

	UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLD-001	Página: 1

## GUÍA DE LABORATORIO

## GUÍA DE LABORATORIO

(formato docente)

INFORMACIÓN BÁSICA					
ASIGNATURA:	Estructura de Datos y Algoritmos				
TÍTULO DE LA PRÁCTICA:	Revisión de elementos de programación (Parte I)				
NÚMERO DE PRÁCTICA:	01	AÑO LECTIVO:	2023 A	NRO. SEMESTRE:	III
TIPO DE PRÁCTICA:	INDIVIDUAL				
	GRUPAL	X	MÁXIMO DE ESTUDIANTES	4	
FECHA INICIO:	04/05/23	FECHA FIN:	07/05/23	DURACIÓN:	2 horas
<b>RECURSOS A UTILIZAR:</b> <ul style="list-style-type: none"> <li><a href="https://git-scm.com/book/es/v2">https://git-scm.com/book/es/v2</a></li> <li><a href="https://guides.github.com/">https://guides.github.com/</a></li> <li><a href="https://www.w3schools.com/java/default.asp">https://www.w3schools.com/java/default.asp</a></li> </ul>					
<b>DOCENTE(s):</b> Dra. Karim Guevara Puente de la Vega Mg. Richart Smith Escobedo Quispe Mg. Edith Giovanna Cano Mamani					

# Revisión de elementos de programación

## OBJETIVOS TEMAS Y COMPETENCIAS

### OBJETIVOS

- Conocer el editor de texto Vim.
- Aprender a manejar el sistema de control de versiones Git y utilizar GitHub para trabajar de manera colaborativa.
- Realizar ejercicios con el lenguaje de Programación Java.

### TEMAS

- Editor Vim
- Java
- Git
- GitHub

### COMPETENCIAS

## CONTENIDO DE LA GUÍA

### MARCO CONCEPTUAL

- Editor Vim
  - Vim es un editor de texto muy eficiente para programar en consola que no tiene nada que envidiar a los editores gráficos.
- Instalar Vim en:
  - GNU/Linux
    - Para instalar Vim en cualquier distribución GNU/Linux use sus mismos repositorios
    - `apt-get install vim`
  - MS Windows
    - Para descarga en sistemas MS Windows <https://www.vim.org/download.php>
  - MacOS
    - Para instalar Vim en sistemas MacOS puede usar brew

- `brew install macvim`

- Java

- Java es un lenguaje de programación orientado a objetos
- La documentación de Java 11 se encuentra en: <https://docs.oracle.com/javase/specs/jls/se11/html/index.html>
- Un curso muy interesante en videos se puede encontrar en: <https://www.youtube.com/playlist?list=PLw8RQJQ8K1yQDqPyDRzt-h8YIBj96OwMP>
- Instalar Java

- GNU/Linux

- Para descarga en sistemas GNU/Linux puede usar OpenJDK
- `apt-get install openjdk-11-jdk`

- MS Windows

- Para descarga en sistemas MS Windows <https://www.oracle.com/java/technologies/downloads/>

- MacOS

- Para descarga en sistemas MacOS <https://www.oracle.com/java/technologies/downloads/>

- Git

- Git es un sistema de control de versiones creado por Linus Torvalds. Es eficiente, confiable.

- Instalar Git

- GNU/Linux

- Para descarga en sistemas GNU/Linux desde <https://git-scm.com/download/linux>
- `apt-get install git`

- MS Windows

- Para descarga en sistemas MS Windows <https://git-scm.com/download/win>

- MacOS

- Para descarga en sistemas MacOS <https://git-scm.com/download/mac>
- `brew install git`

