

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

PUC Minas Virtual

Pós-graduação *Lato Sensu* em Engenharia de *Software*

Trabalho de Conclusão de Curso

Clever App

Samanta Gimenez Fluture

São Paulo
Outubro 2021

Trabalho de Conclusão de Curso

Sumário

Trabalho de Conclusão de Curso	3
1. Cronograma de trabalho	5
2. Diagrama de casos de uso	7
3. Requisitos não-funcionais	8
4. Protótipo navegável do sistema	10
5. Diagrama de classes de domínio	12
6. Modelo de componentes	13
6.1. Padrão arquitetural	13
6.2. Diagrama de componentes	14
6.3. Descrição dos componentes	15
7. Diagrama de implantação	16
8. Plano de Testes	17
9. Estimativa de pontos de função	23
10. Informações da implementação	24
11. Referências	25

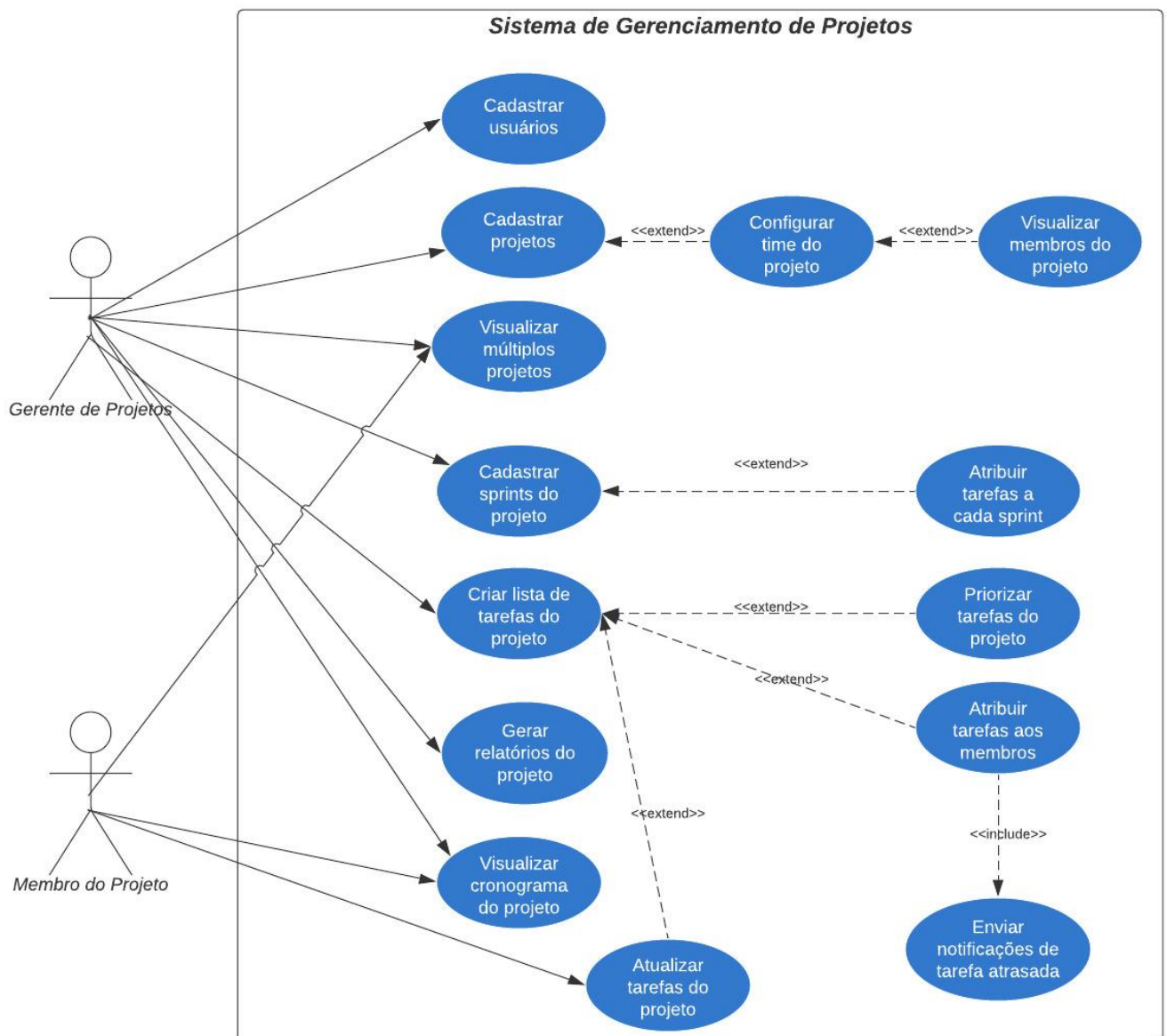
1. Cronograma de trabalho

Datas		Atividade / Tarefa	Produto / Resultado
De	Até		
23/08/2021	27/08/2021	1. Ler materiais de apoio e escolher o sistema/escopo a ser trabalhado no TCC	Escolha do sistema/escopo
30/08/2021	30/08/2021	2. Criar cronograma de trabalho do TCC com tarefas e datas de início e fim	Cronograma de trabalho a ser cumprido
31/08/2021	02/09/2021	3. Criar diagrama de casos de uso do escopo escolhido, contendo atores e usos identificados	Diagrama de casos de uso do sistema
03/09/2021	05/09/2021	4. Listar requisitos não-funcionais previstos para a aplicação, incluindo aqueles importantes para a arquitetura do sistema ou que impactem a solução proposta ponto de vista arquitetural, ou seja, os que terão impacto na definição da proposta da solução	Lista de requisitos não-funcionais
06/09/2021	12/09/2021	5. Criar protótipo navegável do sistema no Figma, incluindo: tela inicial, três casos de uso principais e navegação entre as telas	Protótipo navegável do sistema (no Figma e breve vídeo em AVI mostrando a navegação)
13/09/2021	17/09/2021	6. Criar diagrama de classes de domínio de todo o sistema, incluindo: classes de entidade, seus atributos, métodos e relacionamentos	Diagrama de classes de domínio
20/09/2021	21/09/2021	7. Como parte do modelo de componentes de todo o sistema, escolher o padrão arquitetural para o desenvolvimento da aplicação e das tecnologias que serão utilizadas	Padrão arquitetural escolhido
22/09/2021	24/09/2021	8. Criar diagrama de componentes da aplicação representando todos os componentes, indicando os elementos do padrão arquitetural e as interfaces entre eles	Diagrama de componentes
25/09/2021	26/09/2021	9. Fazer descrição breve dos componentes, indicando: papel dentro do padrão arquitetural escolhido, quais serão reutilizados, quais serão adquiridos, quais precisam ser desenvolvidos	Descrição dos componentes
27/09/2021	28/09/2021	10. Criar o diagrama de implantação de todo o sistema, indicando o mapeamento dos elementos de software da arquitetura para os elementos de hardware onde eles executarão	Diagrama de implantação
