**UNIVERSIDAD DE CONCEPCIÓN**

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA



**549253 – TALLER DE APLICACIÓN TIC I**

Profesor Vicenzo Caro

**Miniproyecto I**

Benjamín Ceballo Gil

Pedro Núñez Moraga

José Figueroa Jara

**Resumen**

En el presente informe se reporta el progreso en el proyecto de aplicación y desarrollo de sistemas de dispositivos de entretenimiento electrónico mediante el uso de infraestructura basada en el sistema de microcomputadores Raspberry Pi 3.

Este proyecto ha sido denominado como “Miniproyecto 1”, el cual está siendo actualmente desarrollado por nuestro equipo a lo largo de una serie de sesiones de trabajo de laboratorio en el marco de la asignatura *Taller de Aplicación TIC I*, impartida por el *Departamento de Ingeniería Eléctrica* de la *Universidad de Concepción*.

El Miniproyecto 1 consta de dos partes, siendo la primera el desarrollo e implementación de una máquina recreativa que recrea en una escala más pequeña un juego electrónico conocido como *Grid* y, la segunda, la utilización del Microcomputador Raspberry Pi 3 para el desarrollo de una consola de videojuegos sencilla especializada en la emulación de juegos de video para plataformas antiguas.

Es de este modo que gradualmente hemos ido avanzando paso a paso con la planificación, diseño, ensamblaje, experimentación mediante ensayo y error, pruebas y aplicación de los sistemas antes mencionados como parte de un proceso integral de capacitación grupal en el manejo de hardware y software como una metodología para la resolución de problemas, siendo esta la misión de la actual forma de la ingeniería en sus varias especialidades en el siglo XXI.

A continuación, procedemos a presentar en mayor detalle los pasos seguidos para desarrollar la serie de plataformas electrónicas de entretenimiento desde cero como una demostración de las capacidades de trabajo que nos presentan los microcomputadores Raspberry Pi 3.

**Tabla de Contenidos**

[Dispositivo de tecnologías de entretenimiento electrónico. 1](#_Toc178254173)

[1.1 Máquina Recreativa emuladora del juego *ACTIVATE!* 1](#_Toc178254174)

[1.2 Consola de Videojuegos Retro mediante SO *RETROPIE* 1](#_Toc178254175)

[Actividad n°1. Grid de *ACTIVATE!* 2](#_Toc178254176)

[Ítem 1.1 2](#_Toc178254177)

[1.1.a) Enunciado 2](#_Toc178254178)

[1.1.b) Esquemático Circuitos 2](#_Toc178254179)

[1.1.c) Resultados y Comentarios 2](#_Toc178254180)

[Actividad n°2. Consola RetroPie 3](#_Toc178254181)

[Ítem 2.1 3](#_Toc178254182)

[2.1.a) Enunciado 3](#_Toc178254183)

[2.1.b) Resultados y Comentarios 3](#_Toc178254184)

[Anexo A. Información Adicional 3](#_Toc178254185)

[A.1) Grid de ACTIVATE! 3](#_Toc178254186)

[A.1.1 Acerca del juego ACTIVATE!: 3](#_Toc178254187)

[A.1.2 Acerca de RetroPie 4](#_Toc178254188)

Dispositivo de tecnologías de entretenimiento electrónico.

* 1. Máquina Recreativa emuladora del juego *ACTIVATE!*

La primera parte de nuestro *Miniproyecto I* ha sido desarrollar el prototipo de hardware que consiste en una interfaz de cinco botones pulsadores que indicaran al software el equivalente de pisar una de las cuadrículas en el juego original, así como también cinco sensores adicionales, siendo estos LEDs RGB alineados para conformar un arreglo de cinco “cuadrículas”, los cuales van fluctuando su estado de activación (color) según lo dispuesto en el software que controla el juego, conectadas también a los mencionados botones para así dar funcionalidad al juego; para sumar al ambiente del juego también se utilizó un Buzzer para generar sonidos característicos en cada momento del juego. Todo esto montado en una placa protoboard que está debidamente conectada al microcomputador Raspberry Pi y configurada según los parámetros por nosotros definidos para estos fines. El sotfware que controla el desarrollo del juego está desarrollado en el lenguaje Python.

* 1. Consola de Videojuegos Retro mediante SO *RETROPIE*

En esta etapa se comienza el desarrollo de una consola emuladora para videojuegos retro mediante el empleo del sistema operativo *RetroPie*. Este proceso ha sido principalmente uno de aplicación de software en el cual, mediante una imagen de instalación del ya mencionado SO, se procede a realizar una instalación en limpio del mismo en una unidad de almacenamiento externa legible para el microcomputador Raspberry Pi 3.

A continuación se realiza también el debido proceso de configuración de la misma para que pueda trabajar en base a especificaciones definidas por nuestro equipo y finalmente las pruebas necesarias para mostrar que el producto se encuentra en un estado funcional.

Finalmente se realizó también la grabación y producción de un videotutorial para guiar a usuarios no informados sobre cómo pueden realizar este proceso por sus propios medios.

