

Lab 5: Vistas de arquitetura

Análise de Sistemas

Universidade de Aveiro

Diana Rocha 98524, Diogo Correia 90327,
Gonçalo Maranhão 73150, João Fernandes 93460

Grupo 105

2021/2022

Departamento de Eletrónica, Telecomunicações e Informática



Conteúdo

1	5.1 a)	2
2	5.1 b)	3
3	5.1 c)	4
4	5.2 a)	5
5	5.2 b)	6
6	5.3 a)	8
7	5.3 b)	9

5.1 a)

Esta aplicação apresenta duas componentes que estão diretamente ligadas à apresentação de conteúdo, o BroadcastEngine que apresenta conteúdo no "feed" do utilizador e o BlogViewer, que é o próprio utilizador, que vê esse conteúdo.

Ambas componentes requerem o uso da componente ConversionManagement, que dá sentido e organiza os dados que devem ser apresentados no blog.

Esta componente, por sua vez, requer uma componente que fornece esses dados, que neste caso é o BlogDataSource.

Para que haja uma monitorização do que acontece dentro da base de dados, é requerida uma componente, Log4j, para fazer, então, o tal "logging" das ações.

5.1 b)

Log4j é uma biblioteca em Java especializada em fazer logging de ações num software. Isto é, ao registar o que acontece durante o fluxo de uma aplicação, permite facilitar bastante no processo de debugging quando algo corre mal.

5.1 c)

Para identificar este componente, teríamos de colocar o seguinte trecho de código no ficheiro **build.gradle**.

```
dependencies {  
    implementation 'org.apache.logging.log4j:log4j-api:2.17.2'  
    implementation 'org.apache.logging.log4j:log4j-core:2.17.2'  
}
```

5.2 a)

Arquitetura multicamada de um sistema operativo iOS, encontrada no seguinte site:
<https://www.dotnettricks.com/learn/xamarin/understanding-xamarin-ios-build-native-ios-app>

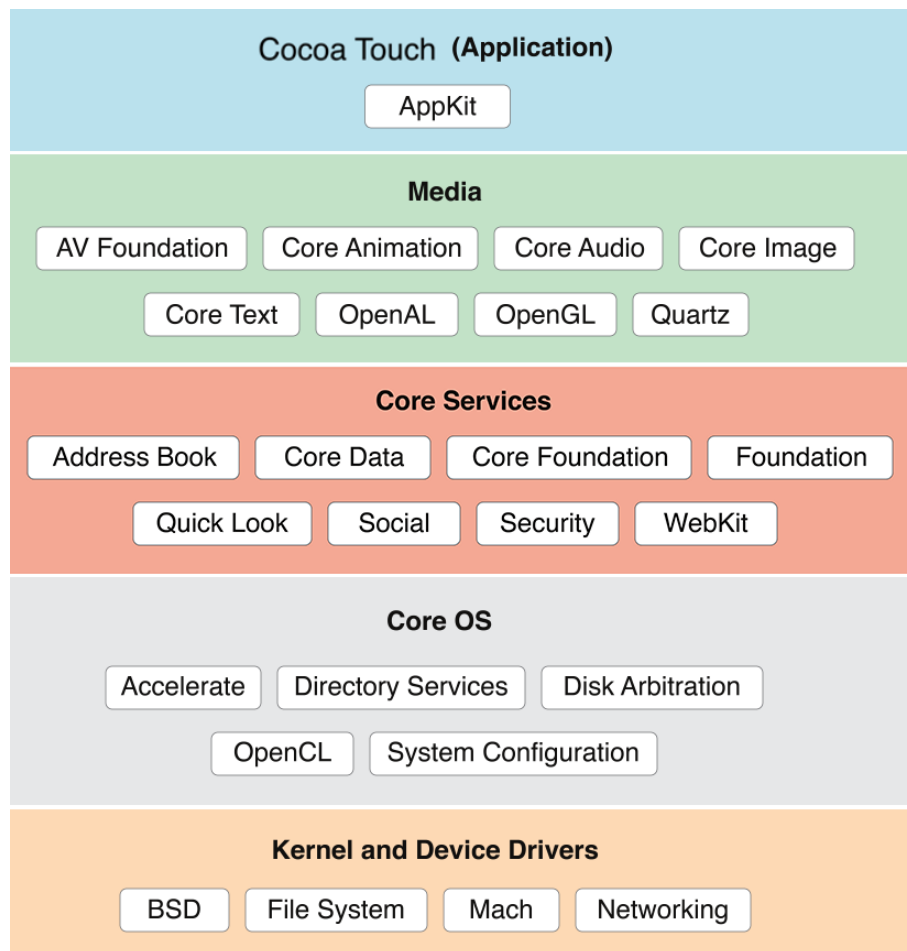


Figura 4.1: Arquitetura multicamadas de um sistema iOS

5.2 b)

