**UNIVERSIDADE DE BRASÍLIA**

**DANILO YUJI FUKUDA**

**GABRIEL DE CARVALHO TOLLINI**

**THIAGO XAVIER DA SILVA**

**VINÍCIUS DE SOUSA SANTANA**

**VITOR PEREIRA TAVARES DA COSTA**

**SISTEMA DE ADMINISTRAÇÃO CONDOMINIAL - SISCOND**

**Brasília**

**2017**

**DANILO YUJI FUKUDA – 120169258**

**GABRIEL DE CARVALHO TOLLINI – 140020454**

**THIAGO XAVIER DA SILVA – 110066766**

**VINÍCIUS DE SOUSA SANTANA – 13/0067385**

**VITOR PEREIRA TAVARES DA COSTA– 140165584**

**SISTEMA DE ADMINISTRAÇÃO CONDOMINIAL - SISCOND**

Trabalho apresentado à disciplina Engenharia de Software da Universidade de Brasília. 1º semestre, turma B.

**Professor: Gustavo Cordeiro Galvão Van Erven**

Brasília, 05 de junho de 2017.

**ES012017**

**Projeto**

Desenvolvimento de sistema para auxiliar na administração do Condomínio Pitanguinhas com o objetivo de tornar mais transparente a gestão orçamentária além de promover a integração dos condôminos com a administração do condomínio e, em última análise, tornar mais dinâmica e cômoda a prestação de alguns serviços.

**Metodologia de desenvolvimento**

Será utilizada a metodologia de desenvolvimento XP na qual os integrantes do grupo assumirão os seguintes papéis:

* Programador: Todos
* Cliente: Grupo xxxxx  (Maysa, Geraldino, Miguel)
* Testador: Vitor
* Rastreador: Vinícius
* Treinador: Vitor e Vinícius
* Consultor: Thiago
* Chefão: Grupo Massive Dynamics

**Stakeholders**

Durante as reuniões com o cliente foram identificados os seguintes stakeholders.

* **Administração**

Composta pelo síndico, subsíndico e conselho fiscal, este stakeholder possui problemas de comunicação com os condôminos, que não atendem em grande número as convocações para reuniões, bem como de clareza e objetividade na prestação de contas do condomínio, feita durante essas reuniões de baixo quórum e em um curto espaço de tempo. Sendo assim, é necessário que sejam melhoradas a forma de comunicação entre administração e condôminos e a maneira como é feita a prestação de contas.

* **Condôminos**

Com relação ao atendimento as reuniões convocadas, o prazo mínimo estipulado por lei é demasiadamente longo para alguns, o que faz com que muitos percam a noção da data, mesmo que notificados via carta e que sejam afixados avisos nos murais.

Quanto a prestação de contas, o problema consiste em uma discriminação mais detalhada dos gastos que vem resumidos no boleto da taxa de condomínio, além do curto espaço de tempo para examinar os balancetes durante as reuniões, visto que não é viável a confecção e distribuição para cada condômino.

Com isso, é necessário melhorar o sistema de convocação de reuniões, a maneira como a prestação de contas é feita e por fim, seria bem recebida uma otimização no método de reserva dos espaços comuns (salão de festa, churrasqueiras, quadra de esporte) bem como no de correio, visto que por vezes temos problemas com atraso na entrega de encomendas e afins por desencontro de informações.

* **Funcionários**

A maioria dos problemas relacionados a funcionários está na questão do gerenciamento das entregas de correio visto que as correspondências e encomendas após recebidas na portaria são distribuídas nas caixas de correio ou ficam aguardando a retirada. Como alguns moradores passam o dia fora, torna-se difícil, algumas vezes, proceder determinadas entregas. Portanto, é necessário que o processo de gerenciamento do correio seja aperfeiçoado para dirimir os problemas atuais.

**Arquitetura e justificativa**

Para o desenvolvimento deste projeto será adotada a arquitetura cliente servidor de três camadas, sendo escolhida a plataforma da Oracle, que inclui o banco de dados 12C; e servidor web Apex Listener, em conjunto com a ferramenta Application Express, também da Oracle. Essa combinação permitirá o desenvolvimento do sistema no modelo SaaS de maneira ágil e prática, visto que toda a infraestrutura já é disponibilizada pela Oracle sem que haja a necessidade de executar quaisquer tarefas de configuração ou integração.

As principais razões dessa escolha estão no fato da dispersão de conhecimentos em temos de linguagens e paradigmas de programação do grupo. Com essa escolha o esforço de aprendizagem se resume a lidar com a ferramenta e com PL/SQL, não se retirando o foco do emprego das metodologias do desenvolvimento ágil.

Quanto a ferramenta, vale destacar que todo o trabalho é desenvolvido já na nuvem da Oracle, em ambiente disponibilizado pela própria empresa, porém é possível configurar seu próprio ambiente bastando ter um servidor com o banco de dados Oracle configurado.

