

**UNIVERSIDAD DE CONCEPCIÓN**  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA



**549305-1 – Taller de Aplicación TIC II**

Profesor: Vincenzo Caro Fuentes

**Miniproyecto 1**

Benjamín Ceballo Gil  
José Figueroa Jara  
Pedro Núñez Moraga

Concepción, 27 de abril de 2025

## Resumen

En el presente, detallamos los progresos y avances realizados en el marco de nuestro llamado “Miniproyecto 1”, el cual es una instancia de aprendizaje y perfeccionamiento en el cual nuestro equipo ha dedicado a probar aplicaciones prácticas para solución de problemas tecnológicos aprendiendo a emplear Hardware y Software de Arduino en conjunto a reforzar nuestros conocimientos y habilidades anteriores para nuestro perfeccionamiento constante como integrantes de la comunidad de las ciencias de la ingeniería.

El Miniproyecto 1 consiste de dos partes

La primera fue la creación de un dispositivo electrónico interactivo programable para fines recreativos, emulando con hardware básico la función de un control para el videojuego musical “Guitar Hero” empleando de base un microcontrolador Arduino UNO R3 con el cual se logró implementar una forma más rudimentaria de dicho videojuego con algunas características modificables por el usuario para proveer una experiencia más amena para el usuario durante la ejecución como la velocidad, el volumen y otros y dando un puntaje en base al desempeño durante la partida.

La segunda parte consiste en combinar el hardware de Arduino con software ejecutado de manera nativa en un computador de control para con ello ejecutar una versión modificada del famoso programa *El Juego de la Vida* de Conway con algunas funciones y características nuevas para así ensayar el uso de interfaces gráficas y de creación de programas empleando el lenguaje de programación Python para con ello diversificar nuestras herramientas de trabajo para futuros proyectos.

Todo esto ha sido realizado en sesiones de laboratorio semanales llevadas a cabo entre Marzo y Abril del presente año 2025 de manera grupal y llevando una bitácora del progreso en base a copias de las versiones del código correspondientes a cada sesión así como material fotográfico y videográfico que será presentado más adelante.

## Actividad n°1. Control Interactivo de Guitar Hero

---

A continuación se detalla sobre el desarrollo del control interactivo de Guitar Hero.

### Ítem 1.1

#### 1.1.a) Enunciado

Esta primera actividad consiste en implementar una versión básica del juego “Guitar Hero” utilizando una placa Arduino. Para ello, se desarrollará un piano digital que permita reproducir cinco notas musicales utilizando un buzzer, cinco botones y cinco LEDs RGB. La finalidad es que cada botón represente una tecla de la guitarra de Guitar Hero, los cuales se deberán presionar una vez se ilumine su respectivo LED, reproduciendo una nota musical específica.

Con estos elementos, preparen un circuito en el que cada botón esté vinculado a un LED RGB y a una nota musical distinta. A partir de esta base, desarrollos un programa en Arduino que simule un piano de 5 teclas, considerando los siguientes aspectos:

- Cada vez que se presione uno de los botones, el sistema deberá:
  - ✓ Encender el LED RGB correspondiente con un color específico.
  - ✓ Reproducir una nota musical predefinida con una duración determinada a través del buzzer.
  - ✓ Mostrar en el monitor serial un mensaje indicando qué botón fue presionado y qué nota fue reproducida.
- Además, el código deberá implementar dos comandos especiales que se envían desde el monitor serial para añadir funcionalidad al sistema:
  - ✓ Comando Vxxx: Modifica el volumen de salida, siendo xxx un número entre 000 y 255 que representa el valor de PWM aplicado al buzzer.
  - ✓ Comando Tx: Cambia el banco de tonos utilizados. El índice actual del banco puede ser aumentado (T1) o disminuido (T0), permitiendo al usuario cambiar el set de 5 notas disponibles durante la ejecución del programa. Para mayor simplicidad al momento de trabajar con las notas, tomen como referencia el Anexo “Reproducción de Notas”.

