UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

INGENIERÍA DE SOFTWARE 2

HERRAMIENTAS PARA LA GESTIÓN DE LA CONFIGURACIÓN DE SOFTWARE

Integrantes del equipo:

UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

CORPORACIÓN UNIVERSITARIA REMINGTON

Carrera : Ingeniería en Sistemas-2024

Video de YOUTUBE de presentación en equipo: https://youtu.be/WhHnXMvHVZ4

Alumnos

1. Brayhan Stiven Orrego Valencia — **correo:** brayhan.orrego.0551@miremington.edu.co
2. Jimmis Jhoan Simanca Rojas — **correo:** jimmis.simanca.7363@miremington.edu.co
3. Oscar Germanico Motta Riaño — **correo:** oscar.motta.5521@miremington.edu.co
4. Valentin Ducuara Leguizamo — **correo:** valentin.ducuara.6611@miremington.edu.co
5. Osvaldo Andres Peña — **correo:** osvaldo.pena.6343@miremington.edu.co

UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

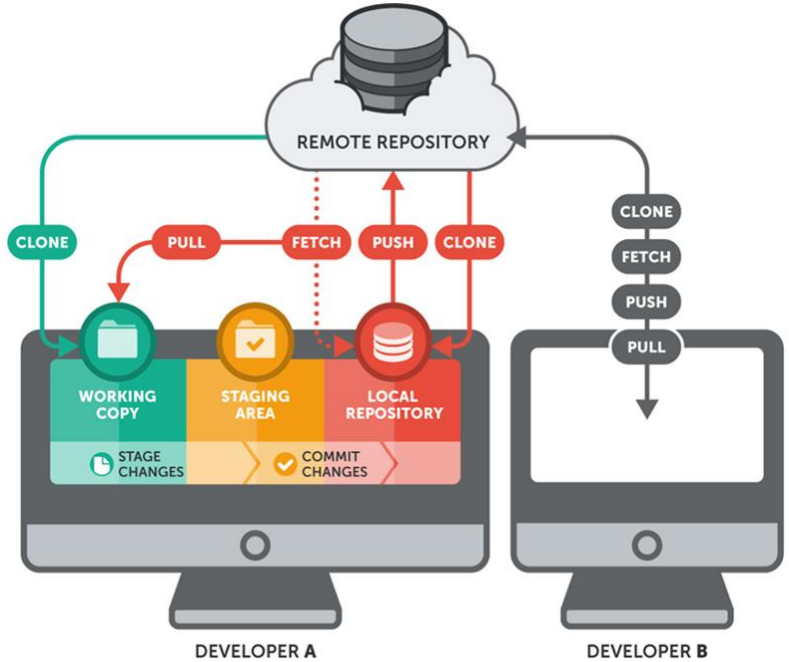
Control de Versiones en Desarrollo de Software

El control de versiones permite gestionar cambios y versiones de código, facilitando la colaboración, recuperación y mejora continua en el desarrollo de software.

UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Git

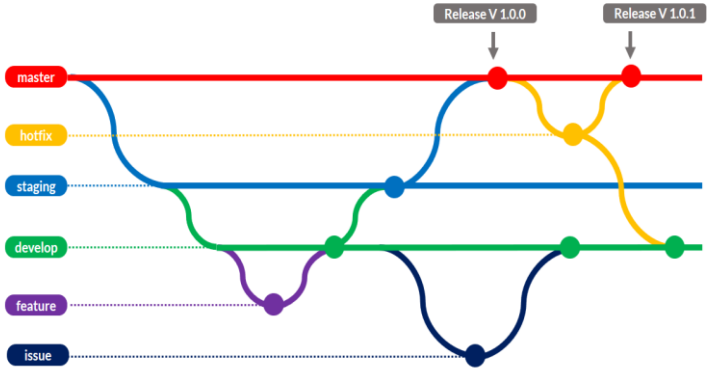
Git es un sistema de control de versiones distribuido que permite trabajar en múltiples ramas, con historial detallado y fusiones flexibles.



UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Control de Cambios en Git

Git usa commits y ramas para gestionar cambios, integrados a través de pull requests. Facilita la colaboración y revisión del código.



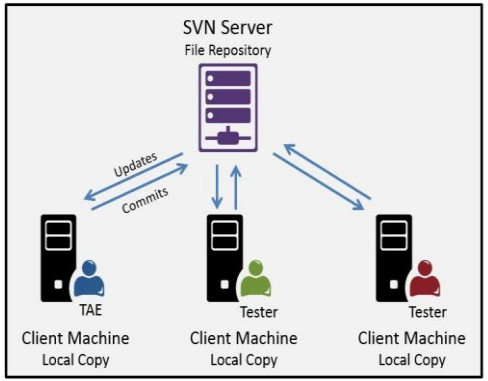
soy DSIT

UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Subversion (SVN)

Subversion es un sistema centralizado, donde los cambios se almacenan en un único repositorio y las revisiones son numeradas secuencialmente.

