

Propuesta Técnica

9 de septiembre de 2013

Índice

1. Propuesta de servicios	3
1.1. Propósito	3
1.2. Objetivo	3
1.2.1. Objetivo general	3
1.2.2. Objetivos específicos	3
1.2.3. Tecnología a utilizar	3
1.3. Tipos de usuarios	6
1.4. Funcionalidades globales	6
2. Metodología de desarrollo	6
2.1. Roles en la metodología SCRUM	8
2.1.1. Rol product owner	8
2.1.2. Cliente	8
2.1.3. Scrum Master	8
2.1.4. Scrum Team	8
2.2. Componentes de SCRUM	9
2.2.1. Sprint o Iteración	9
2.2.2. La duración del sprint	10
2.2.3. La meta del sprint	10
2.2.4. Product Backlog	10
2.2.5. Sprint Backlog	10
2.3. Reuniones scrum	10
2.3.1. Planificación del Sprint	10
2.3.2. Seguimiento del Sprint o reunión diaria	11
2.3.3. Revisión del Sprint	11

2.3.4.	Retrospectiva	11
2.3.5.	Control de calidad	11
2.3.6.	Documentación	11
2.3.7.	Capacitación	11
2.3.8.	Transferencia de tecnología	12
2.3.9.	Implantación y puesta en marcha	12
2.4.	Ventajas de scrum	12
2.4.1.	Simple	12
2.4.2.	Ágil	12
2.4.3.	Flexible	12
2.4.4.	Colaborativa:	12
3.	Planificación	13
3.1.	Detalle de los entregables	14
4.	Plazo de conclusión del contrato	14
5.	Estimación de esfuerzo	14
5.1.	Estimación usando Puntos de Función (PF)	14
5.1.1.	Cálculo de puntos de función sin ajustar	14
5.1.2.	Ajuste de complejidad técnica	15
5.1.3.	Cálculo de los puntos de función ajustados	15
5.1.4.	Cálculo del esfuerzo	16
5.1.5.	Cálculo de la duración del proyecto	16
6.	Propuesta económica	16
6.1.	Costos operativos	16
6.2.	Costos del personal	17
6.3.	Costo total	17
7.	Plan de pagos	18
8.	Gestión de riesgos	18

1. Propuesta de servicios

1.1. Propósito

La presente propuesta técnica tiene como propósito exponer nuestra alternativa de solución al problema de la empresa de TIS para una juez virtual para la Olimpiada Internacional de Informática. Además el documento servirá de guía para el proceso de desarrollo del producto de software y formará parte indivisible del contrato estipulado entre la empresa TIS y el proponente, esperando que cumpla y satisfaga con las exigencias emitidas por parte de la empresa licitante.

1.2. Objetivo

1.2.1. Objetivo general

Desarrollar un juez virtual, que funcione a nivel departamental, que permita a olimpistas enfrentarse en un modo de evaluación, como el que se usa la competencia mundial.

1.2.2. Objetivos específicos

- Desarrollo de componentes de gestión de usuarios.
- Desarrollo de componentes del modulo de evaluador.
- Desarrollo de componentes del gestión de competición.
- Desarrollo de componentes del gestión de problemas
- Realización de la gestión de calidad del sistema integrado

1.2.3. Tecnología a utilizar

Para el cumplimiento de los objetivos de esta propuesta y las condiciones establecidas en el pliego de especificaciones de la empresa TIS, se ha considerado desarrollar el sistema del «Juez Virtual» con las siguientes herramientas de software con una pequeña descripción:

Herramienta	Software
Servidor web	Apache
Gestor de base de datos	MySQL
Lenguaje de programación	PHP
Framework de apoyo	Symfony
Biblioteca de apoyo	jQuery
Sistema de control de versiones	Git
Entorno integrado de desarrollo	Netbeans
Composición de documentación	LaTeX
Editor de texto plano	vim
Diseño de Base de Datos	MySQL Workbench

El Servidor web Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation, es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual, también presenta características altamente configurables.

