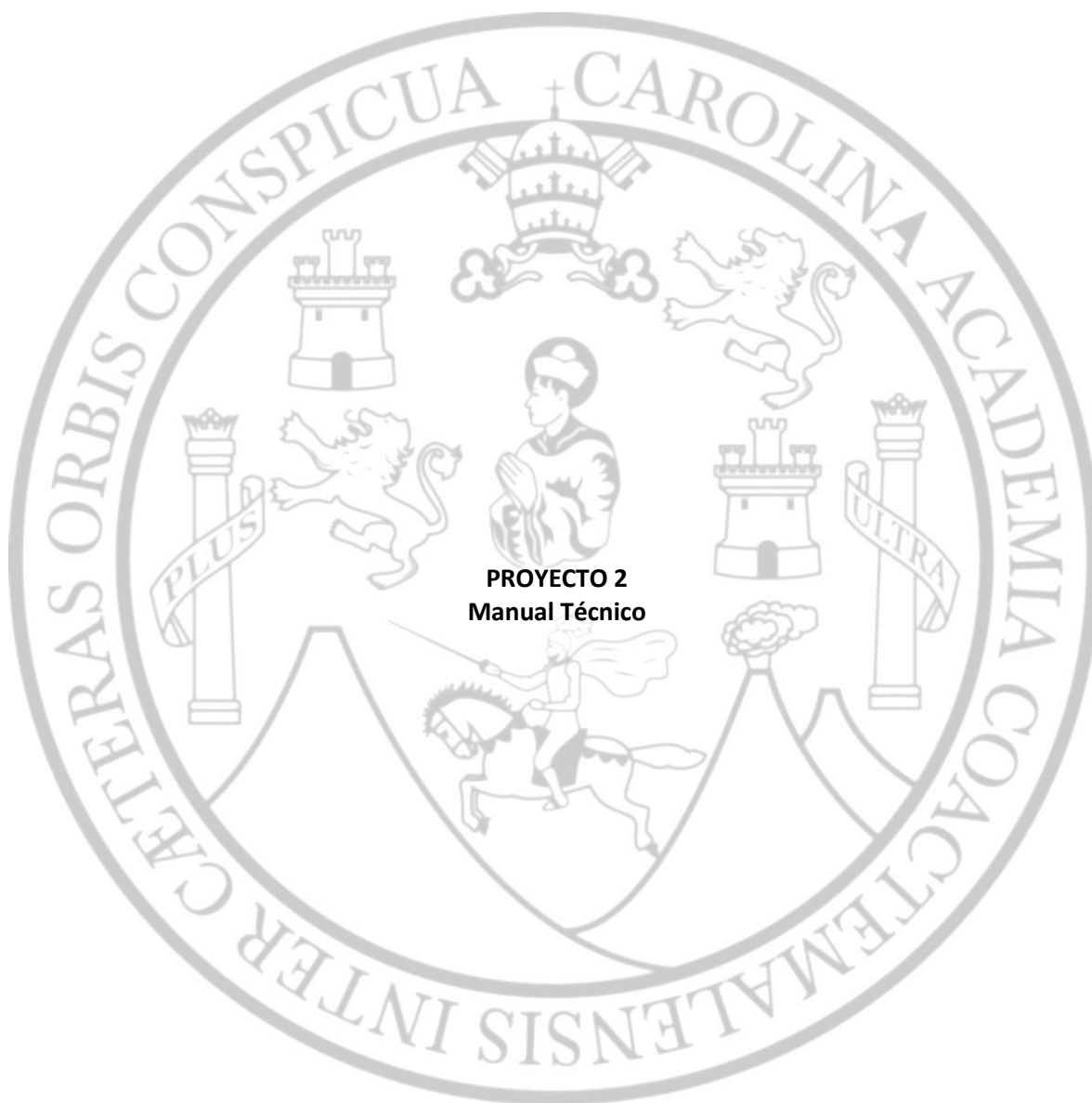


UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERIA
ESCUELA DE CIENCIAS Y SISTEMAS



