****

**FACULTAD DE ESTUDIOS EN AMBIENTES VIRTUALES**

**CONSTRUCCION DE SOFTWARE**

**GUÍA 4: JAVA E INTERFACES GRAFICAS**

**AUTORES**

**JENNY ESPERANZA TIBAMOSO ALBA**

**TUTOR**

**COBO CAMPO LUIS ARMANDO**

**BOGOTA 2020**



**TABLA DE CONTENIDO**

[Guía 3 - Programación Orientada por Objetos en Java 3](#_Toc49723439)

[Actividad 1 - Manejo de Clases en Java 3](#_Toc49723440)

[2.1 Diagrama de UML 6](#_Toc49723441)

[Actividad 2 - Manejo de características básicas de la POO con Java (Grupal) 6](#_Toc49723442)

[Actividad 3 - Manejo de características avanzadas de la POO en Java (Grupal) 7](#_Toc49723443)

[1.1 UML: 8](#_Toc49723444)

# Guía 3 JAVA E INTERFACES GRAFICAS

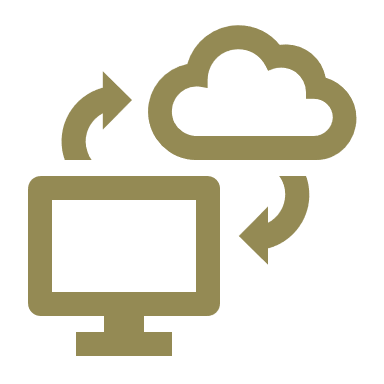
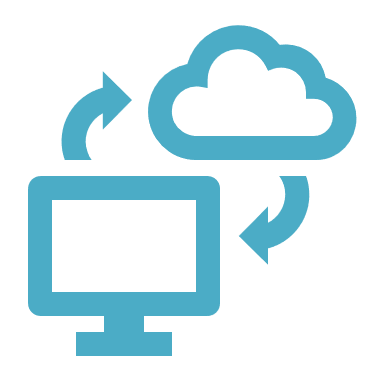
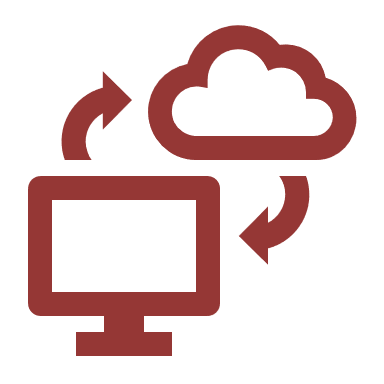
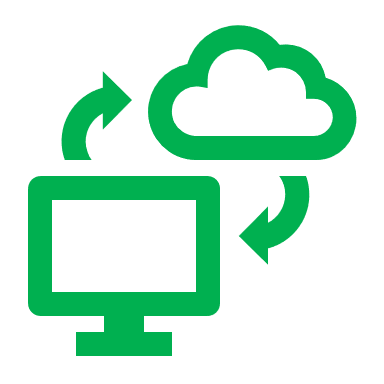
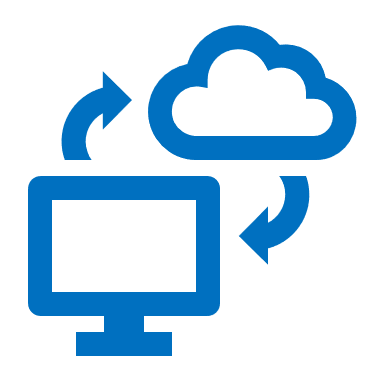
## ACTIVIDAD 1. COMPARACION DE HERRAMIENTAS CON GUTHUB.

* Tabla comparativa entre las herramientas de versionado y trabajo en equipo para la construcción de software (individual).

