

**UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”
UNESP – FACULDADE DE CIÊNCIAS**

ANDRESSA CRISTINA MOUTINHO TEIZEN

PROJETARE:

Sistema Integrado de Gestão de Tarefas e Projetos

**BAURU – SP
2019**

Andressa Cristina Moutinho Teizen

PROJETARE:

Sistema Integrado de Gestão de Tarefas e Projetos

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de Informação da Universidade Estadual Paulista “Júlio de Mesquita Filho”, como requisito à obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. José Remo Ferreira Brega

BAURU – SP

2019

Resumo

O projeto desenvolvido e detalhado neste documento surgiu da necessidade de empresas de pequeno porte controlarem suas tarefas e projetos em conjunto. Muitas ferramentas fazem um controle eficaz de tarefas, outras auxiliam no gerenciamento de projetos de forma bem completa, porém, poucas conseguem controlar estas duas frentes com a mesma eficiência, focando nas necessidades de empresas com uma complexidade interna relativamente baixa. Assim, o sistema Projetare busca juntar o controle de tarefas e projetos focando suas funcionalidades nas necessidades de empresas de menor porte e tem como desafio fazê-lo sem que usabilidade do sistema seja afetada pela complexidade do mesmo.

Palavras-chave: Gestão de tarefas. Controle de projetos. Empresas.

Abstract

The project detailed in this document emerged from the small-sized companies need to control their tasks and projects together. Many tools control tasks effectively, others help project management very well, but few can control these two fronts with the same efficiency, focusing on the needs of companies with relatively low internal complexity. Thus, the Projetare system seeks to join tasks and projects control focusing their functionalities on the smaller companies needs and has the challenge of doing so without the system usability being affected by his complexity.

Keywords: Task management. Project control. Companies.

GLOSSÁRIO

Dashboard	Dashboards são painéis que mostram métricas e indicadores importantes para alcançar objetivos e metas traçadas de forma visual, facilitando a compreensão das informações geradas (NASCIMENTO, 2017).
Kanban	Quadro para registrar tarefas e ações por meio de simbologias visuais (ABRANTES, 2019).
Backlog	O backlog é um indicador de tempo, usado na Gestão da Manutenção. Em meio a tantos outros indicadores, podemos classificar o backlog como o indicador que mede o acúmulo de atividades pendentes de finalização (TELES).
GANTT	O Gráfico de Gantt ou Diagrama de Gantt, é uma ferramenta visual para controlar o cronograma de um projeto ou de uma programação de produção, um painel com as tarefas que precisam ser realizadas, a relação de precedência entre elas, a pretensão de início, sua duração, responsável e previsão de término (LEÃO, 2019).

LISTA DE ABREVIATURAS

EPP – Empresas de Pequeno Porte

ME – Microempresas

MVC – Model-View-Controller

LISTA DE ILUSTRAÇÕES

Figura 1 - Interface Bitrix24	14
Figura 2 - Interface Redmine.....	16
Figura 3 - Interface Jira	17
Figura 4 - Diagrama de Classes.....	19
Figura 5 - Diagrama de Atividades - Projeto	20
Figura 6 - Diagrama de Casos de Uso	21
Figura 7 - Árvore do projeto.....	23
Figura 8 - Camadas do padrão Repositório	25
Figura 9 - Implementação do repositório base	26
Figura 10 - Interface do repositório base.....	27
Figura 11 - Exemplo de injeção de dependência	28
Figura 12 - Classe Startup.cs	28
Figura 13 - Objetos utilizados no MVC e suas interações.....	30
Figura 14 - Ilustração do papel Modelo e Visão do padrão MVC	30
Figura 15 - Tela de autenticação.....	31
Figura 16 - Tela inicial da Visão de Tarefas	32
Figura 17 - Detalhes de tarefa.....	33
Figura 18 - Tela inicial da Visão de Projetos	34
Figura 19 - Detalhes do projeto	34
Figura 20 - Tela inicial da Visão de Recursos	35
Figura 21 - Lista de tarefas.....	36
Figura 22 - Dados exportados da lista de tarefas.....	37
Figura 23 - Quadro kanban	38
Figura 24 - Visualização do Gantt de tarefas	39

LISTA DE TABELAS

Tabela 1 - Comparação de Funcionalidades de Gerenciamento de Tarefas	17
Tabela 2 - Comparação de Funcionalidades de Gerenciamento de Projetos	18
Tabela 3 - Papéis dos projetos na estrutura da solução	24
Tabela 4 - Comparação de Funcionalidades de Gerenciamento de Tarefas - Projetare.....	41
Tabela 5 - Comparação de Funcionalidades de Gerenciamento de Projetos - Projetare.....	42

SUMÁRIO

1. INTRODUÇÃO	11
1.1. Objetivos.....	12
1.2. Estrutura.....	12
2. DETALHAMENTO DO PROBLEMA.....	13
3. SOLUÇÕES EXISTENTES	14
3.1. BITRIX24.....	14
3.2. Redmine	15
3.3. Jira.....	16
3.4. Análise	17
4. ESPECIFICAÇÃO.....	19
4.1. Diagramas	19
5. IMPLEMENTAÇÃO.....	23
5.1. Tecnologias utilizadas.....	23
5.2. Estrutura do projeto	23
5.2.1. Padrão Repositório.....	24
5.2.2. Injeção de dependência	27
5.2.3. MVC	28
5.3. Visões e uso	30
5.3.1. Visão de Tarefas.....	32
5.3.2. Visão de Projetos	33
5.3.3. Visão de Recursos	35
5.3.4. Funcionalidades comuns.....	35
5.3.5. Kanban.....	37
5.3.6. Gantt.....	38
5.4. Testes e validações	40
6. CONCLUSÃO	41

6.1. Diferenciais	41
6.2. Sugestões para projetos futuros	42
REFERÊNCIAS	43

1. INTRODUÇÃO

Atualmente no Brasil, cerca de 98,5% do total de empresas privadas são Microempresas (ME) ou Empresas de Pequeno Porte (EPP) (SEBRAE, 2018), ou seja, a grande maioria das empresas privadas, e foi visando atingir esse público que o projeto descrito neste trabalho foi desenvolvido.

Analizando as necessidades dessa faixa empresarial, notou-se que existia uma carência no mercado em relação a ferramentas de gestão de projetos focadas a este público. As ferramentas para este tipo gestão pesquisadas, em sua maioria, dispunham de grande quantidade de recursos para controle e análise de projetos, porém, estas possuíam duas deficiências quando aplicadas aos cenários de Empresas de Pequeno Porte e Microempresas, que eram:

- Grande quantidade de recursos não utilizados e grande custo pago por estes recursos. A maior parte dos recursos existentes eram de difícil aplicação para cenários menores, porém, as ferramentas não dispunham pacotes com funcionalidades voltadas para estes diferentes cenários com preços mais acessíveis;
- Falta de recursos para a gestão das tarefas dentro dos projetos.

Ou seja, as ferramentas de gestão de projetos eram focados no macro gerenciamento, assim, optou-se por alterar o foco das pesquisas e buscar ferramentas de gestão de tarefas, que possuísem funcionalidades para o controle de projetos, porém, como foco secundário.

Neste novo cenário de pesquisa, foram encontradas diversas ferramentas focadas para o público alvo da mesma, ou seja, focadas no microgerenciamento de atividades. Mas mesmo estas ferramentas, deixavam muito a desejar neste controle, pois não integravam estas duas frentes de gerência, apenas dispunham de pequenas funcionalidades para a manutenção de projetos e focavam a maior parte de suas funcionalidades e relatórios nas tarefas em si.

Foi então que decidiu-se desenvolver o *Projetare* e este trabalho destina-se a descrever a implementação deste software cujo objetivo é integrar a gestão de projetos e o controle de tarefas, focando suas funcionalidades nas necessidades de ME e EPP, que são empresas com recursos mais limitados e que de maneira geral

não possuem usuários especialistas para extrair informações complexas das ferramentas.

Logo, o principal desafio deste trabalho foi integrar estas duas frentes de gestão sem se tornar superficial em nenhuma delas e sem prejudicar a usabilidade do sistema, mantendo as interfaces do mesmo simples e intuitivas e dispondo de funcionalidades direcionadas ao perfil do usuário conectado.

1.1. Objetivos

O objetivo da elaboração deste sistema foi integrar a gestão de projetos e o controle das tarefas destes projetos, com uma interface simples e com funcionalidades voltadas às necessidades de empresas de menor porte.

1.2. Estrutura

Este documento está dividido em capítulos. No Capítulo 2, está detalhado o problema que envolve os objetivos do sistema. O Capítulo 3 apresenta as soluções já existentes no mercado para o problema abordado. No Capítulo 4, estão descritas todas as especificações do sistema desenvolvido. No Capítulo 5, apresenta-se a implementação e o funcionamento do projeto desenvolvido. No Capítulo 6 estão relacionadas as conclusões do trabalho quanto ao alcance dos objetivos e, por último, as referências utilizadas.

2. DETALHAMENTO DO PROBLEMA

Hoje em dia, no Brasil, a grande maioria das empresas são microempresas ou de pequeno porte. Estas buscam soluções para atualizar e auxiliar no crescimento dos negócios que caibam em seus orçamentos enxutos. Uma necessidade das mesmas é a de controlar suas tarefas e projetos, internos e externos, para assim organizar o cronograma interno e entregar as demandas de seus clientes dentro do prazo.

Para atender o cenário dessas empresas, a ferramenta deve retornar informações sobre o negócio, tais como custos e duração dos projetos, quantidade de tarefas, dentre outros dados que possibilitem a estas promover melhorias contínuas em seus processos.

Além disso, é importante que a ferramenta abranja diferentes níveis da organização, para que não sejam necessárias vários sistemas para se atingir os objetivos, o que acaba sendo ineficaz e custoso, pois é necessário integrar estas para se atingir o objetivo final de se controlar com eficiência tarefas e projetos em conjunto.

