



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**DESCIFRADO DE CÓDIGO FUENTE DE
SOFTWARE PARA CONTROL DE
EQUIPO MÉDICO DENTRO DE UN
QUIRÓFANO INTELIGENTE**

INFORME DE ACTIVIDADES PROFESIONALES

Que para obtener el título de
Ingeniero en Computación

P R E S E N T A

Luis Esteban Serrano Bermúdez

ASESOR DE INFORME

Ing. Alberto Templos Carbajal



Ciudad Universitaria, Cd. Mx., 2024

*A mi madre que a pesar de todo siempre me apoyó y pudo sacarme adelante a pesar
de todas las dificultades.*

A mi hermana, por su compañía en los momentos más difíciles de mi vida.

A mi tía Cati que supo ser mi segunda mamá y cuidarme todos estos años.

A mi papá que nos acogió y educó a mi hermana y a mí como sus hijos.

A mi tío Raúl que me guió en los momentos que más lo necesitaba.

*A mi madrina y prima Lilia por su ayuda y motivación para lograr alcanzar todo
mis objetivos.*

Resumen

En el presente documento, se busca demostrar que el autor cuenta con las aptitudes y capacidades suficientes para desempeñarse como ingeniero en computación, aplicando los conocimientos adquiridos durante la carrera para la comprensión, desarrollo y ejecución de código fuente de software.

La empresa, donde el autor llevó a cabo este trabajo, posee los derechos del código de un software de control de dispositivos médicos ubicados dentro de un quirófano. No obstante, el desarrollo y mantenimiento estaban a cargo de una empresa externa. Al comenzar en el puesto, se le encomendó la tarea de replicar su operación en un entorno controlado con todos los dispositivos disponibles para comprobar su funcionamiento y rendimiento.

Para lograr el objetivo principal, fue preciso que el autor observara un quirófano inteligente en uso dentro de un hospital, así como su instalación y comunicación con los dispositivos instalados. También fue necesario que analizara una computadora con todo el hardware y software adicional imprescindible para su correcto desempeño.

Finalmente, se busca que el autor obtenga la comprensión necesaria para poder manipular los diferentes módulos disponibles en la aplicación, mejorar su rendimiento, minimizar los errores y, por último, cambiar su diseño. Además, se buscará reducir los costos de instalación, mantenimiento y soporte del software en todas sus etapas, así como proporcionar una respuesta más rápida a problemas y errores en su uso diario.

Índice general

Resumen	II
Índice de figuras	II
1. Objetivos	1
2. Marco Teórico	2
2.1. Quirófano Inteligente	3
2.1.1. Infraestructura	3
2.1.2. Software de Control de Equipo Médico	4
3. Antecedentes del Tema	7
4. Definición del problema	11
4.1. Presentación del proyecto	11
4.1.1. Obtención de código fuente	11
4.1.2. Equipo de cómputo prototipo	12
5. Metodología SCRUM	14
5.1. ¿Qué es SCRUM?	14
5.2. ¿En que consiste la metodología SCRUM?	15
5.3. ¿Por qué Scrum es importante para el desarrollo de software? .	16
6. Participación profesional	18
6.1. Refactorización y reestructuración de código fuente	18
6.1.1. Análisis de la aplicación	21
6.1.2. Pruebas	45
6.1.3. Sugerencias de mejoras en el código	58

6.1.4. Presentación del Proyecto	61
7. Resultados	67
8. Conclusiones	69
Bibliografía	70

Índice de figuras

2.1. Elementos Quirófano Inteligente	4
2.2. Pantalla Principal	5
3.1. Quirófano Inteligente	8
3.2. Quirófano Inteligente	9
5.1. Metodología SCRUM	15
6.1. Error de BD	22
6.2. Base de Datos	23
6.3. Capture Studio	24
6.4. Eye4	25
6.5. AppInterfaz	25
6.6. Menú principal	26
6.7. Pantallas menús de configuración	30
6.8. Pantalla Informes y cronómetro	30
6.9. Diagrama de Flujo Monitoreo Gases	31
6.10. Pantallas Mesa QX	32
6.11. Diagrama de Flujo Mesa Quirúrgica	33
6.12. Pantalla música	34
6.13. Diagrama de Flujo Música	34
6.14. Pantalla videollamda	35
6.15. Pantalla Luz Ambiental	35
6.16. Diagrama de Flujo Luz Ambiental	36
6.17. Pantalla Historial del paciente	37
6.18. Pantalla cámara ambiental	37
6.19. Pantalla Lámparas Satelitales	38

6.20. Diagrama de Flujo Lámpara Satelital	39
6.21. Pantalla Gestion y Video	39
6.22. Pantallas Mesa QX	40
6.23. Diagrama de Flujo Gestión y Video	41
6.24. Tarjeta QX-0030	42
6.25. Tarjeta QX-0024	42
6.26. Tarjeta QX-0209	43
6.27. Cámara QX-0016	44
6.28. Software de Video	45
6.29. Conexiones Sensado de Gases y RGBW	47
6.30. Cámara QX-0016	48
6.31. Conexiones Lámparas Satelitales	50
6.32. Pantalla cámara ambiental	62
6.33. Pantallas de monitoreo	63
6.34. Pantallas de monitoreo de 38"	63
6.35. Pantalla cámara ambiental	64
6.36. Pantallas de control	65
6.37. Pantallas audiovisuales	65
6.38. Pantalla cámara ambiental	66
6.39. Pantallas menú de configuración	66

1.

Objetivos

Como ingeniero en computación y atendiendo a las necesidades de la oferta laboral como desarrollador de firmware, se me asignó la responsabilidad de estudiar y comprender el funcionamiento de los Quirófanos Inteligentes a través de la aplicación propia de la empresa. Para ello se presentó el proyecto de descifrado del código fuente.

Dentro de las metas que se proyectaron al iniciar fue el de comprender el código fuente para así tener el control total del desarrollo del software, esto con la finalidad de minimizar costos de actualizaciones, adecuaciones, mantenimiento e instalación que estaban a cargo de una empresa externa, a la cual se le tenía que solicitar su apoyo cada vez que había un nuevo proyecto en un hospital o alguna falla que se presentara. De esta manera las instalaciones pasarían a estar a cargo del personal de servicio; las correcciones de bugs y evaluación de desempeño lo revisaría el personal de Tecnologías de la información (TI) y las nuevas características se investigaría por Diseño y Desarrollo para que de esta forma desde la instalación hasta los reportes de fallas quedarían a cargo de todo el personal de la empresa.

Por último, la actualización en materia de UI/UX (User Interface/User Experience), con el objetivo de expandirse a nuevos clientes con diferentes configuraciones acorde a las necesidades de los diferentes quirofanos.

2.

Marco Teórico

Las compañías enfocadas en los servicios de la salud, proveen desde materiales médicos farmacológicos, hasta tecnología a hospitales públicos y privados. Debido a la transformación digital de años recientes las compañías han adoptado estos cambios tecnológicos para mejorar la eficiencia y calidad en la atención al paciente. Entre ellos destaca el software para controlar el equipo médico, que al integrarse con dispositivos presentes en los quirófanos constituyen los ahora llamados Quirófanos Inteligentes, que tienen como objetivo optimizar la gestión dentro de ellos y mejorar la seguridad del paciente.

Dentro de la empresa donde se realizó el proyecto, está dividida en varios sectores lo que le ha permitido que se logre enfocar de manera consistente en diversos aspectos, desde la instalación hasta el mantenimiento de los productos y servicios. Gracias a esta división se logran desarrollar soluciones tecnológicas personalizadas para mejorar la gestión del sistema y la atención al paciente.

Para este proyecto las actividades se centrará en el área de Innovación, mayoritariamente enfocado en la subárea de Diseño y Desarrollo como la principal ya que ahí es donde desarrollé la gran mayoría del trabajo; el área de Servicio y por último TI ya que por la estructura planeada, los proyectos pasan primero al área de Diseño donde se lleva a cabo toda la parte de modelado del software para después pasar por el equipo de servicio el cual se encarga de la instalación de los equipos y dispositivos médicos dentro de la sala al igual que el cableado necesario para su funcionamiento y finalmente TI realice la implementación del software y los servicios requeridos para su correcto funcionamiento en sitio.

El desarrollo de software es un campo en crecimiento en la industria de

la salud y las empresas dedicadas a ello contribuyen significativamente a la mejora de la atención médica y operativa en los hospitales.

2.1. Quirófano Inteligente

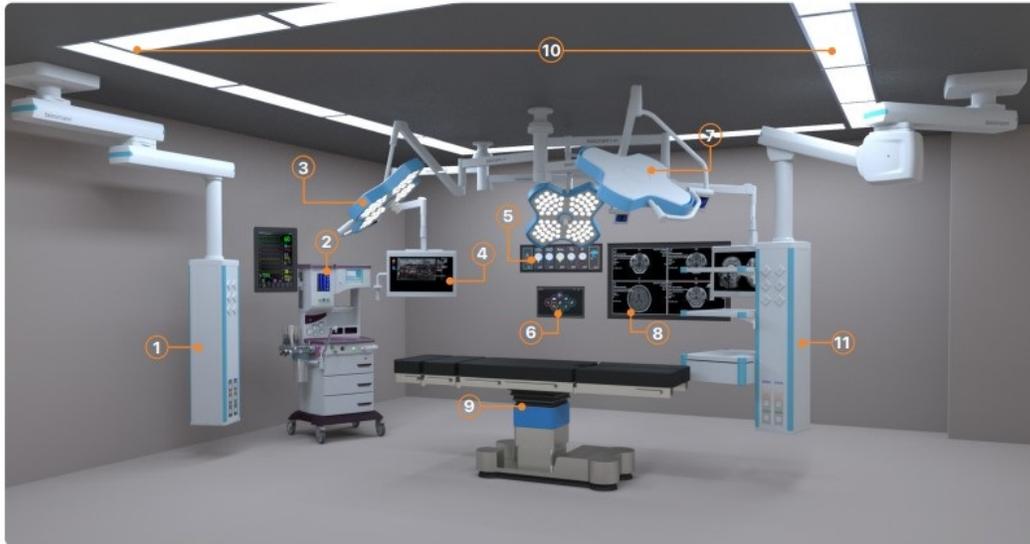
El quirófano inteligente objetivo de este trabajo, fue creado en 2016 por expertos y profesionales de la salud, consta de un software que realiza la integración de equipamiento dentro de una sala de cirugía, mediante el uso de un panel de control ubicado en una pantalla touchscreen que apoya al personal médico con diversas funcionalidades como el manejo de mesas quirúrgicas a distancia, control de lámparas, direccionamiento de imágenes a los distintos medios de despliegue en la sala, grabación y edición de videos, comunicación en tiempo real con salas de capacitación, auditorios, congresos y otros lugares de enseñanza, instalado en el hospital Ángeles Pedregal convirtiéndose en el primer quirófano inteligente en México [Biossmann, 2020].

2.1.1. Infraestructura

La infraestructura de este quirófano inteligente consta de varios monitores empotrados en la pared o montados sobre brazos articulados, junto al tendido del cableado para audio, video, alimentación y comunicación con las tarjetas de control. También se incluye la instalación de dispositivos médicos como mesas, luces RGB, lámparas quirúrgicas, sensores, torres de suministros y la computadora desde la cual se controlará la interfaz. Todos estos elementos están integrados de manera que son controlables desde la pantalla de control, facilitando la monitorización continua de las condiciones del quirófano para permitir una mejor toma de decisiones por parte del equipo médico.

El proceso de instalación de un quirófano inteligente debe iniciarse desde el área comercial. Así, se pueden obtener los requerimientos específicos del hospital y proporcionar la infraestructura adecuada a sus necesidades. La elección del modelo de mesas y lámparas quirúrgicas, la instalación de luces ambienta-

les y la determinación de la cantidad y tamaño de las pantallas dentro de la sala quirúrgica dependerán en gran medida de estos requisitos comerciales.



- | | | |
|--|---|--|
| 1) Torre de suministros para anestesia | 5) Pantallas de 38" formato amplio | 9) Mesa quirúrgica |
| 2) Unidad de anestesia | 6) Pantalla de control 24" | 10) Lámparas para iluminación ambiental RGBW |
| 3) Lámparas quirúrgicas | 7) Lámpara quirúrgica con cámara HD integrada | 11) Torre de suministro motorizada con torre de endoscopia |
| 4) Pantallas de 32" | 8) Pantalla touch 55" | |

Figura 2.1: Algunos elementos de un quirófano inteligente, (Imagen tomada del manual de usuario, consultado el 27/11/2023)

2.1.2. Software de Control de Equipo Médico

Los softwares de control de equipo médico son encargados de integrar los recursos médicos de un paciente y los equipos biomédicos de una unidad quirúrgica de manera digital. Su objetivo principal es garantizar el orden en la sala de operaciones, el fácil acceso a estudios de imagenología, la reducción de espacio utilizado por algunos equipos biomédicos y ofrecer más seguridad y comodidad en las intervenciones quirúrgicas [, 2022].

Entre las ventajas que ofrecen se encuentran:

- Acceso a los estudios de imagenología por medio del PACS (Picture Archiving and Communication System) del hospital.



Figura 2.2: Pantalla principal del software. Tomada del manual [1, 2022]).

- Control integrado del quirófano en una sola pantalla (mesa quirúrgica, lámparas satelitales, iluminación ambiental y música).
- Lectura de los parámetros del quirófano (presiones de las tomas murales de los gases medicinales, temperatura y humedad relativa).
- Reloj digital y cronómetro.
- Interconexión con equipos biomédicos de exploración y visualización.
- Cámara satelital HD con capacidad de grabación.
- Despliegue de estudios de imagen, señal de video de equipos conectados y visión de la cámara satelital en cualquiera de las pantallas secundarias de la sala.
- Acceso a la ficha de identificación del paciente.
- Llamada y video llamada.

-
- Cámara ambiental del quirófano con opción para transmitir video y sonido vía intranet o Internet.
 - Acceso a música desde Internet (YouTube o Spotify).
 - Controles Administrativos

3.

Antecedentes del Tema

Con el avance de la tecnología, los requerimientos del personal médico en quirófanos han evolucionado para facilitar el trabajo de los asistentes y apoyar al médico cirujano en los procesos quirúrgicos. El control de dispositivos y el acceso a la información del paciente dentro de un quirófano resultan de suma importancia. Es en este contexto que el desarrollo de software se convierte en un método de ayuda para la integración de dispositivos, el manejo digital de la información y la conectividad con el exterior [ETKHO, 2020].

Surge así la idea de los quirófanos inteligentes, con el objetivo de abordar todas las necesidades de los médicos. Esta concepción se impulsa mediante la integración de prácticas industriales y fabricación tradicional con el mundo tecnológico, llevada a cabo por las “fábricas inteligentes”. Estas fábricas se basan en sistemas que combinan estructuras físicas con software informático y tecnología de comunicación, lo que hoy se conoce como Internet de las cosas (IoT) e Industria 4.0.

La revolución causada por la integración de la tecnología llevó a la equipación de las salas quirúrgicas con los elementos necesarios para realizar diagnósticos e intervenciones de manera eficiente. Dentro de estas salas se encuentran monitores de grado médico para acceder a toda la información digital del paciente, visualizar dispositivos médicos externos y consultar intervenciones similares. Estos quirófanos permiten el control absoluto de los dispositivos médicos integrados, proporcionan información en tiempo real de la sala y posibilitan la comunicación externa. Todo ello es gestionado por una computadora y un panel de control táctil, vocal o por sensor de movimiento. Su uso no solo

minimiza los riesgos para los pacientes al permitir diagnósticos e intervenciones simultáneos, sino que también reduce los tiempos al evitar traslados entre diferentes áreas del hospital [Villar Mir, 2019].

El Hospital General de Castellón fue pionero en la instalación del primer quirófano inteligente del mundo en 2006, con una inversión de 900,000 euros en ese momento. Este quirófano incorporó el estándar de imagen HDTV (alta definición, de 1,080 líneas) y una plataforma universal para endoscopia que permite la total compatibilidad con endoscopios de cualquier especialidad.

Los quirófanos inteligentes se caracterizan por ser espacios amplios con escasa presencia de cables en el suelo, ya que todas las conexiones se realizan por dentro de las paredes, llegando a un rack en el exterior del quirófano. Además, implementan tecnología inalámbrica y cuentan con torres de conexiones colgadas al techo, donde se encuentran tomas de corriente, conexiones de video y tomas de gases. También incluyen monitores anclados en la pared o colgados en brazos para facilitar su limpieza, desplazamiento y acceso dentro del quirófano.



Figura 3.1: Quirófano Inteligente (Imagen cortesía de https://www.biossmann.com/quiروفano_inteligente/, consultada el 26/12/2023)

De entre los elementos más comunes en un Quirofono Inteligente se encuentran:

- Mesas de operaciones, ya sea con controles alámbricos o inalámbricos.

-
- Cámaras ambientales que permiten la grabación a conveniencia de la sala de operaciones.
 - Lámparas quirúrgicas que proporcionan iluminación intensa y enfocada al área de trabajo para mejorar la visibilidad y precisión durante las cirugías.



Figura 3.2: Muestra de un quirófano inteligente. (Imagen cortesía de [System, 2023], <https://blog.hospitalangeles.com/posts/quirofanos-inteligentes-la-evolucion-de-la-cirugia>)

Así es como los cirujanos en México reconocen la utilidad de tener todos los dispositivos al alcance y lograr controlarlos a lo largo de una cirugía, además de mejorar la eficiencia y coordinación en el quirófano para optimizar el flujo de trabajo, reducir tiempos y minimizar posibles obstáculos físicos. En consecuencia, grupos de ingenieros en México se han propuesto innovar y desarrollar una manera de interconectar todos los dispositivos en una computadora. Esta iniciativa toma como inspiración los avances realizados en Europa y Asia con los quirófanos híbridos, con el objetivo de crear un sistema que permita el control eficiente de los elementos de un quirófano, apoyándose en las nuevas tecnologías inalámbricas y protocolos de comunicación [Cué Palero, 2019].

