Software Requirements Specification

for

<Argentina's Dream Team>

Version 1.0 approved

Prepared by <Fernandez Manuel, Moreno Michael, Urquiza Carlos>

<Universidad Nacional de Río Cuarto>

<03/04/2023>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

Este software requirement specification (SRS) está definiendo detalladamente la descripción de la arquitectura, las especificaciones y funcionalidades del proyecto Argentina's Dream Team.
Este documento está preparado para ambos, el equipo desarrollador y los usuarios. El propósito es desarrollar un software de trivia basado en la selección Argentina de fútbol tricampeona. El público objetivo es todo aquel que tenga interés en aprender sobre el tema.

1.2 Document Conventions

Creado con Google Docs y exportado a .pdf

Termino	Descripción
Trivia	Un sistema de preguntas que utiliza distintas técnicas de validación.
Recompensa y Castigo	Un sistema que recompensa las preguntas bien respondidas y ejecuta un castigo por las malas.
Progreso	Un sistema de niveles , donde el usuario avanza sobre los mismos y se determina su progreso.
Leaderboard	Tabla de puntuaciones sobre el progreso del usuario, comparando su progreso respecto otros usuarios.

1.3 Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

Este documento está hecho para todos los individuos que participan o supervisan el proyecto Argentina's Dream Team

1.4 Product Scope

El producto final es una aplicación web que provee a los usuarios un sistema de trivia especializado en la temática de la selección argentina tricampeona. Esta aplicación debe soportar conexión entre usuarios y la red de internet. Los usuarios tienen que estar habilitados a:

- *Responder preguntas*
- *Conocer las respuestas correctas e incorrectas*
- *Visualizar su progreso*
- *Consultar su progreso respecto a otros usuarios*
- *Sugerir preguntas y reportar errores*

1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2. Overall Description

2.1 Product Perspective

El proyecto Argentina's Dream Team es un nuevo software, autocontenido, destinado a ser una aplicación web.

2.2 Product Functions

Sistema de usuario :

- El usuario se registra por única vez a través de un nombre de usuario único y una contraseña.
- Se ingresa una única vez, cuando se inicia la aplicación.

Sistema de trivia:

- Carga de preguntas y respuestas, con solo una correcta
- Las preguntas pueden aparecer de distintas formas.
- Se aceptan sugerencias de preguntas o reportes de errores.

Sistema de recompensas y castigos:

- Se compensará al usuario de alguna manera las respuestas que sean respondidas de manera correcta y se castigaran aquellas que no, afectando a su progreso en cada caso.

Registro sobre el progreso del aprendizaje de cada usuario:

- Seguimiento del usuarios a través de un sistema de niveles que determinarán el aprendizaje alcanzado.

Diferenciación de niveles:

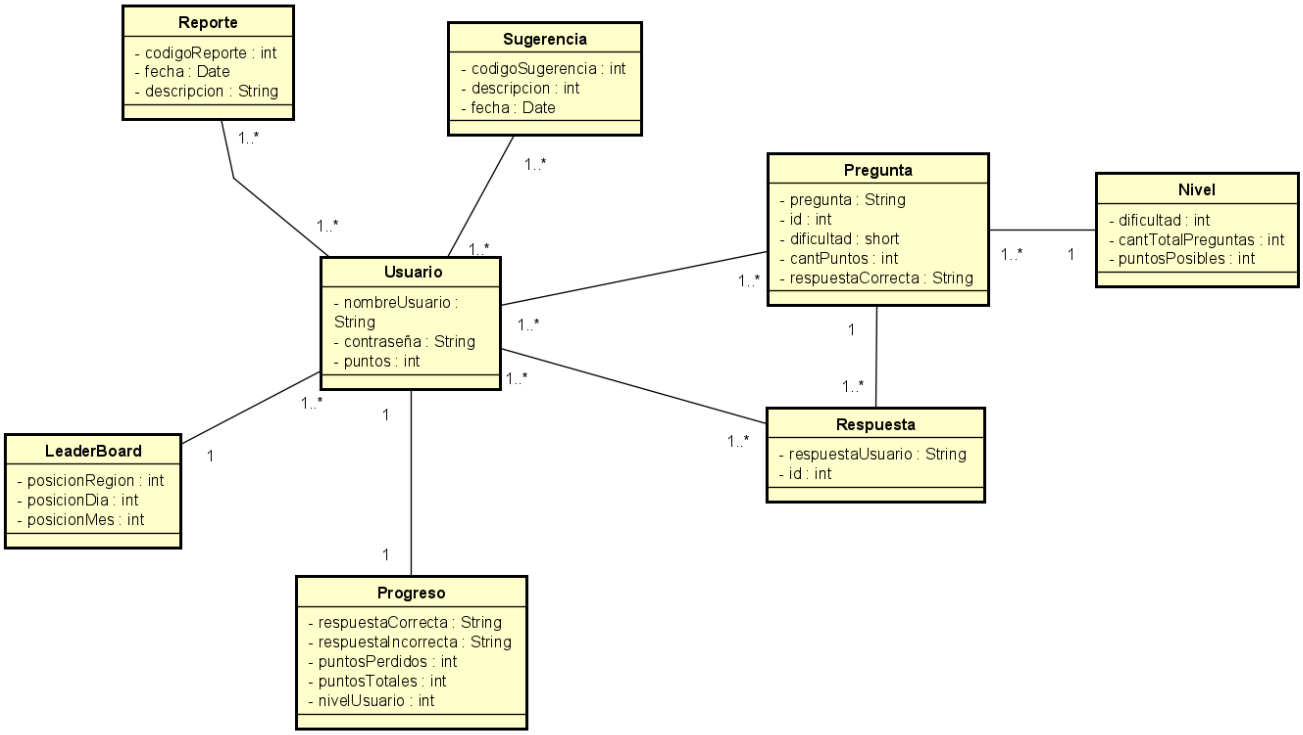
- Cada usuario pertenece a un nivel de dificultad, este será ascendiente en dificultad.
- Luego de cierto progreso, se accedera al siguiente nivel, en caso de un mal desempeño, bajara de nivel.

Sistema de puntos:

- Los puntos son asignados luego de cada trivia.
- Con un máximo determinado, por cada fallo, se descuenta sobre el total.
- Si tiene menos que cierta cantidad, se restará sobre el total de puntos del usuario.
- El mínimo de puntos es 0(cero)

Leaderboard:

- Se comparan los puntos de cada usuario y con ellos se crea una tabla de clasificación



2.3 User Classes and Characteristics

Habrá 2 tipos de usuarios, los administradores y los jugadores.
Los administradores podrán ingresar preguntas y respuestas, banear usuarios, corregir errores, y acceder a los datos de todos los usuarios.

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

2.4 Operating Environment

El componente principal de Argentina's Dream team es el software, el cual será implementado mediante 2 lenguajes, el Back-end será desarrollado en un lenguaje de programación llamado Ruby y el Front-end será desarrollado en HTML y CSS. El software resultante será una aplicación web que podrá accederse desde cualquier navegador.

2.5 Design and Implementation Constraints

La principal restricción de diseño es la plataforma. La primera versión del software estará desarrollada para poder ejecutarse desde un navegador web desde un ordenador (Pc, Notebook,etc). Por lo tanto, se espera no tener soporte para plataformas móviles (Teléfonos celulares con sistema Android, IOS ,etc.)
Otra restricción es garantizar una interfaz eficiente en cuanto a uso de recursos, además de ser lo suficientemente intuitiva para que cualquier usuario pueda utilizarla.

2.6 User Documentation

El usuario deberá superar un tutorial, éste será obligatorio para los usuarios que utilicen la aplicación por primera vez, y dicho apartado quedará disponible.

2.7 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).>

3. External Interface Requirements

3.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

3.4 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.>

4. 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.>

4.1 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:

4.2 System Feature 2 (and so on)

5. Other Nonfunctional Requirements

5.1 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.>

5.2 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.>

5.3 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.>

5.4 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.>

5.5 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.>

6. 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.>