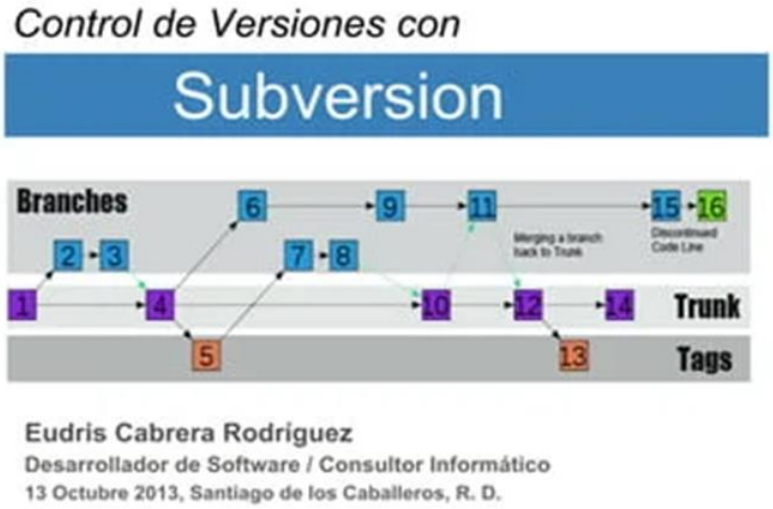
Subversion's Architecture



UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Control de Cambios en SVN

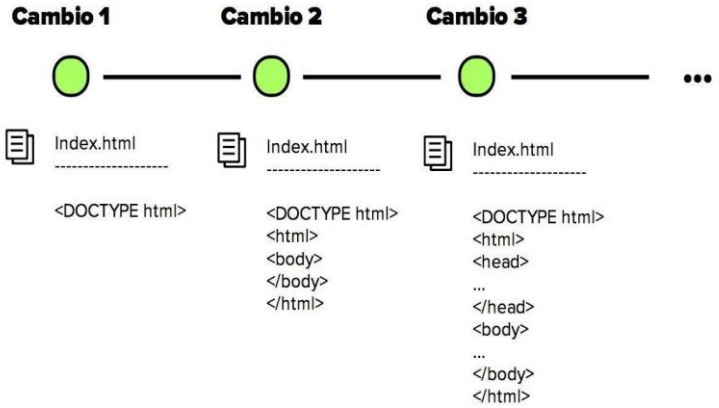
SVN asigna números de revisión a cada cambio, permitiendo recuperar versiones anteriores y resolver conflictos durante las fusiones.



UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Historial de Cambios

Permite rastrear quién hizo cada cambio, cuándo y por qué, facilitando auditorías y corrección de errores.



UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Reversión de Cambios

Si se introduce un error, el software puede revertirse a una versión estable anterior fácilmente.

UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Colaboración Eficiente

Múltiples desarrolladores pueden trabajar simultáneamente en el mismo proyecto sin sobrescribir el trabajo de otros.

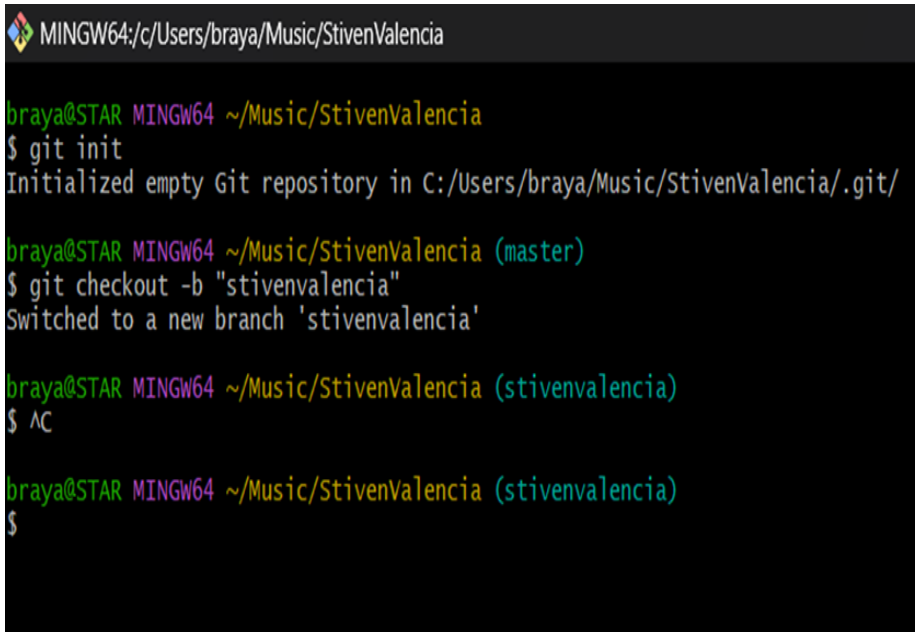
UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Seguridad y Copia de Seguridad

El control de versiones actúa como copia de seguridad automática del código y permite control de acceso.

