Notas:

* Los textos en cursiva de cada sección son indicaciones que deben eliminarse y sustituirse por los contenidos descritos en cada apartado.
* La tabla de revisiones debe reflejar la última versión válida del documento

Miembros del grupo

|  |  |
| --- | --- |
| **Nombre** | **Apellidos** |
| Miguel | Carballo |
| Marcos | Gago |
| David | Martínez |
| Alberto | Meneses |
| Esteban | Vidiella |

Versión

|  |  |  |
| --- | --- | --- |
| **Versión** | **Fecha** | **Comentarios** |
| 0.0 | 21/02/2020 | Dia de planificación, no hay demo a la vista |

Especificación de Requisitos de Software

Formato IEEE 830

[**Introducción**](#_npbjqvklurii) **2**

[1.1 Propósito](#_vgwhlvydyflq) 2

[1.2 Alcance](#_3iupoz3hz3yz) 2

[1.3 Definiciones, siglas y abreviaturas](#_pn85vcrayoqt) 2

[1.4 Referencias](#_dyc67je6wpfy) 2

[**Descripción global**](#_ni14snwmfxl) **2**

[2.1 Perspectiva del producto](#_5i1gydnccy2y) 2

[2.2 Funciones del producto](#_b59y1d9x0isf) 3

[2.3 Características del usuario](#_xmsbjdpnxcyh) 3

[2.4 Restricciones](#_t86wm43dboq8) 3

[2.5 Suposiciones y Dependencias](#_sks1bcxzgtfo) 3

[2.6 Requisitos Futuros](#_iwj7gw6nqcco) 3

[**Requisitos Específicos**](#_9sxyzcqgclpt) **4**

[3.1 Interfaces externas](#_1zstwsnnohx3) 5

[3.2 Funciones](#_o477g2sdhruf) 5

[3.3 Requisitos de rendimiento](#_77ox7212z8qg) 6

[3.4 Restricciones de diseño](#_wgqfqie9iqr) 6

[3.5 Atributos del sistema](#_ubfl91nn8z5t) 6

[3.6 Otros requisitos](#_mp2if0w0k4sl) 6

[**Apéndices**](#_5sd0lcjml53) **6**

## Introducción

### 1.1 Propósito

Nuestra aplicación provee a usuarios información de la enfermedad que ellos mismos han buscado; aunque también ofrece la posibilidad de encontrar la enfermedad más posible teniendo en cuenta los síntomas que nos introduce el usuario. Los usuarios son principalmente personas hipocondríacas, o que prefieran auto diagnosticarse.

### 1.2 Alcance

Habrá dos prototipos respecto a Searchill: una aplicación java con una base de datos externa y una página HTML con de nuevo, una base de datos externa. El usuario podrá elegir el modo de búsqueda (mediante síntomas encontrar enfermedad o simplemente buscar la enfermedad). Al encontrar enfermedad, si se está satisfecho con el diagnóstico el usuario podrá darle una review positiva, de manera que a la hora de calcular la heurística se tendrá en cuenta la review positiva; a parte al encontrar la enfermedad, el usuario puede pedir mas información sobre ella, de manera que podrá abrirse una pagina web como las de la OMS para obtenerla.

### 1.3 Definiciones, siglas y abreviaturas

Enfermedad: referencia a la clase de enfermedad que vamos a crear compuesta de síntomas

Síntoma: sensaciones que percibe el usuario, procedentes de una o varias enfermedades

Popularidad: número otorgado a la enfermedad en función de lo popular que sean los síntomas de la enfermedad

Prioridad: forma en la que aparecerán las enfermedades en función de parámetros que encajen

Estaciones: referente a verano, otoño, invierno y primavera

Base: forma de acortar base de datos

HTML:

### 1.4 Referencias

*En esta subsección se mostrará una lista completa de todos los documentos referenciados en la ERS.*

|  |  |
| --- | --- |
| **Título del documento** | **Referencia** |
| Especificacion software del buscador de enfermedades | ERS1-REF1 |

## Descripción global

### 2.1 Perspectiva del producto

La aplicación cuya interfaz de cara al usuario es HTML actúa conjuntamente con el código de Java y las bases de datos que este utiliza

### 2.2 Funciones del producto

* El usuario podrá buscar enfermedades por su nombre.
* El usuario podrá introducir síntomas que padece y el sistema le dará ciertas enfermedades que encajan con sus datos y síntomas.
* El sistema podrá dar una breve descripción de la enfermedad on-code.
* Si el usuario quiere el sistema puede abrir una url contiendo más información sobre la enfermedad.
* El usuario puede introducir datos de sí mismo para hacer la búsqueda más precisa.

