**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**NÚCLEO DE EDUCAÇÃO A DISTÂNCIA

Pós-graduação *Lato Sensu* em Desenvolvimento Web Full Stack

**Renan Souza de Sá**

**CONTROLE DE PROJETOS E APONTAMENTOS SEM CONSIDERAR CUSTO**

Belo Horizonte

2022

**SUMÁRIO**

[**1. Apresentação 3**](#_heading=h.gjdgxs)

[**1.1. Contexto 3**](#_heading=h.30j0zll)

[1.2. Público alvo 6](#_heading=h.7txmz3uoq44b)

[1.3. Requisitos 7](#_heading=h.3znysh7)

[**2. Modelagem 10**](#_heading=h.a0flm7pn28ts)

[2.1. Diagrama de casos de uso 10](#_heading=h.tyjcwt)

[2.2. Atores 10](#_heading=h.3dy6vkm)

[2.3. Detalhamento dos casos de uso 11](#_heading=h.1t3h5sf)

[2.4. Projeto de Interface 17](#_heading=h.m2sa0aqmospg)

[Tela de acesso 17](#_heading=h.7986wabwibph)

[Tela principal 17](#_heading=h.tmsv8jmc7t9q)

[Telas de Parâmetro e Acessibilidade 18](#_heading=h.9t0hc3n468z)

[Telas de Cadastro de Usuário 20](#_heading=h.fpo0kk33c7ta)

[Telas de Cadastro de Função 23](#_heading=h.fslpkiaqyhzr)

[Telas de Cadastro de Equipe 25](#_heading=h.oiulr83cal9q)

[Telas de Cadastro de Projeto 27](#_heading=h.f6woutpj6lgq)

[Telas de Apontamento 30](#_heading=h.klmjipkc4smp)

[Telas de Análise 33](#_heading=h.noje2zl7i2jh)

[2.5. Diagrama de classes 35](#_heading=h.xlt318ipxpxw)

[**3. Projeto 38**](#_heading=h.t04jdoi7hq2k)

[3.1. Arquitetura de software 38](#_heading=h.j960i9yin5vo)

[Definição tecnológica 42](#_heading=h.inzu4jbuwotn)

[3.2. Arquitetura da informação 42](#_heading=h.kjdh9ngpteic)

[**4. Testes 44**](#_heading=h.lnxbz9)

[**Plano de teste 44**](#_heading=h.q26uq8952f53)

[**Evidências de testes realizados 44**](#_heading=h.62h87ckmogv5)

[**5. URLs 50**](#_heading=h.grmab16n88az)

[5.1. Aplicação web 50](#_heading=h.1ksv4uv)

[5.2. Repositório código-fonte 50](#_heading=h.44sinio)

[5.3. Vídeo de apresentação do trabalho 51](#_heading=h.2jxsxqh)

# 1. Apresentação

## 1.1. Contexto

Durante o desenvolvimento de um projeto de software é necessário o acompanhamento do esforço investido pela equipe para a análise, desenvolvimento, validação e implantação. Esse acompanhamento pode ser necessário para mensurar a lucratividade e/ou a precisão entre o que foi estimado e os recursos que foram utilizados, além disso em alguns modelos de negócio, esse acompanhamento pode ser utilizado para faturamento das horas trabalhadas em intervalos definidos junto ao cliente.

Pode-se presumir que é um acompanhamento de grande importância para uma empresa que desenvolve software e que precisa monitorar se sua operação está fluindo adequadamente ou criar estratégias de negócio baseadas nestes dados, e em um cenário de fábrica de software, modelo com desenvolvimento mais pontual e dinâmico do que o desenvolvimento evolutivo de um único produto, esse acompanhamento passa a ser crucial.

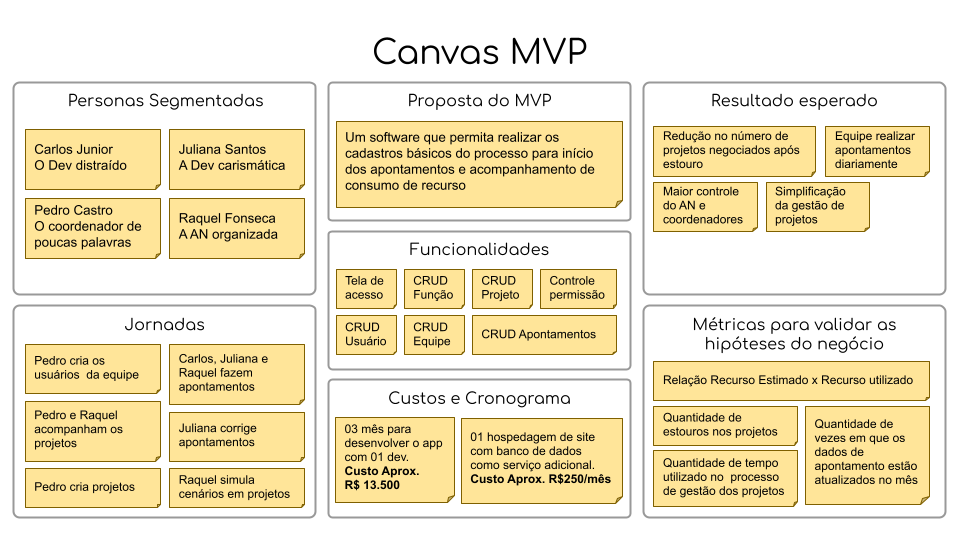
Neste contexto, uma ferramenta que permita o controle de estimativas e acompanhamento de recursos alocados agregaria valor ao ciclo de produção de um software auxiliando a gestão das informações.

Um fator importante neste tipo de ferramenta é a usabilidade, sendo assim deve ser possível para cada perfil de usuário acessar e utilizar com poucos cliques as funcionalidades, por se tratar muitas vezes de uma tarefa repetitiva e constante em sua jornada de trabalho.

Por exemplo, para que um desenvolvedor possa realizar seus apontamentos no início e no final de cada atividade do seu dia, um formulário com vários campos de preenchimento não parece ser a melhor opção, uma vez que suas atividades podem não seguir um cronograma bem definido ocorrendo a necessidade de alternar várias vezes entre atividades ao longo do dia.

Outro aspecto fundamental neste tipo de ferramenta é a capacidade de simular e organizar os dados gerando informações que deem suporte à gestão estratégica das demandas. Um exemplo de como a falta ou o entendimento tardio de algumas informações pode ser prejudicial para a operação é o caso de um projeto ultrapassar a quantidade de horas estimadas e a gestão perceber apenas após as horas terem terminado, a negociação com o cliente fica mais complexa.

Estas são algumas oportunidades dentre outras que podem ser identificadas em uma empresa que utiliza um software para gestão de projetos e apontamentos que não possui funcionalidades com foco em um controle ágil e simples, e que não permita a utilização em fluxos de trabalho cuja alternância entre demandas, priorização e negociação com o cliente são constantes inviabilizando a elaboração de cronogramas muito detalhados.

****

## 1.2. Público alvo

Apesar do controle de apontamentos ser uma atividade utilizada por empresas que atuam no desenvolvimento de software das formas mais variadas, este trabalho olha especialmente para o modelo de fábrica de software. Ambientes que trabalham nesse modelo se diferenciam por ter uma rotina de trabalho mais dinâmica com projetos que podem variar de tamanho (de algumas horas à meses) e com poucos ou nenhum projeto com objetivo de evolução, em grande parte do tempo trabalhando em integrações ou melhorias pontuais.

Flexibilidade, simplicidade e autonomia se tornam pilares de uma boa ferramenta para controle de apontamentos neste cenário, pois em muitos casos os ritos das práticas ágeis consolidadas no mercado podem não ser viáveis e os fluxos de entrada de demanda podem variar.

Por exemplo: pense nas práticas que o Scrum descreve, com reuniões de abertura de sprint, dailys, reuniões de fechamento de sprint, elas parecem se encaixar muito bem em projetos maiores, porém tentar seguí-las em uma situação onde o cliente precisa de um ajuste em lote para um cadastro incorreto de seus produtos provavelmente atrapalha mais do que ajuda, principalmente por ser uma tarefa de poucas horas e alta prioridade para o cliente.

O mesmo se aplica a uma ferramenta de controle que presume um fluxo rígido para que seja possível realizar os apontamentos, fazendo com que o desenvolvedor tenha que anotar em papel e criar lembretes para realizar o apontamento após todas as etapas exigidas pela ferramenta estarem finalizadas, criando aberturas para falhas ou atrasos nos apontamentos.

Logo, mesmo que possa ser utilizado em qualquer cenário, os alvos deste trabalho são ambientes em que as mudanças são constantes e os fluxos de trabalho tenham como característica a necessidade de mais flexibilidade, sendo assim a solução seguirá a filosofia: “Começar, mesmo que sem todos os dados que gostaríamos, é melhor do que começar tardiamente ou nem começar”.

## 1.3. Requisitos

Os requisitos detalhados abaixo foram identificados com o auxílio da metodologia Lean Inception, portanto sugiro que veja o anexo [Levantamento usando Lean Inception](#_heading=h.4boehhms6e54).

*Requisitos funcionais*

* Controle de acesso a aplicação
  + Deve ser possível acessar usando usuário, senha e o código do google authenticator
    - O usuário deve ter sido cadastrado/aprovado antes
    - Esses usuários podem acessar a Web API, se for parametrizado
  + Deve ser possível acessar com e-mail do google
    - O usuário possui as permissões padrões definidas na aplicação
    - Deve ter o cadastro aprovado posteriormente
    - No primeiro login após cadastrado/aprovado deve ser solicitada a criação da senha e nome de usuário
* Parametrização do controle de acesso
  + Pode ser dada permissão por usuário, função e/ou equipe
  + A ordem de sobreposição das permissões é usuário > função > equipe, ou seja, caso um desenvolvedor esteja em uma função que não permita abrir a aplicação X, porém, o seu usuário tenha essa permissão, ele abrirá a tela
  + Deve ser parametrizada as permissões padrões para acessos não aprovados
  + As permissões das aplicações deve ser por nível de tela e de funcionalidade
  + A ordem de sobreposição das permissões é funcionalidade > tela, ou seja, mesmo que um usuário possa utilizar uma tela, ele pode não conseguir executar todas as funcionalidades
* Tela para cadastro de usuário
  + A tela deve ter as funcionalidades básicas de criação, leitura, atualização e remoção dos dados
  + Nesta tela terá a funcionalidade de aprovação dos usuários que acessaram de modo autônomo
  + Nesta tela deve apresentar os dados em grid e/ou totalizadores e gráficos
* Tela para cadastro de função
  + A tela deve ter as funcionalidades básicas de criação, leitura, atualização e remoção dos dados
* Tela para cadastro de equipe
  + A tela deve ter as funcionalidades básicas de criação, leitura, atualização e remoção dos dados
  + Nesta tela deve apresentar os dados em grid e/ou totalizadores e gráficos
* Tela para cadastro de projeto
  + A tela deve ter as funcionalidades básicas de criação, leitura, atualização e remoção dos dados
  + Nesta tela deve apresentar os dados em grid e/ou totalizadores e gráficos
* Tela para cadastro de apontamento
  + A tela deve ter as funcionalidades básicas de criação, leitura, atualização e remoção dos dados
  + Essa tela é um formulário para entrada de apontamentos feitos fora do fluxo principal de trabalho
* Tela de apontamentos
  + Deve listar todos os projetos que o usuário tem vinculados
  + Permitir o apontamento usando botões de iniciar e encerrar atividade
  + Trazer os dados ordenados por último apontamento feito
  + Caso um novo apontamento seja iniciado encerrar o anterior
  + Permitir informar uma observação no apontamento
  + Nesta tela deve apresentar os dados em grid e/ou totalizadores e gráficos
* Geração de dados de produtividade a partir dos lançamentos de apontamento
  + Esses dados devem ser gerados em todos os níveis (usuário, função, equipe, projeto)
  + Esses valores devem estar presente nas telas de seus respectivos cadastros
  + Devem apoiar na simulação/análise de projetos
* Tela para análise de projetos
  + Permitir ao usuário projetar qual a quantidade de recurso que será utilizado ou a data de entrega baseado no histórico do projeto
  + Mostrar os dados no formato Grid e/ou totalizadores e gráficos
* Web API
  + Deve ser possível exportar dos dados no formato JSON
  + Deve permitir importar dados no formato JSON
  + Dados disponíveis:
    - Usuário
    - Função
    - Equipe
    - Projeto
    - Apontamento
    - Simulação

*Requisitos não funcionais*

* As telas devem ser responsivas
* As telas devem ter um padrão consistente de layout para facilitar a usabilidade
* Todas as funcionalidades devem apresentar resposta à ação do usuário com mensagens de texto ou indicadores visuais
* Ícones devem ser condizentes com seu contexto na tela e devem sempre ter texto alternativo (tooltip)
* A aplicação deve ter mais de um conjunto de cores em seu padrão visual para atender ao público com características especiais no sentido da visão (sensibilidade e/ou daltonismo)
* A aplicação deve ter um modo de leitura para o público com baixa ou nenhuma capacidade no sentido da visão
* A aplicação deve ter um modo de atalhos de clique único para o público com características especiais na coordenação motora

# 2. Modelagem

## 2.1. Diagrama de casos de uso

O diagrama de caso de uso elaborado para esse trabalho é apresentado ao final do tópico “2.3. Detalhamento dos casos de uso”, e nele é possível ter uma visão macro das principais funcionalidades e como elas se relacionam.

O diagrama foi dividido em quatro partes, para que fosse possível colocá-lo neste documento de modo a ficar legível para leitura, porém, as primeiras três partes se relacionam diretamente, sendo as partes dois e três continuação da parte um. Por fim, a parte quatro apresenta as funcionalidades de importação e exportação de dados que foi idealizado para ocorrer via Web Service API.

## 2.2. Atores

Segue abaixo a lista dos atores apresentados no diagrama de caso de uso e uma descrição resumida deles:

* Usuário sem cadastro: Trata-se dos indivíduos que desejam acessar a aplicação, porém, fazem isso através do login usando uma conta de e-mail do Google. Estes usuários obrigatoriamente não tiveram seu acesso autorizado ou negado por algum usuário previamente cadastrado e com permissões para isso.
* Usuário com cadastro: Trata-se dos indivíduos que acessam a aplicação com usuário e senha pessoais. Estes usuário obrigatoriamente tiveram seu acesso autorizado por um usuário previamente cadastrado e com permissões para isso.
* Usuário com permissão: Trata-se dos indivíduos que possuem cadastro autorizado e que receberam permissões que os possibilita consultar, autorizar ou negar acesso aos demais usuários.
* Usuário: Categoria genérica que engloba todos os demais tipos de usuário citados acima.

