

Proposta de Trabalho para Projeto de TCC

Igor Santos

26 de novembro de 2014

1 Proposta de Projeto de TCC

Semestre:	2
Nome do Aluno:	Igor Gomes dos Santos
Matrícula:	2011.01.12519-5
Professor Orientador:	José Carlos Millan
Título Provisório do Projeto:	Insert Coin

1.1 Introdução

O projeto a ser desenvolvido é um Sistema Pessoal de Planejamento Financeiro, ou de Controle Orçamentário, ou ainda, um Administrador de Finanças. Seu objetivo é transformar a organização financeira de uma pessoa ou de uma residência, facilitando o manuseio e controle de entrada e saída do dinheiro e das contas bancárias, bem como do cartão de crédito – neste, alterando a forma como o usuário comumente vê o “plástico”.

Em verdade, o principal diferencial do produto em relação à concorrência atual no mercado será o controle de cartões de crédito. O brasileiro tem o costume de considerar o cartão como um “gasto”, uma despesa que atrapalha o bom andamento do orçamento familiar, sendo assim um “mal necessário” devido aos seus benefícios no adiamento de compras.

O objetivo do **Insert Coin** nesse sentido é auxiliar o usuário no entendimento dos seus cartões de crédito, e integrá-los ao seu fluxo de caixa normal. Tal fluxo é comumente baseado na entrada de poucas fonte de renda (de alto volume), e a saída em diversas despesas mensais (de volume menor). O cartão de crédito, por ser um concentrador de despesas que adia o pagamento de diversos eventos para um único dia, tende mais confundir do que a ajudar. O usuário precisa compreender bem a dinâmica dos dias de fechamento e de abertura de nova fatura, do “melhor dia para compra”, o dia do vencimento e quais são os valores possíveis de pagamento – e quais as consequências de não pagar a fatura em cheio. Além disso, o lançamento de despesas ocorre com diversos dias de diferença entre o evento e seu aparecimento na fatura parcial. Tudo isso

vai muito além do fluxo simples de sua conta-corrente e seu cartão de débito, que consiste em receber o salário do mês, e lançamentos instantâneos do uso do cartão de débito.

1.2 O Projeto

Inicialmente chamado de **Insert Coin**, ele é principalmente um sistema baseado no lançamento de entradas e saídas de uma ou mais contas de finanças. Ele se assemelha à organização de um Livro-Caixa empresarial, onde o usuário marcará todos os valores recebidos e gastos de uma determinada conta.

Atualmente existem diversos sistemas no mercado, tanto web quanto móvel, que atendem a esse tipo de usuário, auxiliando na visualização e controle das despesas mensais. Boa parte deles tem o mesmo princípio, inclusive. Considerando esse ponto de vista, o sistema será pouco diferente do que já existe no mercado e, por exemplo, não incentivará a migração dos usuários de concorrentes para o nosso projeto.

Por outro lado, somente um dos sistemas encontrados tenta suprir as peculiaridades do mercado de cartões de crédito nacional, de forma muito vaga e primariamente informativa. O método a ser utilizado aqui é bem mais ativo: ele auxilia o usuário a entender como o cartão de crédito afeta seu fluxo de caixa e insere, como despesas comuns, os eventos nos quais o cartão foi utilizado. O objetivo é evitar a segregação comum que o brasileiro faz em considerar o cartão de crédito com uma mais uma conta/boleto a pagar, como é a eletricidade ou a água da casa. No cartão são feitos gastos assim como ocorrem na conta de débito: lazer, vestuário, transporte, educação, etc. Portanto, porque devemos pensar em “reduzir os gastos do cartão” ao invés de “reduzir os gastos mensais”?

Trabalhando de forma unificada será mais fácil para o usuário compreender o destino do dinheiro daquele mês. Também será possível começar a planejar e organizar metas de redução dos custos, ou objetivos de economia de forma eficiente e focada. Essa unificação ocorrerá a partir da visualização das despesas daquele mês com relação às entradas; o usuário não precisará se perder em filtros de datas distintos entre a conta-corrente e o cartão de crédito – que naturalmente possuem fluxos temporais diferenciados. O sistema também evitará a dúvida comum “será que já posso usar o cartão de crédito?”, auxiliando o usuário a entender as datas importantes de seus cartões, e as inserindo na rotina que ele já conhece e compreende – a de sua conta corrente e salário.

