando al centro de la mesa favoreciendo el contacto visual. También cuenta con un proyector que se encuentra arriba de la mesa, apuntando una de las paredes de la oficina a la que todos los desarrolladores pueden ver sin ningun problema.

Cabe aclarar que la oficina cuenta con una máquina expendedora de café instantáneo para elevar la calidad y el ambiente de trabajo de los desarrolladores.

* 1. **Plan de aceptación del producto**

*No aplica.*

1. **Plan de procesos de apoyo**
   1. **Plan de administración de configuración**

*No aplica.*

* 1. **Plan de pruebas**

*No aplica.*

* 1. **Plan de documentación**

-Entrevista ID:01 - Bestnid.

-Entrevista ID:02 - Bestnid.

-STD-830 Especificación de requisitos de software.

-DER (Diagrama de Entidad Relación) - Bestinid.

-Planificación de Riesgos - Bestnid.

-Diseño de Interfaz - Bestnid.

-Manual de Usuario - Bestnid.

* 1. **Plan de aseguramiento de calidad**

*No aplica*

* 1. **Plan de revisiones y auditorías**

*No aplica.*

* 1. **Plan de resolución de problemas**

*No aplica.*

* 1. **Plan de administración de terceros**

*No aplica.*

* 1. **Plan de mejoras en el proceso**

*No aplica.*

1. **Planes adicionales**

Plan de seguridad:

A continuación se encuentra una especificación de los elementos que protegerán al software de accesos, usos y sabotajes maliciosos:

* Empleo de algoritmos de cifrados para contraseñas de clave simétricas, como RSA.
* El servidor en el cual Bestnid opere va a tener su propios métodos de seguridad contra accesos de escritura no autorizados.
* Existencia de un sistema de redundacia de informacion para asegurar la integridad de los datos.

Plan de mantenibilidad:

El sistema requiere de poco mantenimiento, el cual debe ser llevado a cabo por un desarrollador. El mantenimiento se deberá hacer una vez por mes, revisando informes de errores y actualizando la versión de la aplicación.

Plan de backup:

La documentación del proyecto está respaldado en el sistema de cloud hosting de Google, Google Drive, al que tienen accesos todos los miembros del equipo.

La misma también estará alojada en un repositorio en Github, junto con el código fuente, el cuál es clonado (descargado) por cada miembro del equipo a un repositorio local, lo que hace que toda la información esté respaldada de forma distribuida entre los miembros del equipo.