



Área de Ingeniería En Computación

Compiladores e Intérpretes

Entrega final

Generador

Estudiantes:

Ignacio Castillo Montero - 2021579556

Miguel Cubero Valverde - 2023271617

José Miguel González Barrantes 2023087564

Alejandro Gutierrez Chaves - 2023066266

Felipe Masís Calderón - 20230478825

Profesor:

Aurelio Sanabria Rodríguez

30 de Junio

I Semestre, 2025

Justificación

Este programa nació de una idea rara pero que nos hizo sentido: mezclar programación con cocina. No fue algo súper planeado desde el inicio, simplemente se dio. Pensamos que si tantas veces nos explicaron algoritmos como “recetas”, ¿por qué no llevarlo más allá y hacer un lenguaje completo con esa lógica? Y en el camino, descubrimos que esa comparación nos ayudaba a entender mejor cómo funcionan los lenguajes de verdad.

Desde el inicio, nuestro objetivo fue más que solo hacerlo funcionar. Queríamos que este programa sirviera como una herramienta para aprender, para experimentar, y para hacer la programación más accesible. Al crear el analizador, el generador y el sistema que convierte lo escrito en nuestro lenguaje a Python, logramos entender mejor lo que hay detrás de los lenguajes de programación. Dejamos de ser solo usuarios del código y pasamos a ser quienes lo diseñan.

Durante el proceso, también nos dimos cuenta de que construir un lenguaje no es solo cosa de técnicos. Es pensar en el usuario, en cómo explicar bien una idea, y en cómo traducirla sin perder el sentido. Tuvimos que hacer elecciones, priorizar claridad, y aprender a equilibrar lo simple con lo funcional.

Este programa fue una mezcla entre lo técnico y lo creativo. Y aunque empezó con una metáfora, terminamos convencidos de que sí: programar se parece bastante a cocinar. Tenés ingredientes, seguís pasos, podés improvisar, y al final el resultado es algo propio. Esa fue, y sigue siendo, la motivación que nos empujó a terminar este programa.

Lecciones aprendidas

Ignacio Castillo Montero:

En mi opinión, las lecciones aprendidas de esta fase final son menores ya que la mayor parte del trabajo está hecha. Aparte del hecho de que se aprende cómo es que se usan las plantillas, por lo menos en Python, también sirve como recordatorio de lo importante que es estar bien familiarizado con todo el lenguaje en todo momento. Si

algo se dejó mal hecho en fases tempranas, esa decisión te persigue y termina pasándote la factura al final, cuando es más complicado corregir.

Miguel Cubero Valverde:

Algo que me quedó muy claro es que trabajar en algo que uno mismo inventa suena vacilón al inicio, pero si no hay reglas claras desde el principio se vuelve una pesadilla. Aprendí que definir bien la gramática desde el inicio es clave, porque después se conecta con todo lo demás: los analizadores, el generador, etc. No se puede andar improvisando porque después todo falla en cadena. También entendí mejor cómo funcionan los compiladores y transpiladores, cosas que antes uno usaba sin saber del todo qué pasaba por detrás.

José Miguel González Barrantes:

Una de las cosas que más aprendí fue cómo una idea creativa puede transformarse en algo útil si se trabaja con constancia y lógica. El proceso de pasar de recetas de cocina a instrucciones reales de código fue más complicado de lo que pensé. Me di cuenta de que hay mucho detalle técnico que uno suele pasar por alto cuando solo programa para entregar. Aquí tocó pensar en cómo otros van a entender el lenguaje y cómo interpretarlo, lo cual me dejó una mejor perspectiva sobre diseño de software.

Alejandro Gutiérrez Chaves:

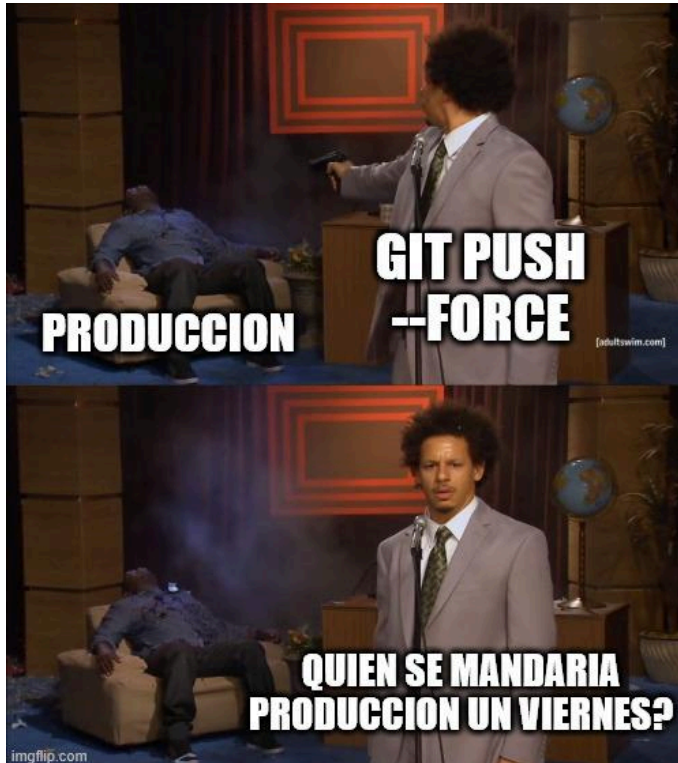
Me sorprendió lo mucho que se aprende intentando replicar lo que ya hace un lenguaje de programación, desde lo más básico como imprimir hasta construir estructuras más complejas. Entendí que programar no siempre es solo escribir código, sino pensar cómo ese código se genera, cómo se estructura y qué lo hace comprensible. Esta experiencia me cambió un poco la forma en la que veo la programación: no solo como herramienta, sino como lenguaje en serio, que tiene sus reglas, su cultura y su forma de crecer.

Felipe Masís Calderón:

Aprendí que en este tipo de programas hay que tener respeto al detalle. Uno se puede emocionar con ideas originales o creativas, pero si no se bajan a tierra con una

estructura clara, no sirven. También noté que hacer herramientas para otros (aunque sean simples) es más difícil que hacerlas para uno mismo. Hay que pensar en los errores, en cómo se lee el código, en que todo tenga sentido. Me dejó muy claro que la programación no es solo lógica, también es comunicación

Memazos





imgflip.com



imgflip.com

EL DIA DE MI BODA

**EL DIA QUE
WINDOWS 11 MUERA**



**ESCRIBIR
CÓDIGO
DIRECTAMENTE
EN PYTHON**



**ESCRIBIR
TU PROPIO
LENGUAJE**