

umaFrancesinha

Relatório Final



Universidade do Porto

Faculdade de Engenharia

FEUP

Mestrado Integrado em Engenharia Informática e Computação

Sistemas Distribuídos

Grupo 10:

Paulo Bordalo Marcos – 201100759

João Manuel Ferreira Trindade – 201109221

Pedro Miguel Sousa Santos - 201102999

Rui Pedro Meneses da Rosa Neves – 201109350

Faculdade de Engenharia da Universidade do Porto
Rua Roberto Frias, sn, 4200-465 Porto, Portugal

31 de Maio de 2014

Introdução

umaFrancesinha, é um serviço de avaliação de restaurantes destinado a entusiastas e apreciadores deste tão tradicional prato portuense. Esta aplicação permite que os utilizadores, quando apreciam esta iguaria, possam avaliar a qualidade da confecção atribuindo uma classificação ao restaurante, podendo também incluir uma breve descrição e uma fotografia. O serviço permite assim, a um utilizador, consultar restaurantes numa determinada zona, ou por nome e ver assim, a classificação deixada por outros utilizadores, podendo assim tomar uma decisão mais informada, alargando os seus horizontes gastronómicos.

Este projecto divide-se em três módulos dependentes: a aplicação do utilizador final, em que este pode consultar restaurantes, atribuir classificações, deixar comentários ou inserir fotografias; o segundo módulo é destinado aos proprietários de restaurantes, estes podem utilizar a nossa plataforma para divulgar promoções/eventos a toda a nossa base de utilizadores na forma de notificações; o terceiro módulo diz respeito ao sistema de controlo que gere a base de dados e estabelece a comunicação entre os diferentes módulos.

A diferença mais significativa entre a proposta inicial e o resultado final é a plataforma a que as aplicações se destinavam. A proposta inicial consistia numa aplicação destinada a dispositivos móveis Android, enquanto a versão final agora apresentada, destina-se a computadores pessoais. Esta alteração deveu-se à falta de conhecimento e domínio da programação da API do Android. Como a finalidade desta unidade curricular é interiorizar os conhecimentos de sistemas distribuídos e não a programação de dispositivos moveis, optamos então por fazer esta alteração. De resto, todas as funcionalidades apresentadas na proposta inicial foram cumpridas.

Índice

Introdução.....	2
Índice	3
O Problema.....	4
Funcionalidades	4
O Desenvolvimento	4
A solução	5
Elementos Extra / Dificuldades	10
Conclusão	10

O Problema

A dificuldade do sistema que desenvolvemos reside na comunicação entre as partes, cliente-servidor, na gestão e no tratamento da informação recebida.

A aplicação cliente, sempre que é executada, precisa de fazer pedidos ao servidor para assim obter informação que possa disponibilizar ao utilizador. Isto é válido tanto para a aplicação-cliente dos utilizadores finais, como dos proprietários de restaurantes. Esta troca de informação poderia levar a problemas de sincronização.

O sistema deve ainda garantir que os utilizadores são notificados quando um proprietário de restaurante decide partilhar uma promoção/desconto.

Outro problema encontrado era a possibilidade de um utilizador mal-intencionado, poder atribuir classificações negativas a todos os restaurantes, numa tentativa de ter o seu restaurante como melhor classificado. Uma possível solução passaria por introduzir moderadores humanos que avaliassem o comportamento de cada utilizador. Para poupar os custos envolvidos com isso, o nosso serviço trata automaticamente de identificar estes utilizadores que tentam denegrir restaurantes com votações.

Funcionalidades

A lista seguinte introduz as principais funcionalidades do nosso serviço:

- Votação em restaurante com nota, comentário e fotografia.
- Pesquisa de restaurantes por nome ou cidade.
- Consulta de informação geral do restaurante.
- Consulta de classificações mais recentes.
- O utilizador pode introduzir no sistema um restaurante caso ainda não exista.
- O utilizador é notificado sempre que os restaurantes em que deu nota positiva tenham novidades.
- Controlo de votações negativas.
- Gestão geral de dados de utilizador e preferências.

O Desenvolvimento

O desenvolvimento foi feito na linguagem Java, utilizou-se o IDE Eclipse, e incluiu-se as bibliotecas externas de tratamento de JSON e SQLITE.

