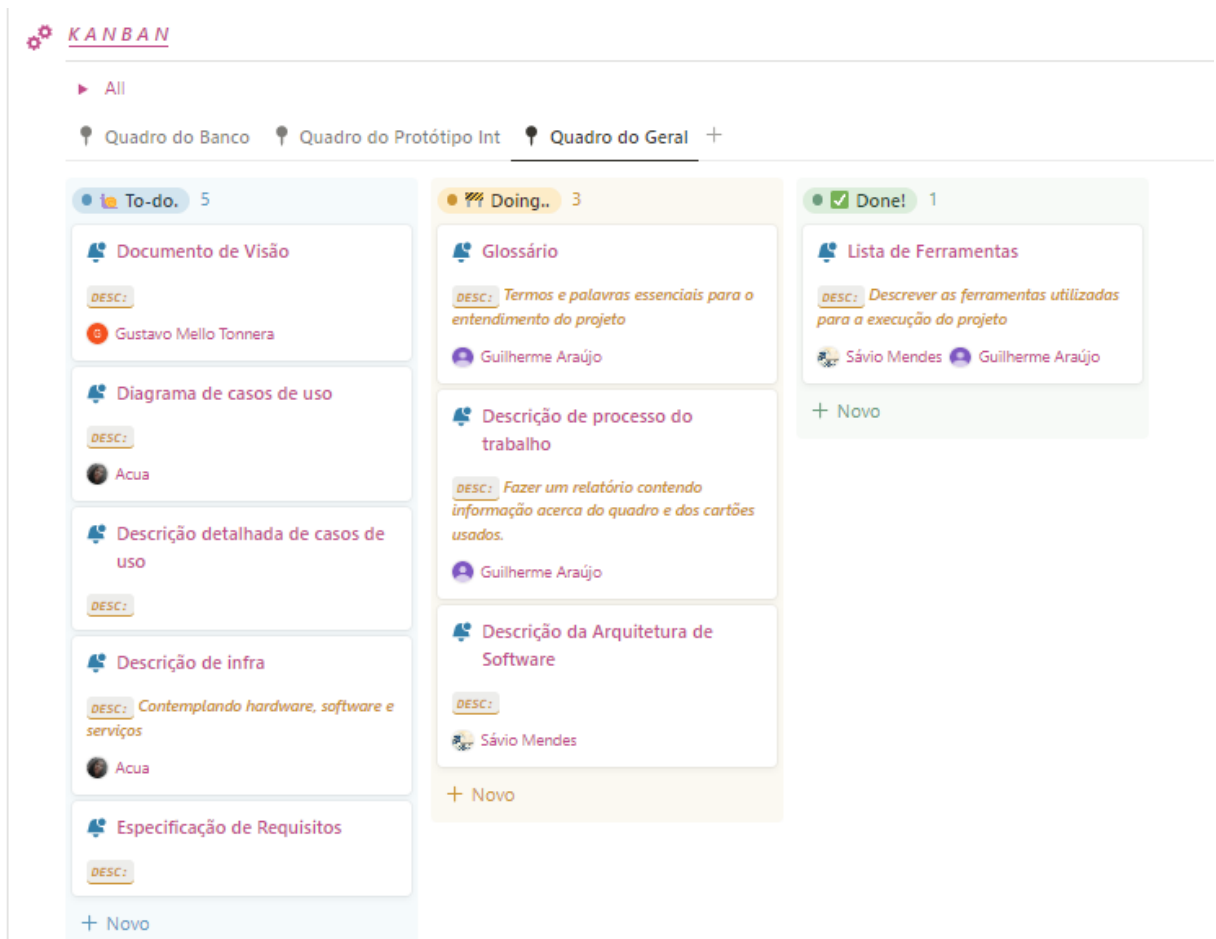


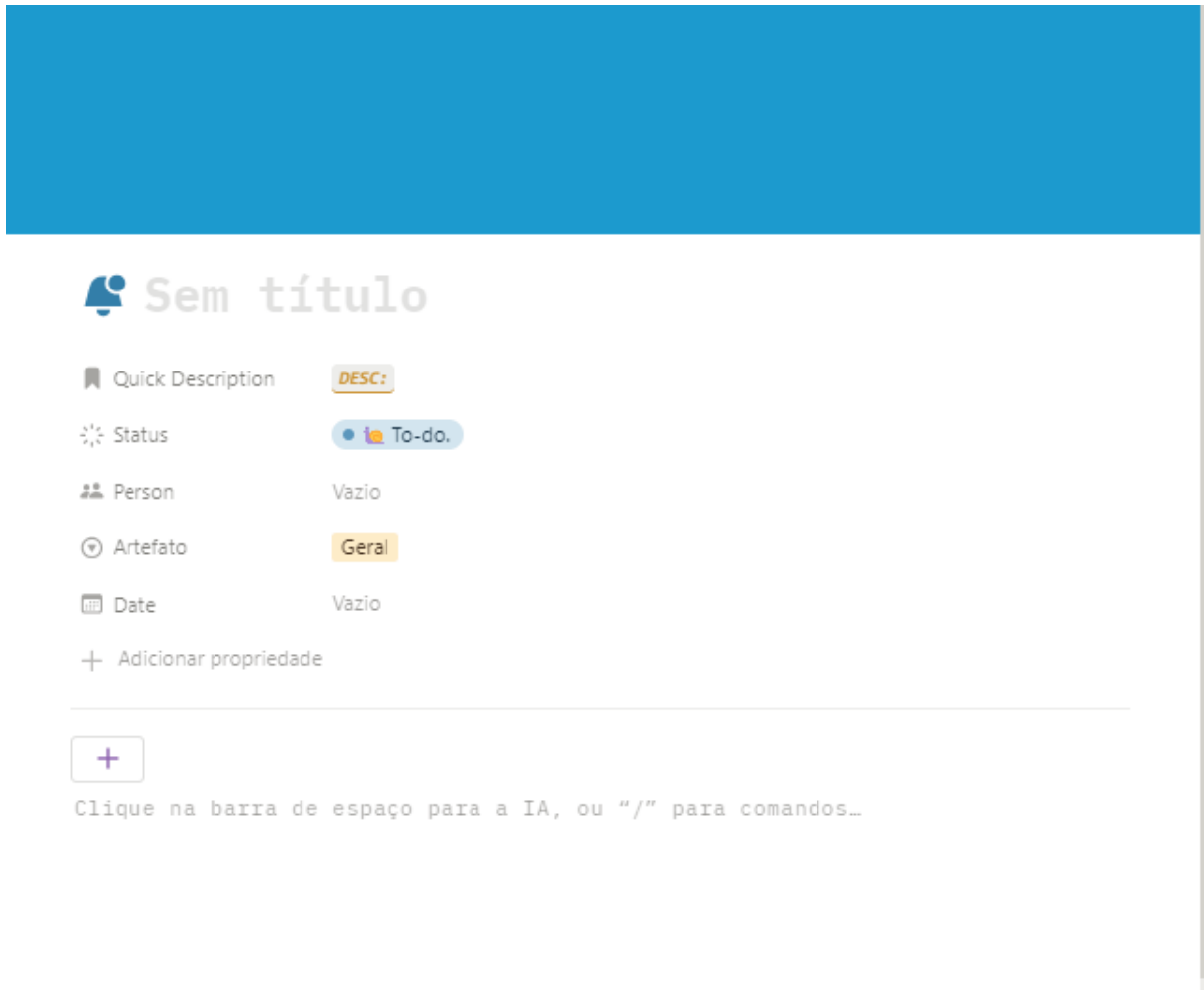
Descrição do processo de gerenciamento

Para o desenvolvimento de software é necessário ter o máximo de organização possível, uma vez que assume-se um compromisso com os clientes. Visto isso, existem diversas ferramentas de gerenciamento que podem ser usadas para melhor compreensão do que deve ser feito, do que está sendo feito e o que já foi feito. Para este trabalho acadêmico a equipe optou pelo uso da ferramenta Notion, que é gratuita e simples de utilizar. O ambiente ficou dividido em 3 quadros: Quadro do Banco, que ficou com projeto físico e lógico do banco de dados para aplicação Kanban, Quadro do Protótipo, que ficou com os wireframes da aplicação Kanban e o Quadro Geral, que ficou com as distribuições das tarefas. Na imagem abaixo encontra-se o Quadro Geral do Kanban que a equipe utilizou.



Gerenciamento de Quadros

Cada quadro em cada coluna possui campos que nos permitem melhor gerenciar o desenvolvimento de software. A imagem abaixo mostra um quadro *default* da coluna a fazer (*to-do*), o campo *artefato* indica em qual conjunto de quadros este novo quadro ficará.



A partir dessa separação por quadros a equipe consegue ficar ciente do que deve ser feito, o que está sendo feito por outros colegas e o que ainda falta ser feito, permitindo assim o gerenciamento de tempo individual para cada membro da equipe.

Gerenciamento de tempo

Para melhor gerir o tempo da equipe entre as tarefas e visto a correria do semestre letivo os membros da equipe discutiram e chegaram a conclusão que aplicar princípios da metodologia SCRUM seria uma boa escolha, daí ficou decidido fazer sprints com intervalo de 7 dias, no começo houve dificuldades pois as tarefas ainda não estavam separadas, mas após 2 sprints o ambiente no Notion já estava organizado com as tarefas, assim possibilitando a equipe separar o que cada membro iria fazer. Após analisar a divisão das tarefas e o tempo hábil por membro, a equipe decidiu estender o intervalo entre sprints para 14 dias.

