

# Projeto de TCC - Insert Coin

Igor Santos

12 de dezembro de 2014

# *Sumário*

<b>1</b>	<b>Proposta do Projeto de TCC</b>	p. 4
1.1	O Projeto . . . . .	p. 4
1.2	Método de Trabalho . . . . .	p. 6
1.3	Previsão de Alocação de Recursos . . . . .	p. 7
	Recursos Humanos . . . . .	p. 7
	Recursos Materiais . . . . .	p. 7
1.4	Cronograma de Trabalho . . . . .	p. 8
<b>2</b>	<b>A Empresa e o Negócio</b>	p. 9
2.1	Histórico . . . . .	p. 10
2.2	Mercado Consumidor . . . . .	p. 10
2.3	Concorrência . . . . .	p. 11
2.4	Premissas e Restrições ao Projeto . . . . .	p. 11
<b>3</b>	<b>Os Sistemas Atuais</b>	p. 14
3.1	Principais Concorrentes . . . . .	p. 14
3.2	Outros Sistemas . . . . .	p. 15
3.3	Motivações para o Novo Sistema . . . . .	p. 16
	3.3.1 Problemas dos Sistemas Atuais . . . . .	p. 17
	3.3.2 Situação Desejada . . . . .	p. 17
<b>4</b>	<b>O Sistema Proposto</b>	p. 18
4.1	Requisitos do Sistema . . . . .	p. 18

4.1.1	Design . . . . .	p. 18
4.1.2	Implementação . . . . .	p. 18
4.2	Casos de Uso . . . . .	p. 20
4.2.1	Especificações . . . . .	p. 21
4.3	Diagrama de Topologia . . . . .	p. 24
4.4	Modelo Conceitual de Classes . . . . .	p. 25
4.5	Modelo Conceitual de Dados . . . . .	p. 27
<b>Bibliografia</b>		p. 28

# *1 Proposta do Projeto de TCC*

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 fontes 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 por completo. 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.

## **1.1 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. Portanto, se considerarmos somente esta *feature*, 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. Para remediar isso tentaremos inovar na interface e na facilidade de entrada de dados, que é um dos grandes obstáculos para a introdução deste tipo de sistema na rotina do usuário.

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 gastos, 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.

## 1.2 Método de Trabalho

O desenvolvimento do sistema será gerido a partir da área de *Issues* do **Github**. Lá é possível criar tarefas, categorizar (a partir de *tags*), delegar e ordenar temporalmente. O andamento do projeto será acompanhado a partir de *milestones* configurados no mesmo subsistema, todos com claras datas de finalização.

O **Github**, além de possuir um subsistema de tarefas, serve principalmente para armazenar na *cloud* um servidor centralizado de *git*. Dessa forma temos automaticamente um sistema eficiente de backup do código do sistema. Ele também participa do ambiente auxiliando a entender o histórico de desenvolvimento no decorrer do tempo; o *git* armazena todas as alterações de código que forem feitas, o que nos ajuda a analisar nossa produtividade e facilmente encontrar os responsáveis por partes do sistema – útil, por exemplo, quando ocorrer um *bug* inesperado e que precisa ser rapidamente corrigido.

Por fim, para organizar o fluxo de código, será utilizado o *GitFlow*[1] uma metodologia de versionamento que organiza as *tags* de um repositório de forma lógica. A organização precisa desse sistema nos permite criar um fluxo contínuo de *deploys*; podemos configurar nossos servidores de produção para *escutarem* a *tag master* e sempre que houver novos *commits*, eles podem importar a nova versão - isso é feito a partir de *git hooks*, ou automaticamente por alguns sistemas de *PaaS*.

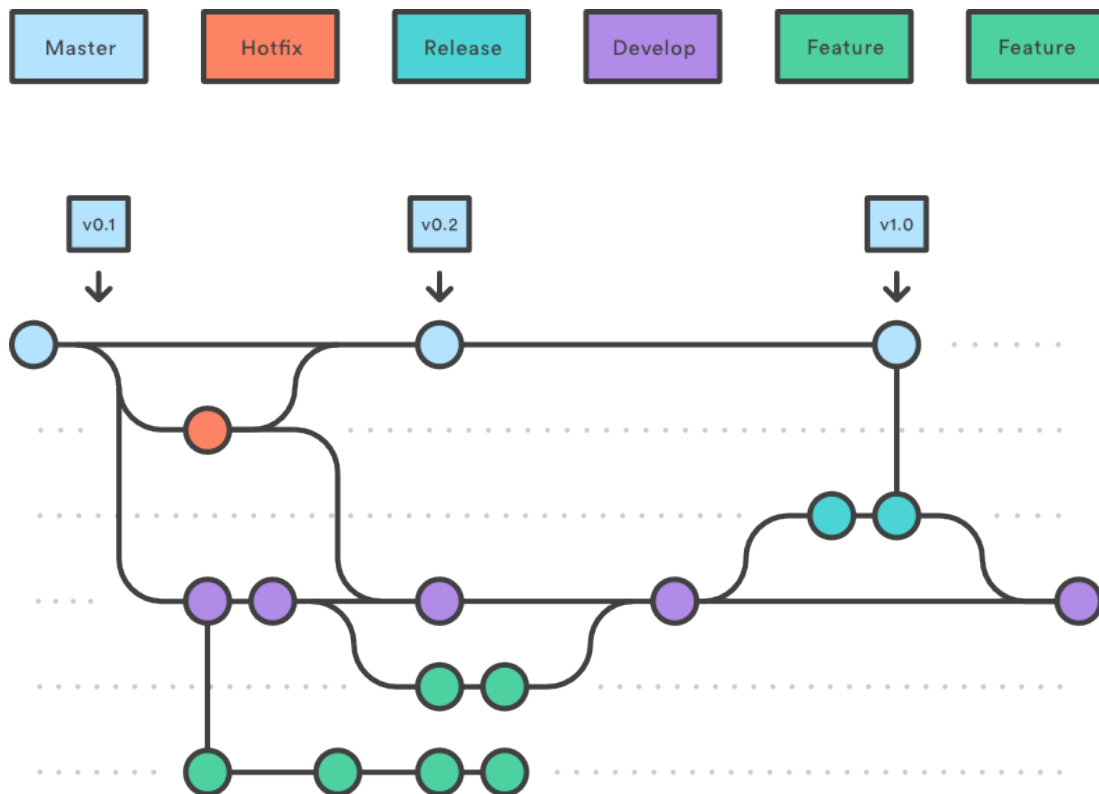


Figura 1.1: Exemplo de como *GitFlow* funciona, exibindo os diversos *branches* da metodologia.

