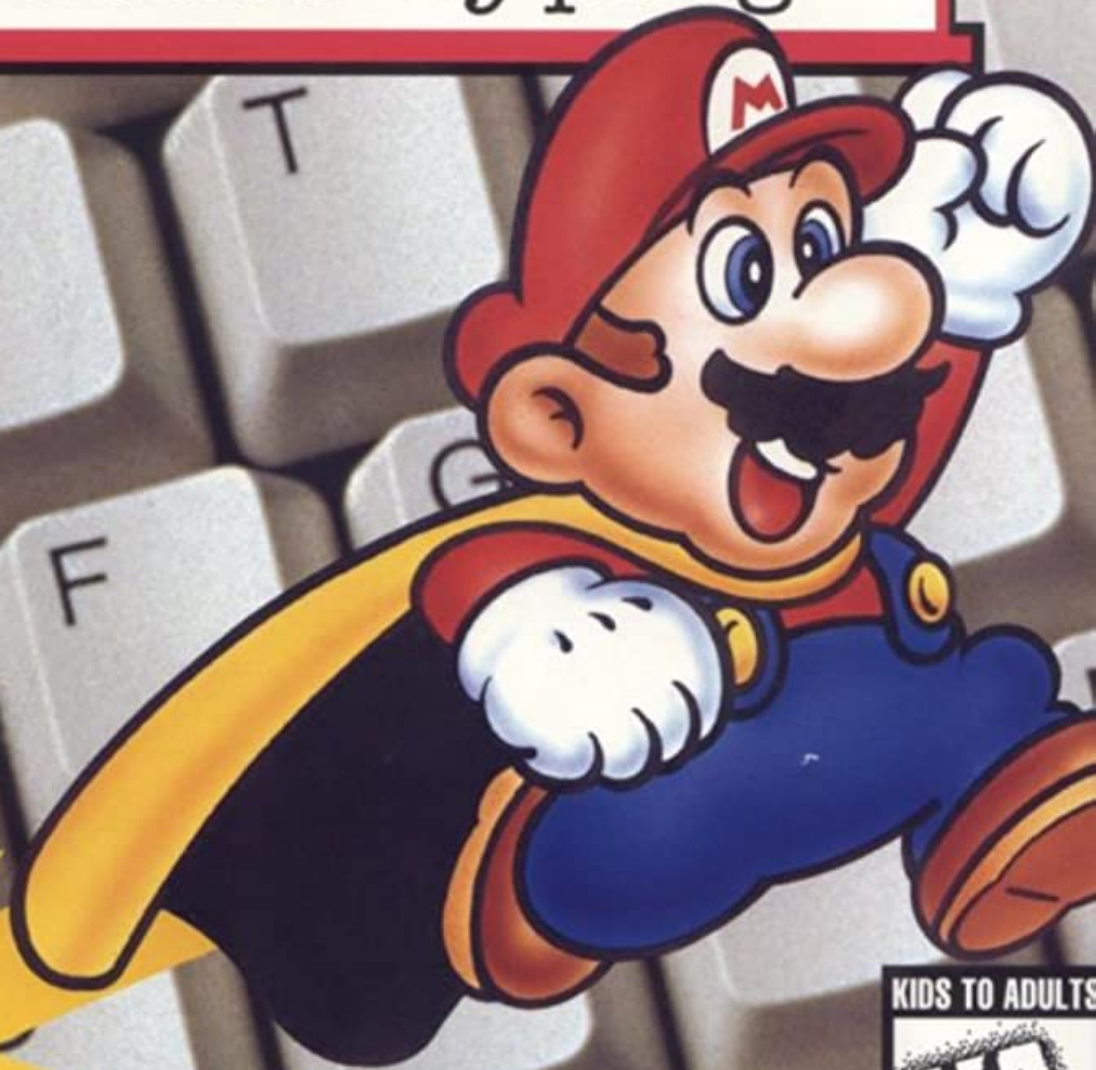


Tarea 2

# Tutormec

Agosto, 2022

MARIO  
Teaches Typing™



Interplay™  
BY GAMERS. FOR GAMERS.™

KIDS TO ADULTS  
K-A  
AGES 8 & UP

## Objetivo

Desarrollar una aplicación utilizando el lenguaje de programación Ensamblador, que permita comprender al estudiante el arranque de un Sistema Operativo.

## Datos Generales

- **Fecha de Entrega:**  
Martes 30 de Agosto de 2022  
antes de las 23:59:59 GMT-6.
- **Fecha de Revisión:**  
Miércoles 31 de Agosto de 2022.
- **Lenguaje:**  
Ensamblador para x86
- **Recurso Humano:**  
Individual
- **Valor de la asignación:** 5 %

## Profesor

Kevin Moraga  
kmoragas@ic-itcr.ac.cr  
Escuela de Computación

## Introducción

El teclado QWERTY es la distribución de teclado más común. Fue diseñado y patentado por Christopher Sholes en 1868 y vendido a Remington en 1873. Su nombre proviene de las primeras seis letras de su fila superior de teclas. La distribución QWERTY se diseñó con el propósito de lograr que las personas escribieran más rápido distribuyendo las letras de tal forma que se pudieran usar las dos manos para escribir la mayoría de las palabras. Otro de los objetivos principales era separar las letras más usadas de la zona central del teclado, para evitar que se atascaran las máquinas de escribir de primera generación, y aunque hoy en día no es necesario, se sigue utilizando mayoritariamente esta distribución.

