



# Trabalho de Web – *Athena*

Banco de materiais de estudo colaborativo

## **Disciplina**

**Desenvolvimento de Software para Web**

Profa. Valeria Maria da Silva Pinheiro

## **Equipe**

Alexsandro Martins Alves – 541581

Jona Ferreira de Sousa – 539700

Luan Roger Santos Clementino – 517173

Vitor Costa de Sousa – 536678

Yvens Almeida Girão – 542481

## **Links Úteis**

[Github](#)

[Athena](#)

# Sumário

<b>1. Definição da Persona</b>	<b>3</b>
1.1. Quem	3
1.2. Contexto	4
1.3. Experiências Prévias/Preferências	4
1.4. Problemas/Necessidades/Desejos	5
1.5. Soluções Existentes	7
<b>2. Especificação dos requisitos</b>	<b>9</b>
2.1. Requisitos Funcionais	9
2.2. Requisitos não funcionais	10
2.3. Regras de Negócio	10
<b>3. Prototipação</b>	<b>11</b>
<b>4. Desenvolvimento da aplicação</b>	<b>12</b>

# 1. Definição da Persona



Persona: Bryan, 23 anos

## 1.1. Quem

### 1.1.1. Qual a profissão dele (a)?

Atualmente, trabalha informalmente como desenvolvedor de software backend em uma software house que ele iniciou com amigos.

### 1.1.2. Qual o nível de escolaridade dele (a)?

Ensino médio completo, mas está fazendo uma formação em Análise e Desenvolvimento de Sistemas para ampliar seus conhecimentos na área de software.

### 1.1.3. Como ele(a) se descreve?

Se considera proativo e competente com as atividades que lhe são repassadas; possui interesse em aprender coisas novas; habilidades de comunicação clara com os envolvidos no projeto.

### 1.1.4. Quais os medos/preocupações/frustrações dele (a)? Por quê?

Sente-se “travado” ao lidar com conceitos complexos, considerando o tempo que precisa despender para isso; não consegue gerenciar o tempo muito bem quando há muitas

atividades a serem feitas; não lida muito bem com erros, geralmente não aprendendo com eles.

## 1.2. Contexto

### 1.2.1. Quais tarefas da sua rotina ele (a) realiza utilizando aplicações (web, mobile, desktop)?

Utiliza aplicações de software diariamente, como: para consultar documentações e ferramentas pelo navegador do PC ou smartphone; acessar sua conta bancária via aplicativo mobile nativo; se comunicar com sua equipe via plataforma de reunião online.

## 1.3. Experiências Prévias/Preferências

### 1.3.1. Quais partes das aplicações citadas ele (a) gosta? Por quê?

Ele aprecia interfaces limpas, com menus bem organizados, documentações com exemplos práticos e respostas rápidas às suas ações. Plataformas como o navegador e editores de código com recursos de autocompletar e documentação embutida ajudam muito em seu fluxo de trabalho. Em apps bancários, valoriza quando as informações estão dispostas de forma objetiva e de fácil acesso.

### 1.3.2. Quais partes dessas aplicações ele (a) não gosta? Por quê?

Fica facilmente frustrado com interfaces confusas, excesso de opções que desviam o foco e documentações muito teóricas ou genéricas. Ele tende a se perder quando precisa seguir caminhos longos para concluir uma tarefa simples. Também se incomoda com softwares pesados, que exigem muito da máquina ou são lentos no carregamento.

### 1.3.3. Em quais dispositivos ele (a) utiliza essas aplicações?

Utiliza principalmente o notebook para tarefas de desenvolvimento e estudo, mas recorre frequentemente ao smartphone para consultas rápidas, leitura de documentação ou operações bancárias. O uso é dividido entre ambiente desktop (Windows) e mobile (Android).

#### 1.3.4. Como ele (a) aprende a usar um software (perguntando a outras pessoas, pesquisando, lendo manuais, mexendo)?

Prefere testar na prática e descobrir os recursos por conta própria. Quando é preciso, utiliza modelos de IA ou busca vídeos no YouTube . Raramente pergunta diretamente para alguém, a menos que esteja em um contexto de equipe ou colaboração.

