



Julho de 2011

Projecto final de curso | **Rede Social**



Docente orientador: Professor Alexandre Pereira

Discente: Ruben Tiago Lopes Costa

20083714 – ruben_costa_16@hotmail.com

Índice de conteúdos

Resumo	5
Abstract	6
1. Introdução.....	7
2. Planeamento e enquadramento.....	8
2.1. ASP.NET	8
2.2. Android	9
3. Desenho lógico do sistema.....	10
3.1. Arquitectura e descrição do sistema	10
3.2. Use Case	12
3.3. Lista de Requisitos	13
3.3.1. Requisitos funcionais.....	13
3.3.2. Requisitos não funcionais	15
3.3.3. Requisitos extra (relevantes)	16
4. Diagramas	17
4.1. Diagrama de actividades Login ASP.NET	17
4.2. BPMN Login Android	18
4.3. Diagrama de sequência Publicar imagem ASP.NET	19
4.4. BPMN Publicar imagem Android.....	20
4.5. Modelo de dados	21
4.5.1. Dados pessoais	21
4.5.2. Dados multimédia	21
4.5.3. Informações para o utilizador.....	22
4.5.4. Definições de conta	22
4.5.5. Administração	23
4.5.6. Restantes tabelas	23
5. Android	24

5.1.	Activity	25
5.1.1.	O que é?	25
5.1.2.	Como funciona?.....	26
5.2.	ListActivity - principais dificuldades	28
Considerações finais e perspectivas		32
Referências		34
Glossário		35
Netografia		36
	ASP.NET.....	36
	Android.....	36
	SQL Server.....	37

Índice de ilustrações

ILUSTRAÇÃO 1: CAMADAS DO SOFTWARE ANDROID (SIMPLIFICADO)	9
ILUSTRAÇÃO 2: ARQUITECTURA DO SISTEMA	10
ILUSTRAÇÃO 3: USE CASE	12
ILUSTRAÇÃO 4: DIAGRAMA DE ACTIVIDADES LOGIN ASP.NET	17
ILUSTRAÇÃO 5: BPMN LOGIN ANDROID	18
ILUSTRAÇÃO 6: DIAGRAMA DE SEQUÊNCIA PUBLICAR IMAGEM ASP.NET	19
ILUSTRAÇÃO 7: DIAGRAMA DE SEQUÊNCIA PUBLICAR IMAGEM ASP.NET	20
ILUSTRAÇÃO 8: MODELO DE DADOS – DADOS PESSOAIS	21
ILUSTRAÇÃO 9: MODELO DE DADOS – DADOS MULTIMÉDIA	21
ILUSTRAÇÃO 10: MODELO DE DADOS – INFORMAÇÕES PARA O UTILIZADOR	22
ILUSTRAÇÃO 11: MODELO DE DADOS – DEFINIÇÕES DE CONTA	22
ILUSTRAÇÃO 12: MODELO DE DADOS – ADMINISTRAÇÃO	23
ILUSTRAÇÃO 13: MODELO DE DADOS – RESTANTES TABELAS	23
ILUSTRAÇÃO 14: ECRÃ PRINCIPAL DA APLICAÇÃO ANDROID	25
ILUSTRAÇÃO 15: CICLO DE VIDA DE UMA ACTIVITY	26
ILUSTRAÇÃO 16: EXEMPLO DA ESTRUTURA DA CLASSE ENVIAFOTOS.JAVA	28
ILUSTRAÇÃO 17: ECRÃ DA CLASSE ENVIAFOTOS.JAVA	30

Resumo

Esta é a década em que a internet aumentou exponencialmente de utilizadores, sendo actualmente a primeira escolha para qualquer utilizador no que diz respeito a pesquisa, divertimento e até relacionamento. Com esta evolução e acompanhando as necessidades dos utilizadores, começaram a surgir as actuais Redes Sociais, permitindo assim que as pessoas estejam mais unidas e cada vez mais conectadas umas às outras, com várias formas e tipos de relacionamentos, onde podem partilhar informações, valores, interesses e objectivos em comum.

Este projecto tem como objectivo a realização de uma Rede Social que permite aos seus utilizadores a criação de um perfil onde podem colocar e armazenar conteúdos multimédia (imagens, vídeos e músicas), informações pessoais, podendo ainda comunicar e estar sempre ligados a amigos e potenciais conhecidos. Em acréscimo, foi desenvolvida uma aplicação para telemóveis com sistema Android, que permite aos utilizadores poderem continuar a partilhar conteúdos multimédia e a integração de uma funcionalidade de localização de amigos.

Para este projecto foram então utilizadas diversas tecnologias, através das quais foi criado um site em ASP.NET, a aplicação para telemóvel desenvolvida em JAVA, um Webservice em C# e foi utilizada uma Base de Dados relacional SQL Server para armazenar todo o conteúdo do projecto. Na interface visual foi utilizado CSS e HTML para o layout do site e XML para o layout da aplicação Android, aplicação esta já em fase final de HTML5 para reprodução de vídeo e musica.

Abstract

This is the decade that the Internet users increased exponentially, being the first choice nowadays for any user when it comes to research, fun and even relationship. With this development and monitoring the needs of users began to appear the current Social Networks, allowing the union of people and progressively more connected to each other, with a variety of forms and types of relationship, where they can share information, values, interests and objectives in common.

This project have the purpose the implementation of a Social Network that allows its users to create a profile where they can publish and store multimedia content (images, videos and music), personal information, and can also communicate and stay connected to your friends and potential acquaintance. In addition, was developed an application for Android phones, allowing users to be able to continue sharing multimedia content and the integration of a feature for locating friends.

For this project were used different technologies, where was created an ASP.NET site, the mobile application developed in JAVA, a WebService in C # and was used a SQL Server relational database to store the entire contents of the project. In the visual interface was used CSS and HTML to layout of site and XML for layout of Android application, which is already in the final stages of HTML5 to video and music reproduce.

1. Introdução

Apenas com os conhecimentos adquiridos ao longo da licenciatura em Engenharia Informática e tendo como objectivo a nível profissional continuar a desenvolver software que tenha interface gráfica do utilizador, decidi propor para o presente Projecto Final de Curso, algo que me motivasse e desse prazer em desenvolver, mas que também fosse útil e inovador. De forma a acrescentar valor a este projecto, bem como a nível pessoal, considereei apropriado a realização de um estudo pessoal que incidisse sobre a recente plataforma móvel Android.

Para juntar estas duas tecnologias, decidi realizar algo que tem adquirido cada vez mais utilizadores, as Redes Sociais, onde a ligação entre as duas tecnologias referidas se encontra na partilha de ficheiros multimédia.

2. Planeamento e enquadramento

O projecto consiste num Sistema Distribuído [a] onde Clientes fixos (clientes provenientes de computadores) e Clientes móveis (clientes provenientes de telemóveis) comunicam com o servidor que contém todos os ficheiros necessários para o sistema funcionar. Este sistema é baseado num modelo Cliente/Servidor, onde o cliente faz pedidos para o servidor e este lhe retorna uma resposta.

Este projecto tem a implementação de tecnologias que estão em crescimento e que serão certamente o futuro da Web e dos dispositivos móveis.

2.1. ASP.NET

Para a realização do site é utilizada a plataforma ASP.NET, que é uma plataforma da Microsoft para o desenvolvimento de aplicações Web, onde o seu principal ambiente de desenvolvimento é o Visual Studio. Neste foram criadas todas as páginas HTML e ASPX. Para a reprodução do conteúdo multimédia, que é o principal objectivo para o desenvolvimento do site, foi inserido, já em fase final do projecto e com a orientação do Professor Alexandre Pereira, a utilização de HTML5, também devido a algumas dificuldades em encontrar um player open source para fazer tal reprodução. Foi utilizado inicialmente um plugin do JQuery para fazer a reprodução de mp4 e mp3 (ficheiros de vídeo e áudio suportados pelo sistema Android), onde encontrei algumas dificuldades em obter o resultado final esperado e daí a migração para o HTML5, que apesar da versão final ser esperada apenas para 2012 e, também não me oferecendo o resultado pretendido, será certamente o futuro da Web, ficando assim um início da sua implementação.

2.2. Android

A aplicação Android foi desenvolvida em linguagem Java utilizando o Eclipse como o IDE e um plugin ADT para fazer a instalação da componente Android. O Android tem vindo a crescer ao longo dos últimos anos devido à sua flexibilidade, ao seu livre desenvolvimento sem qualquer custo e à sua fácil migração para os fabricantes. Inicialmente desenvolvido pela Google, é uma plataforma móvel que é executada sobre um Kernel Linux [b] e onde cada aplicação é executada numa instância de uma máquina virtual Dalvik (Dalvik VM) [c] que por sua vez reside num processo gerido pelo Kernel Linux [1].

