

**SERVIÇO NACIONAL DE APRENDIZAGEM COMERCIAL - SENAC**

**CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS**

**PROJETO INTEGRADOR**

**NeoNotus**

**Porto Alegre**

**2025**



**SERVIÇO NACIONAL DE APRENDIZAGEM COMERCIAL - SENAC**

**CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS**

**PROJETO INTEGRADOR**

**NeoNotus**

**Andrei de Souza Fischborn**

**Porto Alegre**

**2025**

**AGRADECIMENTOS**

XXXXXXXXXXXXXXXXX

.

# **1 INTRODUÇÃO**

**Sistema de Perguntas e Respostas com notícias**

## ****1.1 Apresentação Geral do Projeto****

O **NeoNotus** é um sistema gamificado de perguntas e respostas baseado em notícias reais obtidas via RSS. O objetivo do projeto é transformar o consumo de informação em uma atividade interativa e divertida. O usuário escolhe uma categoria de notícias, visualiza um resumo e, em seguida, responde perguntas geradas automaticamente com base no conteúdo.

O diferencial do sistema está na integração entre fontes de notícias confiáveis e a geração dinâmica de questões, classificadas por níveis de dificuldade (fácil, médio e difícil). O progresso do jogador é salvo, permitindo acompanhar seu desempenho e revisitar conteúdo. A aplicação será construída em Python, com uso das bibliotecas feedparser para leitura de RSS e para persistência dos dados.

## ****2 Objetivos****

### ****2.1 Objetivo Geral****

Projetar e desenvolver um sistema interativo de perguntas e respostas com base em notícias obtidas por RSS, promovendo aprendizado e engajamento do usuário.

### ****2.2 Objetivos Específicos****

* Organizar sessões de jogo com base em categorias e níveis de dificuldade;
* Importar e salvar notícias de fontes confiáveis via RSS;
* Gerar perguntas de múltipla escolha automaticamente a partir do conteúdo das notícias;
* Armazenar o histórico de respostas dos usuários;
* Exibir o progresso do jogador com base em acertos e erros.

# **3 JUSTIFICATIVA**

A ideia do projeto **NeoNotus** surgiu a partir da observação do desinteresse de grande parte dos estudantes e leitores casuais em consumir notícias diariamente, especialmente em formatos longos e densos. Em um cenário onde a desinformação circula com facilidade e o acesso à informação confiável é essencial, surgiu a proposta de transformar esse processo em algo interativo e educativo.

Foram realizadas pesquisas exploratórias em comunidades online como Reddit (subreddits voltados à educação e tecnologia), fóruns de jogos educacionais, além da análise de aplicativos como "Perguntados", "Kahoot!" e plataformas de quiz informativo. Também foram levantadas avaliações de usuários sobre o uso de ferramentas de gamificação no aprendizado. A partir dessas observações, identificou-se uma oportunidade de desenvolver um sistema que usa notícias reais como base para gerar perguntas de múltipla escolha, promovendo o engajamento com o conteúdo de maneira divertida, acessível e útil para estudantes, professores e qualquer pessoa que queira se manter informada.

# **4 METODOLOGIA**

Para o desenvolvimento do **NeoNotus**, será adotada uma abordagem baseada na metodologia ágil, mais especificamente inspirada no modelo **Scrum**, devido à sua capacidade de adaptação e organização em ciclos curtos. O desenvolvimento será dividido em **sprints semanais**, permitindo a entrega incremental das funcionalidades principais, com testes e melhorias constantes baseadas no feedback do usuário.

O foco será em construir um produto funcional desde as primeiras versões, priorizando os módulos essenciais: importação de notícias via RSS, geração automática de perguntas, armazenamento de progresso do jogador e uma interface simples e acessível para interação. Cada entrega será validada quanto à usabilidade e ao impacto da funcionalidade no objetivo central do projeto: incentivar a leitura de notícias e o aprendizado de forma lúdica.