3. SOLUÇÕES EXISTENTES

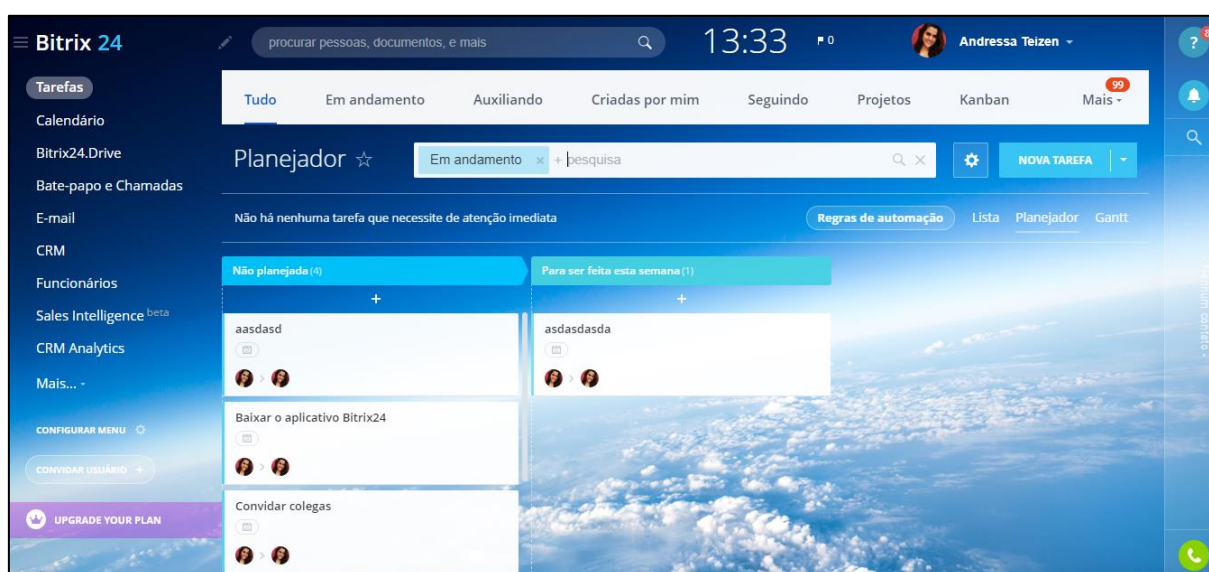
Neste capítulo serão abordados três sistemas que possuem funcionalidades parecidas com o sistema desenvolvido. São estes BITRIX24, Redmine e Jira, cujas principais características estão apresentadas nas seções 3.1, 3.2 e 3.3, respectivamente. A seção 3.4 apresenta a análise e comparação destas ferramentas.

3.1. BITRIX24

O BITRIX24 (Bitrix24, 2019) é uma ferramenta online de gerenciamento de tarefas e projetos em conjunto com a gestão de clientes. Ela possibilita a criação de tarefas em um backlog e o controle do andamento das mesmas em um quadro Kanban ou através de um Gantt. Também permite o rastreo do tempo de execução das tarefas e a abertura de tarefas de fonte externa (chamados). O BITRIX24 possui relatórios customizados do andamento das tarefas e um dashboard com gráficos e estatísticas das mesmas.

Porém, a ferramenta é um pouco limitada no gerenciamento de projetos e sua interface é confusa, como é possível observar na Figura 1, pois dispõe muitas informações ao mesmo tempo. Ela não possui uma visão Gantt por projetos, além de não possibilitar a edição das informações dispostas neste. A ferramenta também não disponibiliza uma visualização das tarefas por recurso.

Figura 1 - Interface Bitrix24



Fonte: BITRIX24, 2019

3.2. Redmine

O Redmine (Redmine, 2019) é uma ferramenta de controle de tarefas com código aberto e com a possibilidade de se vincular componentes que agregam funções ao sistema. Nele pode-se cadastrar tarefas em um backlog ou em um projeto e dentro das tarefas possui diversas funcionalidades que facilitam o controle destas, tais como: possibilidade de inclusão de checklists no corpo da tarefa, controle do tempo real gasto na execução dela, anexo de arquivos, dentre outros. Além disso, no Redmine pode-se separar as tarefas em pacotes de entregas, facilitando o gerenciamento de pacotes pré definidos. Permite também definir campos personalizados para as tarefas, possibilitando a personalização desta para atender ao universo de quem a utiliza, e disponibiliza a visualização destas tarefas em quadros kanban.

Porém, o Redmine tem poucas ferramentas para o gerenciamento de projetos, o mesmo possibilita a visualização do encadeamento de projetos com o Gantt, porém o gerenciamento desse encadeamento não pode ser feito pela visão e não há automatização de estimativas de entrega das tarefas, o que dificulta o gerenciamento de projetos.

A ferramenta também não possui uma visualização de cronograma de recursos e não possibilita a divisão da execução de uma tarefa entre outros recursos.

O foco do Redmine é disponibilizar o máximo possível de recursos para que, à partir de algumas configurações, possa atender a vários tipos de demanda diferentes, com isso a usabilidade do sistema fica afetada, como pode ser visto na Figura 2, onde a tela de inclusão de tarefas ocupa um espaço grande e desnecessário, pois assim, pode-se configurar os campos exibidos. Outro problema é que, por ser OpenSource e componentizado por diferentes fornecedores, as interfaces não seguem um padrão visual, o que também prejudica a usabilidade.

Figura 2 - Interface Redmine

The screenshot displays the Redmine web interface for creating a new task. The top navigation bar includes links for 'Página Pessoal', 'Projetos', 'Easy Gantt', 'People', 'Administração', and 'Ajuda'. The user is logged in as 'gamastf.andressa.teizen'. The main navigation tabs are 'Visão geral', 'Atividade', 'Tarefas' (selected), 'Wiki', 'Easy Gantt', and 'Configurações'. The 'Nova tarefa' form contains the following fields:

- Privado**: A checkbox option.
- Tipo**: A dropdown menu currently set to 'Bug'.
- Assunto**: A text input field.
- Descrição**: A rich text editor with formatting options (Bold, Italic, Underline, Strikethrough, Code, Header, List, Table, etc.).
- Estado**: A dropdown menu set to 'Novo'.
- Prioridade**: A dropdown menu set to 'Normal'.
- Atribuído a**: A dropdown menu for assigning the task.
- Tarefa principal**: A search input field.
- Data de fim**: A date input field with the format 'dd/mm/aaaa'.
- Tempo estimado**: A text input field followed by 'Horas'.

Fonte: Elaborado pelo autor

3.3. Jira

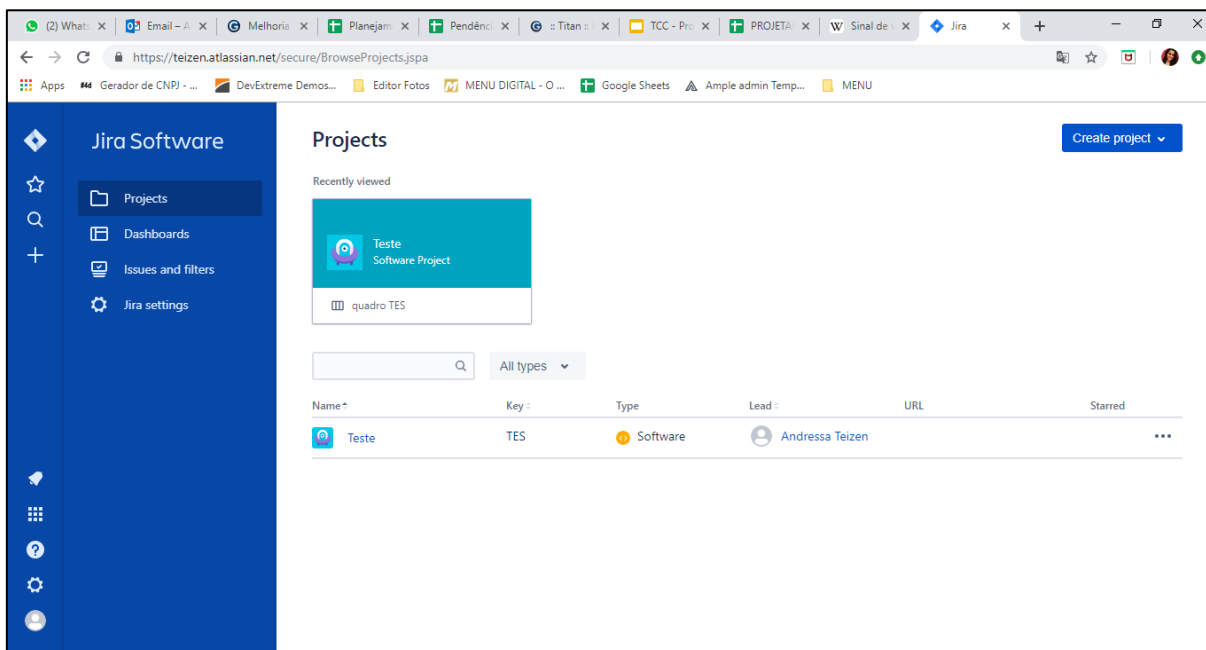
O Jira (JIRA, 2019) é uma ferramenta de gerenciamento de tarefas e projetos componentizada, ou seja, ela possui um núcleo principal de funcionalidades, mas disponibiliza por valores a parte outras funções.

Esta possui um interface simples, como é possível visualizar na Figura 3, e nela é possível a criação de tarefas em backlogs, com diversos recursos, como rastreo do tempo, inclusão de anexos, campos personalizados.

Permite também a visualização de tarefas e projetos em Gantt, bem como a manutenção destes à partir dessa visualização.

Porém, suas funcionalidades estão distribuídas entre seus componentes e estes são vistos de maneira secundária. Assim, algumas funções importantes como o rastreo do tempo, não recebem o devido foco, dificultando o uso destes.

Figura 3 - Interface Jira



Fonte: JIRA, 2019

3.4. Análise

Os três sistemas analisados possuem diversas funcionalidades voltadas para o controle de tarefas, mas poucas para o gerenciamento de projetos e principalmente dos recursos alocados nestes. O Redmine e o Bitrix24 pecam muito na interface, já o Jira, possui uma interface limpa em seu núcleo, mas quando se tratam de funcionalidades fora desse núcleo, a usabilidade fica prejudicada.

Na Tabela 1 é possível se comparar as funcionalidades de gerenciamento de tarefas das três ferramentas e na Tabela 2 as funcionalidades de gerenciamento de projeto.

Tabela 1 - Comparação de Funcionalidades de Gerenciamento de Tarefas

	KANBAN	BACKLOG	CHECKLIST	RASTREIO TEMPO
BITRIX24	✓	✓		✓
JIRA	✓	✓	✓	✓
REDMINE	✓	✓	✓	✓

Fonte: Elaborado pelo autor

Tabela 2 - Comparação de Funcionalidades de Gerenciamento de Projetos

	GANTT	CRONOGRAMA RECURSOS	MÚLTIPLOS RECURSOS	CUSTOS
BITRIX24	✓			
JIRA	✓	✓	✓	
REDMINE	✓			

Fonte: Elaborado pelo autor

À partir destas é possível ver nas tabelas que o Jira é a ferramenta que mais abrange a gestão de tarefas e projetos em conjunto, porém, mesmo as ferramentas possuindo as funcionalidades descritas, muitas não são completas, principalmente o controle das atividades diretamente no quadro Gantt, recurso importante na gestão de projetos.

