INVERTIGACION

CARLOS ARCENIO ROCHA BARRIOS



UNIVERSIDAD DE LA COSTA CUC

**INGENIERIA DE SISTEMAS**

**BARRANQUILLA**

**2017**

INVESTIGACION

CARLOS ARCENIO ROCHA BARRIOS

**TRABAJO INVESTIGATIVO**

**ING. ROBERTO MORALES ORTEGA**



UNIVERSIDAD DE LA COSTA CUC

**FACULTAD DE INGENIERIA**

**INGENIERIA DE SISTEMAS**

**BARRANQUILLA**

**CONTENIDO**

**Pag**

[**1.** INTRODUCCION 4](#_Toc304442591)

[**2.** ¿Qué ES GIT? 5](#_Toc304442592)

[**3.** HISTORIA DE GIT 5](#_Toc304442593)

[**4.** OTROS TIPOS DE SISTEMAS DE CONTROL DE VERSIONES 6](#_Toc304442596)

[**4.1** ACCUREV 6](#_Toc304442597)

[**4.1.1** VENTAJAS 6](#_Toc304442598)

[**4.1.2** DESVENTAJAS 6](#_Toc304442598)

[**4.1** ACCUREV 6](#_Toc304442597)

[**4.1.1** VENTAJAS 6](#_Toc304442598)

[**4.1.2** DESVENTAJAS 6](#_Toc304442598)

[5. 20 COMANDOS DE GIT 15](#_Toc304442600)

[6. QUE ES GITHUB 16](#_Toc304442604)

[7. LAS 5 MEJORES FUNCIONALIDADES DE GITHUB 17](#_Toc304442608)

[7.1 Titulo 1 17](#_Toc304442609)

[7.1.1 Subtitulo 1 17](#_Toc304442610)

[7.1.1.1 Subtitulo 2 17](#_Toc304442611)

[8. QUE UN BRANCH O RAMA 18](#_Toc304442612)

[9. QUE ES UN PULL REQUEST. 19](#_Toc304442613)

[10. QUE ES UN HOOK 20](#_Toc304442614)

[11. LOS 5 MEJORES PROYECTOS OPENSOURCE ALOJADOS EN GITHUB 21](#_Toc304442615)

[11.1 Anexo 1 21](#_Toc304442616)

[11.2 Anexo 2 21](#_Toc304442617)

# INTRODUCCION

En este trabajo de investigación se mostraran la definición de un sistema de control de versiones muy conocido y usado por muchos desarrolladores en el mundo llamado Git. Hablaremos un poco de la historia de Git y resaltaremos las características, ventajas y desventajas de otros sistemas de control de versiones. Dentro de este trabajo también se podrán encontrar 20 comandos Git y su función.

Encontraremos también la respuesta a ¿Qué es Github? Y sus cinco funcionalidades más relevantes. También tocaremos el tema de ¿Qué es una rama o branch?, para que sirve y como se genera. Encontraremos respuestas a ¿Qué es un pull request?, ¿Qué es un hook?, y finalmente seleccionaremos los 5 mejores proyectos opensource alojados en Github e indicaremos sus funcionalidades, características, imágenes, ventajas y desventajas.

# ¿Qué ES GIT?

Git un software de control de versiones diseñado por Luis Torvalds y fue pensado en la confiabilidad y eficiencia del mantenimiento de versiones de aplicaciones cuando tienen un gran número de archivos de código fuente.

Al hablar de control de versiones, se habla de la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo.

# historia de git

Como muchas de las grandes cosas en esta vida, Git comenzó con un poco de destrucción creativa y encendida polémica. El núcleo de Linux es un proyecto de software de código abierto con un alcance bastante grande. Durante la mayor parte del mantenimiento del núcleo de Linux (1991-2002), los cambios en el software se pasaron en forma de parches y archivos. En 2002, el proyecto del núcleo de Linux empezó a usar un DVCS propietario llamado BitKeeper.

