



UNIVERSIDADE LUSÓFONA
de Humanidades e Tecnologias
Humani nihil alienum

Relatório Final de Curso do Curso de Licenciatura em Engenharia Informática

Ano Letivo 2015/2016



Aplicação Multiplataforma

Professora Coordenadora:

Professora: Inês Oliveira

Alunos:

21103065 – Luis Manuel Pacheco Figueiras

21104167 - Rúben de Oliveira Carvalho

Índice

	Pág.
Resumo	4
Abstract (em Inglês)	5
Introdução	6
Enquadramento teórico (com revisão bibliográfica)	7
➤ Tipos de Aplicações Mobile (Nativas ou Web)	7
➤ Tecnologias Nativas (Android, ObjectiveC, entre outros)	10
➤ Tecnologias Web	11
✓ HTML5	11
✓ CSS3	11
✓ Javascript	11
✓ Mobile J-Query	12
➤ Frameworks e Tecnologias Baseadas em Javascript, HTML5 e CSS	12
✓ PhoneGap	12
✓ JSON	13
➤ Frameworks e Tecnologias para Aplicações Nativas	13
✓ XAMARIN	13
➤ Frameworks e Tecnologias para Aplicações Nativas, WEB e Híbridas	13
✓ TELERIK	13
Análise e Desenho	14
➤ Prova de conceito	14
➤ Diagramas de Casos de Uso	15
➤ Arquitetura da Aplicação (Tecnologias Usadas)	16

➤ Ferramentas Utilizadas no Desenvolvimento	17
➤ Desafios no Desenvolvimento	17
✓ Comunicação entre aplicação e servidor:	18
✓ Enviar informação através do servidor:	18
✓ Utilização de Google Maps API:	20
✓ Bug de formatação JavaScript:	20
✓ Má execução da aplicação em dispositivos móveis	21
✓ Problemas de execução de código	21
Funcionalidades	22
✓ Login	22
✓ Registar	23
✓ Adicionar anúncios	24
Resultados	25
➤ Pagina Principal	25
➤ Login	25
➤ Registrar	26
➤ Barra Principal	27
✓ Home	27
✓ Perfil	27
✓ Categorias	27
✓ Camera	27
✓ Adicionar Anúncios	29
➤ Sub-barra do perfil do anúncio	29
➤ Limitações	30
Conclusões e trabalho futuro	31
Bibliografia	32
Glossário	33

Resumo

A tecnologia é algo que evolui a cada dia que passa, e o seu desenvolvimento tem sido exponencial ao longo dos últimos anos. Consequentemente tornou-se uma arma poderosa para todo o tipo de serviço, desde comunicação, informação, lazer, etc.

Assim sendo, o objetivo é associar o máximo conforto e mobilidade para usar estes serviços e para isso estes mesmos foram introduzidos dentro de dispositivos móveis a partir de aplicações.

O interesse de todas as entidades que criam aplicações é poder estendê-las para o maior número de consumidores possível e para isso necessitam de fazer com que a aplicação funcione em todas as plataformas e sistemas. Devido à diferença das linguagens de programação usadas para cada sistema operativo é praticamente necessário refazer a aplicação para cada sistema pretendido e adaptá-la a cada plataforma em que esta possa ser executada.

No entanto hoje já é possível contornar esse problema. Existem tecnologias capazes de criar aplicações que podem ser executadas em mais do que um sistema operativo e adaptam-se automaticamente à respetiva plataforma. Ou seja estão preparadas para qualquer tipo de dispositivo e resolução.

Assim sendo o objetivo deste projeto foi explorar estas tecnologias multiplataforma, nomeadamente as que têm como base o HTML5 e JavaScript.

Como prova de conceito foi desenvolvida uma aplicação no qual, os consumidores fazem publicidade sobre o seu serviço, pondo apenas informação base como nome, morada, descrição e um preço exemplo.

Além de fazer publicidade ao negócio esta permite encontrar quais os serviços mais próximos da localização do utilizador e disponibiliza o trajeto para a opção selecionada.

Os utilizadores podem ainda enviar feedback aos serviços através de comentários.

Abstract (em Inglês)

Technology has been a subject with a lot of developing and improvement since the last 10 years. It has become a powerful weapon to be used by every kind of service, namely communication, information, leisure, etc.

The main goal is to ally comfort and mobility do this services. Therefore all these services were implemented in mobile devices by developing applications to run on these devices.

The main interest of every entity that creates any kind of mobile application is to extend their service and make it available to the biggest amount of users possible and to do that they need to reproduce the main application to every Operative System and platforms they wish to run due to the difference of the programming languages used and the different requirements of different platforms.

However, now it's possible to solve that problem by using multi-platform technologies to create these mobile applications. Multi-Platform technologies allow us to make the application only once, and run it on every system and every platform with automatic adaptations to any requirement.

Therefore, this project's goal was to explore these technologies using mainly HTML5 and JavaScript.

To make something concrete we made a mobile application where users advertise their business posting basic information like name, address, description and model price list.

Besides the advertising users can also find the nearest services posted on the application with all the steps to get to the location, and even give feedback to the service entity by posting comments

Introdução

Hoje em dia está cada vez mais repartida a variedade de sistemas operativos (SO) a usar nos dispositivos móveis. Apesar de a maioria utilizar IOS ou Android, os grandes rivais de mercado, temos também concorrentes como Windows Phone

Assim sendo quando queremos fazer uma aplicação (APP) é importante que esta chegue a maior parte da população possível. Este objetivo torna-se, contudo, complicado pois o desenvolvimento das aplicações é diferente para cada Sistema Operativo. Por exemplo:

- ✓ IOS é desenvolvido a partir duma linguagem de programação nova, criada pela Apple chamada de Swift que vem baseada de Objective C
- ✓ Android usa maioritariamente Java
- ✓ Windows Phone usa .Net, C#, VB ou C++

Assim, atualmente, é praticamente impossível construir uma aplicação igual para cada SO. Caso quiséssemos que estas aplicações fossem compatíveis também noutras plataformas como tablets e até mesmo desktops seria também necessário remodelações para cada um destes dispositivos.

Por isto mesmo foram criadas formas de criar aplicações suportadas por todos (ou maior parte) os SO e plataformas.

Sendo assim, o nosso objetivo é procurar conhecer melhor este conceito de multiplataforma optando também por desenvolver uma aplicação que seja capaz de ser executada em vários dispositivos com diferentes sistemas operativos e explorar as funcionalidades que esta oferece.

A aplicação desenvolvida como prova de conceito consiste assim num serviço de divulgação de negócio (Anúncios) que faculta informações ao cliente incluindo as direções necessárias até ao local do serviço, por exemplo restaurantes numa zona do país desconhecida para o utilizador. Desta forma o utilizador poderá encontrar informações como localização, preço e até comentários do serviço que necessita.

Enquadramento teórico

Tipos de Aplicações Mobile

As aplicações mobile podem ser nativas ou web. As **aplicações nativas** são aplicações desenvolvidas apenas para um sistema operativo, ou seja estas só podem ser executadas no respetivo SO enquanto as **aplicações Web** são aplicações desenvolvidas para correrem por exemplo num browser daí a não necessitarem sequer de ser obrigatoriamente instaladas nos dispositivos.

