Desarrollo de videojuego para la enseñanza de medidas de control del Dengue

Plan Gestión de la Configuración

Versión 2.0

Revisión Historia

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Versión** | **Descripción** | **Autor** |
| 03/10/2014 | 1.0 | Especificación finalidad documento | José Omar Colorado Pérez |
| 04/10/2014 | 2.0 | Creación estrategia Gestión configuración | José Omar Colorado Pérez.  Mateo Rincón Estrada.  Daniela Guzmán Bedoya. |
| 06/10/2014 | 3.0 | Ajustes del documento en general, correcciones y aclaración de información. | Daniela Guzmán Bedoya |
|  |  |  |  |
|  |  |  |  |

Tabla de contenidos

1. Introducción 4

1.1 Propósito 4

1.2 Alcance 4

1.3 Definiciones, Acrónimos y Abreviaciones 4

1.4 Referencias 4

1.5 Visión General 4

2. Gestión de configuración del Software 4

2.1 Organización, responsabilidades e interfaces 4

2.2 Herramientas, entorna o e infraestructura 4

3. Programa para la gestión de la configuración 6

3.1 Identificación de configuraciones 6

3.1.1 Métodos de identificación 6

3.1.2 Líneas base del Proyecto 6

3.2 Configuración y control de cambios 7

3.2.1 Proceso de aceptación de cambios y aprobación. 7

3.2.2 Comité de control de Cambios (CCB) 7

3.3 Configuración de contabilidad de estado 7

3.3.1 Proyecto de Medios de Almacenamiento y Proceso de estreno 8

3.3.2 Reportes y auditorias 8

4. Hitos 9

5. Entrenamiento y recursos 10

Plan de gestión de la configuración

# Introducción

El presente documento mostrara el plan de gestión de la configuración adoptado por el grupo de trabajo tanto para el manejo en el desarrollo, la creación de versiones como para la corrección de errores y seguimiento de cambios. Así como lineamientos para la gestión de los archivos contenidos dentro del proyecto, específicamente en la herramienta SCM.

## Propósito

Se tratara de definir los lineamientos para gestionar los elementos involucrados en el proyecto, además de controlar las versiones de prueba que se irán construyendo para dar soporte a los elementos funcionales que el cliente aceptará.

## Alcance

Podrá verse afectado la especificación de algún caso de uso, si se aceptase un cambio según los criterios que se exponen en este documento, asimismo la gestión de la herramienta GitHub será administrada bajo estos estándares.

## Definiciones, Acrónimos y Abreviaciones

Unity: Es un motor de desarrollo de videojuegos.

Sprite: Imágenes usualmente en formato .png que se emplean para la creación de personajes 2D en motores de desarrollo como Unity.

Kanban: Técnica empleada para la asignación de tareas en todas las etapas de un proyecto.

SCM: Gestor de manejo de código, en el caso nuestro gestor de las configuración de archivos creados por los integrantes del grupo de trabajo.

CM: Configuration Managed (Gestión de la configuración).

## Referencias

Se hace referencia de los sitios Web de las herramientas empleadas para la gestión de la configuración, así como del documento por el cual se crearon los lineamientos presentes en este documento.

Sitio Web herramienta SCM (<https://github.com/>)

Sitio Web gestor de cambios (<https://www.mantisbt.org/>)

CMMI-DEV, V 1.3. (2010). Gestión de configuración. Software Engineering Institute, 243-257.

## Visión General

El documento se encuentra distribuido en tres grande ítems, primero, punto 2 del documento, Gestión de configuración del software describiremos los elementos a integrar en la herramienta SCM, el seguimiento del trabajo y se describirá las herramientas a emplear para hacer la gestión de configuración por parte del gerente, por otra parte se hace una visión del producto final que se espera obtener.

En el ítem 3 del presente documento se describen las estrategias para el manejo de archivos resultantes, la corrección de errores, la solicitud y seguimiento de cambios dentro del proyecto. Se trata de dar un visión general de los reportes de avance y auditoria del proyecto.

En el ítem número 4, se identifican los hitos en los cuales la CM podría sufrir cambios dependiendo de las necesidades del cliente o grupo de trabajo. Finalmente el ítem 5 contiene un cuadro con las habilidades con las que debería contar el equipo desarrollador para hacer frente a este plan de configuración empleando las herramientas descritas.

# Gestión de configuración del Software

En esta etapa se gestionara los entornos de desarrollo que serán implementados para llevar a cabo el desarrollo del proyecto, se hace una visión del producto final que se espera obtener, teniendo así un alto índice de efectividad.