La licencia que usa es Apache License, que es una licencia de software libre creada por la Apache Software Foundation (ASF).

Gestor de base de datos MySQL se ejecuta como un servidor que proporciona múltiples acceso de los usuarios a una serie de bases de datos, con términos de Licencia Pública General GNU, también cuenta con un uso comercial, varias ediciones de pago están disponibles, y ofrecen una funcionalidad adicional.

Lenguaje de programación PHP es acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje de 'scripting' de propósito general y de código abierto que está especialmente pensado para el desarrollo web y que puede ser embebido en páginas HTML. Su sintaxis recurre a C, Java y Perl. La meta principal de este lenguaje es permitir a los desarrolladores web escribir dinámicamente y rápidamente páginas web generadas, aunque se puede hacer mucho más con PHP.

La licencia PHP es la licencia bajo la que se publica el lenguaje de programación PHP. De acuerdo a la Free Software Foundation es una licencia de software libre no copyleft y una licencia de código abierto según la Open Source Initiative. Debido a la restricción en el uso del término "PHP", no es compatible con la licencia GPL.1.

Framework Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón Modelo Vista Controlador. Para

empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

Cuenta con la licencia MIT es una de tantas licencias de software que ha empleado el Instituto Tecnológico de Massachusetts (MIT, Massachusetts Institute of Technology) el texto de la licencia no tiene copyright, lo que permite su modificación.

Biblioteca de JavaScript jQuery permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones (FLV) y agregar interacción con la técnica AJAX a páginas web.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos

Sistema de control de versiones Git es un software, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.

Cuenta con una licencia GNU GPL v2 más ampliamente usada en el mundo del software y garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software.

Netbeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo (dentro los cuales completa a PHP), es un producto libre y gratuito sin restricciones de uso. Cuenta con dos licencias Common Development and Distribution License (Licencia Común de Desarrollo y Distribución o CDDL) y GNU General Public License 2

Composición de documentación L^AT_EX orientado especialmente a la creación de libros, documentos científicos y técnicos que contengan fórmulas matemáticas.

LaTeX es software libre bajo Licencia Pública del Proyecto LaTeX(LPPL) .

Vim es un editor de texto plano presente en todos los sistemas UNIX, se distribuye bajo una licencia Charityware (del inglés charity: caridad) compatible con Licencia pública general de GNU.

MySQL Workbench es una herramienta visual de diseño de bases de datos que integra desarrollo de software.

Cuenta con licencia General Public License (GNU) Las versiones a ser utilizadas de Apache, MySQL y PHP serán definidas de acuerdo al software instalado en el laboratorio de TIS.

1.3. Tipos de usuarios

1. Olimpista.
2. Administrador.
3. Miembro de comité académico.

1.4. Funcionalidades globales

Gestión de usuarios La gestión de usuarios consta de la creación, modificación y depuración de usuarios, los tipos de usuarios se especifican en 1.3. Cada tipo de usuario tendrá establecido sus privilegios con respecto al Sistema.

Modulo evaluador Consta de un sub-sistema que recibe una solución a un problema y ejecuta los casos de prueba correspondientes al problema, luego devuelve un puntaje de acuerdo a los resultados de la ejecución de los casos de prueba.

Gestión de competición Permite crear una competición entre olimpistas, se podrá seleccionar el modo de la competición, es decir si la competición va ser verdadera o solo sera una practica. Una vez creada la competición se la podrá dar inicio y también se la podrá finalizar. También se contempla la creación de reportes relacionados a la competición, como ser un ranking de una competición determinada.

Gestión problemas Permite la subida nuevos problemas, la modificación de problemas y dar de baja un problema.

2. Metodología de desarrollo

Nuestro marco de trabajo para la gestión y desarrollo de software estará basada en un proceso iterativo e incremental, utilizando un componente en entornos basados en el desarrollo Ágil de software SCRUM.

