**Estratégia de teste**

Este documento define a estratégia de testes utilizada no projeto HermesBar Management, com o objetivo de registrar quais técnicas utilizamos para garantir que o produto está de acordo com as especificações.

Não utilizamos nenhum método formal para gerenciar e definir quando os testes seriam realizados e entregues. A medida que os módulos foram ficando prontos, fomos realizando os testes de acordo com a técnica utilizada e citada abaixo. Nossos “releases” foram feitos através do **Kanbanize,** conforme citado anteriormente, à medida que os módulos ficavam prontos, integrantes da equipe testavam o as novas funcionalidades e depois disso outros integrantes testavam novamente para garantir que a saída dos dados estavam conforme o especificado.

Durante o desenvolvimento do software utilizamos como técnica de teste os testes de **caixa – preta,** onde definimos dados de entrada e os esperados dados de saída, sem saber focar em como os dados de entrada foram processados para gerar a informação final. Ao final de cada ciclo de testes, os dados de saída são comparados com aqueles definidos nos casos de uso e regras de negócio. A aprovação depende da paridade entre eles e aceitação do cliente.

Utilizamos esta técnica porque a técnica de caixa-preta é útil para todos os níveis de teste, desde os de componentes até os de sistema e aceitação, pois é fundamental que o software atenda suas especificações. Além de simples não ocupou muito tempo da equipe como a técnica de caixa-branca por exemplo, onde teríamos que estudar se a maneira como os dados foram processados é segura e correta.

Além da técnica de caixa preta, utilizamos também o teste de **Regressão.**

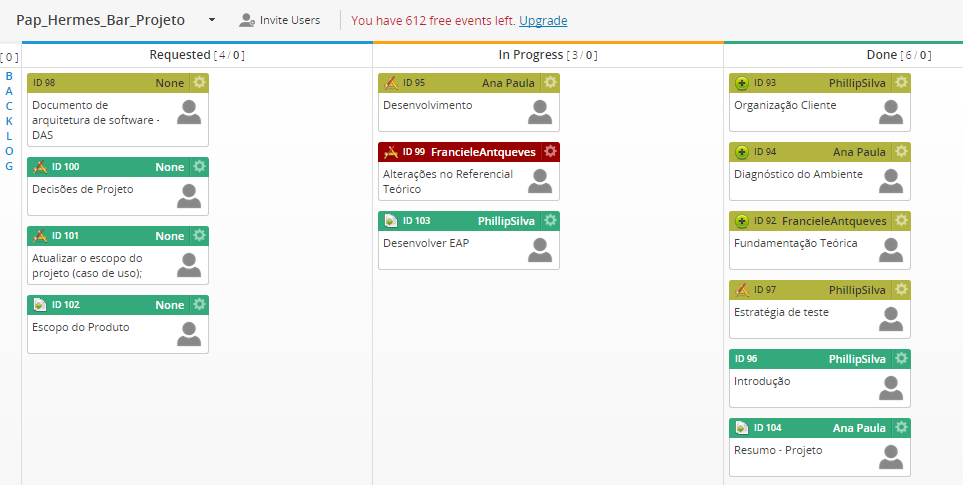
Utilizamos o teste de regressão de maneira que cada atualização no código fonte ou implantação de novos módulos testávamos o sistema desde o primeiro caso de uso, isso para garantir que o que foi acrescido no sistema não prejudicou o que já estava implementado.

