**Documentación proyecto de análisis**

**Web IDE**

Sebastian Noreña Meglan - 1701313412

Nicolas Rivera Jaramillo - 1701313965

Universidad de Caldas

**Abril 2019**

# 

[**Documentacion Web IDE**](#_njoceq9gt9zs) **2**

[1. Analizador](#_ilnnhvkcqr67) 2

[1.1 Lexico](#_6oanrmk1icuy) 2

[1.2 Sintáctico](#_kly6qxpa2na3) 2

[1.3 Semántico](#_eefynnhrcbo7) 3

[2. Interfaz grafica](#_wvgswpf7cy8o) 3

[2.1 Detalles tecnicos](#_rtd7mlrmd80j) 3

[2.2 Explicación de navegación.](#_3itholkernjy) 3

[3. Que se hizo](#_5dbyth4d9q1g) 12

[4. Que no se completo.](#_d2149ngl1t5f) 12

[4.1 Porque no se hizo.](#_9zoev5pgc1ph) 12

[5. Experiencias](#_281lgukxl597) 13

[6. Anexos](#_1aszjm9lcdje) 13

# 

# Documentacion Web IDE

## 1. Analizador

### 1.1 Lexico

En el analizador léxico se ingresa lo básico del lenguaje, que serán los **tokens** y las **palabras reservadas.**

Las palabras reservadas son como su nombre lo indica, palabras especiales para el lenguaje que sirven como indicativo en el programa; por ejemplo, en el IF de un programa se reconoce la palabra clave IF, de tal forma que se sabe que si se encuentra esa palabra, es porque se quiere aplicar aquel método.

Los tokens, a diferencia de las palabras reservadas, pueden contener un valor diferente a solo su nombre, o incluso más de un valor, además de que su valor puede cambiar dependiendo al caso; por ejemplo, se pueden crear tokens por cada signo de operaciones matemáticas (+, -, \*, /), pero también un token que representa nombres de variables, pero si la variable que se identificó es igual a una palabra reservada, se tomará como una palabra reservada. Para la creación de los tokens se debe tener en cuenta el **lenguaje de expresiones regulares,** ya que para que ciertas letras y símbolos sean identificados como solo esa letra o símbolo, hay que saber si no significan otra cosa como expresión regular.

### 1.2 Sintáctico

El analizador sintáctico contiene las combinaciones de **bloques** que se reconocerán en el lenguaje; estos bloques constan de tokens, palabras reservadas, e incluso otros bloques, de tal forma que se pueda no solo identificar si la combinación de palabras reservadas y tokens es correcta, sino que tambien, que significa la combinación de estas; por ejemplo, no solo puede identificar si un método IF está bien escrito, sino que además identifica que es un método IF.

Además, se especifica qué procedencia tienen las palabras clave y los tokens, en una lista de forma ordenada de menor a mayor importancia, de tal manera que tenga en cuenta ciertos elementos antes que otros; por ejemplo, las multiplicaciones y las divisiones, por leyes matemáticas, deben ser resueltas antes que las sumas y las restas, por lo cual las primeras tienen mayor importancia que las ultimas mencionadas.

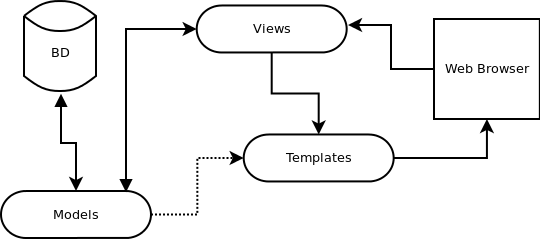
### 1.3 Semántico

(Está incluido dentro de la clase Sintáctico)**:** El analizador semántico se encarga de que los bloques en el analizador sintáctico manejen la información dentro de ellos; por ejemplo, si hay un bloque encargado de reconocer la suma de 2 números, el semántico se encarga de sumarlos, o si hay un bloque responsable de identificar métodos IF, se encarga de resolver el método, y lo que se encuentra dentro de él.

