

Universidad Tecnológica Nacional

Proyecto Final

Automatización de Sala Snoezelen para Escuela
de Educación Especial

Autor:

- Massaroni, Juan Francisco

*Proyecto final presentado para cumplimentar los requisitos académicos
para acceder al título de Ingeniero Electrónico en la
Universidad Tecnológica Nacional
Facultad Regional Paraná*

Noviembre de 2019

Declaración de autoría:

Declaro que el Proyecto Final “Automatización de Sala Snoezelen para Escuela de Educación Especial” y el trabajo realizado es propio.

Declaro:

- Este trabajo fue realizado en su totalidad, o principalmente, para acceder al título de grado de Ingeniero Electrónico, en la Universidad Tecnológica Nacional, Regional Paraná.
- Se establece claramente que el desarrollo realizado y el informe que lo acompaña no han sido previamente utilizados para acceder a otro título de grado o pre-grado.
- Siempre que se ha utilizado trabajo de otros autores, el mismo ha sido correctamente citado. El resto del trabajo es de autoría propia.
- Se ha indicado y agradecido correctamente a todos aquellos que han colaborado con el presente trabajo.
- Cuando el trabajo forma parte de un trabajo de mayores dimensiones donde han participado otras personas, se ha indicado claramente el alcance del trabajo realizado.

Firmas:

-
-

Fecha:

Agradecimientos:

Mis más sinceros agradecimientos están dirigidos a mis padres, hermanas y abuela, que siempre me apoyaron emocional y económicamente para poder llegar a cumplir el objetivo de ser un profesional.

A mis amigos y colegas, por el apoyo y motivación para seguir adelante.

Un agradecimiento especial a la institución y a los docentes de la carrera por la formación y dedicación para transmitir los conocimientos que hoy poseo.

Massaroni, Juan Francisco

Universidad Tecnológica Nacional

Abstract

Facultad Regional Paraná

Ingeniero en Electrónica

Automatización de Sala Snoezelen para Escuela de Educación Especial

Massaroni, Juan Francisco

Abstract:

Multi-Sensorial or Snoezelen stimulation is a form of therapy applied on people that has some or several disabilities, people that have their motor or neurological development compromised or severe cognitive deficiencies.

I have tried to simplify the therapist task, allowing to select the stimuli before the patient arrives for treatment, easing the therapy session and making the most of its time.

This project has a central controller that commands the room's system through pre-established routines, activating different devices autonomously following the recorded steps. These routines can be configured from Laptop or PC using a software developed for this task, which allows the user to upload to the controller the desired configuration.

Python language was used to program the controller application, and C# programming language to develop the PC application.

The system is constituted by a low-cost computer in charge of storing and playing the configured routines, has outputs for Audio, Video and eight relays that activate the room's devices. Additionally, it generates a Wi-Fi hotspot that enables the configuration and provides an Access Point for internet connection, everything on a wireless environment.

User's interaction with the controller is achieved through an infrared remote control which allows to activate the configured routines, but this can also be accomplished using an USB mouse and selecting the routine on the screen.

Keywords:

Automation, Stimulation, Multi-sensory, Therapy

Resumen:

La estimulación multisensorial o Snoezelen es un tipo de terapia que se utiliza con personas que tienen alguna o varias discapacidades, personas con afectaciones en su desarrollo motor, neurológico o con deficiencias cognitivas severas.

Se buscó simplificar la tarea del terapeuta brindando la posibilidad de seleccionar los estímulos en una etapa previa al tratamiento del paciente, pudiendo así facilitar la terapia y aprovechar al máximo el tiempo de cada sesión.

El proyecto consiste en un controlador central que comanda el sistema de la sala mediante rutinas preestablecidas, activando los distintos dispositivos de la sala de manera autónoma. Dichas rutinas pueden ser configuradas desde una PC, mediante un software desarrollado para tal fin, que permite cargar en el controlador la configuración deseada.

Se utilizó el lenguaje Python para la aplicación del controlador y lenguaje C# para el desarrollo de la aplicación para la PC.

El sistema se compone de una computadora de bajo costo encargada de almacenar y reproducir las rutinas configuradas, con salidas de audio, video y ocho relevadores que activan los elementos de la sala. Además, genera un espacio Wi-Fi que permite realizar la configuración de forma inalámbrica y puede actuar como punto de acceso a internet.

La interacción del controlador con el usuario se realiza a través de un control remoto infrarrojo que permite activar las rutinas configuradas, pero también se puede realizar la misma acción utilizando un ratón USB y seleccionando la rutina en pantalla.

Palabras Clave:

Automatismo, Estimulación, Multi-sensorial, Terapia

Reconocimientos:

Un reconocimiento especial a los docentes de la cátedra Proyecto Final por su dirección y ayuda constante, en especial por su orientación metodológica y su continuo estímulo durante todo el proceso.

Además, quiero agradecer a la institución Escuela de Educación Especial N° 8, "Berta E. Goitia de Jurado" de la localidad de Monte Caseros, Corrientes, por brindarme el espacio y la posibilidad de trabajar en sus instalaciones para concretar el proyecto.

Índice:

1.	Capítulo 1: Introducción	1
1.1	¿Qué es Snoezelen?	2
1.2	¿A quién va dirigido?	3
1.2.1	Discapacidad intelectual	3
1.2.2	Discapacidad sensorial	3
1.2.3	Enfermedades psiquiátricas.....	3
1.2.4	Envejecimiento normal y demencias	4
1.2.5	Parálisis cerebral	4
1.3	Problemática actual	4
1.4	Objetivo del proyecto	5
1.5	Estudio de mercado	5
1.5.1	Target.....	5
1.5.2	Pruebas de concepto. ¿Es un producto útil?.....	5
1.5.3	Pruebas de producto. Competencia. ¿Qué hay en el mercado?	6
1.5.4	Análisis de recursos	6
1.5.5	Análisis de costos.....	7
1.5.6	Ciclo de vida del producto	7
2.	Capítulo 2: Desarrollo	8
2.1	Diagrama de bloques	8
2.2	Descripción general del Hardware	8
2.2.1	Placa CPU	9
2.2.2	Conectividad Wi-Fi	11
2.2.3	Salidas de Video.....	13
2.2.4	Salidas de Audio.....	13
2.2.5	Pines GPIO.....	14
2.2.6	Control remoto infrarrojo.....	15
2.2.7	Convertor de nivel lógico	16
2.2.8	Módulo de Relevadores	17
2.3	Programas y herramientas utilizadas	17
2.3.1	Sistema Operativo “Raspbian”.....	17
2.3.2	OMXPlayer	18
2.3.3	FEH	18
2.3.1	HostApd y DNSmasq.....	19
2.3.2	Lenguaje Python y Herramienta PyQt5	20
2.3.3	Lenguaje C# y Visual Studio 2019.....	21
2.4	Desarrollo de aplicación para Raspbian	21
2.5	Desarrollo de aplicación para Windows	30
2.6	Diseño Final	36
2.6.1	Diseño final	36
2.6.2	Prestaciones.....	36
2.6.3	Fotos del hardware y capturas de pantalla de software.....	37
3.	Capítulo 3: Resultados	40
4.	Capítulo 4: Análisis de Costos	41
4.1	Plan de Negocios	41
4.2	Costos y Retorno de Inversión	43
4.3	Plan de Marketing	44
5.	Capítulo 5: Discusión y Conclusión.....	45
5.1	Conclusión.....	45
5.2	Mejoras en futuros desarrollos	45
6.	Capítulo 6: Literatura Citada.....	46

Lista de Figuras:

Figura 1 – Introducción. Perfil de las personas con discapacidad	1
Figura 2 – ¿Qué es Snoezelen? Ejemplo sala de estimulación	2
Figura 3 – Problemática Actual. Estado de la sala al comienzo del proyecto.....	4
Figura 4 – Diagrama de bloques. Esquema completo del sistema.....	8
Figura 5 – Placa CPU. Vista superior Raspberry Pi 3 B+	9
Figura 6 – Placa CPU. Descripción de componentes y características	10
Figura 7 – Placa CPU. Almacenamiento: Tarjeta SD	11
Figura 8 – Conectividad Wi-Fi. Rayos X del chip de conectividad inalámbrica	12
Figura 9 – Conectividad Wi-Fi. Antena PCB.....	12
Figura 10 – Salidas de Video. Conector HDMI	13
Figura 11 – Salidas de Audio. Conector Jack 3.5mm	14
Figura 12 – Pines GPIO. Pinout de los 40 pines	14
Figura 13 – Control remoto infrarrojo. Mando y receptor	15
Figura 14 – Control remoto infrarrojo. Código de letras: NuMicro Mini5X	16
Figura 15 – Conversor de nivel lógico. Placa del adaptador	16
Figura 16 – Módulo de Relevadores. Disposición de componentes	17
Figura 17 – Sistema Operativo “Raspbian”. Logo de Raspbian Stretch.....	18
Figura 18 – HostApd y DNSmasq. Red Wi-Fi vista desde Windows 10 y Android.....	19
Figura 19 – Lenguaje Python y Herramienta PyQt5. Logo de PyQt	20
Figura 20 – Lenguaje C# y Visual Studio 2019. Logos de C# y Visual Studio	21
Figura 21 – Desarrollo de aplicación para Raspbian. Entorno de desarrollo IDLE.....	21
Figura 22 – Desarrollo de aplicación para Raspbian. Entorno de desarrollo Qt Designer	24
Figura 23 – Desarrollo de aplicación para Raspbian. Prototipo del diseño del control en pantalla ..	25
Figura 24 – Desarrollo de aplicación para Raspbian. Aplicación para conversión de UI a Python...	26
Figura 25 – Desarrollo de aplicación para Raspbian. Diagrama de flujo (Parte 1)	29
Figura 26 – Desarrollo de aplicación para Raspbian. Diagrama de flujo (Parte 2)	30
Figura 27 – Desarrollo de aplicación para Windows. Entorno de desarrollo Visual Studio 2019	31
Figura 28 – Desarrollo de aplicación para Windows. Detalle de la interfaz de usuario	31
Figura 29 – Desarrollo de aplicación para Windows. Menú desplegable de acciones	32
Figura 30 – Desarrollo de aplicación para Windows. Selector de duración	33
Figura 31 – Desarrollo de aplicación para Windows. Selector de cantidad de pasos	33
Figura 32 – Desarrollo de aplicación para Windows. Configuración de estímulos físicos.....	33
Figura 33 – Desarrollo de aplicación para Windows. Configuración de estímulos multimedia.....	34
Figura 34 – Desarrollo de aplicación para Windows. Estado de conexión a la red	34
Figura 35 – Desarrollo de aplicación para Windows. Diagrama de flujo completo	35
Figura 36 – Diseño Final. Esquemático del hardware	36
Figura 37 – Diseño Final. Disposición de componentes en carcasa	37
Figura 38 – Diseño Final. Receptor infrarrojo en carcasa	37
Figura 39 – Diseño Final. Aplicación para Raspbian: Control de Rutinas	38
Figura 40 – Diseño Final. Aplicación para Raspbian: Control Manual	38
Figura 41 – Diseño Final. Aplicación para Windows: Interfáz gráfica de usuario	39
Figura 42 – Diseño Final. Aplicación para Windows: Carga de configuración.....	39
Figura 43 – Resultados. Prueba de latencia y pérdida de paquetes.....	40
Figura 44 – Resultados. Pruebas de alcance Wi-Fi	40
Figura 45 – Costos y Retorno de Inversión. Cálculo de retorno de inversión	44

Lista de Tablas

Tabla 1 – Introducción. Estudio sobre perfil de las personas con discapacidad	1
Tabla 2 – Análisis de Costos. Estudio de los costos estimados a junio 2019.....	7
Tabla 3 – Resultados. Pruebas de fiabilidad del mando a distancia.....	40
Tabla 4 – Costos y Retorno de Inversión. Costos del Hardware de la sala.....	43

Lista de Abreviaciones

API	Application Programming Interface
APROM	Application Program Read Only Memory
CEC	Consumer Electronics Control
CPU	Central Processing Unit
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DSI	Display Serial Interface
GPIO	General Purpose Input-Output
GPU	Graphical Processing Unit
GUI	Graphical User Interface
HDMI	High-Definition Multimedia Interface
I2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
IDLE	Integrated Development and Learning Environment
INDEC	Instituto Nacional de Estadística y Censos
LCD	Liquid Crystal Display
LED	Light-Emitting Diode
LXDE	Lightweight X11 Desktop Environment
OMS	Organización Mundial de la Salud
PC	Personal Computer
PCB	Printed Circuit Board
PIXEL	Pi Improved Xwindow Environment, Lightweight
PWM	Pulse Width Modulation
RAM	Random Access Memory
SFTP	Secure File Transfer Protocol
SOC	System On Chip
SPI	Serial Peripheral Interface
SSH	Secure SHell
SSID	Service Set IDentifier
TFTP	Trivial File Transfer Protocol
TRRS	Tip/Ring/Ring/Sleeve
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
Wi-Fi	Wireless Fidelity (originalmente una marca comercial)

Dedicado a:

Especialmente a mis padres, que fueron mis pilares fundamentales para convertirme en Ingeniero, siempre impulsándome a crecer profesionalmente y como persona.

A mis hermanas, que siempre estuvieron presentes brindando su apoyo incondicional en todo momento.

A mi abuela que, a pesar de la distancia, siempre supo acompañar en todo lo que me propuse realizar.

1. Capítulo 1: Introducción

Según el INDEC, en 2018, el 10,2% de la población de la Argentina¹ tiene algún tipo de discapacidad. En términos absolutos, se corresponde con una estimación de 3.571.983 personas.

En términos estadísticos, podemos observar que el 59,0% de la población con alguna discapacidad declara experimentar solo una dificultad, mientras que el 30,5% señala tener dos dificultades o más, y el 10,5% tiene solo certificado de discapacidad vigente.

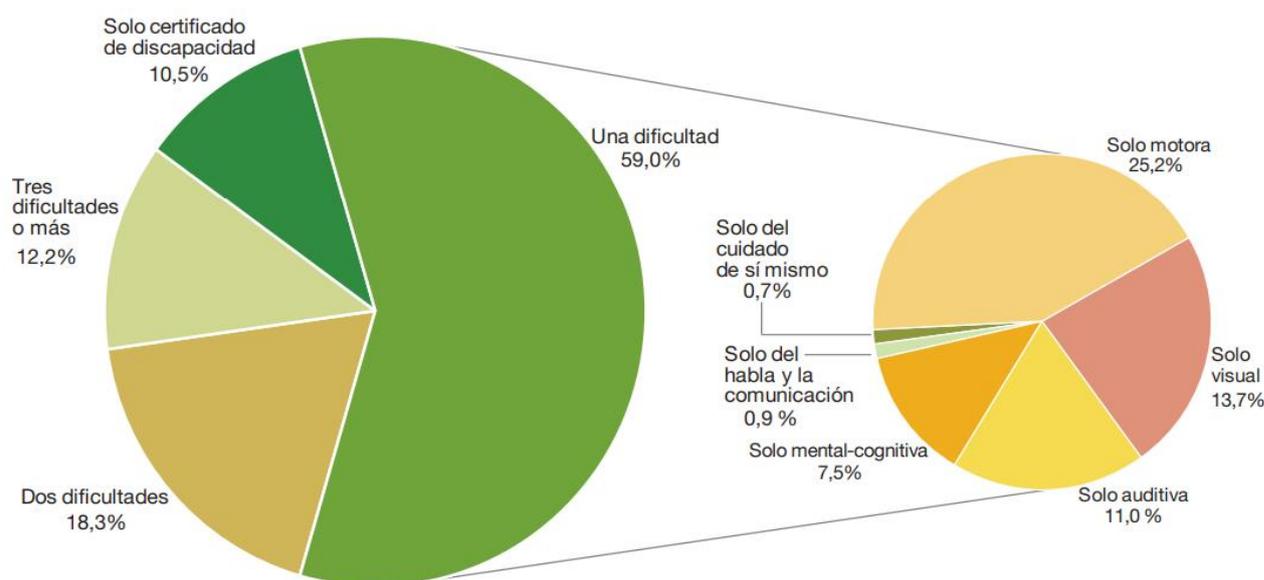


Figura 1 – Introducción. Perfil de las personas con discapacidad¹

Cantidad y tipo de dificultad	Total	6 a 14 años	15 a 64 años	65 años y más
		%		
Total	100,0	100,0	100,0	100,0
Una dificultad	59,0	51,5	61,6	56,7
Solo motora	25,2	5,3	24,6	30,2
Solo visual	13,7	11,2	18,5	7,3
Solo auditiva	11,0	4,9	8,5	15,9
Solo mental-cognitiva	7,5	24,9	8,6	2,3
Solo del cuidado de sí mismo	0,7	///	///	///
Solo del habla y la comunicación	0,9	4,7	0,9	///
Dos dificultades	18,3	14,8	15,4	23,2
Tres dificultades o más	12,2	13,5	9,1	16,6
Solo certificado de discapacidad	10,5	20,2	13,9	3,6

Tabla 1 – Introducción. Estudio sobre perfil de las personas con discapacidad¹

¹ INDEC (2018). Estudio Nacional sobre el Perfil de las Personas con Discapacidad. Recuperado de: <https://www.indec.gov.ar>

Según la Organización Mundial de la Salud (OMS, 2001), se considera que la discapacidad es toda limitación en la actividad y restricción en la participación, originada en la interacción entre la persona con una condición de salud y los factores contextuales (entorno físico, humano, actitudinal y sociopolítico), para desenvolverse en su vida cotidiana, dentro de su entorno físico y social, según su sexo y edad.

1.1 ¿Qué es Snoezelen?

Las salas de estimulación sensorial, o Snoezelen, son una herramienta o recurso que permite trabajar el despertar sensorial a través de la acción y la experimentación. Suponen un cambio metodológico en la estimulación de niños y niñas, personas con discapacidad y personas mayores, donde el sujeto se convierte en el protagonista de sus propios aprendizajes y el educador/terapeuta pasa a ser un mediador de este proceso. Conlleva la creación de espacios, que cuenten con estímulos controlados que favorezcan la percepción de sensaciones y ayuden a adquirir el aprendizaje por medio del descubrimiento. Según la visión de Snoezelen, el mundo en el que estamos se basa en múltiples sensaciones, ya sean lumínicas, acústicas, olfativas, etc. Estas sensaciones las percibimos a través de los órganos sensoriales, es por eso que este tipo de estimulación pretende potenciar todas las entradas sensoriales, ya que es una forma de conectarnos con nuestro entorno, es decir con el mundo.



Figura 2 – ¿Qué es Snoezelen? Ejemplo sala de estimulación

En alumnos discapacitados, el aula multisensorial consigue potenciar las relaciones positivas fuera del estrés del centro educativo y logrando mejoras importantes en la lógica, concentración y coordinación. En concreto, en los alumnos con discapacidad cognitiva, se ha demostrado que la aplicación de estímulos controlados es una de las terapias más ventajosas, ya que mejora la calidad de vida de estas personas tanto aumentando el grado de autonomía, la tolerancia, el control emotivo y la comprensión de términos familiares. Así, las aulas Snoezelen sirven de instrumento de comunicación no-verbal, dado que potencian la relajación y aportan incentivos que no se obtendrían de otra forma. Los resultados más positivos del empleo de aulas multisensoriales se han obtenido en personas con autismo, de hecho, la utilización de materiales Snoezelen consigue reducir hasta un 75% el daño auto-provocado y la agresividad de este tipo de pacientes, frente a cualquier otro tipo de terapia.

1.2 ¿A quién va dirigido?

En un principio, la estimulación Snoezelen fue desarrollada para aplicarse a personas con discapacidad intelectual, por eso su principal campo de aplicación ha sido esta patología, aunque se han descrito múltiples campos donde se han observado beneficios en su aplicación.

