Fatec
Lins
Prof. Antonio Seabra

Análise e Desenvolvimento de Sistemas

Projeto 2Do | Gerenciador Pessoal

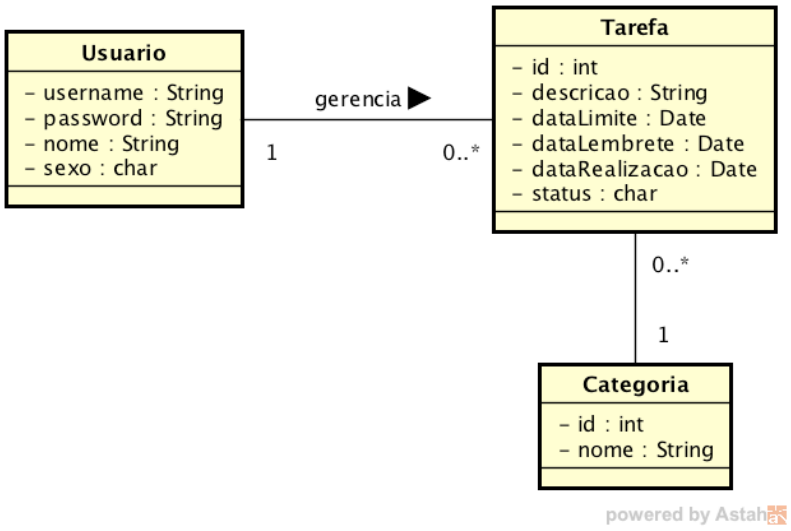
Projeto para prática e consolidação de conhecimentos adquiridos na disciplina de Programação para Dispositivos Móveis

Um dos maiores desafios de qualquer pessoa é gerenciar suas tarefas pessoais e profissionais.

O **objetivo** deste projeto é desenvolver uma ferramenta online em que pessoas possam gerenciar suas tarefas.

Diagrama de Classes

O seguinte Diagrama de Classes define os requisitos do sistema.



De acordo com o diagrama, deseja-se:

- Manter um cadastro de usuários que têm acesso ao sistema e que podem gerenciar suas tarefas;
- Manter um cadastro de tarefas, cadastradas e gerenciadas por usuário;
- Manter um cadastro de categorias como, por exemplo, “Lazer”, “Profissional”, “Financeiro”, para organizar as tarefas por tipos.

Primeira Etapa

ENTREGA
11/11/2015

1) Criar um banco de dados chamado ***todo***, usando ***SQLite***, que contém 3 tabelas, conforme especificadas a seguir:

Tabelas

Categoria

Nome do Campo	*Tipo*	*PK*	*FK*	*O que armazena*
id	INTEGER	X		Um número sequencial
nome	TEXT			O nome da categoria

Usuario

Nome do Campo	*Tipo*	*PK*	*FK*	*O que armazena*
username	TEXT	X		O nome de usuário para *login*
password	TEXT			A senha de usuário para *login*
nome	TEXT			O nome completo da pessoa
sexo	TEXT			F (Feminino) ou M (Masculino)

Tarefa

Nome do Campo	*Tipo*	*PK*	*FK*	*O que armazena*
id	INTEGER	X		Um número sequencial
descricao	TEXT			A descrição que especifica a tarefa
dataLimite	INTEGER			A data que define o encerramento do prazo para cumprimento da tarefa
dataLembrete	INTEGER			A data que define o dia em que o sistema deve emitir alguma notificação sobre a tarefa pendente
dataRealizacao	INTEGER			A data em que a tarefa foi realizada
status	TEXT			P (pendente) ou R (realizada)
username	TEXT		X	Identifica o "dono" da tarefa
idCategoria	INTEGER		X	Identifica a categoria à qual pertence a tarefa

2) Criar um projeto no ***Android Studio***. Neste projeto, criar:
   a. o pacote ***modelo***;
   b. dentro do pacote ***modelo***, as classes `Categoria`, `Usuario` e `Tarefa`, que apenas declaram os atributos exibidos no Diagrama de Classes ilustrado na página anterior e os respectivos métodos *getters* e *setters*;
   c. a classe `DatabaseHelper`, responsável pela criação do banco de dados e das respectivas tabelas;
   d. a classe `UsuarioDao`, que implementa os métodos:
      - `inserir(Usuario)`, para cadastrar novos usuários;
      - `atualizar(Usuario)`, para atualizar dados do usuário (com exceção do nome de usuário);
      - `excluir(String)`, para excluir o cadastro de usuários, inclusive todas as tarefas a ela relacionadas;

- `validar(String, String)`, para validar os dados de um usuário que deseja acesso ao aplicativo.

e. uma `Activity` que servirá como tela de *login* para o aplicativo.