|  |  |
| --- | --- |
| Software Versionado | Descripción |
| SVN Y CVS | * Los comandos son muy parecidos. La migración de CVS a SVN de bajo costo. * Los commits son atómicos y más eficientes. Si el este sufre una interrupción, no corrompe el repositorio. * CVS estudia diferencias entre archivos basándose en líneas. SVN en cambio lo hace comparando bytes, lo que permite parches de menor tamaño. * SVN trabaja en base a revisiones que involucran todo el proyecto. CVS en cambio, funciona con historiales independientes por cada archivo, o sea, no es fácil identificar una corrección de bug que involucra cambios en distintos archivos, porque cada uno tiene distintos números de revisión * En CVS no se puede renombrar un archivo. Hay que borrarlo, y agregar la copia como un archivo nuevo con otro nombre. Lo peor es que el nuevo archivo no hereda el historial, por lo tanto, se pierde. SVN permite renombrar archivos e incluso cambiarlos de directorio manteniendo todo el historial. |
| SVN Y GIT | * La principal diferencia este estas dos herramientas es que SVN es centralizado y Git es distribuido. * Con SVN no podemos conectarnos al repositorio local es decir no podemos ejecutar comit. * Con Git podemos conectarnos al repositorio local y realizar confirmaciones por medio del comité. * SVN es mas fácil de aprender su manejo. * Git es para los desarrolladores que no están conectados continuamente al repositorio central. * SVN no está bien definido con ramas de trabajo y combinación de estas. * Git es mas rápido en la ejecución y tiene ramas bien definidas y se pueden realizar combinaciones de estas. |
| GIT Y GITHUB | * GitHub es una plataforma que permite tener repositorios locales en la nube para varios desarrolladores puedan trabajar en el mismo proyecto y ver las ediciones de este en tiempo real. * Git es un software local que permite a los desarrolladores guardar instantáneas de sus proyectos a lo largo del tiempo * GitHub permite asignar roles, tareas, permite que su repositorio se ha de nivel público. * Git los desarrolladores no pueden leer ni ver los cambios que se han aplicado de manera local ni en tiempo real. |

• Mapa conceptual o mental con el reconocimiento de las diferentes funcionalidades de la plataforma GitHub (individual).

Git.moodle.org



git-remote + git-fetch

git-fetch /git-pull

Copia de trabajo

+

Repo desarrollador

git-remote + git-fetch

git-fetch /git-pull

git-remote

+ git-pull

Git-push

Git-fetch

Git-pull

Git-clone

Git-push

Git-fetch

Git-pull

Git-clone

Git-push

Git-fetch

Git-pull

Repositorio compartido

@jenny.edu.co

## Actividad 2. Componentes de la GUI y acceso a bases de datos

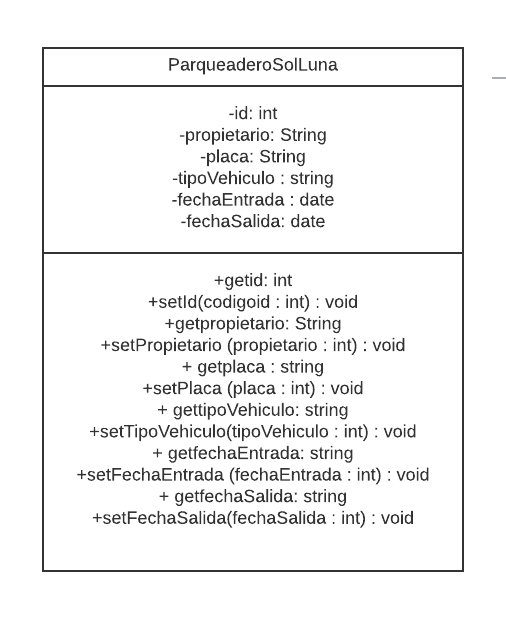
• Programa completo en Java previamente desarrollado en la guía 3, sea publicado en la plataforma GitHub y compartido entre todos los miembros del grupo (grupo).

Actividad 2. Componentes de la GUI y acceso a bases de datos

Programa: Para esta Guía realice la solución al siguiente problema. El parqueadero Sol Luna requiere:

**Registrar** la entrada del vehículo (Nombre del propietario, placa del vehículo, tipo de vehículo), **Modificar y buscar (**Nombre del propietario, placa del vehículo, tipo de vehículo, fecha de ingreso, parqueadero, se debe mostrar la tabla), pagar el tiempo que se encuentre en el parque (placa) se deben guardar los datos en la **base de datos h2.**

### UML



### Enlace de GitHub

# Referencias

García, P. (2019). *nerion.* Obtenido de Subversion vs Git : https://www.nerion.es/blog/subversion-vs-git/

martins, p. ,. (04 de Enero de 2011). *epidataconsulting.* Obtenido de Sistemas de Versionado de Código: http://www.epidataconsulting.com/tikiwiki/tiki-index.php?page=Versionado#Comparaci\_n\_entre\_herramientas\_SVN\_y\_CVS\_