1.2.1 Discapacidad intelectual

La discapacidad intelectual (antes denominada retraso mental) es una alteración en el desarrollo del ser humano. Se caracteriza por significativas limitaciones, tanto en el ejercicio intelectual como en las conductas adaptativas. Genera anomalías en el proceso de aprendizaje, como la adquisición lenta e incompleta de las habilidades cognitivas durante el crecimiento de la persona, lo cual conduce finalmente a limitaciones sustanciales en el desarrollo corriente. Se evidencia antes de los 18 años de edad y afecta alrededor del 2% de la población general.

Este fue el primer campo donde se aplicó la técnica Snoezelen y actualmente se utiliza ampliamente en los diferentes grados o niveles de esta discapacidad. Hay varios estudios donde se han constatado una mejoría de las capacidades generales de estas personas, mediante la estimulación de sus órganos y capacidades sensoriales. En 1998, Kenyon y cols. comprobaron que la estimulación multisensorial en personas con discapacidad intelectual producían cambios en su conducta, así como un incremento del estado de relajación y de su bienestar. En relación al aspecto comunicativo y emocional, se ha observado que las personas con discapacidad intelectual que realizan este tipo de intervención incrementan su comunicación y sus emociones positivas en detrimento de los patrones negativos usuales en esta discapacidad.

1.2.2 Discapacidad sensorial

Se trata de los trastornos en los órganos de los sentidos. Incluye los trastornos relacionados con la vista, el oído y el lenguaje. Son patologías muy importantes de considerar dado que conllevan graves efectos psico-sociales. Producen problemas de comunicación del paciente con su entorno.

Se pueden beneficiar de esta técnica especial aquellos sujetos con dificultades en el lenguaje, en la percepción sensorial, con déficit visual (parcial/totalmente ciegos), auditivo (sordos), motórico (parálisis cerebral), plurideficientes (sordomudos), entre otros.

1.2.3 Enfermedades psiquiátricas

La discapacidad psiquiátrica, o enfermedad mental, describe una amplia gama de condiciones mentales y emocionales. En diferentes estudios se comprobó que la intervención sensorial en personas que padecen una enfermedad mental, les ayudaban a reducir su nivel ansiedad y estrés y aumentaban los sentimientos de bienestar y confianza. Además, en algunos estudios se ha observado que este tipo de estimulación ayuda a reducir la sintomatología psiquiátrica que ocasiona alteraciones conductuales.

1.2.4 Envejecimiento normal y demencias

En la actualidad, la estimulación sensorial en personas con demencia, sobretodo demencia tipo Alzheimer está tomando fuerza y se están realizando diferentes investigaciones para observar que beneficios pueden obtener estas personas.

La mayoría de las intervenciones en demencia se realizan en personas en fases avanzadas. Algunos de los beneficios observados han sido: una reducción de los sentimientos de ansiedad y estrés; una importante reducción de las alteraciones conductuales durante la sesión y en un periodo de tiempo corto al finalizarla; y un significativo efecto positivo en las conductas apáticas, depresivas y agresivas, así como una disminución de las conductas desinhibidas.

Además, se ha observado que las sesiones continuas favorecen el incremento en la interacción y la comunicación de estas personas con el entorno, así como una mayor independencia en las actividades de la vida diaria.

1.2.5 Parálisis cerebral

En este campo, también se ha constatado que la intervención Snoezelen tiene cierta efectividad. Diferentes estudios muestran, una mejora en la calidad de vida de estas personas. Y en otro estudio, se observó una mejora en el nivel de alerta y atención ante estímulos presentados por vía sensorial visual y auditiva, aunque dicha mejoría no se mantenía después de las sesiones. Además, se constató un mayor nivel de relajación durante e incluso una hora después de la sesión.

1.3 Problemática actual

La motivación principal para la realización de este proyecto surgió de observar la escasez de recursos tecnológicos que poseen las escuelas públicas en general, pero particularmente esta institución de educación especial del interior de la provincia de Corrientes, la cual posee un muy alto porcentaje de niños provenientes de familias de muy bajos recursos económicos que no disponen de acceso a este tipo de estimulación fuera del ámbito escolar.



Figura 3 – Problemática Actual. Estado de la sala al comienzo del proyecto

Se desea que la escuela pueda ampliar y mejorar su servicio terapéutico-educativo, es decir, aquel que implementa acciones de atención tendientes a promover la adquisición de conductas saludables y adecuados niveles de autonomía e independencia, mediante la incorporación de modelos de interacción de metodologías y técnicas del ámbito terapéutico, pedagógico y recreativo.

Concretar este proyecto puede conllevar una importante mejoría en las posibilidades de rehabilitación de un alumno que, a través de metodologías y técnicas específicas instrumentadas por un equipo interdisciplinario, pueda permitirle alcanzar un nivel funcional óptimo para una adecuada integración social.

1.4 Objetivo del proyecto

Actualmente, la institución cuenta con un espacio de estimulación Snoezelen que posee el equipamiento mínimo, con activación de modo manual y sin posibilidad de diseñar las sesiones de estimulación. El objetivo de este proyecto fue brindar una herramienta integral que permita diseñar las sesiones, seleccionando el tiempo y duración de cada estímulo, para posibilitar la utilización del recurso centrado en cada individuo y su necesidad específica. Esto permite a los profesionales de la institución diseñar previamente la sesión y focalizar el trabajo enteramente en el alumno sin desviar su foco de atención al realizar la manipulación de los elementos de la sala manualmente.

1.5 Estudio de mercado

1.5.1 Target

Se define como TARGET o Población Objetivo del presente proyecto a todo profesional y/o institución que brinden servicios de tipo Terapéutico-Educativos o de Rehabilitación. Por ejemplo, instituciones que se encuentren constituidas o posean las modalidades de “Estimulación Temprana”, “Centro de Día”, “Centro Educativo Terapéutico”, “Servicios para personas con Discapacidad en general” o “Centro de Rehabilitación”.

En cuanto al target geográfico, inicialmente se focaliza en el mercado regional del Litoral Argentino, con más de 50 centros de educación especial como potenciales clientes, muchos de los cuales no disponen de la tecnología ofrecida. No se descarta una posterior expansión al mercado nacional.

1.5.2 Pruebas de concepto. ¿Es un producto útil?

El concepto es atractivo para el sector y se adapta completamente al comprador, además brinda la flexibilidad de agregar o quitar funcionalidades a la hora de adaptar el producto a cada necesidad.

Teniendo en cuenta la gran población que puede estar interesada en utilizar el producto y la escasez de oferta en el target geográfico planteado, se considera que puede atraer a todos los segmentos de la región, como ser instituciones educativas o centros de atención y rehabilitación de personas con discapacidad tanto del sector público como privado.

1.5.3 Pruebas de producto. Competencia. ¿Qué hay en el mercado?

El producto que se propone posee una plataforma tecnológica actual además de ofrecer una gran versatilidad y adaptabilidad. Se considera que satisface todas las necesidades de los posibles compradores ya que brinda una alternativa de bajo costo dentro de la región.

En el mercado se encuentran múltiples oferentes que utilizan tecnología poco versátil como la comunicación serial (COM) cableada y una plataforma poco adaptable a los cambios en el hardware de la sala.

Otra gran diferencia se encuentra en el costo final de cara al consumidor, se estima que el producto se podrá ofrecer por un precio hasta un 30% menor que la mejor oferta actual en el mercado.

Actualmente la competencia (Playtime, Centro CAMAC, Winter of Life) se encuentra concentrada en la provincia de Buenos Aires, brindando una ventaja geográfica para la venta, distribución y posterior puesta en marcha dentro de la región.

1.5.4 Análisis de recursos

Ya que la escuela no cuenta con un presupuesto para dedicar a esta ampliación se intenta realizar proyecto puramente “ad honorem” se intenta utilizar todos los recursos disponibles, tanto de parte del alumno como de la institución.

Se caracterizan los recursos con los que contamos al momento de iniciar el proyecto.

Alumno UTN:

- > Raspberry Pi 3B+ Element14;
- > Módulo de 8 Relés (5V 10A);
- > Fuente de PC con salidas de 5V y 12V;
- > Elementos varios de electricidad (cables, toma-corrientes, cable canal).

Institución Educativa:

- > Espacio físico con acondicionador de aire;
- > Consola potenciada con entrada y salida de audio;
- > Micrófono profesional;
- > Proyector HDMI instalado;
- > Notebook HP dedicada para el aula;
- > Espejo con tira de luces LED;
- > Láser de figuras multipunto;
- > Elementos no-electrónicos para estimulación.

Si bien se cuenta con una base sólida para iniciar el proyecto, aún faltan elementos adicionales para completar la experiencia multisensorial, como ser luces de colores, aromatizadores y un burbujero.

Además del hardware para la sala, es necesario el desarrollo de una aplicación (para Windows o Android) que pueda comandar las acciones y realizar la puesta en marcha de cada sesión.

1.5.5 Análisis de costos

Habiendo evaluado múltiples opciones para realizar esta solución se ha optado por un núcleo central comandado por un Raspberry Pi 3B+, conectado a una salida de potencia a Relés que pueda comandar los diferentes elementos de la sala (Luces, Aromatizadores, Burbujas). El principal motivo para la utilización de este dispositivo es su bajo costo y la capacidad de manejar múltiples salidas de 5v de tensión, otra ventaja es que posee un chip Wi-Fi integrado, además salida de Audio Plug 3.5mm y Video HDMI.

El estudio de costos* estimado es:

Cantidad/ Horas	Producto	Concepto	Precio por Unidad/Horas	Envío	Subtotal
2	Difusor Aromático Eléctrico	Aroma	\$ 2.213,7	\$ 0,0	\$ 4.427,5
1	Consola Potenciada 4canales Cabezal	Audio	\$ 6.389,0	\$ 0,0	\$ 6.389,0
1	Micrófono profesional		\$ 650,0	\$ 305,0	\$ 955,0
1	Máquina De Burbujas Bubble Master	Burbujas	\$ 3.625,0	\$ 0,0	\$ 3.625,0
1	Líquido Para Máquina De Burbujas 5L		\$ 449,0	\$ 410,0	\$ 859,0
1	Raspberry Pi 3B+ 64gb Memoria y Fuente	Hardware de control	\$ 3.359,0	\$ 0,0	\$ 3.359,0
1	Módulo de 8x Relés 5v 10a		\$ 415,8	\$ 285,0	\$ 700,8
-	Materiales (Cables, Tomas, Cable Canal)		\$ 4.000,0	\$ 0,0	\$ 4.000,0
1	Proton Tacho Luz Negra 18 LED	Luces	\$ 1.899,0	\$ 0,0	\$ 1.899,0
3	Luz RGB Proton Par 36 LED		\$ 700,0	\$ 305,0	\$ 3.015,0
1	Esfera Espejada Bola Boliche 40 + Motor		\$ 1.931,3	\$ 0,0	\$ 1.931,3
1	Spot Tacho Led Blanco		\$ 599,0	\$ 330,0	\$ 929,0
1	Laser Lluvia Audio-rítmico Multipunto		\$ 599,0	\$ 285,0	\$ 884,0
50	Instalación y puesta en marcha	Mano de Obra	\$ 200,0	\$ 0,0	\$ 10.000,0
120	Desarrollo aplicación p/Windows	Software de control	\$ 150,0	\$ 0,0	\$ 18.000,0
120	Desarrollo aplicación p/Raspberry		\$ 150,0	\$ 0,0	\$ 18.000,0
1	Proyector Viewsonic - 3000 Lum (Usado)	Video	\$ 9.999,0	\$ 0,0	\$ 9.999,0
TOTAL					\$ 88.972,6

Tabla 2 – Análisis de Costos. Estudio de los costos estimados a junio 2019

*Todos los costos son estimados y sujetos a cambios por stock/inflación.

1.5.6 Ciclo de vida del producto

Teniendo en cuenta los componentes a utilizar, y como serán de uso dedicado para la sala, se estima un ciclo de vida superior a los 5 años ya que se utilizará una tecnología de comunicación Wireless Wi-Fi de las bandas de 2.4GHz o 5GHz. Esto brinda la posibilidad de utilizar cualquier dispositivo que pueda correr el software de la Interfaz de Usuario y conectarse a la sala mediante Wi-Fi.

2. Capítulo 2: Desarrollo

2.1 Diagrama de bloques

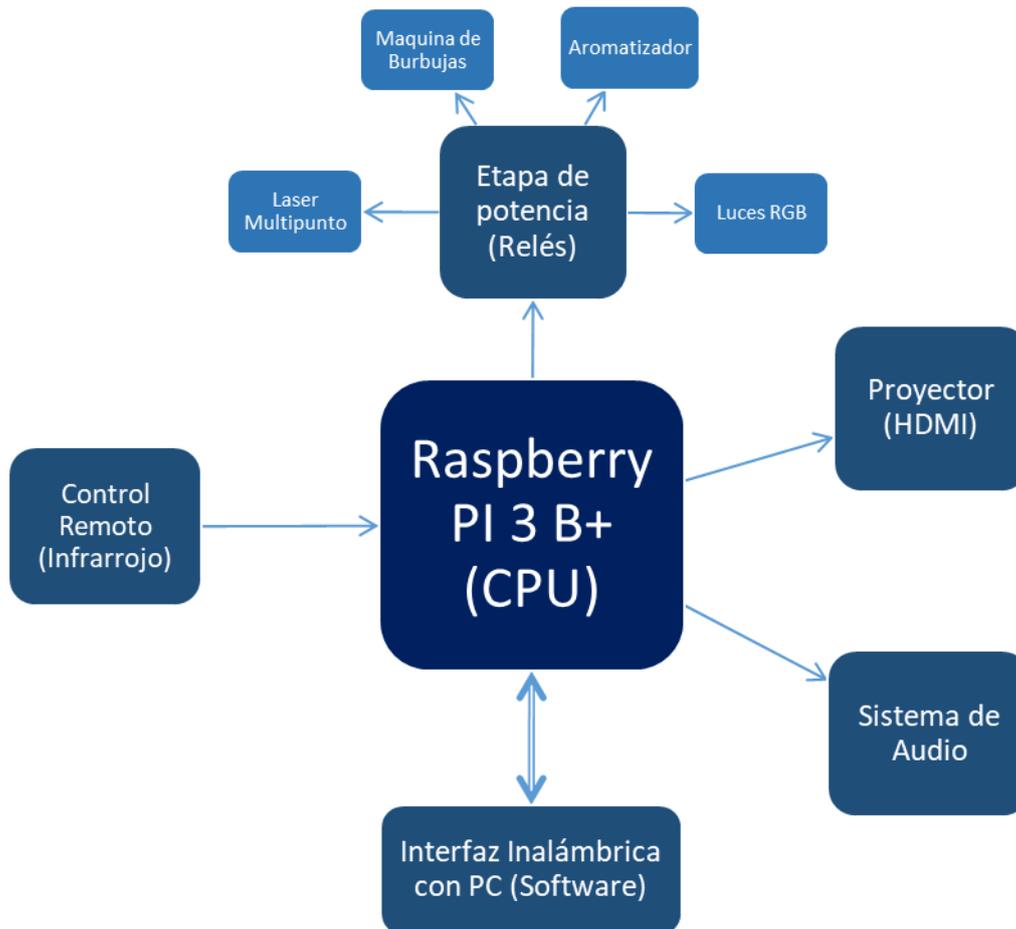


Figura 4 – Diagrama de bloques. Esquema completo del sistema

2.2 Descripción general del Hardware

El propósito del sistema es proporcionar una herramienta integral que permita diseñar sesiones de estimulación, seleccionando el tiempo de activación y duración de cada uno de los estímulos. El objetivo principal es automatizar la utilización de los componentes de la sala de forma sencilla, pero brindar la posibilidad de modificar la configuración de manera consistente sin necesidad de un hardware específico, ofreciendo versatilidad y adaptabilidad a cualquier entorno.

Se utiliza una CPU basada en un Raspberry PI 3B+ que es una computadora de bajo costo y se encarga de controlar el hardware necesario para llevar a cabo la tarea. Además, se ha desarrollado una aplicación para la misma, que es capaz de tomar la selección del usuario mediante un control remoto (infrarrojo) y actuar en consecuencia. El sistema contará con 4 posibles configuraciones, completamente programables, que podrán ser activadas y desactivadas por el usuario mediante el control remoto infrarrojo. La CPU tendrá salidas de video, audio y relevadores para controlar los componentes de la sala.

Además, se ha diseñado y desarrollado un software para Windows que permite conectar una PC o Laptop a la CPU de la sala (Raspberry) y realizar la configuración de cada sesión que luego será seleccionable mediante el control remoto infrarrojo.

El proyecto completo comprende, además del hardware específico, el diseño e implementación de las aplicaciones que se utilizan. Una de ellas se utiliza para el control de la sala y se desarrolló para el sistema operativo Raspbian (basado Debian, una distribución de Linux), y la segunda, que se utiliza para la configuración de la misma, se desarrolló para el sistema Windows. Finalmente se realiza el conexionado completo y la puesta en marcha de cada uno de los componentes de la sala, asegurando que las especificaciones de hardware sean las correctas para cada dispositivo.

2.2.1 Placa CPU

Como CPU en el sistema se utiliza es una placa de desarrollo llamada Raspberry Pi 3 B+. Se hizo un análisis en relación a costo/beneficio, comparando precio y prestaciones de las distintas placas que hay en el mercado, obteniendo como resultado que la misma es la más conveniente para los recursos requeridos.

Esta placa, más allá de sus limitaciones, posee un hardware capaz de realizar las tareas de procesamiento necesarias con una performance muy buena y a un bajo costo económico. Ofreciendo conectividad Wi-Fi integrada, así como capacidad de manejar audio, video y salidas de control de forma nativa, sin necesidad de elementos anexos.



Figura 5 – Placa CPU. Vista superior Raspberry Pi 3 B+

La Raspberry Pi 3 B+ es una placa de desarrollo con unas dimensiones de 8.5cm por 5.6cm que ensambla en su circuito un chipset Broadcom BCM2837B0 de cuatro núcleos ARM Cortex-A53 (64 bits) a 1.4 GHz de frecuencia de reloj, con una memoria RAM integrada de 1GB DDR2. Dicha combinación es capaz de mover con soltura videojuegos livianos y aplicaciones varias, además de disponer de una gran potencia de procesamiento para otros tipos de tareas.

La GPU encargada de los gráficos es la Broadcom VideoCore IV, una solución Dual Core compatible con Open GL y Open GV, que permiten llegar a resoluciones de video “Full HD” con soltura.

Además, ofrece conectividad inalámbrica Wi-Fi de 2.4GHz/5GHz IEEE 802.11.b/g/n/ac, conectividad Bluetooth 4.2 BLE y un puerto de red FastEthernet 10/100 Gbps con la posibilidad de anexas un adaptador Gigabit Ethernet sobre USB 2.0 con un máximo teórico de 300 Mbps.