PROYECTO 2
Manual Técnico

RAFAEL ANGEL CHOCOJ XINICO
201314826
PRIMER SEMESTRE 2020

Requerimientos De Entorno de Programación

- Javascript
- Typescript
- Go
- NodeJs
- Css
- Html
- Jison

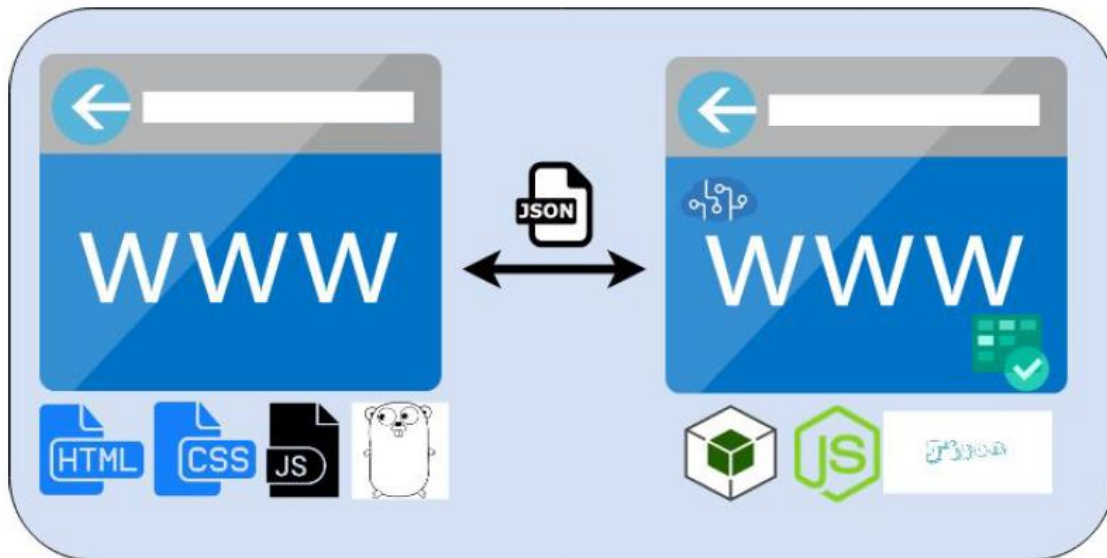
Arquitectura

Frontend (Vista Cliente)

Desarrollado mediante el uso de javascript, css y html para la construcción de una página lanzada mediante un servidor http, el cual utiliza lenguaje Go.

Backend (Servidor Analizador)

Desarrollado utilizando nodejs con javascript para levantar un servidor que recibire peticiones REST y realizara el analisis lexico, sintactico y la detección de copias entre 2 proyectos desarrollados en java de manera que se obtengan reportes sobre el grado de similitud de los mismos.



