



## 1. Datos Generales de la asignatura

<b>Nombre de la asignatura:</b>	<b>Sistemas de control de versiones</b>
<b>Clave de la asignatura:</b>	<b>TDD-2103</b>
<b>SATCA<sup>1</sup>:</b>	<b>2-3-5</b>
<b>Carrera:</b>	<b>Ing. en Sistemas Computacionales</b>

## 2. Presentación

### Caracterización de la asignatura

Esta asignatura aporta al perfil del Ingeniero en Sistemas Computacionales las siguientes habilidades:

- Crear e iniciar repositorios para proyectos nuevos y retomar proyectos legados.
- Decidir qué flujos de trabajo y estrategias de ramas puedes usar para controlar tus versiones según tus necesidades.
- Entender cómo resolver conflictos al incluir cambios de otras ramas y otros desarrolladores.
- Repositorios de bibliotecas
- Encontrar defectos con la ayuda de Git.

Sistemas de control de versiones aporta la capacidad de Validación y testing como un proceso continuo que se lleva a cabo durante todo el ciclo de vida del software, desde que se comienza a programar hasta que, luego de ser implementado y utilizado.

El control de versiones ayuda a los equipos a resolver este tipo de problemas al realizar un seguimiento de todos los cambios individuales de cada colaborador y al contribuir a evitar que el trabajo concurrente entre en conflicto. Los cambios realizados en una parte del software pueden ser incompatibles con los que ha hecho otro desarrollador que está trabajando al mismo tiempo. Este problema debería detectarse y solucionarse de manera ordenada sin bloquear el trabajo del resto del equipo. Además, en todo el desarrollo de software, cualquier cambio puede introducir nuevos errores por sí mismo y el nuevo software no es fiable hasta que se prueba. De este modo, las pruebas y el desarrollo van de la mano hasta que está lista una nueva versión.

Los sistemas de control de versiones son una categoría de herramientas de software que ayudan a un equipo de software a gestionar los cambios en el código fuente a lo largo del tiempo. El software de control de versiones realiza un seguimiento de todas las modificaciones en el código en un tipo especial de base de datos. Si se comete un error, los desarrolladores pueden ir atrás en el tiempo y comparar las versiones anteriores del código para ayudar a resolver el error al tiempo que se minimizan las



interrupciones para todos los miembros del equipo.

**Intención didáctica**

La asignatura se organiza en cuatro temas, integrados por contenidos conceptuales, Fundamentos de Sistemas de Control de Versiones, Sistemas de control local de versiones, Sistemas de control de versiones en la nube y Clientes Gráficos para Sistemas de control de versiones, la cual será capaz de desarrollar en el estudiante la habilidad para emplear las nuevas tecnologías para el control de versiones:

En el primer tema el estudiante será capaz de identificar y relacionarse con los diferentes tipos de plataformas para el control de versiones.

En el segundo tema el estudiante será capaz de identificar y trabajar con los comandos del sistema de control local de versiones, como lo es GIT, así como su configuración.

En el tercer tema el alumno será capaz de configurar el sistema de control de versiones GIT HUB, y trabajar con las operaciones básicas de GitHub.

En el cuarto tema el estudiante será capaz de identificar y trabajar con Clientes Gráficos para Sistemas de control de versiones.

**3. Participantes en el diseño y seguimiento curricular del programa**

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Instituto Tecnológico de Tuxtepec	Departamento de Sistemas y Computación	Módulo de Especialidad.

**4. Competencia(s) a desarrollar**

<b>Competencia(s) específica(s) de la asignatura</b>
Desarrollar aplicaciones donde se involucren varios desarrolladores, aprenderá y practicará los fundamentos, funcionalidades y diferentes flujos de trabajo en los que se puede usar sistemas de control de versiones, herramientas como control de versiones, desde proyectos personales hasta proyectos de gran tamaño en los que se involucran decenas de desarrolladores.



### 5. Competencias previas

Dominio de sistema operativo Linux, Windows y el Shell a nivel de usuario consola de comandos desde la terminal.

### 6. Temario

No.	Temas	Subtemas
1	Fundamentos del control de versiones	1.1 ¿Que es y porque es importante el control de versiones? 1.2 Características fundamentales 1.3 CVS centralizado/distribuido 1.4 Software para el control de versiones 1.4.1 Git 1.4.2 CVS 1.4.3 Mercurial 1.4.4 Monotone
2	Sistemas de control local de versiones.	2.1. Que es y para que sirve. 2.2 Que es el comando Git. 2.3 Fundamentos y arquitectura interna de Git. 2.4 Configuración de Git . 2.5 Comandos de Git. 2.5.1 Obtención y Creación de proyectos. 2.5.2 Captura. 2.5.3 Ramificación y fusión. 2.5.4 Compartir y actualizar proyectos. 2.5.5 Inspección y comparación. 2.5.6 Parcheo.
3	Sistemas de control de versiones en la nube.	3.1. Que es y para que sirve. 3.2Arquitectura interna de GitHub . 3.3 Configuración de GitHub. 3.4 Operaciones básicas de GitHub
4	Clientes Gráficos. para Sistemas de control de versiones	4.1.GitKraken 4.2 Sublime merge 4.3 Git-cola 4.4 SmartGit 4.5 Git GUI