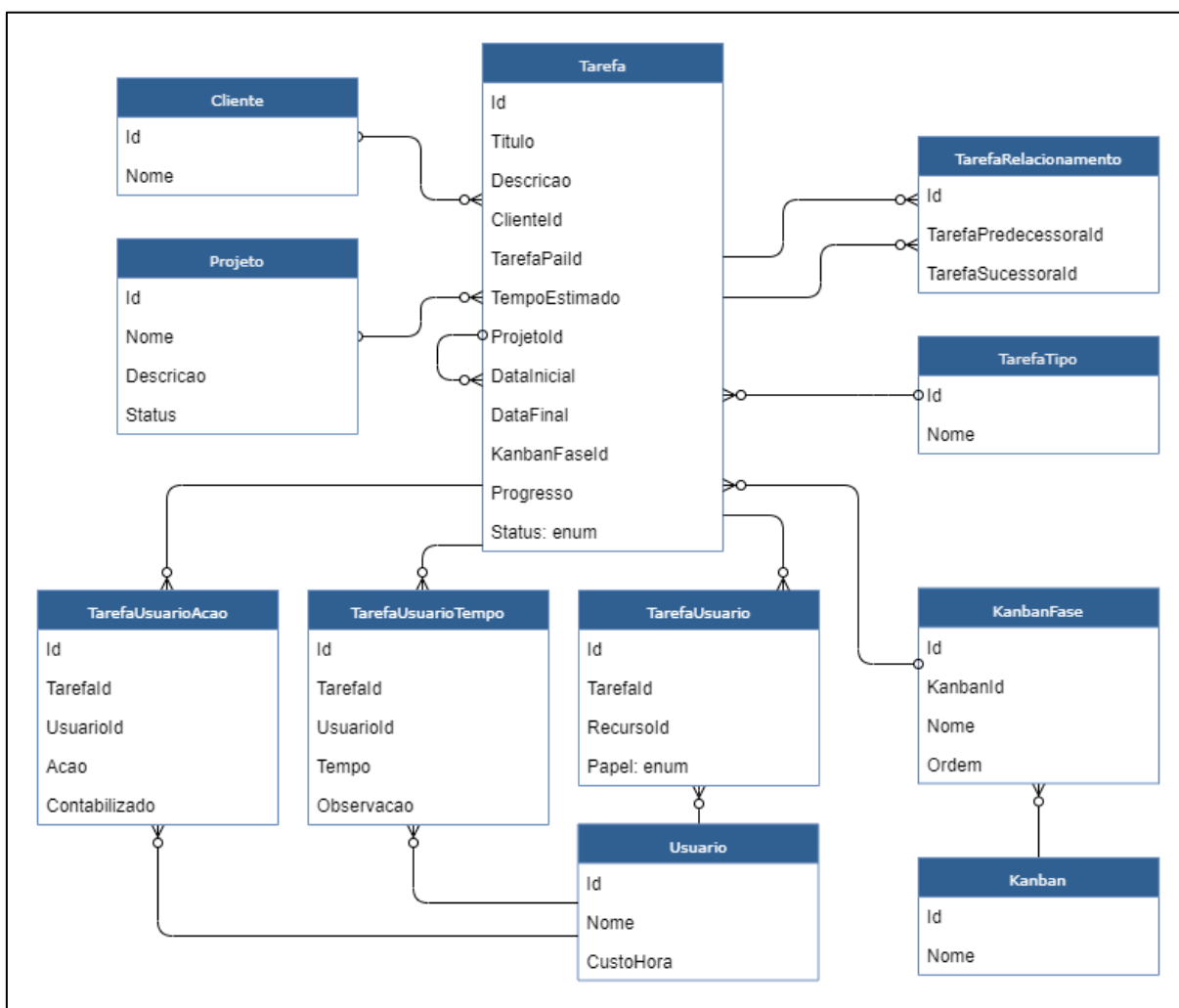
4. ESPECIFICAÇÃO

Para melhor ilustrar a modelagem do projeto desenvolvido, seguem os diagramas UML feitos em uma etapa anterior ao desenvolvimento do sistema para nortear o mesmo.

4.1. Diagramas

O primeiro diagrama desenvolvido foi o de classes, que demonstra as classes que o sistema contém, suas propriedades e o relacionamento entre estas, conforme pode ser visto na Figura 4 abaixo.

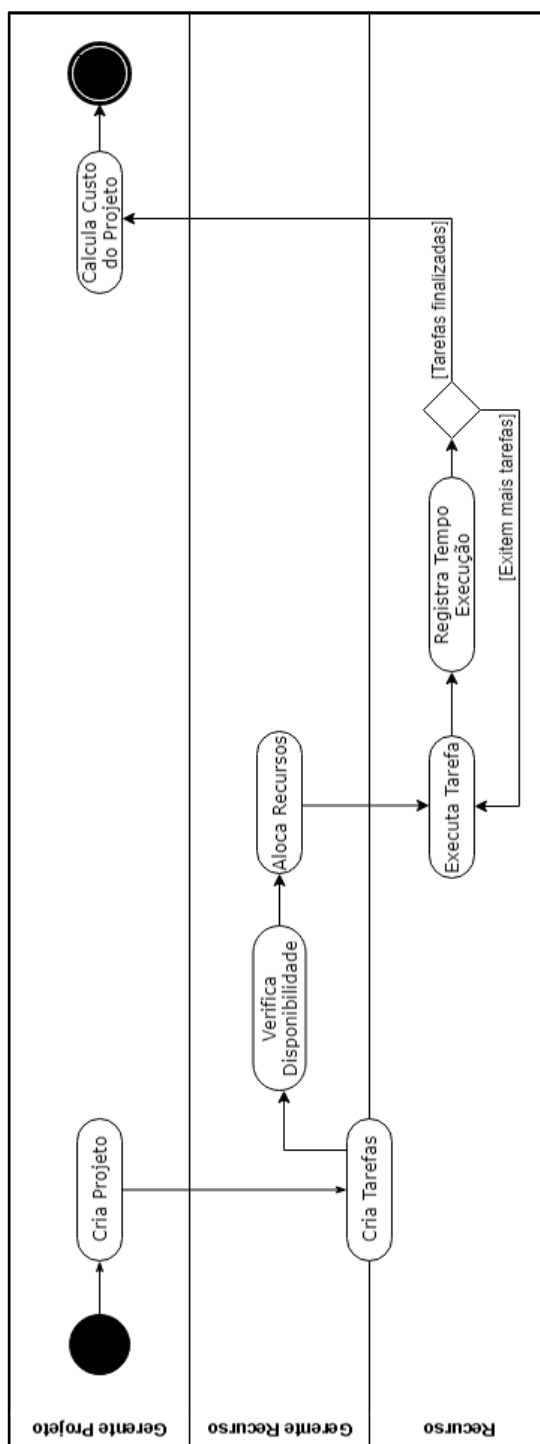
Figura 4 - Diagrama de Classes



Fonte: Elaborado pelo autor

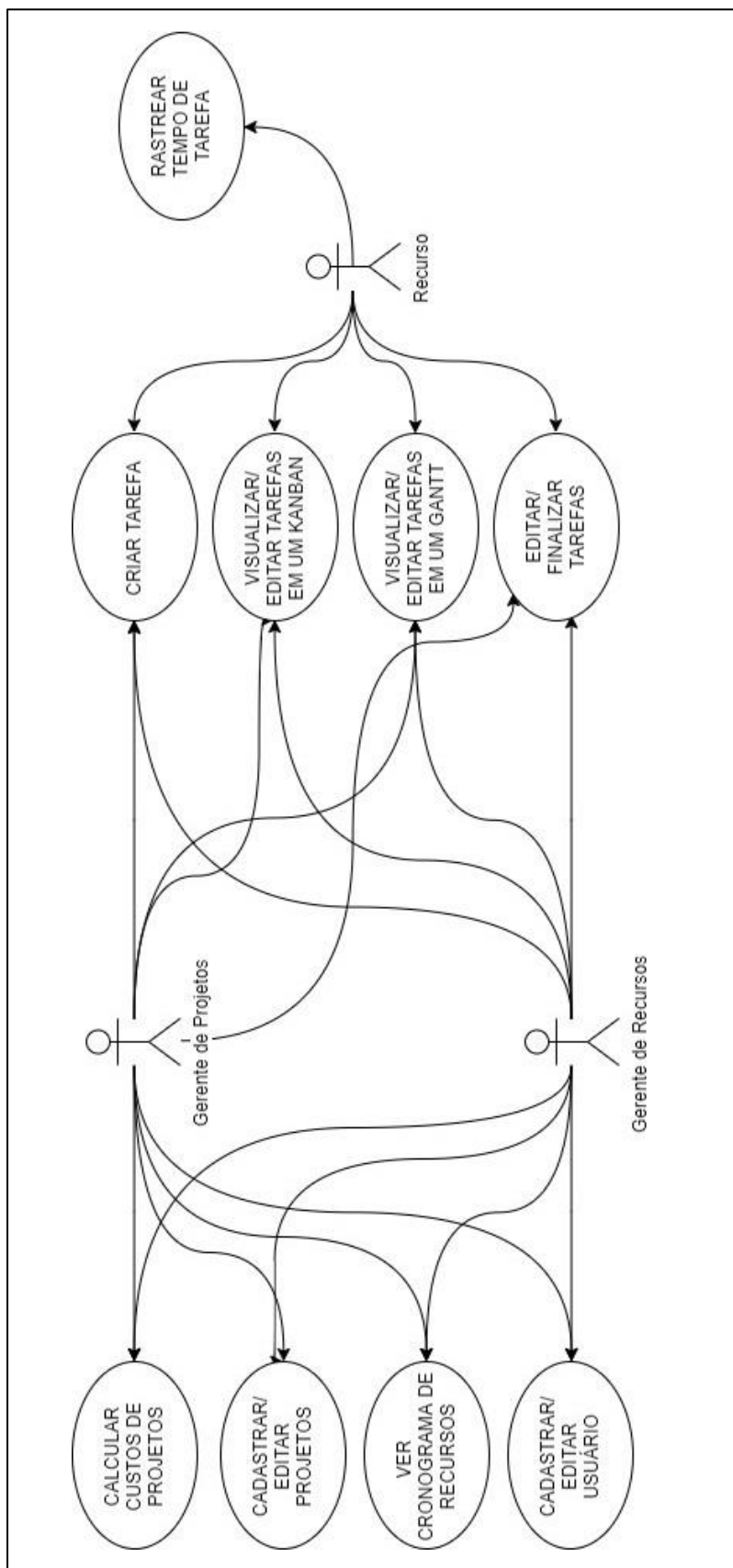
Realizou-se também um diagrama de atividades do sistema (Figura 5), que detalha a vida útil de um projeto na ferramenta, desde sua criação até o retorno de custo que este gera.

