

Manejo del Ambiente Controlado Testify

OSLO

Ojeda Valeria – Sly Eduardo

Levipichun Emilio – Oyarzo Malena



En este documento se debe establecer cómo será implementado el Ambiente Controlado, por ejemplo con software existente, o si se realizará de otra forma como a través de una página Web del grupo, backups, etc

Tabla de contenido

Definición del Ambiente Controlado	4
Descripción del uso del Ambiente Controlado	4
Implementación de Cambios	5
<i>Estructura General del Ambiente Controlado</i>	<i>6</i>
<i>Ubicación física del Servidor.....</i>	<i>6</i>
<i>Utilización del Ambiente Controlado.....</i>	<i>6</i>
Responsabilidades	7

Manejo del Ambiente Controlado

Definición del Ambiente Controlado

El ambiente controlado descrito en el presente documento será implementado utilizando Git junto a GitHub Desktop, donde se versionarán y almacenarán los artefactos y el código del proyecto. Adicionalmente se realizarán Backups periódicos de la información en los ordenadores locales y en el servidor local detallados en el “plan gestión de configuración”.

Descripción del uso del Ambiente Controlado

Cada miembro del equipo deberá realizar commits frecuentes en las ramas asignadas, y se utilizará la rama 'Main' para las fusiones definitivas una vez que los cambios hayan sido aprobados por el Líder del proyecto y el Administrador de la configuración. La creación de nuevas ramas para funcionalidades específicas será obligatoria, y antes de fusionar con la rama 'Main', se requerirá aprobación mediante un pull request.

Los cambios deberán ser registrados mediante la herramienta de control de versiones Github, estos cambios pueden ser empleados por cualquier integrante del grupo de desarrollo OSLO que incluyen:

Nomenclatura:

Los cambios son realizados mediante:

1. “commit”: Confirmación de un cambio en el repositorio local que tendrá por título el nombre de la rama involucrada y un título referente al cambio, seguido por una descripción detallada y el “push” al repositorio origen.

Ejemplo de commit:

- **Título:** “Documentación – Actualización de plantillas base”.
- **Descripción:** “Se añade la plantilla base para el estándar de documentación OSLO”.

Integrantes:

- Líder del proyecto: Ojeda Valera.
- Administrador de configuración: Levipichun Emilio.
- Analista: Ojeda Valera, Oyarzo Malena.

- Administrador SQA: Levipichun Emilio.
- Documentador: Ojeda Valera, Oyarzo Malena.
- Arquitecto: Sly Eduardo, Levipichun Emilio.
- Cliente: Cátedra de laboratorio de desarrollo.
- Programador: Sly Eduardo, Levipichun Emilio.

Implementación de Cambios

Este apartado detalla el procedimiento para llevar a cabo el cambio una vez aprobado:

Una vez realizada una entrega importante o aprobación por parte del cliente, fin de iteración o fin de fase se procederá a realizar los cambios mediante un “Merge” de las ramas operativas de desarrollo a la rama principal de producción “Main”, asegurando que se sigan las pautas del control de versiones. Las modificaciones serán documentadas y registradas en GitHub/Git mediante el uso de pull requests y merges para aprobar cambios, para ello se designará como encargado al Administrador de la configuración para la fusión de ramas y al líder del proyecto para la evaluación y posibles conflictos entre ramas.

“Merge”: Confirmación de la unión entre 2 ramas/branches para traer cambios de las líneas bases de desarrollo a la línea base principal de producción “Main”.

Ejemplo de Merge:

- **Título:** “Merge **From** Documentación **To** Main”.
- **Descripción:** “Se realiza fusión de la rama documentación al Main para integrar los documentos/entregables definitivos y que superaron una iteración, fase o entrega crítica obligatoria”.
- **Conflictos:** Para ramas en conflictos se deberá solicitar al líder del proyecto que verifique y controle los cambios a aceptar para aprobar la solicitud de pull requests, mediante la designación de un “Reviewer” y un “Assignees” para la revisión, control y aprobación de cambios en conjunto.

Estructura General del Ambiente Controlado

El proyecto contará con un repositorio en GitHub que gestionará las ramas y los repositorios referente a las líneas bases específicas para el control de versiones y almacenamiento de los artefactos del proyecto.

Ramas:

1. **Documentación:** almacena los documentos entregables y los artefactos relacionados con el desarrollo de software.
2. **Desarrollo:** utilizada para la codificación y el desarrollo activo del software.
3. **Main:** Rama principal y línea base oficial del proyecto que funciona como entorno de producción, donde se fusionan los cambios revisados y aprobados de las ramas Documentación y Desarrollo.

Repositorios

Cada rama definida contiene exactamente la misma estructura de repositorio para evitar conflictos entre ramas:

1. **Fases_de_desarrollo:** repositorio que organiza el trabajo de acuerdo con las fases del modelo RUP: inicio, elaboración, construcción y transición. Cada fase contiene los artefactos correspondientes a su etapa del desarrollo.
2. **Logo:** Contiene las imágenes relacionadas con el equipo de desarrollo, como logotipos o gráficos de presentación.
3. **Templates:** Almacena planillas propias con formato estándar establecido por el equipo OSLO, así como plantillas externas de referencia PSI.

Ubicación física del Servidor

La ubicación del servidor fijo a confirmar.

El control de versiones y almacenamiento se gestionará a través de GitHub, asegurando la disponibilidad y seguridad de los datos en un entorno en la nube.

Utilización del Ambiente Controlado

El acceso al repositorio está configurado como privado y restringido únicamente a los miembros del equipo de desarrollo y supervisores. Cada miembro tendrá permisos específicos de acuerdo a su rol, siendo los programadores responsables del código, el Administrador de SQA responsable de la revisión de documentos y el Líder del proyecto el encargado de controlar y aprobar los cambios durante la fusión de ramas en conflictos.

Responsabilidades

Cada miembro del equipo es responsable de la correcta gestión de sus versiones. El responsable de la administración de la configuración será el encargado de revisar periódicamente el estado del repositorio, y en caso de detectar inconsistencias o errores en el manejo de versiones, deberá notificar al Líder del proyecto, al Administrador SQA para registrar el/los incidentes y a los miembros correspondientes para su corrección inmediata.

Las siguientes responsabilidades llevadas a cabo por sus correspondientes responsables son definidas para ejecutar el proceso de gestión de configuraciones:

- Diseñador y codificador: Eduardo Sly – Levipichun Emilio.
 1. Gestionar versiones del código y documentación.
 2. Crear y revisar pull requests en GitHub relacionados a la codificación.
 3. Aplicar cambios siguiendo los procedimientos definidos.
 4. Crear espacios de trabajo online con disponibilidad permanente (Codespaces).
- Líder del proyecto: Valeria Ojeda.
 1. Aprobar cambios críticos relacionados a conflictos entre versiones y fusiones de múltiples ramas.
 2. Supervisión del presente plan para la gestión de configuraciones.
- Administrador SQA - Administrador de Configuración: Levipichun Emilio.
 1. Realizar revisiones periódicas/rutinarias del sistema de configuración.
 2. Validar la trazabilidad y consistencia de cambios.
- Analista - Documentador: Malena Oyarzo – Valeria Ojeda.
 1. Responsable de organizar, mantener y garantizar la información generada durante todo el proceso de desarrollo.
 2. Debe validar y garantizar el almacenamiento adecuado para una fácil recuperación y actualización.