



**Ciências
ULisboa**

Faculdade
de Ciências
da Universidade
de Lisboa

Design de Software Relatório do 2º Projeto - IoT

Guilherme Guimarães - 46375
Catarina Guerreiro - 46426
Mário Teixeira - 47074

Introdução:

Este projeto tem como objetivo a concepção e implementação duma linha de produtos de software e a utilização de diferentes mecanismos na gestão de variabilidade. Esta fase será apenas um protótipo para esta linha de produtos, por isso é importante tomar boas decisões e deixar a documentação necessária para futuro trabalho na nossa linha de aplicações IoT.

O AspectJ será usado para oferecer ao utilizador uma variedade de produtos na aplicação devido ao Aspect Oriented Programming. O AOP (Aspect Oriented Programming) permite a definição de aspetos, de forma programática, explícita e separada de outros elementos disponíveis.

O Middleware Bezirk é utilizado para tratar da comunicação com os dispositivos de hardware, apesar de nesta fase ainda não ser preciso lidar com a incorporação de dispositivos de hardware concretos, vamos, no nosso código ter classes que simulam estes dispositivos. O Bezirk utiliza uma arquitetura Publish/Subscribe o que possibilita adicionar e remover componentes sem limitações e também permite a cada componente publicar/subscrever eventos que ao circular no event bus permitirão despoletar ações noutros componentes.

Decisões de Implementação:

No nosso projeto, para a persistência dos dados, decidimos usar quatro ficheiros que irão armazenar os dados e guardá-los, para que não seja necessário voltar a introduzir esses dados cada vez que utilizarmos o protótipo. Os ficheiros servem para guardar os contactos do utilizador, guardar informações sobre avisos, guardar padrões de atividade e guardar padrões de inatividade. Os ficheiros têm os seguintes nomes respetivos: contactos.txt, avisos.txt, atividades.txt e inatividades.txt..

Os produtos que decidimos criar são:

- Produto com interface em inglês para cegos com monitor de atividade;
- Produto com interface em inglês para cegos com funcionalidade botão;
- Produto com interface em inglês para surdos com monitor de atividade;
- Produto com interface em inglês para surdos com funcionalidade botão.

Para gerar as variações de alertas que são despoletados após o carregamento de botão ou uma atividade/inatividade são utilizados aspetos. Estas variações de alertas foram diferenciadas da seguinte forma:

- Para os utilizadores cegos é utilizado um sintetizador de voz.
- Para os utilizadores surdos é utilizado um sistema de luzes inteligentes, que na verdade são simuladas no output, pois nesta fase ainda não recorreremos a hardware.

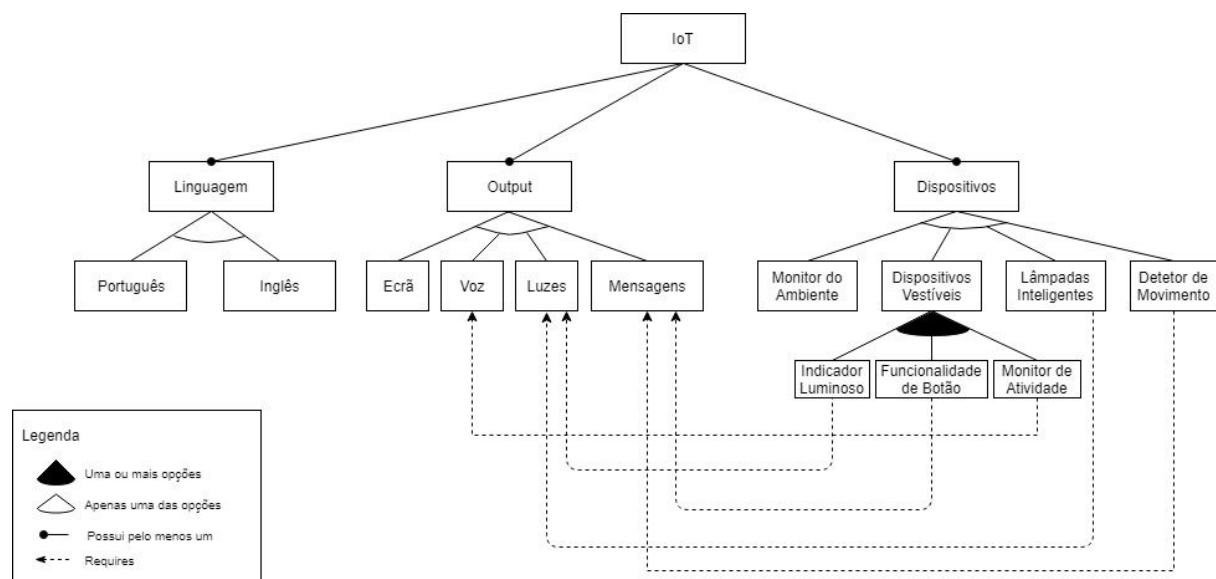
Quando um utilizador cego utiliza o sistema, em caso de atividade o sintetizador de voz irá emitir a mensagem "Alert! An activity was detected.", em caso de inatividade o sintetizador de voz irá emitir a mensagem "Alert! It's been a while since I saw you moving" e em caso de carregamento de botão o sistema envia uma mensagem de socorro a todos os contactos do utilizador.

