Trabalho Prático

Meta 1 - 5 valores

Pretende-se que seja desenvolvido um sistema distribuído para auxiliar grupos de amigos ou colegas no registo e gestão de despesas partilhadas (por exemplo, quando passam férias em conjunto) para que, no final, seja facilitada a tarefa de acerto de contas. A inspiração para este tema de trabalho prático foi a aplicação Splitwise (https://www.splitwise.com/). Globalmente, quando um dos elementos do grupo efetua um pagamento (por exemplo, uma portagem ou os pequenos-almoços de vários elementos), é inserido no sistema um novo registo como indicação de quem a pagou, quais são os elementos do grupo pelos quais esta deve ser dividida (por questões de simplificação, em partes iguais), o seu valor e a data. O sistema apresenta, com base na informação introduzida, quanto é que cada pessoa deve e tem a receber de cada um dos outros elementos do grupo. Também é possível registar o acerto/liquidação de contas (pagamento de valores em dívida entre elementos).

O sistema a desenvolver deve obedecer aos requisitos funcionais e arquiteturais definidos nas próximas secções. A implementação tem de ser realizada recorrendo à linguagem Java utilizando os conceitos estudados ao longo da unidade curricular. O sistema é composto por três tipos de componentes: **cliente**, aplicação através da qual os utilizadores interagem com o sistema; **servidor principal**, aplicação responsável por toda a lógica de negócio e gestão dos dados; e **servidor de *backup***, aplicação que interage com o servidor para manter uma cópia de segurança atualizada dos dados. Para efeitos de gestão e persistência dos dados, os dois tipos de servidor recorrem a uma base de dados SQLite.

1 Requisitos Funcionais

O sistema distribuído pretendido deve oferecer as seguintes funcionalidades aos **utilizadores**:

- **Registo de um novo utilizador**, sendo este caraterizado por um nome, um número de telefone, um endereço de email e uma *password*. O endereço de email deve ser único e serve de *username*;
- **Autenticação de um utilizador** já registados através do respetivo endereço de email e *password*;
- **Edição dos dados de registo**;

- **Criação de um novo grupo**, sendo este caracterizado por um nome (por exemplo, “Viagem finalistas deis-isec 2025 - Andorra”) que deve ser único no sistema. O utilizador que cria o grupo passa automaticamente a integrá-lo (primeiro elemento);
- As operações executadas no contexto de um determinado grupo são acessíveis apenas aos seus elementos;
- A aplicação cliente deve permitir a **seleção do grupo corrente/atual**, ou seja, o utilizador escolhe um dos grupos a que pertence e, a partir de esse momento, as operações que executar referem-se implicitamente a esse grupo;

- **Criação de convites para adesão a um grupo**, sendo os destinatários identificados através dos seus emails de registo no sistema;
- **Visualização** automática e atualizada dos **convites de adesão recebidos/pendentes**;
- **Aceitação e recusa de convites** de adesão a grupos;
- **Lista dos grupos** a que pertence o utilizador autenticado;
- **Edição do nome de um grupo** por qualquer um dos seus elementos;
- **Eliminação de um grupo** e respetivos dados, desde que não exista qualquer conta por saldar/valor em dívida (ou seja, não podem existir situações de "o elemento X deve a quantia Z ao elemento Y" / "o elemento Y tem a receber a quantia Z do elemento Y").
- **Saída de um grupo** se ainda não existir qualquer despesa associada ao utilizador;

- **Inserção de uma despesa** associada ao grupo corrente, por qualquer um dos seus elementos, com: data; descrição; valor; quem pagou; e os elementos com quem é partilhada (pode não incluir quem pagou);
- **Visualização do valor total de gastos** efetuados pelo grupo corrente;
- **Visualização do histórico das despesas** associadas ao grupo corrente, ordenadas cronologicamente, com todos os detalhes, incluindo a identificação de quem a inseriu no sistema (pode não ser quem efetuou a despesa);
- **Exportação, para um ficheiro em formato CSV, da lista de despesas** associadas ao grupo corrente, ordenadas cronologicamente e detalhada (ver exemplo na Figura 1);
- **Edição dos campos de uma despesa** já introduzida no sistema;
- **Eliminação de uma despesa**;

- Para efeitos de liquidação/acerto das contas, **inserção de um pagamento efetuado por um elemento** do grupo corrente **a outro elemento** do mesmo grupo, com indicação de: quem pagou; quem recebeu; data; e valor;
- **Listagem dos pagamentos efetuados** entre elementos do grupo;
- **Eliminação de um pagamento efetuado** por um elemento a outro elemento;

- **Visualização dos saldos** do grupo corrente com, para cada elemento, indicação do:
  - gasto total;
  - valor total que deve;
  - valor que que deve a cada um dos restantes elementos;
  - valor total que tem a receber;
  - valor que tem a receber de cada um dos restantes elementos;

- ***Logout***.