# **5 ARQUITETURA DO SISTEMA (MODELAGEM)**

## ****5.1 Requisitos****

### ****5.1.1 Requisitos Funcionais****

* **RF01**. O sistema deve importar notícias automaticamente a partir de fontes RSS configuradas.
* **RF02**. O sistema deve permitir ao usuário selecionar uma categoria de notícia.
* **RF03**. O sistema deve gerar perguntas de múltipla escolha com base no conteúdo da notícia.
* **RF04**. O sistema deve apresentar as perguntas ao usuário uma por vez, com quatro alternativas.
* **RF05**. O sistema deve registrar as respostas do usuário e indicar se estão corretas ou incorretas.
* **RF06**. O sistema deve calcular e exibir a pontuação total da sessão.
* **RF07**. O sistema deve armazenar o histórico de sessões jogadas pelo usuário.
* **RF08**. O sistema deve permitir configurar o nível de dificuldade (fácil, médio, difícil).
* **RF09**. O sistema deve permitir visualizar o conteúdo da notícia associada à pergunta.
* **RF10**. O sistema deve exibir o feedback de desempenho ao final da sessão.
* **RF11**. O sistema deve possibilitar ao usuário reiniciar uma sessão a qualquer momento.
* **RF12**. O sistema deve indicar quais perguntas foram acertadas e quais foram erradas.
* **RF13**. O sistema deve permitir a atualização das fontes RSS cadastradas.
* **RF14**. O sistema deve gerar novas perguntas a cada sincronização com as notícias mais recentes.

### ****5.1.2 Requisitos Não Funcionais****

* **RNFO1**: O sistema deve rodar locamente, sem precisar de login.
* **RNFO2:** O sistema deve ser desenvolvido utilizando a tecnologia de programação Python.
* **RNFO3**: O sistema deve utilizar o banco de dados SQlite3.
* **RNFO4**: O sistema deve ter uma experiência de usuário automatizada.

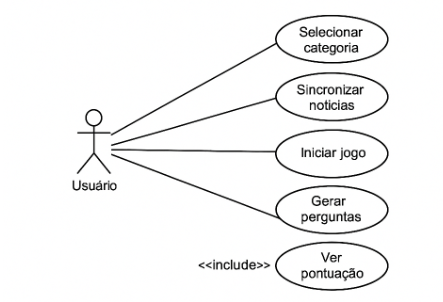
### 5.3 Modelagem Funcional

A modelagem funcional descreve as principais interações entre os usuários e o sistema com base em suas funcionalidades. Abaixo, listamos os atores identificados no sistema:

#### **Lista dos Atores**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| | **Ator** | **Descrição** | | --- | --- | | Usuário | O usuário pode iniciar sessões de estudo e cadastrar playlists, além de editá-las, excluí-las e acompanhar o progresso nas atividades propostas. | |

### 5.4 Diagrama de Caso de Uso



### 5.4.1 Detalhamento dos Casos de Uso

Nesta seção será apresentado o detalhamento dos seguintes casos de uso:  
 CSU01 – Iniciar Sessão de Jogo  
 CSU02 – Cadastrar Fonte RSS  
 CSU03 – Editar Fonte RSS  
 CSU04 – Excluir Fonte RSS  
 CSU05 – UC Gerenciar Fontes  
 CSU06 – Configurar Dificuldade  
 CSU07 – UC Gerar Perguntas  
 CSU08 – Responder Pergunta  
 CSU09 – Calcular Pontuação  
 CSU10 – Exibir Resultados  
 CSU11 – Sincronizar Notícias  
 CSU12 – Visualizar Notícia  
 CSU13 – Acessar Histórico  
 CSU14 – Exportar Relatórios

### CSU01 – Iniciar Sessão de Jogo

| **Caso de Uso** | **1 – Iniciar Sessão de Jogo** |
| --- | --- |
| **Objetivo** | Permitir ao usuário começar uma nova sessão de perguntas baseadas em notícias. |
| **Ator** | Usuário |
| **Pré-condições** | O sistema deve já ter perguntas geradas a partir de notícias atualizadas. |

**Cenário Principal**

1. O usuário seleciona "Iniciar Sessão de Jogo";
2. O sistema carrega as perguntas conforme a dificuldade configurada;
3. A sessão é iniciada e a primeira pergunta é exibida.

### CSU02 – Cadastrar Fonte RSS

| **Caso de Uso** | **2 – Cadastrar Fonte RSS** |
| --- | --- |
| **Objetivo** | Permitir que o usuário adicione novas fontes RSS para obter notícias. |
| **Ator** | Usuário |
| **Pré-condições** | A URL deve ser válida e corresponder a um feed RSS. |

**Cenário Principal**

1. O usuário acessa o menu "Fontes RSS" e clica em "Adicionar nova fonte";
2. O sistema solicita a URL da nova fonte;
3. O usuário informa a URL e confirma;
4. O sistema valida a fonte e salva.