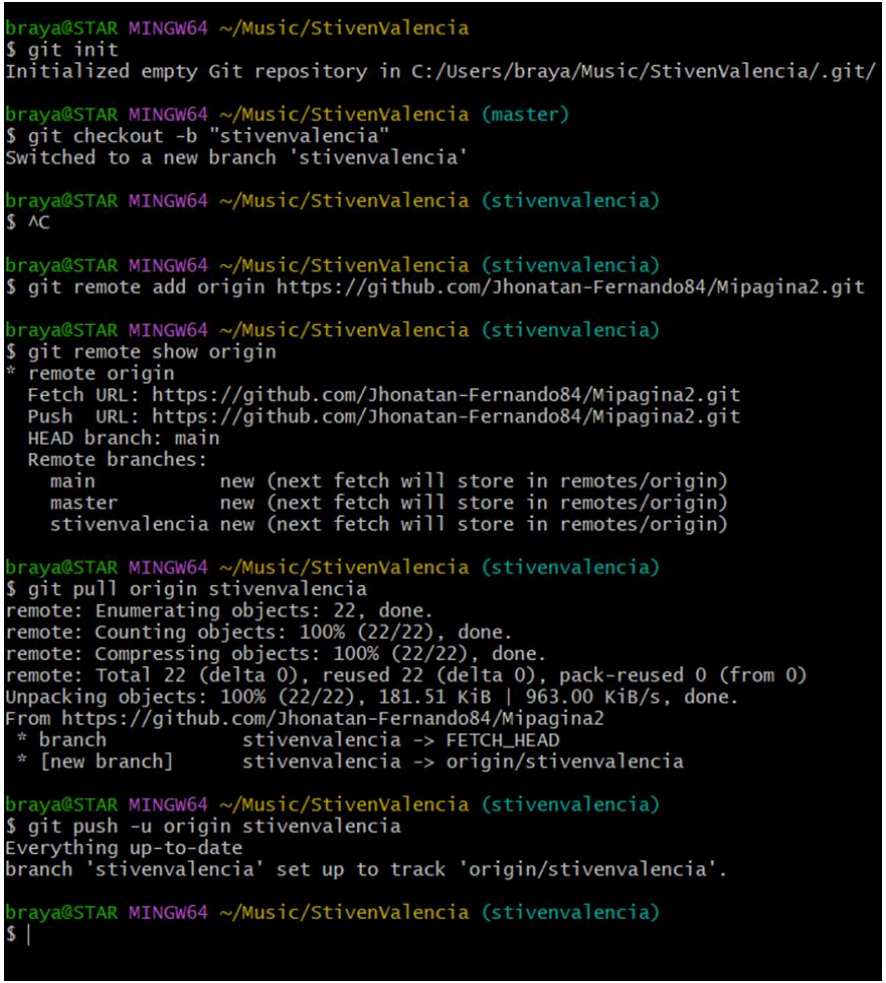
2-.Ejemplo práctico de cómo se realiza el control de versiones implementando Git y GitHub.

- 1-Primero se entra al repositorio que se quiere participar
- 2.-Creamos una carpeta vacía para descargar el repositorio
- 3.-Iniciamos git en la carpeta creada
- 4-Con el comando git init iniciamos git en esta carpeta
- 5-Con el comando git checkout -b "stivenvalencia" nos conectamos a la rama Stiven valencia y en caso de no existir la creamos con este mismo



UNIREMINGTON
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

- 6-. Copiamos el vínculo del repositorio que queremos participar
- 7-. Con el comando git remote add origin https://github.com/Jhonatan-Fernando84/Mipagina2.git nos conectamos al repositorio
- 8-. El comando git remote show origin permite ver más información sobre el remoto origen en Git



- 9-.Con el comando git pull origin stivenvalencia descargamos en nuestra carpeta local el repositorio remoto
- 10-. Con el comando git push -u origin stivenvalencia subimos los cambios realizados
- 11-. Con el comando git commit -m "Cambios realizados prueba" hacemos un comentario corto pero detallado de que cambios se realizaron
- 12-.Con el comando git push -u origin stivenvalencia subimos nuestros cambios y commit al repositorio remoto para ser aprobado o rechazado por el dueño del repositorio