#### Ítem 1.2

A partir del código base desarrollado en el ítem anterior, implementen ahora una versión básica

del juego Guitar Hero, donde una fila de LEDs representa las notas musicales que el jugador deberá reproducir secuencialmente presionando los botones correspondientes. Para ello, deberán considerar los siguientes aspectos en el desarrollo del circuito y del código en Arduino:

- Se debe crear un menú de selección inicial, visible en el monitor serial, donde el jugador pueda elegir:

- ✓ La dificultad del juego: la dificultad modificará la velocidad a la que se reproducen las notas (es decir, el tiempo disponible para presionar el botón correcto).
- ✓ Una canción predefinida: se deben definir al menos tres melodías diferentes. Estas pueden ser creadas o buscadas en internet (por ejemplo: Mario Bros, Harry Potter, Star Wars, entre otras).

- Las notas de la canción seleccionada deberán almacenarse en una estructura secuencial (por ejemplo, una matriz) y reproducirse una a una, encendiendo el LED correspondiente a cada nota en el orden de la melodía. La velocidad de reproducción dependerá del nivel de dificultad seleccionado.

- El jugador deberá presionar el botón correspondiente al LED que se encuentra iluminado en ese momento. Si acierta, el juego:

- ✓ Suma una cantidad de puntos base.
- ✓ Aumenta una racha de aciertos consecutivos. Las rachas multiplican los puntos base, en los siguientes niveles:
  - 2x: al alcanzar 5 aciertos consecutivos.
  - 4x: al alcanzar 10 aciertos consecutivos.
  - 8x: al alcanzar 15 aciertos consecutivos.
  - 16x: al alcanzar 20 aciertos consecutivos.
- ✓ Reinicia la racha si el jugador comete un error.

- Si el jugador presiona una tecla incorrecta, el juego:

- ✓ Resta puntos del total.
- ✓ Reinicia la racha.
- ✓ Si el puntaje llega a 0 puntos, el juego termina automáticamente y se reproduce un tono de "Game Over" en el buzzer.

- Una vez todas las notas de la canción han sido reproducidas, el juego finaliza y se deben contemplar las siguientes situaciones:

- ✓ Si el jugador completó correctamente todas las notas, el código reproducirá una melodía de victoria y mostrará el puntaje total en el monitor serial.

- ✓ Si el jugador no completó todas las notas, pero su puntaje no llegó a cero, se muestra únicamente el puntaje obtenido, sin reproducir ninguna melodía.
- ✓ En ambos casos, se puede ofrecer reiniciar el juego o finalizar.

