Universidad Nacional Autónoma de México
Facultad de Ingeniería

Primera Entrega Compilador

Equipo: Wombat

Integrantes:

Romero Andrade Cristian	Arquitecto
Arguelles Macosay Mariana	Integradora
Rocha García Erick Hazel	PMP

Entrega: 16 de septiembre del 2019

1. Introducción

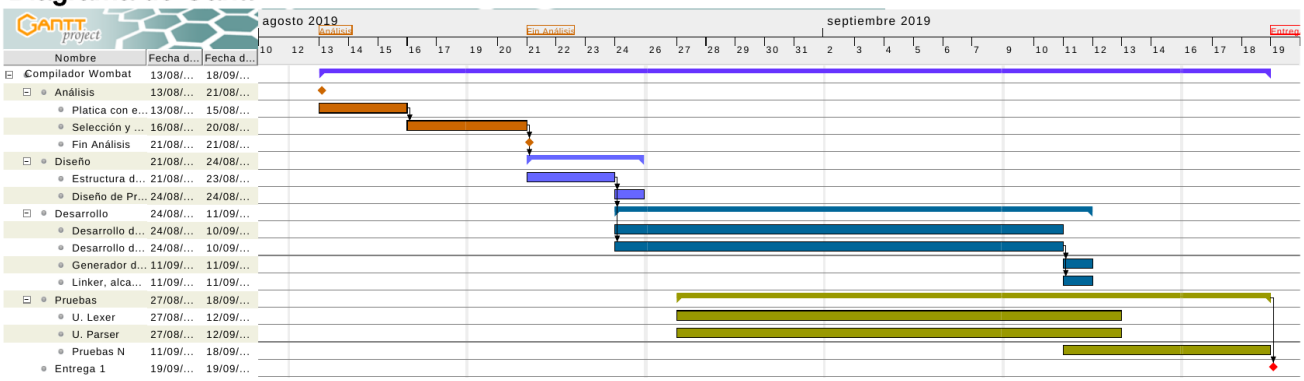
2. Plan de Proyecto

Compilador Wombat 17/09/2019

Tarea 2

Nombre	Fecha de inicio	Fecha de fin
Compilador Wombat	13/08/19	18/09/19
Análisis	13/08/19	21/08/19
Platica con el Stakeholder	13/08/19	15/08/19
Selección y estudio de herramientas	16/08/19	20/08/19
Fin Análisis	21/08/19	21/08/19
Diseño	21/08/19	24/08/19
Estructura de Compilador	21/08/19	23/08/19
Diseño de Pruebas	24/08/19	24/08/19
Desarrollo	24/08/19	11/09/19
Desarrollo del Lexer, alcance 1	24/08/19	10/09/19
Desarrollo del Análizador, alcance 1	24/08/19	10/09/19
Generador de Código ASM, alcance 1	11/09/19	11/09/19
Linker, alcance 1	11/09/19	11/09/19
Pruebas	27/08/19	18/09/19
U. Lexer	27/08/19	12/09/19
U. Parser	27/08/19	12/09/19
Pruebas N	11/09/19	18/09/19
Entrega 1	19/09/19	19/09/19

Diagrama de Gantt 4



3. Arquitectura del Compilador

La arquitectura del compilador es simple utilizado el gestor de proyectos de *elixir mix*, la cual genera las herramientas suficientes para las pruebas y compilación. La siguiente estructura que se decidió fue la siguiente:

- _build
- config
- ejemplo
  - return_2.c
- lib
  - compiladorwombat.ex
  - wc2
    - analizador.ex
    - arbol.ex
    - code_gen.tex
    - lexer.ex
    - linker.ex
  - Makefile
  - miscelanea
  - mix.exs
  - mix.lock
  - README.md
  - test
    - compilador_wombat_test.exs
    - stage_1[1]
    - test_helper.exs

Ya que así se tiene separado las etapas del compilador y es fácil extender y dar mantenimiento a dicha estructura tomando a todos los archivos necesarios de manera independiente.

4. Plan de Pruebas y de "Suites"

El plan consiste en pasado un tiempo de desarrollo de los módulos del compilador, se comenzará las pruebas mediante una simple regla, "Lo que se debe de recibir, lo que se debe de mandar", por ejemplo en el caso del lexer, lo que este recibe es una lista de cadenas de caracteres ["int", "main", etc] y lo que debe de mandar en una lista y/o tupla de *tokens* [**:kint**, **:kmain**, etc]. Ya completado esta "etapa" de pruebas, se comenzará con las pruebas de **Nora**, la cual se espera solo tener errores en sentencias especificas conservando el funcionamiento original.

5. Conclusiones

[1] Pruebas de Nora