## 2.3. Detalhamento dos casos de uso

Diagrama de caso de uso - Parte 1: Acesso a aplicação

O acesso à aplicação, seja por usuário com login nominal e pessoal ou por usuário sem cadastrado utilizando uma conta do Gmail, ocorre pela mesma tela. Após o acesso ambos são direcionados para a tela principal da aplicação, porém, antes disso o usuário com cadastro tem suas permissões validadas, e o usuário sem cadastro tem o registro do acesso salvo para avaliação posterior de autorização ou recusa e as permissões previamente definidas para o acesso sem cadastro são utilizadas para ele.

Uma vez na tela principal apenas as funcionalidades permitidas devem estar habilitadas para serem utilizadas pelo usuário. Essa parametrização de permissões deve ser estabelecida previamente por algum usuário com cadastro e permissões para fazer isso na aplicação.

Com relação ao acesso de um usuário sem cadastro, ele poderá ocorrer apenas com as permissões para esse tipo de acesso enquanto seu usuário não é verificado devidamente. Após a verificação, ou o usuário terá seu acesso negado e deste modo não poderá mais acessar com o e-mail utilizado, ou o usuário terá o acesso autorizado e em seu próximo acesso será solicitada a criação da senha e cadastramento do google autenticador para ser utilizado como dupla autenticação nos acessos.

Diagrama de caso de uso - Parte 2: Menu de parametrização

Assim que estiver na tela principal da aplicação, caso tenha permissão, o usuário poderá acessar e alterar as definições relacionadas com as permissões por usuário, função ou equipe, além de alterar as permissões padrões para acessos sem cadastro, caso queira. Também é no menu de parametrizações que o usuário poderá alterar as próprias configurações de acessibilidade conforme desejar.

Diagrama de caso de uso - Parte 3: Demais menus da aplicação

Assim que estiver na tela principal da aplicação, caso tenha permissão, o usuário poderá acessar as telas de cadastro (Usuário, Função, Equipe e Projeto) e fazer os ajustes desejados. O usuário poderá acessar a tela de apontamentos para registrar suas atividades ou fazer ajustes de apontamentos, gerando dados para apoiar na estimativa de produtividade dele.

Outro recurso possível é a tela de simulação de projetos que permitirá ao usuário ter uma noção de tempo baseado no histórico de apontamentos do projeto e/ou no indicador de produtividade da equipe e dos colaboradores. As consultas de informações permitirão ao usuário ver os dados em grid ou através de gráficos e totalizadores, auxiliando na análise realizada.

Diagrama de caso de uso - Parte 4: Importação e exportação de dados

Para importação e exportação de dados, uma Web API estará disponível para que usuários com cadastro possam acessar e automatizar o processo de cadastro de informações ou coleta de dados para acompanhamento. Todos os dados da aplicação devem estar disponíveis para esse processo, com exceção das informações de parametrização de permissões e acessibilidade, sendo obrigatória a utilização da aplicação para isso.

Diagrama de caso de uso - Parte 1: Acesso a aplicação

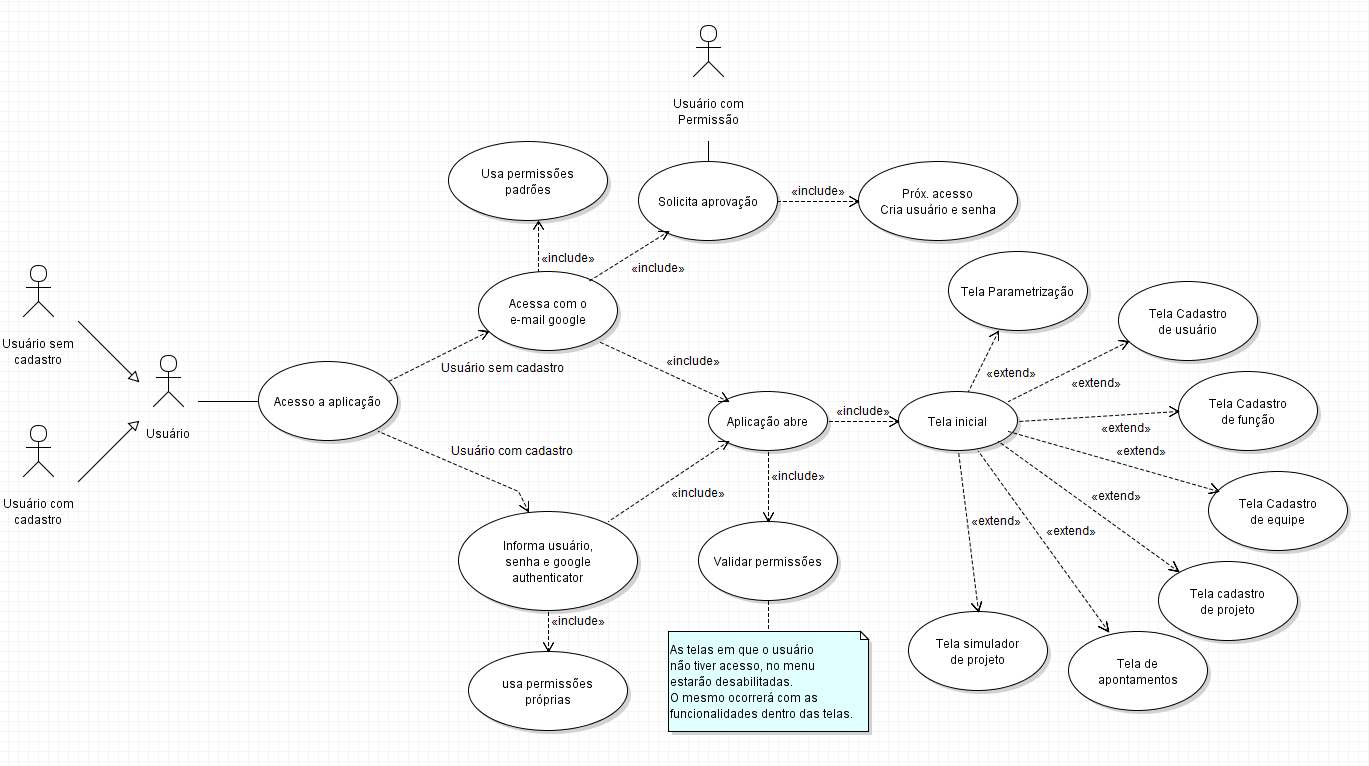
****

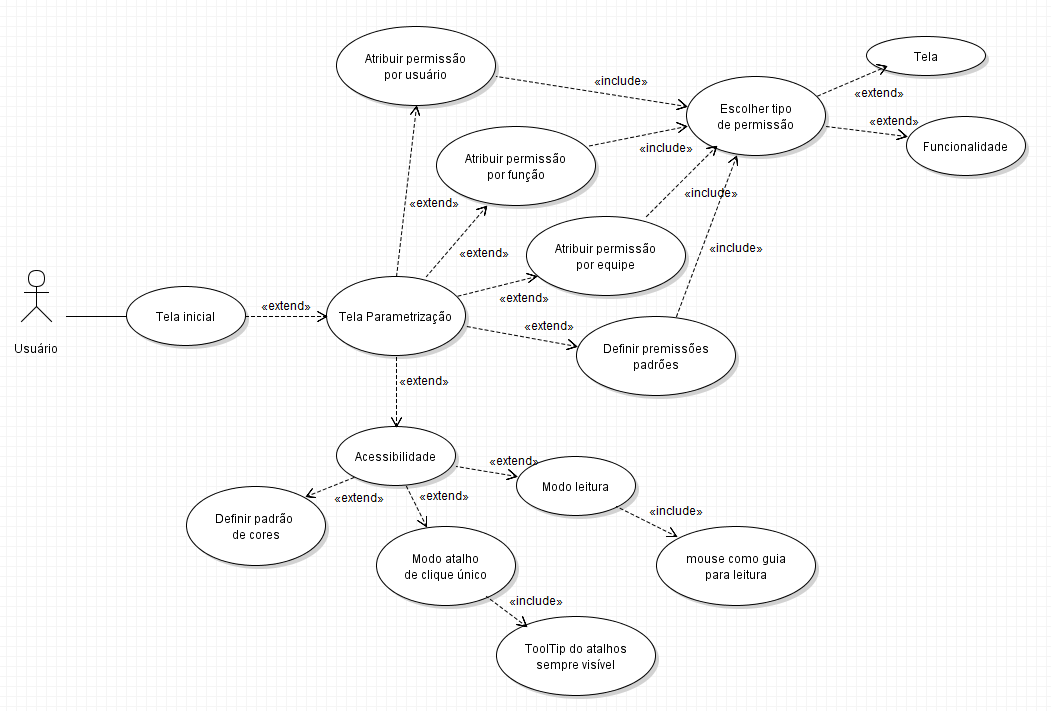
Diagrama de caso de uso - Parte 2: Menu de parametrização

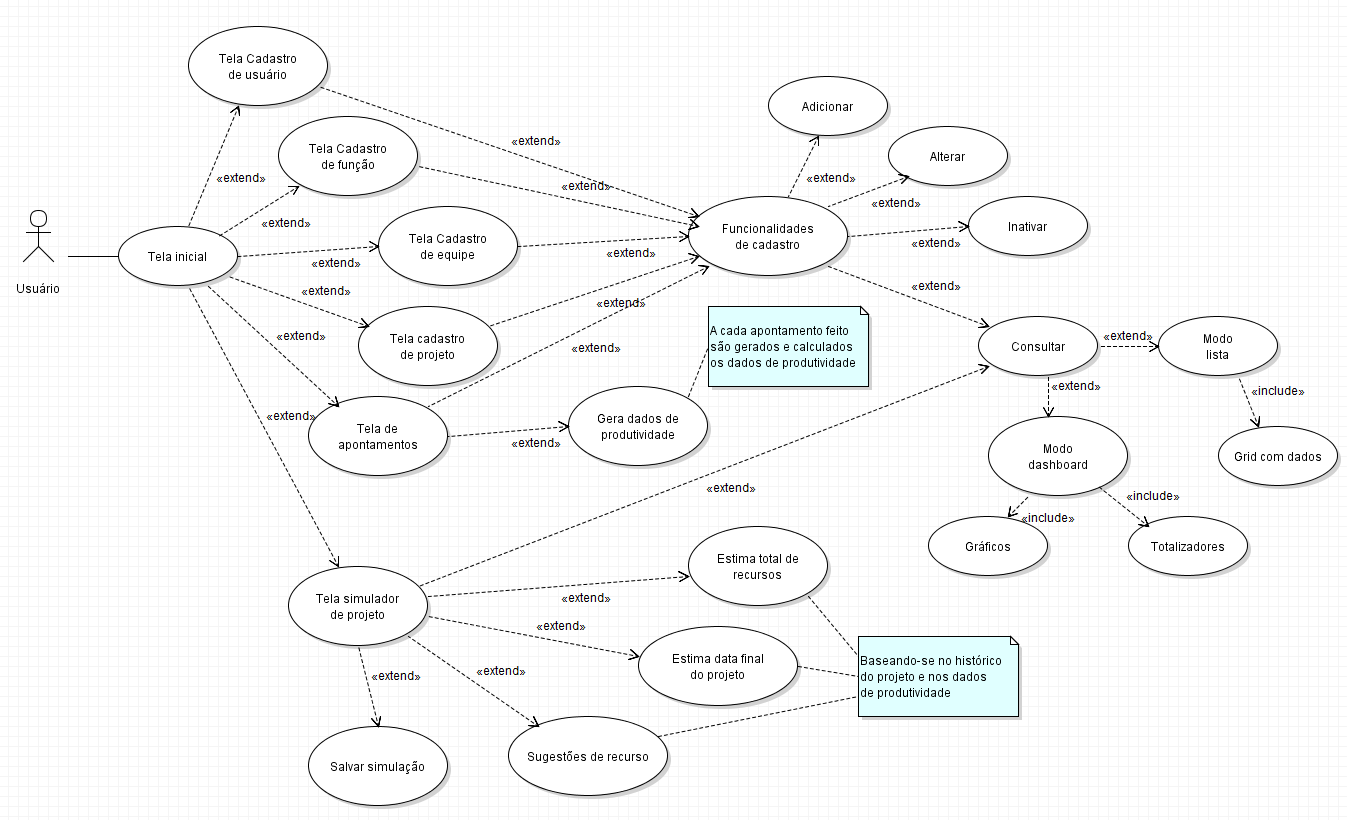
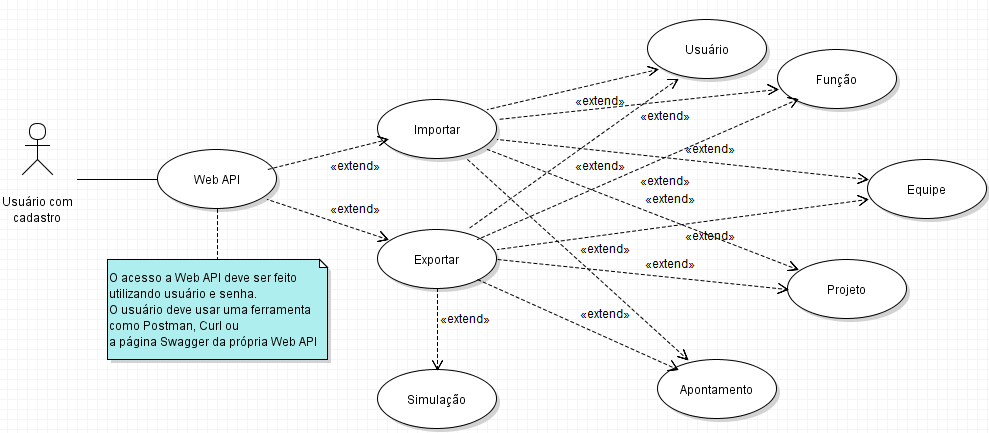
Diagrama de caso de uso - Parte 3: Demais menus da aplicação

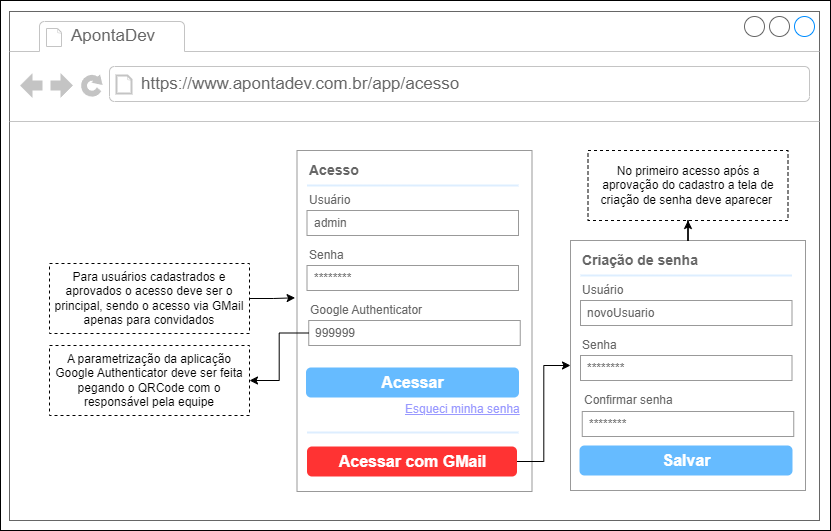
Diagrama de caso de uso - Parte 4: Importação e exportação de dados

****

## 2.4. Projeto de Interface

### Tela de acesso

Ao acessar a aplicação web a primeira tela que o usuário verá é a tela de acesso. Nela ele poderá informar os dados pessoais para acesso, redefinir a senha usando o link “Esqueci minha senha” ou acessar a aplicação usando a conta do Gmail através do botão “Acessar com o GMail”.

