



**INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA**

**Área Departamental de Engenharia de Electrónica e Telecomunicações e de Computadores**



## **Rain Computing: Orquestração de ações executadas por grupos de dispositivos IoT**

**António Peixoto de Assunção Santos**

(Licenciado em Engenharia Informática e de Computadores)

Projecto Final para obtenção do Grau de Mestre  
em Engenharia Informática e de Computadores

Orientador : Doutor Luís Manuel da Costa Assunção

Júri:

Presidente: Doutor Nuno Miguel Machado Cruz

Vogais: Doutor Carlos Jorge de Sousa Gonçalves  
Doutor Luís Manuel da Costa Assunção

DEZ, 2020





**INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA**

**Área Departamental de Engenharia de Electrónica e Telecomunicações e de Computadores**



## **Rain Computing: Orquestração de ações executadas por grupos de dispositivos IoT**

**António Peixoto de Assunção Santos**

(Licenciado em Engenharia Informática e de Computadores)

Projecto Final para obtenção do Grau de Mestre  
em Engenharia Informática e de Computadores

Orientador : Doutor Luís Manuel da Costa Assunção

Júri:

Presidente: Doutor Nuno Miguel Machado Cruz

Vogais: Doutor Carlos Jorge de Sousa Gonçalves  
Doutor Luís Manuel da Costa Assunção

DEZ, 2020



*Aos meus pais, à minha esposa Eugénia, aos meus  
filhos Júlio e Raul, aos meus irmãos.  
A todos os professores e colegas que contribuíram  
para que o começo desta última caminhada fosse  
possível, em especial ao Professor Doutor Luís  
Assunção que me acompanhou nas últimas etapas.  
Aos meus amigos.*



# **Agradecimentos**

Um profundo agradecimento ao Professor Doutor Luís Assunção que me conduziu nesta etapa final do curso e que, desde a primeira hora, me motivou por forma a ultrapassar os desafios encontrados, permitindo a conclusão deste trabalho.



# Resumo

No mundo da *Internet of Things* (IoT) surgiram soluções tecnológicas de sistemas para armazenar e processar na *Cloud* grandes volumes de dados originados em IoT.

Os dispositivos IoT utilizam comunicações sem fios criando a necessidade de níveis intermédios (*fog computing*) para agregar e filtrar os dados antes de serem enviados para a *Cloud*. Para além de gerar dados, por exemplo, valores de temperatura, de sinais vitais como o ritmo cardíaco, alguns dispositivos podem também executar ações.

Contrariamente ao fluxo de dados Dispositivos-*Cloud*, é muito importante considerar cenários *Cloud*-Dispositivos onde os dados são enviados da *Cloud* para os dispositivos, que podem ser utilizados por pessoas, durante a execução de qualquer atividade. Consequentemente identifica-se um problema - como coordenar as ações que cada pessoa deve executar, ao receber dados no seu dispositivo IoT.

Esta tese propõe como solução o sistema *Rain Computing* como analogia com a chuva que cai das nuvens e inunda rios e lagos. Assim, dados previamente definidos na *Cloud* (planos de ação) são enviados para inundar dispositivos, com intermediação de coordenadores, que sincronizam as ações executadas em cada dispositivo.

A arquitetura do sistema *Rain Computing* assenta em módulos desacoplados, com interfaces e mensagens de comunicação bem definidas: i) O Centro de Controlo executa-se na *Cloud* e permite aos utilizadores definir e agendar a execução de planos de ação em múltiplos dispositivos; ii) O módulo intermédio Coordenador executa-se num servidor ao nível *fog computing* e sincroniza a execução das ações do plano nos dispositivos; iii) Adicionalmente também é disponibiliza uma biblioteca de suporte ao desenvolvimento de aplicações em múltiplos cenários.

São também apresentados dois estudos de caso, o *Maestro Digital* e o *Luzes de Palco* que permitiram avaliar e validar a operacionalidade do sistema *Rain Computing* para o desenvolvimento de múltiplos cenários de controlo de dispositivos IoT.

