



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Departamento de Electronica e Informatica

"ENFOQUE EN ASEGURAMIENTO DE CALIDAD (QA) PARA PLATAFORMA DIGITAL DE PROYECTO DE GESTIÓN DE VISITAS HOSPITALARIAS"

Camilo Ignacio Thenoux Concha

TRABAJO DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO EN
INFORMÁTICA

Profesor Guia: Dagoberto Cabrera

Profesor Correferente: Diego Caceres



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: ENFOQUE EN ASEGURAMIENTO DE CALIDAD (QA) PARA PLATAFORMA DIGITAL DE PROYECTO DE GESTIÓN DE VISITAS HOSPITALARIAS

Nombre del candidato(a): Camilo Ignacio Thenoux Concha

Carrera / Grado: Ingeniería en Informática

Campus: Viña del mar

Departamento: Departamento de Electrónica e Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, DAGOBERTO CABRERA TAPIA, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 21/09/2026

Firma:

Estudiante o Candidato(a):

Fecha: 21/09/2026

Firma:



ÍNDICE:

ÍNDICE:	1
1 INTRODUCCIÓN:	3
1.1 CONTEXTO Y ANTECEDENTES:	3
1.2 DEFINICIÓN DEL PROBLEMA:	3
1.3 BREVE DESCRIPCIÓN DE PROPUESTA SOLUCIÓN:	3
1.4 OBJETIVO GENERAL Y OBJETIVOS ESPECÍFICOS:	4
1.5 METODOLOGÍA DE TRABAJO:	4
1.6 ORGANIZACIÓN DE CAPÍTULOS:	5
2 MARCO TEÓRICO:	6
2.1 EXPERIENCIA DE USUARIO (UX):	6
2.2 FRONT-END:	6
2.3 BACK-END EN EL DESARROLLO DE SISTEMAS DIGITALES:	6
2.4 ASEGURAMIENTO DE CALIDAD (QA) EN EL DESARROLLO DE SOFTWARE:	7
2.5 DIGITALIZACIÓN EN EL ÁMBITO HOSPITALARIO:	7
3 CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA:	8
3.1 CONTEXTO DE LA INSTITUCIÓN:	8
3.2 SITUACIÓN ACTUAL DEL PROCESO DE VISITAS:	8
3.3 ACTORES INVOLUCRADOS:	9
3.4 COMPETENCIA Y SOLUCIONES EXISTENTES:	9
3.5 ALCANCE DEL PROYECTO:	10
4 CAPÍTULO 2: MARCO CONCEPTUAL:	12
4.1 ASEGURAMIENTO DE CALIDAD DE SOFTWARE (SQA):	12
4.2 MODELOS DE CALIDAD DE SOFTWARE: ISO/IEC 25010:	14
4.3 QA VS PRUEBAS DE SOFTWARE:	16
4.4 ESTRATEGIAS Y NIVELES DE PRUEBA:	16
4.5 PRUEBAS DE USABILIDAD COMO PARTE DEL QA:	18
4.6 MÉTRICAS DE CALIDAD Y SYSTEM USABILITY SCALE:	21
4.7 QA EN SISTEMAS DE SALUD Y SALUD DIGITAL:	23
4.8 RELACIÓN DEL MARCO DE QA CON EL PROYECTO VISITACONNECT:	24
5 CAPÍTULO 3: PROPUESTA DE SOLUCIÓN:	27
5.1 VISIÓN GENERAL DE LA SOLUCIÓN (VISITACONNECT):	27
5.2 ALCANCE Y PERFILES DE USUARIOS CONSIDERADOS:	27
5.3 REQUISITOS Y CRITERIOS DE ACEPTACIÓN ORIENTADOS A CALIDAD:	28
5.4 ESTRATEGIA DE ASEGURAMIENTO DE CALIDAD APLICADA:	32
5.5 DISEÑO DE PRUEBAS: CASOS DE PRUEBA Y MÉTRICAS:	33
6 CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN:	49
7 CAPÍTULO 5: CONCLUSIONES:	50
7.1 CUMPLIMIENTO DE OBJETIVOS DEL PROYECTO:	50
7.2 PRINCIPALES RESULTADOS DESDE EL ENFOQUE DE QA:	50
7.3 VALIDACIÓN DE LA SOLUCIÓN EN UN CONTEXTO REALISTA:	51
7.4 ALCANCE Y LIMITACIONES:	51
7.5 RECOMENDACIONES:	51
8 REFERENCIAS:	53



Resumen: El presente trabajo aborda la problemática de la gestión manual de visitas en el Hospital Dr. Gustavo Fricke, la cual genera largas esperas, desinformación y sobrecarga administrativa. Frente a ello, se desarrolló VisitaConnect, una plataforma digital que busca modernizar y optimizar este proceso.

Esta propuesta fue concebida priorizando a sus usuarios finales. Gracias a las técnicas de Diseño UX, pudimos capturar las necesidades reales de los visitantes, del personal y de los médicos.

Con el fin de garantizar una experiencia de usuario óptima y un correcto funcionamiento de la plataforma, se llevaron a cabo diversas pruebas de compatibilidad y de peticiones mediante solicitudes (GET/POST). Al tratarse del desarrollo de un Producto Mínimo Viable (MVP), la validación se enfocó en la estabilidad del sistema, prescindiendo de pruebas de estrés masivas o simulaciones de alta concurrencia simultánea en esta etapa inicial.

Con esa información, definimos la estructura de interacción con el sistema y desarrollamos prototipos funcionales. La solución ofrece: reservas de horarios web, alertas automatizadas por email, un sistema de chequeo con QR y una interfaz administrativa completa para el manejo de accesos y permisos.

Los resultados muestran un sistema que facilita la autonomía de los visitantes, reduce la carga operativa del personal, mejora la trazabilidad del proceso. Además, la plataforma ayuda a fortalecer el control y la seguridad dentro del entorno hospitalario.

La importancia de este proyecto reside en mostrar cómo una herramienta tecnológica bien pensada puede mejorar la productividad, la disponibilidad y el enfoque humano en los cuidados médicos, creando beneficios reales tanto para la vivencia de las personas involucradas como para la administración del centro.

Palabras Clave: Gestión hospitalaria; Experiencia de usuario; Plataforma digital; Accesibilidad; Eficiencia

1 INTRODUCCIÓN

1.1 CONTEXTO Y ANTECEDENTES

Los hospitales son esenciales en el sistema de salud público de Chile, pero el constante flujo de personas ha revelado graves fallos en la administración de las visitas.

Esta ineficiencia no solo perjudica la experiencia de los usuarios, sino que también satura al personal administrativo, que gasta tiempo valioso en la repetitiva gestión manual de accesos y consultas. Con más de 2.5 millones de personas en lista de espera (Minsal, 2023), la tensión del sistema es palpable. Aunque se ha avanzado en la digitalización de la atención clínica, los procedimientos de visitas familiares siguen siendo anticuados y desorganizados, lo cual afecta negativamente la percepción general sobre la calidad hospitalaria.

1.2 DEFINICIÓN DEL PROBLEMA

El principal inconveniente que tiene el Hospital Dr. Gustavo Fricke en cuanto al manejo de acceso de visitantes es la persistencia de un proceso totalmente manual. Este método genera una serie de problemas que afectan negativamente a usuarios y funcionarios por igual.

En consecuencia, la vivencia de los visitantes resulta “negativa”, dado que se ven obligados a lidiar con procesos desordenados, reiterativos e ineficaces. Por su parte, el equipo administrativo se encuentra abrumado debido a la obligación de contestar las mismas preguntas repetidamente, lo cual demora el servicio y les obstaculiza dedicarse a sus responsabilidades principales.

Considerando la importancia del acompañamiento hospitalario, es fundamental que los familiares y cercanos puedan acceder a toda la información requerida de forma rápida y organizada. Sin la implementación de un sistema digital diseñado específicamente para la gestión de visitas, esta optimización es inconseguible. Continuar con el sistema actual sólo refuerza una percepción de caos y falta de modernización dentro del recinto hospitalario.

1.3 BREVE DESCRIPCIÓN DE PROPUESTA SOLUCIÓN

VisitaConnect surge como una propuesta de desarrollo para un hospital de alta demanda en la Región de Valparaíso, con el propósito de ofrecer un sistema de reserva de visitas simple, ordenado y eficiente tanto para familiares como para personas cercanas a los pacientes. El objetivo es agilizar la operación hospitalaria, disminuyendo las demoras y optimizando las tareas del equipo administrativo. La plataforma busca automatizar y suprimir los trámites innecesarios, liberando al personal de la gestión repetitiva.



Para asegurar la facilidad de uso, VisitaConnect funcionará adecuadamente en cualquier dispositivo (móvil, tableta o PC). Un lector de códigos QR será clave para acelerar y dar seguridad al proceso de verificación en la entrada.

El personal clínico y administrativo contará con un módulo centralizado para supervisar la información de pacientes, la ocupación y el registro de entradas. Esta herramienta permitirá, además, restringir el acceso bloqueando identificaciones (RUTs) si es necesario.

El servicio se enfoca, al comienzo, en visitantes mayores de edad en contextos estándar. Quedan fuera del alcance inicial aquellas situaciones que requieran protocolos especiales (como visitas a pacientes de alto riesgo). Esta restricción facilita un desarrollo y una puesta en marcha más segura y controlada, al enfocarse en un grupo de usuarios claramente definido.

1.4 OBJETIVO GENERAL Y OBJETIVOS ESPECÍFICOS

Objetivo General:

Crear, poner en marcha y probar la plataforma digital VisitaConnect para administrar las visitas hospitalarias. Esto se logrará aplicando metodologías de aseguramiento de calidad (QA) para garantizar que sea fácil de usar (usabilidad), accesible, rápida (eficiencia) y estable (confiabilidad) dentro del ambiente de un hospital.

Objetivos Específicos:

- Establecer los criterios de calidad del sistema conforme a estándares de usabilidad, accesibilidad, seguridad y eficiencia, adaptados a las necesidades y particularidades del contexto hospitalario.
- Elaborar y ejecutar evaluaciones de calidad (tanto funcionales como no funcionales) para confirmar el adecuado desempeño de los componentes de programación de visitas, alertas, verificación por QR y gestión administrativa.
- Medir la experiencia del usuario a través de pruebas de usabilidad con visitantes, personal administrativo y clínico, para detectar fallos, limitaciones y áreas de mejora.
- Analizar la confiabilidad y seguridad de la plataforma, considerando la gestión de accesos y la protección de datos sensibles de pacientes y visitantes.
- Sugerir modificaciones y refinamientos al modelo inicial, fundamentados en los hallazgos de las evaluaciones, con el propósito de mejorar el rendimiento, disminuir incidencias y lograr una edición definitiva sólida y efectiva.

1.5 METODOLOGÍA DE TRABAJO

Se empleó Scrum como marco ágil para planificar, construir y validar VisitaConnect de forma incremental. La meta fue generar valor rápidamente con un MVP, minimizar riesgos y



asegurar la calidad a través de ciclos cortos de desarrollo con feedback continuo (metodología QA-Driven).

Un enfoque impulsado por el QA garantiza que el producto no solo opere correctamente, sino que también satisfaga estándares de calidad verificables y provea una experiencia de usuario coherente y satisfactoria.

Fase I — MVP funcional (primer release)

- En la primera fase se desarrolló un MVP funcional, es decir, una versión inicial del sistema que incorporó los elementos básicos de la gestión de visitas. Aunque no contaba con un diseño refinado ni con todas las funcionalidades previstas, permitió validar la idea central del proyecto: la posibilidad de reservar visitas hospitalarias de manera digital y registrar su trazabilidad. Este primer bloque fue clave para obtener retroalimentación temprana, detectar falencias y comprobar la viabilidad de la solución.

Fase II — Plataforma casi completa (segundo release)

- La segunda fase se orientó a construir una plataforma casi completa, integrando la mayor parte de los módulos definidos en el backlog. Durante este ciclo se añadieron funciones críticas como el envío de confirmación por correo, el control administrativo más detallado y la generación de códigos QR, quedando pendiente únicamente la descarga de estos a través del correo electrónico y el refinamiento del diseño del front-end. Esta etapa marcó un avance sustancial hacia el producto final, acercando el prototipo a una versión de uso real dentro del hospital.

Fase III — Cierre y pulido al 100% (tercer release)

- La tercera y última etapa consiste en el refinamiento y fortalecimiento de la plataforma. Se culminaron las características pendientes, se mejoró el desempeño general, se potenció la seguridad y se unificó el diseño visual. Esta fase es la entrega final del proyecto, asegurando una plataforma estable, productiva y preparada para su implementación en el entorno real.

