Universidad Nacional de Córdoba

Facultad de Ciencias Exactas Físicas y Naturales

Los piratas

Ingeniería de Software

PacMan

Plan de Gestión de Configuraciones

Autores:

- Arnolfo Emanuel
- Landaeta Ezequiel
- Montero Felipe
- Olocco Laureano

Versión del documento: 1.4

Historial de versiones

Versión	Fecha	Cambios
1.1	22 de abril	Primera versión
1.2	29 de abril	Se agregan sección 2 y 3
1.3	30 de abril	Se agrega sección 4 y 5
1.4	1 de mayo	Se revisa y corrige todo el documenta con todo el equipo de trabajo
1.5	1 de junio	Cambio en esquema de ramas y directorios

Página de aprobación

A continuación, se enumeran las personas a las que se les solicitará su aprobación para cada cambio importante de este plan.
*Las aprobaciones no son necesarias en caso de que el cambio sea menor.

Líder del CCB	Fecha	Firma
Arnolfo Emanuel		

Gerente de pruebas	Fecha	Firma
Olocco Laureano		

Gerente de ingeniería o de release	Fecha	Firma
Landaeta Ezequiel		

Gestor global de configuración	Fecha	Firma
Montero Felipe		

1. Introducción

1.1 Propósito y alcance

En el siguiente documento se detalla el Plan de Gestión de Configuración para el proyecto de software a realizar. El objetivo de este documento es detallar la configuración de los requisitos, documentos, software y herramientas utilizadas en este proyecto.

Se define como procesar los cambios propuestos al sistema, se decide qué componentes del sistema modificar, cómo gestionar las distintas versiones y cómo distribuir las distintas versiones para los distintos clientes.

1.2 Propósito del SCM

- Garantizar la coherencia en la práctica de actividades de SCM.
- Mantener la integridad del producto durante todo el ciclo de vida.
- Gestionar las versiones de los componentes que hacen a la aplicación.
- Informar a los grupos e individuos afectados sobre el estado del proyecto.

1.3 Herramientas de gestión de configuración

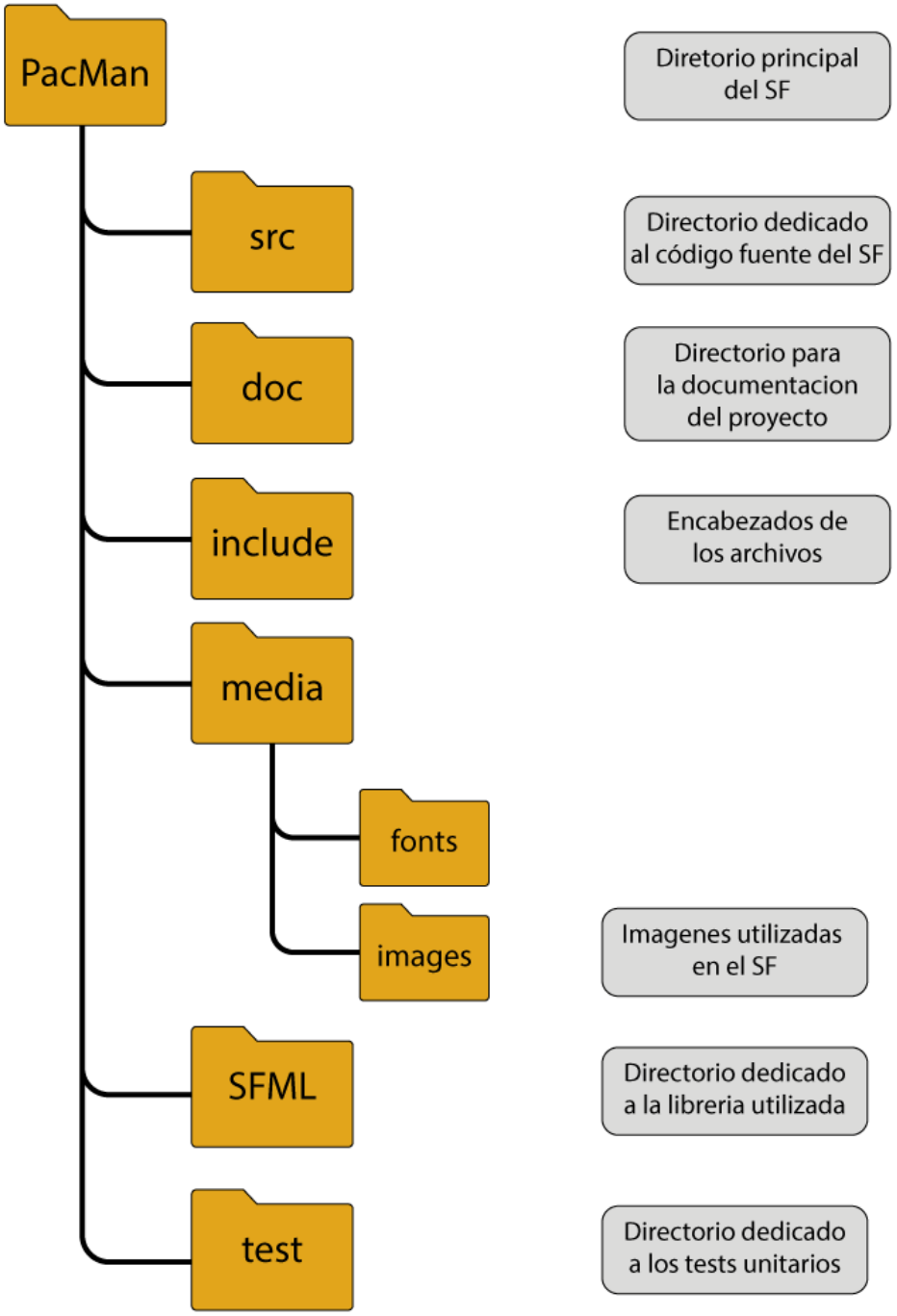
Herramienta	Propósito	Dirección
GitHub	Repositorio del código y gestión de versiones	https://github.com/TheRabbitt/Ingenieria-de-Software---Los-Piratas
Circle-CI	Control de Integración continua	https://app.circleci.com/pipelines/github/TheRabbitt
GitHub Issues	Seguimiento de mejoras y bugs	https://github.com/TheRabbitt/Ingenieria-de-Software---Los-Piratas/issues

