<PetAPP>

Plan de Gestión de la Configuración

Versión <1.6>

Historial de Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| <18/abr/2016> | <1.0> | <Se completaron de forma parcial punto 1, punto 2 y punto 3> | <Luis Gianpierre Portella> |
| <19/abr/2016> | <1.1> | <Se terminaron algunos puntos (revisiones y consultas pendientes), se agregó la tabla RACI > | <Luis Gianpierre Portella – Alexander Medina Escalante> |
| <01/may/2016> | <1.2> | <Se agregó el alcance del documento, el glosario, se reajusto las responsabilidades del equipo de trabajo y se adicionaron elementos de gestión de la configuración> | <Hans Soto Rojas – Carlos Zarate Carpio> |
| <02/may/2016> | <1.3> | <Se actualizaron elementos de gestión de la configuración> | <Gustavo Feijoo Oliveira> |
| <06/may/2016> | <1.4> | <Se completaron y actualizaron varios puntos por corregir y agregar, se incluyeron archivos adicinales: RACI (esta adjunto con el proyecto, ya no es una imagen), diagrama Entidad Relación> | <Gianpierre Portella Bravo Hans Soto Rojas  Carlos Zarate Carpio> |
| <20/jun/2016> | <1.5> | <Se completaron cambios en línea base> | <Hans Soto Rojas  Carlos Zarate Carpio> |
| <20/jun/2016> | <1.6> | <Se completaron cambios en línea base, Control de cambios y Control y Auditoría> | <Hans Soto Rojas> |

Índice

1. Introducción

1.1 Propósito del Documento

1.2 Alcance del documento

1.3 Definiciones y acrónimos.

2. Gestión de la configuración de software

2.1 Organización, Responsabilidades

2.2 Tools, Environment, and Infrastructure

3. Gestión de la configuración

3.1 Identificación de la configuración

3.1.1 Estándar de nomenclatura

3.1.2 Elementos de gestión de la Configuración

3.1.3 Roles y Responsabilidades

3.1.4 Líneas Base

3.2 Control de Cambios

3.2.1 Proceso de solicitud de cambios y aprobación

3.2.2 Comité de control de Cambios.

3.3 Contabilidad y Auditoría de la Gestión de la Configuración

3.3.1 Almacenamiento del proyecto

3.3.2 Proceso de generación de releases.

3.3.3 Reports and Audits

Plan de Gestión de la Configuración

# Introducción

## Propósito del Documento

## Este documento tiene como propósito realizar la planificación de evolución del producto software propuesto en cada versión de su desarrollo, y empleando buenas prácticas de gestión de la configuración (SCM) se logrará construir versiones de esta más adecuadas con el propósito de llegar a tener un producto final que se adecue a las necesidades de un cliente.

## Alcance del Documento

## El alcance de este documento es tener un plan de gestión de la configuración en el cual se encuentres todos los parámetros necesarios que permitan seguir una línea de desarrollo estable con posibilidad a cambios a futuro sin mucho esfuerzo. Los detalles a contener podrían ser: herramientas a usar, software a utilizar, especificación de los responsables con su respectiva área a manejar, etc.

## Definiciones y acrónimos.

## Para obtener un mayor detalle de las siglas y palabras remitirse al documento de Glosario (GLOSS).

# DB: Data Base (Base de datos).

# IDE: Integrated Development Environment (Ambiente de Desarrollo Integrado).

# RF: Fundamental Requeriment (Requerimiento Fundamental).

# RUP: Rational Unified Process (Proceso Racional Unificado).

# SCM: Software Configuration Management (Gestión de la Configuración de Software).

# SCRUM: Es un proceso en el que se aplican de manera regular [un conjunto de buenas prácticas](https://proyectosagiles.org/fundamentos-de-scrum) para trabajar colaborativamente, en equipo, y obtener [el mejor resultado posible](https://proyectosagiles.org/beneficios-de-scrum) 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](https://proyectosagiles.org/historia-de-scrum). Es una palabra, no siglas.