A base de dados utilizada foi SQLITE, na última versão disponível.

No desenvolvimento gráfico utilizou-se a extensão WindowBuilder para o IDE Eclipse, e a framework Swing.

A solução

A nossa solução consiste numa implementação de um tradicional sistema de cliente servidor.

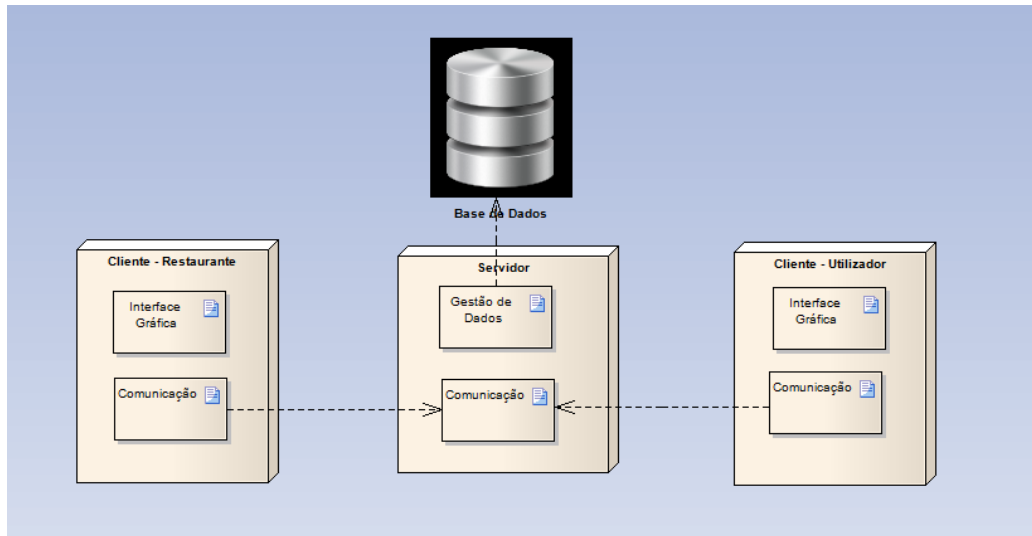
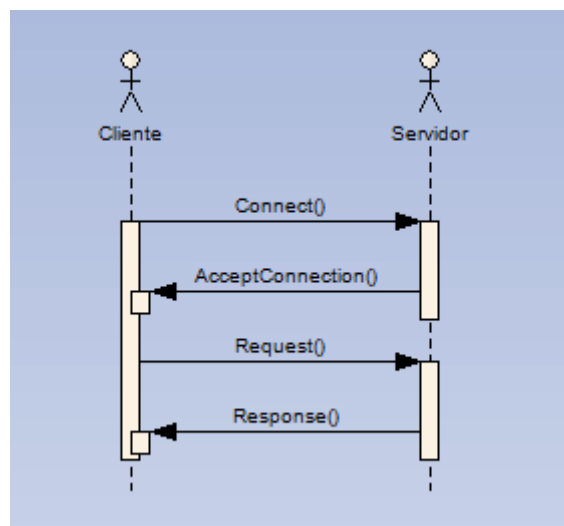


Imagem 1: Diagrama de Desenvolvimento

Na parte da comunicação entre os nossos módulos utilizamos o protocolo REST. Este é um protocolo de comunicação entre cliente e servidor em que o cliente efetua pedidos ao servidor. Estes pedidos devem apenas ser do tipo “GET”, “POST”, “PUT” ou “DELETE”. Os pedidos do tipo “GET” destinam-se à obtenção de informação sem envio de qualquer tipo de objecto, o servidor responde a estes pedidos enviado uma resposta no formato JSON. Os pedidos POST destinam-se a actualizar os dados no servidor, nos pedidos POST, é o cliente que envia para o servidor um objecto em formato JSON com a nova informação, que depois é tratada pelo servidor. Os pedidos do tipo “PUT” são para inserção de nova informação no servidor e também é enviado o objecto JSON por parte do cliente, respondendo o servidor com códigos, indicando a existência de erros ou não. Por último a mensagem DELETE destina-se a informar o servidor de uma informação que deve ser apagada.