**Fluxo Alternativo**  
A. O usuário informa uma URL inválida:

1. O sistema exibe mensagem de erro e solicita nova entrada.

### CSU03 – Editar Fonte RSS

| **Caso de Uso** | **3 – Editar Fonte RSS** |
| --- | --- |
| **Objetivo** | Permitir ao usuário alterar uma URL de fonte já cadastrada. |
| **Ator** | Usuário |
| **Pré-condições** | A fonte deve já estar cadastrada. |

**Cenário Principal**

1. O usuário acessa a lista de fontes e seleciona "Editar";
2. O sistema permite a alteração da URL;
3. O usuário altera e confirma;
4. O sistema atualiza a fonte e exibe mensagem de sucesso.

### CSU04 – Excluir Fonte RSS

| **Caso de Uso** |  | **4 – Excluir Fonte RSS** |
| --- | --- | --- |
| **Objetivo** |  | Permitir que o usuário remova uma fonte cadastrada. |
| **Ator** |  | Usuário |
| **Pré-condições** |  | A fonte deve existir e estar listada. |

**Cenário Principal**

1. O usuário acessa a lista de fontes;
2. O usuário seleciona a opção "Excluir";
3. O sistema solicita confirmação;
4. O usuário confirma e o sistema remove a fonte.

### CSU05 – UC Gerenciar Fontes

| **Caso de Uso** | **5 – UC Gerenciar Fontes RSS** |
| --- | --- |
| **Objetivo** | Permitir ao usuário visualizar, adicionar, editar ou excluir fontes RSS. |
| **Ator** | Usuário |
| **Pré-condições** | O sistema deve estar em funcionamento local. |

**Cenário Principal**

1. O usuário acessa a aba de Gerenciamento de Fontes;
2. O sistema exibe todas as fontes cadastradas;
3. O usuário escolhe entre adicionar, editar ou excluir uma fonte;
4. O sistema executa a ação correspondente.

### CSU06 – Configurar Dificuldade

| **Caso de Uso** | **6 – Configurar Dificuldade** |
| --- | --- |
| **Objetivo** | Permitir que o usuário defina o nível de dificuldade (fácil, médio, difícil) das perguntas. |
| **Ator** | Usuário |
| **Pré-condições** | O sistema deve ter categorias de dificuldade implementadas. |

**Cenário Principal**

1. O usuário acessa as configurações do jogo;
2. O sistema exibe as opções de dificuldade;
3. O usuário seleciona uma das opções;
4. O sistema salva a configuração e exibe confirmação.

### CSU07 – UC Gerar Perguntas

| **Caso de Uso** | **7 – UC Gerar Perguntas** |
| --- | --- |
| **Objetivo** | Gerar automaticamente perguntas baseadas no conteúdo das notícias mais recentes. |
| **Ator** | Sistema |
| **Pré-condições** | O sistema deve estar sincronizado com pelo menos uma fonte RSS válida. |

**Cenário Principal**

1. O sistema verifica se há novas notícias nas fontes RSS;
2. O conteúdo das notícias é analisado por inteligência textual;
3. O sistema gera perguntas de múltipla escolha com base nas manchetes, subtítulos e conteúdo.

### CSU08 – Responder Pergunta

| **Caso de Uso** | **8 – Responder Pergunta** |
| --- | --- |
| **Objetivo** | Permitir que o usuário selecione uma das alternativas de resposta da pergunta apresentada. |
| **Ator** | Usuário |
| **Pré-condições** | O jogo deve estar em execução e uma pergunta deve estar ativa. |

**Cenário Principal**

1. O sistema exibe uma pergunta com 4 alternativas;
2. O usuário escolhe uma das alternativas;
3. O sistema armazena a resposta e verifica se está correta.

### CSU09 – Calcular Pontuação

| **Caso de Uso** | **9 – Calcular Pontuação** |
| --- | --- |
| **Objetivo** | Avaliar o desempenho do usuário com base nas respostas corretas. |
| **Ator** | Sistema |
| **Pré-condições** | O usuário deve ter respondido pelo menos uma pergunta. |

**Cenário Principal**

1. Ao final da sessão, o sistema contabiliza as respostas corretas;
2. A pontuação é calculada de acordo com o nível de dificuldade e número de acertos;
3. O resultado é exibido na tela final.

### CSU10 – Exibir Resultados

| **Caso de Uso** | **10 – Exibir Resultados** |
| --- | --- |
| **Objetivo** | Mostrar ao usuário seu desempenho após uma sessão de perguntas. |
| **Ator** | Usuário |
| **Pré-condições** | A sessão de jogo deve ter sido finalizada. |

**Cenário Principal**

1. O sistema apresenta a pontuação final;
2. O usuário pode visualizar estatísticas como acertos, erros, tempo médio por pergunta, etc.

