

GESTÃO DE DESPESAS DE PARTILHADAS

Henrique Nobre de Figueiredo

isep Instituto Superior de
Engenharia do Porto

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

2016

Relatório do trabalho efetuado no âmbito da disciplina de Redes Inteligentes e Serviços do
1º Ano do Curso de Mestrado em Engenharia Eletrotécnica e de Computadores
Área de Especialização de Telecomunicações



Departamento de Engenharia Eletrotécnica
Instituto Superior de Engenharia do Porto
14 de janeiro de 2016

Índice

ÍNDICE	V
ÍNDICE DE FIGURAS.....	VII
ÍNDICE DE TABELAS	IX
ACRÓNIMOS	XI
1. INTRODUÇÃO	1
1.1. CONTEXTUALIZAÇÃO	1
1.2. OBJETIVOS.....	1
1.3. ORGANIZAÇÃO DO RELATÓRIO.....	2
2. MODELOS DE CASOS DE USO.....	5
2.1. CASOS DE Uso.....	5
3. APLICAÇÃO.....	9
3.1. ARQUITETURA	9
3.2. BASE DE DADOS	10
3.3. <i>BACK-END</i>	12
3.4. <i>FRONT-END</i>	13
4. TESTES E RESULTADOS	17
4.1. REGISTO	17
4.2. CONSULTA.....	17
4.3. CRIAÇÃO.....	18
5. CONCLUSÕES	19
REFERÊNCIAS DOCUMENTAIS.....	21

Índice de Figuras

Figura 1	Casos de Uso de Registo e Administração	6
Figura 2	Processo de criação de listas e convite de utilizadores para a mesma	7
Figura 3	Processo de criação de submissão	7
Figura 4	Processo de eliminação de submissão.	8
Figura 5	Fecho de contas de uma lista	8
Figura 6	Exemplo de arquitetura de uma Aplicação <i>Web</i>	9
Figura 7	Interação do <i>back-end</i> em Perl com a Base de Dados e Servidor MIME assim como a interação com o cliente.....	13
Figura 8	Página de <i>Login</i>	14
Figura 9	Página de Registo	14
Figura 10	Página "MinhasListas" de um utilizador com 2 listas subscritas.....	15
Figura 11	Página "MinhaLista" da "Lista de Rinte 2"	15
Figura 12	Página de criação de submissões	16
Figura 13	Página de convite de utilizadores	16
Figura 14	<i>Pop-up</i> para confirmação de "Acerto de contas".....	16
Figura 15	Página "lista" apenas acedível por um Administrador da aplicação.....	16

Índice de Tabelas

Tabela 1	<i>Users</i>	10
Tabela 2	Autenticacao.....	11
Tabela 3	Listas	11
Tabela 4	Submissao.....	12

Acrónimos

BD - Base de dados

HTTP - *HyperText Transfer Protocol*

SQL - *Structured Query Language*

HTML - *HyperText Markup Language*

CSS - *Cascading Style Sheets*

MIME - *Multipurpose Internet Mail Extensions*

1. INTRODUÇÃO

1.1. CONTEXTUALIZAÇÃO

No âmbito da disciplina de Redes Inteligentes e serviços, após o estudo de tecnologias na área das aplicações e serviços *Web* surgiu a necessidade de desenvolver uma aplicação *Web*. Assim, este relatório pretende apresentar uma aplicação desenvolvida assim como todo o sistema a ela envolvente que permite o seu correto funcionamento [1].

O tema “Gestão de Despesas de Partilhadas” foi proposto com o objetivo de tentar desenvolver uma plataforma apontada para uma gestão económica de despesas partilhadas pelos utilizadores. A ideia surge do elevado numero de estudantes que partilham casa e consequentemente, despesas. A utilização desta plataforma permite ainda *updates* regulares sobre alterações de saldo por email, automatizando assim a sua autogestão [7].

1.2. OBJETIVOS

O objetivo principal deste trabalho consiste em proporcionar a qualquer pessoa que pretenda ter despesas partilhadas com um ou mais utilizadores uma plataforma para gestão de saldos, através de um serviço de submissões. Para isso dividiu-se este objetivo em múltiplas tarefas, como as seguintes:

- Desenvolvimento de casos de uso;
- Estruturação de uma Base de Dados (BD);
- Desenvolvimento de *back-end* da aplicação;

- Desenvolvimento de *front-end* da aplicação;
- Testes e respetivos resultados.

