Laboratorio 02

Tema: Git y GitHub

Docente	Escuela	Asignatura
M.Sc. Ing. Richart Smith Escobedo Quispe rescobedoq@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web Semestre: III Código: 1702122

Laboratorio	Tema	Duración
02	Git y GitHub	06 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	06 Mayo 2024	10 Mayo 2024

Índice

1. Especificaciones del Laboratorio

1.1. Objetivos del curso

- Proporcionar los conocimientos y habilidades para el uso de las principales metodologías de análisis y diseño de sistemas.
- Brindar los conocimientos para la utilización de técnicas para el análisis y diseño de sistemas web.
- Proporcionar conocimientos y habilidades en el manejo de herramientas para el desarrollo de sistemas Web.
- Desarrollar sistemas de información dentro de una arquitectura cliente servidor.

1.2. Objetivos del laboratorio

- Utilizar el sistema de control de versiones Git.
- Utilizar la plataforma GitHub para replicar y administrar proyectos de programación.

1.3. Equipos, materiales y temas

- Sistema Operativo (`GNU/Linux` de preferencia).
- Editor de texto plano (`GNU Vim` de preferencia).
- Navegador Web: Chrome, Firefox, Edge, Brave, Opera, etc.
- `Git`.
- Cuenta en `GitHub` asociada al correo institucional.

2. Marco teórico

2.1. PowerShell

- PowerShell es una solución de automatización de tareas multiplataforma compuesta por un shell de línea de comandos, un lenguaje de secuencias de comandos y un marco de gestión de configuración. PowerShell se ejecuta en Windows, Linux y macOS.
- Instalar PowerShell en MS Windows: `https://learn.microsoft.com/en-us/powershell/scripting/install/installing-powershell-on-windows`

2.2. La W3C y los Estándares Web

- El World Wide Web Consortium o simplementa la W3C, desarrolla estándares y pautas para ayudar a todos a construir una web basada en los principios de accesibilidad , internacionalización , privacidad y seguridad.
- Los estándares web son planos, o bloques de construcción, de un mundo conectado digitalmente consistente y armonioso. Se implementan en navegadores, blogs, motores de búsqueda y otro software que potencia nuestra experiencia en la web.
- Página de la W3C: https://www.w3.org/
- Estándares: `https://www.w3.org/standards/`

2.3. Vim

- Vim es un editor de texto altamente configurable creado para hacer que la creación y el cambio de cualquier tipo de texto sean muy eficientes. Se incluye como "viçon la mayoría de los sistemas UNIX y con Apple OS X.
- Vim es muy estable y se desarrolla continuamente para mejorar aún más. Entre sus características se encuentran:
  - árbol de deshacer persistente de varios niveles
  - amplio sistema de complementos
  - soporte para cientos de lenguajes de programación y formatos de archivo
  - poderosa búsqueda y reemplazo
  - se integra con muchas herramientas
- Manual Vim:`http://ftp.vim.org/pub/vim/doc/book/vimbook-OPL.pdf`
- Descarga: `https://www.vim.org/download.php`

Listing 1: Vim en GNU/Linux

```
$ sudo apt-get install vim
```

Listing 2: Vim en MacOSX

```
$ brew install macvim
```

2.4. Visual Studio Code

- Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte integrado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes y tiempos de ejecución (como `C++, C#, Java, Python, PHP, Go, .NET`).
- Videotutoriales: `https://code.visualstudio.com/docs/getstarted/introvideos`
- Descarga: `https://code.visualstudio.com/Download`

2.5. Git

- `Git` es un sistema de control de versiones distribuido gratuito y de código abierto diseñado para manejar todo, desde proyectos pequeños hasta proyectos muy grandes, con rapidez y eficiencia.
- Documentación: `https://git-scm.com/doc`
- Descargas: `https://git-scm.com/downloads`

2.6. GitHub

- `GitHub` es una plataforma de hospedaje de código para el control de versiones y la colaboración. Este permite que tú y otras personas trabajen juntos en proyectos desde donde sea.
- Documentación: `https://docs.github.com/es`

