

---

# Software Requirements Specification

for

***codeQuest***

*Versión 1.0*

*Universidad Nacional de Río Cuarto*

*Miércoles 5 de abril, 2023*

**Hecho por:**    *Agustin Dosantos*        *Nicolas Mercado*

*Santiago Sosa*                      *Juan I. Villanueva*

# **1. Introducción**

## **1.1 Propósito**

*Este documento SRS define una descripción detallada de las especificaciones y funcionalidades del proyecto codeQuest. El propósito de dicho proyecto es crear una aplicación en donde un usuario registrado pueda aprender un lenguaje de programación, mediante una etapa de validación de respuestas y luego una fase de evaluación, en donde se pueda poner a prueba lo aprendido hasta el momento*

## **1.2 Convenciones del documento**

*El presente documento será desarrollado usando Google Docs, en donde los cuatro integrantes del equipo de desarrollo harán sus aportes. La extensión será “.pdf”.*

## **1.3 Audiencia esperada y recomendaciones de lectura**

*Este documento SRS está dirigido a los docentes de la asignatura Análisis y Diseño de Sistemas de la Universidad Nacional de Río Cuarto y a otros desarrolladores.*

## **1.4 Alcance del producto**

*Aplicación creada para el uso de personas interesadas en la programación, sin importar su conocimiento previo en el tema.*