1.3. ORGANIZAÇÃO DO RELATÓRIO

Este relatório encontra-se dividido em cinco capítulos. No primeiro capítulo é feita uma introdução ao tema desenvolvido neste trabalho. O segundo capítulo apresenta os casos de uso da aplicação. O terceiro capítulo diz respeito à aplicação em si, à sua arquitetura, à base de dados que a suporta e ao seu desenvolvimento. No capítulo 4 são mencionados alguns testes efetuados e modos de operação. No capítulo final, são feitas algumas conclusões relativas à aplicação desenvolvida e mencionadas algumas possíveis melhorias ou acréscimo de funcionalidades.

2. MODELOS DE CASOS DE USO

Os modelos de casos de uso são uma ferramenta que descreve como diferentes tipos de utilizadores interagem com um sistema para solucionar um problema. Assim, a partir desta ferramenta é possível obter informações como as metas dos utilizadores, as interações entre os utilizadores e o sistema e o comportamento necessário do sistema para satisfazer as metas. Os elementos de modelo mais importantes são: casos de uso, atores e relações entre eles.

2.1. CASOS DE USO

Os casos de uso são um elemento de modelo que representam uma série de funcionalidades que servem de interação entre os diversos atores. Este elemento é composto pelo nome do caso de uso e pela sua especificação.

No caso desta aplicação, a parte Administrativa apenas tem uma função descrita no ponto 2.1.1, isto pois se pretende que a aplicação seja o mais autónoma possível. Assim sendo de seguida são apresentados os casos de uso dos utilizadores comuns: Registo (que será apresentado juntamente com a Administração pelas suas interdependências), Criação de

listas/Convite de Utilizadores para listas, Criação de submissões, Eliminação de submissões e Fecho das contas.

2.1.1. ADMINISTRAÇÃO & REGISTO

A administração da aplicação está diretamente relacionada com o registo, a função administrativa consiste na verificação da veracidade dos utilizadores, isto é, tem a capacidade de verificar quais são os utilizadores existentes, assim como eliminar (Marcar como eliminado na base de dados) utilizadores não verificados por email. O registo de um utilizador comum passará por um formulário no qual registará um email que terá de conformar para o subsequente *login*. A Figura 1 é uma representação do processo de registo e administração na aplicação.

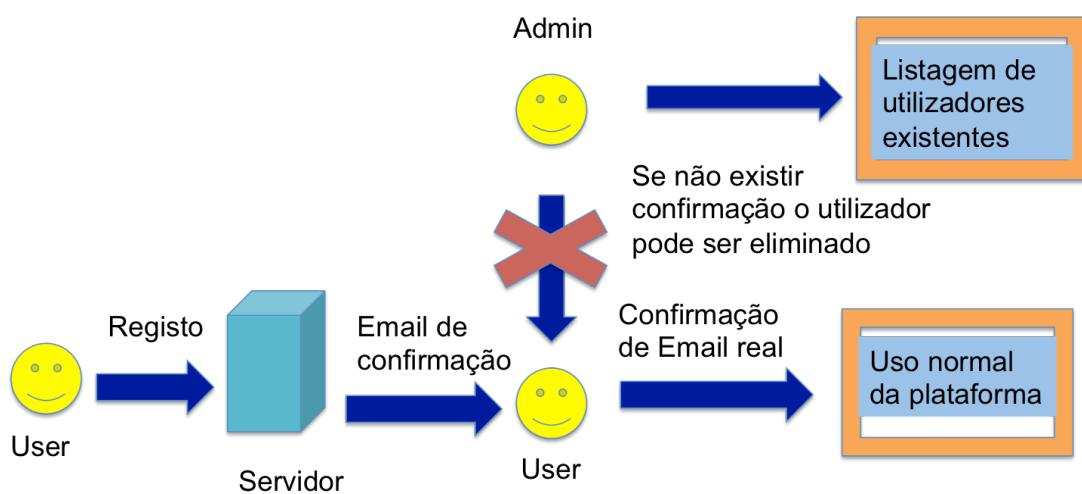


Figura 1 Casos de Uso de Registo e Administração

2.1.2. CRIAÇÃO/CONVITE DE UTILIZADORES PARA LISTAS

A criação de listas assim como o convite de utilizadores é feita de forma semelhante e daí estar agrupada num tópico. A criação da lista apenas implica atribuir um nome à mesma, enquanto que após a criação bastará inserir o email do utilizador a convidar para a lista (sendo que este carece de confirmação). A Figura 2 é uma representação gráfica destes processos.

