



1. Datos Generales de la asignatura

Nombre de la asignatura:	Sistemas de control de versiones
Clave de la asignatura:	TDD-2103
SATCA¹:	2-3-5
Carrera:	Ing. en Sistemas Computacionales

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

Competencia(s) específica(s) de la asignatura
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

7. Actividades de aprendizaje de los temas

1. Fundamentos del control de versiones.	
Competencias	Actividades de aprendizaje
<p>Específica(s): Investiga e identifica las diferentes plataformas para el control de versiones.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> Habilidades para buscar, procesar y analizar información procedente de diversas fuentes. Capacidad de abstracción, análisis y síntesis. Capacidad de trabajo en equipo. 	<ul style="list-style-type: none"> Investigar las características de los diferentes sistemas para el control de versiones y realiza un resumen. Investigar las diferencias entre CVS centralizado y distribuido, y realiza un cuadro comparativo.
2. Sistemas de control local de versiones.	
Competencias	Actividades de aprendizaje
<p>Específica(s): Gestiona, planifica, diseña, programa, prueba, implementa y controla las versiones automáticas de los cambios que se realizan sobre uno o varios ficheros de un servicio local.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> Habilidades para buscar, procesar y analizar información procedente de diversas fuentes. Capacidad de abstracción, análisis y síntesis Capacidad de aplicar los conocimientos en la práctica Capacidad para identificar, plantear y resolver problemas Capacidad de trabajo en equipo 	<ul style="list-style-type: none"> Ejecutar los comandos de Git de obtención y Creación de proyectos, Captura, Ramificación y fusión, Compartir y actualizar proyectos, Inspección y comparación y de Parcheo y elabora una bitacora donde se muestren los resultados. Realizar prácticas de laboratorio utilizando un git. Comparar el código de un archivo, y ver las diferencias entre versiones. Restaurar versiones antiguas. Fusionar cambios entre distintas versiones. Trabajar con distintas ramas de un proyecto, por ejemplo la de producción y desarrollo. Facilitar la colaboración entre el equipo de desarrollo y la incorporación controlada de modificaciones al código del proyecto.

3. Sistemas de control de versiones en la nube.	
Competencias	Actividades de aprendizaje
<p>Específica(s): Gestiona, planifica, diseña, programa, prueba, implementa y controla las versiones automáticas de los cambios que se realizan sobre uno o varios ficheros de un servicio en la nube.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> Habilidades para buscar, procesar y analizar información procedente de diversas fuentes. Capacidad de abstracción, análisis y síntesis Capacidad de aplicar los conocimientos en la práctica Capacidad para identificar, plantear y resolver problemas Capacidad de trabajo en equipo 	<ul style="list-style-type: none"> Realizar operaciones básicas Crear un repositorio, Clonar un repositorio, Actualizar un repositorio local con cambios remotos, Crear un commit, Actualizar un repositorio remoto con cambios locales, Crear una rama local y hacerla más tarde remota. Importar cambios de una rama en otra. Realizar prácticas de laboratorio utilizando un github. Comparar el código de un archivo, y ver las diferencias entre versiones. Restaurar versiones antiguas. Fusionar cambios entre distintas versiones. Trabajar con distintas ramas de un proyecto, por ejemplo, la de producción y desarrollo. Facilitar la colaboración entre el equipo de desarrollo y la incorporación controlada de modificaciones al código del proyecto.
4. Clientes Gráficos. para Sistemas de control de versiones	
Competencias	Actividades de aprendizaje
<p>Específica(s): Manejo de clientes gráficos para dar mantenimiento a los repositorios con código fuente que contiene muchos archivos utilizando herramientas colaborativas.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> Habilidades para buscar, procesar y analizar información procedente de diversas fuentes. Capacidad de abstracción, análisis 	<ul style="list-style-type: none"> Configuración de la aplicación Cliente. Realizar prácticas de laboratorio utilizando los diferentes clientes gráficos de CVS. Utilizar herramientas colaborativas tales como atribuir confirmaciones y crear solicitudes de extracción. Ejecutar operaciones básicas de Cliente Grafico:

<p>y síntesis</p> <ul style="list-style-type: none"> • Capacidad de aplicar los conocimientos en la práctica • Capacidad para identificar, plantear y resolver problemas • Capacidad de trabajo en equipo 	<ul style="list-style-type: none"> -Fork de un repositorio. -Clonar un repositorio. -Abrir un repositorio. -Elementos de la interfaz. -Añadir Cambios. -Sincronización con repositorios remotos. -Trabajo en múltiples branches. -Pull Requests. -Características Adicionales.
--	---