Este enfoque hacia la interconexión y control centralizado en quirófanos no solo representa un avance tecnológico, sino también una respuesta a las demandas específicas de los profesionales de la salud. La adaptación de estas soluciones a las necesidades locales destaca la importancia de la innovación contextualizada y la colaboración entre distintas disciplinas para mejorar la calidad de la atención médica.

4.

Definición del problema

4.1. Presentación del proyecto

Al iniciar el proyecto, el gerente del área de innovación y diseño proporcionó una explicación detallada y una presentación del software, destacando su aplicación en entornos hospitalarios y los objetivos esperados por parte del equipo. Posteriormente, el coordinador asignó el orden de las tareas, basándose en los requerimientos mínimos de funcionamiento. Para dar inicio, era necesario contar con los siguientes elementos:

1. Código fuente del software.
2. Equipo de cómputo para el desarrollador.
3. Equipo prototipo para pruebas de la aplicación.
4. Visita a un hospital con el software instalado y en funcionamiento.
5. Dispositivos médicos con los que el software interactuaría.

4.1.1. Obtención de código fuente

Para acceder al código de la aplicación, el primer paso fue solicitar acceso al repositorio donde se encuentra almacenado, para ello se gestionó a través del departamento de TI que, a través de la cuenta institucional de la empresa. Con ello se otorgó el acceso al repositorio de GitLab, suite que permite gestionar, administrar, crear y conectar los repositorios con diferentes aplicaciones de todo el software propiedad de la empresa.

Previo a la descarga del proyecto, se procedió a identificar el lenguaje de programación en el que estaba desarrollado. Este paso fue necesario para preparar el equipo de trabajo con las condiciones mínimas requeridas para la depuración y ejecución del código. Al examinar los archivos, se notó que estos tenían la extensión .cs, lo que llevó a la conclusión de que la aplicación estaba programada en C#. Se optó por utilizar el Entorno de Desarrollo Integrado (IDE) Visual Studio 2022 (VS) para analizar el código y realizar la depuración. Este entorno facilita el desarrollo de aplicaciones para la plataforma .NET, proporcionando las bibliotecas necesarias para trabajar con el lenguaje C# y aprovechar los recursos visuales de la interfaz de la aplicación.

A continuación, se procedió a instalar el software de versionado Git. Esto permitió descargar el proyecto desde el repositorio de la empresa, y de esta forma iniciar con la depuración del código. Además, Git proporciona una solución eficiente para versionar los cambios realizados, facilitando la ejecución de la aplicación en equipos con características más modestas y permitiendo la implementación de modificaciones conforme se avanza en el desarrollo del proyecto [Chacon and Straub, 2014].

Siguiendo con el desarrollo, y con el proyecto ya descargado, no se halló ningún tipo de documentación, manual o instructivo que pudiera brindar orientación sobre la ejecución del mismo. Esta carencia dificultó el inicio rápido de la aplicación. Por consiguiente, se optó por utilizar la depuración paso a paso para obtener una comprensión más detallada del flujo de la aplicación y lograr una primera ejecución de la aplicación.

4.1.2. Equipo de cómputo prototipo

Después de lograr un primer inicio y de observar los módulos que comprendía la aplicación, se consultó con el coordinador y el gerente acerca de las características de las computadoras actuales en las que se ejecutaba la aplicación. Esto condujo a la entrega de unos primeros equipos en desuso que eran capaces de mantener la aplicación en ejecución.

Posteriormente, y debido a las condiciones del desarrollo, también se optó por actualizar los equipos, buscando una reducción en el costo de adquisición de los mismos y mejorar el hardware que ya se tenía.

Estos equipos constan de una computadora de alto rendimiento, donde, dependiendo de la configuración del quirófano inteligente, cambiaban ligeramente sus componentes como se menciona a continuación.

- El equipo para una configuración 4x4 (4 entradas - 4 salidas) consta de:
 - Procesador Intel Core i9-10900K @ 3.70 GHz.
 - Motherboard Gigabyte B460M DS3H AC V2.
 - Memoria RAM 32 GB DDR4 3200MHz, 2 Módulos de 16 GB.
 - Tarjeta de Video Nvidia GeForce RTX 3060.
 - Tarjeta Capturadora de Video AVerMedia CL314H1.
 - Disco Principal SSD 256 GB, Disco Secundario HDD 1TB.
 - Windows 10 Pro.
 - Fuente de poder 650W 80 Plus Bronze.
 - Lector/Quemador de discos DVD.

- Mientras que una configuración 8x8 (8 entradas - 8 salidas) es:
 - Procesador Intel Core i9-10900K @ 3.70 GHz.
 - Motherboard Asus Prime Z590-A.
 - Memoria RAM 32 GB DDR4 3200MHz, 2 Módulos de 16 GB.
 - Tarjeta de Video Matrox D-Series D1450 x2 unidades.
 - Tarjeta Capturadora de Video AVerMedia CL314H1 x2 unidades.
 - Disco Principal SSD 256 GB, Disco Secundario HDD 1TB.
 - Windows 10 Pro.
 - Fuente de poder 750W 80 Plus Bronze.
 - Lector/Quemador de discos DVD.

5.

Metodología SCRUM

Debido a la naturaleza de los proyectos de la empresa, se optó por la metodología más efectiva, que en estos casos resultó ser SCRUM, dada su relevancia y aplicación exitosa en diversos proyectos de implementación de software, donde el objetivo principal es el control permanente del estado del software [Velasco et al., 2021].

Como lo menciona Barrios et al. (2011), el vínculo interdisciplinario entre licenciados e ingenieros es lo que permite aplicar la metodología como una práctica dentro del desarrollo y establecer como objetivo la iniciación de un proceso formal de la gestión, así como la creación de una estructura organizativa que predisponga a la empresa en futuras etapas de crecimiento.

5.1. ¿Qué es SCRUM?

SCRUM, en esencia, es un marco de gestión utilizado para organizar equipos de forma autónoma y colaborativa con el objetivo de alcanzar metas comunes. Este enfoque describe un conjunto de reuniones, herramientas y funciones diseñadas para facilitar la entrega eficiente de proyectos. Las prácticas de SCRUM capacitan a los equipos al permitirles gestionarse y aprender de manera autónoma basándose en la experiencia, lo que facilita su adaptación a cambios imprevistos. Además, SCRUM demuestra ser efectivo en la resolución rentable y sostenible de problemas complejos [AWS, 2023].

5.2. ¿En que consiste la metodología SCRUM?

La metodología SCRUM se caracteriza por los siguientes principios y valores:

■ **Principios:**

- **Transparencia:** Los equipos de trabajo son conscientes de los retos que pueden experimentar los demás equipos. Debe haber reuniones periódicas entre los propietarios del proyecto y los equipos de trabajo para evitar la falta de comunicación y cuellos de botella de información.
- **Reflexión:** Se incorporan puntos de reflexión frecuentes en el marco para permitir a los miembros del equipo revisar los avances logrados. La información adquirida es usada para realizar estimaciones y pla-

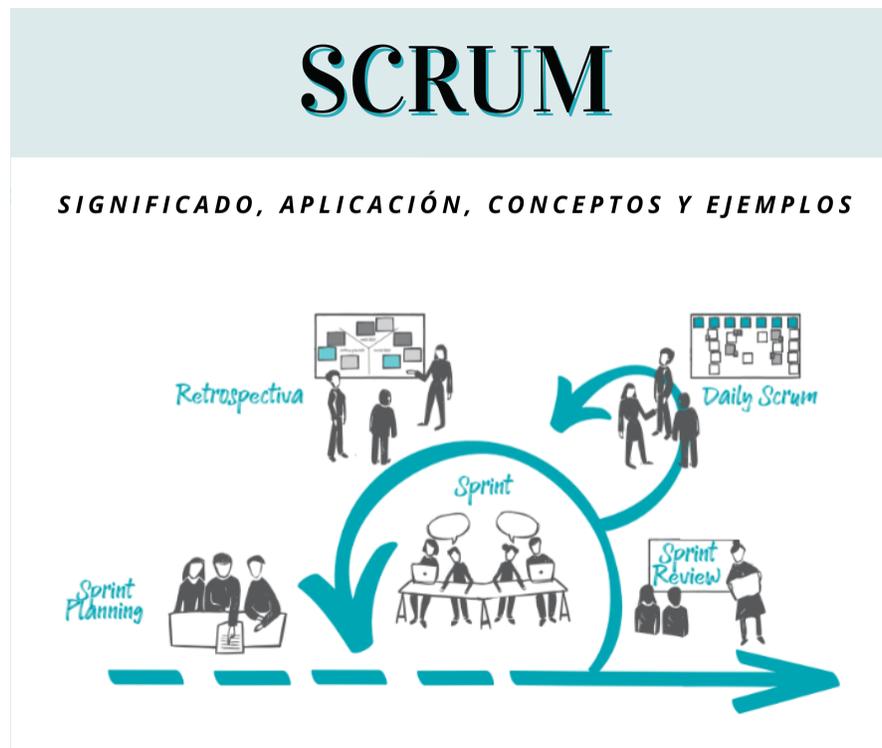


Figura 5.1: Metodología SCRUM (Imagen tomada de Internet, consultada el 15/12/2023)

nificaciones a futuro, resultando en proyectos planeados de manera más eficiente, dentro del presupuesto y plazos previstos.

- **Adaptación:** Los miembros del equipo pueden volver a priorizar las tareas en función de los requisitos cambiantes de los clientes. Deciden qué tareas se completan primero y cuáles se retoman en el futuro.

■ **Valores:**

- **Compromiso:** Los miembros del equipo están comprometidos con las tareas y objetivos basados en el tiempo y se dedican a la mejora continua para encontrar la mejor solución.
- **Valentía:** Los equipos demuestran valentía al formular preguntas abiertas y que suponen un reto. Mantienen conversaciones honestas y transparentes para encontrar la mejor solución.
- **Concentración:** Se centrarán en las tareas seleccionadas para proporcionar resultados en un plazo limitado.
- **Actitud receptiva:** Los miembros del equipo se muestran receptivos a las nuevas ideas y oportunidades que apoyan el aprendizaje individual y la calidad general del proyecto.
- **Respeto:** Los miembros del equipo de trabajo respetan a los directores del proyecto, a los demás y al proceso.

5.3. ¿Por qué Scrum es importante para el desarrollo de software?

La metodología SCRUM permite a los equipos responder de manera rápida ante los requisitos cambiantes, evitando que los presupuestos se desborden. Es importante debido a los siguientes puntos:

- **Capacidad de mantener la calidad en situaciones que plantean desafíos.** Esto implica que los requisitos se mantienen relevantes y viables

durante un corto periodo de tiempo. La retroalimentación y las revisiones permiten la mejora continua.

- **Un mayor rendimiento de la inversión.** Se priorizan los requerimientos en función de los análisis de riesgos y se enfoca en desarrollar un producto inicial que pueda llegar a los usuarios con el objetivo de tener retroalimentación en una fase temprana.
- **Equipos más felices y productivos.** Las estructuras de equipos que se organizan y administran por cuenta propia permiten a los miembros ser más creativos e innovadores.
- **Métricas relevantes que mejoran las estimaciones.** Cada equipo elige sus propias métricas para medir el desempeño del proyecto. Estiman los plazos, los presupuestos y las métricas de calidad en función de su experiencia y capacidades.

La metodología SCRUM tuvo un impacto beneficioso en el desarrollo del proyecto, especialmente porque se trataba de un proyecto interno donde las consultas a los desarrolladores iniciales (quienes formaban parte de una empresa externa) resultaban imposibles. Gracias a la autonomía facilitada por la metodología utilizada, los involucrados lograron un avance continuo. Esto se debió a la capacidad de realizar pruebas continuas en las diversas secciones del código y recibir retroalimentación por parte de los propietarios del software.

Además, esta aproximación permitió visualizar áreas de oportunidad en el código inicial, identificando posibles métodos más eficaces para su implementación. Esta dinámica de trabajo favoreció la adaptabilidad y la mejora constante del proyecto, a pesar de la limitada disponibilidad para consultas externas.

6.

Participación profesional

6.1. Refactorización y reestructuración de código fuente

En el desarrollo de software, el cambio es un factor constante que ejerce presión sobre el sistema, generando la necesidad de analizar su ciclo de vida. En este contexto, el énfasis recae principalmente en el mantenimiento, el cual se desglosa en las siguientes categorías:

1. **Mantenimiento Perfectivo:** Se realizan tareas destinadas a perfeccionar el sistema de software, pudiendo incluir la implementación de nuevos requerimientos.
2. **Mantenimiento Correctivo:** Se llevan a cabo tareas para corregir errores identificados en el sistema de software.
3. **Mantenimiento Adaptativo:** Implica realizar tareas de mantenimiento para adaptar el sistema de software a los cambios en su entorno. Esto puede requerir la implementación de nuevos requerimientos.
4. **Mantenimiento Preventivo:** Su propósito es prever posibles averías o desperfectos en el sistema, anticipándose a problemas potenciales.

A partir de esta perspectiva, se comprende que el desarrollo de software constituye un proceso continuo de evolución mediante sucesivas liberaciones de versiones. En este contexto, el código fuente experimenta cambios, y es

esencial identificar los componentes que permanecen estables y aquellos que necesitan atención. Este reconocimiento resulta fundamental para prevenir el deterioro y la eventual inestabilidad o dificultad de mantenimiento del código, especialmente ante la incorporación de nuevos requerimientos.

Esta problemática ha llevado a la ingeniería de software a investigar y desarrollar estrategias para mejorar el código ya existente. En este sentido, la reestructuración del código fuente ha surgido como una herramienta fundamental. Esta práctica implica la aplicación de transformaciones al código con el objetivo de hacerlo más fácil de cambiar y comprender. La reestructuración busca, ante todo, reducir la susceptibilidad a errores cuando se aplican futuros cambios. En consecuencia, se convierte en una valiosa herramienta para minimizar el esfuerzo requerido durante la fase de mantenimiento en el ciclo de vida del desarrollo de software.

El objetivo principal de la reestructuración de software es mantener o aumentar su valor, dado su constante evolución impulsada por las solicitudes de características de los usuarios, puede volver a los programas más rígidos y difíciles de modificar. Además, se busca incrementar la posibilidad de reutilizar componentes en otros sistemas. Finalmente, es crucial evaluar cómo este proceso contribuirá al valor actual del sistema.

Algunas técnicas de reestructuración orientadas a las modificaciones de código fuente son las siguientes:

- a) **Estilo de programación:** Busca estructurar el código de manera que sea más fácil de comprender, mejorando su aspecto mediante una estandarización del estilo y sustituyendo secciones con otras más comprensibles.
- b) **Paquetes / Código Reusable:** Esta técnica implica la utilización de componentes, paquetes y bibliotecas que han sido utilizados previamente.
- c) **Control de flujo:** La reestructuración se realiza, en parte, para mejorar y simplificar el flujo de un programa.

d) **Documentación o Actualización de la documentación:** Consiste en mantener actualizada la documentación de un sistema, facilitando así la reingeniería y la aplicación de ingeniería inversa.

Como el software está en constante evolución, en muchas ocasiones el proceso que se lleva a cabo tiende a generar una gran “bola de barro”. Esto ocurre con frecuencia debido a que se desarrolla bajo la presión de cumplir con fechas límite, la optimización de recursos u objetivos ficticios, sumado a la falta de comprensión del sistema o la ausencia de bases de una buena arquitectura.

La refactorización, por lo tanto, se presenta como un intento de mejorar la producción de código reutilizable. El desarrollo se convierte en un proceso aún más complejo, ya que es el resultado de varias iteraciones de diseño. En este contexto, la refactorización surge como “El proceso en el cual se aplican cambios en un sistema de software de forma tal que no altere el comportamiento externo del código, mejorando su estructura interna”.

Refiriéndonos al comportamiento externo del código, es crucial preservarlo durante el proceso de refactorización. Esto implica que el conjunto de entrada y salida debe permanecer inalterado antes y después de aplicar el proceso. Sin embargo, desde el punto de vista de entrada y salida, la situación es ambigua debido a otros factores que influyen en el comportamiento del software actual, tales como:

1. **Software en tiempo real:** Donde el proceso de refactorización debe preservar las restricciones temporales.
2. **Software embebido:** Donde se debe preservar el consumo de memoria

En última instancia, vale la pena mencionar que las características del software se dividen en externas (robustez, extensibilidad, rendimiento, reusabilidad) e internas (comprensibilidad, legibilidad, corrección, redundancia). La refactorización de código afecta principalmente a las características internas al eliminar código redundante y cambiar nombres de métodos o variables. No

obstante, también puede repercutir en las características externas, como el rendimiento. El polimorfismo y las optimizaciones de los compiladores actuales juegan un papel clave en la mejora del rendimiento durante el proceso de refactorización [Méndez et al., 2010].

6.1.1. Análisis de la aplicación

Con los requerimientos cubiertos para el funcionamiento del software, se diseñó un plan de acción para realizar el estudio del código, todo apoyado por la metodología SCRUM.

En este plan se realizó una primer reunión para exponer todos los puntos, alcances que se debían cumplir al finalizar el proyecto y la periodicidad para presentar los avances. En esta primer reunión, denominada Sprint Planing, se acordó realizar las revisiones cada 2 semanas, de manera que hubieran avances significativos en el descifrado del código y en caso de ser necesario presentar la solicitud de uso de algún dispositivo que interactue con el sistema.