Figura 5 - Diagrama de Atividades - Projeto



Fonte: Elaborado pelo autor

Figura 6 - Diagrama de Casos de Uso



Fonte: Elaborado pelo autor

No diagrama de casos de uso da Figura 6, vê-se quais os atores do sistema (Gerente de Projetos, Gerente de Recursos e Recurso), bem como os acessos às funcionalidades que estes possuem. A partir deste é possível notar que, de maneira geral, as funcionalidades disponíveis no sistema são:

- Cálculo de custos de projetos;
- Cadastro/edição de projetos;
- Visualização do cronograma dos recursos;
- Manutenção de usuários do sistema;
- Rastreamento de tempo;
- Criação e edição de tarefas;
- Finalização de tarefas;
- Visualização e edição de tarefas em quadros kanban;
- Visualização e edição de tarefas em gráficos Gantt.

5. IMPLEMENTAÇÃO

O sistema foi desenvolvido em plataforma Web e é online, para garantir a integridade dos dados exibidos. Além disso, por se tratar de um projeto desenvolvido em parceria com a empresa Gama Soluções em Tecnologia, o mesmo contém o padrão visual da empresa e faz uso de componentes disponibilizados por esta para facilitar a disposição das informações.

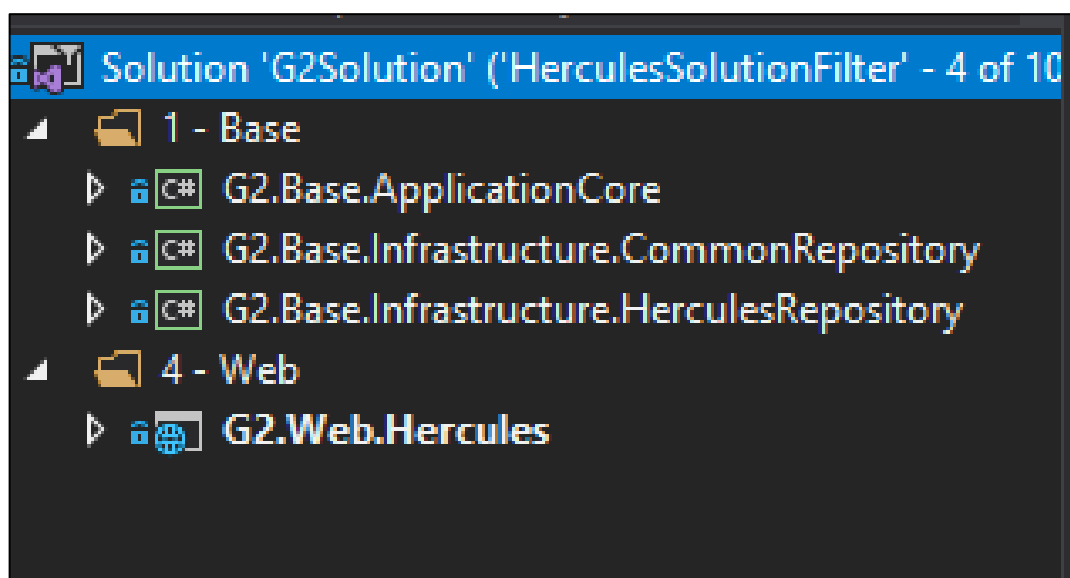
5.1. Tecnologias utilizadas

O sistema Projetare foi desenvolvido em Asp .Net Core 2.2 (MICROSOFT, 2019a), com uso de HTML 5 (W3C, 2017), Javascript (W3SCHOOLS, 2019), CSS (W3C, 2018) e JQuery (JQUERY, 2019) na programação da sua interface visual. O banco de dados é SQL Server (MICROSOFT, 2019b). Para facilitar e agilizar o desenvolvimento da ferramenta foi acrescentado ao projeto uma biblioteca de componentes chamada DevExtreme (DEVEXPRESS, 2019).

5.2. Estrutura do projeto

Para a estruturação do sistema, foram criados quatro projetos na solução G2Solution, conforme pode-se ver na Figura 7.

Figura 7 - Árvore do projeto



Fonte: Elaborado pelo autor

Estes projetos foram criados para implementar na solução um padrão de estruturação de projetos chamado de padrão Repositório abordado na seção 5.2.1 e o detalhamento a função de cada um destes nesta estrutura pode ser entendido na Tabela 3.

Tabela 3 - Papéis dos projetos na estrutura da solução

Projeto	Tipo	Responsabilidade
ApplicationCore	Biblioteca de Classes	Contém as entidades de domínio e a interface para as operações do banco de dados
CommonRepository	Biblioteca de Classes	Bibliotecas que persistem os dados no Banco de Dados, comuns a todos os projetos que a solução tem ou possa ter
HerculesRepository	Biblioteca de Classes	Bibliotecas que persistem os dados no Banco de Dados, exclusivas do projeto detalhado neste trabalho
Hercules	MVC	Aplicação web com estrutura MVC (descrita na seção 5.2.3) que utiliza as bibliotecas dos projetos anteriores através da Injeção de Dependência (descrito na seção 5.2.2)

Fonte: Elaborado pelo autor

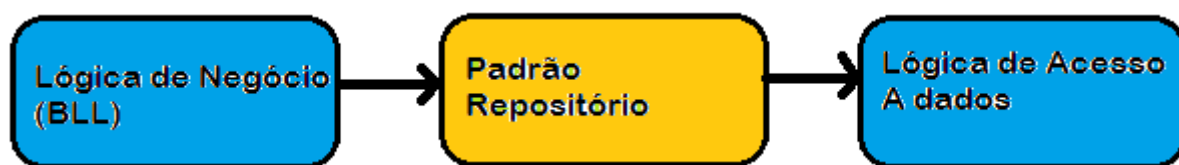
5.2.1. Padrão Repositório

O conceito de padrões de projeto em computação, foi originado da área de arquitetura e urbanismo, onde o padrão descreve um problema que ocorre repetidas vezes em um contexto, bem como a ideia chave de solução para o

problema, de modo que, se utiliza esta solução repetidas vezes, sem a obrigatoriedade de ser implementada da mesma maneira (ALEXANDER,1977).

O padrão Repositório é a separação da lógica de acesso a dados da lógica de negócio. Neste padrão, as entidades de domínio e a lógica de acesso a dados se comunicam usando interfaces, e isso esconde os detalhes do acesso a dados da camada de negócios, ou seja, a camada de negócio pode acessar os objetos de dados sem ter conhecimento da arquitetura de acesso a dados correspondente. Na Figura 8 ve-se o fluxo da informação neste padrão.

Figura 8 - Camadas do padrão Repositório



Fonte: MACORATTI, 2016

Este padrão foi adotado para facilitar a manutenção do sistema e evitar a replicação de código, pois caso posteriormente a solução possua outras interfaces, como um aplicativo, estas poderão utilizar a mesma regra de negócio que a interface web criada.

Para facilitar a implementação do padrão Repositório foi criado um repositório base G2Repository, que implementa funções básicas comuns a todos os repositórios da solução, bem como a interface desse repositório, chamada IG2Repository. A implementação e a interface do repositório base da G2Solution, podem ser vistos nas Figuras 9 e 10, respectivamente.

Figura 9 - Implementação do repositório base

```
namespace G2.Base.Infrastructure.CommonRepository.Data.Common
{
    public class G2Repository<T> : IG2Repository<T> where T : G2BaseEntity
    {
        protected readonly G2Context _dbContext;

        public G2Repository(G2Context dbContext)
        {
            _dbContext = dbContext;
        }

        /// <summary>
        /// Procura a entidade pelo Id
        /// </summary>
        /// <param name="Id"></param>
        /// <returns></returns>
        public T GetById(Guid Id)
        {
            try
            {
                var dados = GetAll();

                return dados.SingleOrDefault(t => t.Id == Id)
                    ?? throw new Exception("Registro não encontrado");
            }
        }
    }
}
```

Fonte: Elaborado pelo autor

Figura 10 - Interface do repositório base

```

namespace G2.Base.ApplicationCore.Interfaces.Common
{
    public interface IG2Repository<T> where T : G2BaseEntity
    {
        /// <summary>
        /// Procura a entidade pelo Id
        /// </summary>
        /// <param name="Id"></param>
        /// <returns></returns>
        T GetById(Guid Id);

        /// <summary>
        /// Atualiza ou insere os dados da entidade
        /// </summary>
        /// <param name="entity">Entidade a ser inclusa/alterada</param>
        /// <param name="UsuarioId">Usuário de inclusão/alteração</param>
        /// <param name="EmpresaFilialId">Filial de inclusão</param>
        /// <returns></returns>
        void InsertOrReplace(T entity, Guid UsuarioId, Guid EmpresaFilialId);

        /// <summary>
        /// Retorna todos os registros
        /// </summary>
        /// <returns></returns>
    }
}

```

Fonte: Elaborado pelo autor

5.2.2. Injeção de dependência

Injeção de Dependência é uma das formas de implementação do padrão de inversão de controle (FOWLER, 2004), usado para evitar o alto nível de acoplamento de código dentro de uma aplicação, pois o alto acoplamento pode prejudicar o desenho de uma arquitetura, tornar a sua manutenção custosa e até mesmo dificultar a sua evolução.

Uma dependência é simplesmente um objeto que a sua classe precisa para funcionar. O padrão define que essa dependência deve ser injetada na classe que a utiliza, ou seja, obrigatoriamente ela precisa ser externa a essa classe.

No projeto essas dependências são injetadas nos controladores pelo construtor, para que estes possam fazer o acesso aos dados e a execução de regras de negócio.

Figura 11 - Exemplo de injeção de dependência

```

namespace G2.Web.Hercules.Controllers
{
    public class KanbanController : Controller
    {
        private readonly IKanbanRepository _kanbanRepository;

        public KanbanController(IKanbanRepository kanbanRepository)
        {
            _kanbanRepository = kanbanRepository;
        }
    }
}

```

Fonte: Elaborado pelo autor

Na Figura 11 temos um exemplo da aplicação da injeção de dependência na solução. O controlador KanbanController, que está contido no projeto web, recebe a injeção do repositório kanban, para que possa fazer acesso às regras de negócio contidas neste.

A origem desta injeção é configurada na classe Startup.cs do projeto, conforme visto na Figura 12.