Otros detalles importantes a destacar son los puertos que posee:

- > 40 pines GPIO
- > Salida de video HDMI
- > 4 x Puertos USB 2.0
- > Puerto CSI para conectar una cámara.
- > Puerto DSI para conectar una pantalla táctil
- > Jack 3.5mm para audio estéreo o video compuesto
- > Micro SD
- > Micro USB (alimentación)
- > Power-over-Ethernet (PoE)

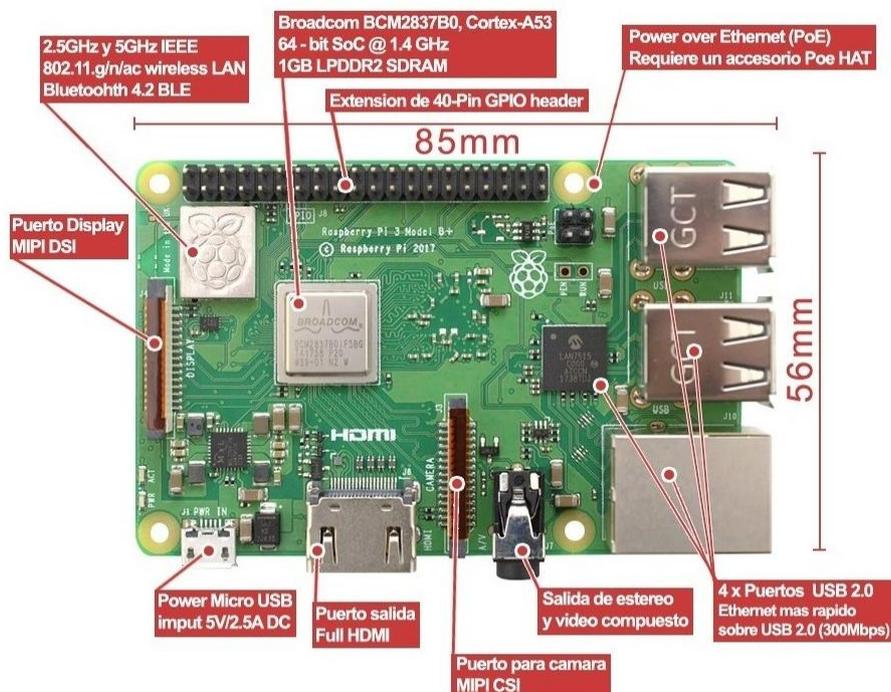


Figura 6 – Placa CPU. Descripción de componentes y características

Para realizar este proyecto, resultaron de especial interés los 40 pines GPIO que permiten tomar como entrada el comando a distancia (control infrarrojo) y actuar sobre los relevadores de salida. Además de las salidas de video HDMI y audio estéreo para la reproducción de elementos multimedia, así como también, la capacidad de conectar cualquier dispositivo USB, como ser un ratón y/o un teclado tanto cableado como inalámbrico.

La Raspberry Pi no tiene un disco duro tradicional, para ello dispone de un lector/ranura para memorias SD, un sistema de almacenamiento en estado sólido. El arranque del sistema se hará desde la propia tarjeta SD, con lo que debido a que tiene que albergar todo el sistema operativo, es necesario que la tarjeta sea de al menos 2 GB de capacidad para almacenar todos los archivos requeridos. Es necesario enfatizar la Clase de la memoria a utilizar, para garantizar el óptimo rendimiento esta debe ser Clase 10 o superior, para obtener una velocidad de transferencia de datos mínima de 10 MB/s.

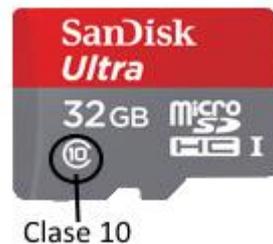


Figura 7 – Placa CPU. Almacenamiento: Tarjeta SD

Para poder arrancar el S.O. será necesario primero instalar (flashear) un sistema operativo en la tarjeta antes de poder trabajar con ella. Tras el arranque inicial de la SD se puede trabajar con almacenamiento de algún dispositivo de disco por USB.

Teniendo en cuenta que el proyecto requiere que la CPU almacene las rutinas y archivos multimedia, será necesario una memoria SD con espacio suficiente para alojar una buena cantidad de datos, es decir, con una capacidad mayor a 16GB de almacenamiento.

Como se describió anteriormente el sistema operativo utilizado es Raspbian, una versión de Linux basada en Debian, especialmente desarrollada para la placa Raspberry, siendo uno de los más populares y con mejor rendimiento para esta placa. Viene preinstalado con software educativo, para programación y para uso general.

Otra razón fundamental por la cual se eligió utilizar este dispositivo es el soporte que posee, con infinidad de proyectos y herramientas que resultaron de especial interés para este proyecto, además de la existencia de una vasta documentación para las mismas.

2.2.2 Conectividad Wi-Fi

Una parte fundamental del desarrollo de este proyecto se basa en la posibilidad de obtener una conectividad inalámbrica fiable, por lo que se evalúa en detalle la circuitería que corresponde a este módulo en la placa de desarrollo.

En la Figura 8, a continuación, se observa una imagen de rayos X del SoC que muestra lo que podemos encontrar debajo del encapsulado del módulo “Wi-Fi + Bluetooth”.

El chip que ocupa el mayor espacio debajo de la cobertura metálica es el Cypress CYW43455, con un módulo Wireless 802.11ac de banda dual (2.4 y 5.6 GHz) y un módulo Bluetooth 4.2, los cuales son superiores a sus predecesores encontrados en la Pi 3 y Pi Zero.

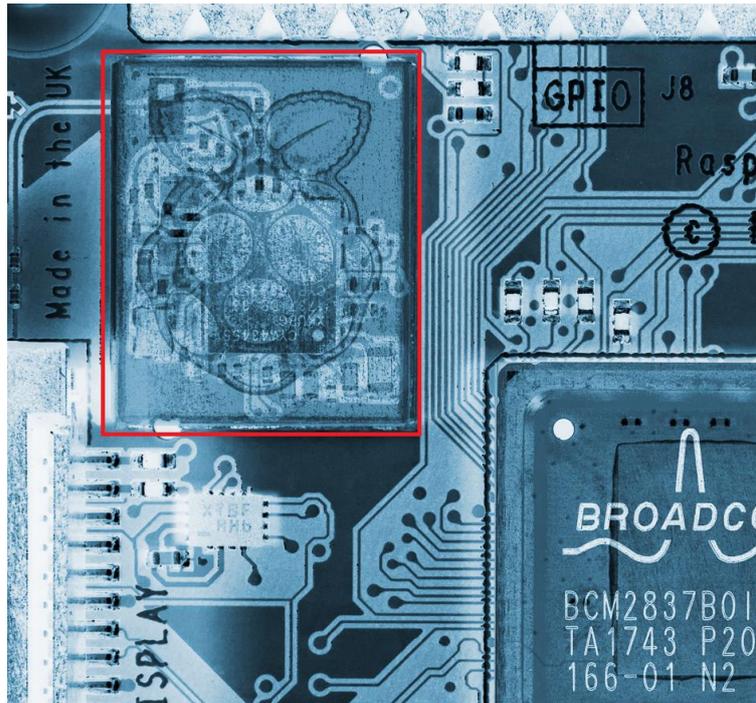


Figura 8 – Conectividad Wi-Fi. Rayos X del chip de conectividad inalámbrica ²

En el lado posterior del PCB se puede observar una antena en forma de un triángulo traslúcido. Esta tecnología está licenciada por Proant AB y trabaja mediante una cavidad triangular a través del plano de tierra de la placa, dentro de la cual pueden resonar las radiofrecuencias y ser enfocadas hacia el extremo estrecho (y exterior) del triángulo, desde donde son transmitidas a través del resto de la circuitería Wireless.

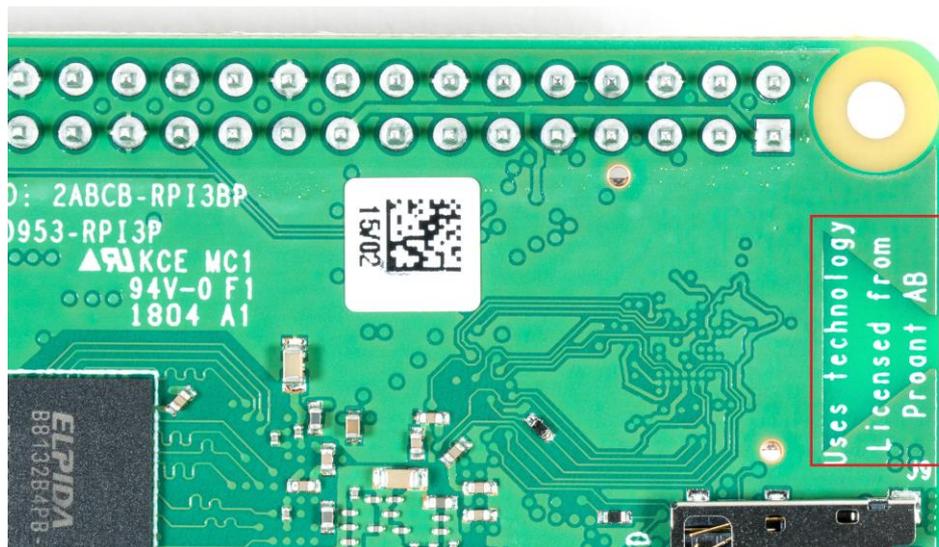


Figura 9 – Conectividad Wi-Fi. Antena PCB

En términos de performance, la combinación de utilizar el PCB para la modulación y la cobertura metálica para la disipación del calor posibilita que el SoC se desempeñe de una manera más rápida y fiable.

² Macdonald, S (2018). A deep dive into the Raspberry Pi 3 B+ hardware. Recuperado de: <https://blog.pimoroni.com/>

2.2.3 Salidas de Video

Para la salida de video la Raspberry posee 3 opciones: un Conector RCA o Video Compuesto (PAL y NTSC), una Interfaz DSI para paneles LCD y un conector HDMI.

El video compuesto está diseñado para conectar la Raspberry Pi a los antiguos dispositivos de pantalla. Como su nombre lo indica, el conector crea un compuesto de colores que se encuentran dentro de una imagen (rojo, verde y azul) y lo envía por un sólo cable al dispositivo de visualización, comúnmente las viejas TV de tubo de rayos catódicos (CRT). La calidad, sin embargo, no será la mejor. Las conexiones de video compuesto son significativamente más propensas a la interferencia, faltas de claridad y funcionan a una resolución limitada.

Una segunda salida de video que tiene la Raspberry es la conocida como Display Serial Interface (DSI), que se utiliza en los monitores de pantalla plana de las Tabletas y Smartphones.

Una mejor calidad de imagen puede obtenerse usando el conector HDMI (High Definition Multimedia Interface). A diferencia de la conexión analógica de video compuesto, el puerto HDMI proporciona una conexión digital de alta velocidad para mostrar imágenes de píxeles perfectos tanto en monitores de PC como en televisores de alta definición.



Figura 10 – Salidas de Video. Conector HDMI

Al utilizar el puerto HDMI, la Raspberry Pi puede desplegar imágenes a la resolución de 1920×1080 Full HD. A esta resolución, el detalle sobre la pantalla es significativamente superior.

Para el presente proyecto nos interesa la salida de video HDMI, por su capacidad de obtener la mejor calidad de video e imagen, aprovechando al máximo las prestaciones del hardware disponible.

2.2.4 Salidas de Audio

Para la salida de audio posee un conector de audio Jack de 3,5mm, además del propio HDMI. Si se está usando el puerto HDMI de la Raspberry Pi, obtener el audio es sencillo: cuando está configurado apropiadamente, el puerto HDMI transporta ambas señales, la de video y la de audio. Esto significa que conectando un único cable a la pantalla es suficiente para sacar video y audio. Si el display no tiene entrada HDMI, o no es capaz de reproducir audio, se tendrá que utilizar la salida de audio Jack.

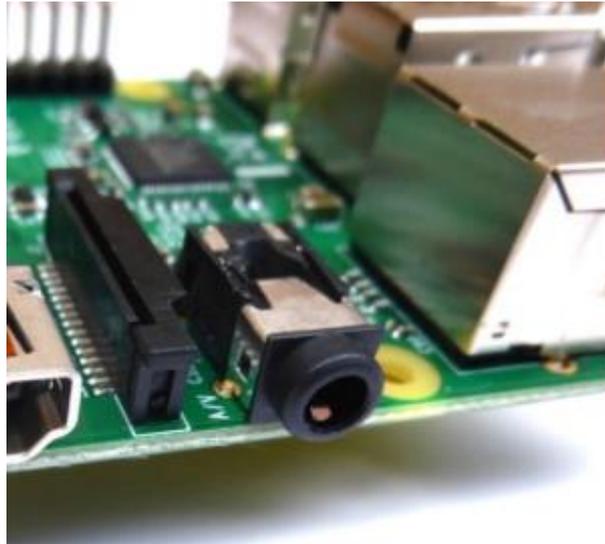


Figura 11 – Salidas de Audio. Conector Jack 3.5mm

Este conector es un zócalo de 4 polos que transmite señales de audio y video como se comentó anteriormente, el mismo es similar a los enchufes que se encuentran en otros dispositivos multimedia, como iPod, reproductores de MP3 y teléfonos inteligentes.

Este tipo de conector se lo conoce comúnmente como "TRRS". Donde los cuatro conductores del mismo llevan video, audio izquierdo, audio derecho y tierra.

2.2.5 Pines GPIO

Todas las Raspberry Pi de 40 pines (modelos B 2, 3 y Zero) comparten la distribución de pines. Se pueden conectar los dispositivos directamente o a través de algunos accesorios que los reproducen en una protoboard.

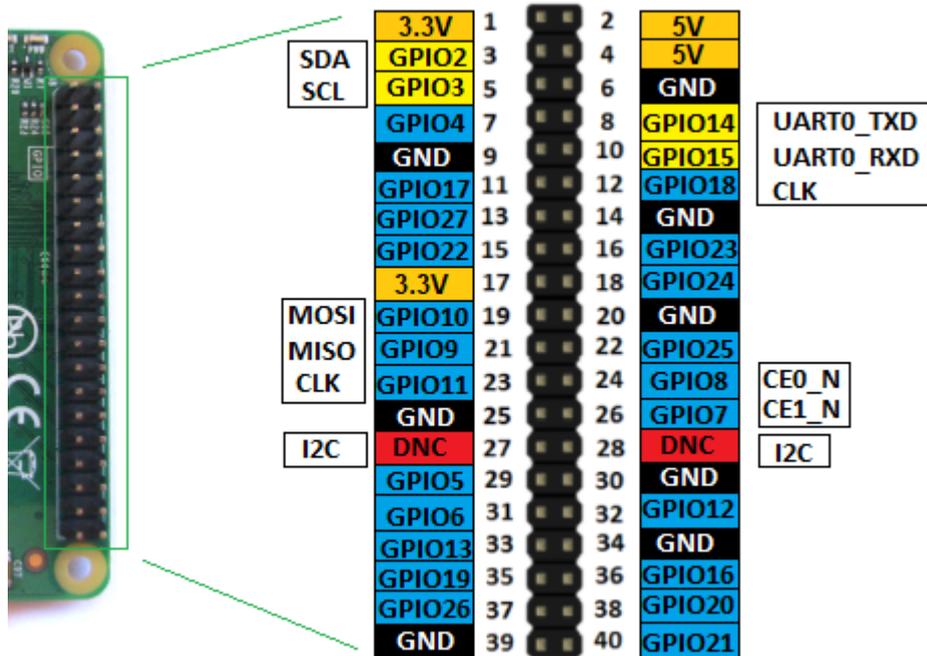


Figura 12 – Pines GPIO. Pinout de los 40 pines

La principal diferencia de estos pines con los de, por ejemplo, Arduino u otros microcontroladores respecto de su interacción con el mundo físico es que las Raspberry

no tienen entradas analógicas ni salidas PWM. Para añadir esas funciones, y controlar motores u otros dispositivos, necesitaremos placas externas las llamadas HAT (Hardware Attached on Top), similares a las shields de Arduino. Algo que sí posee son las interfaces de comunicación: I2C, SPI y UART (puerto serie).

Otra diferencia fundamental frente a modelos como Arduino es que su voltaje de funcionamiento es 3.3V en lugar de 5V. Por lo que necesitaremos adaptar los niveles lógicos al recibir la información de sensores o dispositivos que entreguen 5V.

2.2.6 Control remoto infrarrojo

El módulo utilizado recibe el comando de un control remoto infrarrojo y puede controlar 4 salidas discretas mediante el accionamiento de los botones del mismo.



Figura 13 – Control remoto infrarrojo. Mando y receptor

Este producto está pensado para ser “Plug & Play” por lo tanto, no es necesario ningún programa o librería especial, el simple conexionado de sus pines es suficiente.

Especificaciones técnicas del circuito:

- > Voltaje de operación: 5V DC
- > Corriente de trabajo: 270mA
- > Señal de Control: TTL (3.3V o 5V)
- > Nº de canales a manejar con este control: 4 CH
- > Tiempo de acción: 5-10 ms

Se puede encontrar de manera comercial y está pensado para encender/apagar módulos de Relés a distancia, con un alcance probado de 5 metros.

Este módulo posee un microcontrolador encargado de recibir la señal del receptor infrarrojo y accionar las salidas correspondientes según sean seleccionadas.

Estudiando los microcontroladores de la serie NuMicro Mini51 se puede observar que poseen distintas características según el modelo, la Figura 14, a continuación, describe las principales diferencias entre cada una de ellas.

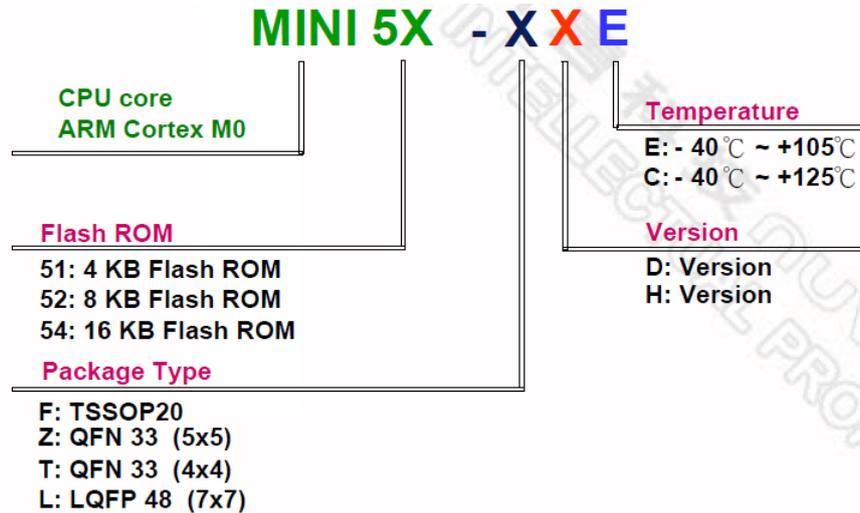


Figura 14 – Control remoto infrarrojo. Código de letras: NuMicro Mini5X³

El dispositivo ensambla en su circuito un microcontrolador NuMicro MINI54-FDE con un encapsulado de montaje superficial TSSOP20 de 6.5mm de largo por 4.4mm de ancho.

Dentro de las características más importantes se pueden mencionar que posee un núcleo ARM Cortex-M0 que puede funcionar hasta 24MHz, 16KB de memoria APROM, 2KB de memoria RAM, 2 Timers de 32 bits, 4 canales ADC de 10 bits y conectividad UART, SPI e I²C. Teniendo en cuenta que la tarea que realiza el procesador es bastante sencilla, ya que simplemente evalúa la entrada y activa las salidas, las características antes mencionadas son suficientes para realizar la tarea de una manera efectiva y fiable.

2.2.7 Conversor de nivel lógico

Para poder recibir la señal del control remoto infrarrojo se utiliza un adaptador de nivel unidireccional, que posee 4 pines que pueden tomar 5V y convertirlos a 3.3V:

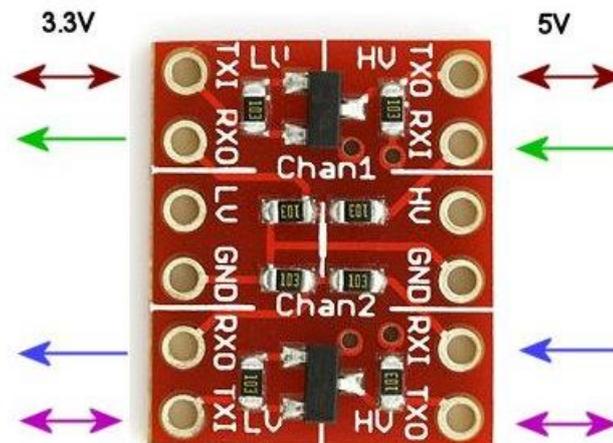


Figura 15 – Conversor de nivel lógico. Placa del adaptador

Los pines RX son unidireccionales (5V a 3.3V), en cambio, los pines TX son bidireccionales. El adaptador necesita tanto 5V como 3.3V de alimentación en cada lado respectivamente, además de la apropiada referencia a tierra.