De acuerdo a la metodología también era necesario el realizar reuniones diarias entre el equipo, sin embargo, al ser un proyecto donde solo estuvo involucrado un desarrollador, las reuniones diarias con el coordinador del proyecto se realizaban al inicio de la jornada laboral, presentando los detalles descubiertos así como preguntar las dudas sobre alguna función de la aplicación.

Requerimientos de software y hardware

Cubiertos los puntos del plan y el seguimiento, llegó el momento de iniciar el análisis. Para ello, se utilizaron los primeros equipos en desuso para revisar los componentes de hardware y el software instalado, con el objetivo de verificar cuáles son aquellos que interactúan con la aplicación y en el equipo para compararlo con la descripción detallada en las hojas de especificaciones.

Entre el software que se encontró en el equipo destacan:

- Microsoft SQL Server

- Microsoft SQL Server Management Studio
- AVerMedia Capture Studio
- AVerMedia SDK
- Eye4
- Camera Finder

Al usar la depuración “paso a paso” para analizar con más detalle el arranque de la aplicación y sin realizar cambios en el código fuente, se encontró un mensaje de “Error de conexión con BD” (Fig. 6.1) demostrando que hacía falta la conexión con la base de datos de MySQL

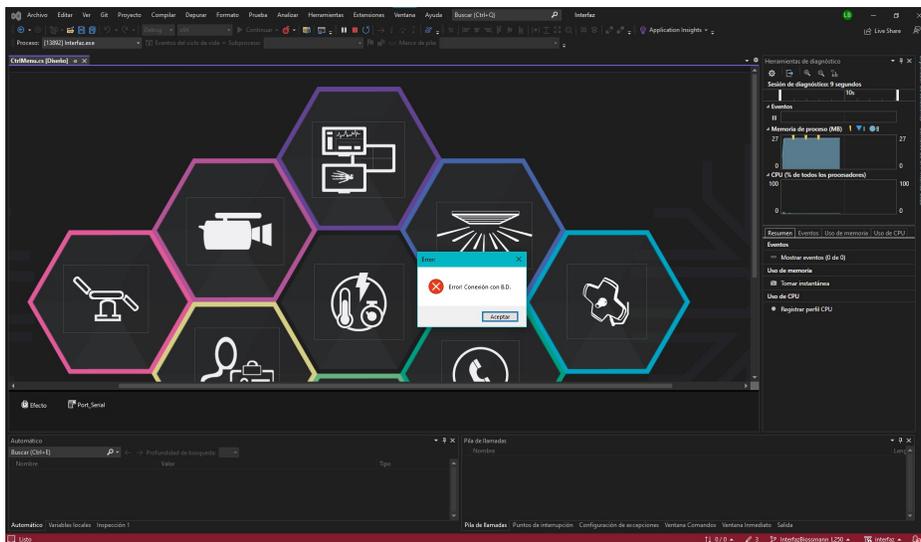


Figura 6.1: Mensaje de Error de la Base de Datos (Captura de pantalla tomada en el momento de la depuración.)

Debido a ello se recurrió a usar los equipos en desuso entregados para revisar la manera en la que esta diseñada la base de datos (Fig. 6.2), y tratar de replicarla en la computadora de trabajo para continuar con la revisión. Al encontrar las tablas, ahora la tarea fue el encontrar una manera de clonar la base de datos y copiar las carpetas a las que se hace referencia al momento de iniciar la aplicación, ya que estas almacenan archivos útiles para los médicos.

Para obtener una comprensión más profunda del proyecto y su relación con la base de datos, se solicitó al coordinador la oportunidad de presenciar una instalación en sitio, con el propósito de entender los requisitos necesarios para el correcto funcionamiento. En consecuencia, se programó una visita a un hospital, que se encontraba en construcción en esos momentos, incluyendo los quirófanos inteligentes que estaban siendo instalados por los desarrolladores anteriores. Durante esta visita, se comprobó el software que se estaba instalando y se comparó con el que se encontraba en los equipos antiguos. Como punto adicional, se recopiló información sobre el funcionamiento de las tablas y se adquirió conocimiento sobre los desafíos que pueden surgir en un entorno real de operaciones.

Una vez comprendido el funcionamiento de la base de datos, entonces se procedió a usar las bases de datos de equipos anteriores para clonarlas y crear un script que pudiera funcionar de manera general [De León, 2020]. De esta manera se consiguió tener una manera rápida para generar la base de datos en instalaciones nuevas.

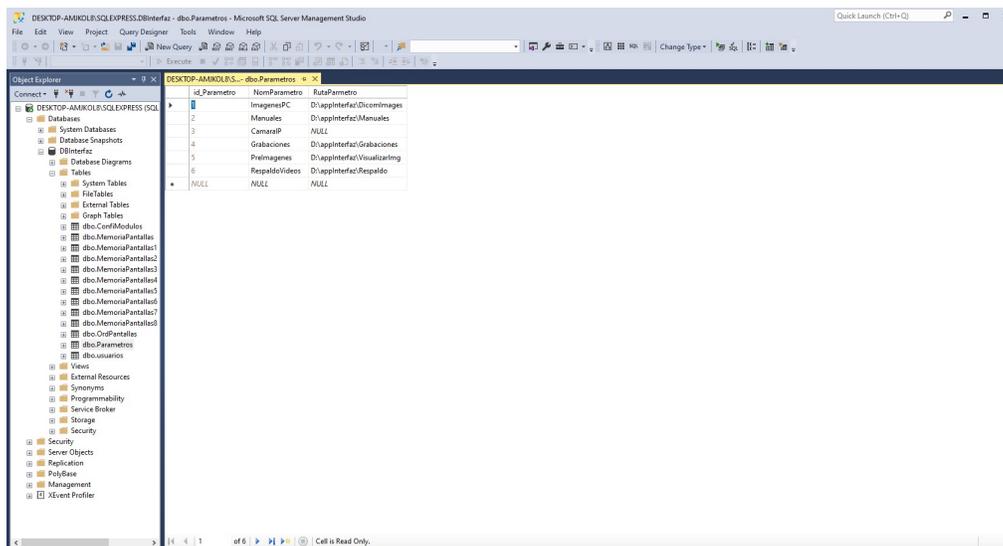


Figura 6.2: Base de Datos de la aplicación.

A continuación se revisó el funcionamiento de las capturadoras de video por lo cual se analizó cómo se relacionan y se configuran para que estas funcionen

de manera correcta, principalmente tomando en cuenta que los puertos de entrada se configuran en diferentes resoluciones, además de necesitar el SDK de AVerMedia para poder hacer uso de las funciones dentro de la aplicación.

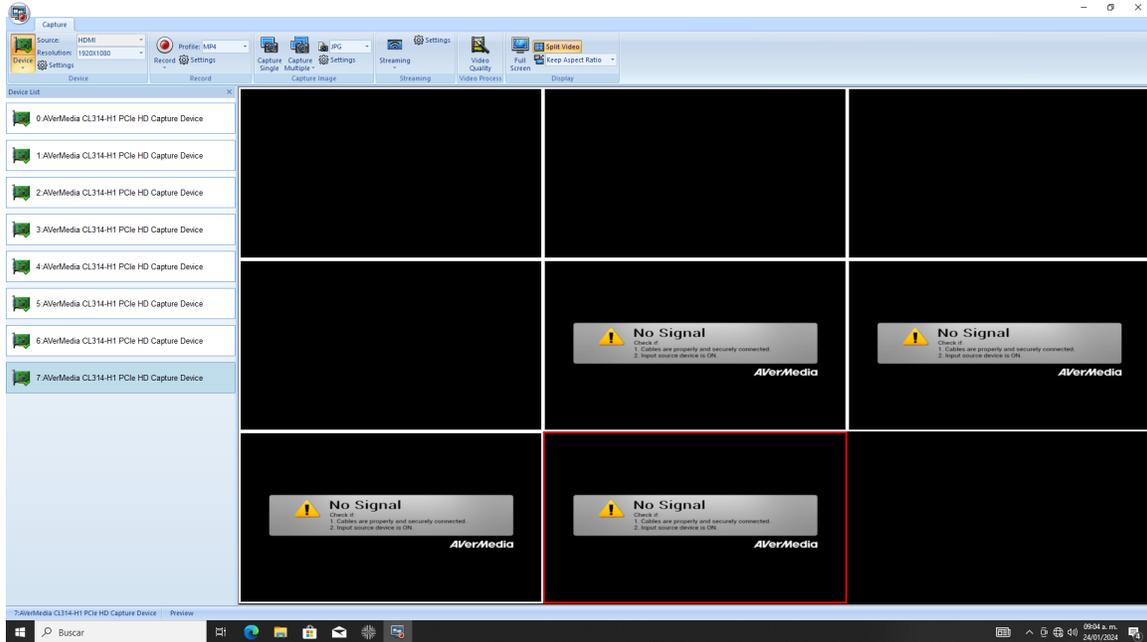


Figura 6.3: Aplicación Capture Studio con 4 pantallas habilitadas y 4 deshabilitadas.

Finalmente para poder conectar la cámara ambiental es requisito usar la aplicación de Camera Finder con lo cual es posible detectar todas las cámaras de video vigilancia conectadas en la misma red, y esto ya opcionalmente, para comprobar que la cámara es la que se encuentra en el mismo quirófano se usa la aplicación de Eye4 para previsualizar la imagen.

Como último requisito, sin ser un software si es necesario crear una carpeta en el disco secundario con las carpetas para grabaciones de video, manuales de ayuda y almacenamiento temporal de archivos de imagenología.

Interpretación y análisis de cada módulo

Inicio de la aplicación

Iniciando el análisis del código y con la base de datos establecida en la máquina de prueba, se procedió a reiniciar la aplicación utilizando el método de

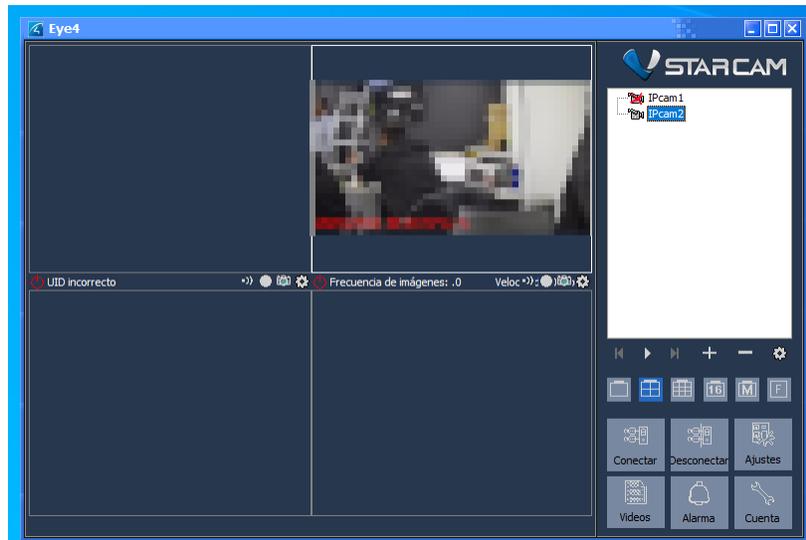


Figura 6.4: Aplicación Eye4 para visualización de cámara ambiental.

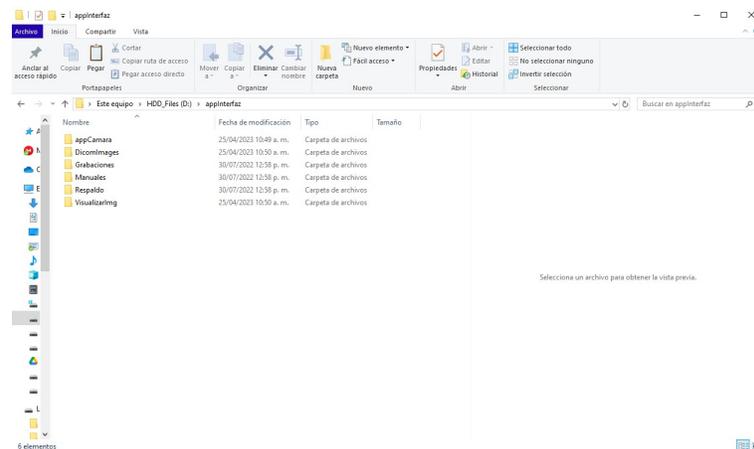


Figura 6.5: Carpeta appInterfaz necesaria para archivos de la aplicación.

depuración “paso a paso”. El objetivo era comprender la interacción entre la base de datos y los dispositivos médicos que podían ser controlados.

Al ejecutar la aplicación, se envía una solicitud a la base de datos para asignar la posición de las pantallas conectadas, elegir el modelo de los dispositivos médicos conectados y ubicar las carpetas con los archivos necesarios. Luego, se realiza una limpieza de las carpetas de imágenes DICOM, eliminando archivos remanentes de sesiones anteriores, y comienza a asignar las ubicaciones de las pantallas según el índice registrado en la base de datos. Finalmente, asigna todas las rutas de las carpetas a variables para acceder a los archivos durante la ejecución de la aplicación.

Concluidos estos procesos, la aplicación avanza al menú principal, donde se pueden acceder a los distintos módulos disponibles en la aplicación.

Panel Maestro y Menú Principal

En el menú principal, se encuentran los iconos de los módulos a los cuales se puede acceder. Cabe destacar que, dependiendo de los requerimientos de cada hospital, a veces no se utilizan ciertos módulos, por lo cual solo se desactivan pero siguen apareciendo en el menú.

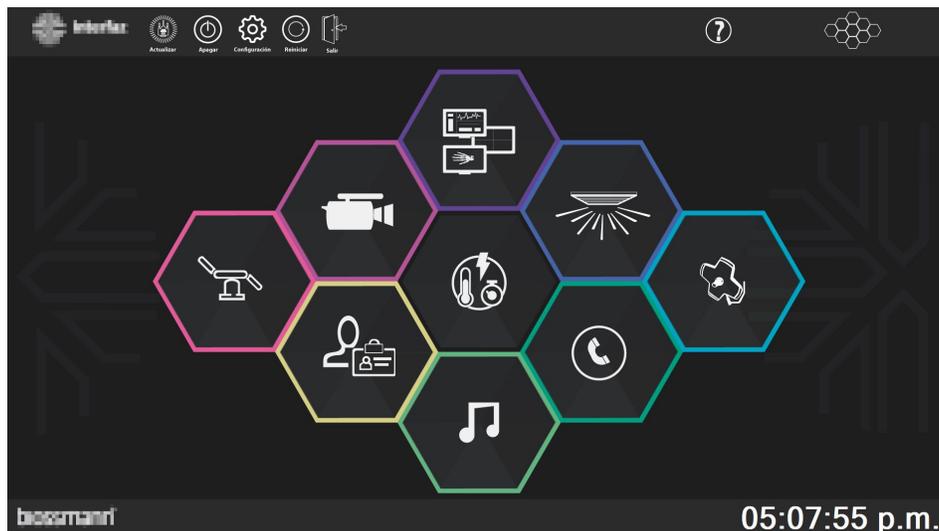


Figura 6.6: Panel maestro donde se sitúa el menú principal

Una vez que se carga el menú principal, la primera acción que se realiza

es intentar comunicarse por el Puerto COM3 con una tarjeta de control. Tras revisar las referencias, se determinó que esta tarjeta es la encargada de mostrar la información de los sensores de gases quirúrgicos. A continuación, utilizando las ubicaciones de la base de datos, se colocan los paneles en las pantallas para poder enviar las imágenes y los videos de la aplicación.

En este punto, ya se tiene construido el panel maestro, que es el que contiene a los formularios de los módulos cuando son seleccionados. Además, también contiene el menú principal, con el cual se puede acceder a los módulos y realizar acciones como apagar, reiniciar, salir, configurar los dispositivos médicos, acceder a los manuales de ayuda y visualizar el reloj con la hora actual.

Continuando con el funcionamiento, después de intentar hacer la conexión con la tarjeta de informes de gases, se cargan los botones del menú, lo que permite regresar a la pantalla principal desde cualquier módulo. También se presenta el botón de Ayuda, que dependiendo del módulo cargará el PDF correspondiente. Otro botón disponible es el de Teclado, que muestra un teclado en pantalla cuando es necesario (se encuentra desactivado y solo se habilita en aquellos módulos que permiten el ingreso de texto). El botón de “Configuración” abre una ventana emergente que muestra las configuraciones de la mesa quirúrgica, la cantidad de lámparas instaladas, el tipo de quirófano que se va a usar (4x4 u 8x8), la cámara ambiental, la información del equipo y un video demostrativo que muestra el uso de la aplicación para nuevos usuarios. Finalmente, se presentan los botones de “Apagado” y “Reiniciado” del equipo, con los cuales el personal autorizado puede reiniciar o apagar el sistema. Como dato adicional, existe un botón de “Actualizar”; sin embargo, al inspeccionar el código, no se encontró ninguna función asignada a este botón.

El funcionamiento de los botones de “Menú”, ubicados en la parte superior, abre los formularios que solicitan el usuario y contraseña dependiendo de la autorización requerida (Administrador o Servicio). De manera similar, la configuración de dispositivos también es una forma a la que se accede y se presenta como una ventana. Estas funciones se encuentran fuera del código del panel

maestro, ya que se pueden llamar desde cualquier módulo de la aplicación. Sin embargo, dado que el panel maestro es el que contiene todos los paneles, la llamada a las funciones se sitúa dentro del código del panel maestro.

- **Submenú Cerrar**

Este submenú se enfoca en cerrar la aplicación y se utiliza en las funciones de los botones “Salir”, “Apagar” y “Reiniciar”. Esta función realiza una comprobación del usuario y la contraseña, cifrándolas mediante Hash MD5. Si la comprobación es correcta, se abre un mensaje de confirmación que, al aceptar la acción, comienza cerrando la comunicación serial, abre una ventana de Chromium para cancelar todos los subprocesos e inicia el proceso de borrar los objetos relacionados con las tarjetas de captura de video. Además, cierra las ventanas ubicadas en las pantallas secundarias, elimina los archivos temporales de las imágenes y, finalmente, verifica la acción (Salir, Apagar o Reiniciar) para ejecutar únicamente el proceso de finalizar la aplicación o continuar y ejecutar el proceso de apagado o reinicio según corresponda.