## **1.5 Referencias**

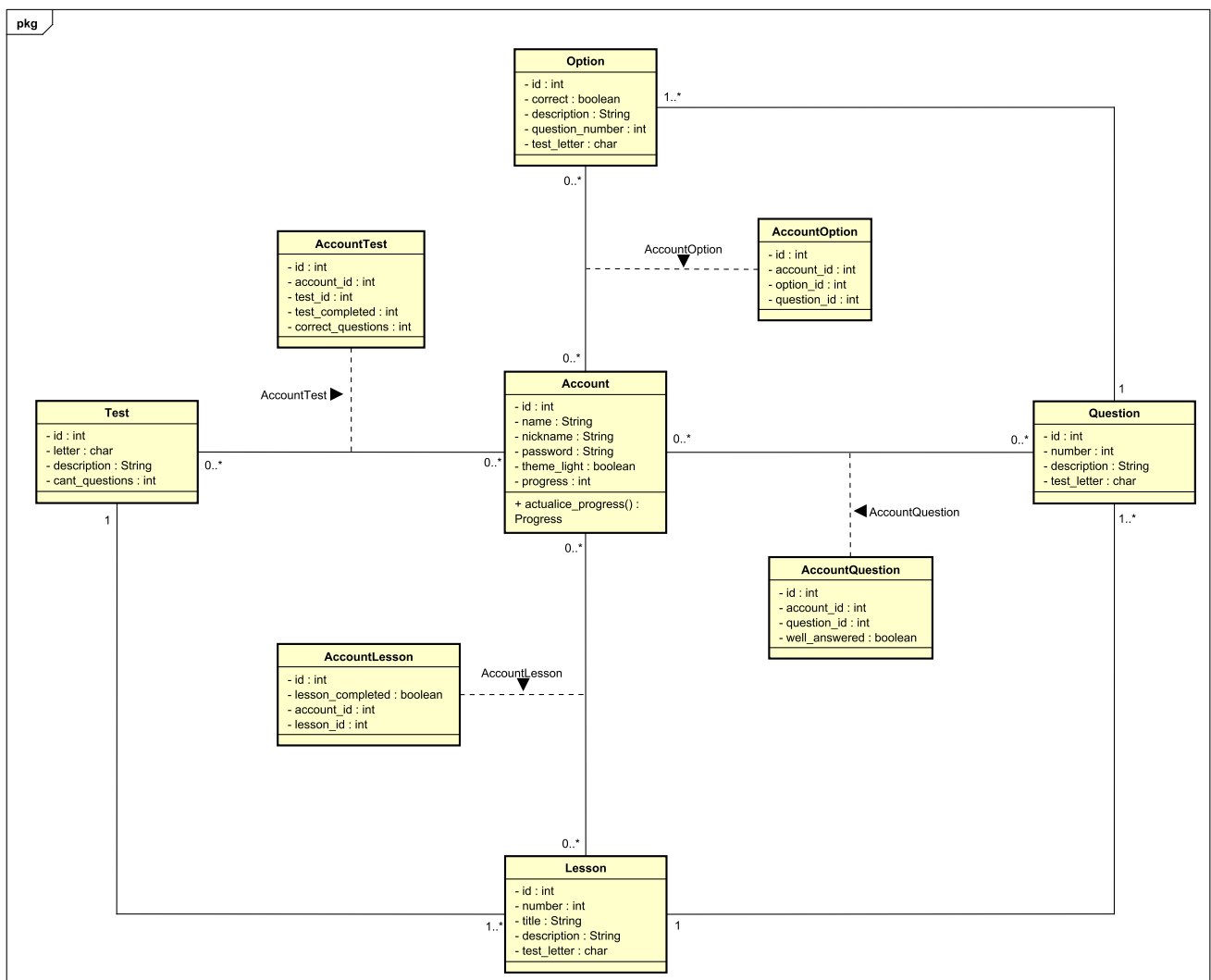
*Por ahora no hay ninguna referencia.*

## 2. Descripción General

### 2.1 Perspectiva del producto

Este proyecto surge como alternativa para aprender a programar en el lenguaje de programación Haskell, brindándoles a los usuarios una ayuda, continuidad en el aprendizaje y a su vez evaluación de su propio desempeño. Se propone una aplicación web que le permita, principalmente, a un usuario aprender este lenguaje de programación, en dos etapas: una de aprendizaje y otra de evaluación.

Diagrama de Clase del Producto:



### 2.2 Funciones del producto

La siguiente lista ofrece un breve resumen y descripción de las principales características y funcionalidades de la aplicación codeQuest. Las características se dividen en dos categorías principales: core features (características principales) y additional features (características adicionales). Las características principales son esenciales para el funcionamiento de la aplicación y las que el equipo de desarrollo cree que deben ser implementadas para que la aplicación cumpla con su propósito, mientras que las características adicionales simplemente agregan nuevas

funcionalidades que pueden ser de interés para potenciales usuarios. Estas últimas características solo se implementarán según lo apruebe el cliente y además lo permita el tiempo.

### **2.2.1 Core Features (Características Principales)**

1. *Registro Del Usuario y Bienvenida*
  - *Permite que el usuario se loguee en la aplicación.*
  - *Solo ocurre la primera vez que el usuario ingresa a la aplicación, o luego de cerrar sesión.*
2. *Ver el Progreso de Aprendizaje*
  - *Permite al usuario visualizar el progreso de aprendizaje.*
3. *Menú de configuración*
  - *Muestra todas las las configuraciones disponibles*
  - *Le permite al usuario modificar aspectos estéticos, funcionales, y de accesibilidad.*
4. *Sesion de Aprendizaje*
  - *Le permite al usuario adquirir conocimientos sobre Haskell.*
5. *Sesion de Evaluacion*
  - *Le permite al usuario poner en práctica los conocimientos adquiridos en la etapa de aprendizaje.*

### **2.2.2 Additional Features (Características Adicionales)**

7. *Guardar Snippets*
  - *Le permite al usuario guardar snippets de código para tenerlos a disposición más cómodamente.*
  - *Estos snippets se visualizan en una sección concreta de la aplicación.*
8. *Información del Perfil*
  - *Le permite al usuario visualizar la información relacionada a su perfil en la aplicación.*
9. *Guardar credenciales de inicio de sesión*
  - *Se guardan en el navegador las credenciales de la sesión de usuario para que no sea necesario que se loguee cada vez que desee ingresar.*
10. *Login con cuentas de Google, Apple o GitHub*
  - *Le permite al usuario loguearse en la aplicación utilizando las cuentas de Google, Apple o Github.*
11. *Validación de Identidad por Correo Electrónico.*
  - *Le permite al usuario validar su identidad a través del correo electrónico (se le envía un código que el usuario debe ingresar en la aplicación).*
12. *Pantallas de Carga*
  - *Se muestra en la pantalla de carga cuando se lo requiera.*
13. *Pantalla de Error*
  - *Se muestra una pantalla de error cuando una conexión entre la aplicación y el servidor haya fallado.*
14. *Foro para compartir los snippets creados*
  - *Usando un subreddit, crearemos un foro para que los usuarios puedan publicar y discutir los snippets creados.*

## **2.3 Clases de usuarios y sus características**

*Si bien la aplicación está diseñada para todo tipo de personas interesadas en aprender los diferentes lenguajes que se ofrecen, podemos distinguir a los estudiantes según su nivel de aprendizaje previo:*

- *Estudiantes principiantes: Este grupo está constituido por todas las personas sin conocimiento previo del lenguaje haskell, que deberán empezar desde cero.*
- *Estudiantes intermedios: Estos estudiantes poseen conocimientos acerca del lenguaje pero buscan reforzarlos.*
- *Estudiantes avanzados: Grupo compuesto de los estudiantes con amplio conocimiento acerca del lenguaje Haskell que solo buscan consultar una parte puntual del contenido.*