1.6 ORGANIZACIÓN DE CAPÍTULOS

En el siguiente capítulo abordaremos el marco teórico, donde explicaremos los conceptos a ver dentro de la tesina, para dar un mejor entendimiento de la misma

2 MARCO TEÓRICO

2.1 EXPERIENCIA DE USUARIO (UX)

Según Donald Norman, la usabilidad es necesaria, pero insuficiente; la experiencia completa incluye factores emocionales y contextuales que influyen directamente en la satisfacción final del usuario [1].

El diseño UX busca lograr interacciones útiles, usables, deseables y accesibles. Para esto, Peter Morville desarrolló el modelo Panal UX, el cual define siete dimensiones fundamentales: utilidad, usabilidad, deseabilidad, accesibilidad, credibilidad, encontrabilidad y valor [2].

La UX es vital en el contexto clínico, donde los usuarios se manejan bajo estrés o emergencia. Es imperativo que el diseño priorice la claridad, accesibilidad y empatía para minimizar el esfuerzo que requiere entender y operar el sistema.

El Diseño Centrado en el Usuario (DCU), especificado en la norma ISO 9241-210, exige que el desarrollo tecnológico se base en el conocimiento detallado de los usuarios, sus tareas y su entorno de trabajo [3]. Esta perspectiva es esencial para guiar la creación de perfiles y prototipos que respondan a las necesidades auténticas del hospital.

Por último, las evaluaciones de usabilidad son fundamentales en el control de calidad UX, ya que permiten detectar fallas, verificar la eficiencia, asegurar la accesibilidad y cuantificar el nivel de satisfacción [4].

2.2 FRONT-END

El front-end corresponde a la capa visual e interactiva entre el usuario y el sistema. Se construye mediante tecnologías como HTML5, CSS3 y JavaScript, junto con frameworks como React o Angular, que facilitan la creación de interfaces dinámicas y modulares [5].

En el sistema VisitaConnect, el front-end es esencial para la experiencia, ya que garantiza un diseño que es tanto accesible como adaptable a distintos dispositivos (responsive). El rendimiento y la accesibilidad de esta capa se pueden evaluar mediante herramientas especializadas como Lighthouse o PageSpeed Insights [6].

Además, la accesibilidad web se rige por las Web Content Accessibility Guidelines (WCAG 2.1), las cuales buscan garantizar que los sitios sean utilizables por personas con diversas capacidades, un requisito clave en contextos hospitalarios [7].

2.3 BACK-END EN EL DESARROLLO DE SISTEMAS DIGITALES

El back-end gestiona la lógica central del sistema, el manejo de bases de datos y la protección de la información. Para este desarrollo se implementó la arquitectura Modelo-Vista-Controlador (MVC) que separa la lógica de negocio de la interfaz de usuario, optimizando la escalabilidad y el mantenimiento [8].

El stack tecnológico contempla opciones como .NET, Node.js, Python (Django) o Java Spring, junto con bases de datos relacionales del tipo SQL Server o MySQL.

En el contexto hospitalario, el tratamiento de datos sensibles requiere cumplimiento obligatorio con la Ley N°19.628 chilena de protección de privacidad y con el estándar internacional ISO/IEC 27001, que establecen los protocolos de seguridad para la custodia de información personal [9][10].

El proceso de control de calidad del back-end incorpora pruebas unitarias, de integración y de seguridad, asegurando así la confiabilidad y el rendimiento del sistema [11].

2.4 ASEGURAMIENTO DE CALIDAD (QA) EN EL DESARROLLO DE SOFTWARE

El QA comprende prácticas estructuradas para asegurar la conformidad del software con los estándares. A diferencia del QC, se caracteriza por su enfoque preventivo ante posibles errores [12].

En entornos ágiles como Scrum, la calidad se valida iterativamente en cada ciclo (sprint). El proceso incluye la ejecución de pruebas automatizadas, la revisión de código y la incorporación del feedback del usuario, siguiendo el modelo de QA-Driven Development [13].

Los marcos normativos fundamentales que rigen los procesos de calidad son:

- ISO/IEC 25010: Especifica características de calidad como usabilidad, rendimiento, seguridad y mantenibilidad
- ISO 9001: Define los requisitos para sistemas de gestión de calidad organizacional
- IEEE 829: Establece los lineamientos para documentación de pruebas y procedimientos formales [14][15][16].

Estas normas sirven como guía para validar los módulos desarrollados en VisitaConnect, incluyendo reserva, validación QR y gestión administrativa.

2.5 DIGITALIZACIÓN EN EL ÁMBITO HOSPITALARIO

La modernización de los procesos hospitalarios representa una tendencia fundamental en la actualización de los sistemas de salud públicos. En Chile, el programa Hospital Digital del Ministerio de Salud, tiene el fin de mejorar la trazabilidad y reducir los tiempos de espera mediante herramientas tecnológicas [17].

El Centro Nacional en Sistemas de Información en Salud (CENS) refuerza esta visión, destaca que la interoperabilidad y las plataformas digitales son elementos vitales para fortalecer tanto la eficiencia como la seguridad en la provisión de servicios de salud [18].

VisitaConnect se enmarca dentro de esta estrategia de modernización digital, aportando una solución que no solo optimiza la experiencia de los visitantes, sino que también mejora la gestión administrativa de los hospitales [19].

3 CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

3.1 CONTEXTO DE LA INSTITUCIÓN

El Hospital Dr. Gustavo Fricke es un hospital público de alta complejidad de la Región de Valparaíso, que recibe diariamente un alto flujo de pacientes, acompañantes y visitas. Como establecimiento de salud, debe compatibilizar la necesidad de entregar un entorno seguro y controlado con el derecho de los pacientes a recibir contención familiar durante su hospitalización.

En este contexto, la gestión de visitas se vuelve un proceso clave: no solo impacta en la experiencia de los pacientes y sus familias, sino también en la carga de trabajo del personal administrativo y de seguridad, y en el cumplimiento de normativas internas relacionadas con aforo, horarios de visita y trazabilidad de las personas que ingresan al recinto.

3.2 SITUACIÓN ACTUAL DEL PROCESO DE VISITAS

Actualmente, el proceso de registro y control de visitas en el hospital se realiza de forma manual. De manera general, el flujo funciona de la siguiente manera:

1. La persona llega al mesón central del hospital y solicita realizar una visita.
2. En el mesón se verifica la información del paciente y se registran los datos del visitante de forma manual (por ejemplo, en planillas o listados).
3. Luego, el visitante debe dirigirse al guardia de seguridad, quien autoriza el ingreso y permite el paso hacia los ascensores.
4. El visitante sube hasta el piso correspondiente y, al llegar, debe pasar nuevamente por el mesón del piso para que se registre su ingreso a la unidad.
5. Finalmente, y solo después de estos pasos, la persona puede realizar la visita al paciente.

Este proceso presenta diversas limitaciones:

- Múltiples puntos de control y registro para una misma visita, lo que genera demoras y tiempos de espera excesivos.
- Dependencia de registros manuales que pueden contener errores, información incompleta o ilegible.
- Falta de trazabilidad centralizada de las visitas (es difícil obtener una vista consolidada de quién ingresó, a qué hora y a qué paciente visitó).



- Dificultad para controlar el aforo y el cumplimiento de los horarios de visita, especialmente en horas punta.
- Carga administrativa adicional para el personal de mesones y guardias, que deben responder consultas, verificar datos y registrar información una y otra vez.

En resumen, el proceso actual es poco eficiente, genera una experiencia confusa para los visitantes y no aprovecha las posibilidades de digitalización para simplificar y ordenar la gestión de visitas.

3.3 ACTORES INVOLUCRADOS

En el problema identificado intervienen diversos perfiles de usuario:

- Visitantes: familiares o amigos de pacientes hospitalizados que necesitan agendar o realizar una visita. Son quienes viven directamente la experiencia de espera, controles y registros manuales.
- Pacientes: personas hospitalizadas que dependen del sistema de visitas para recibir acompañamiento y apoyo emocional de sus cercanos.
- Personal de mesón central: funcionarios administrativos encargados de verificar la información del paciente, registrar datos de visitantes y entregar indicaciones.
- Personal de mesones de piso o unidades clínicas: funcionarios que vuelven a registrar el ingreso del visitante a la unidad correspondiente y responden dudas sobre horarios y normas.
- Guardias de seguridad: responsables de permitir o denegar el acceso, controlar el flujo de personas y velar por la seguridad del recinto.
- Personal clínico (enfermería, técnicos, médicos): si bien no gestionan directamente el registro, se ven afectados por el flujo de visitas (por ejemplo, cuando hay demasiadas personas en una sala o cuando no se respetan horarios).
- Área de gestión/soporte tecnológico: potencialmente involucrada en la futura implementación y mantención de una solución digital.

Cualquier propuesta de mejora debe considerar las necesidades, limitaciones y nivel de alfabetización digital de todos estos actores.

3.4 COMPETENCIA Y SOLUCIONES EXISTENTES

A nivel general, existen distintas soluciones digitales para la gestión de visitas y control de acceso en instituciones, como:

- Sistemas genéricos de control de acceso con credenciales o tarjetas.
- Plataformas de gestión de visitas corporativas que permiten agendar y registrar ingresos en empresas u oficinas.
- Formularios en línea o herramientas básicas (por ejemplo, planillas compartidas) para preinscripción de visitas.

Sin embargo, estas soluciones presentan varias limitaciones cuando se intenta aplicarlas directamente a la realidad de un hospital público como el Hospital Dr. Gustavo Fricke:

- No están diseñadas específicamente para el contexto hospitalario, donde existen normas estrictas de aforo, horarios de visita y particularidades por unidad (UCI, pediatría, etc.).
- Muchas son soluciones privativas o de alto costo, difícilmente adaptables al presupuesto y a la infraestructura tecnológica de un hospital público.
- No consideran de forma explícita la experiencia de usuario para distintos perfiles (visitantes, guardias, administrativos, personal clínico), lo que puede generar interfaces complejas o poco intuitivas.
- En algunos casos, la solución se limita a un solo punto del proceso (por ejemplo, el ingreso en portería), sin ofrecer una visión integrada desde la reserva de la visita hasta el registro en la unidad clínica.

Por otro lado, en otras realidades —como clínicas privadas o instituciones en el extranjero— este problema se ha abordado mediante portales web o aplicaciones móviles que permiten agendar visitas con anticipación, generar códigos QR para el ingreso y centralizar el registro. No obstante, esas soluciones no siempre son directamente transferibles al contexto local, ya sea por idioma, normativa, costos o porque no responden a las necesidades específicas del hospital y de sus usuarios.

En este escenario, se hace necesario desarrollar una solución adaptada al contexto del Hospital Dr. Gustavo Fricke, que ponga el foco en la usabilidad, en la claridad de los flujos y en la reducción de la carga operativa del personal.

3.5 ALCANCE DEL PROYECTO

El alcance del Trabajo de Título, desde la perspectiva del Aseguramiento de Calidad (QA), se centra en:

- La validación y verificación del prototipo web funcional: Diseñar y ejecutar estrategias de pruebas enfocadas en un modelo funcional y no en un sistema productivo definitivo.



- La cobertura de pruebas para los flujos principales de:
 - Solicitud y reserva de visitas (peticiones y respuestas del sistema).
 - Generación, lectura y correcta validación de códigos QR.
 - Visualización, gestión de accesos y consistencia de datos en las interfaces de perfiles internos (guardias y administrativos).
- El control de calidad a nivel de prototipo aislado: Evaluar el comportamiento de la plataforma de forma independiente, prescindiendo de pruebas de integración con sistemas hospitalarios reales existentes (clínicos o de gestión interna), orientando los esfuerzos a dejar lineamientos de QA para una futura implementación.
- Evaluación de la usabilidad y consistencia del proceso: Asegurar que los criterios de aceptación cumplan con el enfoque en la experiencia de usuario y la optimización del flujo actual, priorizando esto por sobre la evaluación de una infraestructura tecnológica compleja.