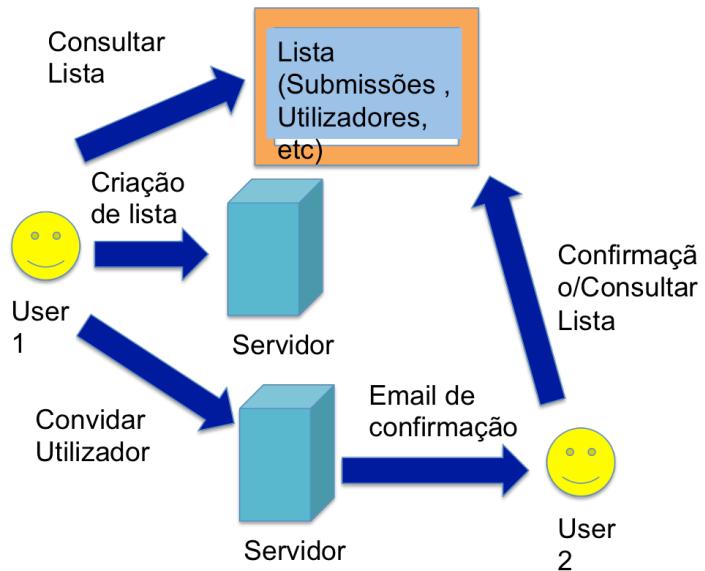


Figura 2 Processo de criação de listas e convite de utilizadores para a mesma

2.1.3. CRIAÇÃO DE SUBMISSÕES

A criação de submissões é uma das tarefas mais importantes da aplicação, esta consiste num conjunto de funções que inserem a submissão na base de dados e regularizam os valores dos saldos dos utilizadores, enviando-lhes uma notificação por email caso estes o desejem. A criação de submissões torna-se a tarefa mais complexa desta aplicação. A Figura 3 mostra uma representação simples deste processo.

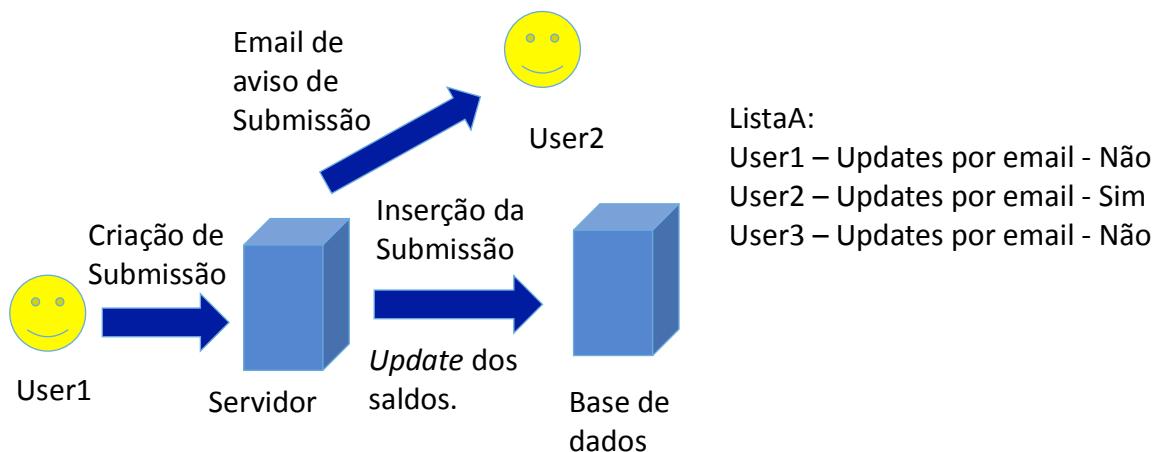


Figura 3 Processo de criação de submissão

2.1.4. ELIMINAÇÃO DE SUBMISSÕES

A eliminação de submissões é uma tarefa análoga à criação de submissões, sendo que serão feitos os *updates* nos saldos novamente e a submissão é marcada como eliminada. Na Figura 4 podemos ver a tarefa de eliminação.

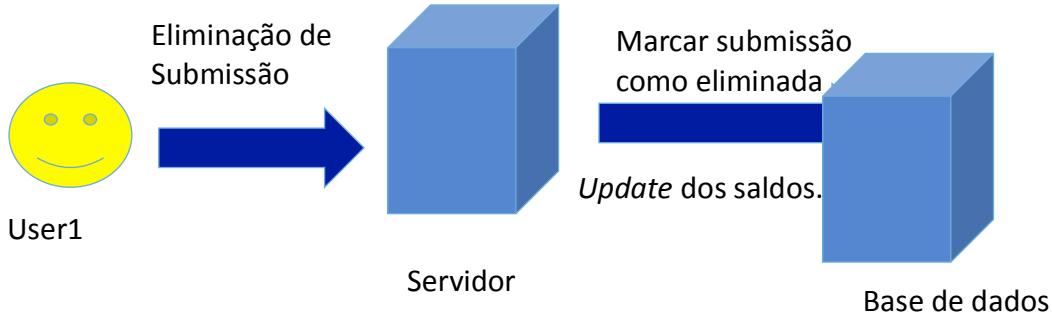


Figura 4 Processo de eliminação de submissões.

