

# Conference Buddy CB 0.4

## Projeto Integrador

Preparado por

**Grupo 5**

**Nuno Alves - 32349**

**João Ribeiro - 32022**

**Filipe Correia - 31637**

Preparado para

**Professor Fernando Birra**

Última	27 Junho, 2013
Modificação:	
Versão	1.0
Documento:	<i>Relatório Final</i>

# Índice

<b>Histórico de versões .....</b>	<b>3</b>
<b>1.0 Introdução .....</b>	<b>4</b>
<b>2.0 Apresentação Geral do Sistema .....</b>	<b>4</b>
2.1 Contexto .....	4
2.2 Descrição do problema .....	5
2.3 Proposta de solução.....	5
2.4 Restrições gerais .....	9
2.5 Pressupostos e dependências.....	9
<b>3.0 Modelo do Domínio.....</b>	<b>10</b>
3.1 Dicionário de Termos .....	10
3.2 Diagrama de Classes .....	10
<b>4.0 Arquitectura e Desenho do Sistema .....</b>	<b>12</b>
4.1 Arquitectura .....	12
4.2 Modelo de Dados .....	14
<b>5.0 Desenho e Interface.....</b>	<b>15</b>
5.1 Filosofia Geral e Modalidades de Interação .....	19
5.2 Aspectos Gerais da Interface Gráfica .....	21
5.3 Destaques .....	22
<b>6.0 NFRs e Validação.....</b>	<b>25</b>
6.1 NFRs .....	25
<b>Apêndices.....</b>	<b>26</b>

**Histórico de versões**

Versão	Data	Autor(es)	Detalhes/Descrição
1	27 Jun. 13	Nuno Alves, João Ribeiro, Filipe Correia	

## **1.0 Introdução**

Este projeto criado na cadeira de Projecto Integrador tem como objectivo criar uma aplicação para tablets Android com o intuito de melhorar a experiência do participante numa conferência melhorando a abordagem ou mesmo resolvendo alguns destes problemas.

## **2.0 Apresentação Geral do Sistema**

### **2.1 Contexto**

Numa conferência científica de informática existem vários processos que ainda são feitos por papel (por exemplo, consultar o programa da conferência) que têm trazem alguns inconvenientes (por exemplo, se por alguma razão for preciso mudar o programa, as pessoas que só tem acesso ao programa em papel não ficam informadas) ou meios de comunicação “organização – participantes” pouco eficientes, dando ao participante uma experiência menos boa da conferência.

Este sistema é referente à concepção de uma Aplicação para Tablet Android que tem como objectivo melhorar a experiência do participante numa conferência melhorando a abordagem ou mesmo resolvendo alguns destes problemas.

## 2.2 Descrição do problema

Estes são os principais problemas identificados:

### **1) User – Story: Utilizar a Aplicação para mais que uma conferência;**

É bastante penoso ter de descarregar uma aplicação diferente de cada vez que queremos ir a uma conferência. Ter de aprender a mexer novamente na aplicação, ter de criar novamente uma conta para a aplicação (caso se aplique), informações que já tínhamos são perdidas... Acreditamos ser necessário que uma única aplicação sirva para várias conferências.

### **2) User – Story: Ver o Programa da Conferência;**

É essencial que este problema seja resolvido e bem resolvido. Das informações que temos acesso durante a conferência, o programa e detalhes do mesmo são o mais importante. Garantir que o programa está actualizado, dar acesso a muito mais informação do que um panfleto e de forma mais eficiente e dinamica do que um site web.

### **3) User – Story: Ser possível fazer download de pdfs e ou imagens relacionados com uma apresentação;**

Não sendo essencial, seria muito bom que durante a conferência (ou mesmo no fim da mesma, caso a organização assim queira) seja possível fazer o download do material relacionado com a conferência. É uma forma de poupar papel (para a organização) e muito comodo para o participante que tem acesso imediato e digital ao conteúdo.

### **4) User-Story: Poder construir a própria agenda;**

Tão essencial como poder ver o Programa da conferência é conseguir construir a própria agenda. Pretende-se que o possamos fazer ao mesmo tempo que vemos o programa geral, ficando assim simples e imediato de tomar decisões sobre o que queremos (porque podemos ver detalhes dos eventos) ou fazer alterações.

### **5) User-Story: Ter acesso a notícias dadas pela organização.**

Em conferências pequenas pode não haver grande problema, mas em conferências grandes, difundir uma noticia, por mais pequena e simples que seja pode ser bastante problemático. Queremos então acesso a noticias geradas pela organização.

**6) User-story: Conseguir ver na aplicação os posters disponíveis na conferência.**

Em conferências grandes muitas vezes não temos tempo para ver todos os posters que gostaríamos. Ou gostamos muito de um poster e queremos tê-lo em formato digital para mais tarde o podermos ler sossegadamente. Queremos portanto ter acesso aos posters através da aplicação.

**7) User-Story: Uma maneira fácil de trocar contacto com alguém.**

Não é incomum acabar uma conferência com um pequeno monte de cartões de visita com quem quisemos trocar contacto. E ao fim de um tempo nem nos lembramos da cara da pessoa com quem trocamos contacto. Seria portanto ótimo conseguir fazer isso através da aplicação de forma simples e eficiente. Queremos também poder gerir os contactos e também exportá-los para fora da aplicação.

## **Proposta de solução**

Segue-se a lista de funcionalidades da aplicação, qual a user-story (problema) que lhe deu origem e algumas decisões de implementação relevantes para essa funcionalidade:

### **1) Possibilidade de ver conferências disponíveis e executar o download de uma delas para utilização na aplicação;**

User – Story que lhe deu origem: Utilizar a Aplicação para mais que uma conferência;

Solução e Decisões de implementação:

Estão disponíveis para download várias conferências que a organização das mesmas tenha colocado no servidor da aplicação. Para facilitar este processo é possível utilizar CSVs para importar dados para o servidor. Escolhemos o formato CSV por ser extremamente simples de criar e perceber, além de ter um tamanho mais pequeno que a alternativa estudada (XML).

Só são mostradas conferências cuja a data de fim da mesma não ultrapasse a data actual.

É possível fazer download de qualquer conferência, vervo programa geral e detalhes dos autores, mas não é possível fazer download de papers / posters sem estar inscrito na conferência.

A inscrição do user na conferência é feita pela organização e o email de inscrição tem de ser igual ao email do user na aplicação para haver ligação entre inscrito - user.

É possível remover conferências da Aplicação (que tenham sido descarregadas).

### **2) - Possibilidade de ver programa geral da conferência, sessões e detalhes das sessões;**

User – Story que lhe deu origem: Ver o Programa da Conferência;

Solução e Decisões de implementação:

É possível consultar o programa completo da conferência, inicio e fim das sessões e outras actividades, ver detalhes das mesmas e informação sobre os autores.

Para fácil navegação, podemos ver um dia de cada vez e para saber os detalhes da sessão é só “clicar” na mesma.