**Palavras-chave:** Cloud computing, Fog computing, Internet of Things, Coordenação Distribuída.



# Abstract

In the world of *Internet of Things* (IoT), technological solutions naturally emerged with systems based on large volumes of data originating from IoT devices and processed in the *Cloud*.

IoT devices use wireless communications creating the need for intermediate levels (*fog computing*) to aggregated and filtered before being sent to the *Cloud*. In addition to generating data, for example, temperature values, vital signs such as heart rate, some devices can also perform actions.

Contrary to the data flow *Devices-Cloud*, it is very important to consider scenarios *Cloud-Devices* where data is sent from *Cloud* to devices, which can be used by people, during the execution of any activity, consequently a problem is identified - how to coordinate the actions that each person must perform, when receiving data on his IoT device.

This thesis proposes as a solution the *Rain Computing* system as an analogy with the rain that falls from the clouds and floods rivers and lakes. Thus, data previously defined in the *Cloud* (action plans) are sent to flood devices, with the intermediation of coordinators, who synchronize the actions performed on each device..

The *Rain Computing* system architecture is based on decoupled modules, with well-defined communication messages and interfaces: i) The Control Center module runs on *Cloud* and allows users to define and schedule the execution of action plans on multiple devices; ii) The intermediate module Coordinator runs on a server at the *fog computing* level and synchronizes the execution of the plan's actions on the devices; iii) In addition, a library to support application development in multiple scenarios is also available.

Two case studies are presented, *Maestro Digital* and *Luzes de Palco*, which allowed to evaluate and validate the operability of the *Rain Computing* system for the development of multiple control scenarios for IoT devices.

**Keywords:** Cloud computing, Fog computing, Internet of Things, Distributed Coordination.



# Índice

<b>Lista de Figuras</b>	<b>xvii</b>
<b>Lista de Tabelas</b>	<b>xix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Cloud/Fog Computing e dispositivos IoT . . . . .	1
1.2 Contexto tecnológico . . . . .	2
1.3 Questões em aberto . . . . .	3
1.4 Motivação . . . . .	3
1.5 Definição do problema . . . . .	4
1.6 Organização do documento . . . . .	7
<b>2 Trabalho Relacionado</b>	<b>9</b>
2.1 Agendamento ( <i>scheduling</i> ) de tarefas em dispositivos IoT . . . . .	9
2.2 Controlador de dispositivos IoT com decisão em tempo real . . . . .	10
2.3 Coordenação de aplicações IoT para cidades inteligentes . . . . .	11
2.4 Coordenação de dispositivos inteligentes em missão . . . . .	12
<b>3 Proposta de Solução</b>	<b>13</b>
3.1 Especificação de requisitos . . . . .	13
3.1.1 Requisitos funcionais . . . . .	14
3.1.2 Requisitos não funcionais . . . . .	14
3.2 Modelo de interação entre os módulos envolvidos . . . . .	15
3.3 Modelo de adesão ao sistema <i>Rain Computing</i> . . . . .	17