Após a aprovação do acesso, no acesso seguinte ao clicar no botão “Acessar com o GMail” a tela de criação de senha será apresentada para o usuário criar a senha. A configuração do google autenticador deve ser feita entre um membro da equipe e o novo usuário em um momento posterior do processo.

### 

### Tela principal

Após o acesso o usuário será direcionado para a tela principal da aplicação. Nesta tela o usuário terá acesso a todos os menus e funcionalidades definidas nas regras de permissão do seu perfil. As telas e funcionalidades que o usuário não tiver acesso apareceram inativas para ele não respondendo a cliques.

O usuário poderá usar alguns componentes predefinidos como gráficos, totalizadores e ícones de atalho para personalizar sua tela inicial e facilitar a utilização da ferramenta em sua rotina de trabalho. Para personalizar, ele apenas precisará clicar em áreas indicadas na tela e escolher o elemento que deseja que apareça no local.

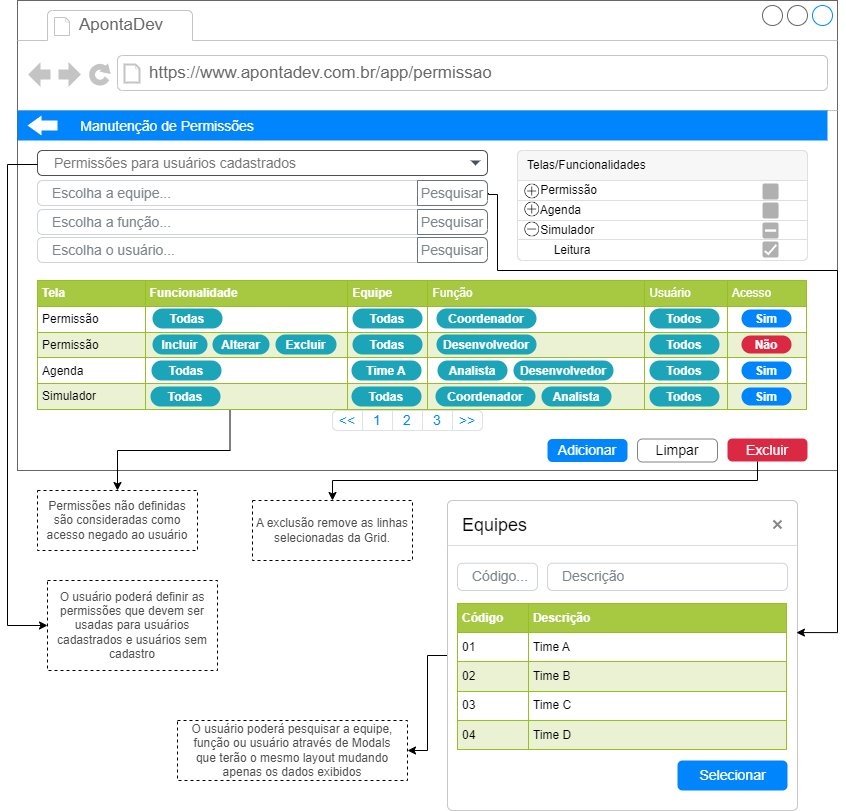
### 

### Telas de Parâmetro e Acessibilidade

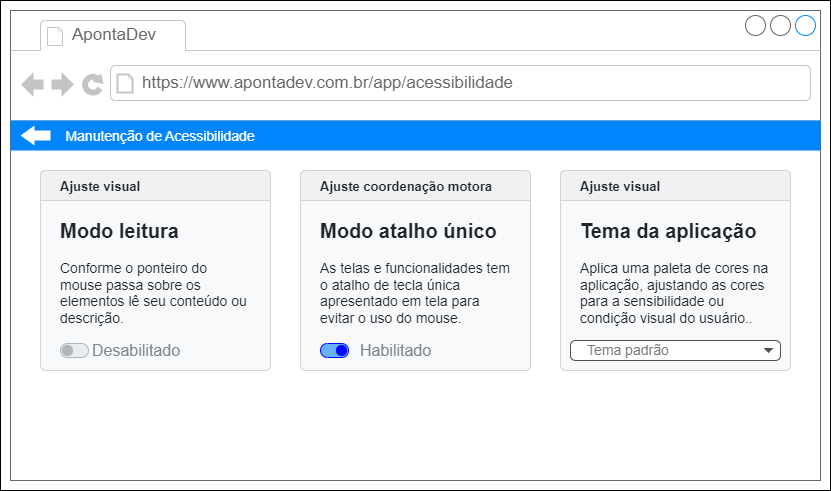
Ao acessar o menu Parâmetro e submenu Permissão o usuário verá a tela que o permite fazer manutenções nas regras de permissões da aplicação.

Na parte superior da tela temos os componentes que são usados para informar qual o perfil da regra que criaremos (Para usuários cadastrados ou acesso autônomo), quais os alvos da regra que será criada (equipe, função e/ou usuário) e quais telas e funcionalidades serão contempladas. Quando necessário o componente terá telas de pesquisa para facilitar a busca das informações.

Logo abaixo temos a grid que lista todas as regras existentes e que permite ao usuário selecionar as linhas para fazer alterações pontuais ou em lote e excluir as linhas se desejar. A grid tem um controle de navegação para que seja possível ver todas as regras caso existam muitas.

Na parte inferior da tela temos os botões para que o usuário possa executar as ações de adicionar, limpar e excluir. O adicionar cria uma nova regra a partir dos dados informados nos componentes da tela, caso uma ou mais linhas da grid estejam selecionadas, o adicionar atualizará elas com os dados informados nos componentes da tela. O limpar remove ou volta para os valores padrões os componentes da tela, com exceção da grid. Por fim, o excluir remove as regras que estejam selecionadas na grid no momento do clique.Ao acessar o menu Parâmetro e submenu Acessibilidade o usuário verá a tela que permite ajustar algumas definições de acessibilidade da aplicação com opções predefinidas.

As opções disponíveis para configurar são:

* Modo leitura - O usuário informará se deseja que quando passar o mouse sobre o texto ele seja lido.
* Modo atalho único - Quando ativado esse modo sempre apresenta as teclas de atalho nos elementos e seu objetivo é que com uma única tecla o usuário possa usar as funcionalidades.
* Tema da aplicação - O usuário poderá escolher dentre temas predefinidos aquele que mais facilita a utilização da aplicação.

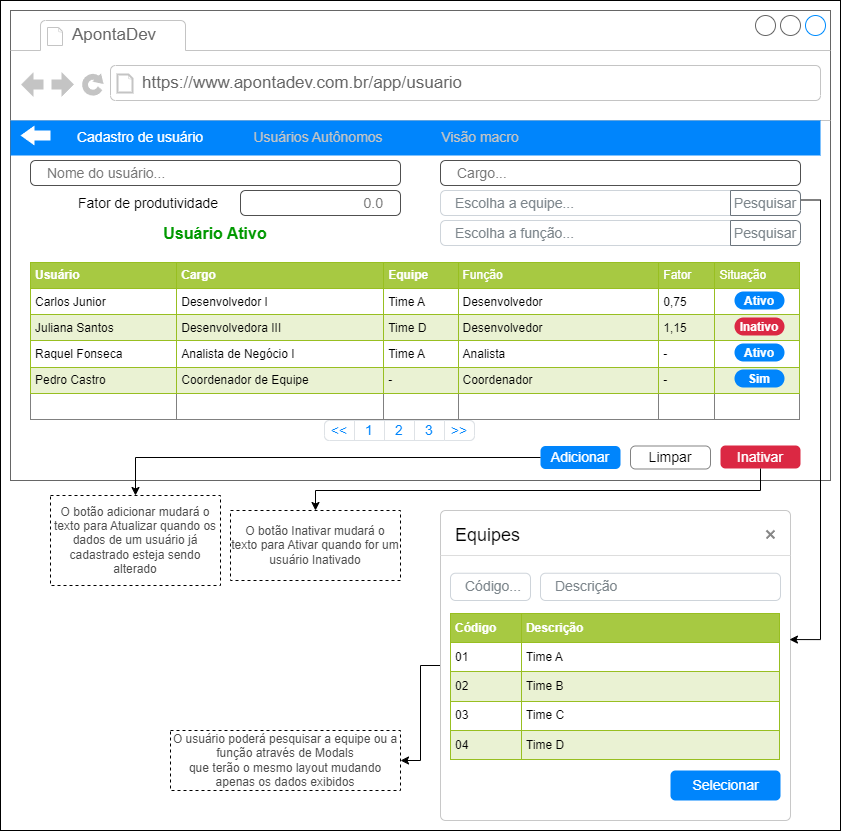
### 

### Telas de Cadastro de Usuário

Ao acessar o menu Cadastro e submenu usuário, o usuário verá a tela que permite dar manutenção no cadastro de usuários da aplicação. No menu desta tela o usuário terá as opções Cadastro de usuário, Usuário autônomo e visão macro.

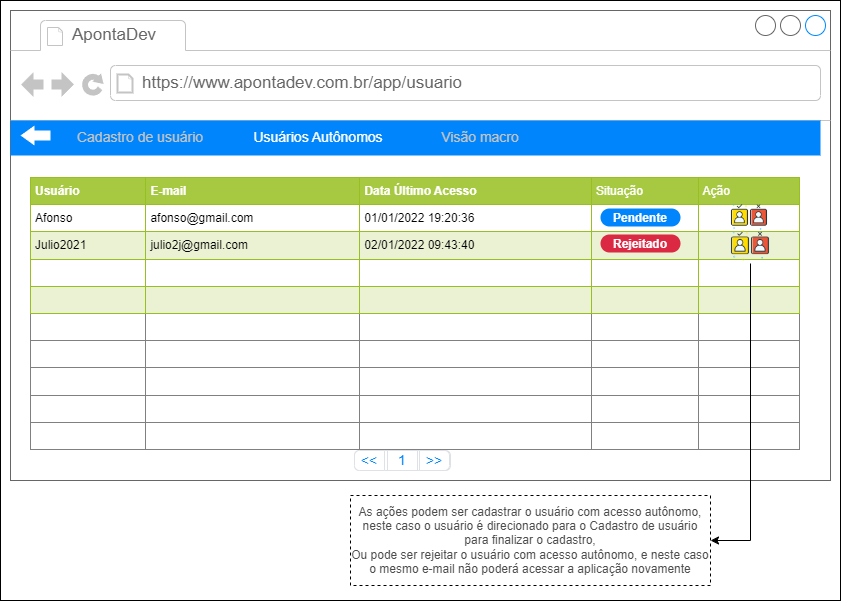
No cadastro de usuário será possível visualizar na grid todos os usuários cadastrados (ativos ou inativos), e preenchendo os dados nos componentes da parte superior da tela será possível criar um novo usuário ou atualizar informações em linhas selecionadas da grid. O limpar remove ou volta para os valores padrões os componentes da tela, com exceção da grid, e o botão inativar altera o estado dos usuários selecionados na grid para inativados.

O campo Fator de produtividade será apenas de leitura e apresentará o valor calculado a partir dos apontamentos realizados pelo usuário na ferramenta.

Em usuários autônomos, o usuário terá uma grid listando todos os acessos feitos de modo autônomo e poderá usar a coluna ações escolher entre rejeitar ou aprovar o acesso.

Ao rejeitar o acesso a linha permanece sendo exibida na grid, porém, com a situação indicando que está rejeitada, e o usuário não poderá acessar mais a aplicação com o mesmo e-mail. Mas, caso algum usuário altere a linha de rejeitada para aprovada, o usuário é liberado na aplicação novamente.

Ao aprovar um acesso a linha desaparece da grid, e começa a ser exibida na aba cadastro de usuário onde os dados podem ser completados e no próximo acesso o usuário criará a senha pessoal.

Então nenhuma ação for feita a linha fica como pendente, e o usuário pode ter acesso, porém sempre como acesso autônomo seguindo as regras gerais de permissão para esse tipo de acesso.

Em visão macro, o usuário verá totalizadores e gráficos relacionados aos dados do cadastro de usuário e acessos de usuários autônomos. Os dados serão apenas para consulta e serão pré-definidos, não sendo possível customizar a visualização.

### 

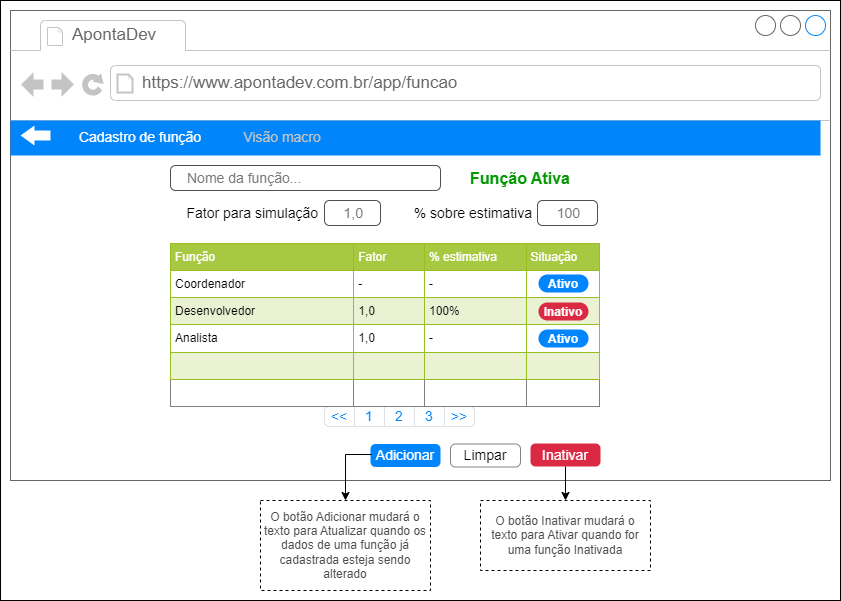
### Telas de Cadastro de Função

Ao acessar o menu Cadastro e submenu Função, o usuário verá a tela que permite dar manutenção no cadastro de funções da aplicação. No menu desta tela o usuário terá as opções cadastro de função e visão macro.