A tabela seguinte, que foi retirada do site (<http://pplware.sapo.pt/internet/apps-nativas-vs-web-apps-quais-devo-escolher/>), compara as vantagens e desvantagens dos dois tipos de aplicações. As vantagens estão com cor verde; as desvantagens, com cor vermelha.

	Aplicações Web	Aplicações Nativas
Integração	1)O acesso hardware do dispositivo é mais limitado embora haja API de geolocalização e outras, a flexibilidade é menor. O acesso APIs de software(ex. Sistema de notificações). Também é incompatível.	1)O acesso à câmara, botões, acelerómetro e outros sensores, GPS, sistema de notificações, widgets nativos, contactos, entre outros, abre exponencialmente o leque de oportunidades/possibilidades. 2)A manutenção de estado da aplicação (temporário ou persistente) é muito mais fácil e versátil.

	2)É mais fácil manter o estatuto da aplicação, providenciar modo offline, multitasking, etc.	
Abrangência	1)Qualquer dispositivo com um browser lhes pode aceder, ou seja, toda a miríade de dispositivos móveis e também desktop	1)São destinados a um só sistema; se se quiser abranger mais, há que fazer um projeto quase do zero.
Ciclos de Lançamento	1)Basta um simples deploy e o produto e o produto esta no ar: pequenas alterações estão em produção em segundos. O processo é muito mais ágil.	1)O processo de packaging e deployment pode ser moroso. Isto é especialmente patente sempre que é necessária uma alteração (ex. na Apple Store, pode demorar dias e até ser recusada).
Tempo e custos	1)O desenvolvimento web é muito mais generalizado. Assim, o TtM é muito menor, tanto no lançamento como em versões posteriores. 2)Usam as tecnologias standard e já conhecidas da web (HTML, CSS,	1)Há menos programadores de apps nativas que para a web em geral. Pior: é preciso diferentes programadores por cada sistema que se queira abranger. 2)Usam tecnologias nativas (ex. Swift); mesmo que seja Java, ainda existe uma curva de aprendizagem e portanto,

	<p>JavaScript), pelo que há muito mais programadores com experiência disponíveis. Alterações são muito mais rápidas (muitas vezes, basta um Save e um F5).</p> <p>3) Há imensos dispositivos e browsers (e suas versões) para suportar, pelo que é um ecossistema aberto e instável.</p>	<p>são menos universais. Alterações (pequenas ou grandes), obrigam a um deploy (no simulador ou dispositivo).</p> <p>3) São oferecidos SDK, frameworks e um sistema base fixo, aos programadores, os quais propiciam um ecossistema mais controlado e fechado (só há preocupações quando é lançada nova versão de S.O.).</p>
Comunicação	<p>1) É menos comum e mais difícil criar uma campanha de marketing à volta de uma web app.</p>	<p>1) É mais fácil de comunicar e publicitar um app nativa em eventuais campanhas.</p> <p>É provavelmente por isso que este tipo de app é muito mais popular. Por outro lado, o facto de estar numa loja cria uma dinâmica própria.</p>
Monetização	<p>1) É preciso implementar o seu sistema de pagamento se quiser criar um app paga, o que é incomum.</p>	<p>1) Sendo que já há uma loja, com sistema de pagamento integrado, é algo que é garantido à partida.</p>

Usabilidade	<p>1) Provavelmente, o UX é pior, pois não se usam os componentes nativos e a velocidade final é inferior.</p>	<p>1) É mais fácil se obter um UX superior (incluindo acessibilidade). A velocidade é superior porque o código é nativo e não há downloads (da interface em si) envolvidos.</p> <p>2) A curto prazo, é pior, pois é preciso ir à loja, instalar (por vezes, colocar a palavra-chave) e esperar.</p> <p>3) A longo prazo, é melhor, pois uma instalação cria um ícone (e um conceito) dedicado no sistema (embora isso seja possível nas web, não é tão linear/comum).</p>

Tecnologias Nativas

As tecnologias nativas são essencialmente para os sistemas operativos Android e OS X/iOS.

Android é um sistema operativo desenvolvido com foco principal em dispositivos móveis (Smartphones, tablets ,etc.) com tela sensível ao toque. Está presente também em Televisões, carros, relógios de pulso, câmaras digitais, etc. É atualmente desenvolvido pela Google e tem como base o núcleo Linux.

Java é uma linguagem orientada a objetos que está a ser desenvolvida ao longo dos tempos. A grande vantagem desta linguagem é criar aplicações que podem ser executadas em várias plataformas desde que tenham a “Java Virtual Machine”. Esta funcionalidade deve-se ao facto do código java ser compilado para um “bytecode” que vai ser interpretado pela “Java virtual machine” e consequentemente executado no hardware do computador.

Objective-C é a linguagem usada para desenvolver sistemas operativos como **OS X** e **IOS**. Maior parte da sintaxe foi herdada de C como por exemplo: as expressões, declaração e chamada de funções, entre outros. Também, obteve sintaxe do Smalltalk, ou seja, é uma sintaxe para aspetos orientados a objetos. Esta linguagem transfere mensagens no estilo Smalltalk para C.

Tecnologias Web

São essencialmente baseadas em **HTML5**, **CSS** e **Javascript**. O **HTML5** é uma linguagem focada em representar e estruturar conteúdo web. Ao longo dos tempos esta tecnologia tem vindo a evoluir, incorporando deste modo novos recursos.

Hoje em dia, **HTML5** ocupa o modelo do **HTML** e **XHTML**.

O **CSS3** é uma linguagem de folhas de estilo que define a aparência das páginas web.

O **Javascript** é uma linguagem de programação interpretada, ou seja, esta é executada dentro de programas e/ou outras linguagens de programação. É uma linguagem com orientação a objetos e é atualmente a principal para programação client-side em browsers isto porque executa scripts do lado do cliente e interage com o usuário sem a necessidade deste script passar pelo servidor, controlando o browser, realizando comunicação assíncrona e alterando o conteúdo do documento exibido.

Atualmente existe a tecnologia **JQuery** que simplifica ainda mais o uso destes scripts que interagem com o **HTML**. Esta biblioteca de **JavaScript** disponibiliza uma API suportada por todos os browsers que simplifica toda a manipulação de eventos, animações e o **HTML** em geral.

Dentro das várias frameworks de **Jquery** existentes, há uma especificamente para mobile que se chama **Mobile J-query**. Este permite construir web sites e aplicações que se adequam automaticamente a qualquer SO e/ou plataforma, juntamente com Javascript e CSS.

Frameworks e Tecnologias Baseadas em Javascript, HTML5 e CSS

PhoneGap é um framework de desenvolvimento de aplicações para dispositivos móveis. Para desenvolvimento das aplicações este software usa JavaScript, HTML5 e CSS3 em vez de depender de API'S individuais para desenvolvimento aplicações IOS ou Android.