- `git init`

- Crea un nuevo proyecto local, se crean archivos en el directorio oculto `.git`

- `git init`
- `git config`
  - Establece variables de configuración. Por ejemplo para los commits se necesita los datos del desarrollador. Se puede especificar el editor y hasta el tiempo que deseas almacenar tus credenciales en la cache y otras cosas más
  - `git config --global user.name "Richart Smith Escobedo Quispe"`
  - `git config --global user.email richarteq@gmail.com`
  - `git config --list`
  - `git config user.name`
  - `git config --global core.editor "code --wait"`
  - `git config --global credential.helper 'cache --timeout=3600'`
- `git status`
  - Permite verificar el estado de los archivos
  - `git status`
- `git add`
  - Añade archivos al staging area. El punto "." agrega todos
  - `git add HolaMundo.java`
  - `git add .`
- `git commit`
  - Sube los archivos al área de staging, en la máquina local. La opción -m permite escribir el mensaje en línea
  - `git commit -m "Probando el Hola Mundo"`
- `git clone`
  - Clona un repositorio remoto como un repositorio local, en el cual se puede hacer push
  - `git clone <URL_GITHUB>`
- `git remote`
  - Persigue un repositorio remoto para hacer push.
  - `git remote add origin <URL_GITHUB>`
- `git push`
  - Permite subir archivos al repositorio remoto
  - `git push -u origin main`
- `git show`
  - Muestra detalles del commit actual
  - `git show`
- `git log`
  - Permite ver un resumen de los commit realizados
  - `git log`
  - `git log --pretty=oneline`

- `git log --graph --pretty=oneline --abbrev-commit --all`
- `git log --pretty=format:"%h - %an, %ar : %s"`
- `git log -p -2`
- 6bb6b6e - Richart Escobedo Quispe, hace 3 minutos : Git - GitHub
- 6bb6b6e - Richart Escobedo Quispe, hace 12 minutos : Git - GitHub
- b36a9fd - Richart Escobedo Quispe, hace 14 minutos : Git - GitHub
- e86aac8 - Richart Escobedo Quispe, hace 22 minutos : Git - GitHub
- e58f653 - Richart Escobedo Quispe, hace 31 minutos : first commit
- 5747062 - Richart Escobedo Quispe, hace 35 minutos : first commit
- 
- git diff
  - Permite comparar los cambios en los archivos
  - `git diff 6bb6b6e 6bb6b6e`
- git branch
  - Permite ver las ramas existentes o crea una rama alternativa al proyecto principal
  - `git branch -a`
  - `git branch prueba1`
  - `git branch`
  - \* main
  - prueba1
  -
- git checkout
  - Permite regresar a versiones anteriores o saltar a otra rama
  - `git checkout prueba1`
  - `git branch`
  - main
  - \* prueba1
  -
- git pull
  - Permite descargar los cambios del repositorio remoto al directorio local
  - `git pull`
  - `git checkout nueva_rama`
  - `git pull <remote repo>`
  - `git checkout main`
  - `git pull --rebase origin`

## EJERCICIO RESUELTO POR EL DOCENTE

## Primer repositorio en GitHub

- Creamos un nuevo proyecto en GitHub

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \*



rescobedoq ▾

Repository name \*

proyecto ✓

Great repository names are short and memorable. Need inspiration? How about **legendary-octo-system**?

Description (optional)

Repositorio de ejemplo de un proyecto



**Public**

Anyone on the internet can see this repository. You choose who can commit.



**Private**

You choose who can see and commit to this repository.

**Initialize this repository with:**

Skip this step if you're importing an existing repository.

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

**Add .gitignore**

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾

**Choose a license**

A license tells others what they can and can't do with your code. [Learn more.](#)

License: None ▾

You are creating a public repository in your personal account.

Create repository

- 
- Crearemos un repositorio local usando git init
  - `pwd`
  - `/home/richart/eda/lab01`
  - `git init`
- Crearemos un archivo Readme.md con contenido Markup
  - `echo "# Aquí irá mi informe de laboratorio" > README.md`
- Agregaremos este archivo al staging area usando git add .
  - `git status`
  - En la rama main
  -

- No hay commits todavía
- 
- Archivos sin seguimiento:
- (usa "git add ..." para incluirlo a lo que se será confirmado)
- README.md
- no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa "git add" para hacerles seguimiento)
- 
- `git add README.md`
- Hacemos un primer commit en nuestro repositorio local
  - `git commit -m "Mi primer proyecto en github"`
- Asociamos el repositorio local con el repositorio remoto
  - `git remote add origin <URL_GITHUB>`
- Actualizamos el repositorio remoto
  - `git push -u origin main`
- Ahora podemos verificar en GitHub que nuestro repositorio se actualizó con el proyecto local