### 2.3 Características del usuario

|  |  |
| --- | --- |
| **Tipo de usuario** | Usuario estándar |
| **Criterio** | Todos aquellos que posean la aplicación deben tener acceso a la base de datos. |

### 2.4 Restricciones

*Esta subsección describe aquellas limitaciones que se imponen sobre los desarrolladores del producto:*

* *Limitaciones del hardware*
* *Interfaces con otras aplicaciones*
* *Fechas de auditoría (en vuestro caso del profesor)*
* *Lenguaje(s) de programación*
* Java mediante Eclipse IDE y HTML.
* *Protocolos de comunicación*
* El equipo de desarrollo está siempre intercomunicado entre sí, pero se podrá contactar con él vía correo electrónico.
* *Consideraciones acerca de la seguridad*
* Cumplir con la Ley Orgánica de protección de datos.

### 2.5 Suposiciones y Dependencias

*Esta subsección describe aquellos factores que, si cambian, pueden afectar a los requisitos. Por ejemplo, los requisitos pueden presuponer una cierta organización de ciertas unidades de la empresa, o pueden presuponer que el sistema correrá sobre cierto sistema operativo. Si cambian dichos detalles en la organización de la empresa, o si cambian ciertos detalles técnicos, como el sistema operativo, puede ser necesario revisar y cambiar los requisitos.*

Asumimos que el back-end (Java y la base de datos) se comunicará mediante apis con el front-end (HTML); restringimos el uso al sistema operativo Windows por el momento.

### 2.6 Requisitos Futuros

Se prevé una forma de almacenar los datos de los usuarios en una estructura valida e incluir una gran cantidad de enfermedades y síntomas mas específicos para hacer la búsqueda mas sencilla.

## Requisitos Específicos

*Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que satisfaga estos requisitos, y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos. Todo requisito aquí especificado describirá comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas.*

*Esta es la sección más larga e importante de la ERS. Deberán aplicarse los siguientes principios:*

* *El documento debería ser perfectamente legible por personas de muy distintas formaciones e intereses.*
* *Deberán referenciarse aquellos documentos relevantes que poseen alguna influencia sobre los requisitos.*
* *Todo requisito debería ser unívocamente identificable mediante algún código o sistema de numeración adecuado.*
* *Lo ideal, aunque en la práctica no siempre realizable, es que los requisitos posean las siguientes características*
  + *Corrección: La corrección de la ERS implica que el sistema implementado será el sistema deseado.*
  + *No son ambiguos: Cada requisito tiene una sola interpretación. Para eliminar la ambigüedad inherente a los requisitos expresados en lenguaje natural, se deberán utilizar gráficos o notaciones formales. En el caso de utilizar términos que, habitualmente, poseen más de una interpretación, se definirán con precisión en el glosario*
  + *Completos: Todos los requisitos relevantes han sido incluidos en la ERS. Deben incluir todas las posibles respuestas del sistema a los datos de entrada, tanto válidos como no válidos*
  + *Consistentes: Los requisitos no pueden ser contradictorios. Un conjunto de requisitos contradictorio no es implementable.*
  + *Clasificados: Normalmente, no todos los requisitos son igual de importantes. Los requisitos pueden clasificarse por importancia (esenciales, condicionales u opcionales) o por estabilidad (cambios que se espera que afecten al requisito). Esto sirve, ante todo, para no emplear excesivos recursos en implementar requisitos no esenciales.*
  + *Verificables: La ERS es verificable si y sólo si todos sus requisitos son verificables. Un requisito es verificable (testeable) si existe un proceso finito y no costoso para demostrar que el sistema cumple con el requisito. Un requisito ambiguo no es, en general, verificable. Expresiones como a veces, bien, adecuado, etc. introducen ambigüedad en los requisitos. Requisitos como “en caso de accidente la nube tóxica no se extenderá más allá de 25Km” no es verificable por el alto costo que conlleva.*

