



Lineamientos, roles y responsabilidades del equipo

Integrantes:

Yendry Badilla

Alejandro Vasquez Oviedo

Brian Wagemans Alvarado

Steven Castro

Algoritmos y Estructuras de Datos II

Luis Diego Noguera Mena

Miércoles 12 de agosto, 2020

Contenido

Introducción.....	3
Lineamientos	4
Roles y responsabilidades	5
Plan de iteraciones	6
Bitácora	8

Introducción

Al trabajar en conjunto con otras personas es inevitable que a lo largo de la realización de cualquier tarea se termine por presentar algún conflicto que afecte de forma directa la efectividad del equipo, donde se termina por desperdiciar tiempo vital necesario para alcanzar una completitud idónea ante el trabajo realizado, es por esta razón que en ese documento se establecerán los lineamientos necesarios para que el trabajo ocurra de manera armónica y provechosa en todo momento, evitando así los problemas más comunes. Su fin es además el de delimitar de manera clara y concisa los roles de cada uno de los miembros, así como sus respectivas responsabilidades. Otro de los apartados pretende mostrar el plan de iteraciones planteado por el equipo con referencia a las historias de usuario y a la carga que tiene cada miembro.

Lineamientos

1. Los miembros del equipo se comprometen a responder los diferentes mensajes.
enviados por cualquier otro integrante independientemente la plataforma en la que se
reciba dicho mensaje.
2. Las reuniones realizadas por algún miembro del equipo son de carácter obligatorio.
3. Se deben realizar reportes de las tareas realizadas durante cada sesión individual o
grupal de trabajo.
4. Cada integrante es responsable de velar por la completitud de sus entradas en la
bitácora.
5. Las tareas asignadas deberán reportarse como concluidas dentro del período
internamente definido.
6. En caso de que se presente algún inconveniente que imposibilite la asistencia a
reuniones o completitud de tarea asignada debe reportarse a los demás miembros lo
antes posible.

Roles y responsabilidades

Coordinador: Alejandro Vásquez.

Investigador: Yendry Badilla, Steven Castro, Alejandro Vásquez, Brian Wagemans.

Implementador: Yendry Badilla, Steven Castro.

Como se determina en los roles cada uno de los miembros del equipo son responsables de investigar según sea la tarea que les fue asignada, de forma más concreta Steven Castro es responsable de la investigación acerca del Algoritmo de Huffman y MD5, Brian Wagemans de las instrucciones que se requieran realizar en la base de datos, Yendry Badilla de los temas referentes a la elaboración de la RestAPI, su conexión con la base de datos, y Alejandro Vásquez sobre el manejo de archivos y la realización del diff por parte del servidor, así como del diseño inicial de la base de Datos. Los principales encargados de la implementación de los resultados de las diferentes investigaciones son los miembros Yendry Badilla y Steven Castro, siendo estos los responsables de la integración completa de las funcionalidades en un único programa final.

Plan de iteraciones

1. Primera iteración

- 1.1. Yo como usuario quiero guardar los archivos en los que estoy trabajando en un manejador de versiones para tener un respaldo de mi trabajo en la nube (got sync)
- 1.2. Yo como usuario quiero poder crear un repositorio en el manejador de versiones con un nombre significativo para mí para tener fácil acceso a mis archivos (got init)
- 1.3. Yo como usuario quiero tener un archivo que ignore los archivos especificados dentro de él para que no se suban al manejador de versiones (.gotignore)
- 1.4. Yo como usuario quiero una función para agregar archivos al commit (got add)
- 1.5. Yo como usuario quiero crear commits para poder actualizar los archivos que se encuentran en el servidor (got commit)
- 1.6. Yo como usuario quiero que el servidor utilice un REST API para que los comandos sean fáciles de escribir

La distribución de tareas en la primera iteración, siguiendo los roles previamente establecido, es de la siguiente forma:

- 1.1 Yendry Badilla, Alejandro Vásquez, Brian Wagemans
- 1.2 Yendry Badilla, Alejandro Vásquez, Brian Wagemans
- 1.3 Alejandro Vásquez, Brian Wagemans
- 1.4 Yendry Badilla, Alejandro Vásquez
- 1.5 Steven Castro, Brian Wagemans

- 1.6 Yendry Badilla
2. Segunda iteración
- 2.1. Yo como usuario quiero un manejador de versiones para trabajos en colaboración con otras personas.
 - 2.2. Yo como usuario quiero poder regresar un archivo a una versión anterior (got rollback)
 - 2.3. Yo como usuario quiero deshacer cambios en el repositorio local (got reset)
 - 2.4. Yo como usuario quiero que el servidor utilice MySQL para almacenar los datos del repositorio

La distribución de tareas en la segunda iteración, siguiendo los roles previamente establecido, es de la siguiente forma:

- 2.1 Yendry Badilla, Steven Castro
 - 2.2 Yendry Badilla, Alejandro Vásquez
 - 2.3 Brian Wagemans, Steven Castro
 - 2.4 Yendry Badilla, Alejandro Vásquez
3. Tercera iteración
- 3.1. Yo como usuario quiero un comando para que se muestren todos los comandos posibles en la interfaz (got help)
 - 3.2. Yo como usuario quiero poder ver el estado de mi commit actual (got status)

La distribución de tareas en la tercera iteración, siguiendo los roles previamente establecido, es de la siguiente forma:

- 3.1 Brian Wagemans
- 3.2 Steven Castro, Yendry Badilla, Alejandro Vásquez

Bitácora

1 de agosto

Alejandro: se investiga a profundidad la utilización de MySQL en ArchLinux, así como MariaDB.