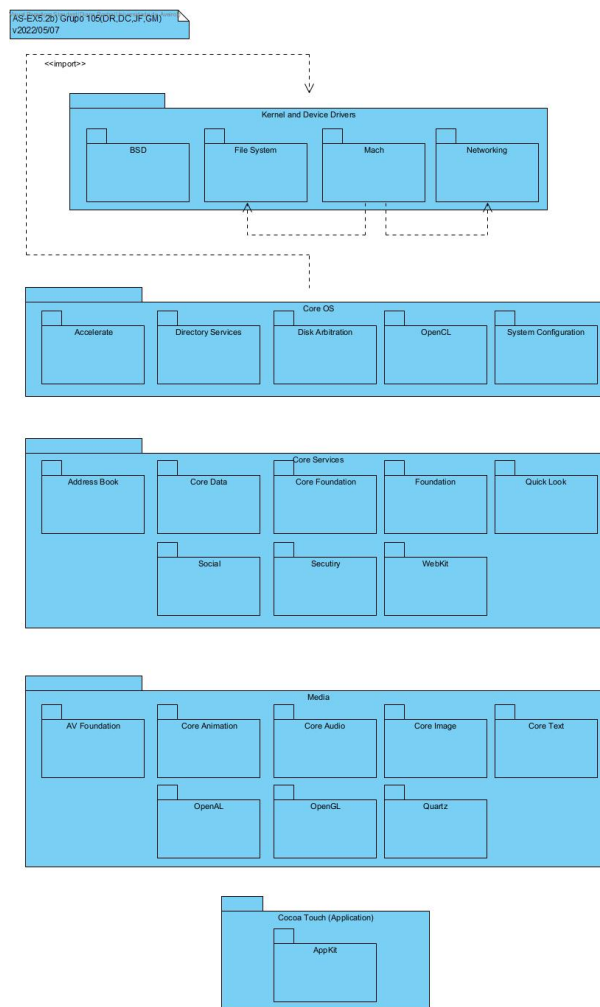


Figura 5.1: Diagrama de pacotes da Arquitetura multicamadas de um sistema iOS

Kernel and Device Drivers: Na base desta arquitetura encontra-se o Kernel, que é o programa que faz a ponte entre o hardware do dispositivo e tudo o resto que compõe o sistema operativo, e os Device Drivers que criam compatibilidade com dispositivos periféricos. O Kernel, dentro de outras funcionalidades, implementa um sistema de ficheiros e protocolos de comunicação de rede.

Core OS: Nesta camada encontram-se tecnologias e estruturas que implementam serviços que derivam diretamente da camada anterior. O que se encontra nesta camada apresenta ainda um nível baixo de abstração. Aqui são usadas frameworks como:

- OpenCL, bom para desenvolver programas que interajam com estruturas de baixo nível, como, por exemplo, o Kernel
- Accelerate, que usa inteligência artificial para aumentar a performance do sistema e tornar o sistema energético mais eficiente
- Disk Arbitration, para controlar a entrada e saída de periféricos

Core Services: Esta camada apresenta serviços que fornecem informação e features que são essenciais para o funcionamento das aplicações de alto nível. Algumas das frameworks usadas neste nível de abstração são:

- Foundation/Core Foundation, fornece funcionalidades de backend às aplicações, como armazenamento de dados, networking, interações com data e hora, etc...
- Security, tal como o próprio nome indica, fornece controlos de acesso a dados e software
- Core Data, serve para guardar o estado das aplicações localmente, para, por exemplo, poderem ser usadas no modo offline

Media: Esta camada trata de otimizações gráficas, animações e efeitos visuais que ocorrem nas aplicações. Tem, por isso, variadas frameworks e aplicações que tratam do processamento de imagem e áudio.

Cocoa Touch: Ambiente de desenvolvimento para criação de aplicações que vão correr no sistema operativo iOS. É a camada de mais alto nível da arquitetura do sistema e é onde se encontra a AppKit, que é a biblioteca com todos os objetos necessários para desenvolver a interface de uma aplicação.

5.3 a)