Complementarmente ao citado, utilizamos o GitHub para centralizar a documentação, e o Trello para fazer o acompanhamento do desenvolvimento do trabalho (tarefas e etapas desenvolvidas).

**SPIKES**

A principal decisão do projeto girou em torno da escolha da plataforma da Oracle para o desenvolvimento. Até que esta decisão fosse efetivamente tomada, foram feitos testes no Heroku com Django, treinamento em outras ferramentas e ao fim se concluiu pela inviabilidade do uso destas tecnologias, citadas por último, devido a disparidade tanto no nível de conhecimento dos componentes do grupo, bem como na previsão de que o tempo necessário para que todos atingissem um nível de razoável de domínio que permitisse o desenvolvimento independente seria muito grande, o que comprometeria o desenvolvimento e até mesmo a entrega do trabalho.

**Planejamento das iterações**

As iterações foram planejadas de forma que as etapas previstas só fossem vencidas com o consentimento e validação do cliente. Para tanto, estabelecemos um método no qual a construção do artefato inicial da etapa fosse feita com o cliente, estudada e aperfeiçoada pelo grupo e por fim validada/modificada em nova reunião com o cliente.

Isso foi aplicado nas histórias de usuários, protótipos LO-FI, *storyboards*, bem como na conclusão das histórias de usuário, momento no qual foi verificado se o funcionamento de fato estava de acordo com o que havia sido pensado.

Com relação aos clientes, além de reuniões presenciais previamente marcadas, também foi estabelecido contato direto via WhatsApp para breves esclarecimentos a respeito do desenvolvimento.

Quanto ao desenvolvimento *interna corporis*, o grupo estabeleceu reuniões semanais para discutir e acompanhar o desenvolvimento, tirar dúvidas e fazer as entregas previstas. De modo resumido o planejamento da iteração envolveu as seguintes etapas:

* Histórias de usuário;
* Protótipos LO-FI UI;
* *Storyboards*;
* Tarefas;
* Desenvolvimento (TDD);
* Teste de validação; e
* Validação do cliente.

Com isso, realizamos o trabalho em duas iterações, onde na primeira, constituída de dez histórias, realizamos nove, e na segunda, com apenas três, completamos todas.

**Histórias de usuário**

As seguintes histórias de usuário foram escritas em conjunto com o cliente, sendo refinadas pela equipe de desenvolvimento e revalidadas com o cliente que nesse momento designou valores de um a três, sendo um menos relevante e três mais relevante, para cada história.

Após esta fase de construção inicial, as histórias foram classificadas em conjunto pelos desenvolvedores, sendo posteriormente distribuídas entre os desenvolvedores, que por sua vez definiu um tempo estimado de desenvolvimento em horas/semanas, com cada semana composta por vinte horas distribuídas em sete dias.

**1ª Iteração (20 pontos | 18 completos)**

* **Danilo**

1. **Autenticação de usuário**

            Como usuário eu quero acessar o sistema através de um método de autenticação. Para garantir a segurança das informações disponibilizadas e do uso do meu perfil.

* Classificação cliente (relevância): 2
* Classificação desenvolvedor (dificuldade): 2
* Tempo estimado de desenvolvimento (semanas): 1

1. **Controle individual de taxas de condomínio**

            Como morador eu quero visualizar as minhas faturas das taxas condominiais. Para controlar os pagamentos e confirmar as quitações já realizadas.

* Classificação cliente (relevância): 1
* Classificação desenvolvedor (dificuldade): 3
* Tempo estimado de desenvolvimento (semanas): 1

* **Vitor**

1. **Controle de reserva do salão multiuso**

            Como morador eu quero fazer a reserva do salão multiuso. Para garantir a disponibilidade do espaço em uma data específica.

* Classificação cliente (relevância): 1
* Classificação desenvolvedor (dificuldade): 2
* Tempo estimado de desenvolvimento (semanas): 1

1. **Cadastro de receitas e despesas**

            Como membro da administração eu quero cadastrar as despesas e receitas do mês. Para gerenciar o controle fiscal das contas do condomínio.

* Classificação cliente (relevância): 3
* Classificação desenvolvedor (dificuldade): 3
* Tempo estimado de desenvolvimento (semanas): 1

* **Vinícius**

1. **Controle de correio**

            Como funcionário eu quero cadastrar as correspondências dos moradores. Para controlar a distribuição dos itens recebidos.

* Classificação cliente (relevância): 1
* Classificação desenvolvedor (dificuldade): 1
* Tempo estimado de desenvolvimento (semanas): 1

1. **Notificação de correspondência**

            Como usuário eu quero receber uma notificação de correspondência aguardando retirada. Para poder proceder no recebimento do item.

* Classificação cliente (relevância): 1
* Classificação desenvolvedor (dificuldade): 1
* Tempo estimado de desenvolvimento (semanas): 1

* **Thiago**

1. **Cadastrar usuários do sistema**

            Como administrador eu quero gerenciar o acesso de usuários. Para definir o perfil de acesso do indivíduo.

