

# IST Tickets

## App Android



João Bravo (84390)

**Instituto Superior Técnico**

*Mestrado Integrado em Engenharia Física Tecnológica*

*LID - Prof. Luís Viseu Melo*

*Junho 2018*

---

### Resumo

*Neste relatório, introduzem-se alguns conceitos básicos do desenvolvimento de aplicações para Android, para melhor compreensão do restante conteúdo. Explica-se também sucintamente o motivo da escolha deste projeto e o funcionamento da aplicação desenvolvida, IST Tickets. Finalmente, detalham-se algumas nuances do código desenvolvido.*

### CONTEÚDO

I	Definições e Funções Básicas	1
II	Motivo	2
III	Funcionamento da app	2
IV	Alguns detalhes do código desenvolvido	4
	Referências	5

---

### I. DEFINIÇÕES E FUNÇÕES BÁSICAS

- **Atividade:** cada ecrã, com a sua interface de utilizador (*UI*) específica, que interage com o utilizador.
- **Layout:** estrutura da *UI* de cada atividade.
- **View:** objeto gráfico do layout, como um botão ou uma caixa de diálogo.
- **Android Studio:** IDE da *Google* utilizada para desenvolver e compilar a app.
- **Manifest:** ficheiro *.xml* que descreve os componentes da app, nomeadamente, o pacote Java usado, as permissões requeridas e a hierarquia de atividades.
- **Ficheiros *.java*:** implementam todas as classes, objetos e métodos usados, as atividades da app e as reações associadas a cada ação do utilizador.
- **Ficheiros *.xml*:** inúmeras funções, tal como implementar a hierarquia de atividades, o *layout* de cada atividade, caixa de diálogo e menu, e definir as cores, estilos e *Strings* usadas na app.
- **Ficheiro *.apk*:** ficheiro compilado com todo código e imagens usadas, pronto a instalar num sistema *Android* (depois do processo de alinhamento e assinatura digital com um certificado, já efetuados na versão anexada).
- **Web API:** neste contexto, refere-se à interface que permite o acesso a dados e recursos de um serviço *web*, usando o protocolo *HTTP* (ou seja, *urls*).
- **API endpoints:** extremos de um canal de comunicação de um serviço usados para comunicar com a API, representados como *urls* neste caso.

## II. MOTIVO

Este projeto foi escolhido com o intuito de criar uma ferramenta que permitisse verificar, *online* e em tempo real, qual a senha a ser correntemente atendida em cada serviço da secretaria do IST. Desta forma, os alunos e (não) docentes não teriam de esperar na fila, facilitando o processo de atendimento e melhorando o fluxo nos corredores dos edifícios do IST. Atualmente, o *IST Tickets* já incorpora informação sobre os serviços do Núcleo de Mobilidade e Cooperação Internacional e da secretaria do campus Taguspark, aglomerando assim todos os serviços em que está totalmente implementado o sistema digital de senhas, bem como informação sobre os horários e contactos dos mesmos. Optou-se ainda por desenhar a app em inglês, de forma a possibilitar a estudantes estrangeiros o uso da mesma.

## III. FUNCIONAMENTO DA APP

A app *IST Tickets* divide-se em 4 ecrãs, ou atividades. Ao abrir a app, clicando no ícone desenvolvido de raiz no *Photoshop CS6* para se assemelhar ao do IST, surge a atividade *Main* (figura 1). A partir desta, é possível consultar informações sobre os serviços do IST já implementados. O botão *Schedules* permite consultar os horários de todos os núcleos num só lugar (figura 2). O botão *About* permite visualizar algumas informações sobre a origem da app, bem como os contactos de todos os núcleos (figura 3). Já os três botões centrais (*Academic Office*, *Mobility & International Office* e *TagusPark Office*) permitem consultar informações em tempo real sobre os serviços de cada núcleo.

