

Software Requirements Specification for <Argentina's Dream Team>



ARGENTINA'S DT



Version 1.0 approved

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

<Universidad Nacional de Río Cuarto>

<03/04/2023>

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 References	1
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	2
2.5 Design and Implementation Constraints	2
2.6 User Documentation	2
2.7 Assumptions and Dependencies	3
3. External Interface Requirements	3
3.1 User Interfaces	3
3.2 Hardware Interfaces	3
3.3 Software Interfaces	3
3.4 Communications Interfaces	3
4. System Features	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

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 (solamente los 5 mejores).

1.3 Intended Audience and Reading Suggestions

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
- Ver posición en leaderboard
- Comprar cartas

1.5 References

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 :

Al momento de ingresar a la aplicación "Argentina 's Dream Team" , se mostrará el menú principal a través de una interfaz donde el usuario tendrá 2 opciones, crear una cuenta si es la primera vez que ingresa o iniciar sesión en una cuenta ya existente mediante usuario y contraseña.

Si se elige crear una cuenta el usuario deberá elegir un nombre de usuario, una contraseña y su confirmación(mínimo 8 letras, una mayúscula y un número).

Al iniciar sesión se pedirá su nombre de usuario y contraseña. Todo esto se mostrará mediante una interfaz en pantalla, con instrucciones para guiar al usuario.

Sistema de trivia:

Argentina 's Dream Team contará con un sistema de trivia el cual tendrá una base de datos, en la que se cargaran las preguntas con sus debidas respuestas.

Cada pregunta será cargada con un ID único.

Cada pregunta tendrá 1 respuesta correcta.

Las preguntas se mostrarán a través de una interfaz en donde el usuario podrá ver la pregunta y 4 opciones . El usuario contará con 5 vidas (en caso de no tener el máximo de vidas , se le otorgará una vida cada 6 horas) las cuales se conservarán si el usuario responde correctamente y perderá 1 vida por cada respuesta incorrecta. Además el usuario será premiado con puntos si la respuesta es correcta.

Cada respuesta respondida correctamente, aportará al usuario 3 puntos.

El usuario podrá visualizar la cantidad de respuestas seguidas que ha contestado de forma correcta (racha) y quedará registrado como dato de su perfil.

Luego de 10 preguntas, aparecerá una pregunta "bonus" que contará con una mayor dificultad y valdrá el doble de puntos que una pregunta normal.

El usuario podrá sugerir preguntas y reportar errores a través de un apartado que estará disponible en el menú principal de la aplicación.

Registro sobre el progreso del aprendizaje de cada usuario:

El progreso de cada usuario se registra y se mide a través de la cantidad de puntos totales.

El progreso de cada usuario se agrupa en las siguientes categorías:

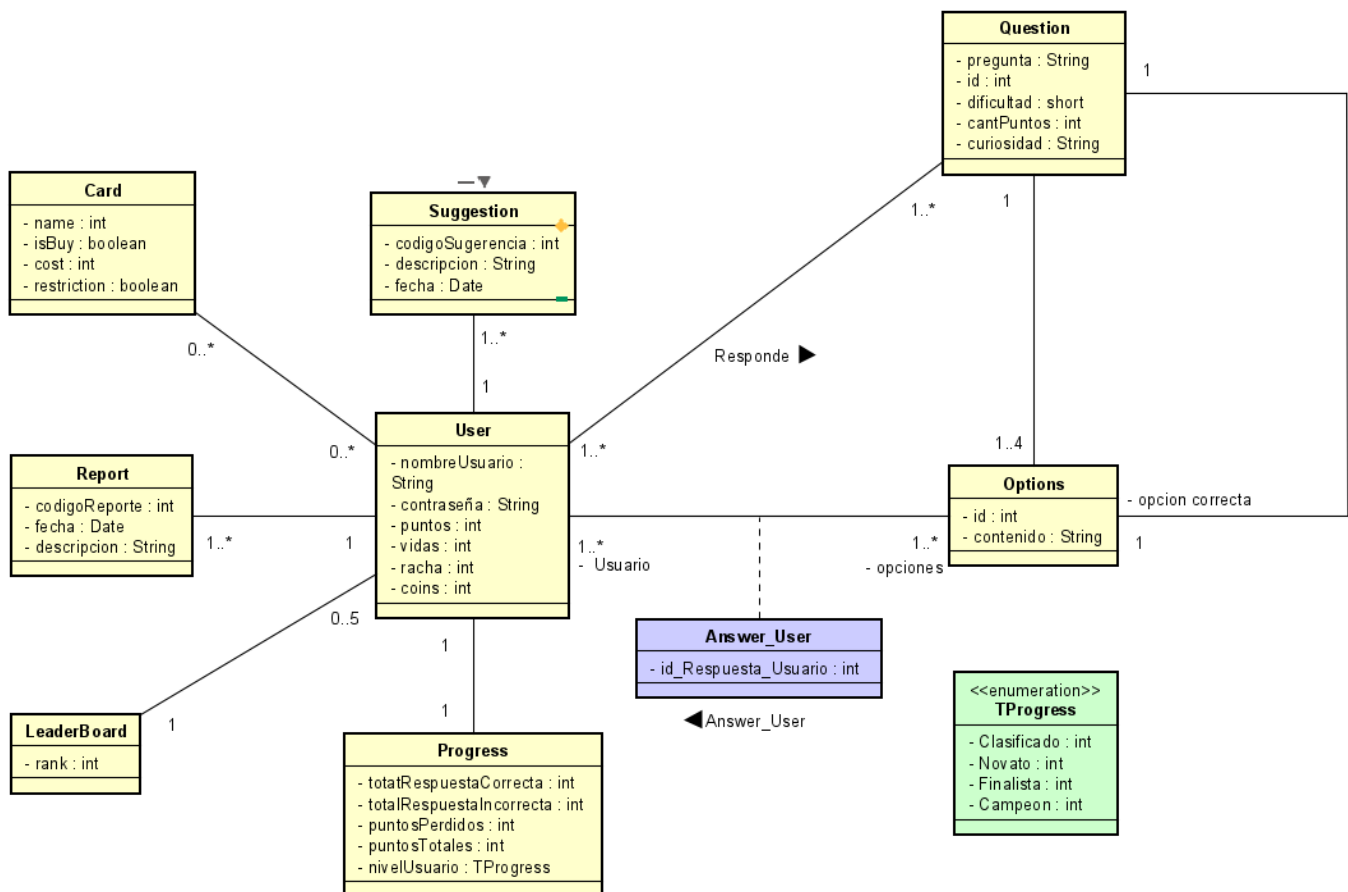
- Clasificado: La categoría del usuario al iniciar el juego (0 puntos)
- Novato: Se alcanza esta categoría al llegar a 100 puntos
- Finalista: Se alcanza esta categoría al llegar a 200 puntos
- Campeón: Se alcanza esta categoría al llegar a 300 puntos