# Requerimientos

## Requerimientos Funcionales

- Programar en ensamblador el booteo desde una unidad de USB
- Una vez que bootee desde el USB, cargará única y exclusivamente un programa llamado: Tutormec.

## Tutormec

Este programa consiste en un Tutor de Mecanografía, que se describe a continuación:

- Se le presenta al usuario una interfaz donde las letras que se tienen que presionar aparecen de derecha a izquierda, utilizando las primeras 10 líneas de la pantalla.
- En el centro de la pantalla se encuentran al menos 5 bloques, con una separación entre ellos, esta separación será utilizada para que “bajen” las letras.
- El usuario deberá de intentar “cachar” las letras cuando estas se encuentren por la separación de los bloques.
- Correrá una animación de la letra pasando por en medio de los bloques.
- En el lado inferior, se le deberá de indicar al usuario: cuál dedo debería de usar para poder “cachar” la letra.

## Requerimientos Técnicos

- El desarrollo se debe de realizar utilizando el lenguaje de programación definido en los datos generales.
- Se utilizará el BIOS con MBR (legacy mode) como mecanismo de booteo.
- La tarea deberá de funcionar en el laboratorio de la Universidad.

# Aspectos Administrativos

## Entregables

- Código fuente del programa que cumpla los requerimientos funcionales y técnicos.
- Binario del programa, compilado para una arquitectura x86.
- Fuente de la documentación en Latex o Markdown.
- PDF con la documentación.

## Evaluación

- Sector de Arranque: 30 %.
- Tutormec: 50 %.
- Documentación: 20 %

En caso que el booting no funcione, la tarea **no** se evaluará.

## Documentación

Las siguientes son las instrucciones para la documentación:

1. **Introducción:** Presentar el problema. Puede “reciclar” partes del enunciado de la tarea programada.
2. **Ambiente de desarrollo:** Indicar las herramientas usadas para implementar la tarea.
3. **Estructuras de datos usadas y funciones:** Se debe describir las principales funciones y estructuras utilizadas en la elaboración de esta asignación.
4. **Instrucciones para ejecutar el programa:** Presentar las consultas concretas usadas para correr el programa para el problema planteado en el enunciado de la tarea y para los casos planteados al final de esta documentación.
5. **Actividades realizadas por estudiante:** Este es un resumen de las bitácoras de cada estudiante (estilo timesheet) en términos del tiempo invertido para una actividad específica que impactó directamente el desarrollo del trabajo, de manera breve (no más de una línea) se describe lo que se realizó, la cantidad de horas invertidas y la fecha en la que se realizó. Se deben sumar las horas invertidas por cada estudiante, sean concientes a la hora de realizar esto el profesor determinará si los reportes están acordes al producto entregado.
6. **Autoevaluación:** Indicar el estado final en que quedó el programa, problemas encontrados y limitaciones adicionales. Adicionalmente debe de incluir el reporte de commits de git. Por otro lado, también debe incluir una calificación con la rúbrica de la sección “Evaluación”.
7. **Lecciones Aprendidas** del proyecto: Orientados a un estudiante que curse el presente curso en un futuro.
8. **Bibliografía** utilizada en la elaboración de la presente asignación.
9. Es necesario documentar el código fuente.

## Aspectos Adicionales

Aún cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones:

1. Si no se entrega documentación, automáticamente se obtiene una nota de 0.
2. Si el código no compila se obtendrá una nota de 0, por lo cual se recomienda realizar la defensa con un código funcional.
3. El código debe ser desarrollado en Ensamblador para x86, en caso contrario se obtendrá una nota de 0.
4. Si no se siguen las reglas del formato del envío a través de Google Drive se obtendrá una nota de 0.
5. La revisión de la documentación será realizada por parte del profesor, no durante la defensa del proyecto.
6. Cada grupo tendrá como máximo 20 minutos para exponer su trabajo al profesor y realizar la defensa de éste, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo cual se recomienda tener todo listo antes de ingresar a la defensa.
7. Cada excepción o error que salga durante la ejecución del proyecto y que se considere debió haber sido contemplada durante el desarrollo del proyecto, se castigará con 2 puntos de la nota final de la presente asignación.
8. Cada grupo es responsable de llevar los equipos requeridos para la revisión, si no cuentan con estos deberá avisar al menos 2 días antes de la revisión al profesor para coordinar el préstamo de estos.
9. Durante la revisión podrán participar asistentes, otros profesores y el coordinador del área.
10. Cualquier indicio de copia será calificado con una nota de 0 y será procesado de acuerdo al reglamento.