2.1.5. FECHO DAS CONTAS

Fechar as contas simplesmente faz um *reset* ao saldo de todos os utilizadores da lista, assim como marca todas as submissões da lista como eliminadas. Consequentemente todos os membros da lista serão notificados por email. Na Figura 5 podemos ver uma representação de um fecho de contas.

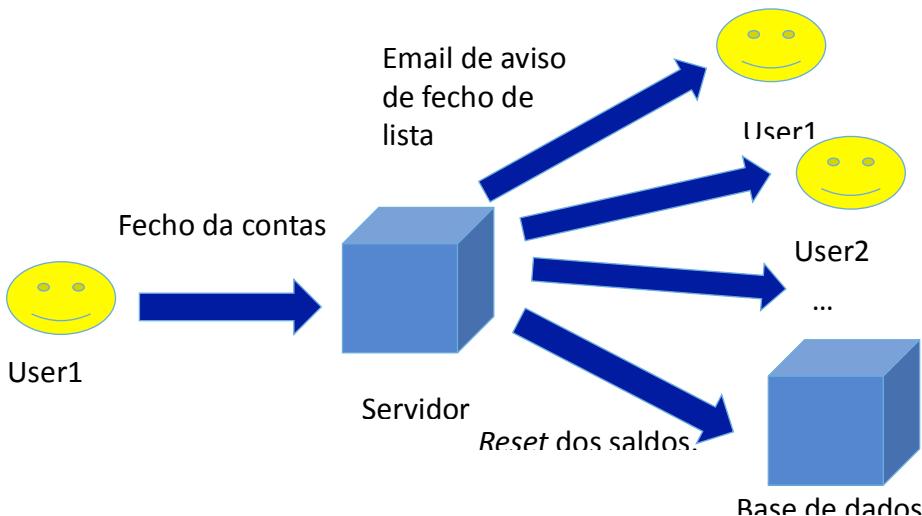


Figura 5 Fecho de contas de uma lista

3. APLICAÇÃO

3.1. ARQUITETURA

Tratando-se este trabalho de uma aplicação *Web*, esta foi desenvolvida tendo em conta o modelo atual da arquitetura das aplicações *Web*. Esta arquitetura é baseada no modelo cliente/servidor sendo composta por três camadas. A primeira camada é composta pela aplicação cliente, que neste caso será um navegador *Web*. É nesta camada que também são executados os *scripts* que tornam as páginas dinâmicas (Sendo que neste caso a maioria das páginas são estáticas) assim como tratamento gráfico das páginas. A segunda camada diz respeito à aplicação servidora, aquela que implementa as funcionalidades de atendimento dos pedidos dos clientes gerando respostas dinâmicas. Na aplicação criada, esta camada será composta por um ficheiro Perl que recebendo um pedido *HyperText Transfer Protocol* (HTTP) acedem à base de dados presente na terceira camada de modo a efetuar o que lhe foi requisitado no pedido HTTP [2]. A terceira camada representa um servidor de base de dados. Nesta camada é onde se encontra a base de dados que será apresentada na subsecção 3.2 deste relatório. A Figura 6 representa a arquitetura de uma aplicação *Web* atualmente.

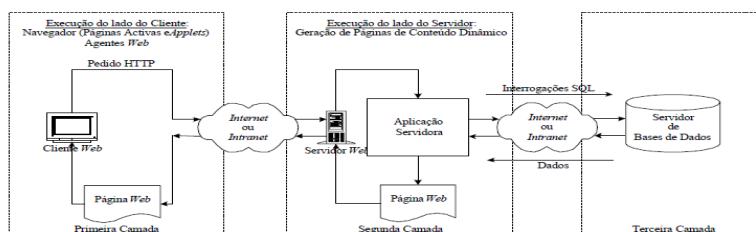


Figura 6 Exemplo de arquitetura de uma Aplicação *Web*

3.2. BASE DE DADOS

Após o desenvolvimento de cada caso de uso, passou-se para o desenvolvimento da base de dados. Nela foram criadas 4 tabelas sendo que duas são específicas do utilizador e outras duas são para listas e submissões respectivamente.

3.2.1. *USERS*

Esta tabela serve para guardar a informação pessoal do utilizador. É composta por 12 colunas. A primeira coluna é a coluna do “Nro_user”, é uma coluna de números inteiros cujo valor é auto incrementado, com o objectivo de atribuir um número único para cada utilizador. As colunas da segunda à sétima contêm a informação pessoal de cada utilizador: primeiro nome (F_name), último nome (L_name), email, data de nascimento (Bday), sexo (sex) e cidade (City). Em oitavo lugar temos a opção de o utilizador ser administrador do servidor (SU). A nona e a décima coluna são respectivas a emails automáticos quando alguém submete alguma numa lista do utilizador (*Updates*) ou quando o saldo do utilizador numa lista fica negativo (*Alert*). As ultimas duas colunas são a verificação por email da conta (Verificado) e a verificação da desativação da conta esteja (Existente).