## 2. Interfaz grafica

### 2.1 Detalles tecnicos

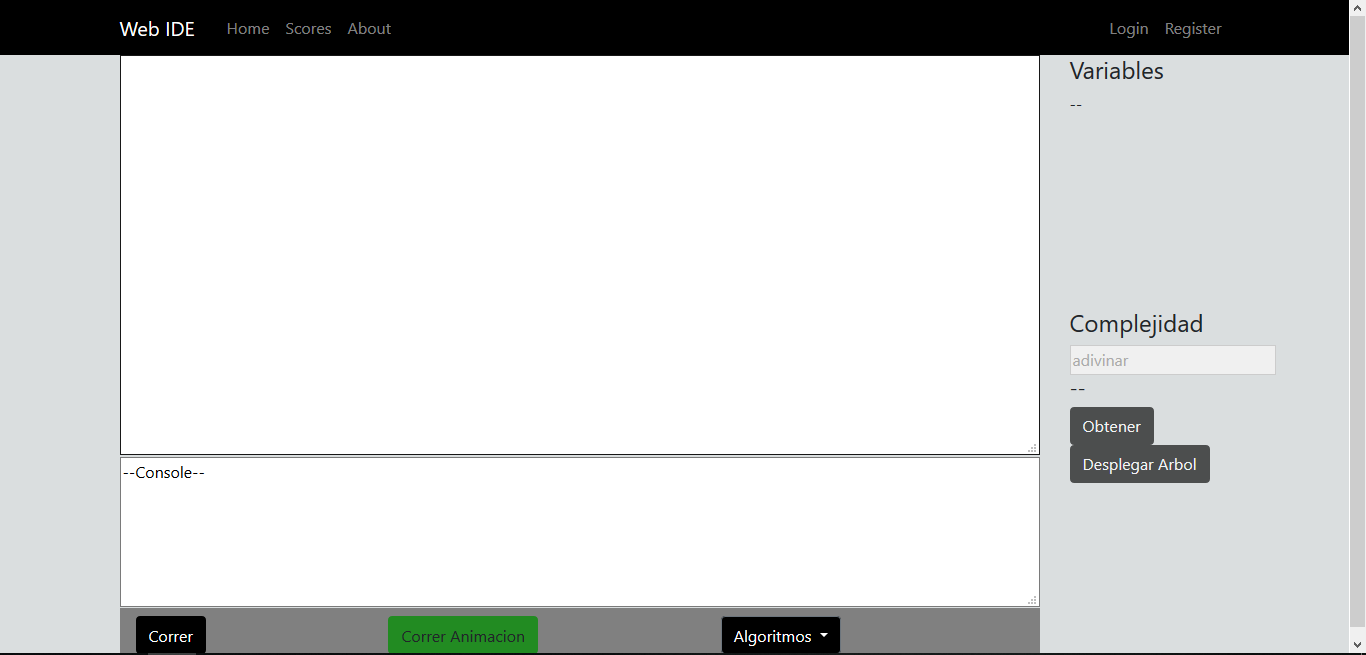
* La interfaz fue realizada en Django 2.2 que usa el modelo vista template, ilustrado en la siguiente imagen



