**Documento de aseguramiento de calidad**

**Proyecto MindSoft**

**Versión 1.0**

**Control de versiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| **18/09/2024** | **1.0** | **Creación del documento.** | **Victor Caceres** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **Índice**

[**Índice 3**](#_q5noaarc639j)

[**1. Introducción 4**](#_q0vn4ea1dwnc)

[**2. Objetivos del QA 4**](#_1z9qald14zgd)

[**3. Alcance 4**](#_hbr0sggqqkys)

[**4. Metodología de QA 4**](#_tor0z5pw8r3y)

[**5. Checklist de Revisión 4**](#_s9jdrltt23hp)

[**6. Plan de Revisión y Validación 7**](#_ulor1qsdmff7)

[**7. Identificación de Riesgos 7**](#_257zu88zdn4f)

[**8. Conclusiones 7**](#_e5a3x88yy2uy)

[**9. Documentación y Control de Versiones 7**](#_bkidgqkyme9)

# 

# **1. Introducción**

Este documento establece los criterios de calidad y los procesos de revisión para los entregables de la etapa de análisis del proyecto MindSoft, una aplicación diseñada para estudiantes universitarios enfocada en el monitoreo de su bienestar emocional. Asegurar la calidad de los documentos es esencial para cumplir con las expectativas del público objetivo y facilitar el desarrollo de una herramienta eficaz que contribuya a la salud mental de los usuarios.

# **2. Objetivos del QA**

* Asegurar la calidad de los documentos generados en la fase de análisis.
* Verificar que los requisitos y entregables estén completos y alineados con las expectativas del proyecto.

# **3. Alcance**

* Este documento cubre los siguientes entregables:
  + Project Charter
  + Documento de Negocio
  + Historias de Usuario (HU01 - HU07)
  + Documento de Especificaciones de la UI
  + Documento de Guía de Estilos
  + Documento de Especificación de la Base de Datos
  + Documento de Arquitectura de Software
  + Documento de Análisis de Requerimientos

# **4. Metodología de QA**

* **Revisión por Pares**: Realizar sesiones de revisión entre miembros del equipo.
* **Checklist de Revisión**: Utilizar una lista de verificación para cada documento.
* **Entrevistas de Validación**: Reuniones con stakeholders para validar las historias de usuario.

# **5. Checklist de Revisión**

## **Project Charter**

* Estructura clara (secciones).
* El nombre del proyecto representa adecuadamente el propósito.
* La definición del producto es precisa.
* Los objetivos están bien definidos y son medibles.
* Los hitos están claramente definidos y las fechas son realistas.
* Las partes interesadas y su rol están correctamente identificados.
* Ortografía del documento.
* Control de cambios documentado.

## **Documento de Negocio**

* Estructura clara (secciones).
* Los objetivos están bien definidos y son medibles.
* Los procesos están claramente definidos.
* La descripción de actividades está correctamente definida.
* Ortografía del documento.

**Observación:** Se encuentran faltas ortográficas en el documento de negocio.

* Control de cambios documentado.

## **Historia de Usuario 01**

* Estructura clara (secciones).
* Se tiene una definición clara de la historia de usuario.
* Se incluye el formato “Como [rol], quiero [acción] para [beneficio]”.
* Prioridad de la historia establecida.
* Validaciones de la historia establecida.
* Control de cambios documentado.

## **Historia de Usuario 02**

* Estructura clara (secciones).
* Se tiene una definición clara de la historia de usuario.
* Se incluye el formato “Como [rol], quiero [acción] para [beneficio]”.
* Prioridad de la historia establecida.
* Validaciones de la historia establecida.
* Control de cambios documentado.

## **Historia de Usuario 03**

* Estructura clara (secciones).
* Se tiene una definición clara de la historia de usuario.
* Se Incluye el formato "Como [rol], quiero [acción] para [beneficio]".
* Prioridad de la historia establecida
* Control de cambios documentado.

## **Historia de Usuario 04**

* Estructura clara (secciones).
* Se tiene una definición clara de la historia de usuario.
* Se incluye el formato “Como [rol], quiero [acción] para [beneficio]”.
* Prioridad de la historia establecida.
* Validaciones de la historia establecida.
* Control de cambios documentado.

## **Historia de Usuario 05**

* Estructura clara (secciones).
* Se tiene una definición clara de la historia de usuario.
* Se Incluye el formato "Como [rol], quiero [acción] para [beneficio]".
* Prioridad de la historia establecida
* Control de cambios documentado.

## **Historia de Usuario 06**

* Estructura clara (secciones).
* Se tiene una definición clara de la historia de usuario.
* Se Incluye el formato "Como [rol], quiero [acción] para [beneficio]".

* Prioridad de la historia establecida.
* Validaciones de la historia establecida.
* Control de cambios documentado.

## **Historia de Usuario 07**

* Estructura clara (secciones).
* Se tiene una definición clara de la historia de usuario.
* Se incluye el formato “Como [rol], quiero [acción] para [beneficio]”.
* Prioridad de la historia establecida.
* Validaciones de la historia establecida.
* Control de cambios documentado.