### **3) - Possibilidade de download dos papers;**

User – Story que lhe deu origem: Ser possível fazer download de pdfs e ou imagens relacionados com uma apresentação;

É possível fazer download dos Papers associados às apresentações através da aplicação. Para isso temos de através do programa da conferência (ou através da agenda pessoal) aceder aos detalhes da sessão, e clicar no evento de que queremos fazer o download do paper. Resolvemos colocar desta forma por uma questão de organização.

### **4) - Possibilidade de definir agenda pessoal de acordo com o Programa da Conferência;**

User-Story que lhe deu origem: Poder construir a própria agenda;

Podemos construir uma agenda pessoal a partir do Programa de Conferência. Podemos escolher sessões a que queremos atender, marca-las e desmarcalas no Programa de Conferência ou na Agenda pessoal.

Temos também uma funcionalidade de Notificação que, se mudar o horario ou sala de uma sessão a que estamos inscritos, a aplicação apresenta uma notificação sobre o sucedido. Outras alterações são feitas automaticamente sem notificações.

#### **5) – Feed de Notícias**

User-Story que lhe deu origem: Ter acesso a notícias dadas pela organização.

Para cada conferência existe um feed de notícias actualizado automaticamente que estabelece um canal de comunicação da organização para os participantes.

#### **6) - Ver lista dos posters**

User-story que lhe deu origem: Conseguir ver na aplicação os posters disponiveis na conferência.

Podemos ver a lista dos posters presentes na conferência, detalhes sobre o mesmo assim como descarregar o Poster e o Artigo que foi aprovado para que o Poster entre na conferência.

O Poster pode ser visto na lista dos Posters.

#### **7) – Trocar Contacto**

User-Story que lhe deu origem: Uma maneira fácil de trocar contacto com alguém.

Para resolver este problema acabamos por criar várias funcionalidades:

Podemos procurar por um user especifico numa conferência, podemos aceder ao perfil desse user e pedir uma troca de contacto. Se esse pedido for aceite podemos ver e exportar o contacto desse user.

Os pedidos são mostrados ao user por meio de notificações. Podemos também remover um contacto aceite.

É também possivel pedir um contacto de um autor se este for user da aplicação.

#### **Outras decisões / características:**

A nossa aplicação tem um nível minimo de segurança, Para aceder à aplicação temos de ter uma conta e para fazer download de material da conferencia, temos de estar registados na mesma e o email da conta verificado.

Para verificar o email, optamos pelo sistema de mandar um link de verificação para esse email.

É também possivel relacionar autores com users da conferência. Basta que o email de registo como autor seja o mesmo do user registado na conferencia. Isto permite que, quando acedemos ao perfil de um autor, possamos também aceder ao seu perfil de contacto, caso este seja um user da aplicação.



## 2.3 Restrições gerais

Esta aplicação foi concebida para Tablets com sistema Android versão 4. O tablet tem de ter acesso à Internet.

## 2.4 Pressupostos e dependências

**Internet:** Existe Wi-fi na conferência. Apesar da aplicação ter muitos dados offline, muitas das funcionalidades dependem da ligação ao servidor.

**Tablet pessoal:** Supõem-se que o tablet é pessoal. A aplicação não está preparada para receber logins de vários utilizadores.

## 3.0 Modelo do Domínio

### 3.1 Dicionário de Termos

**User:** Utilizador registado na aplicação.

**Block:** Relativo a uma sessão numa conferencia. Dentro de uma sessão podemos ter várias apresentações relativas ao mesmo tema. Demos o nome de Block porque no nosso servidor já existia uma classe chamada Session relative às sessões dos utilizadores logados.

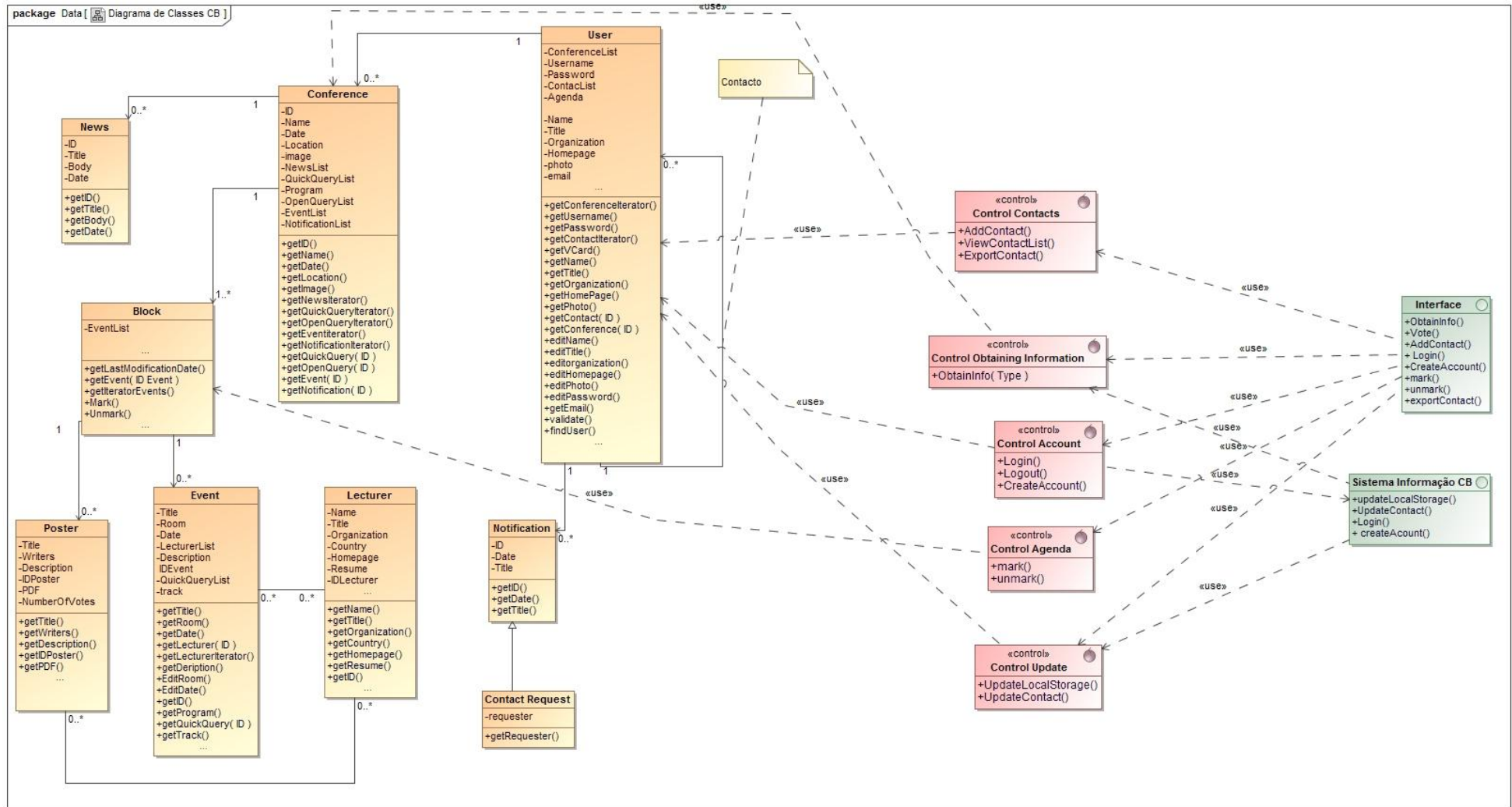
**Participante:** Todos os participantes registados na conferencia.