29/09/2021	30/09/2021	11. Definir plano de testes, considerando os três casos de uso contemplados no protótipo navegável, sendo dois casos de testes para cada caso de uso	Plano de testes
01/10/2021	03/10/2021	12. Preencher planilha de contagem de pontos de função, considerando todo o	Planilha de estimativa de pontos de função

Clever App

		sistema e colocá-la no repositório do GitHub	
04/10/2021	10/10/2021	13. Codificar os três casos de uso contemplados no protótipo navegável do sistema, tanto front end quanto back end	Repositório no GitHub devidamente configurado com os códigos prontos e funcionando
11/10/2021	15/10/2021	14. Revisar trabalho e deixar o PDF formatado segundo template, pronto para entrega	PDF revisado e formatado, pronto para entrega
17/10/2021	17/10/2021	15. Entregar arquivo no sistema AVA, no formato e prazo estabelecidos	Entrega do arquivo final

2. Diagrama de casos de uso



3. *Requisitos não-funcionais*

1. Segurança

1. a. Comportar diferentes perfis de acesso para controle de segurança, usando login e senha como forma de autenticação

1. a. Agregar conjunto de funcionalidades específicas a cada nível de usuário, sendo dividido em:

- Gerente de Projetos (usuário master, pode criar usuários e projetos),
- Líder do Projeto (responsável pelo projeto e todas as funcionalidades deste),
- Membro do Projeto (parte da equipe do projeto, podendo acessar o projeto no qual faz parte e fazer certas edições e adições)

2. Usabilidade

2. a. Desenhar interface de fácil visualização tanto dos projetos de forma global quanto de suas etapas individuais, contemplando divisão por cores (exemplo: verde, amarelo e vermelho para remeter ao estado atual como um "farol") e formatação visual dos dados (exemplo: linha do tempo, cards "Kanban" e gráficos)

2. b. Incluir na interface um dashboard para servir de resumo e de menu, melhorando a navegabilidade entre as múltiplas funções e projetos

2. c. Oferecer acesso direto e visual aos responsáveis por cada projeto, incluindo fotos, nomes e contatos

3. Portabilidade

3. a. Ser desenvolvido de forma responsiva e interativa, usando tecnologias como HTML, CSS e JavaScript, para que todas as funcionalidades sejam acessadas da mesma forma padrão, tanto na Web quanto em dispositivos móveis

4. Desempenho

- 4. a. Deve suportar diversos acessos e projetos simultâneos, agregando novos times e projetos conforme o crescimento da organização
- 4. b. Não deve ter interface e funcionalidades "pesadas" e que "custosas" em termos de conexão, processamento e memória, pois será o sistema deverá ser acessado múltiplas vezes ao dia, de diversos dispositivos, conectados tanto ao Wifi quanto à rede móvel, dependendo do local em que o membro da equipe estiver