### 1.1.b) Esquemático Circuitos

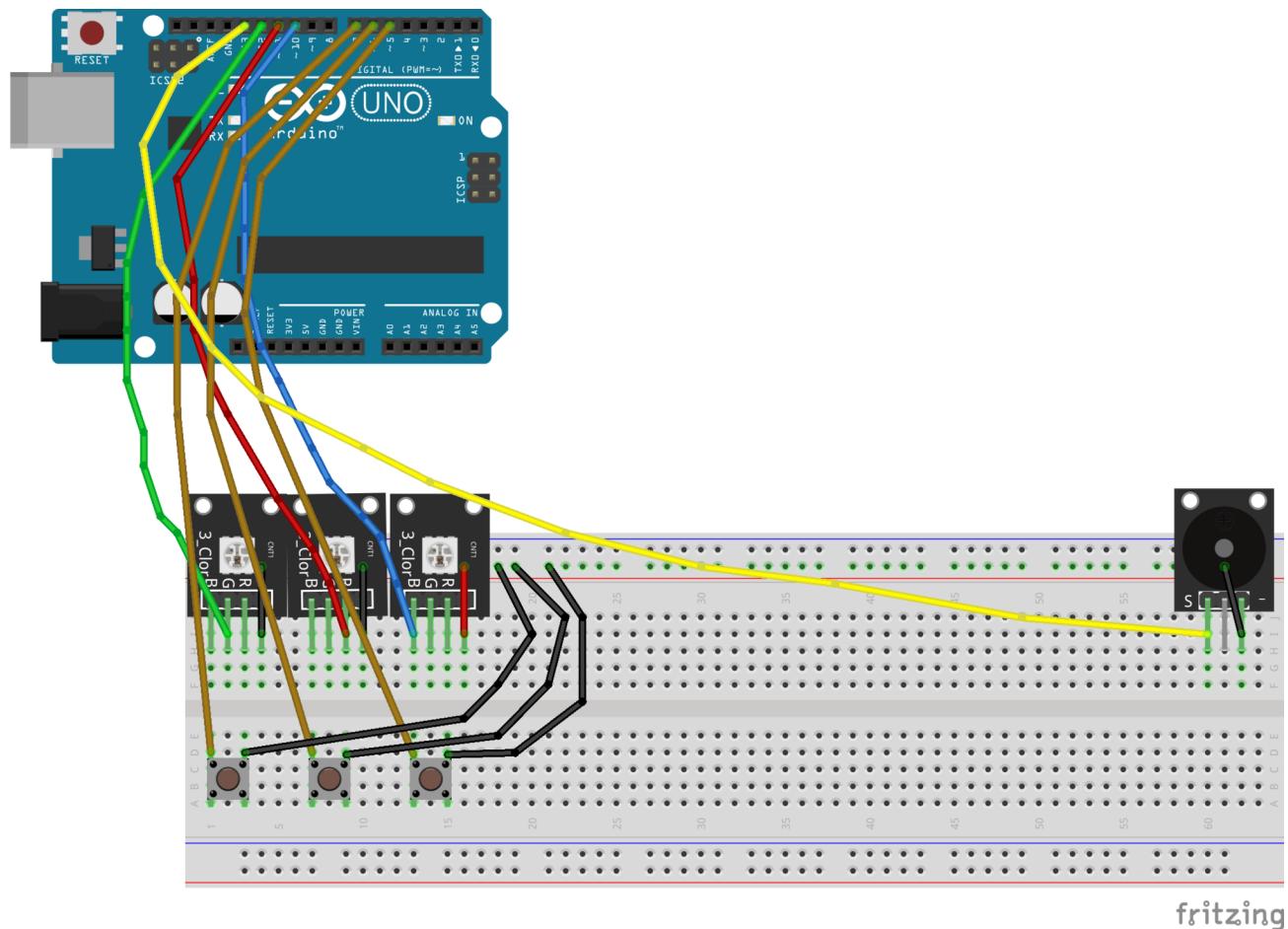


Figura 1: Esquemático del Circuito.

Aquí podemos ver la actual disposición del hardware utilizado, siendo este unos cinco botones y unos cinco LEDs RGB conectados de manera debida a la placa de Arduino además de un zumbador para salida de audio, empleando nuestra protoboard y conectado el Arduino a un computador del equipo.

### 1.1.c) Resultados y Comentarios

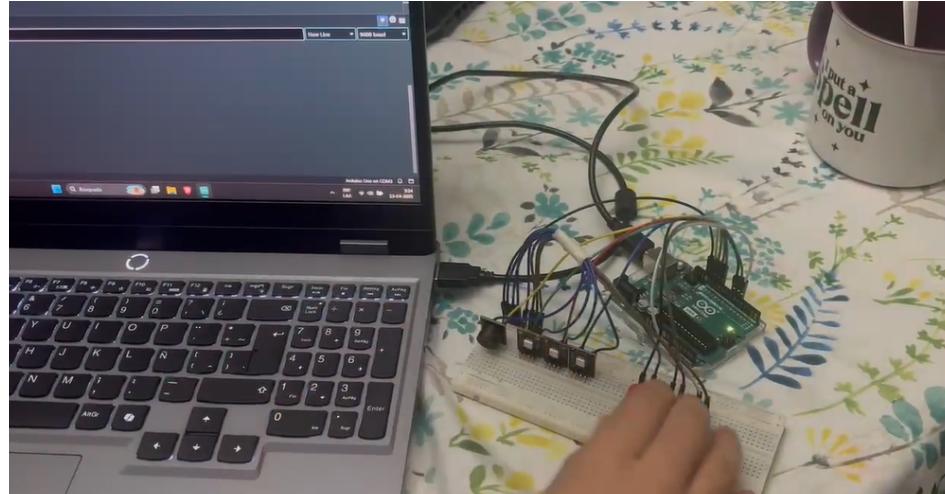


Figura 2: Ensayos de funcionamiento.