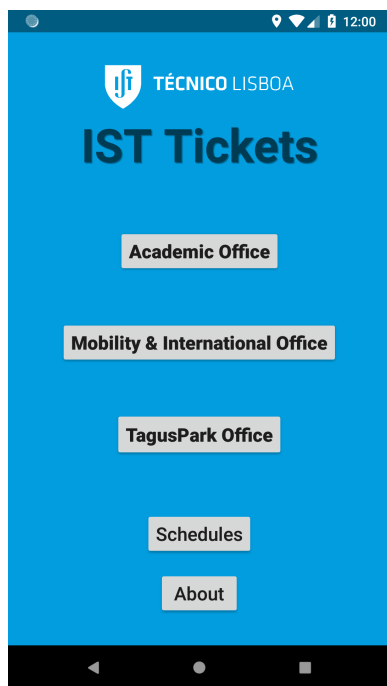


Figura 1: Atividade Main

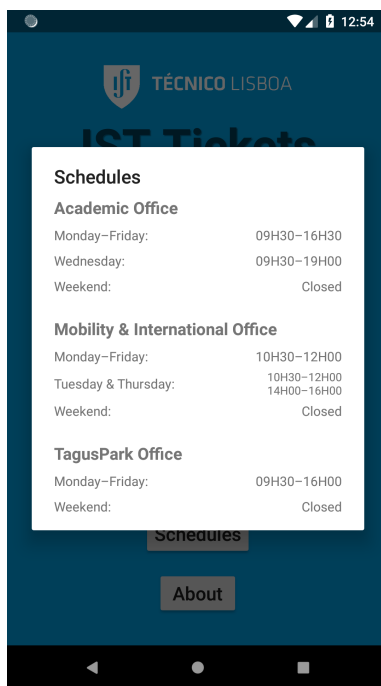


Figura 2: Caixa de diálogo com horários dos núcleos