A tabela seguinte mostra as plataformas e funcionalidades suportadas pelo PhoneGap:

Característica	iPhone / iPhone 3G	iPhone 3GS e mais novos	Android 1,0- 4,4	Windows Phone
Acelerômetro	SIM	SIM	SIM	SIM
Câmera	SIM	SIM	SIM	SIM
Bússola	N/D	SIM	SIM	SIM
Contactos	SIM	SIM	SIM	SIM
Arquivo	SIM	SIM	SIM	SIM
Geolocation	SIM	SIM	SIM	SIM
Mídia	SIM	SIM	SIM	SIM
Rede	SIM	SIM	SIM	SIM

Notificação (alerta, som, vibração)	SIM	SIM	SIM	SIM
Armazenamento	SIM	SIM	SIM	SIM

O **JSON (JavaScript Object Notation)** é um formato leve para troca de dados computacionais. É uma alternativa ao uso de XML pois torna-se mais fácil de escrever neste formato. JSON é construído por 2 estruturas:

- Um grupo de pares de nomes e/ou valores, que em termos de linguagens de programação podemos dizer, objetos, structs, hash table, associative array, etc.
- Uma lista de valores ordenada, ou seja, arrays, vetores, listas ou sequencias.

Frameworks e Tecnologias para Aplicações Nativas

O **Xamarin** torna possível a criação de aplicações nativas para cada sistema operativo móvel a partir da linguagem C#, dando acesso a API's nativas, UI, etc.

Tem também disponível uma extensão para Visual Studio e também o próprio IDE Xamarin Studio que permite fazer aplicações híbridas, ou seja, para qualquer plataforma, também sempre baseado em código C#.

Frameworks e Tecnologias para Aplicações Nativas, WEB e Híbridas

Tererik é uma framework de desenvolvimento de aplicações para dispositivos desktops e móveis. Disponibiliza ferramentas de software para o desenvolvimento destes. Os seus instrumentos estão direcionados para auxiliar diversas tecnologias, tais como .NET, ASP.NET, AJAX, ASP.NET MVC, WPF, Silverlight e do Windows/Windows Phone. Em 2011, introduziu suporte para HTML5 e JavaScript. Recentemente, passou a

disponibilizar também uma plataforma de desenvolvimento para aplicações nativas, web e híbridas.

Análise e Desenho

A prova de conceito

A aplicação desenvolvida permite a escolha categorias de serviços/negócios/lojas, tais como:

- Restaurantes
- Shopping
- Combustível
- Mecânica
- Eventos

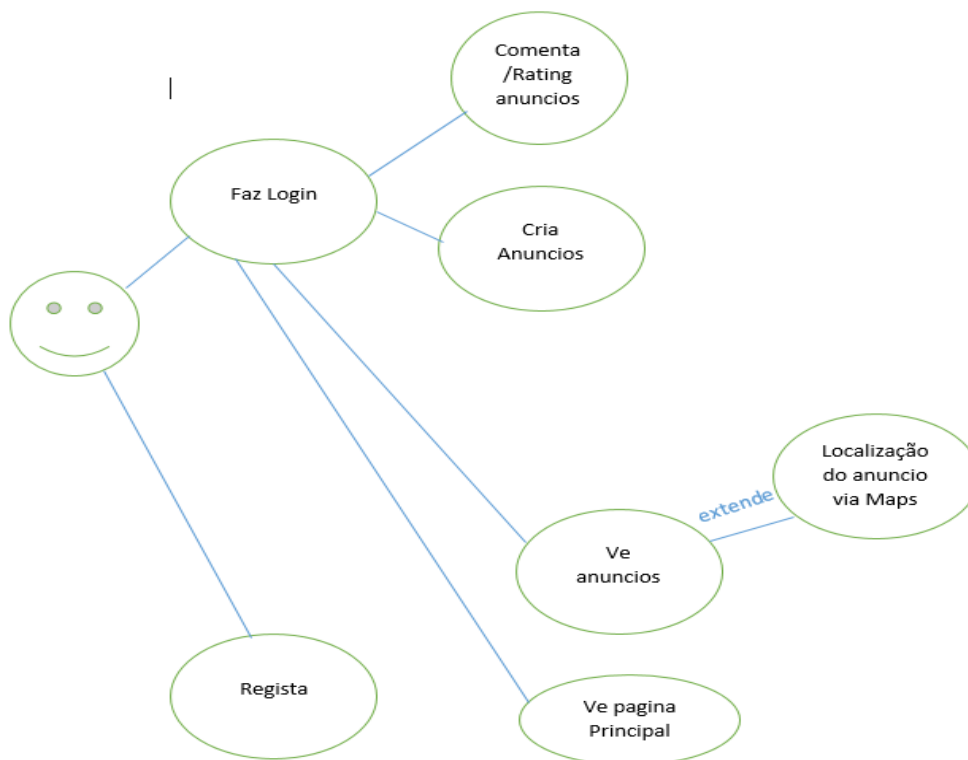
Ao escolher uma das categorias o utilizador deve ter acesso a informações tais como:

- Informação Geral
- Preços
- Comentários
- Localização
- Contactos

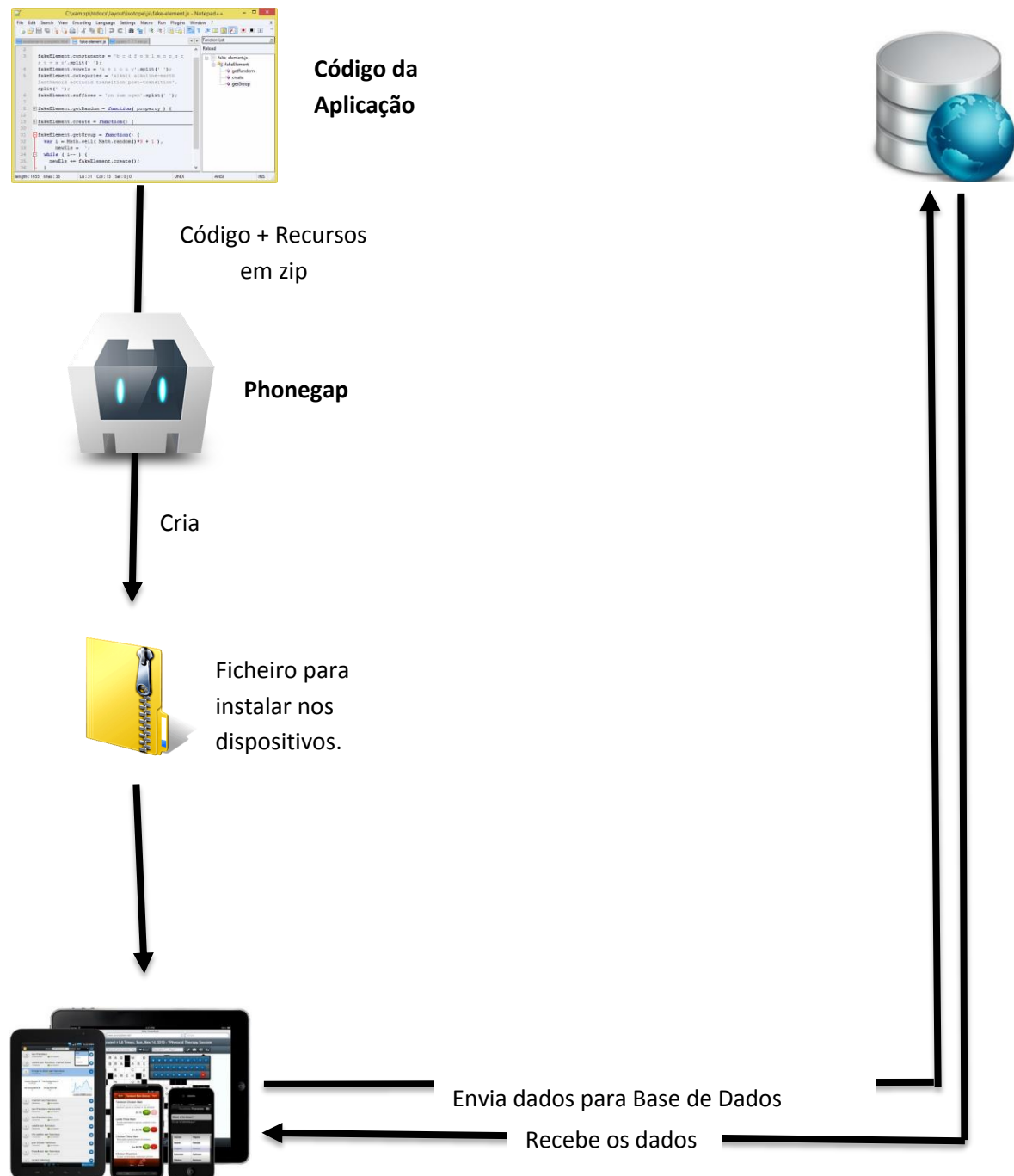
Estas informações são respetivas a anúncios localizados a uma distância de 50Km da posição atual do mesmo. Além de visualizar toda esta informação, os utilizadores podem ainda deixar feedback no anúncio (comentário). Todos estes anúncios que são apresentados nas categorias, são introduzidos por utilizadores que queiram promover o seu negócio juntando-se assim a esta rede.