Gramática utilizada

```
INICIO
INICIO : IN_CLASE EOF
IN_CLASE : CLASE      ;

CLASE : IMPORTS 'class' identificador '{' LIS_INSTRU_CLASS '}'
      | 'class' identificador '{' LIS_INSTRU_CLASS '}'
      | error
      ;
IMPORTS : IMPORTS 'import' identificador ';'
        | 'import' identificador ';'
        ;

LIS_INSTRU_CLASS : LIS_INSTRU_CLASS INSTRU_CLASS
                  | INSTRU_CLASS
                  ;
INSTRU_CLASS : DECLARACION
              | MAIN
              | METODO
              | FUNCION
              ;

LIS_INSTRUCCIONES : LIS_INSTRUCCIONES INSTRUCCION
                   | INSTRUCCION
                   ;

INSTRUCCION : DECLARACION
             | ASIGNACION
             | PRINT
             | WHILE
             | DOWHILE
             | FOR
             | IF
             | SWITCH
             | LLAMADA_FUN_ALONE
             /*solo para bucles*/
             | 'continue' ';'
             | 'break' ';'
             /*solo para main y metodos*/
             | 'return' ';'
             | 'return' EX ';'
             ;
/*lista de variables*/
DECLARACION : TIPO LISTA_ID '=' EX ';'
;
```

```

        | TIPO LISTA_ID ';'
        ;

LISTA_ID : LISTA_ID ',' identificador
        | identificador
        ;

/*sentencias inicio*/
WHILE : 'while' CONDICION BLOQUE_LISINSTRUCCIONES
        ;
BLOQUE_LISINSTRUCCIONES : '{' LIS_INSTRUCCIONES '}'
        | '{' '}'
        ;

CASE_LISINSTRUCCIONES : LIS_INSTRUCCIONES
        |
        ;
CONDICION : '(' EX ')'
        ;

CON_FOR : EX
        ;

DOWHILE : 'do' BLOQUE_LISINSTRUCCIONES 'while' CONDICION ';'
        ;

FOR : 'for' '(' INIFOR CON_FOR ';' EX ')' BLOQUE_LISINSTRUCCIONES
        ;
INIFOR : DECLARACION
        | ASIGNACION
        ;

IF : 'if' CONDICION BLOQUE_LISINSTRUCCIONES
    | 'if' CONDICION BLOQUE_LISINSTRUCCIONES 'else' BLOQUE_LISINSTRUCCIONES
    | 'if' CONDICION BLOQUE_LISINSTRUCCIONES 'else' IF
    ;

SWITCH : 'switch' CONDICION '{' LIS_CASES C_DEFAULT '}'
        | 'switch' CONDICION '{' C_DEFAULT '}'
        ;

LIS_CASES : LIS_CASES CASE
        | CASE
        ;
CASE : 'case' EX ':' CASE_LISINSTRUCCIONES

```

```

        ;
C_DEFAULT : 'default' ':' CASE_LISINSTRUCCIONES
        |
        ;
/*fin sentencias*/
TIPO : 'int'
      | 'double'
      | 'boolean'
      | 'char'
      | 'String'
      ;

ASIGNACION : identificador '=' EX ';'
          ;

PRINT : 'System' '.' 'out' '.' 'print' '(' EX ')' ';'
      | 'System' '.' 'out' '.' 'println' '(' EX ')' ';'
      ;
/*inicio funciones, eventos*/
MAIN : 'void' 'main' '(' ')' BLOQUE_LISINSTRUCCIONES
      ;
METODO : 'void' identificador '(' LIS_PARAMETROS ')' BLOQUE_LISINSTRUCCIONES
        | 'void' identificador '(' ')' BLOQUE_LISINSTRUCCIONES
        ;

FUNCION : TIPO identificador '(' LIS_PARAMETROS ')' BLOQUE_LISINSTRUCCIONES
        | TIPO identificador '(' ')' BLOQUE_LISINSTRUCCIONES
        ;

LIS_PARAMETROS : LIS_PARAMETROS ',' TIPO identificador
               | TIPO identificador
               ;

LIS_PARCALL : LIS_PARCALL ',' EX
            | EX
            ;

LLAMADA_FUN : 'identificador' '(' LIS_PARCALL ')'
            | 'identificador' '(' ')'
            ;

LLAMADA_FUN_ALONE : 'identificador' '(' LIS_PARCALL ')' ';'
                  | 'identificador' '(' ')' ';'
                  ;

```

```
EX : '-' EX %prec
    | '!' EX

    | EX '++'
    | EX '--'

    | EX '+' EX
    | EX '-' EX
    | EX '*' EX
    | EX '/' EX
    | EX '^' EX
    | EX 'mod' EX

    | EX '<' EX
    | EX '>' EX
    | EX '>=' EX
    | EX '<=' EX
    | EX '==' EX
    | EX '!=' EX
    | EX '||' EX
    | EX '&&' EX
    | 'entero'
    | 'decimal'
    | 'true'
    | 'false'
    | 'cadena'
    | 'char'
    | 'identificador'
    | LLAMADA_FUN
    | '(' EX ')'
;
```