O caso de uso que descreve este funcionamento é descrito assim:



Foi utilizado o *toolkit* do modo gráfico do java, SWING, para o nosso projecto ter uma interface não só bonita como acessível a quem a utilize. Em baixo estão algumas imagens dos módulos que efectuamos:

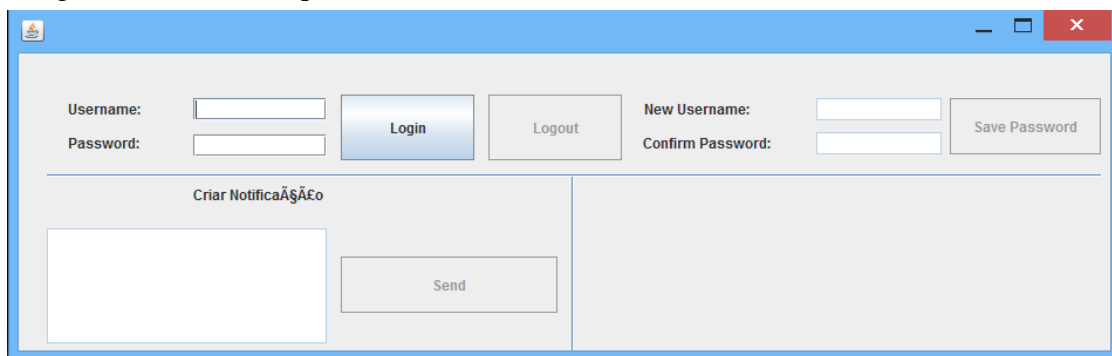


Imagem 2: Cliente para Gestão de Restaurantes.



Imagem3: Página Inicial Cliente Utilizador.

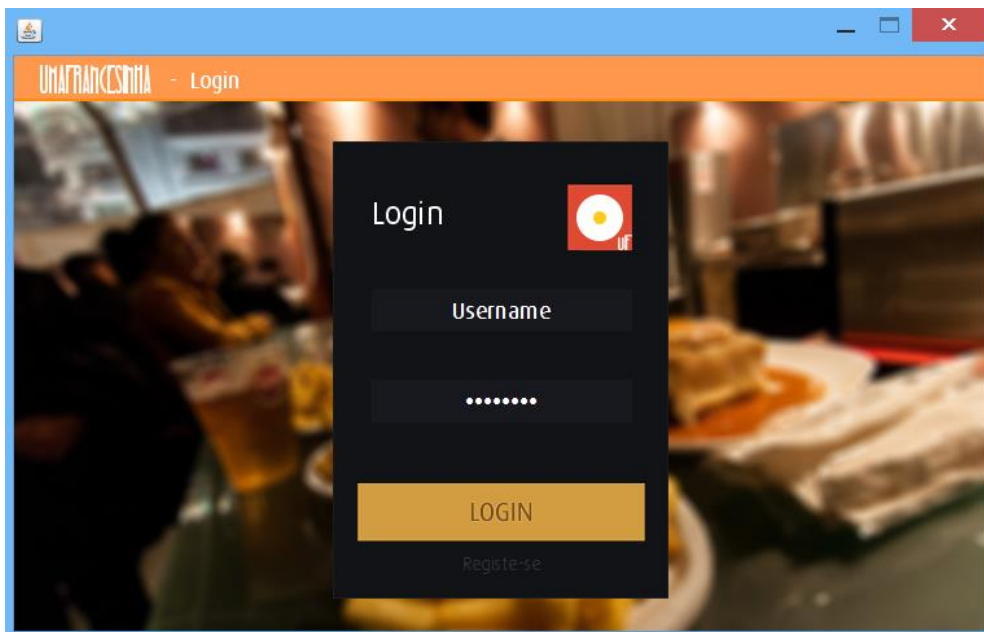


Image4: Pagina de Login de Cliente Utilizador.