**Lecturer:** Refere-se aos autores de um paper ou poster. Chamam-se Lecturers pois queríamos colocar também quem vai apresentar o evento /poster, mas como não podemos saber isso de ante-mão, este passou a ser apenas o autor. Não mudamos o nome pois o projecto já estava muito avançado, teríamos de fazer uma revisão profunda do projecto para mudar todas as referências.

**Event:** Referente a uma apresentação/workshop/outro dentro de um block.

**Track:** Referente a um tema que abrange vários blocks.

## 3.2 Diagrama de Classes



## 4.0 Arquitectura e Desenho do Sistema

### 4.1 Arquitectura

#### Sistema Geral

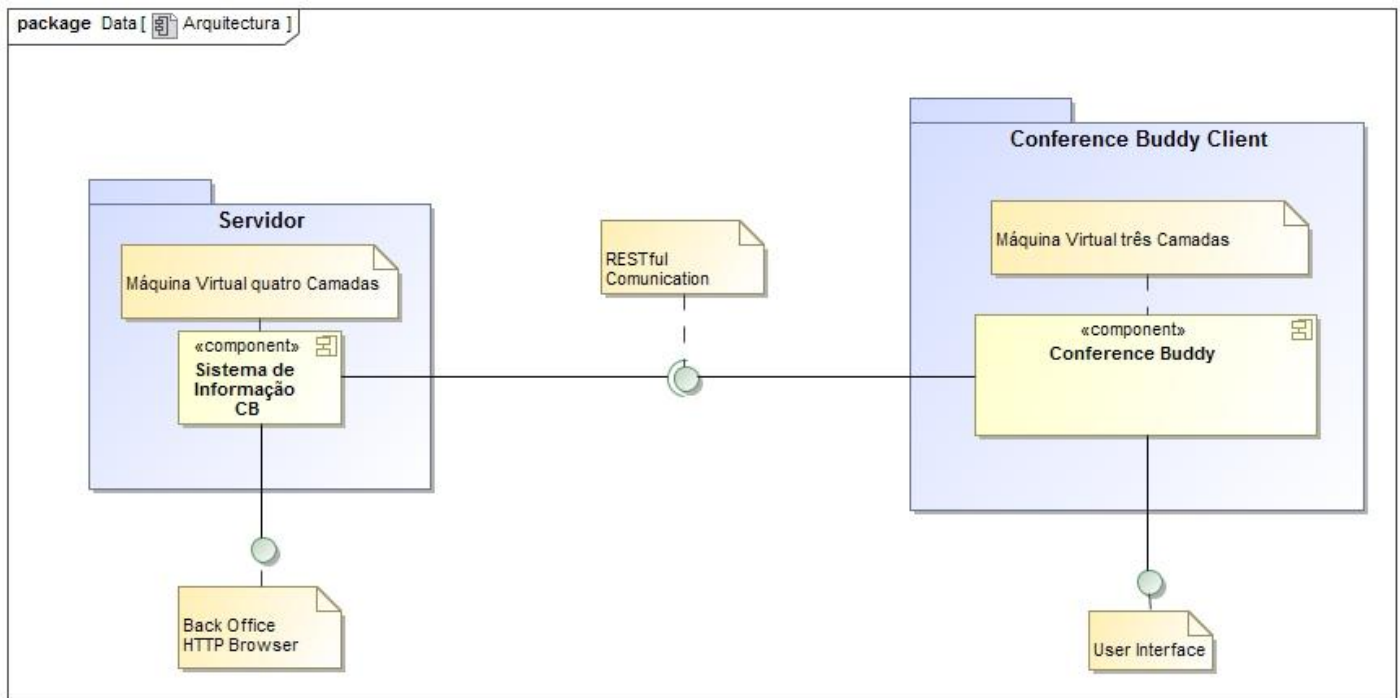


Figura 1

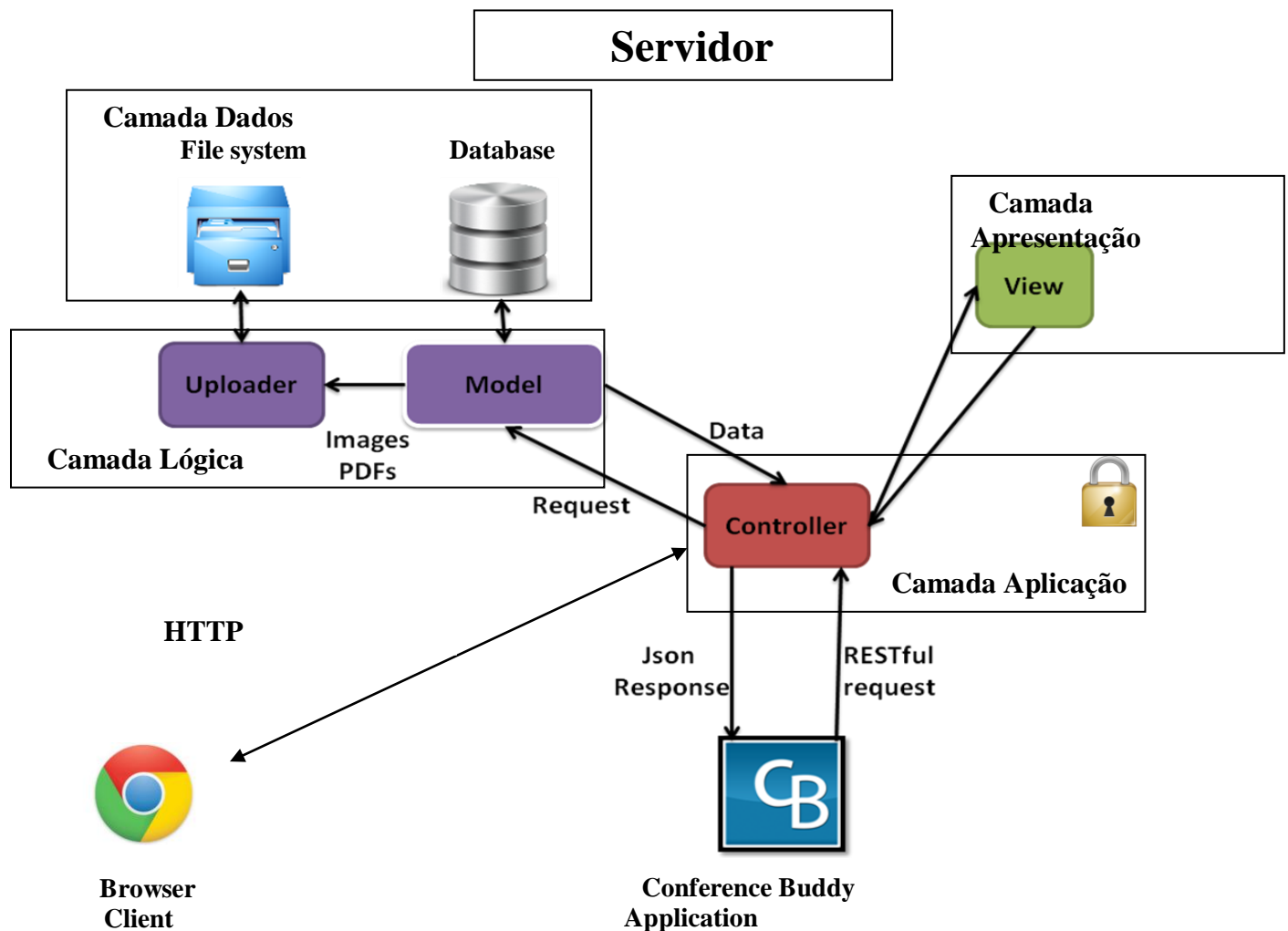
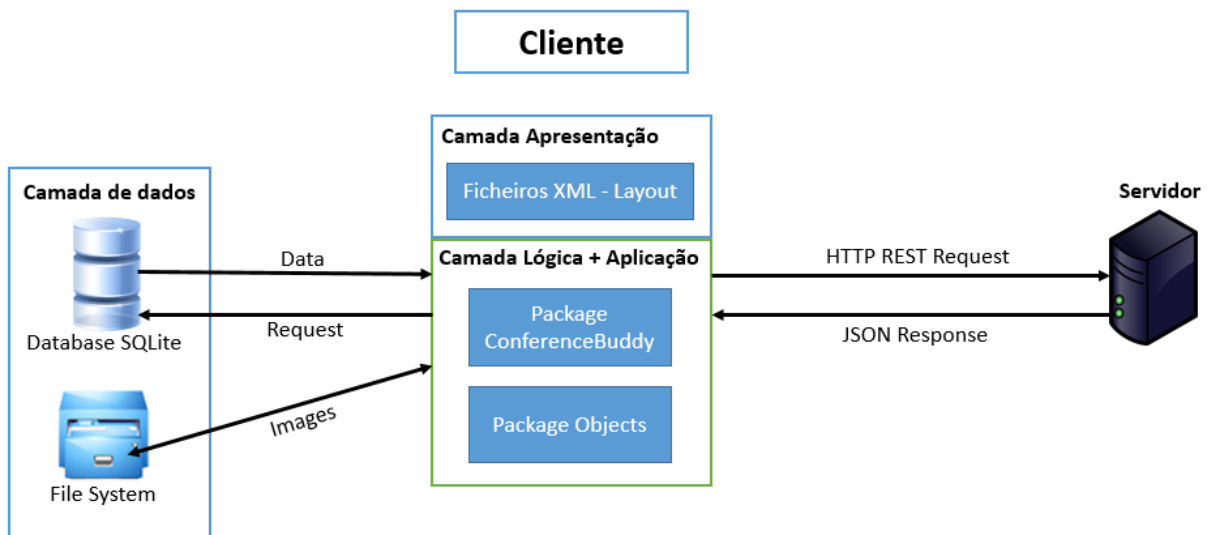


Figura 2

