

**Control de Lectura: GitHub**

**NOMBRE DEL ESTUDIANTE:**

Shaffik Samir Samandé Mujica

Universidad de las Fuerzas Armadas - ESPE

Segundo Semestre de Carrera en TICS – PREGRADO SII 2024/2025

PRIMER PARCIAL

Asignatura de Programación Orientada a Objetos – NRC: 1322

Ing – Luis Enrique Jaramillo Montaño

20 de noviembre de 2024

Sangolquí – Ecuador

**Informe de Deberes de Programación Orientada a Objetos**

**Curso:** Fundamentos de Programación  
**Profesor:** Luis Enrique Jaramillo Montaño  
**Fecha:** miércoles, 20 de noviembre de 2024

**I. Portada**

1. **Título del Proyecto:**
   * Control de lectura: GitHub
2. **Datos del Estudiante:**
   * Shaffik Samir Samandé Mujica
   * 202483894L00440302
   * sssamande@espe.edu.ec
3. **Fecha de Entrega:**
   * Miércoles, 20 de noviembre del 2024

**II. Índice**

1. Introducción
2. Objetivos del Proyecto
3. Metodología
4. Desarrollo del Proyecto
5. Resultados
6. Conclusiones
7. Referencias
8. Anexos

**III. Introducción**

* **Descripción del Proyecto:**
  + En este presente informe, se familiariza hacia los alumnos, el uso de herramientas de control de versiones o de lectura, especialmente Git y GitHub. Después de crear un programa simple en el lenguaje de programación Java, los alumnos se familiarizarán con el trabajo en equipo, el control de versiones o de lectura y la carga del código en un repositorio remoto. Para lograrlo, se desarrolló un programa básico para enviar el mensaje "Hola mundo" a la consola, luego se usó GitHub para alojar el código y seguir las mejores prácticas de desarrollo de software.
* **Antecedentes:**
  + En POO, el control de versiones o de lectura, se ha convertido en una herramienta esencial para el desarrollo de software profesional, especialmente cuando se trabaja en equipos o en proyectos de mediana y gran escala. Git es uno de los sistemas de control de versiones más utilizados en la industria y su integración con plataformas como GitHub permite una gestión eficiente del código fuente y la colaboración entre desarrolladores. Desde ese punto de vista, aprender a utilizar estas herramientas en las primeras etapas de la preparación académica es fundamental para desarrollar las habilidades necesarias en un entorno profesional.

**IV. Objetivos del Proyecto**

* **Objetivo General:**
  + Familiarizar con el uso de Git y GitHub para gestionar versiones del código, mediante la elaboración y publicación de un proyecto sencillo realizado en el lenguaje de programación Java.
* **Objetivos Específicos:**
  + Crear un programa en Java que implemente una funcionalidad básica de impresión en consola, como "Hola Mundo".
  + Aprender a utilizar Git para gestionar el código fuente, desde la creación del repositorio hasta la realización de commits.
  + Publicar el código fuente al repositorio individual en GitHub, verificando la correcta configuración y uso de la plataforma.

**V. Metodología**

* **Herramientas y Tecnologías Utilizadas:**
  + Java
  + Computadora/Laptop con sistema operativo Windows
  + GitHub
  + Git
  + OnlineGDB o Apache NetBeans IDE
* **Procedimiento:**
  + **Análisis y definición de problemas:**

**Descripción del problema:** El análisis principal, es familiarizar a los alumnos con las herramientas de control de versiones o de lectura, especialmente Git y GitHub, fundamentales para el desarrollo de software profesional. Aunque el programa es muy sencillo de crear, se consideró necesario incluir este ejercicio en los temas de Fundamentos de Programación para que los estudiantes puedan familiarizarse con las mejores prácticas de gestión de código desde el inicio del curso.

El problema planteado consiste en dos aspectos importantes:

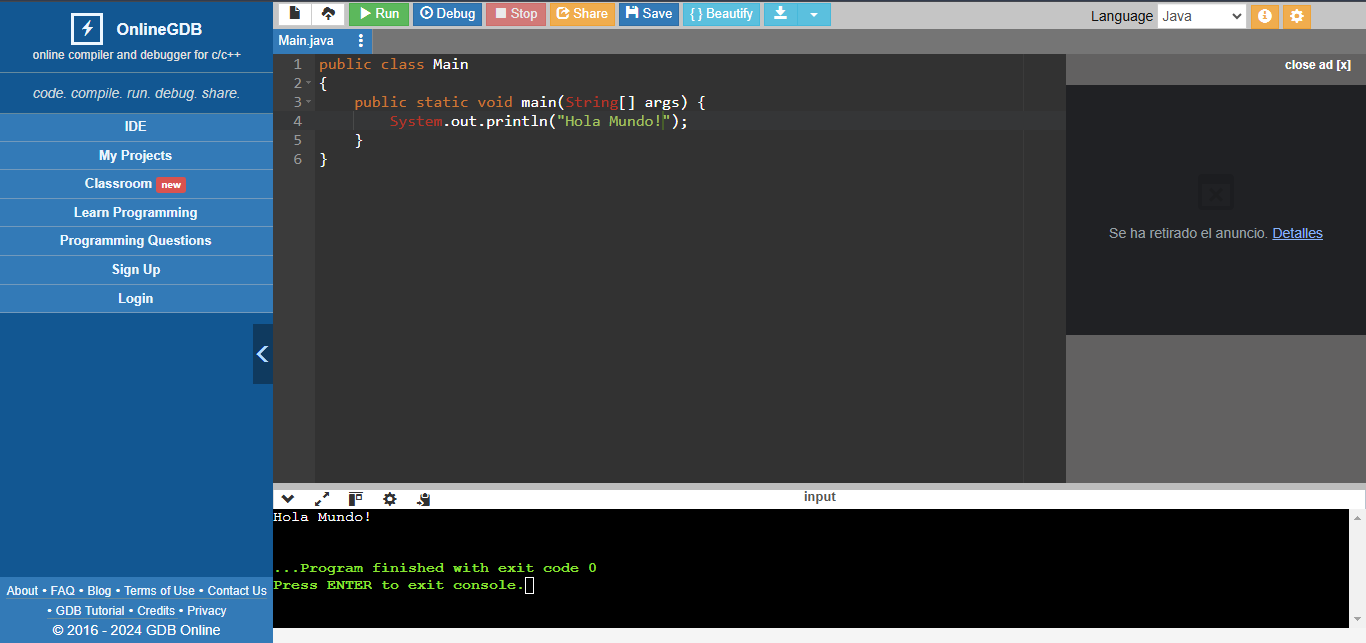
* **Desarrollar un programa en Java que sea fácil de funcionar,** que permita al alumno hacer la práctica de la creación del código en un entorno de desarrollo.
* **Aprender a utilizar y practicar Git y GitHub,** para gestionar el código de manera apropiada, y empezar a elaborar con un sistema de control de versiones o de lectura, que será esencial en la carrera profesional. Esto implica desde la creación de un repositorio en Git hasta el uso de comandos básicos como *git add, git commit* y *git push*.

Este ejercicio no solo implica la creación de un programa simple, sino también la adopción de buenas prácticas desde el punto de vista del control de versiones o de lectura, lo que permite una gestión más ordenada y escalable del código a medida que los proyectos se vuelvan más complejos.

Al final, el alumno no solo sea capaz de escribir un programa básico, sino también de mantener el control de sus versiones o de lectura, realizar cambios de forma eficiente y documentar los avances del proyecto a través de los *commits* en GitHub.

**VI. Desarrollo del Proyecto**

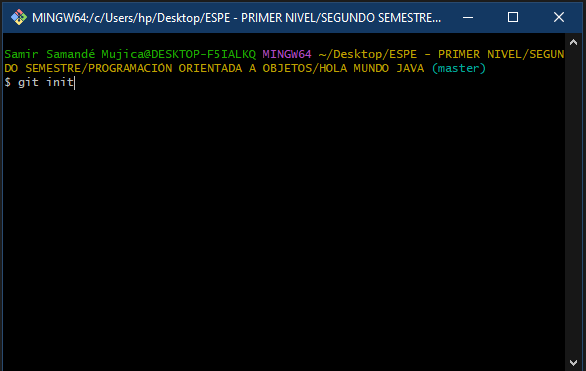
* **Implementación del Código**
  + El primer paso fue crear un programa sencillo en Java que cumpliera con la debida función de imprimir el mensaje de “Hola Mundo” en la consola. Es un ejemplo clásico utilizado para familiarizarse con la sintaxis y la básica estructura de un lenguaje de programación.
* **Código Fuente:**
  + El código fuente del programa con explicaciones se presenta a continuación. También se incluyen capturas de pantalla del código y su ejecución en Java.



* **Public class Main {:** Significa la clase, que es el que contiene el código y se accede desde otras partes del programa.
* **Public static void main(String[] args) { :** Es el punto de entrada del programa, indicando que comenzará la ejecución en este método.
* **System.out.println(“Hola Mundo”); }:** Es la acción que realiza el programa: imprime o da el mensaje en la consola el mensaje “Hola Mundo”.

Este programa es el primer paso en el aprendizaje de Java porque presenta a los alumnos la estructura básica del programa y cómo utilizar herramientas de salida estándar. Aunque es un programa simple, proporciona una base sólida para comprender conceptos básicos como clases, métodos e interacciones básicas de la consola.

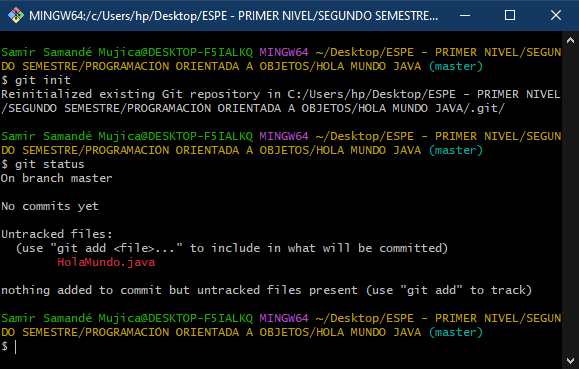
* **Creación del Proyecto en Java:**
  + El primer paso fue crear el archivo *HolaMundo.java* utilizando el entorno de desarrollo integrado (IDE). En este archivo, se escribió el código que imprime el mensaje “Hola Mundo” en la consola.
* **Creación del Repositorio con Git.**
  + Una vez elaborado y hecho las pruebas del código, el siguiente paso fue crear un repositorio en Git en la carpeta local donde se encontraba el archivo Java. Para ello, se utilizaron los siguientes comandos en la terminal:



Esto permitió crear un repositorio vacío en el directorio del proyecto, donde se empezarían a registrar los cambios.

* **Verificación del Estado del Repositorio:**

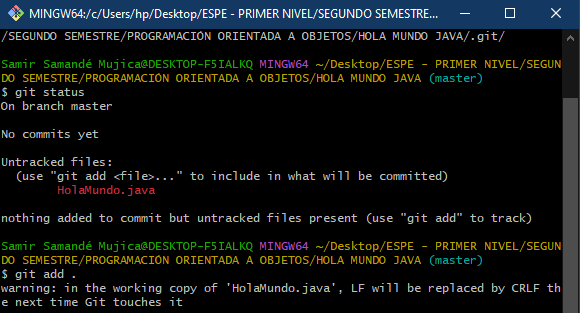
Antes de realizar cualquier adición o *commit*, es importante verificar el estado del repositorio utilizando el comando *git status*. Mostrando qué archivos han sido modificados, y cuáles están listos para ser añadidos al *commit* y cuáles aún no se han identificado.



Al ejecutarlo, nos mostrará la información que han cambiado desde el último *commit*, los archivos que han sido modificados, pero no añadidos al área de *staging,* y cualquier archivo que no haya sido identificado aún. Es importante porque permite al usuario observar las modificaciones que han sido realizadas y cuáles no han sido comprometidas.

* **Adición del Archivo al Repositorio:**

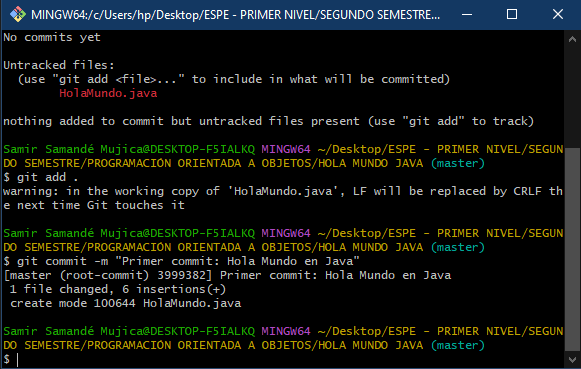
Luego, se añadió el archivo creado al índice de Git, utilizando el comando *git add .* , que indica a Git que incluya todos los archivos modificados en el siguiente *commit*.



Este comando prepara los archivos para el próximo *commit*. El comando *git status* puede ser ejecutado de nuevo para comprobar que los archivos ahora están en el área de *staging*, listos para ser confirmados.

* **Realización del Primer Commit:**

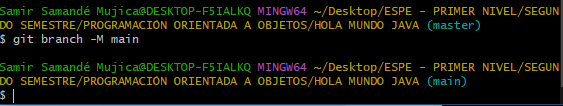
Posteriormente, se realizó el primer *commit* para guardar el estado actual del proyecto. Es importante que cada commit esté acompañado de un mensaje que describa los cambios realizados.



Este commit, sirve para guardar la primera versión del código en el repositorio local.

* **Cambio de la Principal Rama a *main* con *git branch -M main*:**

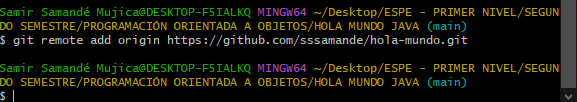
En Git, por defecto, la principal rama se llama *master*. No obstante, muchos repositorios han adoptado como *main* que es el nombre la principal rama para ser más inclusivos. Para cambiar el nombre de la rama principal de *master* a *main*, se utiliza el siguiente comando:



Este comando *git branch -M main* realiza un cambio de nombre o un *“rename”* de la principal rama a *main.* La opción *-M* se aseguró en hacer el cambio de nombre de la rama, incluso si ya existe una rama llamada *main*.

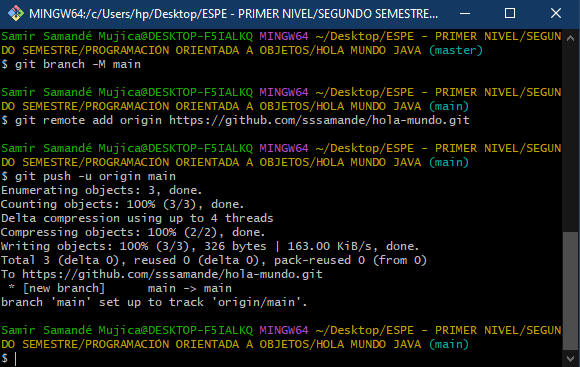
* **Creación del Repositorio en GitHub:**

Posteriormente, se creó el repositorio en GitHub, para que el proyecto de forma remota se aloje. Con tal efecto, se accedió a la plataforma web de GitHub y se creó un nuevo repositorio con un nombre apropiado o específico. Después, se vinculó el repositorio local con el remoto mediante el siguiente comando:



* **Subida del Proyecto a GitHub:**

Una vez vinculado el repositorio local con el remoto y cambiada la principal rama a *main,* se procedió a publicar los archivos al repositorio de GitHub con el siguiente comando:



Este comando publica el código al repositorio de GitHub, permitiendo que el proyecto esté disponible de forma remota, y es accesible desde cualquier dispositivo. La opción *-u ­*indica que se debe identificar la rama *main* de forma predeterminada, lo que significa que en futuros *git push* y *git pull*, no será necesario específica el nombre de la rama.

* **Resultados obtenidos:**
* Una vez subido el proyecto a GitHub, se comprobó que el código estuviera correctamente reflejado en la plataforma. Además, se realizaron pruebas para verificar que el programa con el lenguaje de programación Java ejecutara el mensaje “Hola Mundo” sin errores.
* También, se aseguró que el repositorio en GitHub estuviera disponible públicamente y que el historial de *commits* estuviera bien redactado.

**VII. Resultados**

* **Análisis de Resultados:**
  + El proceso de creación y publicación del código en GitHub se completó con éxito. El programa Java se ejecutó exitosamente, mostrando el mensaje esperado en la consola. Además, se ha configurado correctamente el repositorio de GitHub, permitiendo acceder al código desde cualquier dispositivo conectado a Internet.
* **Comparación con Objetivos:**
  + Los resultados cumplen con los objetivos asumidos, demostrando una correcta implementación y funcionamiento de los códigos realizados en Java y la publicación del código en el repositorio a través de Git hacia GitHub.

**VIII. Conclusiones**

* **Conclusiones Generales:**
  + El uso de Git y GitHub es importante para gestionar los proyectos de programación y trabajar de manera accesible en equipo.
  + La práctica de utilizar herramientas de control de versiones en pequeños proyectos es una forma excelente de familiarizarse con el funcionamiento, antes de aplicar las herramientas aún en los complejos proyectos.
  + La creación de proyectos sencillos, como el programa “Hola Mundo”, es esencial para consolidar los básicos conocimientos en cualquier lenguaje de programación.
* **Recomendaciones:**
  + Seguir practicando con Git y GitHub para mejorar la fluidez en el uso de las herramientas.
  + Realizar los proyectos más complejos que impliquen la participación con otros desarrolladores y el manejo de ramas y funciones.
  + Aprender a utilizar las descripciones detalladas y claras en los *commits*, para que el historial de cambios sea más comprensible para otros colaboradores.

**IX. Referencias**

* **Bibliografía:**

Hilari, F. [Hilari Web] (29 de septiembre de 2022). *72 – COMO SUBIR PROYECTO LOCAL A GITHUB para compartir el código fuente del sistema (PHP y MYSQL)* [Archivo de video]. YouTube. <https://www.youtube.com/watch?v=eWL2UY2_aTM>

Somos Devs. (4 de mayo de 2022). *Como hacer un hola mundo en java* [Archivo de Vídeo]. YouTube. <https://www.youtube.com/watch?v=2_KAErlVcbU>

ATLASSIAN. (s.f.). *Configuración de un repositorio*. https://www.atlassian.com/es/git/tutorials/setting-up-a-repository

**X. Anexos**

