Documento de Arquitetura

Conteúdo

Introdução

Este documento visa oferecer os detalhes das principais partes da arquitetura e dar uma visão geral do comportamento do aplicativo PocketPlanner. Neste documento serão abordados os seguintes tópicos: Principais casos de uso, requisitos não funcionais, ambiente, visão física, visão lógica, controle de acesso, ferramentas, padrões tanto de nomes quanto de projeto e frameworks utilizados.

Visão Geral

O sistema PocketPlanner é um aplicativo inicialmente voltado para a plataforma Android e posteriormente estendido para WEB que visa, através de uma interface simples, amigável e intuitiva, facilitar o gerenciamento de finanças pessoais dos usuários.

Ele visa facilitar o controle das finanças do usuário permitindo que o mesmo cadastre seu salário e outras eventuais fontes de renda, bem como suas despesas mensais e o sistema fará o balanceamento de ambas, mostrando se os gastos estão compatíveis com a renda do usuário, mantendo sua renda mensal sob controle . A principal funcionalidade do sistema é que as pessoas que o utilizarem poderão, ao pagar uma despesa programada no mundo real, informar ao aplicativo que pagaram essa despesa, anexando uma foto de um comprovante de pagamento ou comentário, e o sistema automaticamente colocaria o status dessa despesa para "pago" e recalcularia e exibiria o que ainda resta da renda no mês atual.

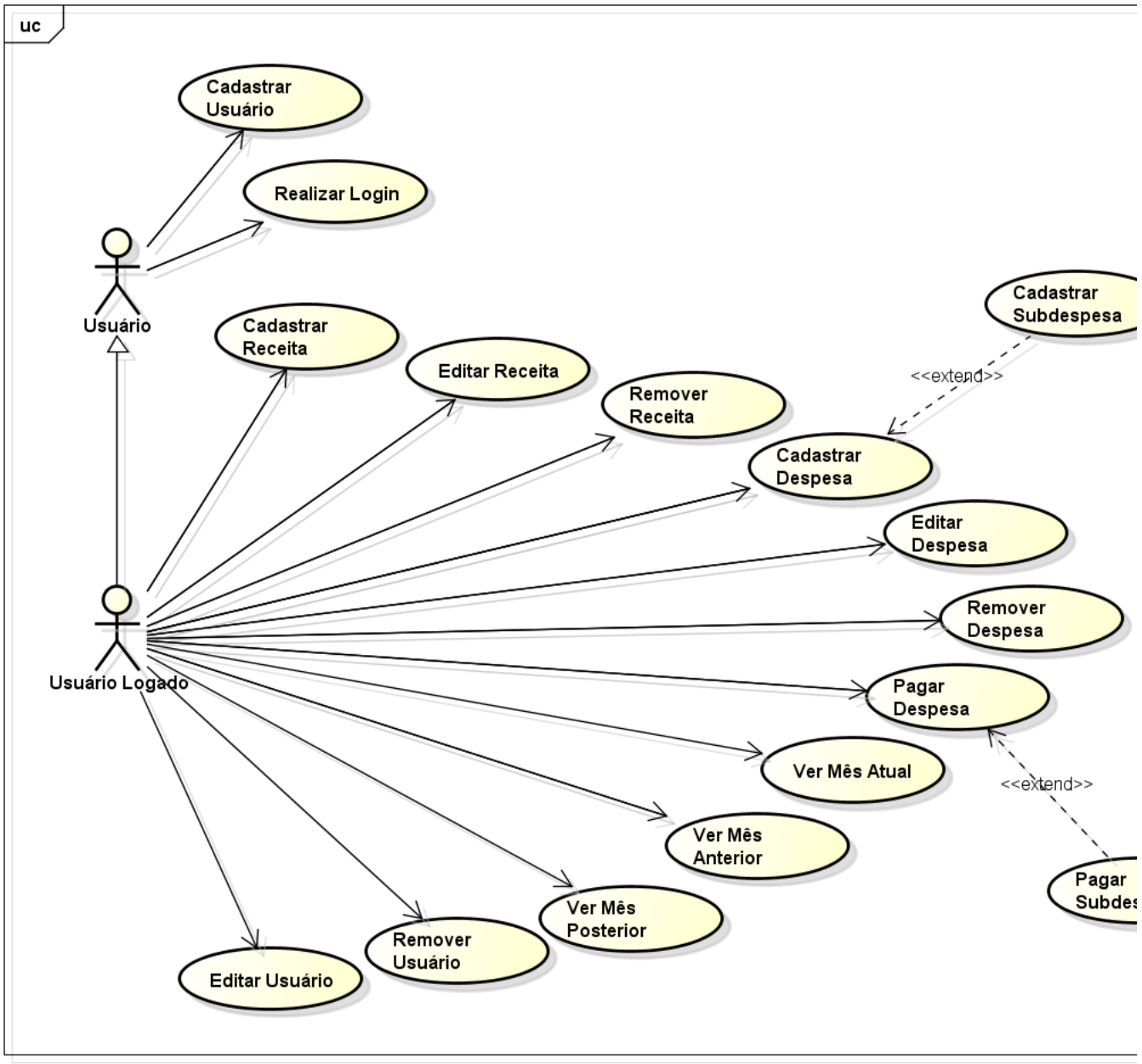
Se no fim do mês a receita não estiver zerada (saldo positivo, caso o valor das despesas for menor que o valor da receite, ou negativo, caso o valor das despesas for maior que o valor da receita), será dada a opção de replicar esse valor para o próximo mês, nesse caso a receita será somada ou subtraída do saldo do mês anterior.