A **figura 2** representa a arquitectura do servidor. A aplicação comunica com o mesmo através de **pedidos REST** e obtém a **resposta** correspondente em **JSON**.

Para poder fazer pedidos, o user tem de estar logado na aplicação.

O **controller** controla a parte aplicacional do servidor, tendo os metodos definidos necessários para tratar e manipular toda a informação. A **View** contém os templates de resposta tanto das paginas HTML como das respostas em JSON. O **Model** faz a abstracção do acesso à base de dados e o **uploader** uma abstracção do acesso ao file system (necessário para guardar imagens e PDFs que foram carregados para o servidor).

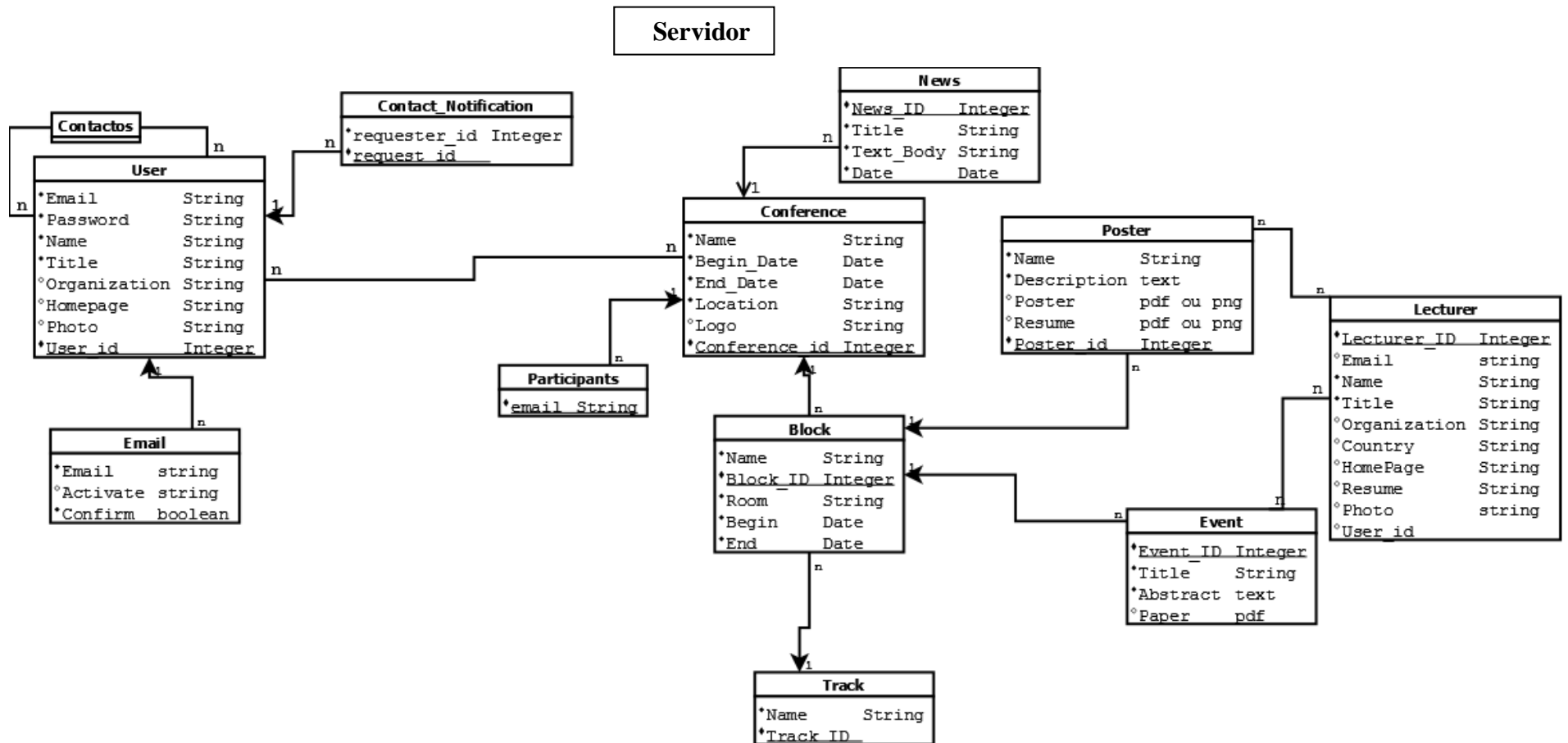


Neste diagrama verificamos a Arquitetura do Cliente. Este é composto pelas camadas de Apresentação que contém os ficheiros XML a partir dos quais o layout da aplicação é construído, e pelas camadas Lógica + Aplicação.