Leaderboard:

Se crea una tabla de posiciones donde el usuario puede consultar si se encuentra entre los 5 mejores jugadores de la aplicación a nivel global.

Cards/store:

Se contará con una tienda donde se podrán comprar cartas de los jugadores, estas costarán monedas que se ganarán al responder bien, también tendrán ciertas restricciones sobre algunas cartas para poder comprarlas (como costo o cierta categoría en el progreso), una vez compradas podrás usar la foto de la carta como fondo a la hora de ver tu progreso.



2.3 User Classes and Characteristics

Habr  1 solo tipo de usuarios, es decir, los jugadores.

Los usuarios podr n 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 ser  implementado mediante 2 lenguajes, el Back-end ser  desarrollado en un lenguaje de programaci n llamado Ruby y el Front-end ser  desarrollado en HTML y CSS. El software resultante ser  una aplicaci n web que podr  accederse desde cualquier navegador.

2.5 Design and Implementation Constraints

La principal restricci n de dise o es la plataforma. La primera versi n del software estar  desarrollada para poder ejecutarse desde un navegador web desde un ordenador (Pc, Notebook, etc). Por lo tanto, se espera no tener soporte para plataformas m viles (Tel fonos celulares con sistema Android, IOS ,etc.)

Otra restricci n es garantizar una interfaz eficiente en cuanto a uso de recursos, adem s de ser lo suficientemente intuitiva para que cualquier usuario pueda utilizarla.

2.6 User Documentation

El usuario deber  superar un tutorial,  ste ser  obligatorio para los usuarios que utilicen la aplicaci n por primera vez, y dicho apartado quedar  disponible.

2.7 Assumptions and Dependencies

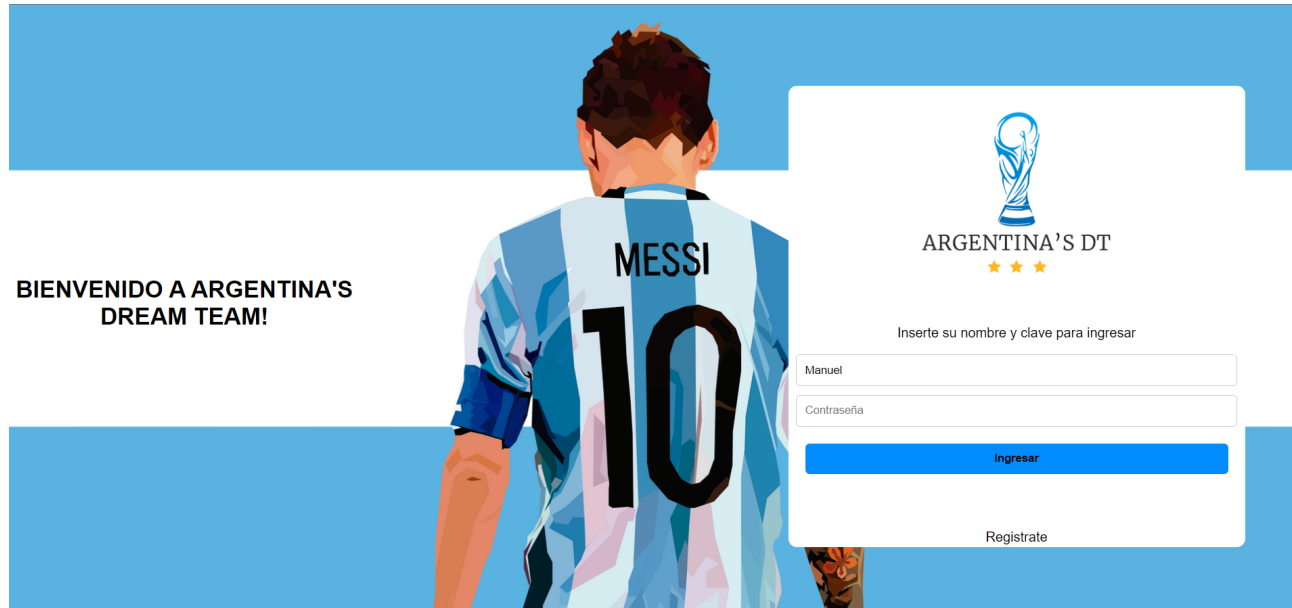
<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions 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