Na Tabela 1 podemos ver um exemplo de *output* desta tabela.

Tabela 1 *Users*

Nro_User	F_name	L_name	Email	Bday	Sex	City	SU	Updates	Alert	Verificado	Existente
1	Henrique	Figueiredo	nf.henrique@gmail.com	1994-01-02	0	Braganca	1	1	1	1	1
2	Silvia	Nobre	nobresi@gmail.com	1959-10-27	1	Braganca	0	1	1	1	1

3.2.2. AUTENTICACAO

Esta tabela é uma segurança de modo a não misturar dados pessoais com a palavra-passe da conta. Em conjunto com a tabela *Users*, esta permite que se faça *login* na página e consequentemente a aceder a todas as funcionalidades da aplicação. É composta por 2 colunas, a coluna “Nro_User” que faz referência à tabela *Users* e a coluna “Password” que guarda a palavra-passe associada à conta com um determinado “Nro_User”.

A Tabela 2 mostra nos um exemplo de dois utilizadores registados.

Tabela 2 Autenticacao

Nro_User	Password
1	root
2	mae

3.2.3. LISTAS

A Tabela 3 serve para gerir cada lista criada pelos utilizadores, sendo que cada entrada corresponde a um utilizador de modo a ser usável mais eficientemente. É composta por 5 colunas. A primeira é a “Nro_list” que representa um número específico associado a cada lista. Apesar desta característica este valor não pode ser auto incrementado à semelhança das duas tabelas anteriores, pois como referido anteriormente cada entrada refere-se a um utilizador. Assim sendo deve-se programar com precaução de modo a não repetir números para listas diferentes. A segunda coluna “Nro_User” identifica o utilizador pertencente a determinada lista (Que é referente, é claro, à primeira tabela *Users*). A terceira coluna é o “Saldo” do utilizador nessa lista, este valor encontra-se em euros vezes mil, de modo a que os valores na coluna possam ser do tipo inteiro. A quarta coluna trata-se do “Nome” da lista e a quinta é uma coluna de verificação da existência da lista “Fechada”.

Tabela 3 Listas

Nro_list	Nro_User	Saldo	Nome	Fechada
1	2	0	Lista de compras Afonso Costa	0
1	3	601	Lista de compras Afonso Costa	0
1	1	-601	Lista de compras Afonso Costa	0

3.2.4. SUBMISSAO

Esta tabela mantém o registo de todas as submissões. É composta por 8 colunas. À semelhança das tabelas anteriores, a primeira coluna trata-se de uma coluna composta por números inteiros os quais são auto incrementados à medida que novas entradas são adicionadas na tabela, isto serve de identificação para cada submissão. A segunda coluna identifica quem fez a submissão pelo seu número (Nro_User). A coluna 3 é a lista onde a submissão foi feita. As colunas 4 e 5 apresentada a informação relativa a cada submissão (“Data”, “Descricao”). A coluna “Pagantes” tem um multiplicador de uso da submissão por

cada utilizador (Número de utilizador) da lista. As ultimas duas colunas (“Eliminada” e “Valor”), são referentes à existência da submissão e o valor submetido vezes 100 em euros. A Tabela 4 permite perceber melhor a estrutura da tabela.

Tabela 4 Submissao

Sub_id	Nro_User	Nro_list	Data	Descricao	Pagantes	Eliminada	Valor
1	3	1	2016-01-03	Compras	0x2,2x3,1x1	1	5000
3	3	1	2016-01-06	Coisas de brincar	0x2,1x3,2x1	0	1230

3.3. BACK-END

O conceito de *back-end* e o conceito de *front-end* servem para distinguir o que se refere à camada de acesso aos dados e o que se refere à apresentação da aplicação. O *back-end* é a camada de acesso aos dados. É a partir dela que o *front-end* disponibiliza a informação necessária ao seu utilizador.

Na aplicação desenvolvida o *back-end* é composto um ficheiro principal Perl (.psgi (Módulo Dancer2)) composto por uma série de funções para cada caso de uso da aplicação. Esta camada permite-nos fazer a ligação com as bases de dados *My Structured Query Language* (MySQL) assim como toda a lógica referente ao tratamento dos dados. Pelas características de aplicabilidade de Perl, certa parte do *front-end* é feito neste ficheiro principal, pois se torna a forma mais fácil de integrar os dados das bases de dados em *HyperText Markup Language* (HTML). É importante referir também que por estarmos a utilizar o Módulo Dancer2 e este nos criar os diretórios e subdiretórios da aplicação, toda a parte de front-end a seguir referida é integrada nestes diretórios, facilitando o seu acesso pelo *back-end*. Pelas funcionalidades deste módulo foram integrados diferentes tipos de pedidos HTTP, *POST* e *GET*.