3.4	Arquitetura lógica . . . . .	18
3.4.1	Representação do plano de execução . . . . .	18
3.4.2	Fluxos de comunicação . . . . .	24
3.5	Sumário . . . . .	34
<b>4</b>	<b>Implementação do Sistema <i>Rain Computing</i></b>	<b>35</b>
4.1	Arquitetura da implementação . . . . .	36
4.1.1	Arquitetura do centro de controlo . . . . .	37
4.1.2	Arquitetura do coordenador . . . . .	46
4.1.3	Biblioteca <i>Rain</i> . . . . .	51
4.2	Disponibilidade e sincronismo dos dispositivos e coordenadores . . . . .	56
4.3	Exemplo de um cenário de troca de mensagens . . . . .	57
4.4	Sumário . . . . .	58
<b>5</b>	<b>Operacionalidade do Sistema <i>Rain Computing</i></b>	<b>61</b>
5.1	Instalação e configuração . . . . .	61
5.1.1	Centro de controlo . . . . .	61
5.1.2	Coordenador . . . . .	62
5.1.3	Dispositivos . . . . .	62
5.2	Validação e estudos de caso . . . . .	62
5.2.1	<i>Maestro Digital</i> . . . . .	63
5.2.2	<i>Luzes de Palco</i> . . . . .	71
5.2.3	Avaliação dos resultados . . . . .	79
<b>6</b>	<b>Conclusões e Trabalho Futuro</b>	<b>81</b>
6.1	Conclusões . . . . .	81
6.2	Trabalho futuro . . . . .	83
	<b>Referências</b>	<b>85</b>
<b>A</b>	<b>Maestro Digital</b>	<b>i</b>
A.1	Plano de execução . . . . .	i
A.2	Registos de testes . . . . .	xii
A.2.1	Centro de controlo . . . . .	xii
A.2.2	Coordenador . . . . .	xv
A.2.3	Dispositivos . . . . .	xviii

**B Luzes de Palco** **xxi**

B.1 Plano de execução . . . . . xxi

B.2 Registos de testes . . . . . xxxi

B.2.1 Centro de controlo . . . . . xxxii

B.2.2 Coordenador . . . . . xxxii

B.2.3 Dispositivos . . . . . xxxii



# Lista de Figuras

3.1	Desenho do modelo de processamento de um plano . . . . .	16
3.2	Desenho do modelo de adesão ao sistema . . . . .	17
3.3	Arquitetura Lógica Global . . . . .	19
3.4	Partes do plano de execução . . . . .	19
3.5	Fluxo Global de Comunicação . . . . .	24
4.1	Arquitetura geral da implementação do sistema <i>Rain Computing</i> . . .	36
4.2	Arquitetura do centro de controlo . . . . .	37
4.3	Tópicos Pub/Sub do centro de controlo . . . . .	39
4.4	Modelo de dados do centro de controlo . . . . .	40
4.5	Permitir a criação de plano no centro de controlo . . . . .	41
4.6	Editar atributo <i>dados</i> . . . . .	42
4.7	Permitir o agendamento da execução de planos . . . . .	42
4.8	Agendar a execução de um plano . . . . .	43
4.9	Arquitetura do coordenador . . . . .	46
4.10	Modelo de dados do coordenador . . . . .	47
4.11	Tópicos Pub/Sub do coordenador . . . . .	48
4.12	Configuração geral do coordenador . . . . .	50
4.13	Gestão de dispositivos no coordenador . . . . .	50
4.14	Composição da biblioteca <i>Rain</i> . . . . .	51
4.15	Utilização da biblioteca <i>Rain</i> . . . . .	53
4.16	Exemplo de troca de mensagens <i>Rain</i> . . . . .	57

5.1	<i>Maestro Digital</i> : Exemplo de troca de mensagens . . . . .	69
5.2	<i>Maestro Digital</i> : Mensagens do estudo de caso . . . . .	70
5.3	<i>Luzes de Palco</i> : Exemplo de troca de mensagens . . . . .	77
5.4	<i>Luzes de Palco</i> : Mensagens do estudo de caso . . . . .	78