Fuera del alcance del QA: No se contempla dentro de este trabajo el aseguramiento de calidad en un entorno de producción real, pruebas de rendimiento ante escenarios de estrés masivo, ni la certificación de integraciones con sistemas corporativos del hospital. El límite de la responsabilidad técnica radica en la entrega de una solución validada a nivel de prototipo, estableciendo las bases de calidad para su eventual desarrollo futuro

4 CAPÍTULO 2: MARCO CONCEPTUAL

4.1 ASEGURAMIENTO DE CALIDAD DE SOFTWARE (SQA)

El Aseguramiento de Calidad de Software (Software Quality Assurance, SQA) corresponde al conjunto de actividades planificadas, sistemáticas y verificables orientadas a proporcionar confianza de que los requisitos de calidad serán cumplidos, tanto en el proceso de desarrollo como en el producto final. En términos de gestión de calidad, QA se entiende como la parte enfocada en “dar confianza” del cumplimiento de requisitos, diferenciándose del Control de Calidad (QC), que se centra en detectar defectos sobre entregables ya construidos [20].

A diferencia de considerar la calidad como una etapa puntual al final del desarrollo, el SQA se concibe como una práctica transversal que acompaña todo el ciclo de vida del software, desde la definición y validación de requisitos hasta el mantenimiento. Esto se alinea con marcos normativos que describen procesos de ciclo de vida y promueven su control y mejora continua [21]. En este enfoque, el SQA tiene un carácter principalmente preventivo: busca reducir la ocurrencia de defectos mediante estándares, revisiones, criterios de aceptación, seguimiento y medición, utilizando pruebas como una práctica relevante, pero no única, dentro del aseguramiento.

Para evitar definiciones ambiguas de “calidad”, es recomendable apoyarse en modelos que definan características evaluables. La norma ISO/IEC 25010 define un modelo de calidad de producto con características y subcaracterísticas que permiten especificar, medir y evaluar la calidad del sistema [22]. Esto es especialmente útil cuando el software es usado por distintos perfiles y en contextos operacionales exigentes.

En un sistema como VisitaConnect, donde existe interacción con usuarios y operación en contexto hospitalario, la calidad no se limita al funcionamiento funcional, sino que suele priorizar atributos tales como usabilidad, confiabilidad y seguridad, entre otros, para asegurar que el sistema sea comprensible, consistente y apropiado para su contexto de uso [22].

El SQA se implementa mediante prácticas que cubren el proceso y el producto. Entre las más habituales se encuentran:

- Planificación de calidad: definición de objetivos, alcance, roles, artefactos y criterios de aceptación/salida.
- Definición de estándares y guías: reglas de documentación, convenciones de interfaz, criterios mínimos de usabilidad y lineamientos de seguridad.
- Revisiones y auditorías de artefactos: inspección de requisitos, flujos, prototipos, casos de prueba y evidencias, para detectar inconsistencias tempranas.

- Trazabilidad de requisitos: relación requisito → criterio de aceptación → caso de prueba → resultado.
- Verificación y validación: ejecución de pruebas funcionales, de regresión, y de usabilidad; registro de defectos, severidad y acciones correctivas.
- Métricas y monitoreo: indicadores para observar tendencias y respaldar decisiones.

Estas prácticas se encuentran descritas y formalizadas en estándares de procesos SQA como IEEE 730, el cual establece requisitos para iniciar, planificar, controlar y ejecutar procesos de aseguramiento de calidad en proyectos de software [23].

En VisitaConnect, el SQA se traduce en convertir la "calidad" en criterios concretos y verificables, alineados con necesidades del entorno hospitalario. En forma aplicada, se expresa en:

1. Definir criterios de calidad del producto (qué significa "calidad" para VisitaConnect):
 - Usabilidad (flujo claro, facilidad de aprendizaje, minimización de errores).
 - Confiabilidad (comportamiento consistente y tolerancia a fallos).
 - Seguridad (control de acceso, permisos y protección de datos).
Basándose en un modelo de calidad como ISO/IEC 25010 [22]
2. Prevención y control durante el desarrollo:
 - Criterios de aceptación por funcionalidad.
 - Revisiones de requisitos y prototipos para reducir ambigüedad.
 - Trazabilidad entre requisitos, diseño y pruebas.
3. Evidencias de calidad mediante pruebas y métricas:
 - Pruebas funcionales (incluyendo casos de error).
 - Pruebas de usabilidad (tasa de finalización por tarea, errores por tarea y observaciones).
 - Medición de usabilidad percibida con System Usability Scale (SUS), una escala de 10 ítems ampliamente utilizada para obtener una evaluación global de usabilidad [24].

4.2 MODELOS DE CALIDAD DE SOFTWARE: ISO/IEC 25010

Para estructurar de manera objetiva qué se entiende por “calidad” en un producto de software, es habitual utilizar modelos de calidad que permitan definir, descomponer y evaluar el producto mediante características y subcaracterísticas. En la familia SQuaRE, la norma ISO/IEC 25010 propone un modelo de calidad de producto que funciona como referencia para especificar requisitos de calidad y orientar su evaluación [14].

En su edición 2011, ISO/IEC 25010 organiza la calidad de producto en ocho características: adecuación funcional, eficiencia de desempeño, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad y portabilidad [14]. Estas características permiten pasar desde una noción general (“el sistema debe ser bueno”) a criterios concretos (“el sistema debe ser usable, seguro, confiable, etc.”), facilitando la comunicación entre interesados y la trazabilidad de los requisitos de calidad durante el ciclo de vida.

Es importante señalar que existe una actualización del estándar (ISO/IEC 25010:2023) que revisa el modelo de calidad de producto y mantiene el propósito de servir como marco de referencia para especificar y evaluar calidad [22]. En este trabajo, el modelo se utiliza principalmente como marco conceptual y de justificación, alineando los atributos de calidad priorizados con las necesidades del dominio hospitalario y con las métricas aplicadas en la evaluación del prototipo.

Dentro de ISO/IEC 25010, la usabilidad es especialmente relevante en sistemas interactivos donde participan usuarios con distintos niveles de experiencia digital y con restricciones de tiempo o carga operativa. La norma descompone la usabilidad en las siguientes subcaracterísticas [14]:

- Capacidad de reconocimiento de la adecuación (appropriateness recognizability): grado en que el usuario puede reconocer si el sistema es apropiado para su necesidad.
- Aprendibilidad (learnability): facilidad con la que el usuario aprende a utilizar el sistema.
- Operabilidad (operability): facilidad para operar y controlar la interfaz durante el uso.
- Protección frente a errores del usuario (user error protection): capacidad del sistema para prevenir errores o mitigar sus consecuencias.
- Estética de la interfaz de usuario (user interface aesthetics): claridad visual y presentación consistente que favorece la comprensión.
- Accesibilidad (accessibility): posibilidad de uso por personas con diversas capacidades y condiciones de uso.

Estas subcaracterísticas permiten traducir “que sea fácil de usar” en aspectos observables y medibles, lo que resulta útil para planificar pruebas de usabilidad, definir criterios de aceptación y establecer mejoras iterativas del diseño.

En VisitaConnect, ISO/IEC 25010 se utiliza como marco para justificar y delimitar los atributos de calidad más relevantes del prototipo, considerando que la plataforma es utilizada por perfiles heterogéneos (por ejemplo, visitantes, guardias y personal administrativo) y en un entorno que exige minimizar errores y tiempos de atención.

De forma aplicada, el modelo aporta en:

1. Justificación del foco en usabilidad

La diversidad de usuarios y contextos de uso (primer uso, alta demanda, necesidad de rapidez) vuelve crítico asegurar:

- Aprendibilidad (primer uso sin capacitación extensa),
- Operabilidad (flujo simple y controlable),
- Protección frente a errores (validaciones y prevención de acciones incorrectas),
- Accesibilidad (uso desde distintos dispositivos y condiciones),
- Reconocimiento de adecuación (que el usuario entienda rápidamente “para qué sirve” y “qué hacer”).

Esto permite argumentar que la usabilidad no es un “extra”, sino una condición de calidad del producto [14].

2. Definición de criterios de evaluación del prototipo

A partir de las subcaracterísticas de usabilidad, se derivan criterios observables para evaluar el prototipo, por ejemplo: claridad de flujos, facilidad de aprendizaje, reducción de errores por validaciones, y consistencia visual. Complementariamente, el modelo permite mantener en vista otros atributos relevantes para el dominio (seguridad y confiabilidad), aun cuando la evaluación se concentre principalmente en usabilidad [14], [22].

3. Vinculación con métricas y evidencias

El marco ISO/IEC 25010 ayuda a alinear instrumentos de evaluación (por ejemplo, tareas de prueba, registro de errores, observaciones) con atributos de calidad definidos. En particular, la medición de usabilidad percibida mediante SUS se utiliza como evidencia cuantitativa complementaria, reforzando la evaluación de la calidad desde el punto de vista del usuario [24].

4.3 QA VS PRUEBAS DE SOFTWARE

En el ámbito del desarrollo de software, es común que los términos Aseguramiento de Calidad (QA) y pruebas de software (testing) se utilicen de manera intercambiable. No obstante, corresponden a conceptos distintos y complementarios. Desde la perspectiva de gestión de calidad, QA se define como la parte de la gestión de calidad orientada a proporcionar confianza en que los requisitos de calidad serán cumplidos [25]. Esta definición refuerza que QA no se limita a ejecutar pruebas, sino que se relaciona con prácticas planificadas que permiten sustentar dicha “confianza” mediante evidencia.

Por su parte, las pruebas de software constituyen una actividad centrada en el producto, cuyo propósito es evaluar el comportamiento del sistema y detectar desviaciones respecto de lo esperado. En este sentido, el testing puede entenderse como un conjunto de procesos, actividades y documentación que guían la planificación, diseño, ejecución y reporte de resultados de prueba. La serie ISO/IEC/IEEE 29119 establece un marco internacional para la gestión y realización de pruebas de software, definiendo conceptos y lineamientos aplicables a distintos tipos de proyectos y organizaciones [27].

Una forma práctica de diferenciar ambos enfoques es considerar su orientación principal:

- QA (aseguramiento): se enfoca en el proceso y en cómo se trabaja para aumentar la probabilidad de cumplir los requisitos de calidad; por ejemplo, mediante criterios, revisiones, estándares, trazabilidad y control de cambios [26].
- Testing (pruebas): se enfoca en el producto y en generar evidencia mediante la evaluación y verificación del sistema en ejecución, reportando defectos y resultados [27].

Relacionado con lo anterior, la práctica de Verificación y Validación (V&V) aporta un marco complementario: los procesos de V&V permiten determinar si los productos de cada etapa cumplen sus requisitos y si el sistema satisface su uso previsto y necesidades de usuario. Este enfoque está formalizado en estándares como IEEE 1012, que describe requerimientos de V&V a lo largo del ciclo de vida y según niveles de integridad [28].

En el proyecto VisitaConnect, concentra principalmente componentes visibles de evaluación (planificación de pruebas, casos de prueba, pruebas con usuarios y métricas de calidad). Sin embargo, conceptualmente se adopta QA como un enfoque transversal: las decisiones de diseño, definición de criterios de calidad y evaluación del prototipo se entienden como acciones coordinadas para respaldar con evidencia el cumplimiento de requisitos, y no únicamente como una etapa final de “detección de fallos”.

4.4 ESTRATEGIAS Y NIVELES DE PRUEBA

Dentro del aseguramiento de calidad, las pruebas de software se organizan mediante niveles y estrategias que permiten planificar qué se probará, en qué etapa y con qué profundidad, de modo de obtener evidencia suficiente del cumplimiento de requisitos. La serie ISO/IEC/IEEE 29119 proporciona un marco internacional para el testing, estableciendo

conceptos generales y procesos para gobernar, gestionar e implementar actividades de prueba en distintos contextos y modelos de desarrollo [29], [30].

Los niveles de prueba (también llamados *test levels* o *test stages*) agrupan actividades de testing que se gestionan en conjunto y que se asocian con responsabilidades específicas dentro de un proyecto. En términos generales, se distinguen cuatro niveles principales: pruebas de unidad (o componente), integración, sistema y aceptación [32].

- Pruebas de unidad (componente): se enfocan en validar partes individuales del sistema (módulos, componentes o funciones) de manera aislada.
- Pruebas de integración: se centran en verificar las interacciones entre componentes o sistemas ya integrados, identificando fallas de comunicación, contratos de interfaz o dependencias [32].
- Pruebas de sistema: evalúan el sistema integrado como un todo para verificar que cumple los requisitos especificados [32].
- Pruebas de aceptación: se orientan a determinar si el sistema satisface criterios de aceptación, necesidades de usuario y procesos de negocio, permitiendo decidir si se acepta o no la solución [32].

En proyectos como VisitaConnect, donde el objetivo central es validar un prototipo funcional orientado a experiencia de usuario, los niveles con mayor relevancia práctica suelen ser:

- Pruebas de sistema, porque permiten validar el comportamiento end-to-end del prototipo (flujos completos y reglas principales) en relación con requisitos funcionales y restricciones del contexto de uso.
- Pruebas de aceptación, porque conectan el sistema con la expectativa real del entorno (usuarios o representantes), determinando si el prototipo "resuelve el problema" desde el punto de vista operativo y del usuario.

Además del nivel, las pruebas pueden clasificarse según su objetivo principal:

- Pruebas funcionales: verifican que el sistema implementa correctamente las funciones definidas por los requisitos (qué hace el sistema).
- Pruebas no funcionales: evalúan requisitos y atributos de calidad que no describen funciones específicas, sino cómo se comporta el sistema (por ejemplo, rendimiento, confiabilidad, usabilidad, mantenibilidad, etc.) [32].

En VisitaConnect, esta distinción permite estructurar la evaluación de manera coherente: por un lado, se valida que los flujos críticos funcionen; por otro, se evalúa la calidad del uso (por ejemplo, usabilidad) mediante métricas e instrumentos complementarios.

Una estrategia frecuente para definir el alcance y prioridad de pruebas es el risk-based testing, donde la selección y priorización de actividades de prueba se guía por el nivel de riesgo (impacto y probabilidad) asociado a funcionalidades o atributos de calidad [32]. De forma consistente, ISO/IEC/IEEE 29119-2 plantea el uso de un enfoque basado en riesgos para orientar la planificación y focalizar esfuerzos en los elementos más relevantes del sistema [30].

Aplicado a VisitaConnect, esto implica priorizar la validación de aquellos procesos que, por criticidad operativa, podrían generar mayor impacto si fallan o si resultan confusos para el usuario. Por ejemplo:

- Reserva/solicitud de visita (inicio del flujo y punto de mayor frecuencia).
- Validación de QR / confirmación de visita (instancia crítica de control y autorización).
- Visualización y gestión de visitas (consulta rápida, trazabilidad y apoyo a la operación).

Considerando lo anterior, el énfasis de pruebas para la tesina se estructura en:

1. Validación funcional mediante pruebas de sistema

Se diseñan y ejecutan casos de prueba sobre los flujos críticos (reserva, validación QR y visualización/gestión), incluyendo escenarios normales y casos de error (datos inválidos, estados no permitidos, duplicidad, etc.). Esta aproximación se alinea con la lógica de procesos de testing descrita en ISO/IEC/IEEE 29119-2 (planificación, diseño, ejecución y reporte) [30].

2. Validación con usuarios como prueba de aceptación

Se complementa lo funcional con evaluación mediante usuarios o representantes del contexto, para determinar si el prototipo cumple expectativas y criterios de aceptación (comprensión, fluidez del flujo, reducción de errores y adecuación al contexto). Esta idea se vincula directamente con la definición de *acceptance testing* en glosarios y marcos de testing [32].

3. Evaluación no funcional con foco en usabilidad

Dado que el propósito del prototipo es mejorar experiencia y eficiencia del proceso de registro/gestión, se incorporan pruebas no funcionales centradas en usabilidad, registrando métricas (tasa de finalización por tarea, errores por tarea y observaciones) y utilizando instrumentos estandarizados (como SUS, desarrollado en el punto correspondiente).

4.5 PRUEBAS DE USABILIDAD COMO PARTE DEL QA

Las pruebas de usabilidad son una técnica de evaluación en la que usuarios representativos intentan completar tareas típicas con un sistema (o prototipo) mientras un evaluador observa y registra su desempeño, dificultades y percepciones. Este tipo de evaluación

permite recopilar evidencia sobre si el diseño apoya o dificulta el logro de objetivos en un contexto de uso específico, entendiendo la usabilidad como el grado en que un producto puede ser usado para alcanzar metas con efectividad, eficiencia y satisfacción [33].

Desde el enfoque de diseño centrado en el usuario, la evaluación con usuarios es un mecanismo fundamental para reducir el riesgo de construir una solución que no se ajuste a necesidades reales: el diseño se refina de forma iterativa a partir de los hallazgos obtenidos al observar a personas reales usando el sistema en escenarios realistas [3]. En términos de QA, esto implica que la usabilidad se trata como un atributo de calidad que debe ser planificado, medido y mejorado, no sólo “revisado al final”.

Una prueba de usabilidad se construye sobre escenarios y tareas que reflejan actividades reales del usuario y permiten evaluar el flujo de principio a fin. En el ámbito práctico, guías de referencia en usabilidad destacan que las tareas deben estar alineadas con objetivos del usuario, ser claras, y permitir observar comportamientos como errores, confusión, retrocesos y tiempos de ejecución [35], [36].

Además, las pruebas pueden ser:

- Moderadas (moderated): un facilitador guía la sesión, plantea tareas y realiza preguntas de seguimiento (por ejemplo, aclarar qué entendió el usuario o por qué tomó cierta decisión).
- No moderadas (unmoderated): el usuario realiza las tareas de forma autónoma, normalmente mediante una plataforma o guión, y el evaluador analiza registros (video, clics, tiempos, etc.).

Esta distinción se usa ampliamente porque impacta el tipo de datos que se obtiene y el control del entorno de evaluación [35].

Las pruebas de usabilidad permiten combinar evidencia cuantitativa y cualitativa, lo cual es especialmente valioso en QA cuando se busca justificar cambios de diseño con resultados observables:

- Cuantitativos:
 - Tasa de éxito por tarea (si el usuario completó o no el objetivo).
 - Número y tipo de errores (por tarea o por flujo).
 - Tiempo de realización (cuando corresponde y el contexto lo permite).
Estas métricas se utilizan comúnmente para describir desempeño y eficiencia del diseño [33], [35].
- Cualitativos:

- Comentarios verbales, dudas, frustraciones, expectativas, sugerencias.
 - Observaciones del evaluador sobre puntos de confusión, decisiones erróneas, lenguaje ambiguo o elementos de interfaz poco visibles.
-
- La evidencia cualitativa permite explicar el “por qué” detrás de los resultados numéricos y orientar acciones correctivas concretas [36].

Como complemento estandarizado, puede aplicarse el System Usability Scale (SUS) para obtener una medida global de usabilidad percibida y facilitar comparaciones entre iteraciones [24].

En evaluación formativa (mejora del diseño), se ha difundido el criterio de que grupos pequeños por iteración pueden descubrir la mayoría de problemas relevantes, especialmente si se realizan varias rondas de prueba. Jakob Nielsen propone que, en muchos casos, probar con alrededor de 5 usuarios por iteración permite detectar una parte significativa de problemas y que es más eficiente invertir en múltiples ciclos que en una sola prueba grande [37]. En trabajos posteriores, el mismo autor enfatiza el valor de la evaluación con un tamaño de muestra (n) reducido por ronda como práctica común, destacando que este depende del objetivo (formativo vs. cuantitativo/sumativo) [38]. Para estudios con propósito cuantitativo (por ejemplo, estimar métricas con mayor precisión), se recomiendan muestras mayores [39].

En VisitaConnect, las pruebas de usabilidad forman parte del plan de QA porque permiten:

1. Validar comprensión y desempeño por perfil
Determinar si perfiles heterogéneos (visitante, guardia, administrativo, personal clínico) pueden completar tareas clave sin ayuda excesiva, observando efectividad, eficiencia y satisfacción en el contexto de uso [33].
2. Detectar problemas de diseño antes de escalar
Identificar flujos confusos, textos ambiguos, validaciones insuficientes o pasos innecesarios, reduciendo el riesgo de construir una solución a gran escala con problemas de interacción persistentes [3], [36].
3. Recolectar evidencia para justificar mejoras
Integrar resultados cuantitativos (éxito por tarea, errores, SUS) y cualitativos (citaciones textuales y observaciones), respaldando decisiones de rediseño con evidencia y manteniendo trazabilidad entre hallazgos y cambios aplicados [35], [24].

4.6 MÉTRICAS DE CALIDAD Y SYSTEM USABILITY SCALE

Las métricas permiten cuantificar la calidad del producto y comparar versiones o alternativas de diseño, aportando evidencia objetiva para la toma de decisiones. En evaluación de UX/usabilidad, estas métricas se alinean con el enfoque de *quality in use*, donde se busca medir resultados del uso en términos de efectividad, eficiencia y satisfacción en un contexto específico [40].

En pruebas de usabilidad y evaluación de interacción, es común emplear métricas de desempeño (*performance metrics*) y percepción (*self-reported metrics*), entre ellas [41]:

- Tasa de finalización de tareas (Task success / completion rate): porcentaje de usuarios que logra completar una tarea (con o sin ayuda, según el criterio definido). Esta métrica se utiliza como indicador directo de efectividad del diseño, al reflejar si el usuario pudo alcanzar el objetivo propuesto [41].
- Número de errores por tarea: cantidad de acciones equivocadas o decisiones incorrectas durante la ejecución (p. ej., seleccionar una fecha inválida, confundir botones, omitir un paso). Permite identificar puntos críticos del flujo y evaluar si el sistema previene o reduce errores [41].
- Tiempo para completar la tarea (Time-on-task): duración requerida para lograr el objetivo. Se usa como indicador de eficiencia, especialmente útil cuando las tareas representan procesos reales y el contexto permite medir tiempos sin distorsión por aprendizaje o intervención del evaluador [41].
- Métricas subjetivas: escalas de percepción (satisfacción, facilidad, confianza, carga percibida). Estas métricas capturan la dimensión subjetiva de la experiencia y permiten complementar el desempeño observado con la percepción del usuario [40], [41].

En conjunto, estas medidas facilitan explicar no solo “si funciona”, sino qué tan bien funciona para el usuario y en qué puntos del flujo aparecen fricciones.

El System Usability Scale (SUS) es una de las métricas subjetivas más utilizadas para obtener una evaluación global de usabilidad percibida. Brooke lo desarrolló como un instrumento “rápido” para recopilar una medida subjetiva general tras el uso de un sistema, compuesto por 10 ítems con respuesta tipo Likert y un cálculo que entrega un puntaje global entre 0 y 100 [24]. Además, el propio Brooke describe su origen en un programa de ingeniería de usabilidad en Digital Equipment Co. y su propósito de permitir comparaciones rápidas entre versiones [24], [42].

El SUS se caracteriza por [24], [43]:

- Rápido de administrar (cuestionario breve) y aplicable a múltiples tipos de sistemas.

- Estandarizado y ampliamente usado, lo que facilita comunicar resultados y comparar iteraciones.
- Entregar un indicador global, útil como métrica “resumen” que se complementa con métricas de desempeño (éxito, errores, tiempo).

Aunque el SUS genera un puntaje de 0 a 100, su interpretación suele basarse en benchmarks y rangos prácticos. Investigaciones han propuesto escalas interpretativas para asociar puntajes con niveles comprensibles (por ejemplo, “aceptable”, “bueno”, “excelente”), facilitando la comunicación a equipos no especializados [43].

En particular:

- Bangor, Kortum y Miller reportan que los puntajes en los 70 tienden a considerarse aceptables, en los 80 buenos, y en los 90 excepcionales, mientras que valores bajo ~70 sugieren problemas de usabilidad que requieren atención [43].
- Trabajos posteriores discuten el uso de referencias como 68 (promedio) y 80 (sobre el promedio) como puntos de comparación, destacando que los benchmarks pueden variar según tipo de producto [44].
- En contextos de salud digital, revisiones y estudios han respaldado el uso del benchmark de 68 como referencia ampliamente aceptada para interpretar resultados de SUS [45].

En consecuencia, el SUS debe interpretarse como una medida comparativa y contextual: es especialmente útil para comparar iteraciones del mismo producto o contrastar resultados entre perfiles de usuarios en el mismo contexto de uso.

En VisitaConnect, estas métricas se integran como parte del QA para generar evidencia trazable sobre la calidad del prototipo:

- Éxito por tarea para verificar si los usuarios completan flujos críticos (reserva de visita, validación de QR, visualización/gestión de visitas).
- Errores por tarea para identificar puntos de confusión (por ejemplo, campos mal interpretados, selección incorrecta, pasos omitidos).
- Tiempo por tarea como indicador complementario de eficiencia cuando las condiciones de prueba lo permiten.
- SUS para obtener un puntaje cuantitativo de usabilidad percibida y comparar percepción entre perfiles (visitantes vs. guardias, etc.), además de evaluar si las mejoras entre iteraciones se traducen en mayor usabilidad percibida [24], [43], [44].

Así, las métricas no reemplazan la visión amplia del QA, sino que la fortalecen: permiten demostrar avances de calidad con indicadores comparables, complementando los hallazgos cualitativos (observaciones y citas textuales) con resultados cuantificables [40], [41].

4.7 QA EN SISTEMAS DE SALUD Y SALUD DIGITAL

En el ámbito de la salud digital, el aseguramiento de calidad (QA) adquiere una relevancia particularmente alta, debido a que el software puede influir directamente en procesos sensibles y en la seguridad de la operación. A diferencia de otros dominios, los sistemas de salud suelen manejar información personal y operar en escenarios donde los errores pueden provocar consecuencias relevantes (por ejemplo, confusiones, atrasos, autorizaciones incorrectas o exposición de información), por lo que resulta crítico definir y aplicar controles de calidad desde etapas tempranas del desarrollo. Este enfoque se alinea con la necesidad, destacada por la OMS, de que las iniciativas de salud digital estén guiadas por estrategias robustas que consideren gobernanza, implementación y principios para generar valor sin introducir riesgos. [49]

Uno de los factores que vuelve crítica la calidad en salud digital es la sensibilidad de los datos tratados (pacientes, familiares, personal y procesos internos). En este contexto, el QA debe incorporar prácticas que aseguren control de acceso, consistencia del proceso, y mecanismos que permitan trazabilidad (quién realizó qué acción y cuándo), de forma coherente con las políticas del establecimiento. Desde una mirada normativa, estándares del dominio de salud abordan explícitamente requisitos asociados a seguridad y ciclo de vida: por ejemplo, IEC 82304-1 establece requisitos de seguridad (safety) y seguridad informática (security) aplicables a productos de software de salud que operan en plataformas de cómputo generales, considerando el ciclo de vida completo (diseño, desarrollo, validación, instalación y mantención). [46]

Si bien VisitaConnect no se plantea como un “dispositivo médico”, estos marcos son útiles conceptualmente porque refuerzan que, en salud digital, QA no se limita a “probar que funciona”, sino que debe contemplar aspectos de seguridad del producto, documentación y validación en un contexto con riesgos reales. De forma complementaria, IEC 62304 define procesos de ciclo de vida para software de dispositivos médicos, reforzando la importancia de prácticas sistemáticas como planificación, verificación y validación dentro del desarrollo. [47]

En sistemas de salud, la usabilidad no es solo una dimensión estética o de conveniencia, sino que se vincula con la seguridad operativa y con la continuidad del trabajo. Evidencia institucional y técnica sostiene que un diseño deficiente puede contribuir a errores y a interrupciones del flujo de trabajo: AHRQ, por ejemplo, describe que una mala usabilidad en registros clínicos electrónicos puede provocar errores, comprometer seguridad del paciente y aumentar carga cognitiva, afectando el desempeño del personal. [51]

En la misma línea, NIST ha desarrollado iniciativas y marcos enfocados en definir y evaluar la usabilidad de sistemas de Health IT con procedimientos medibles y repetibles,

precisamente porque la usabilidad se relaciona con uso seguro, eficiente y efectivo para múltiples perfiles (clínicos, administrativos, pacientes y otros). [50]

En un hospital público como el Hospital Dr. Gustavo Fricke, esta dimensión es especialmente crítica porque:

- Muchos usuarios pueden no ser expertos en tecnología (p. ej., familiares o visitantes), por lo que la aprendibilidad y la claridad del flujo influyen directamente en el éxito del proceso;
- Guardias y personal administrativo suelen operar bajo alta carga, por lo que el sistema debe apoyar el trabajo (reducir pasos, prevenir errores) y no introducir fricción adicional;
- La solución debe aportar seguridad y trazabilidad del proceso de visitas sin agregar riesgos por confusión, ambigüedad o exposición indebida de información.

Diversos estudios muestran que instrumentos estandarizados de usabilidad se utilizan de forma frecuente en salud digital. En particular, el System Usability Scale (SUS) ha sido aplicado ampliamente para cuantificar la usabilidad percibida en aplicaciones de salud digitales; por ejemplo, Hyzy et al. Analizan su uso para *digital health apps* y concluyen que el SUS y su benchmark ampliamente adoptado (media 68, SD 12.5) son adecuados para evaluar este tipo de soluciones. [52] Esto lo vuelve útil para comparar iteraciones, apoyar decisiones de rediseño y comunicar resultados a actores no técnicos, especialmente cuando se complementa con métricas de desempeño (éxito por tarea, errores) y evidencia cualitativa (observaciones y citas textuales).

En VisitaConnect, estas consideraciones se integran al QA al tratar el prototipo como una solución de salud digital orientada a proceso: se prioriza la usabilidad (claridad de flujos y reducción de errores), se contemplan medidas básicas de seguridad de la información visible en pantalla (principio de mínima exposición de datos y control por perfil) y se busca asegurar trazabilidad del proceso en la lógica de interacción. Además, el uso de pruebas con usuarios y métricas como SUS permite sustentar, con evidencia, si el sistema facilita el trabajo operativo del hospital y si resulta comprensible para perfiles heterogéneos, coherente con la orientación de la OMS hacia intervenciones digitales que consideren beneficios, riesgos, aceptabilidad y factibilidad. [48], [49], [52]

4.8 RELACIÓN DEL MARCO DE QA CON EL PROYECTO VISITACONNECT

Los conceptos desarrollados en este capítulo se vinculan directamente con la forma en que se planifica y ejecuta el aseguramiento de calidad en el Trabajo de Título, ya que permiten definir qué se entiende por calidad, cómo se evalúa y con qué evidencia se justifica la mejora del prototipo.

En primer lugar, el Aseguramiento de Calidad de Software (SQA) entrega el marco para comprender la calidad como un enfoque planificado y transversal que acompaña el ciclo de

vida del desarrollo, evitando tratarla como una etapa final de "revisión" posterior a la construcción. Esta visión se alinea tanto con la lógica de procesos de ciclo de vida descritos en ISO/IEC/IEEE 12207 [21] como con estándares orientados a asegurar procesos de calidad en proyectos de software, como IEEE 730 [23]. En consecuencia, VisitaConnect adopta QA como un componente que inicia en la definición de requisitos y criterios de calidad, continúa en el diseño del prototipo y culmina en su evaluación mediante pruebas y métricas.

En segundo lugar, el modelo ISO/IEC 25010 se utiliza como base para estructurar la calidad del producto y, particularmente, para priorizar la usabilidad como atributo clave en un sistema de gestión de visitas hospitalarias [14], [22]. Esta elección se justifica porque el sistema está orientado a perfiles heterogéneos (visitantes y personal del hospital) y requiere flujos claros y robustos. En términos operacionales, este marco guía la definición de criterios de evaluación y la selección de métricas enfocadas en la interacción.

Asimismo, la distinción entre QA y testing permite posicionar correctamente las actividades realizadas en la Entrega 2. Mientras que el testing se enfoca en evaluar el producto y detectar fallas, el QA comprende un enfoque más amplio orientado a asegurar el proceso y prevenir defectos mediante criterios, revisiones, trazabilidad y planificación [27], [30], [32]. Bajo esta lógica, las pruebas funcionales y de usabilidad aplicadas en VisitaConnect se entienden como evidencias dentro de una estrategia de calidad, no como el único mecanismo de QA.

En cuarto lugar, las pruebas de usabilidad y sus métricas asociadas aportan evidencia concreta sobre la calidad de la experiencia ofrecida por el prototipo, coherente con el enfoque de diseño centrado en el usuario de ISO 9241-210 [3]. La medición de desempeño mediante tasa de éxito, errores y tiempos, junto con la evaluación subjetiva, se sustenta en prácticas comunes de medición de UX/usabilidad [41] y en normas de medición de calidad en uso [40]. En particular, el System Usability Scale (SUS) aporta un indicador estandarizado y ampliamente utilizado para cuantificar usabilidad percibida [24], complementando la evidencia observacional y cualitativa recogida durante las sesiones de prueba.

Finalmente, la literatura y marcos de salud digital refuerzan la pertinencia de evaluar rigurosamente la usabilidad y la seguridad en soluciones utilizadas por pacientes, familiares y personal. Recomendaciones y estrategias de la OMS enfatiza la necesidad de implementar intervenciones digitales con criterios de calidad y evaluación que consideren riesgos y contexto [48], [49]. A su vez, estándares de software de salud (como IEC 82304-1) remarcan la importancia de requisitos de seguridad del producto y consideraciones de ciclo de vida en este tipo de soluciones [46]. Esto refuerza que, para un hospital público, la solución debe ser comprensible para usuarios no expertos y no debe entorpecer el trabajo bajo carga del personal, manteniendo control y trazabilidad del proceso.

En síntesis, el marco conceptual de QA no solo respalda la metodología aplicada en la segunda entrega, sino que también fundamenta las decisiones de diseño, prueba y mejora continua incorporadas en VisitaConnect: definir calidad (SQA), estructurar atributos



(ISO/IEC 25010), planificar pruebas (niveles/estrategias), evaluar usabilidad con métricas y SUS, y justificar su relevancia en el contexto de salud digital [21], [23], [14], [27], [40], [24], [48].

5 CAPÍTULO 3: PROPUESTA DE SOLUCIÓN

5.1 VISIÓN GENERAL DE LA SOLUCIÓN (VISITACONNECT)

VisitaConnect es una plataforma digital orientada a optimizar el registro, control y validación de visitas en un contexto hospitalario, reemplazando procesos manuales y fragmentados por un flujo centralizado, trazable y más eficiente. La solución busca disminuir los tiempos de atención en el acceso, reducir errores de registro y mejorar la coordinación entre los distintos actores involucrados en la gestión de visitas, permitiendo que el proceso sea más ordenado y seguro tanto para visitantes como para el personal del establecimiento.

A diferencia de una implementación centrada únicamente en “hacer que funcione”, VisitaConnect se diseñó con un enfoque explícito en aseguramiento de calidad (QA) y usabilidad, considerando desde el inicio criterios de interacción clara, consistencia visual, prevención de errores y facilidad de aprendizaje para perfiles con distintos niveles de experiencia digital. Este enfoque se respalda con actividades planificadas de evaluación (pruebas funcionales y de usabilidad), junto con métricas que aportan evidencia del desempeño del prototipo y su alineación con las necesidades reales del entorno hospitalario.

5.2 ALCANCE Y PERFILES DE USUARIOS CONSIDERADOS

El alcance de VisitaConnect comprende el diseño y evaluación de un prototipo funcional de plataforma web orientada a la gestión de visitas hospitalarias, incorporando los flujos esenciales para registrar y controlar el acceso de visitantes. La solución incluye funcionalidades como registro de visitas, visualización de información asociada a la visita, validación mediante código QR en el punto de control y un flujo de consulta/seguimiento que permite apoyar la trazabilidad del proceso. En términos de calidad, el alcance considera además la aplicación de actividades de QA asociadas al prototipo, tales como definición de criterios de aceptación, ejecución de pruebas funcionales y realización de pruebas de usabilidad con métricas.

Como exclusiones, VisitaConnect no aborda la integración con sistemas clínicos internos (por ejemplo, ficha clínica electrónica), ni contempla procesos institucionales avanzados como interoperabilidad con bases de datos externas, automatización de agendas médicas, control biométrico, analítica avanzada, ni despliegue productivo con infraestructura hospitalaria real. Tampoco se incluye, dentro del alcance de este trabajo, una operación a escala institucional con políticas completas de ciberseguridad, continuidad operacional o monitoreo 24/7, dado que la propuesta se centra en validar la solución desde una perspectiva de diseño y calidad de uso.

Para asegurar que la solución respondiera a necesidades reales, se consideraron tres perfiles de usuario clave: (1) Visitantes, quienes requieren un proceso simple, claro y rápido para registrarse y acceder sin errores; (2) Guardias de seguridad, responsables de verificar visitas y autorizar el ingreso, por lo que necesitan una validación ágil y confiable (por ejemplo, lectura/confirmación de QR) y reducción de ambigüedades; y (3) Personal clínico/administrativo, que interactúa con la gestión de visitas desde una perspectiva

operativa (apoyo a la coordinación, revisión de información y control del flujo). La definición explícita de estos perfiles es fundamental para el aseguramiento de calidad, ya que permite establecer criterios de aceptación realistas por tipo de usuario, diseñar casos de prueba representativos y planificar evaluaciones de usabilidad que reflejen condiciones y tareas propias del contexto hospitalario.

5.3 REQUISITOS Y CRITERIOS DE ACEPTACIÓN ORIENTADOS A CALIDAD

Para asegurar que VisitaConnect no sólo “funcione”, sino que además entregue una experiencia consistente y confiable, los requisitos se definieron como funcionales (RF) y no funcionales (RNF). Cada uno incorpora un criterio de aceptación verificable, redactado en formato “Se considera correcto si...”, y se organiza según las fases de liberación (releases) del proyecto: Fase I, Fase II y Fase III - [Tabla 5-1 y Tabla 5-2]. Esta estructura permite planificar pruebas representativas y demostrar trazabilidad entre lo construido, lo probado y la evidencia obtenida.

Tabla 5-1. Requisitos funcionales (RF) y criterios de aceptación por fase

ID	Requisito Funcional	Criterio de Aceptación
Fase I	MVP Funcional (Primer Release)	
RF-01	Registro de visita	El visitante puede ingresar los datos mínimos requeridos y el sistema confirma la creación de la visita asignándole un identificador único.
RF-02	Validación de campos	El sistema impide guardar los datos de una visita cuando se omiten campos obligatorios, desplegando mensajes de error claros en la interfaz.
RF-03	Consulta de visita	El usuario puede visualizar de forma correcta el detalle completo de una visita registrada (a través del correo electrónico).

RF-04	Trazabilidad básica (estado)	Cada visita nueva es registrada con un estado inicial definido y estandarizado (por ejemplo: "pendiente").
Fase II	Plataforma Avanzada (Segundo Release)	
RF-05	Confirmación por correo	Al registrar una visita, el sistema envía automáticamente un correo de confirmación al destinatario configurado con la información resumida de la reserva.
RF-06	Generación de código QR	El sistema genera un código QR único asociado a la visita, el cual puede ser visualizado e interactuado de forma correcta desde la plataforma.
RF-07	Validación de QR en acceso	La interfaz del guardia permite verificar el QR de acceso y el sistema devuelve un resultado inequívoco sobre su condición (válida, no válida, no encontrada o expirada).
RF-08	Control administrativo detallado	El personal clínico y administrativo puede gestionar el flujo de accesos mediante acciones definidas, tales como cambiar estados, revisar información y aplicar filtros por fecha o estado.
Fase III	Cierre y Optimización (Tercer Release)	

RF-09	Búsqueda y filtrado	El sistema permite localizar visitas mediante criterios específicos (nombre, RUT/ID, fecha, estado), entregando resultados consistentes con los registros existentes.
RF-10	Entrega de QR por correo	El correo de confirmación incluye el código QR de forma accesible (ya sea mediante enlace o archivo adjunto), permitiendo al usuario obtenerlo de manera directa.
RF-11	Manejo de errores controlado	Ante fallas esperables (QR inválido, registro no encontrado o campos mal ingresados), el sistema mantiene su estabilidad operativa y muestra una respuesta clara y accionable al usuario.
RF-12	Roles y permisos consolidados	Cada perfil de usuario (visitante, guardia, personal interno) visualiza y ejecuta estrictamente las acciones y datos que corresponden a sus permisos asignados.

Fuente: Elaboración Propia

Tabla 5-2. Requisitos no funcionales (RNF) y criterios de aceptación por fase

ID	Requisito No Funcional	Criterio de Aceptación
Fase I	MVP Funcional (Primer Release)	
RNF-01	Usabilidad base (comprensibilidad)	Los usuarios pueden completar las tareas principales de forma intuitiva sin requerir

		capacitación previa, interactuando con mensajes claros y comprensibles.
RNF-02	Consistencia mínima de interfaz	Los componentes similares (botones, formularios, alertas) mantienen el mismo comportamiento visual y nomenclatura en todas las interfaces del MVP.
RNF-03	Disponibilidad del prototipo	El sistema se ejecuta y permanece operativo durante las sesiones de prueba sin presentar fallas críticas que impidan completar los flujos establecidos.
Fase II	Plataforma Avanzada (Segundo Release)	
RNF-04	Eficiencia del flujo (tiempos)	Las tareas críticas (como el registro de visitas y la validación de accesos) se completan de manera fluida y sin pasos innecesarios, según lo cuantificado en las pruebas.
RNF-05	Confiabilidad de validación	La validación de los códigos QR entrega resultados consistentes y deterministas ante condiciones idénticas y repetidas (mismo QR implica el mismo estado).
RNF-06	Compatibilidad básica (dispositivo)	La plataforma web posee un diseño adaptativo (<i>responsive</i>) que garantiza su correcta usabilidad tanto desde computadores de escritorio como desde <i>smartphones</i> en escenarios reales.

Fase III	Cierre y Optimización (Tercer Release)	
RNF-0 7	Seguridad en accesos	Las funciones y vistas administrativas exigen autenticación obligatoria, asegurando que las acciones sensibles queden estrictamente restringidas según el rol asignado.
RNF-0 8	Rendimiento percibido	Las interfaces principales responden sin retrasos perceptibles para el usuario, asegurando que la validación del QR no introduzca tiempos de espera que afecten el control de acceso físico.
RNF-0 9	Mantenibilidad del prototipo	Los módulos de software están organizados de forma que permiten realizar ajustes, correcciones o cambios visuales puntuales sin alterar ni romper las funcionalidades ya validadas.

Fuente: Elaboración Propia

5.4 ESTRATEGIA DE ASEGURAMIENTO DE CALIDAD APLICADA

El Aseguramiento de Calidad de Software (SQA) en VisitaConnect tuvo como objetivo entregar confianza sobre la calidad del prototipo a lo largo del ciclo de diseño y construcción, asegurando principalmente: (1) confiabilidad del prototipo para ejecutar los flujos críticos sin fallas, (2) consistencia de interfaz (UI) para reducir confusión entre pantallas y perfiles, (3) reducción de errores mediante validaciones y manejo controlado de situaciones esperables, y (4) alta usabilidad, considerando que el sistema se utiliza en un contexto hospitalario con distintos niveles de experiencia digital y bajo presión de tiempo.

Para cumplir este objetivo, se ejecutaron actividades de QA planificadas y sistemáticas. En primer lugar, se realizó la revisión de requisitos y consistencia de flujos, verificando que los procesos definidos (registro, consulta y control/validación) fueran coherentes con los perfiles de usuario y con el alcance del proyecto. Luego, se llevó a cabo la definición de casos de prueba, estableciendo criterios de aceptación verificables ("se considera correcto si...") y pruebas orientadas a cubrir los escenarios más críticos. Posteriormente, se efectuó una prueba piloto, con el fin de ajustar tareas, instrucciones y el guión del moderador,

asegurando que el protocolo de evaluación fuera claro antes de aplicar las pruebas con participantes. Con el protocolo ajustado, se ejecutaron pruebas funcionales para validar el comportamiento esperado del sistema y su respuesta ante errores comunes. En paralelo, se desarrollaron pruebas de usabilidad moderadas utilizando la técnica de pensamiento en voz alta, complementadas con la aplicación del cuestionario SUS, para obtener evidencia cuantitativa y cualitativa sobre la experiencia de uso. Finalmente, se realizó el registro de hallazgos y mejoras, documentando incidencias, oportunidades de ajuste y cambios aplicados al prototipo, reforzando el enfoque de mejora continua propio del SQA.

5.5 DISEÑO DE PRUEBAS: CASOS DE PRUEBA Y MÉTRICAS

Las pruebas de VisitaConnect se diseñaron para validar la solución desde dos perspectivas complementarias: (1) evidencia cuantitativa, orientada a medir desempeño del sistema (tasa de éxito por tarea, número de errores, tiempos, tiempos de respuesta), y (2) evidencia cualitativa, orientada a comprender la experiencia de uso (comentarios, confusiones, percepciones y observaciones durante la ejecución). Esta estrategia permite sustentar el aseguramiento de calidad (SQA) no solo en “que funcione”, sino en qué tan usable, consistente y confiable resulta para cada perfil.

Tareas evaluadas por perfil

- Visitantes
 - Crear una reserva/registro de visita con datos válidos.
 - Identificar confirmación de la reserva y comprender el resultado del proceso.
 - Recibir y entender el correo de confirmación (incluyendo el QR cuando aplica).
- Guardias de seguridad
 - Validar QR en el acceso y tomar decisión según el resultado (autorizada/rechazada).
 - Manejar escenarios de error (QR inválido, expirada, visitante bloqueado).
- Personal clínico/administrativo
 - Ejecutar control administrativo sobre visitas (por ejemplo: bloqueo de visitante por RUT).
 - Verificar que las restricciones impacten el flujo del sistema (no permitir nuevas reservas/validaciones).

Evidencia recolectada (métricas)

- Cuantitativa: tasa de éxito por tarea, número de errores por tarea, tiempo por tarea, tiempos de respuesta (en validación QR y creación de reserva), puntaje SUS.
- Cualitativa: comentarios espontáneos, observaciones del moderador, puntos de confusión y sugerencias de mejora (pensamiento en voz alta).

A continuación se muestra la confección de cada tipo de prueba (funcionales, integración, usabilidad, rendimiento/carga y seguridad). [Tabla 5-3, 5-4, 5-5 y 5-6]

Tabla 5-3. Casos de prueba funcionales

ID	Caso de Prueba	Precondición	Pasos de Ejecución	Resultado Esperado
F1	Crear reserva correctamente	1. El usuario "Visitante" tiene acceso al módulo de reservas/registro. 2. El sistema está operativo y permite seleccionar fecha/hora.	1. Ingresar al módulo "Reservar visita". 2. Ingresar RUT válido. 3. Completar datos mínimos obligatorios. 4. Seleccionar fecha disponible en el calendario. 5. Seleccionar hora disponible. 6. Revisar el resumen de la reserva.	1. El sistema registra la reserva sin errores. 2. Muestra mensaje visible: "Reserva confirmada". 3. Genera un identificador único asociado. 4. Genera un código QR asociado y visible. 5. La reserva queda almacenada para consultas.

			7. Presionar el botón "Confirmar / Reservar". 8. Esperar respuesta del sistema.	
F2	Crear reserva con datos inválidos	1. El usuario "Visitante" tiene acceso al formulario. 2. El sistema aplica validaciones y reglas de negocio a los campos.	1. Ingresar al módulo de reservas. 2. Ingresar un RUT inválido (formato o DV incorrecto). 3. Intentar continuar sin corregir el RUT. 4. Seleccionar una fecha fuera de rango o pasada. 5. Dejar campos obligatorios vacíos. 6. Presionar "Confirmar / Reservar".	1. El sistema impide completar la reserva. 2. Muestra errores claros (ej: "RUT inválido", "Campo obligatorio"). 3. No se crea ningún registro en la base de datos. 4. Permite corregir campos sin bloquearse.

F3	Envío de correo de confirmación	1. Existe una reserva creada exitosamente (F1). 2. Funcionalidad de envío de correo habilitada. 3. Se ingresó un correo válido.	1. Completar el flujo de creación de reserva (F1). 2. Verificar indicación de envío en la interfaz. 3. Acceder a la bandeja de entrada del correo ingresado. 4. Buscar correo con asunto alusivo a la reserva. 5. Abrir el correo y revisar el contenido. 6. Verificar datos principales (fecha/hora/ID). 7. Verificar disponibilidad del código QR.	1. El correo llega correctamente al destinatario. 2. El contenido es coherente con la reserva. 3. El QR se encuentra accesible (adjunto o enlace). 4. No existen errores visuales ni enlaces rotos.
F4	Validación de QR en la entrada (QR válido)	1. Usuario "Guardia" con acceso al módulo de validación.	1. Iniciar sesión con rol de Guardia. 2. Acceder al módulo	1. El sistema reconoce el QR y halla la reserva. 2. Muestra el mensaje:

		2. Existe una reserva válida con QR generado (F1). 3. El QR no está expirado ni bloqueado.	"Validación de QR". 3. Escanear el QR válido o ingresar el ID manual. 4. Confirmar la validación (si aplica). 5. Observar el resultado en pantalla. 6. Repetir la validación para verificar consistencia.	"Visita autorizada". 3. Despliega información del visitante y la reserva. 4. El estado de la visita cambia a "válida". 5. El sistema responde sin demoras notorias.
F5	Validación con QR inválido o expirado	1. Usuario "Guardia" con acceso al módulo de validación. 2. Se dispone de un QR alterado o de una reserva expirada.	1. Acceder al módulo de validación como guardia. 2. Escanear un QR inválido (inexistente). 3. Observar la respuesta en pantalla. 4. Escanear un QR de una	1. Para QR inválido: muestra "Código inválido" y deniega acceso. 2. Para QR expirado: muestra "Visita expirada" y deniega acceso. 3. No se producen caídas ni

			reserva ya expirada. 5. Observar la respuesta del sistema.	errores técnicos.
F6	Módulo administrativo: bloquear visitante	1. Usuario con rol Admin o Administrativo autenticado. 2. Existe un visitante registrado e identificado por RUT.	1. Iniciar sesión con rol administrativo. 2. Acceder al módulo "Gestión de visitantes". 3. Buscar al visitante por su RUT. 4. Seleccionar el registro correspondiente. 5. Ejecutar la acción "Bloquear". 6. Confirmar la acción en la ventana emergente. 7. Verificar que cambie el estado a "bloqueado".	1. El sistema aplica el bloqueo sin errores. 2. El visitante cambia visualmente a estado bloqueado. 3. El sistema entrega un mensaje de confirmación. 4. El bloqueo persiste de forma correcta en la BD.

F7	Restricción: visitante bloqueado no puede reservar	1. Existe un visitante en estado bloqueado (F6). 2. El usuario intenta reservar con el RUT bloqueado.	1. Ingresar al módulo de reservas como visitante. 2. Ingresar el RUT que se encuentra bloqueado. 3. Completar los demás campos obligatorios. 4. Seleccionar fecha y hora disponibles. 5. Presionar "Confirmar / Reservar".	1. El sistema impide estrictamente crear la reserva. 2. Muestra alerta: "Visitante bloqueado / No autorizado". 3. No se genera ningún ID de reserva ni código QR. 4. No se alteran los registros de visitas activas.
----	---	---	---	--

Fuente: Elaboración Propia

Tabla 5-4. Casos de prueba de integración

ID	Caso de Integración	Precondición	Pasos de Ejecución	Resultado Esperado
I1	Reserva → Base de Datos (Persistencia)	1. Sistema operativo con conexión activa a la base de datos. 2. Usuario visitante con	1. Ingresar al módulo "Reservar visita". 2. Completar el formulario con un RUT válido y	1. Existe un registro nuevo en la base de datos que corresponde exactamente a

		acceso al módulo de reservas. 3. Se cuenta con datos válidos para la creación de un registro.	campos obligatorios. 3. Seleccionar fecha y hora disponibles. 4. Presionar "Confirmar / Reservar" y verificar el mensaje en interfaz. 5. Identificar el ID de reserva generado (o la combinación RUT + fecha/hora). 6. Acceder a la base de datos mediante la herramienta de administración. 7. Consultar las tablas de reservas filtrando por el identificador. 8. Verificar los campos persistidos	la reserva realizada. 2. Los datos almacenados coinciden con los ingresados en la interfaz. 3. El registro posee su identificador único y el estado inicial "pendiente". 4. No se generan registros duplicados para una misma transacción.
--	--	---	---	--

			(estado inicial, fecha, hora).	
I2	QR Generado → Validador (Interoperabilidad QR)	1. Existe una reserva creada correctamente con su código QR asociado (F1). 2. El QR está visible en la plataforma o disponible en el correo electrónico. 3. Se dispone del mecanismo de lectura (cámara o ingreso manual).	1. Crear una reserva válida (F1) hasta visualizar el código QR. 2. Obtener el QR desde la pantalla o desde el correo electrónico. 3. Acceder al módulo de validación de QR con el rol de Guardia. 4. Escanear el código QR (o ingresar el ID de forma manual). 5. Observar el valor alfanumérico reconocido por el sistema. 6. Repetir la lectura con el mismo QR para comprobar consistencia.	1. El código QR es legible y la cámara lo captura sin errores. 2. El valor decodificado corresponde a una referencia válida (ID de reserva existente). 3. El módulo de validación interpreta el valor y localiza la visita de forma correcta. 4. Las lecturas repetidas producen exactamente el mismo resultado.

I3	Módulo Admin → Módulo Guardias (Impacto de bloqueo)	1. Usuario Admin con permisos para bloquear visitantes. 2. Usuario Guardia con acceso al módulo de validación. 3. Existe una reserva activa para un visitante con su QR generado. 4. El visitante (RUT) es identificable y bloqueable.	1. Crear una reserva válida para un visitante (F1) y asegurar su QR. 2. Iniciar sesión como Admin y acceder al módulo administrativo. 3. Buscar al visitante por RUT, ejecutar "Bloquear" (F6) y confirmar. 4. Cerrar sesión de Administrador. 5. Iniciar sesión como Guardia y abrir el módulo de validación. 6. Escanear el QR asociado a la reserva del visitante que fue bloqueado. 7. Evaluar el resultado y el mensaje desplegado en la pantalla.	1. El bloqueo aplicado por el Admin se propaga en tiempo real al módulo del guardia. 2. El sistema rechaza la validación mostrando un mensaje inequívoco (ej: "Visitante bloqueado / No autorizado"). 3. No se permite cambiar el estado de la visita a "autorizada" o "validada". 4. El sistema mantiene su estabilidad operativa y responde con claridad.
----	---	--	--	---

Fuente: Elaboración Propia

Tabla 5-5. Métricas de usabilidad por tarea

Métrica	Definición / Cómo se mide	Evidencia / Instrumento
Tasa de éxito por tarea	Porcentaje de participantes que logran completar de forma correcta la tarea asignada sin requerir asistencia externa.	Checklist de finalización de tareas.
Nº de errores por tarea	Conteo total de incidencias registradas durante el flujo (fallas de validación, retrocesos involuntarios, confusiones en la interfaz o intentos fallidos).	Registro de observación de QA.
Tiempo por tarea	Duración cronometrada en segundos/minutos desde el inicio del flujo hasta la confirmación de término de la tarea.	Cronometraje individual por participante.
System Usability Scale (SUS)	Escala global de usabilidad estandarizada, calculada mediante un puntaje final de 0 a 100 aplicado de forma posterior a la sesión de pruebas.	Cuestionario psicométrico SUS.
Pensamiento en voz alta (<i>Think Aloud</i>)	Registro cualitativo de las expresiones verbales, dudas espontáneas y dificultades manifestadas por el usuario durante la interacción.	Notas de campo y bitácora del moderador.

Fuente: Elaboración Propia

Tabla 5-6. Casos de prueba de rendimiento/carga (En entorno local)

ID	Caso de Rendimiento / Carga	Precondición	Pasos de Ejecución	Resultado Esperado
R1	Carga concurrente: múltiples usuarios creando reservas simultáneamente (ej.: 50 usuarios)	1. Sistema desplegado en entorno ejecutable con acceso estable a BD y servicios asociados. 2. Se cuenta con un método para simular concurrencia (múltiples navegadores o herramientas de carga). 3. Existe disponibilidad de fechas y horas para permitir los registros.	1. Definir el objetivo de concurrencia (ej: 50 intentos en un intervalo corto). 2. Preparar datos de prueba generados (RUT/IDs y correos válidos). 3. Ejecutar en paralelo el flujo de creación de reserva en múltiples sesiones simultáneas. 4. Ingresar datos, seleccionar fecha/hora y confirmar en cada sesión. 5. Registrar tiempo de respuesta, tasa de finalización y	1. El sistema mantiene la operación estable y sin caídas durante la concurrencia. 2. La mayoría de las solicitudes completan el flujo exitosamente. 3. Los tiempos de respuesta se mantienen dentro del umbral aceptable (promedio < 3s). 4. No se generan errores críticos del servidor (500) ni bloqueos de interfaz. 5. Las reservas se

			errores visualizados. 6. Revisar la base de datos para detectar duplicados o registros incompletos. 7. Analizar los logs del sistema para identificar fallas de servidor o <i>timeouts</i>.	registran de forma consistente en BD, sin duplicados ni campos nulos.
R2	Escaneo/validación de QR en "hora punta" (alto ritmo de validación)	1. Usuario con rol de Guardia activo y con acceso al módulo de validación. 2. Se dispone de un set de 20 a 30 QRs válidos y algunos inválidos/expirados para la muestra. 3. El módulo de validación está operativo y conectado a la BD.	1. Preparar el conjunto de códigos QR generados previamente. 2. Iniciar sesión como Guardia y abrir el módulo correspondiente. 3. Simular escenario de alta demanda validando 20 QRs de forma consecutiva en un lapso de 5 minutos. 4. Registrar el tiempo transcurrido	1. El sistema responde de forma rápida y consistente para cada validación, sin afectar el flujo de acceso. 2. Los resultados se muestran de manera inmediata e inequívoca en la interfaz. 3. No se presenta una degradación progresiva del rendimiento del software

			desde la acción de escaneo hasta la respuesta en pantalla. 5. Interpolar casos inválidos o expirados para medir la respuesta del sistema bajo excepciones. 6. Verificar la ausencia de congelamientos o retrasos acumulativos.	con el paso del tiempo. 4. El módulo de validación no sufre caídas ni errores críticos de servidor. 5. Los estados de las visitas se actualizan correctamente en la BD a pesar del alto ritmo de lectura.
--	--	--	---	--

Fuente: Elaboración Propia

Tabla 5-7. Casos de prueba de seguridad

ID	Caso de Seguridad	Precondición	Pasos de Ejecución	Resultado Esperado
S1	Intento de acceder a módulo administrativo sin rol autorizado	1. Existe al menos un usuario autenticado sin permisos de administrador (Visitante o Guardia). 2. Existe una ruta, vista o	1. Iniciar sesión con un usuario no administrativo (o permanecer anónimo). 2. Intentar acceder al panel administrativo mediante la navegación	1. El sistema deniega estrictamente el acceso al módulo administrativo. 2. Si el usuario no está autenticado, se le redirige automáticamente

		panel de administración implementado. 3. El sistema cuenta con control de autenticación y autorización por roles.	normal de la interfaz. 3. Intentar forzar el acceso ingresando directamente la URL o ruta administrativa en el navegador. 4. Intentar ejecutar una acción sensible (ej: bloquear visitante) si la UI llega a exponer el módulo. 5. Registrar el comportamiento del sistema (redirecciones, denegaciones).	al login o se bloquea la vista. 3. Si posee un rol incorrecto, se despliega un mensaje claro de acceso denegado. 4. No se exponen datos administrativos ni acciones sensibles a perfiles no autorizados. 5. Es imposible procesar o persistir operaciones de administración desde un rol no correspondiente.
S2	Datos sensibles en tránsito (HTTPS / protección de datos)	1. El sistema está ejecutándose en un entorno donde es posible verificar los protocolos de red. 2. Existen interfaces activas que recopilan datos personales	1. Abrir la plataforma desde el navegador y validar si la URL implementa el protocolo HTTPS y el candado de seguridad. 2. Acceder a los formularios de registro de visitas, inicio de	1. Toda comunicación de datos se realiza mediante HTTPS en entornos de despliegue públicos. 2. No se presentan advertencias críticas de certificados



		(RUT, nombre, credenciales).	sesión y validación.	inválidos en el navegador.
			3. Enviar los formularios y verificar que no existan alertas de "sitio no seguro" o contenido mixto.	3. Los datos privados ingresados no quedan expuestos en cadenas de texto de la URL ni en logs visuales.
			4. Inspeccionar que no se expongan datos sensibles como parámetros visibles en la URL.	4. Si el prototipo corre localmente sin HTTPS, se declara explícitamente como limitación del entorno, manteniendo el estándar de no exposición de variables.
			5. Revisar la consola de desarrollador del navegador en busca de advertencias de seguridad.	

Fuente: Elaboración Propia



6 CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN

La validación de VisitaConnect se abordó desde una perspectiva estrictamente funcional, utilizando el marco de la Feria de Software como un entorno controlado de simulación. Si bien la plataforma no fue puesta a prueba ni desplegada en el escenario real del Hospital Dr. Gustavo Fricke, esta instancia permitió exponer el prototipo a ciclos de interacción práctica con usuarios y observadores, demostrando la viabilidad operativa y la comprensibilidad de los flujos críticos del sistema para los perfiles objetivos.

Desde un enfoque de Aseguramiento de Calidad de Software (SQA), la estrategia se estructuró mediante pruebas de validación funcional orientadas a certificar el comportamiento correcto de los requisitos. El proceso cubrió aspectos clave como: pruebas funcionales (verificación de creación de reservas, flujos de validación y emisión de QR), pruebas de integración a nivel de componentes (coherencia entre módulos y persistencia básica de datos) y pruebas de usabilidad cualitativas (observación directa, pensamiento en voz alta y la aplicación experimental del cuestionario SUS). Adicionalmente, se verificaron de forma dirigida controles básicos de seguridad por roles y la estabilidad general de la interfaz durante las jornadas de evaluación.

