



Actividad 4 - Utilizando sistemas de control de versiones

Dayanna Patricia Coral Martinez

dcoralma@estudiante.ibero.edu.co

IdBanner: 100138687

Ingeniería de Software, Corporación Universitaria Iberoamericana

28102024_C2_202434: Mantenimiento de Software

Profesora: Carmen Emilia Rubio Vanegas

30 de Noviembre del 2024

Introducción

El proceso de mantenimiento de software es esencial en la vida del sistema desarrollado, pues permite que sus características y cualidades se conserven, estén actualizadas y se adapten a las necesidades cambiantes del usuario. En el presente trabajo se busca desarrollar una actividad de aprendizaje práctica y aplicada al mantenimiento de software con la presencia de una de las tareas importantes, revisión de versiones del software.

La versión del software es una de las actividades principales dentro del mantenimiento del software, y para llevar a cabo la práctica del software se utilizarán herramientas de código abierto, Open Source, como Git y GitHub para que los estudiantes sean habilitados a llevar a cabo el mantenimiento del ciclo de vida de un proyecto de manera colaborativa.

Mediante esta actividad, se logra robustecer y validar conceptos básicos del proceso de mantenimiento de Software, poner en práctica y desarrollar de iteración con los sistemas de revisiones.

En cuanto a las habilidades técnicas-actitudinales, se fomentará y reforzará la noción del trabajo colaborativo, responsable con los horarios y citas de reunión y trabajo en equipo. Además, se integrarán conocimientos clave como la gestión del cambio y trabajo en equipo.

Historias de usuario

Historia de Usuario 1: Registro de Deportistas

Como: Administrador del torneo

Quiero: Registrar los datos de los deportistas que participarán en el torneo

Para que: Se pueda llevar un control detallado de los jugadores, sus equipos y posiciones

Criterios de Aceptación:

El sistema debe permitir ingresar datos como nombre, apellido, edad, equipo, posición y número de camiseta.

Debe validar que la edad del jugador cumpla con los requisitos del torneo (14-18 años).

Los datos deben guardarse en la base de datos en la colección deportistas.

Historia de Usuario 2: Gestión de Árbitros

Como: Organizador del torneo

Quiero: Registrar y gestionar la disponibilidad de los árbitros

Para que: Puedan asignarse de manera eficiente a los partidos programados

Criterios de Aceptación:

El sistema debe permitir ingresar datos como nombre, apellido, experiencia y disponibilidad.

Debe validar que los árbitros registrados tengan un mínimo de 1 año de experiencia.

Los datos deben almacenarse en la colección arbitros y ser consultables al programar un encuentro.

Historia de Usuario 3: Creación del Calendario de Partidos

Como: Administrador del torneo

Quiero: Programar los encuentros entre los equipos participantes

Para que: Se garantice una planificación adecuada del torneo

Criterios de Aceptación:

El sistema debe permitir definir fecha, hora, ubicación, equipos participantes y árbitro asignado.

Debe verificar la disponibilidad de árbitros y evitar conflictos de horarios entre partidos.

La información debe registrarse en la colección encuentros.

Historia de Usuario 4: Registro de Resultados de Partidos

Como: Árbitro del torneo

Quiero: Registrar los resultados de los partidos al finalizar cada encuentro

Para que: La tabla de posiciones y estadísticas de los equipos se actualicen automáticamente

Criterios de Aceptación:

El sistema debe permitir ingresar el puntaje de ambos equipos al final del partido.

La información debe reflejarse en la colección resultados y actualizar la colección equipos.

Debe recalcular la posición de los equipos en la tabla tras cada actualización.

Historia de Usuario 5: Consulta de la Tabla de Posiciones

Como: Participante del torneo

Quiero: Consultar la tabla de posiciones en tiempo real

Para que: Pueda conocer el desempeño de mi equipo y el de los demás equipos

Criterios de Aceptación:

El sistema debe mostrar la posición de los equipos, partidos jugados, victorias, derrotas y puntos obtenidos.

Los datos deben actualizarse automáticamente tras cada partido.

La tabla debe ser accesible desde cualquier dispositivo con conexión a internet.

Conclusiones

Conocimiento técnico robusto: con esta actividad, los integrantes del grupo tuvieron la oportunidad de aprender el proceso de versionamiento de software y pusieron en práctica los comandos de Git esenciales, que son add, commit, branch y push.

La creación del proyecto en GitHub permitió a los estudiantes trabajar juntos y desarrollar habilidades concomitantes para trabajar juntos en como grupo. Esto incluye responsabilidad compartida, trabajos cronológicamente iguales y relación abierta entre los miembros.

La presente experiencia ayudó a consolidar el procesamiento de los procesos de mantenimiento, donde se resalta la importancia de las ventajas presentadas por el control de versiones como una herramienta esencial para garantizar la trazabilidad y la organización durante el proceso de desarrollo continuado del software.

Esta actividad contribuyó a fomentar el compromiso, la disposición a la solución de problemas en manera grupales y la orientación con respecto a las actitudes hacia el uso de las herramientas tecnológicas contemporáneas, colaborando con el desarrollo no solo técnico, sino personal y profesional de los alumnos

Bibliografía

Guzmán, D. I., Islas, U. C., Corona, C. P., & Pedroza-Méndez, B. E. (2014). Metodología ágil Scrumban en el proceso de desarrollo y mantenimiento de software de la norma MoProSoft. *Res. Comput. Sci.*, 79, 97-107.

MERIZALDE, C., & GABRIEL, S. (2024). Diseño de software para el registro del historial de mantenimiento de equipos en las pequeñas y medianas empresas (PYMES) del sector de servicios técnicos profesional de computadoras en la ciudad de Guayaquil durante el periodo 2015-2016 (Tesis de licenciatura).

Pérez Carvajal, R. J. (2016). Mantenimiento del software (UF1894). Málaga: IC Editorial. (Capítulo 1, pp. 11-31; Capítulo 2, pp. 35-57).

Pressman, R. S. (2014). Ingeniería de Software: Un enfoque práctico (9ª ed.). McGraw-Hill.

Sommerville, I. (2011). Ingeniería de Software (9ª ed.). Pearson Educación.

Vargas-Vargas, I., Estupiñán-Díaz, S., & Díaz-Molina, A. (2017). Actualidad mundial de los sistemas de gestión del mantenimiento. *ICIDCA. Sobre los Derivados de la Caña de Azúcar*, 51(2), 10-16.

Software Engineering Body of Knowledge (SWEBoK). (2014). IEEE Computer Society. <https://www.computer.org/education/bodies-of-knowledge/software-engineering>