As funcionalidades implementadas em Perl no *back-end* são as seguintes:

- Listar e criar os utilizadores gerais da aplicação;
- Listar e criar listas;
- Convidar utilizadores para listas;

- Confirmar convites e registos por e-mail [3];
- Eliminar utilizadores não confirmados (Função de Administração);
- Listar e criar submissões;
- Eliminar listas e submissões (Marcar como eliminadas).

A Figura 5 apresenta o esquema de conexão entre os diversos componentes de *back-end* desenvolvidos na aplicação.

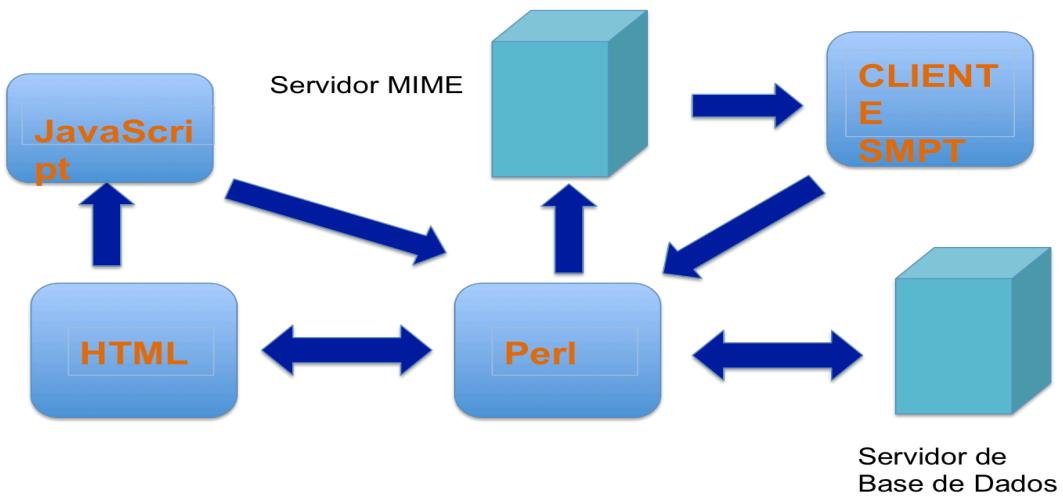


Figura 7 Interação do *back-end* em Perl com a Base de Dados e Servidor MIME assim como a interação com o cliente.

3.4. FRONT-END

O *front-end* é a camada que serve de interação com o utilizador, é através desta que o cliente comunica com o *back-end*.

Na aplicação desenvolvida o *front-end* foi desenvolvido em HTML [4]. Usou-se HTML5 para se tirar o máximo partido desta linguagem que em conjunto com o *Cascading Style Sheets* (CSS) 3 foi possível desenhar uma interface gráfica simplista mas funcional e bastante apelativa [5]. A interface desenhada, dependendo do tipo de utilizador poderá ter acesso a mais ou menos páginas desenvolvidas. Foram desenvolvidas variadas páginas com o mesmo *template* gráfico, mantendo o Navegador, o Cabeçalho e o Rodapé constantes ao longo do trabalho.

A primeira vez que se acede à aplicação seremos forçados a fazer o *login* ou a registar-mos nos dado que todas as outras partes da aplicação se encontram fechadas enquanto a sessão não for iniciada.

Na Figura 7 e Figura 8 podemos ver a página de *login* e registo, respectivamente.

The screenshot shows a dark-themed web application interface. At the top center is the title "Quem paga o quê?!" in green. On the left, there is a sidebar with a "HOME" icon and input fields for "Email" and "Password", followed by a "Login" button and links for "Não tem credenciais?" and "Registe-se". The main content area contains a brief summary of the project: "Este trabalho foi desenvolvido no âmbito da cadeira Redes Inteligentes e Serviços, tem como objectivo a criação de uma Aplicação Web. Foi projectada a criação de uma aplicação de gestão de recursos económicos de pessoas com despesas partilhadas. Este sistema implica o registo e login e implica ainda o uso de um e-mail válido. Se tiverem questões relativas à aplicação por favor consultem o relatório referente ao mesmo! Espero que se divirtam! :D". At the bottom right, it says "Copyright Henrique Figueiredo".