2 de agosto

Yendry: se investiga acerca de RestAPI, conceptos, funciones e implementaciones.

Todos los integrantes: se discute la tecnología que será utilizada para la elaboración del RestAPI que funcionará junto al servidor, analizando las diferentes bibliotecas provistas en distintos lenguajes. Se determina que se trabajará en Windows debido a la incompatibilidad inherente de Arch con MySQL para cumplir con los requerimientos del proyecto.

3 de agosto

Alejandro: se inicia el planeamiento de la estructura de la base de datos, con las tablas y sus respectivas columnas para un correcto almacenamiento de los datos necesarios para el proyecto. Se propone una reunión de todo el equipo para delimitar de mejor manera los detalles de dicho planeamiento.

Yendry: se investiga acerca de la forma para comunicar una RestAPI en JavaScript con la base de datos, tanto la conexión propia como la forma de realizar solicitudes

Todos los integrantes: se realiza una reunión para explicar, analizar y terminar de desarrollar la lógica y arquitectura de la base de datos a utilizar en el proyecto, así como conocer los resultados de investigación y realizar pruebas respecto a la conexión RestAPI-MySQL.

4 de agosto

Yendry: continúa la investigación sobre el funcionamiento del RestAPI

Todos los integrantes: se realiza otra reunión para continuar definiendo la estructura de la base de datos y se inicia la investigación sobre cómo enviar solicitudes HTTP desde el cliente en C++

5 de agosto

Steven: se profundiza en la investigación para realizar solicitudes a la base de datos desde Nodejs.

Yendry: se crea la aplicación de consola en el cliente.

Alejandro: se investiga sobre la obtención de un input por parte del usuario así como su respectiva conversión a string

6 de agosto

Yendry: se investiga sobre la mejor forma para almacenar los datos de cada archivo dentro de la base de datos, se propone usar TEXT para facilitar la comparación entre los archivos.

7 de agosto

Brian-Alejandro: se realiza un método para obtener los diferentes parámetros introducidos por el usuario en cada comando, siendo funcional para n parámetros, así como la reestructuración necesaria del cliente para la correcta interpretación.

Yendry-Alejandro: se investiga el método para enviar solicitudes desde C++ hacia el RestAPI, así como se repasa la estructura actual de la base de datos en caso de que requiera correcciones

Yendry: investigación general sobre el manejo de metadata, tipos, endpoints, establecimiento de rutas y el flujo del programa.

8 de agosto

Yendry: intento de corrección de errores de la lista.

Yendry-Steven: se trabaja en el comando add del proyecto.

9 de agosto

Alejandro: se investiga acerca del manejo de archivos, la obtención de la información necesaria para la base de datos, búsqueda en subdirectorios y lectura de los mismos.

Investigación de las funciones diff y patch.

Yendry-Steven: investigación de la biblioteca curl así como su implementación correcta.

Requiere investigación para envío de parámetros.

Todos los miembros: prueba general del comando add, init, commit, falta manejo de datos por parte del RestAPI, se implementa el algoritmo de Huffman

10 de agosto

Steven: investigación y documentación del huffman.

Brian: documentación y ordenamiento de GitHub, se comienza a colocar las issues previamente definidas en GitHub, funciones de delimitaciones de parámetros en los comandos.

Yendry-Alejandro: se hacen ajustes en el comando commit del cliente, se comienzan a adaptar estos cambios en el servidor, se reajusta la base de datos, se crea la función para reconstruir el archivo a partir de una biblioteca diff para JS, falta probar esta función. Se plantea la lógica de los comandos faltantes así como se crea el esqueleto base de dichas operaciones.

11 de agosto

Steven: se investiga sobre MD5 y se incorpora una implementación al repositorio. Se provee la forma para dejar de utilizar Nodemon al ejecutar el servidor. Se termina la documentación del algoritmo de Huffman y se realiza en su completitud la del MD5.

Brian: se soluciona el problema para verificar si existe o no un archivo base. Se completan las issues en GitHub.

Yendry-Alejandro: se corrigen errores al eliminar la lista de add en el cliente, se investiga la forma de obtener el responde del HTTP request con curl, se investiga e implementa la escritura de un responde en el archivo de destino en el cliente. Se corrige el error previo con

los relative paths donde se necesita un forward slash y no un backslash. Se realizan los comandos reset, rollback y sync, falta probar estos dos últimos. Se plantea la lógica para el comando status file.

12 de agosto

Brian: se realiza el diagrama de clases basado en el diseño original del proyecto.

Steven: generar el doxygen del proyecto.

Yendry-Alejandro: se prueban los comandos sync y rollback, funcionan de la forma esperada. Se comienza a plantear la lógica para el comando sync, se crea en su totalidad y se prueba. Se realizan ajustes menores al código.