### 3.1 Interfaces externas

*Se describirán los requisitos que afecten a la interfaz de usuario, interfaz con otros sistemas (hardware y software) e interfaces de comunicaciones. Debe ser una descripción detallada de todas las entradas y salidas del sistema del software, incluyendo:*

* *la descripción de propósito*
* *la fuente de entrada o destino de salida*
* *el rango válido*
* *exactitud y/o tolerancia*
* *las unidades de medida*
* *tiempos*
* *las relaciones a otras entradas/salidas*
* *el formato de pantalla /organización*
* *el formato de ventanas/organización*
* *los formatos de los datos*
* *los formatos de los comandos*
* *fin de mensajes*

### 3.2 Funciones

*Esta subsección (la más larga del documento) deberá especificar todas aquellas acciones (funciones) que deberá llevar a cabo el software. Si se considera necesario, podrán utilizarse notaciones gráficas.*

*Las funciones pueden organizarse de múltiples formas, sugiriendo las siguientes:*

* *Por tipos de usuario: Distintos usuarios poseen distintos requisitos. Para cada clase de usuario que exista en la organización, se especificarán los requisitos funcionales que le afecten o tengan mayor relación con sus tareas.*
* *Por objetos: Los objetos son entidades del mundo real que serán reflejadas en el sistema. Para cada objeto, se detallarán sus atributos y sus funciones. Los objetos pueden agruparse en clases. Esta organización de la ERS no quiere decir que el diseño del sistema siga el paradigma de Orientación a Objetos.*
* *Por objetivos: Un objetivo es un servicio que se desea que ofrezca el sistema y que requiere una determinada entrada para obtener su resultado. Para cada objetivo o sub- objetivo que se persiga con el sistema, se detallarán las funciones que permitan llevarlo a cabo.*
* *Por estímulos: Se especificarán los posibles estímulos que recibe el sistema y las funciones relacionadas con dicho estímulo.*
* *Por jerarquía funcional: funciones que comparten entradas, salidas o datos internos. Se detallarán las funciones (entrada, proceso, salida) y las subfunciones del sistema.*

### 3.3 Requisitos de rendimiento

*Se detallarán los requisitos relacionados con la carga que se espera tenga que soportar el sistema. Por ejemplo, el número de terminales, el número esperado de usuarios simultáneamente conectados, número de transacciones por segundo que deberá soportar el sistema, etc. También, si es necesario, se especificarán lo requisitos de datos, es decir, aquellos requisitos que afecten a la información que se guardará en la base de datos. Por ejemplo, la frecuencia de uso, las capacidades de acceso y la cantidad de registros que se espera almacenar (decenas, cientos, miles o millones).*

*Todos que estos requisitos deben declararse en las condiciones mensurables. Por ejemplo, 95% de las transacciones se procesarán en menos de 1 seg.*

### 3.4 Restricciones de diseño

*Todo aquello que restrinja las decisiones relativas al diseño de la aplicación: Restricciones de otros estándares, limitaciones del hardware, etc. Por ejemplo, un requisito de rastro de auditoría puede declarar que deben grabarse todos los cambios a un banco de datos de la nómina en un archivo del rastro con los valores antes del proceso y después del proceso.*

La comunicación entre los back end y el front end; cuan rápida sea la API de interconexión entre html y

### 3.5 Atributos del sistema

Disponibilidad: la disponibilidad debe de ser continua mientras que el acceso a internet y las bases de datos sea continuo.

### 3.6 Otros requisitos

*Cualquier otro requisito que no encaje en otra sección*

## Apéndices

*Pueden contener todo tipo de información relevante para la ERS pero que, propiamente, no forme parte de la ERS. Por ejemplo: formatos de entrada/salida de datos, por pantalla o en listados o resultados de análisis de costes.*