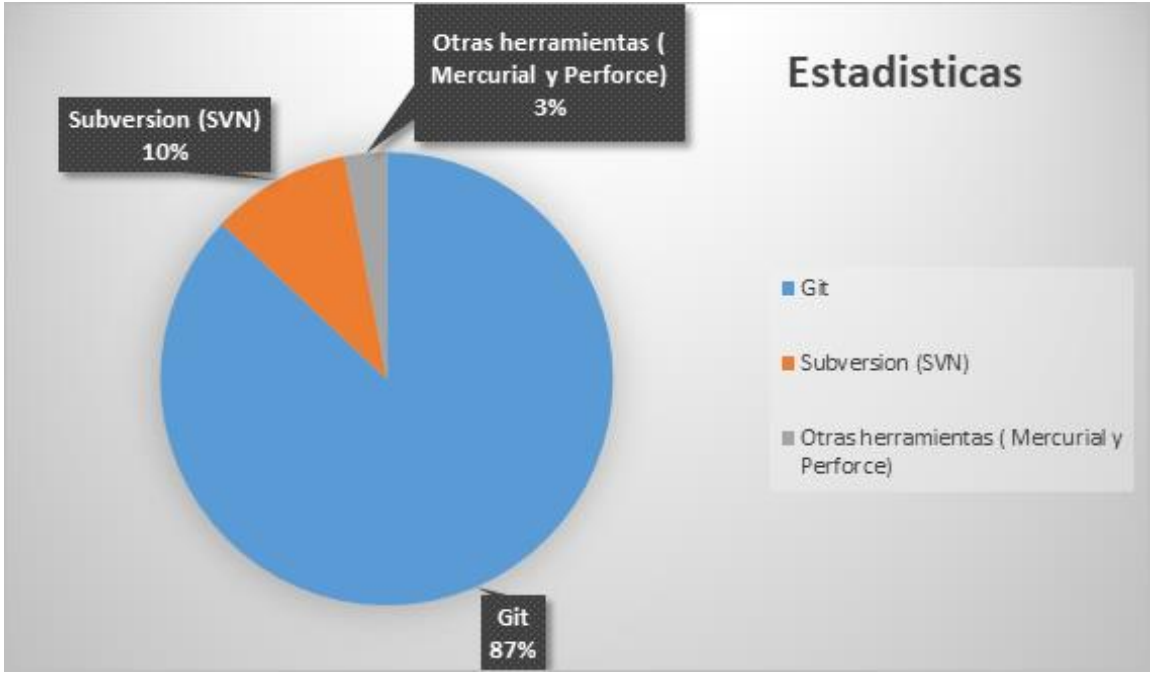

```
braya@STAR MINGW64 ~/Music/StivenValencia
$ git init
Initialized empty Git repository in C:/Users/braya/Music/StivenValencia/.git/

braya@STAR MINGW64 ~/Music/StivenValencia (master)
$ git checkout -b "stivenvalencia"
Switched to a new branch 'stivenvalencia'

braya@STAR MINGW64 ~/Music/StivenValencia (stivenvalencia)
$ AC

braya@STAR MINGW64 ~/Music/StivenValencia (stivenvalencia)
$ git remote add origin https://github.com/Jhonatan-Fernando84/Mipagina2.git

braya@STAR MINGW64 ~/Music/StivenValencia (stivenvalencia)
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/Jhonatan-Fernando84/Mipagina2.git
  Push  URL: https://github.com/Jhonatan-Fernando84/Mipagina2.git
  HEAD branch: main
  Remote branches:
    main           new (next fetch will store in remotes/origin)
    master         new (next fetch will store in remotes/origin)
    stivenvalencia new (next fetch will store in remotes/origin)

braya@STAR MINGW64 ~/Music/StivenValencia (stivenvalencia)
$ git pull origin stivenvalencia
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 22 (delta 0), reused 22 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (22/22), 181.51 KiB | 963.00 KiB/s, done.
From https://github.com/Jhonatan-Fernando84/Mipagina2
 * branch            stivenvalencia -> FETCH_HEAD
 * [new branch]      stivenvalencia -> origin/stivenvalencia

braya@STAR MINGW64 ~/Music/StivenValencia (stivenvalencia)
$ git push -u origin stivenvalencia
Everything up-to-date
branch 'stivenvalencia' set up to track 'origin/stivenvalencia'.

braya@STAR MINGW64 ~/Music/StivenValencia (stivenvalencia)
$ git commit -m "Cambios realizados prueba"
On branch stivenvalencia
Your branch is up to date with 'origin/stivenvalencia'.

nothing to commit, working tree clean

braya@STAR MINGW64 ~/Music/StivenValencia (stivenvalencia)
$ AC

braya@STAR MINGW64 ~/Music/StivenValencia (stivenvalencia)
$ git push -u origin stivenvalencia
Everything up-to-date
branch 'stivenvalencia' set up to track 'origin/stivenvalencia'.

braya@STAR MINGW64 ~/Music/StivenValencia (stivenvalencia)
$
```

Bonus:

- Es común cometer errores al subir cambios a la rama main en GitHub. Git ofrece varias herramientas para revertir estos cambios y mantener tu historial limpio.
- Para estos casos tenemos los siguientes comandos
- git revert: Ideal para deshacer cambios específicos y mantener un historial limpio.
- git reset: Útil para retroceder varias confirmaciones o eliminar cambios no deseados.
- git checkout: Sirve para cambiar de rama o restaurar archivos individuales

Miraremos las estadísticas más relevantes de las herramientas de Gestión de Configuración de Software más importantes.

Popularidad de las Herramientas de SCM (2023):

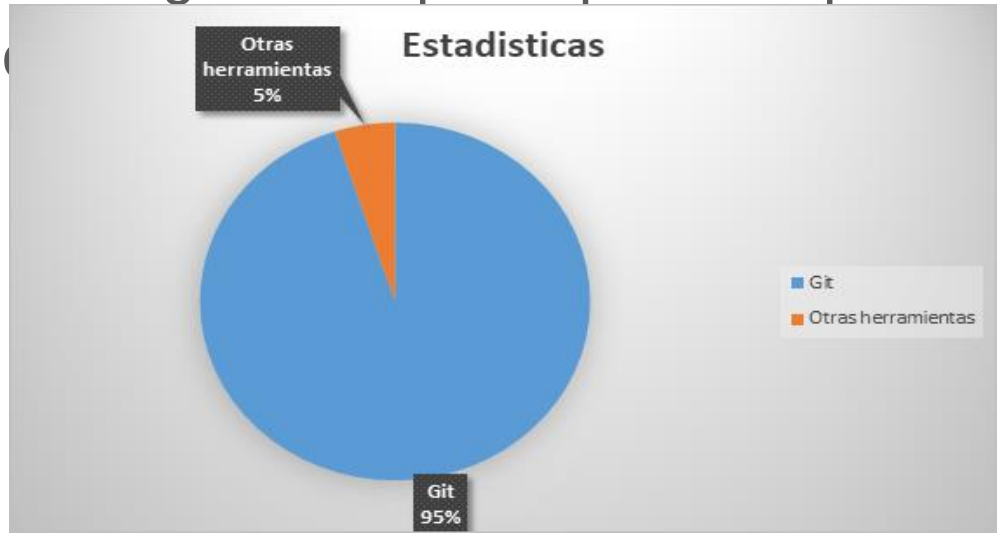


UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Ahorro de Tiempo:

Preferencias por Sectores:

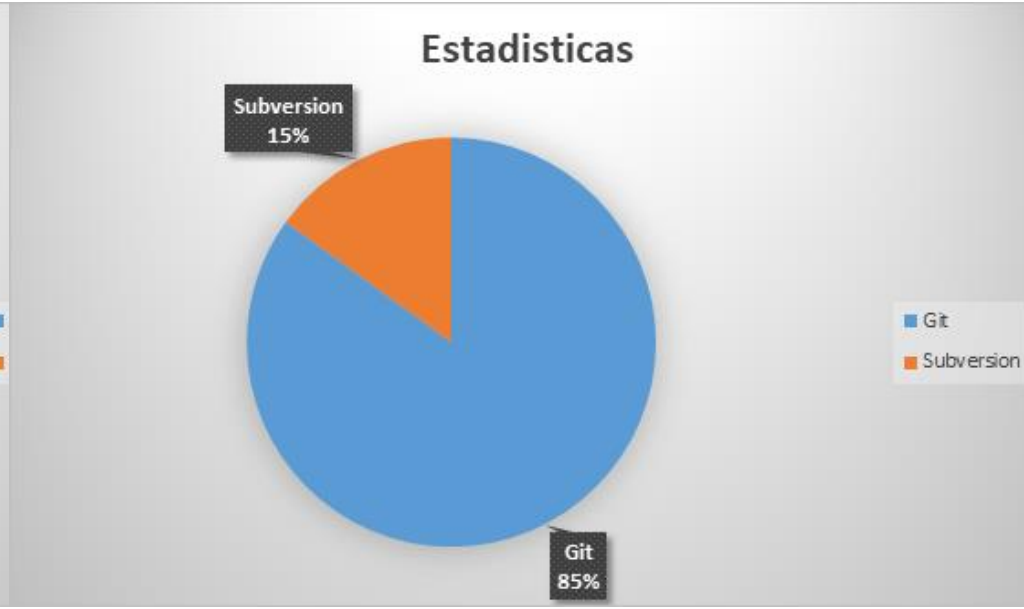
Este gráfico aplica para empresas que estas empezando y



Distribución Geográfica del Uso de SCM:

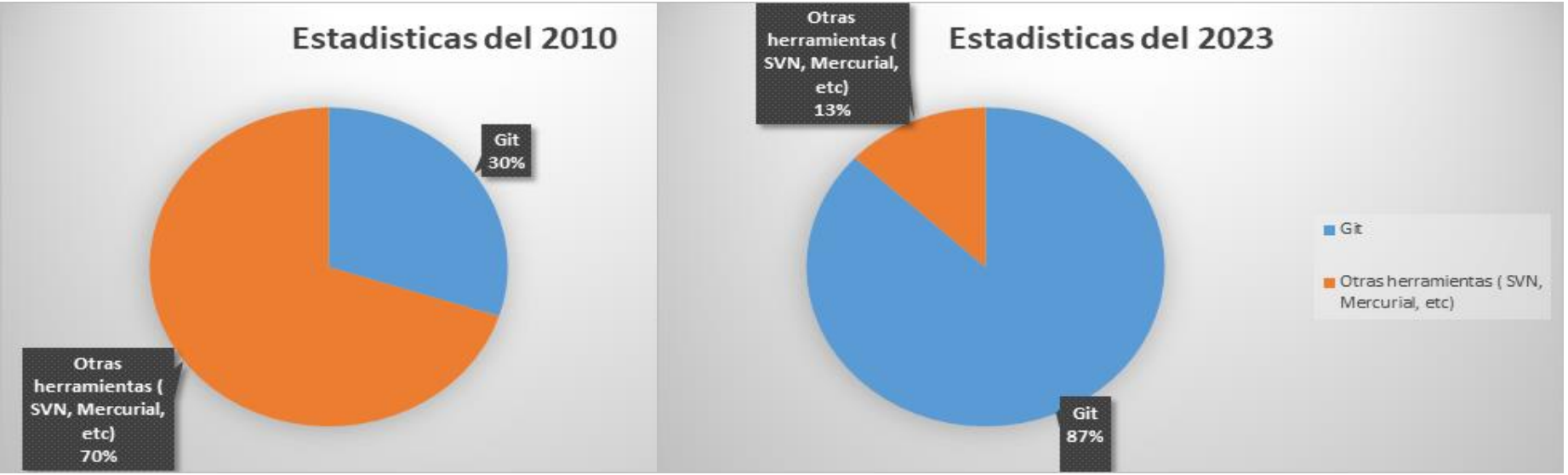
América del Norte y Europa

Asia y Latinoamérica



UNIREMINGTON
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Crecimiento del Uso de Git (2010-2023)