Nesta última camada é onde a aplicação trata de todo o processamento de dados, quer estes venham do servidor ou da base de dados local, e também do processamento do layout. Para comunicar com o servidor esta faz pedidos HTTP REST ao servidor aos quais este responde com respostas JSON. Esta camada trata estes pedidos e guarda localmente informações que estas respostas contenham.

A camada de dados é composta pela base de dados SQLite que contém toda a informação localmente, que foi descarregada do servidor e contém ainda uma pasta onde são guardadas todas as imagens que estejam relacionadas com conferências, *lectures* e contactos.

## 4.2 Modelo de Datos



**Tabelas [Servidor]:**

Muitas das tabelas são comuns ao cliente e ao servidor. Segue-se uma explicação breve das tabelas que aqui aparecem:

**User:** Users registados na aplicação. Para aceder à aplicação têm de fazer um login que requer o email e password registado neste User.

**Contact Notification:** Notificação de contactos; Quando um user pede um contacto a outro User, este ultimo fica com um pedido registado nesta tabela. Quando aceitar ou recusar, o registo é eliminado.

**Contacts:** Quando um pedido de contacto é aceite, é feito um registo nesta tabela.

**Conference:** Registo dos dados referentes às conferências disponiveis na Aplicação.

**News:** Registo dos dados referentes a cada noticia.

**Block:** Registo referente a cada Sessão numa conferência. Tivemos de dar o nome de block a esta tabela em vez de 'session', pois quando construímos o login no servidor, o Ruby on Rails já tinha o termo sessão relativo a cada login no servidor.

**Track:** Registo da Track referente a cada Sessão (Block).

**Event:** Os eventos associados a cada bloco como talks, workshops entre outros estão abstraidos nesta tabela.

**Poster:** É o unico "evento" que está considerado à parte dos restantes, pois necessita de um tratamento diferente. Está também referente a um bloco.

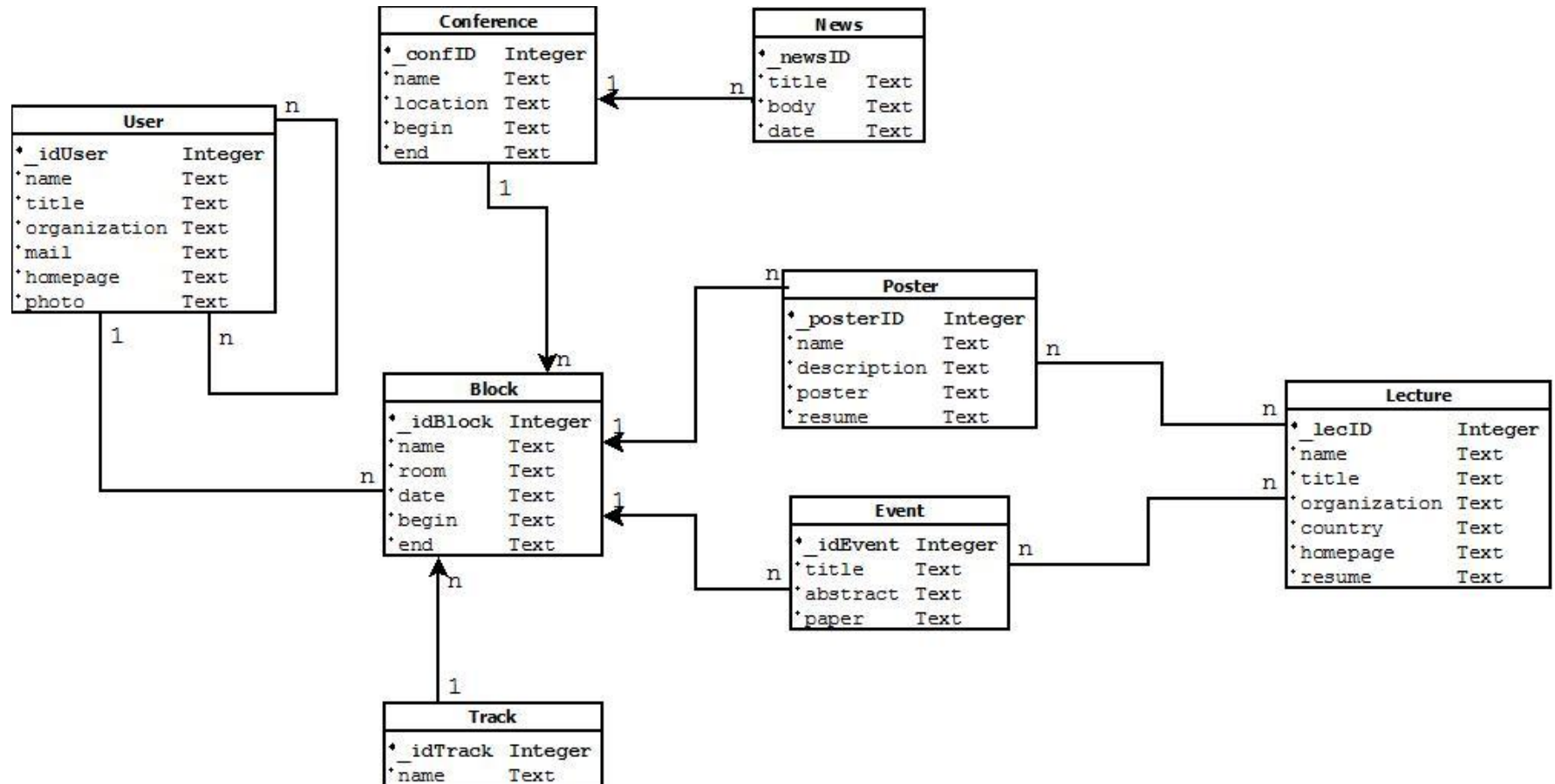
**Lecturer:** São os autores referentes a cada evento / poster.

**Email:** Lista de emails de cada User. Além do email de registo que serve de login, um user pode ter mais alguns mails que pode associar à conta.

