

DOCUMENTO DE ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE PARA EL SISTEMA DZONOTMAP

Versión 0.1

3 de abril de 2022

Realizó:

- **Vicente Nava Montoya**
- **Rodrigo Urtecho Quintal**
- **Raúl Antonio Villanueva Hernández**

CONTROL DE DOCUMENTACIÓN

Control de Configuración

Título:	Especificación de Requerimientos para el sistema DzonotMap
Autores:	<ul style="list-style-type: none">• Vicente Nava Montoya• Rodrigo Urtecho Quintal• Raúl Antonio Villanueva Hernández
Fecha:	3 de abril de 2022

Histórico de versiones

Versión	Fecha	Estado	Responsable	Nombre de archivo
0.1	03/04/2022	B	Vicente Nava Montoya	ERS.docx

Estado: (B)orrador, (R)evisión, (A)probado

Histórico de cambios

Versión	Fecha	Cambios

CONTENIDO

1 DESCRIPCIÓN GENERAL	4
1.1 PERSPECTIVA DEL PRODUCTO.	4
1.1.1 <i>Interfases del sistema.</i>	4
1.1.2 <i>Interfases de usuario.</i>	4
1.1.3 <i>Interfases de hardware.</i>	4
1.1.4 <i>Interfases de software.</i>	5
1.1.5 <i>Interfases de comunicación.</i>	5
1.1.6 <i>Restricciones de memoria.</i>	5
1.1.7 <i>Operaciones.</i>	5
1.1.8 <i>Requerimientos de adaptación del sitio.</i>	¡Error! Marcador no definido.
1.2 FUNCIONES DEL PRODUCTO.	¡ERROR! MARCADOR NO DEFINIDO.
1.3 USUARIOS.	6
1.3.1 <i>Características Generales.</i>	6
1.3.2 <i>Clases.</i>	6
1.4 RESTRICCIONES GENERALES.	6
1.5 SUPUESTOS Y DEPENDENCIAS.	6
1.6 ASIGNACIÓN DE REQUERIMIENTOS.	¡ERROR! MARCADOR NO DEFINIDO.
2 REQUERIMIENTOS ESPECÍFICOS	7
2.1 INTERFASES EXTERNAS.	¡ERROR! MARCADOR NO DEFINIDO.
2.2 FUNCIONES (REQUERIMIENTOS FUNCIONALES).	7
2.3 DESEMPEÑO DE REQUERIMIENTOS (REQUERIMIENTOS NO FUNCIONALES).	8
2.4 REQUERIMIENTOS DE BASES DE DATOS LÓGICAS.	¡ERROR! MARCADOR NO DEFINIDO.
2.5 RESTRICCIONES DE DISEÑO.	¡ERROR! MARCADOR NO DEFINIDO.
2.5.1 <i>Estándares de conformidad.</i>	¡Error! Marcador no definido.
2.6 DOCUMENTACIÓN DE USUARIO	9
2.7 ATRIBUTOS DEL SISTEMA DE SOFTWARE.	9
2.7.1 <i>Confiabilidad.</i>	9
2.7.2 <i>Disponibilidad.</i>	¡Error! Marcador no definido.
2.7.3 <i>Seguridad.</i>	9
2.7.4 <i>Mantenimiento.</i>	10
2.7.5 <i>Portabilidad.</i>	10
3 APÉNDICES	11

1 DESCRIPCIÓN GENERAL

<Se describen los factores generales que afectan al producto y sus requerimientos. No se deben enunciar requerimientos específicos, sino que se mencionarán los antecedentes generales de éstos (requerimientos de alto nivel). Todos los requerimientos serán definidos a detalle en la sección 3. No es necesario que exista texto entre este punto y el siguiente.>

Perspectiva del producto.

<Se pone al producto en perspectiva con otros productos o proyectos. Si el producto es independiente y totalmente autocontenido, si es una nueva versión y/o se va a sustituir algún otro, aquí debe establecerse. Si se define un producto que es componente de uno más grande entonces se deben describir los componentes del sistema o proyecto mayor e identificar las interfases entre dicho sistema y el software que se está especificando.