O que mais cria dúvidas no cartão de crédito é que sua fatura é finalizada cerca de 10 dias antes do dia de pagamento. Aquele período é quando o usuário não está certo de onde

as despesas se encaixam melhor: se no mês atual ou no seguinte. Essa confusão se acentua pois as datas da fatura dificilmente encapsulam o período de mês com o qual o usuário está financeiramente acostumado: seu fluxo mensal de entrada do salário. Se ele decide colocar o vencimento do cartão para uma data próxima de seu salário, o fechamento da fatura fica distante do final do mês; se decide fechar a fatura no final do mês, o pagamento fica distante das outras contas que ele comumente paga. E muitas vezes algumas despesas de uma determinada só serão pagas dali a dois meses, o que gera confusão e certa negligência na hora de conferir a origem dos gastos.

A proposta é coletar as datas importantes do cartão e orientar o usuário para que, sempre que possível, a fatura seja paga em cheio no dia do vencimento (em débito automático, por exemplo). Esta é, de acordo com os especialistas da área, a melhor forma de conviver com o cartão, pois não gera juros adicionais em cima do que não foi pago. A partir daí, o usuário terá configurada sua conta-corrente do sistema para que as despesas que ocorrerem no cartão de crédito sejam visualizadas simultaneamente às da conta, no período que corresponda ao pagamento da fatura. Conforme vão surgindo novas despesas no cartão de crédito, o usuário poderá visualizar diretamente seu o impacto no orçamento dos meses seguintes. Também será possível entender de forma prática a fatura do cartão do mês em visualização, juntamente com as contas a serem pagas e as despesas comuns em débito – afinal de contas, é na conta-corrente que a ação financeira acontece.

1.3 Problemas a serem resolvidos

- Desconhecimento de quanto dinheiro é recebido por mês, e quanto é gasto;
- Desconhecimento das despesas mensais;
- Falta de direcionamento e de objetivos financeiros, ocasionados pela desorganização;
- Dificuldade de adequação das faturas fixas do cartão de crédito ao fluxo de caixa tradicional do brasileiro;
- Problemas no entendimento das datas importantes do cartão de crédito;
- Ocultação de diversas despesas, tanto ocasionais como mensais, que passam a fazer parte do orçamento familiar como uma única entidade, “a fatura do cartão de crédito”, que tem como características mais comuns sua desinformação e grandiosidade;

- Descoberta de “despesas-surpresa”, que não foram consideradas até a chegada da fatura do mês – e portanto, geram readequação do orçamento do mês; possivelmente, com o corte de algum gasto comum ou mesmo com endividamento temporário.

1.4 Vantagens para o usuário

- Entendimento simples do fluxo de caixa pessoal e da própria situação financeira;
- Dados reais para preparação de metas de gastos e objetivos de economia;
- Libertação do “pesadelo do cartão de crédito”, que passa a funcionar como uma conta-corrente fictícia e atrelada à real, ao invés de ocultar as despesas que no orçamento pessoal aparece como mais uma conta da casa;
- O cartão de crédito passa a ser um aliado da casa, gerando conforto com os benefícios em serviços que costumam trazer, e não mais as dores-de-cabeça tradicionais por causa das despesas não-previstas.

1.5 Aprendizado para o aluno

O projeto será desenvolvido baseado nos conhecimentos prévios do aluno em sistemas distribuídos, interfaces web e móveis, e organização de dados. O objetivo técnico do sistema é conciliar a experiência pessoal em controle de finanças com o desafio prático de um projeto real e de grande porte nas áreas citadas. Ele unificará diversas áreas que originalmente só foram vistas pelo aluno em separado – como desenvolvimento de servidor de aplicação distribuída, aplicação consumidora de API, e aplicativo móvel com acesso a informações da internet.

O resultado do projeto será um sistema sólido e fluido, que permitirá aos usuários a escolha do melhor ambiente para utilização de acordo com seu estado e necessidade atuais, proporcionando elevada experiência de uso. Pelo lado do aluno, isso representará um grande desafio no que tange a unificação de interfaces distintas sob um mesmo ideal de sistema, e sob uma mesma base de dados e sistema de comunicação.

O sistema exigirá documentação de alto nível, para possibilitar o desenvolvimento a médio e longo prazos. Portanto, também haverá um trabalho intenso no que tange a análise dos diversos subsistemas que irão compor o projeto, bem como a documentação prévia, corrente e posterior do trabalho desenvolvido.