Ao implementar as funcionalidades descritas, tenha em consideração os seguintes aspetos:

- As funcionalidades, com exceção do registo e autenticação, só estão disponíveis para utilizadores autenticados;
- Não podem ser criados utilizadores com emails iguais;
- De um modo genérico, as aplicações cliente e servidor necessitam de suportar a execução simultânea de diversas operações (notificações assíncronas, etc.). Desta

forma, deve recorrer-se aos mecanismos de programação concorrente estudados (e.g., *threads* e mecanismos de sincronização) sempre que se justificar a sua utilização;

- Existem aspetos relacionados com funcionalidades e características arquiteturais e protocolares que estão omissos neste enunciado. Os grupos de trabalho têm total liberdade para lidar com essas questões e implementarem soluções da forma que melhor entenderem. Em caso de dúvida, devem contactar um dos docentes.

```
“Nome do grupo”
“Viagem finalistas deis-isec 2025 - Andorra”

“Elementos”
“Ana”;”Juliette”;”Berta”;”Joel”;”Francisco”

“Data”; “Responsável pelo registo da despesa”; “Valor”;”Pago por”;“A dividir com”
”30-09-2024”;”Juliette“;”10”;”Francisco”;”Ana”;”Joel”
”02-10-2024”;”Francisco“;”34.5”;”Francisco”;”Ana”;”Joel”
”02-10-2024”;”Berta“;”60”;”Ana”;”Francisco”;”Joel”;”Berta”
”04-10-2024”;”Juliette“;”45”;”Juliette”;”Juliette”
```

Figura 1 – Exemplo hipotético de ficheiro CSV com listagem das despesas associadas a um grupo

2 Requisitos Arquiteturais e Protocolares

A arquitetura geral do sistema pretendido é apresentada na Figura 2, sendo constituída por um servidor principal, eventuais servidores de *backup*, clientes e bases de dados SQLite (a principal no servidor principal e réplicas desta nos servidores de *backup*). Os dois tipos de servidor correm no mesmo domínio de difusão e formam um *cluster* que oferece o serviço pretendido (requisitos funcionais explicitados na Secção 1). Cada servidor acede, de forma exclusiva, a uma base de dados SQLite local com a informação necessária ao funcionamento do sistema. Esta informação é atualizada pelo servidor principal, na sequência das interações com os utilizadores, e propagada às restantes bases de dados, que devem manter-se consistentes com a principal.

Os princípios de funcionamento e de interação das aplicações servidor principal, servidor de *backup* e cliente são descritos nas secções seguintes. Em todos os cenários expostos, a comunicação pode ser feita da forma que os grupos de trabalho entenderem ser mais apropriada. Por exemplo, podem ser utilizadas estratégias baseadas em cadeias de caracteres ASCII ou em objetos serializados.