El **modelo incremental** se basa en incrementos, cada incremento sigue el modelo cascada y posee características de retro alimentación.

Estos modelos iterativos se basan en dividir el proyecto de desarrollo en varias etapas, llamadas iteraciones. Las iteraciones son cortas (unas cuantas semanas, excepto en proyectos enormes) y su duración es fija (no puede alargarse: si hay retrasos, estos se incluyen en otra iteración).

La idea central es que, en cada una de esas iteraciones, se construye una parte pequeña del sistema (esto se llama a veces "desarrollo incremental"). Para esa parte del sistema, se realiza todo el proceso: análisis, diseño, programación y pruebas. Se acaba la iteración con un ejecutable que incluye todas las partes del sistema construidas hasta el momento. Los aspectos del sistema con más riesgo (por ejemplo, la arquitectura) se construyen en las primeras iteraciones. El esquema de un modelo iterativo se muestra gráficamente en la siguiente figura.

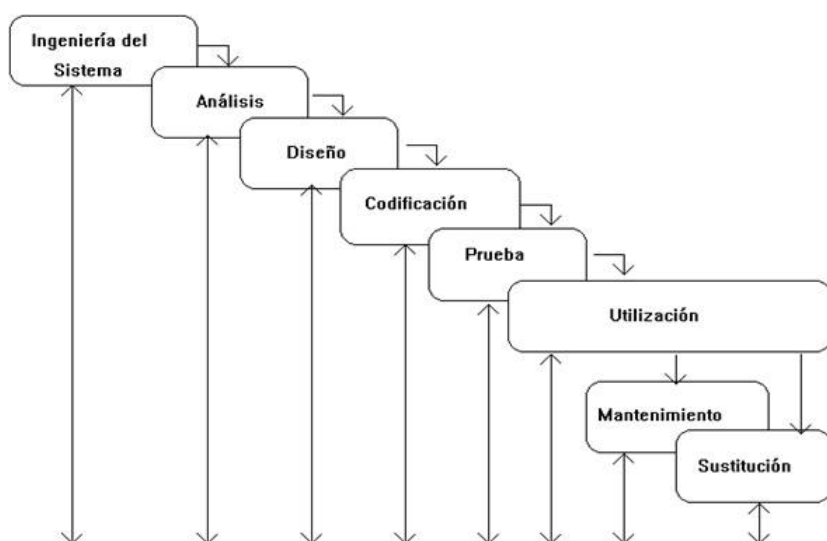


Figura 2.1

En este sentido podemos considerar SCRUM como el paradigma de metodología de desarrollo ágil, definiendo la forma de abordar un proceso de desarrollo de software de forma ágil y liviana, a través de la descripción de un conjunto de roles, componentes y organización de la actividad diaria.

2.1. Roles en la metodología SCRUM

2.1.1. Rol product owner

Persona que trabaja junto con el **Cliente** con conocimientos de negocio e informática sus responsabilidades son :

- Representante de todas las personas interesadas en los resultados del proyecto.
- Es quien coordina las actividades administrativas o funcionales que deben realizar los usuarios en implementación a las actividades y requerimientos del equipo de desarrollo.
- Es quien se encarga del financiamiento del proyecto y de que los requerimientos o necesidades de los usuarios sean satisfechos en tiempo y alcance.

Sobre el Product Owner recae la responsabilidad de definir el conjunto de requerimiento.

2.1.2. Cliente

- Es el **representante** de todas las personas interesadas en los resultados del proyecto (internas o externas a la organización, promotores del proyecto y usuarios finales [idealmente también debería ser un usuario clave] o consumidores finales del producto).
- Definir los objetivos del producto o proyecto.
- Colaborar con el equipo para planificar, revisar y dar detalle a los objetivos de cada iteración.