A função deve ser entendida como uma forma de agrupar os profissionais por características e acessos similares, portanto o usuário poderá criar uma função chamada desenvolvedor e atribuir ela para membros do time com cargos de senioridades diferentes ou criar funções mais específicas como desenvolvedor I e desenvolvedor II e atribuir elas a membros da equipe usando como regra a senioridade do cargo.

Em cadastro de função, o usuário poderá criar a função preenchendo os componentes abaixo:

* Nome da função - Um nome definido pelo usuário
* Fator para simulação - Valor decimal que representa o fator ideal para o cargo ao ser analisado nas simulações. Esse valor não corresponde ao obtido através dos apontamentos, mas um valor ideal.
* % sobre estimativa - Valor de 0 a 100 que indica quanto de um valor estimado espera-se que essa função utilize. Esse campo será usado para sugestão do consumo de horas do projeto.

A grid apresentará a lista de função (ativas ou inativas) cadastradas, e os botões de ação na parte inferior da tela permitem ao usuário adicionar (ou alterar caso tenham linhas selecionadas na grid), limpar os dados (exceto da grid) e inativar funções que estiverem selecionadas na grid.

Em visão macro, o usuário verá totalizadores e gráficos relacionados aos dados do cadastro de função. Os dados serão apenas para consulta e serão pré-definidos, não sendo possível customizar a visualização.

### 

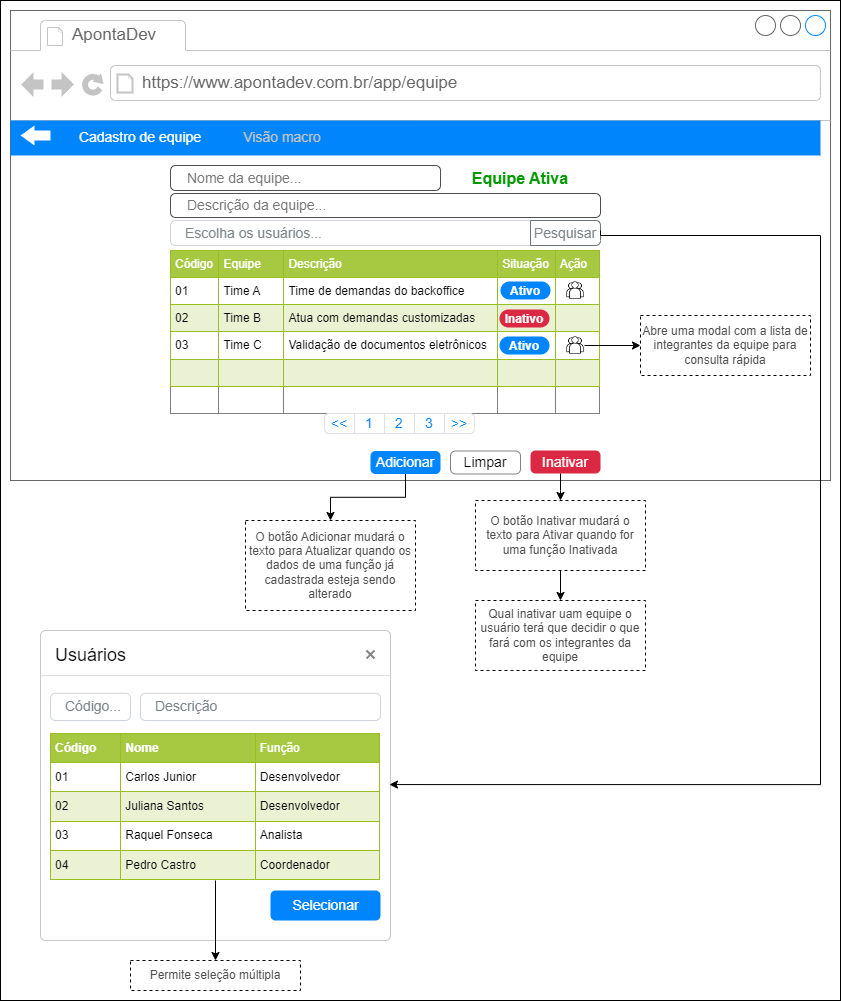
### Telas de Cadastro de Equipe

Ao acessar o menu Cadastro e submenu Equipe, o usuário verá a tela que permite dar manutenção no cadastro de funções da aplicação. No menu desta tela o usuário terá as opções cadastro de equipe e visão macro.

Em cadastro de equipe, o usuário poderá criar um nome para a equipe, colocar uma descrição (opcional) e indicar quais usuários farão parte da equipe que está sendo criada. Como a inclusão de um usuário em uma equipe pode ser feita através da aplicação de cadastro do usuário e do cadastro de equipe, essa informação será sempre a mesma em ambas as telas, evitando confusos.

A grid da tela listará todas as equipe criadas (ativas ou inativas) e na coluna ação haverá a opção do usuário abrir uma pequena tela listando todos os membros da equipe.

Na parte inferior da tela os botões permitem ao usuário executar as operações básicas de adicionar (ou alterar caso tenham linhas selecionadas na grid), limpar os dados (exceto da grid) e inativar equipes selecionadas na grid.



Em visão macro, o usuário verá totalizadores e gráficos relacionados aos dados do cadastro de função. Os dados serão apenas para consulta e serão pré-definidos, não sendo possível customizar a visualização.

### 

### Telas de Cadastro de Projeto

Ao acessar o menu Cadastro e submenu Projeto, o usuário verá a tela que permite dar manutenção no cadastro de projetos da aplicação. No menu desta tela o usuário terá as opções cadastro de projeto e visão macro.

Em cadastro de projeto, o usuário terá na parte superior os componentes para preencher os dados e verá o status do projeto. Segue abaixo lista de dados:

* Nome do projeto: Campo aberto para digitação e obrigatório
* Código externo: Campo aberto para digitação e opcional
* Nome do cliente: Campo aberto para digitação e opcional
* Descrição do projeto: Campo aberto para digitação e opcional
* Tipo de projeto: Campo de seleção que permite informar se o projeto é para uso interno (Horas não faturadas) ou uso externo (projetos pagos por clientes)
* Previsão de conclusão: Campo do tipo data para que o usuário estipule qual a previsão inicial de terminar o projeto
* Vigência do projeto:Campo do tipo data inicial e final para o usuário indicar o período em que o projeto estará ativo para apontamento.
* Horas estimadas: Campo numérico para o usuário informar a previsão inicial de horas que serão consumidas no projeto

Uma grid com os projetos cadastrados atualmente na aplicação e uma coluna de ações para que o usuário possa associar os integrantes do projeto e com possibilidade de ver o histórico de alterações no projeto. Também terá o controle de navegação da grid caso existam muitas linhas.

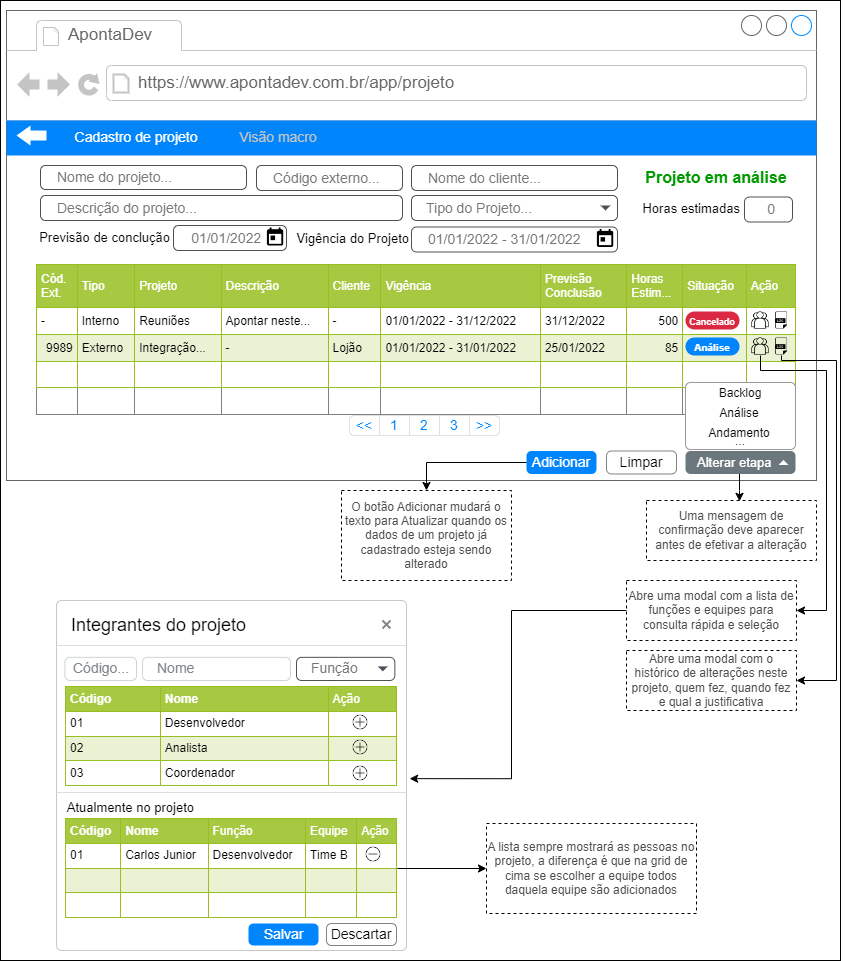
Na parte inferior da tela, temos os botões de ação:

* Adicionar: Usado para criar um novo projeto, ou caso existam linhas selecionadas na grid, este botão atualizará as linhas selecionadas com as informações dos componentes
* Limpa: Usado para limpar ou voltar para o valor padrão os componentes da tela (exceto a grid)
* Alterar etapa: Usado para alterar a etapa do projeto
  + Backlog: O projeto foi criado, mas ainda não entrou na esteira de desenvolvimento
  + Análise: O projeto está sendo analisado para refinamento de escopo, prazo e estimativa de horas
  + Andamento: O projeto iniciou o desenvolvimento com o time
  + Concluído: O projeto foi encerrado
  + Cancelado: O projeto é cancelado e não pode mais ser utilizado

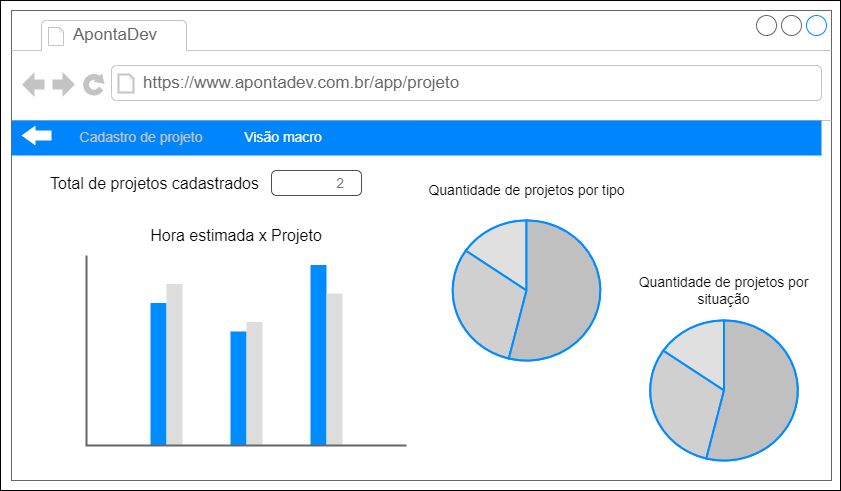
No modal Integrantes do Projeto, o usuário poderá selecionar uma equipe, uma função ou um membro do time individualmente tornando mais rápido o processo de inclusão de pessoas ao projeto.

Nos três campos da modal, o usuário pode selecionar se deseja pesquisar por Equipe, Função ou Usuário, e digitar o código ou a descrição para que na primeira grid apareça as opções disponíveis. Na coluna ação da grid superior é possível ao usuário usar o botão de inclusão para listar os integrantes por usuário na grid inferior. Na grid inferior, o usuário tem o botão remover na coluna ação para retirar do projeto algum integrante.

O botão salvar confirma as alterações realizadas e o botão descartar ignora as alterações feitas na modal.



Em visão macro, o usuário verá totalizadores e gráficos relacionados aos dados do cadastro de projeto. Os dados serão apenas para consulta e serão pré-definidos, não sendo possível customizar a visualização.



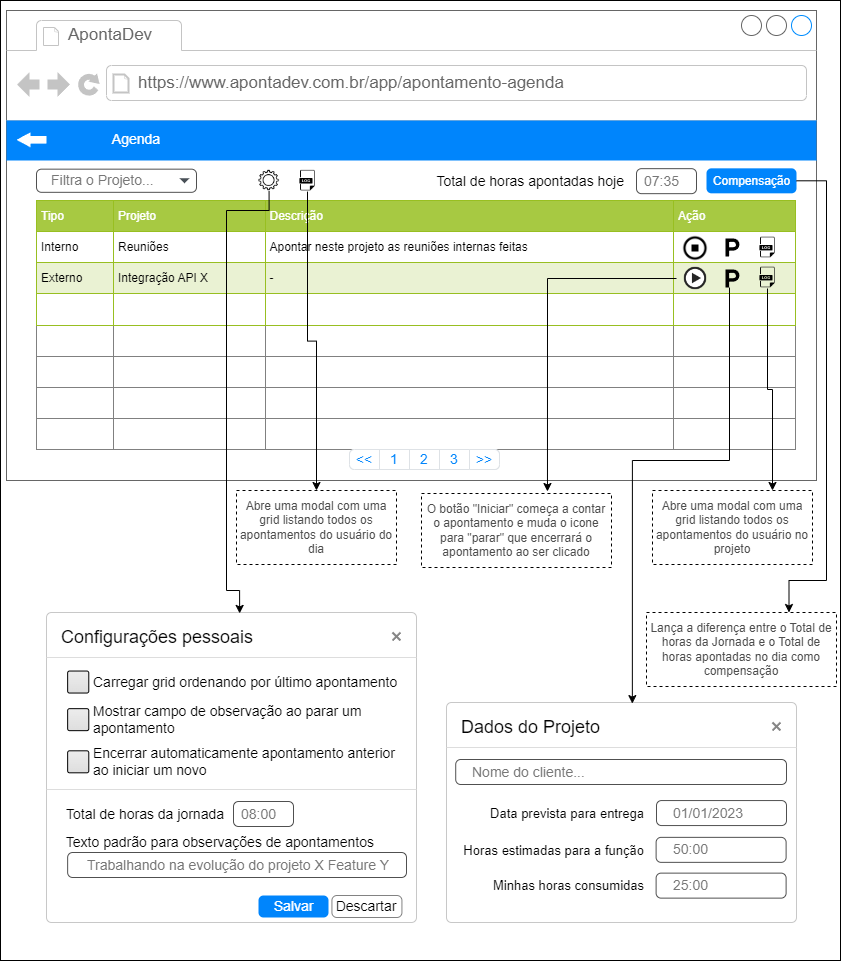
### Telas de Apontamento

Ao acessar o menu Apontamentos e submenu Agenda, o usuário verá a tela que permite realizar os apontamentos para projetos em que o usuário está vinculado.

