**INVESTIGACIÓN 1**

**Carné 22007685 – María Fernanda Morales Álvarez**

**Fernanda.morales@galileo.edu**

**Ciencia de Datos en Python**

**Inteligencia de Negocios**

**Universidad Galileo**

**Qué es Git:**

Git es un sistema de control de versiones gratuito y de código abierto, creado originalmente por Linus Torvalds en 2005. A diferencia de los antiguos sistemas centralizados de control de versiones, como SVN y CVS, Git está distribuido: cada desarrollador tiene el historial completo de su repositorio de código de manera local. De este modo, la clonación inicial del repositorio es más lenta, pero las operaciones posteriores, como commit, blame, diff, merge y log son mucho más rápidas.

Git incluye las funcionalidades de crear ramas y fusiones y reescribir historiales de repositorios, lo cual ha dado como resultado muchas herramientas y flujos de trabajo innovadores y eficaces. Las solicitudes de incorporación de cambios son una herramienta popular con la que los equipos pueden colaborar en las ramas de Git y revisar con eficacia el código de los demás. Git es el sistema de control de versiones más utilizado en el mundo hoy en día, y se considera el modelo actual de desarrollo de software.

Git es un Software cuyo propósito principal es llevar un registro de los cambios realizados en archivos de la computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

Utilizando este software se puede llevar un registro de los cambios de los archivos de un proyecto en el que se está trabajando, independientemente el lenguaje y también nos permitirá trabajar junto a otras personas que quieran sumarse a nuestro proyecto. O nos permitirá sumarnos a trabajar en los proyectos de otros programadores de una manera muy sencilla.

Este software trabaja de dos maneras, en local y en remoto. Pero para comprender mejor esto debemos saber que es un repositorio. Puesto que lo Git hace en realidad es crearnos y administrar un repositorio local para trabajar o no en un repositorio remoto.

**Básicamente, Git funciona del siguiente modo:**

1. Crea un "repositorio" (proyecto) con una herramienta de alojamiento de Git (por ejemplo, Bitbucket)
2. Copia (o clona) el repositorio a tu máquina local
3. Añade un archivo a tu repositorio local y confirma ("commit") los cambios
4. Envía ("push") los cambios a la rama principal
5. Haz cambios en tu archivo con una herramienta de alojamiento de Git y confírmalos
6. Extrae ("pull") los cambios a tu máquina local
7. Crea una rama ("branch", versión), haz algún cambio y confírmalo
8. Abre una solicitud de incorporación de cambios ("pull request": propón cambios a la rama principal)
9. Fusiona ("merge") tu rama con la rama principal

**Diagrama del funcionamiento de Git**

Diagrama

Descripción generada automáticamente

**Qué es GitHub:**

Es un portal creado para alojar el código de las aplicaciones de cualquier desarrollador, y que fue comprada por Microsoft en junio del 2018. La plataforma está creada para que los desarrolladores suban el código de sus aplicaciones y herramientas, y que como usuario no sólo puedas descargarte la aplicación, sino también entrar a su perfil para leer sobre ella o colaborar con su desarrollo.

Se trata de una de las principales plataformas para crear proyectos abiertos de herramientas y aplicaciones, y se caracteriza sobre todo por sus funciones colaborativas que ayudan a que todos puedan aportar su granito de arena para mejorar el código.

Como buen repositorio, el código de los proyectos que sean abiertos puede ser descargado y revisado por cualquier usuario, lo que ayuda a mejorar el producto y crear ramificaciones a partir de él. Y si prefieres que tu código no se vea, también pueden crearse proyectos privados.

Como su nombre indica, la web utiliza el sistema de control de versiones Git diseñado por Linus Torvalds. Un sistema de gestión de versiones es ese con el que los desarrolladores pueden administrar su proyecto, ordenando el código de cada una de las nuevas versiones que sacan de sus aplicaciones para evitar confusiones. Así, al tener copias de cada una de las versiones de su aplicación, no se perderán los estados anteriores cuando se va a actualizar.

Así pues, Git es uno de estos sistemas de control, que permite comparar el código de un archivo para ver las diferencias entre las versiones, restaurar versiones antiguas si algo sale mal, y fusionar los cambios de distintas versiones. También permite trabajar con distintas ramas de un proyecto, como la de desarrollo para meter nuevas funciones al programa o la de producción para depurar los bugs.

Las principales características de la plataforma es que ofrece las mejores características de este tipo de servicios sin perder la simplicidad, y es una de las más utilizadas del mundo por los desarrolladores. Es multiplataforma, y tiene multitud de interfaces de usuario.

Así pues, Github es un portal para gestionar las aplicaciones que utilizan el sistema Git. Además de permitirte mirar el código y descargarte las diferentes versiones de una aplicación, la plataforma también hace las veces de red social conectando desarrolladores con usuarios para que estos puedan colaborar mejorando la aplicación.

