UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
CURSO. LABORATORIO DE ANÁLISIS Y DISEÑO 2
HOJA DE TRABAJO NO.1
NOMBRE: MARÍA FERNANDA RODRÍGUEZ SANTOS
CARNÉ: 201020946
FECHA DE ENTREGA: 10/08/2017

HOJA DE TRABAJO NO.1

PARTE I

¿Qué es un sistema de control de versiones?

Es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que se pueda recuperar las versiones específicas más adelante. Cualquier tipo de archivo que se encuentra en un ordenador puede ponerse bajo control de versiones. Se utiliza principalmente en la industria informática para controlar distintas versiones del código fuente dando lugar a los sistemas de control de código fuente.

Existen varios tipos de sistemas de control de versiones como los locales, los distribuidos y los centralizados.

Referencia, *Git - Acerca del control de versiones*. (2017). *Git-scm.com*. Retrieved 10 August 2017, from https://git-scm.com/book/es/v1/Empezando-Acerca-del-control-de-versiones

¿Qué es un repositorio?

Es un sitio centralizado donde se almacena y mantiene información digital, normalmente archivos informáticos o bases de datos. Por lo que se podría definirse como la base de datos fundamental del diseño, no sólo guarda datos, sino también algoritmos de diseño y, en general, elementos de software necesarios para el trabajo en la programación.

Referencia, *Repositorio*. (2017). *Mastermagazine.info*. Retrieved 10 August 2017, from https://www.mastermagazine.info/termino/6534.php

DEFINIR:

- ✓ Árbol: En un árbol se pueden ir creando ramas que contengan información importante respecto al tiempo de vida de un programa, en el que es de suma importancia la constancia del desarrollo de información.
- ✓ Revisión: Es una forma de verificar que la información que se está desarrollando sea verídica.
- ✓ Release: Es el lanzamiento de una parte del sistema que ha pasado por un proceso de desarrollo por lo que ha habido revisiones posteriores que conllevan al software lanzar nuevas actualizaciones.
- ✓ Rama: Es un nodo que contiene información, que será útil dependiendo del contexto en el que se está desarrollando.
- ✓ Etiqueta: Puede identificar la información sobre el objeto que se está creando.
- ✓ Versión: Son mejoras del sistema que se ven reflejados respecto al tiempo.
- ✓ Commit: Confirmación de cambios, hacia un destino.
- ✓ Update: Actualización de cambios de algún proyecto que está desarrollo constantemente.
- ✓ Check out: La verificación o movimiento de objetos que contienen información diferente a otros componentes.
- ✓ Merge: Combinar información que está en ramas con otras ramas nuevas.

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
CURSO. LABORATORIO DE ANÁLISIS Y DISEÑO 2
HOJA DE TRABAJO NO.1
NOMBRE: MARÍA FERNANDA RODRÍGUEZ SANTOS
CARNÉ: 201020946
FECHA DE ENTREGA: 10/08/2017

DIFERENCIA ENTRE TRUNK Y BRANCH:

TRUNK (tronco): Es una línea principal de desarrollo, en donde se llevan los cambios complejos todo el tiempo. La manera ideal es la compilación y hacer pruebas en todo momento.

BRANCH (branch): Es cuando se hacen cambios importantes que romperán de alguna manera la compilación, las pruebas, los experimentos o intentos de optimización, para ello se debe crear nuevas ramas de desarrollo, en resumen es una copia del código o de la rama de la que se deriva, en ella se harán nuevos cambios, se integrará los arreglos que se puedan haberse haciendo en el trunk, una vez terminando este desarrollo se integrarán los nuevos cambios en el trunk.

Referencia, Trunk, branch y tag. (2017). *Mundo Geek*. Retrieved 10 August 2017, from http://mundogeek.net/archivos/2011/01/13/trunk-branch-y-tag/

¿Qué es y por qué es necesaria la integración continua?

Es la práctica de realizar despliegues incrementales basados en la funcionalidad de lo que los desarrolladores "liberan" y que son validados mediante suites de pruebas automatizadas (quizás manuales) en ambientes que no son exactamente producción.

¿Por qué es necesaria? Hoy en día es considerada una filosofía de trabajo ya que las formas están alineadas bajo un mismo tema en que el producto está listo para ser liberado. Por lo que esto cambia el paradigma del proyecto y se incorpora cambios de todo el equipo en el software generando así cambios incrementales y siempre listos para enviarlos a producción. Una de las principales funcionalidades que implementan los desarrolladores son las pruebas unitarias y potencialmente automatizadas de la interfaz de usuario y envía los cambios al repositorio de fuentes. Es una manera de lograr más seguridad en los cambios que se realizan con el tiempo en el software ya que se notifican por medio de pruebas unitarias si algo está saliendo mal.

Referencia, *La integración continua revoluciona el desarrollo de software en Redbee*. (2017). *SearchDataCenter en Español*. Retrieved 10 August 2017, from http://searchdatacenter.techtarget.com/es/cronica/La-integracion-continua-revoluciona-el-desarrollo-de-software-en-Redbee

Gomez, D. (2017). *Integración Contínua*. *Dos Ideas*. Retrieved 10 August 2017, from https://dosideas.com/noticias/metodologias/309-integracion-continua

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
CURSO. LABORATORIO DE ANÁLISIS Y DISEÑO 2
HOJA DE TRABAJO NO.1
NOMBRE: MARÍA FERNANDA RODRÍGUEZ SANTOS
CARNÉ: 201020946
FECHA DE ENTREGA: 10/08/2017

Hacer un diagrama donde se explique el proceso de la integración continua