2.7. Arquitectura Git

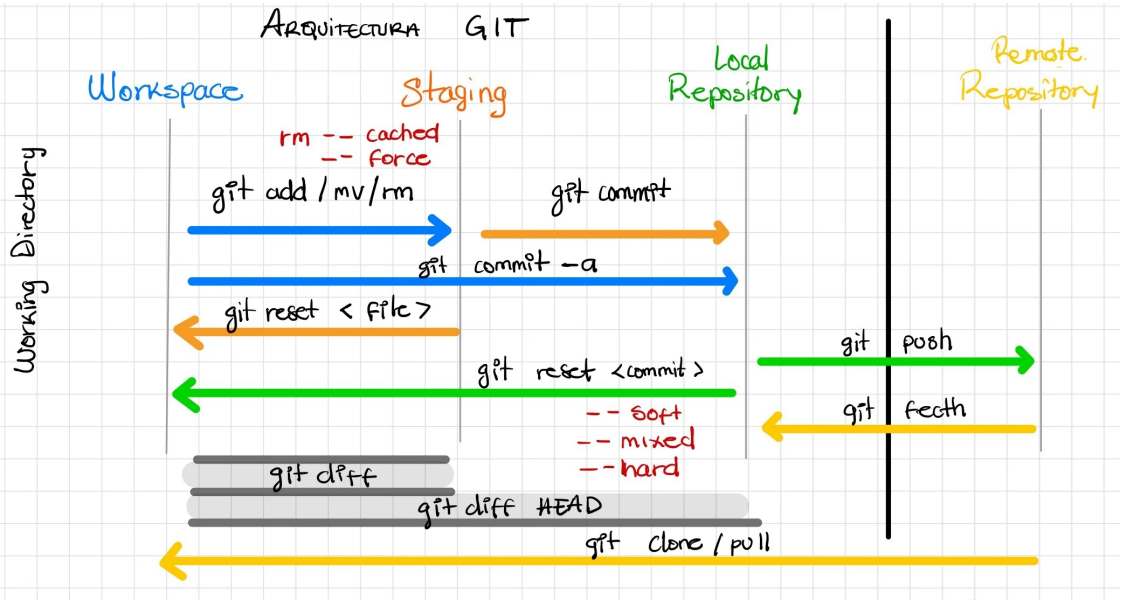

Figura 1: Arquitectura Git

- Git tiene tres estados principales en los que se pueden encontrar tus archivos: confirmado (committed), modificado (modified), y preparado (staged).
  - Confirmado: significa que los datos están almacenados de manera segura en tu base de datos local.
  - Modificado: significa que has modificado el archivo pero todavía no lo has confirmado a tu base de datos.
  - Preparado: significa que has marcado un archivo modificado en su versión actual para que vaya en tu próxima confirmación.
- Esto nos lleva a las tres secciones principales de un proyecto de Git: El directorio de Git (Git directory), el directorio de trabajo (working directory), y el área de preparación (staging area).

2.8. Estándares de codificación

- Los estándares de código son una serie de reglas definidas para un lenguaje de programación, o bien, un estilo de programación específico.
- El estilo garantiza que todos los ingenieros que contribuyen a un proyecto tengan una forma única de diseñar su código, lo que da como resultado una base de código coherente, asegurando una fácil lectura y mantenimiento.
- El uso de estándares es muy importante en la calidad de software, sin embargo mantener todos los proyectos cumpliendo a la perfección con esto no es una tarea fácil, requiere un gran esfuerzo y constancia por parte del equipo de desarrollo.
- Mientras más y más compañías han adoptado estándares, todavía hay aquellas que realizan el desarrollo de sus proyectos sin ellos.