## 1.3 Previsão de Alocação de Recursos

### Recursos Humanos

- 1 Analista de Sistemas Sênior;
- 1 Designer multi-plataforma (*web* e móvel);<sup>1</sup>
- 1 Analista de marketing *freelancer* (para o marketing inicial do produto; futuramente, integrará a equipe permanente).

### Recursos Materiais

- Instâncias no *Heroku* (é possível começar com pouco e expandir facilmente no futuro[2]) com as seguintes configurações:
  - *Dynos web* (máquinas que processam *requests* web) rodando **PHP 5.5**;
  - *Dynos de background*
  - Servidor **PostgreSQL**, também facilmente escalável e seguro[3];

- Servidor de Cache HTTP **Varnish** em cada *dyno* para otimizar os requests estáticos ou repetidos;
- Os seguintes add-ons a partir da plataforma do *Heroku*:
  - Servidor de *Memcache* **MemCachier** para caches temporários de aplicação;
  - **Mandrill by MailChimp** para envio de e-mails de lembretes;
  - **NewRelic** para monitoria da performance da aplicação e do banco de dados;
  - **Rollbar** para monitoria de erros nas aplicações.

## 1.4 Cronograma de Trabalho



Figura 1.2: Gráfico de Gantt, descrevendo o planejamento do projeto a partir de sua linha do tempo



## 2 *A Empresa e o Negócio*

O objetivo do projeto é entrar num mercado já saturado de aplicativos similares com uma lista de diferenciais para atrair os consumidores que ainda não utilizam sistemas automatizados de gestão financeira ou se sentem insatisfeitos com as soluções que usam atualmente. Para isso, a equipe de desenvolvimento focará numa usabilidade simples e no menor atrito possível para entrada de novos lançamentos. E, por outro lado, o diferencial em funcionalidades terá como carro-chefe o tratamento do cartão de crédito de forma diferenciada das contas corrente e poupança, como nenhum aplicativo do mercado faz atualmente.

Todos os responsáveis pelo projeto tem larga experiência prática em diversos métodos de organização financeira e são usuários de diferentes sistemas do mercado. Portanto, todos são hábeis de opinar e desenvolver o projeto da melhor forma possível, com boas sugestões de funcionalidades e ao mesmo tempo evitando os erros (de acordo com nosso ponto de vista) já cometidos pela concorrência.

O sistema funcionará no modo *freemium*, onde os usuários terão acesso a um grupo limitado de funcionalidades e precisarão pagar mensalmente caso desejem utilizar as funções avançadas. A lista final do que será de uso gratuito e do que será pago só será de fato conhecida quando o escopo do projeto for fechado; no entanto, segue um exemplo de algumas funcionalidades mais complexas e que fazem sentido entrar na lista *premium*:

- Gráficos e análises financeiras sobre os gastos mensais e anuais
- Ter mais de duas contas (sendo uma corrente e uma poupança<sup>1</sup>)
- A possibilidade de planejar o pagamento parcial de uma fatura de cartão de crédito
- Uso do aplicativo móvel em modo offline, permitindo sincronia de dados quando o aparelho entrar numa conexão *WiFi*

---

<sup>1</sup>Isso caracterizaria um usuário avançado, visto que o caso mais comum é o brasileiro ter uma conta-corrente onde recebe seu pagamento (ou uma conta-salário e uma corrente), e talvez uma poupança onde ele guarda suas economias, e diversos cartões de crédito.

- Integração entre mais de um usuário numa mesma conta – por exemplo, um casal usando o mesmo sistema a partir de dois *logins* diferentes

## 2.1 Histórico

A empresa responsável pelo **Insert Coin** foi criada especialmente para o projeto, a partir da sintonia de ideias entre o Analista e o Designer a cargo. A ideia surgiu quando o Analista se sentia insatisfeito com o sistema de gestão que utilizava e não conseguiu encontrar substituto eficaz no mercado. O que mais lhe incomodava era não ser fácil controlar os gastos em cartão de crédito, pois as aplicações disponíveis só possuíam a distinção entre múltiplas “contas”, e o ciclo de vida de uma fatura acabava por ficar misturado com o fluxo de caixa da conta-corrente, por exemplo. A ideia foi fomentada aos poucos e então surgiu a oportunidade de, junto ao seu colega de graduação, iniciar a estruturação e o desenvolvimento do projeto.

## 2.2 Mercado Consumidor

O público-alvo do projeto tem uma ou mais de cada uma das seguintes características:

### 1. Em relação às suas finanças:

- Nunca se informou sobre educação financeira antes, mas começou a se interessar sobre o assunto;
- Já tem um certo nível de organização financeira, mas ela é falha ou rudimentar;
- Tem suas planilhas eletrônicas financeiras, mas são complexas de usar e engessadas;
- Encontra problemas de se manter no azul, e está em busca de soluções para sua falta de controle;

### 2. Em relação à tecnologia:

- Possui um smartphone e tem um uso aceitável do mesmo, mas não usa o computador com frequência;
- Possui ou não um smartphone e se sente confortável navegando na Internet;
- Possui ou não um smartphone e utiliza planilhas eletrônicas para controlar suas finanças;

Ou seja, nosso principal mercado consumidor tem algum contato com tecnologia e se sente confortável a usá-la, seja no computador ou num dispositivo móvel, e tem algum interesse em educação e organização financeira, seja por dificuldades anteriores ou por puro interesse pessoal.

## 2.3 Concorrência

Atualmente no mercado há diversas empresas focadas em produtos similares ao que propomos neste projeto. No entanto, foi identificado que nenhum deles oferece todas as funcionalidades que planejamos implementar, o que significa que, após o lançamento vamos estar à frente da concorrência e aptos a conquistar novos clientes, sejam estes experientes em outros produtos ou interessados em experimentar algo novo.