Este dispositivo se utiliza para proteger los pines del procesador de la Raspberry y evitar sobretensiones.

³ NuvoTon (2014). NuMicro MINI51 DE Series Datasheet. Recuperado de: <https://www.alldatasheet.vn>

2.2.8 Módulo de Relevadores

La placa contiene 8 Relés fabricados por Songle, capaces de manejar cargas de hasta 250V de tensión y 10A de corriente. Cada canal posee aislamiento eléctrico por medio de un optoacoplador e indica su estado mediante un led. Se activa la salida normalmente abierta (NO: Normally Open) al recibir un "0" lógico y desactiva la salida con un "1" lógico.

Su diseño facilita el trabajo con Raspberry Pi, al igual que con muchos otros sistemas como Arduino, Módulos ESPxxx o PIC. Para la programación se utilizan una serie de timers mediante los cuales se decide cuando se activan y desactivan cada uno de los relevadores.

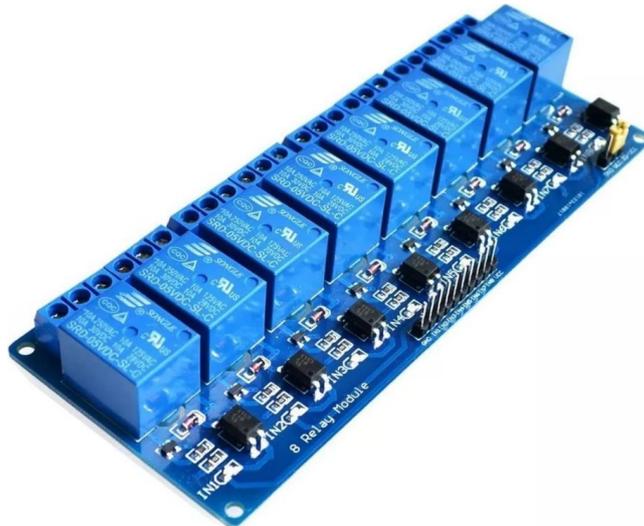


Figura 16 – Módulo de Relevadores. Disposición de componentes

Entre las cargas que se pueden manejar tenemos: bombillas de luz, luminarias, motores AC (220V), motores DC, solenoides, electroválvulas, calentadores de agua y una gran variedad de actuadores más. Por esto, este dispositivo será el encargado de activar y desactivar los dispositivos de la sala (luces, aromatizadores, burbujas, etc.) en los tiempos estipulados por la CPU.

2.3 Programas y herramientas utilizadas

Para completar el proyecto, es necesario desarrollar aplicaciones que sean capaces de configurar y controlar la automatización, para esto se utilizaron distintos programas, herramientas y entornos de desarrollo. Para la aplicación de la Raspberry Pi se utilizó el Lenguaje Python y la herramienta "PyQt5", en cambio, para realizar el programa para la PC se utilizó el Lenguaje C# y el entorno de desarrollo "Visual Studio 2019".

2.3.1 Sistema Operativo "Raspbian"

El sistema operativo utilizado en la Raspberry Pi está basado en Debian, una de las distribuciones GNU/Linux más importantes, considerado uno de los sistemas operativos más estables del mundo. Raspbian está altamente optimizado para la línea de procesadores ARM de bajo performance que posee este dispositivo. Fue creado por Mike Thompson y Peter Green como un proyecto independiente y todavía se encuentra en desarrollo constante. Existen múltiples versiones de Raspbian, incluyendo Raspbian Buster y Raspbian Stretch.

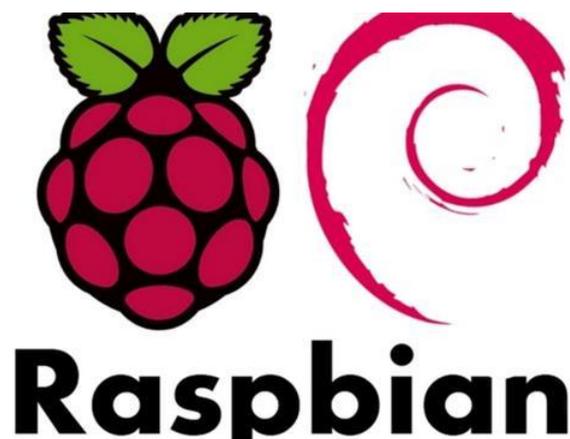


Figura 17 – Sistema Operativo “Raspbian”. Logo de Raspbian Stretch

La versión utilizada, Raspbian Stretch, cuenta con el kernel (núcleo) Linux 4.14 LTS de tipo Monolítico, el cual permite un gran manejo de memoria física y virtual, manejo heterogéneo de memoria para GPU y soporte completo de HDMI CEC para la Raspberry Pi. Emplea PIXEL como su principal entorno de escritorio, el cual está compuesto por un escritorio LXDE modificado y el controlador de ventanas Openbox. Lo más importante es que, al estar diseñado particularmente para este dispositivo, posee los controladores (drivers) específicos y completa compatibilidad con el Hardware que posee. Además, incluye pre-instaladas varias herramientas de uso rápido, como ser un Navegador Web liviano como Chromium, un reproductor multimedia, un visor de imágenes, un software de álgebra como Mathematica, entre otras.

2.3.2 OMXPlayer

Con la instalación base de Raspbian viene incluido un reproductor multimedia, llamado OMXPlayer, desarrollado por Edgar Hucek como parte de su Proyecto Kodi.

Puede ser comandado mediante línea de códigos y posee la capacidad de reproducir muchos de los formatos más populares de audio y video. Usa una aceleración de hardware denominada OpenMAX, la cual es oficialmente soportada por Raspberry Pi.

Para utilizarlo simplemente basta con invocar el comando:

```
omxplayer -opciones <nombre del archivo multimedia>
```

Resulta de especial interés notar que es el reproductor oficial y está incluido en el sistema operativo Raspbian, además que brinda la capacidad de invocarlo mediante línea de código, directamente desde la aplicación desarrollada en Python.

2.3.3 FEH

FEH es un visor de imágenes desarrollado por Tom Gilbert y Daniel Friesel. Es muy liviano, rápido y puede ser administrado mediante línea de comandos.

Puede mostrar imágenes HD, utiliza muy pocos recursos y posee todas las funcionalidades que se necesitan a la hora de realizar la tarea.

Se instala muy fácilmente mediante el comando de súper-usuario:

```
sudo apt-get install feh
```

Luego, puede ser invocado desde la aplicación desarrollada mediante:

```
feh -opciones <nombre del archivo de imagen>
```

2.3.1 HostApd y DNSmasq

Antes de discutir estas herramientas, se debe caracterizar una importante nomenclatura que se utilizará, el concepto de daemon. En sistemas operativos que realizan múltiples tareas, un “daemon” (en Linux) o “servicio” (Windows) es un tipo especial de proceso informático que no interactúa con el usuario, es decir, se ejecuta en segundo plano. Usualmente se ejecutan durante el arranque del Sistema Operativo.

Para garantizar la conectividad inalámbrica al CPU de control de la sala, se utilizará el chip Wi-Fi que posee la Raspberry Pi para generar un “Hotspot Wi-Fi”. Para esto se necesita crear una red inalámbrica, y permitir el acceso de dispositivos a la misma.

HostApd es un daemon utilizado para crear puntos de acceso Wi-Fi (IEEE 802.11) y servidores de autenticación (WPA/WPA2/EAP, servidores de autenticación RADIUS, servidores EAP, etc.). Esto es necesario para que los dispositivos sean capaces de conectarse a la red que genera la CPU.

Instalaremos HostApd con el siguiente comando de súper-usuario:

```
sudo apt-get install hostapd
```

DNSmasq es daemon que puede generar un servidor DNS (resolución de nombres), DNS Caching (resolución de nombres almacenando consultas en Caché), TFTP (protocolo de transferencia de archivos) y DHCP (distribución automática de parámetros de configuración de red). Esto es necesario para permitir que la Raspberry asigne dinámicamente una dirección a cada dispositivo que se conecte a ella.

Para instalar DNSmasq utilizaremos el siguiente comando (se necesitan permisos root):

```
sudo apt-get install dnsmasq
```

Estos pasos todavía no son suficientes para lograr nuestra red Wireless. Se realiza la configuración de los parámetros de red, estableciendo a la Raspberry Pi para actuar como un enrutador, siendo capaz de recibir conexiones entrantes y asignar direcciones de red (IPv4) dinámicamente.

Para este proyecto, la red configurada tiene como nombre “Sala Snoezelen” (SSID):

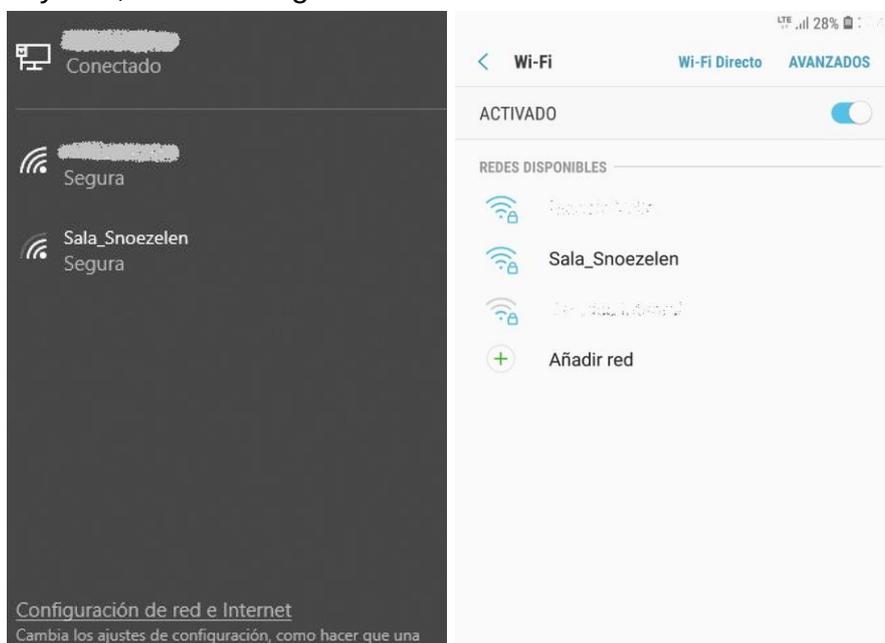


Figura 18 – HostApd y DNSmasq. Red Wi-Fi vista desde Windows 10 y Android

Actúa como un Punto de Acceso Wi-Fi y asigna direcciones IPv4, de clase C, dentro del rango de red comprendido dentro de 192.168.200.x. Esta red es capaz de brindar servicio de internet a los dispositivos conectados a ella, siempre que la Raspberry Pi posea una conexión activa a internet en su puerto Ethernet.

2.3.2 Lenguaje Python y Herramienta PyQt5

Python es un lenguaje de programación interpretado que hincapié en la legibilidad de su código, se trata de un lenguaje multi-paradigma, es decir, puede estar enfocado a Programación Orientada a Objetos, Programación Imperativa o Programación Funcional. Uno de sus puntos más fuertes es que es compatible con múltiples plataformas de desarrollo. Posee una licencia de código abierto que es compatible con la licencia pública general de GNU. Este lenguaje de programación se ha clasificado constantemente entre los 5 más populares desde 2018, compitiendo con Java, C y C++.

Las principales diferencias que se encuentran con otros lenguajes es que no usa corchetes para delimitar bloques, las declaraciones de final de línea (punto y coma) son opcionales y tiene mucho menos excepciones sintácticas que otros lenguajes, como C o Pascal. Por ejemplo, para crear una variable, no es necesario declarar su tipo ya que el intérprete de Python es lo suficientemente inteligente para hacerlo en tiempo de compilación. Por todas estas facilidades, y además porque Raspberry Pi ha adoptado este lenguaje por defecto, se ha decidido realizar en este lenguaje el desarrollo de la aplicación para este dispositivo.



Figura 19 – Lenguaje Python y Herramienta PyQt5. Logo de PyQt

Ahora, en cuanto a la herramienta PyQt5 podemos decir que es un módulo que se utiliza para crear interfaces gráficas de usuario (GUIs) y compatibilidades directas con el entorno de desarrollo Qt v5, con más de 600 clases y 6000 funciones disponibles. Se implementa con más de 35 módulos de extensión y permite que Python se utilice como lenguaje de desarrollo en aplicaciones con interfaz de usuario en todas las plataformas incluidas, ya sea Raspbian (Linux), iOS, Android o Windows.

2.3.3 Lenguaje C# y Visual Studio 2019

C# es un lenguaje de programación multi-paradigma, desarrollado por Microsoft como parte de su plataforma .NET. Es uno de los lenguajes desarrollados para la infraestructura de lenguaje común, es decir, permite que las aplicaciones escritas en este lenguaje puedan luego ejecutarse en distintas plataformas, tanto de software como de hardware, sin necesidad de reescribir o recompilar su código fuente.

Su sintaxis deriva de la estructura básica de C/C++, pero además agrega objetos de la plataforma .NET con mejoras provenientes de otros lenguajes. Aunque C# surge como parte de la plataforma .NET, es un lenguaje de programación independiente y ya posee un compilador implementado que provee el marco Mono-DotGNU, el cual permite generar programas para distintas plataformas, como Windows, Unix, Android, iOS, entre otros.



Figura 20 – Lenguaje C# y Visual Studio 2019. Logos de C# y Visual Studio

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE) para Windows, Linux o MacOS, compatible con 36 lenguajes de programación distintos (C++/C#, VB .NET, Java, Python, y más), y con entornos de desarrollo web, como ASP.NET o Django.

En el presente proyecto se utilizó esta herramienta para el desarrollo de la aplicación de Windows, en su versión “Community” que es una versión gratuita para desarrolladores independientes, solo es necesario una cuenta Microsoft para su utilización.

Es importante destacar que, para la realización de la aplicación para la PC, se utilizó la interfaz de programación de aplicación gráfica (API) llamada “Windows Forms” que se incluye como parte de Microsoft .NET Framework y proporciona acceso a los elementos de la interfaz gráfica de Microsoft Windows a partir de la versión .NET 2.0 del Framework.

2.4 Desarrollo de aplicación para Raspbian

El desarrollo de la aplicación en lenguaje Python se realizó en 3 etapas.

1º. Utilizando el entorno de desarrollo IDLE, que viene incluido por defecto en Raspbian,

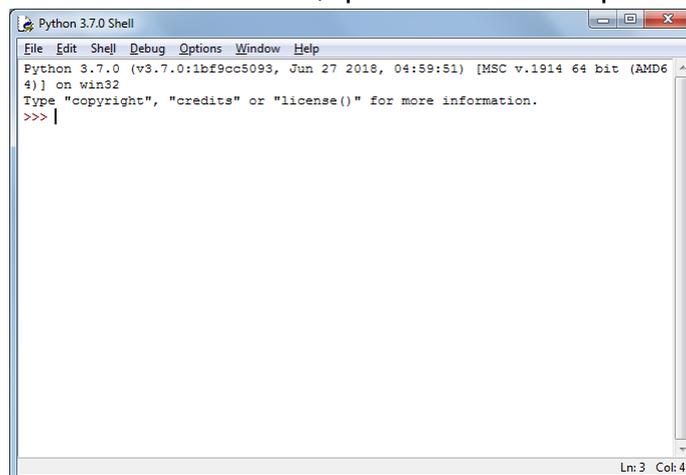


Figura 21 – Desarrollo de aplicación para Raspbian. Entorno de desarrollo IDLE

Se crea un archivo de configuración que se utiliza a modo de librería personalizada por la interfaz gráfica desarrollada en la segunda etapa.

En este archivo se preestablecen los pines que se utilizarán como entradas y salidas:

```
#PINES
in1=17, in2=18, in3=22, in4=23
relay1=5, relay2=6, relay3=12, relay4=13, relay5=19, relay6=20, relay7=21, relay8=26
```

Además, almacena las banderas utilizadas para evaluar el estado de las entradas provenientes del control remoto infrarrojo o la selección del usuario en pantalla.

```
#BANDERAS
bandera_1 = True
bandera_2 = True
bandera_3 = True
bandera_4 = True
```

Para el control de las entradas/salidas GPIO se utiliza la librería estándar llamada RPI.GPIO, la cual contiene funciones que inicializan, leen o establecen el comportamiento de los pines de la Raspberry Pi.

```
import RPi.GPIO as GPIO
```

Se utilizan interrupciones para detectar la interacción del usuario con el control remoto:

```
def activar_int():
    if bandera_GPIO:
        try:
            global bandera_1,bandera_2,bandera_3,bandera_4
            bandera_1=GPIO.input(in1)
            bandera_2=GPIO.input(in2)
            bandera_3=GPIO.input(in3)
            bandera_4=GPIO.input(in4)
            GPIO.add_event_detect(in1, GPIO.BOTH, callback=checkear)
            GPIO.add_event_detect(in2, GPIO.BOTH, callback=checkear)
            GPIO.add_event_detect(in3, GPIO.BOTH, callback=checkear)
            GPIO.add_event_detect(in4, GPIO.BOTH, callback=checkear)
        except:
            print ("No se pudo activar Interrupciones")
    else:
        print ("No se inicializaron los GPIO")
```

Este código provoca que la función de callback, “checkear”, sea llamada cada vez que una de las entradas cambia de estado. Dentro de esa función se realiza la evaluación de cada una de los pines y se establecen los valores de las banderas, hasta que la siguiente interrupción ocurra, repitiendo la evaluación.

Cuando el usuario no puede (o no desea) utilizar el control remoto, debemos tener la opción de deshabilitar estas interrupciones y resetear las banderas:

```
def desactivar_int():
    if bandera_GPIO:
        try:
            global bandera_1,bandera_2,bandera_3,bandera_4, bandera_Todos
            bandera_Todos = False
            GPIO.remove_event_detect(in1)
            bandera_1=True
            GPIO.remove_event_detect(in2)
            bandera_2=True
            GPIO.remove_event_detect(in3)
            bandera_3=True
            GPIO.remove_event_detect(in4)
            bandera_4=True
        except:
            print ("No se pudo desactivar Interrupciones")
    else:
        print ("No se inicializaron los GPIO")
```

Se verá luego que se puede utilizar la aplicación de interfaz gráfica para controlar las banderas de manera manual, mediante botones en pantalla.

Además de las mencionadas, esta librería contiene las siguientes funciones:

def iniciar_GPIO(): Permite inicializar todos los pines GPIO utilizados por el sistema como entradas o salidas según corresponda.

def limpiar_GPIO(): Permite resetear todos los pines GPIO y sus estados. Es necesario inicializarlos nuevamente para volver a utilizarlos.

def leer_banadera(num): Devuelve el estado de la bandera “num”, utilizado en cada interrupción para evaluar la selección del usuario.

def leer_relay(num): Devuelve el estado de la salida “num”, sirve para controlar el estado de cada relé en tiempo de ejecución.

def cambiar_relay(num): Permite cambiar el estado de la salida que se pasa como referencia “num”. Esto se utiliza para activar/desactivar los relés de salida.