Github permite que los desarrolladores alojen proyectos creando repositorios de forma gratuita. Pero hay que tener una cosa en mente, y es que para poder subir gratis los proyectos deberán ser de código abierto. Y no quieres que tu aplicación sea de código abierto, la plataforma también tiene una versión de pago para alojar proyectos de forma privada.

Github también ofrece una serie de herramientas propias con las que complementar las ventajas que ya tiene el sistema Git de por sí solo. Por ejemplo, puedes crear una Wiki para cada proyecto, de forma que puedas ofrecer toda la información sobre él y anotar todos los cambios de las diferentes versiones.

También tiene un sistema de seguimiento de problemas, para que otras personas puedan hacer mejoras, sugerencias y optimizaciones en los proyectos. Ofrece también una herramienta de revisión de código, de forma que no sólo se pueda mirar el código fuente de una herramienta, sino que también se pueden dejar anotaciones para que su creador o tú mismo después si es tu proyecto las podáis revisar. Se pueden crear discusiones también alrededor de estas anotaciones para mejorar y optimizar el código.

A su vez, se pueden encontrar gráficos para ver cómo trabajan los desarrolladores en sus proyectos y bifurcaciones del proyecto, viendo las actualizaciones realizadas a partir de la primera versión o los cambios que se han realizado. Y por último también se incluyen características de redes sociales, como un sistema para seguir a tus creadores favoritos y ni perderte sus actualizaciones.

**Markdown y sus comandos**

Markdown es una herramienta de conversión de texto a HTML para escritores web. Markdown le permite escribir utilizando un formato de texto sin formato fácil de leer y escribir, y luego convertirlo a XHTML estructuralmente válido (o HTML).

Por lo tanto, "Markdown" es dos cosas: (1) una sintaxis de formato de texto sin formato; y (2) una herramienta de software, escrita en Perl, que convierte el formato de texto sin formato a HTML. Consulte la página Sintaxis para obtener detalles relacionados con la sintaxis de formato de Markdown. Puedes probarlo, ahora mismo, usando el Dingus en línea .

El objetivo principal del diseño de la sintaxis de formato de Markdown es hacer que sea lo más legible posible. La idea es que un documento con formato de Markdown se pueda publicar tal cual, como texto sin formato, sin que parezca que ha sido marcado con etiquetas o instrucciones de formato. Si bien la sintaxis de Markdown se ha visto influenciada por varios filtros de texto a HTML existentes, la mayor fuente de inspiración para la sintaxis de Markdown es el formato de correo electrónico de texto sin formato.

La sintaxis es muy sencilla y cuenta con varias opciones diferentes para algunos de sus elementos. Básicamente, se trata de añadir ciertos caracteres al inicio de la línea o antes y después de los elementos a los que vamos a aplicar el formato. Veamos a continuación tan sólo algunos ejemplos para hacernos una idea de su sencillez de uso.

Como hemos visto en el ejemplo anterior, los encabezados se crean poniendo almohadillas. El número de almohadillas que pongamos se corresponderá con el nivel de encabezado que queremos usar. Por ejemplo, si queremos un encabezado de nivel 2 (h2), pondremos dos almohadillas; si queremos uno de nivel 4 (h4) pondremos cuatro almohadillas. Así, del 1 al 6, que son los encabezados que tiene el HTML.

Para enfatizar, usamos los asteriscos antes y después de las palabras. Dado que hay dos formas de enfatizar, cursivas y negritas, usaremos un asterisco antes y después para las cursivas y dos asteriscos para las negritas, ambos sin espacios.

**REFERENCIA BIBLIOGRÁFICA**

<https://www.atlassian.com/es/git#:~:text=B%C3%A1sicamente%2C%20Git%20funciona%20del%20siguiente,(%22commit%22)%20los%20cambios>

<https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores>

<https://pythones.net/git-tutorial/#%C2%BFQue_es_Git>

<https://www.google.com/search?q=que+es+Markdown+y+sus+comandos&ei=9V7gY8WLHa6wqtsPhveVyA4&ved=0ahUKEwiF2tSH5v_8AhUumGoFHYZ7BekQ4dUDCA8&uact=5&oq=que+es+Markdown+y+sus+comandos&gs_lcp=Cgxnd3Mtd2l6LXNlcnAQAzIFCAAQogQyBQgAEKIEOgoIABBHENYEELADSgQIQRgASgQIRhgAULQLWNESYO4aaAFwAHgAgAGpBogByh6SAQkyLTEuNS0xLjSYAQCgAQHIAQjAAQE&sclient=gws-wiz-serp#kpvalbx=_E2rgY4qAIoas5NoPxY6skAk_5>

<https://daringfireball.net/projects/markdown/>