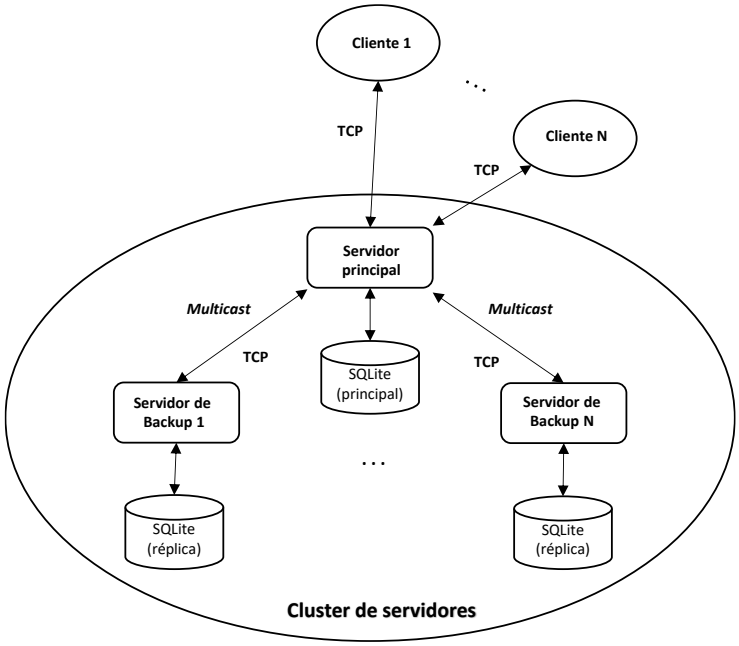

Figura 2 - Arquitetura geral do sistema

2.1 Servidor principal

- Deve ser lançado indicando, na linha de comando:
  - um porto de escuta TCP, onde irá aguardar pela conexão de clientes;
  - o caminho da diretoria de armazenamento da sua base de dados SQLite (ficheiro);
- Aguarda continuamente por pedidos de ligação TCP de clientes no porto indicado;
- Depois da fase de arranque, um servidor envia, a cada 10 segundos, uma mensagem de *heardbeat* para o porto 4444 do endereço *multicast* 230.44.44.44;
- Os *heartbeats* incluem, no mínimo, a seguinte informação: número de versão atual da sua base dados local e porto de escuta TCP automático no qual aguarda pedidos de ligação de servidores de *backup* para obtenção de uma cópia integral da base de dados;
- Um *heardbeat* também é emitido depois de ser feita qualquer alteração à base de dados local, com inclusão adicional da *query* SQL que foi executada (inserção, atualização ou eliminação);
- Cada base de dados SQLite tem um número de versão associado que é incrementado sempre que é feita uma atualização (este valor pode ser guardado na base dados numa tabela, com uma única coluna e linha, não relacionada com as restantes tabelas do modelo físico);
- Se, na fase de arranque, o servidor principal não possuir qualquer base de dados local criada, cria uma nova com número de versão 0 (estrutura criada, mas sem dados inseridos).
- Depois de um servidor proceder à atualização da sua base de dados local e às réplicas nos servidores de *backup*, os seus clientes, conectados via TCP, devem ser notificados

para que, de um modo assíncrono, procedam à atualização das suas vistas/informação apresentada aos respetivos utilizadores.

2.2 Servidores de *backup*

- Situam-se no mesmo troço de rede (domínio de difusão) do servidor principal, formando um *cluster*;
- São lançados indicando, na linha de comando, o caminho da diretoria para armazenamento da uma réplica da base de dados existente no servidor principal. Se a diretoria não estiver vazia, a aplicação deve terminar;
- Aguardam continuamente pela receção de *heartbeats* enviados pelo servidor principal para o porto 4444 do endereço de *multicast* 230.44.44.44. Se não detetarem qualquer *heartbeat* durante 30 segundos, terminam;
- Na fase de arranque, começam por obter, via TCP, ligando-se ao porto anunciado pelo servidor nos seus *heartbeats*, uma cópia integral da base de dados do servidor principal (ficheiro), guardando-a localmente na diretoria indicada através da linha de comando. Durante uma operação de transferência da base de dados, deve ser garantido que não é feita qualquer alteração aos dados pelo servidor principal;
- Depois da fase de arranque, a receção de um *heartbeat*, sem *query* de atualização, com um número de versão da base de dados diferente do valor local, provoca o final da execução do servidor de *backup*.
- Depois da fase de arranque, a receção de um *heartbeat*, com *query* de atualização, com um número de versão da base de dados diferente do valor local + 1, leva, igualmente, ao final da execução do servidor de *backup*.

2.3 Clientes

- São lançados fornecendo o endereço e o porto de escuta TCP do servidor principal através da linha de comando;
- Começam por solicitar o *username* (email) e a *password* ao utilizador, para feitos de autenticação, ou o conjunto de dados necessários ao registo de um novo utilizador;
- Estabelecem uma ligação TCP com o servidor principal e enviam a informação de autenticação ou de registo.
- Quando a autenticação falha ou as credenciais não são enviadas no espaço de 60 segundos, o servidor encerra a ligação TCP com o cliente;
- A informação (mensagens) trocada entre os clientes e o servidor principal pode assumir a forma que os grupos entenderem ser mais apropriada;
- As vistas dos clientes, que podem ser em modo texto ou gráfico, devem ser atualizadas de forma assíncrona.
- O código dos clientes deve estar estruturado em dois componentes distintos: vista (interação como o utilizador, menus, etc.) e comunicação (gestão da ligação TCP, envio e receção de mensagens, etc.).

3 Estrutura da Base de Dados

A Figura 3 apresenta uma sugestão para o modelo Entidade-Relacionamento (ER) das bases de dados SQLite acedidas pelos dois tipos de servidor, a partir do qual pode ser definido e implementado o modelo físico.