#### 1.3.5. Ele (a) gosta de fazer as coisas passo a passo ou prefere atalhos?

Gosta de atalhos e otimizações sempre que já está familiarizado com o sistema. No entanto, em ambientes novos ou complexos, segue o passo a passo inicialmente até se sentir seguro.

#### 1.3.6. Ele (a) lembra melhor das coisas vendo imagens/representações gráficas, lendo texto ou ouvindo áudio?

Lembra melhor ao ver imagens, esquemas visuais e fluxogramas. Textos longos não são tão eficazes, e ele não costuma usar recursos em áudio.

#### 1.3.7. No mundo virtual ele (a) gosta de interação social e compartilhamento de acontecimentos ou prefere se manter reservado?

Apesar de trabalhar em equipe, é mais reservado no mundo virtual. Participa de fóruns ou comunidades apenas quando necessário e não costuma compartilhar detalhes da vida pessoal online.

### 1.4. Problemas/Necessidades/Desejos

#### 1.4.1. Quais problemas ele/ela enfrenta na sua rotina que podem ser resolvidos pela aplicação a ser projetada?

- **Dificuldade em entender conceitos complexos de forma isolada:** Ele perde muito tempo buscando informações em fontes dispersas. Muitas vezes, os conteúdos são técnicos demais ou não têm exemplos práticos, o que torna o aprendizado mais lento e frustrante.

- **Sobrecarga de materiais disponíveis na internet sem curadoria:** Ele encontra muito conteúdo online, mas tem dificuldade em saber o que é confiável, atualizado e relevante para o seu nível de conhecimento.
- **Falta de tempo para organizar os próprios materiais de estudo:** Ele não mantém um repositório próprio estruturado de anotações, e isso o atrapalha quando precisa revisar conceitos ou compartilhar com colegas de equipe.
- **Pouca exposição a outras formas de raciocínio:** Por estudar de forma independente, sente falta de ver como outras pessoas resolvem o mesmo problema, o que o ajudaria a desbloquear sua própria compreensão.
- **Falta de incentivo ou automotivação:** Por estar fora de um ambiente acadêmico formal, sente dificuldade em manter uma rotina disciplinada de estudos sem algo que o estimule a continuar.

#### 1.4.2. O que a aplicação a ser projetada pode ter para ajudar a resolver estes problemas?

- **Sistema de curadoria:** Funcionalidade de avaliação para que ele possa encontrar os conteúdos mais relevantes e confiáveis.
- **Materiais em múltiplos formatos (texto, esquemas visuais, códigos interativos):** Como ele tem memória visual e gosta de exemplos práticos, a aplicação pode destacar conteúdos com diagramas, fluxogramas, e snippets de código prontos para testar.
- **Sistema de bookmarks e anotações pessoais integradas:** Permite salvar conteúdos para revisar depois, adicionar notas rápidas e organizar seu próprio banco pessoal dentro da aplicação.
- **Sistema de recompensas leve (gamificação opcional):** Para mantê-lo motivado, a aplicação pode oferecer pequenas conquistas (ex: "contribuiu com 3 materiais", "completou uma trilha de estudos", "comentou em 5 materiais"). (parece uma funcionalidade interessante)

## 1.5. Soluções Existentes

1.5.1. Existem outras aplicações que resolvem os problemas (ou partes dos problemas) citados? Quais? De que forma estas aplicações ajudam?

- **Notion e Google Drive (compartilhamento informal):** Usados para guardar e compartilhar anotações com amigos ou colegas de equipe. No entanto, essas ferramentas não foram feitas especificamente para estudo colaborativo, e exigem organização manual.
- **GitHub (repositórios de estudo):** Utilizado para acessar ou manter repositórios com conteúdos técnicos, listas de exercícios, roteiros de estudo e códigos. Embora útil, não é amigável para quem está nos níveis iniciais e tem uma curva de aprendizado mais técnica.
- **Stack Overflow / Dev.to / Medium:** Plataformas com conteúdos úteis, mas os materiais são isolados (não formam uma sequência de aprendizado) e geralmente voltados para resolução de dúvidas ou artigos mais avançados.
- **Chat GPT / Google Gemini (IAs generativas):** Ferramentas que permitem sintetizar conteúdos com diferentes níveis de profundidade e linguagem. Auxiliam nas fases exploratórias do aprendizado, oferecendo explicações iniciais, exemplos e resumos. No entanto, o uso constante pode criar dependência, e muitas vezes falta validação comunitária ou referência de fonte, o que compromete a confiabilidade e o senso crítico do usuário.