Figura 12 - Classe Startup.cs

```

services.AddScoped<ITarefaRepository, TarefaRepository>();
services.AddScoped<ITicketRepository, TicketRepository>();
services.AddScoped<IProjetoRepository, ProjetoRepository>();
services.AddScoped<IKanbanRepository, KanbanRepository>();
services.AddScoped<IKanbanFaseRepository, KanbanFaseRepository>();
services.AddScoped<IKanbanEtiquetaRepository, KanbanEtiquetaRepository>();
services.AddScoped<ITarefaKanbanEtiquetaRepository, TarefaKanbanEtiquetaRepository>();

```

Fonte: Elaborado pelo autor

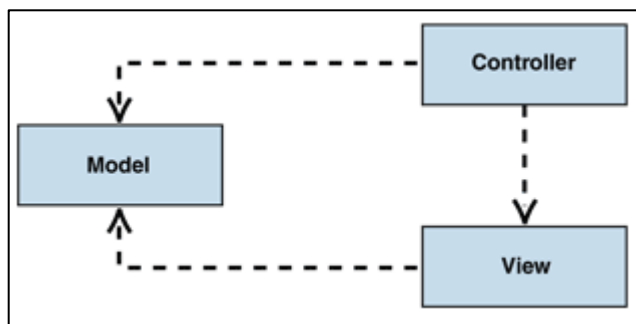
5.2.3. MVC

O padrão Model-View-Controller (Modelo-Vista-Controlador) (MVC) possui uma arquitetura que possibilita a divisão do projeto em camadas muito bem definidas. Cada uma delas, o Model, o Controller e a View, executa o que lhe é definido e nada mais do que isso.

A comunicação entre interfaces e regras de negócios é definida através de um controlador, e é este controlador que torna possível a separação entre as

camadas. Quando um evento é disparado na interface gráfica, como um clique em um botão, a interface se comunica com o controlador que por sua vez se comunica com as regras de negócios. Na Figura 13, é possível visualizar como é realizada a comunicação entre as camadas.

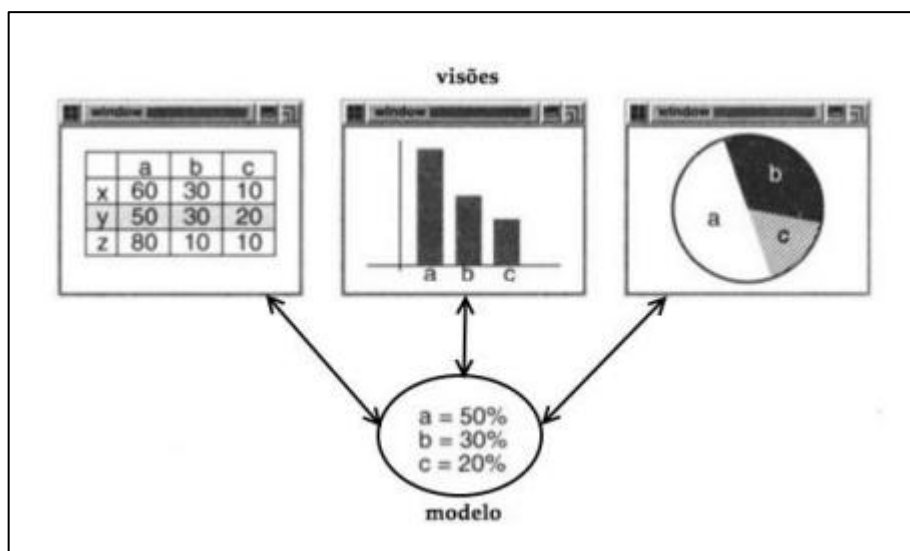
Figura 13 - Objetos utilizados no MVC e suas interações



Fonte: MICROSOFT, 2014

A utilização deste padrão traz como benefício o isolamento das regras de negócios da lógica de apresentação. Isto possibilita a ligação de várias interfaces gráficas com as mesmas regras de negócio e facilita a manutenção destas interfaces, que tendem a mudar mais rapidamente que a lógica de negócio. Na Figura 14 ve-se de forma simplificada, sem o papel do controlador uma ilustração do potencial do padrão, que possibilita a ligação de múltiplas visões a um mesmo modelo, fornecendo assim, diferentes apresentações.

Figura 14 - Ilustração do papel Modelo e Visão do padrão MVC



Fonte: GAMMA, 2000

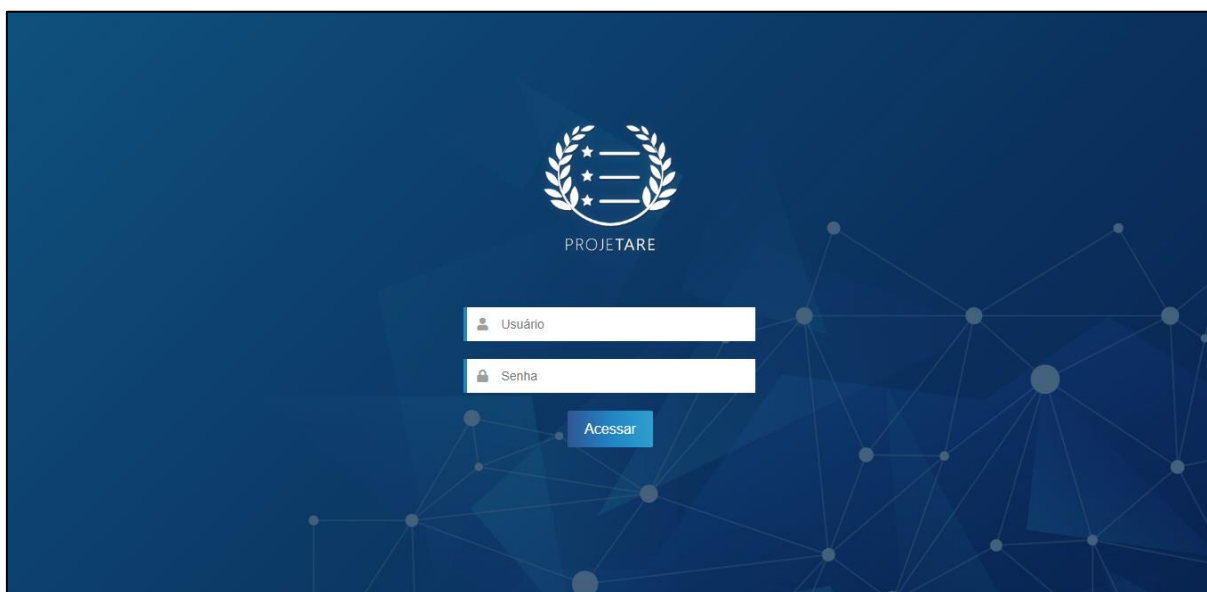
5.3. Visões e uso

Visando o objetivo deste trabalho, o sistema Projetare foi separado em três tipos de visualizações de informações, para otimizar e organizar o acesso às funcionalidades disponíveis para cada perfil de usuário, são essas visões: Visão de

Tarefas, Visão de Projetos e Visão de Recursos. Cada perfil de usuário está vinculado a uma visão diferente, cada qual focada no seu elemento principal e com facilidades para acessos de funções voltadas para este elemento, porém, com funcionalidades comuns dentre estas. Conforme exemplificado no diagrama de casos de uso deste trabalho, um exemplo destas funções comuns é a criação de tarefas. Assim, dada a importância desta funcionalidade para todos os perfis de usuário, de todas as telas do sistema, independentemente da visão, é possível acessar o botão para realizar esta ação.

A separação da visões é feita imediatamente após o usuário realizar a autenticação no sistema, pois é neste momento que o usuário é identificado, bem como seu perfil. Assim, a única tela do sistema comum a todas as visões, é a de autenticação (Figura 15), que faz esta separação.

Figura 15 - Tela de autenticação



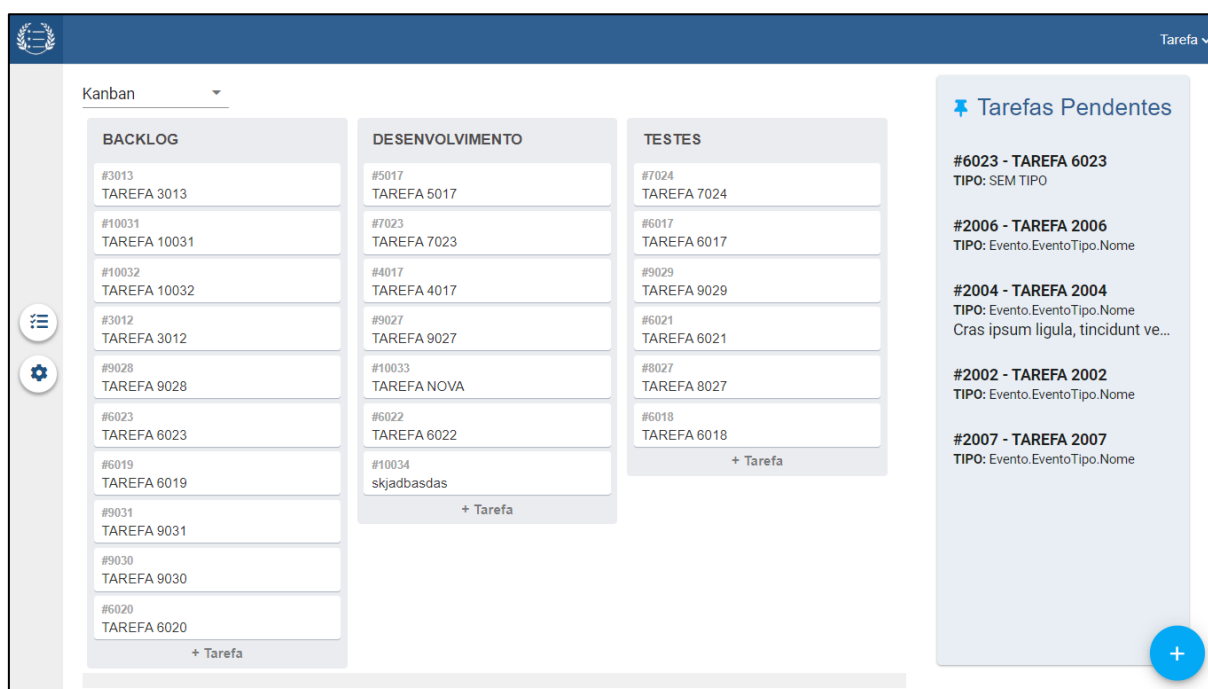
Fonte: Elaborado pelo autor

A seguir, são descritas como estas funções estão distribuídas em cada uma das três visões, bem como o elemento principal e o perfil vinculado neste modo de visualização.

5.3.1. Visão de Tarefas

Esta é a visão mais simples do sistema, focada no usuário com perfil de recurso, o elemento principal desta visão é a tarefa. Tem acesso fácil para criação e manutenção das tarefas do usuário conectado e de outros usuários ligados a este, controle de tempo das atividades exercidas, visualização das tarefas em quadro kanban, gantt e lista de suas pendências.