1.4 Glosario

- CCB: comité de Control de Cambios.
- CMP: Plan de gestión de las configuraciones.
- CM: Gestión de Configuraciones .
- SCM: Administración de la Configuración de Software.
- Tag: Etiqueta.
- CI: Ítem de Configuración.
- UML: Lenguaje Unificado de Modelado.
- Developer: Encargado de la programación del software.
- Máster: Vigila el cumplimiento de la metodología . Define prioridades, roles y proyectos.
- Téster: Es el encargado de crear los test unitarios y de sistema.

2. Esquema de Directorios y Propósitos

2.1 Estructura de directorios y propósitos de cada uno



2.2 Normas de etiquetado y nombramiento de archivos

Para el nombramiento de las entregas del proyecto PacMan, utilizaremos un patrón de tres dígitos separados por puntos **‘.’** . Con un prefijo ‘v’ haciendo referencia a la palabra “versión”, por ejemplo: PacMan v1.0.0. Los dígitos representan lo siguiente:

- Primer dígito: Representa las entregas que significan un nuevo paquete con muchos cambios y funciones nuevas o diferentes a la anterior entrega.
- Segundo dígito: Representa cambios o adición de funcionalidades.
- Tercer dígito: Representa correcciones de bugs o errores encontrados.

Todos estos comenzarán con el 0, excepto el primero que comenzará en 1, e incrementará a medida que se avanza en el desarrollo del trabajo.

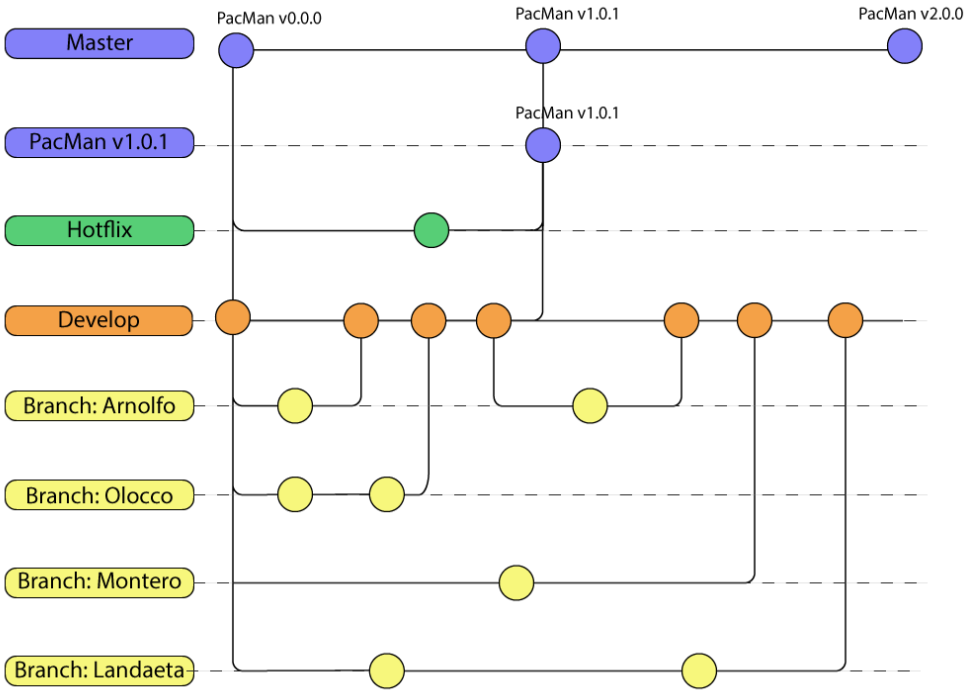
Para los documentos adicionales del proyecto, como lo es este mismo, se usará un patrón de dos dígitos separados por puntos **‘.’**. Los dígitos representan lo siguiente:

- Primer dígito: Representa las entregas de nuevos documentos, asociados a las entregas oficiales del proyecto.
- Segundo dígito: Representa correcciones en el documento.