- 
- Cree una clase Java HolaMundo.java que imprima un saludo, compílelo, ignore archivos binarios, agregue archivo al staging area, haga commit y súbalo al repositorio GitHub.
  - `vim HolaMundo.java`
  - `public class HolaMundo`
  - `{`
  - `public static void main(String args[]) {`
  - 
  - `/** */`
  - `System.out.println ("¡Hola mundo!");`

- 
- }
- }
- 
- java -version
- openjdk version "11.0.14" 2022-01-18
- 
- javac HolaMundo.java
- java HolaMundo
- ¡Hola mundo!
- 
- vim .gitignore
- \*.class
- .gitignore
- 
- git add HolaMundo.java
- git commit -m "Hola Mundo"
- git remote <URL\_GITHUB>
- git push -u origin main
- Realizaremos un estudio de Java desde: <https://www.w3schools.com/java/default.asp>

## EJERCICIOS PROPUESTOS

- Cree una cuenta de usuario en GitHub usando su correo institucional.
- [opcional por ahora] Configure su cuenta de estudiante (<https://education.github.com/pack>).
- Cree un nuevo proyecto personal y desarrolle el ejercicio resuelto en clase. Haga 3 commits como mínimo y muéstrelos. Commit para "¡Hola mundo!", otro para "Bienvenida al curso" y otro para imprimir su nombre.
- Cree un proyecto grupal para trabajo colaborativo (de 3 a 5 integrantes).
- Cree un archivo por cada tema del manual de java (<https://www.w3schools.com/java/default.asp>), haga commit e incluyalo en su informe grupal (Divídanse los temas).
  - Java Tutorial
  - Java Methods
- Cree ramas para cada integrante y cada cierto tiempo una las ramas al main. No elimine nada para evidenciar ramas, main y commits.

## CUESTIONARIO

- ¿Por qué Git y GitHub son herramientas importantes para el curso?



- ¿Qué conductas éticas deberían promocionarse cuando se usa un Sistema de Control de Versiones?
- ¿Qué son los estándares de codificación?

## REFERENCIAS

- <https://git-scm.com/book/es/v2>
- <https://guides.github.com/>
- <https://www.w3schools.com/java/default.asp>
- Nano vs. Emacs vs. Vim (Editor Wars!) - <https://www.linuxtrainingacademy.com/nano-emacs-vim/>
- Emacs vs Vim - [https://www.linuxteaching.com/article/emacs\\_vs\\_vim](https://www.linuxteaching.com/article/emacs_vs_vim)
- Vim or Emacs? The Debate is over... - <https://cmd.com/blog/vim-or-emacs-the-debate-is-over/>
- Por qué un editor de texto de hace 40 años machaca al "todopoderoso" Atom - <https://www.xataka.com/aplicaciones/por-que-un-editor-de-texto-de-hace-40-anos-le-da-sopas-con-ondas-al-todopoderoso-atom>
- Tutorial de Vim - <http://www.truth.sk/vim/vimbook-OPL.pdf>
- Teclado en Vim - <http://www.viemu.com/vi-vim-cheat-sheet.gif>
- Cómo Configurar VIM como VS Code - <https://www.youtube.com/watch?v=XgQFzi3Vkc8>
- Empezando el Curso de Java - <https://www.youtube.com/playlist?list=PLw8RQJQ8K1yQDqPyDRzt-h8YIBj96OwMP>
- The Java® Language Specification Java SE 11 Edition - <https://docs.oracle.com/javase/specs/jls/se11/html/index.html>
- The Java™ Tutorials - <https://docs.oracle.com/javase/tutorial/>
- Java Course - [http://www.vias.org/javacourse/wrapnt4F38D8\\_object\\_oriented\\_programming.html](http://www.vias.org/javacourse/wrapnt4F38D8_object_oriented_programming.html)