Un diagrama de bloques mostrando los componentes más importantes del sistema más grande, sus interconexiones y las interfases externas puede ser de mucha utilidad.

Identificar las principales interfases externas del producto de software que se está especificando (la descripción no debe ser detallada) y describir el hardware de computadoras y el equipo periférico a ser usado (esta es una descripción global).

Se deben comentar las razones por las que ciertas restricciones de diseño serán especificadas más adelante en la ERS.>

Interfases del sistema.

<Se lista cada interfaz del sistema y se identifica la funcionalidad del software para lograr que el requerimiento del sistema y la descripción de la interfase correspondan. Esta interfase es tanto hacia dentro como hacia fuera.>

Interfaces de usuario.

<Se debe especificar lo siguiente:

a) Las características lógicas de cada interfaz entre el producto de software y sus usuarios. Esto incluye aquellas características de configuración (por ejemplo, formatos de pantalla, esquemas de páginas o ventanas, contenido de algún reporte o menú, formatos de mensajes, etc.), necesarias para obtener los requerimientos de software.

b) Todos los aspectos de optimización de la interfase con la persona que usa el sistema. Esto es, incluir una lista de qué hace y qué no hace el sistema desde la perspectiva del usuario.>

Interfaces de hardware.

<Se especifican las características lógicas de cada interfaz entre el producto de software y los componentes de hardware del sistema. Esto incluye características de

configuración (números de puertos, conjuntos de instrucciones, etc.). Se mencionan los dispositivos que pueden ser soportados, cómo son soportados, y los protocolos.>

Interfaces de software.

<Se especifica el uso de otros productos de software requeridos (por ejemplo sistemas manejadores de datos, sistema operativo o alguna utilidad específica) y las interfaces con otras aplicaciones desarrolladas (por ejemplo con el sistema de contabilidad). Para cada producto de software hay que especificar:

- *Nombre.*
- *Mnemónico.*
- *Número de especificación.*
- *Número de versión.*
- *Fuente (quien lo provee, lugar donde está el código).*

Para cada interfase hay que:

- *Explicar el tipo de interacción del software relacionado con este producto de software.*
- *Definición de la interfase en términos del contenido y formato del mensaje. No es necesario detallar alguna interfase bien documentada, pero sí hacer referencia al documento que define la interfase requerida.>*

El sistema hará uso de la API de Google Maps para posibilitar la visualización de cenotes en el mapa.

Interfaces de comunicación.

<Se deben especificar todas las interfaces de comunicación tales como los protocolos de la red local, etc.>

Restricciones de memoria.

<Se debe especificar alguna característica aplicable y los límites de la memoria primaria y secundaria.>

Operaciones.

<Se especifican las operaciones normales y especiales requeridas por el usuario tales como:

- *Las distintas formas o modos de operación en la organización del usuario.*
- *Períodos de operación interactiva y períodos de operaciones desatendidas.*
- *Funciones que soporten procesamiento de datos.*
- *Operaciones de respaldo y recuperación.*

Nota: Esto es especificado algunas veces como parte de la sección Interfaces de Usuario.>

Usuarios.

Características Generales.

<Se describen las características generales del usuario deseado del producto, se incluye su nivel de estudios, experiencia y su profesionalismo técnico. No se debe utilizar para enunciar requerimientos específicos, pero se deben dar razones del por qué ciertos requerimientos son especificados más adelante (sección 3).>

Clases.

<Identificar las diversas clases de usuarios que usarán el producto. Las clases pueden ser diferenciadas por frecuencia de uso, subconjunto de funciones utilizadas, experiencia técnica, niveles de privilegios o seguridad, nivel de educación o experiencia. Distinguir los más importantes de los menos.>

Restricciones generales.

<Provee una descripción general de otros productos que limiten las opciones de los desarrolladores. Estas son:

- *Políticas regulatorias.*
- *Limitaciones de hardware.*
- *Interfases con otras aplicaciones.*
- *Operaciones paralelas.*
- *Funciones de auditoría.*
- *Funciones de control.*
- *Requerimientos de lenguajes de alto nivel.*
- *Protocolos de comunicación.*
- *Lo crítico de la aplicación.*
- *Consideraciones de seguridad.>*

Supuestos y dependencias.

