



INSTITUTO TECNOLÓGICO DE COSTA RICA  
ESCUELA DE COMPUTACIÓN

---

## Tarea 2 de Sistema Operativos

---

Semestre II

*Realizado por:*  
Jonathan Guillén 201261579

Mayo 2021

# Índice general

<b>1. Introducción</b>	<b>1</b>
<b>2. Ambiente de Desarrollo</b>	<b>2</b>
2.1. Lenguaje . . . . .	2
2.2. Ambiente . . . . .	2
2.3. Manejo de versiones . . . . .	2
<b>3. Estructuras de datos usadas y funciones</b>	<b>3</b>
<b>4. Instrucciones para ejecutar el programa</b>	<b>4</b>
4.1. Pre-requisitos . . . . .	4
4.2. Compilación . . . . .	4
4.3. Ejecución . . . . .	4
<b>5. Actividades realizadas por estudiante</b>	<b>5</b>
<b>6. Autoevaluación</b>	<b>6</b>
<b>7. Lecciones Aprendidas</b>	<b>7</b>

# Capítulo 1

## Introducción

Frogger (Furoggā) es un juego de arcade desarrollado por Konami y originalmente publicado por Sega. El objetivo del juego es guiar a las ranas a sus hogares, una por una, cruzando carreteras y ríos con muchos peligros. Este juego ha encontrado su lugar en la cultura popular, incluyendo la televisión y la música. El juego originalmente se iba a llamar “Highway Crossing Frog”, pero los ejecutivos de Sega pensaron que no capturaba la verdadera naturaleza del juego, así que fue cambiado a Frogger.

En esta tarea consiste realizar una versión jugable del clásico juego que corra al momento de encender la computadora, o por su defecto una máquina virtual .

# Capítulo 2

## Ambiente de Desarrollo

### 2.1. Lenguaje

El lenguaje usado para la tarea es Ensamblador, concretamente usando NASM.

### 2.2. Ambiente

El ambiente de desarrollo que se utilizo es SublimeText3, con un plugin que permite escribir código NASM, como si fuera un IDE.

### 2.3. Manejo de versiones

Para este proyecto se decidió el uso de github como herramienta para manejo de versiones. Para poder tener la tarea en cualquier maquina y ejecutar el proyecto. Se puede optar por descargar el proyecto desde la pagina o bien clonando el proyecto.

Para descargar el proyecto se puede hacer por medio de la pagina de Github, este [este link](#). Este le descargara el proyecto comprimido en un archivo .zip.

En caso que no se pueda o no se quiera descargar el proyecto desde la pagina se puede clonar el proyecto. Para esto se tiene que acceder a la terminal de linux, después moverse a la carpeta que quiere clonar el proyecto e ingresar el siguiente comando:

---

```
git clone  
git@github.com:JhonnyGuillen/TareaSistemasOperativos-Frogger.git
```

---

Listing 2.1: Link para descargar el proyecto.

# Capítulo 3

## Estructuras de datos usadas y funciones

Para esta tarea las funciones utilizadas fueron para el arranque del programa como sistema operativo. Tambien se utiliza una funcion para imprimir los mensajes de texto que se utilizan dentro del programa. Se pensaba utilizar otras funciones mas, pero por problemas en el desarrollo no se pudieron implementar.

# Capítulo 4

## Instrucciones para ejecutar el programa

### 4.1. Pre-requisitos

Para la presente tarea es necesario tener o bien una maquina física o virtual para poner que arranque usando la tarea como sistema de arranque. En el caso de las pruebas hechas se uso VirtualBox para utilizar una maquina virtual para las pruebas de la tarea.

Para poder utilizar la tarea en el arranque de la maquina virtual se tiene que seguir los siguientes pasos:

- Se crea una nueva maquina virtual en el virtualbox.
- Se selecciona que esta maquina virtual sea de tipo “Other” y como “Other/Unknown”, de ahí se prosigue el proceso normal para montar la máquina virtual.
- Cuando ya este creada la Máquina Virtual se accede a las configuraciones y se le va al apartado de almacenamiento.
- Se agrega un controlador de tipo floppy y se agrega la imagen de Frogger.img.

Ya con esto se procede a arrancar la máquina virtual para compilar.

### 4.2. Compilación

Para compilar la tarea se debe arrancar la maquina virtual creada en el pre-requisito, al momento de que esta arranque inicia el programa.

### 4.3. Ejecución

Para ejecutar el archivo compilado se debe usar el siguiente comando.

# Capítulo 5

## Actividades realizadas por estudiante

Cuadro 5.1: Tabla de Actividades.

<b>Dia</b>	<b>Actividad</b>	<b>Duración</b>
18 de marzo 2021	Instalación de plugin de NASM en SublimeTex3	2 Horas
22 de marzo 2021	Practica con el lenguaje y familiarización con la sintaxis	3 Horas
24 de marzo 2021	Implementación de los mensajes de la interfaz del juego	2 Horas
26 de marzo 2021	Búsqueda de manejo de arreglos en NASM	4 Horas
28 de marzo 2021	Continuación de búsqueda de manejo de arreglos	2 Horas
29 de marzo 2021	Búsqueda de comando de arranque en NASM	3 Horas
31 de marzo 2021	Volver a buscar como rehacer todas las funcionesdel código.	4 Horas
1 de abril 2021	Buscando como arreglar la tarea	5 Horas
2 de abril 2021	Desarrollo de la documentación de la Tarea	3 Horas
6 de abril 2021	Creación del repositorio de la tarea	1 Horas

# Capítulo 6

## Autoevaluación

La tarea cumple con la parte del arranque, pero al momento que no se pudo solucionar el problema de que al implementar el arranque dejaba de funcionar las demás funciones. El Código no está completo por la razón ya explicada.

Personalmente me pondría una nota de 30, porque en si no cumple la idea de la asignación, pero si me esforcé en hacer la tarea.



# Capítulo 7

## Lecciones Aprendidas

- Crear el repositorio desde el primer día de la asignación de la tarea, se me volvió a olvidar hacerlo el primer día.
- Investigar mas sobre el arranque del sistema, ya que por no querer investigar sobre otra forma de realizar el arranque no pude arreglar lo demás que ya había hecho y dejo de funcionar con el arranque.
- Planificar mejor los tiempos ya que afecto con el desarrollo de otro proyecto.