# Lista de Tabelas

3.1	Identificação do plano . . . . .	20
3.2	Responsáveis pelo plano . . . . .	21
3.3	Papéis envolvidos no plano . . . . .	21
3.4	Contextos do plano . . . . .	22
3.5	Ordens (ações) envolvidas na execução do plano . . . . .	23
3.6	Cabeçalho de Mensagem. . . . .	25
3.7	Mensagem de erro . . . . .	26
3.8	Mensagem registrar . . . . .	27
3.9	Mensagem descobrir . . . . .	28
3.10	Mensagem aderir . . . . .	29
3.11	Mensagem convidar . . . . .	30
3.12	Mensagem retirar . . . . .	30
3.13	Mensagem planear . . . . .	31
3.14	Mensagem de contexto . . . . .	31
3.15	Mensagem ordenar . . . . .	32
3.16	Mensagem sincronizar . . . . .	33
5.1	Maestro: Identificação do plano . . . . .	64
5.2	Maestro: Responsáveis pelo plano . . . . .	64
5.3	Maestro: Papéis envolvidos no plano . . . . .	65
5.4	Maestro: <i>Contextos</i> do plano . . . . .	66
5.5	Maestro: <i>Ordens</i> (ações) envolvidas na execução do plano . . . . .	67
5.6	Maestro: <i>Contexto</i> inicial da música . . . . .	67

5.7	Maestro: <i>Ordem</i> de música . . . . .	68
5.8	<i>Luzes de Palco</i> : Interface nativa da lâmpada . . . . .	73
5.9	<i>Luzes de Palco</i> : Interface nativa da lâmpada: Cor . . . . .	73
5.10	<i>Luzes de Palco</i> : Identificação do plano . . . . .	74
5.11	<i>Luzes de Palco</i> : Responsáveis pelo plano . . . . .	75
5.12	<i>Luzes de Palco</i> : Papéis envolvidos no plano . . . . .	75
5.13	<i>Luzes de Palco</i> : <i>Contextos</i> do plano . . . . .	76
5.14	<i>Luzes de Palco</i> : Ordens (ações) envolvidas na execução do plano . . . . .	76



# Introdução

Neste capítulo enquadra-se o contexto do trabalho, indicando as motivações e questões em aberto no problema a resolver. O capítulo termina com a descrição da organização deste documento.

## 1.1 Cloud/Fog Computing e dispositivos IoT

O mundo tem vindo a evoluir a um ritmo cada vez mais acelerado em quase todas as suas vertentes e a informática está presente em quase todas elas. Esta é, sem qualquer margem para dúvida, uma das áreas que mais se tem desenvolvido e a um ritmo cada vez mais frenético. As áreas da informática que mais se têm destacado nos últimos tempos são a inteligência artificial [1], o processamento de grandes quantidades de dados e extração de informação, também referida como *big data* [2], a computação na nuvem (*cloud computing*) [3] e a internet das coisas (*Internet-of-Things* (IoT) [17]).

O termo *cloud* faz parte da linguagem corrente dos nossos dias. Para o cidadão comum a *cloud* é percebida como um grande armazém de dados, mas na verdade a *cloud*, para além da capacidade de armazenamento é composta por serviços diversificados que permitem uma capacidade de processamento em larga escala, e outros serviços como por exemplo o reconhecimento de faces de pessoas e outras coisas em fotos [15]. Estas características permitem o processamento e armazenamento de informação, dispensando custos de manutenção com reais benefícios económicos. A *cloud* também tem desvantagens, como por exemplo: a latência provocada pela distância do cliente às máquinas que suportam o processamento na *cloud* e a falta de privacidade resultante da entrega dos dados de negócio do cliente que,

muitas vezes, são suporte essencial à sua sobrevivência e têm que permanecer confidenciais. Embora a *cloud* permita efetuar cópias de segurança do software e dos dados, caso algo aconteça aos dados os negócios podem sofrer fortes consequências.

Para além da *cloud* (nuvem) estão também na ordem do dia o *fog computing* [10] (nevoeiro) e o *edge computing* (orla) [6], que permitem aproximar os dispositivos IoT ao centro de processamento de dados, alavancando assim a diminuição da latência nas comunicações.

Atualmente existe uma grande discussão sobre se o *fog computing* e o *edge computing* são, ou não, a mesma coisa [12]. No âmbito deste trabalho vamos assumir que *edge computing* e *fog computing* são semelhantes face ao objetivo de ajudar a diminuir a latência nas comunicações e a quantidade de dados enviados para a *cloud* pelos dispositivos IoT, possibilitando a utilização de técnicas de filtragem e agregação de dados.