1.5.2. Quais características positivas/essenciais dessas outras aplicações?

- **Organização por tópicos e tags** facilita a busca.
- **Conteúdos didáticos com exemplos práticos** e, às vezes, interativos.
- **Foco em tecnologia** com linguagem acessível e visual moderno.
- **Trilhas de conhecimento ou sequências lógicas** ajudam a manter um ritmo de aprendizado.

- **Respostas rápidas e adaptadas ao nível do usuário**, no caso das IAs generativas.

### 1.5.3. Quais características negativas/dispensáveis dessas outras aplicações?

- **Falta de personalização no aprendizado:** muitas plataformas não se adaptam ao estilo e ritmo do usuário.
- **Experiência fragmentada:** precisa alternar entre várias ferramentas para estudar, organizar e compartilhar conteúdos.
- **Dependência de inteligência artificial:** pode limitar a autonomia do usuário se não for usada com criticidade.
- **Pouca abertura para colaboração direta:** não é possível editar, complementar ou comentar nos materiais de forma interativa em muitas dessas plataformas.



## 2. Especificação dos requisitos

### 2.1. Requisitos Funcionais

Código	Descrição
RF1	O sistema deve permitir que o usuário crie uma conta por meio de uma interface de login. Utilizando o email, definindo uma senha e um nome de usuário.
RF2	O sistema deve permitir que o usuário se autentique por meio de email e senha em uma interface de login.
RF3	O sistema deve exibir ao usuário autenticado, em um dashboard, os artigos/arquivos que foram feitos upload para a plataforma de um determinado usuário autenticado.
RF4	Deve ser possível fazer o upload novos arquivos para a plataforma, definindo o nome do arquivo, título do artigo, autor do artigo e tags (palavras-chave)
RF5	Deve ser possível explorar os arquivos/artigos divulgados por outros usuários em uma interface intitulada “Explorar”
RF6	Deve ser possível realizar pesquisas por título do artigo, autor do artigo ou tags
RF6	Deve ser exibido uma aba “Em alta”, com os materiais mais favoritos.
RF7	Deve ser possível visualizar as informações do artigo/arquivo que foi feito o upload por outro usuário
RF8	Deve ser possível realizar o download do arquivo de um artigo específico.
RF9	Deve ser possível deletar os artigos do usuário autenticado, por meio do dashboard
RF10	Deve ser possível atualizar as informações de título do artigo, nome do autor e arquivo do artigo que foi feito o upload posteriormente pelo usuário autenticado
RF11	Deve ser possível encerrar a seção de um usuário autenticado
RF12	Deve ser possível um usuário favoritar artigos

## 2.2. Requisitos não funcionais

Código	Descrição
RNF1	As informações e registros dos metadados dos arquivos devem ser armazenadas em um banco de dados relacional PostgreSQL
RNF2	As informações do usuário como nome de usuário, email, senha devem ser armazenadas em um banco de dados relacional PostgreSQL
RNF3	As senhas dos usuário devem estar criptografadas usando scrypt para armazenar no banco
RNF4	Os blobs dos arquivos devem ser armazenados e gerenciados pelo MinIO
RNF5	A comunicação para upload de arquivos devem ocorrer por meio de HTTPS
RNF6	O sistema deve permitir que até 10 usuários façam um upload simultaneamente sem degradar a performance.

