

Universidad San Carlos de Guatemala

Centro Universitario de Occidente

Teoría de sistemas 2

Grupo 4



Manual Técnico - Proyecto 2

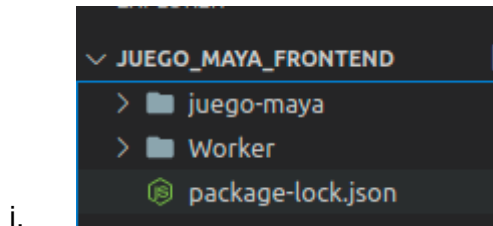
Requerimientos mínimos y herramientas a instalar:

- Sistema operativo:
 - Windows 10.
 - Linux.
 - Ubuntu 16.04 LTS en adelante.
- Herramientas necesarias: las siguientes versiones son las recomendadas para evitar todo tipo de conflictos de dependencias.
 - Angular 13.
 - npm 6.
 - Node 12.
 - Express.
 - MySQL 8.
 - git.
- IDE's: para tener un uso más cómodo se recomienda tener instalado el siguiente IDE:
 - Visual Studio Code.
- Espacio de almacenamiento:
 - 300 mb libres para la app.
 - 700 mb para VSCode.
 - Si se tiene más espacio es mejor, dado que Angular descarga muchos recursos.

- RAM:
 - 4 gb mínimo.

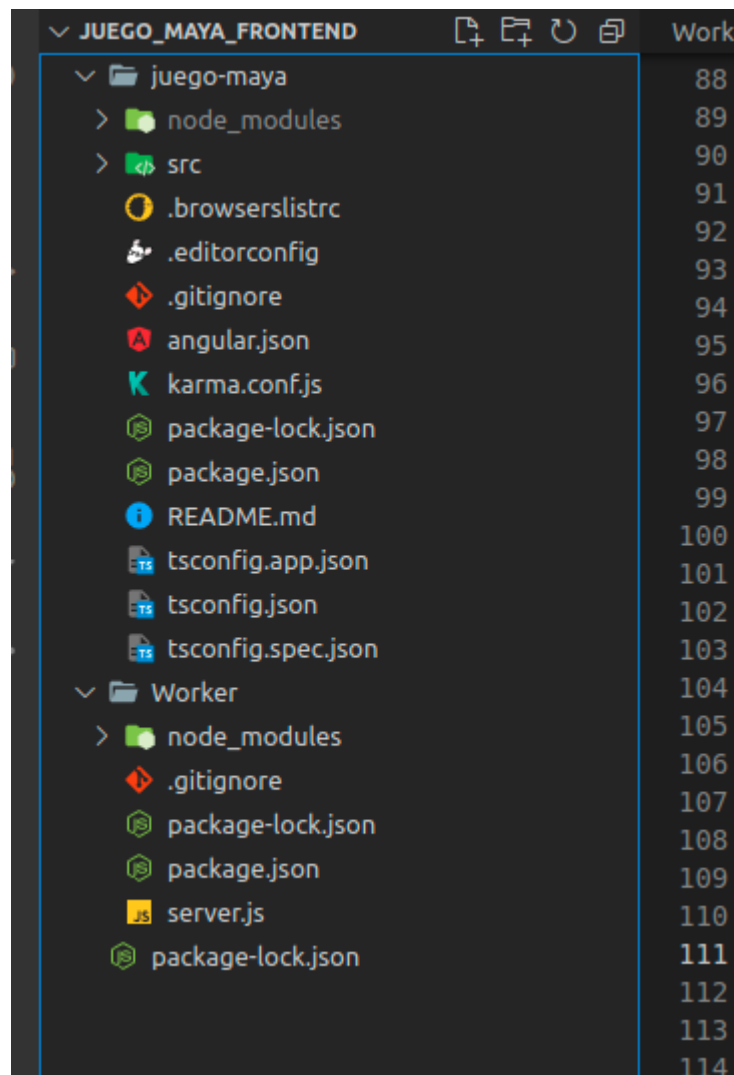
Descarga de la aplicación y pasos para la instalación:

1. Descargar la aplicación como un zip o clonar el repositorio con git desde el repositorio: https://github.com/jamesg19/JUEGO_MAYA_FRONTEND/tree/main
 - a. La carpeta quedaría de la siguiente forma:



2. Antes de utilizar la aplicación, es necesario ejecutar el script de extensión “.sql” dentro del repositorio.
3. Cuando la base de datos ya está instalado, procedemos a descargar las dependencias:
 - a. En una terminal dentro de la carpeta “juego-maya” ejecutamos el código siguiente:
 - i. `npm install`
 - b. En una terminal dentro de la carpeta “Worker” ejecutamos el siguiente código:
 - i. `npm install`

- c. Esto descargará todas las dependencias y recursos a utilizar, bajo el nombre de “node_modules”, esto para los 2 proyectos.



Modo de ejecución del software y los servicios:

Una vez instalado todo lo necesario para la ejecución del programa, ejecutamos los siguientes comandos:

1. Dentro de la carpeta “juego-maya”:
 - a. `ng serve --open`
2. Dentro de la carpeta “Worker”:
 - a. `node server.js`
3. El programa está funcionando de forma eficiente.

Explicación algoritmo de transformación de números decimales a maya y viceversa:

- Número maya a decimal: "baseNToDecimal"
 - Función que transforma un número en base N a sistema decimal (base 10), recibe los parámetros:
 - numeros: cadena de números separados por coma.
 - base: base N en la que está el número ingresado.
 - Ejemplo:
 - baseNToDecimal("15 7 16 3", 20) → Entrada.
 - El número ingresado está en base 20 y transformado a decimal produce "123123".

```
function baseNToDecimal(numeros, base){
  let numeroResultate = 0; //variable que contiene el valor final
  const listadoNumeros = numeros.toString().split(" "); //separamos
  por espacios

  for(let i = 0; i < listadoNumeros.length; i++){ //recorremos cada
  numero
    let aux = listadoNumeros[i]; //obtenemos el numero actual
    let nivel = (listadoNumeros.length - i - 1); //obtenemos el
  nivel o posicion de derecha a izquierda
    numeroResultate += aux * Math.pow(base, nivel);

    /*
      el nivel es la posicion en la que se encuentra el numero
      por ejemplo, 1 10, 1 está en el nivel 1, mientras que el 10
  está en la posicion 0
      este nivel se usa para calcular el valor real del numero en
  esa posicion

      Ej: 1 * base^1 + 10 * base^0, vemos que mientras mas alto el
  nivel, mas alto el valor
      de cada numero
    */
  }

  return numeroResultate; //retornamos el valor
}
```

- Número decimal a maya: "baseNToDecimal"
 - Función que transforma un número en decimal (base 10) a sistema base N (o maya):
 - numeros: cadena/número en sistema decimal.
 - base: base N en la que se quiere el número de salida esperado.
 - Ejemplo:
 - baseNToDecimal("123123", 20) → Entrada.
 - El número ingresado esta en decimal es "123123", se espera que el número resultado sea en base 20, dando como resultado "15 7 16 3".
 - **Nota:** en sistema maya, esos números van de arriba hacia abajo.

```
function decimalToBaseN(numeros, base){
  let valor = ""; //variable que contiene el valor final, poseera
  varios caracteres separados por espacio

  let resultado = Math.trunc(numeros / base); //el resultado es igual
  al numero dividido la base
  let residuo = numeros % base; //se calcula si tiene residuo

  if(resultado > 0){ //si el resultado es mayor a 0, significa que se
  agrega otro nivel/digito
    return decimalToBaseN(resultado, base).concat(" ", residuo); // +
  " " + residuo; //concatenamos al valor recursivo el residuo
  }else{
    return valor.concat(residuo); //concatenamos el residuo
  }

  /*
  Ejemplo: 60, base 20
    se ingresa 60, resultado: 60/20 = 3  Residuo: 0
    se ingresa 3, resultado: 0          Residuo: 3
    se retorna 3
    se retorna "3" concatenado a 0
    Salida "3 0"

  */
}
```