2.1.3. Scrum Master

Responsable de que el equipo desempeñe sus actividades de una manera eficiente y organizada, también elimina obstáculos que impiden que el equipo alcance el objetivo del sprint.

2.1.4. Scrum Team

Es un equipo formado de 7 (+/-) 3 integrantes que se encargan de revisar los requerimientos, la tecnología disponible y evalúan los conocimientos para determinar de manera colectiva cómo incrementar la funcionalidad del software, son responsables

de elegir las historias de usuarios para una determinada iteración y se caracteriza por ser auto gestionado y auto organizado.

El cliente y el equipo tienen que *acordar la definición de completado* de los objetivos / requisitos en el proyecto:

- Debe asegurar que el incremento de producto es potencialmente entrégale al cliente al finalizar cada iteración, que no hay tareas pendientes que puedan impedir utilizar los resultados del proyecto lo antes posible. De este modo, el cliente podrá tomar decisiones correctas cuando al final de cada iteración el equipo le haga una demostración de los requisitos completados: cambiar las prioridades en función de la velocidad de desarrollo, solicitar una entrega del producto desarrollado hasta ese momento, etc.
- Debe incluir lo necesario para considerar de manera clara que el cliente obtendrá lo que necesita según sus criterios de entregables y de calidad (producto construido, probado, documentado, refactorizado para conseguir calidad interna/mantenibilidad, etc.).

Además de esta definición de completado, a cada objetivo hay que asociarle sus condiciones de satisfacción, preferiblemente en forma de casos de prueba de aceptación, en el momento de crear la lista de requisitos priorizada (Product Backlog).

Notar que la definición de completado también sirve como base para identificar las tareas necesarias para conseguir cada objetivo/requisito, en la reunión de planificación de la iteración (Sprint planning). Para cada objetivo se crearán más tareas que la definición de completado (o menos) y con más significado. Por ejemplo, respecto a que el objetivo tiene que estar construido, pueden aparecer varias tareas, del estilo construir el componente X, modificar la pantalla Y , modificar la BBDD, preparar el script de carga, etc.

2.2. Componentes de SCRUM

Término general para cualquier tipo de información creada, producida, cambiada o utilizada por el equipo Scrum.

2.2.1. Sprint o Iteración

Es la base de desarrollo de Scrum, su duración se encuentra entre 15 a 30 días, solo el Scrum Master puede abortar el sprint si lo considera no viable por alguna de estas razones:

- La tecnología establecida no funciona
- El equipo ha tenido interferencias

2.2.2. La duración del sprint

Los sprint cortos permite al equipo ser ágil es decir cambiar de dirección frecuentemente, pero los sprint largos tampoco están mal el equipo consigue más tiempo para realizar sus actividades y recuperarse de los problemas que surjan en el sprint y así cumplir la meta.

2.2.3. La meta del sprint

La meta del sprint debería responder a la pregunta fundamental ¿Por qué realizamos este sprint? Las respuestas pueden ser «**Cumplir con las expectativas del cliente.**»

2.2.4. Product Backlog

Contiene los **objetivos/requisitos** de alto nivel del producto o proyecto. **Product Backlog** se suelen expresar en forma de historias de usuario. Para cada **objetivo/requisito** se indica el valor que aporta al cliente y el coste estimado de completarlo.

2.2.5. Sprint Backlog

Especifica la serie de tareas que se van a desarrollar, que descompone las funcionalidades del Product backlog en las tareas necesarias para construir un incremento o una parte completa y operativa del proyecto.

2.3. Reuniones scrum

2.3.1. Planificación del Sprint

Para la planificación se deberán tomar como base las prioridades y necesidades del cliente, se determina cuáles y cómo van a ser las funcionalidades que incorporara el producto tras el siguiente sprint, consta de dos partes:

En la **primera** parte se decide qué elementos de la pila de producto se van a desarrollar. En la **segunda** parte se desglosa estos para determinar las tareas necesarias y estimar el esfuerzo para cada una, también procedemos a asignarlas a cada uno de los miembros del equipo.

