

# WAD

## WEB APPLICATION DOCUMENT

SCHOOLAHEAD

Autores: Antonio Angelo Teixeira

Eduardo França Porto

Elisa Flemer

Matheus Macedo Santos

Pedro Gattai

Tainara Rodrigues Teixeira

Victor Carvalho

Data de criação: 03/05/2022

## Controle do Documento

### Histórico de revisões

Data	Autor	Versão	Resumo da atividade
03/05/2022	RIGHTML	0.1	Atualizamos as seções referentes aos entregáveis da semana dois.
13/05/2022	RIGHTML	0.2	Adicionado o guia de estilos.
30/05/2022	RIGHTML	0.3	Adicionado diagrama de entidades, relacionamentos do banco de dados, arquitetura da solução e documentação da API

---

# Sumário

## [Visão Geral do Projeto](#)

[Empresa](#)

[O Problema](#)

[Objetivos](#)

[Objetivos gerais](#)

[Objetivos específicos](#)

[Descritivo da Solução](#)

[Partes Interessadas](#)

## [Análise do Problema](#)

[Análise da Indústria](#)

[Análise do cenário: Matriz SWOT](#)

[Proposta de Valor: Value Proposition Canvas](#)

[Matriz de Risco](#)

## [Requisitos do Sistema](#)

### [3.1. Persona](#)

[Juliana Santos](#)

[27 anos](#)

[Beth Silva](#)

[38 anos](#)

[Diretora de escola](#)

[Mini biografia](#)

[Apaixonada por educação](#)

[Quase 30 anos de experiência com docência](#)

[Formada em história](#)

[Nível básico de entendimento da tecnologia](#)

[Paulo Freitas](#)

### [3.2 Histórias dos usuários \(user stories\)](#)

## [Arquitetura do Sistema](#)

[Módulos do Sistema e Visão Geral \(Big Picture\)](#)

[Descrição dos Subsistemas](#)

[Users Stories dos subsistemas](#)

[Requisitos de software](#)

[Tecnologias Utilizadas](#)

## [UX e UI Design](#)

[Wireframe + Storyboard](#)

[Design de Interface - Guia de Estilos](#)

## [Projeto de Banco de Dados](#)

[Modelo Conceitual](#)

[Modelo Lógico](#)

## [Teste de Software](#)

[Teste Unitário](#)

[Teste de Usabilidade](#)

## [Análise de Dados](#)

## [Manuais](#)

[Manual do Usuário](#)

[Manual do Administrador](#)

## [Referências](#)

## [Apêndice](#)

# **1. Visão Geral do Projeto**

## **1.1. Empresa**

A Falconi Consultores de Resultados oferece consultoria para pequenas e médias empresas com os principais objetivos de aperfeiçoamento da gestão, definição de metas e ratificação de resultados. Seu está direcionado para um mercado altamente aquecido. Com mais de 700 consultores, é a maior empresa desse ramo no mercado nacional.

## **1.2. O Problema**

A maior dificuldade das escolas no quesito construção de metas reais é que apenas os dados não são suficientes para mostrar o que deverá ser feito a partir de uma metrificação do ensino. Através dos resultados obtidos (sejam elas nacionais como o ENEM, ou simplesmente uma metrificação de uma prova específica) não será disponibilizado sugestões de como melhorar, e consequentemente obter resultados excepcionais para assim conseguir uma melhoria efetiva. Nosso principal problema é o abismo que gera entre a falta de comunicação entre os dados e as escolas, justamente por não existir um diagnóstico que relacione de forma direta e objetiva o que poderia ser feito.

## **1.3. Objetivos**

### **1.3.1. Objetivos gerais**

- Consultar diagnóstico online com questões objetivas para a consultoria de possíveis melhorias de oportunidades para as escolas (públicas e privadas);
- Ter a possibilidade de comparação com os dados escolares, sejam municipais, estaduais, federais e de redes privadas;
- Ter a possibilidade de editar o questionário;
- Gerar escalabilidade, visando atingir mais clientes para a Falconi;

### **1.3.2. Objetivos específicos**

- Oferecer uma interface online, automática e intuitiva para a realização dos diagnósticos de agenda educacional e agenda de gestão da Falconi;

- Oferecer uma interface online de consulta de índices educacionais brasileiros filtrável por região, estado, município, rede e escola;
- Providenciar relatórios detalhados, na própria plataforma e por e-mail, com gráfico de radar e proposta de melhoria como resposta aos diagnósticos preenchidos por gestores escolares;
- Centralizar, organizar e comparar informações intra- e interescolares em uma mesma rede, salvando o histórico de resultados e de índices para consultas futuras;
- Propiciar a criação de novos leads para a Falconi através de diversos call to actions.

## 1.4. Descritivo da Solução

Aplicação web com diagnósticos gratuitos e instantâneos de gestão de redes de ensino, com relatório de pontos de melhoria e consulta imediata a índices educacionais de instituições semelhantes em todo o Brasil.

Principais funcionalidades:

- Questionário diagnóstico de agenda educacional que produz relatórios com análise da situação atual, gráficos e propostas de melhoria.
- Questionário diagnóstico de agenda de gestão que produz relatórios com análise da situação atual, gráficos e propostas de melhoria.
- Sistema de login que salva resultados anteriores para comparação futura.
- Resultados acessíveis tanto pela plataforma quanto por e-mail.
- Capacidade de o administrador adicionar, editar, deletar e duplicar questões e questionários.
- Consulta dos principais índices educacionais com filtragem por estado, município, rede e escola.
- Compara o diagnóstico com outros feitos no passado, podendo ter noção do progresso da instituição.
- Redirecionamento para contratação da consultoria após os questionários.

Como usar:

Para usuários gestores (de unidade ou de rede), SchoolAhead oferece duas funcionalidades principais: diagnósticos educacionais e de gestão com base nas respostas a certas perguntas e a consulta imediata de dados educacionais em todo o Brasil.

Para fazer isso, o usuário deve se dirigir à tela principal da plataforma e clicar em "Avalie sua escola" caso queira fazer o diagnóstico. Nesse ponto, será necessário fazer login ou criar uma

conta se ela ainda não existir. Essa conta exigirá o nome do gestor, e-mail, senha, escola e informações sobre a instituição, como número de aluno e séries contempladas.

Feita a conta, o gestor poderá escolher seu questionário, ler a respeito de sua metodologia e clicar em “Iniciar” para começar a respondê-lo. Cada questão conterá cinco alternativas, de modo que o usuário terá de ser o mais honesto possível para obter resultados confiáveis. Tais questões estão posicionadas uma acima da outra e ordenadas em várias páginas diferentes, uma para cada categoria ou eixo (grupo de questões de tema similar). A cada seção, será possível voltar para a anterior, porém não se poderá pular para a próxima seção antes de se terminar a atual.

Ao final do questionário e clicado no botão “Finalizar”, o usuário será automaticamente redirecionado para a tela de resultados. Nela, as conclusões tiradas a partir das respostas através da metodologia Falconi estarão visíveis e compartilháveis. Para sair desta página, basta clicar no menu do campo superior direito e depois em “Login”. Os cookies no computador reconhecerão a conexão e renderizar a página do gestor, onde ele poderá ver suas notas antigas (tanto da rede como de todas as unidades que a compõem, no caso de gestores de rede), refazer algum questionário, contatar a Falconi diretamente, alterar seus dados cadastrais e finalizar a conexão, saindo de sua conta. No caso de desconexão acidental, seu progresso no questionário será salvo.

No mesmo menu, clicando em “Consultar índices educacionais”, é possível pesquisar dados do MEC e INEP filtrados por região, estado, município e escola. A informação será disponibilizada em texto e em diagrama para melhor compreensão. Ademais, será possível comparar certa unidade ou rede com outras em uma única tela.