O sistema ainda permite que uma despesa possua subdespesas, para o caso do usuário definir que gastará, por exemplo, 100 reais com gadgets, ele poderá informar que gastou 10 reais com um gadget, 20 com outro gadget, e então o sistema irá somar todas essas subdespesas para compor o valor da despesa "pai".

O PocketPlanner também dará suporte a despesas planejadas, como no caso de um parcelamento ou financiamento. Nesse caso o sistema dará a opção de, ao cadastrar uma despesa, informar se ela se repetirá, por quanto tempo e o qual o valor dos juros (se houver)

Principais Casos de Uso

O Diagrama de Casos de Uso é um Diagrama UML que tem por o objetivo facilitar a comunicação entre os analistas e o cliente, nesse sentido o cliente, ao ver um diagrama de Casos de Uso, deve conseguir visualizar sem maiores dificuldades as principais funcionalidades de seu sistema. Um diagrama de Caso de Uso descreve um cenário que mostra as funcionalidades do sistema do ponto de vista do usuário.



Os casos de uso mais significativos que merecem ser descritos no documento de arquitetura do PocketPlanner são: Cadastrar Receita, Cadastrar Despesa, Cadastrar Sub-despesa Pagar Despesa.

Cadastrar Receita: Este caso de uso permite que o usuário adicione quantas fontes de renda ele possuir. A receita possuirá 5 campos: Nome (Obrigatório), Descrição (Opcional), Valor (Obrigatório), Repetir (Obrigatório), Opções de Repetição (Opcional caso a opção "Repetir" seja marcada). O usuário poderá escolher se a receita será um evento único (ex: 13º salário), ou se irá se repetir nos meses posteriores. Caso seja escolhida a opção repetição, haverá algumas possibilidades de repetição, como "repetir indefinidamente", "repetir por X meses" ou "repetir até mês Y/Z" (Y = Jan, Fev, Mar, ...; Z = 2013, 2014, 2015, ...). A receita total do usuario deverá aparecer na tela principal do sistema e será composta pelo total de todas as receitas

Cadastrar Despesa: Este caso de uso permite que o usuário adicione quantas despesas ele quiser. A despesa possuirá 5 campos: Nome (Obrigatório), Descrição (Opcional), Valor (Obrigatório), Repetir (Obrigatório), Opções de Repetição (Opcional caso a opção "Repetir" seja marcada). Ao criar uma despesa o usuario também escolherá entre criar uma despesa única (padrão) ou criar uma despesa composta. Uma despesa composta implicará que essa despesa será um total das sub-despesas. Ainda no campo das despesas compostas, serão apresentadas duas opções, a despesa composta com custo fixo: O valor da despesa será fixado e o total das sub-despesas não poderá ultrapassar esse valor; a despesa comporta com custo variável: o valor da despesa não será informado e o total desta será a soma de todos os valores das suas sub-despesas. Todas as despesas contarão com um campo Status que, por padrão, virá preenchido como "Não-pago". Ao cadastrar uma despesa, sua fonte ficará na cor vermelha, indicando que ela ainda não foi paga