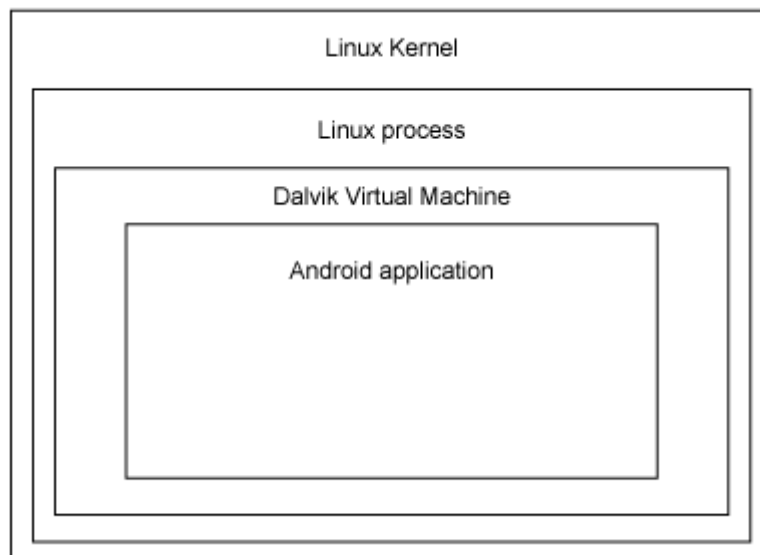


Ilustração 1: Camadas do software Android (simplificado)

3. Desenho lógico do sistema

3.1. Arquitectura e descrição do sistema

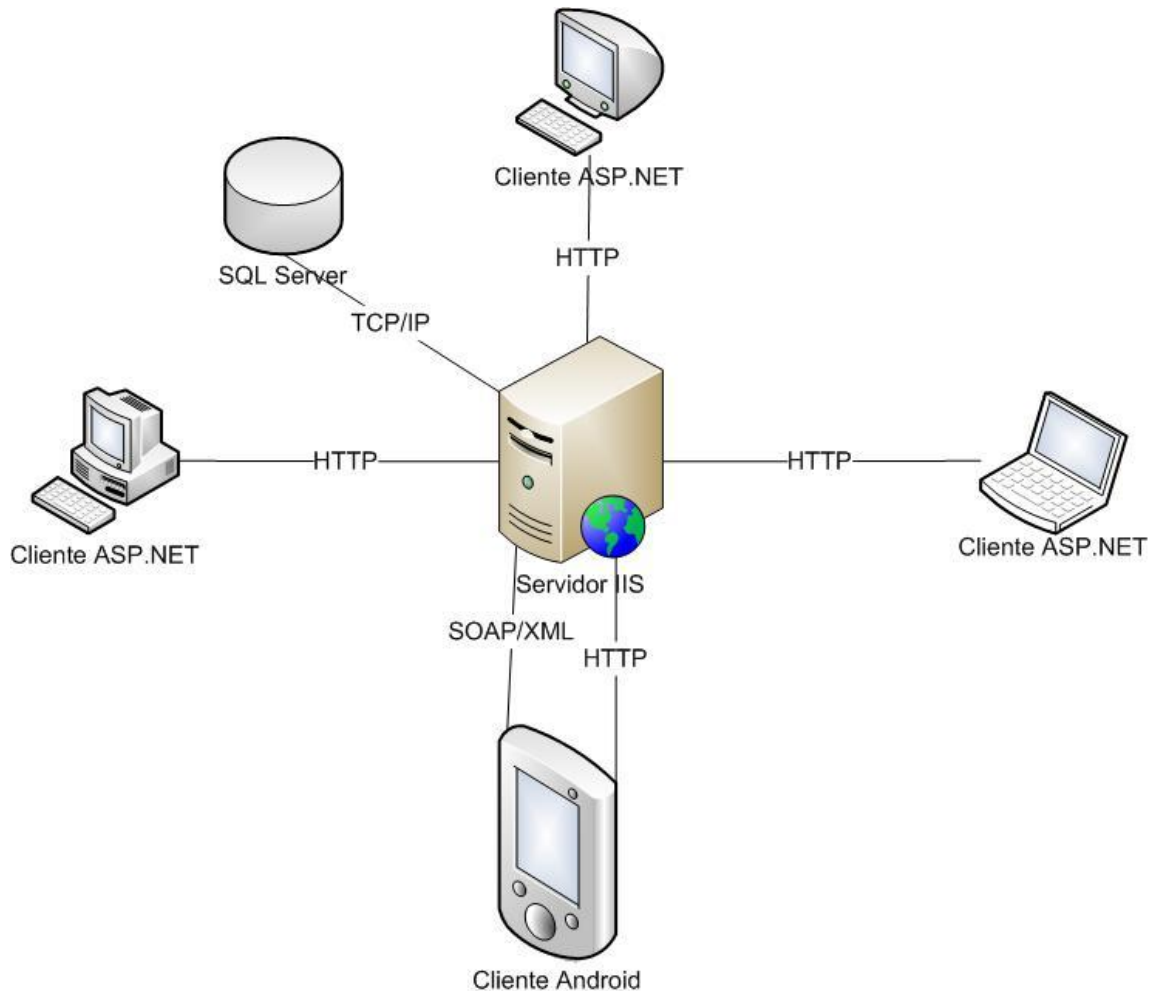


Ilustração 2: Arquitectura do Sistema

O “centro” do projecto está no Servidor IIS, que é um local do computador onde se colocam ficheiros, podendo assim simular um servidor no meu computador. O servidor contém assim todas as páginas HTML e ASPX para os clientes ASP.NET, contendo assim todo o conteúdo multimédia dos utilizadores, onde este está organizado com uma pasta para cada utilizador. Possui também o Webservice em C# para fazer a interoperabilidade com o sistema Android. O servidor tem uma ligação com o SQL Server que é a base de dados de todo o sistema.

Os Clientes ASP.NET fazem pedidos HTTP ao Servidor onde este faz a compilação da página e traduz o código para HTML para este ser interpretado pelo browser do Cliente. Nos Clientes Android existem dois tipos de pedidos ao Servidor:

- Pedidos SOAP/XML [d]

É utilizado para fazer o envio e recepção de dados escritos, como também para fazer o envio dos ficheiros multimédia por parte do Cliente.

- Pedidos HTTP

Servem para fazer os pedidos dos ficheiros multimédia por parte do Cliente e estes lhe serem apenas mostrados no seu telemóvel sem serem armazenados fisicamente.

3.2. Use Case

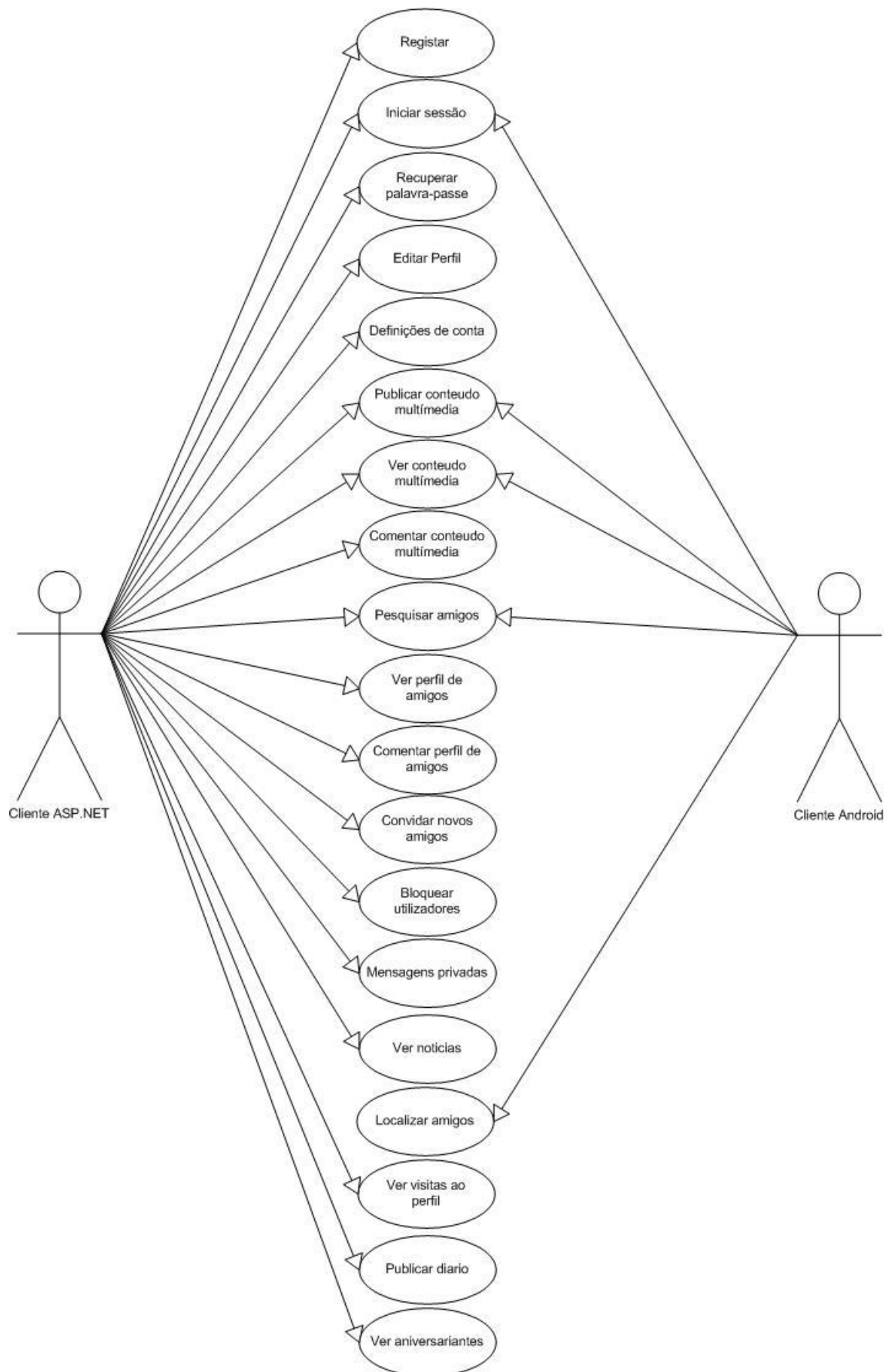


Ilustração 3: Use Case

3.3. Lista de Requisitos

A proposta inicial incidia na realização de um simples site onde fosse permitido o registo de utilizadores, fazer a ligação entre amigos, inserir fotos e poder comentá-las. Para a aplicação Android seria desenvolvido uma aplicação que permitia enviar fotografias que estão na memória do telemóvel para o site através da comunicação com um Webservice. Depois de desenvolvida esta parte inicial, foi acrescentada a partilha de ficheiros de áudio e vídeo bem como a sua reprodução, permitindo também a visualização na aplicação Android de tais ficheiros.