5. Integração Externa

- 5. a. Poder ser integrado com o sistema de calendário utilizado pelos membros das equipes da organização, facilitando o acesso às datas e cronogramas
- 5. b. Poder ser integrado com a plataforma de email utilizada pelos membros das equipes da organização, para recebimento automatizado de notificações

4. Protótipo navegável do sistema

O protótipo navegável do sistema foi criado usando a ferramenta Figma. Abaixo, estão links para acessar o arquivo em: formato original (Figma), em JPG Alta Resolução (GitHub) e em vídeo (Youtube). Todos possuem a navegação entre telas.

- Arquivo Figma: <https://www.figma.com/file/ZtWCeADekem1C4SHjLp8zg/CleverApp-Wireframe?node-id=0%3A1>
- JPG Alta Resolução: <https://github.com/samantafluture/clever-front/blob/main/info/ux-ui-prototype/wireframe-navegation.jpg>
- Vídeo: <https://youtu.be/r9gBmMNxyPY>

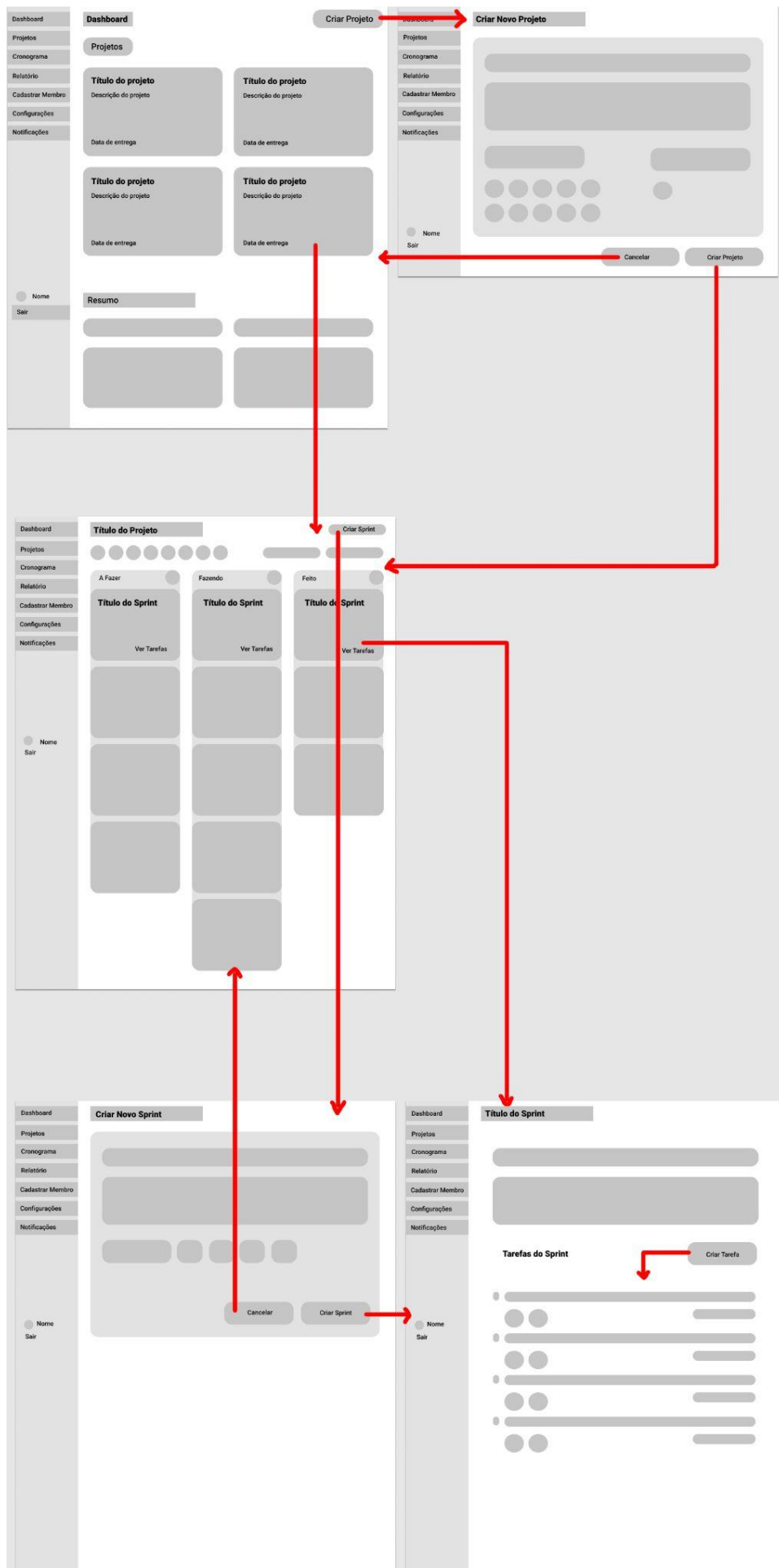
Foi criada a tela principal ("Dashboard") e os três principais casos de uso:

- Cadastrar um novo projeto
- Cadastrar um novo sprint de um projeto existente
- Cadastrar tarefas em um sprint

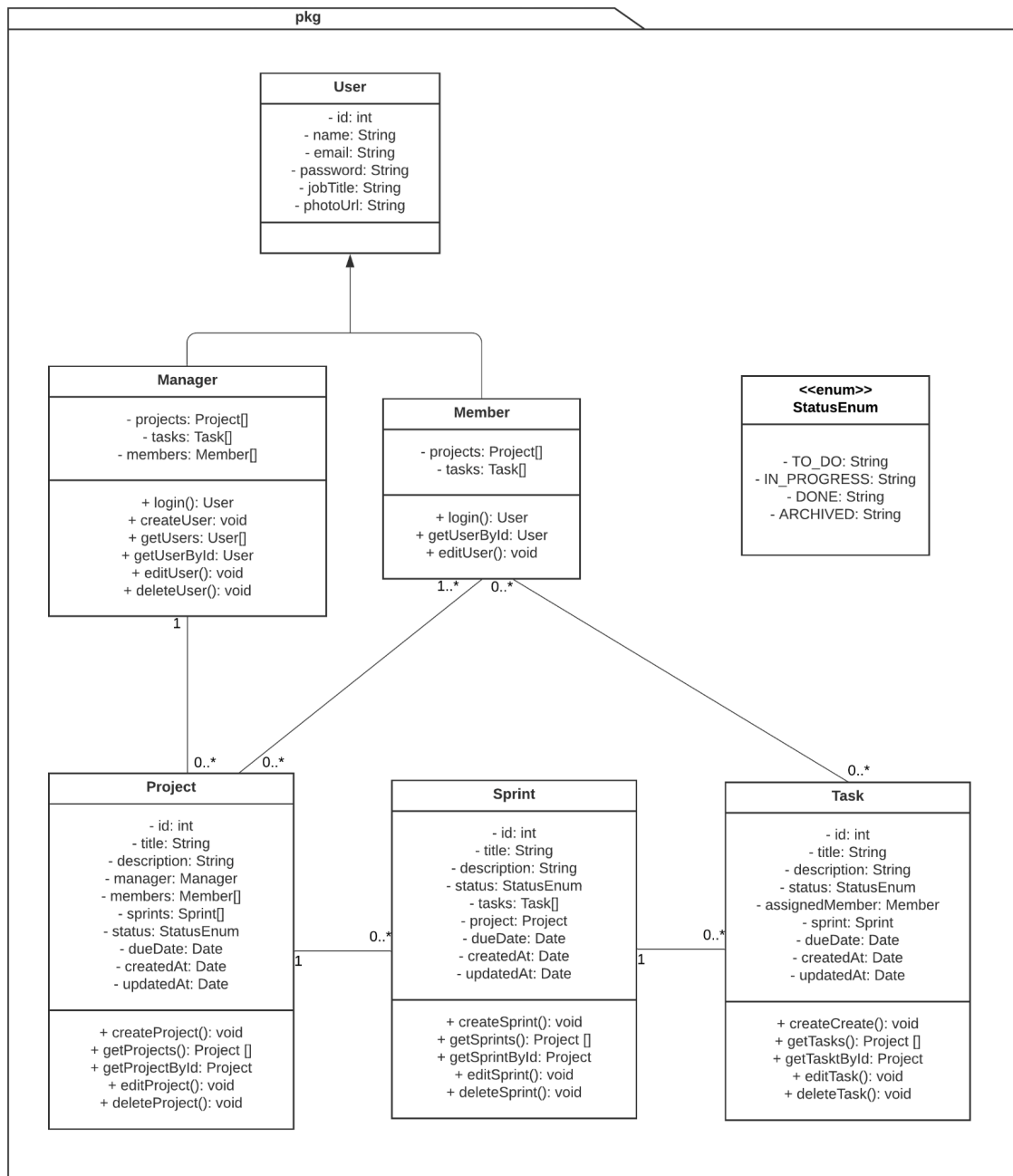
Além disso, é possível também: visualizar múltiplos projetos, visualizar os sprints em um quadro "Kanban" e visualizar uma lista de tarefas por sprint.

Na página a seguir, está uma imagem em menor resolução do protótipo (acessível também via link do GitHub).

Trabalho de Conclusão de Curso – Engenharia de *Software* - PMV



5. Diagrama de classes de domínio



Modelo de componentes

6.1. Padrão arquitetural

Nesta proposta, será usado o Padrão Arquitetural MVC (Model-View-Controller).
Abaixo, a lista das tecnologias que serão utilizadas na implementação.