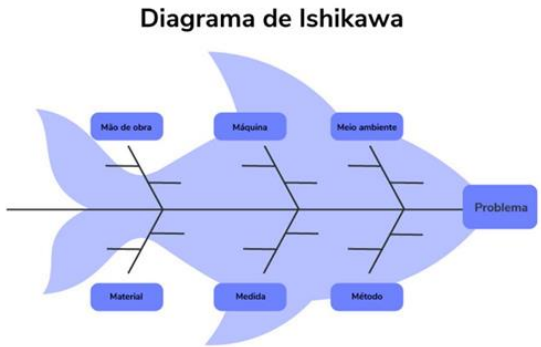
UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Importancia de la Gestión de Riesgos

La gestión de riesgos se ha vuelto crucial para las empresas, ya que previene pérdidas significativas y ayuda a evitar problemas futuros en los proyectos.

Técnicas para Identificación de Riesgos

El Análisis de Causa-Raíz usa diagramas Ishikawa para identificar causas; la Técnica Delphi recopila opiniones de expertos, y el análisis DOFA evalúa debilidades y oportunidades.



Técnicas de Identificación de Riesgos (Continuación)

Las listas de verificación ayudan a identificar riesgos conocidos, mientras que el brainstorming fomenta la colaboración del equipo para generar ideas sobre riesgos y soluciones.

UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Estándares de Calidad en Software

Los estándares ofrecen un marco para medir la calidad del software.

Se clasifican en estándares que miden el producto, el proceso y específicos para TI.

Estándar ISO 25000

Enfocado en evaluación y mejora de la calidad del software, incluyendo mantenibilidad y usabilidad.

Evalúa características como adecuación funcional, fiabilidad, usabilidad, eficiencia, y más.

UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Estándar ISO 9126

Permite medir la calidad del software a través de siete características claves.

Proporciona métricas para cuantificar la calidad del software y asegurar la satisfacción del cliente.

UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Conclusión

La gestión de riesgos y estándares de calidad son cruciales para el éxito en proyectos de software, previniendo problemas y garantizando un producto final de calidad.

UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

¿Que es la estimación de Proyectos?

La estimación de proyectos se refiere al proceso de estimar el tiempo, los costos y los recursos necesarios para completar un proyecto

De esa manera, se puede estar seguro de que se tiene todo lo necesario para ejecutar y completar el proyecto con éxito, desde los recursos hasta el presupuesto correcto.

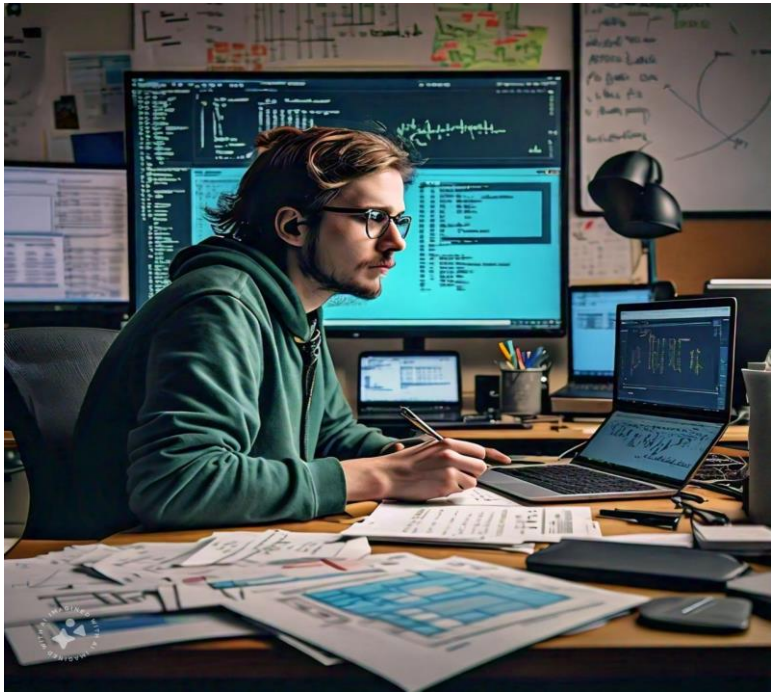
Importancia de Estimar en un Proyecto

UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Planificación

La programación claro en proyectos depende de previsiones detalladas de recursos y tiempo para crear un horario efectivo y evitar malos entendidos .

Las estimaciones en proyectos ayudan a planificar operaciones, asignar recursos, identificar interdependencias y evitar excesos de costes e interrupciones, protegiendo la reputación.



Valentin Ducuara Leguizamon

Control de Recursos

El control de recursos garantiza disponibilidad y uso eficiente de empleados, materiales y financiamiento, evitando retrasos y costos innecesarios mediante estimaciones precisas.

El control efectivo de recursos permite planificar asignación, equilibrar carga de trabajo, evitar agotamiento y ajustar en tiempo real para cumplir objetivos y presupuesto.

1. Vesperman, J. (2006). Essential CVS (2nd ed.). O'Reilly Media.

UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Expectativas de Stakeholders

La gestión de expectativas de stakeholders requiere transparencia y comunicación clara, basada en estimaciones precisas, para evitar malentendidos y cumplir con sus intereses.