3.3.1. Requisitos funcionais

- ASP.NET
 - Realização de um registo para acesso à área privada, tanto no site como na aplicação Android
 - Início de sessão para área privada
 - Recuperação de palavra-chave
 - Utilização de cookies para login automático (*)
 - Utilização de SHA-1 para protecção da palavra-chave antes de a guardar na base de dados (*)
 - Convite de amizade realizado através do site ou via e-mail (*)
 - Criação de um perfil que permite integrar informações pessoais
 - Permitir que outros utilizadores vejam as últimas acções de outro utilizador (*)
 - Visualização do perfil de amigos e desconhecidos
 - Troca de mensagens privadas entre utilizadores
 - Visualização de utilizadores que visitem o perfil de outro utilizador (*)
 - Comentário tanto ao perfil de um utilizador como aos conteúdos multimédia (apenas imagens e vídeos)
 - Criação de um diário pessoal ou público (*)
 - Visualização de aniversários de amigos (*)

- Bloquear utilizadores onde ambos os utilizadores ficam sem acesso a qualquer informação um do outro (*)
 - Pesquisa de amigos através de um nome ou através de e-mail (*)
 - Área de definição de conta, onde pode:
 - Fazer a alteração de e-mail (*)
 - Fazer a alteração de palavra-chave (*)
 - Escolher quem pode ter acesso ao perfil pessoal e fazer comentários (*)
 - Escolher quem pode ter acesso ao conteúdo multimédia e fazer comentários (*)
 - Escolher música de perfil (*)
 - Permitir que diário seja público ou privado (*)
 - Permitir que os utilizadores vejam as últimas acções de outro utilizador (*)
 - Deixar que outros utilizadores vejam que um utilizador visitou o seu perfil (*)
 - Desbloquear utilizadores (*)
 - Desactivar conta (*)
 - Contactar administração (*)
 - Fazer o carregamento de ficheiros JPG, PNG, MP3 e MP4 até 20MB
 - Escolher foto de perfil visível a todos os utilizadores
 - Área administrativa onde permite:
 - Ver número de utilizadores registados e número de utilizadores que cancelaram a conta (*)
 - Ver mensagens enviadas pelos utilizadores e responder para o respectivo e-mail (*)
 - Ver lista de erros que acontecem no site (*)
 - Fazer o download da aplicação Android para instalar no telemóvel
- Android
 - Iniciar sessão para área privada
 - Ver todos os conteúdos multimédia pessoais (*)
 - Permitir ou não que outros utilizadores vejam a minha localização actual via GPS (*)

- Envio de ficheiros multimédia para o servidor
- Pesquisar amigos (*)
- Ver todo o conteúdo multimédia de um amigo caso este o permita (*)
- Ver localização geográfica de um amigo (*)
- Marcar rota da posição de um utilizador ate outro (*)

(*) Requisitos extra aos inicialmente propostos

3.3.2. Requisitos não funcionais

No documento inicial propus-me a fazer a criação de álbuns de fotografias, o qual não foi desenvolvido, pois considerei não ser necessário com os novos requisitos inseridos no site. O site deixou de ser apenas uma Rede Social onde os seus utilizadores publicavam somente imagens, passando a ser um site onde os utilizadores poderiam publicar fotos, vídeos e músicas.

Em relação ao upload de vídeos, ficou algo inutilizável com a migração para o HTML5. O HTML5 obrigou-me em certa forma a deixar de ter o IE8 como browser predefinido para testes e passar a utilizar o Google Chrome, pois este suportaria a reprodução tanto de áudio como de vídeo (na altura não migrei para o IE9 porque ainda utilizava a reprodução de vídeo através do plugin JQuery [2] o que não funcionava no Google Chrome devido à falta de codecs). Acontece que o HTML5 requer codecs específicos para a reprodução dos ficheiros e em relação à reprodução de vídeo exige que este possua o codec de vídeo H.264 e codec de áudio ACC [3]. Apesar do problema relativo aos vídeos inseridos não serem produzidos no HTML5, segundo o Docente Alexandre Pereira, este é um contra-tempo que me ultrapassa, sendo portanto ignorado. Como solução para este problema, conforme orientação do Docente Alexandre Pereira, deveria ter uns vídeos previamente convertidos com os codecs necessários, para depois os utilizar na demonstração do projecto.

Como tentativa de conversão de vídeo, procedi à utilização do ffmpeg [4] onde foram realizados testes na conversão de um vídeo mp4, sem os codecs necessários para ser reproduzido no HTML5, para um vídeo mp4 com os codecs inseridos. Deste modo o ffmpeg começava a converter o vídeo mas ultrapassava os limites do mesmo e só parava a conversão manualmente. Não compreendendo a razão para o que acabei de descrever

e, mesmo após pesquisar [5], não descobri a solução para este problema. A utilização do ffmpeg ficou então a ser utilizada unicamente para a criação de Thumbnail dos vídeos, para os quais se faz o upload, tanto no site como na aplicação Android [6].

3.3.3. Requisitos extra (relevantes)

Como requisitos extra, executei primeiramente a reprodução dos ficheiros multimédia no telemóvel, onde, utilizando como base a componente VideoView que vem com o plugin ADT do Android, construí um player de vídeo e de música. Inicialmente, era realizada a leitura dos bytes do ficheiro original para um ficheiro temporário, armazenado na memória interna do telemóvel, só sendo reproduzido depois da totalidade dos bytes serem escritos para o ficheiro temporário. No entanto, este método tinha a desvantagem de ocupar memória no telemóvel, o que tornava impossível a visualização do ficheiro caso o utilizador não tivesse memória disponível, tendo ainda a desvantagem de não permitir o Streaming [e] do ficheiro.

O segundo requisito extra, corresponde à localização de amigos através de GPS. Esta funcionalidade consiste na utilização dos mapas da Google e do componente MapView do sistema Android, onde é possível marcar a posição geográfica de um utilizador. Para poder usufruir inteiramente desta funcionalidade, permito ainda que, caso o user deseje, seja marcada a rota de um utilizador ao outro. Para a realização desta funcionalidade utilizo um pedido HTTP ao site <http://maps.google.com>, onde completando o URL com as coordenadas do utilizador A até ao utilizador B, este devolve-me um ficheiro KML que contém toda a descrição que é utilizada pelo Google Maps para fazer a marcação de rotas. Nesse ficheiro retiro toda a descrição da rota, todas as coordenadas para fazer a marcação da mesma, e após de possuir todos os dados referidos, desenho a rota sobre o MapView [6]. Adicionalmente ao descrito, poderia ainda fazer a simulação GPS de uma posição à outra, onde iria mostrar a constante actualização da posição do utilizador, fazendo aparecer descrições no ecrã e exibindo para onde o utilizador deveria ir. Mas, partilhando da opinião do Docente Orientador, já era algo que se distanciava do objectivo do tema proposto.

4. Diagramas

De seguida, serão apresentados alguns diagramas que representam alguns aspectos importantes do projecto, mostrando também as diferenças de acções entre as duas tecnologias, nomeadamente o Início de Sessão feito no lado do cliente ASP.NET e no lado do cliente Android, tal como o carregamento de um ficheiro no lado do cliente ASP.NET e no lado do cliente Android.