En 2005, la relación entre la comunidad que desarrollaba el núcleo de Linux y la compañía que desarrollaba BitKeeper se vino abajo, y la herramienta dejó de ser ofrecida gratuitamente. Esto impulsó a la comunidad de desarrollo de Linux (y en particular a Linus Torvalds, el creador de Linux) a desarrollar su propia herramienta basada en algunas de las lecciones que aprendieron durante el uso de BitKeeper. Algunos de los objetivos del nuevo sistema fueron los siguientes:

* Velocidad
* Diseño sencillo
* Fuerte apoyo al desarrollo no lineal (miles de ramas paralelas)
* Completamente distribuido
* Capaz de manejar grandes proyectos (como el núcleo de Linux) de manera eficiente (velocidad y tamaño de los datos)

Desde su nacimiento en 2005, Git ha evolucionado y madurado para ser fácil de usar y aún conservar estas cualidades iniciales. Es tremendamente rápido, muy eficiente con grandes proyectos, y tiene un increíble sistema de ramificación (branching) para desarrollo no lineal.

# OTROS TIPOS DE SISTEMAS DE CONTROL DE VERSIONES

## Accurev

Es una herramienta para la gestión de la configuración de código fuente que integra un gestor de incidencias basado en flujos que maneja de forma eficiente desarrollo paralelo a escala global. También contempla un servidor para replicación. Algunas ventajas y desventajas:

### **VENTAJAS**

* AccuRev captura y controla las relaciones entre bases de código en entornos paralelos utilizando una arquitectura basada en streams. Esto permite a los equipos almacenar el trabajo de forma seguro y probarlo antes de que sea compartido con otros.
* El proceso de [guardado](https://es.wikipedia.org/wiki/Commit) en AccuRev consta de dos pasos. Los usuarios pueden realizar un check-in privado a su espacio de trabajo (workspace) en una operación denominada keep antes de compartirlo con el resto del grupo.[4](https://es.wikipedia.org/wiki/AccuRev#cite_note-4) Posteriormente pueden promover los cambios al stream superior mediante el comando promote.
* AccuRev se integra con diversos [sistemas de gestión de incidencias](https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_incidencias) y herramientas de [gestión de proyectos](https://es.wikipedia.org/wiki/Software_de_gesti%C3%B3n_de_proyectos).
* AccuRev permite estructuras de streams remotos y replicación para equipos distribuidos. Los servidores de réplica funcionan como una caché local que acelera las lecturas y escrituras mientras mantiene sincronizada la base de código con el repositorio central.

### **DESVENTAJAS**

* Los workspaces de los desarrolladores no reciben automáticamente los cambios de su stream superior, sino que lo hacen bajo petición del usuario.
* Se pueden definir streams estáticos (snapshots) para mantenimiento de versiones ya liberadas que no deban recibir nuevas actualizaciones de otras líneas de desarrollo.
* Es posible impedir la propagación de cambios que supongan no pasar las [pruebas de integración](https://es.wikipedia.org/wiki/Pruebas_de_integraci%C3%B3n).

## [CA SCM](https://en.wikipedia.org/wiki/CA_Software_Change_Manager)

CA SCM es un [software de](https://en.wikipedia.org/wiki/Computer_software)herramienta para la [gestión de la configuración](https://en.wikipedia.org/wiki/Configuration_management) ( [control de revisiones](https://en.wikipedia.org/wiki/Revision_control) , [SCM](https://en.wikipedia.org/wiki/Software_configuration_management) , etc.) de [código fuente](https://en.wikipedia.org/wiki/Source_code) y otros activos de desarrollo de software.

### **VENTAJAS**

# METODOLOGIA EXPERIMENTAL – PUEDE SER UN NOMBRE MAS ALUCIVO A ESTA PARTE

“Toda la parte experimental (***Metodología Experimental, Resultados, Análisis y Discusión***) debe estar escrito bajo normas ICONTEC, en tercera persona y en *máximo* ***30 HOJAS.***  Aquí debe relatarse toda la parte práctica del proyecto, esto es, la parte experimental o la parte desarrollada por el autor. “

Metodología experimental es donde deben explicarse los métodos usados, los equipos, software y procedimientos para el desarrollo de la parte experimental de la tesis.

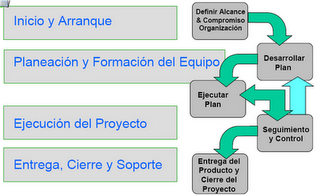
## Titulo 1

### Subtitulo 1.

#### Subtitulo 2.

A continuación se muestra la forma, tipo de letra etc. Para citar las figuras, tablas y ecuaciones. Para marcar la figura, tabla o ecuación usar la ayuda de Word referente a Insertar título de la ventana de referencias.