- **Mensaje Cargando y Mensaje de Información**

Este sencillo mensaje se muestra al usuario cuando un módulo tarda en iniciar y se cierra automáticamente una vez que se han cargado todos los componentes necesarios. El mensaje específico que aparece depende del módulo que se intenta abrir.

En cuanto a los mensajes de información, operan de manera similar, pero con la diferencia de que estos proporcionan información sobre alguna acción o error al intentar ejecutar alguna operación. Estos mensajes se cierran al hacer clic en “Aceptar”.

- **DEMO**

El Demo consiste en una serie de acciones sincronizadas con una imagen GIF que ilustra algunas funciones y pantallas disponibles en la aplicación. La sincronización se logra mediante temporizadores, llamando a las

funciones de cada módulo para comunicarse con alguna tarjeta serial y controlar algún dispositivo. Esta función se repite de manera indefinida hasta que el usuario decide detenerla.

- **Keyboard**

El keyboard es un teclado en pantalla que se abre cuando se hace click en algun campo donde permita la introducción de texto, este tambien se puede ocultar por si es necesario para visualizar la pantalla completa y despues volver a restaurarlo.

- **Ayuda**

La función de ayuda despliega documentos PDF dependiendo del módulo que esté en el panel principal. Para ello, se debe contar con la carpeta “appInterfaz/Manuales” que contenga los documentos PDF que se van a mostrar y tener asignada la ruta en la base de datos.

- **Configuración**

Finalmente, está el submenú de Configuración, que es el más importante de los mencionados, ya que permite seleccionar los modelos de la Mesa quirúrgica, la cantidad de lámparas, muestra la información del equipo, ejecutar el DEMO, dar de alta la cámara ambiental y elegir la cantidad de pantallas instaladas. Los cambios se verán reflejados en cada módulo, habilitando algunos botones en los módulos correspondientes y cambiando las imágenes de las pantallas para que se ajusten a los cambios. Estos cambios también se reflejan en la base de datos, de modo que, en caso de cerrar la aplicación o apagar el equipo, se conserven los ajustes realizados.

Pantalla de Informes

El primer módulo que se analizó fue el de informes. Este módulo se encarga de presentar en la pantalla de 38” la información de los gases quirúrgicos. También posee un cronómetro para utilizar en la cirugía, con 3 tiempos de

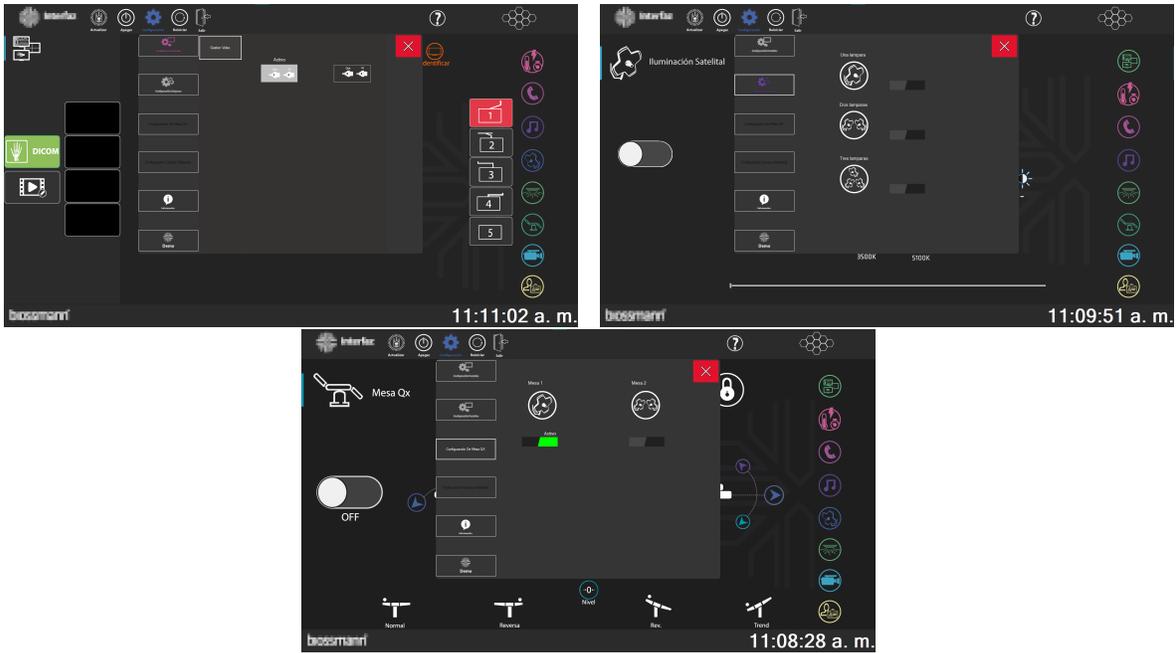


Figura 6.7: Pantallas de los menús de configuración.



Figura 6.8: Pantalla principal del monitoreo de gases y cronómetro.

pausa que se pueden almacenar para compararse entre sí. Aunque este módulo utiliza la pantalla super wide, no es indispensable para su funcionamiento, ya que es completamente operativo al presentar la información en la pantalla principal de 24". Sin embargo, la pantalla secundaria muestra más información en una posición de mayor visibilidad para los demás asistentes.

La comunicación se inicia al ejecutar la aplicación, ya que la información se presenta en la pantalla secundaria. Esta abre la comunicación serial por el puerto COM3 a 9600 baudios. Una vez que se logra establecer la comunicación, se toman los datos de la trama enviada por la tarjeta de gases y se separan los datos para generar las cadenas de texto que se mostrarán en ambas pantallas.

En la parte superior se encuentra un botón con el cual se puede cambiar a la pantalla del cronómetro. (Este botón no afecta a la pantalla secundaria, dejando el panel de información de gases quirúrgicos siempre a la vista). Este cambio permite cronometrar los procesos quirúrgicos. El cronómetro también es visible en la pantalla secundaria, junto a la información de los gases quirúrgicos y la hora real.

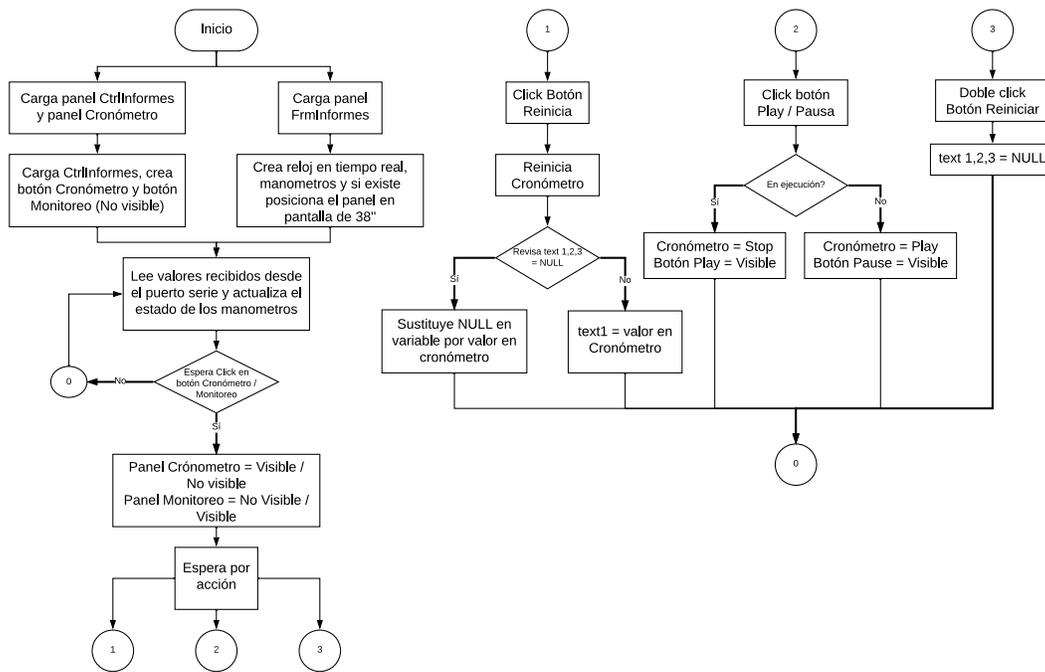


Figura 6.9: Diagrama de flujo del monitoreo de gases quirúrgicos.

Mesa Quirúrgica



Figura 6.10: Pantallas de mesa quirúrgica 7050 (izq.) y 8050 (der.).

La mesa de operaciones o quirúrgica fue el siguiente módulo analizado. Este consiste en poder controlar la mesa desde la pantalla principal al conectarse con una tarjeta serial por el puerto de comunicaciones COM10 a 9600 baudios. El funcionamiento es bastante simple, ya que se implementa una función a través de la cual se envían las tramas de control a una tarjeta externa con un emisor infrarrojo que la mesa recibirá e interpretará esas instrucciones. En este módulo por defecto se encuentra en la configuración de mesa avanzada o para el modelo 8050, ya que se cuenta con 2 modelos: 7050, versión básica con menor cantidad de movimientos, y 8050, que contiene toda la gama de movimientos disponibles para el cirujano. En ambas versiones, la comunicación es igual; el cambio de modelo se encuentra en deshabilitar los botones con los movimientos no disponibles para el modelo 7050 y cambiando el diseño de la pantalla para acoplarse al modelo seleccionado.

En las primeras versiones, era necesario realizar una acción de sincronización para que emisor y receptor quedaran emparejados y se pudieran realizar las acciones solicitadas.

Entre las acciones que pueden realizarse en este módulo se encuentran:

- Encendido/Apagado.
- Bloqueo/Desbloqueo.
- Subir/Bajar la posición de la cama.

-
- Inclinarse hacia los lados la posición de la cama.
 - Desplazar la posición de la cama.
 - Elevar/Bajar la posición de la cabeza y los pies de la cama.

También contiene algunas funciones predefinidas como:

- Reverse: Permite cambiar la posición de la cabeza a los pies.
- Normal: Revierte la función de Reverse anterior.
- Trend: Inclina la cama elevando la cabeza y dejando los pies más cerca al suelo.
- Reverse Trend: Realiza la acción inversa a la anterior elevando los pies y acercando la cabeza al suelo.
- Posición Cero: Coloca la cama en su posición inicial.

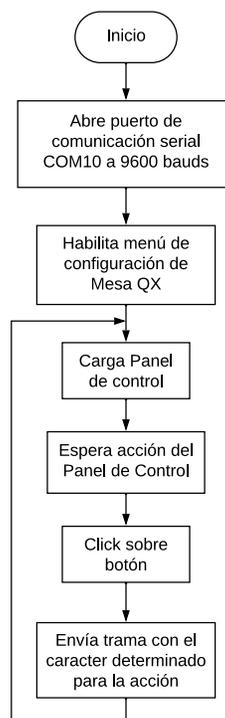


Figura 6.11: Diagrama de Flujo de la mesa quirúrgica.

Música

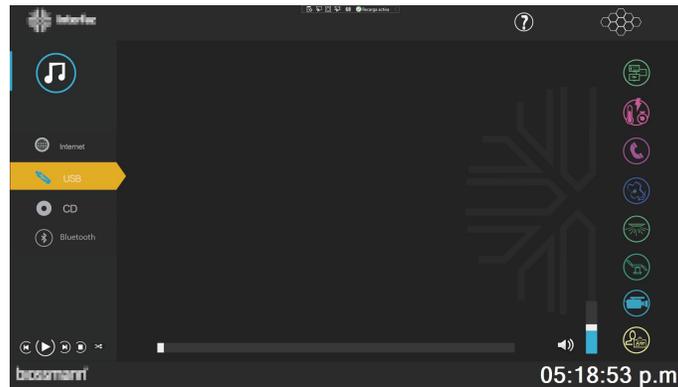


Figura 6.12: Pantalla de reproductor de música.

La sección de música consta de 2 reproductores en línea: YouTube y Spotify, y un reproductor local con el cual se pueden escuchar música guardada en el equipo o en un medio extraíble. La reproducción en línea se consigue al abrir una ventana usando el complemento de Chromium para abrir las pestañas con la dirección URL de las páginas web, mientras que la reproducción de medios locales es a través del reproductor de medios disponible desde los controles de .NET de Visual Studio.

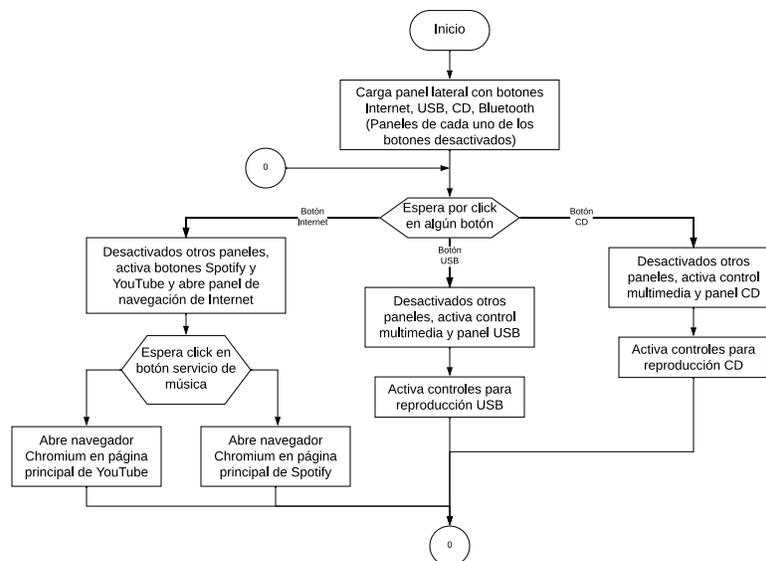


Figura 6.13: Diagrama de Flujo del módulo de música.

Videollamada

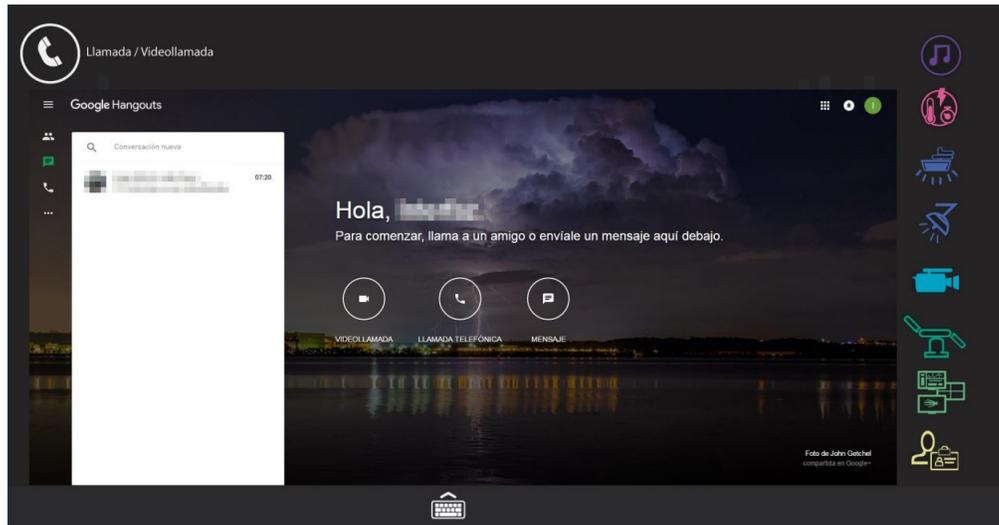


Figura 6.14: Pantalla de ejemplo de Hangouts.

Igual que la función anterior, la videollamada se carga en un panel que abre una ventana de Chromium en la que se redirige a la página de Hangouts de Google. Para los momentos en que se comenzó el proyecto, el servicio de Hangouts ya no se encontraba disponible, por lo cual solo se comprobó la conexión. Dentro del código, la única función que se ejecutaba era la de cargar la página web dentro del panel.

Luz Ambiental



Figura 6.15: Pantalla de control luces RGBW.

Con los paneles RGB, se utiliza la misma tarjeta de control de los gases.

En este caso, se envía una trama en lugar de recibirla. Los datos se codifican dependiendo de la posición del cursor en el círculo cromático, lo cual permite seleccionar tonalidades más acordes a las necesidades del momento. En otro caso, se puede elegir un tono directamente desde la paleta de colores. En ambos casos, es posible cambiar la intensidad del color seleccionado.

El proceso inicia con la selección de un color, el cual se transforma en codificación RGB en un valor de 0 a 255. Posteriormente, estos valores se envían a una función que les agrega una cabecera y una cola. Finalmente, se codifican en little endian (LSB - MSB, donde el bit menos significativo va en primer lugar) y se envían a la tarjeta para que esta se comunice con el controlador propio de los paneles RGB y se realice el cambio de tonalidad e intensidad de la sala.

