**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**](#_heading=h.gjdgxs) **3**

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

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

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

[**2. Modelagem**](#_heading=h.2et92p0) **9**

[2.1. Diagrama de casos de uso](#_heading=h.tyjcwt) 9

[2.2. Atores](#_heading=h.3dy6vkm) 9

[2.3. Detalhamento dos casos de uso](#_heading=h.1t3h5sf) 9

[2.4. Projeto de Interface](#_heading=h.4d34og8) 9

[2.5. Diagrama de classes](#_heading=h.2s8eyo1) 10

[**3. Projeto**](#_heading=h.17dp8vu) **10**

[3.1. Arquitetura de software](#_heading=h.3rdcrjn) 10

[3.2. Arquitetura da informação](#_heading=h.26in1rg) 10

[**4. Testes**](#_heading=h.lnxbz9) **10**

[**5. URLs**](#_heading=h.35nkun2) **11**

[5.1. Aplicação web](#_heading=h.1ksv4uv) 11

[5.2. Repositório código-fonte](#_heading=h.44sinio) 11

[5.3. Vídeo de apresentação do trabalho](#_heading=h.2jxsxqh) 11

[**REFERÊNCIAS**](#_heading=h.z337ya) **12**

[**LISTA DE ANEXOS**](#_heading=h.ghspku6kjc3t) **13**

[Anexo: Levantamento usando Lean Inception](#_heading=h.4boehhms6e54) 13

# 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 e que travam os processos.

## 

## 

## 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 fazer anotações 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

*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 indicado isso
  + 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
  + Neste 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 casos de uso oferece uma visão global dos casos de uso e dos atores que dele participam.

## 2.2. Atores

Liste cada um dos atores que participarão dos casos de uso, oferecendo uma descrição sucinta para cada um deles. Os atores são todos aqueles que interagem com o sistema (usuários, outros sistemas, etc.). Agora, como já estamos falando da modelagem do sistema, já podemos incluir os usuários responsáveis pela administração do sistema

## 2.3. Detalhamento dos casos de uso

Nesta seção, cada caso de uso deve ser detalhado. Esse detalhamento deve incluir uma descrição do caso de uso, a lista de atores que participam do caso de uso, as pré e pós-condições e os fluxos de eventos (básico, alternativo, de exceção, sub-fluxos, etc.).

## 2.4. Projeto de Interface

Apresente o fluxo do usuário no sistema e o layout das interfaces, incluindo uma descrição dos campos e comandos em cada uma delas. Como forma de layout, use um wireframes ou as interfaces já implementadas.

Se necessário, use um diagrama de estados para representar o comportamento de uma interface ou fluxo de um processo associado à funcionalidade em questão.

## 2.5. Diagrama de classes

Apresente o diagrama de classes da aplicação web, descrevendo, sucintamente, as classes e as relações entre elas.

# 3. Projeto

## 3.1. Arquitetura de *software*

Descreva o padrão arquitetural que sua aplicação adotará (cliente-servidor, MVC, P2P, etc.), justificando a escolha de cada opção tecnológica.

Apresente diagramas que retratem a arquitetura da solução (recomenda-se o [modelo C4](https://c4model.com/)) com os componentes planejados para a sua aplicação Web, tanto aqueles que serão executados do lado do cliente quanto do lado do servidor e como será a relação entre eles.

## 3.2. Arquitetura da informação

Descreva como a informação estará organizada na aplicação – hierarquias, categorias, rótulos (palavras-chave), etc.

Descreva também como será a navegação pelo espaço de navegação, os mecanismos de busca e de recuperação de informações. Opcionalmente inclua um diagrama hierárquico que mostre como as funcionalidades estão distribuídas pelo sistema.

# 4. Testes

Apresente os planos de testes (testes de software, testes heurísticos ou testes de usabilidade) além dos registros de testes realizados em sua aplicação Web.

# 5. URLs

## 5.1. Aplicação web

Apresente o endereço em que sua aplicação web está hospedada, além de quaisquer orientações e restrições (ex.: senha) para usá-la.

## 5.2. Repositório código-fonte

Inclua o código da sua aplicação web em um repositório e indique a URL. A inclusão desse código servirá como base para garantir a autenticidade dos trabalhos.

## 5.3. Vídeo de apresentação do trabalho

Inclua o link para o vídeo com a apresentação da sua aplicação. Seu vídeo deve ter duração máxima de 5 minutos e deve apresentar de forma sucinta o seu projeto. Concentre-se principalmente na apresentação da sua aplicação, mostrando o funcionamento de cada um dos requisitos solicitados na proposta escolhida por você.

O software OBS Studio é uma boa sugestão de software para gravação de screencast.

# REFERÊNCIAS

Como um projeto de aplicativo não requer revisão bibliográfica, a inclusão das referências não é obrigatória. No entanto, caso você deseje incluir referências relacionadas às tecnologias ou às metodologias que foram usadas no seu trabalho, relacione-as de acordo com o modelo a seguir.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

# 

# LISTA DE ANEXOS

## Anexo: Levantamento usando Lean Inception