# UI: User Interface (Interfaz de Usuario).

# Gestión de la configuración de software

## Organización, Responsabilidades

Encargado de Cliente:

* Carlos Zarate Carpio.
* Hans Soto Rojas.
* Rances Ramos Ramírez.
* Gustavo Feijoo Oliveira.
* Elvis Maldonado Rodriguez.

Encargado de Servidor:

* Luis Gianpierre Portella.
* Alexander Medina Escalante.

Grupo PetAPP:

* Administrador: Rances Ramos Ramírez.
* Scrum Master: Carlos Zarate Carpio.
* Server Team Leader: Alexander Medina Escalante.
* Issue Manager: Rances Ramos Ramírez.
* Client Team Leader: Hans Soto Rojas.
* Server Testing Leader: Luis Gianpierre Portella.
* Product Owner: Luis Gianpierre Portella.
* Client Testing Leader: Elvis Maldonado Rodriguez.
* Integration Manager: Gustavo Feijoo Oliveira.

## Tools, Environment, and Infrastructure

Este proyecto estará hecho con arquitectura Cliente-Servidor, realizando el server en PHP brindando servicios que serán consumidos por un cliente en ASP.Net.

Las herramientas necesarias para realizar este proyecto serán:

* IDE PHPStorm por el lado del servidor en lenguaje PHP.
* IDE Visual Studio 2013 o superior por el lado del cliente en lenguaje C-sharp con el framework ASP.Net
* El gestor de BD en MySQL usando la utilidad PHPMyAdmin.
* El servidor de aplicaciones web será APACHE en su versión 7.0.

|  |  |  |
| --- | --- | --- |
| **Recurso**  **Hardware** | **Especificación** | **Utilidad** |
| Servidor | Procesador de un solo núcleo  RAM 8GB  Disco Duro 16 GB  Sistema Operativo Ubuntu | Se utilizará como un hosting de PHP. Aquí se desplegará el servicio web para ser consumido y se almacenará la información tratada en una base de datos local de MySQL. |
| Cliente | Procesador Intel core i5  RAM 4GB  Disco Duro 500GB  Sistema Operativo | Se desplegará la aplicación web cliente en ASP.Net para consultar la información de la BD en MySQL y visualizar la información en pantalla (Navegador). |

|  |  |
| --- | --- |
| **Recurso**  **Software** | **Descripción** |
| Sistema operativo | Ubuntu 15.04 |
| Lenguajes de Programación | PHP 5.5 (Apache2), C# - HTML5 (ASP.Net) y MySQL(Apache) |
| Ofimática | Microsoft Office 2013 (Word, Excel y Power Point) |
| Entorno de desarrollo Integrado | * PHPStorm. * SublimeText. * SourceTree. * BitBucket. * Visual Studio 2013 o superior. * PHPMyAdmin. |
| Base de datos | MySQL 5.7 |
| Control de Versiones | Git (https://github.com/gianpierrepb/PetAPPEVO\_Cliente.git) |
| Repositorio | Github. Usuarios:   * Hans Soto Rojas: hanssr * Carlos Zarate Carpio: carlosZarate20 * Rances Ramos Ramírez: * Gustavo Feijoo Oliveira: gusXvo * Elvis Maldonado Rodriguez: (emrodriguez) * Luis Gianpierre Portella: * Alexander Medina Escalante: |
| Integración Continua | Jenkins |

# Gestión de la configuración

## Identificación de la configuración

### Estándar de nomenclatura

Los artefactos de software serán nombrados, marcados e identificados dentro del esquema de gestión de la configuración de la siguiente manera:

1. Historias de Usuario: Por su estándar “HU”, ejemplo: HU\_01.
2. Base de Datos: Por su estándar “DB”, ejemplo: DB\_01.
3. Interfaz de usuario: Por su estándar “UI”, ejemplo: UI\_01.
4. Plan de Gestión de la Configuración: Por su abreviatura “PGC”, ejemplo : PGC\_01
5. Backlog del producto: Por su abreviatura “BLP”, ejemplo: BLP\_01.
6. Estructura de descomposición del trabajo: Por su estándar “WBS”, ejemplo: WBS\_01.
7. Ejecutables: Por su extensión “EXE”, ejemplo: EXE\_01.
8. Diagrama de clases: Por si abreviatura “DC”, ejemplo: DC\_01
9. Modelo de Entidad Relacion: Por abreviatura “MER”, ejemplo: MER\_01.