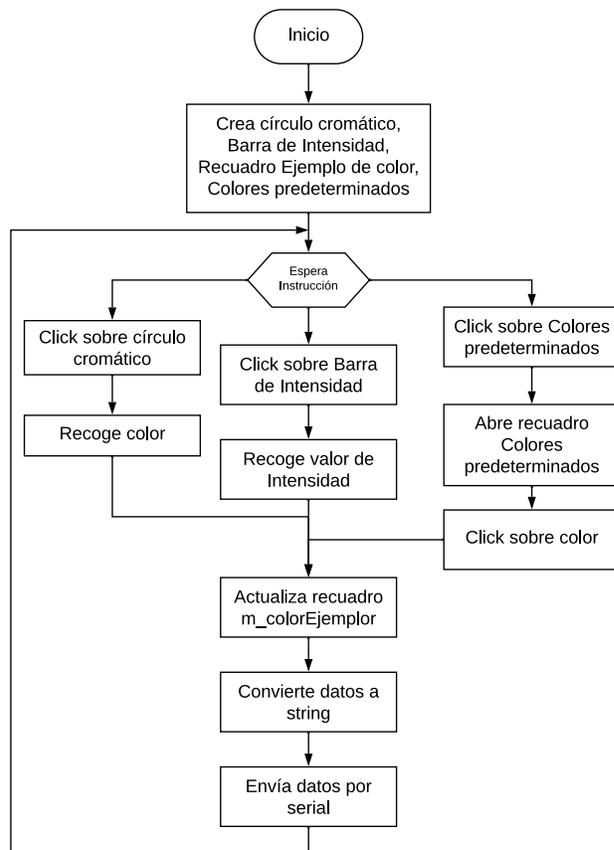


Figura 6.16: Diagrama de Flujo de la luz ambiental.

Historial Paciente

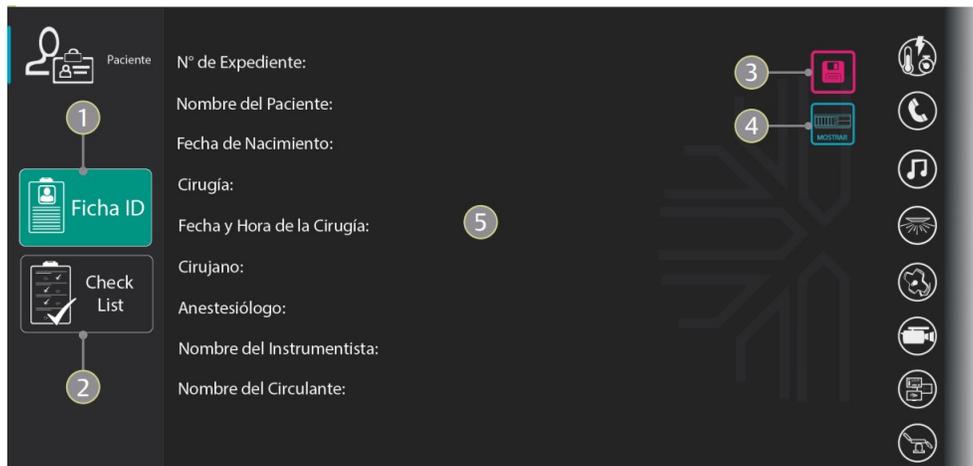


Figura 6.17: Pantalla inicial del Historial del paciente.

En un inicio, este módulo no contenía ninguna función; solo estaba la pantalla donde se veía que se podían llenar algunos datos. En el DEMO también se podía observar que en algún momento esta pantalla servía para mostrar y llenar información del paciente.

Cámara Ambiental

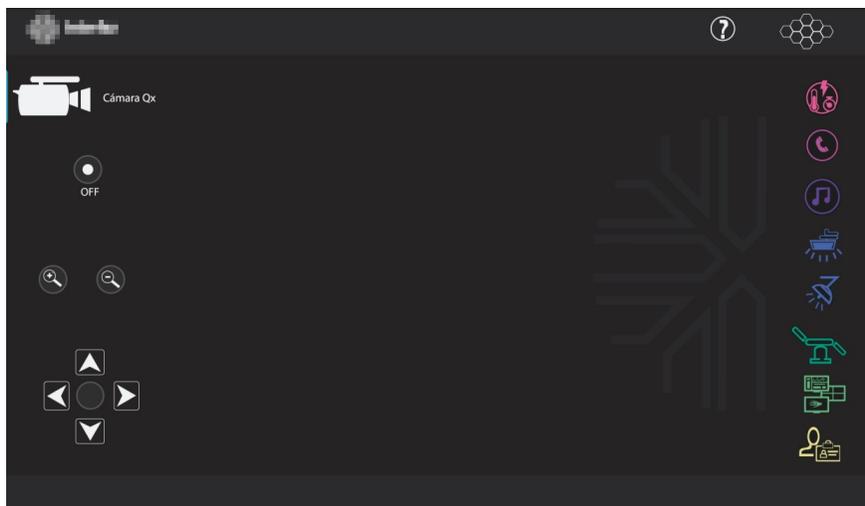


Figura 6.18: Pantalla de ejemplo de la cámara ambiental.

La cámara ambiental era un software independiente que se cargaba dentro

de un panel; sin embargo, los derechos de este código no estaban disponibles contrario al resto de la aplicación, lo cual provocó que este módulo fuera inoperante en un inicio y con los recursos disponibles. La única información disponible era que la cámara era del fabricante VStarCam, por lo cual solo se limitó a descargar el software de reconocimiento y de conexión de la cámara.

Lámparas Satelitales

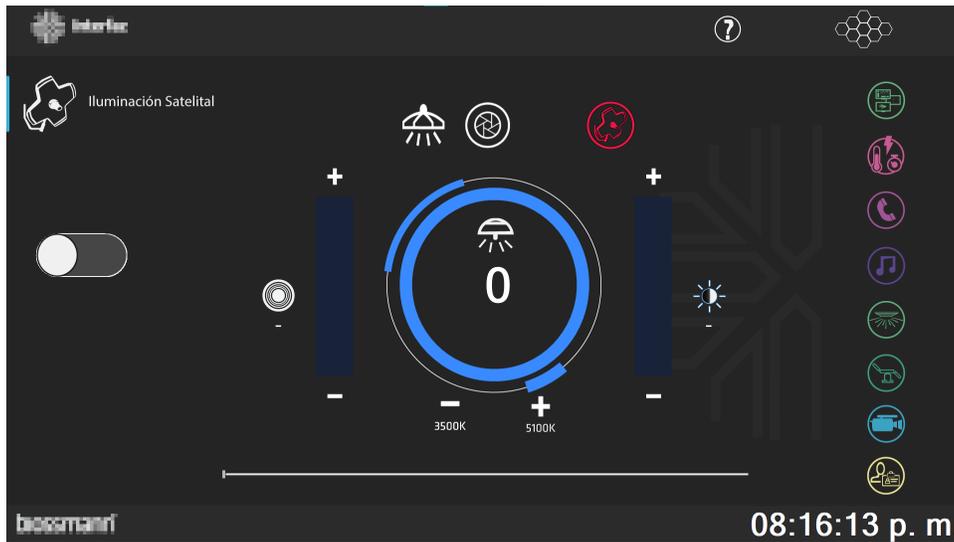


Figura 6.19: Pantalla de control de las lámparas satelitales.

Las lámparas son el penúltimo módulo; constan de una pantalla con barras deslizantes para seleccionar el brillo y el campo de luz que mostrarán las lámparas, y un selector de lámpara para indicar cuál se está controlando. Al igual que en la mesa, se cuenta con el botón de encendido y, en la configuración, se puede indicar cuántas lámparas se instalan por sala.

Un punto importante a destacar es que, inicialmente, se utilizó una versión para lámparas 250, las cuales tenían control sobre la intensidad, campo y calidez, mientras que más tarde se proporcionó el código de las lámparas 190, que eran más limitadas en funciones.

En ambas versiones, se puede encender y apagar la cámara instalada en la cúpula principal.

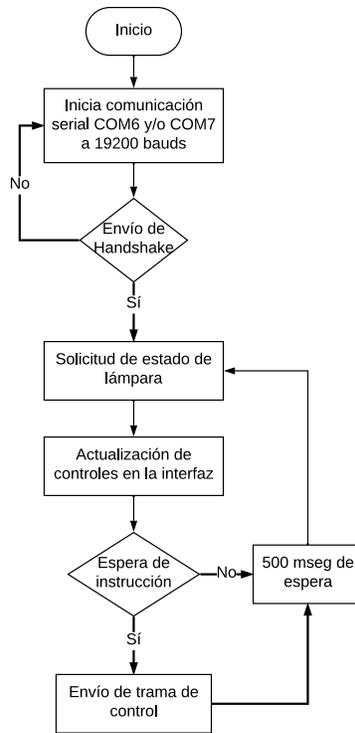


Figura 6.20: Diagrama de Flujo de las lámparas satelitales.

Gestión y Video

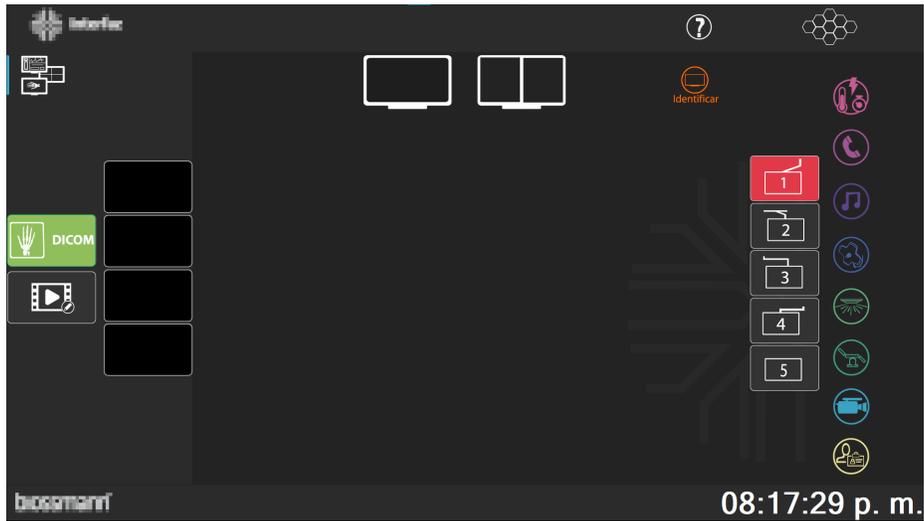


Figura 6.21: Pantalla inicial Gestión y Video.

El módulo de video es uno de los más complejos, ya que incluye diversas funciones y factores que pueden afectar su funcionamiento. En la primera parte,

se utiliza capturas de video que muestran el video de los dispositivos conectados a los puertos. Para que estas se muestren en la aplicación, se deben habilitar los puertos para permitir la reproducción y captura. Además de estas funciones, se agrega el envío de video a las pantallas externas y la división para tener múltiples videos en una misma pantalla. También se debe tener en cuenta que cada ventana se debe seleccionar para elegir hacia qué pantalla o división se posicionará, logrado mediante botones que muestran una previsualización de los videos. Como característica final, además de capturar el video para su almacenamiento en el disco duro de la PC, también se pueden editar los videos para generar recortes de fragmentos específicos del video, que también pueden ser exportados.



Figura 6.22: Pantallas de visualización de DICOMs y edición de video.

Dentro del módulo, se incluye la visualización de imágenes DICOM, que son imágenes médicas que normalmente se encuentran alojadas en el servidor del hospital. Aunque también pueden ser traídas por el paciente para que el cirujano las consulte durante la operación. En el primer escenario, existe la función para acceder al servidor y descargar las imágenes de manera temporal en el almacenamiento interno del equipo sin tener que cerrar la aplicación. Entre las funciones disponibles, además de enviarlas a las pantallas secundarias, es posible realizar zoom, rotarlas, modificar el contraste y brillo y, en caso de ser un compilado de imágenes, reproducirlas secuencialmente para simular un video. La conexión con el servidor permite tanto la descarga de las imágenes como la subida, asociándolas con el nombre del paciente y el nombre de la cirugía.

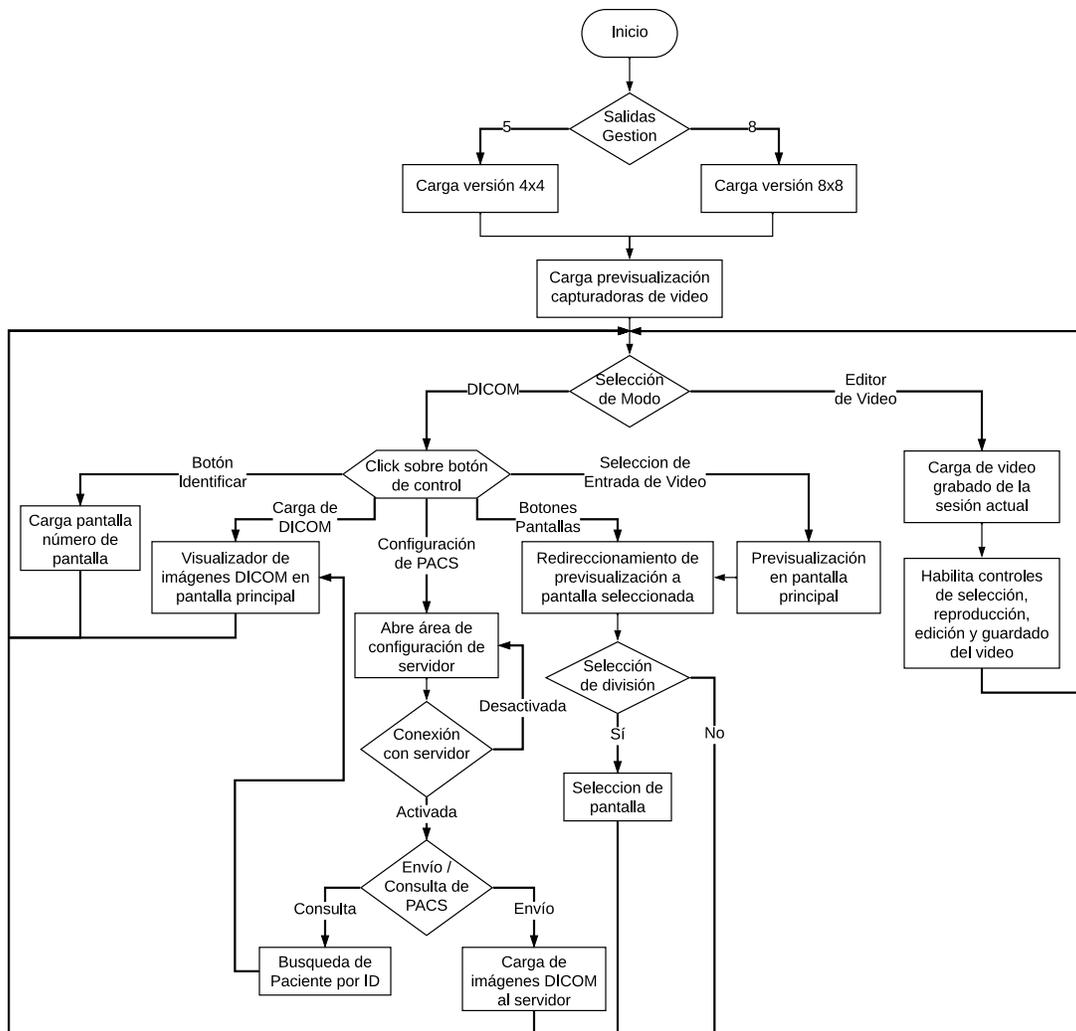


Figura 6.23: Diagrama de Flujo del módulo de Gestión y Video.

Requisitos de cada módulo

Al finalizar el análisis de cada módulo y su flujo de trabajo, se identificaron algunos requisitos externos necesarios para su correcta operación. Principalmente, se trataba de hardware adicional que se comunica con el software para controlar un dispositivo presente en el quirófano. En otros casos, era necesario contar con software adicional que permitiera el uso de bibliotecas propias para llevar a cabo determinadas tareas.

Luz Ambiental e Información de Gases

Uno de los primeros requisitos identificados para los módulos fue la tarjeta

de gases y control de luz ambiental. Esta tarjeta se distingue por utilizar un adaptador DB9 a USB para establecer comunicación serial con los dispositivos de sensado y control.

La comunicación se realiza mediante el protocolo serie RS232, que posibilita la transmisión bidireccional de datos (dúplex), permitiendo el control de las luces RGB al mismo tiempo que recibe la información de los sensores de presión de los gases.

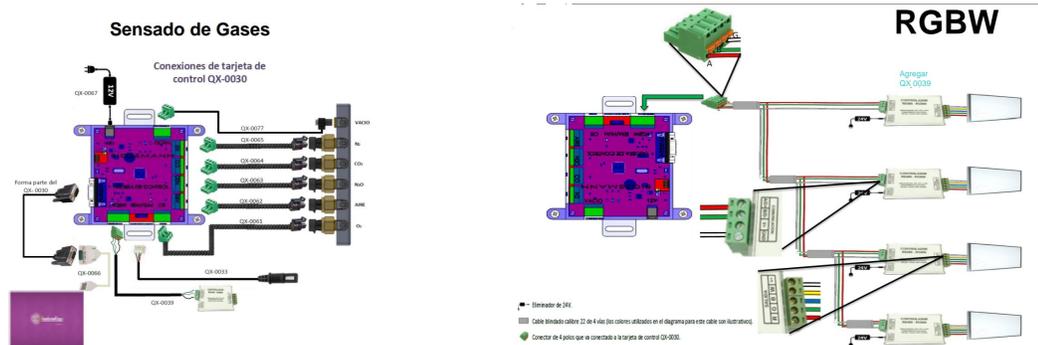


Figura 6.24: Tarjeta de sensado de gases y control de luz ambiental.

Mesa Quirúrgica

Las mesas quirúrgicas se controlan mediante tarjetas que incorporan emisores infrarrojos, emitiendo señales que las mesas pueden interpretar para llevar a cabo las acciones solicitadas por el usuario. Estas tarjetas se conectan a través del protocolo serial RS485. Aunque este protocolo permite la comunicación dúplex, en el firmware solo se utiliza el envío de datos en ambos modelos de mesa.

