**Software Requirements**

**Specification**

**for**

**<Project>**

**Version 1.0 approved**

**Prepared by <author>**

**<Universidad Nacional de Río Cuarto>**

**<03/04/2023>**

***Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.***

# Table of Contents

[Table of Contents ii](#_TOC_250033)

[Revision History ii](#_TOC_250032)

1. [Introduction 1](#_TOC_250031)
   1. [Purpose 1](#_TOC_250030)
   2. [Document Conventions 1](#_TOC_250029)
   3. [Intended Audience and Reading Suggestions 1](#_TOC_250028)
   4. [Product Scope 1](#_TOC_250027)
   5. [References 1](#_TOC_250026)
2. [Overall Description 2](#_TOC_250025)
   1. [Product Perspective 2](#_TOC_250024)
   2. [Product Functions 2](#_TOC_250023)
   3. [User Classes and Characteristics 2](#_TOC_250022)
   4. [Operating Environment 2](#_TOC_250021)
   5. [Design and Implementation Constraints 2](#_TOC_250020)
   6. [User Documentation 2](#_TOC_250019)
   7. [Assumptions and Dependencies 3](#_TOC_250018)
3. [External Interface Requirements 3](#_TOC_250017)
   1. [User Interfaces 3](#_TOC_250016)
   2. [Hardware Interfaces 3](#_TOC_250015)
   3. [Software Interfaces 3](#_TOC_250014)
   4. [Communications Interfaces 3](#_TOC_250013)
4. [System Features 4](#_TOC_250012)
   1. [System Feature 1 4](#_TOC_250011)
   2. [System Feature 2 (and so on) 4](#_TOC_250010)
5. [Other Nonfunctional Requirements 4](#_TOC_250009)
   1. [Performance Requirements 4](#_TOC_250008)
   2. [Safety Requirements 5](#_TOC_250007)
   3. [Security Requirements 5](#_TOC_250006)
   4. [Software Quality Attributes 5](#_TOC_250005)
   5. [Business Rules 5](#_TOC_250004)
6. [Other Requirements 5](#_TOC_250003)

[Appendix A: Glossary 5](#_TOC_250002)

[Appendix B: Analysis Models 5](#_TOC_250001)

[Appendix C: To Be Determined List 6](#_TOC_250000)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

# Introduction

## Purpose

Esta Software Requirement Specification (SRS) define con detalle la estructura, especificaciones, requerimientos funcionales, no funcionales y de dominio del proyecto <Project>. Este documento está centrado en orientar tanto a desarrolladores como a usuarios.

El propósito del proyecto <Project> es desarrollar un software de trivia basado en conceptos varios de la programación. Su público objetivo es principalmente estudiantes y entusiastas de la programación que deseen poner a pruebas sus conocimientos.

## Document Conventions

|  |  |
| --- | --- |
| **Termino** | **Significado** |
| Trivia | Un sistema que emplea diversas técnicas de validación de preguntas. |
| Progreso | Sistema de niveles y logros. |
| Ranking | Tabla comparativa del progreso de los usuarios de la aplicación (50 mejores). |

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

## Intended Audience and Reading Suggestions

Este documento está dirigido a todas las personas involucradas en la supervisión o participación en el proyecto <Project>. El documento brinda una breve introducción al proyecto, a sus objetivos y distintos tipos de requerimientos.

## Product Scope

El producto final es una aplicación que provee a los usuarios un sistema de trivia enfocada en la temática de programación. La misma debe soportar conexión entre usuarios y red de internet. El usuario debe estar habilitado a:

* Responder preguntas
* Conocer las respuestas correctas e incorrectas
* Seguimiento del progreso y logros
* Reportar errores
* Ver posición en el ranking

## References

# Overall Description

## Product Perspective

## El proyecto <Project> es un nuevo software auto-contenido desarrollado como una aplicación web y destinado a ofrecer a los usuarios una forma entretenida de reforzar sus conocimientos sobre conceptos de programación.