Diagrama de Casos de Uso

Funcionalidades:



Arquitetura Técnica da Solução



A aplicação começa por ser desenvolvida a partir do IDE Notepad++, ficheiros HTML, Javascript, CSS e PHP foram codificados a partir deste. Estes ficheiros (excepto ficheiros PHP, estes ficam alojados num servidor) são introduzidos na pasta “www” da pasta do Projeto pré criada pelo Phonegap. Seguidamente a pasta do projeto é

comprimida em Zip e enviada fazendo upload no endereço <https://build.phonegap.com/apps/>. A partir daqui o website disponibiliza para download os ficheiros de instalação da aplicação para Android, Windows Phone e IOS. Instala-se no dispositivo e a aplicação fica disponível para usar. Como esta requer envio e retorno constante de dados, foi criada uma Base de Dados num servidor externo onde estão alojados também os ficheiros PHP que são invocados pela aplicação.

Ferramentas Utilizadas no Desenvolvimento

Visual Studio:

- O projeto inicialmente foi criado no Visual Studio e todo o código era escrito neste mesmo. Com a extensão Telerik era possível simular a aplicação depois de compilada em dispositivos virtuais Android, IOS e Windows Phone.

Notepad ++:

- Todos os ficheiros de código pertencentes à aplicação (Javascript, CSS, HTML e PHP) foram criados a partir do Notepad++

Mysql Workbench:

- Usado para visualizar e editar a Base de Dados criada num servidor externo.

Desafios no Desenvolvimento

Ao longo do desenvolvimento deste projeto tivemos algumas dificuldades no qual maior parte foram superadas.

As dificuldades foram as seguintes:

- **Comunicação entre aplicação e servidor:** Um dos grandes entraves do projeto foi fazer com que a aplicação comunicasse com um servidor externo e assim pudesse executar as funções PHP que estavam neste mesmo servidor.

À partida iríamos usar o método \$POST do JQuery mobile que envia algo para o servidor e recebe uma resposta através duma função “callback”, mas após vários testes este método não teve qualquer sucesso.

Assim sendo, tentámos usar o método \$GET cujo objetivo é trazer informação do servidor dando a possibilidade de enviar informação como parâmetros. Este método resultou e deu-nos a possibilidade de executar todas as funções necessárias que dependiam do servidor, tais como: Login, Registar, Load da informação de perfis e de anúncios, etc.

- **Enviar informação através do servidor:** Outra ocasião a que podemos chamar de ligeira dificuldade foi a forma de enviar uma quantidade dinâmica de informação através do servidor.

Esta informação seria dinâmica porque nunca sabíamos ao certo o volume desta pois este pode ser alterado ao longo do tempo, por exemplo se quisermos seleccionar da base de dados a lista de utilizadores registados, esta varia conforme o registo de novas entidades. Portanto teríamos de preencher um array dentro dum ciclo e depois enviá-lo.

Em PHP, toda a informação que é enviada de volta para a aplicação é codificada via **JSON**, e ao início quando as funções PHP retornavam uma quantidade sempre certa de informação, este retorno podia ser predefinido por nós sem necessitar de preencher qualquer array e desta forma tudo corria na perfeição. Quando chegámos à fase em que a quantidade de informação já não era constante ao longo do tempo, necessitámos de fazer com que o JSON codificasse um array de informação e assim o enviasse. Depois de várias tentativas todos os retornos estavam em branco, não conseguíamos fazer com que o ficheiro PHP mostrasse o array que tínhamos introduzido no echo. Depois de pesquisar chegámos à conclusão que a informação codificada por JSON tem de estar na codificação Unicode UTF-8. Assim sendo toda a Base de Dados foi convertida e prefinida para uma codificação UTF-8 e a partir daqui todos os valores dentro do array do PHP já seriam codificados pelos JSON e enviados para a aplicação com sucesso.

```
$dbuser = 'root';
$dbpass = '';
$host = '127.0.0.1';
$db = 'tfc_db';
mysql_connect($host, $dbuser, $dbpass) or die(
mysql_select_db($db) or die(mysql_error()));

$user=$_GET['user'];
$pass=$_GET['pass'];

$sql=mysql_query("SELECT * FROM tfc_db.user W
if (mysql_num_rows($sql) > 0){
$result=mysql_query("SELECT username,id_u
$row = mysql_fetch_assoc($result);
$username = $row['username'];
$user_id = $row['id_user'];
$name = $row['name'];
$subname = $row['subname'];
$email = $row['email'];
echo json_encode( array( "result"=>"succe
}else{
echo json_encode( array( "result"=>"fail"
```

Código: PHP

```
<script type="text/javascript">
var submitClicked = function (argument) {
$("#submit").click(function (e) {
$.get('http://127.0.0.1/php/login.php', {
user: $("#user").val(),
pass: $("#pass").val()
}, function (answer) {
if (answer.result == "success") {
var userInfo = {
"user_id": answer.user_id,
"username": $("#user").val(),
"email": answer.email,
"name": answer.name,
"subname": answer.subname,
};
//Guardar informação da var "userInfo" na localStorage.
localStorage.setItem('loggedUser', JSON.stringify(userInfo));
window.location.replace("perfil.html");
} else {
alert("User ou Pass Errados");
}
}, "json"
));
}
$(document).on('ready', submitClicked);
</script>
```

Código: Javascript

- **Utilização de Google Maps API:** A aplicação disponibiliza a visualização da morada introduzida pelo utilizador quando cria o anúncio para que este possa confirmar a mesma.