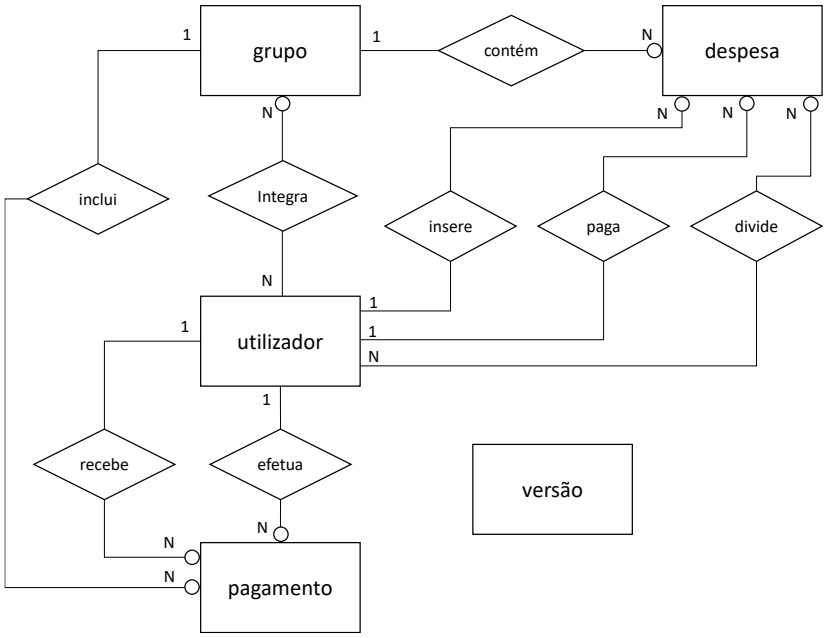

Figura 3 - Modelo Entidade-Relacionamento

4 Extras

A interface do utilizador da aplicação cliente descrita neste enunciado pode ser implementada em modo consola (i.e., texto). Os aspetos fundamentais considerados na avaliação base são se cumpre as funcionalidades pretendidas e se apresenta toda a informação necessária de forma adequada aos utilizadores. No entanto, os grupos que apresentem uma aplicação que cumpra minimamente os requisitos essenciais e que tenha uma interface do utilizador gráfica (GUI) funcional e completa, terão uma bonificação extra que poderá ir até aos 7.5% da nota atribuída. Ou seja, um trabalho avaliado em 80% que tenha a totalidade deste extra passa a valer 86% (80% + 80% * 7.5%).

5 Considerações Gerais

Deve ter-se em consideração os seguintes aspetos:

- O trabalho deve ser realizado preferencialmente por grupos de três alunos, não podendo este valor ser ultrapassado;
- A constituição dos grupos deve ser registada através da plataforma Moodle;

- Esta primeira meta do trabalho prático deverá ser entregue até ao dia **25 de novembro de 2024**, **às 8h00**, através da plataforma InforEstudante, num ficheiro ZIP com a designação *PD-24-25-F1-TP-Gx.zip*, sendo *x* o número do grupo;
- Haverá uma penalização de 10% por cada hora de atraso na entrega;
- O ficheiro referido no ponto anterior deve incluir o código fonte (ficheiros ".java") e a documentação produzida, assim como os ficheiros auxiliares necessários à execução e teste das aplicações sem necessidade de recorrer a qualquer IDE (e.g., o *byte code* gerado e respetivas *batch files* e/ou ficheiros do tipo *jar* executáveis) – não devem ser entregues as pastas e respetivos conteúdos correspondentes ao projeto criado no IDE;
- As opções tomadas durante o projeto (e.g., aspetos não especificados no enunciado, variações devidamente justificadas ao nível das funcionalidades implementadas ou do modo de interação entre os diversos componentes, tratamento de anomalias, etc.), os aspetos relevantes do sistema desenvolvido (pormenores de implementação, diagramas temporais, estrutura/modelo físico da base de dados, etc.) e o manual de utilizador devem ser devidamente documentados de um modo sintético num documento do tipo PowerPoint;
- No documento referido na alínea anterior, é aconselhável a utilização de figuras e capturas de ecrã;
- Não é expectável que diferentes grupos apresentem soluções iguais. Caso este cenário se verifique, os grupos envolvidos terão de se justificar. **A deteção de situações de plágio leva a uma atribuição direta de 0 valores na nota do trabalho aos alunos de todos os grupos envolvidos**;
- Será valorizado o facto de o código das aplicações desenvolvidas ter sido estruturado de uma forma modular, com uma separação clara entre lógica de funcionamento, lógica de comunicação, lógica de acesso aos dados e interface do utilizador, podendo esta ser em modo texto ou gráfico conforme já mencionado.