Camada View:

- Angular, Angular Material CDK e jspdf

Camada Model e Controller:

- API construída com NestJS, JWT, Logger e Mailer
- Integração com base de dados usando TypeORM

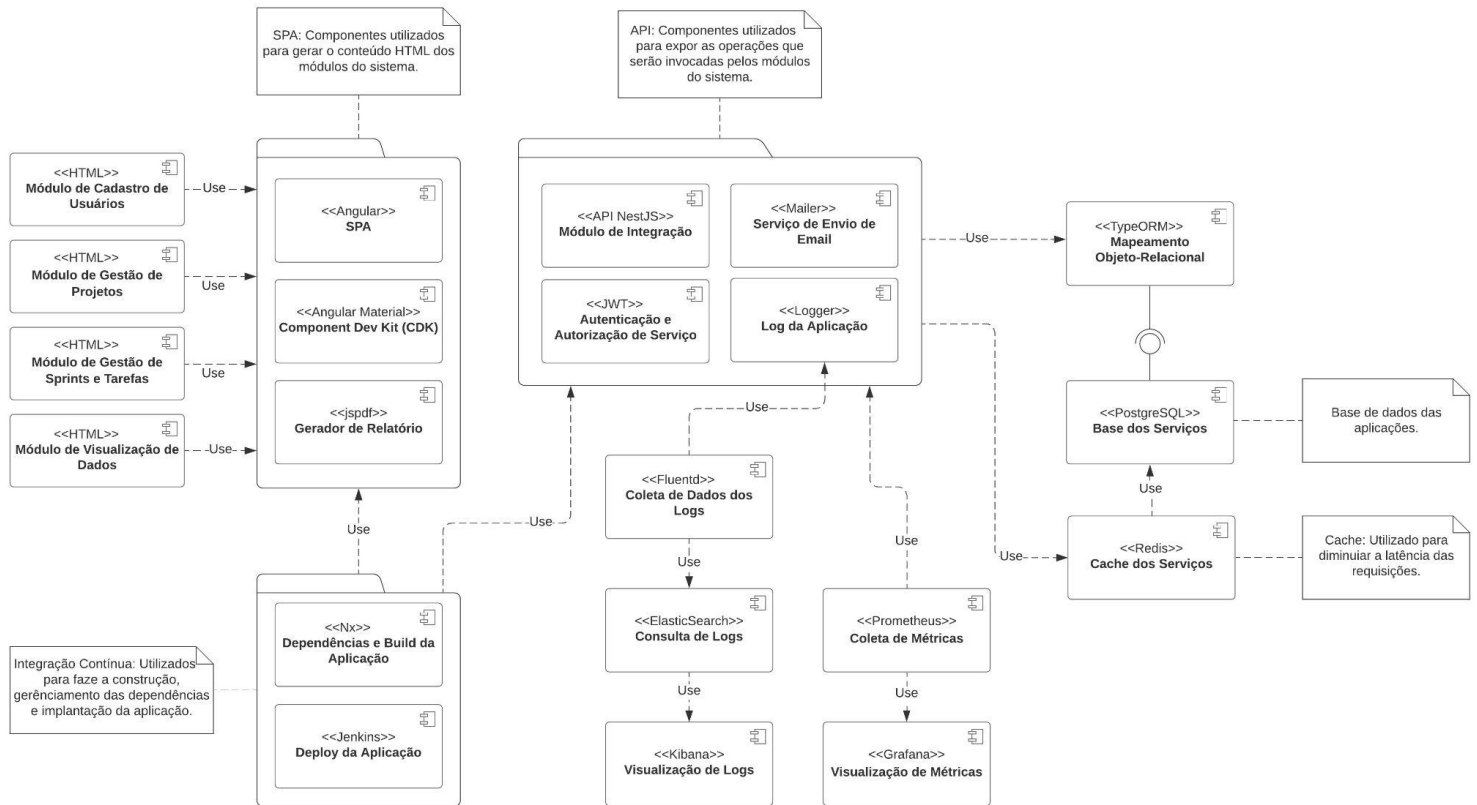
Bancos de Dados:

- PostgreSQL e Redis (para cache)

Tecnologias para build, deploy e integração contínua:

- Nx e Jenkins (gerenciamento de dependências, build e deploy)
- Elasticsearch, Fluentd e Kibana (logs da aplicação)
- Prometheus e Grafana (monitoramento da aplicação)

