

Sistema de Control de Versiones

Yerson Coaquira Calizaya

18 de agosto de 2018

1. Introducción

Este es un pequeño trabajo de investigación para el curso de Base de Datos II. Básicamente el trabajo consta de tres partes: Los objetivos, el Marco Teórico y las Conclusiones.

2. Marco Teórico

2.1. El SCV

Un sistema de control de versiones es una herramienta que registra todos los cambios hechos en uno o más proyectos, guardando así versiones del producto en todas sus fases del desarrollo. Las versiones son como fotografías que registran su estado en ese momento del tiempo y se van guardando a medida que se hacen modificaciones al código fuente.

Un sistema de control de versiones debe proporcionar:

- * Mecanismo de almacenamiento de los elementos que deba gestionar.
- * Posibilidad de realizar cambios sobre los elementos almacenados (ej. modificaciones parciales, añadir, borrar, renombrar o mover elementos).
- * Registro histórico de las acciones realizadas con cada elemento o conjunto de elementos (normalmente pudiendo volver o extraer un estado anterior del producto).

En los principales tipos de SCV tenemos en las siguientes gráficas:

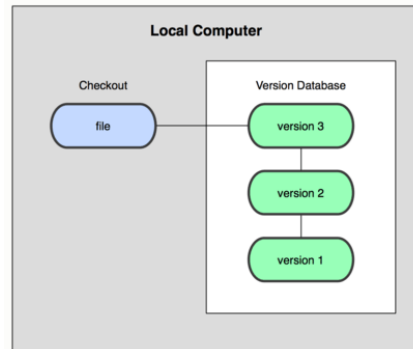


Figura 1: Sistemas de Control de Versiones Locales

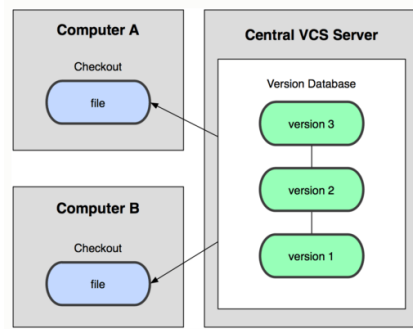


Figura 2: Sistemas de Control de Versiones Centralizados

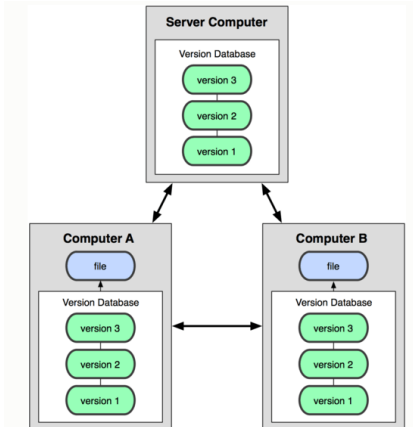


Figura 3: Sistemas de Control de Versiones Distribuidos



2.2. El Git

Es un software para controlar versiones de nuestras aplicaciones, ejem: Estamos realizando un ERP y queremos compartirlo entre todos los desarrolladores de nuestro equipo para que los cambios que hagan puedan ser sincronizados en uno solo.

2.2.1. GitHub

Es un host para Git que nos permite almacenar proyectos open sources. Es decir que los proyectos que alojemos van a ser públicos, esto es ideal para compartir código con otros y generar colaboración entre los demás para dar mejoras a los proyectos que se publiquen. Podemos tener un repositorio privado pero tiene un costo. El funcionamiento es fácil:

- * Lo primero que debemos hacer es crear una cuenta en <https://github.com>. La activamos por mail y ya podemos crear nuestros repositorios. Los repositorios de Github son el almacén que utilizamos para guardar nuestro código.
- * Github nos ofrece la opción de crear un repositorio vacío, recomendable cuando vamos a iniciar un nuevo desarrollo, o la opción de importar un proyecto ya existente, elegimos la que más nos convenga y mediante unos pocos comandos de consola configuramos la rama principal de nuestro repositorio, que por defecto se llamará "master".
- * Cada programador puede crear sus propias ramas de desarrollo, donde tiene que llevar a cabo sus modificaciones, sin interferir en el trabajo de sus compañeros. Cuando terminamos y validamos un desarrollo paralelo, lo unimos con la rama principal y todos los miembros del equipo pueden descargar las nuevas modificaciones, sin alterar los desarrollos que estén llevando cabo en ese momento.
- * Después de alojar el repositorio público en Github.com, cualquier usuario de la comunidad podrá aportar ideas, hacer un seguimiento del proyecto, incluso copiarlo y modificarlo a su gusto bajo la misma licencia.



2.3. El VSTS

El Visual Studio Team Services es un servicio que ofrece Microsoft para organizar nuestro proyecto, usemos o no el IDE Visual Studio, que nos puede ser útil.

- * Puede compilar windows, IOS, Android, Java, LInux y Xamarin.
- * Usa su DLS habitual.
- * Prueba y publica sus apps.

Ademas de eso también es de fácil personalización como:

- * Los procesos son editables a través de la Web.
- * No hay que trabajar con XAML.
- * Permite usar conocimientos de script.