Além disso disponibiliza também as direções do local atual do utilizador até ao destino que será a morada do anúncio escolhido. Estes mapas são resultado duma API disponibilizada pela Google.

O problema é que poucos ou nenhuns exemplos de mapas estavam a funcionar na aplicação. Caso acedêssemos á pagina via browser os mapas funcionavam, mas acedêssemos pelo dispositivo, ou seja, por aplicação estes não funcionavam. Ao reler documentações do PhoneGap descobrimos que é necessário introduzir certas permissões e plugins no ficheiro de configuração do Phonegap/aplicação e o que nos faltava era introduzir o plugin de Geolocation através de uma linha de código.

Depois disto todos os maps e funções destes passaram a funcionar.

- **Bug de formatação Javascript:** Quando introduzimos uma listview dentro do body do HTML esta apresenta-se sem qualquer problema e devidamente formatada pelo CSS default do JQuery.

Mas caso queiramos uma listview dinâmica, ou seja, uma listview que é criada e preenchida dentro duma função javascript esta não sofre qualquer formatação. Isto acontece quando queremos apresentar as categorias para o utilizador escolher e os comentários pertencentes aos anúncios. Isto é

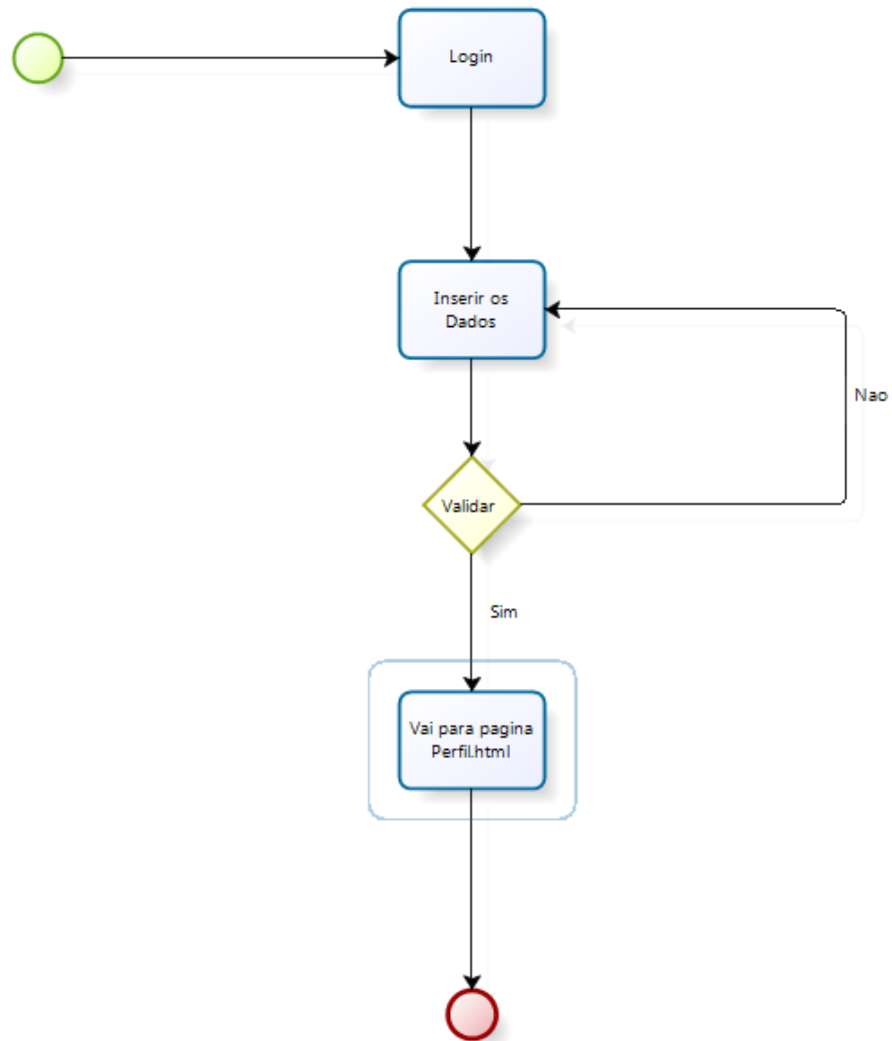
informação que não é constante ao longo do tempo pois esta pode ser alterada, daí a ser preenchida por uma função JavaScript que recebe a informação do servidor.

Após várias tentativas e remodelações no código javascript foi possível colocar esta funcionalidade corretamente.

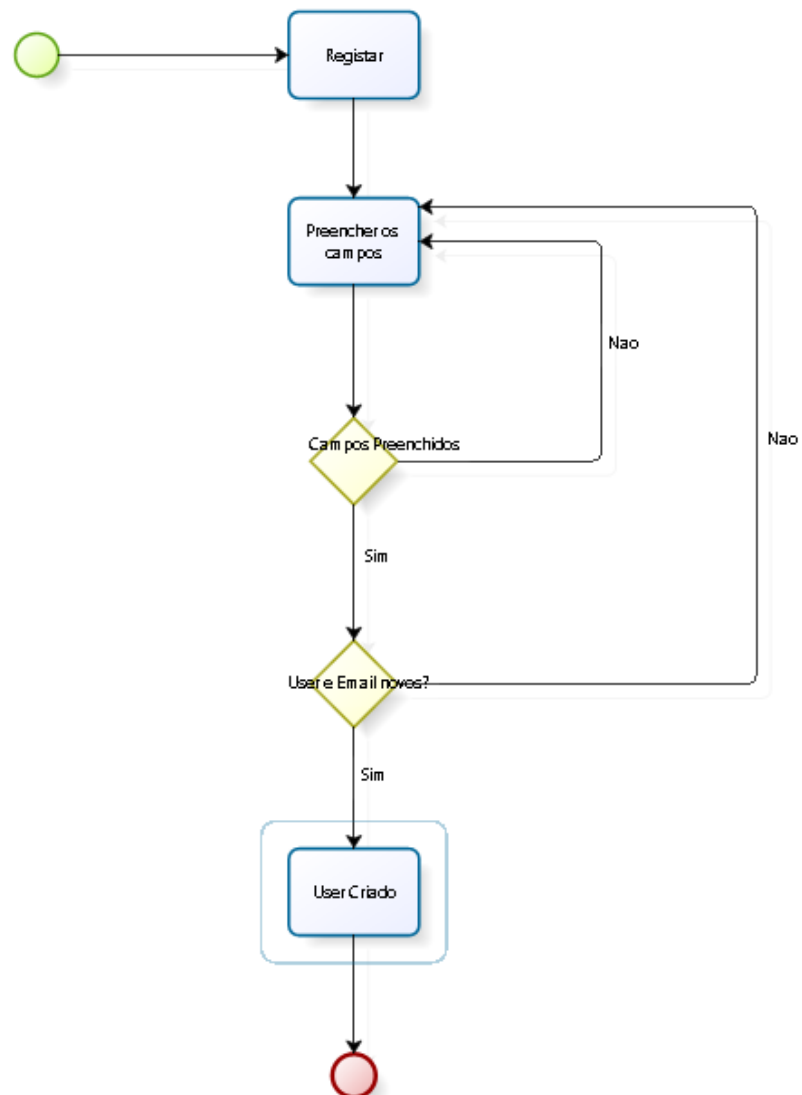
- **Má execução da aplicação em dispositivos móveis:** Depois da aplicação já estar praticamente feita há bastante tempo deparámo-nos com um problema enorme. Depois de efetuarmos algumas alterações na aplicação fizemos o processo todo habitual e instalámos o ficheiro “.apk” no dispositivo Android. Estranhamente a aplicação apresentava-se praticamente sem estilo de página o que indicava que o CSS não estaria a funcionar e com os botões inutilizáveis o que indicava também que o código Javascript não funcionava também. Verificámos que no browser a aplicação corria perfeitamente ao contrário do dispositivo. Experimentámos correr uma versão antiga (que funcionava no dispositivo) e esta estava também “danificada”. Até à data estamos a tentar resolver este problema mas ainda sem sucesso.
- .
- **Problemas de execução de código:** Um dos problemas que se mantém é o facto da página “categorias” e páginas que utilizem o Google maps necessitem de serem atualizadas uma vez para o JavaScript executar como deve ser.