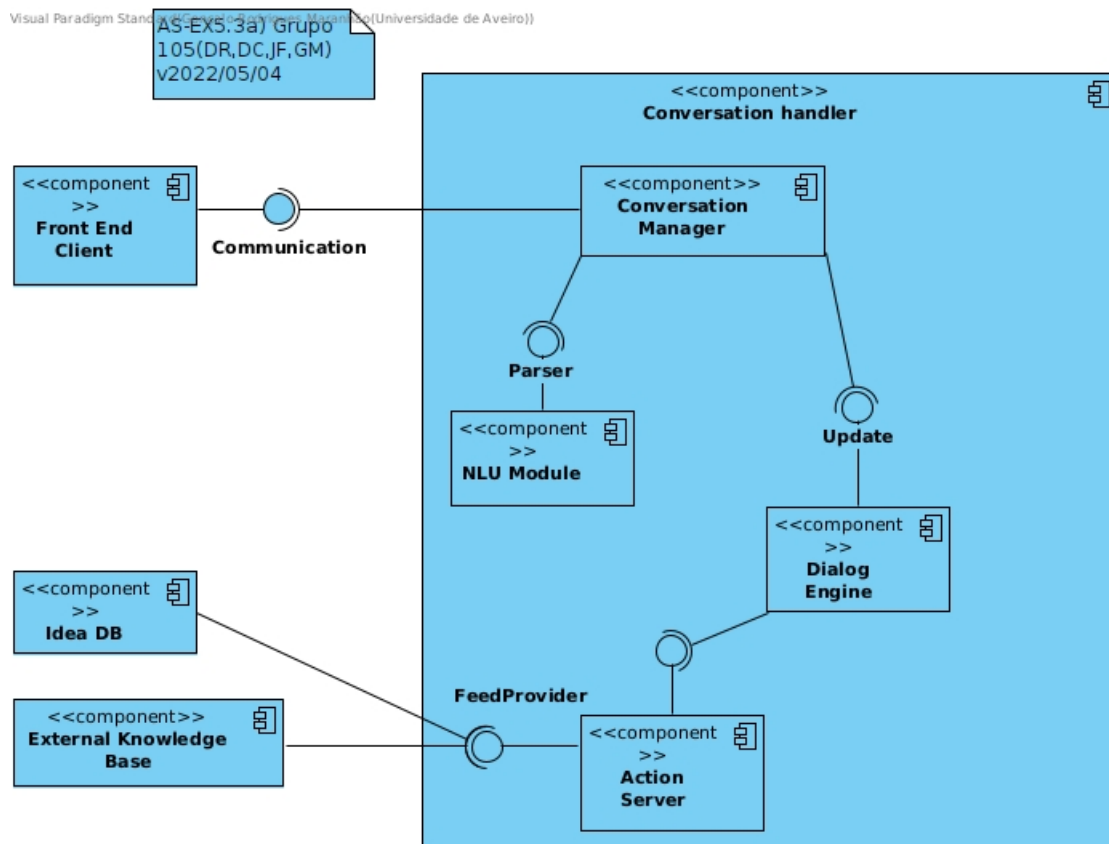


Figura 6.1: Diagrama de Componentes

5.3 b)

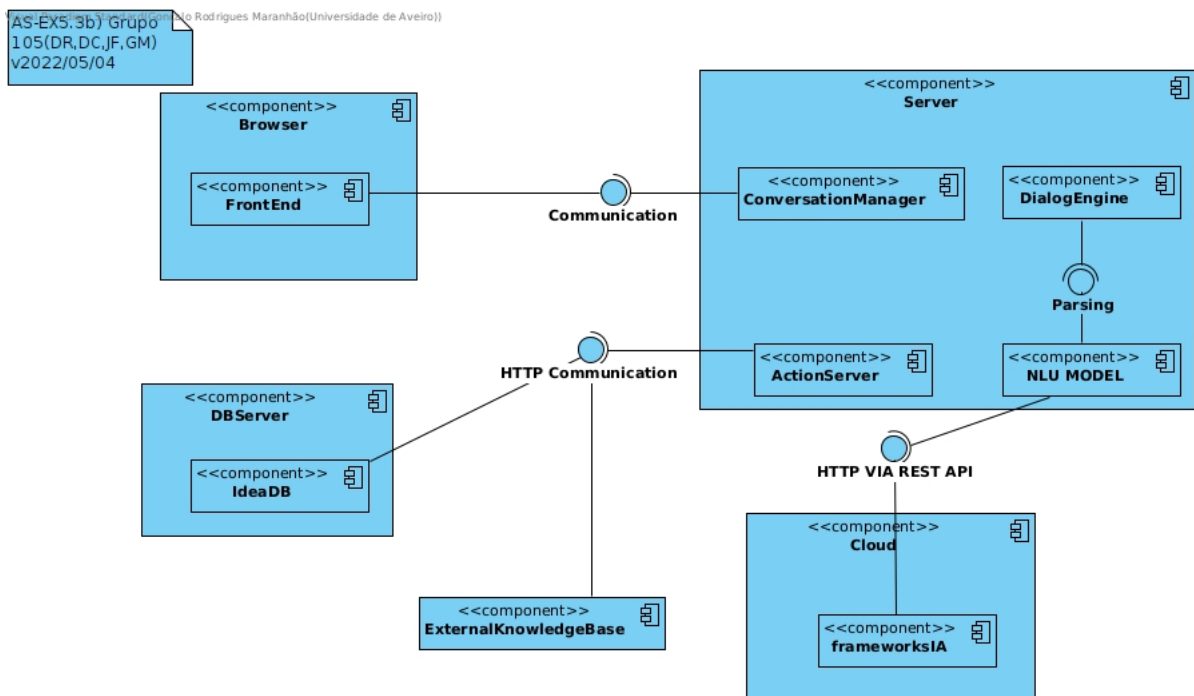


Figura 7.1: Diagrama de Componentes