### Elementos de gestión de la Configuración

1) Especificación del sistema  
  
2) Plan de proyecto   
  
3) Especificación de requisitos: Cualquier PC con conexión a internet.   
  
b. Prototipo ejecutable: Se realiza la conexión entre servidor y cliente con el desarrollo del logueo en la aplicación.

4) a. Descripción del diseño de datos: La distribución de los datos será por la base de datos con el modelo Entidad-Relación, desarrollado en MySQL(Apache)  
  
b. Descripción del diseño arquitectónico : La arquitectura será el Modelo Cliente-Servidor, realizando el server en PHP brindando servicios que serán consumidos por un cliente en ASP.Net.  
  
c. Descripciones del diseño de interfaces: Los mockups del cliente donde se muestra el diseño explorable e intuitivo de la web.  
  
5) Listados del código fuente  
  
6) a. Plan y procedimiento de pruebas  
  
b. Casos de prueba y resultados registrados  
  
7)La especificación de la base de datos:   
  
a. Esquema y estructura de archivos: Tablas y Relaciones  
  
b. contenido inicial: datos en la bd  
  
8) Los documentos de mantenimiento:  
  
a. Los informes de problemas del software se registraran por medio de los resultados de pruebas al sistema  
  
b. Registro de mantenimiento: Tendrán un registro cuando se haga un mantenimiento al sistema.  
  
c. Órdenes de cambios: Tendrán un registro con las actualizaciones a la aplicación.

|  |  |  |
| --- | --- | --- |
| Elemento de Configuración | Descripción y Características | Etapa en que se incorpora a la gestión de la configuración |
| PETAPP\_WBS\_01 | Estructura de descomposición del trabajo a presentar | Etapa de diseño |
| PETAPP\_BLP\_01 | Backlog del producto | Etapa inicial |
| PETAPP\_EXE\_01 | Ejecutables del proyecto a presentar | Etapa de desarrollo |
| PETAPP\_UI\_\* | Interfaz de usuario a través del cual el usuario interactuara con el sistema | Etapa de desarrollo |
| PETAPP\_SC\_01 | Código fuente del proyecto | Etapa de desarrollo |
| PETAPP\_PGC\_01 | Plan de Gestión de la Configuración | Etapa de diseño |
| PETAPP\_DB\_01 | Base de datos que contiene la información que se ingrese en el sistema. | Etapa de desarrollo |
| PETAPP\_HU\_01 | Las historias de usuario definidas | Etapa de diseño |
| PETAPP\_RACI\_01 | Tabla de los roles y responsabilidades | Etapa inicial |
| PETAPP\_MER\_01 | Modelo entidad relación inicial. | Etapa de diseño |
| PETAPP\_DC\_01 | Diagrama de clases creado a partir del modelo entidad relación | Etapa de diseño |
| PETAPP\_SBL\_01 | Sprint Backlog | Etapa de diseño |

### Roles y Responsabilidades

Archivoadjunto.

### Líneas Base

Para cada uno de nuestros tres spring definidos se coloco un tag donde veriamos hasta que punto fue apropado el spring y que nombre fue asignado

Para el spring 1 se le coloco de nombre Version0.1 en el cual se ve con claridad que en este spring está definido nuestro iniciar sesión con usuario y parte de la interfaz gráfica del inicio de la página web.

En este spring se desarrolló los siguientes ciclos de vida del software

Análisis

Se analizaron los requerimientos establecidos para este primer spring para poder realizar la funcionalidad de este.