## Organización, responsabilidades e interfaces

Se asignara a un miembro del equipo de trabajo para que lleve el control de las diversas actividades que se deben realizar para la gestión de la configuración a lo largo del desarrollo del software, esto debido al tamaño reducido de los repositorios a generar, lo cual se hace manejable para todos los integrantes del proyecto.

## Herramientas, entorna o e infraestructura

GitHub es una plataforma SCM que nos permite alojar repositorios de proyectos colaborativos, permitiendo acceder a un único proyecto desde diferentes ambientes de trabajo, lo que facilita el control de versiones y la gestión de cambios dentro del desarrollo. Otra de las funcionalidades de GitHub es poseer un visor de código mediante el cual se puede consultar el contenido de un fichero.

Dentro de las diversas características presentes en la página oficial de la herramienta, podemos encontrar características que pueden ser de gran utilidad como lo son la Wiki, para la documentación referente al repositorio. El sistema de seguimiento de problemas, contribuirá a disminuir los tiempos de soluciones de los problemas dentro del equipo de desarrollo. Algo llamado por los creadores del GitHub como el visualizador de ramas nos brindara la oportunidad de conocer el progreso en los diferentes casos de uso asignados dentro del proyecto.

Mantis Bug Tracker

Citaremos la definición de esta herramienta presente en su sitio Web

“MantisBT es un gestor de incidencias de código abierto que ofrece un delicado equilibrio entre simplicidad y potencia. Los usuarios pueden empezar en cuestión de minutos y empezar a gestionar sus proyectos al tiempo que colaboran con sus compañeros de equipo y los clientes con eficacia.”

A continuación se mostrara detalles de desarrollo tales como:

Tamaño final del producto

Luego de hacer un sondeo por la red del tamaño promedio de las aplicaciones desarrolladas en Unity, que comparten características similares con el proyecto a desarrollar se logra identificar un peso final tentativo de 32 MB en la aplicación móvil para Android y de 15 MB como ejecutable para Windows. Además se practicaron pruebas en el entorno de desarrollo con las cuales se permitiera realizar un cálculo del tamaño final mucho más acertado que al encontrado en la red. Realizando la revisión de los escenarios del juego se estima que se hará uso de 45 Sprites, mejor conocidas como imágenes, estas en formatos .png o .jpg, las cuales no superan los 46 Kb cada una.

Para creación de Scripts se asigna 2 Mb en total, dado lo liviano de estos llegando a ser los archivos que menos espacio ocupan dentro del juego. En el apartado de Audio se emplearan formatos de compresión que disminuyan el uso de disco, buscando garantizar tamaños no superiores a los 400 Kb por cada uno. Finalmente el integrador de todos los elementos de juego, la base del motor Unity está ocupando entre 10 y 15Mb según especificaciones de la página oficial e información obtenida de los foros, para aplicaciones en 2D como es nuestro caso. Al sumar estos valores podemos fijar un tamaño entre 21 Mb y 25Mb en la aplicación final.

Distribución del trabajo en equipo

Se realiza una distribución igual para los cuatro integrantes del equipo de trabajo, de tal manera que a cada uno le correspondió dos de los ocho casos de uso especificados con anterioridad, la forma en que se distribuyó el trabajo a realizar quedo plasmados en el Kanban y las fechas serán estipuladas en el cronograma de trabajo. (Para más información de los casos de uso, remítase a la especificación de casos de uso).

Localización de servidor y equipos de desarrollo

El servidor donde se encuentra alojado el proyecto es brindado por GitHub y es controlado en su totalidad por todos los integrantes del proyecto. Los equipos en donde se realiza la implementación del software deberán estar conectados mediante la aplicación de escritorio que nos brinda la herramienta SCM. Generalmente los equipos sobre los cuales se trabaja están ubicados en la universidad del Quindío, específicamente en la sala de sistema o en las computadoras de los 4 miembros del equipo de desarrollo.

# Programa para la gestión de la configuración

Se describen las estrategias para el manejo de archivos resultantes, la corrección de errores, la solicitud y seguimiento de cambios dentro del proyecto. Se trata de dar un visión general de los reportes de avance y auditoria del proyecto.

## Identificación de configuraciones

Se configurara la forma en que se deben presentar el nombre de los cambios ejecutados en algún archivo, con el fin de llevar un orden, también se aclararan las líneas bases del proyecto.

### Métodos de identificación

La identificación de todos los artefactos que se generen durante el desarrollo, ser hará de la siguiente manera:

1. Siglas: Todo documento, componente, prueba, dato. Etcétera, tendrá al inicio las iniciales del tipo de artefacto. Ejemplo: Caso de uso Salir CU
2. Número secuencial: Número que precederá a las iniciales del tipo de artefacto Servirá para diferenciar entre los diversos elementos de una categoría, así como para saber del orden de creación entre las categorías.
3. Fecha de creación: Como último elemento se plasmara la fecha en la cual el artefacto fue creado, esto a fin de tener otro elemento para la verificación del cronograma de trabajo. La fecha ira luego del número secuencial y estará en el siguiente formato “-DD/MM/AA”.