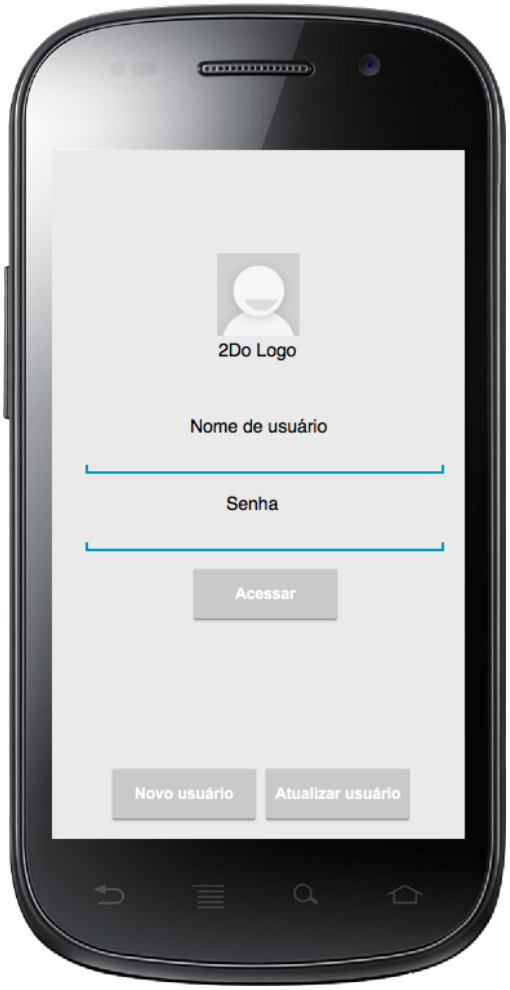


Figura 1. Sugestão de layout para Activity de login.

Segunda Etapa

ENTREGA
18/11/2015

3) Incluir no projeto:
   a. a classe `CategoriaDao`, que implementa os métodos:
      - `inserir(Categoria)`, para cadastrar novas categorias;
      - `atualizar(Categoria)`, para atualizar o nome de uma categoria; a atualização só pode ser autorizada se a categoria estiver vinculada somente a tarefas do respectivo usuário; se a categoria em questão estiver vinculada a tarefas de outros usuários, a atualização não deve ser autorizada;
      - `excluir(int)`, para excluir uma categoria, somente se ela não estiver vinculada a qualquer tarefa.
      - `selecionarTodas()`, para retornar uma lista das categorias cadastradas.
4) As telas para visualização e cadastro de novas categorias.

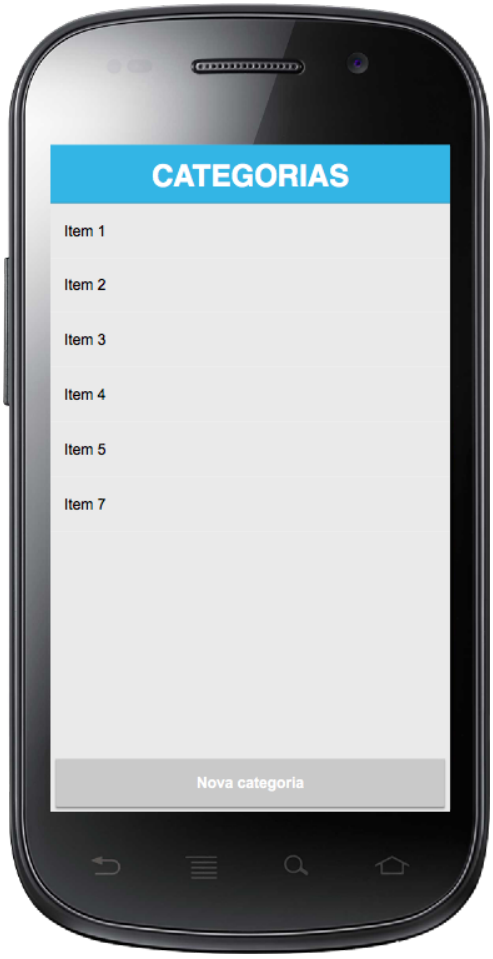


Figura 2.Sugestão de layout para exibição de categorias cadastradas em uma ListView.

Figura 3.Sugestão de layout para tela de alteração ou exclusão de categorias, aberta depois que o usuário seleciona uma das categorias da ListView.

Terceira Etapa

ENTREGA
Até
02/12/2015

5) Incluir no projeto:
    a. a classe `TarefaDao`, que implementa os métodos:
        - `inserir(Tarefa)`, para cadastrar novas tarefas;
        - `atualizar(Tarefa)`, para atualizar os dados de uma tarefa ainda não realizada;
        - `excluir(int)`, para excluir uma tarefa;
        - `selecionarTodas()`, para retornar uma lista de todas as tarefas ainda não realizadas.

As telas para visualização e manutenção (inclusão, atualização e exclusão) de tarefas ficam a critério do desenvolvedor.

Exige-se apenas que a lista de tarefas seja apresentada em uma `ListView`, a fim de facilitar a consulta. Além disso, pede-se também que a marcação de uma tarefa como "realizada" seja feita através da própria `ListView`.

CRITÉRIO DE AVALIAÇÃO

O aplicativo será avaliado por meio das seguintes dimensões, considerando-se as restrições indicadas em cada etapa do projeto:

	Item avaliado	Nota máxima atribuída
Usuário	Inserir	0,5
	Atualizar	0,5
	Excluir	1,0
	Validar (*login*)	1,0
Categoria	Inserir	0,5
	Atualizar	1,0
	Excluir	1,0
	Listar	1,0
Tarefa	Inserir	1,0
	Atualizar	1,0
	Excluir	0,5
	Listar	1,0