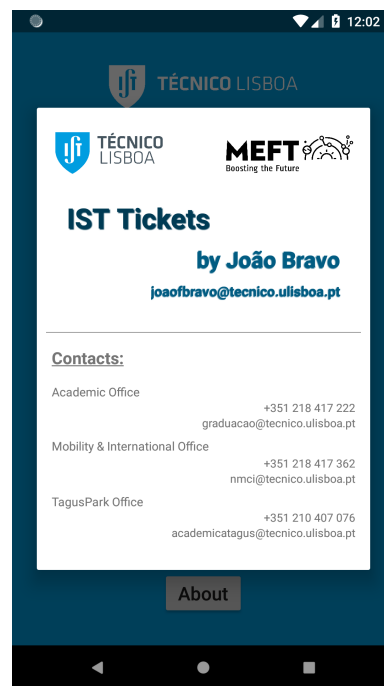
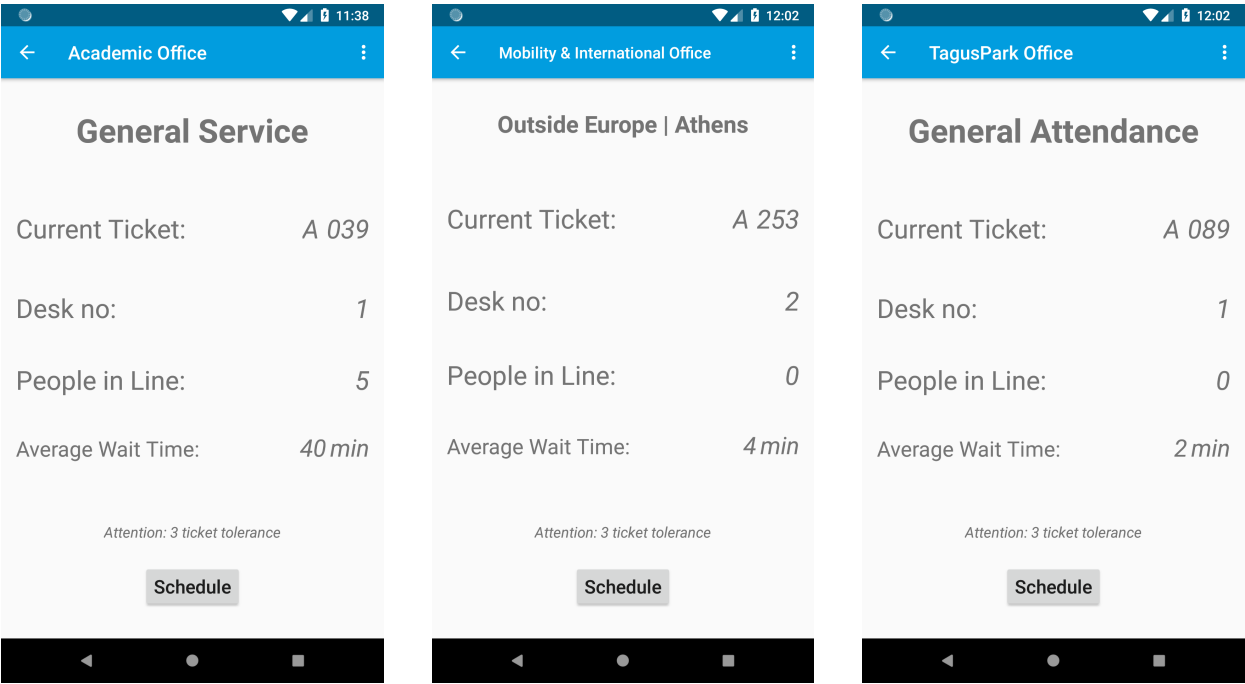
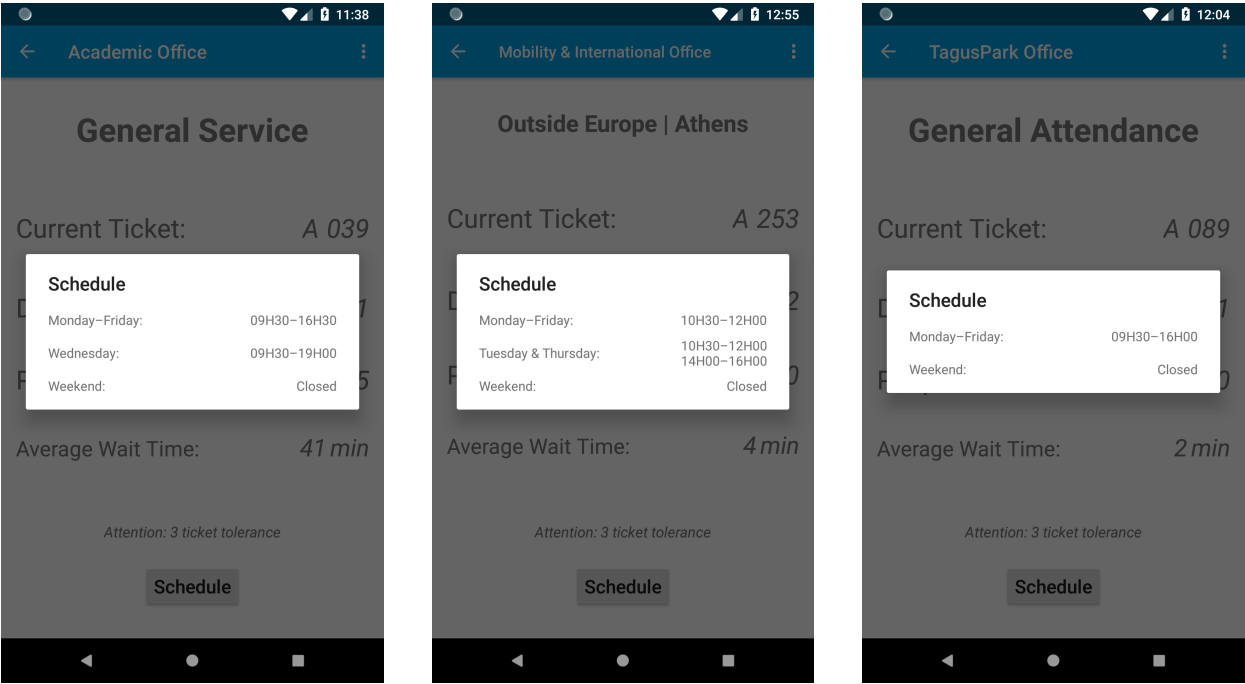


