Especificación de Requisitos de Software

para

Mercedario Recetas

Versión 1.0 en revisión

Hecha por Antonio Parra, Cristian Orbes y Sebastián Ramos

Universidad Cooperativa de Colombia (Sede pasto)

6 de septiembre de 2023

Tabla de contenido

Tabla de contenido ii

Historial de Revisión ii

1. Introduction 1

1.1 Proposito 1

1.2 Convenciones de documento 1

1.3 Audiencia destinada y sugerencias de lectura 1

1.4 Visualizacion del producto 1

1.5 Referencias 1

2. Descripcion General 2

2.1 Perspectiva del producto 2

2.2 Funciones del producto 2

2.3 Clases de usuario y caracteristicas 2

2.4 Entorno operativo 2

2.5 Limitaciones de diseño e implementación 2

2.6 Documentación de usuario 2

2.7 Asunciones y/o dependencias 3

3. Requerimientos de interfaces externas 3

3.1 Interfaces de usuario 3

3.2 Interfaces de hardware 3

3.3 Interfaces de software 3

3.4 Interfaces de comunicación 3

4. Caracteristicas del sistema 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Historial de Revisión

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Fecha** | **Razón de cambio** | **Versión** |
| Antonio Parra | 6/09 | Agregando contenido | 0.1 |
| Antonio Parra | 27/09 | Traduciendo contenido | 1.0 |

# Introducción

## Propósito

Este documento cubre la primera versión del sistema, una aún en revisión para verificar su correcto funcionamiento y documentación. El producto a entregar se trata de un software encargado para el restaurante Mercedario Recetas, este software está planeado para ser usado únicamente por el personal del restaurante involucrado en la preparación de los distintos platos servidos en el mismo.

## Convenciones del documento

N/A.

## Audiencia destinada y sugerencias de lectura.

Este documento está destinado a las manos del equipo de desarrollo, al propietario del producto a entregar, al usuario común y al usuario recurrente, al ingeniero destinado a realizar mantenimiento del software entregado y al administrador del proyecto. Tomad este documento como una guía de lo que van a encontrar en el programa entregado a la empresa.

## Visualización del producto.

Mercedario recetas necesitaba un software que le ayudase a gestionar la preparación de sus recetas y controlar los ingredientes utilizados en cada receta, se necesitaba un sistema que le permitiera sus usuarios observar las recetas, los ingredientes utilizados y la forma de preparar los platos, de manera que tuvieran a su disposición todas las características necesarias a la hora de cocinar los platos.

## Referencias

La especificación de los requisitos del producto se encontrará en la hoja de cálculo “Tablas de requisitos” realizada en Excel y encontrada en la carpeta en línea del proyecto Mercedario Recetas. Los diagramas del producto se encontrarán en la carpeta en línea del proyecto Mercedario Recetas, Graficas, en esta se encontrarán los diagramas de flujo, contexto y clase respectivamente.

Repositorio de GitHub:

<https://github.com/AntBPV/Ingenieria_de_requisitos/tree/main/Mercedario%20Recetas>

# Descripción general

## Perspectiva del producto

El diagrama de contexto que otorgara la perspectiva que se tiene del producto estará disponible en el repositorio de GitHub vinculado a continuación: <https://github.com/AntBPV/Ingenieria_de_requisitos/tree/main/Mercedario%20Recetas>

## Funciones del producto

El diagrama de clase que permitirá la visualización de las funciones que tiene el producto estará disponible en el repositorio de GitHub vinculado a continuación: <https://github.com/AntBPV/Ingenieria_de_requisitos/tree/main/Mercedario%20Recetas>

## Clases de usuario y características

Asistente de cocina: Usuario común del programa, no tiene permisos especiales ni accesos limitados. Su función se limita a observar y preparar recetas.

Chef: Usuario especial del programa, sus permisos especiales radican en editar, agregar y eliminar cualquier receta guardada en sistema.