Gerenciamento de Produção

Para organização na produção dos artefatos a equipe entrou em consenso de que a ferramenta GitHub seria melhor por também preencher o portfólio dos membros da equipe, daí os artefatos encontram-se armazenados em um repositório público e, mais uma vez, a equipe decidiu em consenso tentar aplicar o que está em evidência no mercado de trabalho, exemplo disso são os padrões adotados para submissão ao repositório remoto (commit) conforme mostra a imagem abaixo.

Padrão de Commit

- Usaremos commits no modo imperativo e descritivo em inglês

```
$emoji $tipo_commit - $descricao_breve
```

- `:page_facing_up: docs` : Quando enviar alguma alteração em algum `README.md` ou acréscimo de algum artefato;
- `:zap: feat` : Quando enviar codificação nova (feature);
- `:wrench: fix` : Quando enviar alguma correção em artefatos e/ou codificações;
- `:file_folder: organize` : Quando enviar alteração que organiza artefatos e/ou codificações;

Exemplos de cada tipo de commit:

1. 📄 docs - Add descriptive README.md's
2. ⚡ feat - Add card component
3. ✂ fix - Limits card's description to X chars
4. 📁 organize - move `index.js` to root

A tentativa de padronização teve um resultado positivo já que por se tratar de um trabalho acadêmico quanto mais simples ficasse melhor, ao final os membros da equipe concluíram que tiveram uma boa evolução técnica e conseguiram obter insights que com certeza ajudaram não só na vida acadêmica mas também na vida profissional ao entrar em projetos de grande porte, com uma equipe capacitada, comprometida que segue os padrões e tem regras de negócios bem definidas.