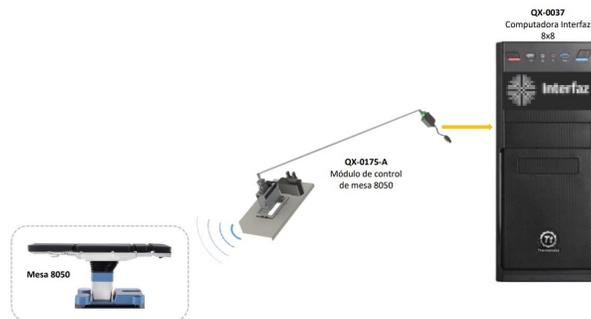


Figura 6.25: Tarjeta de control infrarrojo de Mesa quirúrgica

Lámparas Satelitales

La comunicación con las lámparas resultó ser algo más compleja, ya que al contar con dos modelos y revisar la documentación correspondiente, se descubrió que las funciones para la comunicación eran distintas. Ambos modelos utilizan el protocolo RS485, pero para los modelos 250 es necesario solicitar el estado de la lámpara, mientras que las 190 lo envían de manera automática. También se identificó una diferencia en la longitud de la trama, ya que las versiones de 190, al tener menos características, presentan una longitud menor en comparación con el modelo 250. Estas diferencias se lograron identificar gracias a la consulta de la documentación que existía de los modelos de lámparas que pertenecen al área de electrónica de Diseño y Desarrollo.

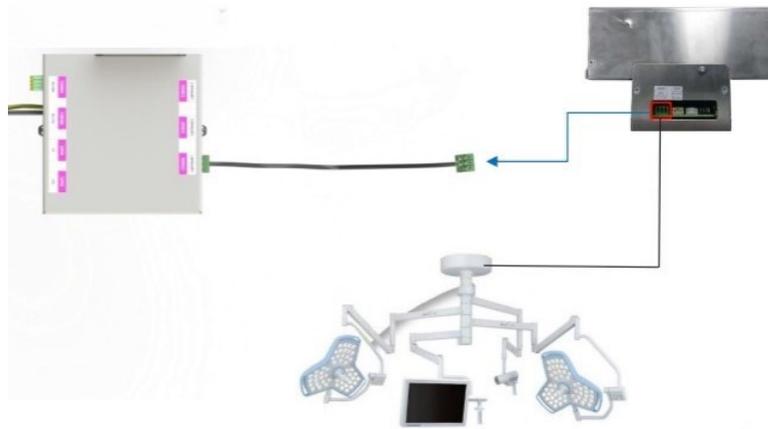


Figura 6.26: Conexión de lámparas satelitales con controlador y tarjeta de control

Cámara Ambiental

La cámara ambiental a diferencia de la que se encuentra en las lámparas satelitales se conectan a través de la red, esto ocasionó que además de requerir una conexión a red y un dispositivo capaz de interconectar estos dispositivos como un switch, es necesario que se den de alta las computadoras con un software propio del fabricante que escanea las redes en busca de los dispositivos y otro que permite la visualización, ya que al no contar con el desarrollo propio de la cámara y al no contar con el SDK ya que se encontraba discontinuado en la página oficial, se optó por crear una instancia de la aplicación para que este

se abriera de forma adicional a la aplicación principal.



Figura 6.27: Cámara ambiental

Gestión y Video

Como ya se había mencionado anteriormente, en el software encontrado en los equipos en desuso, los principales requisitos en la gestión de video son el software de Capture Studio y el SDK. Estos deben estar instalados en el equipo y deben corresponder al modelo de la capturadora de video.

El segundo requerimiento que se encontró fue que se necesitaban bibliotecas que decodificaran las imágenes DICOM. Para ello, se necesitó descargar el software KLite Codec Pack, con lo cual se cargaban las bibliotecas necesarias para visualizar las imágenes. Estas imágenes se encuentran en formato .dcm, el estándar con el que se comprimen es JPEG2000, el cual es comúnmente usado en imágenes médicas debido a su avanzado sistema de compresión y funciones de datos de archivo.

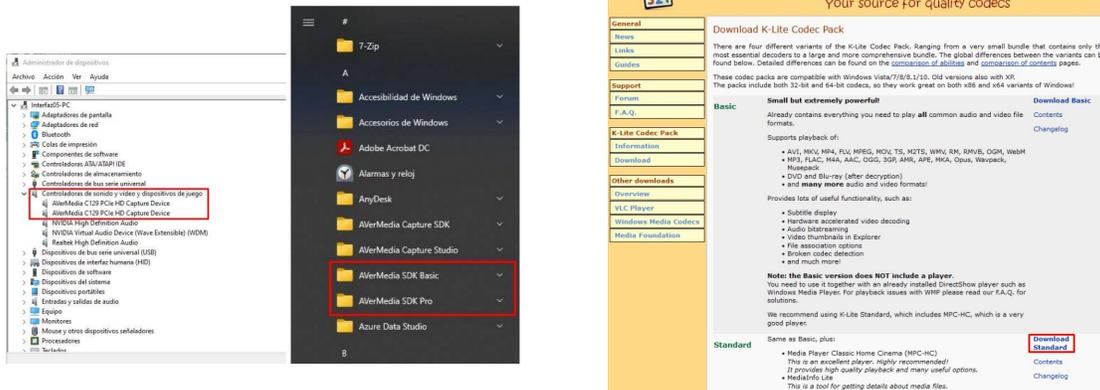


Figura 6.28: Software necesario para el módulo de gestión y video.

6.1.2. Pruebas

El autor Glenford J. Myers señala que las pruebas para los desarrolladores deben aportar un valor añadido a la aplicación, y propone un enfoque alternativo: “La prueba es el proceso de ejecución de una aplicación con la intención de encontrar errores”.

Siguiendo esta perspectiva, al realizar pruebas con el objetivo de encontrar errores, se pueden crear escenarios con mayor probabilidad de detección de fallos. Es importante comprender los términos asociados a las pruebas: error, defecto y fallo, según la definición del ISTQB (International Software Testing Qualifications Board ó Comité de Internacional de de Certificaciones de Pruebas de Software en español):

- Error: Es provocado por la acción humana.
- Defecto: Se deriva de un error de implementación.
- Fallo: Son la consecuencia de un defecto, por lo cual se obtienen resultados no deseados.

Es crucial distinguir entre estos términos, ya que un error puede generar un defecto, que a su vez puede causar un fallo en la aplicación. Sin embargo, no todos los defectos resultan en fallos.

Las pruebas se llevan a cabo mediante la ejecución de casos de prueba diseñados para encontrar errores. Dos técnicas comunes son:

- **Pruebas de caja negra:** Consiste en probar la aplicación sin conocer el comportamiento interno y centrandonos en el comportamiento incorrecto de acuerdo a las especificaciones, es decir en este tipo de practicas sólo se toman en cuenta las entradas y las salidas de la aplicación.
- **Pruebas de caja blanca:** Estas pruebas consiste en analizar la estructura interna del aplicación para obtener los resultados proyectados.

Es poco productivo utilizar solo una de estas técnicas, ya que la caja negra requiere probar todas las entradas y verificar los resultados, mientras que la caja blanca implica evaluar todos los caminos lógicos de la aplicación. Por lo tanto, es importante establecer límites y objetivos claros para las pruebas.

Pruebas con dispositivos en ambiente controlado

Después de revisar los módulos y establecer como objetivos principales el poder controlar todos los dispositivos asociados con la aplicación, se realizaron los casos de prueba sobre las tarjetas de control para analizar la interacción entre dispositivo y aplicación.

Tarjeta QX-0030 de Sensado de Gases y Control RGBW

- **Prueba de Sensores de Gases:** Se utilizaron sensores de presión de gases conectados a la tarjeta de control para verificar la comunicación y la interpretación de los datos en la aplicación. Para ello se realizaron pruebas de conexión usando el programa Terminal antes de probar directamente en la aplicación, con el fin de asegurar que esta estuviera enviando información a la velocidad especificada en las hojas de datos, y que las tramas tuvieran la estructura correcta para ser interpretadas por la aplicación.

Una vez completada la verificación de la comunicación entre la PC y la tarjeta, se iniciaron las pruebas de funcionamiento con la aplicación.

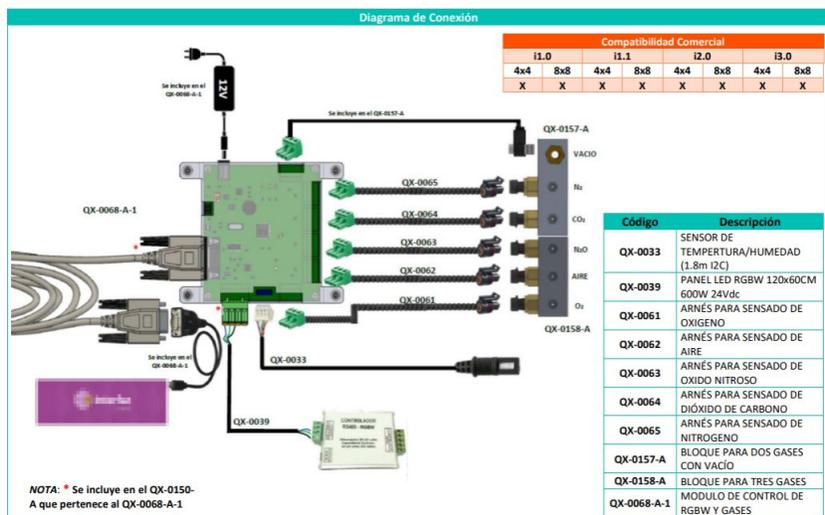


Figura 6.29: Diagrama de conexiones del sensor de gases y control RGBW.

Durante estas pruebas, se buscó confirmar la comunicación y la interpretación de los datos enviados por las tarjetas de cada uno de los gases. Para lograr esto, además de conectar los sensores uno a uno a los puertos mientras la aplicación estaba en funcionamiento, se varió el flujo de gas a través del sensor para obtener diferentes valores en la tarjeta con el fin de comparar los valores de los manómetros con los mostrados en pantalla.

El objetivo de estas pruebas era identificar posibles problemas con la conexión de los sensores en sus respectivos puertos, así como evaluar la fiabilidad de los sensores y la interpretación del código para mostrar los valores de los gases medicinales. Los resultados de estas pruebas mostraron que la interpretación de los datos era correcta, al igual que el despliegue de los valores en pantalla.

- **Prueba de Control de Paneles RGBW:** Continuando con las pruebas, se procedió con el módulo de luz ambiental, el cual se encarga de controlar paneles RGBW a través de un puerto situado en la tarjeta junto a los que recuperan los datos de los sensores de gases.

Se volvió a utilizar el programa Terminal para intentar enviar una trama con la estructura que maneja la aplicación para controlar las luces. Esta

prueba resultó exitosa al enviar una trama con valores hexadecimales de varios colores en RGB, logrando cambiar el color en los paneles.

Posteriormente, utilizando la aplicación, se inició el control de las luces a través del módulo. Se logró codificar el color que se mostraba visualmente en un círculo cromático y en una paleta de colores con tonos predeterminados. Es importante destacar que ambos controles que se encontraban en la interfaz de usuario fueron personalizados por parte de los desarrolladores originales. Sin embargo, a pesar de conocer cómo funcionaban las tramas y tener la facultad de cambiarlos, se decidió conservar los controles debido a la familiaridad que ya se tenía con ellos.

Durante las pruebas, no se presentaron mayores complicaciones al probar alguna tonalidad. La única observación fue el ligero retraso en los cambios de color después de seleccionar alguno.

Tarjeta QX-0024-A de Control de Mesa Quirúrgica

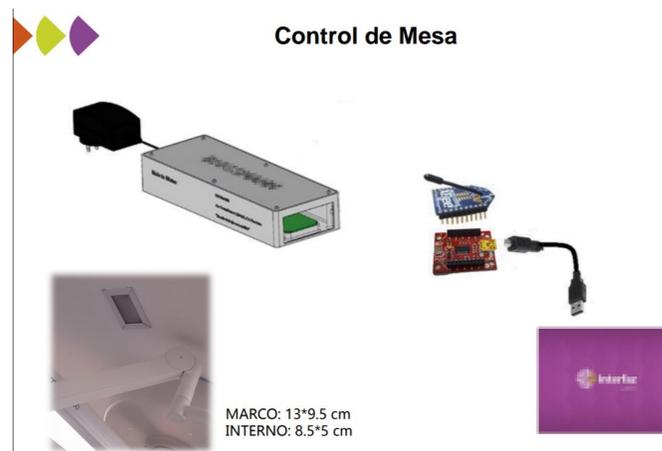


Figura 6.30: Cámara ambiental

A continuación, se comenzó con la prueba de la tarjeta de control de la mesa. Al igual que con la anterior, no hay comunicación bidireccional con los dispositivos, por lo que la visualización del movimiento de la mesa es la única prueba con la que se cuenta.

Al iniciar estas pruebas y tratar de enviar las instrucciones a la tarjeta de control, se encontró que los únicos controles que funcionaban correctamente eran el de encendido y anclaje de la mesa. Estas primeras pruebas se realizaron usando únicamente la tarjeta de control, y para comprobar el funcionamiento, se utilizaba la cámara de los celulares para verificar que los LEDs se encendieran al enviar la instrucción. Esto permitió verificar que solo los controles anteriormente mencionados estaban actuando correctamente.

Por lo tanto, se decidió utilizar nuevamente el software Terminal para intentar enviar las tramas de control para la mesa. En este caso, se logró enviar las tramas de manera única, consiguiendo leves movimientos de la mesa. Las pruebas hasta el momento se realizaban usando ratón y teclado por lo cual, continuando con el análisis del código, se descubrió el uso de una librería personal para los botones de movimiento de la mesa, con la que al utilizar el monitor táctil y dejar presionado durante largo tiempo el panel, este por configuración de fábrica realiza la función de clic izquierdo, y con esta librería, se puede utilizar como una función de mantener presionado el botón, encontrando así la causa del fracaso de los primeros intentos de control.

Finalmente se encontró al realizar estas pruebas fue que al enviar las instrucciones, en un punto debido a la forma de envío de las tramas, la instrucción de paro se mezclaba con la anterior trama de movimiento, provocando una confusión de lectura de datos provocando que los LEDs se quemaran al permanecer en un envío de datos continuo. Este error se identificó gracias al uso del osciloscopio, ya que al utilizar este dispositivo, se verificó que no existía un espacio entre las tramas enviadas, lo que el microcontrolador situado en la tarjeta consideraba como una sola trama.

Módulo de Control para Lámparas QX-0209-A

Después de obtener las lámparas, comenzaron las pruebas, pero desafortunadamente solo se pudo contar con el modelo de lámparas 250, el más actual, por lo que estas pruebas solo se pudieron realizar con este tipo de lámparas.

Con respecto a la conexión de las lámparas, se utiliza un adaptador USB

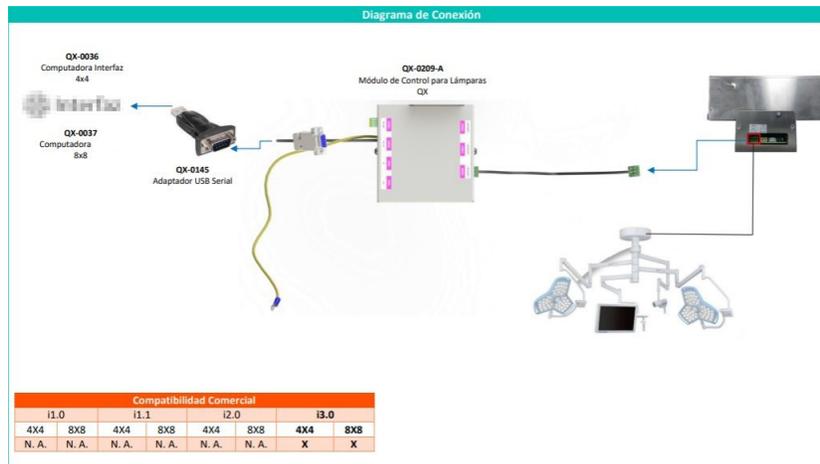


Figura 6.31: Diagrama de conexiones de lámparas satelitales.

- RS232, que interactúa con el controlador de las mismas. Este adaptador permite al usuario utilizar las funciones de la lámpara desde la aplicación.

Las primeras pruebas se enfocaron en lograr que la aplicación se comunicara con las lámparas y mostrara el estado de las cúpulas (se le denomina cúpula a la parte de la lámpara que proporciona la iluminación). Al usar la configuración que se encontró en el código, se logró establecer la comunicación entre la aplicación y la lámpara. Inicialmente, no se percibió el estado de la lámpara en los controles deslizantes, ya que estas necesitan estar encendidas. Entonces, se intentó usar el botón de encendido para comprobar si este era funcional, logrando así encender y apagar las lámparas instaladas. En este mismo escenario, se intentó cambiar de puerto COM para simular el uso de la lámpara 3, ya que en los menús de configuración se pueden seleccionar hasta 3 cúpulas. Sin embargo, en este caso no se logró debido a que el código se encontraba incompleto y, a pesar de que los controles estaban presentes, no tenían funciones asociadas. Por lo tanto, se procedió con las siguientes pruebas.

En las siguientes pruebas, se verificó que los cambios realizados desde las lámparas se reflejaran en la aplicación y viceversa. Se aprovechó para comprobar que los controles funcionaran correctamente. Después de probar el encendido y apagado desde ambas partes, se continuó con el control de la intensidad, el campo de iluminación y la calidez de la luz. En todos los casos, se encontró

un retraso en el envío de la instrucción y el cambio en el estado de la lámpara. Analizando el código, se descubrió que el uso de los `backgroundWorkers` era la causa de estos retrasos en ambos casos. Como última prueba, se mencionó que había una función para cambiar el tono de las lámparas a color verde, la cual se encontraba en desarrollo pero no se había entregado. Por lo tanto, utilizando las hojas de datos de las lámparas, se encontraron las instrucciones de envío de la trama para usar esa función. Para probar esta función de manera rápida y sin hacer modificaciones aún al código, se utilizó nuevamente la aplicación de Terminal para enviar la secuencia, logrando de esta manera la función antes mencionada (MIS).