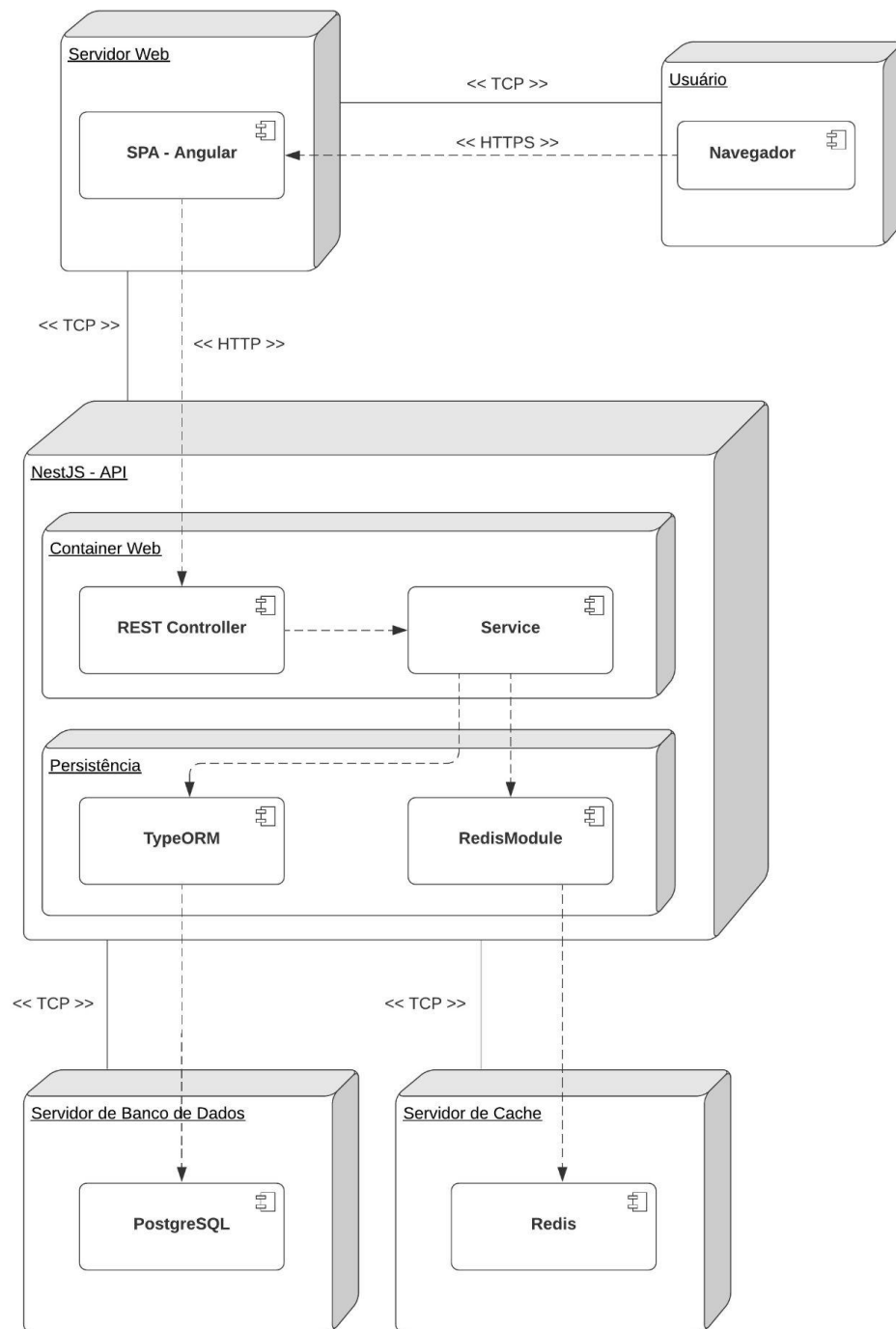
6.2. Diagrama de componentes



6.3. Descrição dos componentes

Número	Componente	Descrição
1	Angular	Componente da camada de apresentação (view), responsável por gerar a interface de acesso web responsiva para o usuário (tanto no desktop, quanto em dispositivos móveis)
2	Angular Material CDK	Componente da camada de apresentação, responsável por fazer a gestão de comportamento e estado dos principais componentes de interface
3	jspdf	Componente da camada de apresentação responsável por gerar relatórios em PDF
4	NestJS, JWT, Mailer e Logger	Componente API RESTful da camada de modelo e controller utilizado para expor as operações que serão invocadas pelos módulos do sistema. JWT é um serviço proprietário da Auth e precisa ser adquirido.
5	TypeORM	Componente ORM (mapeamento objeto-relacional) responsável por fazer a integração com a base de dados, expondo serviços
6	PostgreSQL	Base de dados da aplicação
7	Redis	Recurso utilizado para salvar dados temporários e diminuir a quantidade e latência de requisições. Este serviço é proprietário e precisa ser adquirido.
8	Elasticsearch, Fluentd e Kibana (EFK)	Componentes responsáveis pela busca e visualização dos dados de logs da API. Estes serviços são proprietários e precisam ser adquiridos.
9	Prometheus e Grafana	Componentes responsáveis pela coleta e visualização de métricas do sistema. Grafana é um serviço proprietário e precisa ser adquirido.
10	Integração Contínua	Componentes utilizados para construir, gerenciar dependências e implantar os módulos dos sistemas