Interfaz de inicio:

Al ingresar a la p gina web, se le presenta al usuario la opci n de iniciar sesi n (usuario y contrase a) o registrarse si no posee una.



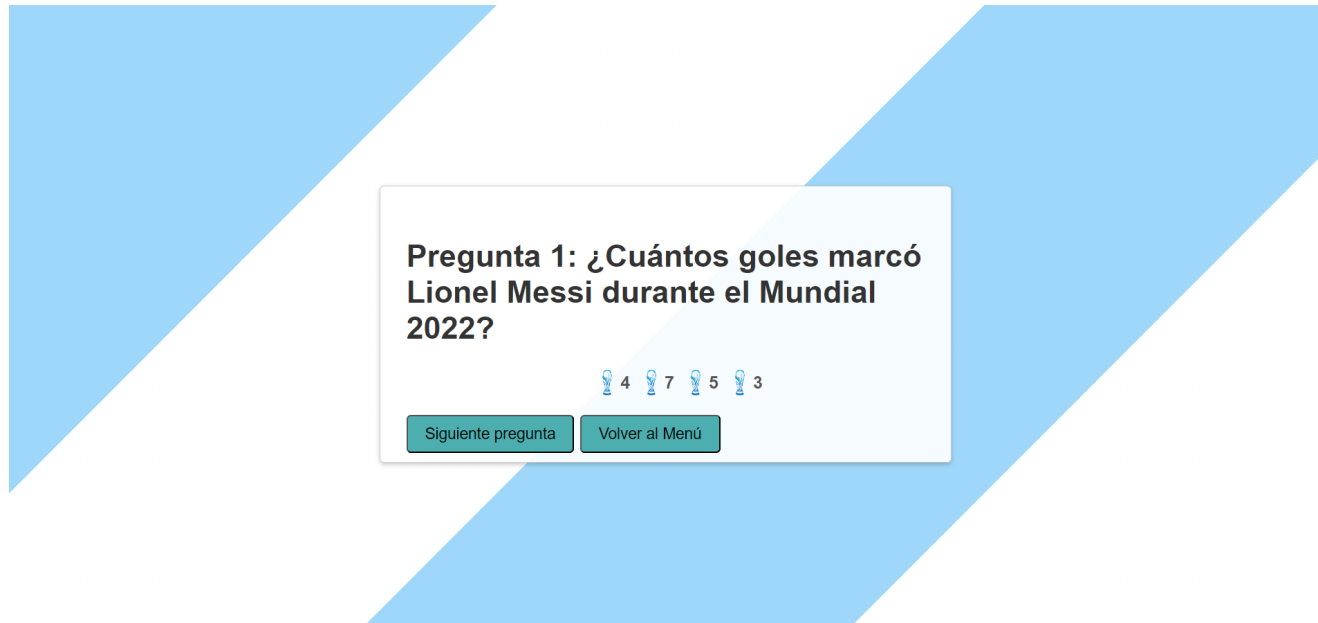
Interfaz de Registro:

Al registrar una cuenta, le pediremos que nos proporcione un usuario, una contraseña y fecha de nacimiento. Con esto tendrá registrada su cuenta.



Interfaz de Juego:

Al momento de jugar, se le presentará al usuario la pregunta a responder, con sus respectivas 4 opciones, con una de ellas correcta.



3.2 Hardware Interfaces

Este producto tiene como objetivo funcionar en dispositivos como computadoras de escritorio a través de acceso a la web.

3.3 Software Interfaces

Este producto tendrá conexiones principalmente con una base de datos donde quedarán almacenados todos los datos del mismo. En detalle, se podrá encontrar: información de los usuarios, preguntas, respuestas, reportes y sugerencias.

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.>