4.1. Diagrama de actividades | Login ASP.NET

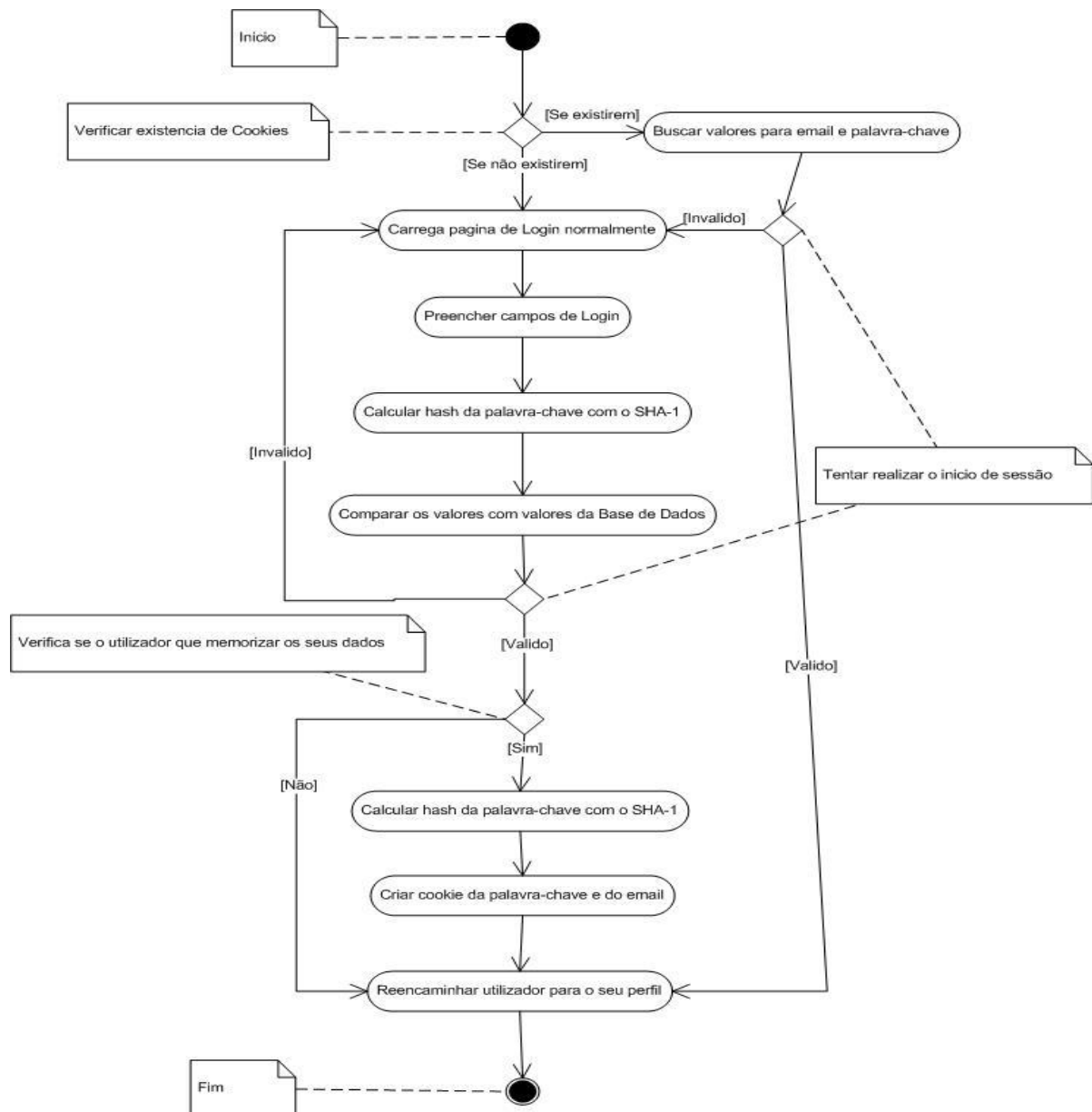


Ilustração 4: Diagrama de actividades | Login ASP.NET

4.2. BPMN | Login Android

Para a realização de login no lado do Cliente Android, é utilizado SOAP para troca de variáveis entre as duas tecnologias [7]. É utilizada a base de dados SQLite para guardar os dados no telemóvel caso o utilizador pretenda que o login seja feito automaticamente e não precise de estar sempre a inserir os dados para iniciar a sessão. De seguida demonstro como é efectuado o processo de login entre o telemóvel Android e o Servidor Web.

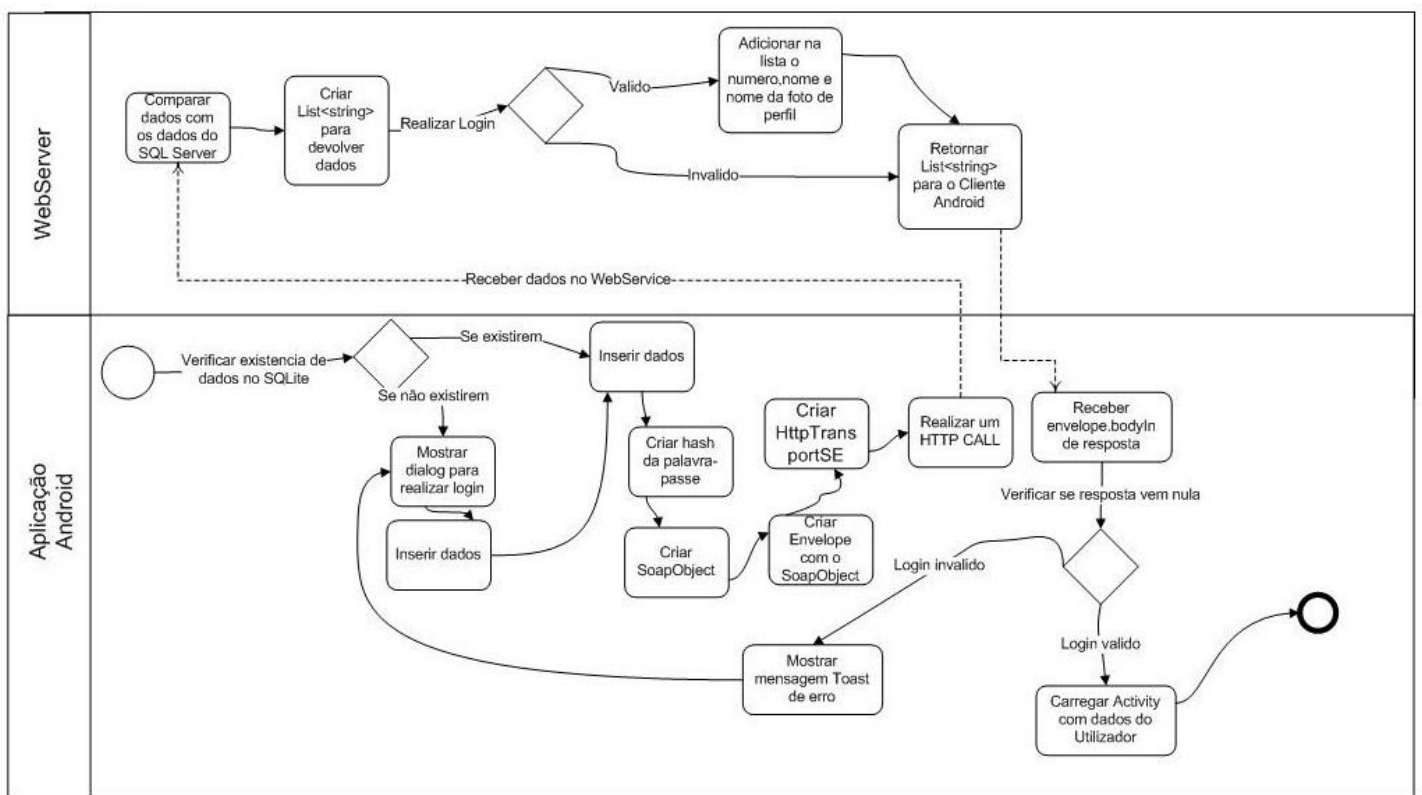


Ilustração 5: BPMN | Login Android

4.3. Diagrama de sequência | Publicar imagem ASP.NET

O processo de publicação de ficheiros por parte do site ASP.NET, é praticamente semelhante para os três tipos de ficheiros permitidos. Estes são carregados na página `carregaFiles.aspx`, onde é gerado um nome para o ficheiro utilizando a funcionalidade do `DateTime.Ticks` do C# [8], prevenindo assim que ficheiros com o mesmo nome inseridos pelo utilizador, venham posteriormente a criar problemas na pasta de ficheiros do utilizador no lado do servidor. São gravados depois no lado do Servidor consoante a sua extensão.

Como já foi referido, para cada utilizador é criada uma pasta com o seu respectivo número, onde dentro dessa pasta são criadas três pastas, uma para cada tipo de ficheiro, mantendo assim o conteúdo dos utilizadores mais organizado.

No entanto, existem pequenas diferenças no tratamento dos ficheiros. No caso das imagens (exemplo em baixo), a imagem deve ser redimensionada, não só para questões de design do site, mas também para evitar erros no Cliente Android quando são efectuados pedidos HTTP para visualização das imagens. No caso dos vídeos, o único tratamento é a criação de Thumbnail dos vídeos, questão já abordada anteriormente. Relativamente às músicas é somente necessário que se grave o nome original da música na base de dados e dar-lhe o nome de gravação gerada pelo `DateTime.Ticks`.