<Se listan cada uno de los factores que afecten los requerimientos. Estos factores no imponen restricciones de diseño al software, sino que los cambios a ellos (factores), pueden afectar los requerimientos. Los supuestos son contrarios a los hechos, por ejemplo, el uso de componentes de terceros, componentes comerciales que se planeen usar, temas relacionados al ambiente de desarrollo (equipo, SO, etc).>

2 REQUERIMIENTOS ESPECÍFICOS

<Debe contener todos los requerimientos de software a un nivel suficiente de detalle que permita, a los desarrolladores, diseñar sistemas que los satisfagan, y que las personas encargadas de la verificación, puedan probar el sistema usándolos.

A través de esta sección, cada requerimiento especificado deberá ser externamente percibible por usuarios, operadores y otros sistemas externos. Estos requerimientos deben incluir una descripción mínima de cada entrada al sistema, cada salida del sistema, y todas las funciones que se realicen en el sistema como respuesta a una entrada o soporte de una salida. Como esta es la parte mas grande e importante de una ERS, se deben aplicar los siguientes principios:

- Los requerimientos específicos se deben enunciar de acuerdo a las características siguientes: correctos, no ambiguos, completos, consistentes, ordenados por importancia y/o estabilidad, verificables, modificables y rastreables.*
- Los requerimientos específicos deben tener referencia cruzada con documentos relacionados.*
- Todos los requerimientos deben ser identificados de manera única.*
- Poner atención cuidadosa a la organización de los requerimientos para obtener una máxima legibilidad.*
- Se debe mencionar al dueño del requerimiento.*
- Los requerimientos deben ser de carácter cuantitativo y no cualitativo y capaces de ser probados. Cada requerimiento debe ser presentado con su criterio de aceptación.*
- Se debe de justificar el requerimiento, diciendo el por qué del mismo.*

No es necesario que exista texto entre este punto y el siguiente.>

Funciones (requerimientos funcionales).

<Los requerimientos funcionales deben definir acciones fundamentales que tomarán lugar en el software, en la aceptación y procesamiento de entradas y en el procesamiento y generación de salidas. Se incluyen las siguientes:

- a) Validación de entradas.*
 - b) Secuencias de operación exactas.*
 - c) Respuestas a situaciones anormales, incluyendo:*
 - 1. Sobreflujo.*
 - 2. Facilidades de comunicación.*
 - 3. Manipulación y recuperación de errores.*
 - d) Efectos de parámetros.*
 - e) Relación de salidas con las entradas, incluyendo:*
 - 1. Entrada/salidas de secuencias.*
 - 2. Fórmulas de conversión de entrada/salida.*
-

Puede ser apropiado dividir los requerimientos funcionales en subfunciones o subprocesos. Esto no implica que el proceso de diseño de software deba ser dividido también.>

1. La aplicación debe dar la posibilidad al usuario de acceder al sistema con una cuenta de usuario o como invitado.
Si el usuario selecciona la opción de acceder con cuenta:
 - 1.1 La aplicación debe mostrar un cuadro de diálogo con los campos de entrada “usuario” y “contraseña”.
 - 1.2 La aplicación debe permitir al usuario crear una cuenta por primera vez.
 - 1.3 La aplicación debe informar al usuario de si los datos que introdujo no coinciden con los registros existentes en la base de datos.
 - 1.4 La aplicación debe permitir al usuario recuperar su contraseña por medio de correo electrónico.*Si el usuario selecciona la opción de acceder como invitado:*
 - 1.5 La aplicación debe informar al usuario que ingresando al sistema de esta forma no podrá hacer uso de la funcionalidad de agregar cenotes ni de escribir opiniones.
2. La aplicación debe poder presentarse en español e inglés.
3. La aplicación debe permitir al usuario alternar entre idiomas en cualquier momento.
4. Después de que el usuario acceda al sistema, la aplicación debe mostrar la pantalla principal (ver apartado *Interfaces de usuario*).
5. La aplicación debe permitir al usuario navegar en un mapa, donde podrá visualizar los distintos cenotes que otros usuarios hayan agregado al sistema.
 - 5.1 La aplicación debe permitir al usuario seleccionar un cenote en el mapa
 - 5.2 Cuando el usuario seleccione un cenote, se desplegará la pantalla “Vista de cenote” (ver apartado *Interfaces de usuario*).
 - 5.3 La aplicación debe mostrar al usuario la información del cenote seleccionado. Esta información debe ser:
 - 5.4 La aplicación debe mostrar al usuario las opiniones que hayan escrito otros usuarios acerca del cenote seleccionado
6. La aplicación debe permitir a los usuarios con cuenta registrar un nuevo cenote.
 - 6.1 Al registra un nuevo cenote, la aplicación debe permitir a los usuarios marcar la ruta del mismo en el mapa.

