



Universidad Francisco Marroquín
Facultad de Ciencias Economicas
Computer Science
Algoritmos y Complejidad
Prof. Ernesto Rodríguez - erodriguez@unis.edu.gt
Aux. Juan Roberto Alvarado

Proyecto Final – Code Crawler

Fecha de entrega: 22 de Noviembre, 2018 - 11:59pm

Instrucciones: Resolver cada uno de los ejercicios siguiendo sus respectivas instrucciones. El trabajo debe ser entregado a través de Github, en su repositorio del curso, colocado en una carpeta llamada "Proyecto Final". Al menos que la pregunta indique diferente, todas las respuestas a preguntas escritas deben presentarse en un documento formato pdf, el cual haya sido generado mediante Latex.

1 Descripción General

El objetivo de este proyecto es aplicar algunos de los algoritmos aprendidos en clase en una situación de la vida real. Su tarea es implementar un *crawler* que permita indexar y buscar objetos en código **Javascript** (es6), similar al programa Hoogle.

Para ello, su programa debe inspeccionar código fuente Javascript y utilizar las expresiones **require** para ubicar los demás archivos y construir un índice de objetos (métodos, constantes, funciones, clases, etc.). Aparte el programa debe proveer una interfaz (gráfica, web, línea de comandos, rest, etc.) para buscar objetos. Usted debe utilizar su creatividad para diseñar la interfaz de tal forma que permita filtrar los resultados de formas útiles. Por ejemplo, hacer una búsqueda solamente de clases que hereden de la clase "Foo". Para todo objeto indexado por su crawler, se debe construir **el árbol de búsqueda** que se utilizó para llegar al objeto a partir del archivo origen.

2 Evaluación

El proceso de evaluación consistirá de los siguientes pasos:

1. Se le proveerá a su programa un archivo Javascript para que construya el índice. Su programa debe buscar enunciados **require** para descubrir los archivos restantes.
2. Luego de terminar el proceso de indexado, se probará el buscador mediante algunas búsquedas de prueba. Por ejemplo:
 - Buscar las clases con nombre similar a "Foo"
 - Buscar los métodos que acepten 2 parámetros
 - Buscar las clases que heredan de "Bar"

3. Por ultimo, se verificara que para cada resultado de la busqueda, se haya construido el correspondiente **arbol de busqueda**

Tambien se llevara a cabo una competencia contra los demas compañeros. Se utilizara un proyecto Javascript particular, el cual sera indexado y se realizaran 10 busquedas (no se revelaran exactamente cuales) en el codigo de dicho proyecto. Los trabajos que mejor localizan los objetos buscados seran los ganadores de la competencia. Se premiaran de la siguiente manera:

1. **Primer lugar:** 5 puntos extra en el curso
2. **Segundo lugar:** 3 puntos extra en el curso
3. **Tercer lugar:** 1 punto extra en el curso

3 Recomendaciones

Para facilitar el procesamiento del codigo fuente, se recomienda procesar los archivos Javascript con un **lexer** ya existente. Algunos de los existentes son:

1. Pygments (Python): <https://bitbucket.org/birkenfeld/pygments-main/src/default/pygments/lexers/javascript.py>
2. v8 (C++): <https://v8.dev/>
3. Language Javascript (Haskell): <https://hackage.haskell.org/package/language-javascript>
4. Nashorn (Java): <https://openjdk.java.net/projects/nashorn/>
5. Rhino (Java): <https://github.com/mozilla/rhino>

4 Entrega

El trabajo se debe entregar a traves de **Github**. Por favor entregar un proyecto de calidad profesional tal como lo espera la industria y academia de tecnologia esto incluye:

- Asegurarse que el programa pueda ser compilado en la plataforma para la cual fue escrito. Idealmente si puede compilarse en cualquier plataforma
- Instrucciones detallando como compilar y utilizar el programa
- Utilizar un sistema de compilación: Make, CMake, Ant, Maven, Pip, etc.
- Si la plataforma lo permite, hacerlo un paquete. Ejemplo: NPM o Pip
- Detallar los requisitos del sistema para compilar y ejecutar el programa
- Ejemplos y pruebas unitarias