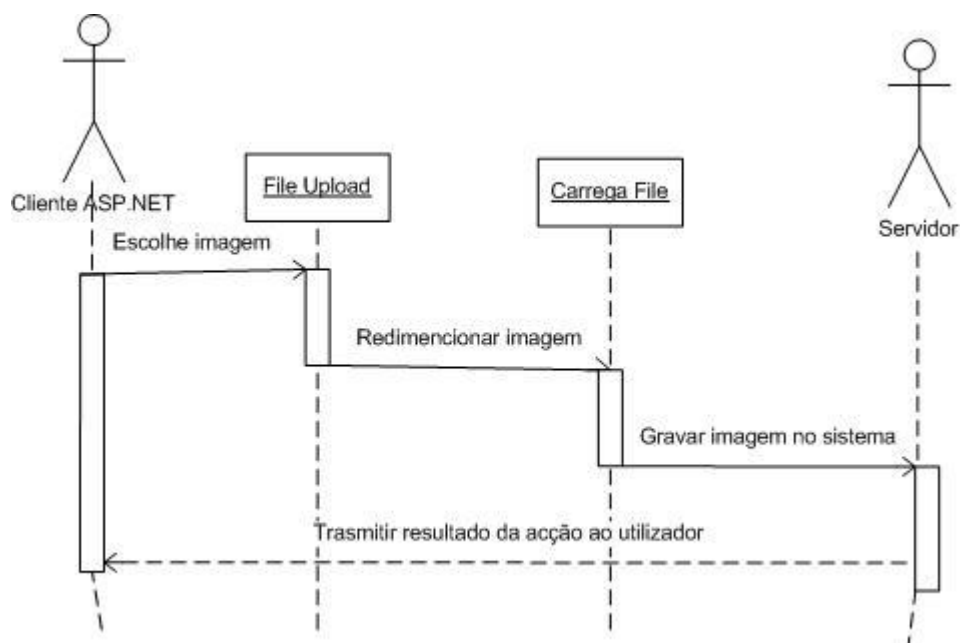


Ilustração 6: Diagrama de sequência | Publicar imagem ASP.NET

4.4. BPMN | Publicar imagem Android

O processo de envio de ficheiros do Cliente Android para o Servidor é um processo mais extenso (mas o tratamento dos ficheiros é o mesmo que no ponto anterior). Este foi também um dos problemas mais relevantes com que me deparei no início do estudo da plataforma Android, também devido à falta de conhecimento na utilização de WebServices. Inicialmente, no lado do cliente Android, era lido um ficheiro para um array de bytes, e enviado para o Webservice. Aí, eram retirados os bytes lidos, tentando assim escrever o ficheiro. Porém, no lado do servidor não conseguia ler os bytes enviados pelo Android. Após pesquisar, descobri que era necessário fazer um encode dos bytes do ficheiro para Base64, onde posteriormente, no lado do servidor, seria efectuado o seu decode, permitindo assim que o ficheiro fosse escrito correctamente. Mais tarde, ao fazer os mesmos passos para o envio de músicas e vídeos, deparei-me com outro problema, o qual consistia no facto de a aplicação Android não fazer o encode dos ficheiros. Este problema foi bastante mais simples de solucionar, pois bastou perceber que quando se faz o encode para Base64, este só faz o encode até um determinado tamanho de bytes. Logo para fazer a transferência de ficheiros do Android para o servidor, vou enviando 500Kb (máximo suportado para fazer o encode para Base64) do ficheiro de cada vez para o servidor, onde este os vai juntando ao respectivo ficheiro e assim o construindo [9].

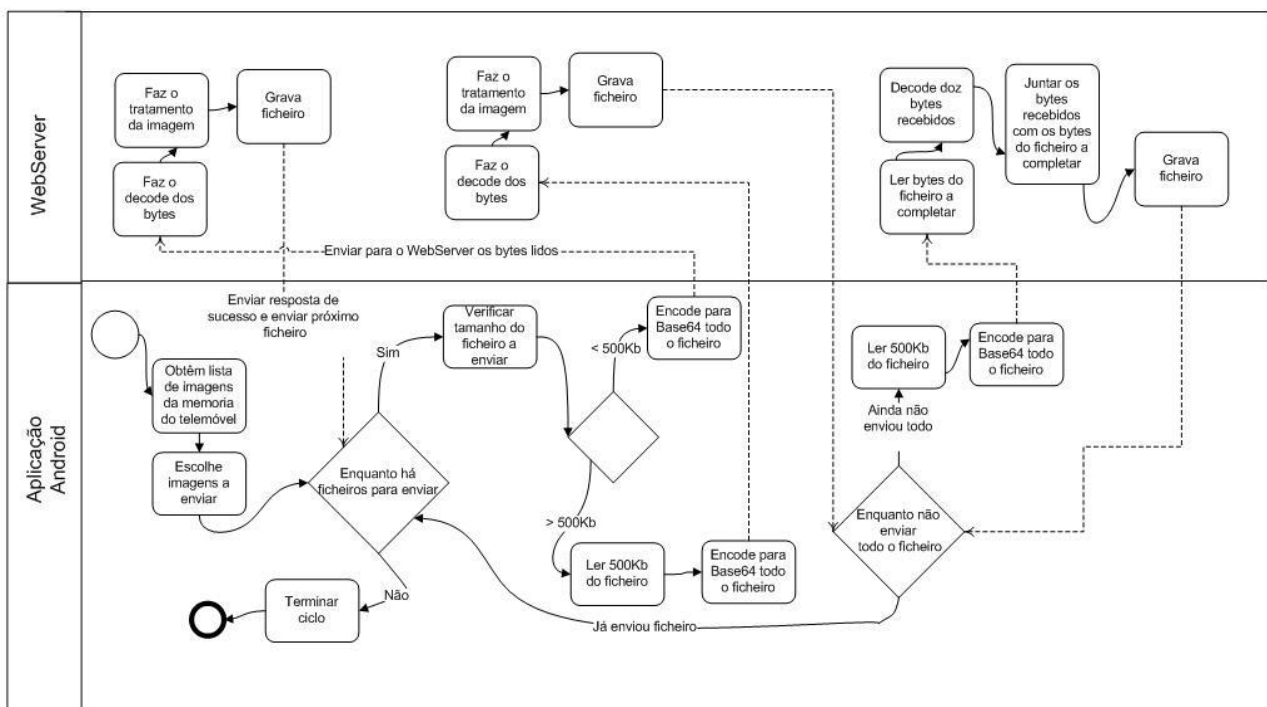


Ilustração 7: Diagrama de sequência | Publicar imagem ASP.NET

4.5. Modelo de dados

Devido ao modelo de dados ser bastante extenso, apresento-o de seguida dividido por categorias.