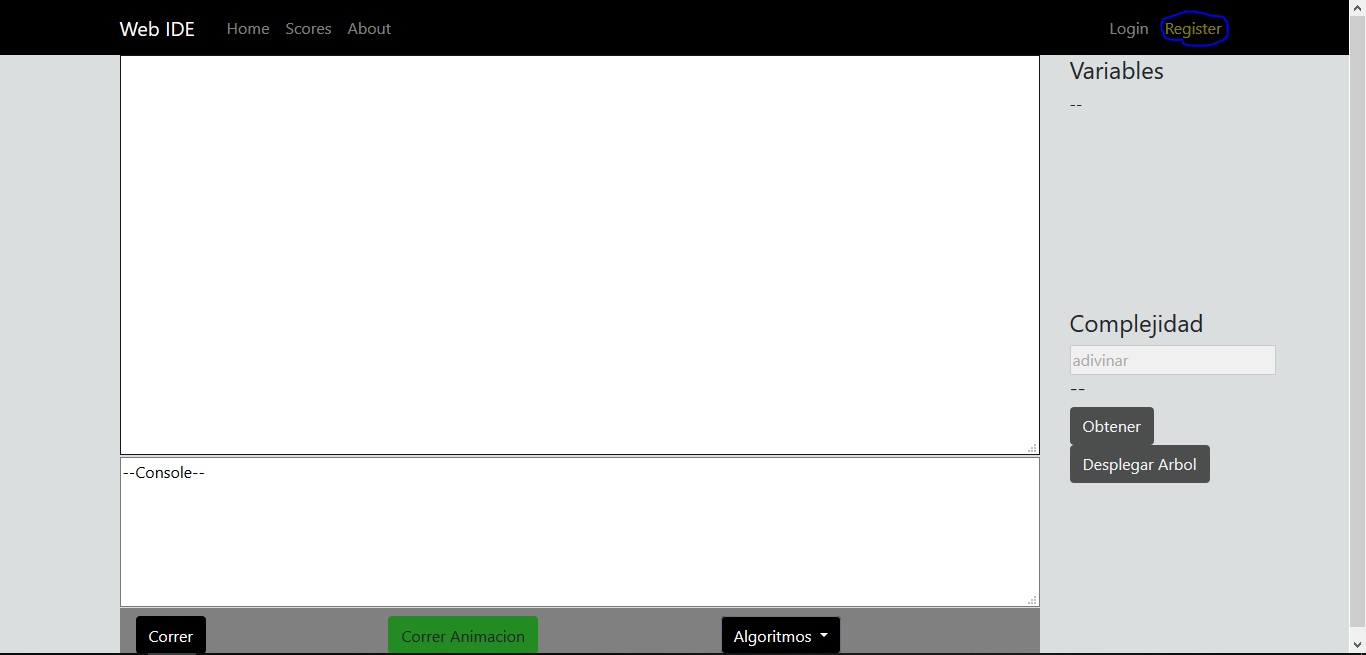
* Todas las peticiones de almacenamiento de base son llevados a través de POST con una funcionalidad de Django que llamada csrf\_token(cross site request forgery) que incluye cifrado de datos y previene ataques de suplantación.
* La base de datos usa sqlite3 que es un sistema gestor de bases de datos livianos, que generalmente se usa en dispositivos móviles.
* Para el estilo visual se utilizó Bootstrap 4 y la librería django crispy forms, entre otras que vienen por defecto en django.
* Para el soporte de imágenes se usó la librería Pillow que es compatible con Django.

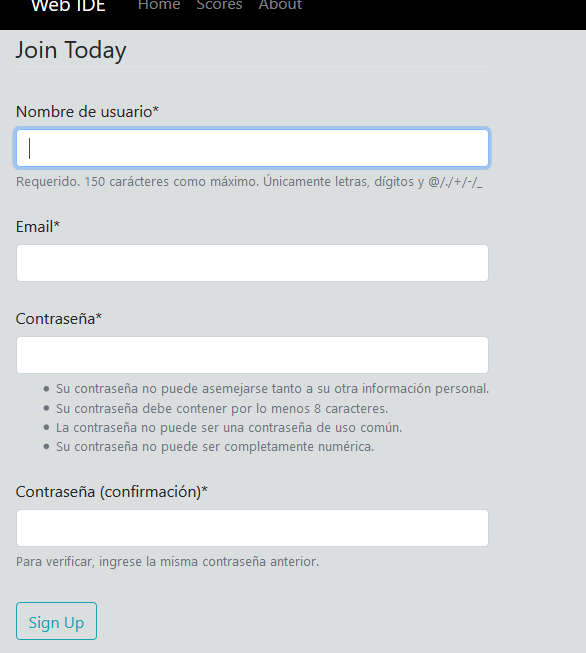
### 2.2 Explicación de navegación.

Aquí se puede ingresar nuestros algoritmos correrlos y ver su ejecución, pero la opción de guardado, entre otras opciones no están disponibles a menos que el usuario se loguee primero.