Na parte superior da tela, o usuário verá os componentes:

* Filtra o Projeto: Campo de seleção que permite pesquisar um projeto específico que esteja na grid da tela
* Configuração: Com ícone de engrenagem, abre uma modal que possibilita ao usuário definir algumas configurações pessoais desta tela
* Histórico: Com ícone de log, abre uma modal com a lista de apontamentos feitos na data atual de utilização
* Total de horas apontadas hoje: Um campo apenas para leitura que apresentará o total de horas apontadas na data atual da utilização
* Compensação: Um botão que permite lançar a diferença entre o que foi apontado no dia e o total esperado de apontamento no dia, como apontamento de compensação de horas para os casos em que o expediente é encerrado mais cedo

Na grid serão listados todos os projetos em que o usuário esteja vinculado, e na coluna ação da grid o usuário poderá:

* Iniciar/Parar: Inicia a contagem de tempo para um apontamento e marca o encerramento do apontamento registrando ele no histórico do usuário
* Dados do Projeto: Abre uma modal que apresenta os principais dados do projeto
* Histórico: Abre uma modal que lista o histórico dos lançamentos de apontamento para o projeto da linha clicada

Ao acessar o menu Apontamentos e submenu Manutenção, o usuário verá a tela que permite realizar a inclusão ou ajuste nos apontamentos.

Na parte superior da tela, o usuário verá os componentes:

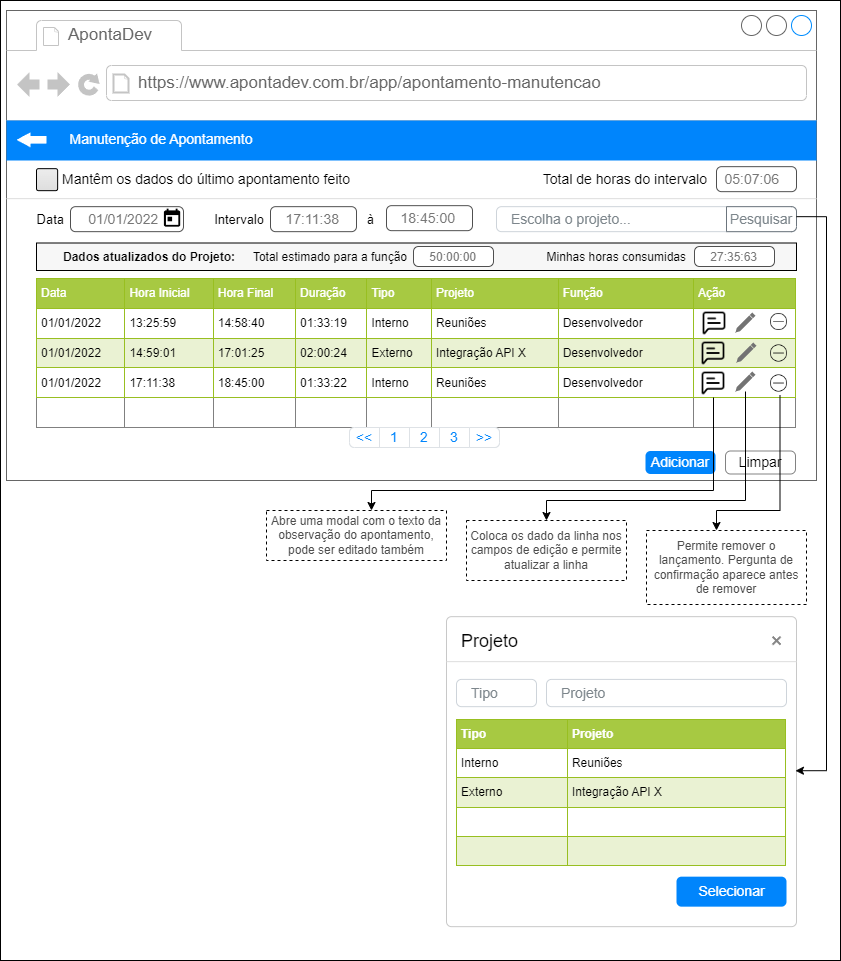
* Mantêm os dados do último apontamento feito: Indicador que se marcado mantêm os dados do último apontamento feito na tela para iniciar um novo apontamento
* Total de horas do intervalo: Campo de horas apenas para leitura que totaliza para o usuário as horas dos apontamentos que foram filtrados na grid
* Data: Campo do tipo data que o usuário utilizará para criar um novo apontamento ou para atualizar a data de algum apontamento
* Intervalo: Campos do tipo hora para o usuário informar a hora inicial e a final do apontamento que será criado ou para dar manutenção em apontamentos existentes
* Escolha o projeto: Campo para pesquisa do projeto que será utilizado em criar um novo apontamento ou para dar manutenção em apontamentos existentes
* Dados Atualizados do Projeto: Após selecionar um projeto no campo “Escolha o projeto” os campos para para leitura deste grupamento são atualizados

A grid listará os apontamentos existentes do usuário, e na coluna ação o usuário poderá:

* Abrir uma modal para ver ou editar uma observação relativa ao apontamento selecionado
* Permite remover o apontamento selecionado

Na parte inferior da tela, temos os botões de ação:

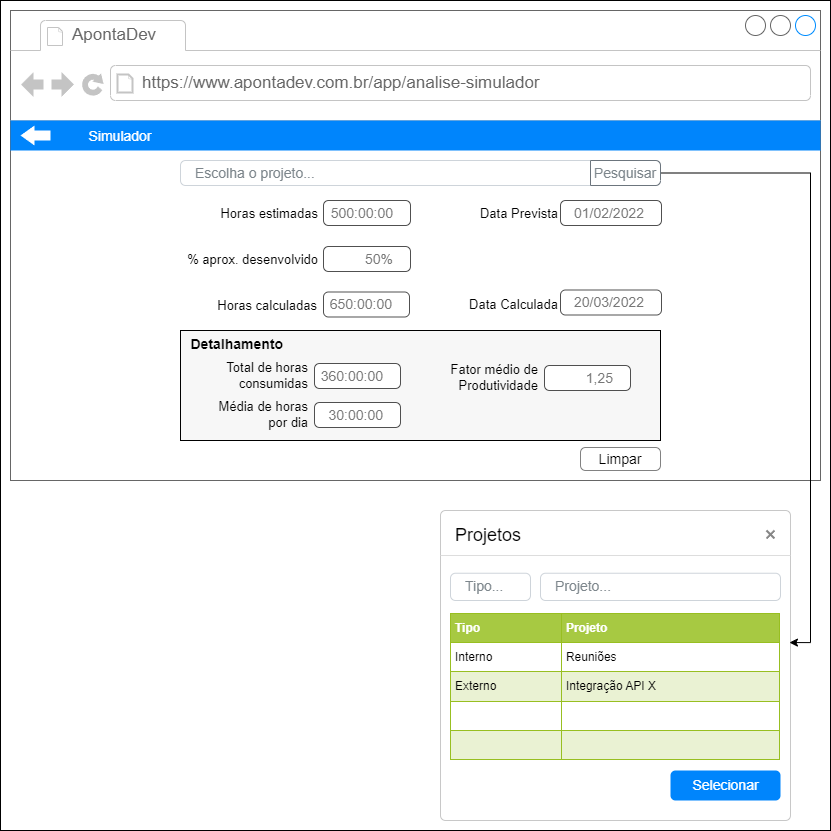
* Adicionar: Usado para criar um novo apontamento, ou caso existam linhas selecionadas na grid, este botão atualizará as linhas selecionadas com as informações dos componentes
* Limpa: Usado para limpar ou voltar para o valor padrão os componentes da tela (exceto a grid)



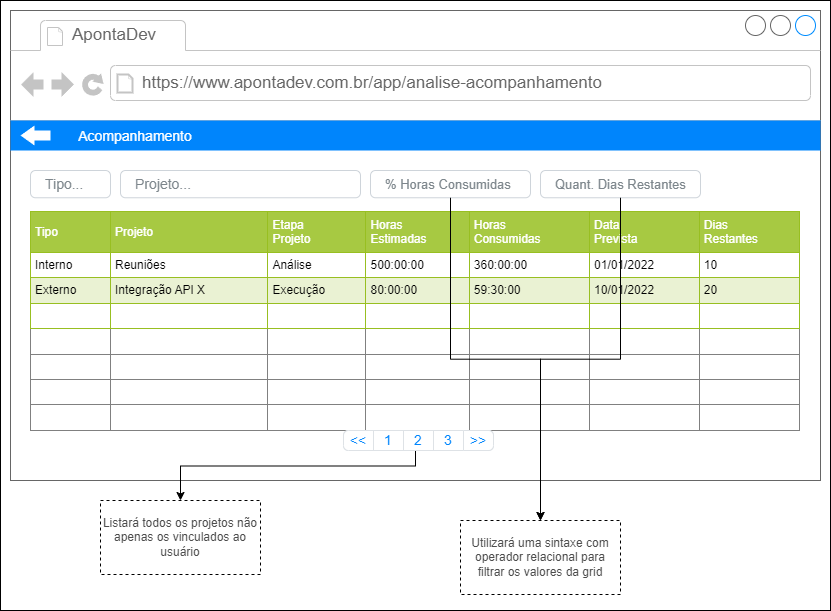
### Telas de Análise

Ao acessar o menu Análise e submenu Simulador, o usuário verá a tela que permite realizar simulações em projetos existentes na aplicação.

Neste tela o usuário deve escolher o projeto que deseja simular, os dados principais do projeto serão exibidos e após o usuário informar um percentual aproximado do desenvolvimento concluído do projeto a projeção será exibida nos campos Horas calculadas e Data calculada.

No grupamento Detalhamento serão exibidos os valores considerados no cálculo para que o usuário possa confirmar e entender a previsão final.

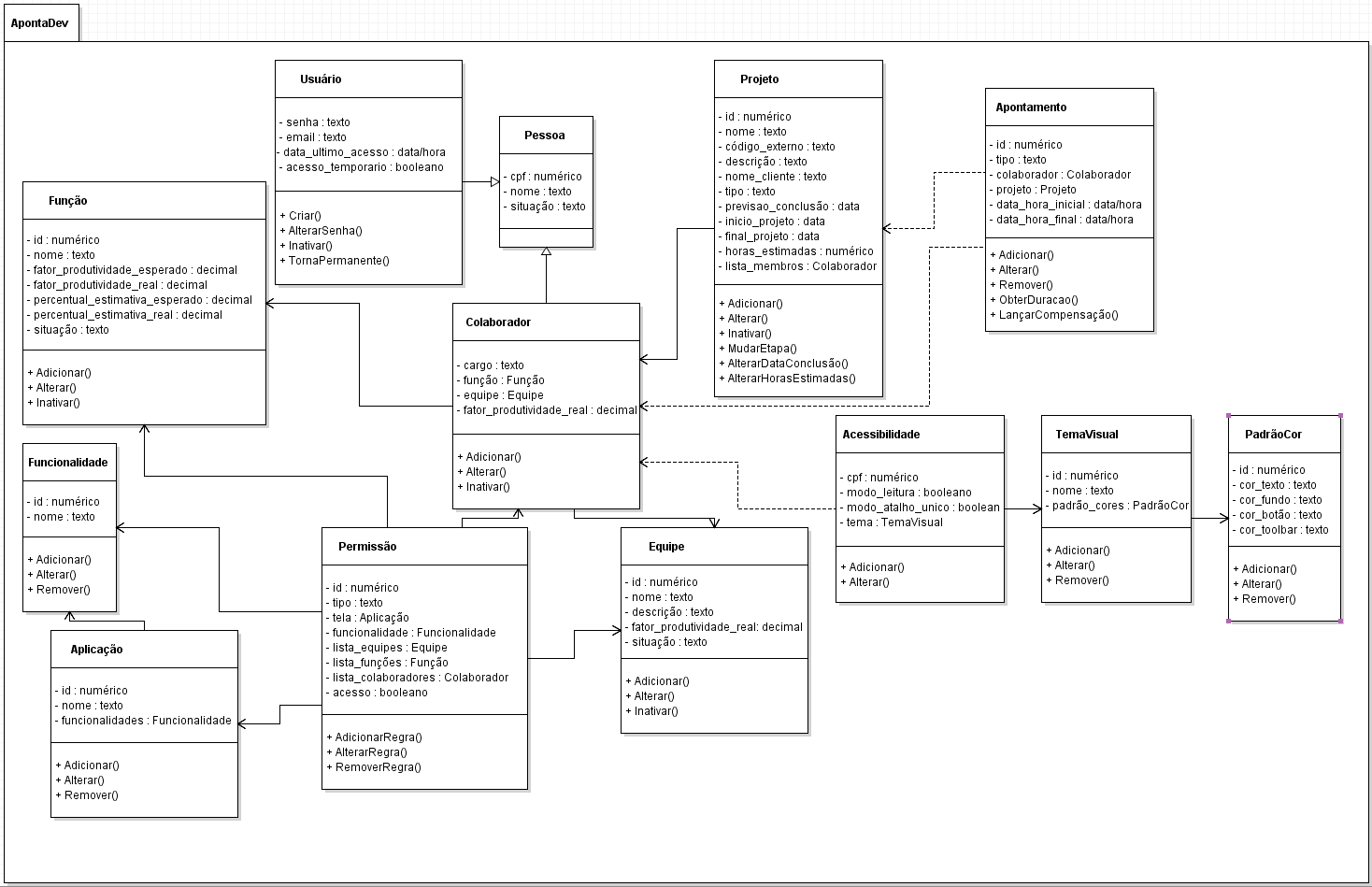
Ao acessar o menu Análise e submenu Acompanhamento, o usuário verá a tela que permite realizar simulações em projetos existentes na aplicação.

O usuário poderá utilizar os quatro componentes da parte superior da tela para filtrar os projetos da grid. E na grid serão listados todos os projetos vinculados ao usuário para acompanhamento das principais informações do projeto.