Gestion y Video

Estas pruebas resultaron ser las más extensas ya que este módulo se compone de tres secciones:

- Visualización de imágenes DICOM.
- Direccionamiento de pantallas.
- Captura de video.

Para la primer parte se tenía que lograr cargar una carpeta completa de imágenes en secuencia que debían verse de manera cíclica para dar la sensación de estar viendo un video. También fue necesario poder cargar las imágenes desde un dispositivo extraíble y desde el servidor del hospital. Esta última prueba se tuvo que realizar hasta que se tuvo acceso al ambiente productivo. Como ya se mencionó, era necesario el uso de la biblioteca `KLite` para poder visualizar estas imágenes. La primer prueba consistió en poder visualizar las imágenes de manera individual y poder manipularlas, es decir, hacer zoom, moverlas a lo largo y ancho de la pantalla, rotar las imágenes y cambiar el contraste y brillo. La siguiente prueba fue cargar un conjunto de imágenes que simulaban un video. Al conseguirlo, se repitieron las mismas pruebas, agregando además el poner pausa y reiniciar la reproducción y pasar imagen por imagen.

Concluidas estas pruebas, se procedió a intentar proyectar las imágenes en una pantalla extra y lograr controlarla desde la pantalla principal. Estas pruebas no generaron mayores conflictos y sirvieron para comprender el uso de los controles de pictureBox de Windows Forms cuando se utilizan en la visualización de imágenes DICOM y no en video.

La segunda parte de las pruebas resultó ser más complicada ya que a pesar de usar los drivers, SDK y las aplicaciones oficiales que proporcionaba AVerMedia, no se lograba hacer funcionar las tarjetas dentro de la aplicación. A pesar de que los drivers y el software proporcionado se ejecutaban de manera correcta, se dedujo que lo que causaba conflictos era el SDK ya que al cambiar el modelo también se actualizaron los drivers y el SDK. Ya que en el software funcionaban de manera correcta, se comenzó entonces a tratar el problema desde la perspectiva de solucionar la incompatibilidad del SDK.

Buscando en los foros de AVerMedia, se encontró en una de las preguntas de la comunidad que con una tarjeta diferente se solucionaron problemas similares al usar una versión antigua del SDK, así que se optó por probar con diferentes versiones de SDK disponibles en el centro de descargas. Finalmente, usando la versión 1.2.1.17 se logró dar imagen dentro de una aplicación de prueba. Como se mencionó, antes de iniciar y probar esta solución en la aplicación principal, se utilizó una aplicación de prueba desarrollada en C# y Windows Forms tratando de replicar todas las funciones que se encontraban en la aplicación original.

Las primeras funciones que se buscaron replicar fueron las de duplicar la imagen, es decir, previsualizar la captura de un puerto y enviarla a una nueva pictureBox de diferente tamaño. Luego se intentó cambiar la resolución. Esto dependía del modelo, ya que existen diferentes resoluciones compatibles, descartando aquellas en las que la imagen no se visualiza de manera correcta. La última prueba realizada antes de agregarla a la aplicación principal fue poder realizar una grabación del contenido del puerto de captura. Al conseguir todos estos, se agregaron los cambios y se realizó la prueba con las nuevas

características y el SDK compatible.

Historial del Paciente

No se realizaron pruebas ya que el módulo no tenía funciones implementadas.

Música

Se probó la conexión a servicios de música en línea y la reproducción de archivos musicales desde un dispositivo USB externo sin encontrar problemas.

Videollamada

No se realizaron pruebas debido a la indisponibilidad del servicio de Hangouts.

Cámara Ambiental IP HD QX-0016

No se realizaron pruebas debido a la falta de acceso al código de la aplicación ya que este no fue proporcionado.

Estas pruebas permitieron identificar y resolver problemas en la comunicación y funcionalidad de los dispositivos asociados con la interfaz, así como mejorar la experiencia del usuario al controlar estos dispositivos desde la aplicación.

Pruebas con nuevo hardware

Hasta este punto, las pruebas se habían realizado con equipos presentes en el laboratorio, y en el caso de las tarjetas de captura de video, se instalaron individualmente en el equipo de prueba. Los equipos prototipo aún no se habían entregado, por lo que se tuvo el tiempo suficiente para estudiar el código y realizar las pruebas necesarias, además de buscar la manera de desarrollar un instalador, ya que antes las instalaciones se realizaban copiando la carpeta resultante del debug. Gracias a esto, con el equipo prototipo además de probar que funcionaran todas las características y se proyectaran en todas las pantallas de acuerdo al índice asignado se intentó realizar la instalación con el ejecutable generado, consiguiendo una instalación de la aplicación de manera correcta.

Teniendo un equipo únicamente con el sistema operativo, se procedió a instalar todos los softwares necesarios que se fueron encontrando durante el

análisis y las pruebas del código fuente, además de instalar los codecs de imagen y actualizar los controladores de puertos COM, ya que estos de un inicio no lograban reconocer los convertidores seriales a USB. La parte más complicada de estas pruebas fue instalar el motor y el manejador de la base de datos, ya que por petición de la gerencia se solicitó que todos los softwares se incluyeran en una carpeta de manera que se pudieran instalar en sitio y lo cual resulta complicado al no contar con acceso a Internet en todas las instalaciones, al mismo tiempo se genera un problema ya que se descargan actualizaciones. Ante esto, se sugirió que la instalación de estos softwares, junto con los controladores de las tarjetas de video y actualizaciones de los drivers, se realizaran por parte del proveedor, de esa manera el área de calidad al recibir el producto puede comprobar que el software viene instalado de manera correcta y en caso de que haya algún fallo en la instalación, se puede regresar al proveedor para que se reinstale, y de esta manera, el personal de TI solo se encargue de instalar nuestra aplicación y generar la base de datos con el script que se creó al estudiar las tablas necesarias para ejecutar la aplicación.

El siguiente paso consistió en utilizar el ejecutable para instalar la aplicación en el equipo. Utilizando el software Inno Setup para generar el instalador, también se buscó que se crearan las carpetas necesarias para el funcionamiento, por lo cual al terminar el Asistente de Instalación se verificó que todas las carpetas quedaran en sus respectivas rutas. De esta manera, solo restaba incluir los archivos que vienen dentro de las carpetas que hasta el momento de presentación de los resultados no se habían agregado a la instalación de manera automática.

Concluidas las pruebas de instalación, se siguió con las pruebas de funcionamiento de la aplicación creada. De inicio, se encontró que por permisos de Administrador era necesario que la ubicación de la instalación se realizara sobre la carpeta Documentos, ya que al requerir el acceso a carpetas dentro de sí misma y estar instalada por defecto en disco duro C, estaba limitadas estos accesos. El siguiente detalle que se notó fue que el archivo “config.xml” era ne-

cesario reemplazarlo manualmente, ya que de momento no se había creado un menú para ingresar los datos necesarios para que se editara. Esto no sucedía así con la conexión con el servidor de imágenes DICOM del servidor (PACs), ya que estos sí poseen un menú para ingresar los datos necesarios. Finalmente, hubo un fallo al generar algunas librerías de AVerMedia, ya que se cargó una versión equivocada que poco después se corrigió por la adecuada para que las capturadoras funcionaran de manera correcta. Todos estos detalles se comenzaron a documentar ya que estar próximas las pruebas en un ambiente productivo e involucrar al área de TI es necesario comenzar a capacitar a quienes estarán a cargo de las instalaciones del software.

En este punto de las pruebas y al involucrarse el área de TI, estos apoyaron con el software Orthanc, con el cual se consiguió crear un servidor local para probar la función de PACs en el módulo de gestión y video. Este software permite subir y bajar archivos DICOM. Usando la aplicación, con la cual se logró cargar y descargar las imágenes al servidor creado de manera local, con lo que quedó comprobada la funcionalidad del submódulo de DICOMs.

Al finalizar todas las revisiones e instalar el equipo, se constató que funcionaba de acuerdo a los requerimientos establecidos, por lo que se procedió a comunicar los resultados y solicitar una prueba en un ambiente productivo en alguna sala quirúrgica que estuviera disponible.

Pruebas en ambiente productivo

Una vez cumplidas las pruebas en el laboratorio con el equipo personal y el prototipo, se buscó una locación para instalar el equipo y comprobar su comportamiento en un ambiente sobre el cual no se tiene control total y que puede no estar en óptimas condiciones. Los lugares seleccionados para estas pruebas fueron 3 hospitales dentro de la ciudad. En estos sitios se realizaron diversas pruebas dependiendo de los dispositivos instalados.

Las pruebas realizadas en los tres hospitales incluyeron el control de las luces ambientales, la monitorización de la presión de los gases medicinales, el

cronómetro, la música y la gestión de video. Respecto a los módulos de mesa quirúrgica y lámparas satelitales, en 2 de ellos no se contaba con lámparas 250, ya que al ser salas quirúrgicas de mayor antigüedad, estaban equipadas con lámparas 190. En cuanto a los modelos de mesa, a pesar de existir dos versiones (7050 y 8050), no se presentaron inconvenientes significativos, ya que el control de las mesas era similar, aunque con acciones limitadas en el modelo 7050 al ser más básico. Estas pruebas se mencionan aquí porque en el caso de las lámparas no se pudieron probar debido a que el código proporcionado difería del instalado en las versiones más antiguas, y el código de la mesa solo se pudo probar en el laboratorio con la versión 8050 debido a la limitación de equipos disponibles en el almacén. Por lo tanto, en las salas quirúrgicas se usaron solo las mesas modelo 7050, y las pruebas se realizaron con el fin de comprobar el funcionamiento de la infraestructura.

Tras la instalación del equipo y la ejecución de la aplicación, se comenzó por probar los equipos más importantes de la sala, como la mesa y las lámparas. Uno de los primeros problemas identificados fue la confusión al asignar los puertos COM, ya que no se distinguía qué cable pertenecía a qué dispositivo. Para solucionar estos conflictos, se conectó uno por uno los cables y se probaron las diferentes configuraciones para determinar a qué dispositivo correspondían. Se sugirió además rotular los cables para facilitar su identificación.

Para identificar cada cable, se utilizó el propio módulo para intentar controlar dicho dispositivo. Sin embargo, surgieron dos problemas: primero, al hacer la pregunta si el cable conectado era el correcto o si la configuración del dispositivo era adecuada dentro del código, se tuvo que realizar la prueba con todos los módulos que tienen acceso a los puertos COM. Una vez que se determinó que cada dispositivo era el adecuado, se realizó la prueba completa de cada módulo.

La parte de monitoreo de gases y luz ambiental resultó más sencilla de identificar, ya que al tener instalada la pantalla de 38" y configurar correctamente las tarjetas de control, la información de los gases medicinales apareció inme-

diatamente en la pantalla. Concluida la prueba de monitorización, se procedió con la prueba de la luz ambiental, que se realizó sin mayores complicaciones.

El siguiente caso fue controlar la mesa, siguiendo los mismos pasos, pero ahorrándose la conexión de un cable al ya tener identificado el de monitoreo. Al intentar encender la mesa, hubo algunos problemas de sincronización, ya que los nuevos modelos no incluyeron esa funcionalidad, sin embargo, al intentar anclar la mesa en la superficie, esta se consiguió casi de inmediato. Los siguientes movimientos se llevaron a cabo sin mayor problema, excepto por la situación previamente mencionada de ciclos de envío de instrucciones que provocaban el quemado de los LEDs de transmisión.

A pesar de que los modelos de lámparas no correspondían con los instalados, se intentó comunicarse con estos e intentar encenderlos, lo cual no funcionó debido a tramas de diferentes longitudes, pero era una prueba solicitada por el coordinador.

En cuanto a las conexiones con las pantallas, se encontraron dificultades al momento de dar de alta el nombre, debido a que había pantallas del mismo modelo, lo que provocaba que al direccionar los controles de las pantallas, estos no se desplegaran en el orden correcto o no se mostrara nada. Por lo tanto, fue necesario intentar escribir el nombre de las pantallas de varias maneras hasta lograr que se enviaran correctamente las señales de video. Estos casos incluían el uso de mayúsculas al inicio del nombre, cambio de espacios por guiones medios y bajos y agregar un número que diferenciara a las pantallas, en cada instalación cambiaba el resultado esto debido al Sistema Operativo Windows y su forma de detectar las pantallas.

Una vez direccionadas correctamente las pantallas, se probaron las capturas de video y el envío de imágenes a las pantallas correctas. Con el apoyo del área biomédica del hospital, se proporcionó un monitor de signos vitales con conexión HDMI para conectar a las torres donde se encuentran los puertos de entrada. Al conectar el monitor, se encontró que en algunos funcionaba y en otros no, lo que llevó a investigar por qué solo funcionaba en ciertos puer-

tos. Se descubrió que los primeros dos puertos se configuraban en sitio para la aplicación con una resolución full HD (1080p), mientras que los restantes se configuraban en una resolución de VGA (480p), lo que provocaba que el monitor, al tener una resolución más baja, no se visualizara correctamente en los puertos full HD, siendo necesario bajar la resolución del video desde la aplicación. Con la captura de video no hubo mayores problemas, logrando capturar el video proyectado por el monitor de signos vitales.

Por último, se realizó la prueba de las imágenes DICOM, para la cual fue necesario configurar la PC con el servidor del hospital. Para comprobar que la configuración fuera correcta, se utilizó el botón de PING para verificar que estuvieran en la misma red y se pudiera alcanzar el servidor desde la PC. Si esta configuración era correcta, significaba que se podía conectar al servidor, pero no era garantía de que se pudieran descargar las imágenes. Para ello, la contraseña debía ser correcta, ya que no se podía comprobar hasta que se intentaran descargar imágenes, en cuyo caso se mostrarían los nombres de los pacientes pero no habría imágenes al seleccionar un expediente de imagenología. Si los archivos se descargaban correctamente, era posible visualizarlos y manipularlos como en las pruebas en laboratorio descritas en el capítulo anterior.

Las pruebas restantes no se consideraron necesarias, ya que no requerían dispositivos presentes en el quirófano y se habían realizado las pruebas pertinentes en el laboratorio.

Concluidas las pruebas tanto en laboratorio como en quirófano, se consideró finalizada esta fase con la elaboración del manual de instalación y técnico de la aplicación. Esto permitirá que el área de TI se encargue de llevar a cabo las tareas de instalación y solución de problemas relacionados con la aplicación.

6.1.3. Sugerencias de mejoras en el código

Al finalizar los casos de prueba tanto en laboratorio como en un ambiente real, se identificaron diversas redundancias y funciones obsoletas, así como un

mal uso de ciertas partes del código. Estos problemas ocasionaban retrasos y fallas en algunas secciones de la aplicación, lo que resultaba en su cierre inesperado. Gracias a los conocimientos adquiridos durante la estancia en la Facultad de Ingeniería y al tiempo dedicado al estudio del código, se pudo determinar la manera más eficiente de realizar ciertas tareas. Aunque en el código estas tareas parecían simples, en realidad generaban procesos que ralentizaban la ejecución de otras funciones.

Identificación de las posibles correcciones

En los siguientes módulos, se identificaron funciones que podrían mejorarse de manera inmediata para lograr una mayor fluidez en la interfaz de usuario.

- Videollamada: Se utilizaba Hangouts, una versión obsoleta del servicio de videollamadas de Google, que ya no era accesible. Se propuso cambiar a la versión web de Microsoft Teams, que permite realizar llamadas de voz y transmitir video desde las capturadoras y la cámara montada en las lámparas satelitales. Para realizar este cambio, se continuó utilizando el complemento del navegador Chromium, que carga un navegador web dentro del panel de Windows Forms, al cual se le asigna la dirección URL del sitio al que se desea acceder (Teams). Con este simple cambio, el módulo funcionó completamente.
- Lámparas Satelitales: Aunque estos dispositivos parecían funcionar correctamente, en realidad generaban latencia en los controles debido al uso incorrecto de las funciones de background worker, esto se reflejaba al no realizar las acciones que se enviaban desde el panel de control. Para mejorar esto, se eliminó el background worker y las funciones empaquetadas en una DLL de los desarrolladores anteriores. En cambio se utilizaron las funciones predeterminadas de Windows Forms, lo que mejoró significativamente el tiempo de respuesta en las lámparas. Finalmente se agregaron funciones que nunca se implementaron, como la función de

MIS (colocar las lámparas en verde) y el control de la cámara instalada sobre la cúpula.

Para una actualización posterior se tiene la idea de agregar la selección de modelo de lámparas para que un mismo software pueda controlar cualquier tipo de lámpara.

- Monitoreo de Gases: Se propuso añadir todos los gases médicos junto con una barra de desplazamiento para tener disponibilidad de ellos en cualquier hospital, incluso si alguno no está presente físicamente.

Además de este cambio se implementó en la pantalla de 38" un cambio de la visualización de gases por la información de la cirugía, y poder intercambiar entre ambas informaciones en cualquier momento.

- Mesa Quirúrgica: Se resolvió el principal fallo relacionado con el envío continuo de instrucciones a las tarjetas, agregando una función de parada cada 20 milisegundos. Esto permitió que las tarjetas se detuvieran en la acción de envío de datos a la mesa.
- Gestión y Video: En esta sección, se modificó la forma de inicializar y dirigir las pantallas, ya que los métodos anteriores utilizaban funciones cíclicas que ocasionaban retrasos en la reproducción de imágenes y, en ocasiones, causaban superposición de la ruta en la que debía mostrarse la imagen. Respecto a las imágenes DICOM, se solicitó mejorar su reproducción, pero esto no fue factible debido a la incompatibilidad entre las librerías NuGet y el proyecto. Como solución provisional, se bloqueó la actualización de estas librerías y se documentó este detalle en el manual.