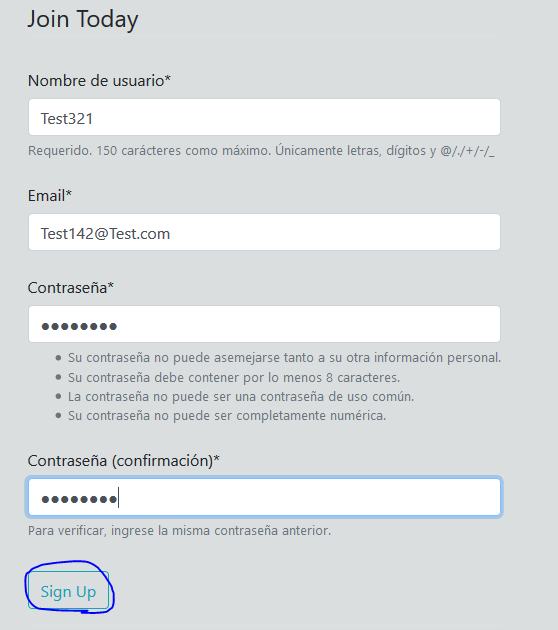


Para ello es necesario registrar en la pestaña disponible en la parte superior derecha.

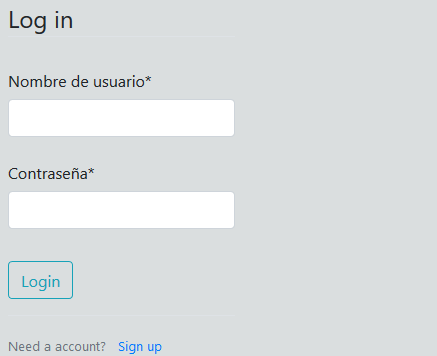




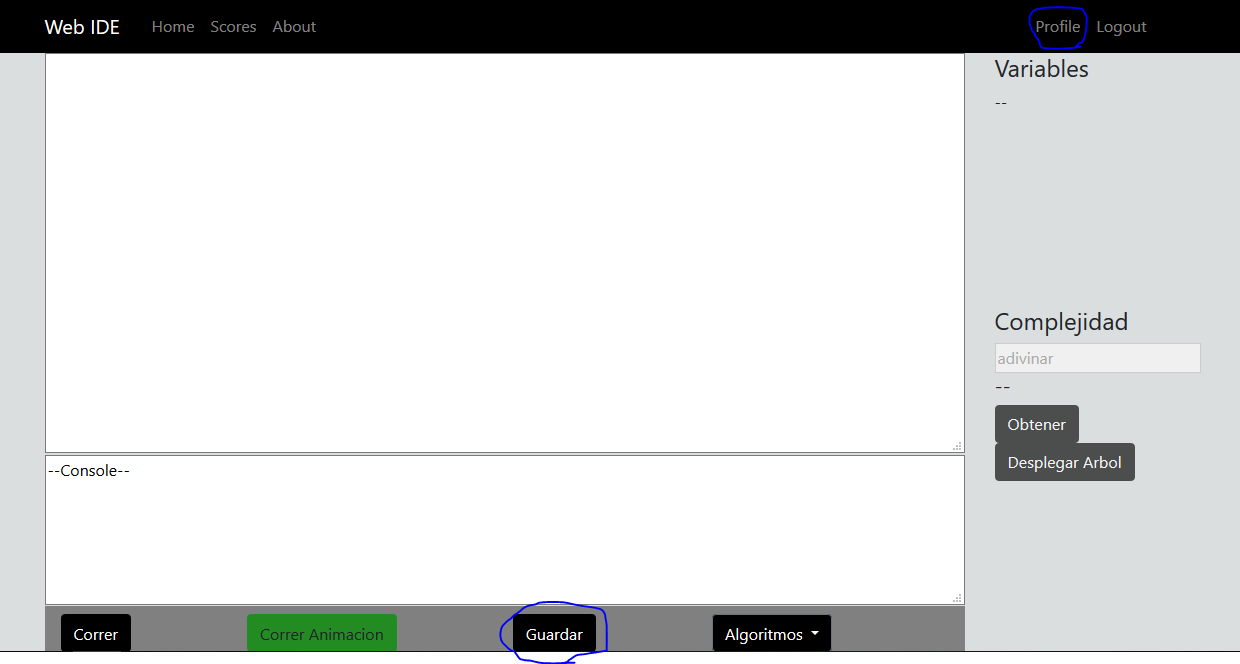
Rellenamos el formulario con los datos requeridos.



