Ricardo Perez Perez

Documento de investigacion

Programacion 2 - Tarea # 1

# Lenguage C

-Es un lenguaje orientado a la implementación de Sistemas Operativos,

-Dispone de las estructuras típicas de los lenguajes de alto nivel pero, a su vez,

dispone de construcciones del lenguaje que permiten un control a muy bajo nivel.

-Los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclar código en ensamblador

con código C.

-Acceder directamente a memoria o dispositivos periféricos.

# Lenguage C++

-Permitan la manipulación de objetos.

-C++ es un lenguaje de programación multiparadigma.

-C++ es la posibilidad de redefinir los operadores (sobrecarga de operadores).

-Hay ciertas palabras clave que se utilizan de modo distinto, como extern y

static.

# C#

-Soporte XML para interacción basada en componentes.

-Acceso a librerías a través de COM+ y los servicios .NET.

-Seguridad a nivel de código basada en mecanismos de confianza.

-Soporte total a los metadatos y a sus posibles extensiones.

-Soporte total a la plataforma COM para permitir la integración con código ya

existente.

# Herramienta Git

Git, es un software de control de versiones diseñado por Linus Torvalds. La pregunta es ¿qué es control de versiones? Pues bien, se define como control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo es decir a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración,y para los que aún no les queda claro del todo, control de versiones es lo que se hace al momento de estar desarrollando un software o una página web. Exactamente es eso que haces cuando subes y actualizas tu código en la nube, o le añades alguna parte o simplemente le editas cosas que no funcionan como deberían o al menos no como tú esperarías.

Algunas de las características más importantes de Git son:

Rapidez en la gestión de ramas, debido a que Git nos dice que un cambio será fusionado mucho más frecuentemente de lo que se escribe originalmente.

Gestión distribuida; Los cambios se importan como ramas adicionales y pueden ser fusionados de la misma manera como se hace en la rama local.

Gestión eficiente de proyectos grandes.

Realmacenamiento periódico en paquetes

## Ordenes básicas

Iniciar un repositorio vacío en unas carpeta específica.

**git init**

Añadir un archivo especifico.

**git add “nombre\_de\_archivo”**

Añadir todos los archivos del directorio

**git add .**

Confirmar los cambios realizados. El “mensaje” generalmente se usa para asociar al commit una breve descripción de los cambios realizados.

**git commit –am “mensaje”**

Revertir el commit identificado por "hash\_commit"

**git revert “hash\_commit"**

Subir la rama(branch) “nombre\_rama” al servidor remoto.

**git push origin “nombre rama”**

Mostrar el estado actual de la rama(branch), como los cambios que hay sin hacer commit.

**git status**

# Github

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git.

## ¿Para que sirve?

GitHub aloja tu repositorio de código y te brinda herramientas muy útiles para el trabajo en equipo, dentro de un proyecto.Además de eso, puedes contribuir a mejorar el software de los demás. Para poder alcanzar esta meta, GitHub provee de funcionalidades para hacer un fork y solicitar pulls.Realizar un fork es simplemente clonar un repositorio ajeno (genera una copia en tu cuenta), para eliminar algún bug o modificar cosas de él. Una vez realizadas tus modificaciones puedes enviar un pull al dueño del proyecto. Éste podrá analizar los cambios que has realizado fácilmente, y si considera interesante tu contribución, adjuntarlo con el repositorio original.

## ¿Qué herramientas proporciona?

En la actualidad, GitHub es mucho más que un servicio de alojamiento de código. Además de éste, se ofrecen varias herramientas útiles para el trabajo en equipo. Entre ellas, caben destacar: