

Aplicación Móvil para la Gestión de Reservas del Hotel Miraflores de la ciudad de Ibarra

Plan de SQA

Versión [2.0]

Historia de revisiones

Fecha	Versión	Descripción	Autor
[04/03/2025]	[2.0]	[Segundo Avance]	[Francisco Quinteros, Xavier Torres, Matías Mosquera]

Contenido

1. PROPÓSITO.....	3
2. REFERENCIAS	3
3. GESTIÓN	3
3.1. ORGANIZACIÓN	3
3.2. ACTIVIDADES.....	4
3.2.1. <i>Ciclo de vida del software cubierto por el Plan</i>	<i>4</i>
3.2.2. <i>Actividades de calidad a realizarse</i>	<i>4</i>
3.2.3. <i>Revisar cada producto</i>	<i>4</i>
3.2.4. <i>Revisar el ajuste al proceso.....</i>	<i>5</i>
3.2.5. <i>Realizar Revisión Técnica Formal (RTF).....</i>	<i>5</i>
3.2.6. <i>Asegurar que las desviaciones son documentadas.....</i>	<i>5</i>
3.2.7. <i>Relaciones entre las actividades de SQA y la planificación</i>	<i>5</i>
3.3. RESPONSABLES	5
4. DOCUMENTACIÓN	6
4.1. PROPÓSITO	6
4.2. DOCUMENTACIÓN MÍNIMA REQUERIDA	6
4.2.1. <i>Especificación de requerimientos del software</i>	<i>6</i>
4.2.2. <i>Descripción del diseño del software</i>	<i>8</i>
4.2.3. <i>Plan de Verificación & Validación</i>	<i>8</i>
4.2.4. <i>Reportes de Verificación & Validación</i>	<i>8</i>
4.2.5. <i>Documentación de usuario</i>	<i>9</i>
4.2.6. <i>Plan de Gestión de configuración</i>	<i>9</i>
4.3. OTROS DOCUMENTOS	9
5. ESTÁNDARES, PRÁCTICAS, CONVENCIONES Y MÉTRICAS	10
5.1. ESTÁNDAR DE DOCUMENTACIÓN	10
5.2. ESTÁNDAR DE VERIFICACIÓN Y PRÁCTICAS	10
5.3. OTROS ESTÁNDARES	11
6. REVISIONES Y AUDITORÍAS	11
6.1. OBJETIVO	11
6.2. REQUERIMIENTOS MÍNIMOS	11
6.2.1. <i>Revisión de requerimientos</i>	<i>11</i>
6.2.2. <i>Revisión de diseño preliminar</i>	<i>11</i>
6.2.3. <i>Revisión de diseño crítico.....</i>	<i>11</i>
6.2.4. <i>Revisión del Plan de Verificación & Validación</i>	<i>11</i>
6.2.5. <i>Auditoría funcional</i>	<i>11</i>
6.2.6. <i>Auditoría física.....</i>	<i>11</i>
6.2.7. <i>Auditorías internas al proceso.....</i>	<i>11</i>
6.2.8. <i>Revisiones de gestión</i>	<i>12</i>
6.2.9. <i>Revisión del Plan de gestión de configuración</i>	<i>12</i>
6.2.10. <i>Revisión Post Mortem</i>	<i>12</i>
6.2.11. <i>Agenda.....</i>	<i>12</i>
6.3. OTRAS REVISIONES.....	12
6.3.1. <i>Revisión de documentación de usuario</i>	<i>12</i>
7. VERIFICACIÓN	12
8. REPORTE DE PROBLEMAS Y ACCIONES CORRECTIVAS	12
9. HERRAMIENTAS, TÉCNICAS Y METODOLOGÍAS	12

10. GESTIÓN DE RIESGOS 13

1. Propósito

Este Plan de Aseguramiento de la Calidad del Software (SQA) tiene como objetivo garantizar que el desarrollo de la aplicación móvil para la gestión de reservas del Hotel Miraflores cumpla con los estándares de calidad, funcionalidad y seguridad establecidos. Su propósito es asegurar la correcta implementación de los requisitos del sistema, minimizando errores, garantizando la usabilidad y optimizando la experiencia del usuario.

El Plan cubre el ciclo de vida del software desde la fase de análisis de requerimientos hasta la fase de implementación y validación, pero no contempla la fase de mantenimiento posterior al despliegue.

Los elementos clave de software cubiertos por este Plan incluyen:

- ✓ Código fuente de la aplicación móvil (Ionic/Angular).
- ✓ Base de datos y estructuras de almacenamiento (MySQL/XAMPP).
- ✓ Interfaces de usuario y experiencia del usuario (UI/UX en Figma).
- ✓ Documentación técnica y manuales de usuario.
- ✓ Seguridad del software basada en OWASP Mobile Top 10.

2. Referencias

- ANSI/IEEE Std 730.1-1989, IEEE Standard for Software Quality Assurance Plans.
- OWASP Mobile Top 10, marco de seguridad para aplicaciones móviles.
- Metodología Scrum, utilizada para el desarrollo ágil del proyecto.

3. Gestión

3.1. Organización

El equipo de calidad estará compuesto por los siguientes roles:

Rol	Responsabilidades
Responsable de SQA (Francisco Quinteros)	Supervisión del cumplimiento de los estándares de calidad del software.
Equipo de Desarrollo (Matías Mosquera)	Implementación de cambios y correcciones con base en los hallazgos de SQA.
Equipo de Pruebas (Xavier Torres)	Ejecución de pruebas de verificación, validación y reporte de fallos.

El **Responsable de SQA** coordinará las actividades de control de calidad en cada fase del desarrollo y asegurará que los procedimientos sean seguidos por el equipo.

3.2. Actividades

3.2.1. Ciclo de vida del software cubierto por el Plan

Este Plan de SQA cubre las siguientes etapas del ciclo de vida del software:

- ✓ Fase de análisis de requerimientos: Definición y documentación de requisitos.
- ✓ Fase de diseño: Creación de arquitecturas de software y diseños UI/UX.
- ✓ Fase de implementación: Desarrollo del sistema y codificación del software.
- ✓ Fase de pruebas y validación: Verificación de que el software cumple con los requisitos establecidos.

Los productos de software que tendrán revisiones de calidad incluyen:

- ✓ Código fuente y estructura de base de datos.
- ✓ Documentación del sistema y manuales de usuario.
- ✓ Interfaces gráficas y flujos de interacción del usuario.
- ✓ Planes de pruebas y reportes de evaluación de calidad.

3.2.2. Actividades de calidad a realizarse

Actividad	Descripción
Revisión de cada producto	Evaluación de los productos entregables mediante checklists y reportes de calidad.
Revisión del ajuste al proceso	Evaluación del cumplimiento de la metodología Scrum en cada iteración del proyecto.
Realización de Revisiones Técnicas Formales (RTF)	Identificación temprana de defectos en los productos mediante reuniones de revisión.
Documentación de desviaciones	Registro y seguimiento de errores encontrados, asegurando su corrección en la siguiente iteración.

3.2.3. Revisar cada producto

Se evaluarán los productos clave del proyecto utilizando listas de verificación y reportes de calidad. Se revisará que no queden correcciones sin resolver en los informes de revisión previos y, en caso de encontrar alguna pendiente, será incluida en la siguiente revisión.

Se utilizarán checklists definidas para cada producto, verificando que cumplan con los estándares de calidad y los requisitos establecidos.

Como salida de esta actividad, se generará el Informe de Revisión de SQA, el cual será distribuido a los responsables del producto para asegurar que sean conscientes de las desviaciones o discrepancias encontradas y tomen medidas correctivas oportunas.

3.2.4. Revisar el ajuste al proceso

Se verificará que cada fase del desarrollo cumpla con la metodología Scrum. Se revisarán los productos clave para asegurar el cumplimiento de las actividades definidas en el proceso de desarrollo. Esta revisión se basará en documentos como:

- ✓ Plan del Proyecto
- ✓ Plan de la Iteración
- ✓ Plan de Verificación

Antes de cada revisión, se verificará que las desviaciones detectadas en informes previos hayan sido corregidas. Como resultado, se generará un Informe de Revisión de SQA que será compartido con los responsables de cada actividad.

3.2.5. Realizar Revisión Técnica Formal (RTF)

Las Revisiones Técnicas Formales (RTF) se realizarán para identificar fallos en la funcionalidad, lógica o implementación del software. Se analizará si el producto cumple con sus especificaciones y estándares definidos.

Se convocará a reuniones de revisión con el equipo de desarrollo y el Responsable de SQA, en las cuales se evaluarán los productos en función de listas de verificación y estándares de calidad. La duración de estas reuniones no debe exceder las dos horas.

Como salida, se generará el Informe de RTF con las observaciones encontradas y las correcciones necesarias.

3.2.6. Asegurar que las desviaciones son documentadas

Todas las desviaciones detectadas en los productos y en las actividades del proceso de desarrollo serán registradas y documentadas. Se verificará que los responsables de cada plan realicen modificaciones cuando sea necesario, asegurando la mejora continua del producto.

Se aplicará un procedimiento formal para la documentación de desviaciones, garantizando su seguimiento y corrección en la siguiente iteración.

3.2.7. Relaciones entre las actividades de SQA y la planificación

Cada actividad de calidad será integrada dentro del cronograma del proyecto. A continuación, se presenta una planificación tentativa:

Actividad	Semana de Ejecución
Planificación de calidad	Semana 1
RTF de estimaciones y mediciones	Semana 6
Validación de prototipo	Semana 8
Pruebas de regresión	Semana 10
Auditoría final de calidad	Semana 12

4. Documentación

4.1. Propósito

El propósito de esta sección es identificar la documentación requerida para garantizar la calidad del desarrollo del software, cubriendo aspectos de verificación y validación (V&V), uso y mantenimiento del sistema.

La documentación definida en este plan permitirá:

- ✓ Asegurar que la implementación del software satisface los requisitos establecidos.
- ✓ Proporcionar un mecanismo formal de revisión y validación de cada fase del ciclo de desarrollo.
- ✓ Cumplir con estándares de calidad de software (IEEE, OWASP, ISO 25010).

Se establecen criterios de revisión y metodologías para confirmar la consistencia de los documentos clave, garantizando la correcta trazabilidad de los procesos de calidad.

4.2. Documentación mínima requerida

La documentación mínima es la requerida para asegurar que la La documentación mínima es aquella imprescindible para garantizar que el sistema cumple con los requisitos del proyecto. Incluye:

- ✓ Especificación de Requerimientos del Software.
- ✓ Descripción del Diseño del Software.
- ✓ Plan de Verificación y Validación (V&V).
- ✓ Reportes de Pruebas y Verificación.
- ✓ Manual de Usuario y Documentación Técnica.

4.2.1. Especificación de requerimientos del software

El documento de Especificación de Requerimientos del Software (SRS) debe detallar de manera clara y precisa cada uno de los requerimientos funcionales y no funcionales del sistema.

Los requisitos deben cumplir con los siguientes criterios:

- ✓ Completitud: No deben existir elementos sin especificar ni dependencias omitidas.
- ✓ Consistencia: No debe haber contradicciones entre los requerimientos.
- ✓ Claridad y no ambigüedad: Cada término técnico debe estar definido en un glosario del proyecto.
- ✓ Verificabilidad: Debe ser posible validar que el producto final cumple con cada requisito.
- ✓ Trazabilidad: Cada requerimiento debe estar vinculado a su correspondiente prueba de validación.

Los atributos de calidad de la aplicación del Hotel Miraflores se clasifican en:

Funcionalidad:

- ✚ Adecuación a las necesidades del usuario.
- ✚ Precisión en la gestión de reservas.
- ✚ Seguridad en la manipulación de datos personales.

Confiabilidad:

- ✚ Recuperación ante fallos del sistema.
- ✚ Registro de auditorías de cambios en reservas.

Usabilidad:

- ✚ Interfaz intuitiva y de fácil aprendizaje.
- ✚ Compatibilidad con dispositivos móviles Android e iOS

Eficiencia:

- ✚ Optimización de tiempos de respuesta.
- ✚ Baja carga en el consumo de recursos del servidor.

Mantenibilidad:

- ✚ Código modular y bien documentado.
- ✚ Facilidad de actualización y escalabilidad del sistema.

- ✓ Cumplimiento con OWASP Mobile Top 10 para evitar vulnerabilidades en la gestión de datos.

4.2.2. Descripción del diseño del software

Este documento especifica cómo se implementará el software para satisfacer los requerimientos.

- ✚ Estructura del sistema: Modelos UML (Casos de uso, Diagramas de Clases y Secuencia).
- ✚ Arquitectura del software: Definición de capas (Front-end, Back-end, Base de Datos).
- ✚ Diseño de la Interfaz de Usuario (UI/UX) en Figma.
- ✚ Modelo de datos: Relación de tablas en MySQL y esquema de API REST.

Este documento debe ser revisado en cada iteración para verificar que cumple con los estándares de calidad definidos.

4.2.3. Plan de Verificación & Validación

El Plan de Verificación y Validación describe los métodos que se utilizarán para comprobar que el software satisface los requerimientos.

✓ **Verificación:** Se validará que cada módulo del sistema cumpla con las especificaciones de diseño y desarrollo.

✓ **Validación:** Se ejecutarán pruebas funcionales para comprobar que el software responde a las necesidades del usuario.

Las actividades de V&V incluirán:

- ✚ Revisión de especificaciones → Asegurar que todos los requisitos han sido documentados y aprobados.
- ✚ Validación del diseño → Confirmar que el sistema propuesto satisface los requerimientos.
- ✚ Pruebas de aceptación → Verificar que el software implementado cumple con los requisitos del cliente.

4.2.4. Reportes de Verificación & Validación

Estos documentos especificarán los **resultados de las pruebas realizadas** en el sistema.

- ✚ **Informe de Pruebas Funcionales:** Evaluación del cumplimiento de los requisitos.
- ✚ **Informe de Pruebas de Seguridad:** Detección de vulnerabilidades con OWASP Mobile Top 10.

📌 **Informe de Pruebas de Usabilidad:** Feedback de usuarios y correcciones aplicadas.

Estos reportes serán utilizados para la toma de decisiones en cada iteración de desarrollo.

4.2.5. Documentación de usuario

El Manual de Usuario debe proporcionar instrucciones detalladas sobre el uso de la aplicación, incluyendo:

- 📌 Descripción de funcionalidades (gestión de reservas, pagos, cancelaciones).
- 📌 Guías paso a paso para la navegación dentro de la aplicación.
- 📌 Listado de errores comunes y sus soluciones.
- 📌 Contacto de soporte técnico y atención al cliente.

✓ Se garantizará que el manual sea claro y accesible para los clientes del Hotel Miraflores.

4.2.6. Plan de Gestión de configuración

Este documento establecerá las metodologías utilizadas para:

- 📌 **Identificar componentes del software** (código fuente, bases de datos, documentación).
- 📌 **Control de cambios y versionado** (GitHub/GitLab).
- 📌 **Registro de versiones y auditorías de configuración.**
- 📌 **Estrategia de recuperación ante fallos.**

✓ Todo el código será gestionado bajo control de versiones en Git con seguimiento de cambios.

4.3. Otros documentos

Además de la documentación principal, se incluirán los siguientes documentos complementarios:

- 📌 **Plan de Desarrollo** → Estrategia y fases del desarrollo del software.
- 📌 **Plan de Proyecto** → Cronograma y asignación de tareas.
- 📌 **Manual de Estándares y Procedimientos** → Normativas de codificación, seguridad y calidad.
- 📌 **Checklist de Revisión de Código** → Evaluación del cumplimiento de buenas prácticas.

- ✓ Todos los documentos estarán **versionados y almacenados** en repositorios seguros.

5. Estándares, prácticas, convenciones y métricas

Esta sección identifica los estándares, prácticas, convenciones y métricas que serán aplicadas para la evaluación de la calidad del software, así como los mecanismos para monitorear y asegurar su cumplimiento.

- ✓ **Uso de estándares IEEE para documentación y pruebas.**
- ✓ **Métricas de rendimiento y usabilidad para evaluar la eficiencia del sistema.**
- ✓ **Cumplimiento con OWASP Mobile Top 10** para mitigar vulnerabilidades de seguridad en aplicaciones móviles.

5.1. Estándar de documentación

Se definirán dos documentos base para la documentación del software:

Estándar de Documentación Técnica:

- Debe ser comprensible y facilitar el mantenimiento del producto.
- Incluir fuentes, diagramas UML y descripciones de modelos de datos.

Estándar de Documentación de Usuario:

- Debe proporcionar instrucciones claras sobre el uso de la aplicación.
- Incluir ejemplos, imágenes y guías paso a paso.

Para la escritura de documentos, se definirán **plantillas estándar** que incluyen:

- ✓ **Encabezado y pie de página con identificación del documento.**
- ✓ **Fuente y tamaño de texto unificados para consistencia.**
- ✓ **Datos mínimos requeridos:** Fecha, versión y responsables del documento.

5.2. Estándar de verificación y prácticas

- ✓ Se seguirán las prácticas definidas en el **Plan de Verificación y Validación (V&V)**.
- ✓ Se aplicará el estándar **IEEE Std 1012-1986** para la planificación de pruebas.
- ✓ Se implementarán revisiones técnicas formales para garantizar calidad del código y cumplimiento con los requerimientos funcionales.

5.3. Otros Estándares

- ✓ **ISO 25010:** Se utilizarán métricas para evaluar la calidad del software en términos de **funcionalidad, rendimiento, seguridad y usabilidad**.
- ✓ **ISO 27001:** Se establecerán controles de seguridad de la información para proteger los datos del sistema.
- ✓ **Scrum Guide:** Se empleará la metodología ágil **Scrum** para la gestión iterativa del desarrollo del software.

6. Revisiones y auditorías

6.1. Objetivo

Se realizarán revisiones y auditorías para evaluar la calidad del sistema en términos de **diseño, funcionalidad, seguridad y gestión del proyecto**.

- ✓ **Auditorías internas al código** para garantizar su estabilidad y seguridad.
- ✓ **Pruebas de seguridad** para detectar vulnerabilidades antes de la implementación final.

6.2. Requerimientos mínimos

6.2.1. Revisión de requerimientos

Se verificará que todos los requisitos especificados por el cliente hayan sido documentados y validados.

6.2.2. Revisión de diseño preliminar

Se analizará la consistencia del diseño técnico antes de su implementación.

6.2.3. Revisión de diseño crítico

Se validará que la arquitectura del software es escalable, segura y cumple con los requerimientos.

6.2.4. Revisión del Plan de Verificación & Validación

Se confirmará que las pruebas cubren todas las funcionalidades esenciales.

6.2.5. Auditoría funcional

Se verificará que todas las funcionalidades han sido implementadas correctamente antes de la entrega final.

6.2.6. Auditoría física

Se revisará que el software y la documentación técnica sean consistentes antes de su liberación.

6.2.7. Auditorías internas al proceso

Se evaluará que el código, el diseño y las pruebas cumplen con los estándares definidos.

6.2.8. Revisiones de gestión

Se realizarán reuniones de control de calidad para garantizar que el proyecto se mantiene alineado con los objetivos.

6.2.9. Revisión del Plan de gestión de configuración

Se validará que los métodos de control de versiones y configuración sean efectivos.

6.2.10. Revisión Post Mortem

Se evaluarán los resultados del proyecto y se documentarán aprendizajes para futuras implementaciones.

6.2.11. Agenda

Se definirán reuniones periódicas para revisar avances y detectar posibles fallos en el desarrollo del software.

6.3. Otras revisiones

6.3.1. Revisión de documentación de usuario

Se verificará que la documentación es clara, completa y fácil de entender para los usuarios finales.

7. Verificación

Se implementarán los siguientes procesos para garantizar que el software cumple con los requisitos funcionales y no funcionales:

- ✓ **Pruebas unitarias y de integración** para evaluar la funcionalidad de cada módulo.
- ✓ **Pruebas de carga y rendimiento** para asegurar la escalabilidad del sistema.
- ✓ **Evaluación de seguridad con OWASP Mobile Top 10** para detectar vulnerabilidades.

8. Reporte de problemas y acciones correctivas

Se documentarán todos los errores encontrados en el sistema y se aplicará un proceso de corrección antes de la entrega final.

- ✓ Registro de problemas en una plataforma de gestión de incidencias (JIRA, Trello).
- ✓ Seguimiento de correcciones en cada iteración del desarrollo.
- ✓ Validación final del sistema antes de la entrega.

9. Herramientas, técnicas y metodologías

Herramientas:

- ✓ **Ionic y Angular** para el desarrollo de la aplicación.

- ✓ **MySQL y XAMPP** para la base de datos.
- ✓ **Postman** para pruebas de API REST.
- ✓ **GitHub/GitLab** para control de versiones y gestión del código.

Técnicas:


- ✓ **Pruebas automatizadas** con herramientas como Cypress o Selenium.
- ✓ **Revisiones de código por pares** para validar estándares de calidad.


Metodología:


- ✓ **Scrum** para iteraciones ágiles y mejora continua en el desarrollo del software.

10. Gestión de riesgos

Se identifican y mitigan los siguientes riesgos potenciales para el sistema:

 **Seguridad:** Posibles ataques cibernéticos que comprometan los datos de clientes.

 **Usabilidad:** Complejidad en la interfaz que afecte la experiencia del usuario.

 **Rendimiento:** Fallas en la escalabilidad de la aplicación en momentos de alta demanda.

Estrategias de mitigación:

- ✓ Implementación de **cifrado AES-256** y autenticación multifactor (2FA) para la seguridad del sistema.
- ✓ Pruebas de usuario para optimizar la experiencia de navegación.
- ✓ Monitoreo de rendimiento con herramientas de análisis en tiempo real.