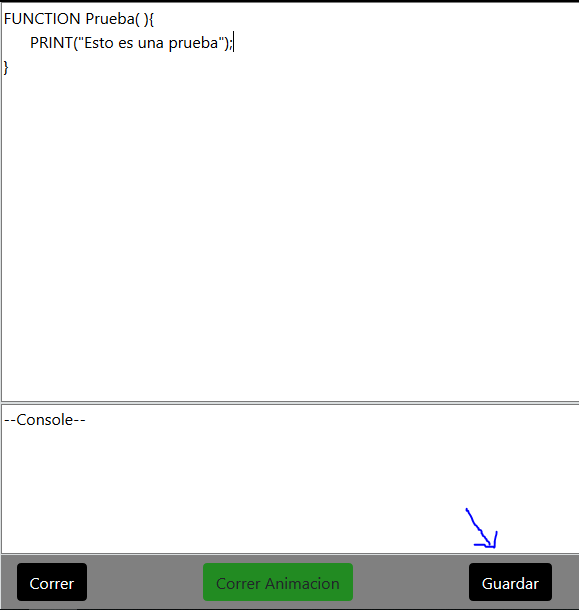
al hacer click en ‘Sign up’ esta nos redireccionará, a la página de login para poder ingresar.



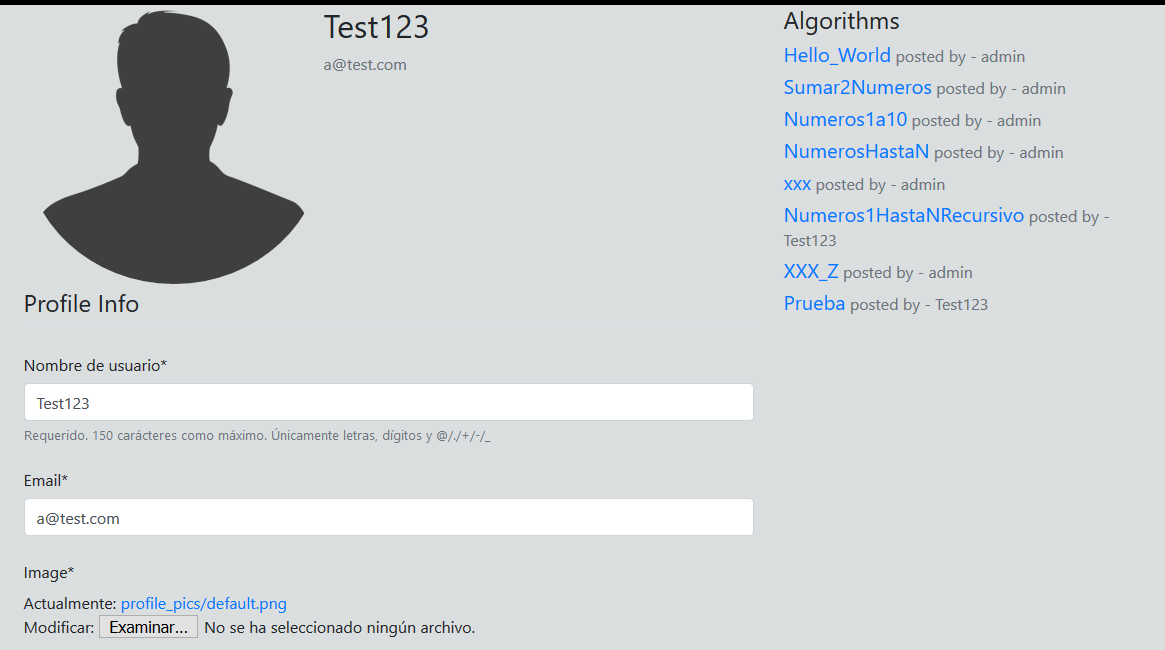
Una vez iniciemos sesión podremos observar que la interfaz ha cambiado. Ya podremos guardar algoritmos y visitar nuestro perfil.