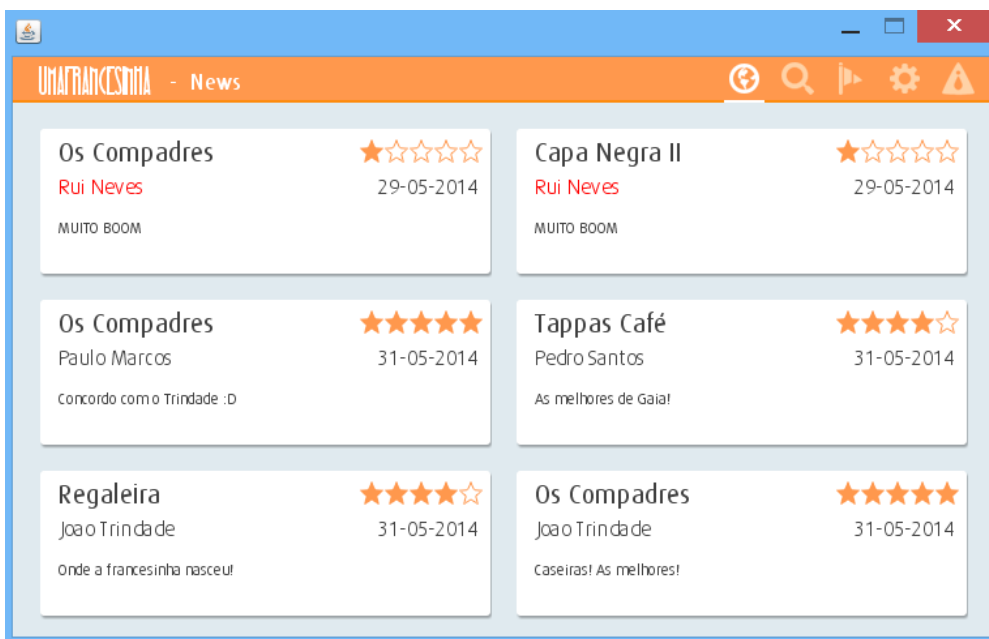


Imagem5: Últimas notícias de Cliente Utilizador.

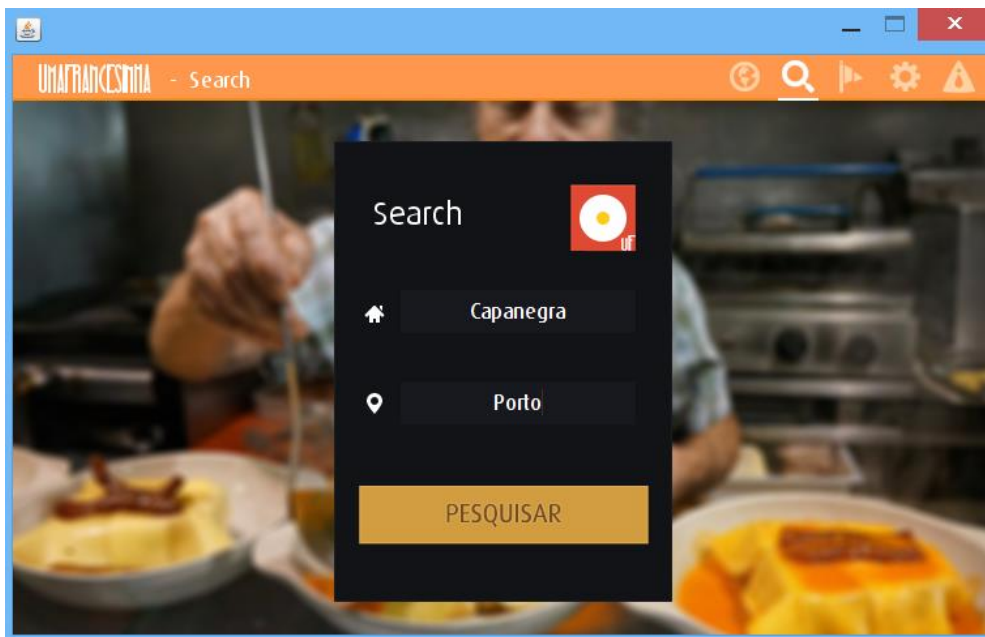


Imagem6: Página de Pesquisa de Cliente Utilizador.