Figura 8 Página de Login

This screenshot shows the registration form on the same dark-themed application. The title "Quem paga o quê?!" is at the top. The registration form includes fields for "Primeiro nome" (First name) and "Sobrenome" (Last name), gender selection ("Sexo: M F"), date of birth ("Data de nascimento: day month year"), city ("Cidade: Cidade"), email ("E-mail: Email") and password ("Password"). There are also two checkboxes for receiving updates and alerts about balance changes, and buttons for "Registrar" (Register) and "Já está registrado? Faça login" (Already registered? Log in). The sidebar on the left is identical to the login page. The bottom right corner credits "Copyright Henrique Figueiredo".

Figura 9 Página de Registo

Após efetuado o registo e de volta à página *Login*, o utilizador efetua a sua autenticação. Caso a autenticação esteja correta, é diretamente redirecionado para a página “MinhasListas”. Nesta página o utilizador tem acesso a uma série de informação e funcionalidades. Consegue ver as listas que já criou e o saldo relativo a elas, assim como

pode criar novas listas, através deste menu conseguimos também aceder a uma página chamada “MinhaLista” que é referente aos dados específicos de cada lista.

Na Figura 9 podemos ver a página “MinhasListas” e na Figura 10 podemos ver a página “MinhaLista” referente à Lista 2 da Figura 9.’

Numero	Nome	O meu saldo	Saldo mais Elevado	Saldo mais Baixo
1	Lista de Rinte	-4.22 €	8.43 €	-4.22 €
2	Lista de Rinte 2	7.45 €	7.45 €	-7.45 €

Figura 10 Página "MinhasListas" de um utilizador com 2 listas subscritas.

Quem Pagou:	Data	Valor	Descrição	Pagantes	Eliminada
Henrique Figueiredo	2016-01-08	22.35 €	Lavandaria+ Carne pro jantar	John x1, Henrique x2,	X

Figura 11 Página "MinhaLista" da "Lista de Rinte 2"

A página apresentada acima é a que permite uma maior utilização por parte do utilizador, isto porque nesta página podemos adicionar submissões (Botão “Nova Submissão”), eliminar submissões (Cruz à frente de cada submissão), adicionar utilizador à lista e finalmente podemos ainda “Acertar as Contas” que fecha a Lista.

Na Figura 11,12 e 13 podemos ver as 3 páginas, Nova submissão, Adicionar Utilizador e o Fecho de contas.



Figura 12 Página de criação de submissões

Figura 13 Página de convite de utilizadores

Quem Pagou:	Data	Valor	Descrição	Pagantes	Eliminada
Henrique Figueiredo	2016-01-08	22.35 €	Lavandaria+ Carne pro jantar	John x1, Henrique x2,	X

Figura 14 Pop-up para confirmação de "Acerto de contas"

Finalmente temos uma página puramente administrativa. Quando o *login* é feito, caso o utilizador seja administrador aparece mais uma opção no navegador para “Listar Users”. Esta opção lista os dados pessoais de todos os utilizadores (com a exceção da password) e permite ao administrador eliminar todos os utilizadores que não tenham validado o seu Email.

Na Figura 14 podemos ver esta página de grupo de utilizadores de exemplo.

Número	Primeiro Nome	SobreNome	Email	Data de nascimento	Sexo	Cidade	SU	Updates	Alertas	Verificado	Existente
1	Henrique	Figueiredo	nf.henrique@gmail.com	1994-01-02	M	Braganca	Sim	Sim	Sim	Sim	Sim
2	Silvia	Nobre	nobres@gmail.com	1959-10-27	F	Braganca	Não	Sim	Sim	Sim	Sim
3	John	Cena	johncenaff@gmail.com	1978-10-16	M	Dallas	Não	Sim	Sim	Sim	Sim
4	A	B	a@gmail.com	2016-02-01	F	Oliveira de azemeas	Não	Não	Não	Sim	Sim

Figura 15 Página "lista" apenas acedível por um Administrador da aplicação.

4. TESTES E RESULTADOS

Este capítulo tem como objetivo revelar alguns testes efetuados assim como os resultados obtidos. Divida-se então este capítulo pelas em 3 funcionalidades principais da aplicação: Registo, Consulta e Criação.

4.1. REGISTO

O Registo de utilizadores nesta aplicação está completamente automatizado para utilizadores gerais, sendo que antes da sua submissão, o seu email é verificado de modo a demonstrar que este é real. Para registar um utilizador com permissões de Administrador, o mesmo terá de ser feito manualmente na base de dados, de modo a não comprometer a segurança da aplicação.

4.2. CONSULTA

Para qualquer utilizador que efetue o *login* poderá consultar todas as suas listas assim como submissões em cada lista e utilizadores em cada lista.