Figura 16 - Tela inicial da Visão de Tarefas



Fonte: Elaborado pelo autor

Na Figura 16, pode-se ver a tela inicial de um usuário com o perfil de recursos, onde há um foco muito grande no quadro kanban, para criação rápida de tarefas em suas fases corretas e também nas suas tarefas pendentes, para que estas possam facilmente ter prosseguimento. À partir de qualquer um destes dois quadros ou da lista de tarefas completa que pode ser acessada pelo menu lateral esquerdo é possível visualizar os detalhes completos das tarefas. Dentro destes detalhes o usuário tem acesso ao controle de tempo, podendo iniciar a execução de uma tarefa, pausar e finalizar de maneira fácil. Após iniciada, a pausa ou finalização da contagem de tempo pode ser executada em qualquer tela do Projetare, pois um link é disposto na barra superior deste.

O detalhe das tarefas, bem como o controle de tempo das mesmas, conforme mostra a Figura 17 abaixo, estão dispostos de maneira a se ter um rápido acesso.

Figura 17 - Detalhes de tarefa

The screenshot displays a task details form for task #5017 (ABERTA). At the top, there is a timer showing 00:02 and a 'FINALIZAR' button with a green checkmark. The form is organized into several sections:

- Header:** Task ID #5017 (ABERTA), Timer 00:02, and a 'FINALIZAR' button.
- Title Section:** 'Título *' with the value 'TAREFA 5017'. To the right are 'Data Inicio' and 'Data Fim' dropdowns, both set to 'Selecione'.
- Metadata Section:** 'Cliente' (Selecione), 'Tipo' (Selecione), 'Responsável' (Selecione), and 'Tarefa Pai' (Selecione).
- Time and Progress Section:** 'Data Prevista' (Selecione), 'Tempo Est. (h)' (23), 'Tempo Executado' (empty bar), 'Progresso' (0% bar), 'Projeto' (Selecione), and 'Kanban - Fase' (DESENVOLVIMENTO).
- Description Section:** A large text area labeled 'Descrição'.
- Footer:** 'FECHAR' and 'SALVAR' buttons.

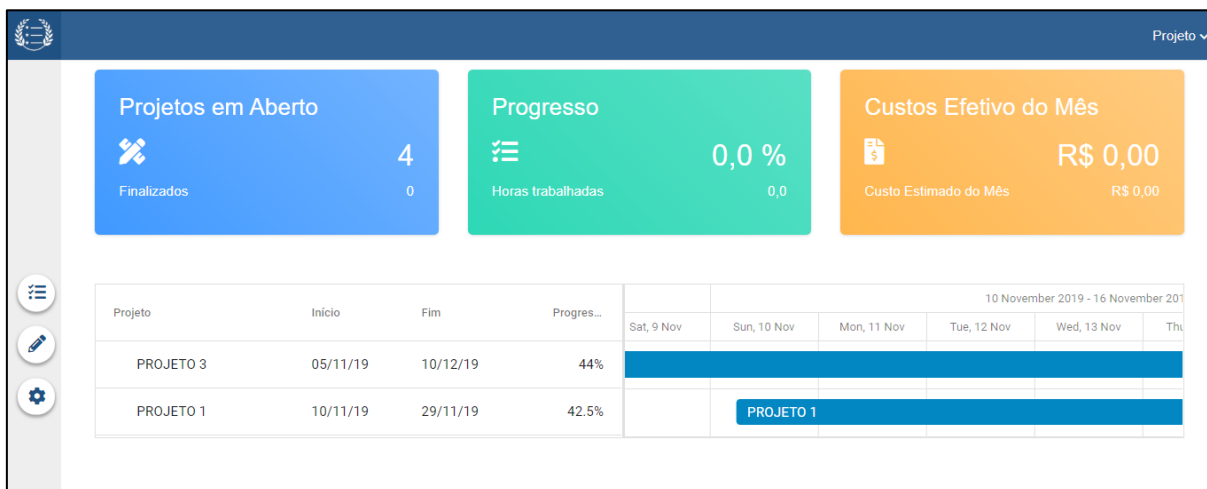
Fonte: Elaborado pelo autor

5.3.2. Visão de Projetos

Nesta visão o elemento principal é o projeto e assim o perfil vinculado a ela é o de gerente de projetos. No dashboard desta visão o sistema dispõe de indicadores de custos e andamento dos projetos no mês, bem como um gráfico Gantt que demonstra o progresso individual de cada um destes em aberto.

Além disso, esta visão dispõe de uma página de detalhes do projeto onde se pode ver o custo estimado, o custo efetivo e outros indicativos importantes do progresso individual do projeto selecionado, conforme pode ser visto na Figura 18.

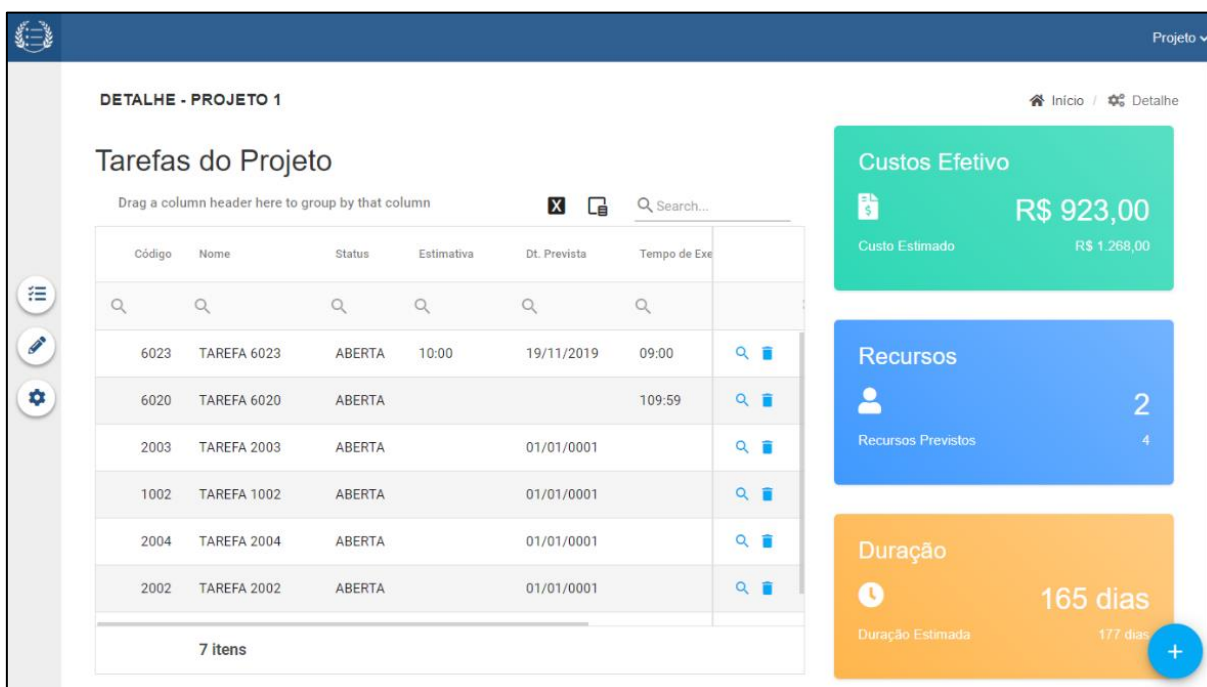
Figura 18 - Tela inicial da Visão de Projetos



Fonte: Elaborado pelo autor

Esta visão também possui acesso ao cadastro de projetos e ao detalhamento das informações de cada um dos projetos em andamento, conforme é possível ver na Figura 19.

Figura 19 - Detalhes do projeto

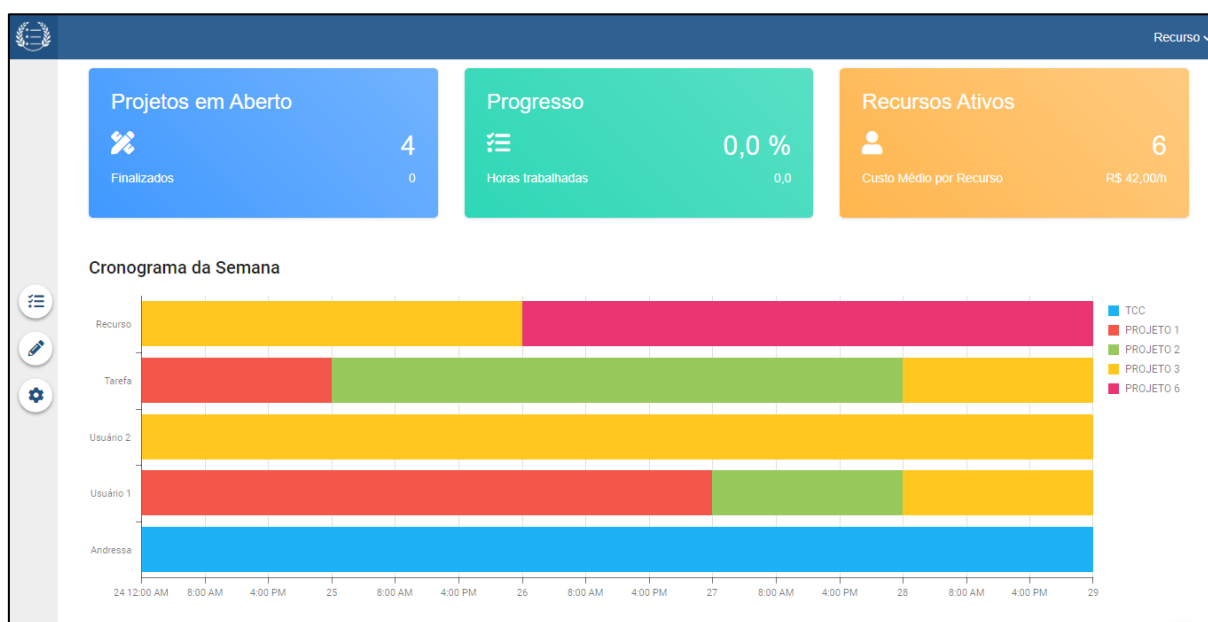


Fonte: Elaborado pelo autor

5.3.3. Visão de Recursos

Nesta, o foco é o recurso e a gestão deste recurso. Assim, o perfil vinculado a ela é o de gerente de recursos. No dashboard do perfil de gerente de recursos o elemento principal é o cronograma dos recursos do sistema, além de determinados indicadores sobre o andamento dos projetos em que estes estão vinculados, conforme mostra a Figura 20.

