**Software Requirements**

**Specification**

**for**

**Road to Senior**

**Version 1.0 approved**

**Prepared by <Campregher Bruno, Centeno Nicolas, Serafín Agustín>**

**<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 y no funcionales, y el dominio del proyecto Road to Senior. Este documento está centrado en orientar tanto a desarrolladores como a usuarios.

El propósito del proyecto Road to Senior es desarrollar un software de trivia sobre programación inspirado en Duolingo. Su público objetivo son principalmente estudiantes y entusiastas que deseen poner a prueba y mejorar sus conocimientos sobre conceptos generales de programación y utilizando el lenguaje Python como medio para mostrar código.

Desde conceptos fundamentales hasta temas más avanzados, cada etapa desafiará a los usuarios con preguntas diseñadas para evaluar y ampliar su conocimiento en el lenguaje.

## Document Conventions

|  |  |
| --- | --- |
| **Término** | **Significado** |
| Trivia | Sistema que permite evaluar conocimientos a través de preguntas. |
| Leccion | Etapa que contiene una porción de preguntas las cuales se centran en uno o mas temas específicos para aprender. |
| Progreso | Estadísticas del usuario tales como la lección actual donde se encuentra, cantidad de preguntas respondidas, etc. |
| Ranking | Tabla comparativa del progreso de los usuarios de la aplicación. |

## Intended Audience and Reading Suggestions

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

## Product Scope

El proyecto "Road to Senior" es un software de trivia diseñado para ayudar a estudiantes y entusiastas de la programación a mejorar sus habilidades en conceptos clave de programación y el lenguaje especifico Python. El software permitirá a los usuarios participar en una serie de preguntas y respuestas estructuradas en diferentes etapas de aprendizaje, cada una destinada a abordar aspectos específicos de la programacion. En cada etapa, los usuarios tendrán acceso a lecciones diseñadas para reforzar el conocimiento existente y ayudar en la comprensión de nuevos conceptos. Estas lecciones podrán incluir texto, ejemplos de código y otros recursos educativos. Además de las preguntas y respuestas de trivia, el software incluirá características adicionales como la pérdida de vida por respuestas incorrectas, la posibilidad avanzar por diferentes etapas centradas en temas exclusivos y un sistema de clasificación para ver la posición en un ranking.

## 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.>*

# Overall Description

## Product Perspective

## El proyecto Road to Senior 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 y Python. Esto se logrará mediante la combinación de preguntas desafiantes, retroalimentación inmediata y una interfaz de usuario intuitiva y atractiva.

## Product Functions

* Gestión de Usuario (iniciar sesión, crear cuenta, modificar ciertos datos de usuario, eliminar progreso).
* Avanzar de lección al responder correctamente todas las preguntas correspondientes a la misma.
* Consultar contenido correspondiente a la lección actual.
* Implementar sistema de vidas y penalizar al usuario al responder de manera incorrecta.
* Permitir al usuario realizar un seguimiento del progreso (consulta de lección alcanzada, cantidad de respuestas correctas, nivel de progreso).
* Implementar sistema de clasificación (ranking) y permitir consulta por parte del usuario.

## 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, consultar contenido informativo relacionado a la leccion y realizar un seguimiento de su progreso.

## Operating Environment

El entorno operativo está diseñado para ser accesible desde una variedad de dispositivos y navegadores web comunes. Los usuarios podrán acceder a la plataforma desde computadoras de escritorio, laptops, tabletas y dispositivos móviles.

Principio del formulario

## Design and Implementation Constraints

* La aplicación será compatible con una variedad de navegadores web, ya sea Chrome, Firefox, Safari, Edge, etc.
* La aplicación será accesible desde dispositivos tales como computadoras de escritorio, notebooks, tablets, teléfonos, etc.
* Debe ser desarrollada en parte utilizando el lenguaje Ruby.

## User Documentation

La documentación para el usuario proporcionará una guía detallada sobre cómo utilizar la aplicación "Road to Senior" y comprender su lógica subyacente.