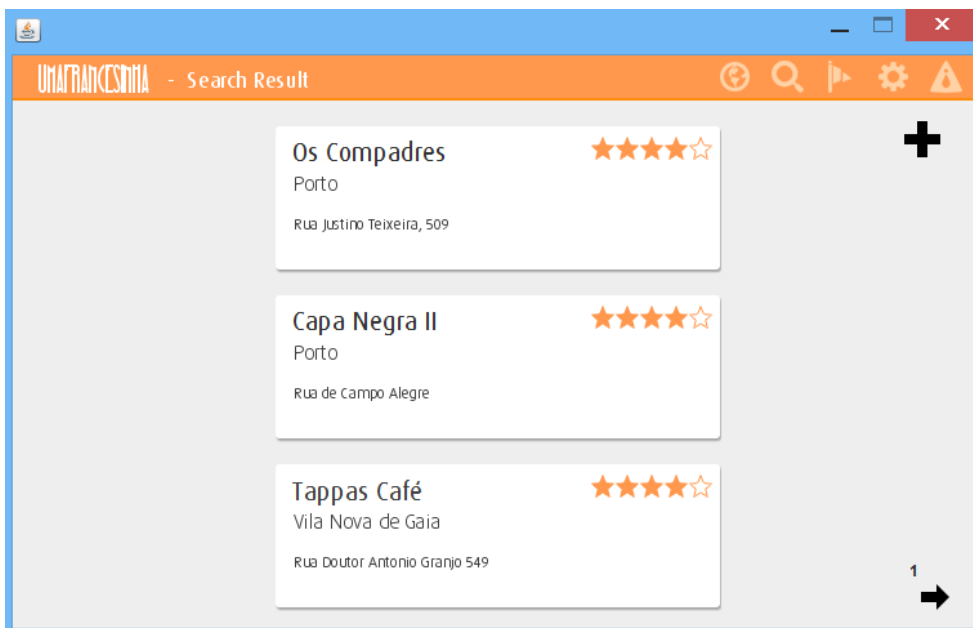


Imagem7: Página de Apresentação de Resultados de pesquisa em Cliente Utilizador.



Imagem8: Página de Apresentação de Restaurante e Inserção de Avaliação (incluindo visualização de fotos das últimas duas avaliações) do Cliente Utilizador.

Elementos Extra / Dificuldades

No lado do servidor, foi essencial termos a leitura do “socket” em “threads”, visto que é um servidor e poderão ser múltiplos os acessos (diversos clientes). A nossa “thread” será para cada cliente e tratará de ler a informação que é enviada pelo “socket” e consequente interpretação. Sempre que uma thread termina inesperadamente, é tratada a excepção lançada.

Visto que implementamos um serviço em que necessitamos de enviar e receber imagens, implementamos um envio pelo protocolo. O cliente envia um “PUT” com o nome da imagem que quer pôr no servidor e o servidor automaticamente fica à espera da imagem enviada por um “ObjectOutputStream”. Com esta funcionalidade, poderiam haver imagens com o mesmo nome na base de dados e no espaço físico do servidor, visto que é um serviço multi-thread. Para isto, encapsulamos no nome da imagem a data, horas, minutos, segundos e milissegundos para garantir que não irão duas imagens com o mesmo nome para o servidor.

Conclusão

Depois de toda a execução deste projecto conseguimos retirar alguma informação sobre o funcionamento de alguns protocolos e verificações essenciais para que as comunicações decorram com a maior normalidade. As maiores dificuldades residiram nos aspectos de sincronização multi-thread para garantir o funcionamento com vários clientes em simultâneo com um mesmo servidor.

Outra dificuldade encontrada durante o desenvolvimento residiu com o envio e manuseamento das fotografias. O envio destes objectos com grande dimensão foi algo problemático e levou a adaptações no processo de envio de mensagens.

No final do projecto, a nossa decisão na substituição da plataforma de desenvolvimento mostrou-se acertada, não pondo de parte o desenvolvimento futuro da aplicação móvel, conseguimos no espaço de tempo dado, uma melhor, e mais robusta aplicação.

Futuramente, e uma vez que o modulo do servidor está a funcionar plenamente, a migração para dispositivos moveis seria bastante mais célere, e acessível, nomeadamente no que diz respeito a Android, dada a semelhança na linguagem de programação.