## 2.3. Regras de Negócio

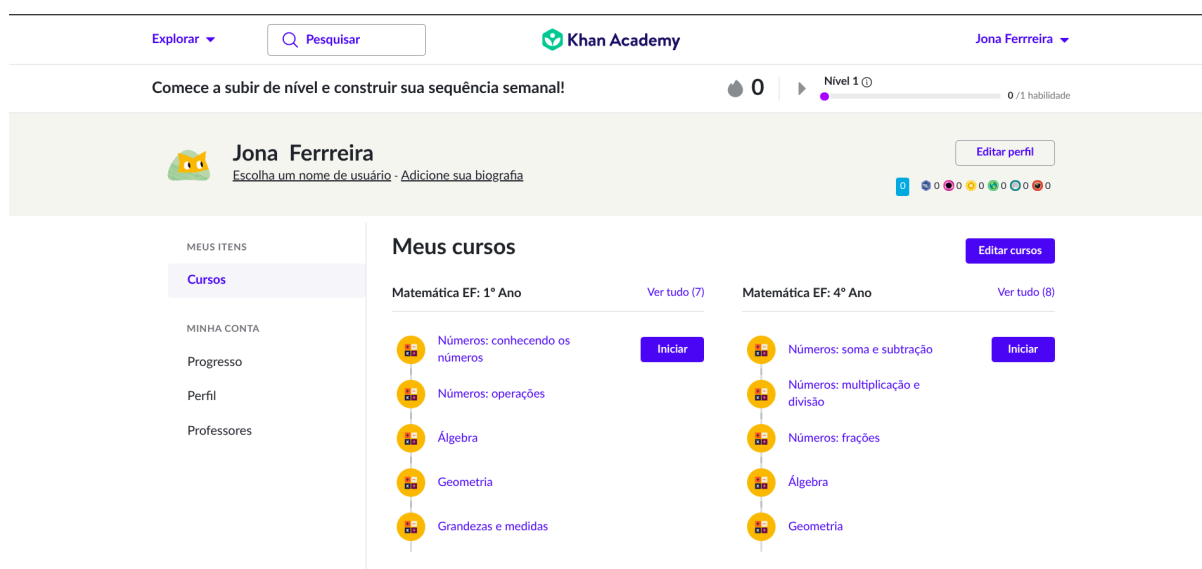
Código	Descrição
RN1 – Acesso ao conteúdo	<ul style="list-style-type: none"><li>Qualquer visitante da plataforma pode visualizar o repositório de arquivos.</li><li>Somente usuários autenticados podem realizar o download de conteúdos da plataforma.</li></ul>
RN2 – Gerenciamento de conteúdo	<ul style="list-style-type: none"><li>Somente usuários autenticados podem fazer uploads ou remover seus próprios conteúdos.</li><li>Administradores têm total controle do gerenciamento de conteúdos da plataforma, podendo deletar e adicionar qualquer arquivo.</li></ul>
RN3 – Arquivos suportados	<ul style="list-style-type: none"><li>Deve-se restringir as extensões de arquivos para upload, como: .pdf, .docx, .md, .txt.</li><li>O sistema deve limitar arquivos maiores que 10mb.</li></ul>
RN4 – Favoritar	<ul style="list-style-type: none"><li>Os materiais podem ser favoritados apenas por usuários autenticados.</li></ul>

### 3. Prototipação

Para a prototipação do sistema, nos baseamos em alguns dos sites que, de certa forma, tangencia o conteúdo da nossa plataforma, como **Seneca Learning** e **Khan Academy**.