7 CAPÍTULO 5: CONCLUSIONES

7.1 CUMPLIMIENTO DE OBJETIVOS DEL PROYECTO

A partir del desarrollo, implementación y evaluación del prototipo VisitaConnect, se concluye que el objetivo general del Trabajo de Título se cumple: se logró crear y probar una plataforma para administrar visitas hospitalarias, aplicando un enfoque de aseguramiento de calidad (QA) orientado a que el sistema sea usable, accesible, eficiente y confiable en un contexto hospitalario.

En cuanto a los objetivos específicos, también se consideran cumplidos, dado que:

- Se definieron criterios de calidad alineados a atributos relevantes para el contexto (usabilidad, accesibilidad, seguridad y eficiencia), estableciendo un “lenguaje común” de calidad para el proyecto.
- Se diseñaron y ejecutaron evaluaciones funcionales y no funcionales para validar módulos críticos como registro/reserva, confirmación, QR y gestión administrativa.
- Se midió la experiencia de usuario con pruebas de usabilidad aplicadas a perfiles representativos (visitantes, guardias y personal interno), con el fin de detectar fallos y oportunidades de mejora.
- Se incorporó un análisis de confiabilidad y seguridad, especialmente asociado a control de accesos, permisos y protección de datos (a nivel de prototipo).
- Se documentaron hallazgos y se propusieron refinamientos basados en evidencia, reforzando la lógica de mejora continua propia de QA.

7.2 PRINCIPALES RESULTADOS DESDE EL ENFOQUE DE QA

Una conclusión clave del trabajo es que VisitaConnect no se construyó únicamente desde la funcionalidad, sino desde un enfoque explícito de QA + usabilidad, incorporando criterios y actividades de evaluación desde el inicio del diseño. En particular, el plan de SQA priorizó: confiabilidad del prototipo, consistencia de interfaz, reducción de errores y alta usabilidad, considerando presión de tiempo y perfiles con distintos niveles de alfabetización digital.

Este enfoque se materializó en actividades concretas: revisión de requisitos y flujos, definición de casos de prueba con criterios verificables, prueba piloto, pruebas funcionales, pruebas de usabilidad moderadas (pensamiento en voz alta) y registro de hallazgos/mejoras.

Además, la estrategia de pruebas se definió para recolectar evidencia cuantitativa (éxito por tarea, errores, tiempos y tiempos de respuesta) y cualitativa (observaciones, confusiones y comentarios), lo que permitió sustentar decisiones de mejora con datos, no solo con percepciones.

7.3 VALIDACIÓN DE LA SOLUCIÓN EN UN CONTEXTO REALISTA

Se concluye que la plataforma VisitaConnect demostró una correcta validez funcional a nivel de prototipo, habiendo sido evaluada con éxito de manera práctica en el marco controlado de la Feria de Software. Esta instancia permitió certificar desde el enfoque de SQA las categorías críticas de pruebas funcionales, de integración de componentes y de usabilidad (mediante observación directa y la aplicación del cuestionario SUS), logrando un comportamiento estable y un control de accesos por roles adecuado para la solución diseñada.

No obstante, se reconoce como una limitación metodológica que la solución no fue desplegada ni sometida a pruebas en el entorno real del Hospital Dr. Gustavo Fricke. Debido a esto, los resultados obtenidos corresponden a un escenario simulado, quedando pendiente para futuras líneas de trabajo la validación en terreno bajo las condiciones operativas, de infraestructura y de resistencia al cambio propias del contexto hospitalario definitivo.

7.4 ALCANCE Y LIMITACIONES

El trabajo presenta alcances claros: se trata de un prototipo web funcional, enfocado en flujos principales (registro/reserva, QR, visualización y gestión) y en evaluación de calidad (criterios, pruebas funcionales y usabilidad con métricas).

Sin embargo, existen limitaciones propias del alcance definido: no se implementó un despliegue productivo ni integración con sistemas clínicos internos; tampoco se cubre una operación institucional completa (p. ej., políticas de ciberseguridad avanzadas, continuidad operacional, monitoreo 24/7).

7.5 RECOMENDACIONES

Como líneas de continuidad y desarrollo futuro para este proyecto, se recomienda:

- Realizar la validación en el escenario operativo real: Trasladar la solución fuera del entorno controlado de simulaciones para iniciar pruebas piloto directamente en las dependencias del Hospital Dr. Gustavo Fricke, permitiendo evaluar el comportamiento de la plataforma ante flujos de usuarios reales y bajo la infraestructura técnica del centro de salud.
- Avanzar hacia un entorno productivo integrado: Evolucionar desde el prototipo funcional actual hacia una implementación definitiva, incorporando integraciones con los sistemas hospitalarios existentes (sistemas clínicos o de gestión interna), definiendo protocolos de acceso especiales según la unidad médica y fortaleciendo los esquemas de seguridad informática.
- Mantener el enfoque orientado a la calidad (*QA-Driven*): Ampliar la estrategia de aseguramiento de calidad mediante nuevas rondas de pruebas iterativas en terreno, incrementando la cobertura de escenarios de error complejos y aumentando la precisión de las métricas de rendimiento y estabilidad bajo condiciones de concurrencia real.



- Profundizar en la evaluación de usabilidad como pilar de calidad: Diseñar la evaluación de la experiencia de usuario como un atributo planificado y medible a lo largo de todo el ciclo de desarrollo (y no solo como una validación final), combinando de manera sistemática evidencia cuantitativa y cualitativa para sustentar las mejoras de la interfaz.

8 REFERENCIAS

- [1] D. Norman, The Design of Everyday Things, Basic Books, New York, 2013.
- [2] P. Morville, User Experience Design, Semantic Studios, 2004.
- [3] International Organization for Standardization, ISO 9241-210:2019 – Ergonomics of Human-System Interaction – Human-Centred Design for Interactive Systems, Geneva, 2019.
- [4] K. Moran, Usability (User) Testing 101, Nielsen Norman Group, 2019.
- [5] Mozilla, MDN Curriculum – The essential skillset for new front-end developers, Mozilla, 2024.
- [6] Google, Introduction to Lighthouse (Lighthouse Documentation), Chrome for Developers, 2025.
- [7] World Wide Web Consortium (W3C), Web Content Accessibility Guidelines (WCAG) 2.1, 2018.
- [8] I. Sommerville, Software Engineering, 10th ed., Pearson, 2016.
- [9] Ministerio de Justicia de Chile, Ley N°19.628 sobre Protección de la Vida Privada, Santiago de Chile, 1999.
- [10] International Organization for Standardization, ISO/IEC 27001:2022 – Information Security Management Systems, Geneva, 2022.
- [11] R. Pressman, Ingeniería del Software: Un Enfoque Práctico, 7ª ed., McGraw-Hill, México D.F., 2014.
- [12] C. Kaner, Software Testing: A Craftsman's Approach, CRC Press, Boca Raton, 2006.
- [13] K. McDonald, Agile Q&A: How is Testing Incorporated into Agile Software Development?, Agile Alliance, 2019.
- [14] International Organization for Standardization, ISO/IEC 25010:2011 – Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Product quality model, Geneva, 2011.
- [15] International Organization for Standardization, ISO 9001:2015 – Quality Management Systems – Requirements, Geneva, 2015.
- [16] IEEE, IEEE 829:2008 – Standard for Software and System Test Documentation, New York, 2008.
- [17] Ministerio de Salud de Chile, Hospital Digital: 5 años de vida, 5 millones de prestaciones, Santiago de Chile, 2024.
- [18] Centro Nacional en Sistemas de Información en Salud (CENS), Guía de Mejores Prácticas para la Integración de Interoperabilidad (documento institucional), Santiago de Chile, 2024.
- [19] SaludDigital.com, El avance de la Salud Digital en el sistema de salud en Chile, 2024.
- [20] International Software Testing Qualifications Board (ISTQB), ISTQB Glossary – Quality assurance, s/f.
- [21] International Organization for Standardization, ISO/IEC/IEEE 12207:2017 – Systems and software engineering – Software life cycle processes, Geneva, 2017.
- [22] International Organization for Standardization, ISO/IEC 25010:2023 – Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Product quality model, Geneva, 2023.
- [23] IEEE, IEEE Std 730-2014 – IEEE Standard for Software Quality Assurance Processes, IEEE, 2014.

- [24] J. Brooke, "SUS: A 'Quick and Dirty' Usability Scale", en P. W. Jordan, B. Thomas, B. A. Weerdmeester y I. L. McClelland (eds.), *Usability Evaluation in Industry*, Taylor & Francis, London, 1996.
- [25] International Organization for Standardization, **ISO 9000:2015** – *Quality management systems – Fundamentals and vocabulary*, Geneva, 2015.
- [26] International Organization for Standardization, *Quality management: The path to continuous improvement* (QA vs QC), ISO, 2024.
- [27] International Organization for Standardization / IEC / IEEE, **ISO/IEC/IEEE 29119-1:2022** – *Software and systems engineering – Software testing – Part 1: General concepts*, Geneva, 2022.
- [28] IEEE Standards Association, **IEEE Std 1012-2016** – *System, Software, and Hardware Verification and Validation*, IEEE, 2016.
- [29] International Organization for Standardization / IEC / IEEE, **ISO/IEC/IEEE 29119-1:2022** – *Software and systems engineering – Software testing – Part 1: General concepts*, Geneva, 2022.
- [30] International Organization for Standardization / IEC / IEEE, **ISO/IEC/IEEE 29119-2:2021** – *Software and systems engineering – Software testing – Part 2: Test processes*, Geneva, 2021.
- [31] International Organization for Standardization / IEC / IEEE, **ISO/IEC/IEEE 29119-3:2021** – *Software and systems engineering – Software testing – Part 3: Test documentation*, Geneva, 2021.
- [32] International Software Testing Qualifications Board (ISTQB), *ISTQB Glossary of Testing Terms* (definiciones: test level/stage, system testing, acceptance testing, non-functional testing, risk-based testing), s/f.
- [33] International Organization for Standardization, **ISO 9241-11:2018** – *Ergonomics of human-system interaction – Part 11: Usability: Definitions and concepts*, Geneva, 2018.
- [35] C. M. Barnum, *Usability Testing Essentials: Ready, Set...Test!*, 2nd ed., Morgan Kaufmann, 2020.
- [36] J. S. Dumas y J. C. Redish, *A Practical Guide to Usability Testing*, revised ed., Intellect, 1999.
- [37] J. Nielsen, "Why You Only Need to Test with 5 Users", Nielsen Norman Group, 2000.
- [38] J. Nielsen, "How Many Test Users in a Usability Study?", Nielsen Norman Group, 2012.
- [39] R. Budiu y K. Moran, "How Many Participants for Quantitative Usability Studies: A Summary of Sample-Size Recommendations", Nielsen Norman Group, 2021.
- [40] International Organization for Standardization, **ISO/IEC 25022:2016** – *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Measurement of quality in use*, Geneva, 2016.
- [41] T. Tullis and W. Albert, *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*, 2nd ed., Morgan Kaufmann, 2013.
- [42] J. Brooke, "SUS: a retrospective", *Journal of Usability Studies*, vol. 8, no. 2, 2013.
- [43] A. Bangor, P. Kortum and J. Miller, "Determining what individual SUS scores mean: Adding an adjective rating scale", *Journal of Usability Studies*, vol. 4, no. 3, pp. 114–123, 2009.
- [44] J. R. Lewis, "Item Benchmarks for the System Usability Scale (SUS)", *Journal of User Experience (UXPA)*, 2018.
- [45] M. Hyzy, et al., "System Usability Scale Benchmarking for Digital Health Apps", 2022.



- [46] International Electrotechnical Commission, *IEC 82304-1:2016 – Health software – Part 1: General requirements for product safety*, IEC, 2016.
- [47] International Electrotechnical Commission, *IEC 62304:2006 – Medical device software – Software life cycle processes*, IEC, 2006.
- [48] World Health Organization, *WHO guideline: Recommendations on digital interventions for health system strengthening*, Geneva, 2019.
- [49] World Health Organization, *Global strategy on digital health 2020–2025*, Geneva, 2021.
- [50] National Institute of Standards and Technology (NIST), *Health Information Technology Usability* (initiative overview), NIST, s/f.
- [51] Agency for Healthcare Research and Quality (AHRQ), *Improving Electronic Health Record Usability for Patient Safety*, AHRQ, s/f.
- [52] M. Hyzy et al., "System Usability Scale Benchmarking for Digital Health Apps," *JMIR mHealth and uHealth*, 2022.