## **1.5. Partes Interessadas**

Dentre os principais stakeholders, tem-se:

1. Falconi: parceiro de negócio e usuário secundário do sistema a ser desenvolvido. Responsável por comunicar o problema e seus requisitos, assim como clarificar dúvidas e validar iterações nas reuniões quinzenais.
2. Inteli: representante legal da equipe RIGHTML. Fornece a infraestrutura e o apoio técnico para o desenvolvimento. Representado pelo orientador de projetos da turma.
3. CEO do Inteli: afetado pelo efeito que seus resultados podem causar na reputação do Inteli.
4. Escritório de projetos: Responsável por captar, negociar e viabilizar a parceria para este projeto.
5. Gestores de escola: usuários diretos da plataforma. Responsáveis por preencher os questionários diagnósticos e iniciar a comunicação com a Falconi, a fim de receber consultoria para otimizar sua gestão. Nesse sentido, seu interesse parte da instantaneidade e acesso ilimitado aos questionários e resultados proporcionados por uma ferramenta online.
6. Gestores de rede escolar: usuários secundários da plataforma. Responsáveis por preencher os diagnósticos para sua rede e analisar os resultados individuais das unidades sob sua gestão.





## **2. Análise do Problema**

A análise do problema tem um alto grau de complexidade, pois gira ao redor de uma característica que boa parte das empresas que dominam o mercado possuem, que é a escala. Dessa forma, a Falconi problematizou as consultorias de gestores educacionais, que demandam muito tempo e esforço por parte da empresa, visto que, os consultores devem entrevistar individualmente os gestores das instituições, com questionários de quase uma centena de perguntas, registrar e armazenar os dados em planilhas, discutir para chegar a um consenso e enfim, avaliar a instituição e projetar os pontos de melhoria. Assim, a empresa viu dificuldade na escalabilidade desse serviço e propôs que, através de um site, fosse possível a automatização dessa avaliação de gestores.

### **2.1. Análise da Indústria**

A Falconi Consultores de Resultados surgiu na década de 1980 como empresa de consultoria privada. Seu objetivo é ajudar organizações a construir resultados excepcionais pelo aperfeiçoamento de seu sistema de gestão. Nesse sentido, tradicionalmente trabalha com empresas privadas, apresentando planos de reestruturação e gestão. Hoje, a Falconi conta com mais de 700 consultores, sendo assim a maior desse mercado no cenário nacional.

O setor de consultoria, no mundo inteiro, encontra-se em grande expansão, havendo uma quantidade de concorrentes muito grande. Com isso, esse mercado está em constante mudança, devido às novas necessidades do mercado. Primeiramente, um número crescente de profissionais têm atuado de forma independente; de fato, 1 em cada 2 profissionais atuará por conta própria, segundo pesquisas da Robert Half, Fórum Econômico Mundial, Comissão Europeia. Desse modo, a Associação Brasileira de Consultores estima que o mercado de consultoria irá crescer mais de 50% somente no Brasil. Nesse ínterim, o acirramento da competitividade impulsionou 85% dos consultores a buscar treinamentos para aprimoramento de suas competências (ABCO, 2019). Sabendo disso, diferenciais como velocidade, que a automação de algumas atividades traz, e assertividade dos resultados, proporcionada por uma equipe bem treinada, são de fundamental importância.

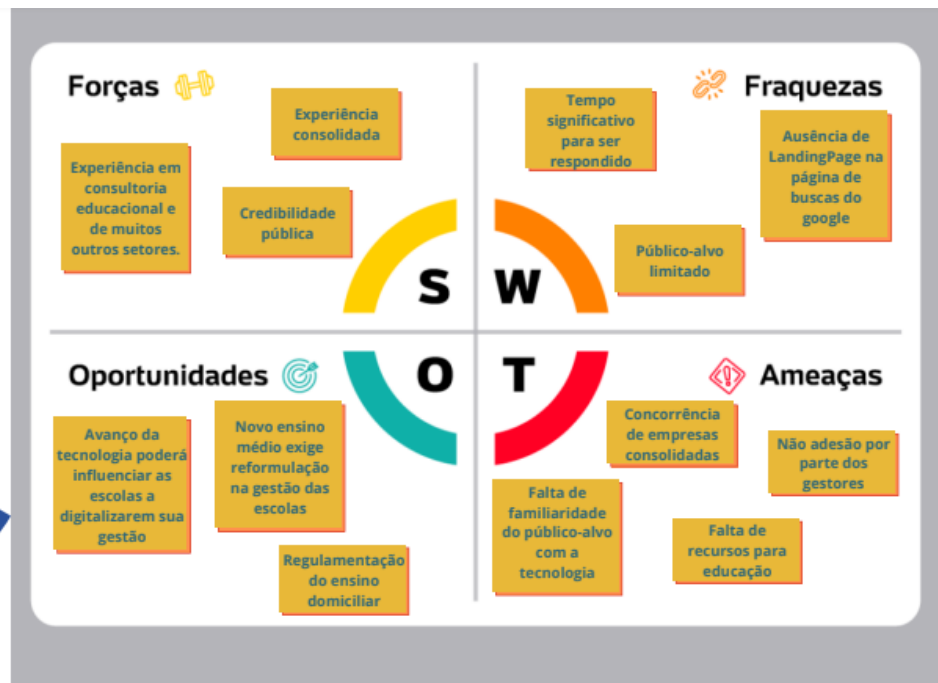
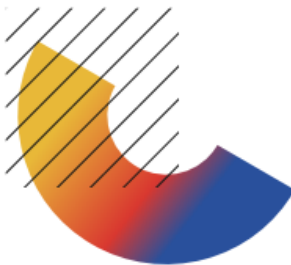
No contexto do poder de barganha, o cliente possui certo poder, por haver uma grande quantidade de concorrentes no mercado, gerando uma maior competitividade. Por outro lado, a Falconi possui uma base de clientes qualificados muito boa, além de um nome consolidado no mercado de consultoria, podendo assim ter um diferencial.

A despeito do mercado de consultoria, podemos afirmar que a barreira de entrada pode ser considerável, haja vista a necessidade de um conhecimento e uma equipe qualificada, porém, mesmo assim, o crescimento dessa indústria é inegável. A importância de haver uma empresa consolidada, como a Falconi, na consultoria é a experiência que ela traz