Diseño

Se diseñaron las primeras pantallas para los requerimientos principales como el login de usuario, la página de inicio.

Implementación

Se implementó el login para que el usuario pueda acceder al sistema.

Para el spring 2 se le coloco de nombre Version0.2 en el cual se ve con claridad que en este spring se levantaron las observaciones del anterior y se agregó el registro de usuario de manera gráfica y el de registrar mascota y se implementó la funcionalidad para que el usuario pueda cerrar sesión.

Los ciclos de vida del software definidos para este spring son:

Análisis:

Se analizaron los requerimientos establecidos para este segundo spring para poder realizar la funcionalidad de este.

Diseño:

Se diseñaron las primeras pantallas para los requerimientos principales como el registro de usuario, registro de mascota.

Implementación

Se implementó el cerrar sesión para que el usuario pueda finalizar una sesión

Para el spring 3 se le coloco de nombre Version0.3 en el cual se ve con claridad que en este spring se levantaron las observaciones del anterior y se agregó el registro de usuarioy el registro mascota de manera funcional y se agrego de manera gráfica el buscar mascota.

Los ciclos de vida del software definidos para este spring son:

Análisis:

Se analizaron los requerimientos establecidos para este segundo spring para poder realizar la funcionalidad de este.

Diseño:

Se diseñaron las primeras pantallas para los requerimientos principales como el registro de usuario, registro de mascota.

Implementación

Se implementó el registro de usuario y de mascota

Pruebas

Se realizaron pruebas unitarias para cada tipo de acción..

## Control de Cambios

### Proceso de solicitud de cambios y aprobación

Se atenderán los cambios con alta prioridad siempre y cuando estos hayan sido requerimientos que no fueron tomados por algún error por parte del equipo de trabajo. De no darse el caso, se organiza una reunión donde se expondrán las peticiones. La solicitud se evaluará por el equipo de trabajo, seguido a esto se entregará una respuesta de forma escrita citando a una reunión conjunta entre las dos partes, de ser aprobado el cambio se expresa el impacto que tendrá, es decir:

* Atrasos
* Costos
* Tiempo invertido
* Y como afecta el cambio a otros requerimientos.

### Comité de control de Cambios.

Para cada nuevo requerimiento que logre surgir se convoca a una reunión en donde se evalúa el impacto por parte de cada miembro del grupo, se discutirán los puntos de vista y se llegará a un acuerdo. Dicho acuerdo tiene como pilar la aceptación de los objetivos propuestos, dando así un plan de trabajo para elaborar el cambio.

## Contabilidad y Auditoría de la Gestión de la Configuración

### Almacenamiento del proyecto

La persistencia de todos los artefactos y documentos del proyecto se harán en tiempo de desarrollo en un ambiente local. Luego de la finalización de este, los cambios realizados o funcionalidades implementadas serán subidas al repositorio GitHub, el cual se encargará de las actividades de versionamiento y persistencia.

Todos los cambios subidos serán comentados en el repositorio describiendo su naturaleza, para quien va dirigido, si surgió como solución a un error y/o si continúe posibles errores contenidos en el mismo.

### Proceso de generación de releases.

Para el presente proyecto no se tomarán acciones directas para el proceso de generación de releases ya que nuestro proyecto estará trabajando bajo integración continua con Jenkis.

Se tomará el ultimo reléase generado bajo integración continua. Luego este reléase será versionado para su posterior entrega al cliente.

### Reports and Audits

Las auditorias se harán de acuerdo al Cronograma de Desarrollo y con el formato preestablecido para las mismas.

Las auditorias tendrán dos filtros:

* Filtro Principal el cual será evaluado por el Gerente del proyecto el Ing. Hans Soto con quien se sostendrán reuniones semanales como mínimo. Para evaluar los avances se confrontará con el cronograma del proyecto.
* Filtro Secundario será realizado por los integrantes del comité que avalan y aprueban los proyectos, poniendo a discusión la viabilidad del proyecto inicialmente, estado del proyecto, avances y finalmente si aprobaron parcial o completamente.