## Assumptions and Dependencies

Se asume:

* Los usuarios contarán con acceso a internet.
* Los usuarios cuentan con conocimientos básicos sobre el uso de navegadores web y el dispositivo en el cual acceden a la aplicación.
* Los usuarios cuentan con dispositivos/navegadores web compatibles.

Dependencias:

* La aplicación podría depender de servicios de terceros (como alojamiento web)
* La aplicación podría depender de bibliotecas o frameworks de código abierto para ciertas funcionalidades
* La aplicación podría depender de actualizaciones para mantener un correcto funcionamiento o resolver problemas específicos.

# External Interface Requirements

## User Interfaces

**Interfaz de inicio:**

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

* Iniciar sesión: permite a los usuarios acceder a sus cuentas existentes.
* Registrarse: permite a los usuarios crear una cuenta.
* Repositorio: redirecciona al usuario al repositorio de GitHub del proyecto.

**Interfaz de inicio de sesión:**

Los usuarios deben ingresar su nombre de usuario y contraseña.

**Interfaz de registro:**

Los usuarios deben ingresar su nombre de usuario, contraseña y dirección de correo electrónico para crear una cuenta.

**Interfaz de bienvenida:**

Dentro de esta el usuario podrá acceder a las diferentes lecciones disponibles, consultar el ranking, ver sus estadísticas de juego, configurar su cuenta y desloguearse.

**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 en computadoras de escritorio/dispositivos móviles con acceso a internet.

## 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, el contenido de todas las lecciones implementadas.

## 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

## Gestión de Usuario

### Description and Priority

### El usuario podrá iniciar utilizando un nombre de usuario y contraseña existentes o crear una nueva cuenta si es un usuario nuevo.

### Stimulus/Response Sequences Creación de cuenta EST: El usuario selecciona una opción para crear una cuenta nueva. RES: El sistema muestra un formulario solicitando al usuario que proporcione un nombre de usuario, una contraseña y una dirección de correo electrónico. EST: El usuario ingresa la información requerida y envía el formulario. RES: El sistema valida la información ingresada y crea la cuenta del usuario, mostrando un mensaje de confirmación. Modificación de cuenta EST: El usuario selecciona una opción para modificar su cuenta desde la configuración de su perfil. RES: El sistema muestra un formulario con las distintas posibles modificaciones que puede realizar sobre su cuenta. EST: El usuario realiza cambios a la información que desea modificar y envía el formulario. RES: El sistema actualiza la información de la cuenta del usuario y muestra un mensaje de confirmación. Eliminación de cuenta EST: El usuario selecciona una opción para eliminar su cuenta desde la configuración de su perfil. RES: El sistema muestra un mensaje de confirmación solicitando al usuario que confirme tal acción. EST: El usuario confirma su decisión de eliminar su cuenta. RES: El sistema elimina la cuenta del usuario y muestra un mensaje de despedida.

### Functional Requirements

### REQ-1: Crear cuenta: el usuario podrá crear una cuenta proporcionando nombre de usuario, contraseña y correo electrónico.

### REQ-2: Modificar cuenta: el usuario podrá modificar su contraseña, correo electrónico o resetear su progreso.

### REQ-3: Eliminar cuenta.

## Sistema de Juego

### Description and Priority

### Sistema de lecciones que los usuarios deben alcanzar progresivamente. Avanzar de lección implica responder correctamente todas las preguntas correspondientes a esta misma. Cada lección presenta preguntas con un aumento en la dificultad. En cada lección, los usuarios tienen acceso a un contenido teórico que les permite comprender conceptos clave y reforzar su aprendizaje antes de responder las preguntas.

### Las preguntas están identificadas por un ID único y cada una presenta cuatro opciones de respuesta, de las cuales solo una es correcta.

### El usuario dispone de 3 vidas. Si un usuario responde incorrectamente una pregunta, perderá una vida. Si el usuario, durante una sesión de juego dentro de una lección, pierde todas las vidas entonces se lo expulsa de esta y pierde el progreso acumulado de la sesión. Para recuperar vidas perdidas, el usuario debe esperar un tiempo determinado (1 minuto), al final del cual se recupera una vida.