Figura 20 - Tela inicial da Visão de Recursos



Fonte: Elaborado pelo autor

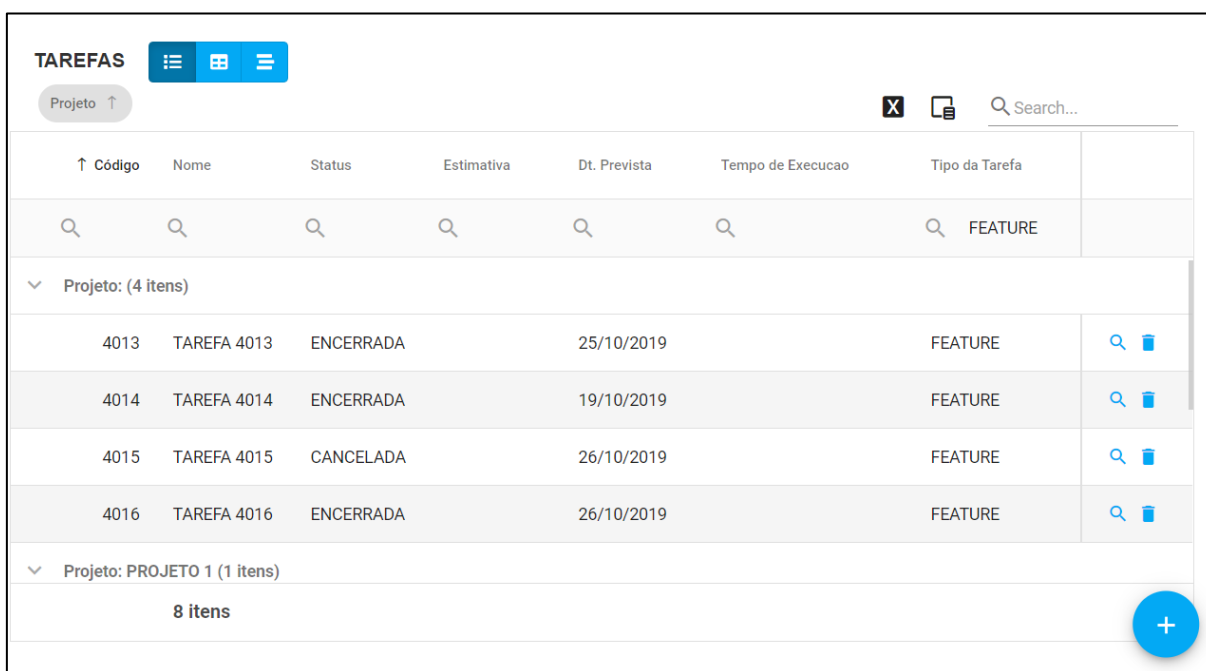
5.3.4. Funcionalidades comuns

Conforme já citado nesta monografia, as visões do Projetare possuem funcionalidades em comum, como:

- Inclusão de tarefas, que pode ser feita através de qualquer página do sistema;
- Visão da listagem de tarefas, com recursos como filtro, ordenação, agrupamento;
- Exportação de dados das tarefas para um arquivo de Excel (escolheu-se este formato, por se tratar de um dos formatos mais aceitos entre sistemas e devido a sua facilidade de customização), para que estas tarefas possam ser importadas em outras

plataformas, gerando a possibilidade de se integrar o sistema com outras soluções, para atender necessidades que as empresas venham a ter de integrações, sem a necessidade de se gerar custos para estas.

Figura 21 - Lista de tarefas

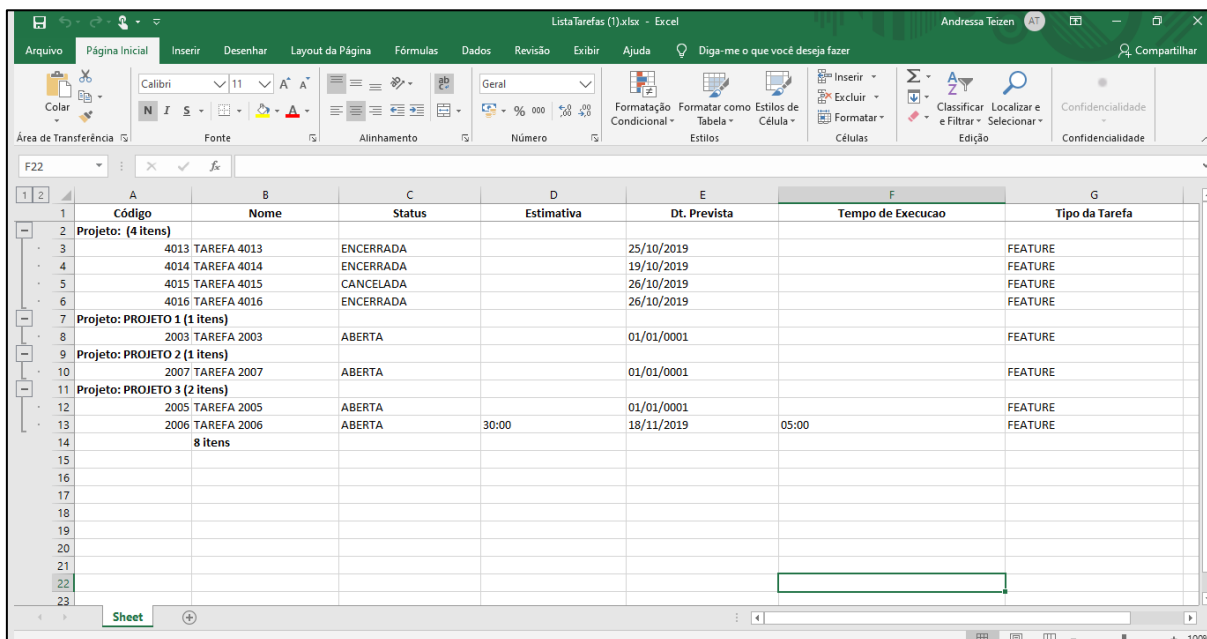


↑ Código	Nome	Status	Estimativa	Dt. Prevista	Tempo de Execucao	Tipo da Tarefa	
🔍	🔍	🔍	🔍	🔍	🔍	🔍 FEATURE	
▼ Projeto: (4 itens)							
4013	TAREFA 4013	ENCERRADA		25/10/2019		FEATURE	🔍 🗑️
4014	TAREFA 4014	ENCERRADA		19/10/2019		FEATURE	🔍 🗑️
4015	TAREFA 4015	CANCELADA		26/10/2019		FEATURE	🔍 🗑️
4016	TAREFA 4016	ENCERRADA		26/10/2019		FEATURE	🔍 🗑️
▼ Projeto: PROJETO 1 (1 itens)							
8 itens							

Fonte: Elaborado pelo autor

Na Figura 21 ve-se a lista de tarefas agrupadas por projeto, filtradas por tipo e ordenada por código, para exemplificar as funcionalidades que esta dispõe e o seu potencial de configuração, adaptável a diversos tipos de empresa focando as informações mais importantes para ela. Na Figura 22, tem-se um exemplo do arquivo gerado na exportação dos dados.

Figura 22 - Dados exportados da lista de tarefas



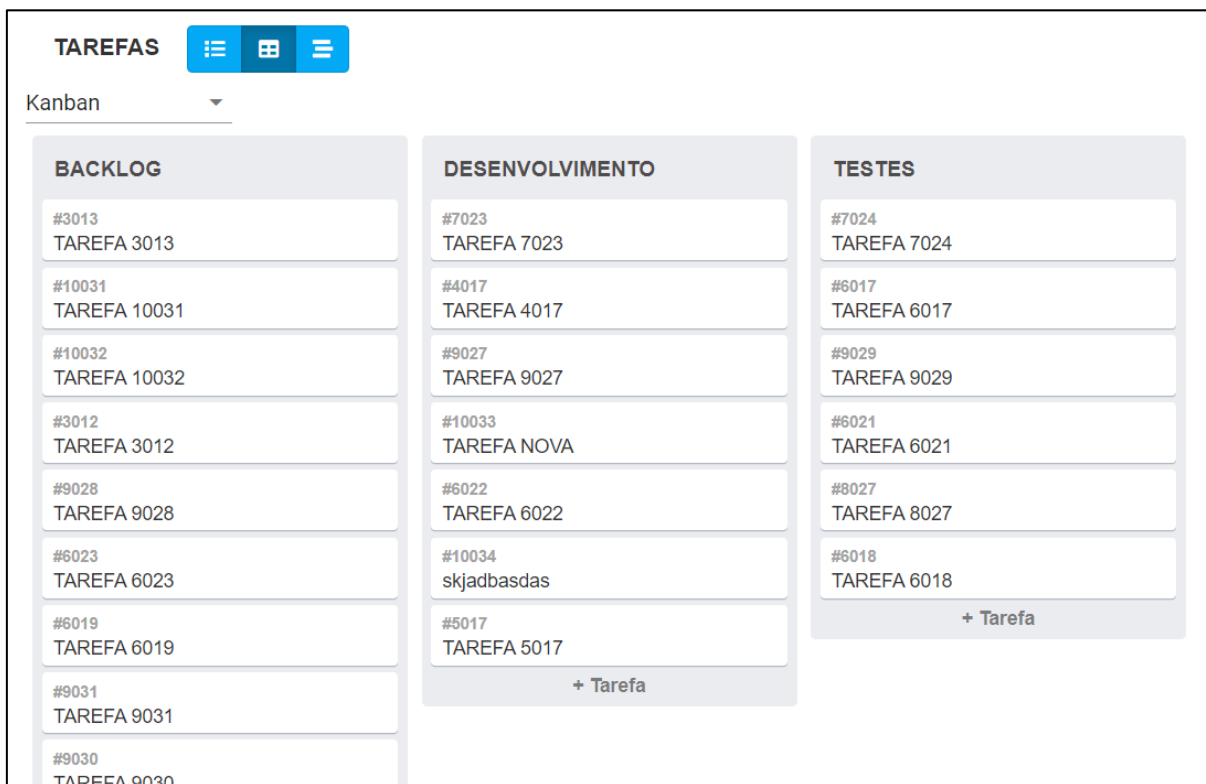
	A	B	C	D	E	F	G
	Código	Nome	Status	Estimativa	Dt. Prevista	Tempo de Execucao	Tipo da Tarefa
2	Projeto: (4 itens)						
3	4013	TAREFA 4013	ENCERRADA		25/10/2019		FEATURE
4	4014	TAREFA 4014	ENCERRADA		19/10/2019		FEATURE
5	4015	TAREFA 4015	CANCELADA		26/10/2019		FEATURE
6	4016	TAREFA 4016	ENCERRADA		26/10/2019		FEATURE
7	Projeto: PROJETO 1 (1 itens)						
8	2003	TAREFA 2003	ABERTA		01/01/0001		FEATURE
9	Projeto: PROJETO 2 (1 itens)						
10	2007	TAREFA 2007	ABERTA		01/01/0001		FEATURE
11	Projeto: PROJETO 3 (2 itens)						
12	2005	TAREFA 2005	ABERTA		01/01/0001		FEATURE
13	2006	TAREFA 2006	ABERTA	30:00	18/11/2019	05:00	FEATURE
14	8 itens						
15							
16							
17							
18							
19							
20							
21							
22							
23							

Fonte: Elaborado pelo autor

5.3.5. Kanban

Um modo de visualização importante do sistema é o quadro kanban, que permite a rápida inclusão e movimentação de tarefas e proporciona, de maneira visual, um retorno sobre o andamento das tarefas programadas, conforme pode-se ver na Figura 23.