Paralelamente a la librería de configuración personalizada, se ha creado un script (llamado proceso2.py) que es el encargado de leer las rutinas configuradas en la Raspberry Pi. Este proceso se encarga de abrir el archivo que contiene los datos guardados, tomarlos y ejecutar la rutina seleccionada por el usuario.

```
with open(archivo, 'r') as f:
    lector = csv.reader(f, delimiter=',')
    for linea in lector:
        diccionario.update({linea[0]:linea[1]})
```

El código anterior abre el archivo de configuración de la rutina a reproducir y guarda en memoria todos los datos que este contiene. Luego, reproduce los archivos multimedia o muestra imágenes mediante subprocessos utilizando el módulo llamado “subprocess” que permite utilizar la línea de comandos para disparar un proceso en segundo plano:

```
if os.path.exists(video_path):
    proceso1 = subprocess.Popen(['omxplayer', '-b', '-o', 'both', video_path])
```

Este código ejecuta en segundo plano el reproductor multimedia OMXPlayer y lo comanda a reproducir el archivo de audio o video que corresponda. Luego continúa con el hilo principal del programa y realiza la activación o desactivación de los relevadores:

```
for relay in accion:  
    configuracion.cambiar_relay(int (relay))
```

A continuación, espera el tiempo estipulado de cada paso antes de continuar con el proceso de lectura de los datos alojados en memoria. Por esto, en el siguiente ciclo, realiza una nueva evaluación de las acciones a realizar y actúa en consecuencia, ya sea, mostrando una nueva imagen o reproduciendo un nuevo archivo multimedia, mientras activa o desactiva las salidas correspondientes.

Es de especial interés mencionar que el script “proceso.py”, encargado de llevar a cabo las acciones que el usuario ha configurado y seleccionado, pero además posee una “rutina por defecto” que se usa para alertar al usuario sobre la imposibilidad del sistema de encontrar la rutina configurada que se ha seleccionado. Además, este script es ejecutado como un subprocesso, lo que permite interrumpir su ejecución si el usuario decide parar la rutina. Esto se programó de esta manera para brindar al usuario un mayor control sobre la sesión de terapia, ya que, con la simple acción de volver a presionar el botón de selección de rutina en el control remoto infrarrojo, puede eliminar los estímulos que actualmente estén presentes en la sala.

2º. Utilizando el entorno de desarrollo Qt Designer, que forma parte de las utilidades y aplicaciones incluidas en el paquete de herramientas de PyQt5,

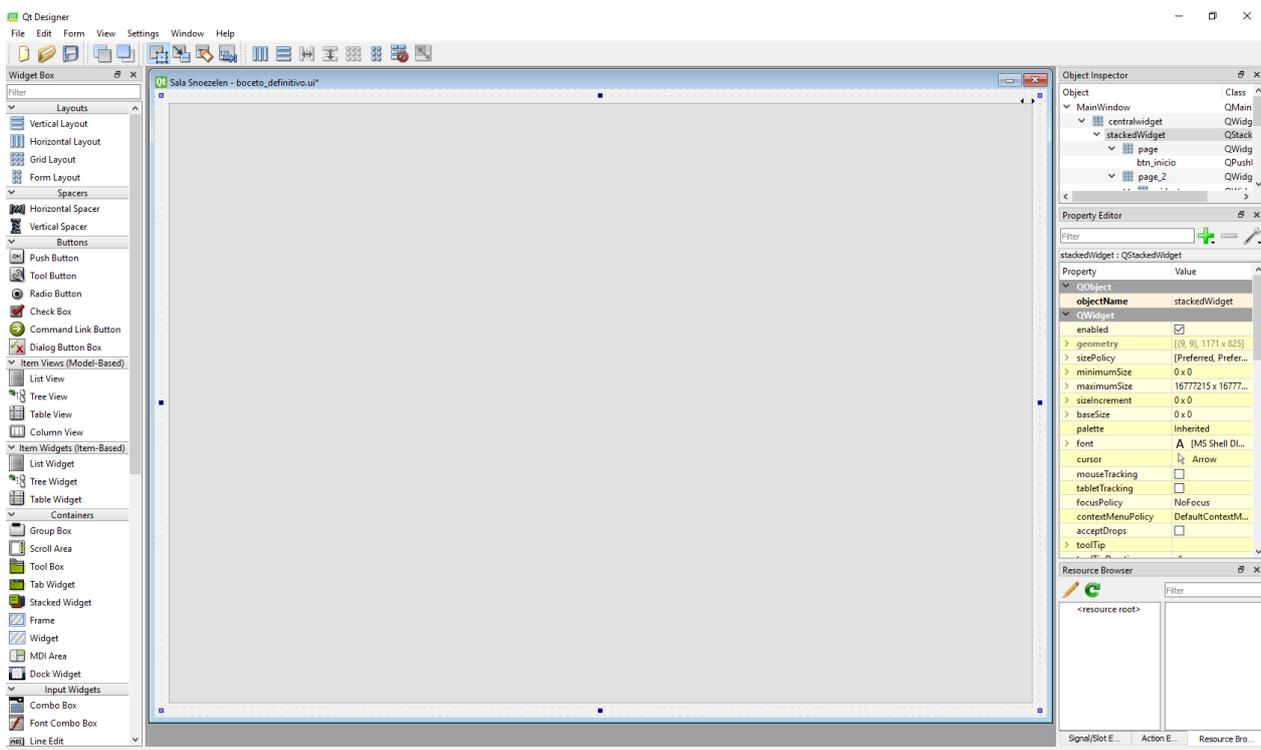


Figura 22 – Desarrollo de aplicación para Raspbian. Entorno de desarrollo Qt Designer

Se ha creado una interfaz gráfica que posee 3 pantallas que llamaremos:

- **Control Automático:** En esta sección las interrupciones del control remoto infrarrojo se encuentran activadas y todo se realiza sin intervención de ningún otro periférico. Esta pantalla solo muestra un botón que ocupa casi la totalidad de la pantalla, el cual al ser presionado desactivará el modo Automático y mostrará el teclado en pantalla.
- **Control de Rutinas:** Al utilizar un ratón para presionar el botón de la pantalla de “Control Automático” se desactivan las interrupciones del control remoto infrarrojo y se despliegan 10 botones en pantalla. 2 de los cuales serán utilizados para regresar al modo de control automático o cambiar al modo manual. En este modo se utilizan 5 botones, 4 para la selección de las rutinas configuradas y el 5to para apagar el dispositivo.
- **Control Manual:** Solo se puede acceder a este modo a través de la pantalla de “Control de Rutinas”, por lo que se necesita un ratón USB conectado a la Raspberry Pi. En este modo se da la posibilidad al usuario de activar/desactivar las salidas de forma individual, con un pequeño retraso de 2 segundos entre cada acción para evitar el mal uso del mismo. Puede ser usado para testear los dispositivos o realizar rutinas de estimulación de forma manual.

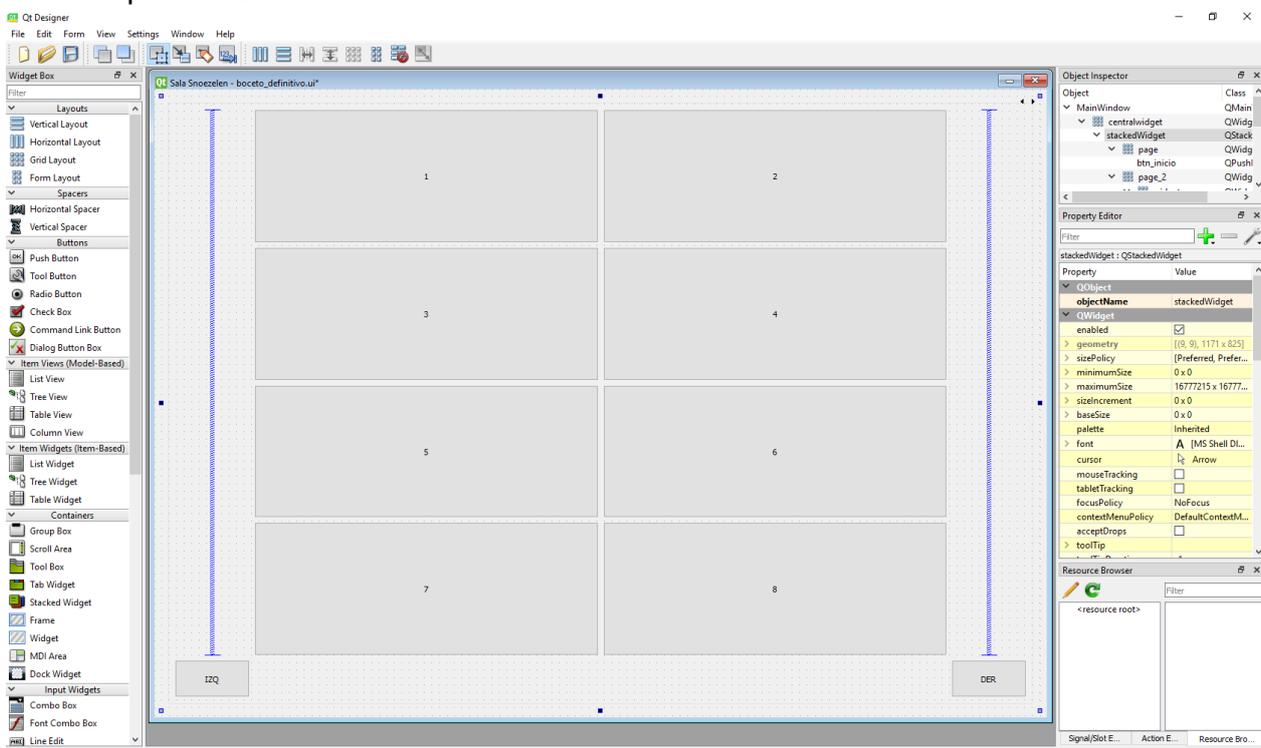


Figura 23 – Desarrollo de aplicación para Raspbian. Prototipo del diseño del control en pantalla

Una vez finalizada la creación del proyecto en Qt Designer, es necesario convertir el archivo `boceto_definitivo.ui` al código en Python. Para esto usamos la herramienta incluida en la carpeta Scripts del intérprete de Python y escribimos:

```
pyuic5 -x boceto_definitivo.ui -o boceto_definitivo.py
```

Esto genera un archivo de extensión “.py” que podremos luego compilar y ejecutar para obtener la interfaz gráfica en pantalla. En la Figura 24, a continuación, podemos ver la localización de la aplicación que se utiliza para este fin.

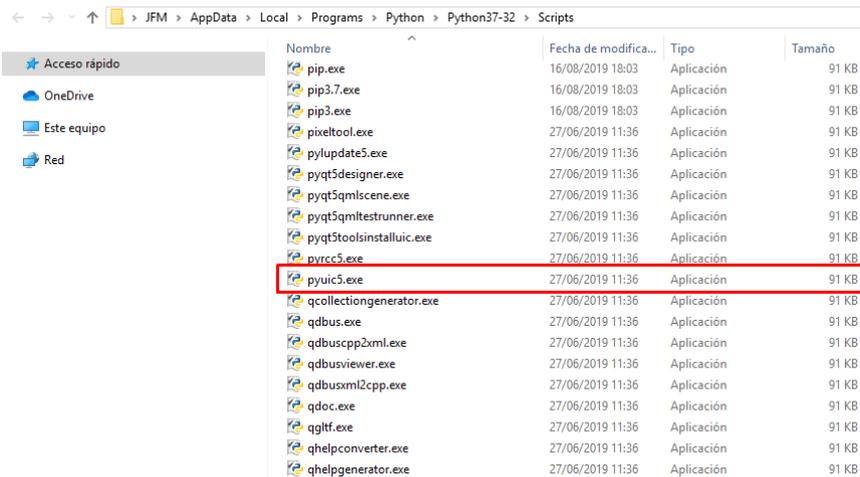


Figura 24 – Desarrollo de aplicación para Raspbian. Aplicación para conversión de UI a Python

Este último paso es muy importante ya que luego, en la 3er etapa, se combina parte del código escrito en la 1er etapa en un archivo que pueda ser compilado y ejecutado.

- 3º. Una vez que se obtiene la interfaz gráfica en formato Python puede modificarse el código para suplir las necesidades de la aplicación. Aquí es donde se realizan las modificaciones necesarias para leer o modificar el estado de los pines de la Raspberry, ejecutar los subprocesos en segundo plano y al mismo tiempo ser capaz de tener una interfaz gráfica que permita al usuario interactuar con el sistema, si así lo desea. Las librerías necesarias para la aplicación son:

```
from PyQt5 import QtCore, QtGui, QtWidgets
#LIBRERIAS ESTANDAR
import subprocess
import sys
import os
import signal
#MIS LIBRERIAS
import configuracion
```

Las primeras son usadas por PyQt5 para generar la interfaz gráfica. Las que aparecen como “librerías estándar” son módulos importados que permiten manejar procesos paralelos (“subprocess”), interactuar con el sistema (“sys” y “os”) y enviar señales a otros scripts o aplicaciones (“signal”). Por último, y como ya se ha mencionado, se utiliza el archivo “configuración.py” como librería, lo que permitirá usar las funciones y variables personalizadas para controlar el comportamiento del sistema y la placa. En cuanto al desarrollo de esta aplicación, resultaría demasiado extenso explicar cada uno de las clases y funciones utilizadas en detalle, por lo que se mencionarán algunas de las modificaciones más importantes que se realizaron (al código creado por Qt Designer) para obtener la versión final de la aplicación. Primero debemos asegurarnos que la interfaz gráfica ocupe la totalidad de la pantalla, por ello, la inicializamos en modo de pantalla completa:

```
MainWindow.showFullScreen()
```

Para asegurar que la pantalla (o proyector) HDMI no interfiere en la sesión de estimulación, es necesario que ésta parezca estar apagada, para esto se establece como estilo de la aplicación el color negro:

```
MainWindow.setStyleSheet("""background-color: Black;""")
```

Al igual que el botón que ocupa la mayoría de la pantalla inicial:

```
self.btn_inicio.setStyleSheet("""background-color: Black; opacity:1;
                                font-size:30px; color: Black; border-style: solid;""")
```

Como podemos observar, el estilo de cualquier elemento GUI puede ser modificado mediante código de estilo CSS, este lenguaje es ajeno a Python, pero lo suficientemente sencillo para ser usado intuitivamente. El estilo del resto de los elementos de la aplicación fue modificado de esta forma, lo cual ofrece la capacidad de modificar color, tamaño, fuente y transparencia de cada uno de manera rápida, sin necesidad de modificar demasiado código.

Para poder utilizar funciones personalizadas con los elementos de la interfaz gráfica, es necesario establecer “conexiones” entre ellos.

```
#CONEXIONES:
self.btn_inicio.clicked.connect(self.cambiar_index)
self.btn_inf_izq.clicked.connect(self.cambiar_index)
self.btn_inf_der.clicked.connect(self.rutina_manual)

self.btn_1.clicked.connect(lambda: self.manejar_botones(1))
...
self.btn_8.clicked.connect(lambda: self.manejar_botones(8))

self.stackedWidget.currentChanged.connect(self.checkear_index)
```

Los primeros 3 botones son los encargados de cambiar de pantalla, estos llaman a las funciones encargadas de realizar estas acciones, las cuales a su vez se encargan de activar o desactivar las interrupciones, activar o desactivar el cursor del ratón o cambiar el estilo, texto y posición de los elementos en pantalla.

Las conexiones realizadas a través de “lambdas” permiten que la interfaz gráfica le envíe un valor a la función que se activa, lo que simplifica el código al evitar la necesidad de crear diferentes funciones para cada uno de los botones.

Por último, el “stackedWidget” posee un evento que es llamado cada vez que se produce un cambio de pantalla, esto es llamado recursivamente al presionar los botones que invocan a la función encargada de cambiar el índice.

Una parte importante del desarrollo consiste en que el sistema sea capaz de controlar el estado de las banderas que indican qué subproceso debe ser llamado en cada acción, para esto se utiliza un temporizador que es invocado en intervalos de 500ms:

```
timer = QtCore.QTimer()
timer.timeout.connect(contar)
timer.setInterval(500)
timer.start()
```

El temporizador siempre está corriendo en segundo plano, por lo que la función personalizada “contar” es llamada regularmente. Esta función es la encargada de manejar los retardos que sean necesarios y también de chequear el estado de las banderas y activar (o interrumpir) el subprocesso seleccionado.

Como se mencionó anteriormente, resulta importante correr los procesos en un hilo o subprocesso distinto al de la interfaz gráfica, esto es para evitar que los elementos UI, incluido el temporizador, queden inhabilitados durante la ejecución de una rutina. Para correr procesos en segundo plano se utiliza nuevamente la librería estándar “subprocess”:

```
if not bandera_proceso:
    proceso=subprocess.Popen(['python3', 'proceso2.py', 'rutina1/rutina1'])
    bandera_proceso = True
```

Este código solo se activa si no hay un proceso corriendo (si lo hay, se realiza otra acción en otra sección del programa). Esta función usa la aplicación que se le pasa como primer argumento ('python3') para generar un subprocesso con el script que se le pasa como segundo argumento ('proceso2.py'), el tercer argumento se usa recursivamente, es decir, Popen se lo pasa como argumento al script para que lo use dentro de su código. Dicho esto, este es el código para activar el subprocesso de la rutina 1, una de las 4 que pueden ser configuradas y modificadas por el usuario. Para realizar el apagado de la Raspberry se utiliza un simple comando de Linux llamado “shutdown”, aunque previamente se agrega un tiempo de espera de 30 segundos (60 cuentas del temporizador de 500ms):

```
elif configuracion.bandera_Todos:
    cuenta -= 1
    if bandera_proceso:
        os.kill (proceso.pid, signal.SIGINT)
        bandera_proceso = False
    if cuenta < 1:
        os.system("shutdown -h now")
    pass
if not configuracion.bandera_Todos:
    cuenta = 60
```

También en este paso nos aseguramos que no quede ningún subprocesso activo antes de iniciar la cuenta regresiva para el apagado, si existe alguno, se usa el comando “kill” para enviarle una señal de interrupción. Esto tiene como objetivo evitar que se apague accidentalmente la Raspberry en medio de una sesión de terapia.

Como podemos observar, la cuenta se reinicia en 60 si en alguna de las activaciones del temporizador el estado de las banderas ha cambiado.

Estas son las modificaciones más importantes que se han realizado al código de la aplicación principal, de todas maneras, el desarrollo completo consta de más de 1000 líneas de código si tenemos en cuenta la librería de configuración y el script que se utiliza como subprocesso en segundo plano.