### Stimulus/Response Sequences Consulta de contenido teórico EST: El usuario selecciona la lección a la cual quiere acceder. RES: El sistema muestra el contenido teórico asociado a tal lección, que presenta conceptos claves para reforzar el aprendizaje, y un botón con el que puede iniciar la sesión de juego de esa lección. Respuesta a preguntas EST: El usuario accede a la sesion de juego correspondiente a la leccion seleccionada anteriormente. RES: El sistema presenta una pregunta dentro de las disponibles junto a las opciones de respuesta posibles. EST: El usuario selecciona una opción como respuesta. RES: El sistema evalúa la respuesta del usuario y le muestra si la opción elegida es correcta o incorrecta, además, actualiza el número de vidas disponibles. Avance de lección EST: El usuario responde correctamente todas las preguntas de la lección en la que se encuentra. RES: El sistema muestra un mensaje de felicitaciones y permite al usuario avanzar a la siguiente leccion de su recorrido.

### Functional Requirements

### REQ-1: Consultar lección: acceder a la lección correspondiente a la etapa en el momento deseado.

### REQ-2: Responder preguntas

### REQ-3: Avanzar de etapa

## Sistema de Progreso

### Description and Priority

### El progreso de cada usuario se registra y se mide en función de diferentes parámetros. Los usuarios pueden realizar un seguimiento de su progreso a lo largo del tiempo, consultando estadísticas como la etapa actual, el nivel de progreso (junior, semi-senior, senior), el número de preguntas respondidas correctamente y el porcentaje de respuestas correctas.

### Stimulus/Response Sequences Consultar progreso EST: El usuario selecciona la opción para consultar su progreso desde su perfil o una sección dedicada del juego. RES: El sistema muestra una pantalla con información detallada sobre el progreso del usuario, incluyendo la etapa actual, el nivel de progreso, el número de preguntas respondidas correctamente y el porcentaje de respuestas respondidas correctamente.

### Functional Requirements

### REQ-1: Consultar progreso

## Sistema de Ranking

### Description and Priority

### Tabla de posiciones donde los usuarios pueden consultar si se encuentran entre los 5 mejores jugadores a nivel global. Esto se determina en función de la mayor cantidad de respuestas correctas proporcionadas por cada usuario y la lección máxima alcanzada.

### Stimulus/Response Sequences Visualizar ranking EST: El usuario selecciona una opción para visualizar el ranking desde una sección dedicada del juego. RES: El sistema muestra una tabla de posiciones que muestra a los 5 mejores jugadores a nivel global, ordenados según la mayor cantidad de respuestas correctas y la máxima lección alcanzada proporcionadas por cada usuario.

### Functional Requirements

### REQ-1: Visualizar ranking

# 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.>

* Rapido tiempo de respuesta.
* Tiempo de carga de la aplicación reducido.
* Compatibilidad con varios dispositivos y navegadores.

## 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

# Trivia: Sistema de preguntas y respuestas diseñado para evaluar y mejorar conocimientos de programación y Python.

# Etapa: Nivel de habilidad y comprensión de los conceptos dados que los usuarios deben alcanzar de forma progresiva.

# Progreso: Medición del avance que cada usuario realiza en cuanto al recorrido del juego.

# Ranking: Tabla de posiciones que muestra a los mejores jugadores a nivel global, basada en la cantidad de respuestas correctas proporcionadas por cada usuario.

# Usuario: Persona que interactúa con la aplicación y utiliza funciones tales como iniciar session, crear cuenta, responder preguntas, consultar su progreso, etc.

# Cuenta: Registro de usuario que contiene información personal y progreso.

# Lección: Recurso educativo que acompaña a cada etapa del juego para reforzar conceptos claves de la misma

# Vida: Elemento del juego que se pierde al responder una pregunta de forma incorrecta, se recupera después de cierto tiempo.

# Interfaz: Refiere a los elementos visuales con los que el usuario interactúa dentro de la aplicación.

# Navegador web: Programa/aplicación que se utilizará para acceder a la aplicación desde computadoras o dispositivos móviles.

# Lenguaje Ruby: Lenguaje de programación utilizado en parte del Desarrollo de la aplicación.

# Perfil: Sección de la aplicación donde se muestran los datos de las cuentas de usuario

# .

# 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.>*