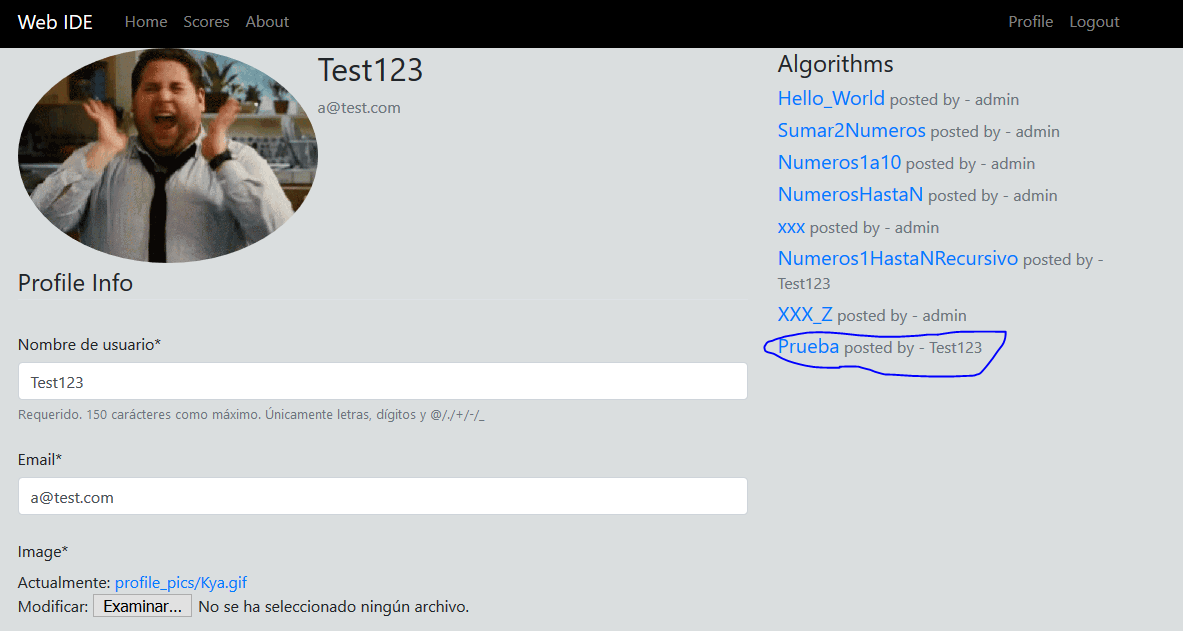
Por ahora guardaremos un algoritmo.

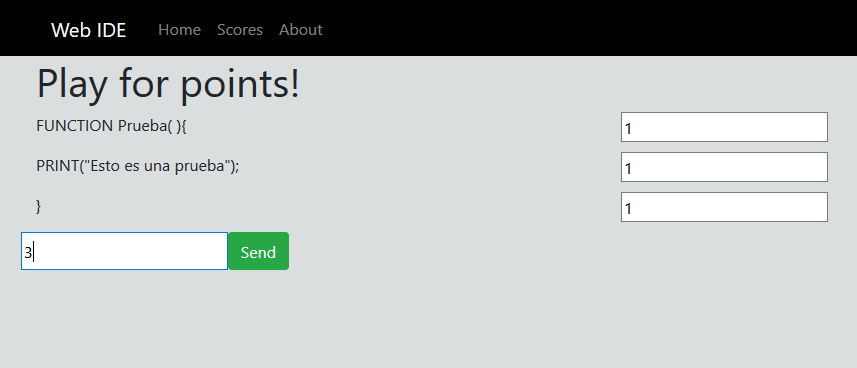


Este algoritmo lo podremos ver en más tarde en nuestro perfil.

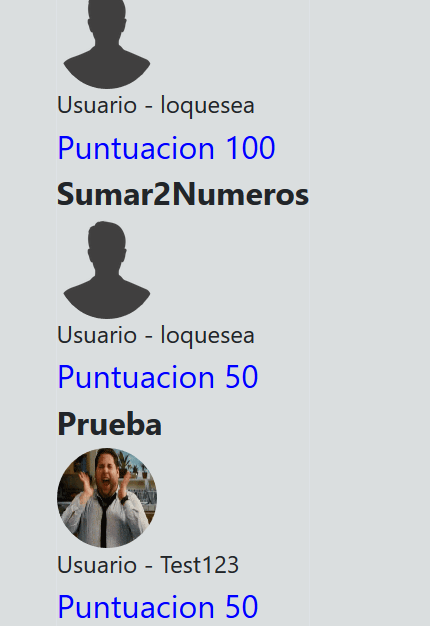


En nuestro perfil podemos ver nuestra información como a sí mismo actualizarla, además a nuestra derecha podremos ver los algoritmos que están disponibles y jugar a adivinar su complejidad.