**Participants:** Lista de participantes da conferência. Contém apenas o Email dos mesmos. Esta tabela é preenchida pela organização para que o servidor possa saber quem está registado na conferência e possa assim dar permissão de acesso a material reservado a quem está registado na conferencia.



Cliente



## **Tabelas [Cliente]**

**User:** Nesta tabela fica guardada a informação do utilizador da aplicação. A relação que tem consigo mesma de n para n representa os contactos do utilizador. Já a ligação que tem com a tabela Blocks representa a Agenda Pessoal que irá ter o \_idUser e o \_idBlock.

**Conference:** Aqui fica guardada a informação relativa a uma conferência que foi descarregada para a aplicação.

**News:** Informação acerca das notícias de uma conferência.

**Block:** Tabela referente à sessão de uma conferência. Aqui contém a hora de início, fim e a data dessa sessão.

**Track:** Tabela referente às tracks existentes.

**Event:** Tabela onde estão representados os eventos (workshops, keynotes, apresentação de papers) que preenchem os blocks.

**Poster:** Tabela onde estão representados os posters que existem numa conferência que também podem preencher os blocks

**Lecture:** Informação relativa a um autor de um poster e/ou paper.

---

## 5.0 Desenho e Interface

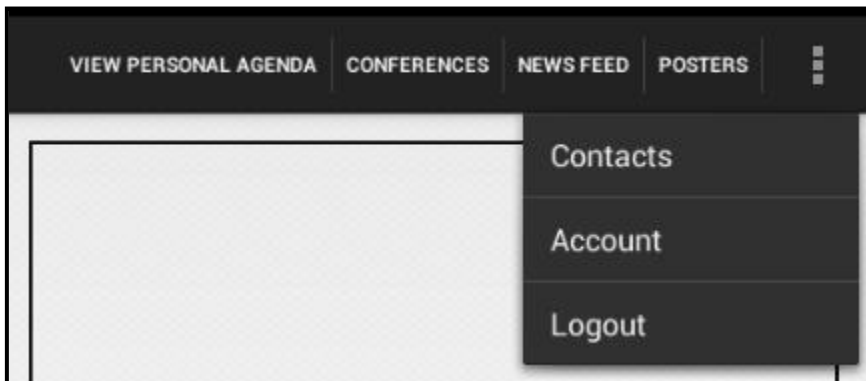
### 5.1 Filosofia Geral e Modalidades de Interação

A interface gráfica foi concebida com a prioridade principal da rapidez de acesso. As interações devem ser completadas com o mínimo de toques possíveis no ecrã para que o utilizador chegue à informação desejada rapidamente.

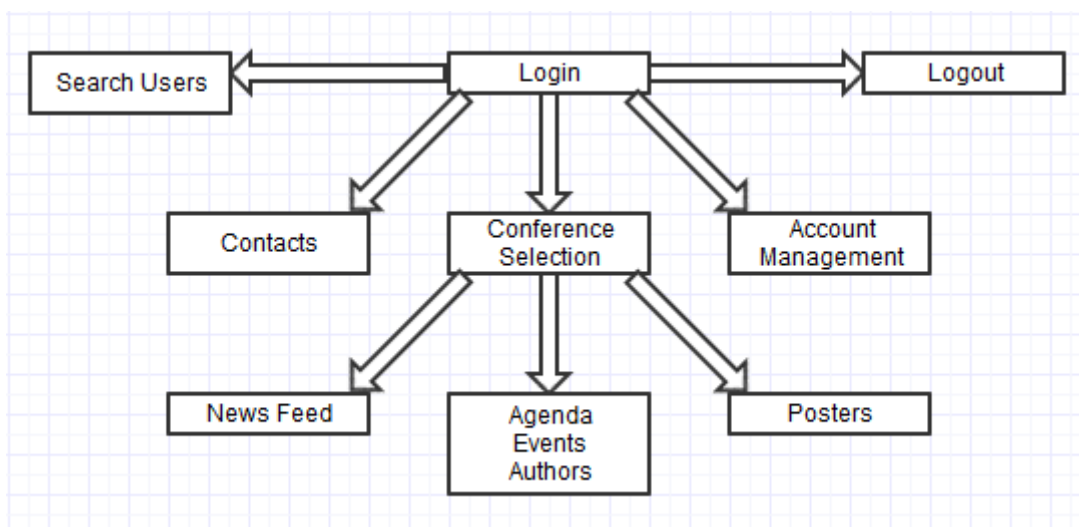
Depois de fazer login, apresenta-se o ecrã de escolha de conferência. Uma vez que algumas funcionalidades são gerais, como a lista de contactos, e outras estão integradas no contexto de uma conferência, como a agenda, o utilizador pode imediatamente aceder a funcionalidades gerais antes de seleccionar qualquer conferência, mas logicamente para visualizar uma agenda, ou uma lista de posters, terá de seleccionar a conferência respectiva.



Como podemos observar neste ecrã de selecção de conferências, as opções disponíveis na action bar (canto superior direito) são: Search Users, Contacts, Account e Logout. Após seleccionar uma conferência a action bar apresentará mais opções.



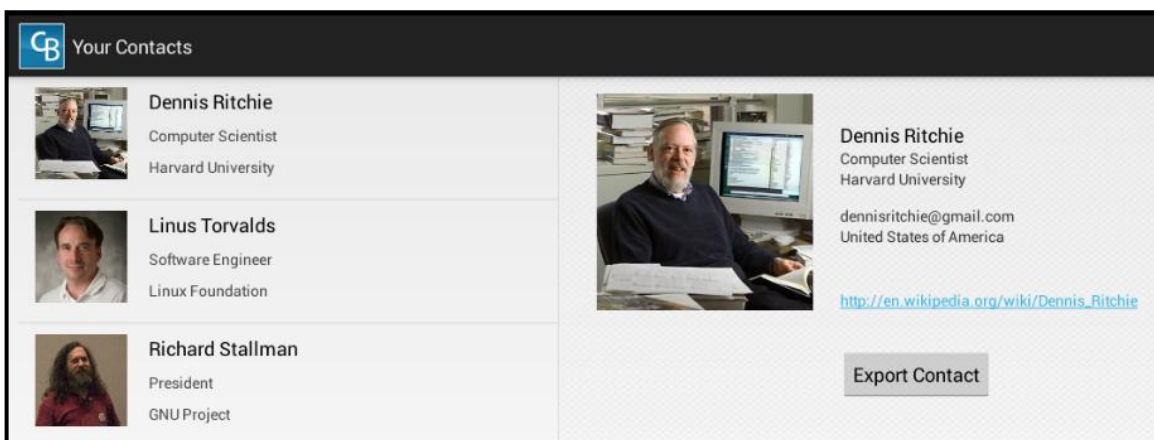
O diagrama abaixo apresenta a hierarquia da navegação dentro da aplicação.