**Conclusão:**

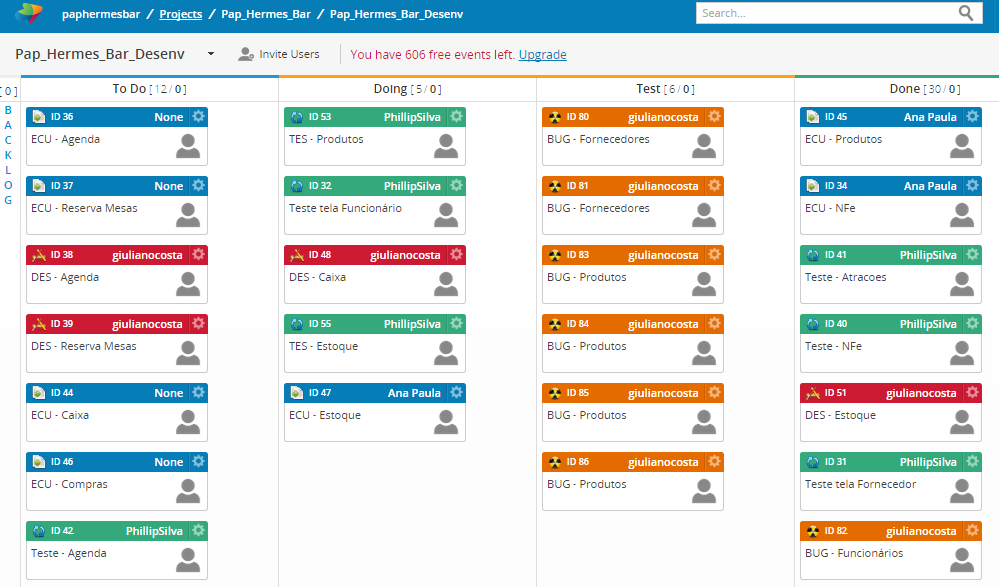
Testes de software são uma parte integrante do desenvolvimento de softwares. A importância dos testes de software no ciclo de vida do desenvolvimento é melhorar a performance, consistência e outros fatores importantes. Ele tem um papel principal a desempenhar na depuração do erro. O teste é uma etapa que ajuda a saber a diferença entre o resultado real e o resultado esperado, isso pode ajudar a melhorar a qualidade de produto. Qualquer software que é produzido sem qualquer teste pode ser perigoso para seus usuários. No ciclo de vida de desenvolvimento de software, o teste é importante do ponto de vista técnico. É muito importante para testar o software e evitar situações perigosas. Confiabilidade no software é um fator importante e é determinado pelos testes. Com a ajuda de testes de desenvolvimento de software, todo o produto pode ser transformado em um produto forte e consistente.

**Kanbanize**

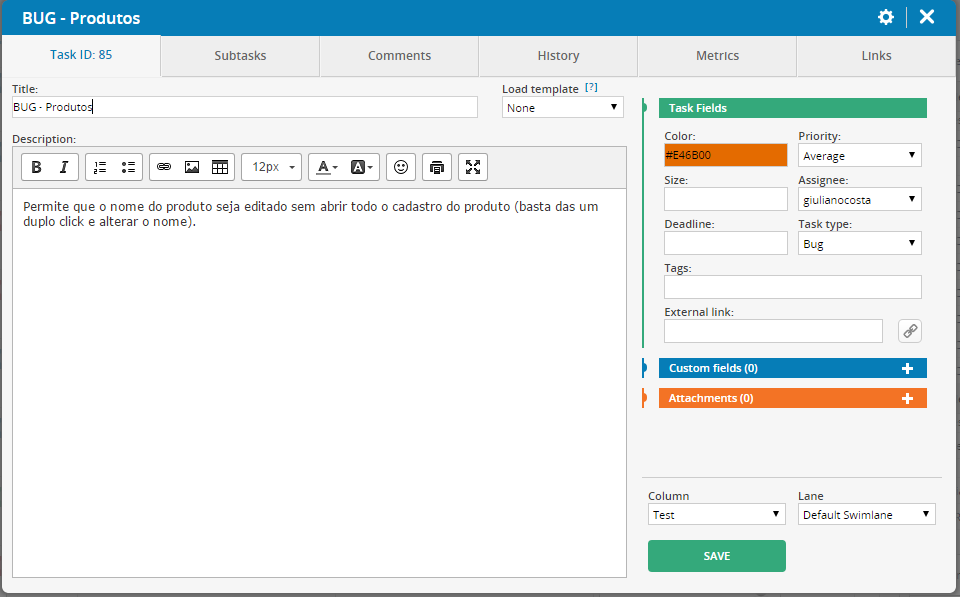
Modelos de gerenciamento de projeto de atividades de documentação.

****

Modelos de gerenciamento de projeto de atividades de desenvolvimento e documentação.



Modelo de visualização para correção de erros no sistema.



As tesks (atividades) que estão no campo *Requested* foram solicitadas e estão esperando ser atribuídas para início, as que estão no campos In Progress já foram atribuídas a um integrante do grupo e está sendo desenvolvida e por fim, as tesks que estão no campos *Done* são as atividades já concluídas pelos integrantes.