- En todo caso se presentaran más problemas sin no se usan estos estándares de codificación.
- Algunos estándares de codificación:
  - Java: `https://amap.cantabria.es/amap/bin/view/AMAP/CodificacionJava`
  - PHP: `https://pear.php.net/manual/en/standards.php`
  - JS: `https://github.com/0granada/js-coding-standards`
  - Python: `https://peps.python.org/pep-0008/`
  - C++: `http://webdiis.unizar.es/asignaturas/PROG1/doc/materiales/sintaxis_cpp.pdf`

2.9. Diagramas

- Es muy conveniente utilizar herramientas que generen diagramas, para incluirlas en la documentación y explicación de código fuente.
- ObjectAid UML Explorer . (eclipse-jee-2021-03) `https://www.eclipse.org/downloads/packages/release/2021-03/r`, `https://drive.google.com/file/d/1n4Z0gGeC65BaRZO9QLOc6W4cROvNAyhQ/view`
- Graphviz es un programa de visualización gráfica de fuente abierta. `https://django-extensions.readthedocs.io/en/latest/graph_models.html`
- GraphStream A Dynamic Graph Library. `https://graphstream-project.org/`
- Drawio: Software de diagramación gratuito y de alta calidad. `https://app.diagrams.net/`

3. Guía de laboratorio

3.1. Directorio de trabajo personal

- Cree su directorio de trabajo personal. (Debe utilizar su usuario institucional.)
- Luego, diríjase a este directorio.

Listing 3: Creando directorio personal

```
$ mkdir -p $HOME/rescobedoq/
```

Listing 4: Dirijíendonos al directorio personal

```
$ cd $HOME/rescobedoq/
```

3.2. Repositorio local

- Cree un directorio para el repositorio local. (Utilizar el estándar pw2-24a)
- Luego, diríjase a este directorio.

Listing 5: Creando directorio para repositoio local

```
$ mkdir -p $HOME/rescobedoq/pw2-24a
```

Listing 6: Dirijíendonos al directorio para repositorio local

```
$ cd $HOME/rescobedoq/pw2-24a
```

3.3. Directorio del laboratorio

- Cree un directorio para el laboratorio correspondiente, dentro del directorio para el repositorio local. (Utilizar el estándar lab01)
- Luego, diríjase a este directorio.

Listing 7: Creando directorio para este laboratorio

```
$ mkdir -p $HOME/rescobedoq/pw2-24a/lab01/exercises/
```

Listing 8: Dirijíendonos al directorio del laboratorio

```
$ mkdir -p $HOME/rescobedoq/pw2-24a/lab01
```

3.4. Archivo .gitignore

- Siempre evalue utilizar el archivo `.gitignore` para no hacer seguimiento a archivos innecesarios, esto es muy importante sobre todo para ignorarlos en el repositorio GitHub.
- Pueden haber varios de estos archivos y estar ubicados estratégicamente en algún directorio; por ejemplo sólo para el laboratorio o para todo el repositorio local.

Listing 9: Creando .gitignore para todo el repositorio local

```
$ vim $HOME/rescobedoq/pw2-24a/.gitignore
```

Listing 10: Creando .gitignore específico para el laboratorio 01

```
$ vim $HOME/rescobedoq/pw2-24a/lab01/.gitignore
```

- Ejemplo de estructura de repositorio Git:

```
rescobedoq/
|--- pw2-24a
|   |--- README.md
|   |--- .gitignore
|   |--- lab01
|       |--- README.md
|       |--- .gitignore
|       |--- exercises
|           |--- holamundo.html
|       |--- latex
|           |--- rescobedoq_pw2_24a_lab01.tex
|           |--- rescobedoq_pw2_24a_lab01.pdf
|   |--- lab02
|       |--- README.md
|       |--- .gitignore
|       |--- exercises
|           |--- index.html
|           |--- myStyle.css
|           |--- myJS.js
|       |--- latex
|           |--- rescobedoq_pw2_24a_lab02.tex
|           |--- rescobedoq_pw2_24a_lab02.pdf
|   |--- lab03
|       |--- README.md
|       |--- .gitignore
|       |--- exercises
|           |--- HelloWorld.py
|       |--- latex
|           |--- rescobedoq_pw2_24a_lab03.tex
|           |--- rescobedoq_pw2_24a_lab03.pdf
```

Listing 11: Ejemplo de .gitignore en proyectos Java

```
*.class
```

Listing 12: Ejemplo de .gitignore en proyectos Python

```
my_env/*
*.pyc
```

3.5. Ejercicios

3.5.1. Repositorio Local

Crear un nuevo repositorio local Git

- Para crear un nuevo repositorio local Git.
- Se crea un directorio oculto .git.

Listing 13: git init

```
$ git init
```

Configurar repositorio

- Establecer 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.

Listing 14: git config

```
$ git config --global user.name "Richart Smith Escobedo Quispe"
$ git config --global user.email rescobedoq@unsa.edu.pe
$ git config --list
$ git config user.name
$ git config --global core.editor "code --wait"
$ git config --global credential.helper 'cache --timeout=3600'
$ git config --global color.ui true
$ git config core.autocrlf false
```

Estado de archivos

- Para mostrar el estado de los archivos en el repositorio Git.

Listing 15: git status

```
$ git status
```

El Staging Area

- Para añadir archivos al Staging Area.
- El punto "..ªgregará todos los archivos.

Listing 16: git add

```
$ git add holamundo.html
$ git add .
```

Enviar archivos al Repositorio Local

- Envía archivos al repositorio local.
- La opción -m permite escribir el mensaje en línea de comandos.

Listing 17: git commit

```
$ git commit -m "Probando el Hola Mundo"
```

Bitácora de envíos

- Permite ver un resumen de los commit(envíos) realizados.

Listing 18: git log

```
$ 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
```

Comparación de envíos

- Permite comparar los cambios de los envíos en los archivos.

Listing 19: git diff

```
$ git diff 6bb6b6e 6bb6b6e
```

3.5.2. Repositorio Remoto

Crear un repositorió GitHub privado

- URL de GitHub: `https://github.com/`
- Nombre del repositorio: pw2-24a
- El repositorio debe ser privado.

- Poner al docente como colaborador para que pueda clonar, sincronizar y revisar.
- No auto crear README.md o licencias desde la plataforma GitHub. (Siempre usar la terminal)
- Usuario GitHub del profesor: @rescobedoq
- La rama principal en GitHub se llama "main".

Generar Tokens para acceso

- Los tokens se deben generar para que se puedan usar al momento de querer acceder a la API de GitHub.
- Genere un token clásico desde: `https://github.com/settings/tokens`
- Copie el token y envíeselo usted mismo por correo electrónico, ya que despues no podra volver a verlo. Si se olvidó elimínelo y genere otro.

Asociar repositorio local con Repositorio GitHub

- Persigue un repositorio remoto para hacer push.

Listing 20: git remote

```
$ git remote add origin https://github.com/rescobedoq/pw2-24a.git
$ git remote -v
```

Cambiar el nombre de la rama actual a main

- Desde el 1 de octubre de 2020, GitHub utiliza "mainçomo rama principal.
- Cambia el nombre de su rama principal a main en lugar del valor predeterminado de git de master.

Listing 21: Rama main en GitHub

```
$ git branch -M main
```

Subir archivos del repositorio local al Repositorio GitHub

- Permite subir archivos al repositorio remoto.

Listing 22: git remote

```
$ git git push -u origin main
```

Clonar un Repositorio GitHub

- Clona un repositorio remoto como un repositorio local, en el cual se puede hacer push. (actualizarlo)
- Los repositorios públicos no solicitan credenciales.
- Los repositorios privados solicitarán el usuario github y el token.

Listing 23: git clone

```
$ git clone https://github.com/rescobedoq/pw2-24a.git
```

Sincronizar Repositorio Local con su Repositorio GitHub

- Permite descargar los cambios del repositorio remoto al directorio local. (sincronizar)
- Los repositorios públicos no solicitan credenciales.
- Los repositorios privados solicitarán el usuario github y el token.

Listing 24: git pull

```
$ git pull
```

4. Tarea

4.1. Descripción

- Elabore un informe individual, de estudio de Git y GitHub.
- Cree su página personal con las recomendaciones dadas por la W3C en su tutorial de CSS: Empezando con HTML y CSS: `https://www.w3.org/Style/Examples/011/firstcss.en.html`.
  - `index.html` - (Menú principal - Página principal de bienvenida al sitio beb)
  - `autor.html` - (Menú principal - Página de presentación del autor) (Se activa Menú Izquierdo)
    - `hobbies.html` - (Menú Izquierdo - Página de fotos y descripciones de sus hobbies.)
    - `ingSoftware.html` - (Menú Izquierdo - Página donde explique ¿Qué es la Ing. de Software desde su punto de vista?. Descripción, especialidades que resalte.)
    - `galeria.html` - (Menú Izquierdo - Página de fotos y descripciones libres que quiera compartir.)
  - `estandaresWeb.html`- (Menú principal - Página donde describa la W3C y algunos estándares web.)
  - `contactame.html`- (Menú principal - Página donde muestra un formulario de contacto.)
- Menú Principal |*Inicio|Autor|Estándares Web|Contáctame|
- Inicio: Página de presentación del sitio. "Bienvenido a ...".
- Autor: Descripción del autor. "Mi nombre es ...".
- Estándares Web: SVG, WOFF, WebRTC, XML.
- Contáctame: Formulario: |nombres:input| |correo:email| |genero:radiobutton| |nacimiento:date| |asunto:input| |contenido:textarea| |enviar:button|
- Menú Izquierdo |*Autor| ->|Hobbies|Ing. de Software|Galería|
- Hobbies: Descripción, tiempo, recomendaciones, fotos.
- Ing. de Software: Acerca de la carrera de Ing. de Software. Futuro. Especialidad.
- Galería: Fotos en barrio, universidad, trabajo u otras actividades, mascota.
- Contactanos: Formulario que guarda en una base de datos. Las consutas que se desean enviar. nombres, correo electrónico, asunto, detalle..
- Utilice todas las recomendaciones dadas por el docente:
  - Antecedentes: Describir antecedentes previos que sean necesarios para desarrollar el laboratorio. (Las entregadas por el docente y/o las que se buscaron personalmente).
  - No usar JavaScript. Sólo HTML y CSS.
  - Commits: Elaborar la lista de envíos que permitiran culminar el laboratorio. (Previo a la implementación)
  - Source: Explicar porciones de código fuente importantes, trascendentales que permitieron resolver el laboratorio y que reflejen su particularidad única. (Sólo en trabajos grupales se permite duplicidad)
  - Ejecución: Muestra comandos, capturas de pantalla, explicando la forma de replicar y ejecutar el entregable del laboratorio.

- Validación: Muestra validación HTML y CSS.

- Utilice DockerFile para realizar operaciones automatizadas en Docker (incluido arrancar el servidor web a traves de un puerto y copiar el proyecto web para acceder desde la máquina anfitrion.)
- Ejemplo: http://127.0.0.1:8085/lab02/

4.2. Pregunta

- Mencione tres aportes a su adquisición de conocimientos que este laboratorio le proporcionó.

4.3. Entregables

- URL al directorio específico del laboratorio en su repositorio GitHub privado donde esté todo el código fuente y otros que sean necesarios. Evitar la presencia de archivos: binarios, objetos, archivos temporales, cache, librerias, entornos virtuales. Si hay configuraciones particulares puede incluir archivos de especificación como: packages.json, requirements.txt o README.md.
- No olvide que el profesor debe ser siempre colaborador a su repositorio que debe ser privado (Usuario del profesor **@rescobedoq**).
- El informe debe estar elabórado en LaTeX
- Usted debe describir sólo los **commits** más importantes que marcaron hitos en su trabajo, adjutando capturas de pantalla, del commit, porciones de código fuente, evidencia de sus ejecuciones y pruebas.
- En el informe siempre se debe explicar las imágenes (código fuente, capturas de pantalla, commits, ejecuciones, pruebas, etc.) con descripciones puntuales pero precisas.
- Agregar la estructura de directorios y archivos de su laboratorio hasta el nivel 4.

```
rescobedoq/
|--- pw2-24a
|   |--- README.md
|   |--- .gitignore
|   |--- lab01
|       |--- README.md
|       |--- index.html
|       |--- autor.html
|       |--- autor
|           |--- hobbies.html
|           |--- ingSoftware.html
|           |--- galeria.html
|       |--- estandaresWeb.html
|       |--- contactame.html
|       |--- .gitignore
|       |--- css
|           |--- style.css
|       |--- exercises
|           |--- HelloWorld.java
|       |--- latex
|           |--- rescobedoq_pw2_24a_lab01.tex
|           |--- rescobedoq_pw2_24a_lab01.pdf
```