Listamos aqui as principais empresas concorrentes até o momento:

<b>Granatum</b>	<b>Organizze</b>	<b>Mint (americano)</b>
Yupee	MyFreeComm Finance	Manubia
Monefy	<i>Outros aplicativos móveis similares</i>	

Foi notado que algumas empresas possuem parceria com algumas instituições bancárias, o que em tese poderia trazer mais clientes destes bancos aos seus aplicativos. No entanto, não há notícia de divulgação destes sistemas dentro das agências ou em outros veículos relacionados aos bancos. Isso nos leva a crer que a parceria foi criada para iniciar um processo de integração bancária, mas por motivos alheios ao conhecimento geral, tal integração não evoluiu com outros bancos. A única exceção a essa regra é o *Mint*, mas que não representa concorrência direta por não fazer parte do mercado nacional. Apesar disso, ele é listado aqui para servir de exemplo sobre como a vida do usuário pode ser facilitada; o sistema do *Mint* possui integração com uma longa, longa lista de instituições financeiras dos Estados Unidos e do Canadá. Isso permite que o usuário não precise criar novas entradas manualmente para se organizar – o sistema busca, em *background*, as informações diretamente com os bancos e administradoras de cartões.

## 2.4 Premissas e Restrições ao Projeto

Os principais objetivos do projeto, a nível de funcionalidades, seguem abaixo. A lista está ordenada aproximadamente por prioridade de implementação, e os itens enfatizados provavelmente farão parte da lista de recursos avançados, oferecidos somente aos usuários *premium*:

1. Lançamento de despesas diárias e visualização em estilo de planilha;
2. Divisão em contas distintas, com tratamento especial para cartões de crédito;
3. Categorias personalizadas de lançamentos, com cores para facilitar identificação;
4. Categorização opcional por meio de pagamento (boleto, cartão, transferência, etc);
5. Inclusão de lançamentos parcelados e periódicos;
6. Filtragem de despesas por mês ou período de livre escolha;
7. *Acompanhamento de metas de valores mensais para categorias definidas;*
8. Planejamento avançado dos próximos meses, a partir dos lançamentos futuros e das metas pretendidas;
9. Tela inicial personalizável, com blocos de resumos, lançamento rápido, saldos, etc;
10. Aplicativo móvel, para facilitar a visualização do orçamento e o registro imediato de despesas feitas na rua. Ele precisará de uma interface inovadora na tela de novo lançamento, de modo a diminuir ao máximo a quantidade de interações necessárias – dessa forma, reduzindo o atrito dessa ação frequente para o usuário;
11. Notificações de lançamentos a vencer via e-mail e no aplicativo móvel;
12. *Resumo histórico, para um entendimento mais amplo da vida financeira do usuário;*
13. *Gráficos de análise financeira por período, categorias, e contas;*
14. Integração com sistema de relatórios de erros e pedidos de novos recursos pelos usuários;
15. Sistema de importação simplificada dos arquivos exportados pelos principais bancos;
16. *Leitor OCR<sup>2</sup> inteligente, para capturar informações de um novo lançamento a partir de uma foto do comprovante de crédito/débito, tirada com o celular;*
17. Equiparação de funcionalidades do sistema web com o aplicativo móvel, de forma que os usuários possam ter liberdade total de usar somente o dispositivo móvel se preferirem;
18. *Integração com sistemas bancários*

---

<sup>2</sup>Reconhecimento Ótico de Caracteres: tecnologia que traduz uma imagem de texto em palavras

Esta lista de funcionalidades previstas para o sistema não é definitiva, o que significa que no decorrer do projeto há a possibilidade de novas ideias surgirem e serem realocadas no planejamento, de acordo com a demanda. Além disso, devido ao seu tamanho, foi decidido que os recursos numerados de 10 em diante serão desenvolvidos numa fase futura do sistema – portanto, fora do escopo deste Trabalho de Conclusão de Curso.

Dito isto, é notável que a maior restrição para o projeto será o fator temporal. O objetivo é que o **Insert Coin** seja um sistema em eterno andamento, com o lançamento gradual de novas funcionalidades. Esse comportamento ajuda a manter os usuários fiéis e um fluxo contínuo de novos clientes. A fidelidade se dará pelos seguintes pontos:

- Saber que os problemas encontrados serão resolvidos em breve;
- O usuário será sempre o foco da empresa, que estará sempre adicionando novos recursos para o melhor aproveitamento de suas informações financeiras;
- Ele poderá participar indiretamente do desenvolvimento do produto, através do *feedback* interativo, reportando problemas encontrados e sugerindo novas funcionalidades que julgar relevante para si.

## 3 *Os Sistemas Atuais*

### 3.1 Principais Concorrentes

Para um melhor entendimento do mercado atual, é necessário que analisemos em maior profundidade os principais sistemas com os quais vamos concorrer por público. É essencial absorver as funcionalidades que são o carro-chefe de cada um destes produtos, para que seja possível balancear nosso projeto e fazer frente ao mercado corrente:

**Organizze** Sistema mais completo atualmente para o usuário final. Possui diversas contas e tratamento especial para cartões, mas tal diferenciação é básica, se restringindo a dividir os lançamentos do cartão em faturas. Possui também metas, que não contribuem para o planejamento mensal ou futuro. Também conta com lançamentos repetidos e parcelados, além de um aplicativo para avaliação dos lançamentos do mês e criação de novos. O sistema possui uma interface clean e bem espaçada, o que pode incomodar alguns usuários que estão acostumados com maior volume de informações (vindos de outros sistemas ou de planilhas, por exemplo).

**Granatum** Segundo maior concorrente, apesar de ter foco maior no público corporativo. O sistema de gestão pessoal está abandonado há vários anos, não recebendo sequer uma melhoria ou correção. No entanto, gradualmente a versão empresarial tem sido renovada, então talvez o plano pessoal seja atualizado também – ou descontinuado. Em matéria de recursos, é muito similar ao *Organizze*, exceto pelo aplicativo móvel ter sido lançado há pouco tempo, e não funcionar corretamente com contas pessoais. Também não possui tratamento especial das despesas de cartão de crédito, e sua listagem de lançamentos se assemelha mais com uma planilha eletrônica do que o outro concorrente.

