

CUESTIONARIO TP ALGORITMOS

**grupo:
“Equipo Rocket”**

**INTEGRANTES:
USHIÑA PAILLACHO BRYAN JHOEL, CANTIGO MARTIN RENZO, TORRE
MAXIMILIANO, ROSELLO EZEQUIEL, PROVOST VALENTIN**

1) ¿Qué es un svn?

Apache Subversion (SVN) es una herramienta de control de versiones basada en un repositorio cuyo funcionamiento se asemeja al de un sistema de ficheros. Es un software libre bajo una licencia de tipo Apache/BSD.

Utiliza el concepto de revisión para guardar los cambios producidos en el repositorio. Entre dos revisiones solo guarda el conjunto de modificaciones (delta), optimizando así al máximo el uso de espacio en disco. SVN permite al usuario crear, copiar y borrar carpetas con la misma flexibilidad con la que lo haría si estuviese en su disco duro local. Dada su flexibilidad, es necesaria la aplicación de buenas prácticas para llevar a cabo una correcta gestión de las versiones del software generado.

SVN permite la colaboración remota, lo que permite la posibilidad de que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración

2) ¿Qué es git?

Git es el sistema de control de versiones más popular hasta la fecha.

Un sistema de control de versiones permite que los desarrolladores puedan administrar su proyecto, ordenando el código de cada una de las nuevas versiones que sacan de sus proyectos para evitar imprevistos. En consecuencia, al tener copias de las versiones anteriores de sus proyectos, los estados anteriores del mismo no se van a perder cuando se actualicen.

Entonces, Git permite que los desarrolladores vean la línea de tiempo completa para los cambios, decisiones y progresión de cualquier proyecto en un solo lugar. Desde el momento en el que acceden al historial de un proyecto, el desarrollador tiene todo el contexto que necesitan para entenderlo y comenzar a contribuir.

Los desarrolladores pueden trabajar en cualquier horario y las colaboraciones pueden tomar lugar en cualquier momento, mientras se mantiene la integridad del código principal.

algunas diferencias entre git y svn son las siguientes:

	SVN	Git
Control de versiones	Centralizada	Distribuida
Repositorio	Un repositorio central donde se generan copias de trabajo	Copias locales del repositorio en las que se trabaja directamente
Autorización de acceso	Dependiendo de la ruta de acceso	Para la totalidad del directorio
Seguimiento de cambios	Basado en archivos	Basado en contenido
Historial de cambios	Solo en el repositorio completo, las copias de trabajo incluyen únicamente la versión más reciente	Tanto el repositorio como las copias de trabajo individuales incluyen el historial completo
Conectividad de red	Con cada acceso	Solo necesario para la sincronización

3) ¿Qué es Github?

GitHub es una plataforma donde puedes almacenar, compartir y trabajar junto con otros usuarios para escribir código.

Almacenar tu código en un "repositorio" en GitHub te permite:

- Presentar o compartir el trabajo.
- Seguir y administrar los cambios en el código a lo largo del tiempo.
- Dejar que otros usuarios revisen el código y realicen sugerencias para mejorarlo.
- Colaborar en un proyecto compartido, sin preocuparse de que los cambios afectarán al trabajo de los colaboradores antes de que esté listo para integrarlos.

La posibilidad de trabajar de forma colaborativa, es posible gracias al software de código abierto Git, en el que se basa GitHub.

4) ¿Qué es un valgrind?

Valgrind es un sistema para debuggear y hacer profiling (investigar el comportamiento de un programa utilizando información obtenida durante su ejecución) sobre programas en Linux.

Una de las principales características de Valgrind es poder detectar automáticamente muchos errores de manejo de memoria, (como por ejemplo memory leaks) y el objetivo es lograr programas más estables.

Valgrind funciona con programas escritos en cualquier lenguaje. Al trabajar directamente sobre los binarios, Valgrind puede analizar programas compilados y programados en cualquier lenguaje de programación, sean compilados, pre-compilados o interpretados.

Está pensado para funcionar principalmente con programas escritos en C y C++, ya que estos son los lenguajes que introducen este tipo de bugs con mayor facilidad. Pero puede ser usado, por ejemplo, en programas escritos incluso en varios lenguajes diferentes (C, C++, Java, Perl, Python, assembly, Fortran, y muchos otros).