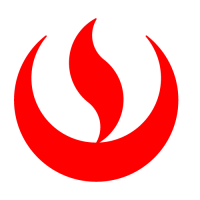
**UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS**

**FACULTAD DE INGENIERÍA**

**CARRERA PROFESIONAL**

**DE CIENCIAS DE LA COMPUTACIÓN**



**Asignatura:**

**CC218 - TEORÍA DE COMPILADORES**

**Sección: CC61**

**Student Outcome 3: La capacidad de comunicarse efectivamente con un rango de audiencias**

**(Nivel de logro 2)**

**Informe de Desempeño**

**Autor**

Jak Cristian Campos Espinoza – U20211A184

**Profesor**

Luis Martin Canaval Sanchez

Lima, noviembre de 2023

**ÍNDICE**

[**Introducción** 3](#_Toc151727535)

[**Objetivos** 3](#_Toc151727536)

[**Marco Teórico** 3](#_Toc151727537)

[**Desarrollo** 3](#_Toc151727538)

[**Método** 3](#_Toc151727539)

[**Tareas** 3](#_Toc151727540)

[**Comunicación** 3](#_Toc151727541)

[**Evidencia** 4](#_Toc151727542)

[**Competencias ABET y Student Outcome** 4](#_Toc151727543)

[**Conclusiones** 4](#_Toc151727544)

# **Introducción**

El presente informe detalla el desarrollo de un intérprete de comandos tipo shell implementado en C++ con el uso de ANTLR (ANother Tool for Language Recognition). El objetivo de este proyecto es crear un intérprete de comandos básico que reconozca y ejecute diversas órdenes típicas de una shell, como listar archivos, cambiar de directorio, crear y eliminar archivos y directorios, entre otros.

# **Objetivos**

* Implementar un intérprete de comandos funcional usando ANTLR en C++.
* Reconocer una serie de comandos básicos y ejecutar acciones correspondientes.
* Permitir interacción y ejecución de comandos a través de la línea de comandos.

# **Marco Teórico**

ANTLR (ANother Tool for Language Recognition) es una herramienta que facilita la creación de analizadores sintácticos para lenguajes de programación. Permite definir gramáticas de manera sencilla y generar analizadores que pueden ser utilizados en distintos lenguajes de programación, incluyendo C++.

# **Desarrollo**

## **Método**

Se empleó ANTLR para definir la gramática de un lenguaje simple de comandos shell. Se generó un analizador léxico y sintáctico utilizando ANTLR, y se implementaron las acciones correspondientes a cada regla gramatical utilizando C++.

## **Tareas**

* Definición de la gramática del lenguaje de comandos.
* Generación del analizador léxico y sintáctico con ANTLR.
* Implementación de acciones para cada regla gramatical en C++.
* Pruebas y depuración del intérprete de comandos.

## **Comunicación**

La comunicación y colaboración en el desarrollo de este proyecto se realizó principalmente a través de herramientas de control de versiones como Git, permitiendo un trabajo colaborativo y un seguimiento de cambios y avances en el código.

## **Evidencia**

La evidencia de este proyecto reside en el código fuente desarrollado, que incluye la gramática definida en ANTLR, la implementación en C++ y el correcto funcionamiento del intérprete de comandos.

## **Competencias ABET y Student Outcome**

ABET: Este proyecto contribuye al desarrollo de habilidades en diseño de lenguajes formales, implementación de analizadores léxicos y sintácticos, y programación en C++.

Student Outcome: Se fortalecen habilidades en la resolución de problemas, comprensión de gramáticas formales, trabajo en equipo y aplicación de conceptos de lenguajes de programación.

# **Conclusiones**

La implementación de una Shell de comandos usando ANTLR y C++ ha sido exitosa, permitiendo reconocer y ejecutar una serie de comandos. Se ha logrado comprender y aplicar conceptos fundamentales de análisis léxico y sintáctico, así como el diseño e implementación de un intérprete básico. Este proyecto sienta las bases para explorar y ampliar funcionalidades en futuros desarrollos de intérpretes y compiladores. El proyecto demuestra la capacidad para aplicar conceptos teóricos en la práctica y sienta las bases para futuras investigaciones en el campo de los lenguajes de programación.