Distribuidor de insumos: El encargado de transportar los insumos al restaurante, es un usuario especial del programa. Sus permisos radican en editar el listado de ingredientes con sus respectivas características y cantidades.

Administrador: Usuario omnipotente en el programa, obtiene todos los permisos del sistema.

## Entorno operativo

El programa deberá ser ejecutado en un sistema Windows o mac con acceso a la consola de comandos, que, a su vez, deberá tener instalado el lenguaje de programación Python y su PATH para poder funcionar correctamente sin necesidad de instalar ningún editor de código o interprete.

## Limitaciones de diseño e implementación

El programa debe ser desarrollado exclusivamente en el lenguaje de programación Python. El programa va dirigido únicamente al personal del restaurante encargado de la preparación de los alimentos y aquellos que suministren al restaurante con insumos. Los datos deben ser guardados en el sistema en el que el programa opera. Ciertas funcionalidades, especificadas en los requisitos deben ser accesibles únicamente al obtener ciertos permisos especiales según su rol en el restaurante.

## Documentación de Usuario

El Usuario final recibirá un manual con las instrucciones de cómo funciona el programa, como acceder a las funciones respectivas de cada rol en el restaurante y como crear su cuenta para interactuar en cada momento con el programa.  
Se les dará una tutoría a los usuarios de cómo utilizar los aspectos básicos del software, como se asignará su rol en la empresa, y que funciones pueden utilizar los cargos especiales.

## Asunciones y/o dependencias

Se asume que el programa será utilizado por todo el personal responsable de preparar los alimentos del restaurante, es decir: Chef, distribuidor de insumos, asistentes de cocina, etc. El programa está planeado para recibir al empleado común como un usuario normal sin permisos especiales.  
El programa depende de que el rol de distribuidor de insumos, cumpla su parte en el programa al ingresar la cantidad y tipo de insumos que ingresan en el restaurante de manera correcta. Para asegurar que el programa cometa un error con los insumos disponibles, se recomienda que varios usuarios puedan editar la sección de insumos del programa.

Se asume que en cada receta se utilizará una cantidad en números enteros de insumos, y que no se registrarán como porciones o fracciones de un insumo ya utilizado. Es decir, no se espera que se reciclen o se utilicen las sobras de un insumo en una receta.

# Requerimientos de interfaces externas

## Interfaces de Usuario

La interfaz de usuario utilizada en el programa únicamente es la pantalla del sistema en el que el programa este operando. En esta se enseñará la consola de sistema en la que se ejecutará el programa.

## Interfaces de Hardware

Las únicas interfaces de hardware que el programa necesita para comunicarse con el usuario se tratan de una computadora de oficina básica en la que el programa pueda ejecutarse y un teclado de escritura para computadora. Gracias a este el usuario se podrá desplazar por los menús y comunicar al sistema las entradas necesarias.

## Interfaces de software

El programa deberá ser ejecutado en un sistema operativo Windows o un sistema operativo mac el cual no se ha logrado especificar sus versiones compatibles.

En caso de usar Windows: Se recomienda utilizar las versiones 8.9 en adelante del sistema operativo. Para que el programa funcione se necesita instalar el lenguaje de programación Python completamente gratis desde la página oficial: <https://www.python.org>

El lenguaje deberá ser instalado en el PATH del sistema operativo para que el programa pueda ser ejecutado directamente desde la consola de sistema o CMD.

## Interfaces de comunicación.

No se conoce ninguna interfaz de comunicación en el proyecto.

# Características del Sistema

A continuación, se presentarán las características funcionales dentro del sistema en Mercedario Recetas, estas características se tratan de distintos procesos redactados de manera detallada para lograr obtener una vista general del producto a obtener, así mismo como poder entender el funcionamiento del mismo y como se ha ideado el proyecto.

## System Feature 1

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## System Feature 2 (and so on)

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>