2.3.2. Seguimiento del Sprint o reunión diaria

Es una reunión breve no más de 15 minutos en la que cada miembro del equipo dice las tareas en las que está trabajando, si se ha encontrado o prevé encontrarse con algún impedimento o actualizar sobre la pila de sprint las ya terminadas.

2.3.3. Revisión del Sprint

Reunión que se realiza al final del sprint en la que el equipo presenta al propietario del producto.

2.3.4. Retrospectiva

Es una reunión en la que debemos analizar las cosas que hemos hecho bien y las cosas que hemos hecho mal durante el sprint.

2.3.5. Control de calidad

Control de calidad implica vigilar el proceso de desarrollo de software en cada iteración para asegurar que se siguen los procedimientos y los estándares de garantía de calidad.

2.3.6. Documentación

La Documentación permitira la facilidad de utilización por parte del usuario, los documentos funcionales y de capacitación seran realizados en la ultima iteración.

2.3.7. Capacitación

La etapa de capacitación está orientada a los usuarios del sistema. Además, se orienta una parte de esta capacitación al o los administradores del sistema. El objetivo final del plan de capacitación es preparar a los distintos usuarios, en el uso y manejo de cada una de las funcionalidades inherentes y necesarias para desarrollar sus labores con el apoyo.

2.3.8. Transferencia de tecnología

La transferencia de tecnología se realiza al momento al culminar el contrato.

2.3.9. Implantación y puesta en marcha

El proceso de implantación, se considera las siguientes actividades:

- Organización del equipo de trabajo del proceso de Implantación
- Instalación y Configuración de Sistema
- Inicio de Operación
- Soporte de puesta en marcha

2.4. Ventajas de scrum

2.4.1. Simple

Se centra especialmente en facilitar el desarrollo rápido, por lo que su complejidad (por ejemplo desde el punto de vista de la documentación a generar o de la organización de equipos) se ha tratado de reducir al máximo.

2.4.2. Ágil

La división del trabajo en pequeñas unidades funcionales (sprints) permite mantener una política de entregas frecuentes de software que ofrecen una visión clara del estado del proceso y permite la introducción de modificaciones.

2.4.3. Flexible

Todo el desarrollo se contempla como un ciclo de iteraciones continuas de desarrollo, lo que facilita la introducción de modificaciones sobre la marcha, mejorando continuamente el proceso.

2.4.4. Colaborativa:

El planteamiento, desde el punto de vista de la organización del equipo, resulta bastante horizontal (en contraposición a una organización jerárquica férrea), otorgando a los miembros del equipo de desarrollo una elevado grado de autonomía y auto-organización de su trabajo.

3. Planificación

De acuerdo al marco de trabajo SCRUM se define una planificación dinámica basada en sprint, que son iteraciones de tiempo fijo. En la reunión para la planificación del sprint se decide, junto con el cliente del producto, el conjunto de historias de usuarios (estos pueden ser vistos como requerimientos funcionales) que van a ser desarrolladas durante el Sprint, considerando solo los días laborales definidos en la parte A.

Descripción	Fecha
Firma de contrato	09 de septiembre, 2013
Inicio de Iteración 0	12 de septiembre, 2013
Fin de Iteración 0	18 de septiembre, 2013
Iteración 1 - Total días: 13	19 de septiembre, 2013
Reunión con cliente	26 de septiembre, 2013
Fin de Iteración	02 de octubre, 2013
Iteración 2 - Total días: 13	03 de octubre, 2013
Reunión con cliente	10 de octubre, 2013
Fin de Iteración	16 de octubre, 2013
Demo de entregable 1	17 de octubre, 2013
Iteración 3 - Total días: 13	17 de octubre, 2013
Reunión con cliente	24 de octubre, 2013
Fin de Iteración	30 de octubre, 2013
Iteración 4 - Total días: 13	31 de octubre, 2013
Reunión con cliente	07 de noviembre, 2013
Fin de Iteración	13 de noviembre, 2013
Demo de entregable 2	14 de noviembre, 2013
Iteración 5 - Total días: 19	14 de noviembre, 2013
Reunión con cliente	21 de noviembre, 2013
Reunión con cliente	28 de noviembre, 2013
Fin de Iteración	03 de noviembre, 2013
Demo de entregable 3	04 de diciembre, 2013