6. Diagrama de implantação



Plano de Testes

Nº	Caso de uso	Objetivo do caso de teste	Entradas	Resultados esperados
1	Cadastrar Novo Projeto	Validar o cadastro com sucesso de um novo projeto	<p>- Logar no sistema com um usuário de perfil Gerente - Acessar a tela de listagem de projetos, 'Projetos', pelo menu</p> <p>- Clicar no botão "Criar Projeto"</p> <p>- Informar os seguintes dados nos respectivos campos: Título: Nova Versão do Aplicativo Móvel para dispositivos iOS Descrição: Refatorar para compatibilidade e adicionar novos recursos solicitados Prazo de entrega: 19/12/2021 Membros: João Silva, Roger Matos, Helena Rodrigues, Marta Marques, Paulo Ferraz, Felipe Soares Sou Responsável por este projeto: Sim</p> <p>- Clicar em "Criar Projeto"</p>	<p>- O sistema apresenta uma mensagem de sucesso informando que o projeto foi criado com sucesso</p> <p>- O sistema retorna para a tela de listagem de projetos</p> <p>- A listagem de projetos aparece atualizada com o novo projeto criado</p>

2	Cadastrar Novo Projeto	Verificar que não é possível inserir no sistema um projeto sem data de entrega	<ul style="list-style-type: none"> - Realizar o processo de criação de um projeto no sistema, informado no caso de teste número 1 - Cadastrar um projeto sem data de entrega - Informar os seguintes dados nos respectivos campos: Título: Nova Versão do Aplicativo Móvel para dispositivos iOS Descrição: Refatorar para compatibilidade e adicionar novos recursos solicitados Prazo de entrega: vazio Membros: João Silva, Roger Matos, Helena Rodrigues, Marta Marques, Paulo Ferraz, Felipe Soares Sou Responsável por este projeto: Sim - Clicar em "Criar Projeto" 	<ul style="list-style-type: none"> - O sistema emite uma mensagem de erro informando que 'data de entrega' é um campo obrigatório e deve ser preenchido - O sistema permanece na tela de cadastro de projeto
---	------------------------	--	--	--

3	Cadastrar Novo Sprint de um Projeto Existente	Validar o cadastro com sucesso de um novo sprint de um projeto existente	<ul style="list-style-type: none"> - Logar no sistema com um usuário de perfil Gerente - Acessar a tela de listagem de projetos, 'Projetos', pelo menu - Selecionar um projeto específico e entrar em sua página de visualização - Clicar no botão "Criar Sprint" - Informar os seguintes dados nos respectivos campos: Título: Recriar tela de splash do aplicativo Descrição: Nova tela com novo design e novas dimensões Prazo de entrega: 01/11/2021 Status: A Fazer Projeto: Nova Versão do Aplicativo Móvel para dispositivos iOS - Clicar em "Criar Sprint" 	<ul style="list-style-type: none"> - O sistema apresenta uma mensagem de sucesso informando que o sprint foi criado com sucesso - O sistema retorna para a tela do projeto em questão, com a lista de seus sprints - A listagem de sprints aparece atualizada com o novo sprint criado, no local de acordo com seu status
---	---	--	--	--

4	Cadastrar Novo Sprint de um Projeto Existente	Verificar que não é possível inserir no sistema um sprint sem data de entrega	<ul style="list-style-type: none"> - Realizar o processo de criação de um sprint no sistema, informado no caso de teste número 1 - Cadastrar um sprint sem data de entrega - Informar os seguintes dados nos respectivos campos: Título: Recriar tela de splash do aplicativo Descrição: Nova tela com novo design e novas dimensões Prazo de entrega: vazio Status: A Fazer Projeto: Nova Versão do Aplicativo Móvel para dispositivos iOS - Clicar em "Criar Sprint" 	<ul style="list-style-type: none"> - O sistema emite uma mensagem de erro informando que 'data de entrega' é um campo obrigatório e deve ser preenchido - O sistema permanece na tela de cadastro de sprint
---	---	---	--	---