8. Práctica(s)

TEMA	PRACTICA	DESCRIPCION
2	1	<ul style="list-style-type: none"> • Emplear una de las opciones de manejadores de versiones local para la elaboración de una de las aplicaciones.
	2	<ul style="list-style-type: none"> • Uso de las operaciones básicas para el control de sistemas de versiones: <ul style="list-style-type: none"> -Obtención y Creación de proyectos. -Captura. -Ramificación y fusión. -Compartir y actualizar proyectos. -Inspección y comparación. -Parcheo.
3	3	<ul style="list-style-type: none"> • Desarrollo en equipo de una aplicación empleando una metodología de desarrollo por medio de uno de los sistemas de control de versiones.
4	1	<ul style="list-style-type: none"> • Instalación de un Cliente Gráfico: Por Ejemplo: <ul style="list-style-type: none"> -GitKraken -Sublime merge -Git-cola -SmartGit -Git GUI
	2	<ul style="list-style-type: none"> • Uso de las operaciones del Cliente Grafico.

9. Proyecto de asignatura

El objetivo del proyecto que planteé el docente que imparta esta asignatura, es demostrar el desarrollo y alcance de la(s) competencia(s) de la asignatura, considerando las siguientes fases:

- **Fundamentación:** marco referencial (teórico, conceptual, contextual, legal) en el cual se fundamenta el proyecto de acuerdo con un diagnóstico realizado, mismo que permite a los estudiantes lograr la comprensión de la realidad o situación objeto de estudio para definir un proceso de intervención o hacer el diseño de un modelo.
- **Planeación:** con base en el diagnóstico en esta fase se realiza el diseño del proyecto por parte de los estudiantes con asesoría del docente; implica planificar un proceso: de intervención empresarial, social o comunitario, el diseño de un modelo, entre otros, según el tipo de proyecto, las actividades a realizar los recursos requeridos y el cronograma de trabajo.
- **Ejecución:** consiste en el desarrollo de la planeación del proyecto realizada por parte de los estudiantes con asesoría del docente, es decir en la intervención (social, empresarial), o construcción del modelo propuesto según el tipo de proyecto, es la fase de mayor duración que implica el desempeño de las competencias genéricas y específicas a desarrollar.
- **Evaluación:** es la fase final que aplica un juicio de valor en el contexto laboral-profesión, social e investigativo, ésta se debe realizar a través del reconocimiento de logros y aspectos a mejorar se estará promoviendo el concepto de “evaluación para la mejora continua”, la metacognición, el desarrollo del pensamiento crítico y reflexivo en los estudiantes.

10. Evaluación por competencias

- La evaluación debe ser continua y cotidiana por lo que se debe considerar el desempeño en cada una de las actividades de aprendizaje, haciendo especial énfasis en:
- Rúbricas o productos, señalados en cada unidad académica dentro de las actividades de aprendizaje.
- Prácticas propuestas y su presentación y exposición en plenaria (algunas se evaluarán por equipo).
- Portafolio de evidencias de Información recabada durante las consultas e investigaciones solicitadas, plasmadas en documentos escritos.
- Exámenes teórico-prácticos para comprobar la efectividad del estudiante en la resolución de casos prácticos.



11. Fuentes de información

- Brown, E., 2019. Web development with Node and Express. 2nd ed. O'Reilly Media.
- Tema 10 - Sistemas de Control de Versiones de última generación (DCA). (2021). <https://www.dlsi.ua.es/asignaturas/dca/teoria/t10.pdf>
- Git - Book. (2021). <https://git-scm.com/book/es/v2>
- Pro Git, el libro oficial de Git. (2021). <https://uniwebsidad.com/libros/pro-git>
- OpenLibra | Libros en la categoría: GIT. (2021). <https://openlibra.com/es/collection/search/category/git>
- Dauxon, S., 2016. *Git*. Saint-Herblain: Editions ENI. <https://books.google.com.mx/books?id=kAP-5cbnmPYC&pg=PA274&dq=Git&hl=es-419&sa=X&ved=2ahUKEwjywryKIJXvAhVDLKwKHTxVAt0Q6AEwAHoECAEQAw>
- ARIAS, A., 2017. Control de versiones de software con git/ control of software versions with git. 2nd ed. <https://books.google.com.mx/books?id=uvw0DgAAQBAJ&pg=PA14&dq=Git&hl=es-419&sa=X&ved=2ahUKEwjywryKIJXvAhVDLKwKHTxVAt0Q6AEwAnoECAkQAw>
- OpenLibra. OpenLibra | Libros en la categoría: NoSQL. <https://openlibra.com/es/collection/search/category/nosql>
- 2021. Práctica: Introducción a los Sistemas de Control de Versiones. Ejecución Controlada de Un Programa. https://campusvirtual.ull.es/ocw/pluginfile.php/2170/mod_resource/content/0/perlexamples/node15.htmlhttps://trainingit.es/curso-git/
- Planetalia.com. Curso de Control de Versiones con SVN. <https://www.planetalia.com/cms/view/curso-control-versiones-svn>