Figura 3: Caixa de diálogo About

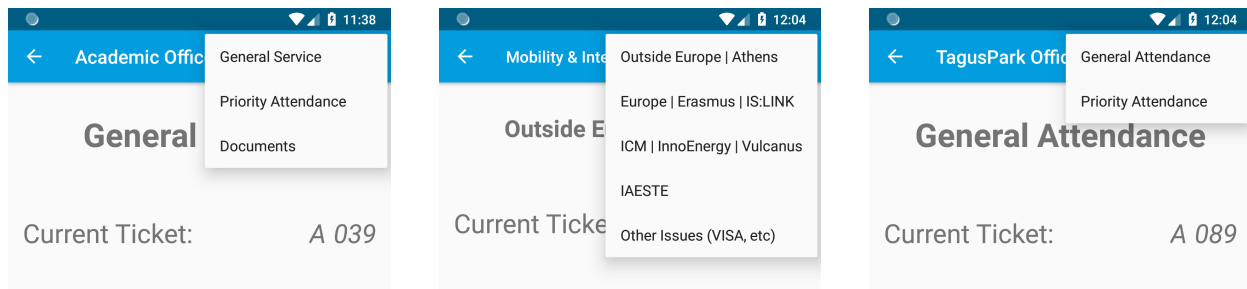
Selecionando um destes três botões, surge a atividade correspondente ao mesmo núcleo (figuras 4). Nesta é possível consultar a senha a ser correntemente atendida, o número do balcão, o número de pessoas em espera nesse serviço e o tempo médio de atendimento do serviço, tendo como referência os registos do próprio dia. Há ainda um aviso em todas as atividades para a tolerância de 3 senhas de atraso, e a possibilidade de consultar o horário do núcleo específico no botão *Schedule* (figuras 5). Estas informações estão disponíveis para todos os serviços de cada núcleo, bastando selecionar o serviço desejado no menu de opções, no canto superior direito da atividade (figuras 6).



**Figura 4:** Atividades de cada núcleo. Por ordem: Secretaria do campus Alameda, Núcleo de Mobilidade e Cooperação Internacional, Secretaria do campus Taguspark

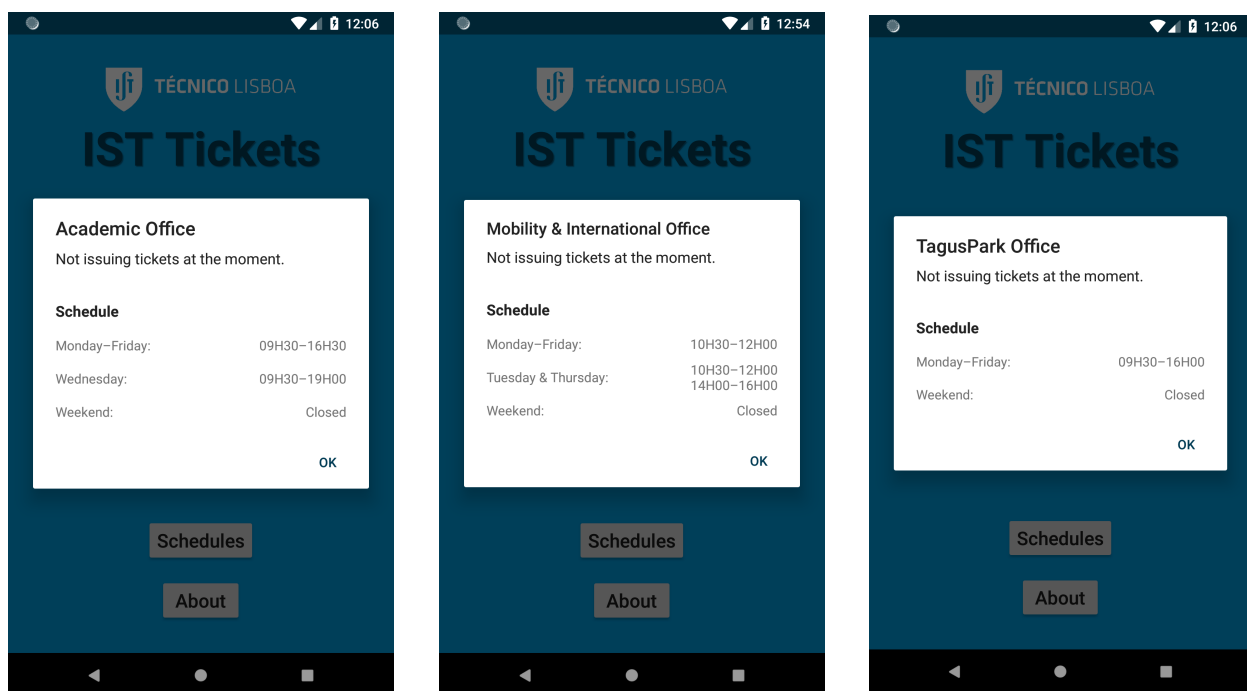


**Figura 5:** Horário dos serviços de cada núcleo. Por ordem: Secretaria do campus Alameda, Núcleo de Mobilidade e Cooperação Internacional, Secretaria do campus Taguspark



**Figura 6:** Menu de serviços de cada núcleo. Por ordem: Secretaria do campus Alameda, Núcleo de Mobilidade e Cooperação Internacional, Secretaria do campus Taguspark

Se os serviços não estiverem ativos, caso da consulta se dar fora do horário do núcleo ou de haver algum imprevisto, é exibida uma caixa de diálogo, ao invés da nova atividade, com a informação de que os serviços não estão ativos e a fornecer senhas, conjuntamente com o horário do mesmo serviço (figuras 7).