### Líneas bases del Proyecto

Se definen 4 líneas bases del proyecto en las cuales se busca realizar revisiones del trabajo que se ha desarrollado y del impacto sobre el cliente. En las primeras tres está abierta la posibilidad de cambios sujeta a restricciones que expondremos en cada una de las líneas, por el contrario en la última línea se cierra la posibilidad de cambios, siendo esta la etapa de entrega final.

1. Elaboración

Aquí se acepta casi todos los cambios que el cliente desee, debido a que es la fase de identificación de las necesidades, y lo que se busca es el mayor grado de satisfacción del cliente entre lo que desea y lo que realmente necesita. Se crean prototipos y bosquejos de escenarios, los cuales deberán ser aprobados por el usuario.

1. Construcción

Luego de tener las bases de trabajo, como lo son el prototipado y la identificación de los elementos finales del sistema (Casos de uso), se procede a iniciar la implementación de estas ideas empleando las herramientas elegidas según los criterios de necesidades definidos. Los cambios o inserción de nuevos elementos en el videojuego están sujetas a aprobación del equipo desarrollador, y afectaran el cronograma de trabajo en caso de ser aprobadas, al igual que el precio inicial estimado.

1. Pruebas

Al cumplir el cronograma de desarrollo se contara con un producto final, que se evaluara según los criterios definidos en el documento de Calidad en uso. Solo se aceptaran cambios en casos de que las funcionalidades no cumplan con lo indicado en la documentación, esto estará soportado por las pruebas que se realicen, pero una mala calificación no será causal de posibles cambios, esto debido a que la usabilidad de los juegos no está relacionada con lo especificado en las funcionalidades. La integración de nuevas funcionalidades podría conformar el inicio de un nuevo proyecto.

1. Entrega final

Al evaluar el producto, tal vez se apliquen algunos cambios poco significativos, con el fin de no retrasar la fecha de entrega final y la capacitación del personal que empleará el videojuego con fines académicos. Cambios e integración de funcionalidades serán cuestiones de un posible proyecto.

## Configuración y control de cambios

En este ítem se definirá, aclarará y establecerá el proceso de aceptación de cambios y aprobación, también se definirá el comité de control de cambios.

### Proceso de aceptación de cambios y aprobación.

En el momento en que se presente la solicitud de un cambio, debe quedar constancia de quien lo solicita, la fecha en que lo realiza y el responsable dentro del equipo de trabajo, el cual recibió la solicitud, también se debe informar al cliente de la fecha en que se dará respuesta a la solicitud y plasmar esto en la plantilla de solicitud de cambios, según las fechas de revisiones de cambios, estas se harán cada 10 días.

Luego de aprobar o no aceptar un cambio solicitado con el cliente, se plasma en la plantilla correspondiente y se le informa a este del resultado, y se solicita un acuse de recibido. Todo esto se hará teniendo como elemento principal la herramienta GitHub; Además de la herramienta Mantis Bug Tracker para la documentación y manejo de cambios, llevando así un control sobre dichos cambios, pero sin dejar de lado la documentación requerida por la organización. De este modo se busca dar soporte a las prácticas de CM PA de CMMI.

Para mayor información de la herramienta SCM, y del gestor de cambios, errores Mantis Bug Tracker consulte la sección de referencia.

### Comité de control de Cambios (CCB)

El CCB está compuesto por el gerente del proyecto y los 4 desarrolladores, y tiene como finalidad determinar la viabilidad de los cambios solicitados. Este análisis será basado en la experiencia y el conocimiento adquirido en el desarrollo del proyecto actual, con esto se podrá determinar el alcance dentro del sistema de los cambios que sean aceptados o la identificación de solicitudes que consideren que se salen de la capacidad de trabajo del equipo. En cualquiera de los casos se debe analizar el día elegido para la gestión de cambios, que subsistemas se verán afectado y en que magnitud, estos resultados serán plasmados en la plantilla correspondiente al cambio solicitado.