Durante el transcurso del desarrollo de esta actividad fueron surgiendo toda una serie de imprevistos que se tuvieron que ir superando en el camino siendo este proceso parte del progreso realizado en la puesta en marcha de la primera mitad de este Miniproyecto

Partiendo en primer lugar con el haber comenzado a aprender sobre la marcha la utilización de hardware y software de Arduino, adecuándose los integrantes de nuestro equipo al nuevo Entorno de Desarrollo Integrado que utiliza C++ lo cual ha sido un pequeño desafío a superar.

Más allá de esto, la adaptación a estas nuevas condiciones de trabajo ha sido rápida y sin muchos inconvenientes. Luego de esto comenzó el testeо del hardware utilizado en el laboratorio mediante la selección de los LEDs y la instalación de la botonera hecha con la protoboard.

Una de las problemáticas surgidas en este proceso ha sido la del zumbador, que por error de interpretación del equipo habíamos conectado tanto con cableado para su control como de alimentación eléctrica de 5V que en este caso en particular no era necesario. Similarmente ocurrió que con el código no se habían declarado correctamente variables relacionadas al tipo de funcionalidad de los botones lo que llevó a tener que corregir el funcionamiento de los bucles de control de los mismos en el programa principal.

Ocurrió también un incidente en una de las sesiones en el cual se estaba teniendo problemas con el cableado de los componentes lo que generaba un flujo de corriente de regreso a la placa y que interrumpía la alimentación y adecuado control de los componentes, de hecho hicimos la observación de que ocurría que debíamos manipular manualmente los cables y moverlos para que el circuito funcionase y por lo tanto se tuvo que recablear el dispositivo y cambiar de placa.

Posteriormente se continuó el trabajo de manera externa fuera del tiempo y espacio de laboratorio lo que permitió que se pudiera desarrollar con más calma, y completar en mejor tiempo la puesta en marcha del control de guitar hero.

El principal problema encontrado fue con el control del buzzer, lo que nos llevó a aprender más del mismo para así evitar tener estos problemas nuevamente a futuro.

Se adjunta con el resto de archivos del proyecto el material videográfico de ensayos del mismo.

Respecto al funcionamiento en sí del programa, en la primera parte de nuestro trabajo se da al usuario la opción de elegir entre un modo “piano” para que pueda probar las luces y sonidos del dispositivo y un modo “juego” para comenzar a jugar una partida del juego programado.

Sobre este último, una vez seleccionado este modo se da al jugador la oportunidad de ingresar comandos mediante el monitor serial digitando la letra inicial correspondiente a la acción deseada para configurar a gusto la dificultad, canción, tono y otros que se quieran emplear durante el juego.

Una vez se elige la dificultad y canción inicia el juego. En esto se va encendiendo el led para indicar la nota y el jugador debe presionar el botón correspondiente. De ser así se registra un acierto en base a la duración entre la activación y la presión del botón y con ello se suma puntaje. Mientras vayan subiendo los aciertos se va sumando la racha y con ello se multiplica el puntaje. Se va presentando el puntaje en el monitor serial y se felicitará o avisará al jugador de sus errores y si el puntaje es cero o menos se termina el juego porque se descuentan puntos por fallar en los aciertos. Eventualmente se hace sonar un pequeño sonido de victoria o derrota.