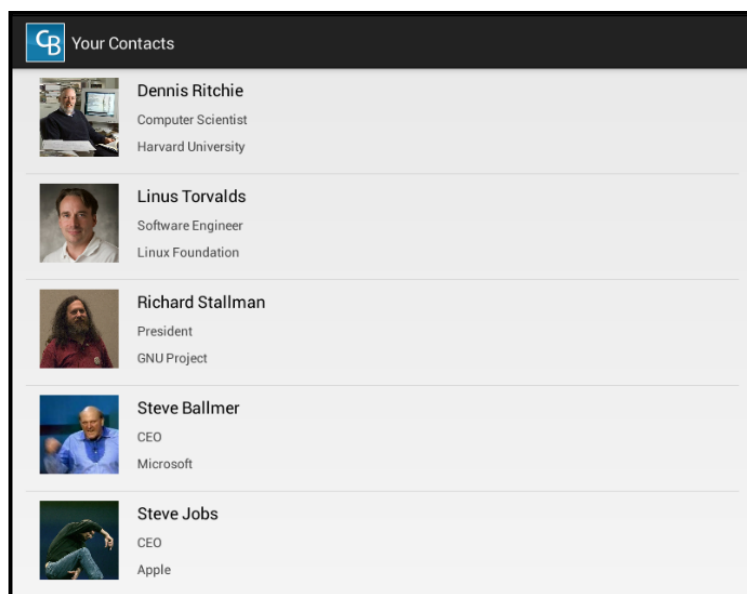
## 5.2 Aspectos Gerais da Interface Gráfica

Os ecrãs da aplicação, salvo algumas excepções, seguem uma estrutura geral composta essencialmente por 2 áreas de interacção. A área de listagem e a área de detalhe. Esta estrutura foi desenhada a pensar numa interacção em modo landscape, embora a aplicação também tenha um desenho específico para o modo portrait.

No caso da lista de contactos, por exemplo, temos à esquerda a lista, com uma thumbnail e nome do contacto. Do lado direito temos o detalhe do contacto, com todas as informações disponíveis.



Abaixo temos um screenshot da mesma lista de contactos, mas em modo portrait. A área de detalhe não é visível, e clicar num contacto dirige a aplicação para um novo ecrã, com os detalhes do contacto seleccionado, obrigando o utilizador a voltar atrás e seleccionar outro contacto, para ver um outro ecrã. O modo landscape é, portanto, o modo de operação preferencial.



### 5.3 Destaques



Fig 1

Na figura 1 vemos o ecrã da agenda de uma conferência (programa principal).

- 1 – Seleccionar dia
- 2 – Bloco de eventos
- 3 – Detalhe de um bloco (nenhum bloco seleccionado)

Ao entrar na agenda, o utilizador pode visualizar a informação de um bloco na zona de detalhe, bastando um clique sobre o respectivo bloco, ou imediatamente acrescentar um bloco à sua agenda pessoal, ou removê-lo, através de um clique prolongado (Fig 2). A aplicação informa o utilizador sobre o efeito da sua acção (Fig 3) e a zona de detalhe é actualizada com as informações do bloco (Fig 4).



Fig 2.

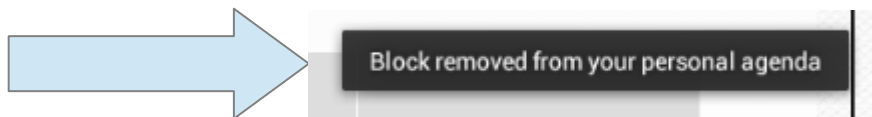


Fig 3.

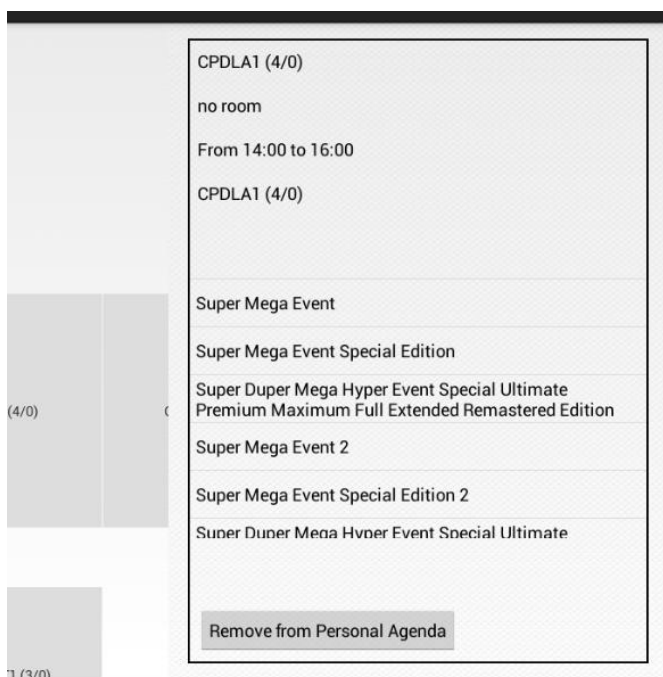


Fig 4.

Como podemos ver na figura 4, na zona de detalhe também existe um botão para adicionar/remover blocos de eventos, neste caso indicativo de que o evento se encontra na agenda pessoal e de que o podemos remover.

Neste screenshot a interface ainda se encontra pouco desenvolvida, e pouco amigável para o utilizador. O bloco seleccionado tem o nome CPDLA1, e por cima do botão “Remove from Personal Agenda” podemos ver a sua lista de eventos, cada um com o seu detalhe próprio. Clicamos num dos eventos e somos levados para o ecrã de detalhe de eventos.