### CSU11 – Sincronizar Notícias

| **Caso de Uso** | **11 – Sincronizar Notícias** |
| --- | --- |
| **Objetivo** | Atualizar o banco local com as notícias mais recentes das fontes cadastradas. |
| **Ator** | Sistema |
| **Pré-condições** | O sistema deve estar conectado à internet. |

**Cenário Principal**

1. O sistema acessa todas as URLs RSS cadastradas;
2. Ele recupera as últimas notícias e armazena no banco de dados;
3. Notícias repetidas não são cadastradas novamente.

### CSU12 – Visualizar Notícia

| **Caso de Uso** | **12 – Visualizar Notícia** |
| --- | --- |
| **Objetivo** | Permitir que o usuário acesse o conteúdo completo de uma notícia. |
| **Ator** | Usuário |
| **Pré-condições** | A notícia deve estar disponível no banco. |

**Cenário Principal**

1. O usuário acessa o histórico de perguntas ou feed de notícias;
2. O sistema exibe o conteúdo completo da notícia (título, resumo, link, data, etc).

### CSU13 – Acessar Histórico

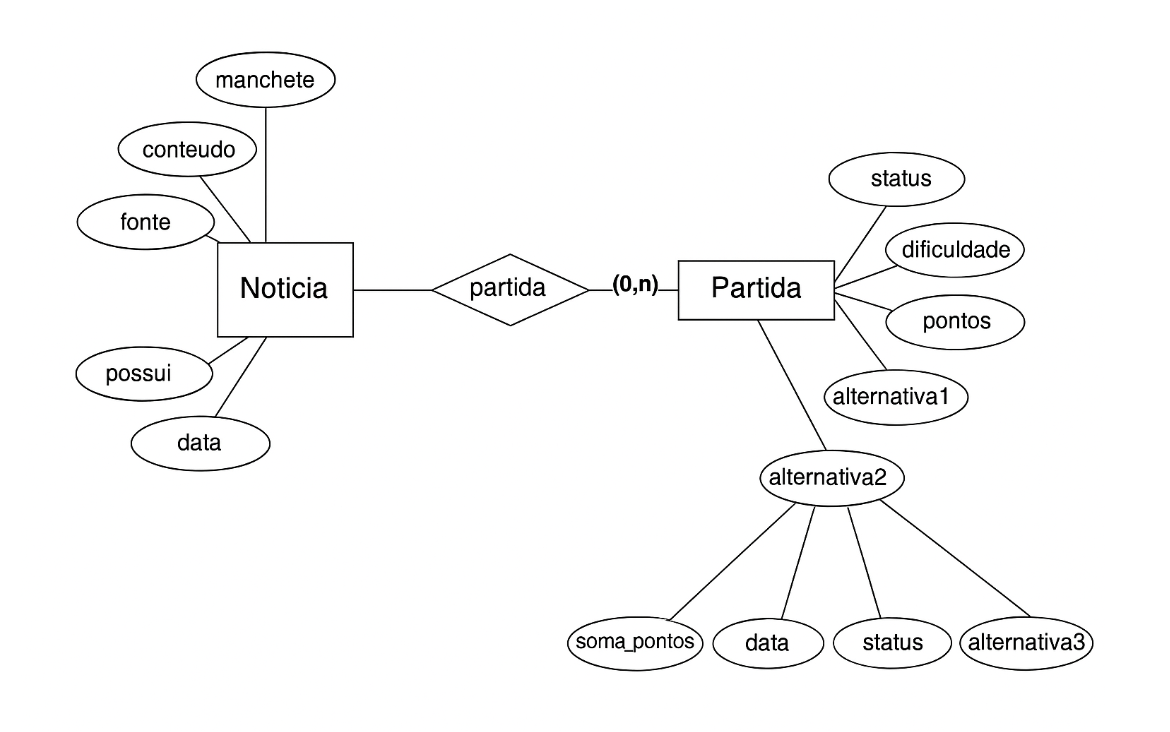
| **Caso de Uso** | **13 – Acessar Histórico** |
| --- | --- |
| **Objetivo** | Permitir que o usuário veja seu desempenho e progresso ao longo do tempo. |
| **Ator** | Usuário |
| **Pré-condições** | O usuário deve ter realizado pelo menos uma sessão de jogo. |

**Cenário Principal**

1. O usuário acessa a aba "Histórico";
2. O sistema exibe a lista de sessões anteriores com pontuação, data e dificuldade.