**Mint** “Concorrente” americano. Líder de mercado da América do Norte, principalmente nos Estados Unidos, onde já se consolidou como ferramenta de gestão financeira pessoal há muitos anos. O principal diferencial é a habilidade de importar automaticamente e sem intervenção todos os lançamentos ocorridos nas contas bancárias e seus cartões, dos mais

diversos bancos. Isso permite ao usuário somente categorizar e organizar as despesas, facilitando bastante a tarefa de analisar os gastos e sua saúde financeira de modo geral. No entanto, por causa de sua natureza focada no histórico bancário, o sistema não permite o planejamento futuro a partir de previsões de gastos e lançamentos adiantados, o que reduz bastante os benefícios para o usuário final. Suas integrações estreitas permitem a monetização facilitada, oferecendo ao cliente diversos outros produtos, como: cartão com taxas mais em conta no caso de serem identificadas altas taxas nos atuais; previdência privada baseada nas economias passadas; avisos sobre taxas inesperadas cobradas pelos bancos; dentre outros.

**Monefy** Um dos aplicativos móveis mais inovadores do mercado. Possui uma interface diferente para lançamento de despesas e receitas, que foca na categorização simplificada a partir de ícones. O aplicativo também possui uma calculadora na tela de valor e descrição, o que facilita a atualização de um lançamento anterior ou o agrupamento de despesas numa só entrada. Ele também possui a opção de alterar a moeda corrente do sistema, apesar de não permitir moedas distintas para cada conta. Sua tela principal consiste de um gráfico que facilmente demonstra ao usuário quais são suas principais fontes de despesas – mas fora isso, não colabora com outros métodos de análise. Há também um widget, que pode até mesmo ser usado diretamente da tela de bloqueio do telefone.

## 3.2 Outros Sistemas

Além dos sistemas digitais, também enfrentaremos um comportamento comum de novos usuários: o receio de migrar de plataforma. Para lidar com a migração de outros sistemas, o usuário poderá importar seus dados, caso o outro suporte a exportação também. No entanto, o maior receio poderá ser a mudança para um sistema informatizado e *online* a partir de uma organização tradicionalmente analógica. Portanto, vamos tentar entender as principais vantagens e desvantagens dos sistemas não-inteligentes mais comuns:

**Caderno de anotações / agenda** O tradicional par *papel & caneta* ganha o cliente no quesito facilidade, pois é extremamente prático fazer uma nova linha para cada nova despesa. No entanto, o caderno perde e muito quando o usuário deseja: identificar uma despesa antiga; fazer correções; planejamento do futuro; fazer análises; dentre outros. Em todos estes pontos a interface digital poderá facilitar, e será necessário demonstrar isso para poder convencer este tipo de usuário a migrar.

**Planilhas eletrônicas** O meio de entrada mais comum para a informatização das finanças de certo grupo de usuários. Ganha na versatilidade e adaptação ao modo de pensar de cada pessoa, e também na facilidade de acesso, visto que normalmente não depende de internet para ser utilizado – apesar de perder em portabilidade. No entanto, a fluidez de uso do sistema é deficiente, visto que será necessário o conhecimento avançado deste tipo de aplicativo para muitos recursos além do básico da gestão financeira, como resumos, análise de longos períodos, ou visualizações gráficas. Além do mais, o usuário sempre corre o risco de ter danos em seu computador e não possuir um backup recente dos arquivos. Para este tipo de usuário precisaremos indicar as vantagens de um sistema mais robusto e online.

### 3.3 Motivações para o Novo Sistema

Como indicado na seção 2.2, há diversos tipos de usuários que poderão se interessar por um sistema como o **Insert Coin**. Para trazê-los ao sistema, será necessário chamar a atenção de cada um deles de forma diferente. Segue aqui uma sugestão de bons motivos para a migração, ou mesmo para iniciarem seu controle financeiro pelo projeto:

**Controle** O sistema dará nas mãos do usuário a possibilidade de ter o controle total sobre seu dinheiro, tendo conhecimento do que entra e sai de suas contas;

**Organização** Visualização dos dados de entradas e saídas dados em diferentes formatos e granularidades;

**Simplicidade** A facilidade de uso permitirá ao usuário usufruir destas qualidades sem sofrer com a perda de tempo, que normalmente ocorre em sistemas robustos e complexos;

**Praticidade** Uma versão móvel simples do sistema é essencial para diminuir o atrito de uma atividade rotineira como o lançamento de despesas cotidianas;

**Futuro** A visualização condensada do passado e as possibilidades de metas e lançamentos futuros levará o usuário à previsão de sua saúde financeira; isso permitirá maior conforto e liberdade em compras parceladas, ou o planejamento de sonhos como a compra da casa própria, um carro novo, ou uma grande viagem.



### 3.3.1 Problemas dos Sistemas Atuais

Nenhum sistema é perfeito; a principal chance de chegar ao mercado com um produto de sucesso é explorando as fraquezas dos concorrentes, e construindo em cima destes conceitos para que não sejam repetidos os mesmos erros.

Foi possível identificar, em diversos produtos menores do mercado, características similares de complexidade desnecessária para o usuário mediano, por exemplo. Esse tipo de comportamento afasta o usuário mais leigo, ao invés de estimulá-lo a uma vida financeira mais saudável.

Outra questão recorrente nas conversas com diversos usuários foi a “preguiça” gerada pelo trabalho rotineiro de controle das despesas. A maioria dos sistemas online exige alguns cliques até chegar à tela de novo lançamento, o que dificulta bastante o uso frequente. Os aplicativos móveis ganham nesse quesito, por estarem “sempre à mão”, sendo normalmente de uso mais prático. Por outro lado, há poucas soluções integradas no mercado, que unam a versatilidade de um sistema móvel com o poder de análise de um sistema *web/desktop*.

### 3.3.2 Situação Desejada

O objetivo final do projeto é dar ao usuário uma vida financeira mais estável e próspera, lhe permitindo visualizar oportunidades de economia que antes não eram visíveis. Dessa forma, o cliente poderá planejar e alcançar sonhos que antes eram muito distantes, a partir da organização de metas de gastos e de economias.

Isso poderá ser atingido com um sistema integrado que possibilite a análise de dados e histórico numa interface simples de entender, aliada a um aplicativo móvel que torne prática e rápida a tarefa cotidiana de acompanhamento financeiro.

## 4 *O Sistema Proposto*

Para uma etapa de desenvolvimento fluida e para o contínuo aprimoramento do sistema, se faz necessária a documentação técnica do mesmo. É de suma importância que estas informações se mantenham “vivas” assim como a própria implementação, de modo que sempre possamos nos apoiar numa documentação sólida e de fácil entendimento sempre que surgirem dúvidas ou para subdividir as tarefas de desenvolvimento em passos lógicos.