4.5.1. Dados pessoais

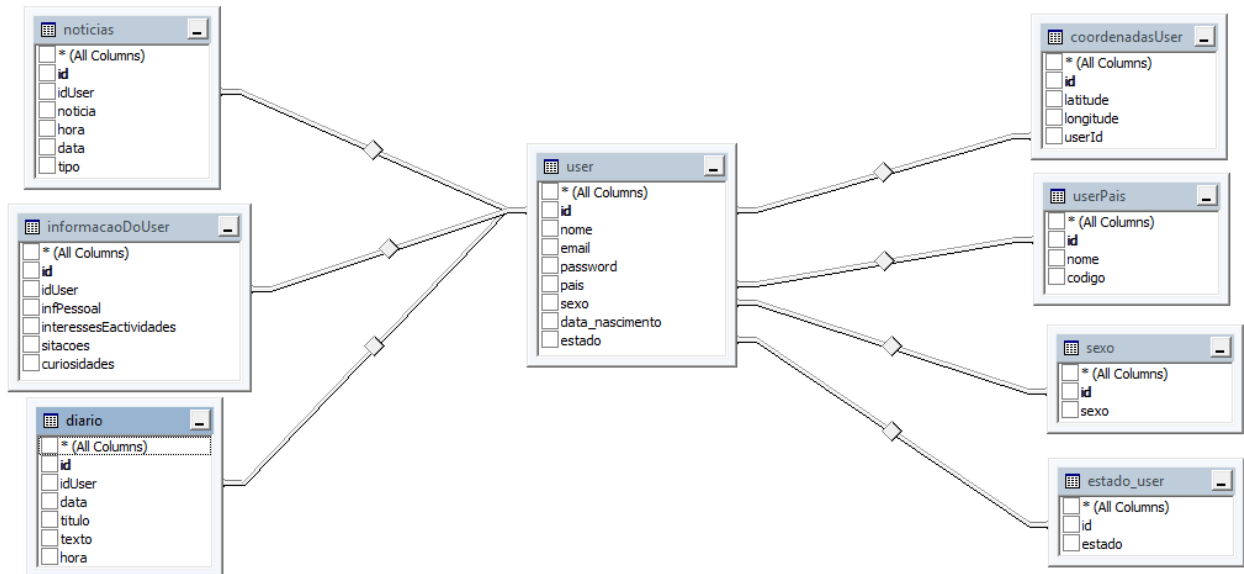


Ilustração 8: Modelo de dados – Dados pessoais

4.5.2. Dados multimédia

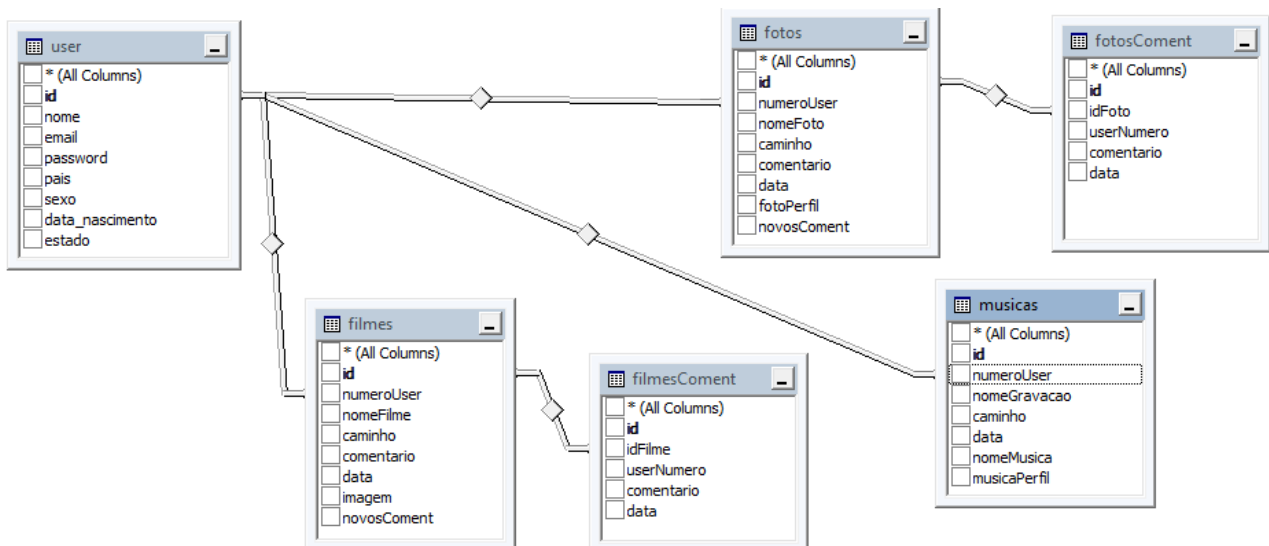


Ilustração 9: Modelo de dados – Dados multimédia

4.5.3. Informações para o utilizador

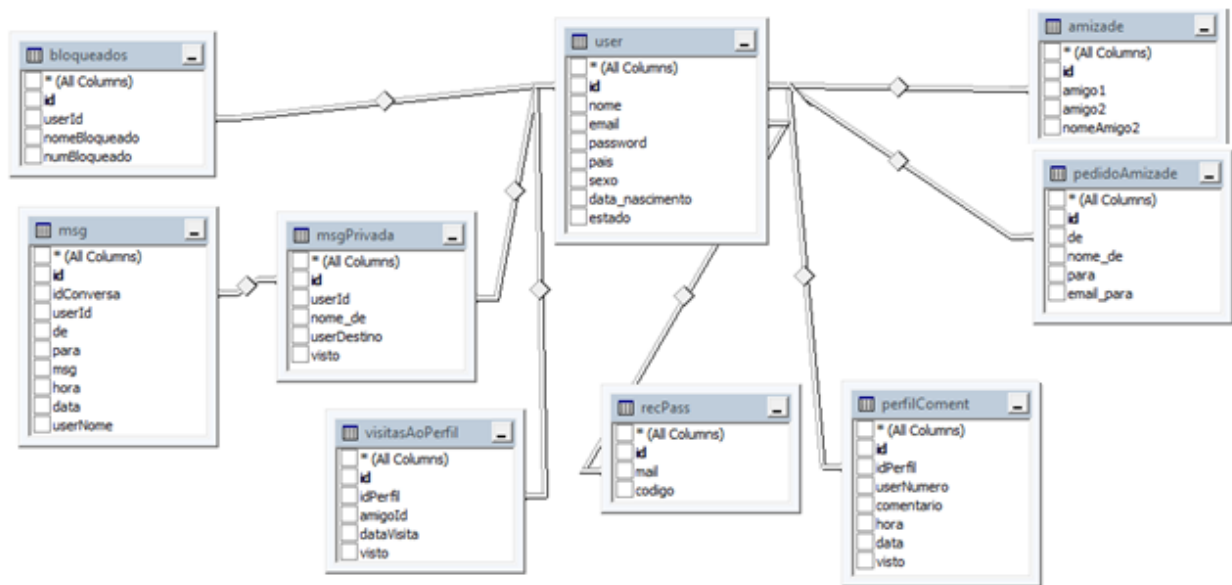


Ilustração 10: Modelo de dados – Informações para o utilizador

4.5.4. Definições de conta

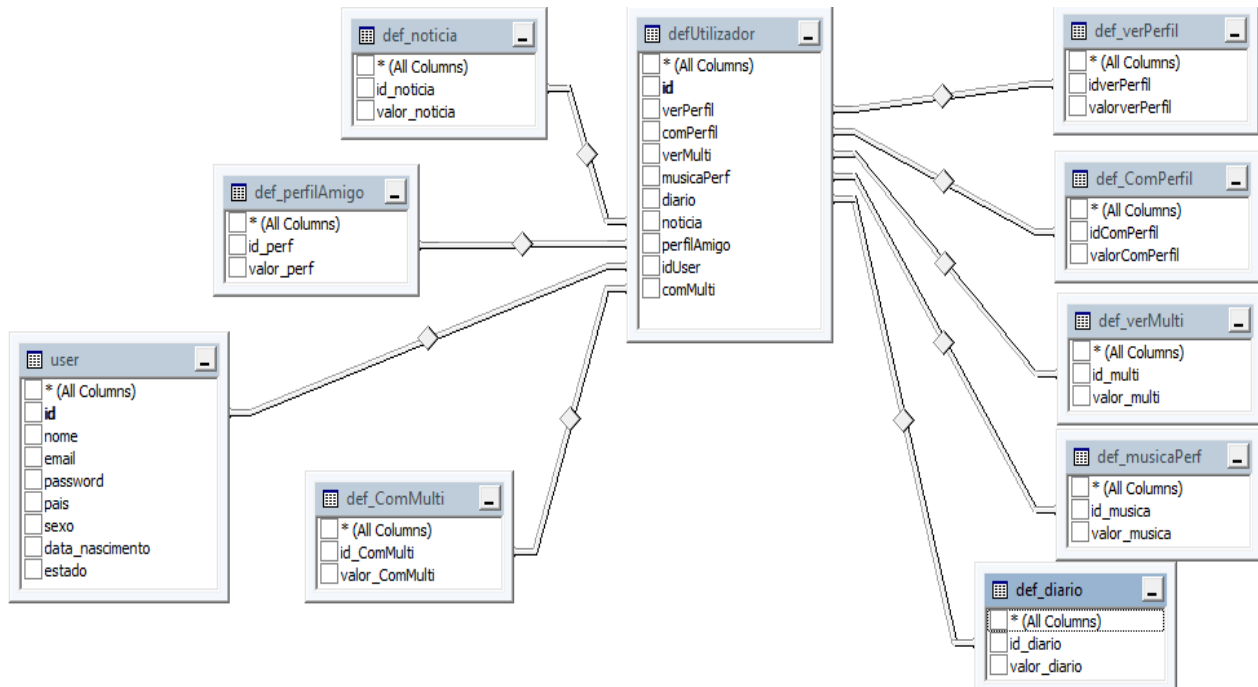


Ilustração 11: Modelo de dados – Definições de conta

4.5.5. Administração

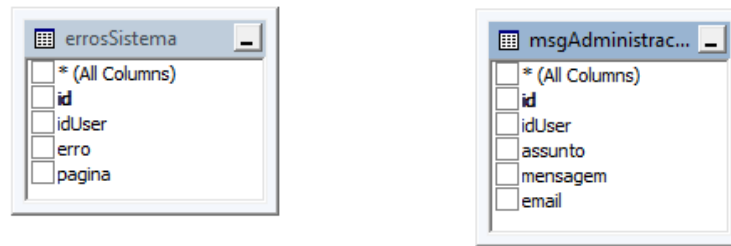


Ilustração 12: Modelo de dados – Administração

4.5.6. Restantes tabelas

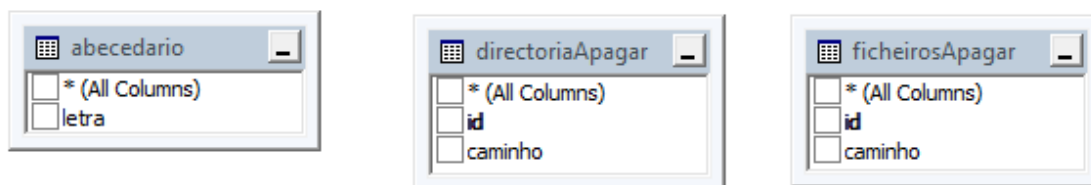


Ilustração 13: Modelo de dados – Restantes tabelas

5. Android

A aprendizagem da programação em plataforma Android foi o principal desafio deste trabalho. Apesar do início atribulado, onde iniciei o estudo com a tentativa de fazer um pequeno bloco de notas, começando também a manusear na base de dados interna do Android (SQLite). Depois das primeiras dificuldades serem ultrapassadas, comecei a tentar fazer um pequeno programa para transferência de ficheiros, fazendo um pequeno servidor em Java utilizando Threads e Sockets. Daqui surgiu então o tema final de curso.

Tendo sido Java a primeira linguagem que adquiri e o facto de ter sido desenvolvida em diversas unidades curriculares ao longo da licenciatura, tornou o processo de desenvolvimento Android mais facilitado, apesar do seu desenvolvimento ser distinto, aumentou o interesse por este devido ao acrescento da sua componente gráfica e de todos os seus componentes extra incluídos.

5.1. Activity

5.1.1. O que é?

Uma Activity, para além da extensão de uma classe, é uma tela de interface gráfica do utilizador. Uma aplicação Android é normalmente composta por uma ou várias Activities, onde cada uma tem o seu próprio objectivo. No caso deste projecto, a aplicação é composta por 12 Activities interligadas entre si. A Activity principal é a `principal.class` que é a Activity que contém o início de sessão para o utilizador e compreende ainda o menu principal, o qual permite ao utilizador navegar entre os vários menus da aplicação.



Ilustração 14: Ecrã principal da aplicação Android

5.1.2. Como funciona?

Cada Activity tem o chamado *Ciclo de Vida* que é representado em baixo:

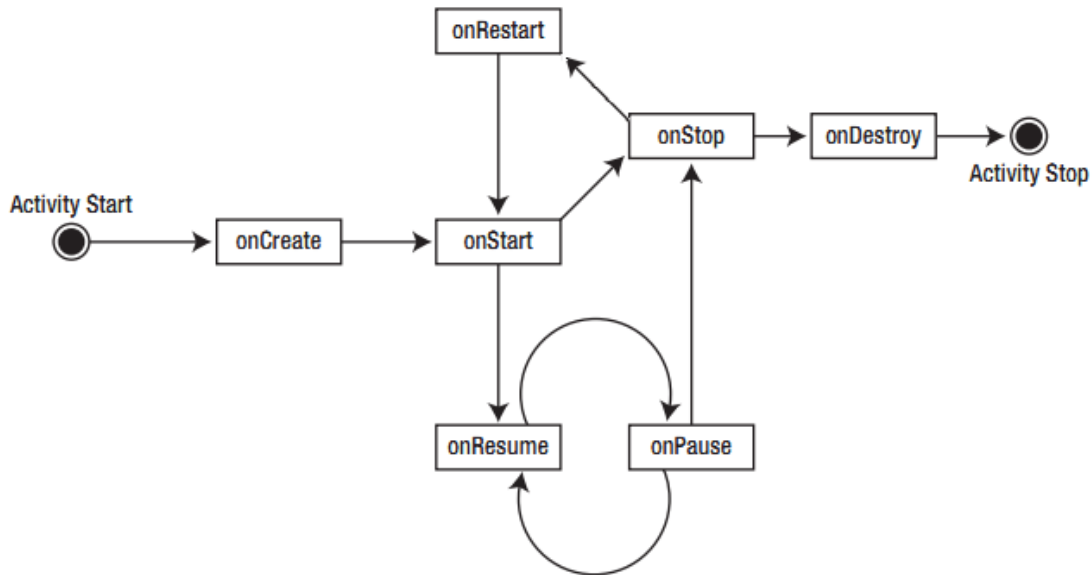


Ilustração 15: Ciclo de vida de uma Activity

Tomando como exemplo a activity principal.class, esta começa por fazer todo o código dentro do método `onCreate()`. Neste caso não tem qualquer definição para o método `onStart()` por isso passa logo para o método `onResume()`. Chegando a este ponto a Activity estará a mostrar o que está representado na Ilustração14 para o utilizador Ruben Costa. O Android tem uma programação orientada a objectos, onde muitos dos seus objectos têm já métodos pré-definidos. Por exemplo, no caso de se utilizar um Button, como qualquer um da Ilustração14, cada um deles tem em si o método `setOnClickListener` que fica activo à espera que haja algum evento sobre ele. Cada botão deve também ser definido no ficheiro XML da correspondente Activity, onde aí é definido o seu id para depois poder ser utilizado na Activity.