5. Rúbricas

5.1. Sobre el Informe

Tabla 1: Rúbrica para el Informe

Informe		**Cumple**	**No cumple**
LaTeX	El informe está en formato PDF desde LaTeX, con un formato limpio (buena presentación) y fácil de leer.	20	0
Observaciones	Respetar la estructura de organización para ubicación de los entregables. Por cada observación dentro del informe se le descontará puntos. Se debe incluir el código fuente latex del informe (`*.tex`)	-	-

5.2. Contenido del Informe

- El alumno deberá autocalificarse, marcando o dejando en blanco las celdas de la columna **Checklist**, deacuerdo a si cumplió o no con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación siempre será sobre la nota mínima aprobatoria, siempre y cuando cumpla con todos lo items. (Máximo 24 horas)
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la tabla de calificación de niveles de desempeño:

Tabla 2: Niveles de desempeño

	Nivel			
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Tabla 3: Rúbrica para contenido del Informe y evidencias

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Repositorio se pudo clonar y se evidencia la estructura adecuada para revisar los entregables. (Se descontará puntos por error o onservación)	4			
2. Commits	Hay porciones de código fuente asociado a los commits planificados con explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4			
3. Ejecución	Se incluyen comandos para ejecuciones y pruebas del código fuente explicadas gradualmente que permitirían replicar el proyecto. (Se descontará puntos por cada omisión)	4			
4. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2			
7. Ortografía	El documento no muestra errores ortográficos. (Se descontará puntos por error encontrado)	2			
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente con explicaciones puntuales pero precisas, agregando diagramas generados a partir del código fuente y refleja un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4			
Total		20			

6. Referencias

- https://learn.microsoft.com/en-us/powershell/scripting/install/installing-powershell-on-windows
- https://www.geeksforgeeks.org/download-and-install-java-development-kit-jdk-on-windows-mac-and-
- https://www.oracle.com/java/technologies/downloads/#java8
- https://www.oracle.com/java/technologies/downloads/#java11
- http://ftp.vim.org/pub/vim/doc/book/vimbook-OPL.pdf
- https://www.vim.org/download.php
- https://code.visualstudio.com/docs/getstarted/introvideos
- https://code.visualstudio.com/Download
- https://git-scm.com/doc
- https://git-scm.com/downloads
- https://docs.github.com/es
- https://github.com/
- https://github.com/settings/tokens
- https://omegaup.com/problem/collection/
- https://amap.cantabria.es/amap/bin/view/AMAP/CodificacionJava
- https://pear.php.net/manual/en/standards.php
- https://github.com/0granada/js-coding-standards
- https://peps.python.org/pep-0008/
- http://webdiis.unizar.es/asignaturas/PROG1/doc/materiales/sintaxis_cpp.pdf
- https://www.eclipse.org/downloads/packages/release/2021-03/r
- https://drive.google.com/file/d/1n4Z0gGeC65BaRZO9QLOc6W4cROvNAyhQ/view
- https://django-extensions.readthedocs.io/en/latest/graph_models.html
- https://graphstream-project.org/
- https://app.diagrams.net/
- https://validator.w3.org/
- https://jigsaw.w3.org/css-validator/
- https://www.w3.org/TR/html401/
- https://developer.mozilla.org/en-US/docs/Glossary/HTML5
- https://www.w3schools.com/html/
- https://lenguajehtml.com/html/
- https://developer.mozilla.org/en-US/docs/Web/CSS
- https://www.w3schools.com/css/

6.1. Otros enlaces

- 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
- 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-tod
- 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-h8YlBj96OwMP
- 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