Se uma lista se encontrar fechada, a mesma não será visível por nenhum utilizador, esta opção é uma salvaguarda aos utilizadores estarem a abrir e fechar listas constantemente (gerando uma quantidade enorme de emails automáticos).

No caso das submissões, caso as mesmas se encontrem eliminadas, poderão ser visualizadas, mas com um traço por cima e todas as suas utilidades estarão desativadas.

Finalmente poderá ser ainda consultada a lista dos utilizadores, caso se tenha privilégios de Administrador. Esta funcionalidade do trabalho é relativamente pequena, pois pretende-se que a aplicação seja o mais autónoma possível (Sem necessidade de gestão por parte de um Administrador).

4.3. CRIAÇÃO

A tarefa mais complexa desta aplicação, quer pelo número de acessos à Base de dados que esta implica, mas também o implica também acesso ao servidor MIME, pois os emails gerados automaticamente só acontece quando “algo” é criado.

Um utilizador poderá criar uma lista apenas dando-lhe um nome, mas de modo a adicionar pessoas à sua lista, estas terão de confirmar a sua participação por email, assim como deverão ter uma conta na aplicação.

A criação de uma submissão carece de uma data, um valor e um multiplicador por cada utilizador a quem a despesa é relativa, este multiplicador permite fazer divisões de despesas mais precisas de acordo com o uso, sendo que o saldo de os utilizadores é ajustado através destes multiplicadores.

Caso os utilizadores tenham especificado o desejo de receber *updates* e/ou alertas, estes serão gerados quando uma submissão acontece, sendo que o utilizador que criou a submissão não receberá nenhum email, obviamente.

5. CONCLUSÕES

Sendo o trabalho incidente no estudo das aplicações *Web*, serviços *Web* e agentes *Web*, pelo que foi proposto o desenvolvimento de uma aplicação com o tema deste relatório “Gestão de Despesas Partilhadas”. Antes do começo do desenvolvimento da aplicação, foi necessário fazer uma preparação cuidadosa tendo em conta todas as possíveis funcionalidades de modo a que se tornasse o processo de desenvolvimento mais funcional.

Os primeiros objetivos delineados para este trabalho foram o desenvolvimento dos casos de uso e da estruturação da base de dados, sendo que a última não deve ser alterada durante o processo e desenvolvimento pois causa uma grande falta de eficiência. Relativamente ao desenvolvimento da aplicação *Web*, este relatório divide-se em duas secções distintas, a parte de *front-end* e *back-end*. Em termos de planeamento do trabalho o desenvolvimento *front-end* desenvolvido primeiro pois era necessário delinear as secções das páginas. Posteriormente foi desenvolvido na totalidade o *back-end* e finalmente completou-se a aplicação com melhorias gráficas e uso de expressões regulares em *front-end* [6]. Finalmente foi feita uma fase de testes para garantir a integridade da aplicação.

As tecnologias utilizadas no desenvolvimento da aplicação foram escolhidas pela sua vasta aplicabilidade e facilidade de utilização e também pela as possibilidades de utilizar diversos tipos de serviços sem a necessidade de implementar uma tecnologia de ponte.

As aplicações *Web* hoje em dia são muito escaláveis. É possível acrescentar sempre mais funcionalidades tornando-as cada vez mais fiáveis, automatizadas e completas. Alguns aspectos que poderiam ser desenvolvidos futuramente nesta aplicação são por exemplo a opção de reabrir listas e de editar submissões. Alguns problemas de segurança poderiam também ser resolvidos, pois é possível contornar alguns sistemas de validação de email. De modo a melhorar a experiência do utilizador poder-se-ia também melhorar a interface gráfica, nomeadamente em relação a tabelas e botões.

Referências Documentais

- [1] “AW_RINTE_2013.pdf.” Accessed December 28, 2015. https://moodle.isep.ipp.pt/file.php/3142/recursos/AW_RINTE_2013.pdf.
- [2] “Dancer2::Tutorial.” Accessed December 28, 2015. <https://metacpan.org/pod/Dancer2::Tutorial>.
- [3] “Simple.pm.” Accessed December 30, 2015. <http://search.cpan.org/~rjbs>Email-Simple-2.208/lib>Email/Simple.pm>.
- [4] “Html.” Accessed December 26, 2015. <http://www.w3schools.com/html/>.
- [5] “Default.asp.” Accessed January 4, 2016. <http://www.w3schools.com/css/default.asp>.
- [6] “Js.” Accessed January 4, 2016. <http://www.w3schools.com/js/>.
- [7] “Www.wiebetaaltwat.nl.” Accessed December 30, 2015. <https://www.wiebetaaltwat.nl/>.