Código:

```
//Definição do botão no ficheiro XML da Activity principal.class

<Button
    android:id="@+id/btn_imagens"
    android:text="Ver imagens"
    android:background="@drawable/button_black"
    android:drawableTop="@drawable/foto2"
    android:layout_width="110px"
    android:layout_height="wrap_content"
    android:textColor="#ffffff"
    android:layout_weight="1">
</Button>

//Declaração do botão na respectiva Activity

Button minhasFotos = (Button) findViewById(R.id.btn_imagens);
minhasFotos.setOnClickListener(this);

// Declaração do método onClick que é o método que trata do evento do botão
que for pressionado

@Override
public void onClick(View v)
{
    Bundle extras = new Bundle();
    Intent intent;

    switch (v.getId())
    {
        case R.id.btn_imagens:
            numFiles = send.numeroDeFiles(numero);
            if(numFiles.get(0).equals("1"))
            {
                intent = new Intent(getApplicationContext(), playFoto.class);
                extras.putInt("numAmigo", numero);
                intent.putExtras(extras);
                startActivity(intent);
            }
            else Toast.makeText(getApplicationContext(), "Não tens imagens para
ver.", Toast.LENGTH_SHORT).show();

            break;
        }
    }
}
```

Um utilizador ao pressionar o botão “Ver imagens” irá dar origem ao evento onClick desse botão, que neste caso vai verificar no lado do servidor se o user tem imagens publicadas, e caso as tenha passa para a Activity playFoto.class. Como não

pretendo fazer nada à Activity principal.class quando estou prestes a abrir uma nova Activity, esta fica simplesmente no estado `onPause()` e é aberta a nova Activity `playFoto.class`, tendo esta o seu próprio ciclo de vida. Ao sair da Activity `playFoto.class` e voltar para a `principal.class`, esta não vai voltar a realizar o método `onCreate()`, realizando apenas o método `onResume()` caso este esteja definido.

5.2. ListActivity - principais dificuldades

Uma das dificuldades encontradas, no início do estudo da plataforma Android, foi perceber o funcionamento de uma Lista. Uma classe em Android pode ser alargada para outras funcionalidades, nomeadamente a `ListActivity`. Uma `ListActivity`, tal como uma Activity, é uma tela de interface gráfica do utilizador com a componente de essa tela ser uma lista de objectos. Depois de alguma pesquisa, percebi que essa classe vai ter uma outra classe que estende um adaptador para essa lista. O adaptador, para uma `ListActivity` com diferentes objectos, mais eficiente para o Android é o `BaseAdapter` [10]. Este adaptador é uma classe predefinida que contém em si cinco métodos que têm de estar obrigatoriamente definidos.

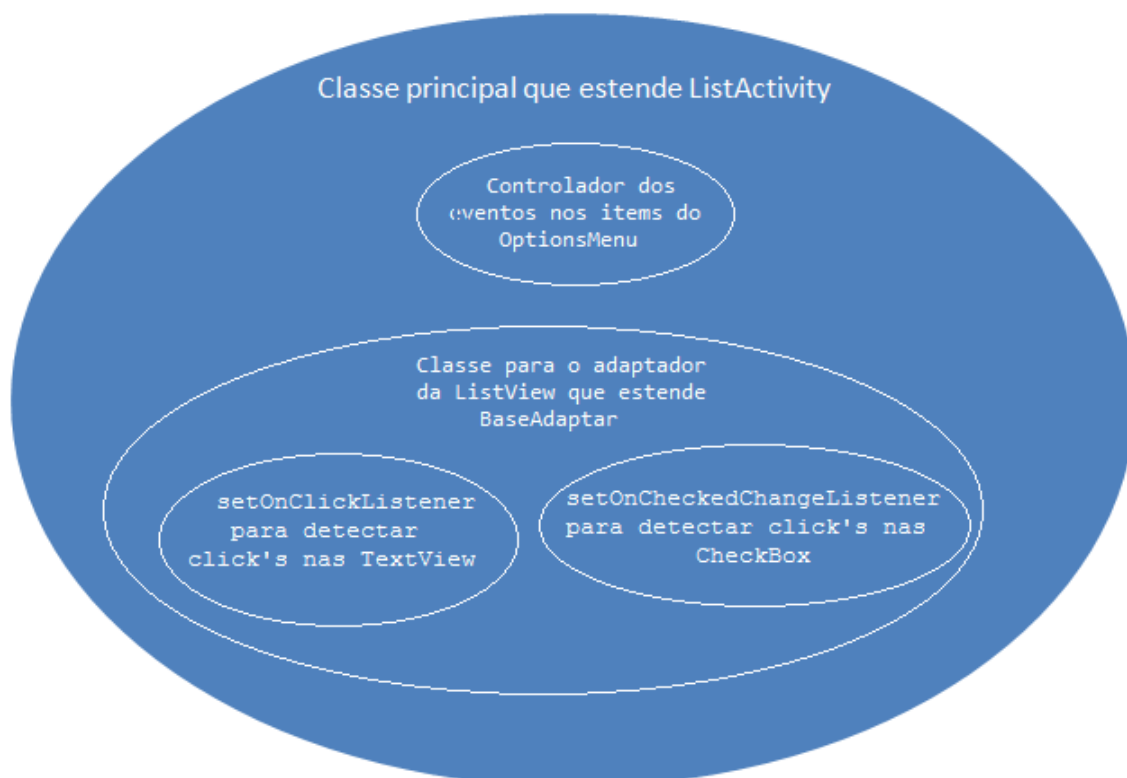


Ilustração 16: Exemplo da estrutura da classe `enviaFotos.java`

Código:

```

public class adapterFotos extends BaseAdapter
{
    public adapterFotos(Context context, Cursor c)
    {
        //retorna numero de elementos da lista
        public int getCount()
        {
        }

        //retorna o objecto de uma determinada posicao
        public Object getItem(int position)
        {
        }

        //retorna o identificador de uma determinada posicao
        public long getItemId(int position)
        {
        }

        //retorna o objecto, que herda de View, a ser mostrado
        public View getView(final int position, View convertView,
        ViewGroup parent)
        {
            //ver se é uma posicao nova ou se ja foi criada
            if(!gravaPos.contains(position))
            {
                //add posicao para n voltar a entrar aqui nessa pos
                gravaPos.add(position);

                //buscar uma nova vista
                convertView = inflater.inflate(R.layout.enviafotosadapter,
                null);

                //criar novos objectos para a nova vista
                holder = new ViewHolder();

                //criar objectos novos para serem postos na nova linha da
                lista
                (...)

                convertView.setTag(holder); //"colar" está nova vista à
                posicao
            }
            //caso seja uma posição ja criada ir buscar essa vista
            else holder = (ViewHolder) convertView.getTag();

            // construir vista que vai aparecendo
            (...)

            return convertView;
        }
    }
}

```

O método mais extenso é o `getView()`, o qual implica um maior manuseio por parte de quem planeia uma lista deste tipo. Aquilo que este método faz, é criar para cada linha que aparece no ecrã uma nova `View`. Se for uma lista com vários elementos que ultrapassem as dimensões do ecrã, à medida que o utilizador vai descendo, novas `View` vão sendo criadas, ou seja, ele só cria uma `View` da linha quando essa linha aparece no ecrã (sendo lógico que para as linhas que nunca aparecerem no ecrã nunca é criada uma `View` delas). Se um utilizador após descer na lista, pretender voltar para cima e ver as linhas já criadas, o que o `BaseAdapter` faz é a reciclagem dessa linha, ou seja, ele vai resgatar a `View` já criada para não a criar novamente, poupando assim recursos ao telemóvel e não estando sempre a criar novas `View`.

No início do desenvolvimento deste tipo de listas, existia o problema de ao andar para baixo na lista, os últimos elementos serem iguais aos primeiros e, ao voltar a subir, os primeiros passavam a ser os últimos. Depois de perceber o funcionamento do `BaseAdapter` e a sua correcta implementação, encontrei um novo problema. Não era feita a reciclagem correcta da lista, existindo elementos que eram sempre criados quando apareciam no ecrã. Para corrigir o problema fiz uma pequena alteração ao método `getView()`, onde tendo um `ArrayList` criado previamente, vou adicionando nele as posições criadas e evitando que uma posição seja criada mais que uma vez, fazendo assim a reciclagem dos elementos correctamente.