Diagrama de flujo completo de la aplicación:

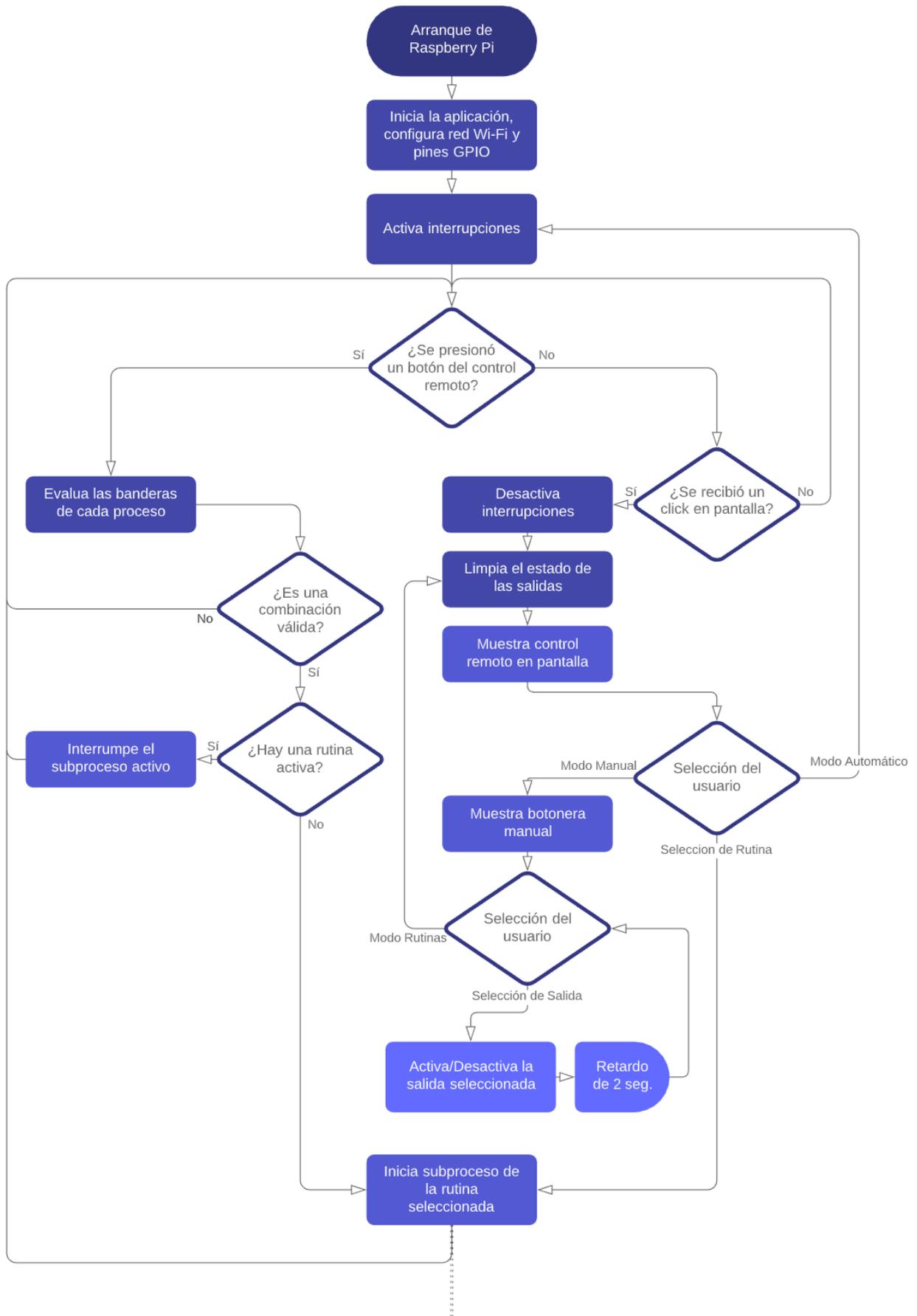


Figura 25 – Desarrollo de aplicación para Raspbian. Diagrama de flujo (Parte 1)

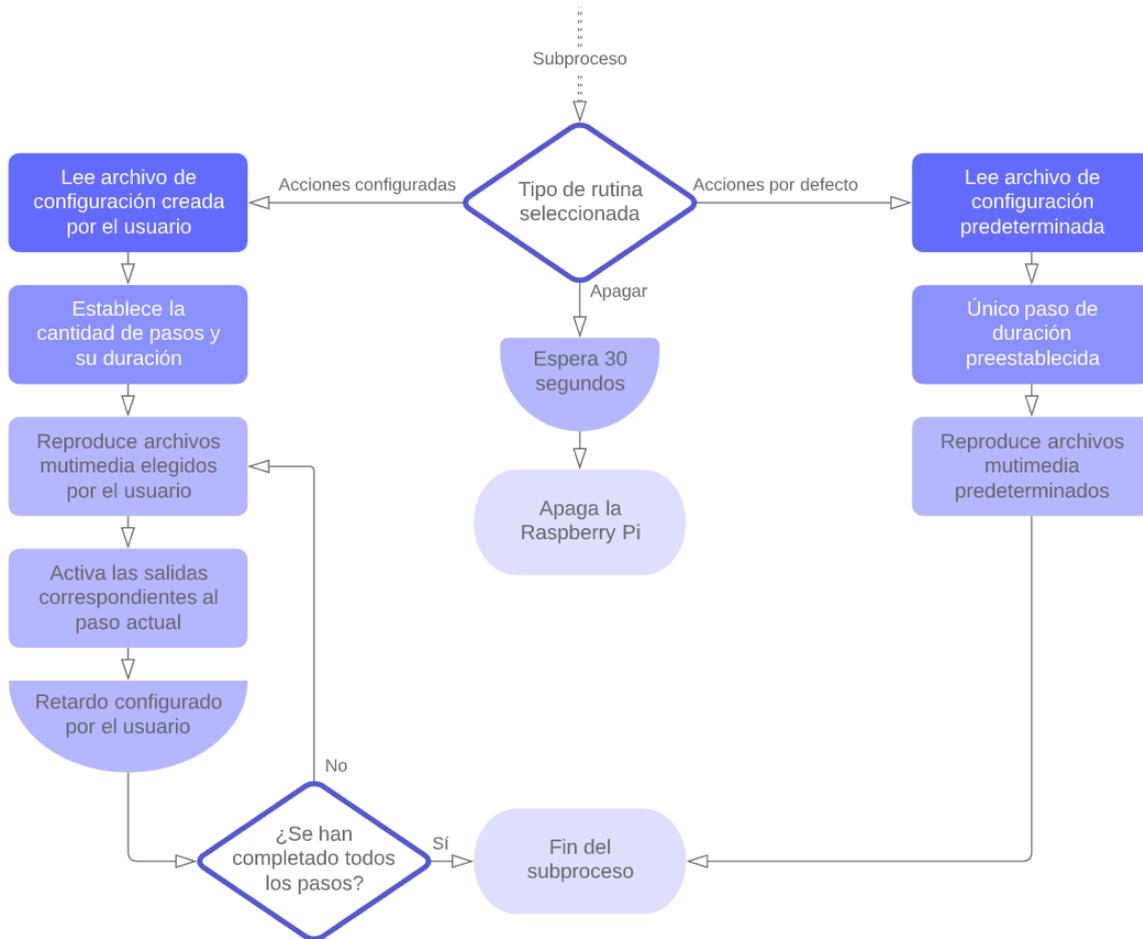


Figura 26 – Desarrollo de aplicación para Raspbian. Diagrama de flujo (Parte 2)

Por último, es importante mencionar que durante todo el desarrollo de esta aplicación se utiliza el intérprete de Python en su versión 3.7 ya que éste posee funciones, llamadas y declaraciones que son incompatibles con las versiones anteriores del mismo.

2.5 Desarrollo de aplicación para Windows

Para completar el sistema es necesario que el usuario sea capaz de crear y guardar sesiones de terapia en la CPU de la sala. Para esto es necesario crear una aplicación que posea una interfaz gráfica sencilla pero capaz de interpretar la selección del usuario y crear el archivo de configuración que necesita la aplicación que comanda la sala.

Debido a la experiencia previa con el lenguaje C# y con la interfaz de aplicación gráfica Windows Forms, se ha decidido utilizar Visual Studio 2019 como plataforma de desarrollo.

En la Figura 27, a continuación, se puede encontrar el entorno visual de desarrollo que ofrece Visual Studio. En el punto 1 se puede ver el cuadro de herramientas, estos son los elementos que podrán ser programados para interactuar con el usuario. En el punto 2 se encuentra el área de trabajo, aquí es donde se pueden colocar las herramientas y manipularlas visualmente. El punto 3 indica la ventana de propiedades, desde aquí se pueden modificar todas las propiedades visuales de cada uno de los elementos en el área de trabajo, también posee una pestaña de eventos en la que se puede definir distintos comportamientos, dependiendo de cada interacción específica con el usuario. Finalmente, en el punto 4, se encuentra la vista de consola, donde veremos los errores y advertencias de compilación y ejecución, o la salida si se ha definido algún mensaje de consola.

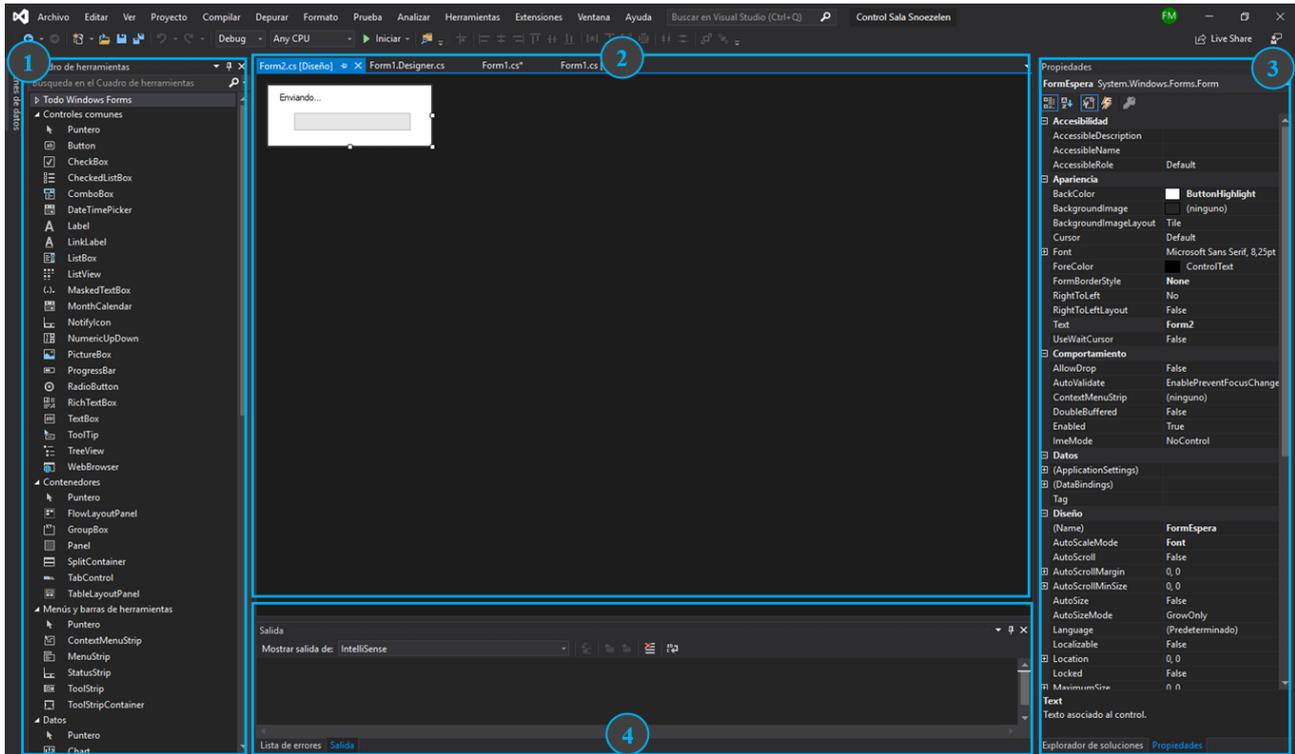


Figura 27 – Desarrollo de aplicación para Windows. Entorno de desarrollo Visual Studio 2019

El desarrollo de esta aplicación es más extenso que el anterior, ya que involucra el uso de herramientas avanzadas en cuanto a sus prestaciones y complejidad de código. Por este motivo, se mencionarán las características más importantes y destacables de la aplicación, sin ahondar en el código escrito.

En la Figura 28, a continuación, se puede observar el diseño final de la aplicación:

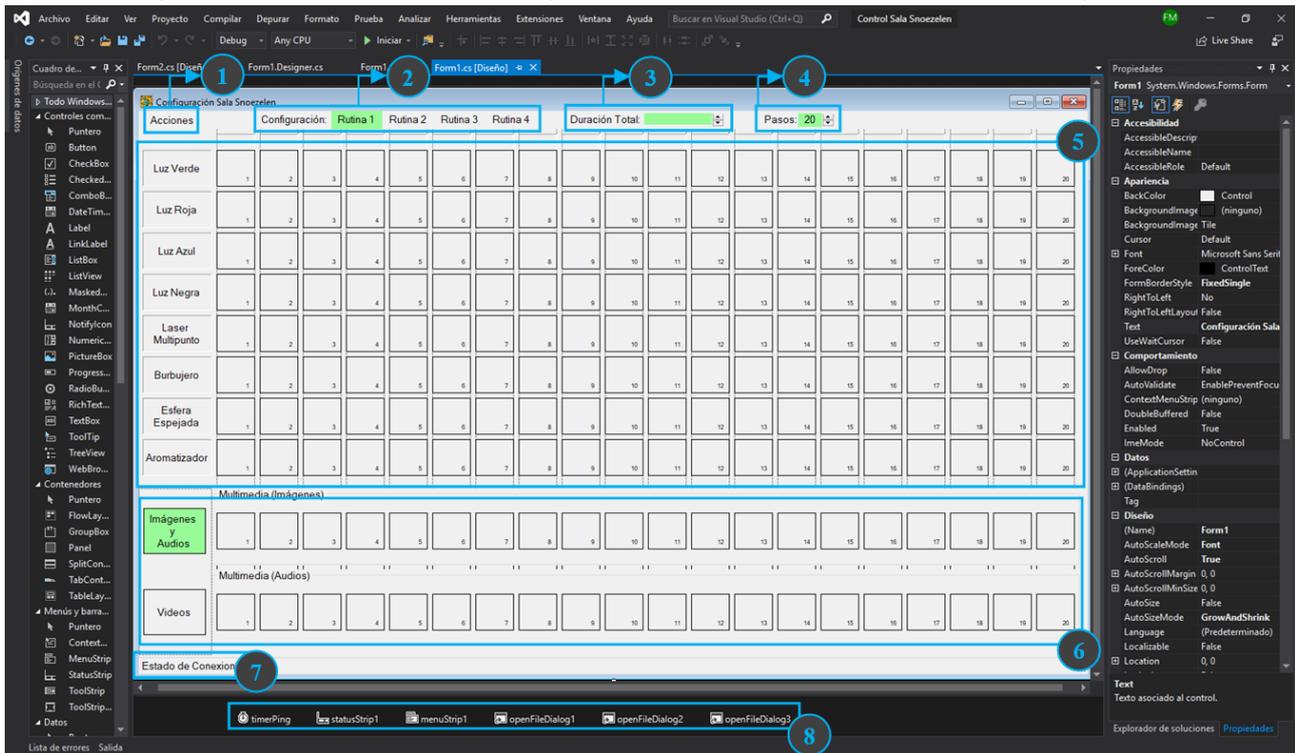


Figura 28 – Desarrollo de aplicación para Windows. Detalle de la interfaz de usuario

Se encuentran señalados los puntos más relevantes a mencionar dentro en el diseño de la aplicación. A continuación, se realizará una descripción en detalle de cada uno de ellos:

- (1) **Acciones:** Este es un menú desplegable que al recibir un click muestra 3 opciones.

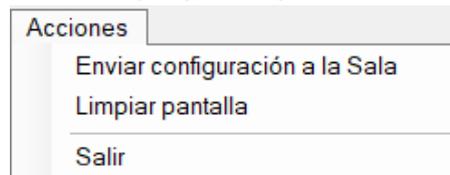


Figura 29 – Desarrollo de aplicación para Windows. Menú desplegable de acciones

Este será nuestro “menú principal” a la hora de realizar la tarea de configuración.

- > **Enviar configuración a la Sala:** A presionar sobre este menú el sistema pregunta al usuario si desea cargar, en la CPU de la sala, la configuración que se encuentra actualmente en pantalla.

Si el usuario confirma el envío el sistema procede de la siguiente forma:

 - 1º. Comprueba el estado de conexión con la red de la sala;
 - 2º. Evalúa cada uno de los elementos en pantalla, para realizar la traducción de la configuración ingresada por el usuario al código necesario para ser interpretado por el script de Python;
 - 3º. Comprueba la existencia de todas las rutas de archivos multimedia ingresadas por el usuario;
 - 4º. Genera un archivo temporal, en la carpeta “Temp” de Windows;
 - 5º. Escribe en el archivo, el código generado en el paso 2;
 - 6º. Si todos los pasos fueron exitosos, envía un comando SSH a la Raspberry Pi para que borre el contenido de la carpeta correspondiente a la rutina a cargar;
 - 7º. Inicia el envío de los archivos a través de la conexión de Wi-Fi, utilizando una conexión SFTP (protocolo seguro de transferencia de archivos);
 - 8º. Muestra un segundo Form, que posee una barra de progreso en marquesina, que indica que los datos están siendo enviados;
 - 9º. Luego de completada la tarea, el archivo temporal es eliminado.

El envío de los datos puede realizarse ya que la Raspberry Pi tiene habilitado su comunicación SSH, por lo que se puede usar la librería SSH.NET tanto para enviar comandos SSH como para abrir conexiones SFTP.
 - > **Limpiar pantalla:** Este menú es el encargado de reiniciar todos los controles de la pantalla, volviéndolos a la configuración que poseen al abrir la aplicación. Requiere confirmación del usuario.
 - > **Salir:** Al igual que el botón cerrar (la X del cuadro de sistema), este menú es el encargado de cerrar la aplicación y liberar los recursos de memoria.
- (2) **Configuración:** Este es el selector de rutinas, definirá cuál de las 4 rutinas se configurará. El control utilizado es lo que se denomina un RadioButton, es decir que permite seleccionar solo una de las 4 opciones y desmarca las demás.
- (3) **Duración total:** El selector de duración establecerá el tiempo total de la sesión de terapia. Es una combinación de 2 controles: Uno llamado NumericUpDown, el cual puede almacenar un valor entero y el usuario tiene la capacidad de modificarlo dentro de un rango predefinido. El segundo control es un TextBox, codificado para que tome el valor del control numérico, lo interprete y lo muestre como una cuenta de minutos. Tanto la modificación mediante los botones (arriba-abajo) o escribiendo el valor en pantalla producen la modificación del valor de duración. El rango de duración establecido es de 1 minuto a 2 horas, es decir, 120 minutos.

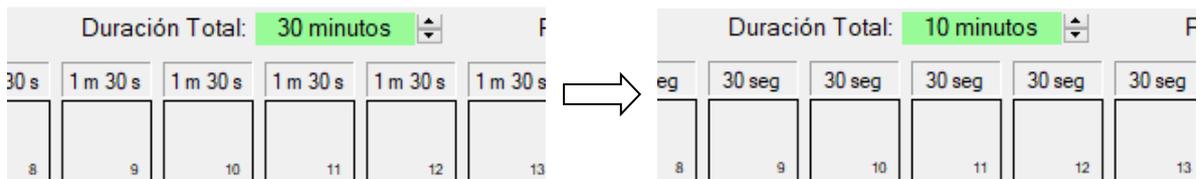


Figura 30 – Desarrollo de aplicación para Windows. Selector de duración

Una característica interesante, que se agregó a la interfaz gráfica, es la muestra dinámica de la duración de cada paso, por lo que, si modificamos el valor numérico de duración, se reflejará el cambio en el tiempo de cada uno de los pasos.

- (4) **Pasos:** Este es el selector de cantidad de pasos y define la cantidad de pasos tendrá nuestra sesión. El control utilizado es de tipo NumericUpDown, el cual se ha detallado previamente. El rango de pasos queda establecido entre 1 y 20.

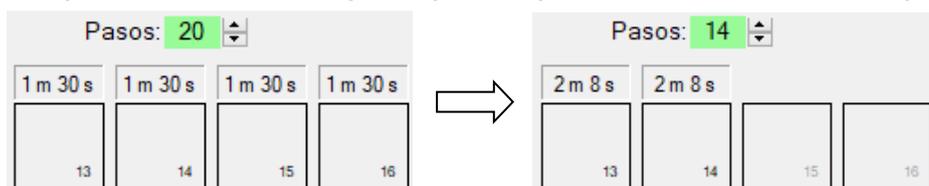


Figura 31 – Desarrollo de aplicación para Windows. Selector de cantidad de pasos

Este control también interactúa dinámicamente con el resto de la interfaz gráfica, al modificar la cantidad de pasos se inhabilita la columna de los pasos que no se utilizarán, y al mismo tiempo, se modifica el valor de tiempo de cada columna. Mediante los controles de duración total y cantidad de pasos, se logra una gran versatilidad en el manejo de los tiempos de sesión y también de cada estímulo.

- (5) **Área de configuración de estímulos físicos:** En esta sección se realiza la configuración de activación y desactivación de los relevadores en cada paso, como su nombre lo dice, configura los estímulos físicos que recibirá el paciente o alumno. Aquí se utilizan controles sencillos llamados CheckBox, que poseen solamente dos valores: Verdadero (Checked) o Falso (!Checked). Se han definido 8 estímulos específicos, con 20 columnas para cada uno. Estos representan a los dispositivos que estarán conectados a los relés y cableados alrededor de la sala de estimulación.