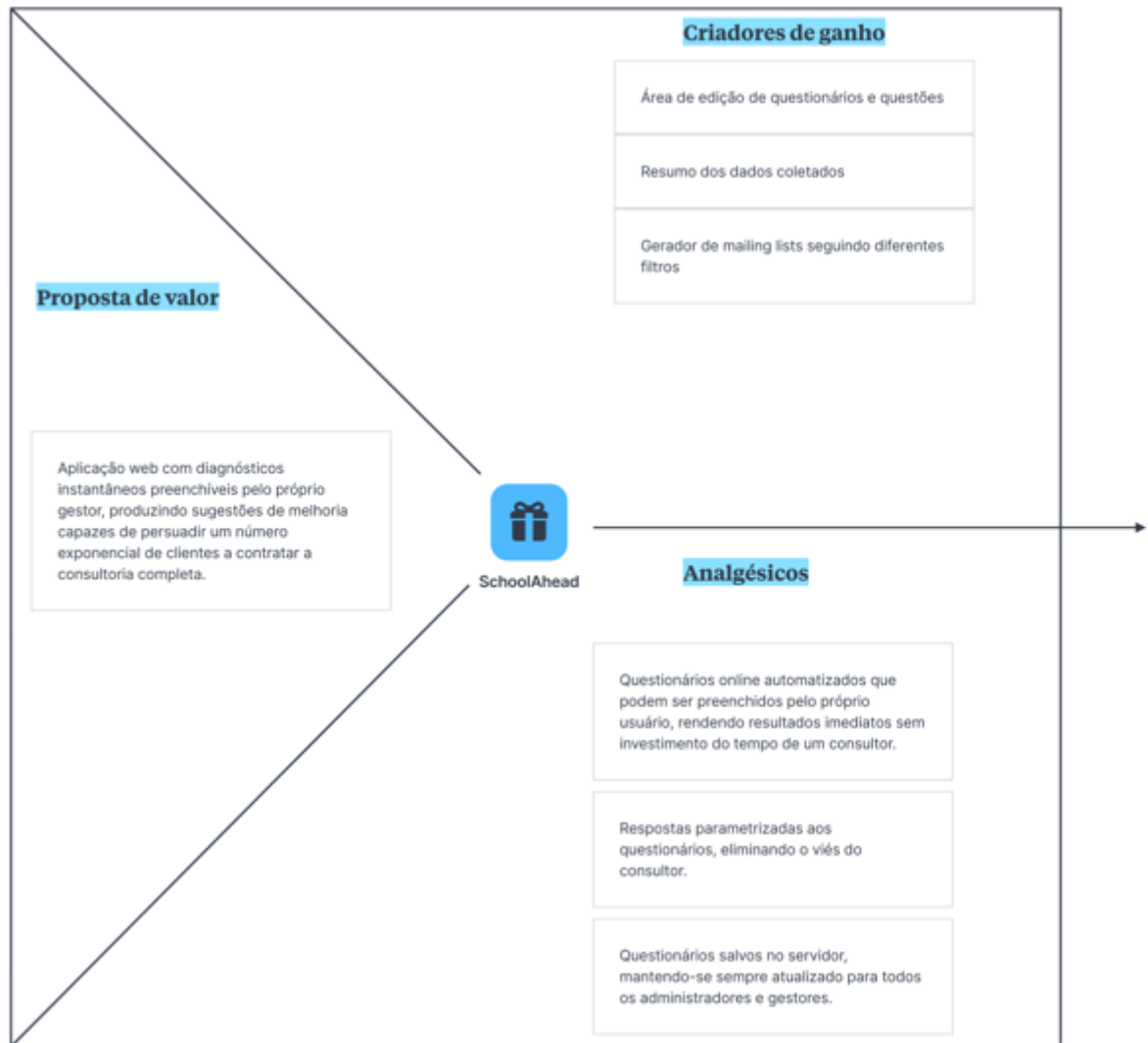
consigo. Para uma empresa consolidada se manter em sua posição privilegiada é treinar os funcionários e buscar por novas soluções.

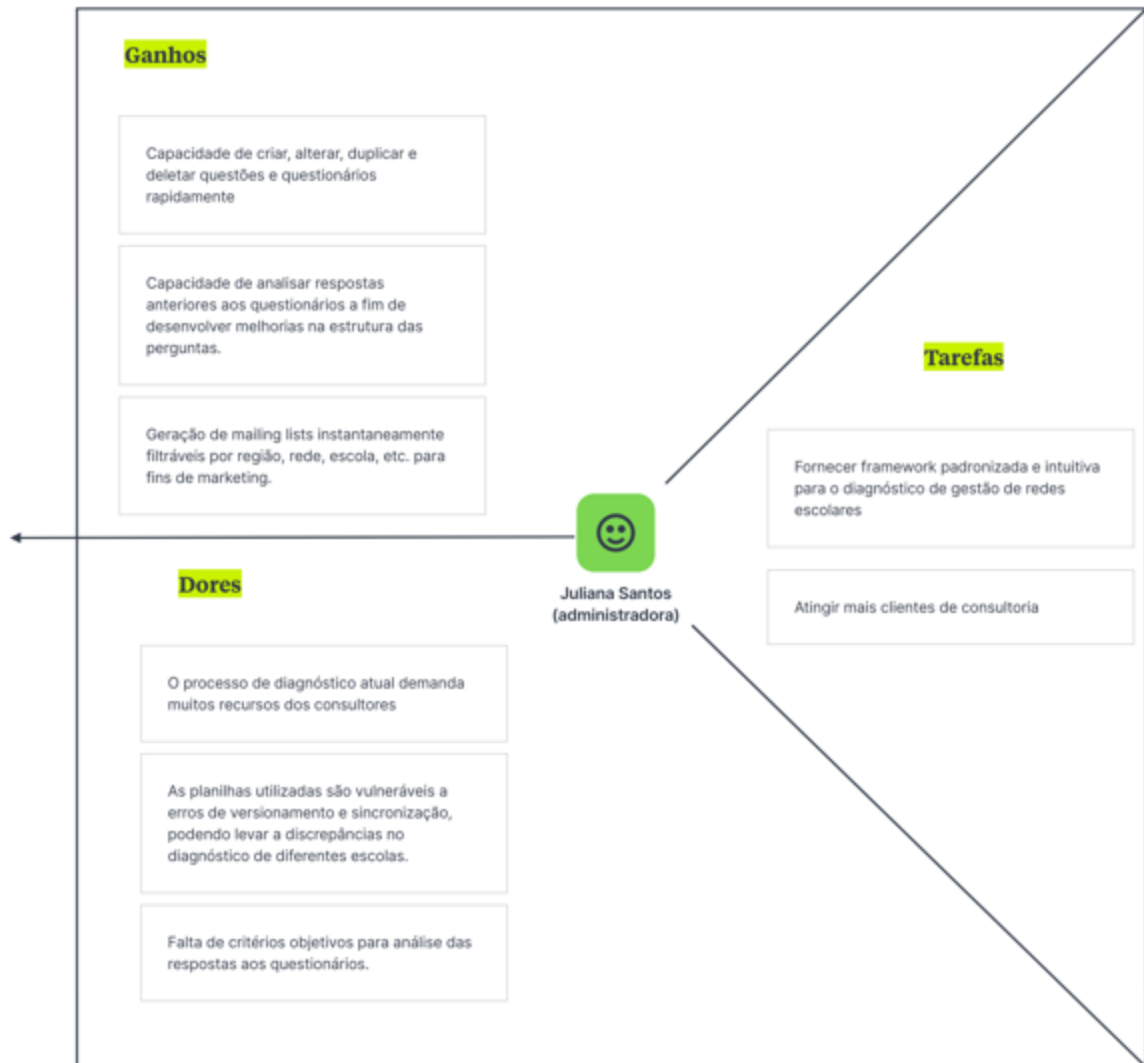
## 2.2. Análise do cenário: Matriz SWOT

### Matriz SWOT



## 2.3. Proposta de Valor: Value Proposition Canvas





## 2.4. Matriz de Risco

Matriz de Risco											
Probabilidade		Riscos					Oportunidades				
Muito Alta	5							Aprimoramento de habilidades técnicas de desenvolvimento web			
Alta	4							Fortalecer o networking			
Médio	3		Falta de energia ou problemas de conexão podem prejudicar reuniões com o parceiro	Dados do MEC e INEP podem sair do ar	Adoecimento de um dos membros do time		Falconi pode dar continuidade ao produto				
Baixa	2		Erros de commit/merge no Github	Documentação incompleta, antiga ou superficial pode prejudicar o desenvolvimento futuro							
Muito Baixa	1				Webhost pode sair do ar						
		1	2	3	4	5	5	4	3	2	1
		Muito Baixo	Baixo	Médio	Alta	Muito Alta	Muito Alta	Alta	Médio	Baixo	Muito Baixo
		Impacto									

## 3. Requisitos do Sistema

### 3.1. Persona

Juliana Santos

27 anos

Usuário administrador da Falconi

#### Mini biografia

- Movida por desafios
- Preza pela excelência
- Formada em administração
- Muito confortável com tecnologia
- Constantemente fazendo cursos de extensão

#### Necessidades/tarefas

- Fornecer framework padronizada e intuitiva para o diagnóstico de gestão de redes escolares
- Atingir mais clientes
- Melhorar o processo diagnóstico da consultoria com base nos dados dos gestores que já o utilizaram

#### Dores

- O processo de diagnóstico atual demanda muitos recursos dos consultores
- As planilhas utilizadas são vulneráveis a erros de versionamento e sincronização, podendo levar a discrepâncias no diagnóstico de diferentes escolas.
- Falta de critérios objetivos para análise das respostas aos questionários.