Apesar deste adaptador ser bastante eficiente, no método `getView()`, depois de ser feita



Ilustração 17: Ecrã da classe `enviaFotos.java`

ou não a reciclagem de uma posição, para cada objecto dessa View tem que haver a indicação do valor que vai obter. Utilizando como exemplo o projecto desenvolvido, no qual utilizei uma lista, onde cada linha contém uma CheckBox, uma ImageView e uma TextView (Ilustração 17). Focando apenas as CheckBox, é necessário criar uma funcionalidade extra para cada vez que uma determinada CheckBox aparece no ecrã, para que saiba se nessa posição essa CheckBox deve estar Checked ou não. Por esta razão, não basta fazer a reciclagem da lista, sendo necessário haver um componente extra que resolva o problema mencionado, caso contrário o valor dessa CheckBox será perdido quando um utilizador navega pelos varios elementos da lista. Neste caso em particular, assim como em outros, a resolução do problema consiste na construção de um array, que para o caso das CheckBox optei por fazer um array de booleanos, onde as posições correspondiam às mesmas posições que as de cada CheckBox da lista, e assim, sempre que um elemento de uma posição aparece, vejo na posição do array se essa CheckBox é para ser Checked ou não. Para guardar os valores das CheckBox, utilizo a propriedade `setOnCheckedChangeListener()` de cada CheckBox, vou alterando as respectivas posições com True e False.

Considerações finais e perspectivas

Tendo em consideração apenas os três anos de experiência em programação, posso afirmar que fiquei bastante satisfeito com o resultado final do presente Projecto. Apesar de não o considerar um projecto com um nível de dificuldade demasiado elevado, considero-o no entanto, um projecto bastante extenso, ambicioso e que só foi possível ser concretizado devido a toda a pesquisa e investigação realizada, de forma a colmatar défices de conhecimento e obstáculos que foram surgindo ao longo de todo o seu percurso. Devido à sua extensão e grau de complexidade, foi necessária dedicação e empenho, de forma a conseguir conciliar a sua elaboração e sucesso com as restantes unidades curriculares. Considero assim que todo este processo foi bastante produtivo, contribuindo assim para o meu desenvolvimento pessoal e profissional.

Apesar de alguns contra-tempos, que como já foi referido anteriormente, me ultrapassam, encaro que o Projecto apresentado se encontra concluído, com um produto final de qualidade e rigor, sendo a meu ver o ponto de partida ideal para a continuação do meu aperfeiçoamento e desenvolvimento em software, um ponto de partida para continuar a investir na aprendizagem de novas linguagens e plataformas de programação.

Dos conhecimentos adquiridos, dou principal destaque à plataforma Android, devido ao grau de autonomia que me exigiu, tanto para a pesquisa e investigação científica, como para colocar à prova o meu nível de perseverança e dedicação a uma área que realmente completa a nível profissional. Relativamente ao desenvolvimento ASP.NET, tinha como pretensão iniciar mais um estudo, o qual lamentavelmente não ter realizado ao longo do desenvolvimento deste projecto, devido à falta de tempo. Estudo esse que iria incidir na inserção de mais funcionalidades de JavaScript, neste e em projectos futuros, de forma a tornar o site menos pesado para o servidor.

Como positivo foi também o conhecimento adquirido a partir de um outro componente ao JavaScript, o JQuery, que apesar de não o ter utilizado muito, consegui compreender as suas funcionalidades, pois foi através deste componente que efectuei a reprodução de vídeo e áudio.

Como objectivo a nível profissional, espero conseguir seguir o caminho de desenvolvimento de aplicações com interface gráfica do utilizador, onde garantidamente o estudo e desenvolvimento de aplicações Android será para perdurar, mesmo que este seja apenas um desenvolvimento fora da área profissional.

Para concluir, gostaria ainda de agradecer a orientação do Docente Alexandre Pereira, que se mostrou sempre disponível e presente, auxiliando na resolução das dificuldades apresentadas.

Referências

- [1] - <http://www.ibm.com/developerworks/br/library/os-android-devel>
- [2] - <http://jquery.malsup.com/media/video.html>
- [3] - http://www.w3schools.com/html5/html5_video.asp
- [4] - <http://www.ffmpeg.org>
- [5] - <http://johndyer.name/post/ffmpeg-Settings-for-HTML5-h264-mp4-theora-ogg-vp8-webm.aspx>
- [6] - <http://ramcrishna.blogspot.com/2008/09/playing-videos-like-youtube-and.html>
- [6] - http://mapki.com/wiki/Google_Map_Parameters
- [7] - <http://stackoverflow.com/questions/1052300/how-to-call-a-net-webservice-from-android-using-ksoap2>
- [8] - <http://msdn.microsoft.com/en-us/library/system.datetime.ticks.aspx>
- [9] - <http://androidcodemonkey.blogspot.com/2010/03/how-to-base64-encode-decode-android.html>
- [10] - <http://developer.android.com/resources/samples/ApiDemos/src/com/example/android/apis/view/List14.html>

Glossário

[a] **Sistemas distribuídos** – é uma referência à computação paralela e descentralizada, realizada por dois ou mais computadores conectados através de uma rede, cujo objectivo é concluir uma tarefa em comum.

[b] **Kernel Linux** – Controlador de todo o hardware do telemóvel. Pode ser visto como uma interface entre os programar e todo o hardware. Cabe a este, a tarefa de permitir que todos os programas sejam executados.

[c] **Máquina Virtual Dalvik** – Máquina Virtual desenvolvida pela Google lançada juntamente com o SDK do Android. Criada para sistemas operacionais UNIX com o intuito de melhorar a gestão do processo e isolamento de threads.

[d] **SOAP/XML** – SOAP é um protocolo para troca de informações estruturadas em plataformas distribuídas. Baseia-se em linguagem XML para o formato das mensagens

[e] **Streaming** – consiste na reprodução do ficheiro no lado do cliente, antes da totalidade deste estar carregada.

Netografia

ASP.NET

- <http://www.mondor.org/captcha.aspx>
 - <http://blog.bluesam.com/2009/07/triple-des-encryption-implementation.html>
 - <http://www.macoratti.net/aspnct12.htm>
 - <http://weblogs.asp.net/jgalloway/archive/2008/01/08/large-file-uploads-in-asp-net.aspx>
 - <http://ramcrishna.blogspot.com/2008/09/playing-videos-like-youtube-and.html>
 - <http://jquery.malsup.com/media/video.html>
 - <http://forums.asp.net/t/1251514.aspx/1?MaxLength+ignored+in+multiline+textbox>
 - <http://snippets.dzone.com/posts/show/216>
 - <http://www.codeproject.com/KB/ajax/ImageAutoComplete.aspx>
 - <http://support.microsoft.com/kb/318425>
 - <http://forums.asp.net/t/911510.aspx/1>
 - <http://weblogs.asp.net/markmcdonnell/archive/2008/03/09/resize-image-before-uploading-to-server.aspx>
 - <http://stackoverflow.com/questions/4819794/sha1-c-equivalent-of-this-java>
 - <http://www.maujor.com/tutorial/boxshadow.php>
 - http://www.dynamicdrive.com/style/csslibrary/item/corp_blue_css_menu/
 - <http://www.mxstudio.com.br/Conteudos/Dreamweaver/Cores.htm>
 - <http://www.asp.net/AJAX/AjaxControlToolkit/Samples/ModalPopup/ModalPopup.asp>
- [X](#)

Android

- <http://greppcode.com/file/repository.greppcode.com/java/root/jdk/openjdk/6-b14/sun/misc/CharacterEncoder.java?av=f>
- <http://androidsamples.blogspot.com/2009/06/displaying-list-of-video-files-stored.html>
- <http://www.devx.com/wireless/Article/39810/1954>
- <http://moazzam-khan.com/blog/?p=490>
- <http://mobiforge.com/developing/story/using-google-maps-android>
- <http://androidsamples.blogspot.com/2009/06/how-to-display-thumbnails-of-images.html>
- <http://www.helloandroid.com/tutorials/how-display-custom-dialog-your-android-application>
- <http://www.java-tips.org/java-se-tips/java.io/reading-a-file-into-a-byte-array.html>
- <http://developer.android.com/resources/samples/ApiDemos/src/com/example/android/ids/apis/view/List14.html>
- <http://stackoverflow.com/questions/2432951/how-to-display-a-two-column-listview-in-android>
- <http://www.androidpeople.com/android-viewflipper-example>

- <http://ruibm.com/?p=184>
- http://wiki.openstreetmap.org/wiki/YOURS#Routing_API
- <http://www.dcfusion.com/post.cfm/apk-mime-type>
- http://mapki.com/wiki/Google_Map_Parameters
- <http://developer.android.com/guide/topics/location/obtaining-user-location.html>
- <http://developer.android.com/resources/tutorials/views/index.html>
- <http://www.felipesilveira.com.br/2008/08/10-motivos-para-nao-ter-um-iphone>
- <http://www.brighthub.com/mobile/google-android/articles/33240.aspx>
- <http://www.warriorpoint.com/blog/2009/05/24/android-how-to-switch-between-activities/>
- <http://developer.android.com/guide/topics/ui/notifiers/toasts.html>
- http://findicons.com/pack/2380/android_style_icons_r1
- http://issuu.com/ricardoogliari/docs/como_criar_interfaces_graficas_com_android
- <http://www.devx.com/wireless/Article/40842/0/page/2>

SQL Server

- <http://sqlfromhell.wordpress.com/2010/04/18/habilitando-o-acesso-remoto-no-sql-server-2008/>