1.6 Metodologia e Tecnologias a serem utilizadas

O projeto será elaborado utilizando as seguintes ferramentas:

1.6.1 Para a documentação

- Análise dos requisitos e objetivos do projeto em $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ (para portabilidade e flexibilidade de desenvolvimento dos documentos);
- Diagramas UML (para fluxos de dados, estados, e das aplicações).

1.6.2 Para a administração do projeto

- Código e documentos disponíveis em repositórios versionados no Github¹;
- Organização e priorização de tarefas e milestones integrados ao repositório.

1.6.3 Para o desenvolvimento

Servidor de comunicação distribuída:

- Baseado em *REST*² e desenvolvido em PHP com a biblioteca *Restler*³;
- Banco de dados *MySQL* ou *MongoDB*.⁴ A decisão dentre um dos dois dependerá de análise futura, de acordo com o formato mais adequado para transporte e armazenagem dos dados de fluxo de caixa.

Interface de usuário web:

- JavaScript com *framework MV*⁵ – provavelmente o *Ember*, por fornecer tecnologias avançadas de interface, aliadas ao processamento facilitado de dados remotos via *REST*;

¹Sistema online de repositórios Git.

²*Representational State Transfer* - Transferência de Estado Representativo, modelo de abstração de computação distribuída baseado em hipermídia.

³Biblioteca PHP que abstrai os recursos *REST* para conceitos tradicionais de Orientação a Objetos.

⁴*MySQL* é um banco de dados relacional, enquanto o *MongoDB* é não-relacional.

⁵Sigla “guarda-chuva” que abriga diversos formatos de organização de framework originalmente baseados no modelo *MVC*, mas que usam outros tipos de componentes para cumprir tarefas de roteamento, controle de requisições e comportamentos do usuário.

- Folhas de estilo *Less*⁶ (para melhor manutenção e organização);
- *Layout* fluido e responsivo, para que a utilização do sistema seja agradável tanto em um computador *desktop*, *tablet*, ou celular, mesmo sem utilizar um aplicativo móvel dedicado;
- Capacidade de trabalho *offline* e de sincronização após reconexão, a partir de bibliotecas de armazenamento *client-side* (para em *browsers modernos*).

Interfaces de usuário móvel:

Em primeira instância, o aplicativo web suprirá a maioria dos Casos de Uso móveis. Para além deles, e para a melhoria da *UX*⁷, um aplicativo nativo será criado.

Será utilizada uma plataforma de desenvolvimento única para múltiplos sistemas operacionais (que suporte *iOS* e *Android*, e talvez *Windows Phone* e *Blackberry*). A plataforma ainda não foi definida, mas o principal candidato é o *Appcelerator Titanium*, por permitir o uso de uma só linguagem de desenvolvimento (JS, além do uso opcional de XML para os templates) e a compilação posterior em aplicativos verdadeiramente nativos – diferente de outras plataformas que se baseiam em *WebViews*.

Infraestrutura:

Para execução do servidor de *REST*, Banco de Dados e servidores adicionais de suporte (como servidores de *cache* e *load-balancing*) serão usados um ou mais *PaaS*^{8,9}, de forma a facilitar a manutenção e escalabilidade do sistema. Lidando com a infraestrutura assim também é possível focar mais no processo de desenvolvimento, diminuindo as preocupações desde o início com o hardware e as necessidades dos servidores.

⁶*Less* é um pré-processador de CSS que tem como principal funcionalidade a possibilidade de aninhar referências, de modo a diminuir a repetição de código e evitar erros, facilitando também a manutenção e o entendimento da estrutura.

⁷*User Experience*: uma disciplina de *Interação Humano-Computador* sobre o comportamento, atitudes e emoções relacionados ao uso de um produto ou serviço. Inclui os aspectos práticos, afetivos, significativos, de experiência e de valor da interação homem-máquina.

⁸*Platform-as-a-Service*: tipo de serviço sob demanda onde o cliente paga para ter uma plataforma de desenvolvimento já preparada e pronta para uso, necessitando somente preencher com os requisitos do projeto e códigos – ou com a estrutura de banco de dados, etc.

⁹“Um ou mais” pois nem sempre um serviço só supre todas as necessidades do sistema, e portanto diferentes provedores são utilizados para se obter um ambiente completo.