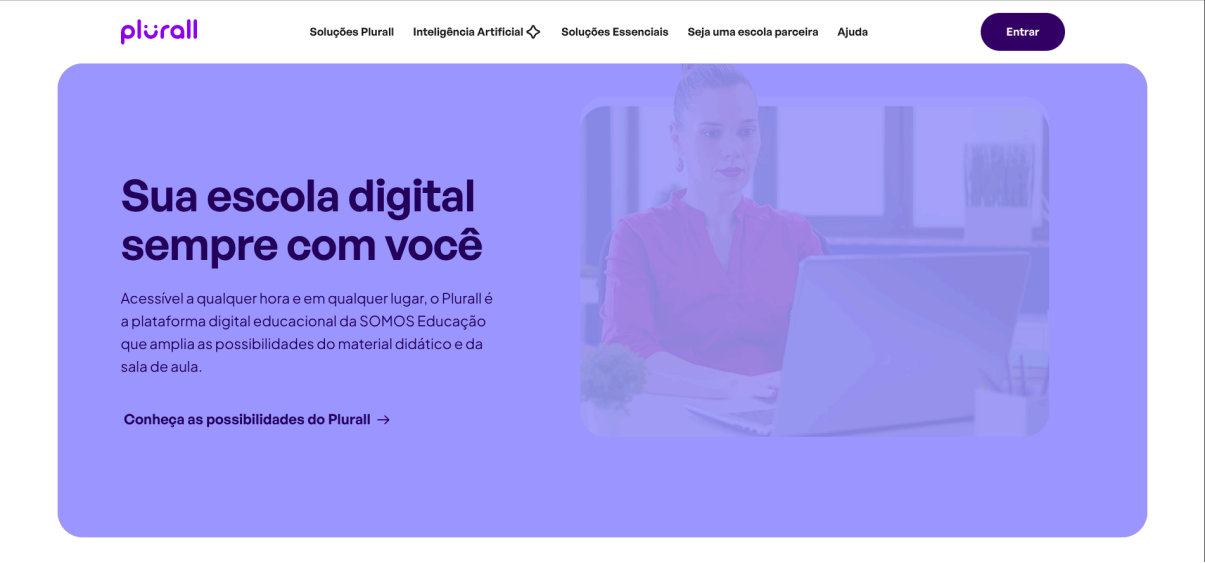


Tela principal da plataforma **Seneca Learning**.



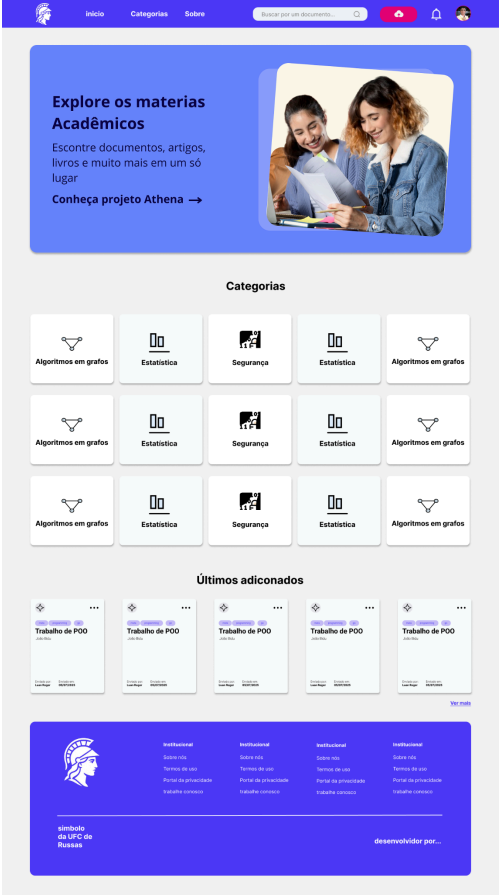
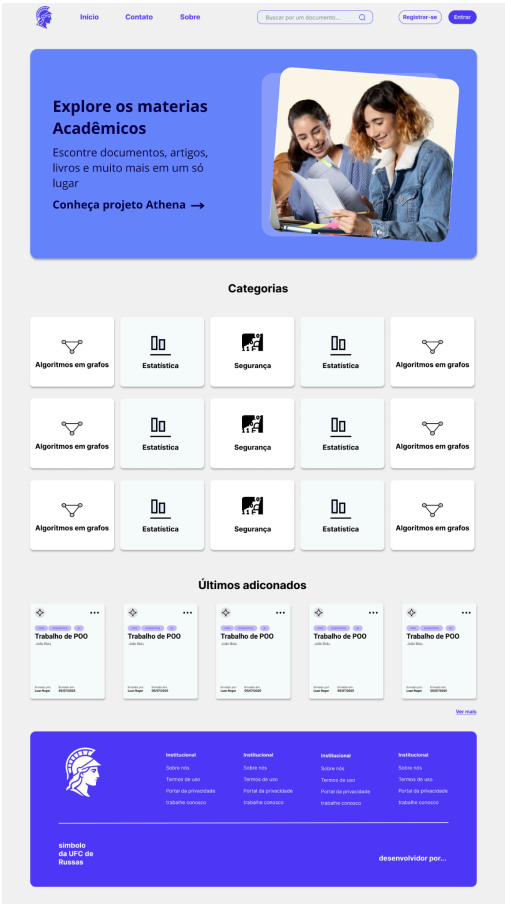
Tela do usuário da plataforma **Khan Academy**.


Já em termos de aparência, consideramos importante a definição de uma paleta de cores atrativas e dado que o escopo da aplicação envolve intrinsecamente os estudos, é coerente o uso do **roxo** como cor primária. Diante disso, tomamos como inspiração o site **Plurall** que utiliza essa mesma paleta e alguns outros padrões de design que incorporamos.



Tela principal do site **Plurall**.

