

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación.

IC5701 – Compiladores e Intérpretes.

Proyecto 5 Generador – Ciruelas Extendido

Profesor: Aurelio Sanabria.

Estudiante:

Tamara Nicole Rodríguez Luna.

3 de junio del 2025.

Justificación

Ciruelas es un lenguaje de programación creado teniendo en mente los costarrriqueñismos, para hacer este lenguaje funcional se hizo el desarrollo de un transpilador y en este proyecto específicamente se hizo una extensión de sus capacidades. Lo que motiva su construcción es diferenciarse de los lenguajes de programación tradicionales, todos ellos están en inglés, pero Ciruelas está en español e incluye diferentes costarrriqueñismos, esto es sumamente importante pues al ser Costa Rica un país tan pequeño con una población de pocos millones no destacamos ni siquiera en la comunidad hispanohablante. Ciruelas es una demostración de nuestra presencia, un producto costarricense que aporta a compartir y preservar nuestra cultura, pues son muy pocos los productos que representan a Costa Rica y al estar expuestos a cientos de millones de personas hispanohablantes se va perdiendo la forma de hablar costarricense. Teniendo eso en cuenta, quienes quieran utilizar Ciruelas podrán hacer una investigación de los diferentes costarrriqueñismos utilizados en este lenguaje para tener una mejor comprensión de la cultura en la que se inspira.

Lecciones aprendidas

Aún en la etapa final se descubren errores de las primeras etapas, la función `echandi_jiménez()` no funcionaba porque el analizador asumía que todas las funciones siempre tenían parámetros, por fortuna fue una única función que debió ser cambiada. Aunque al trabajar con un lenguaje diferente uno se va acostumbrando a la sintaxis única todavía hay resultados que se ven incómodos y valdría la pena seguir trabajando en ellos para obtener un mejor resultado, un ejemplo de esas líneas poco estéticas es:

```
resultado metale llamese hacer_menjunje(resultado / caracter)
```

Esa línea forma parte de un `delete` vuelta y en Python pudo ser hecho con un simple:

```
resultado = caracter * veces
```

Esto es muy probable que sea el resultado de la gramática LL(1) pues al tener esa restricción hubo que crear elementos adicionales para poder diferenciar casos como una invocación de una asignación.

Por otro lado, trabajar en este proyecto viendo las diferentes etapas ayuda a un mejor entendimiento de todo lo que tiene que pasar para ir de un código desconocido o inventado en unas horas a un lenguaje de programación y en este caso al ser Python el objetivo ayuda a tener una idea más clara de lo que se quiere. Durante el desarrollo de la carrera de caracoles varias veces se dio un vistazo al código en Python ya transpilado para poder ver qué estaba fallando, eso ayudó en el caso del fallo de `echandi_jiménez()` sería más difícil intentar hacerlo hacia un lenguaje menos popular como Haskell o ensamblador, al cual la mayoría de estudiantes no está acostumbrados y tendrían que investigar cómo lograr algo similar, por eso pienso que usar Python es la mejor decisión para poder usar el conocimiento previo y enfocarse en lo nuevo que hay que aprender.

Memes