Quando um utilizador surdo utiliza o sistema, em caso de atividade uma luz verde será acesa, em caso de inatividade uma luz vermelha será acesa e em caso de carregamento de botão uma mensagem de socorro será enviada para todos os contactos do utilizador e uma luz amarela será acesa para dar feedback de mensagem enviada ao utilizador.

Features Module:

Como está acima descrito, o nosso protótipo oferece a possibilidade de escolha entre ter a interface em português ou inglês. O output depende de cada produto, sendo que para utilizadores cegos o output é feito através de voz, para os utilizadores surdos o output é feito através de indicadores luminosos e através do ecrã para utilizadores sem deficiência e surdos. O produtos permitem também o envio de mensagens para todos os contactos do utilizador e possuem pelo menos um dispositivo, que pode ser:

- Monitor de ambiente: No caso do dispositivo ser um monitor de ambiente requer o ecrã como forma de output.
- Dispositivo Vestível: No caso de ser um dispositivo vestível este poderá ser composto por uma ou mais componentes (indicador luminoso, funcionalidade botão e monitor de atividade).
- Lâmpadas inteligentes: No caso do dispositivo ser lâmpadas inteligentes requer que uma das formas de output seja através de luzes.
- Detetor de Movimento: No caso do dispositivo ser um detetor de movimento requer o ecrã como uma das formas de output.



Quadro com os produtos da linha:

	Linguagem		Dispositivo Vestível com...					
	Port.	Ing.	Indicador Luminoso	Funcionalidade Botão	Monitor de Atividade	Lâmpadas Inteligentes	Detetor de Movimento	Monitor Ambiente
Prod 1 (cegos)		X			X			
Prod 2 (cegos)		X		X				
Prod 3 (surdos)		X	X		X			
Prod 4 (surdos)		X	X	X				

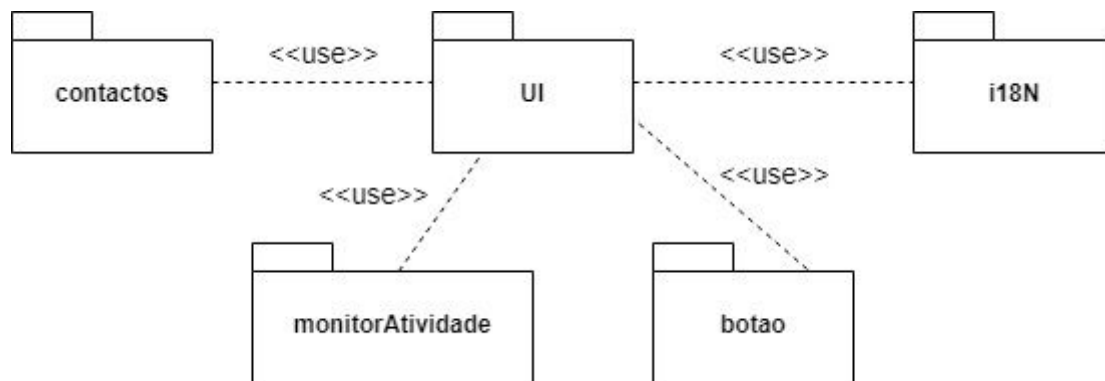
Vista de módulos

No package `monitorAtividade` estão todas as classes referentes à persistência de atividades e inatividades nos ficheiros de texto e as classes que representam que representam os sensores de atividades/inatividades e os eventos de atividade/inatividade.

No package `botao` estão todas as classes que representam a funcionalidade botão, incluindo as classes que representam o aviso que é criado e guardado ao carregar no botão.

No package `contactos` encontram-se todas as classes que representam os contactos do utilizador, nomeadamente a persistência dos contactos no ficheiro de texto.

O package `i18N` contém as mensagens em português e inglês e os ficheiros usados para fazer a conversão para a língua seleccionada, de acordo com as propriedades carregadas.



Vista C&C

Os dispositivos externos representados na imagem (Botão, Monitor de Atividade e Indicador Luminoso) publicam eventos no event bus através do Bezirk Middleware. Estes eventos são tratados recorrendo ao AspectJ.

A nossa interface do utilizador (UI) acede às nossas quatro base de dados (em forma de ficheiros txt), que guardam padrões de atividade e inatividade, avisos e contactos.