Estimaciones claras permiten decisiones informadas, alinean prioridades, construyen confianza, previenen frustración y gestionan expectativas, fortaleciendo relaciones entre stakeholders y partes involucradas.

2. Sink, E. (2011). Version Control by Example. Pyrenean Gold Press.

UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Identificación de Riesgos

La identificación de riesgos utiliza estimaciones precisas para anticipar problemas, detectar áreas de incertidumbre y desarrollar planes de mitigación proactivos en proyectos.

Estimaciones precisas detectan riesgos temprano, permitiendo decisiones preventivas, como capacitación o proveedores alternativos, fortaleciendo la resiliencia y asegurando la entrega exitosa del proyecto.

3. Swicegood, T. (2008). Pragmatic Version Control Using Git. Pragmatic Bookshelf.

Toma de Decisiones

La toma de decisiones en proyectos se basa en información precisa y oportuna, incluyendo estimaciones, para evaluar alternativas y ajustar planes ante cambios imprevistos.

Estimaciones precisas permiten evaluar impacto de decisiones, priorizar tareas, equilibrar calidad, costo y tiempo, y minimizar riesgos de sobre compromisos en proyectos.

4. Somasundaram, R. (2013). Git: Version Control for Everyone. Packt Publishing

UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Evaluación de Viabilidad

La evaluación de viabilidad se basa en estimaciones precisas para determinar factibilidad financiera, técnica y operativa, comparando beneficios esperados con costos y cronograma.

La evaluación de viabilidad analiza disponibilidad de recursos, costos y tiempo, para mitigar riesgos financieros y asegurar sostenibilidad a largo plazo, guiando decisiones informadas.

1. Vesperman, J. (2006). Essential CVS (2nd ed.). O'Reilly Media.

UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Mejora Continua

La estimación en proyectos facilita la mejora continua al comparar desempeño real con objetivos, identificando áreas de mejora y aprendizaje de deficiencias pasadas.

El análisis de estimaciones pasadas identifica errores, afinan técnicas y extrae lecciones aprendidas para mejorar la eficiencia y efectividad en proyectos futuros y fomentar cultura de aprendizaje.

2. Sink, E. (2011). Version Control by Example. Pyrenean Gold Press.

Técnicas que se implementen en la actualidad para realizar estimación de proyectos.

Los objetivos debe ser conciso, que fomenta el compromiso en el equipo para que el proyecto sea exitoso. Debe tener una comunicación efectiva se debe asegurar que todos comprendan qué y por qué se espera alcanzar.

Objetivos claros unen al equipo hacia un fin común. Comunicación efectiva define expectativas y propósito, impulsando compromiso y alineación en la organización.

2. Sink, E. (2011). Version Control by Example. Pyrenean Gold Press.

UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

1. Claridad en los Objetivos

La claridad en los objetivos es fundamental para el éxito de cualquier proyecto. Cuando todos los miembros del equipo comprenden los objetivos de manera precisa, se fomenta la alineación y el compromiso hacia un fin común.

Objetivos claros unen al equipo hacia un fin común. Comunicación efectiva define expectativas y propósito, impulsando compromiso y alineación en la organización.

Una falta de claridad puede dar lugar a confusión, errores y esfuerzos desperdigados que alejan al equipo de sus metas.

Jeffrey A. Hoffer, Joey F. George and Joseph S. Valacich (2018)

2. Facilitación del Trabajo en Equipo

La facilitación del trabajo en equipo es una de las dimensiones más críticas en la gestión de proyectos, y la comunicación efectiva juega un papel esencial en su éxito.

Las herramientas de comunicación, ya sean reuniones, correos electrónicos o plataformas colaborativas, permiten que los integrantes del equipo compartan ideas y progresos, lo cual fomenta la confianza mutua y la colaboración.

1. Vesperman, J. (2006). Essential CVS (2nd ed.). O'Reilly Media.

3. Gestión de Conflictos

UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Es un aspecto esencial en la dinámica de cualquier equipo de trabajo, y una comunicación adecuada es clave para su resolución.

Una comunicación abierta y honesta permite a los miembros del equipo expresar sus diferencias de manera constructiva, facilitando la búsqueda de soluciones.

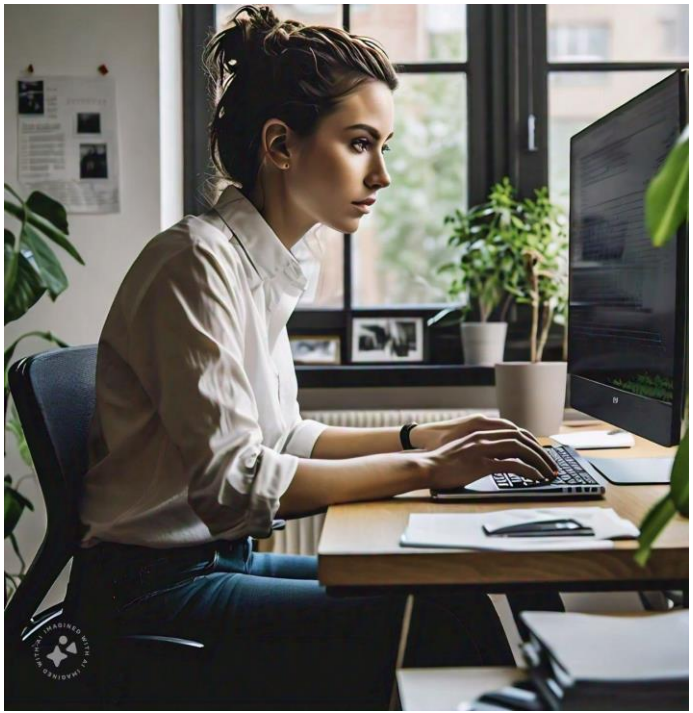
Mediante el uso de técnicas de resolución de conflictos, como la mediación y el compromiso, se pueden transformar situaciones tensas en oportunidades de aprendizaje y mejora.