# Grid de *ACTIVATE!*

Aquí analizaremos y detallaremos los pasos realizados en el proceso de desarrollo de nuestra actividad del miniproyecto.

## 

### Enunciado

Esta primera actividad consiste en implementar una versión (muy) simplificada de Grid, en la que un conjunto de LEDs y botones representará una única fila de la pista original del juego, sin considerar las zonas seguras. Luego, siguiendo una dinámica por turnos y rondas, cada jugador tendrá la oportunidad de presionar la mayor cantidad posible de zonas de acierto dentro de un tiempo límite establecido, con una dificultad que aumentará en cada ronda. Finalmente, ganará el jugador que acumule la mayor cantidad de puntos al completarse todas las rondas.

### Implementación

Para comenzar la construcción del código se definieron todas las funciones que se ocuparían posteriormente en el *maincode*. Las funciones más importantes utilizadas son:

* *entradas\_partida( ):*

Esta función se encargará de solicitar las condiciones iniciales de la partida (entradas), que deberán ser ingresadas por consola por un operador. Las condiciones son validadas para verificar que tienen valores compatibles con la lógica interna del juego (enteras y mayores a cero). Las salidas de esta función son: el número de jugadores, el número de rondas a jugar y el tiempo que tendrá cada jugador por ronda.

* *turno(led, boton, bocina,tiempo, puntajes, jugador):*

Esta función es la que se encarga de ejecutar cada turno de los jugadores. Las entradas son una lista con los leds que se están usando, una lista con los botones correspondientes, el tiempo de duración de la ronda, una lista que almacena los puntajes de los jugadores y un valor entero que representa al jugador que está por jugar. Comienza apagado los LEDs para indicar que el turno comenzará y luego pone cada uno en rojo o en azul (de forma aleatoria). Luego comienza a medir el tiempo y, hasta que este sea igual al tiempo asignado al turno, monitorear si los botones son presionados. Si los botones son presionados se apaga el LED correspondiente y se verifica que color tenía, si era azul se suma un punto al puntaje del jugador y si era rojo se resta un punto. La función no tiene salidas pues edita la lista de puntajes que es entregada como entrada.

Usando estas funciones se construye el *maincode* descrito a continuación:

Se comienza estableciendo los valores iniciales de las variables asociadas al juego y luego se comienza un ciclo *while True* que nos será útil para poder terminar el juego (rompiendo el ciclo). Se inicia le contador de tiempo, el número de rondas a 0 y se crea una variable de validación que permitirá detener el juego si algún jugador es eliminado (esto ocurre si este tiene 0 pts). Luego se comieza el juego como tal que esta regulado por ciclos, un ciclo *while* que verificará las condiciones de juego antes de cada ronda (las condiciones son que aun queden rondas por jugar y que no hayan jugadores eliminados) y un ciclo *for q*ue, a través de la función *turno*, permitirá que todos los jugadores jueguen su ronda. Terminada la partida (por cualquiera de las 2 razones posibles) el juego permitirá al(los) usuario(s) elegir entre iniciar una nueva partida (en este caso, lo descrito anteriormente se repetirá) o dejar de jugar.

### Esquemático Circuitos

### Resultados y Comentarios

Se logró implementar correctamente el juego con todas las funcionalidades solicitadas, la dificultad del ejercicio estuvo mas relacionada con aprender a programar este tipo de hardware y aplicar lógica para concretar el juego como se es esperado.

### Enunciado

Utilizando como base el código del Ítem anterior, modifiquen su juego para incorporar las siguientes funcionalidades:

- En lugar de encender los LEDs de forma completamente aleatoria en rojo o azul, utilicen como base al menos dos de las cuatro secuencias mostradas en el Anexo “Secuencias de LEDs”. De esta forma, la combinación de LEDs durante cada intervalo de tiempo deberá seguir una secuencia determinada, que se elegirá de forma aleatoria entre las que decidan implementar. Cada secuencia seleccionada deberá aplicarse a lo largo de una ronda completa del juego.

- Añadan un contador de aciertos consecutivos para cada jugador, asignando un puntaje adicional después de lograr 4, 8, 16 y 32 aciertos en cadena.

### Implementación

Para esta nueva versión del juego se crean nuevas funciones y se modifican algunas existentes, los principales cambios son:

* Se elimina de la función *randomizador\_colores* con un randomizador que elije dos posibles patrones fijos que contienen 5 secuencias alternadas.
* Se agrega a la función *turno* un conteo de rachas y bonificaciones (una varible almacena el número de aciertos consecutivos y la otra regula la entrega de las bonificaciones de puntaje dependiendo de esta.
* Con las variables antes mencionadas, la función *mostrar\_puntaje* se incluye nuevas variables a mostrar (racha y bonificación)

Luego en el *maincode* se le agrego a la función turno la opción de que hiciera los mismo solo que con el patrón seleccionado aleatoriamente

### Resultados y comentarios

La actividad mantuvo la misma lógica del código anterior solo que la randomizacion de los leds era fija, además fue sencillo agregar las variables de rachas y bonificaciones

# Consola RetroPie

Aquí analizaremos y detallaremos los pasos realizados en el proceso de desarrollo de nuestra actividad del miniproyecto.