Funcionalidades

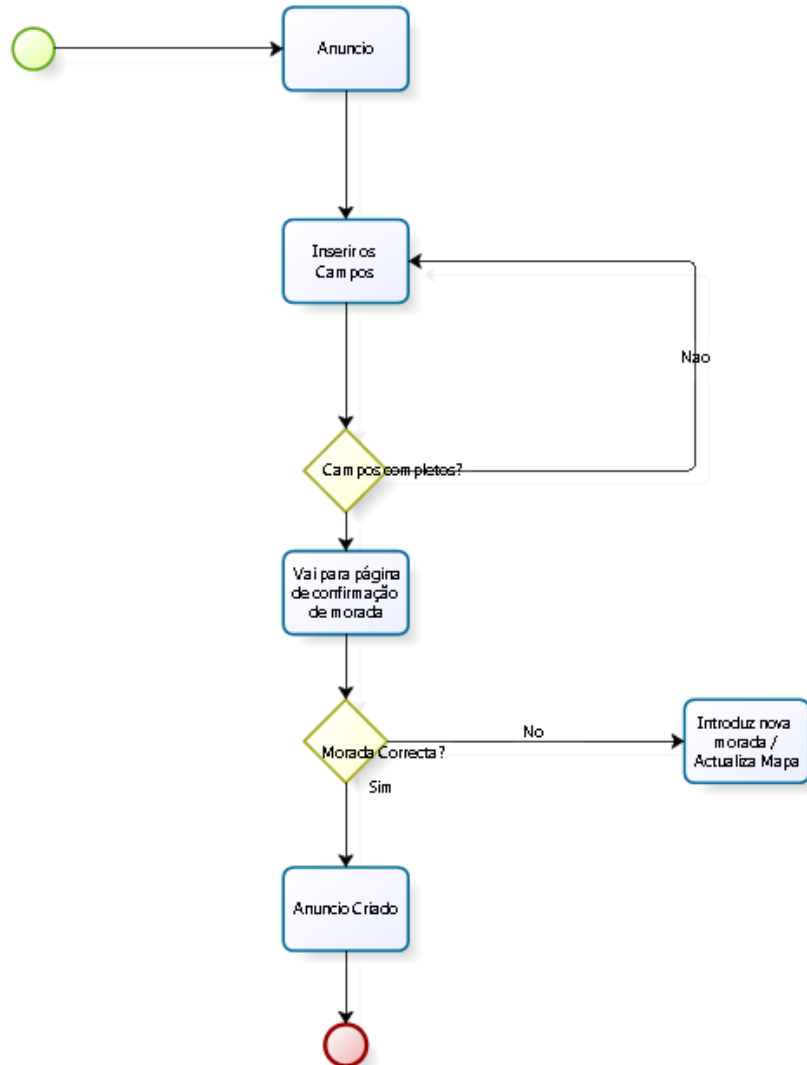
Login



Registrar

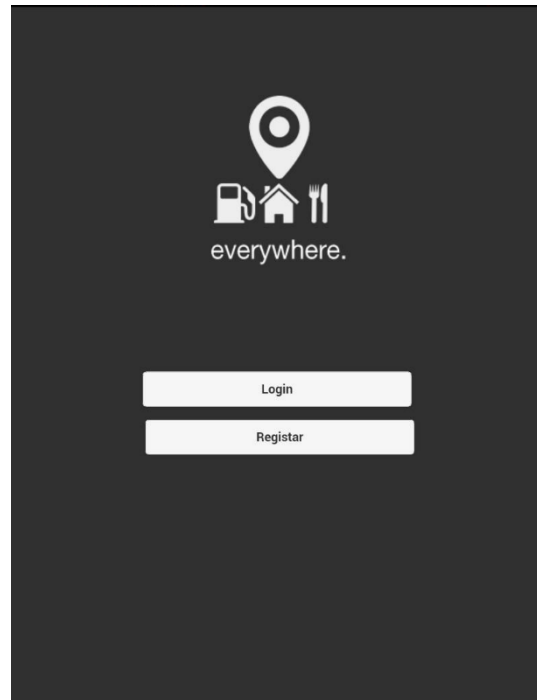


Criar Anuncio



Resultados

Pagina Principal da aplicação



Login: o utilizador vai introduzir os dados no campo, para entrar na página principal onde terá várias opções, tais como Inserir anúncio, perfil (ver o seu perfil), categorias (para escolher uma determinada categoria, onde vai listar uma lista de anúncios).

Pode ainda ver os comentários das outras pessoas sobre o seu anúncio e, ainda consegue inserir comentários sobre a sua experiência num determinado lugar.

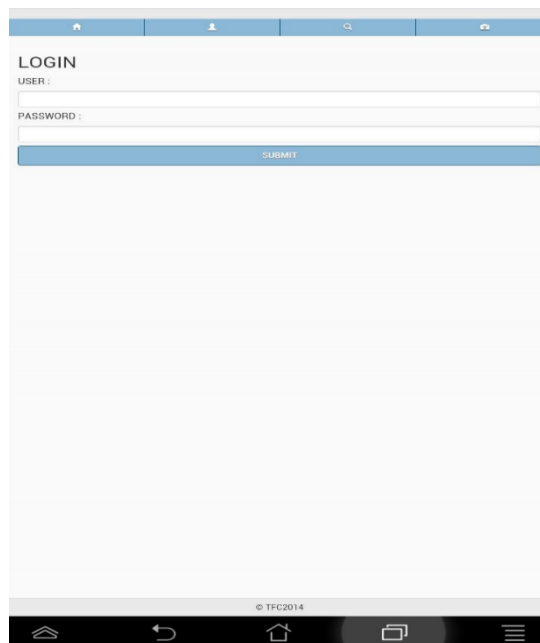
A screenshot of a mobile application's login screen. At the top, there is a blue header bar with four icons: a home icon, a person icon, a magnifying glass icon, and a lock icon. Below the header, the word "LOGIN" is displayed in bold. Underneath, there are two input fields: the first is labeled "USER:" and the second is labeled "PASSWORD:". Below these fields is a blue button with the text "SUBMIT" in white. The background of the screen is a light gray. At the bottom, there is a black bar with five icons: a home icon, a back arrow icon, a home icon, a folder icon, and a list icon. In the center of the black bar, the text "© TFC2014" is visible.