## **Documentos de Especificaciones de la UI**

* Estructura clara (índice, secciones).
* Los objetivos están bien definidos y medibles.
* Los requisitos son claros y detallados.
* El diseño es coherente y sigue las pautas del diseño establecido.
* Los componentes de la interfaz están bien explicados, incluyendo sus funcionalidades.
* Se consideran principios de accesibilidad y usabilidad en el diseño propuesto.
* Se muestran referencias visuales para cada componente.
* Control de cambios documentado.

## **Documento de Guía de Estilos**

* Estructura clara (índice, secciones).
* Los objetivos están bien definidos y son medibles.
* Términos clave definidos.
* Iconografía bien definida.
* Paleta de colores.

**Observación:** Los colores inicialmente elegidos no son tanconvenientes para una aplicación de monitoreo de salud mental.

* Control de cambios documentado.

## **Documento de Especificación de la Base de Datos**

* Estructura clara (índice, secciones).
* El sistema de gestión de base de datos está especificado.
* Las tablas de la base de datos están detalladas.
* El modelo físico de la base de datos está bien elaborado.

**Observación**: Los nombres de los atributos y de las tablas están en inglés y español. Se debe seguir el estándar para nombrar todo en inglés.

* El diccionario de datos está bien elaborado.

**Observación**: Algunos nombres de atributos no coinciden con el modelo lógico de la base de datos. Como el caso de “id\_usuario” en la tabla “user”

* Controles técnicos de seguridad especificados.
* Control de cambios documentado.

## **Documento de Arquitectura de Software**

* Estructura clara (índice, secciones).
* La arquitectura de software utilizada está bien detallada.
* Coherencia y concordancia con el proyecto a realizar.   
  **Observación:** Se describe en la página 4 que es para aplicación web, cuando el proyecto es una aplicación móvil. HTML y CSS no corresponden.
* La vista física está bien elaborada.
* La vista lógica está bien elaborada.
* La vista de despliegue está bien elaborada.  
  **Observación:** El diagrama que corresponde en esta vista es el diagrama de componentes, no el de despliegue.
* La vista de procesos está bien elaborada.  
  **Observación:** El diagrama de casos de uso debe ser actualizado, el diagrama mostrado no es el que se está trabajando actualmente.
* Control de cambios documentado.
* Ortografía del documento.

## **Documento de Análisis de Requerimientos**

* Estructura clara (índice, secciones).
* Objetivo del documento claramente definido
* Correcto desarrollo del diagrama de casos de uso
* Descripción de Casos de Uso
* Detalle de los Requerimientos Funcionales
* Detalle de los Requerimientos No Funcionales  
  **Observación:** No se detalla el requerimiento de “Funcionalidad”, propuesto en el Project Charter
* Los prototipos son coherentes con el caso de uso especificado
* Control de cambios documentado.

# **6. Plan de Revisión y Validación**

* **Frecuencia de Revisiones**: Mensual
* **Métodos de Revisión**: Revisión de documentos.

# **7. Identificación de Riesgos**

* **Riesgos Asociados al Análisis**:
  + Riesgo 1: Ambigüedad en los Requisitos

Descripción: La falta de claridad en los requisitos funcionales y no funcionales puede resultar en malentendidos y en el desarrollo de características que no cumplen con las expectativas del usuario.

Estrategia de mitigación: Realizar revisiones periódicas con los stakeholders para garantizar que los requisitos estén bien definidos y acordados.

* + Riesgo 2: Faltas en Documentación

Descripción: Errores ortográficos y de redacción en la documentación pueden afectar al proyecto y causar confusiones entre los miembros del equipo.

Estrategia de mitigación: Implementar un proceso de revisión por pares en todos los documentos antes de su finalización.

* + Riesgo 3: Inconsistencias en la Nomenclatura

Descripción: La utilización de nomenclaturas no estandarizadas puede generar confusión y dificultar la comunicación entre los miembros del equipo.

Estrategia de mitigación: Establecer una guía de estilo que defina estándares de nomenclatura y presentación que todos los miembros del equipo deben seguir.

# **8. Conclusiones**

Las conclusiones del proceso de aseguramiento de la calidad (QA) para el proyecto MindSoft revelan que, aunque la mayoría de los documentos cumplen con los criterios establecidos, existen áreas clave que requieren atención. Es fundamental corregir las faltas ortográficas y estandarizar la nomenclatura en todos los documentos para mantener un nivel profesional adecuado. Además, se debe ajustar la guía de estilos y los diagramas de arquitectura para asegurar que reflejen correctamente el enfoque del proyecto, que es una aplicación móvil. También es necesario detallar los requisitos de funcionalidad, ya que esto impacta directamente en la implementación. Abordar estas correcciones es esencial para asegurar que MindSoft cumpla con sus objetivos de manera efectiva y proporcione un sólido apoyo a los estudiantes en su bienestar emocional.

# **9. Documentación y Control de Versiones**

* Cada miembro del equipo sube los documentos revisados al repositorio del proyecto en **GitHub**, utilizando el sistema de control de versiones **Git**. Esto garantiza que todas las versiones realizadas durante el proceso de revisión queden correctamente registradas y que cada cambio pueda ser rastreado de manera eficiente. El uso de Git permite gestionar las diferentes versiones de los documentos, facilitando la colaboración, el control de cambios y la trazabilidad de las modificaciones, lo que asegura que cada integrante pueda acceder y revisar el historial completo de ediciones en cualquier momento.