Los siguientes módulos requerían un rediseño completo, al no tener funcionamiento, con permiso de la gerencia se llevaron a cabo los siguientes cambios:

- Cámara Ambiental: Se diseñó un nuevo control de cámara para un modelo actualizado, esto se detallará en el próximo capítulo ya que se planeó hacer un cambio completo de la aplicación.

-
- **Historial de Paciente:** Al estar completamente vacío de funciones y contar solo con una imagen de referencia, se implementó un cuestionario que permite el ingreso de datos del paciente y la cirugía, así como la conexión con el software de monitoreo de signos vitales de la empresa. Esto permite registrar al paciente desde cualquiera de los dos softwares y mostrarlo en las pantallas principales.

Este software de monitoreo de signos vitales diseñado para hacer un seguimiento del paciente durante todo el proceso quirúrgico, proporcionando un respaldo tanto para médicos como para pacientes. Esta aplicación utiliza una base de datos MySQL que se ejecuta en un servidor externo, a diferencia de la aplicación de control de Quirófano Inteligente, cuya base de datos se ejecuta en la misma PC. Permite capturar información sobre el paciente, el personal médico y los detalles de la cirugía. Además, otro aspecto crucial es su capacidad para recopilar información para la evaluación de la cirugía, así como para el control de los medicamentos administrados, el seguimiento de las tendencias de los signos vitales al enlazarse con los monitores de signos vitales presentes en la sala, y el registro de los datos de recuperación. Toda la información recopilada en esta aplicación se almacena en la base de datos para que luego pueda ser impresa y consultada por el médico una vez finalizada la cirugía.

6.1.4. Presentación del Proyecto

Después de completar las pruebas y obtener un mayor entendimiento del proyecto, se identificaron las áreas de mejora en el diseño inicial, proponiendo así las sugerencias de mejoras en el código. Considerando también la incorporación de nuevos dispositivos como la Cámara Ambiental, una nueva PC y tarjetas de control con firmware actualizado, se tomó la decisión de lanzar una nueva versión que integre todas estas actualizaciones. Este proceso incluyó el desarrollo de una interfaz gráfica renovada para proporcionar una experiencia

mejorada a los usuarios.



Figura 6.32: Pantalla de ejemplo de la cámara ambiental.

Debido al descuido en el mantenimiento del software y el ciclo de vida del proyecto, se procedió a actualizar varias librerías durante su análisis para evitar conflictos. La plataforma de destino del proyecto se actualizó al .NET Framework 4.8 desde la versión 3.6, junto con las paqueterías NuGet. Sin embargo, las únicas paqueterías que no se actualizaron fueron las asociadas al navegador Chromium (paqueterías Cef) y las relacionadas con las imágenes DICOM (paqueterías fo-Dicom), ya que eran incompatibles al actualizar.

Se inició entonces la integración de las nuevas características en un nuevo proyecto, tomando como base el proyecto anterior. Se presentaron nuevos diseños para la interfaz de usuario por parte del área de transferencia, con el apoyo de la diseñadora gráfica, quien se encargó de incorporar los colores institucionales y actualizar los iconos de los módulos para que estos correspondieran con los cambios realizados en el software y el hardware. Se decidió realizar este cambio con el objetivo de obtener una interfaz más intuitiva y fluida.

Se optó por continuar utilizando el lenguaje C# junto con Windows Forms, debido a la familiaridad con el lenguaje adquirida a lo largo del proyecto y a la compatibilidad existente con las tarjetas de captura de video, pues el SDK principal de estas tarjetas se encuentra principalmente en este lenguaje,

además de contar con las paqueterías necesarias para las imágenes DICOM, que también permiten la interacción con las alojadas en los servidores de los hospitales.



Figura 6.33: Pantallas renovadas de Información.



Figura 6.34: Pantallas renovadas de monitoreo de la pantalla de 38”.

Dentro de las actualizaciones realizadas en el hardware, se destaca el cambio de la cámara ambiental. Inicialmente, se utilizaba una cámara IP VGA que alcanzaba una resolución máxima de 480p a través de su aplicación de escritorio, sin posibilidad de acceder a la visualización desde la interfaz. Con la nueva cámara, se logra una resolución 4K y además cuenta con un SDK compatible con C# y Windows Forms, lo que facilitó la integración de sus

funciones al proyecto.

Las nuevas funcionalidades agregadas permiten ahora el control del encendido y el movimiento de la cámara. Además, se incorporó una sección en el panel de configuración para asociarla con la PC de control, lo que elimina la dependencia de una aplicación de terceros para visualizar y controlar la cámara.

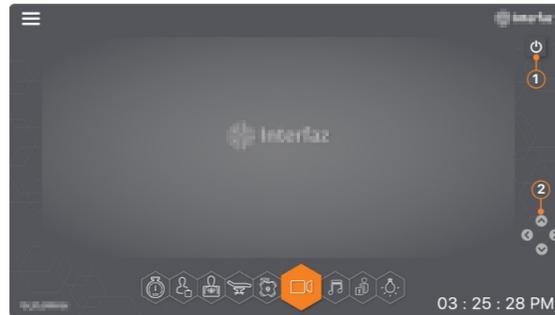


Figura 6.35: Pantalla de ejemplo de la cámara ambiental.

Una actualización que se integró en esta nueva versión consistió en nuevas tarjetas de control para los dispositivos (lámparas, luces y mesa), con el fin de efficientar y reducir los costos de instalación. Estas tarjetas se basaron en las tramas de comunicación con las que se había trabajado, por lo que debieron adaptarse al software y no al contrario. El cambio más significativo realizado en estas tarjetas fue en el modelo de las mesas, que dejaron de incluir receptores infrarrojos y se cambiaron por módulos xBee de radiofrecuencia. Ahora es necesario añadir un receptor en las mesas, el cual se empareja automáticamente con el transmisor que se conecta a la PC.

Como parte de la actualización del software, se inició creando una función en el menú desplegable que permite seleccionar entre dos configuraciones disponibles: avanzada e intermedia. Posteriormente, se integró la versión Lite, la cual habilita más o menos módulos según la sala de operaciones en la que se instala, lo que evita la necesidad de crear “versiones” del software para cada sala. En este mismo menú, se agregaron las configuraciones de los dispositivos asociados, lo que permite configurar las versiones de dispositivos instalados

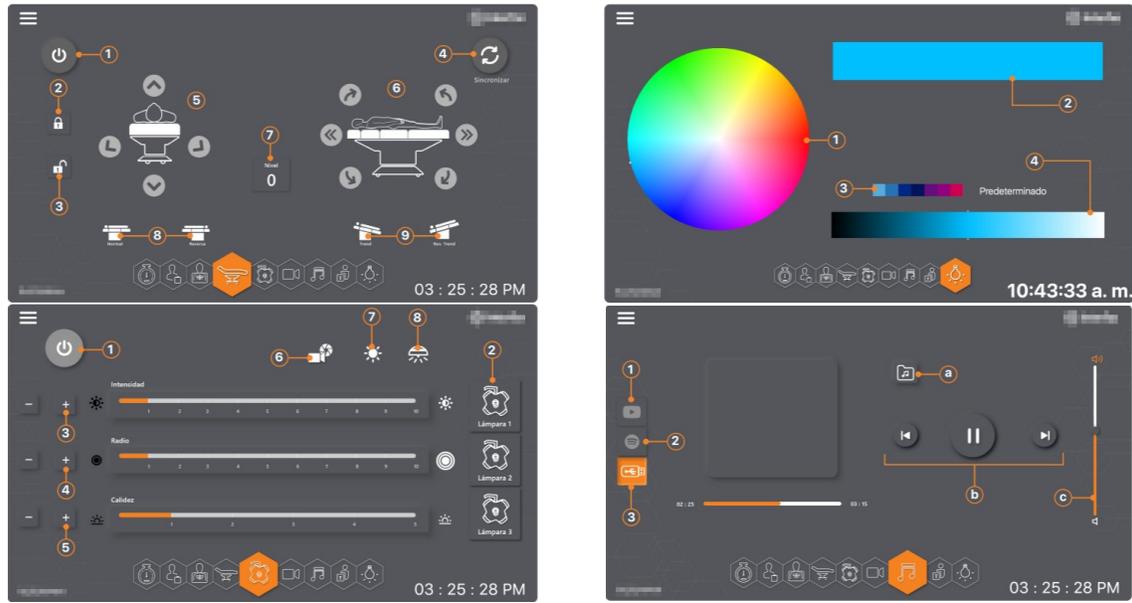


Figura 6.36: Pantallas nuevas de módulos de control.

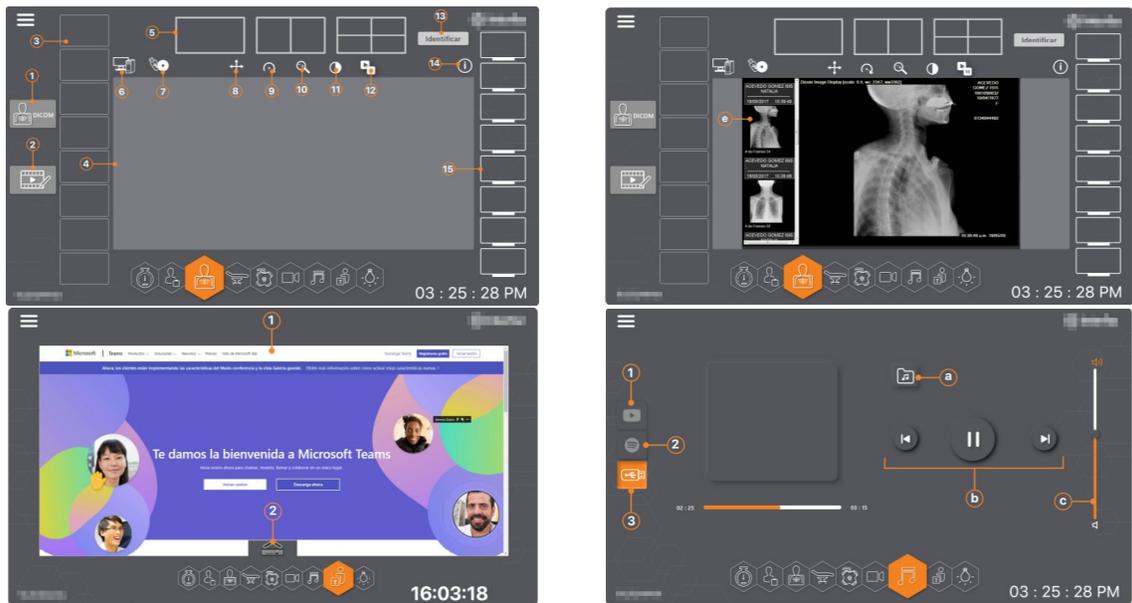


Figura 6.37: Pantallas nuevas de módulos audiovisuales.

para mesas y lámparas, así como la cantidad de cúpulas y pantallas a controlar. Además, como se explicó anteriormente, se incluyen los datos necesarios para conectarse a la cámara IP. Todos los módulos, además de integrar las actualizaciones que se fueron presentando, se rediseñaron con una nueva interfaz más intuitiva acorde al nuevo diseño.



Figura 6.38: Pantalla de ejemplo de la cámara ambiental.



Figura 6.39: Pantallas nuevas de menú de configuración.

7.

Resultados

Una vez completado el nuevo desarrollo del proyecto y tras realizar nuevamente las pruebas pertinentes en el laboratorio, se presentó a la gerencia para su aprobación. Posteriormente, se procedió a preparar una presentación para el equipo Comercial con el fin de familiarizarlos con las nuevas características. Al mismo tiempo, se inició la generación de las hojas de especificaciones para las nuevas PCs y dispositivos, para que los integrantes del equipo de Producto Entrante pudieran aprobar o rechazar los equipos.

Durante estas preparaciones, se llevó a cabo la presentación del producto en ExpoMédica en el WTC, donde se supervisó la instalación. Dado que la instalación se basaba en el proyecto anterior, fue similar en cuanto a los dispositivos, con la excepción de la configuración de la cámara, la cual, al ser un modelo nuevo, requería algunos ajustes adicionales.

Además, durante esta instalación, se identificaron las debilidades que presentaban los ingenieros en cuanto a las instalaciones. Por lo tanto, se agregó una sección de “Errores Frecuentes” en el manual, proporcionando una guía de soluciones para los problemas más comunes que podrían surgir durante la instalación.

Para finalizar el proyecto, se realizó la transferencia del mismo, asignando un número de producto a los nuevos equipos, dispositivos y a los manuales de servicio y de usuario. Así mismo, se llevaron a cabo presentaciones para capacitar a las áreas de Comercial, TI y Servicio, quienes estarán a cargo de las ventas, instalaciones y mantenimiento de los equipos. Por su parte, el equipo de Diseño se encargará de llevar el ciclo de vida de los productos de

control de Quirofano Inteligente y de monitoreo de signos vitales, así como de conceptualizar los nuevos proyectos futuros.

Con estos pasos, se logra el objetivo principal de tener el control del software, que inicialmente estaba a cargo de una empresa externa, lo que generaba costos excesivos para la empresa cada vez que se solicitaba una actualización o nuevos requisitos en la aplicación.

8.

Conclusiones

De acuerdo con los resultados de la exposición realizada en el WTC, se puede concluir que se consiguió descifrar el código de manera correcta para poder llevar a cabo una actualización de todas sus funciones, incluyendo de aquellos dispositivos que interactúan con la aplicación en los cuales se requirió de la participación en conjunto del personal de electrónica ya que ellos eran quienes tenían el control total de los dispositivos y los instrumentos para realizar las pruebas. Esta no fue la única colaboración con otras áreas ya que para el cambio de las pantallas de los módulos se involucró con Transferencia y Tecnología para el diseño gráfico así como también para las presentaciones y el desarrollo de los manuales.

Estas interacciones con otros equipos no relacionados con la ingeniería y en menor medida con la computación ayudó a que se ampliara la comunicación menos técnica y más coloquial para lograr un mejor entendimiento con una mayor cantidad de personas. Esto también facilitó al momento de realizar las presentaciones de capacitación con el personal involucrado en la comercialización e instalación al ser dos grupos con enfoques completamente distintos del mismo producto.

Pasando al análisis del código en este proyecto se utilizó por primera vez la metodología SCRUM para la gestión ya que era necesaria la supervisión constante de los cambios generados para que estos cumplan con los requisitos mínimos. Cada módulo fue necesario el entender su funcionamiento y generar un diagrama de flujo con el objetivo de identificar pasos redundantes y reducir procesos.

Este análisis puso a prueba la habilidad de comprensión hacia los códigos pertenecientes a otros desarrolladores y la capacidad para adaptarse a un entorno de trabajo desconocido, pues en la gran mayoría de proyectos estos ya están en proceso y se debe continuar con lo ya desarrollado por lo que difícilmente se puede iniciar un proyecto completamente desde cero. Así mismo en la realización de pruebas tanto individuales como en conjunto con las áreas involucradas, es necesario que haya una buena comunicación para que el proyecto pueda avanzar de manera correcta y se puedan corregir los fallos encontrados lo más rápido y de la mejor manera posible, ya que la comunicación entre dispositivos es una parte crucial de las aplicaciones.

Con todos los puntos abordados dentro de este trabajo es factible decir que se han cumplido los objetivos propuestos al lograr descifrar el código de una aplicación para su control y transferencia, creando sus manuales y capacitando al personal para su uso.

Bibliografía

- [, 2022] (2022). *Manual de Usuario Software Interfaz*. Biossmann, CDMX, v01 edition. Fecha de Acceso: 18 de Octubre de 2023.
- [AWS, 2023] AWS (2023). ¿en qué consiste scrum? *Amazon Web Services*. Fecha de Acceso: 14 de Octubre de 2023.
- [Biossmann, 2020] Biossmann (2020). Quirófano inteligente qx. *Biossmann*. Fecha de Acceso: 14 de Octubre de 2023.
- [Chacon and Straub, 2014] Chacon, S. and Straub, B. (2014). *Pro git: Everything you need to know about Git*. Apress, second edition.
- [Cué Palero, 2019] Cué Palero, J. A. (2019). *Diseño de un quirófano inteligente. Análisis comparativo de las tecnologías disponibles y necesarias para realizar un quirófano inteligente*. PhD thesis, Industriales.
- [De León, 2020] De León, H. (2020). Sql server para novatos - 9 - copiar una base de datos a otro lugar. <https://www.youtube.com/watch?v=aXkfyPF3nzQ>.
- [ETKHO, 2020] ETKHO (junio 25, 2020). Los quirófanos inteligentes. *ETKHO Hospital Engineering*. Fecha de Acceso: 14 de Octubre de 2023.
- [Méndez et al., 2010] Méndez, M., Garrido, A., Overbey, J., and Tinetti, F. G. (2010). Refactorización en código fortran heredado. In *XVI Congreso Argentino de Ciencias de la Computación*.
- [System, 2023] System, H. A. H. (2023). Quirófanos inteligentes: la evolución de la cirugía. Fecha de Acceso: 16 de Noviembre de 2023.

[Velasco et al., 2021] Velasco, M. V. E., Villacis, J. A. N., Chávez, P. R. S., and Cuchiye, W. C. C. (2021). Revisión sistemática de la metodología scrum para el desarrollo de software. *Dominio de las Ciencias*, 7(4):54.

[Villar Mir, 2019] Villar Mir, J. M. (2019). Los retos de la era digital. In *ANALES DE LA REAL ACADEMIA DE CIENCIAS MORALES Y POLÍTICAS*, pages 241–256. Ministerio de Justicia.