Beth Silva

38 anos

Diretora de escola

#### Mini biografia

- Apaixonada por educação
- Quase 30 anos de experiência com docência
- Formada em história
- Nível básico de entendimento da tecnologia

#### Necessidades/tarefas

- Atingir as metas dos índices de educação propostas pelo estado
- Entender como pode melhorar a escola

#### **Dores**

- Dificuldade em implementar as mudanças necessárias
- Falta de tempo e conhecimento para fazer uma análise profunda da gestão da sua instituição
- Falta de acesso a uma base de dados com várias informações sobre outras escolas para poder realizar uma comparação

### **Paulo Freitas**

45 anos

Secretário da educação em Goiânia – GO

#### **Mini biografia**

- Longa carreira como gestor
- Familiaridade com o meio educacional
- Possui certa experiência com tecnologia, por conta de suas tarefas diárias

#### **Necessidades/tarefas**

- Lida todo dia com tarefas básicas em seu computador
- Saber ,onde há maior carência de verba
- Saber aonde implementar os recursos estatais

#### **Dores**

- Possui grande dificuldade em saber aonde implementar os recursos estatais para melhoria
- Não há dados para saber aonde implementar os recursos estatais
- Índices da educação se encontram abaixo da média

## **3.2 Histórias dos usuários (user stories)**

#### **Juliana Santos (usuário administrador da Falconi)**

- Como Juliana, quero editar questionários para aprimorar nossa metodologia de diagnóstico conforme necessário.
- Como Juliana, quero consultar todos os cadastrados na plataforma para melhor entender nosso público-alvo.
- Como Juliana, quero criar mailing lists com base nos cadastrados para segmentar nossa publicidade da melhor forma possível.

- Como Juliana, quero cadastrar, deletar e alterar informações de usuários cadastrados para facilitar a troca de e-mails, senhas, escola e rede de cada conta, caso necessário.
- Como Juliana, quero acessar os resultados das escolas que contratem a consultoria para familiarizar a equipe com elas antes de iniciar os trabalhos.

#### **Beth Silva (gestora escolar)**

- Como Beth, quero responder aos questionários diagnósticos para descobrir a situação atual da minha escola.
- Como Beth, quero receber um relatório com design bem estruturado, contendo gráficos e hierarquia de informações, para entender melhor os dados apresentados.
- Como Beth, quero receber o relatório em meu e-mail para poder acessá-lo sem me conectar à plataforma.
- Como Beth, quero baixar o relatório em PDF para poder acessá-lo offline.
- Como Beth, quero compartilhar o relatório, em link read-only ou PDF, para que meus colegas de trabalho e superiores possam vê-lo também.
- Como Beth, quero consultar meus resultados anteriores para acompanhar o progresso de minha escola.
- Como Beth, quero facilmente contatar a Falconi para saber mais sobre a consultoria o mais rápido possível.
- Como Beth, quero consultar os resultados de outras escolas da minha rede para entender como minha escola se compara a elas e no que preciso investir mais recursos.
- Como Beth, quero consultar índices educacionais de todo o Brasil para entender como minha escola se compara a outras e no que preciso investir mais recursos

#### **Paulo Freitas (secretário de educação)**

- Como Paulo, quero saber os índices de educação no meu estado e país.
- Como Paulo, quero poder contactar a Falconi para saber como resolver meus problemas
- Como Paulo, quero saber o índice de todas as escolas que eu gerencio, seus problemas e como foram resolvidos para saber como distribuir recursos.

## **4. Arquitetura do Sistema**

### **4.1. Módulos do Sistema e Visão Geral (Big Picture)**

Diagrama representando hardware e software.



Mapa ou organograma com os módulos que existem no sistema.

Por exemplo, um portal principal, em seguida as áreas de acordo com perfil de acesso. Um painel administrativo para controle e gestão, por exemplo.

E tudo no servidor em nuvem, no nosso caso, Heroku.

Pode usar uma ferramenta do tipo x-mind, draw.io, etc.

## **4.2. Descrição dos Subsistemas**

Aqui detalhar cada subsistema com suas funcionalidades, ou seja, o que tem em cada módulo.

Pode usar uma ferramenta do tipo x-mind, draw.io, etc.

### **4.2.1. Users Stories dos subsistemas**

Dentre as users stories da seção 3.2 quais correspondem a cada subsistema.

### **4.2.2. Requisitos de software**

Dentre o total de tecnologias utilizadas na aplicação, da seção 4.4, quais correspondem a cada subsistema.

### **4.2.3. Documentação via Postman**

#### **Networks (redes escolares)**

Para acessar as redes de escola registradas, faça uma requisição GET através da URL “/networks”. Os resultados serão ordenados alfabeticamente pelo atributo “name”.

Para inserir redes, faça uma requisição POST através da URL “/networkinsert” com os parâmetros “name” (text) e “type” (text “public” ou “private”), respectivamente nome da rede e tipo público ou privado. O “id” é autoincrementado.

Para deletar redes, faça uma requisição POST através da URL “/networkdelete” com o parâmetro “id” (integer).

Para atualizar redes, faça uma requisição POST através da URL “/networkupdate” com as modificações desejadas nos parâmetros “name” (text) e “type” (text “public” ou “private”) e o id do registro desejado.

#### **Network managers (gestores de rede)**

Para acessar os gestores de rede resgistrados, faça uma requisição GET através da URL “/networkmanagers”. Os resultados serão ordenados alfabeticamente pelo atributo “name”.

Para inserir gestores, faça uma requisição POST através da URL `/networkmanagerinsert` com os parâmetros `“cpf”` (numeric), `“email”` (text), `“name”` (text) e `“network_id”` (integer), respectivamente o CPF do gestor, seu e-mail, seu nome completo e o id da rede que ele gerencia.

Para deletar gestores, faça uma requisição POST através da URL `/networkmanagerdelete` com o parâmetro `“cpf”` (integer).

Para atualizar gestores, faça uma requisição POST através da URL `/networkmanagerupdate` com as modificações desejadas nos parâmetros `“network_id”` (numeric), `“email”` (text), `“name”` (text) e utilize o `“cpf”` para localização do registro. Note que não é possível modificar o cpf de um gestor já cadastrado no sistema. Assim, se esse feito for necessário, dever-se-á excluir o cadastro existente e registrar o gestor novamente com as informações corretas.

### **School (Escolas)**

Para acessar as escolas registradas, faça uma requisição GET através da URL `/schools`. Os resultados serão ordenados alfabeticamente pelo atributo `“name”`.

Para inserir escolas, faça uma requisição POST através da URL `/schoolinsert` com os parâmetros `“name”` (text), `“number_of_students”` (integer), `“number_of_employees”` (integer), `“school_census_id”` (integer), `“type_of_institution”` (text entre `“public”` e `“private”`), `“network_id”` (integer) e `“cnpj”` (numeric), referentes ao nome da escola, número de alunos, número de funcionários, ID no Censo Escolar, tipo de instituição, id da rede escolar e CNPJ.

Para deletar escolas, faça uma requisição POST através da URL `/schooldelete` com o parâmetro `“cnpj”` (numeric).

Para atualizar escolas, faça uma requisição POST através da URL `/schoolupdate` com os parâmetros com as modificações desejadas nos parâmetros `“name”` (text), `“number_of_students”` (integer), `“number_of_employees”` (integer), `“school_census_id”` (integer), `“type_of_institution”` (text entre `“public”` e `“private”`) e `“network_id”` (integer). Utilize também o atributo `“cnpj”` (numeric) para localização do registro. Não é possível mudar o atributo `“cnpj”`, sendo necessário excluir e recadastrar o registro com as informações corretas nesse caso.