3.1. Detalle de los entregables

Entregable 1	Módulo evaluador y gestión de usuarios.	17 de Octubre, 2013
Entregable 2	Gestión de competición y gestión de problemas.	14 de Noviembre, 2013
Entregable 3	Sistema completo, integrado con los anteriores entregables. Documentación técnica, de instalación y de usuario.	04 de Diciembre, 2013

Esta tabla de entregables puede estar sujeta a cambios, en cuanto a fechas de entregas y los productos entregables en si, previo acuerdo con Cassiopeia S.R.L. y el cliente.

4. Plazo de conclusión del contrato

Una vez entregado el software y firmado un documento de conformidad por parte de la empresa TIS, se dará por concluida la relación contractual con la empresa TIS. Tales acciones en ningún caso podrán exceder el día 6 de diciembre del 2013.

5. Estimación de esfuerzo

5.1. Estimación usando Puntos de Función (PF)

Es una métrica que permite traducir en un número el tamaño de la funcionalidad que brinda un producto de software desde el punto de vista del usuario, a través de una suma ponderada de las características del producto.

5.1.1. Cálculo de puntos de función sin ajustar

EI : Procesos en los que se introducen datos y que suponen la actualización de cualquier archivo interno.

EO: Procesos en los que se envía datos al exterior de la aplicación.

EQ: Procesos consistentes en la combinación de una entrada y una salida, en el que la entrada no produce ningún cambio en ningún archivo y la salida no contiene información derivada.

ILF: Grupos de datos relacionados entre sí internos al sistema.

EIF: Grupos de datos que se mantienen externamente.

Componente	Bajo	Medio	Alto	Total
EI	4 * 3	0 * 4	0 * 6	12
EO	3 * 4	0 * 5	0 * 7	12
EQ	2 * 3	0 * 5	7 * 6	48
ILF	0 * 7	4 * 10	0 * 15	40
EIF	0 * 5	0 * 7	0 * 10	0
Total				112

5.1.2. Ajuste de complejidad técnica

Nro de factor	Descripción	Valor
1	Comunicación de Datos	1
2	Proceso Distribuido	0
3	Objetivos de Rendimiento	1
4	Configuración de Explotación Compartida	0
5	Tasa de transacciones	1
6	Entrada de Datos en Línea	3
7	Eficiencia con el Usuario Final	2
8	Actualizaciones en Línea	3
9	Lógica de Proceso Interno Compleja	0
10	Reusabilidad del Código	4
11	Conversión e Instalación contempladas	1
12	Facilidad de Operación	1
13	Instalaciones Múltiples	1
14	Facilidad de Cambios	2
Ajuste de complejidad técnica (ACT)		20

5.1.3. Cálculo de los puntos de función ajustados

$$ACT = 20$$

$$PFSA = 112$$

$$PFA = PFSA * [0,65 + [0,01 * ACT]]$$

$$PFA = 112 * [0,65 + [0,01 * 20]]$$

$$PFA = 95,2$$

5.1.4. Cálculo del esfuerzo

$$PFA = 95,2$$

$$\text{Horas por Punto de Función} = 7$$

Esfuerzo	$PFA / (1 / \text{Horas por PF})$ $95,2 / (1 / 7)$ $= 666,4 \text{ [horas/persona]}$
-----------------	--

5.1.5. Cálculo de la duración del proyecto

Número de personas en el proyecto = 3 personas

Duración	$\text{Esfuerzo} / (\text{número de personas en el proyecto})$ $(666,4 \text{ [horas/persona]}) / (3 \text{ [personas]})$ $= 222,13 \text{ [horas] por miembro}$
-----------------	--

Las horas de trabajo son de 20,45 horas a la semana por persona.

