

Manual de procedimiento

How2 - Conectar con el aprendizaje

Introducción

El Campus Virtual How2, se establece con el objetivo de proporcionar una plataforma educativa innovadora. Este manual tiene como objetivo proporcionar una guía detallada para los empleados involucrados en el proyecto de desarrollo del campus virtual. Describe los procedimientos para las actividades de desarrollo, testing, corrección de bugs y soporte, utilizando GitHub como repositorio y control de versiones. Se aplica a todos los empleados, incluyendo analistas de sistemas y programadores, testers y técnicos, involucrados en el desarrollo de How2.

1. Desarrollo

1.1 Análisis de Requerimientos:

Ante una posible actualización de una característica, así como su adición, los analistas de sistemas revisarán los requisitos y elaborarán un documento detallando la propuesta de los cambios o adiciones, y los respectivos diagramas detallando los tiempos para cada tarea, así como el funcionamiento.

1.2 Creación de una Nueva Rama en GitHub:

El programador encargado creará una nueva rama en el repositorio de GitHub para la nueva característica o actualización

1.3 Desarrollo de la Característica:

El programador desarrollará la nueva característica en la rama creada. Al momento de desarrollar, el programador se encargará de que la codificación sea ordenada y precisa.

1.4 Push y Pull Requests:

El programador realizará commits locales y realizará un push a la rama remota. Se creará un Pull Request para la revisión.

1.5 Revisión de Código:

Otro miembro del equipo revisará el código y proporcionará comentarios.

1.6 Integración Continua:

Se realizará la integración continua para garantizar la estabilidad del código.

1.7 Aprobación del Pull Request:

Una vez aprobado, se procederá a fusionar la rama de la actualización/adición con la rama principal.

2. Testing

2.1 Pruebas Unitarias y de Integración

2.1.1 Ejecución de Pruebas:

Los testers ejecutarán pruebas unitarias y de integración en el entorno de desarrollo con el propósito de hallar bugs o mejorar el funcionamiento del sistema.

2.1.2 Informe de Errores:

Cualquier error identificado se informará mediante la apertura de un Issue en GitHub.

2.1.3 Corrección de Errores:

El programador responsable corregirá los errores y creará una nueva versión de la rama.

2.2 Pruebas de Aceptación

2.2.1 Ejecución de Pruebas:

Los testers realizarán pruebas de aceptación en un entorno de prueba separado.

2.2.2 Informe de Errores:

Cualquier error identificado se informará mediante la apertura de un Issue en GitHub.

3. Corrección de Bugs

3.1. Creación de Rama de Corrección:

Cuando se identifica un bug, el programador creará una nueva rama en GitHub para solucionarlo.

3.2. Corrección del Bug:

El programador corregirá el bug y creará un Pull Request para revisión.

3.3. Revisión y Aprobación:

Otro miembro del equipo revisará y aprobará la corrección. Luego se fusionará la rama correctiva con la rama principal.

4. Soporte

4.1. Registro de Solicitudes:

Los técnicos registrarán las solicitudes de soporte en GitHub Issues, a modo de “tickets” y se establecerá su prioridad.

4.2. Asignación y Resolución:

Las solicitudes serán asignadas a los miembros del equipo y se resolverán según la prioridad.

5. Changelog

5.1 Publicación de actualizaciones

Cabe resaltar que cada cambio, sea actualización del sistema, corrección de bugs, solicitudes, adición de funcionalidades, o cualquier otro cambio, estos cambios deberán estar acompañados de un changelog, para tener un registro de cada adición, actualización, o eliminación, y una descripción de dichos cambios, que deberá aclarar la versión actual y la fecha de subida.