3. Gestión de configuración del código

3.1 Esquema de ramas

Para el esquema de ramas a usar en nuestro repositorio hemos decidido que los tres desarrolladores tendrán cada uno su propio branch donde realizarán el avance personal y luego harán los merge con la línea de desarrollo “Develop”, luego cuando se acumule lo necesario para hacer un release se sube a la rama “Master” como una nueva versión. También asignamos una rama especial llamada “Hotflix” para corregir posibles bugs o errores que pueden surgir de las versiones que estén en la “Master”.



3.2 Política de fusión de archivos

Cada uno de los desarrolladores deberá en primer lugar verificar que tengan en su repositorio local la última versión de la rama principal (Master). Una vez chequeado esto deberemos ver si nuestro código pasa todos los casos de prueba (test). Asegurada la correcta funcionalidad, recién ahí se podrá hacer “merge” (o combinación) con la rama principal, aplicando el etiquetado de acuerdo a las normas establecidas previamente.

4.Forma de entrega de los releases

Los releases se obtendrán a través de la descarga del repositorio oficial ubicado en GitHub, cuyo enlace se detalló al inicio del documento. Para poder instalar el paquete de software será necesario desinstalar cualquier rastro de versiones anteriores, lo cual será posible mediante una herramienta de desinstalación que se consigue junto con cualquier versión que se descargue.

El formato del paquete de software será un archivo comprimido .zip cuyo contenido será una carpeta de instalación con los siguientes archivos:

- PacMan.exe: Instalador del Programa de Agenda Semanal.
- PacManUninstaller.exe: Desinstalador de versiones previas.
- Readme.txt: Instrucciones de instalación y configuración apropiadas, página web del repositorio y del foro de bugtracker.
- Changelog.txt: Un registro de cambios de las correspondientes versiones con documentación de los errores conocidos hasta la versión actual.

5. Change Control Board

5.1 Introducción y objetivos

El Change Control Board (Comité de Control de Cambios, en español) es el grupo encargado de evaluar, aprobar o rechazar los pedidos de cambios realizados por diversos agentes internos o externos (clientes).

El comité busca que todo cambio sea considerado por todas las partes y lograr su total autorización antes de su implementación en los planes, documentos y el código de la aplicación.

Se monitorea y controla las peticiones de cambio para establecer luego las bases de los requisitos de configuración.

El CCB debe cumplir con reuniones de trabajo de forma periódica en dónde se definen las políticas de trabajo.

Rol	Responsabilidad	Responsable
Líder del CCB	• Aprobar o modificar cambios • Maneja el CCB • Dar la aprobación final al cambio a realizar y documentarlo. • Organizar reuniones	Montero Felipe
Gerente de ingeniería o de release	• Evaluar los costos de los cambios que se le hagan al sistema y las fechas de entrega.	Landaeta Ezequiel
Gestor global de configuración	• Realizar tareas de planeamiento, diseñar políticas de control de seguimiento y asegurarse que se cumplan	Arnolfo Emanuel
Gerente de pruebas	• Organizar las pruebas • Evaluar los impactos del cambio	Olocco Laureano
Gerente de coordinación	• Documentar todos los cambios realizados	Olocco Laureano

Las reuniones serán semanales, se celebrarán los días viernes (se definirá vía mail semana a semana) a las 16:00 hs mediante videoconferencias por internet para revisar las solicitudes de cambio y definir el cronograma de acciones a realizar en la semana entrante. En las reuniones se evaluará si vale la pena o no el cambio según el costo y la nueva funcionalidad, y en función de la decisión se cierra el pedido de cambio o se le da el visto bueno a los desarrolladores para que comiencen a trabajar.

5.2 Proceso de control de cambios

El proceso se inicia cuando se completa y envía una petición de cambio en la que se describe el cambio requerido al sistema. Esto puede enviarse en un formato de petición de cambio CRF (change request form). Aquí se usa el término cliente para incluir cualquier participante que no sea parte del desarrollo.

Mientras se procesa la petición de cambio, se agrega información al CRF para registrar las decisiones tomadas en cada etapa del proceso, las recomendaciones que conciernen al cambio, los costos del cambio y las fechas de cuando se solicitó, aprobó/desaprobó, implementar y validar el cambio. También se puede incluir una sección que enfatice cómo puede realizarse el cambio.