## 2.5. Diagrama de classes

Segue o diagrama de classes desenhado para o desenvolvimento deste trabalho. Nele é possível ver as classes e o relacionamento entre elas. Abaixo está o detalhamento das classes e como se relacionam:

* Pessoa: É uma classe abstrata que traz os dados básicos que uma pessoa deve ter na aplicação
  + Relacionamentos: Essa classe é herdada pelas classes usuário e colaborador, pois ambas as classes compartilham as mesmas informações básicas



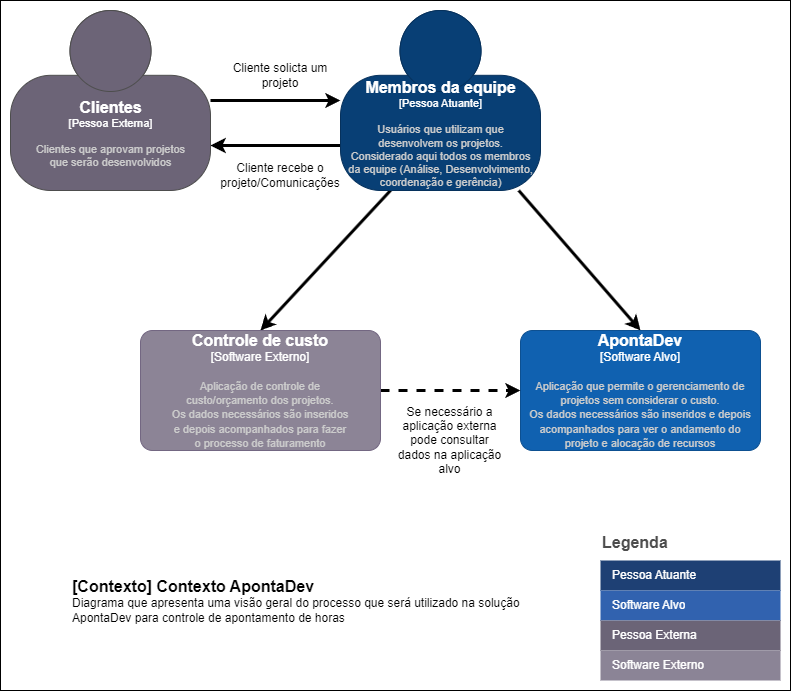
* Usuário: É a classe que apoia no gerenciamento de usuários na aplicação, sendo o termo “usuário” atribuído a todo indivíduo que acessa a aplicação pela tela de acesso
  + Relacionamentos: Usuário herda os dados da classe pessoa
* Colaborador: É a classe que apoia no gerenciamento de colaboradores na aplicação, sendo o termo “Colaborador” atribuído a todo usuário que teve acesso autorizado e atua como integrante dos projetos
  + Relacionamentos: Colaboradores podem ser vinculados a funções, equipe e projetos, colaboradores possuem configurações de permissão e acessibilidade, colaboradores podem fazer apontamentos e herdam os dados da classe pessoa
* Função: É a classe que apoia o gerenciamento de funções na aplicação
  + Relacionamentos: Funções são atribuídas aos colaboradores e tem permissões definidas para elas
* Equipe: É a classe que apoia o gerenciamento de equipes na aplicação
  + Relacionamentos: Colaboradores são designados para equipes, e tem permissões definidas para elas
* Permissão: É a classe que apoia o gerenciamento de permissões na aplicação
  + Relacionamentos: Permissões são atribuídas por aplicação e funcionalidade para colaboradores, funções e/ou equipes
* Aplicação: É a classe que apoia no gerenciamento das aplicações, ela não atuará no cadastramento de aplicações, mas irá recuperar a lista de aplicações da base de dados e utilizará na tela de permissões
  + Relacionamentos: Permissões são atribuídas por aplicação e aplicações possuem um conjunto de funcionalidades
* Funcionalidade: É a classe que apoia no gerenciamento das funcionalidades por aplicação, ela não atuará no cadastramento de funcionalidades, mas irá recuperar a lista da base de dados e utilizará na tela de permissões
  + Relacionamentos: Permissões são atribuídas por funcionalidade e elas pertencem a uma aplicação
* Projeto: É a classe que apoia no gerenciamento de projetos na aplicação
  + Relacionamentos: Projetos tem colaboradores vinculados e estão associados a apontamentos
* Apontamento: É a classe que apoia no gerenciamento de apontamentos na aplicação
  + Relacionamentos: Apontamentos são realizados por um colaborador e estão vinculados a um projeto
* Acessibilidade: É a classe que apoia no gerenciamento das parametrizações de acessibilidade na aplicação
  + Relacionamentos: As configurações de acessibilidade estão ligadas aos colaboradores e utilizam temas visuais predefinidos
* Tema Visual: É a classe que apoia no gerenciamento dos temas que podem ser utilizados na aplicação, ela não atuará no cadastramento de temas, mas irá recuperar a lista da base de dados e utilizará na tela de acessibilidade
  + Relacionamentos: Tema visual é utilizado na tela de acessibilidade e é construído a parte de padrão de cores predefinidos
* Padrão Cor: É a classe que apoia no gerenciamento das cores que podem ser utilizadas na aplicação, ela não atuará no cadastramento de cores, mas irá recuperar a lista da base de dados e utilizará para construção dos temas
  + Relacionamentos: Os padrões de cores são utilizados pelos temas visuais

# 3. Projeto

## 3.1. Arquitetura de *software*

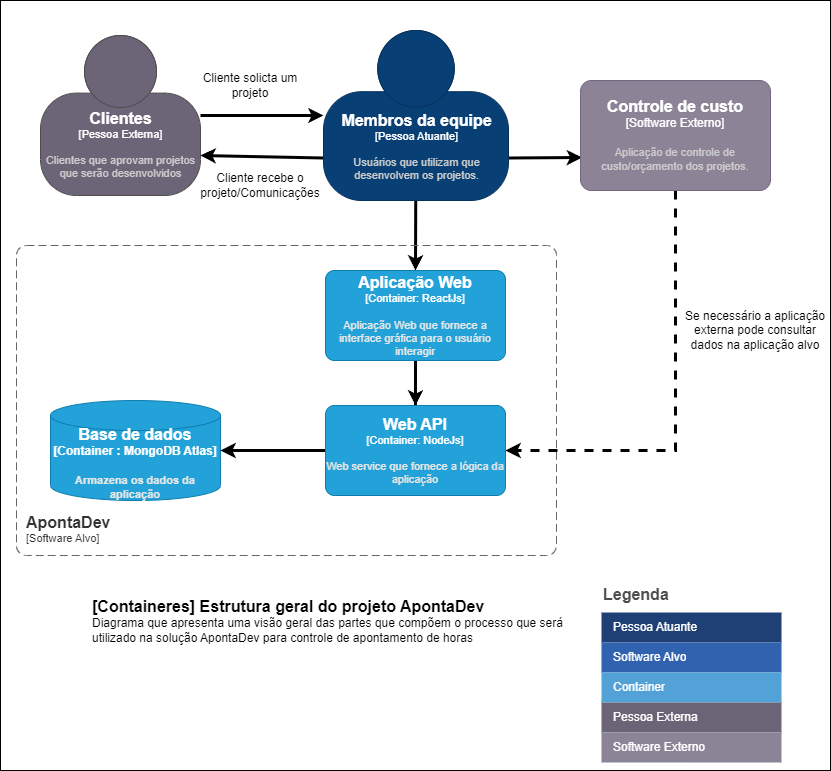
A solução foi idealizada para funcionar na arquitetura cliente-servidor. Essa arquitetura foi escolhida por ser amplamente utilizada e um padrão consolidado na internet, sendo que as características dessa arquitetura favorecem sua utilização na web quando pensamos em aplicações usando o modelo Web Service API no servidor e aplicação web no navegador do cliente criando uma única camada de acesso aos dados e possibilitando a extensão das funcionalidades para aplicações de dispositivos móveis e acesso direto aos dados via Web Service API.

Abaixo temos o diagrama de modelo C4 com visibilidade do contexto da solução, observe que a aplicação será acessada diretamente pelo usuário, e também permite acesso de outras aplicações para automatizações de consulta e alteração de dados via Web Service API.

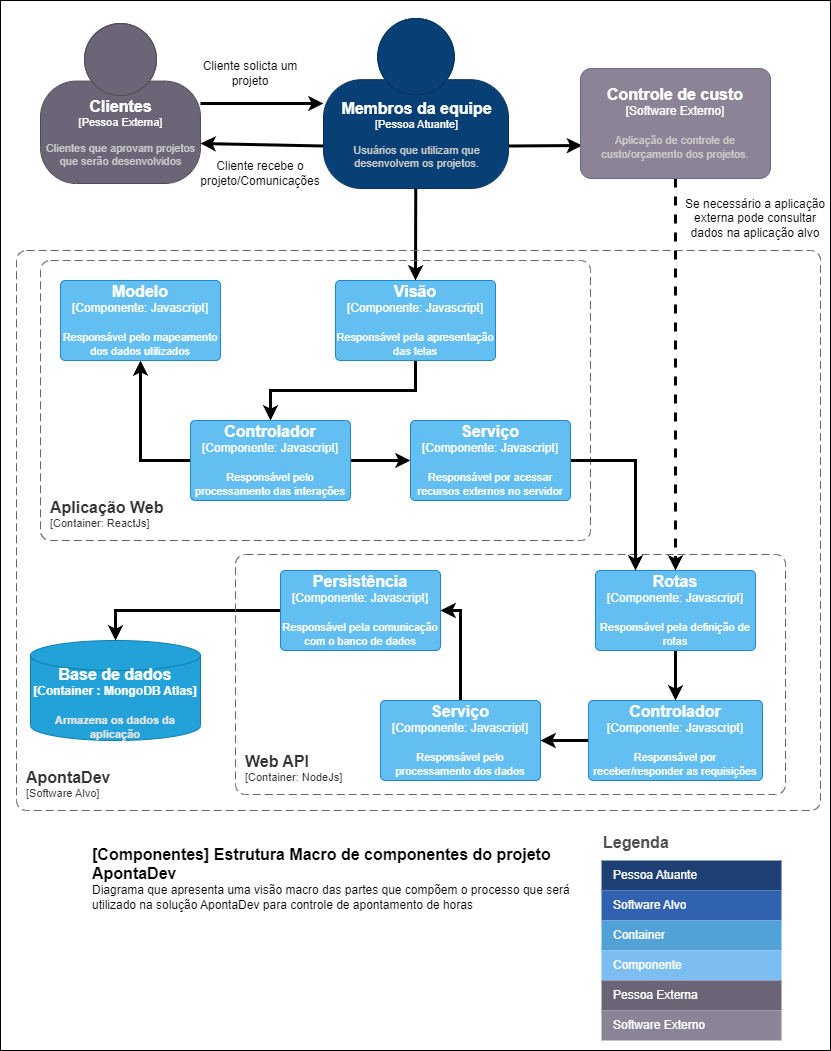


Abaixo temos o diagrama de modelo C4 com visibilidade de contêineres da solução, observe que nesta visibilidade fica evidente a arquitetura cliente-servidor, com o contêiner “Aplicação Web” rodando no navegador do cliente e o contêiner “Web API” rodando no servidor, além desses contêineres temos o “Banco de Dados” que poderia ser executado no servidor, mas neste projeto usaremos um serviço de armazenamento de dados remoto.

Note que o contêiner “Web API” atua como uma camada acima da base de dados, sendo ele o responsável por acessar os dados e repassá-los, processados ou não, para as aplicações requisitantes. Um fator importante neste modelo é que a base de dados não tem acoplamento com as aplicações requisitantes, por isso a base de dados fica "invisível" para as demais aplicações tornando mudanças neste contêiner mais simples de serem feitas.

Abaixo temos o diagrama de modelo C4 com visibilidade de componentes da solução. Note que nesta visão do projeto é possível perceber que a Aplicação Web e a Web API estão usando o padrão de arquitetura de projetos multicamadas baseado no modelo MVC.

Neste modelo a aplicação é dividida em camadas com responsabilidade bem definidas e isso auxilia na organização do projeto, na manutenção do código-fonte e construção da lógica que deve ser implementada.



### Definição tecnológica

Segue a lista de recursos tecnológicos (plataformas, frameworks, linguagens) que serão utilizados para construção deste projeto:

Front-end

* HTML
* CSS
* JavaScript
* ReactJs

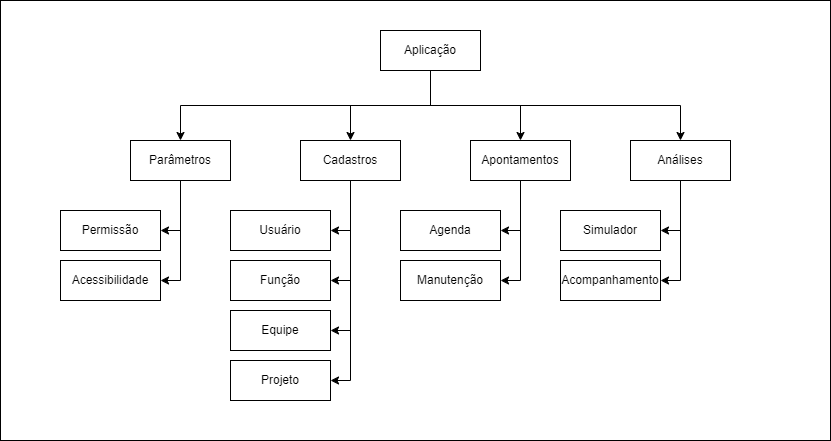
Back-end

* JavaScript
* NodeJs
* ExpressJs
* padrão RESTful

Persistência de dados

* NoSQL - MongoDB
* MongoDB Atlas

## 3.2. Arquitetura da informação

A aplicação apresenta as telas divididas por menus que as agrupam por características semelhantes, como pode ser visto no diagrama hierárquico abaixo:

No menu “Parâmetros” temos as telas que permitem as definições sobre restrições de acesso e como a aplicação deve se comportar com o usuário.

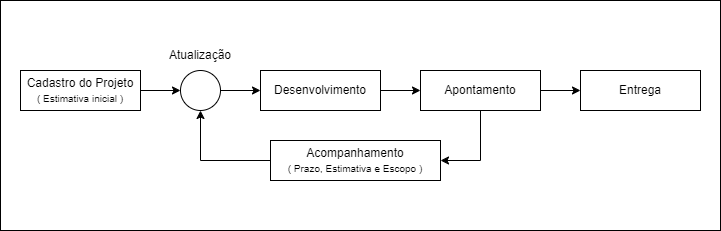
No menu “Cadastros” temos as telas que permitem criar os registros mais básicos da aplicação e que são a base para as demais informações que serão produzidas durante a utilização.

No menu “Apontamentos” temos as telas que permitem a utilização da principal funcionalidade da aplicação, que é o registro de apontamentos em projetos.

Por fim, no menu “Análises” temos as telas que permitem o acompanhamento dos dados produzidos pelos apontamentos para os projetos, equipes, funções e usuários, agregando valor à análise e permitindo a previsão de possíveis necessidades de recursos no projeto.