**Figura 7:** Caixas de diálogos exibidas durante a inatividade de um núcleo, incluindo o respetivo horário

#### IV. ALGUNS DETALHES DO CÓDIGO DESENVOLVIDO

Nesta secção, explicitam-se alguns detalhes e nuances do funcionamento da app, em função do código desenvolvido e das escolhas feitas nesse processo. Restantes comentários relativos ao código e respetivo funcionamento podem ser encontrados de forma detalhada e exhaustiva nos próprios ficheiros em anexo. Inclui-se ainda um ficheiro *.apk* compilado, alinhado e com assinatura digital, estando assim pronto a instalar e usar.

Note-se primeiramente a grande adaptabilidade da app. Foram tomadas várias medidas para garantir esta característica. Todos os *layouts* são do tipo *Constraint Layout*, que permite posicionar e dimensionar cada *View* relativamente a outras *Views* e ao tamanho e densidade de *pixels* de cada ecrã. Para o mesmo efeito adaptativo, agruparam-se *Views* em cadeia (*Spread Chain*), de forma a manter o espaço entre elas proporcional às dimensões do ecrã. A app possui as imagens nela presentes em várias resoluções, usando automaticamente as mais corretas para as dimensões e densidade do ecrã. Utilizou-se também a unidade *sp* (*Scale-independent Pixels*) para as *Strings* usadas, sendo esta a mais adaptável a várias densidades de *pixels* e ao tamanho de fonte escolhido por cada utilizador. No entanto, o sistema *Android* aparenta apresentar alguns problemas no dimensionamento de texto para ecrãs muito grandes, como *tablets* de 8"/9", não sendo capaz de aumentar suficientemente o tamanho da letra. A app é no entanto perfeitamente utilizável, sendo esta apenas uma questão estética. Todas estas escolhas de *layout* podem ser encontradas no ficheiros *.xml* e *.java* anexados.

Optou-se ainda por utilizar a *View Toolbar* ao invés da predefinida, *ActionBar*, como barra de ações das atividades dos núcleos, por ser uma generalização desta última, permitindo alterar o texto nela presente, bem como o seu tamanho e até adicionar *Views*, tornando-a muito mais customizável. Para isso, definiu-se o tema *AppThemeNoActionBar* em *styles.xml*, que remove a *ActionBar* predefinida e definiu-se a *Toolbar* nos ficheiros *.java* de cada atividade como barra de ações padrão.

O *debugging* de cada atividade pode ser facilmente feito, dando o valor *true* à variável booleana *DEBUG* do ficheiro *.java* da respetiva atividade, e com o método *LogcatDebug* criado. Para a versão final (a do *.apk* em anexo, deu-se obviamente o valor *false* a *DEBUG*, de forma a acelerar a resposta da app.

A variável *taskDelay* presente nos *.java* de cada atividade representa o intervalo de verificação dos *endpoints* da *Web API* do IST e atualização de *flags* e das *Views* da *UI*. Optou-se por definir este intervalo como 10 segundos na atualização da *UI* das atividades de cada serviço, e 5 minutos na atualização das *flags* (*servicesIssuing*) da atividade *Main*, respeitantes à (in)atividade de cada núcleo, por este ser um estado estável num prazo de tempo relativamente grande.

Finalmente, verificou-se que, mesmo quando os serviços estão ativos, por vezes a *Web API* do IST não apresenta a informação totalmente atualizada, fornecendo campos *null* para cada dado. Nestas situações esporádicas, através de blocos *try & catch* de *exceptions* presentes nos *.java*, os campos afetados apresentam o sinal *None*. Isto dá-se também antes da primeira senha diária ser emitida.

## REFERÊNCIAS

- [1] *User Guide to Android Studio and App Developing*, by Google, Maio 2018  
<https://developer.android.com/studio>