### **School Managers (gestores de escola)**

Para acessar as os gestores de escola registrados, faça uma requisição GET através da URL `/schoolmanagers`. Os resultados serão ordenados alfabeticamente pelo atributo `“name”`.

Para inserir gestores de escola, faça uma requisição POST através da URL `/schoolmanagerinsert` com os parâmetros `“name”` (text), `“email”` (text), `“school_cnpj”` (numeric) e `“cpf”` (numeric), referentes ao nome, e-mail, CPNJ da escola e CPF do gestor escolar.

Para deletar gestores de escola, faça uma requisição POST através da URL `/schoolmanagerdelete` com o parâmetro `“cpf”` (numeric).

Para atualizar escolas, faça uma requisição POST através da URL `/schoolupdate` com as modificações desejadas nos parâmetros `“name”` (text), `“email”` (integer) e `“school_cnpj”` (integer). Utilize também o atributo `“cpf”` para localizar o registro. Caso seja necessário alterar o `“cpf”`, exclua o registro e recadastre o gestor com as informações corretas.

### **Diagnosis (diagnósticos)**

Para acessar os diagnósticos (agendas e/ou questionários) registrados, faça uma requisição GET através da URL `"/diagnoses"`. Os resultados serão ordenados alfabeticamente pelo atributo `"name"`.

Para inserir diagnósticos, faça uma requisição POST através da URL `"/diagnosisinsert"` com os parâmetros `"name"` (text), `"description"` (text) e `"answer_time"` (integer), referentes ao nome, descrição e tempo (em minutos) necessário para completar o questionário. O id é autoincrementado.

Para deletar diagnósticos, faça uma requisição POST através da URL `"/diagnosisdelete"` com o parâmetro `"id"` (integer).

Para atualizar diagnósticos, faça uma requisição POST através da URL `"/diagnosisupdate"` com as modificações desejadas nos parâmetros `"name"` (text), `"description"` (text) e `"answer_time"` (integer). Utilize também o `"id"` para localizar o registro.

### **Axis (eixos)**

Para acessar os eixos registrados, faça uma requisição GET através da URL `"/axes"`. Os resultados serão ordenados alfabeticamente pelo atributo `"name"`.

Para inserir eixos, faça uma requisição POST através da URL `"/axisinsert"` com os parâmetros `"name"` (text), `"subdivision_name"` (text), `"position"` (text) e `"diagnosis_id"` (integer), referentes ao nome do eixo, nome de suas subdivisões (até o momento `"fator crítico"` ou `"domínio"`), posição no questionário e id do diagnóstico ao qual pertence. O id próprio é autoincrementado.

Para deletar eixos, faça uma requisição POST através da URL `"/axisdelete"` com o parâmetro `"id"` (integer).

Para atualizar eixos, faça uma requisição POST através da URL `"/axisupdate"` com as modificações desejadas nos parâmetros `"name"` (text), `"subdivision_name"` (text) e `"position"` (integer). Utilize também o `"id"` (integer) para localizar o registro.

### **Axis Subdivisions (Subdivisão de Eixos)**

Para acessar os subeixos registrados, faça uma requisição GET através da URL `"/axissubdivisions"`. Os resultados serão ordenados alfabeticamente pelo atributo `"name"`.

Para inserir eixos, faça uma requisição POST através da URL `"/axissubdivisioninsert"` com os parâmetros `"name"` (text), `"axis_id"` (integer) e `"diagnosis_id"` (integer), referentes ao nome e ao diagnóstico e eixo que contém o subeixo. O id é autoincrementado.

Para deletar subeixos, faça uma requisição POST através da URL `"/axissubdivisiondelete"` com o parâmetro `"id"` (integer).

Para atualizar eixos, faça uma requisição POST através da URL `"/axissubdivisionupdate"` com as modificações desejadas nos parâmetros `"name"` (text), `"axis_id"` (integer) e `"diagnosis_id"` (integer). Utilize também o `"id"` (integer) para localizar o registro.

### **Questions (Questões)**

Para acessar as questões registradas, faça uma requisição GET através da URL `"/questions"`. Os resultados serão ordenados alfabeticamente pelo atributo `"position"`.

Para inserir questões, faça uma requisição POST através da URL “/questioninsert” com os parâmetros “text” (text), “position” (integer), “weight” (integer), “axis\_subdivision\_id” (integer), “axis\_id” (integer) e “diagnosis\_id” (integer), referentes ao enunciado, posição no subeixo, peso, subeixo, eixo e diagnóstico aos quais a questão pertence. O id é autoincrementado.

Para deletar questões, faça uma requisição POST através da URL “/questiondelete” com o parâmetro “id” (integer).

Para atualizar questões, faça uma requisição POST através da URL “/questionupdate” com as modificações desejadas nos parâmetros “text” (text), “position” (integer), “weight” (integer), “axis\_subdivision\_id” (integer), “axis\_id” (integer) e “diagnosis\_id” (integer). Utilize também o “id” (integer) para localizar o registro.

### **Options (Opções/Alternativas)**

Para acessar as alternativas registradas, faça uma requisição GET através da URL “/options”. Os resultados serão ordenados alfabeticamente pelo atributo “id”.

Para inserir alternativas, faça uma requisição POST através da URL “/optioninsert” com os parâmetros “text” (text), “weight” (integer), “position” (integer), “question\_id” (integer), “axis\_subdivision\_id” (integer), “axis\_id” (integer), “diagnosis\_id” (integer), referentes ao texto da alternativa, seu peso, sua posição na questão, à questão que a abriga, seu subeixo, eixo e diagnóstico. O id é autoincrementado.

Para deletar alternativas, faça uma requisição POST através da URL “/optiondelete” com o parâmetro “id” (integer).

Para atualizar questões, faça uma requisição POST através da URL “/optionupdate” com as modificações desejadas nos parâmetros “text” (text), “weight” (integer), “position” (integer), “question\_id” (integer), “axis\_subdivision\_id” (integer), “axis\_id” (integer), “diagnosis\_id” (integer). Utilize também o “id” (integer) para localizar o registro.

### **Answers (Respostas)**

Para acessar as respostas registradas, faça uma requisição GET através da URL “/answers”. Os resultados serão ordenados alfabeticamente pelo atributo “id”.

Para inserir respostas, faça uma requisição POST através da URL “/answerinsert” com os parâmetros “extra\_info” (text), “option\_id” (integer), “question\_id” (integer), “axis\_subdivision\_id” (integer), “axis\_id” (integer), “diagnosis\_id” (integer), “school\_id” (integer) e “network\_id” (integer), referentes ao campo de mais anotações, número de alunos na escola associada (a ser utilizado em momento futuro), alternativa, questão, subdivisão do eixo, eixo, diagnóstico, escola e rede aos quais a questão pertence. O id é autoincrementado.

Para deletar respostas, faça uma requisição POST através da URL “/answerdelete” com o parâmetro “id” (integer).

Para atualizar respostas, faça uma requisição POST através da URL “/answerupdate” com as modificações desejadas nos parâmetros “extra\_info” (text), “option\_id” (integer), “question\_id” (integer), “axis\_subdivision\_id”, “axis\_id”, “diagnosis\_id”, “school\_id” (integer) e “network\_id” (integer). Utilize também o “id” (integer) para localizar o registro.

### **Employees (funcionários da Falconi)**

Para acessar os funcionários da Falconi registradas, faça uma requisição GET através da URL “/employees”. Os resultados serão ordenados alfabeticamente pelo atributo “name”.

Para inserir funcionários, faça uma requisição POST através da URL “/employeeinsert” com os parâmetros “email” (string), “name” (string), “type” (string entre “admin” e “consultant”) e “cpf” (numeric).

Para deletar funcionários, faça uma requisição POST através da URL “/employeedelete” com o parâmetro “cpf” (numeric).