Horas al mes por trabajador	$20,45 \text{ [horas/semana]} * 4 \text{ [semanas/mes]}$ $= 81,8 \text{ [horas/mes]}$
------------------------------------	--

Duración en meses	$(\text{Duración en horas}) / (\text{Horas al mes})$ $(222,13 \text{ [horas]}) / (81,8 \text{ [horas/mes]})$ $= 2,71 \text{ [meses]}$
--------------------------	---

6. Propuesta económica

6.1. Costos operativos

Estos costos se refieren a los servicios básicos y otros suministros, un calculo aproximado en bolivianos por mes sería:

Luz	300
Agua	150
Internet	1000
Teléfono	100
Suministros de oficina	100
Agua embotellada	40
Total	1690

6.2. Costos del personal

Estos costos se refieren al salario del personal de la empresa, un calculo aproximado en bolivianos por mes sería:

Gerente general	6960
Secretaria	2784
Limpieza	1044
Administración	4176
Contabilidad	3480
Ingenieros (3)	14616
Total	33060

6.3. Costo total

Las horas totales a trabajarse en el proyecto son:

222, 13 [horas]

La suma de costos del personal y los costos operativos:

$$33060 + 1690 = 34750 \text{ Bs.}$$

En el siguiente cálculo se detalla el costo por hora:

$$((34750)/22 \text{ horas laborales})/8 \text{ horas al día} = 197,44 \text{ Bs./Hora}$$

Cálculo del costo total:

$$197,44 \text{ Bs./Hora} * 222,13 \text{ Horas} = 43857,34 \text{ Bs.}$$

Costo	Importe (Bs.)
Costo total	43857,34
Precio de venta incluyendo utilidades (30 %)	13157,20
Precio final	57014.54

El precio final sería de: Cincuenta y Siete Mil Catorce 54/100 Bs.

7. Plan de pagos

Los pagos se realizarán luego de que nuestra empresa haya realizado una entrega al cliente.

Pago	Evento	Monto (%)
1	Primera entrega 17 de octubre, 2013	25.0 %
2	Segunda entrega 14 de noviembre, 2013	45.0 %
3	Ultima entrega 04 de diciembre, 2013	30.0 %

8. Gestión de riesgos

La gestión de riesgos nos ayuda a prevenir posibles sucesos que eviten que se desarrolle y entregue el software en el tiempo y presupuesto establecido por la empresa. La identificación de riesgos y el plan de contingencias se presenta a continuación:

Identificación de riesgos

Nro. Riesgo	Descripción	Proba.	Impacto	Consecuencias
R1	Incumplimiento de actividades de algún socio de la empresa	40 %	Alto	Posibles retrasos en la planificación
R2	Cambios de gran importancia en los requerimientos	60 %	Medio	Se producen retrasos en la planificación previamente realizada
R3	Problemas con el hardware u otros dispositivos necesarios	5 %	Bajo	Retrasos en el desarrollo del software.
R4	Problemas con la rotación de personal	25 %	Alta	Retrasos en el desarrollo del software.

Plan de contingencias

Nro. Ries-go	Estrategia a Desarrollar
R1	Se realizara una breve planificación para sub sanar este inconveniente evitando de esta forma reformular toda la planificación.
R2	Se realizaran entrevistas planificadas para contemplar el impacto en el desarrollo del software.
R3	Tenemos un contrato con la empresa Infobest para dar soluciones rápidas a problemas técnicos.
R4	La empresa realiza contratos inmediatos de profesionales cualificados