5	Cadastrar Nova Tarefa de um Sprint Existente	Validar o cadastro com sucesso de uma nova tarefa de um sprint existente	<ul style="list-style-type: none"> - Logar no sistema com um usuário de perfil Gerente - Acessar a tela de listagem de projetos, 'Projetos', pelo menu - Selecionar um projeto específico e entrar em sua página de visualização - Clicar em um sprint específico e entrar em sua página de visualização - Clicar no botão "Criar Tarefa" - Informar os seguintes dados nos respectivos campos: Tarefa: Aprovar versão final do design Prazo de entrega: 20/10/2021 Membro: Felipe Soares Essa tarefa é urgente: Sim Sprint: Recriar tela de splash do aplicativo - Clicar em "Criar Tarefa" 	<ul style="list-style-type: none"> - O sistema apresenta uma mensagem de sucesso informando que a tarefa foi criada com sucesso - O sistema retorna para a tela do sprint em questão, com a lista de suas tarefas - A listagem de tarefas aparece atualizada com a nova tarefa criada
---	--	--	--	--

6	Cadastrar Nova Tarefa de um Sprint Existente	Verificar que não é possível inserir no sistema uma tarefa sem data de entrega	<ul style="list-style-type: none"> - Realizar o processo de criação de uma tarefa no sistema, informado no caso de teste número 1 - Cadastrar uma tarefa sem data de entrega - Informar os seguintes dados nos respectivos campos: Tarefa: Aprovar versão final do design Prazo de entrega: vazio Membro: Felipe Soares Essa tarefa é urgente: Sim Sprint: Recriar tela de splash do aplicativo - Clicar em "Criar Tarefa" 	<ul style="list-style-type: none"> - O sistema emite uma mensagem de erro informando que 'data de entrega' é um campo obrigatório e deve ser preenchido - O sistema permanece na tela de cadastro de tarefa
---	--	--	--	---

7. Estimativa de pontos de função

A planilha de estimativa de pontos de função está disponível no repositório do projeto no GitHub, conforme link abaixo.

- <https://github.com/samantafluture/clever-front/blob/main/info/function-point.xlsx>



clever

Planilha de contagem de ponto de função - Versão 2.4

Aplicação : Clever - Gerenciamento de Projetos		Projeto : Clever - Gerenciamento de Projetos					PF IFPUG	82,00
Responsável : Samanta Gimenez Fluture		Revisor :					PF Local do EM	82,00
Empresa : Clever - Gerenciamento de Projetos		Tipo da Contagem : Projeto de Desenvolvimento					PF Local da FS	0,00
Nome da Função	Tipo	Manutenção	TD	AR/TR	Complex.	PF IFPUG	PF Local do EM	PF Local da FS
Membro	ALI		5	1	Baixa	7	7,00	
Cadastrar/Atualizar Membro	EE		5	1	Baixa	3	3,00	
Consultar Membro	CE		5	1	Baixa	3	3,00	
Projeto	ALI		7	3	Baixa	7	7,00	
Cadastrar/Atualizar Projeto	EE		7	3	Alta	6	6,00	
Consultar Projeto	CE		7	3	Média	4	4,00	
Consultar Múltiplos Projetos	CE		5	2	Baixa	3	3,00	
Gerar Dashboard com Análises dos Projetos	SE		3	2	Baixa	4	4,00	
Gerar Cronograma de Projetos	SE		1	1	Baixa	4	4,00	
Gerar Relatório de Projetos	SE		3	3	Baixa	4	4,00	
Sprint	ALI		6	3	Baixa	7	7,00	
Cadastrar/Atualizar Sprint	EE		6	3	Alta	6	6,00	
Consultar Sprint	CE		6	3	Média	4	4,00	
Tarefa	ALI		5	3	Baixa	7	7,00	
Cadastrar/Atualizar Tarefa	EE		5	3	Alta	6	6,00	
Consultar Tarefa	CE		5	3	Baixa	3	3,00	
Enviar Notificação de Tarefa a Membro	SE		3	1	Baixa	4	4,00	

Informações da implementação

Abaixo, está o link do repositório onde o projeto está sendo implementado.

- GitHub: <https://github.com/samantafluture/clever-front>

É possível acessar o código do front-end. No momento, os seguintes casos de uso listados como principais foram implementados usando o framework Angular:

- Visualizar múltiplos projetos
- Cadastrar um novo projeto
- Visualizar os sprints de um projeto
- Cadastrar um novo sprint
- Visualizar tarefas de um sprint
- Cadastrar uma nova tarefa

Foi realizado o *deploy* do código implementado acima através da Vercel, como uma forma de demonstração do projeto.

- Demo atual do projeto (front-end):
<https://github.com/samantafluture/clever-front>

8. Referências

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de banco de dados**. São Paulo: Pearson, 2018. E-book.

LARMAN, Craig. **Utilizando UML e Padrões: Uma introdução à análise e ao projeto orientados a objetos**. Porto Alegre: Bookman, 3a Edição, 2007

MENDES, Marco; **Arquitetura de Sistemas Web**. Apostila do Curso.

PRESSMAN, Roger S. **Engenharia de Software, Uma Abordagem Profissional**. 8ª. Edição, McGraw Hill – Artmed, 2016

SOMMERVILLE, Ian. **Engenharia de Software**. Pearson, 9a Edição, 2011