Portanto, tivemos como resultado as seguintes prototipações, seja para *desktop* ou para *mobile*:





Login


E-mail

Senha

Esqueceu a senha?

Entrar

Não possui conta? [Registre-se](#)



Registro


Nome de Usuário

E-mail

Senha

Confirmar


Já possui conta? [Entre aqui](#)





Inicio

Contato

Sobre







Home > Categorias > Estatística

Abra

Buscar

Trabalho de POO

Trabalho de POO

Trabalho de POO

Trabalho de POO

Trabalho de POO

Trabalho de POO

Trabalho de POO


Trabalho de POO

Trabalho de POO

Trabalho de POO

Trabalho de POO

Trabalho de POO



Institucional

Sobre nós

Termos de uso

Portal de privacidade

Trabalhe conosco

Institucional

Sobre nós

Termos de uso

Portal de privacidade

Trabalhe conosco

Institucional

Sobre nós

Termos de uso

Portal de privacidade

Trabalhe conosco

Institucional

Sobre nós


Termos de uso

Portal de privacidade

Trabalhe conosco

simbolo da UFC de Russas


desenvolvidor por...





Inicio

Contato

Sobre







Home > Categorias > Meus Documentos

Abra

Buscar

Editar

Remover

Trabalho de POO

Trabalho de POO

Trabalho de POO

Trabalho de POO

Trabalho de POO

Trabalho de POO

Trabalho de POO


Trabalho de POO

Trabalho de POO

Trabalho de POO

Trabalho de POO

Trabalho de POO



Institucional

Sobre nós

Termos de uso

Portal de privacidade

Trabalhe conosco

Institucional

Sobre nós

Termos de uso

Portal de privacidade

Trabalhe conosco

Institucional

Sobre nós

Termos de uso

Portal de privacidade

Trabalhe conosco

Institucional

Sobre nós

Termos de uso

Portal de privacidade

Trabalhe conosco

simbolo da UFC de Russas

desenvolvidor por...

Novo upload

Nome do arquivo

Título

Autor

Tags

Cancelar

Enviar

Editar arquivo

Nome do arquivo


Título

Autor

Tags

Cancelar

Salvar




### Login

E-mail

Senha

**Entrar**

Esqueceu a senha?  
Não possui conta? [Registre-se](#)



### Registro


Nome de Usuário

E-mail

Senha

**Confirmar**

Já Possui conta? [Entre aqui](#)




### Explore os materiais Acadêmicos


Encontre documentos, artigos, livros e muito mais em um só lugar.

**Conheça projeto Athena** →



- ### Categorias
- 

Algoritmos em grafos
- 

Estatística
- 

Segurança

## 4. Desenvolvimento da aplicação

Para o desenvolvimento do projeto, buscamos utilizar tecnologias modernas e eficientes para garantir uma aplicação funcional, segura e com boa experiência para o usuário.

Começamos com a decisão das tecnologias, optando por utilizar o [Next.js](#) (meta-framework para React que expande as ferramentas de trabalho) como ambiente principal do desenvolvimento da aplicação, pela sua flexibilidade e suporte a aplicações fullstack. Para a parte de banco de dados, escolhemos o [PostgreSQL](#), integrando com o *headless* ORM [Drizzle](#), que oferece tipagem segura e escrita mais clara das queries em **TypeScript**.

Se tratando do serviço de autenticação de usuários, foi implementada utilizando o [Better Auth](#), que oferece segurança e facilidade de integração com o **Next.js**, garantindo que apenas usuários logados possam interagir com o sistema. Na parte visual, utilizamos o [DaisyUI](#), um plugin do [TailwindCSS](#) que acelera a criação de interfaces com componentes prontos, mantendo um design limpo e padronizado.

Para o armazenamento de arquivos (como PDFs, imagens ou outros materiais), usamos o [MinIO](#), uma solução de armazenamento compatível com **Amazon S3**, ideal para lidar com uploads e downloads de forma segura.

A divisão de tarefas foi feita de forma colaborativa entre os membros da equipe, separando as responsabilidades entre interface, backend, banco de dados e integração com o storage.

Por fim, a aplicação foi hospedada na plataforma [Railway](#), que oferece deploy contínuo e fácil integração com banco de dados e storage, facilitando a publicação do sistema.