## Product Functions

**Sistema de Usuario:**

Una vez se ingresa a la aplicación se mostrará el menú principal donde el usuario tendrá de iniciar sesión (mediante usuario y contraseña) o crear una nueva cuenta.

**Sistema de Trivia:**

La aplicación de basa en un sistema de niveles a alcanzar por el usuario. Se avanza de nivel al responder 10 preguntar de manera correcta. Avanzar de nivel implica un aumento en la dificultad de las preguntas. Al responder incorrectamente a una pregunta, el usuario perderá una vida. Al llegar a cero vidas, se reinicia el nivel actual para continuar. Antes de que un usuario pueda comenzar una nueva partida, debe tener al menos una vida disponible.

Si el usuario no tiene ninguna vida disponible, se le informará que necesita esperar para recuperar al menos una vida antes de poder comenzar una nueva partida.

Las preguntas cuentan con un ID único, 4 opciones y una respuesta correcta.

Después de que un usuario responda correctamente todas las preguntas de un nivel, se llevará a cabo un "Examen Final".

El Examen Final consistirá en un conjunto de 3 preguntas, que incluirán aquellas preguntas que el usuario respondió incorrectamente durante el nivel superado, así como preguntas correspondientes al nivel.

Las preguntas incorrectamente respondidas durante el nivel serán incluidas en el Examen Final para proporcionar al usuario una oportunidad de corregir sus errores y consolidar su aprendizaje.

**Sistema de Vidas:**

Los usuarios comenzarán con 3 vidas. El usuario puede recuperar vidas perdidas esperando un tiempo (8 horas). El tiempo de espera solo se cuenta cuando el usuario no está participando en una partida activa (es decir, no está respondiendo preguntas). Transcurrido el tiempo de espera el usuario recupera 1 vida.

El usuario podrá reportar errores a través de un apartado en la propia pregunta.

**Sistema de Reportes:**

En cada pregunta, los usuarios pueden observar un apartado de reporte para indicar errores en la pregunta o en las respuestas.

**Seguimiento del Progreso y Logros:**

El progreso de cada usuario se registra y se mide a través del nivel alcanzado y logros obtenidos. Se le permite al usuario realizar un seguimiento de su progreso a lo largo del tiempo, incluyendo estadísticas como nivel alcanzado, número de preguntas respondidas correctamente, porcentaje de respuestas correctas y logros desbloqueados.

**Ranking:**

Se crea una tabla de posiciones donde el usuario puede consultar si se encuentre entre los 50 mejores jugadores (mayor nivel alcanzado y cantidad de logros obtenidos) a nivel global.

## User Classes and Characteristics

El sistema está diseñado para satisfacer las necesidades de un único tipo de usuario, denominado “jugadores”.

Los usuarios podrán registrarse, responder las preguntas, reportar errores y realizar un seguimiento de su progreso.

## Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

## User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

# External Interface Requirements

## User Interfaces

**Interfaz de inicio:**

Al abrir la aplicación, los usuarios son recibidos con una pantalla de inicio que les ofrece dos opciones:

* Iniciar sesión: Permite a los usuarios acceder a sus cuentas existentes proporcionando su nombre de usuario y contraseña.
* Registrarse: Para nuevos usuarios, permite crear una cuenta proporcionando un nombre de usuario, contraseña y correo electrónico.

**Interfaz de juego:**

Se le presenta al usuario la pregunta a responder y cuatro opciones correspondientes, junto con un apartado de reporte para informar errores.

## Hardware Interfaces

El producto tiene como objetivo ser accesible y funcional tanto en computadoras de escritorio como en dispositivos móviles, incluyendo teléfonos inteligentes y tablets.

## Software Interfaces

El producto cuenta con conexión a una base de datos donde se almacena información sobre sus usuarios, un conjunto de preguntas y respuestas y, además, reportes o sugerencias que puedan surgir por parte de los usuarios.

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## System Feature 1

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

### 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).>

### 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.>

### 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.>