## 6 BANCO DE DADOS

### 6.1 Modelo Conceitual



### 6.2 Modelo Lógico

## 6.3 Modelo Físico

### Tabela Notícia

CREATE TABLE IF NOT EXISTS Noticia (

id INTEGER PRIMARY KEY AUTOINCREMENT,

manchete TEXT NOT NULL,

conteudo TEXT NOT NULL,

status TEXT NOT NULL DEFAULT 'Este dia',

fonte TEXT NOT NULL,

data TEXT NOT NULL

### Tabela Partida

CREATE TABLE IF NOT EXISTS Partida (

id INTEGER PRIMARY KEY AUTOINCREMENT,

id\_noticia INTEGER NOT NULL,

soma\_pontos INTEGER NOT NULL,

data TEXT NOT NULL,

status BOOLEAN NOT NULL DEFAULT 0,

dificuldade TEXT NOT NULL,

pontos INTEGER NOT NULL,

enunciado TEXT NOT NULL,

alternativa\_certa TEXT NOT NULL,

alternativa1 TEXT NOT NULL,

alternativa2 TEXT NOT NULL,

alternativa3 TEXT NOT NULL,

FOREIGN KEY (id\_noticia) REFERENCES Noticia(id)

### ****7 PROJETO PYTHON****

#### **7.1 Tutorial de Instalação do Python**

Para executar o projeto, é necessário que o Python esteja instalado no computador.

**Passos para instalação do Python:**

1. Acesse o site oficial: <https://www.python.org/downloads>
2. Baixe a versão mais recente do Python compatível com seu sistema operacional.
3. Durante a instalação, marque a opção **“Add Python to PATH”**.
4. Finalize a instalação e, para confirmar, abra o terminal (Prompt de Comando ou PowerShell) e digite:

css

CopiarEditar

python --version

**Bibliotecas utilizadas (instalar com pip):**

bash

CopiarEditar

pip install feedparser

pip install tk

#### **7.2 Projeto com CRUD**

O projeto possui operações de CRUD aplicadas às entidades **Notícia** e **Partida**.

##### **7.2.1 Telas**

* **Tela Inicial:** Apresenta opções de iniciar jogo, visualizar ranking ou sair.
* **Tela de Cadastro de Fontes:** Permite adicionar novos feeds RSS.
* **Tela de Perguntas:** Exibe perguntas geradas a partir das notícias.
* **Tela de Resultados:** Mostra pontuação ao final da partida.

##### **7.2.2 Conexão com Banco**

A conexão com o banco é feita via **SQLite3**, com as seguintes tabelas:

python

CopiarEditar

import sqlite3

con = sqlite3.connect('jogo\_noticias.db')

cur = con.cursor()

Exemplo de criação de tabela:

python

CopiarEditar

cur.execute('''

CREATE TABLE IF NOT EXISTS Noticia (

id INTEGER PRIMARY KEY AUTOINCREMENT,

manchete TEXT NOT NULL,

conteudo TEXT NOT NULL,

status TEXT NOT NULL DEFAULT 'Este dia',

fonte TEXT NOT NULL,

data TEXT NOT NULL

)

''')

cur.execute('''

CREATE TABLE IF NOT EXISTS Partida (

id INTEGER PRIMARY KEY AUTOINCREMENT,

id\_noticia INTEGER NOT NULL,

soma\_pontos INTEGER NOT NULL,

data TEXT NOT NULL,

status BOOLEAN NOT NULL DEFAULT 0,

dificuldade TEXT NOT NULL,

pontos INTEGER NOT NULL,

enunciado TEXT NOT NULL,

alternativa\_certa TEXT NOT NULL,

alternativa1 TEXT NOT NULL,

alternativa2 TEXT NOT NULL,

alternativa3 TEXT NOT NULL,

FOREIGN KEY (id\_noticia) REFERENCES Noticia(id)

)

''')

#### **7.3 Manual de Utilização**

1. **Abrir o sistema:** Execute o arquivo principal (main.py ou equivalente).
2. **Cadastrar Fontes:** Adicione links RSS de sites confiáveis (como G1, BBC etc.).
3. **Iniciar Jogo:** O sistema vai gerar automaticamente perguntas com base nas notícias mais recentes.
4. **Responder Perguntas:** O usuário escolhe a alternativa correta entre 4 opções.
5. **Visualizar Resultado:** Ao fim da rodada, é exibida a pontuação total.
6. **Consultar Histórico:** Através da tela de ranking ou histórico, é possível revisar partidas anteriores.