No fluxo principal de utilização as informações da aplicação seguem a seguinte ordem, e observe que os dados de entrada de apontamentos atuam como uma retroalimentação permitindo a comparação entre o estimado inicialmente e o que está sendo executado ao longo do processo:

1. As parametrizações são utilizadas pelo usuário para definir Se e Como os dados e recursos são disponibilizados.
2. Os dados de cadastro são utilizados para contextualizar o usuário em um ou mais projetos, dentro de uma equipe, trabalhando em uma determinada função.
3. O usuário alimenta os dados de apontamento registrando suas atividades.
4. Os dados de apontamento de todos os usuários criam uma base de dados que nas telas de análise mostram como o projeto caminhou, e permite fazer projeções de tempo e esforço.



# 4. Testes

Para este projeto foram utilizados exploratórios/heurísticos e de integração feitos sem a utilização de automatização de rotinas de código. Abaixo está descrito o plano macro de testes que oferece uma base para os testes na aplicação e na API.

## 

## Plano de teste

Testes na API Web:

* Conexão com serviço web de banco de dados MongoDB Atlas
* Falha na autenticação de um usuário
* Sucesso na autenticação de um usuário retornando o token
* Consumo de endPoints públicos utilizando o token de acesso (GET)
* Consumo de endPoints públicos utilizando o token de acesso para alterações dos dados (POST, PUT, DELETE)
* Consumo de enPoints de desenvolvimento usando token para todos os verbos HTTP

Testes na Aplicação Web:

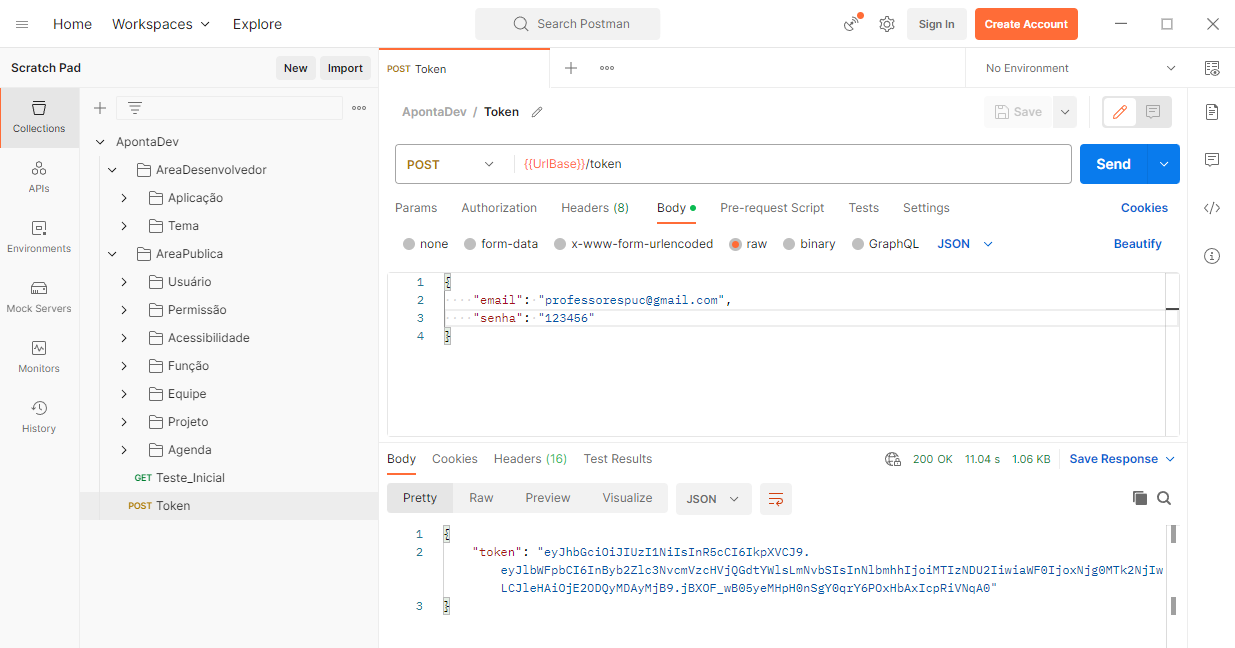
* Redirecionamento de rota para a tela de login quando não houve dados do usuário logado
* Tela de login
  + Acesso com usuário e senha
  + Alteração de senha
* Validação da política de permissão nos menus e funcionalidades das telas
* Validação das funcionalidades de CRUD das telas (as funcionalidades que existirem)
* Aplicação de tema de cores nas telas

## Evidências de testes realizados

Os testes da API foram feitos utilizando a ferramenta Postman. As requisições foram criadas e para apoiar na utilização foi feito uso do recurso de variáveis existente na ferramenta. Com esse recurso foram criadas duas variáveis (URLBase e Token) dentro da pasta “ApontaDev” na aba variables e elas foram utilizadas nas requisições centralizando essas informações e permitindo mudar em apenas um lugar.

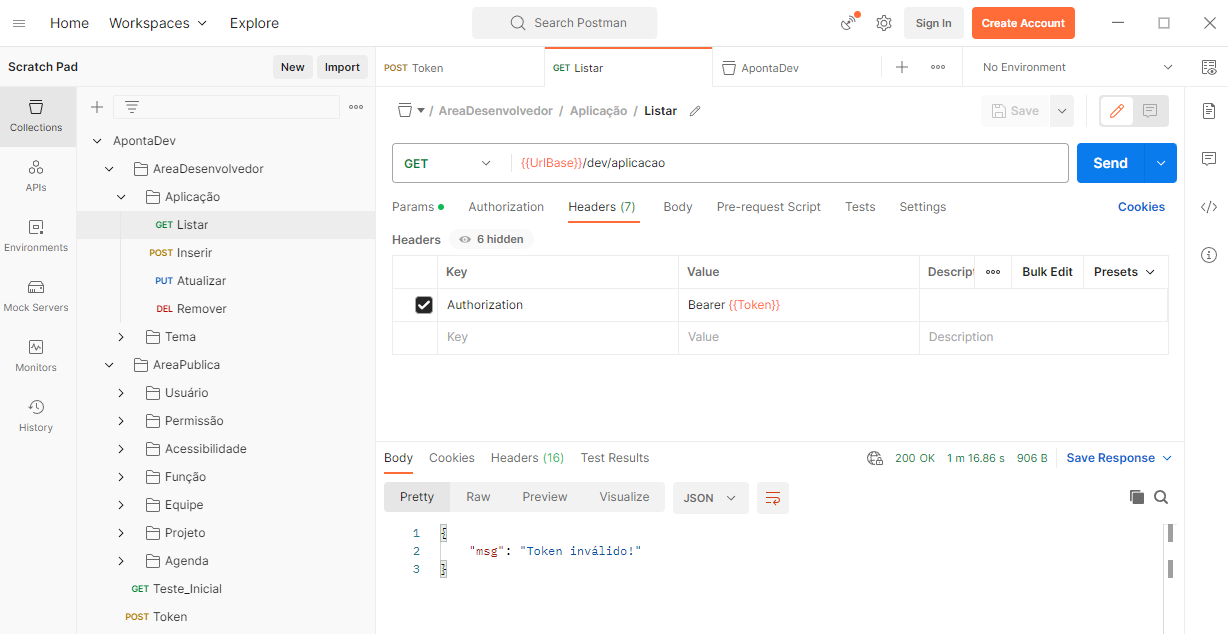
Obtendo o token pela API

Essa requisição não possui Params ou a necessidade de headers adicionais, apenas o body conforme imagem abaixo com dados de um usuário válido.



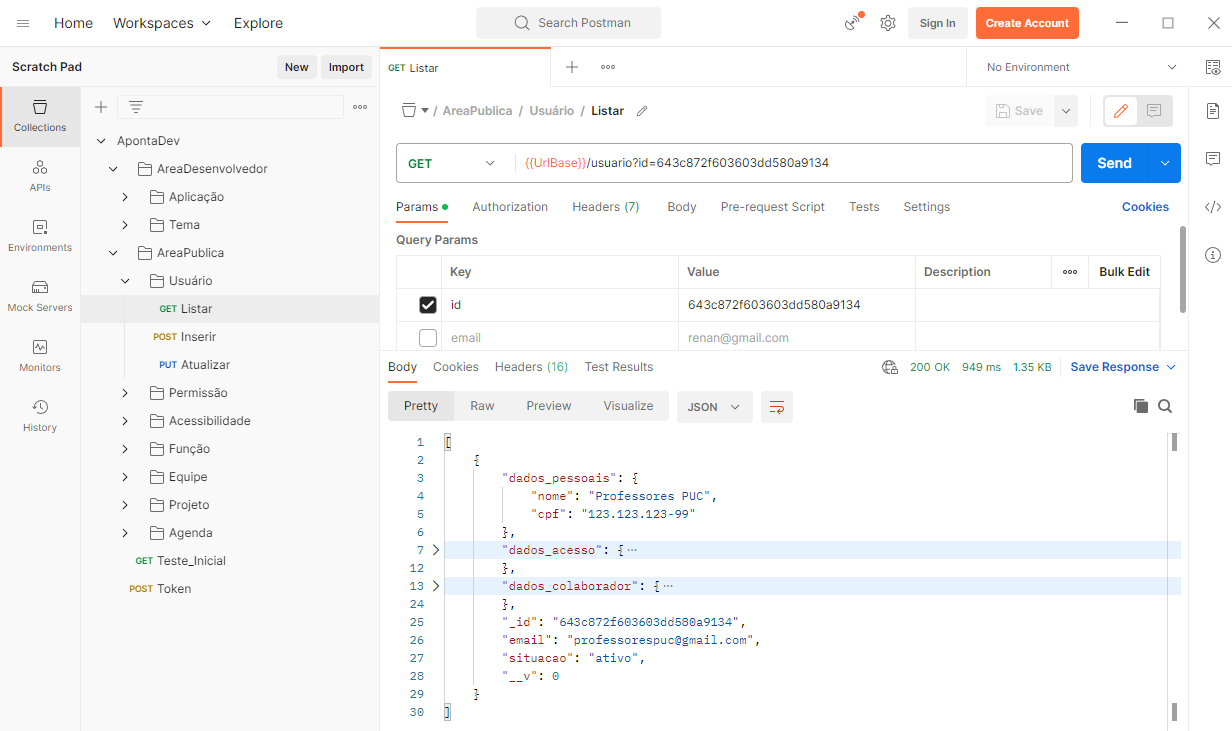
Obtendo uma resposta de token inválido

Para esse teste não foi necessário informar filtros para a consulta (Params / Query Parameters) ou dados no corpo da requisição (body), apenas o Header destacado na imagem com um token inválido.



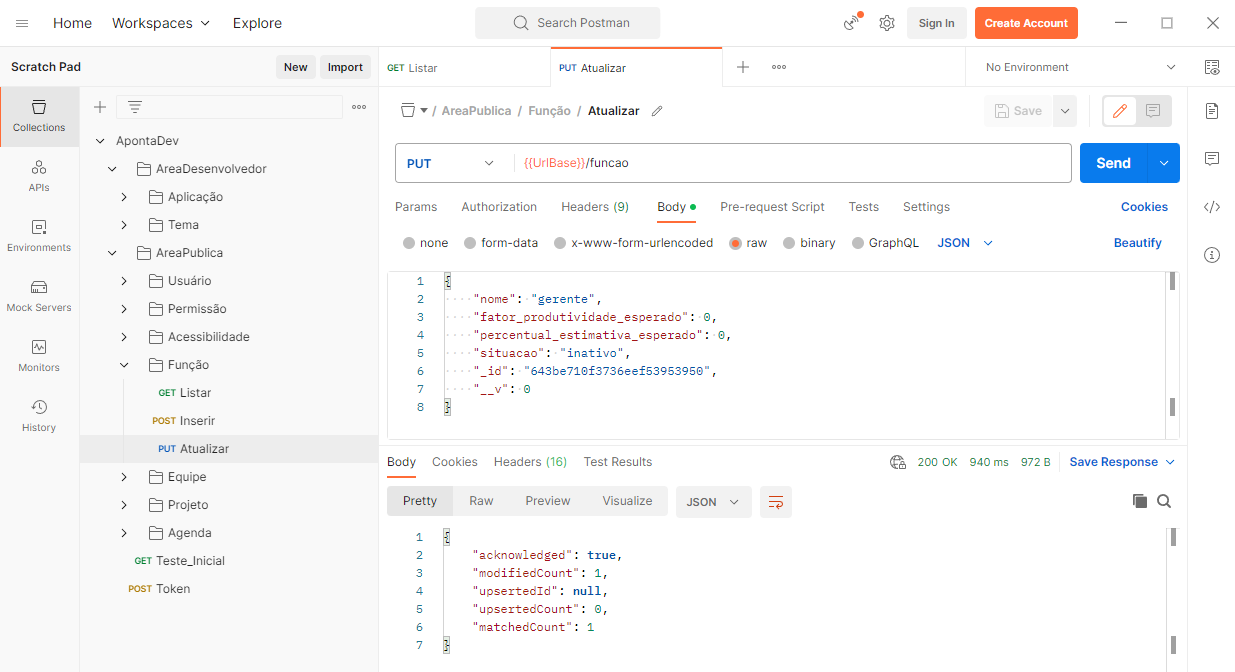
Consultando dados de usuário utilizando filtros na requisição

Neste teste na aba Headers deve existir a chave (Key) "Authorization” com o valor (Value) “Bearer” e o token obtido previamente.