Figura: Login

Registar: o utilizador vai inserir os dados pedidos nos campos, para que ele crie uma conta no qual vai poder usufruir dos privilégios da aplicação.

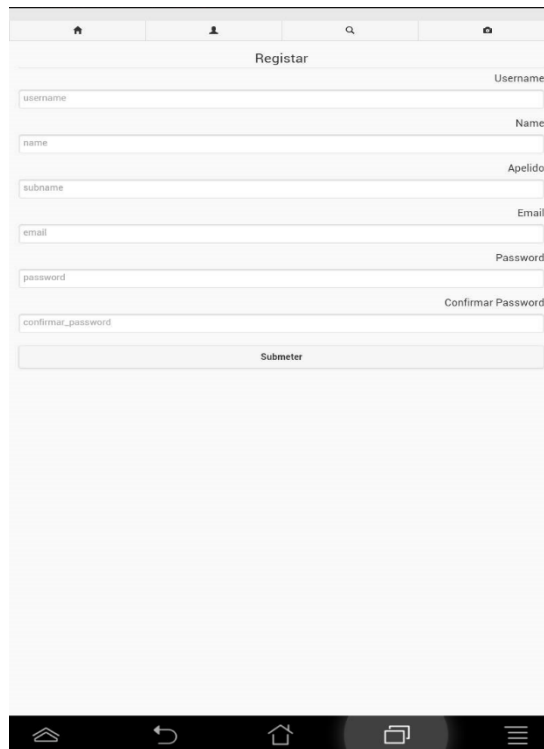
A mobile application registration screen titled "Registrar". At the top, there is a navigation bar with four icons: a home icon, a person icon, a magnifying glass icon, and a lock icon. Below the title, there are several input fields: "username" (labeled "Username"), "name" (labeled "Name"), "surname" (labeled "Apelido"), "email" (labeled "Email"), "password" (labeled "Password"), and "confirmar_password" (labeled "Confirmar Password"). Each field has a small label to its right. At the bottom of the form is a "Submeter" button. The screen is framed by a black mobile OS navigation bar at the bottom with standard icons for back, home, and recent apps.

Figura: Registrar

Barra Principal

Home: retorna para a página principal

Perfil: onde mostra a informação do utilizador, e também pode – se ver os seus comentários feitos (Sub-barra em baixo, onde leva para a página dos comentários)

Camera: Implementação Futura

Categorias: escolha categorias de serviços/negócios/lojas, tais como: Restaurantes, Shopping, Combustível, Mecânica, Eventos.

Ao escolher uma das categorias o utilizador deve ter acesso a informações tais como: Informação Geral, Preços, Comentários, Localização, Contactos.

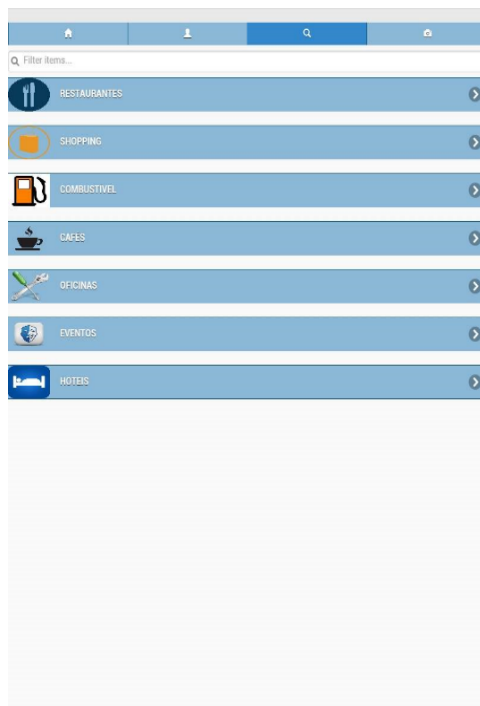


Figura: Categorias

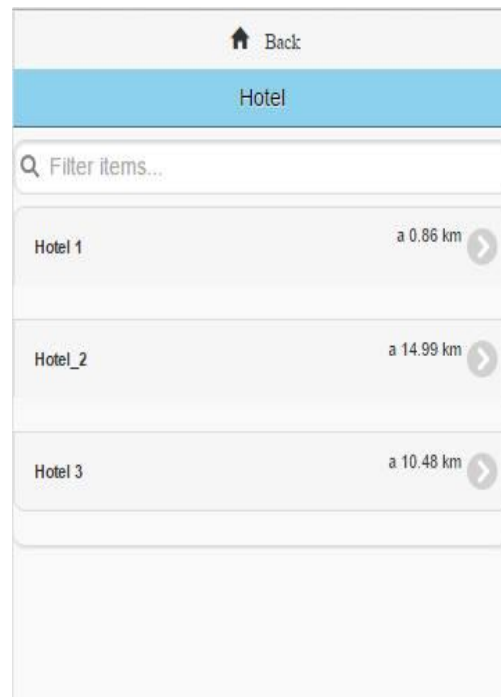
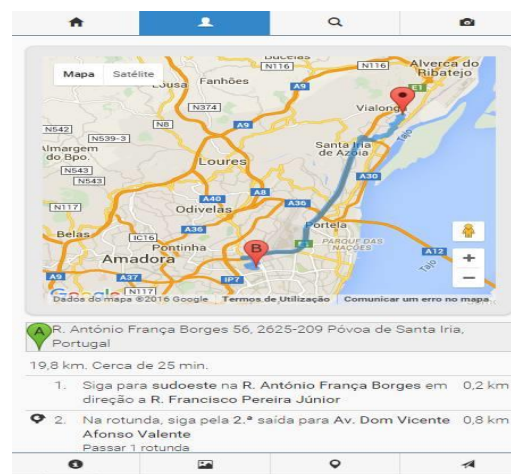
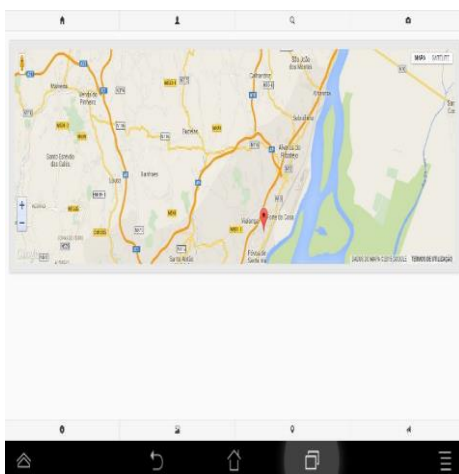


Figura: Anúncios de uma categoria

Estas informações são respetivas a anúncios localizados a 50 Km (máximo) da posição atual do utilizador. Além de visualizar toda esta informação, os utilizadores podem ainda deixar feedback no anúncio (comentário e classificação). Todos estes anúncios que são apresentados nas categorias, são introduzidos por utilizadores que queiram promover o seu negócio juntando-se assim a esta rede.