Este trabalho alarga o leque dos dispositivos normalmente associados ao IoT, passando a incluir todos os tipos de aparelhos informáticos, como computadores pessoais, *tablets*, telefones inteligentes, relógios inteligentes, entres outros.

## 1.2 Contexto tecnológico

Atualmente o paradigma de *cloud computing* é uma realidade indesmentível, proporcionando de forma fácil, e a pedido, o acesso a recursos computacionais de grande capacidade. Assim é atualmente mais económico recorrer a *clouds* públicas, pagando apenas os recursos computacionais realmente utilizados (*on-demand resources*) [3], do que adquirir, instalar, configurar e manter infraestruturas de computação em centros de dados localizados nas próprias organizações. Uma das principais características da *cloud* é o *elastic computing* [9], termo que refere a capacidade de adicionar ou diminuir dinamicamente recursos ao sistema, adaptando-se às variações de carga do negócio.

Os serviços oferecidos pelos fornecedores de sistemas de *cloud computing* são normalmente classificados e oferecidos numa pilha de níveis [22]. O nível mais baixo da pilha é designado como *Infrastructure-as-a-Service* (IaaS) e oferece essencialmente serviços de computação, de armazenamento de dados e de comunicação em rede (*firewall*, redes virtuais VPN, etc), bem como, reencaminhamento de mensagens para efeitos de equilíbrio de carga. O nível intermédio, designado *Platform-as-a-Service* (PaaS), disponibiliza *frameworks* de desenvolvimento de aplicações tipicamente organizados como *Web API* e serviços de armazenamento e acesso a dados (bases de dados relacionais e NoSQL). No topo da pilha existe o nível *Software-as-a-Service* (SaaS), que disponibiliza as designadas aplicações *cloud*, por exemplo redes sociais, correio eletrónico, processamento de fotos e vídeo, gestão de identidades, etc.

O aparecimento do paradigma de *Internet-of-Things* (IoT), onde milhões de dispositivos que utilizam essencialmente redes sem fios para comunicar, introduziu a

necessidade de criar níveis intermédios de comunicação (*fog computing*) onde os dados podem ser agregados e ou filtrados [23], [21].

No entanto, devido à diversidade e complexidade dos casos de aplicação, continua a não existir uma arquitetura unificada para definir arquiteturas de coordenação de múltiplos dispositivos IoT, apesar de múltiplos esforços existentes, por exemplo em [21], [18].

Adicionalmente estes esforços referem requisitos de interação *device-fog-cloud*, contrariamente ao pressuposto de interação proposto neste trabalho em que a interação é *cloud-fog-device* acrescida com requisitos de coordenação e sincronização temporal.

### 1.3 Questões em aberto

Por norma, na computação na *cloud*, a transferência de dados é feita dos dispositivos IoT para a *cloud*, recorrendo, ou não, ao suporte da camada intermédia *fog computing*.

No entanto, é possível questionar os seguintes pontos:

- Como aproveitar o enorme manancial de informação residente na *cloud* para gerir/coordenar todos esses dispositivos IoT que se encontram disponíveis, retirando o máximo proveito de cada um deles?
- Como coordenar um grupo de dispositivos para que estes em conjunto consigam somar mais valor do que a soma do valor da atividade de cada um deles isoladamente?
- Como sincronizar a execução de tarefas nos vários dispositivos por forma a garantir que todos atuam no tempo exato?

### 1.4 Motivação

A motivação para o trabalho é propor e validar soluções para o problema de a partir de dados existentes na *cloud* poder interactuar com dispositivos IoT no sentido de os coordenar na realização de ações em conjunto.

Assim sendo, neste trabalho o foco será direccionado para dados que existem na *cloud* e que serão transferidos de forma coordenada para os dispositivos IoT, utilizando o *fog computing* sempre que tal se justifique, para que estes realizem ações em conjunto.