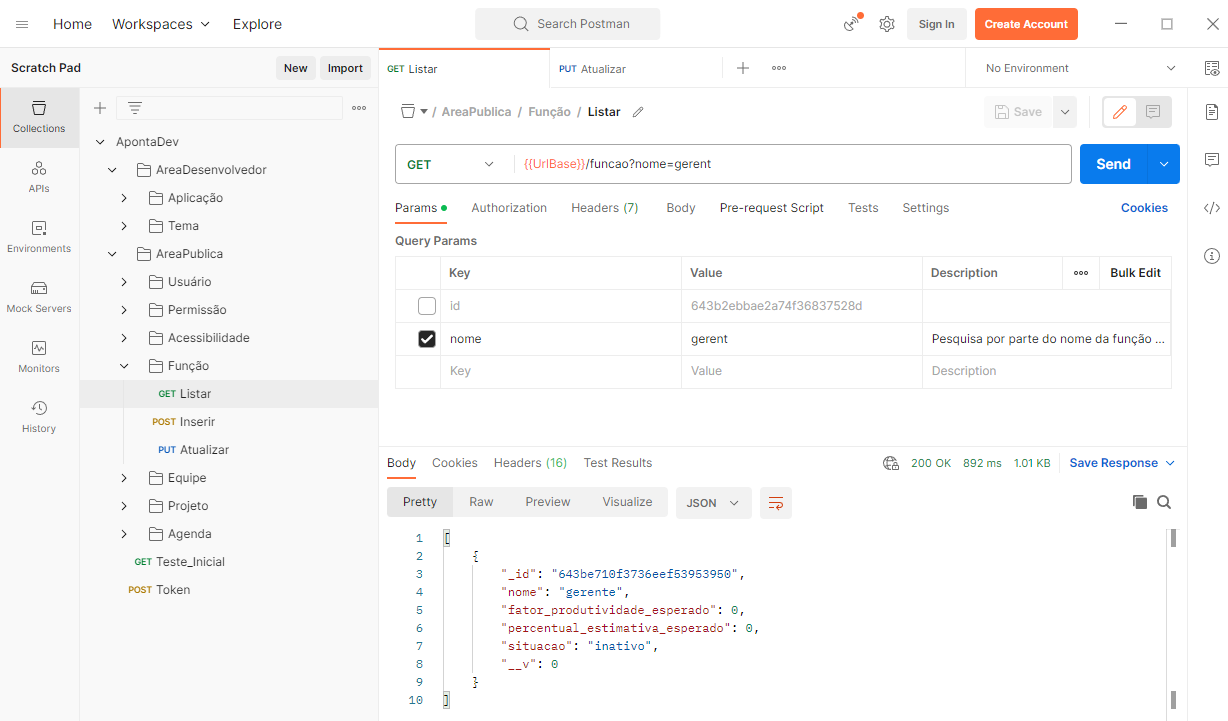


Alterando a etapa de um projeto via API

Neste teste faremos a inativação da função gerente, para isso colocamos os dados no body da requisição e no header a chave “Authorization” com o token. O retorno contém o número de registros alterados.



Para confirmar se a alteração ocorreu, é possível usar o endPoint de consulta filtrando a função que queremos ver.

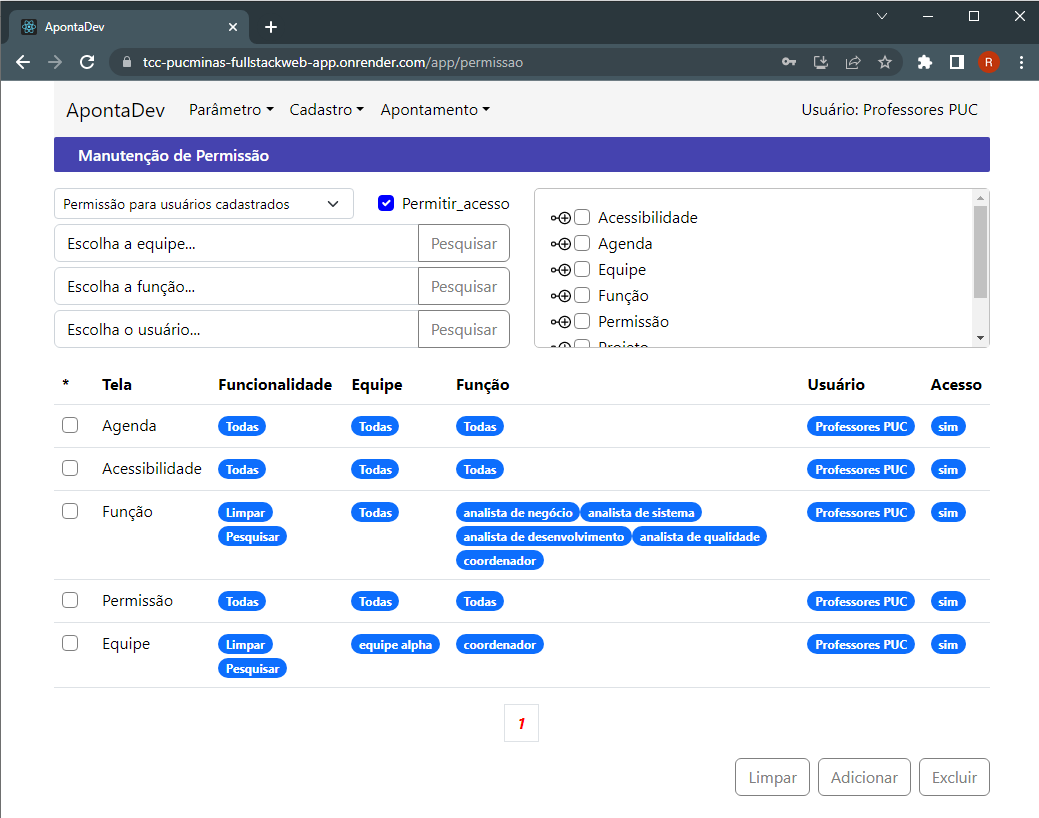


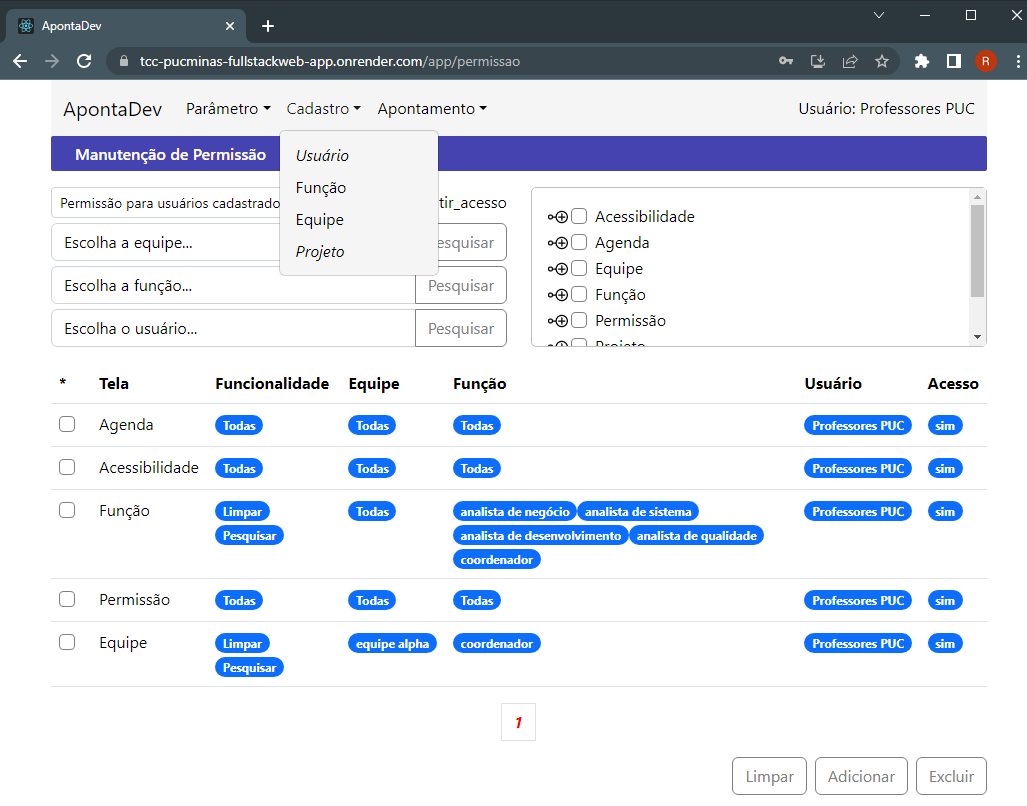
Teste da política de permissão

Para esse teste o usuário “Professores PUC” recebeu permissões nas telas de Permissão, Acessibilidade, Função, Equipe e Agenda. As telas Usuário e Projeto ficaram sem permissão.

Nas telas de Função e Equipe as funcionalidades não foram todas liberadas.

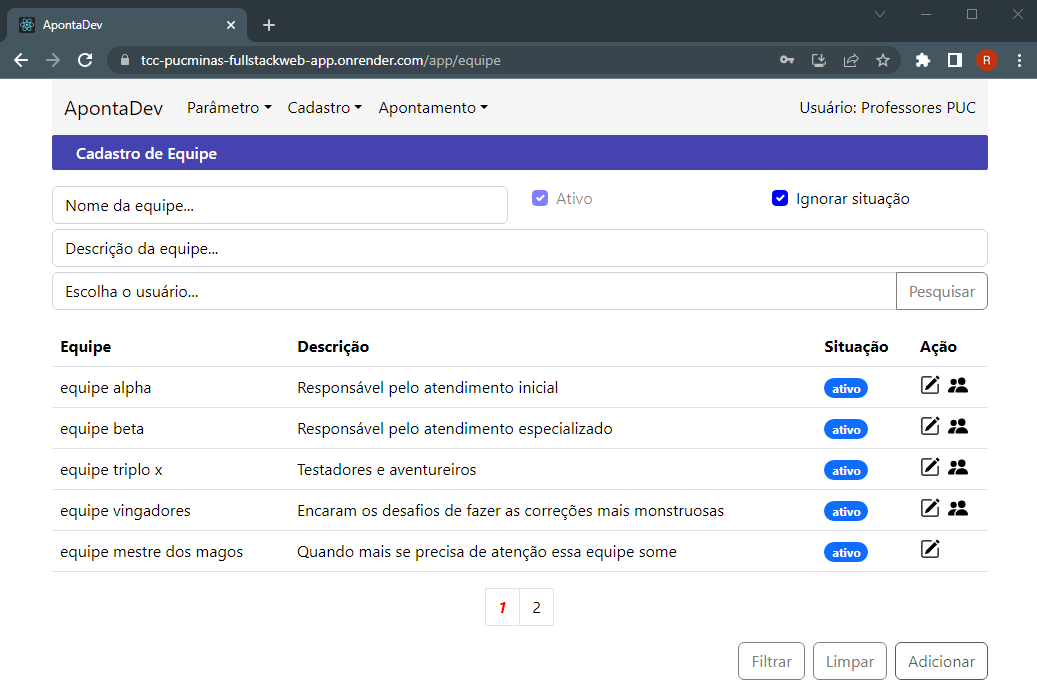
É importante notar que após a alteração das regras de permissão é necessário iniciar uma nova sessão para ter o comportamento esperado. Neste momento do projeto, isso pode ser feito abrindo uma nova aba e acessando novamente ou limpando a aba de Session Storage da aba do navegador.

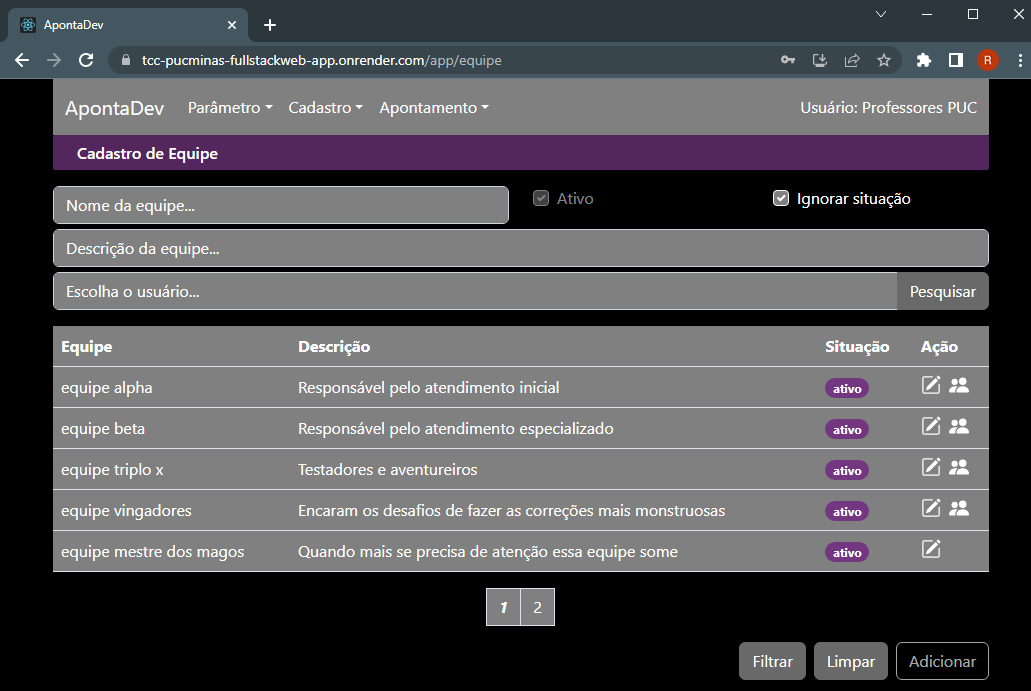




Teste de tema da aplicação

Os temas são “Dia” e “Noite” e neste teste será feita a alteração.





# 5. URLs

## 5.1. Aplicação web

A API e a aplicação web foram disponibilizadas através da plataforma Render (https://render.com) que permite hospedagens gratuitas para pequenos projetos que não precisam de muitos recursos e terão poucos acessos. Esse tipo de serviço viabiliza e apoia significativamente estudantes e projetos em fase de prototipação.

Links das aplicações:

* https://tcc-pucminas-fullstackweb-api.onrender.com
* https://tcc-pucminas-fullstackweb-app.onrender.com

Para apoiar no consumo da API foi disponibilizado no repositório do projeto da API no Github (veja o link no item 5.2 abaixo) o arquivo ApontaDev.postman\_collection.json que contém a coleção de requisições. Esse arquivo deve ser importado na aplicação Postman.

O Postman é uma poderosa ferramenta gratuita que permite criar requisições e consumir APIs web, essa aplicação pode ser encontrada no link https://www.postman.com.

A aplicação web poderá ser acessada com as credenciais do usuário abaixo que foi previamente cadastrado e configurado para essa finalidade:

* Usuário: professorespuc@gmail.com
* Senha: 123456

O mesmo usuário pode ser utilizado para obter o token na API.

## 5.2. Repositório código-fonte

Foi utilizado o software web Github para controle do versionamento remoto por sua grande popularidade, por ser gratuito e por utilizar o software Git como base para o controle entre repositório remoto e repositório local.

Para facilitar no processo de deploy a API e a aplicação web foram colocados em repositórios diferentes. A API pode ser encontrada no repositório do link https://github.com/sarsdev/tcc-desenv-web-fullstack-pucminas-2022-api

O projeto da aplicação web está salvo no repositório do link: https://github.com/sarsdev/tcc-desenv-web-fullstack-pucminas-2022.

## 5.3. Vídeo de apresentação do trabalho

A apresentação está em um diretório no Google Drive e foi compartilhada com o link público abaixo para que possa ser consultada sem a necessidade de liberação de permissão. Nela foi feito um resumo do projeto e uma apresentação geral dos recursos desenvolvidos.

https://drive.google.com/file/d/16-\_xQsNAgEFbMINpkHqGalzJvhQZ8AQu/view?usp=share\_link

**LISTA DE ANEXOS**

Anexo: Levantamento usando Lean Inception