## 

### Enunciado

En esta segunda actividad tendrán que utilizar una Raspberry Pi para crear una consola de

juegos retro utilizando el sistema operativo (SO) RetroPie. Este SO es parte de una plataforma de software que permite emular una amplia variedad de consolas y sistemas de juegos clásicos en una sola interfaz, brindando la posibilidad de revivir la nostalgia de los juegos retro en un dispositivo compacto y de bajo costo.

Ítem 2.1

El objetivo de esta actividad es investigar por sus propios medios los pasos y configuraciones necesarias para instalar y ejecutar una versión funcional de RetroPie en Raspberry Pi. Para ello, deberán redactar su propia guía tutorial donde documenten el paso a paso mínimo necesario, desde los programas utilizados para instalar el SO, la configuración de un joystick o teclado para poder jugar correctamente, los emuladores de consolas disponibles, el método de traspaso de los juegos que hayanconsiderado oportuno, entre otros.

Tengan en cuenta que, como toda guía tutorial, esta debe ser reproducible por cualquier

persona que esté o no esté familiarizada con la plataforma, por lo que deben incluir la mayor cantidad de detalles posibles, e idealmente complementar sus explicaciones con imágenes o capturas de pantalla que consideren relevantes.

### Resultados y Comentarios

Durante la sesión de trabajo de laboratorio del día jueves 26 de septiembre se procedió a realizar el proceso de la implementación del software RetroPie para utilizar a la Raspberry Pi como consola de videojuegos.

Lo primero que se hizo fue el preparar una tarjeta SD virgen/formateada para así grabar en la misma una imagen del sistema operativo RetroPie, en este caso empleando el software Raspberry Pi Imager que de manera automática permite instalar de manera automática el sistema operativo a partir de una imagen de instalación descargable del sitio web de RetroPie.

Una vez conectado todo el hardware, comenzamos con el primer arranque conectando nuestro mouse teclado inalámbrico y configurando los controles de entrada para navegar la interfaz. Se intentó cargar imágenes de ROMs mediante una memoria USB externa pero puesto a que la Raspberry Pi no era capaz de leer dicho dispositivo de almacenamiento usando el sistema RetroPie, se decidió configurar los ajustes de conectividad inalámbrica y hacer llegar a estos archivos mediante el uso de un cliente FTP, *FileZilla*, con el cual se logró exitosamente cargar archivos a la consola.

Finalmente se probaron cuatro tipos de juegos distintos para tres consolas distintas y funcionaba sin problemas en cada uno de los casos. Si bien en términos gráficos no habían anomalías, y los controles permitían manejo fluido de los juegos, el detrimento que encontramos hasta ahora fue la falta de salida de audio que en cierto aspecto genera un detrimento a la experiencia, pero más allá de este detalle se logró cumplir el cometido.

Poco después se grabó y editó una guía en forma de un video tutorial con el cual se planea enseñar a personas que no cuenten con suficiente experiencia a poder configurar sus dispositivos para usar RetroPie de manera sencilla y desde cero.

##### Información Adicional

###### Grid de ACTIVATE!

* + 1. Acerca del juego *ACTIVATE!*:

En Canadá, existe un centro de entretenimiento interactivo conocido como Activate. En

Activate, los jugadores se mueven de sala en sala para completar diferentes juegos que combinan principalmente elementos de ejercicio físico, resolución de problemas, y trabajo en equipo. Entre los juegos disponibles, se encuentra uno llamado Grid, donde los jugadores interactúan con un piso lleno de luces de colores que representan diferentes zonas:

- Zonas seguras, representadas por una luz verde.

- Zonas de acierto, representadas por una luz azul.

- Zonas de trampa, representadas por una luz roja.

- Zonas inactivas, representadas por una luz apagada.

El objetivo del juego es mantenerse dentro de las zonas seguras mientras se espera la

oportunidad para moverse ágilmente a través de la pista para pisar las zonas de acierto, evitando las zonas de trampa que van formando diferentes patrones que cambian de lugar y ritmo con el tiempo.

* + 1. Acerca de RetroPie

RetroPie es un sistema operativo diseñado para microcomputadores Raspberry Pi de código abierto desarrollado en el Reino Unido el cual está diseñado en base a la arquitectura del software de emulación de videojuegos RetroArch el cual permite convertir a estos dispositivo en auténticas consolas de videojuegos retro usando archivos que contienen el código de las memorias de solo lectura (ROM) de los cartuchos y/o imágenes de disco de los juegos para sus respectivas plataformas.

Está diseñado para ser fácil de utilizar y es bastante popular alrededor del mundo. Tiene bastante compatibilidad con diversos tipos de controles que se pueden configurar para tanto navegar la interfaz del sistema operativo, así como también jugar a los juegos que el usuario cargue en la memoria del sistema.

RetroPie incluye también un administrador de archivos para facilitar la transferencia de ROMs los cuales se pueden transferir mediante dispositivos como memorias USB y también de manera inalámbrica utilizando clientes de FTP.