Cadastrar Sub-Despesa: Este caso de uso permite ao usuario cadastar uma sub-despesa ligada a uma despesa "pai" que ele adicionou previamente. A sub-despesa possuirá 3 campos: Nome (Obrigatório), Descrição (Opcional), Valor (Obrigatório). Seu valor será atrelado ao valor da despesa "pai" logo após o cadastro.

Pagar Despesa: Este caso de uso permite que o usuário modifique o status da sua despesa para "pago". Nesse caso ele selecionará a despesa que ele deseja pagar e selecionará a opção "Pagar". Caso a despesa seja única, ele terá a opção de anexar uma foto (possível comprovante de pagamento) e/ou um comentário. Caso a despesa seja composta ele poderá escolher entre pagar a despesa completa ou pagar somente uma sub-despesa. Ao pagar uma despesa, o status dela mudará para "pago" e sua fonte ficará na cor verde, indicando que já foi paga.

Pagar Sub-despesa: Este caso de uso segue o mesmo fluxo do "Pagar Despesa": selecionar a sub-despesa e marcar a opção "pagar", podendo anexar uma foto ou comentário. A sub-despesa, ao ser paga, mudará a cor da fonte para "verde" indicando que ela já foi paga.

Requisitos Não-Funcionais

Abaixo estão citados alguns requisitos não funcionais:

- Controle de acesso, para que os dados das receitas e despesas dos usuários não sejam públicos;
- Senhas criptografadas no banco de dados;
- Design intuitivo;
- Simplicidade de uso.

Requisitos Arquiteturais/Ambiente

Esta seção descreve os requisitos básicos do sistema em relação ao Sistema Operacional e Navegadores suportados, Linguagem utilizada para o desenvolvimento e os tipos de ambiente que ele suportará.

Sistema Operacional

Windows XP ou superior

Navegador

Internet Explorer 6.0 ou superior
Mozilla Firefox 3.0 ou superior
Google Chrome

Linguagem

Java 6 ou superior

Banco de dados

Web: MySQL 5.5 ou superior

Servidor

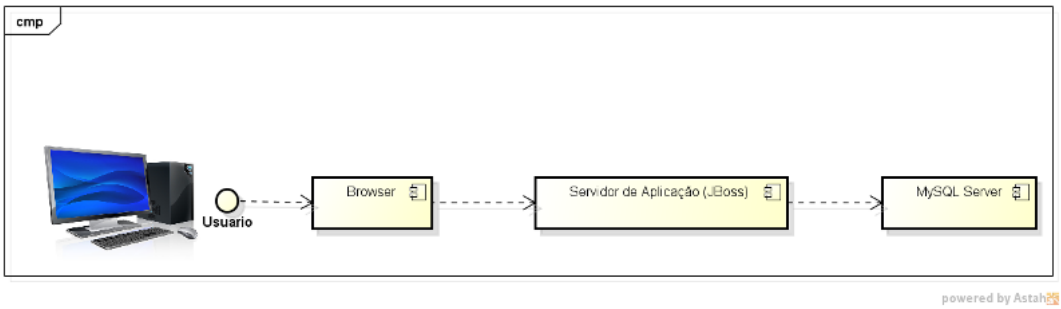
JBoss 7.1

Frameworks

- JSF 2.0
- EJB 3
- JPA 2.0

Visão de Implantação

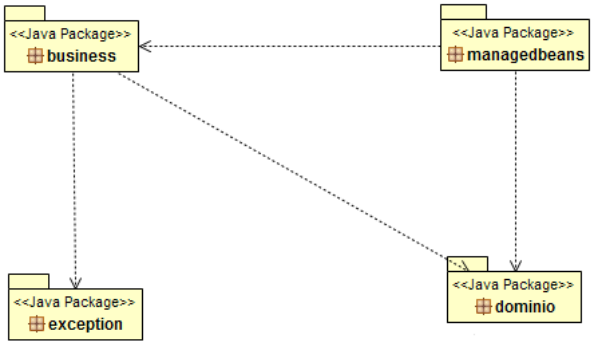
Esta seção descreve, por meio da imagem abaixo, as configurações da rede física (hardware) na qual o software é implantado e executado.