## Configuración de contabilidad de estado

Se usara la plantilla para la descripción de medios de almacenamiento dispuesta en RUP, para describir los medios de almacenamiento

### Proyecto de Medios de Almacenamiento y Proceso de estreno

|  |  |  |
| --- | --- | --- |
| Rol | Tarea de Almacenamiento | Descripción |
| Integrador | Copia en dispositivos externos | Copia de entorno de trabajo en memorias usb, cd. |
| Cualquier Rol | Actualización del espacio de trabajo | Cualquier cambio realizado por un integrante del equipo de desarrollo, el cual se almacena en todas las localizaciones del proyecto. |
| Administrador de configuraciones | Copias de seguridad | Backup del SCM, el cual se almacena en Dropbox, y se categorizan según la fecha de este. |

Lanzamientos

Los posibles Releases que se presenten del videojuego estarán sujetos a revisiones y únicamente se liberará la documentación para el uso final del cliente. Pueden presentarse lanzamientos únicamente para uso del equipo de desarrollo, en pro de la revisión de los escenarios e identificación de errores de primera instancia.

Se calcula que los lanzamientos para el cliente serán 5, uno durante la fase de prototipado, el segundo y tercero corresponderán a las entregas de los niveles definidos en la WBS, la cuarta contendrá el producto a probar con usuario y una guía básica de uso, para aplicar la prueba. La entrega final se hará de manera física y con copias virtuales a los correos del cliente y patrocinador.

### Reportes y auditorias

Documentos que serán presentados al cliente con el fin de brindarle un soporte al software.

* Reportes de integración

Con ese artefacto se pretende recopilar la adaptación de CM en el proyecto, teniendo en cuenta los estándares definidos en este documento para el manejo de la herramienta SCM. Estos reportes contendrán estadísticas, graficas en función del grado de adaptabilidad presente en el proyecto, esto es identificado por el gerente de configuraciones, el cual informara de los resultados al grupo de trabajo.

* Reportes de solución

Se podrán generar reportes de la cantidad de errores reportados por el equipo de trabajo, y el grado de solución de estos en la herramienta GitHub. Estos reportes serán artefactos que den soporte a la calidad del software, de igual modo proveerán una herramienta para conocer la forma de solucionar errores similares a futuro o en otros proyectos. De igual modo estos reportes podrán de efectividad de solución de errores, implementación de nuevas funcionalidades o gestión de cambios. Cualquier consulta de este tipo que se haga sobre el proyecto tendrá un formato de presentación elegido por el integrador o gerente de configuraciones, esto dependiendo de los resultados que necesite compartir.

También podrá hacer un reporte de un elemento de gran importancia, como la inclusión de un nuevo módulo y deberá mostrar el avance de implementación de este en el sistema, con respecto a otros módulos que ya están desarrollados en su totalidad o parte de ella, y servirán como línea base.

# Hitos

Se trabajar sobre hitos que intervengan en la CM, los cuales nos afectan de manera interna e igualmente sean de gran importancia para el usuario.

* Aprobación documentos planeación

Dentro de los puntos clave del desarrollo del proyecto se considera que la aprobación de los documento de planeación, influyen sustancialmente en la CM, dado que se genera gran cantidad de elementos de configuración que deben ser almacenados en la herramienta SCM, teniendo en cuenta la posible variabilidad de estos, así como la característica de ser la base del desarrollo de prototipos iniciales de presentación para el cliente.

* Aceptación prototipo de desarrollo

En el momento en que el cliente acepta el prototipo de desarrollo sobre el cual vamos a trabajar, estos archivos que comprenden el prototipo pasan a ser la línea base sobre la cual se desarrollara, pueden que se presenten cambios en un futuro de los archivos resultantes, pero estos deben ser analizados de forma tal que no afecte el prototipo inicial que el cliente aprobó.

Aceptación producto final

La CM en este punto del proyecto pasa a ser de vital importancia porque nos ayuda a gestionar el producto final que el cliente aprobara, en esta etapa puede que se presenten cambios significativos en la forma en que se maneja la CM, debido a los cambios finales que el cliente quisiera aplicar al sistema y a los cuales deberíamos adaptarnos en caso de aceptarlos.

Desarrollo de pruebas de Usabilidad

Parte de los errores identificados en las pruebas de usabilidad serán corregidos en esta etapa, y será responsabilidad del administrador de configuraciones disponer e informar a los desarrolladores de los posibles cambios que se podrían agregar.

# Entrenamiento y recursos

|  |  |
| --- | --- |
| **Entrenamiento** | **Rol/Recurso** |
| Desarrollo de software empleando GitHub | Cualquiera involucrado |
| Manejo de versiones | Administrador de configuraciones |
| Integración de cambios | Integrador |
| Conocimientos básico Mantis Bug Tracker | Cualquier involucrado |
| Solicitud de cambios | Desarrollador mediante informe a Arquitecto |

GitHub para Windows y equipos donde se clonen versiones del proyecto, las cuales estarán sincronizadas mediante el uso de la herramienta.