Figura 23 - Quadro kanban



Fonte: Elaborado pelo autor

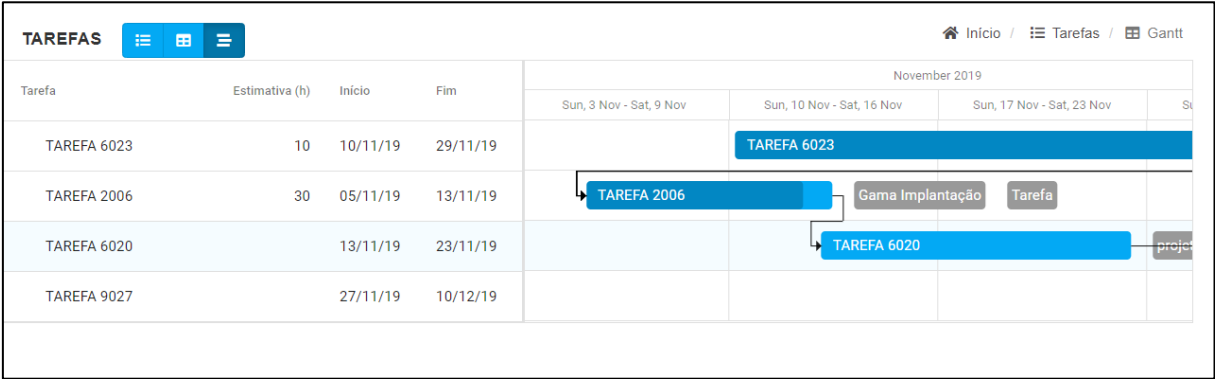
A visualização de tarefas neste quadro está disponível para todos os perfis, porém tem um maior foco na Visão de Tarefas, conforme citado neste trabalho, pois facilita a manutenção das mesmas.

5.3.6. Gantt

A visualização de tarefas no gráfico Gantt, trás ao Projetare recursos importantes para a gestão dos projetos, pois à partir dele é possível encadear tarefas, atualizar o progresso destas, bem como alterar e incluir os responsáveis por estas tarefas. Assim, pode-se ver o impacto que cada tarefa tem no ecossistema do projeto e também pode-se ver o impacto destas no cronograma final.

Na Figura 24, é possível ver o seu funcionamento no sistema.

Figura 24 - Visualização do Gantt de tarefas



Fonte: Elaborado pelo autor

5.4. Testes e validações

Os testes da aplicação foram feitos através da implementação do sistema e uso por parte da empresa Gama Soluções em Tecnologia, que se encaixa no perfil do público-alvo do sistema e que foi responsável por validar as funcionalidades em um ambiente controlado.

A empresa aplicou quatro tipos de testes de caixa-preta, em diferentes etapas, conforme detalhado abaixo:

- Testes unitários: feitos sempre que uma nova funcionalidade era finalizada, em um ambiente de homologação;
- Testes de integração: feitos no ambiente de homologação após todas as visualizações terem seus respectivos desenvolvimentos finalizados;
- Testes de aceitação: feitos pouco antes da finalização do projeto, para validar se os requisitos originais foram atendidos;
- Testes de sistema: a ferramenta foi disponibilizada e utilizada em ambiente de produção da equipe da Gama Soluções, para que a utilização da mesma fosse validada e ajustada à cenários de uso reais.

Após a realização de cada teste, as melhorias e correções propostas foram realizadas para incrementar a ferramenta.

6. CONCLUSÃO

O objetivo do Projetare, ao ser projetado, era o de sanar as dificuldades de gerenciamento de tarefas e projetos e agregar a estas a funcionalidade de gerenciamento dos recursos que executam estas tarefas, para que assim, toda a estrutura hierárquica dentro das empresas de pequeno porte seja atendida pela ferramenta, com um foco nas necessidades que cada função possui.

Esse objetivo é alcançado através de dashboards iniciais direcionados aos diversos perfis de usuários contemplados na ferramenta e de quadros e gráficos que trazem poder de gestão e agilidade ao processo, como o quadro kanban e o gráfico Gantt.

Assim o Projetare permite em uma única frente o controle de projetos, de cronogramas e demandas, a avaliação dos custos dos projetos e a entrada de dados rápida por parte dos recursos, para que estes não percam um longo tempo com a inserção de informações no sistema e ainda permitindo que as camadas mais altas da hierarquia das empresas obtenha o retorno necessário destes recursos.

6.1. Diferenciais

O Projetare além de integrar funcionalidades de diferentes ferramentas, tem como diferenciais que as outras não contemplaram, cronograma dos projetos, durante a execução do mesmo e o cálculo de custos dos projetos, conforme pode ser visto na Tabela 4 e na Tabela 5.

Tabela 4 - Comparação de Funcionalidades de Gerenciamento de Tarefas - Projetare

	KANBAN	BACKLOG	CHECKLIST	RASTREIO TEMPO
BITRIX24	✓	✓		✓
JIRA	✓	✓	✓	✓
REDMINE	✓	✓	✓	✓
PROJETARE	✓	✓		✓

Fonte: Elaborado pelo autor

Tabela 5 - Comparação de Funcionalidades de Gerenciamento de Projetos - Projetare

	GANTT	CRONOGRAMA RECURSOS	MÚLTIPLOS RECURSOS	CUSTOS
BITRIX24	✓			
JIRA	✓	✓	✓	
REDMINE	✓			
PROJETARE	✓	✓	✓	✓

Fonte: Elaborado pelo autor

Nas funcionalidades que o sistema compartilha com as demais soluções existentes, buscou-se aprimorar as mesmas, para obter-se resultados melhores, assim, nas demais ferramentas citadas neste trabalho, o controle das tarefas através do gráfico Gantt é bem superficial, na maioria se limitando a uma visualização das informações, sem manutenção das mesmas. Além disso, no quadro kanban, o foco na praticidade de se incluir uma tarefa sem a necessidade de preencher todas as informações da mesma inicialmente, permite que se controle rapidamente a execução das mesmas, para que só se necessite preencher os demais detalhes na finalização do processo.

Desta maneira, o Projetare conseguiu atingir seu objetivo de integrar o controle das tarefas com a gestão de projetos.

6.2. Sugestões para projetos futuros

As ideias que foram pensadas e não inclusas no sistema atual, mas que podem ser implementadas futuramente são:

- Adaptação do sistema para dispositivos móveis;
- Cadastro e manutenção de cronogramas do usuário;
- Cadastro de chamados externos que originem tarefas.

REFERÊNCIAS

ABRANTES, Letícia. **Entenda o que é e como funciona o método Kanban**. ROCKET CONTENT, 2019. Disponível em: <<https://rockcontent.com/blog/kanban/>>. Acesso em: 25 de jun. de 2019.

ALEXANDER, Christopher. **A pattern language**. Nova York: Oxford University Press, 1977.

BITRIX2424. **Bitrix24**. 2019. Disponível em: <<https://www.bitrix24.net/>>. Acesso em: 15 de mar. de 2019.

DEVEXPRESS. **DEVEXTREME**. 2019. Disponível em: <<https://js.devexpress.com/>>. Acesso em: 14 jun. 2019.

FOWLER, Martin. **Inversion of Control Containers and the Dependency Injection pattern**. 2004. Disponível em: <<https://www.martinfowler.com/articles/injection.html>>. Acesso em: 16 nov. 2019.

GAMMA, Erich. **Padrões de projeto: soluções reutilizáveis de software orientado a objetos**. São Paulo: Bookman, 2000.

JIRA. **Jira Software**. 2019. Disponível em: <<https://br.atlassian.com/software/jira>>. Acesso em: 15 de mar. de 2019.

JQUERY. **JQuery**. What is jQuery?. 2019. Disponível em: <<https://jquery.com/>>. Acesso em: 14 jun. 2019.

LEÃO, Thiago. **Gráfico de Gantt: o que é, como funciona e como montar o seu**. NOMUS. Disponível em: <<https://www.nomus.com.br/blog-industrial/grafico-de-gantt/>>. Acesso em: 25 de jun. de 2019.

MACORATTI, José Carlos. **O padrão repositório revisitado**. IMASTERS. 2016. Disponível em: <<https://imasters.com.br/dotnet/o-padrao-repositorio-revisitado-parte-01>>. Acesso em: 16 de nov. de 2019.

MICROSOFT. **Model-View-Controller**. 2014. Disponível em: <[https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff649643\(v=pandp.10\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff649643(v=pandp.10)?redirectedfrom=MSDN)>. Acesso em: 16 nov. 2019.

MICROSOFT. **ASP.NET Core 2.2: Introdução ao ASP.NET Core**. 2019a. Disponível em: <<https://docs.microsoft.com/pt-br/aspnet/core/?view=aspnetcore-2.2>>. Acesso em: 14 nov. 2019.

MICROSOFT. **SQL SERVER**. 2019b. Disponível em: <<https://www.microsoft.com/pt-br/sql-server>>. Acesso em: 14 jun. 2019.

NASCIMENTO, Rodrigo. **O que é um Dashboard?**. MARKETING POR DADOS, 2017. Disponível em: <<http://marketingpordados.com/analise-de-dados/o-que-e-dashboard-%F0%9F%93%8A/>>. Acesso em: 22 de mar. de 2019.

REDMINE. **Redmine**. 2019. Disponível em: <<https://www.redmine.org/>>. Acesso em: 13 de jun. de 2019.

SEBRAE. **Perfil das Microempresas e Empresas de Pequeno Porte**. 2018. Disponível em: <<http://www.sebrae.com.br/Sebrae/Portal%20Sebrae/UFs/RO/Anexos/Perfil%20das%20ME%20e%20EPP%20-%2004%202018.pdf>>. Acesso em: 22 de mar. de 2019.

TELES, Jhonata. **O que é Backlog e como ele pode me ajudar na Gestão da Manutenção?**. ENGETELES. Disponível em: <<https://engeteles.com.br/o-que-e-backlog/>>. Acesso em: 25 de jun. de 2019.

W3C. **HTML5**. World Wide Web Consortium. 2017. Disponível em: <www.w3.org/TR/html5/>. Acesso em: 14 jun. 2019.

W3C. **CSS**. World Wide Web Consortium. 2018. Disponível em: <<https://www.w3.org/TR/CSS/>>. Acesso em: 14 jun. 2019.

W3SCHOOLS. **Javascript**. Javascript Tutorial. 2019. Disponível em: <www.w3schools.com/js/>. Acesso em: 14 jun. 2019.