- O Web Browser é o cliente do sistema;
- O servidor da aplicação que utilizaremos será o JBoss;
- O servidor de banco de dados utilizado será o MySQL.

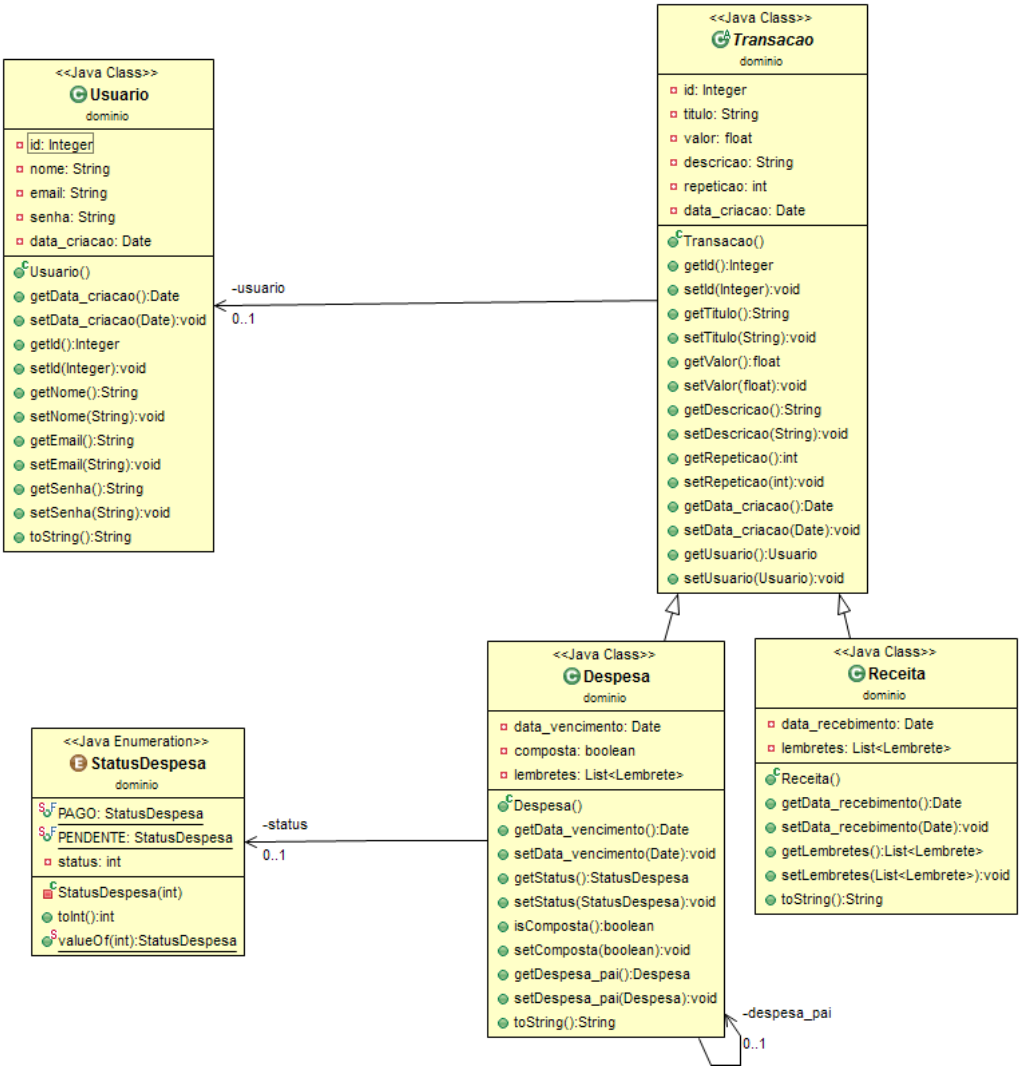
Visão Lógica

O desenvolvimento do sistema é feito em camadas, representadas por pacotes na visão mais baixo nível, que organiza e separa as classes de acordo com sua finalidade. Segue abaixo o diagrama de pacotes para visualização dessa divisão:

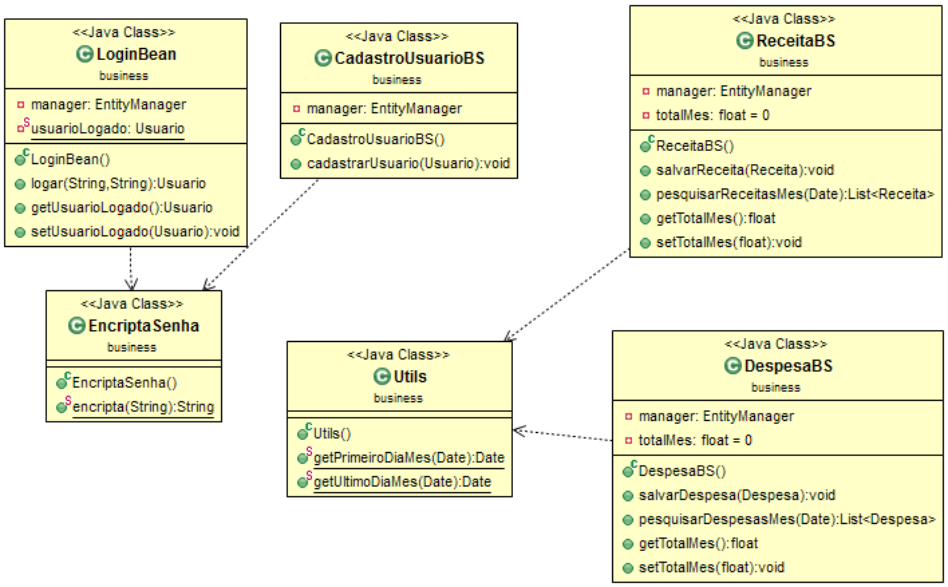


Também abaixo seguem representadas por meio de diagramas de classes as relações entre as classes dentro dos pacotes de domínio e de business:

- Domínio:



- Business



Segurança/Controle Acesso

A segurança do sistema é uma questão de prioridade pois não desejamos que qualquer pessoa, ao ter acesso ao smartphone do usuário, possa entrar no aplicativo e desse modo, ver toda a vida financeira do nosso usuário alvo.

Partindo desse princípio, no primeiro acesso do usuário ao aplicativo, o sistema solicitará que o mesmo crie uma conta para si (login - email e senha - 8 caracteres) e nesse momento o sistema necessita de acesso a internet para replicar esse cadastro no servidor. A partir daí o sistema dará a opção de validar o login todas as vezes que o usuário acessar o aplicativo, ou armazenar o login e senha para entrada automática. Como o sistema não dará suporte a multi-usuários, caso o usuário escolha a primeira opção, o login (email) dele já será automaticamente preenchido, necessitando somente da senha para poder acessar o sistema.

Caso já tenha realizado o primeiro acesso, já tenha criado sua conta e desejar excluir a mesma e cadastrar outra, ao ser excluido o sistema removerá o usuario da base de dados (do servidor e localmente) juntamente com todos os dados ligados a ela e o usuário será redirecionado para a tela de login e o comportamento a partir daí será o mesmo do primeiro acesso ao aplicativo.

Ferramentas

A ferramenta padrão para o desenvolvimento da versão Android será a IDE Eclipse do pacote ADT (Android Developer Tool) disponibilizado pelo site do Google através da url: http://developer.android.com/sdk/index.html.

Para a versão web também será utilizado o Eclipse, com suporte para desenvolvimento J2EE, JBoss e para os frameworks utilizados.

O sistema de controle de versão escolhido foi o GIT através do TortoiseGIT, utilizando como repositório o GitHub.