Figura 32 – Desarrollo de aplicación para Windows. Configuración de estímulos físicos

Estos controles fueron configurados para cambiar de color al posicionar el mouse sobre ellos, y también, al ser activados (clickeados). Una característica adicional es que si se realiza doble-click sobre el nombre del estímulo, (por ejemplo, Luz Verde) serán marcadas (o desmarcadas) todas las casillas disponibles para esa salida.

- (6) **Área de configuración de estímulos multimedia:** En esta sección se pueden seleccionar estímulos de tipo audio-visual, posee dos opciones, “Imágenes y Audio” o “Videos”. También se utilizan controles de tipo CheckBox, pero esta vez con un evento más al ser marcados, ya que abren un OpenFileDialog que permite seleccionar archivos. Esto le permite al usuario navegar a través de su dispositivo y buscar el estímulo que desea cargar en cada uno de los pasos.

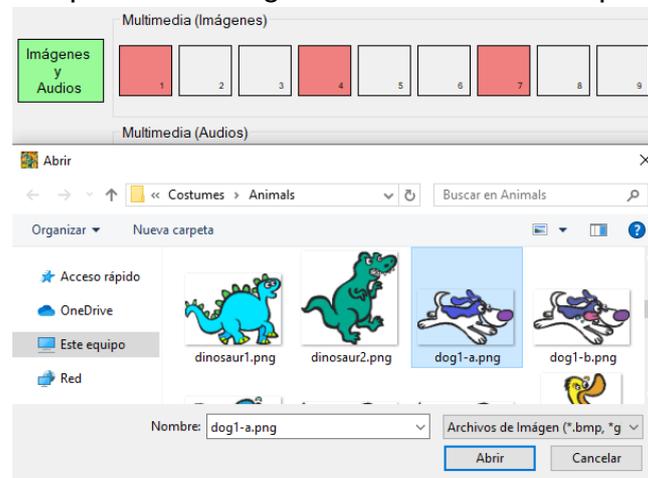


Figura 33 – Desarrollo de aplicación para Windows. Configuración de estímulos multimedia

Se han colocado filtros específicos para cada uno de los archivos, ya sean imágenes, audios o videos, por defecto solo se mostrarán archivos de esos tipos. Como en los estímulos físicos, aquí también tenemos una casilla para cada paso, pero con la particularidad que, si no aparece un estímulo que lo reemplace, la imagen, el audio o el video persisten hasta que finalicen su duración o la sesión.

- (7) **Control de conexión a la red de la sala:** Esta es simplemente una etiqueta dentro de una barra de estado, ToolStripStatusLabel, que reporta el estado de la conexión.



Figura 34 – Desarrollo de aplicación para Windows. Estado de conexión a la red

Dentro del código principal, existe un temporizador que corre en paralelo a la interfaz gráfica y cada 1 segundo evalúa el estado de la conexión con la red de la sala mediante un comando “Ping” a la dirección IP que se asignó a la Raspberry.

- (8) **Herramientas adicionales:** Además de las herramientas observables al inicio de la aplicación, el sistema posee otros elementos que son importantes de mencionar. Uno de ellos es el Timer, un temporizador encargado de llamar a la función que evalúa el estado de conexión a la red de la sala. Luego tenemos StatusStrip, que es la barra de estado ubicada en la parte inferior de la ventana, y MenuStrip, que es la barra de menús en la parte superior. Por último, tenemos los OpenFileDialog que son los controles que permiten explorar archivos en el dispositivo, hay uno para cada tipo de archivo multimedia y tienen filtros específicos de extensiones.

Para explicar el funcionamiento completo de la aplicación se describirán las acciones que realiza el programa mediante un diagrama de flujo, sin tener en cuenta los detalles del código de cada uno de los elementos gráficos.

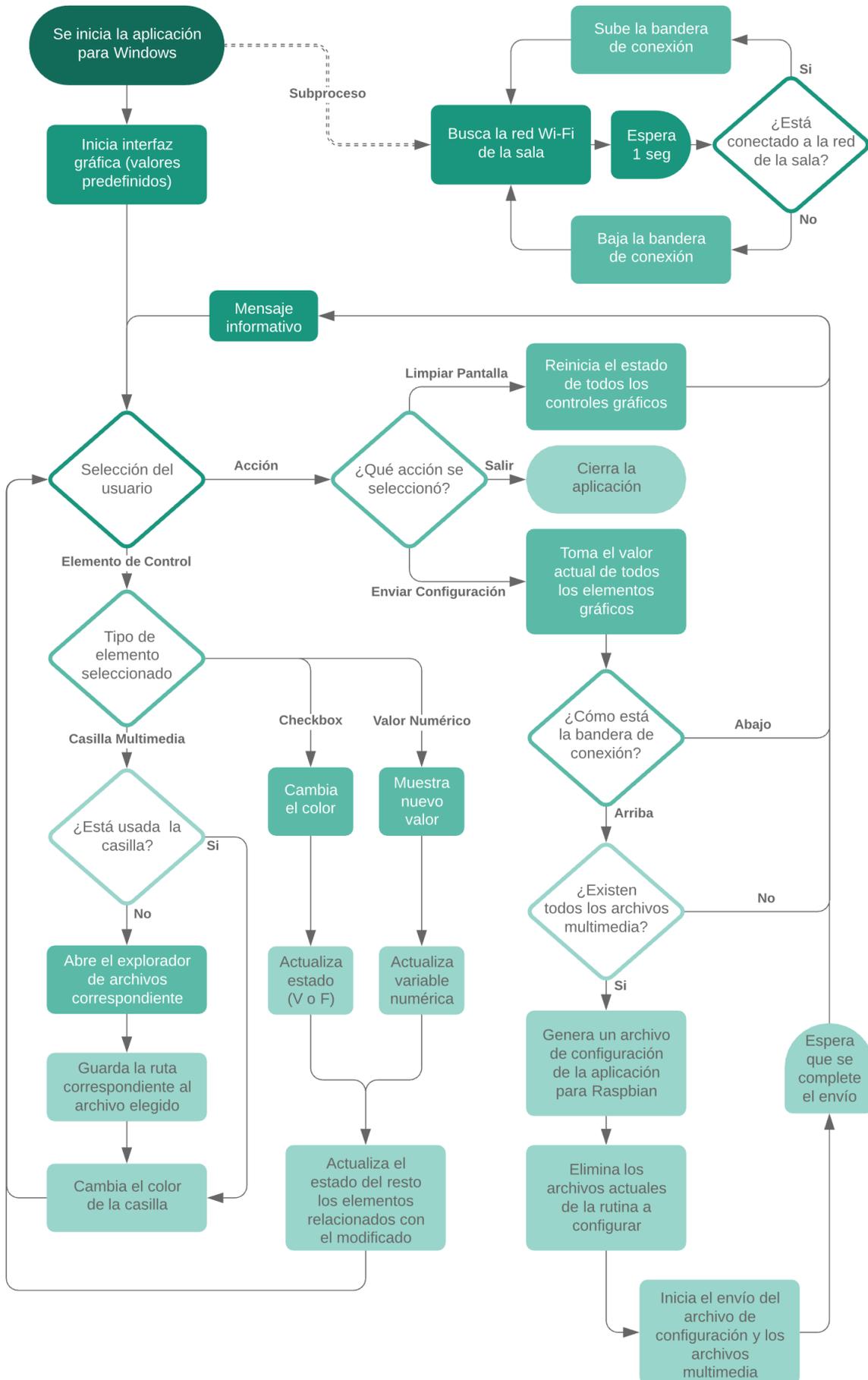


Figura 35 – Desarrollo de aplicación para Windows. Diagrama de flujo completo

2.6 Diseño Final

2.6.1 Diseño final

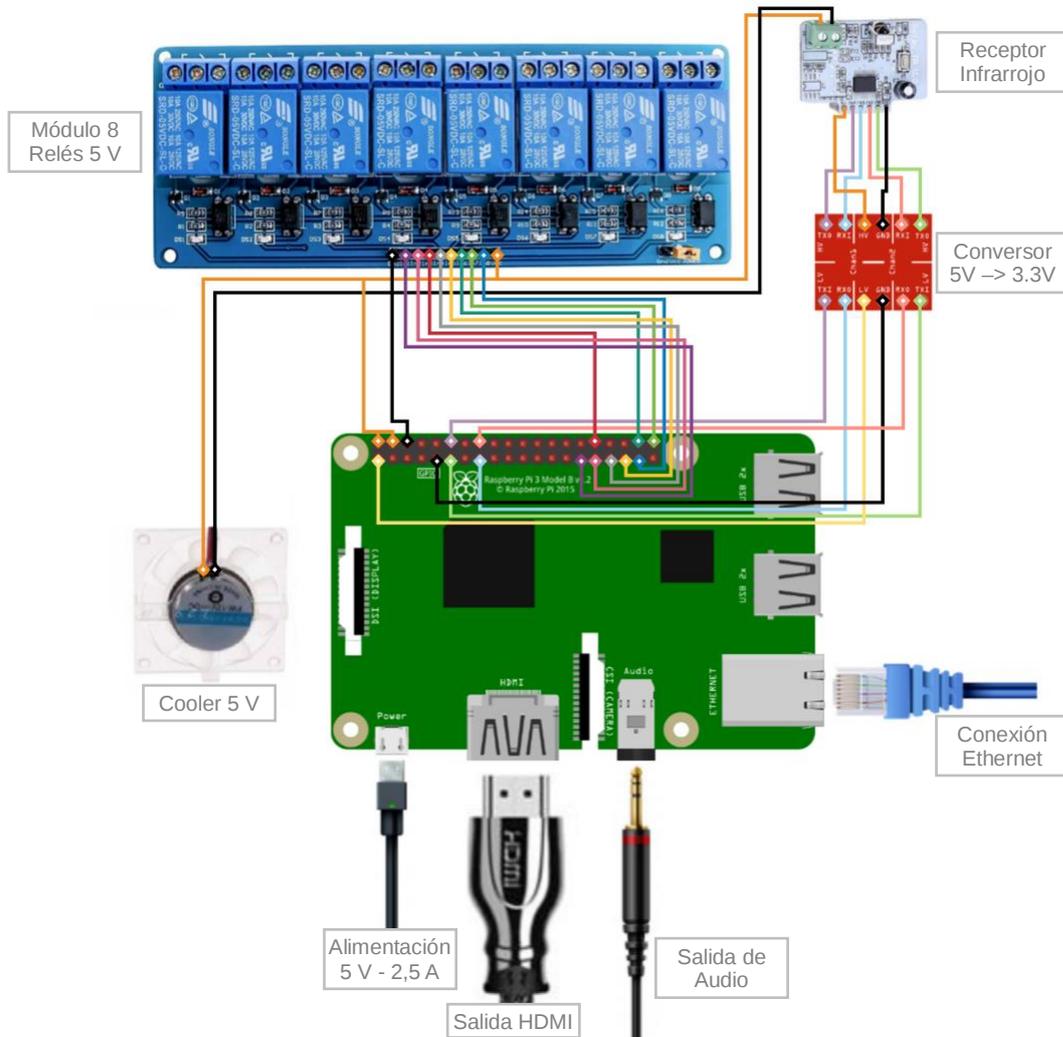


Figura 36 – Diseño Final. Esquemático del hardware

2.6.2 Prestaciones

El dispositivo cuenta con las siguientes prestaciones:

- Alimentación: 5 [V]
- Corriente máxima en Raspberry: 2,5 [A]
- Corriente máxima en Relés: 10[A]
- Rango de temperatura: 10 [°C] - 90 [°C]
- Conectividad:
 - Puerto Ethernet o Gigabit Ethernet (via USB)
 - Wi-Fi 2.4GHz y 5GHz, Norma 802.11b/g/n/ac
- 4 puertos USB 2.0
- Puerto HDMI
- Salida de audio estéreo
- Control remoto infrarrojo
- 8 salidas a relé: 220 [V] y 10 [A]

2.6.3 Fotos del hardware y capturas de pantalla de software

Es importante notar que el proyecto se concreta una vez que la sala es cableada e instalada, para que los actuadores físicos cumplan su función de estimulación. Se presentarán a continuación imágenes del producto previo a su instalación:

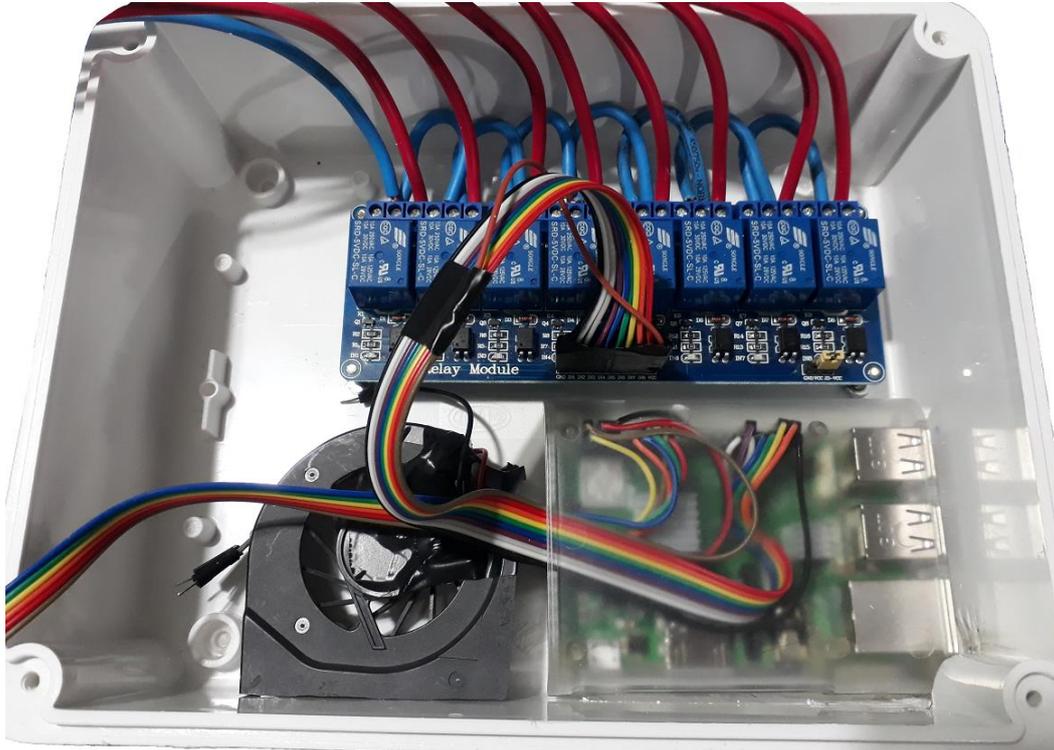


Figura 37 – Diseño Final. Disposición de componentes en carcasa

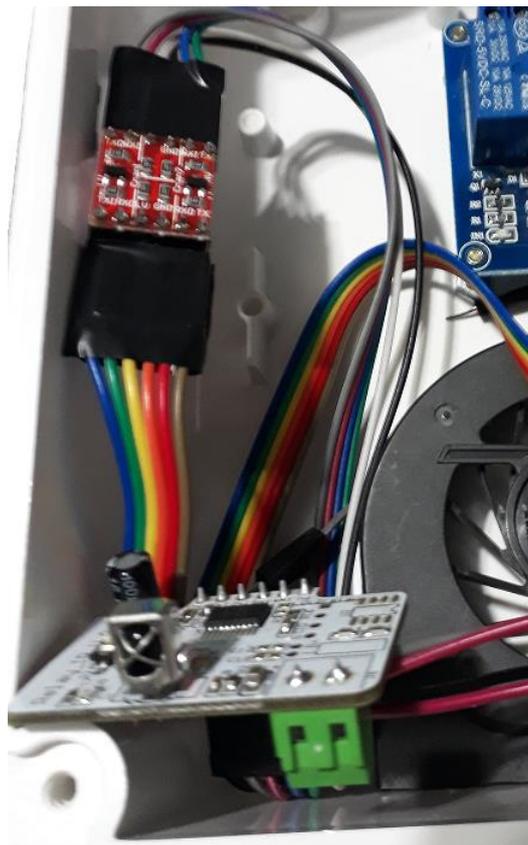


Figura 38 – Diseño Final. Receptor infrarrojo en carcasa

El control remoto infrarrojo es capaz de comandar la aplicación de la sala sin necesidad de mostrar opciones en pantalla. La interfaz para Raspbian no se muestra hasta que el usuario no realiza un click con un ratón conectado a la Raspberry. Cuando esto sucede, observamos la siguiente imagen:



Figura 39 – Diseño Final. Aplicación para Raspbian: Control de Rutinas

Si el usuario presiona en “Auto” la interfaz gráfica vuelve a esconderse y la sala se comanda nuevamente a través del control remoto. En cambio, si el usuario hace click en “Manual” se despliega la siguiente pantalla:



Figura 40 – Diseño Final. Aplicación para Raspbian: Control Manual

A continuación, se presenta la interfaz de configuración de la sala, el usuario será capaz de interactuar con los elementos gráficos para programar la sesión de terapia:

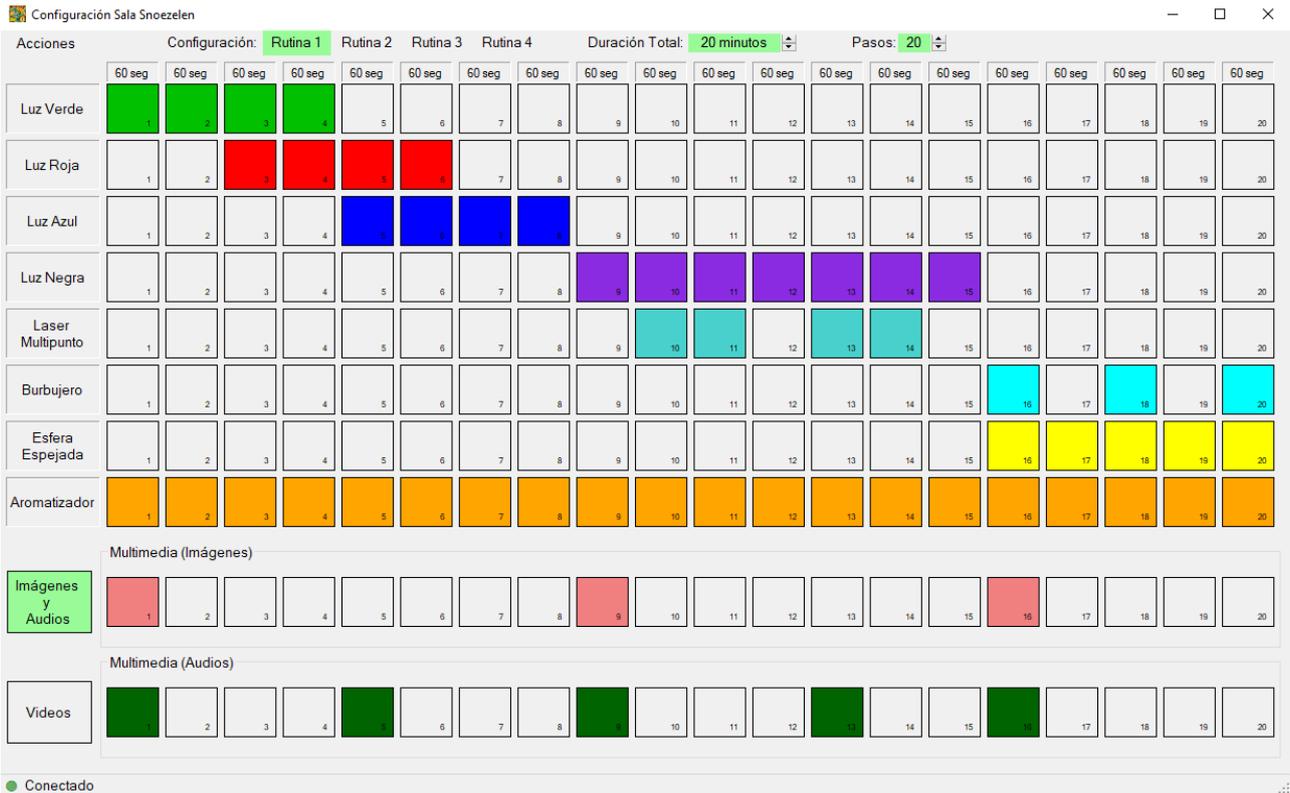


Figura 41 – Diseño Final. Aplicación para Windows: Interfaz gráfica de usuario

Una vez finalizada la configuración, se cargan los datos en la CPU de la sala:

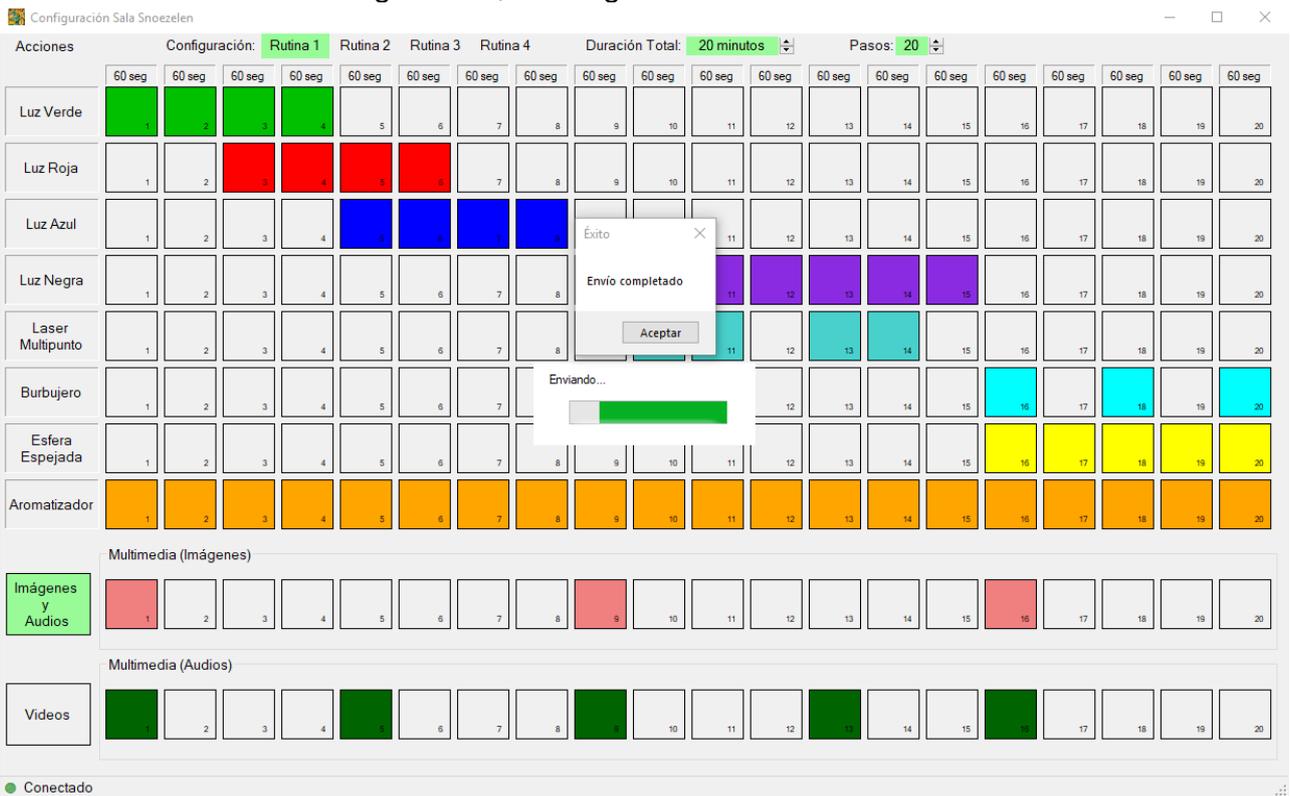


Figura 42 – Diseño Final. Aplicación para Windows: Carga de configuración

3. Capítulo 3: Resultados

En este capítulo se evalúan los resultados de las pruebas del producto. Se han realizado pruebas de:

- > *Control remoto infrarrojo:* Se ha sometido al mando infrarrojo y su circuito receptor a pruebas de alcance y fiabilidad. Se ha realizado el test a distintas distancias, 50 pruebas por cada una, siempre con una línea directa de visión al receptor.

Distancia	Aciertos	% de Éxito
< 1m	50	100%
1m-3m	48	96%
3m-5m	45	90%
> 5m	27	54%

Tabla 3 – Resultados. Pruebas de fiabilidad del mando a distancia

Se puede observar que posee una fiabilidad muy alta (>90%) cuando se lo usa a una distancia menor a 5m, pero no resulta tan confiable a distancias mayores.

- > *Tiempo de respuesta en Relevadores:* Se han sometido a los relevadores a pruebas de estrés en cuanto a la velocidad de respuesta. Se ha observado que comienzan a presentar fallas para tiempos menores a 0.5s. Por este motivo, se limitó la duración mínima de cada estímulo a 3s, lo que se estima que es suficiente para proteger la vida útil de los relevadores.
- > *Fiabilidad de conexión Wi-Fi:* Las pruebas realizadas sobre la conectividad Wi-Fi son del tipo de diagnóstico de red, se han estudiado tiempos de latencia y pérdida de paquetes durante la ejecución de la aplicación desarrollada. Se ha encontrado que el tiempo de latencia promedio es de 2ms y la pérdida de paquetes es nula.

```

Estadísticas de ping:
  Paquetes: enviados = 137, recibidos = 137, perdidos = 0
    (0% perdidos),
  Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 2ms, Máximo = 8ms, Media = 2ms
    
```

Figura 43 – Resultados. Prueba de latencia y pérdida de paquetes

Hay que destacar que las pruebas se realizaron con una intensidad de señal del 73% a aproximadamente 4m del dispositivo.

```

SSID                : Sala_Snoezelen
BSSID               : b8:27:eb:44:5f:c1
Tipo de red         : Infraestructura
Tipo de radio       : 802.11n
Autenticación       : WPA2-Personal
Cifrado             : CCMP
Modo de conexión    : Perfil
Canal               : 7
Velocidad de recepción (Mbps) : 57.8
Velocidad de transmisión (Mbps) : 57.8
Señal              : 73%
Perfil              : Sala_Snoezelen
    
```

Figura 44 – Resultados. Pruebas de alcance Wi-Fi

A una distancia mayor de 5m, comienzan a aparecer errores de comunicación.

4. Capítulo 4: Análisis de Costos

4.1 Plan de Negocios

El modelo de negocios elegido para emprender el proyecto es el modelo CANVAS, a continuación, se estudian los diferentes puntos de su estructura:

Propuesta de valor:

- > Sistema completamente autónomo
- > Sistema configurable de forma inalámbrica
- > Adaptabilidad a las necesidades del usuario
- > Control remoto infrarrojo
- > Conectividad vía Wi-Fi
- > Punto de Acceso a internet
- > Facilidad de uso y configuración
- > Posibilidad de uso de periféricos USB

Segmento de mercado:

- > Terapeutas y profesionales de la salud
- > Centros Educativo-Terapéuticos
- > Centros de Rehabilitación
- > Servicios para personas con discapacidad
- > Spas y centros de relajación
- > Instituciones educativas en general

Canales de comunicación y distribución:

- > Comunicación a través de página web
- > Contactos telefónicos y vía correo electrónico
- > Ventas a través de internet
- > Ventas personales, de forma directa
- > Ventas personalizadas a grandes instituciones
- > Ventas en exposiciones y congresos sobre rehabilitación

Relación con los clientes:

- > Capacitación y fidelización del cliente
- > Tutoriales de uso adecuado del producto
- > Soporte técnico y asistencia de forma personalizada
- > Constante mejora del producto
- > Actualización de aplicaciones
- > Soporte técnico de forma remota

Modelo de ingresos:

- > Venta e instalación del producto
- > Servicio técnico
- > Desarrollo de nuevas funcionalidades
- > Ofertas especiales y opciones de financiación
- > Servicio postventa periódico de mantenimiento

Recursos clave:

- > Componentes electrónicos de calidad
- > Experiencia en electrónica y electricidad
- > Conocimientos de programación y desarrollo de aplicaciones
- > Infraestructura para el desarrollo del producto
- > Mano de obra calificada para realizar la puesta en marcha
- > Conocimiento en ventas y trato con clientes
- > Financiación gubernamental para emprendedores
- > Experiencia en servicio técnico y resolución de problemas

Actividades clave:

- > Desarrollo de software
- > Actualización de hardware y aplicaciones
- > Capacitación y formación
- > Asesoramiento a clientes potenciales
- > Servicio técnico postventa
- > Innovación en software y hardware
- > Cortos tiempos de instalación y puesta en marcha
- > Estrategias de Marketing efectivas

Socios clave:

- > Alianzas estratégicas con importadores de equipos electrónicos
- > Desarrolladores de software
- > Revendedores y técnicos electrónicos
- > Instituciones de rehabilitación con alcance internacional
- > Convenios con instituciones y centros educativos

Estructura de costos:

- > Infraestructura y recursos humanos
- > Costos variables de materias primas
- > Desarrollo y actualizaciones de software
- > Marketing y ventas
- > Gastos fijos y carga impositiva
- > Herramientas y equipamiento
- > Instalación y puesta en marcha del producto

4.2 Costos y Retorno de Inversión

> Costo de Materiales:

El costo del hardware necesario para completar el producto con los dispositivos que se usaron como referencia en el análisis de costos es de aproximadamente U\$S 890.

Cantidad	Producto	Precio	Subtotal
1	Difusor Aromático Eléctrico	U\$S 48,13	U\$S 48,13
1	Consola Potenciada 4 canales Cabezal	U\$S 138,89	U\$S 138,89
1	Micrófono profesional	U\$S 20,76	U\$S 20,76
1	Máquina De Burbujas Bubble Master	U\$S 73,04	U\$S 73,04
1	Líquido Para Máquina De Burbujas 5L	U\$S 18,67	U\$S 18,67
1	Raspberry Pi 3B+ 64gb Memoria y Fuente	U\$S 73,02	U\$S 73,02
1	Módulo de 8x Relés 5v 10a	U\$S 15,23	U\$S 15,23
-	Materiales (Cables, Tomas, Cable Canal)	U\$S 100	U\$S 100
1	Proton Tacho Luz Negra 18 LED	U\$S 41,3	U\$S 41,3
3	Luz RGB Proton Par 36 LED	U\$S 21,85	U\$S 65,55
1	Esfera Espejada Bola Boliche 40 + Motor	U\$S 41,98	U\$S 41,98
1	Spot Tacho Led Blanco	U\$S 20,2	U\$S 20,2
1	Laser Lluvia Audio-rítmico Multipunto	U\$S 19,21	U\$S 19,21
1	Proyector Viewsonic - 3000 Lum (Usado)	U\$S 217,4	U\$S 217,4
	TOTAL		U\$S 893,4

Tabla 4 – Costos y Retorno de Inversión. Costos del Hardware de la sala

> Costo de desarrollo:

El costo de desarrollo de las aplicaciones, que se había estimado en aproximadamente U\$S 800, resultó ser ampliamente mayor (solo se habían estimado 240hs de trabajo).

El trabajo de un Técnico en Electrónica en Argentina se paga \$142 la hora. En cambio, el trabajo de un Desarrollador .Net en Argentina se paga \$170 la hora. Estos datos fueron obtenidos del sitio web de Neuvo.

El tiempo invertido como Técnico en Electrónica es de aproximadamente 120 horas. En cambio, el desarrollo de las aplicaciones ha tomado aproximadamente 800 horas.

Teniendo esto en cuenta, el costo en pesos del desarrollo es:

$$(\$142 \cdot 120) + (\$170 \cdot 800) = \$17040 + \$136000 = \$153040$$

Tomando la cotización del dólar en \$60, tenemos un costo de desarrollo de aproximadamente **U\$S 2550**.

> Retorno de inversión:

Se ha estudiado a la competencia para establecer un precio adecuado de venta. El centro CAMAC ofrece un producto con prestaciones casi idénticas en U\$S 2930.

Como el costo de los materiales es de U\$S 893,4 se busca un precio de venta que pueda ser competitivo, y finalmente se lo establece en U\$S 2020.

Este precio de venta ofrece una ganancia de U\$S 1126,6 por producto, por lo que se necesita vender solamente 2 equipos para cubrir los costos de desarrollo.

Ahora se necesita evaluar la inversión inicial y su efectividad.

Con la perspectiva de vender 12 equipos por año, el costo total de insumos y de desarrollo de software resulta en aproximadamente U\$S 13270,8. Si se considera que la infraestructura y las instalaciones necesarias son solventados a partir del capital inicial, se establece un valor de U\$S 18000 como inversión inicial.

El retorno de inversión para 12 equipos vendidos por año es de:

$$ROI = \frac{\text{Ingreso Anual} - \text{Inversión}}{\text{Inversión}} \times 100 = \frac{(12 * 2020) - 18000}{18000} \times 100 = 34,6\%$$

Figura 45 – Costos y Retorno de Inversión. Cálculo de retorno de inversión

4.3 Plan de Marketing

Para lograr una buena inserción del producto en el mercado se realizarán las siguientes acciones:

- > Se realizará un sondeo de clientes potenciales en la región. Intentando establecer sus necesidades y posibles puntos de inserción.
- > Mediante capacitaciones gratuitas se informará a las comunidades de la zona sobre esta problemática y los aportes que realiza el producto. El objetivo es que el cliente tome conciencia de la necesidad y, al mismo tiempo, conozca el producto.
- > Se extenderán invitaciones de capacitación a las instituciones más importantes de la región que puedan estar interesadas en adquirir el producto.
- > Tomando como referencia el primer dispositivo instalado, se recibirán críticas y consejos de cómo mejorar el producto y el servicio postventa.
- > Uso de redes sociales para la difusión y como punto de contacto. También se utilizarán publicidades online, tipo Google Ads, para dar conocimiento del producto a potenciales clientes.
- > Se desarrollará una página web con un diseño agradable y acorde a la temática. Brindando la información que necesite el usuario sobre la problemática y, por supuesto, el producto ofrecido. Aquí también se pondrán a disposición los links a redes sociales, direcciones de email y teléfonos de contacto.

5. Capítulo 5: **Discusión y Conclusión.**

5.1 **Conclusión**

Al comienzo de este proyecto se propuso crear un sistema integral que permita diseñar sesiones de terapia de estimulación, seleccionando el tiempo y duración de cada estímulo, para brindar al usuario la posibilidad de diseñar previamente la sesión, almacenarla en el CPU y reproducirla luego en repetidas ocasiones.

Se ha logrado obtener:

- > Un equipo confiable y de bajo costo, que se puede adecuar a los requisitos del cliente en términos de dispositivos de estimulación física, mediante el simple intercambio de los elementos conectados a los relevadores. Además, es un dispositivo capaz de generar su propia red y actuar como un Punto de Acceso Wi-Fi en caso de estar conectado a internet.
- > Una aplicación capaz de comandar la sala de forma completamente autónoma, con la capacidad de recibir comandos a distancia mediante un control remoto, como así también, mediante el uso de periféricos USB.
- > Un software para Windows, muy intuitivo y de fácil uso, capaz de crear los archivos de configuración necesarios para diseñar nuevas sesiones de estimulación.

En el mercado local se pueden conseguir otros dispositivos con prestaciones similares, pero ofrecen una tecnología más antigua y a un costo mayor. Por ello, este desarrollo puede resultar especialmente atractivo para los clientes por su capacidad de ser actualizado con una mínima inversión.

Otra gran ventaja es que, una vez que la sala está instalada, toda interacción con el dispositivo de control se realiza de forma inalámbrica. Esto puede parecer trivial, pero en un ambiente donde concurren alumnos y pacientes de diferentes edades, es necesario tomar todas las precauciones y medidas de seguridad posibles.

Durante el desarrollo de este proyecto se han aplicado muchos conocimientos de electrónica, pero también, se han adquirido nuevos en el área de programación y desarrollo de software.

Para concluir, se puede decir que los objetivos del proyecto han sido cumplidos satisfactoriamente, dejando muchas opciones de mejora y ampliación.

5.2 **Mejoras en futuros desarrollos**

Para un futuro desarrollo se intentará conseguir una carcasa personalizada y una disposición de componentes óptima, para evitar tener que adaptar carcasas estándar.

Teniendo en cuenta que el dispositivo posee 4 puertos USB, se pueden agregar hasta esa misma cantidad de periféricos o dispositivos extras. Esto abre una infinidad de nuevas aplicaciones y formas de estimulación, desde luces RGB controlables en tiempo real hasta calentadores que puedan otorgar sensaciones de frío-calor al paciente.

Una segunda posibilidad de mejora es el desarrollo de una aplicación multiplataforma que corra en Smartphones, para configurar y comandar la sala en tiempo real.

Otra interesante mejora sería el uso de HDMI CEC para brindar la posibilidad controlar la Raspberry Pi a través del control remoto del dispositivo de video (con el cable HDMI).

6. Capítulo 6: Literatura Citada

1. Kwok Hwm., To YF Sung HF (2003). The application of a Multisensory Snoezelen room for people with learning disabilities.
<http://forbrain.pt/uploads/documentos/13%20The%20application%20of....pdf>
2. Cid M, Camps Misericordia. Estimulación multisensorial en un espacio snoezelen: concepto y campos de aplicación.
http://www.uliazpi.net/intranet/galeria/uploads/ul_Noticias/01.%20LA%20INTERVE%20NCION%20MULTISENSORIAL.%20LOS%20ESPACIOS%20%20SNOEZELLEN%200.%20MARIA%20JOSE%20CID.pdf
3. Hulsegge, J. y Verheul, A. (1987). Snoezelen: Another World. Rompa Editorial.
4. López-Almela, A. Gómez-Conesa, A. Intervención en demencias mediante estimulación multisensorial (snoezelen).
<https://www.sciencedirect.com/science/article/pii/S0211563811000277>
5. Wikipedia. (2019). Raspbian. <https://en.wikipedia.org/wiki/Raspbian>
6. Wikipedia. (2019). Python. <https://es.wikipedia.org/wiki/Python>
7. Wikipedia. (2019). PyQt. <https://en.wikipedia.org/wiki/PyQt>
8. Wikipedia. (2019). C Sharp. https://es.wikipedia.org/wiki/C_Sharp
9. Wikipedia. (2019). Microsoft Visual Studio.
https://en.wikipedia.org/wiki/Microsoft_Visual_Studio
10. Matthes E. (2015). Python Crash Course: A Hands-On, Project-Based Introduction to Programming. No Starch Press
11. Beazley D., Jones B. (2013). Python Cookbook: Recipes for Mastering Python 3. O'Reilly Media.
12. Raspberry Pi Foundation. (2019) Raspberry Pi hardware.
<https://www.raspberrypi.org/documentation/hardware/raspberrypi/>
13. Croston B., MIT. (2019). RPi.GPIO. <https://pypi.org/project/RPi.GPIO/>
14. Odom W. (2013). Cisco CCNA Routing and Switching 200-120. Cisco Press
15. Renci. (2018). SSH.NET. <https://github.com/sshnet/SSH.NET>
16. Sharp J. (2018). Microsoft Visual C# Step by Step. Microsoft Press
17. Neuvo. (2019). Técnico electrónico: Salario.
<https://neuvo.com.ar/salario/?job=tecnico+electronico>
18. Neuvo. (2019). Desarrollador net: Salario.
<https://neuvo.com.ar/salario/?job=Desarrollador+Net>