Figuras: Mapa/Localização

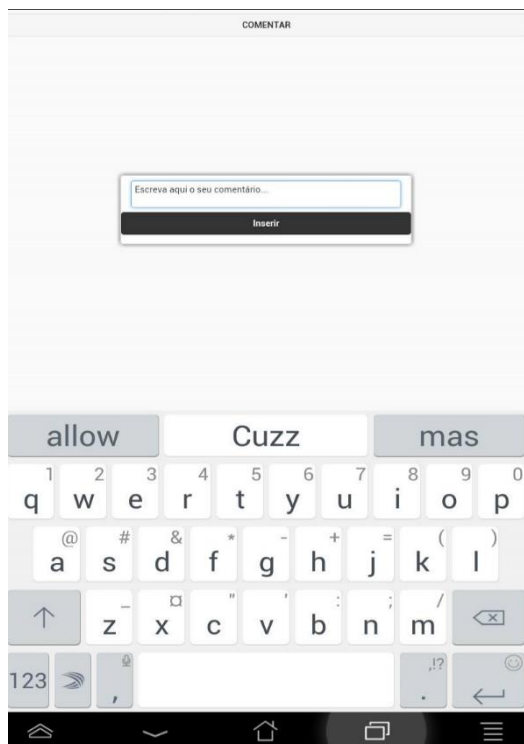
Adicionar Anuncio: o utilizador vai inserir os dados pedidos nos campos, para que ele crie o seu anúncio. (Ver Imagens em baixo)

The image displays two screenshots of a mobile application interface for adding a new advertisement. The left screenshot shows the main form titled "NOVO ANÚNCIO". It includes fields for "TÍTULO", "CATEGORIA" (with a dropdown menu showing "SHOPPING"), "EMAIL", "CONTACT", "LOCALIZAÇÃO" (with sub-fields for "codigo postal", "rua", "cidade", and "pais"), and a table for "PRODUTO" and "PREÇO(€)". The table has three rows, with the first two filled with "Intel i7 950" (250,50) and "Nvidia GeForce 980" (600,30). Below the table is a "DESCRIÇÃO" field. At the bottom is an "Inserir" button. The right screenshot shows the same form, but with the "CATEGORIA" dropdown menu open, displaying a list of categories: Shopping, Restaurante, Hotel, Mecânica, Combustível, Eventos, and Café. The "Inserir" button is also visible at the bottom.

Figuras: Adicionar anúncio

Sub-barra do perfil do anúncio

Esta barra contém um menu que nos permite ver informação sobre o anúncio, a localização e direções para o mesmo e comentários escritos por outros utilizadores.



Figuras: Comentários

Limitações

A maioria das framework não foram testadas nos dispositivos da Apple devido ao facto de serem pago para podermos visualizar.

Conclusões e trabalho futuro

Com este projeto foi possível testar varias tecnologias multiplataforma para assim conseguir ultrapassar o obstáculo que existia em poder criar aplicações apenas uma vez para uma variedade de dispositivos.

Foi possível desenvolver uma aplicação que surge como um serviço de divulgação e avaliação (através do feedback) aliada a um sistema de busca via GPS que alia e aproxima o negócio do consumidor e ao mesmo tempo ajuda pessoas que visitem locais que não conhecem a encontrar serviços que necessitem.

Uma conclusão importante é que apesar das tecnologias multiplataforma demonstrarem potencial para resolver problemas no futuro e tornar mais eficiente e económico a criação de aplicações que não necessitam de ser recriadas consoante os sistemas alvo, não deixa de ser uma tecnologia não nativa, ou seja, a estabilidade ainda não é muito boa em relação ao que as tecnologias nativas oferecem. Para aplicações muito mais rigorosas e personalizadas prevaleceria, à priori, o desenvolvimento em ambiente nativo.

No futuro seria apenas corrigido dois problemas que persistiram e adicionado duas funcionalidades. Seria resolver o problema que impossibilita de correr a aplicação num dispositivo móvel e o facto de certas páginas necessitarem de serem atualizadas para correr as funções corretamente. A funcionalidade adicionada seria a possibilidade de tirar fotos e criar uma galeria para cada utilizador. Tornaria a aplicação um pouco mais social de modo a que as pessoas partilhassem as suas experiencias nos locais visitados a partir de fotos. Outra funcionalidade seria a gestão dos anúncios criados pelo utilizador para que pudesse ter maior controlo sobre estes.

Bibliografia

Referências em texto:

- [1] – iOS: <http://pt.wikipedia.org/wiki/IOS>
- [2] – Android: <http://pt.wikipedia.org/wiki/Android>
- [3] – Windows Phone: https://pt.wikipedia.org/wiki/Windows_Phone
- [4] - HTML5 – <http://pt.slideshare.net/AlexPorfirio/o-que-html5>
- [5] - HTML5 – <http://pt.wikipedia.org/wiki/HTML5>
- [6]-HTML5–[http://www.tracto.com.br/o-que-e-html5-por-que-se-da-tanta-importancia-a-ele/ ???](http://www.tracto.com.br/o-que-e-html5-por-que-se-da-tanta-importancia-a-ele/???)
- [7] – CSS3 – <http://www.criarweb.com/artigos/introducao-a-css3.html>
- [8] – CSS3 – <http://pt.wikipedia.org/wiki/CSS3>

Outros:

<http://pt.wikipedia.org/wiki/Objective-C>

<http://en.wikipedia.org/wiki/PhoneGap>

<http://en.wikipedia.org/wiki/Telerik>

http://pt.wikipedia.org/wiki/Servidor_FTP

<http://pplware.sapo.pt/internet/apps-nativas-vs-web-apps-quais-devo-escolher/>

<http://www.w3schools.com/html/>

http://xamarin.com/?_bt=44014802708&_bk=xamarin&_bm=e&gclid=COyChur68MMCFcHHtAodQxQAnQ

<http://en.wikipedia.org/wiki/Xamarin>

<http://phonegap.com/>

[https://developers.google.com/maps/documentation/javascript/usage?hl=pt-](https://developers.google.com/maps/documentation/javascript/usage?hl=pt-PT)

[PT](#)

<https://developers.google.com/maps/faq?hl=pt-PT>

<http://mundogeo.com/blog/2011/11/01/google-maps-introduz-limite-para-uso-da-api/>

<http://json.org/>

<http://www.w3schools.com/json/>

Anexos (todo o source code do projecto deve ser colocado aqui)

- Se for muito só deve por as partes principais

Anexo de Manual técnico da aplicação (se existir)

- Deve ter isto... com imagens e de forma muito sucinta.

Anexo de Manual de utilizador da aplicação (se existir)

- Deve ter aqui por exemplo, como é que instala a aplicação.

Glossário

- I. **ByteCode** é o código interpretado pela Java Virtual Machine. Este código é o resultado da compilação do código Java com o propósito de ser depois executado no computador.
- II. **Java virtual Machine** é o programa responsável por converter o bytecode do código Java para um programa executável pela máquina onde este foi utilizado.
- III. **SmallTalk** é uma linguagem de programação puramente orientada a objetos, isto é, todos os elementos desta linguagem (classes, métodos, blocos de código) são considerados objetos.
- IV. **XHTML** é uma linguagem baseada em codificação de HTML e com as regras XML que possibilita a criação e apresentação de paginas Web