Para atualizar funcionários, faça uma requisição POST através da URL “/employeeupdate” com as modificações desejadas nos parâmetros “extra\_info” (text), “option\_id” (integer), “question\_id” (integer), “axis\_subdivision\_id”, “axis\_id”, “diagnosis\_id”, “school\_id” (integer) e “network\_id” (integer). Utilize também o “id” (integer) para localizar o registro.

### **Employee\_School (tabela auxiliar)**

Para acessar esta tabela auxiliar, faça uma requisição GET através da URL “/employeeschools”. Os resultados serão ordenados alfabeticamente pelo atributo “email\_employee”.

Para inserir nesta tabela, faça uma requisição POST através da URL “/employeeschools” com os parâmetros “email\_employee” (text) e “school\_cnpj” (numeric), referentes ao e-mail do funcionário e ao CNPJ da escola.

Para deletar registros nesta tabela, faça uma requisição POST através da URL “/employeeschooldelete” com os parâmetros “email\_employee” (text) e “school\_cnpj” (numeric).

Não é possível atualizar registros nesta tabela.

### **Addresses (endereços)**

Para acessar os endereços registrados, faça uma requisição GET através da URL “/addresses”. Os resultados serão ordenados alfabeticamente pelo atributo “cep”.

Para inserir endereços, faça uma requisição POST através da URL “/addressinsert” com os parâmetros “street” (text), “street\_number” (integer), “neighborhood” (text), “cep” (numeric), “city” (text), “state” (text) e “school\_cnpj” (numeric), referentes à rua, número, bairro, CEP, cidade, estado e CNPJ da escola localizada neste endereço.

Para deletar endereços, faça uma requisição POST através da URL “/addressdelete” com o parâmetro “school\_cnpj” (numeric).

Para atualizar endereços, faça uma requisição POST através da URL “/addressupdate” com as modificações desejadas nos parâmetros “street” (text), “street\_number” (integer), “neighborhood” (text), “cep” (numeric), “city” (text) e “state” (text). Utilize também o atributo “school\_cnpj” (numeric) para localizar o registro em questão. Caso seja necessário alterar o “cnpj”, exclua o registro e recadastre o gestor com as informações corretas.

### 4.3. Tecnologias Utilizadas

Colocar em uma tabela as tecnologias utilizadas na aplicação especificando o que é, em que é utilizada no projeto e qual a versão.

Server Side	Função
Node-JS (18.0.1)	Executa scripts em JS dentro do servidor e envia os resultados para a página web
Postman (9.19.0)	Utilizado para testar a API do site
SqLite (3.38.5)	É o banco de dados onde armazena todas as informações dos usuários, perguntas e respostas do questionário .
HTML5	Utilizado para organizar e apresentar toda a parte visual do site
CSS (4.15)	Utilizado para estilizar diferentes partes do site.
JavaScript (ECMAScript 2018)	A linguagem de programação utilizada para todas as funções do site.

Client Side	Função
Ajax	Mandar e receber informações do servidor de forma assíncrona sem interferir na página web.
Browser	O browser que o usuário utiliza para acessar o site podendo ser qualquer um.

## 5. UX e UI Design

Projeto das telas do sistema.

### 5.1. Wireframe + Storyboard

- Link Wireframe: <https://balsamiq.cloud/s68fi9n/p703svo/rFF99>
- Link Storyboard: <https://miro.com/app/board/uXjVO5cDU68=/>

### 5.2. Design de Interface - Guia de Estilos]

O guia de estilo determina as diretrizes de design para a identidade visual de certa empresa, organização e/ou marca. Deve ser seguido à risca para garantir uma imagem consistente na mente do consumidor.

#### Cores



##### **Cor primária: #232228**

Este tom foi escolhido como cor primária por sua neutralidade. Um pouco mais claro que o preto absoluto, combina com diversos outros tons sem oprimir o conjunto geral. É utilizado como cor de fundo no cabeçalho e rodapé.



##### **Cor secundária (de destaque): #F8C000**

Este tom oferece um contraste agradável com a cor primária. Essa cor foi escolhida, pois é a cor do nosso ícone principal, uma lâmpada amarela que representa lucidez e conhecimento que é exatamente o que queremos transmitir com nossa aplicação web.



##### **Cor terciária (call to action): #810DFD**

Essa cor foi escolhida porque chama a atenção, ela casa perfeitamente com o amarelo e com o preto que não é 100%, além de ser nossa cor de call to action (chamadas para ação, ou seja, chamadas para interatividade do usuário).

##### **Cor da fonte**

Quando o background é claro, a cor da fonte é preta (#000). Quando o background é escuro, a cor da fonte é branca (#FFF). Isso garante o contraste e facilidade de leitura.

### Cor de borda: #808080

Esta cor é utilizada nas bordas de certos elementos, a fim de delimitar diferentes seções de uma página. O tom é leve para não competir demais com outras partes do design que merecem mais atenção.

## Tipografia

A escolha de tipografia baseou-se, principalmente, na combinação já consolidada presente na identidade visual do Inteli. Coincidentemente, a fonte Manrope também aparece no guia de estilo da Falconi. Desse modo, a junção de Lucida Console com Manrope ilustra a parceria entre o Inteli e a Falconi.

### Lucida Console, monospace

Utilizado em títulos, no cabeçalho e no rodapé.

Lembra simultaneamente a tipografia de máquinas de escrever antigas e a tipografia utilizada em ambientes integrados de desenvolvimento. Desse modo, traz a ideia de continuidade entre o passado e a modernidade, acomodando clientes de todas as gerações. Essa característica é fundamental para um website que atenderá tanto gestores de mais idade quanto consultores e profissionais mais jovens.

### Manrope, sans serif

Utilizado no corpo do texto. Seus cantos arredondados passam uma impressão amigável ao mesmo tempo em que facilitam a leitura por ser sem serifa. O design simples também promove uma transição sem atrito do Lucida Console.

Deve ser sempre justificado e no tamanho *small* no CSS, a fim de garantir a responsividade.

## Espaçamento

Todo o corpo da página deve ficar dentro da tag `<main>` do HTML, a qual tem como margens laterais “2vh”.

## Bordas

As bordas de contêiner devem ser cinzas, conforme especificado na seção de cores, e conter raio de 2px.

## Ícones





Esse ícone foi escolhido por simbolizar consciência e lucidez, as quais nossa equipe busca com essa aplicação web. Ele está presente em três lugares principais: cabeçalho, rodapé e logo do site. Foi criado com licença de uso comercial no Canva.



Este ícone ilustra os botões de adicionar perguntas no modo de edição de questionário. É simples, e a estilização ocorre no CSS aplicado ao fundo e borda do botão. Disponível no catálogo de ícones do Bootstrap: <https://icons.getbootstrap.com/icons/plus/>.



Este ícone ilustra os botões de excluir perguntas no modo de edição de questionário. É um design simples, pois, como aparece em tamanho pequeno site, detalhes demais deixariam o reconhecimento de seu formato mais difícil. Disponível no catálogo de ícones do Bootstrap: <https://icons.getbootstrap.com/icons/trash3/>.



Ícone que guarda o link à página do Facebook da Falconi no rodapé do site. Como todo link, fica amarelo quando o mouse passa sobre ele. Disponível no catálogo de ícones do Bootstrap: <https://icons.getbootstrap.com/icons/facebook/>.



Ícone que guarda o link à página do Instagram da Falconi no rodapé do site. Como todo link, fica amarelo quando o mouse passa sobre ele. Disponível no catálogo de ícones do Bootstrap: <https://icons.getbootstrap.com/icons/instagram/>.



Ícone que guarda o link à página do Instagram da Falconi no rodapé do site. Como todo link, fica amarelo quando o mouse passa sobre ele. Disponível no catálogo de ícones do Bootstrap: <https://icons.getbootstrap.com/icons/linkedin/>.