* Classificação cliente (relevância): 3
* Classificação desenvolvedor (dificuldade): 1
* Tempo estimado de desenvolvimento (semanas): 1

1. **Transparência das contas do condomínio**

            Como morador eu quero visualisar o relatório financeiro e o balanço do exercício. Para exercer a fiscalização das contas do condomínio.

* Classificação cliente (relevância): 2
* Classificação desenvolvedor (dificuldade): 3
* Tempo estimado de desenvolvimento (semanas): 2

* **Gabriel**

1. **Possuir mural de avisos destinado a veiculação de informações diversas**

            Como usuário eu quero anunciar mensagens em um mural de avisos virtual. Para que eu consiga a colaboração e/ou atenção dos demais usuários.

* Classificação cliente (relevância): 1
* Classificação desenvolvedor (dificuldade): 2
* Tempo estimado de desenvolvimento (semanas): 1

1. **Possuir um meio de validação da prestação de contas da administração (Cancelada)**

            Como membro do conselho fiscal eu quero validar as contas prestadas pela administração. Para garantir a gestão efetiva dos recursos financeiros do condomínio.

* Classificação cliente (relevância): 2
* Classificação desenvolvedor (dificuldade): 2
* Tempo estimado de desenvolvimento (semanas): 1

**2ª Iteração (8 pontos | 8 completos)**

* **Danilo e Vinicius**

1. **Redefinição de senha**

Como usuário eu quero redefinir minha senha para acessar o sistema quando não lembrar mais da senha.

* Classificação cliente (relevância): 2
* Classificação desenvolvedor (dificuldade): 2
* Tempo estimado de desenvolvimento (semanas): 2
* **Gabriel e Vinicius**

1. **Esquema de autorização**

Como cliente eu quero que o sistema tenha suas funcionalidades e páginas disponibilizadas para perfis de acesso compatíveis para poder assegurar o correto funcionamento e interação dos usuários com o sistema.

* Classificação cliente (relevância): 2
* Classificação desenvolvedor (dificuldade): 3
* Tempo estimado de desenvolvimento (semanas): 2
* **Thiago e Vitor**

1. **Cadastro de funcionário**

Como administrador eu quero cadastrar funcionários no sistema para que controlar as informações destes e permitir seu acesso de maneira que possam executar algumas funções através deste.

* Classificação cliente (relevância): 2
* Classificação desenvolvedor (dificuldade): 3
* Tempo estimado de desenvolvimento (semanas): 2

**Relatório de experiência**

O grupo procurou se ater somente a metodologia XP, porém fazendo pequenas modificações para se adequar melhor a realidade de trabalho, tais como a definição de uma semana de 20 horas e a flexibilização do número de pessoas no desenvolvimento de uma funcionalidade, onde tivemos funcionalidades desenvolvidas por uma, duas ou três pessoas a depender da complexidade.

No mais, as funcionalidades em geral puderam ser desenvolvidas de maneira independente e por terem tamanhos, em regra, razoáveis, possibilitaram manter uma integração contínua entre cliente e desenvolvedores durante o desenvolvimento.

Também foi adotado um padrão de codificação que foi seguido por todos e tornou o acompanhamento do desenvolvimento do código muito mais simples, por vezes dispensando requisitar mais explicações quanto a intenção do programador. O padrão consistiu, em codificação utilizando Camel Case, porém com algumas modificações no que toca a construção do código PL/SQL para se adequar melhor a realidade. Com isso, nomes de funções, e procedures foram padronizados, bem como os cabeçalhos de usas definições e os comentários (em inglês).

Na opinião geral do grupo, o trabalho com a metodologia foi uma experiência muito proveitosa, principalmente no que toca ao desenvolvimento com TDD, que de fato provou sua eficiência no desenvolvimento do código, que em regra tendeu a funcionar sem que fossem detectados comportamentos inesperados.

**Lições aprendidas**

Algumas lições foram aprendidas durante o desenvolvimento deste projeto. A primeira delas é a de que para desenvolver um trabalho, seria melhor que os grupos fossem formados por afinidade com as tecnologias.

Não obstante a primeira lição, é possível conduzir um processo de desenvolvimento pleno se as tecnologias/arquitetura escolhidas tiverem uma baixa curva de aprendizagem.

A terceira lição aprendida está em tentar agregar valor ao projeto desenvolvido com diferenciais, nos casos em que um produto já exista no mercado. Assim, achamos válido pesquisar outros produtos semelhantes antes do desenvolvimento de forma a propor inovações para um cliente específico.

A quarta lição é tentar combinar um padrão de testes de unidade, onde possam ser identificados elementos comuns a serem testados e se estabeleça o padrão para a organização dos testes.

A quinta lição que achamos relevante é a de utilizar o máximo de ferramenta que estiver sendo utilizado. Poderíamos ter concentrado todo o projeto dentro do Apex, mas decidimos explorar outras para expandir os conhecimentos, porém durante o projeto surgiram alguns problemas causados pelo uso de várias ferramentas ao mesmo tempo, como manter os arquivos atualizados no GitHub.