## Actividad n°2. The Game of Life

---

### Ítem 2.1

#### Enunciado

The Game of Life fue creado por Conway como un modelo matemático de un sistema celular con el fin de explorar la idea de la vida y la muerte en un entorno abstracto. Conway buscaba un sistema con reglas simples, pero que pudiera producir comportamientos complejos y variados.

La idea detrás de esta actividad es que sean capaces de programar y alterar externamente una versión en Python de The Game of Life, para lo cual pueden considerar el código base mostrado en el Anexo “Código Base de The Game of Life”, donde se implementa el juego para una cuadrícula de 100x100 con sus reglas originales.

De manera inicial, adapten dicho código para implementar el juego en una interfaz gráfica creada con la biblioteca PyQt6, apoyándose de la biblioteca matplotlib para construir la cuadrícula y

de QTimers para actualizar el juego en tiempo real, con una tasa de actualización variable a definir

por ustedes. Luego, utilizando como intermediario la comunicación por protocolo serial, integren un

programa de Arduino que sea capaz de interactuar con su juego de la siguiente forma:

- Cada un intervalo de 10 segundos, el programa de Python debe enviar hacia Arduino la cantidad de células vivas que se encuentran actualmente en la cuadrícula. Luego, definen en Arduino 3 intervalos para evaluar las condiciones de estabilidad, subpoblación y sobre población de las células; vinculando cada estado con un color particular de un módulo LED.

- Desde Arduino, implementen un botón pulsador que sea capaz de reiniciar desde 0 el juego en Python. Para ello, cada vez que se presione el botón, envíen un carácter o un mensaje hacia Python para que la interfaz reaccione a esta acción.

### Ítem 2.2

En esta segunda parte de la actividad, deberán adaptar su juego del ítem anterior bajo el contexto de una invasión zombie, siguiendo nuevas reglas de comportamiento. Cada celda de la cuadrícula representará un individuo en el sistema, y puede encontrarse en uno de los siguientes tres

estados:

0: zombie

1: persona viva

2: muerto

Adicionalmente, cada célula debe poseer un nivel de vida, con un valor máximo de 200 puntos. *Condiciones y reglas del juego*

Durante cada iteración del juego, los individuos evolucionan siguiendo las siguientes reglas:

- Una persona viva pierde 5 puntos de vida por cada zombie en su vecindad inmediata. Si su vida llega a 0, se transforma en zombie con 100 puntos de vida.
- Un zombie pierde 10 puntos de vida si está rodeado por menos de 2 personas viva y 30 puntos de vida si está rodeado por más de 3 personas vivas. Si su vida llega a 0, hay un 10% de probabilidad de que el zombie se cure y se transforme en persona con 100 puntos de vida. En caso contrario, se transforma en muerto.

#### Eventos especiales

El juego debe incluir cuatro tipos de eventos especiales, que pueden alterar el estado de la cuadrícula en zonas específicas:

1. Infección masiva (cuadrada): transforma un bloque 5x5 lleno de personas vivas en zombies con 200 puntos de vida.
2. Infección masiva mutada (forma pulsar): transforma una figura pulsar de 13x13 (ver Anexo “Generación de pulsar”) en el mapa en zombies con 200 puntos de vida.
3. Ritual de purificación: dentro de un área de 15x15, los zombies tienen un 80% de probabilidad de curarse y volver a ser personas vivas con 200 puntos de vida.
4. I am atomic! (área): destruye completamente un área de 25x25, cambiando todas las células de la zona al estado muerto.

Estos eventos especiales deben estar vinculados a botones dentro de la interfaz desarrollada con PyQt6. Cada botón debe permitir activar un único evento especial, y al hacerlo, debe actualizar inmediatamente la cuadrícula.