- Cruz-Alonso, J. A. (17 de Diciembre de 2021).

4. Aumento de la Productividad

en un equipo de proyecto es un objetivo crucial que se ve influenciado directamente por la calidad de la comunicación.

La comunicación también establece expectativas claras sobre plazos y entregables, lo que permite que cada miembro del equipo conozca sus responsabilidades y se comprometa a cumplir con ellas.



- Holcombe, J. (17 de Abril de 2023).

5. Adaptación al Cambio

La adaptación al cambio es un componente inevitable en la gestión de proyectos, especialmente en entornos dinámicos.

La apertura en la comunicación no solo permite que las personas expresen sus preocupaciones o dudas sobre los cambios propuestos, sino que también les da un sentido de propiedad sobre el proceso.

Técnicas (2022). Universidad de la Coruña.Eduardo Mosqueira Rey(Coordinador)

6. Fortalecimiento de Relaciones

El fortalecimiento de relaciones es un factor crucial que impacta directamente en la cohesión y efectividad de un equipo de proyecto.

Una buena comunicación no solo implica informar y compartir actualizaciones, sino también estar dispuesto a escuchar y considerar las perspectivas de los demás.

Jeffrey L.Whitten, Lonnie D.Bentley and Kevin C.Dittman (2021)

7. Retroalimentación Efectiva

La retroalimentación efectiva es importante en el proceso de comunicación de un proyecto, ya que permite la evaluación continua del rendimiento y el avance.

La retroalimentación efectiva en proyectos implica evaluación continua, comunicación constructiva y confianza, mejorando rendimiento, aprendizaje y crecimiento, y fomentando un ambiente colaborativo.

V. Rajaraman; (1974) Analysis and Design of Information

CITAS Y REFERENCIAS BIBLIOGRÁFICAS

- Aldeguer, A. (2024, January 3). *Estimación de Tiempo en Desarrollo de Software: Estrategias Eficientes*. Itequia.

  https://itequia.com/es/estimacion-de-tiempo-en-desarrollo-de-software-estrategias-eficientes/
- orzali, B. (28 de 02 de 2022). *SoftExpert Blog*. Obtenido de SoftExpert Blog: https://blog.softexpert.com/es/identificacion-riesgos/
- Domínguez, I. R. (08 de 03 de 2017). *Calidad del software | MODELOS Y ESTÁNDARES DE UN PRODUCTOl*. Obtenido de Youtube: https://www.youtube.com/watch?v=2lycHFiG-co
- EALDE. (10 de 11 de 2020). *EALDE*. Obtenido de EALDE: https://cursos.ealde.es/blogs/gestion-de-riesgos-direccion-de-proyectos-online/tecnicas-y-herramientas-para-la-identificacion-de-riesgos-en-proyectos#:~:text=Algunos%20ejemplos%20de%20t%C3%A9cnicas%20de%20recopilaci%C3%B3n%20de%20informaci%C3%B3n,de%20deb
- GRUPO DIGITAL 360°. (13 de 12 de 2018). *ISO 9126*. Obtenido de GRUPO DIGITAL 360°: https://grupodigital360.com/iso-9126-metricas-para-calidad-web/
- labitstudio. (s.f.). *labitstudio*. Obtenido de labitstudio: https://labitstudio.com/diferencias-sistemas-control-de-versiones/
- Medina, I. F. (05 de 12 de 2022). *Los estándares de calidad del software más importantes*. Obtenido de hiberus blog: https://www.hiberus.com/crecemos-contigo/los-estandares-de-calidad-del-software-mas-importantes/
- SonarQube. (s.f.). *Documentación de SonarQube 10.6*. Obtenido de SonarQube: https://docs.sonarsource.com/sonarqube/latest/
- Torres, O. P. (14 de 10 de 2022). *pirani*. Obtenido de pirani: https://www.piranirisk.com/es/blog/gestion-de-riesgos-proyectos-de-software

CITAS Y REFERENCIAS BIBLIOGRÁFICAS

- Vesperman, J. (2006). Essential CVS (2nd ed.). O'Reilly Media.
- Sink, E. (2011). Version Control by Example. Pyrenean Gold Press.
- Swicegood, T. (2008). Pragmatic Version Control Using Git. Pragmatic Bookshelf.
- Somasundaram, R. (2013). Git: Version Control for Everyone. Packt Publishing.
- Silverman, M. (2013). Git Pocket Guide: A Working Introduction. O'Reilly Media.
- Pipinellis, A. (2015). GitHub Essentials. Packt Publishing