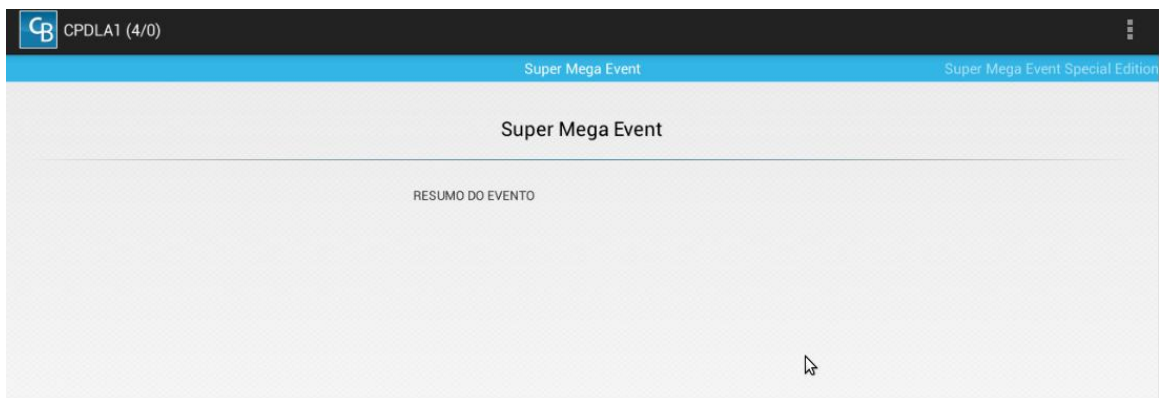
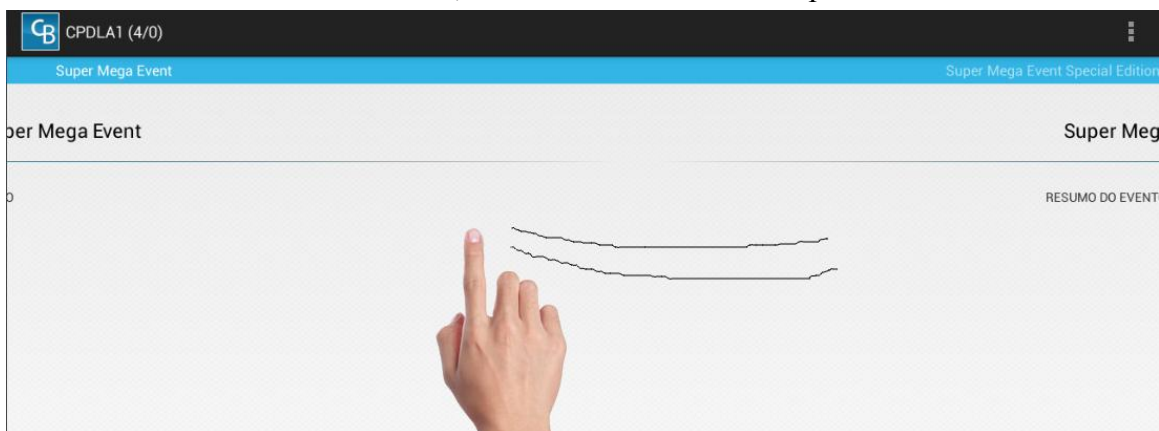


Fig 5.

Neste screenshot também pouco desenvolvido, representativo do ecrã de detalhe dos eventos de um bloco, visualizamos o evento “Super Mega Event”. Por uma questão de conveniência para o utilizador, é possível ver a informação de todos os outros eventos do bloco CPDLA1 sem sair deste ecrã, com um movimento “swipe” ilustrado abaixo.





## 6.0 API RESTful

A API REST do nosso projecto funciona do seguinte modo:

O cliente executa um pedido HTTP ao servidor, pedido esse que pode ser GET, POST ou PUT, levando se necessário informação em formato JSON.

O servidor processa o pedido e a resposta vem também em HTTP, com o *status code* correspondente ao resultado do processamento do pedido (ok, bad request, not found, etc) e se necessário com informação no formato JSON.

A comunicação é feita apenas com pedidos do lado do cliente e respostas do lado do servidor, nunca o contrário.

Em anexo podemos visalzar a API a que o servidor responde.

## 7.0 NFRs e Validação

### 7.1 NFRs

#### - Usabilidade

Para este requisito não funcional ser satisfeito procurámos criar uma aplicação que fosse fácil de usar e de rápida aprendizagem. Para garantirmos isso preocupámo-nos em criar uma interface bastante simples, em que a interação é à base de *touch* e *slide*.

Durante os testes, em que os utilizadores usaram a aplicação, estes não tiveram qualquer dificuldade em ambientar-se à mesma, destacando-se a forma fácil de criar a sua agenda pessoal fosse através do *touch* longo ou do clicar no botão de adicionar um evento à agenda pessoal.

#### - Integridade

Para este requisito não funcional ser satisfeito criámos um sistema em que é obrigatório ter uma conta de aplicação, verificada por e-mail. Assim quem não tiver uma conta não poderá aceder às informações das conferências nem de todos os outros utilizadores.

Para testar este requisito procurámos entrar na aplicação sem conta, o que foi sempre rejeitado pela aplicação que obrigava que fizéssemos sempre login.

#### - Multi Acesso

Para este requisito não funcional ser satisfeito criámos uma base de dados local para guardar informações sobre as conferências e sobre os contactos do utilizador. Desta maneira evita-se a sobrecarga do servidor e permite que toda gente veja a informação que quer e quando quer sem qualquer constrangimento.

## Apêndices

### Application Programming Interface (API) – Servidor

#### -- Perfil --

**Ação:** registar utilizador

**URL:** http://[IP]:3000/users.json

**Método:** POST

**Content-Type:** application/json

**Data example:**

```
{ "email": "nralves@fct.unl.pt", "password": "1234", "password_confirmation": "1234",  
  "name": "Nuno", "title": "Mr.", "organization": "FCT", "homepage": "nuno.fct.unl.pt",  
  "photo": "nuno.bmp" }
```

**Ação:** Login

**URL:** http:// [IP]:3000/sessions.json

**Método:** POST

**Content-Type:** application/json

**Data example:**

```
{ "login": " nralves @ fct.unl.pt", "password": "1234" }
```

**Ação:** Obter Perfil (requere login efectuado com sucesso)

**URL:** http:// [IP]:3000/user/1.json

**Método:** GET

#### -- Conferencia --

**Ação:** Obter Noticias Recentes

**URL:** http:// [IP]:3000/getnews.json

**Método:**POST

**Content-Type:** application/json

**Data exemple:** (Ultima data de noticias recebidas | ID da conferencia)

```
{"date":"2013-06-01 16:36:30" "id":"1"}
```

**Acção:** download da conferencia

**URL:**http://[IP]3000/conferencedownload.json

**Método:**POST

**Content-Type:** application/json

**Data exemple:** (ID da conferencia)

```
{"id":"1"}
```

**Acção:** Obter dados de todas as conferencias disponiveis

**URL:**http://[IP]:3000/getallconferences.json

**Método:**GET

**Acção:**Obter todos os lecturers

**URL:**http://[IP]:3000/lecturers.json

**Método:**GET

**Acção:**download do poster

**URL:** http://[IP]:3000/download\_poster

**Data exemple (ID do Poster):**

```
{"id":"2"}
```

**Acção:**download do paper

**URL:**http://[IP]:3000/download\_paper

**Data exemple (ID do paper):**

```
{"id":"2"}
```

## -- Contacts --

**Acção:** Obter requests

**URL:** http://[IP]:3000/ getrequests

**Método:** GET

**Acção:** Rejeitar pedido de contacto

**URL:** http://[IP]:3000/rejectrequest.json

**Método:** POST

**Content-Type:** application/json

**Data exemple (ID do pedido):**

```
{"id": "1"}
```

**Acção:** Aceitar pedido de contacto

**URL:** http://[IP]:3000/friendships.json

**Método:** POST

**Content-Type:** application/json

**Data exemple (ID do utilizador que fez o pedido):**

```
{"friend_id": "1"}
```

**Acção:** Fazer pedido de contacto

**URL:** http://[IP]:3000/requests.json

**Método:** POST

**Content-Type:** application/json

**Data exemple (ID do utilizador de quem queremos o contacto):**

```
{"user_id": "1"}
```

**Acção:** Remover Contacto

**URL:** http://[IP]:3000/removefriend.json

**Método:** POST

**Content-Type:** application/json

**Data exemple** (ID do utilizador de quem queremos remover o contacto):

```
{"friend_id": "1"}
```