Figura 1. Ejemplo de figura. (Justificado a la izquierda)



Fuente: <http://seminarioaydsistemas.blogspot.com/2009/10/metodologia-de-desarrollo-de-proyectos.html>

(Los formatos, deben preservarse, para introducir el título de una nueva figura, copiar y pegar el formato propuesto y dar actualización de campos o usar insertar referencias cruzadas)

Tabla 2. Ejemplo de tabla. (Justificado a la izquierda y dejar un espacio)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Semanas** | | | | | | | | | | | |
| **ACTIVIDAD** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| **1** |  |  |  |  |  |  |  |  |  |  |  |  |
| **2** |  |  |  |  |  |  |  |  |  |  |  |  |
| **3** |  |  |  |  |  |  |  |  |  |  |  |  |
| **4** |  |  |  |  |  |  |  |  |  |  |  |  |
| **5** |  |  |  |  |  |  |  |  |  |  |  |  |
| **6** |  |  |  |  |  |  |  |  |  |  |  |  |

Nota: Agregue las filas o columnas necesarias. (Los formatos, deben preservarse, para introducir el título de una nueva figura, copiar y pegar el formato propuesto, el automáticamente se numerará)

# RESULTADOS – PUEDE SER UN NOMBRE MAS ALUCIVO A ESTA PARTE

Los resultados obtenidos de los experimentos realizados, las simulaciones calculadas, los software desarrollados o de la aplicación de los protocolos propuestos.

## Titulo 1

### Subtitulo 1.

#### Subtitulo 2.

# ANALISIS Y DISCUSION DE RESULTADOS – PUEDE SER UN NOMBRE MAS ALUCIVO A ESTA PARTE

Donde debe discutirse los resultados obtenidos respecto al marco teórico, la metodología, equipos usados, cálculos teóricos realizados, resultados iniciales esperados etc. Aquí se da el nuevo aporte o ganancia al realizar este desarrollo, todo enlazado coherente y claramente.

A tener en cuenta

* Es quizá la sección más importante del libro.
* La discusión tiene que tener como base los resultados obtenidos y no la bibliografía consultada, esta última sirve como base de apoyo, comparación o negación.
* Generalmente se hace una comparación con los resultados de la bibliografía y los obtenidos

## Titulo 1

### Subtitulo 1.

#### Subtitulo 2.

# CONCLUSIONES

Aquí se plasman las conclusiones que fueron obtenidas del desarrollo práctico del proyecto de grado.

Que hacer:

* Presentar conclusiones generales y específicas, relacionadas a los objetivos.
* Indicar usos, extensiones y limitaciones, si es apropiado.
* Determinar de forma exacta el aporte del desarrollo del trabajo en concordancia a la justificación presentada.

Que **NO** hacer:

* Hacer una lista de afirmaciones obvias de sus resultados.
* Hacer juzgamientos sobre impacto si no era un objetivo estudiado durante el desarrollo.
* Usar palabras imprecisas como “podría”, “debe”, “probablemente” en relación a los resultados.

# RECOMENDACIONES Y TRABAJOS FUTUROS

Son observaciones de la parte práctica sobre aspectos por mejorar si alguien decide en un futuro realizar un avance, comparación, rectificación o rebatimiento al tema estudiado, así como una propuesta de los posibles trabajos futuros que complementarían o profundizarían este desarrollo.

# REFERENCIAS BIBLIOGRÁFICAS

Según normas ICONTEC. Los títulos de libros, seminarios, congresos, artículos etc. que son la base para el desarrollo del proyecto de grado. **No se admiten referencias** de portales de internet que no brinden ningún respaldo p.e. [www.wikipedia.org](http://www.wikipedia.org), [www.elrincondelvago.com](http://www.elrincondelvago.com), [www.monografias.com](http://www.monografias.com) o similares. En la medida de lo posible, las referencias bibliográficas no deben exceder de 10 años atrás.

# ANEXOS

Toda la información que sirva de sustento a los cálculos, diseños o desarrollo de la parte práctica del proyecto así como las normas aplicadas o un resumen de ellas como metodología si fuere el caso.

## Anexo 1

## Anexo 2