Finalmente en la pestaña de scores podremos ver el resultado de dicho intento.



## 3. Que se hizo

* Diagrama de casos de uso.
* Modelo entidad relación.
* Arquitectura
* Documentacion de tokens.
* Interfaz web.
* Manejo de usuarios.
* Sistema de puntuaciones
* Almacenamiento de algoritmos y puntuaciones
* Analisis Lexico de lenguaje.
* Análisis Sintáctico de lenguaje

## 4. Que no se completo.

* Análisis semántico completo del lenguaje
* Animaciones
* Árbol de seguimiento
* Analizador de complejidad completo(Se hizo parcialmente)
* Juego interactivo donde el usuario adivina la complejidad del algoritmo

## 4.1 Porque no se hizo.

**Análisis semántico completo del lenguaje:** Al hacer la transición de analizar solo con lexico y sintactico a usar también semántico, se presentan, según la consola, errores de sintaxis, y si el programa de prueba es más complejo que una simple declaración de variable, también mencionó que el la clase que contiene el método no está definida, aún teniendo en cuenta que se usó el mismo código de pruebas antes y después de aplicar el semántico.

**Animaciones:** Al no haber semántico era un tanto sentido el preparar la animación ya que era un tanto ambiguo lo que dicha animación recibiría, aunque ciertamente era posible implementar una demo.

**Árbol de seguimiento:** Sin el semántico no hay mucho que seguir.

**Analizador de complejidad:** Este era un punto especialmente complejo, en principio si existe un intento de implementación de este, pero se quedó incompleto ya que solo puede saber para ciertos algoritmos y pues al no estar implementado el intérprete.

**Juego interactivo donde se adivina la complejidad del algoritmo:** Depende del semántico

## 5. Experiencias

Python es un lenguaje de programación que gracias a ser de tipado dinámico y de tener una sintaxis que favorece un código legible, es fácil de aprender, a menos de que se esté acostumbrado a programas con una forma de sintaxis muy diferente, como por ejemplo java, lo que puede causar confusión en la realización de ciertos métodos.

PLY es una librería de python con la que se puede crear un analizador lexico y sintactico sin muchos problemas de por medio, ya que solo requiere saber qué palabras clave, tokens(y a que es igual cada token), y como están conformados los bloques para poder realizar estos 2 analizadores; pero la documentación existente de la librería no es lo suficientemente amplia para que la implementación del analizador semántico no cause muchos problemas, ya que no se tiene muy claro cómo se construyen las clases que van a ser llamadas desde el sintáctico,como ejecutar los bloques de una misma línea y qué valor devuelve si se tiene por ejemplo un FOR, un IF, o algún método que pueda producir más de una nueva variable o más de una modificación de variable.

Django es un framework que puede llegar a ser un tanto complejo, sobretodo porque no es tan intuitivo como otros frameworks, ya que requiere la modificación de bastantes archivos de trabajo, sin contar que Python al ser un lenguaje débilmente tipado y esto presenta algunos problemas de claridad y esto se nota más en desarrollos con frameworks como django, ya que por ejemplo al momento de utilizar estos métodos prediseñados no queda muy claro qué tipos datos recibe dicho método, lo que hace más frecuente recurrir a documentación.

## 6. Anexos

Se incluyen:

* Carpetas con diagramas
* Xls con los tokens
* Archivo del programa de Django y el analizador
  + **Nota**: El analizador se encuentra en Proyecto WebIDE\App\_IDE\logic\analizador
  + **Nota 2:** Para ejecutar el proyecto son necesarias las siguientes librerías.
    - django
    - django-crispy-forms
    - pillow
    - ply
  + **Nota 3:** Estas librerías se pueden descargar con el comando ‘pip install <nombre>’