Paralelamente, los eventos especiales también deben poder ser activados desde Arduino mediante botones físicos conectados a la placa. Para ello, deberán implementar un sistema de comunicación serial en el programa de Python, capaz de recibir e interpretar mensajes siguiendo la expresión regular “a-x”, donde:

- “a” identifica que se trata de un mensaje de ataque,

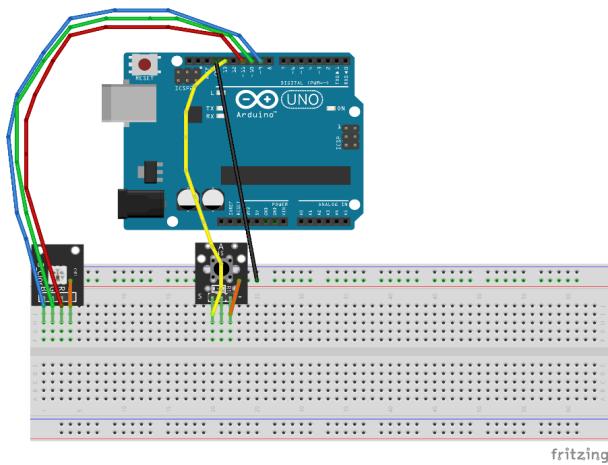
- "x" es un número del 1 al 4 que representa el tipo de evento a ejecutar. Ejemplos:

A-1: *Infección Masiva*, A-2: *Infección Masiva Mutante*, A-3: *Ritual de Purificación*, A-4: *I Am Atomic*, R: *Reinicio*.

El juego debe detectar automáticamente si todo el mapa esta lleno de zombies o de personas vivas. En cualquiera de estos casos, se debe:

1. Reproducir un sonido (audio de victoria o derrota) utilizando bibliotecas como playsound, pygame, QSound, u otra similar.
2. Reiniciar automáticamente el juego, volviendo a una configuración inicial aleatoria.

## 2.1.a) Esquemático Circuitos



## 2.1.b) Resultados y Comentarios

La segunda parte del Miniproyecto comenzó con el desarrollo del juego en sí a partir del código base entregado de manera previa y la creación de la interfaz gráfica actualizada para la ejecución del mismo.

Se estuvo ensayando la modificación de la matriz y la generación de patrones en base a las condiciones puestas en el programa, modificándose también la interfaz gráfica para que el usuario pueda personalizar parámetros de ejecución del programa de manera rápida y sencilla mientras se ejecuta el mismo en tiempo real, generando varios patrones geométricos llamativos y permitiendo juguetear con lo mismo.

Luego de esto se prosiguió a ir elaborando el programa con el juego modificado, para lo cual se consultó la creación de elementos del código de manera digital empleando a ChatGPT para asistir a

los integrantes y se fueron realizando varias correcciones al código por lo mismo, pues tenía este varias imprecisiones y otros errores que limitaban su correcta compilación y ejecución y por lo tanto interrumpiendo el ritmo de progreso de esta parte del proyecto.

Se instaló y conectó la placa Arduino UNO R3 con su protoboard para poder instalar el LED y el botón accionador para así controlar con hardware directamente la ejecución del juego y complementar así también el trabajo e informaciones entregados por la interfaz gráfica.

En esta parte se encontraron problemáticas vinculadas a la instalación correcta del hardware puesto a que en un momento no se habían hecho debidamente las conexiones a tierra de los componentes electrónicos afectando negativamente a su funcionamiento. Además de esto se observó que no se habían hecho las configuraciones correspondientes en el IDE para establecer una conexión serial con el Arduino lo que impedía el poder enviar el código a ejecutarse en el mismo y por lo mismo también impedía la entrada de informaciones por comunicación serial que asisten a controlar parámetros y funciones del modificado juego.

Respecto al funcionamiento del programa se provee una interfaz gráfica con el objetivo de facilitar la interacción del usuario teniendo cuenta de la cantidad de células vivas, infecciones y además permite controlar distintos parámetros tal como se solicita además de poder visualizar más fácilmente los patrones que surgen en la grilla con las configuraciones provistas.

## Anexo A. Información Adicional

### A.1) Registro Fotográfico de Trabajo en Laboratorio.

Video de registros de trabajo en laboratorio y demás:

<https://www.youtube.com/watch?v=ZgZz-DH3pIs>