## **6. Projeto de Banco de Dados**

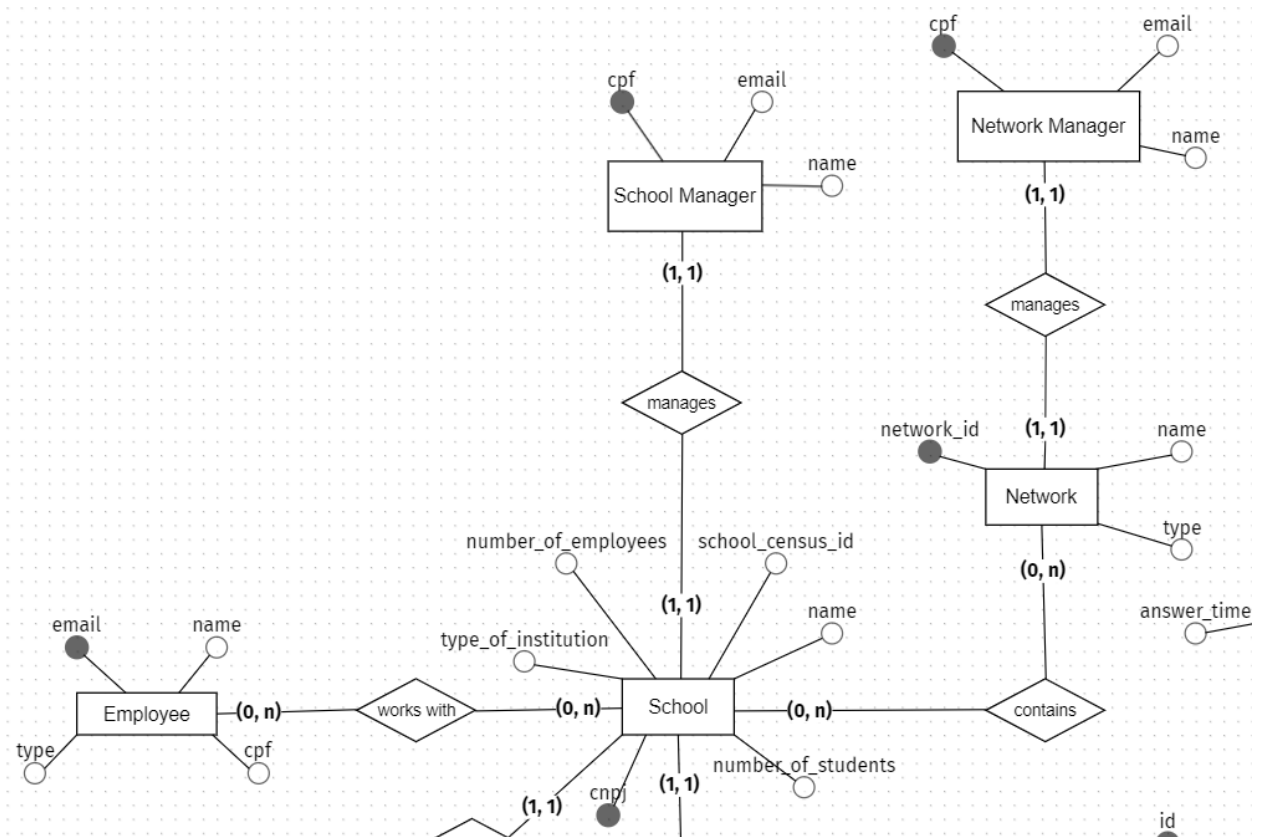
### **6.1. Modelo Conceitual**

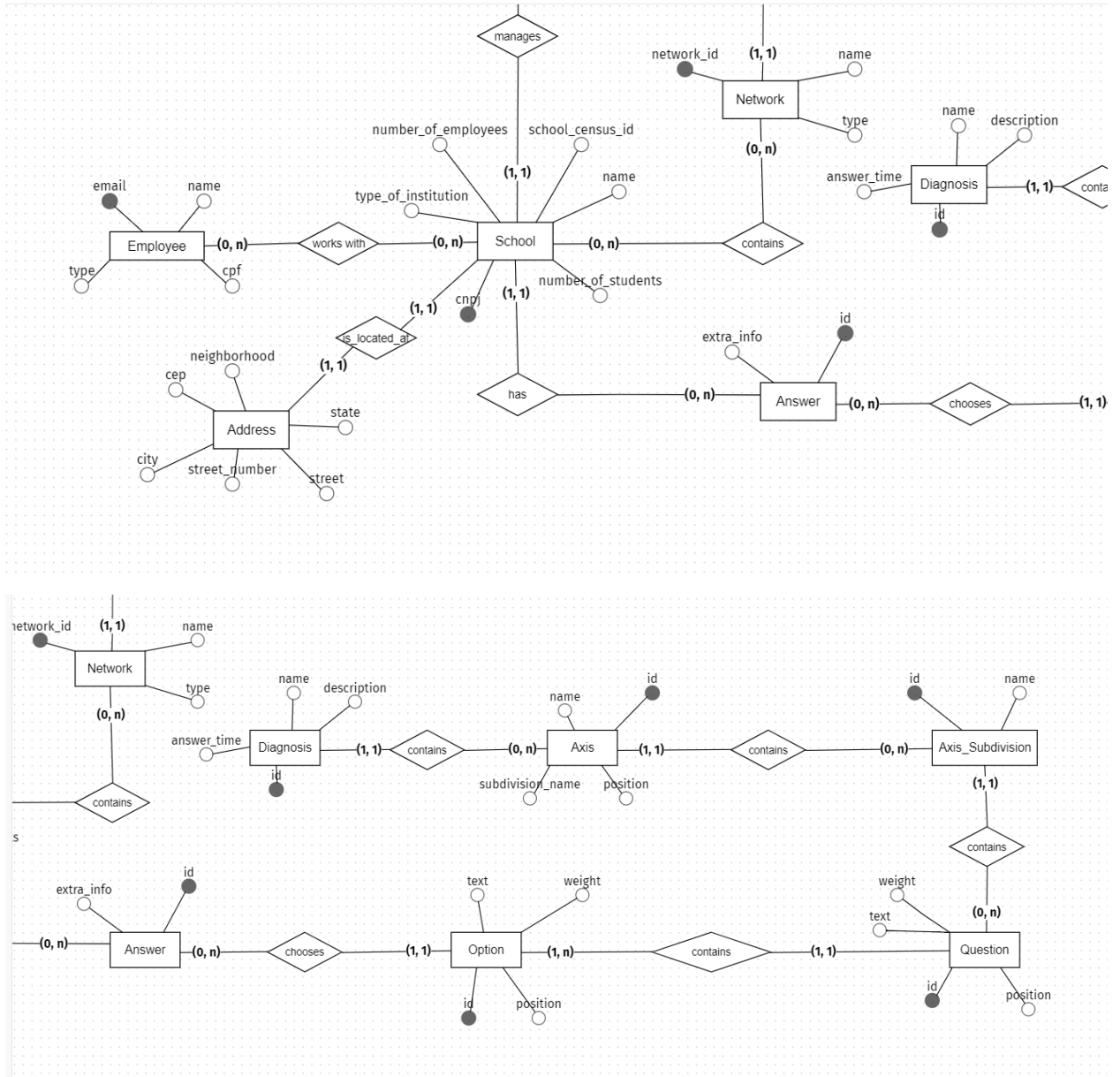
O modelo conceitual deve garantir uma conexão com a realidade. Os 4 tipos de conexões com a realidade são:

- conceitos
- atributos
- identificações
- associações

O Modelo Entidade-Relacionamento - MER

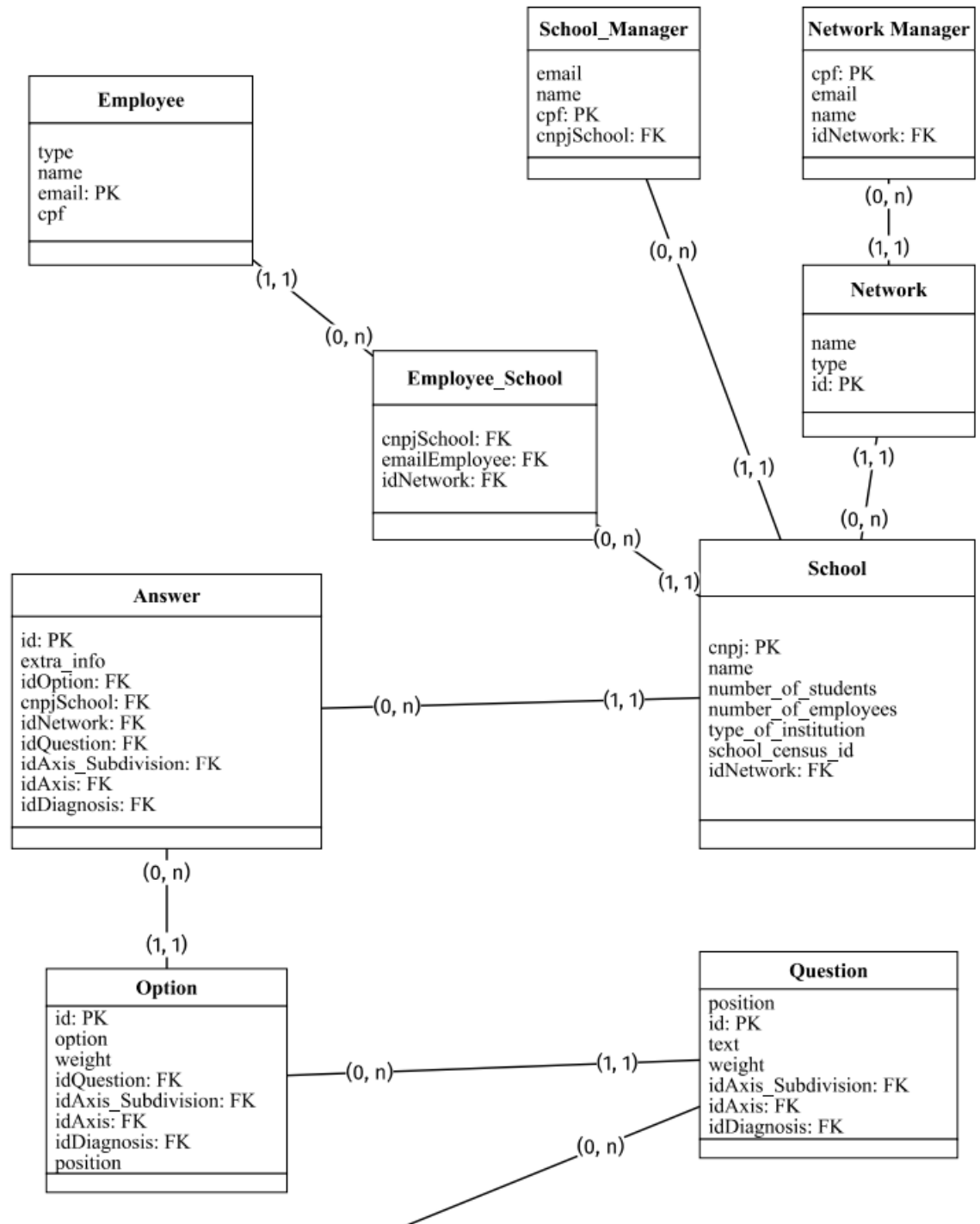
- entidades e tipos de entidades
- atributos e tipos de atributos
- relacionamentos e tipos de relacionamentos

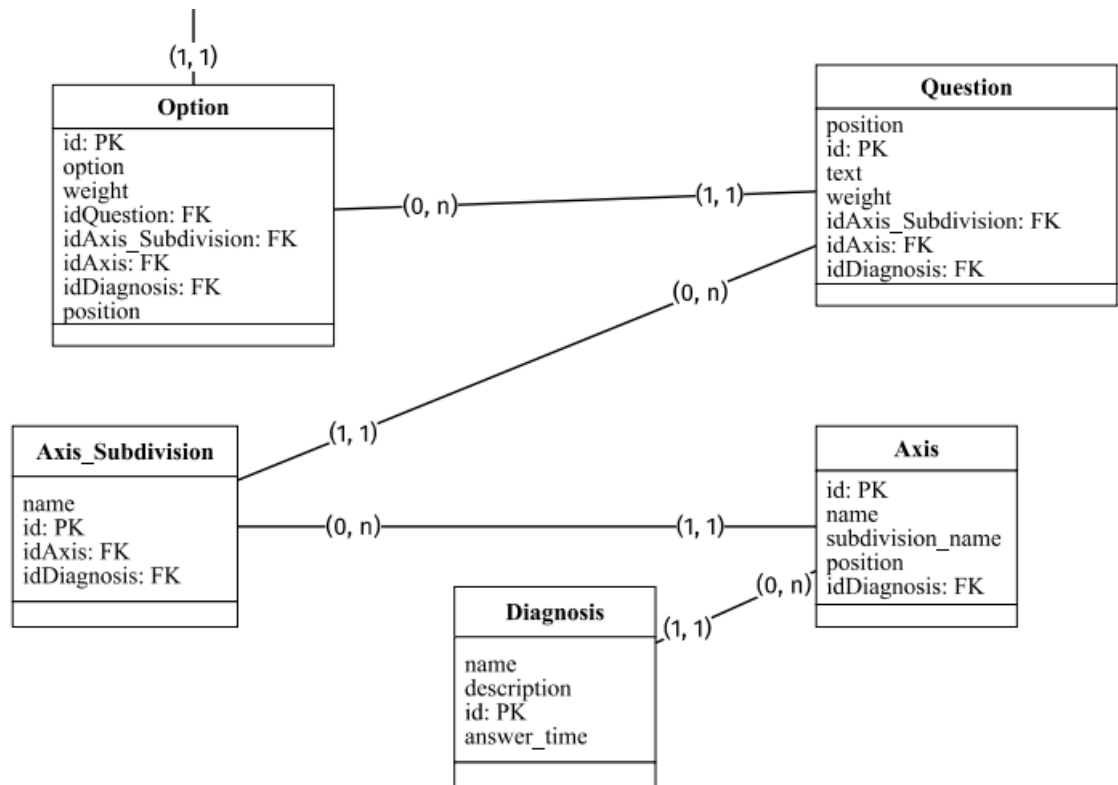




## 6.2. Modelo Lógico

Modelo lógico do banco de dados.





## 7. Teste de Software

### 7.1. Teste Unitário

Evidências dos testes realizados usando o Jest

### 7.2. Teste de Usabilidade

Tabela com dados organizados dos testes realizados

## 8. **Análise de Dados**

Inserir os dashboard ou gráficos das análises dos dados de negócio, usando estatística descritiva com medidas de posição e dispersão.

## **9. Manuais**

### **9.1. Manual do Usuário**

Aqui identificar todos os usuários do sistema

Semana 10 - Artefato: documento contendo instruções da aplicação para o usuário final

### **9.2. Manual do Administrador**



## Referências

Toda referência citada no texto deverá constar nessa seção, utilizando o padrão de normalização da ABNT). As citações devem ser confiáveis e relevantes para o trabalho. São imprescindíveis as citações dos sites de download das ferramentas utilizadas, bem como a citação de algum objeto, música, textura ou outros que não tenham sido produzidos pelo grupo, mas utilizados (mesmo no caso de licenças gratuitas, royalty free ou similares)

## **Apêndice**

Os apêndices representam informações adicionais que não caberiam no documento exposto acima, mas que são importantes por alguma razão específica do projeto.