## **2.4 Desarrollo de operación**

*El sistema será desarrollado en el Entorno Integrado de Desarrollo Visual Studio Code, cuya versión es 1.77, y la plataforma de ejecución de la aplicación será cualquier navegador, entiéndase Chrome, Edge, Firefox, Brave, entre otros.*

## **2.5 Restricciones de diseño e implementación**

*Esta aplicación será diseñada mediante diferentes lenguajes (especialmente Ruby), tendrá diferentes restricciones basadas en el conocimiento de los programadores estudiantes, tiempos limitados, entre otros. La idea es que sea en español, pero tendrá diferentes adaptaciones en un futuro.*

## **2.6 Documentación del usuario**

*El usuario podrá acceder a un manual desde la aplicación, en donde será proporcionado un breve “site-tour” explicando como usar la aplicación y sus herramientas.*

## **2.7 Suposiciones y dependencias**

*Se asume que el lenguaje no sufrirá cambios drásticos, ya que de ser así, habría que cambiar la información almacenada en el sistema la cual se enseña a los usuarios.*

## **3. Requerimientos de la interfaz externa**

### **3.1 Interfaces de usuario**

*La interfaz gráfica de usuario (GUI) será fácil de usar e intuitiva, centrada principalmente en un cuadro de texto, una consola en donde se mostrará la salida de la ejecución del programa, y una explicación de la funcionalidad que se está aprendiendo en ese momento. No usaremos transiciones ni muchos efectos para hacer la aplicación liviana y eficiente.*

### **3.2 Interfaces del hardware**

*No existen requerimientos especiales dado que el uso esperado es mediante navegadores web. Se espera sin embargo que sea útil en cualquier navegador que el usuario desee utilizar.*

### **3.3 Interfaces del Software**

*No existen necesidades de integración con ningún otro software aparte del que se está desarrollando. Toda salida que deba tener un formato de acuerdo a una entidad externa (caso entidades fiscales o gubernamentales) debe ajustarse en el software para su respectiva comunicación o reporte.*

### **3.4 Interfaces de comunicación**

*No existen necesidades de comunicación con otro software.*

## 4. Características del sistema

### 4.1.

*<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. Otros requerimientos no funcionales**

### **5.1 Requerimientos de desempeño**

*Esta aplicación podrá ser usada en un navegador web, preferiblemente en una computadora por cuestiones de adaptabilidad y “responsiveness”. De esta forma cualquier usuario registrado podrá disfrutar y aprovechar la aplicación y no ser limitado por otros factores.*

### **5.2 Requerimientos de seguridad externa**

*El usuario debe estar atento a sus alrededores cuando inicie su sesión en la aplicación, debido a que estará ingresando un e-mail y una contraseña, datos sensibles y personales.*

### **5.3 Requerimientos de seguridad interna**

*Confiabilidad y seguridad para proteger los datos de ingreso al sistema ya que es información sensible.*

### **5.4 Atributos de calidad de software**

*La aplicación podrá ajustar automáticamente su interfaz de usuario para adaptarse a los diferentes tamaños de pantalla en donde sea utilizada. También diseñaremos una UI fácil de usar y entendible para cualquier usuario.*

### **5.5 Reglas de la empresa**

*La información provista en el log-in o sign-in deberá ser verídica y pertenecer al usuario que la introdujo, para así poder brindar información personalizada y relevante para el usuario.*



## 6. Otros requerimientos

*Para almacenar los datos propios de cada usuario registrado, requerimos el acceso a una base de datos, probablemente utilizaremos APIs de Apple, GitHub y Google para facilitar el proceso de registro o de inicio de sesión en la aplicación.*

### **Apéndice A: Glosario**

- *Loguear(se): Proceso en el cual un usuario carga sus datos (e-mail y contraseña) para iniciar sesión en la aplicación.*
- *Site-tour: Breve tutorial sobre el uso de las herramientas y funcionalidades de una aplicación web.*

### **Apéndice B: Modelos de análisis**

*Por el momento, no vemos pertinente ningún modelo de análisis.*

### **Apéndice C: Lista “A ser discutido”**

- *Interfaz Gráfica*
  - *Organización gráfica de las funcionalidades.*
  - *Diseño de la aplicación.*
  - *Flujo de interacción del usuario (menús, secciones, etcétera).*
- *Métodos de evaluación.*
- *Métodos de aprendizaje.*
- *Implementación del site-tour.*