Desempeño de requerimientos (requerimientos no funcionales).

<Especifica los requerimientos numéricos, estáticos y dinámicos incluidos en el software o en una interacción humana con el software completo. Los requerimientos numéricos estáticos incluyen:

a)Número de terminales que deben ser soportadas.

b) Número de usuarios simultáneos que deben ser soportados.

c) Cantidad y tipos de información que debe ser manipulada.

Los requerimientos numéricos estáticos son identificados bajo una sección separada titulada "capacidad".

Los requerimientos numéricos y dinámicos incluyen por ejemplo, el número de transacciones y tareas y la cantidad de datos a ser procesados con o en un cierto período de tiempo para condiciones de trabajo normales y críticas. Todos estos requerimientos deben ser enunciados en términos cuantificables.

Nota: Las limitaciones numéricas aplicadas a funciones específicas son normalmente especificadas como parte de la descripción del apartado del procesamiento en esa función.>

Documentación de usuario

<Listar los componentes de documentación de usuario como manuales de usuario, ayuda en línea y tutoriales que deben ser entregados con el software. Identificar los formatos y estándares de documentación.>

Junto con la entrega del software, debe ser entregado un manual de usuario que explique el uso de la aplicación, para ello se deberá seguir el estándar *IEEE 1063 – IEEE Standard for Software User Documentation*.

Atributos del sistema de software.

<Hay un número de atributos de software que pueden servir como requerimientos. Es importante que los atributos requeridos sean especificados de manera que sus logros sean objetivamente verificados.>

Confiabilidad.

<Se especifican los factores requeridos para establecer la confiabilidad deseada del sistema de software en tiempo de entrega.>

Seguridad.

<Se especifican los factores que protegen al software de accesos accidentales o maliciosos, uso, modificación, destrucciones o divulgación de información no autorizada.

Los requerimientos específicos en esta área podrán incluir la necesidad de:

a) Utilizar ciertas técnicas de criptografía.

b) Llevar una bitácora o historia de los datos.

c) Asignar ciertas funciones a diferentes módulos.

d) Restringir la comunicación entre algunas áreas del programa.

e) Verificar la integridad de datos para variables críticas.

f) Identificación de usuarios.>

Mantenimiento.

<Se especifican los atributos de software que asocien el fácil mantenimiento del software. Puede ser que existan ciertos requerimientos para cierto tipo de modularidad, interfases, complejidad, etc. Estos requerimientos no deben ser definidos en esta sección.>

Portabilidad.

<Se especifican atributos de software que especifican la fácil portabilidad del software a otros servidores y/o sistemas operativos. Esta especificación debe incluir:

- a) Porcentaje de componentes con código dependiente del host.*
 - b) Porcentaje de código que es dependiente del host.*
 - c) El uso de un lenguaje portable probado.*
 - d) El uso de un compilador particular o conjunto de lenguajes.*
 - e) Uso de un sistema operativo particular.>*
-

3 APÉNDICES

<Deben incluir lo siguiente:

a) Ejemplos de formatos de entrada salidas, descripción de estudios de análisis de costos o resultados de encuestas a usuarios.

b) Soporte o referencias de información que puedan ayudar a los lectores de la ERS

c) Descripción de problemas a ser resueltos por el software.

d) Empaquetamientos especiales del código y los medios para encontrar la seguridad, exportación, carga inicial u otros requerimientos.>