### 4.1 Requisitos do Sistema

Para que o **Insert Coin** atinja de forma eficaz seus objetivos será necessário cumprir os requisitos técnicos descritos a seguir.

#### 4.1.1 Design

- Interface limpa e simplificada, de modo a não confundir o usuário desacostumado;
- Categorias de fácil identificação visual;

#### 4.1.2 Implementação

- Certificado de segurança SSL para as comunicações de todos os servidores HTTP (tanto de API quanto de aplicação), em todas as aplicações e subsistemas;

#### Servidor REST

- Autenticação via *Auth header*<sup>1</sup>, e *fallback* com cookie, para facilitar a implementação de diversos tipos de clientes[4];

---

<sup>1</sup>Método de autenticação padrão do HTTP.

- Respostas curtas e compactadas com Gzip, para economia de dados e redução da latência no uso móvel;
- Documentação da API extensa e de rápido entendimento;
- Extensa suíte de testes para garantir que os aplicativos não falharão por causa da API;

### **Aplicativo Web**

- Layout fluido e responsivo, que se adapte a diversos formatos de telas, como celulares, tablets, e telas largas ou estreitas<sup>2</sup> no computador;
- Visões detalhadas dos dados, com todas as opções oferecidas para históricos, resumos e gráficos;

### **Aplicativo Móvel**

- Suporte a diversas plataformas (mínimo: iOS e Android), para evitar a perda de usuários por causa do ambiente;
- Login automático e silencioso após a autenticação inicial;
- Funcionamento *offline* sem afetar a usabilidade ou performance;
- Sincronização de dados manual e automática em *background* (quando encontrar uma conexão WiFi, ou via dados móveis também, caso o usuário deseje);
- Funcionalidades básicas de análise, como históricos, resumos, filtros e alguns gráficos;

---

<sup>2</sup>Uma visualização mais estreita do sistema, principalmente nas telas de lançamentos, pode ser útil em atividades comparativas — quando acessando lado a lado com o histórico bancário, por exemplo

## 4.2 Casos de Uso

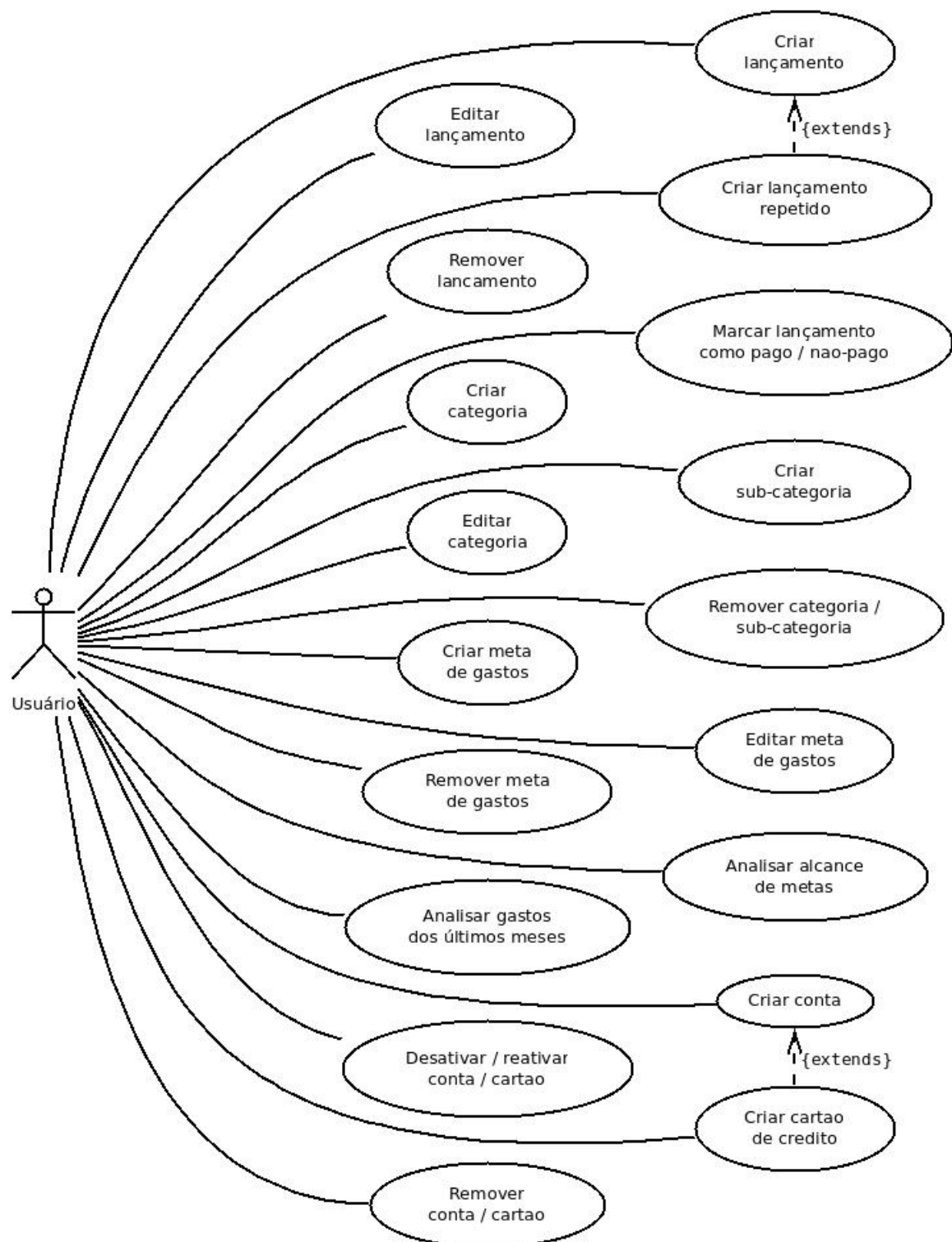


Figura 4.1: Diagrama de Casos de Uso

### 4.2.1 Especificações

#### [UC-01] Criar Lançamento

1. O usuário indica querer criar um novo lançamento de receita/despesa;
2. O sistema pergunta se é receita, despesa ou transferência de fundos;
3. O sistema pergunta qual é o valor, categoria e conta(s) de destino (e origem, no caso da transferência);
4. O sistema pergunta, opcionalmente, se há uma descrição detalhada;
5. O usuário preenche todos os dados;
6. O usuário salva o registro;
7. O sistema armazena os dados;
8. O sistema exibe os dados na planilha ou indica que foram armazenados, permitindo a criação de um novo registro.

#### [UC-02] Criar Lançamento Repetido

*Este Caso de Uso é uma extensão do UC-01. Os passos são inseridos após o passo 4 acima.*

1. O sistema pergunta se a despesa deve ser repetida de forma indefinida ou como despesa parcelada;
2. O usuário define o meio de repetição.

#### [UC-03] Editar Lançamento

1. O usuário indica querer criar um novo lançamento de receita/despesa

#### [UC-04] Remover Lançamento

1. O usuário indica querer criar um novo lançamento de receita/despesa

#### [UC-05] Marcar Lançamento como Pago / não-pago

1. O usuário indica querer criar um novo lançamento de receita/despesa

#### [UC-06] Criar Categoria

1. O usuário indica querer criar um novo lançamento de receita/despesa

**[UC-07] Criar Sub-categoria**

1. O usuário indica querer criar um novo lançamento de receita/despesa

**[UC-08] Editar Categoria**

1. O usuário indica querer criar um novo lançamento de receita/despesa

**[UC-09] Remover Categoria / Sub-categoria**

1. O usuário indica querer criar um novo lançamento de receita/despesa

**[UC-10] Criar Meta de Gastos**

1. O usuário indica querer criar um novo lançamento de receita/despesa

**[UC-11] Editar Meta de Gastos**

1. O usuário indica querer criar um novo lançamento de receita/despesa

**[UC-12] Remover Meta de Gastos**

1. O usuário indica querer criar um novo lançamento de receita/despesa

**[UC-13] Analisar Alcance de Metas**

1. O usuário indica querer criar um novo lançamento de receita/despesa

**[UC-14] Analisar Gastos dos Últimos Meses**

1. O usuário indica querer criar um novo lançamento de receita/despesa

**[UC-15] Criar Conta**

1. O usuário indica querer criar um nova nova conta;
2. O sistema pergunta qual é o tipo de conta desejada: Corrente ou Poupança;
3. O usuário define o tipo de conta;

4. O sistema pergunta o nome desejado para a conta, o saldo atual e o ícone representativo da mesma, a partir de uma lista de opções pré-definidas. Inicialmente, o sistema apresentará um ícone padrão para conta-corrente e outro para conta-poupança;
5. O usuário salva os dados;
6. O sistema cria a conta e uma transação inicial com o saldo corrente;
7. O sistema leva o usuário para a listagem de transações da nova conta, para que o cliente possa populá-la opcionalmente.

#### **[UC-16] Criar Cartão de Crédito**

*Este Caso de Uso é uma extensão do UC-15. Os passos modificados começam no 2.*

1. O sistema também permite o tipo de conta “Cartão”;
2. O usuário escolhe “Cartão”;
3. O sistema também pergunta as datas de fechamento e vencimento da fatura do cartão;
4. O usuário preenche os dados como em UC-15 e o caso de uso prossegue normalmente.

#### **[UC-17] Desativar / reativar conta / cartão**

1. O usuário visualiza a lista de contas ou cartões que possui;
2. O sistema exibe uma opção em cada conta:
  - Para contas ativas, a opção exibe “Desativar”;
  - Para contas inativas, o opção exibe “Ativar”;
3. O usuário seleciona a operação desejada para a conta que deseja modificar;
4. O sistema efetua a operação;
5. O sistema leva o usuário de volta à tela de listagem de contas.

#### **[UC-18] Remover conta / cartão**

1. O usuário visualiza a lista de contas ou cartões que possui;
2. O usuário seleciona qual deseja apagar;
3. O sistema alerta que a operação é irreversível e que todas as transações serão removidas junto com a conta;
  - (a) O usuário confirma e a conta é removida;
  - (b) O usuário muda de ideia e cancela a operação;
4. O sistema leva o usuário de volta à tela de listagem de contas.

### 4.3 Diagrama de Topologia

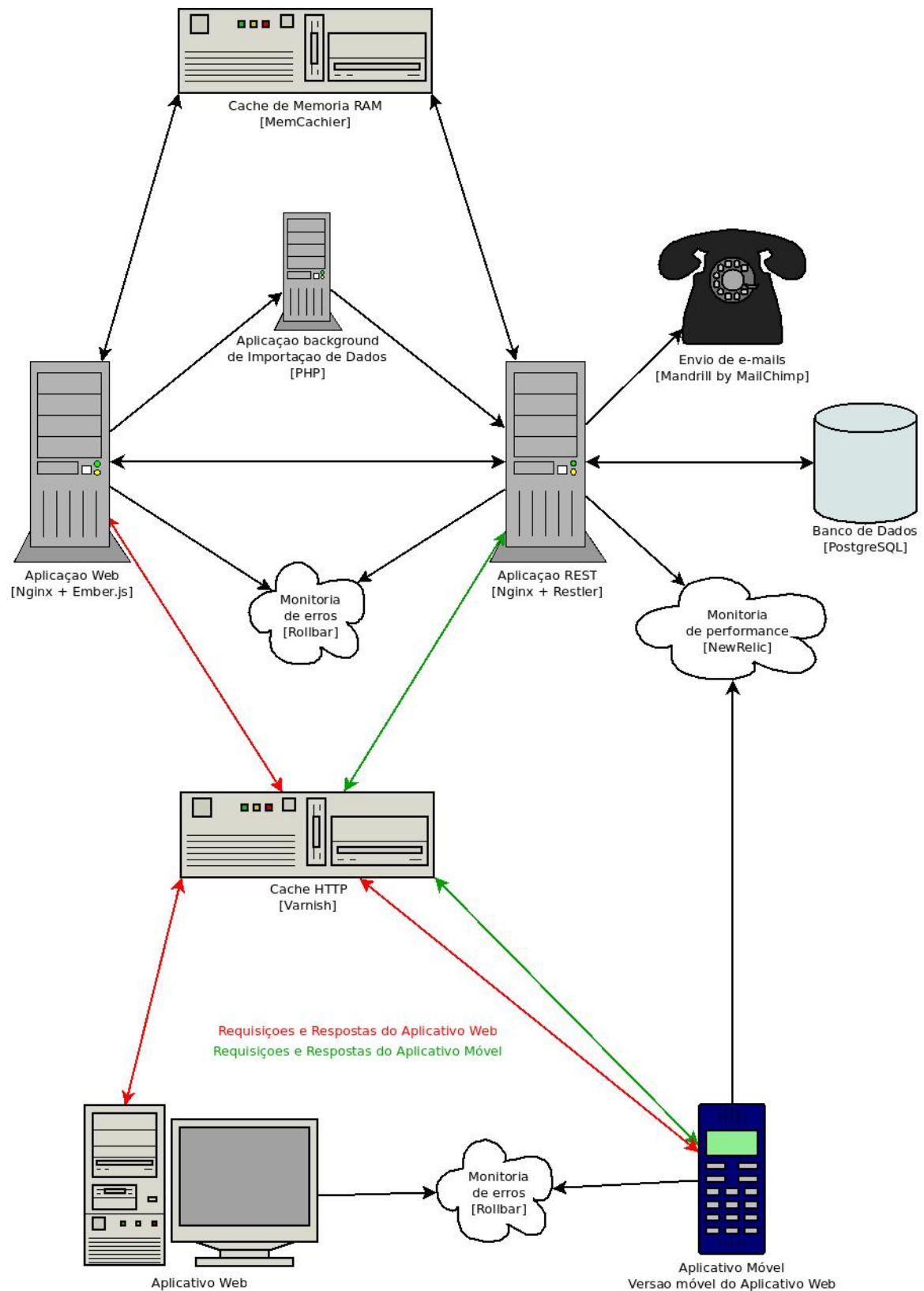


Figura 4.2: Diagrama de Topologia, indicando os diversos servidores de aplicação, de cache, serviços adicionais e o fluxo de comunicação com os aplicativos-cliente



## 4.4 Modelo Conceitual de Classes

Neste diagrama descrevemos as principais classes do Servidor REST do projeto. É possível notar que as classes refletem em boa parte o Modelo de Dados (proposto na seção seguinte, 4.5). Isso ocorre pois o servidor funcionará, principalmente, como um sistema de banco de dados de forma simplificada, armazenando todas as regras de negócio e, em alguns casos, expondo métodos mais inteligentes e ágeis que o tradicional *CRUD*.

O método de espelhamento do banco proporcionará aos aplicativos a maior liberdade possível para serem desenvolvidos e se adaptarem aos seus ambientes, ao invés de se adaptarem ao modelo de dados proposto por um servidor mais enxuto. No entanto, ao mesmo tempo ele garante os requisitos básicos de segurança e funcionalidades, por implementar todas as regras de negócio relacionadas aos dados conjuntamente com os mesmos.

Todos os métodos demonstrados por estas classes são baseados no formato da biblioteca *Restler*, que interpreta classes gerais (não-necessariamente com uma hierarquia em comum) e abre seus métodos públicos num *endpoint HTTP*. Dessa forma, as classes se tornam *recursos*, e os métodos possuem nomes dos *verbos HTTP*; opcionalmente, também são acompanhados de um sufixo que indica uma ação mais específica dentro daquele recurso.

Método	Requisição HTTP	No corpo	Resposta
Category::get()	GET /category	---	Lista de <i>Category</i>
Account::get(\$id)	GET /account/«id»	---	Um objeto <i>Account</i>
Icon::post(\$name, \$img)	POST /icon	name, img	Novo objeto <i>Icon</i>
Goal::putFilled(\$id, \$filled)	PUT /goal/«id»/filled	filled	Novo valor do campo <i>Goal::filled</i>
User::delete(\$id)	DELETE /user/«id»	---	Último estado de User, antes do registro ser removido

Tabela 4.1: Exemplos da conversão das classes para recursos HTTP, com *Restler*

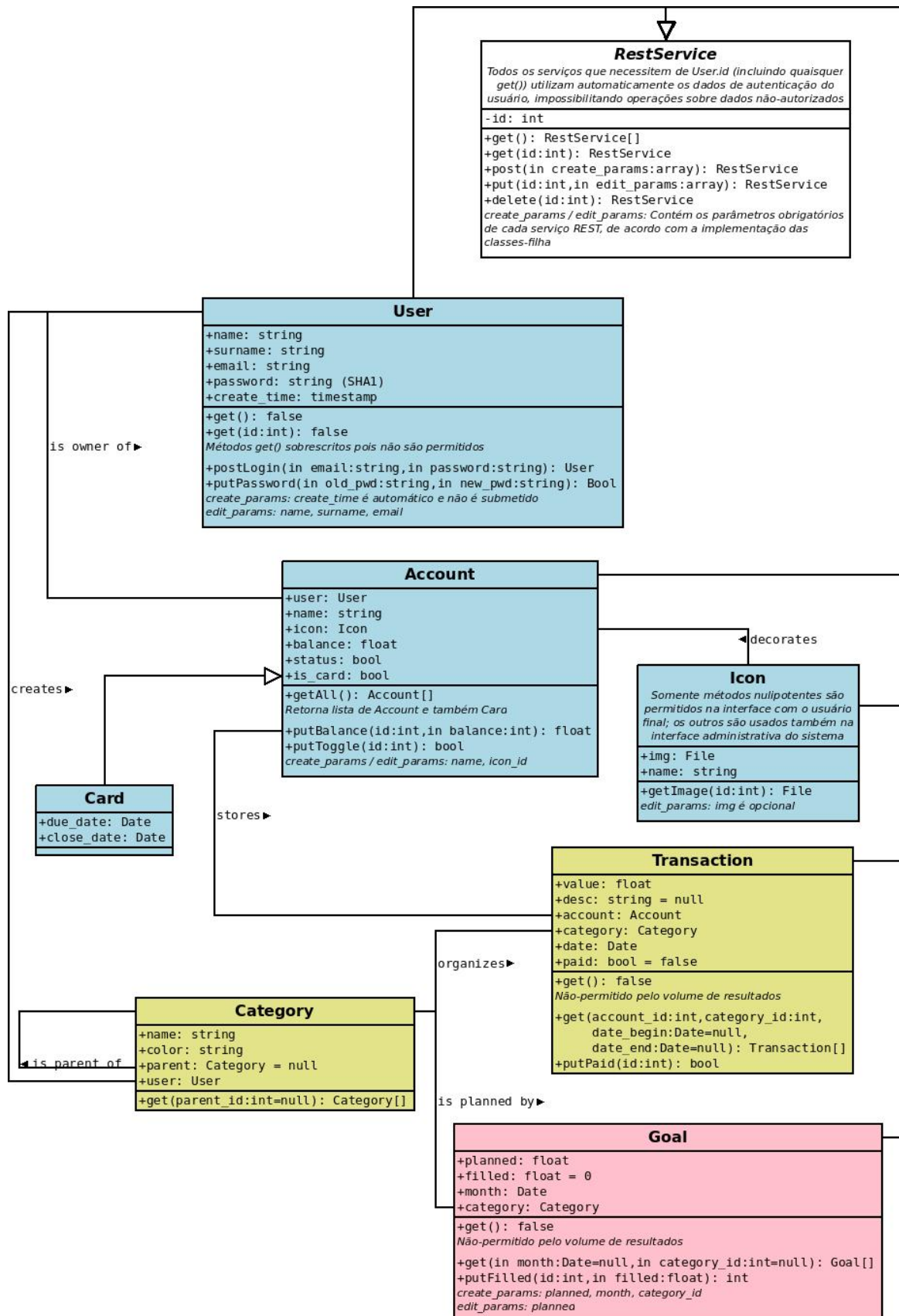


Figura 4.3: Diagrama de Classes, demonstrando as principais classes do sistema REST

## 4.5 Modelo Conceitual de Dados

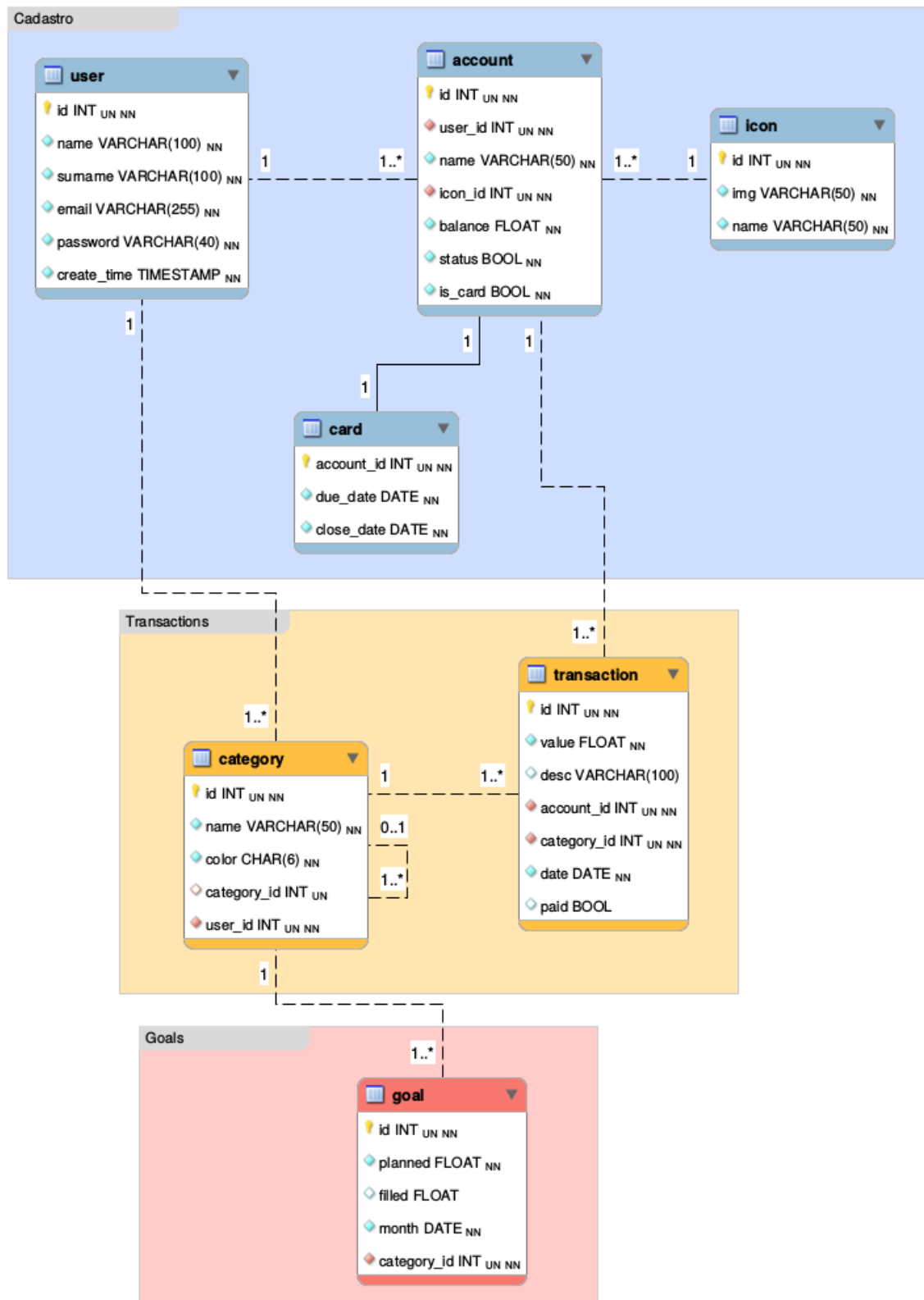


Figura 4.4: Diagrama de Entidade-Relacionamento, descrevendo o modelo de dados proposto para o projeto

## *Bibliografia*

- [1] Vincent Driessen. *A successful Git branching model*. Inglês. 5 de jan. de 2010. URL: <http://nvie.com/posts/a-successful-git-branching-model/> (acesso em 29/11/2014) (ver p. 6).
- [2] Heroku, Inc. *Heroku | Features*. Inglês. Informações sobre escalabilidade, recursos adicionais, monitoramento, etc. 2014. URL: <https://www.heroku.com/features> (acesso em 30/11/2014) (ver p. 7).
- [3] Heroku, Inc. *Heroku | Postgres*. Inglês. 2014. URL: <https://www.heroku.com/postgres> (acesso em 30/11/2014) (ver p. 7).
- [4] Brandon Linton. *Should cookies be used in a RESTful API?* Inglês. 22 de mar. de 2012. URL: <http://programmers.stackexchange.com/questions/141019/should-cookies-be-used-in-a-restful-api> (acesso em 03/12/2014) (ver p. 18).