Embora exista muita discussão à volta da nuvem (*cloud*) e do nevoeiro (*fog computing*), o desafio deste trabalho, fazendo uma analogia com o ciclo da água completo,

é a preocupação com a chuva (mensagens) que cai da nuvem inundando rios e lagos (*fog*) até chegar aos seres vivos (dispositivos IoT) que a consomem.

Assim, os dados que existem na *cloud* são organizados em planos de execução de ações, para os múltiplos dispositivos, em que cada plano é constituído por uma composição de eventos e informação destinados aos vários dispositivos IoT. Estes planos de execução são enviados para uma entidade coordenadora, posicionada na camada *fog computing*, que é responsável pela coordenação/sincronização dos planos e dos vários dispositivos a seu cargo. Para designar esta abordagem será utilizado o termo ***Rain Computing***.

Continuando a analogia do ciclo da água, a maior parte das vezes os seres vivos não consomem a chuva diretamente das nuvens, esta é primeiro armazenada em rios ou lagos, o que equivale à entidade coordenadora dos planos. Da mesma forma que os seres vivos, os dispositivos móveis têm funções muito diferentes uns dos outros consumindo diferentes quantidades de água. Existem ainda seres vivos que, para se adaptarem ao meio ambiente onde vivem, desenvolveram a capacidade de armazenamento de água, como por exemplo como os camelos.

Quem não gostaria de conseguir controlar a chuva? Conseguir indicar o local e a hora da chuva, bem como a quantidade certa. Era ter “sol na eira e chuva no nabal”.

É do conhecimento geral que chuva a menos prejudica o desenvolvimento dos seres vivos. Passando a ideia para o nosso trabalho, a falta de informação pode levar os dispositivos IoT a desaproveitarem as suas capacidades, que podem fazer falta noutro lado. Também não é novidade que chuva a mais é capaz de provocar a destruição do meio ambiente e dos próprios seres vivos, pelo que informação a mais, que não é processada pelos dispositivos, só serve para provocar a degradação dos recursos da rede de comunicações e das máquinas que a produzem, sem gerar qualquer valor. Este equilíbrio é difícil de gerir. O sistema *Rain Computing* procura uma solução para a resolução desta questão.

Assim, dados previamente definidos na *Cloud* (planos de ação) são enviados para inundar dispositivos, com intermediação de coordenadores, que sincronizam as ações executadas em cada dispositivo.

## 1.5 Definição do problema

Com o incremento exponencial de dispositivos móveis ligados em rede existem inúmeros desafios para resolver no que à sua coordenação diz respeito. De um lado temos os dispositivos que apresentam uma grande heterogeneidade, tanto nas suas capacidades de processamento como na sua forma de comunicar com o exterior. Do outro lado temos serviços de gestão e coordenação dos dispositivos com arquiteturas mais homogéneas, apesar de implementarem diferentes requisitos de negócio.

Existem diferentes necessidades de coordenação de dispositivos. Uma delas é exigir que os dispositivos tomem uma ação em simultâneo. Por exemplo, imaginemos a última jornada do campeonato de futebol, onde os jogos entre as equipas que estão ainda a lutar por determinados objetivos têm que ter início ao mesmo tempo. Assim os delegados ao jogo ficam ao telefone com os delegados dos outros jogos, ou com a sede da liga, para coordenarem o início de todos os jogos. Mas, e se os árbitros recebessem no seu relógio a notificação para começarem o jogo, dispensando toda a logística atual, será que iriam obter um resultado mais preciso? Neste caso estamos a falar de um sincronismo com uma precisão de poucos segundos, mas há casos onde a precisão poderá ter que ser maior. Por exemplo, no caso de termos uma orquestra a ser conduzida por um maestro virtual, ou mesmo uma coreografia de drones, que hoje em dia é frequente, onde se chegam a juntar mais de 1000 dispositivos. Esta coordenação tem que ser feita perto do centésimo de segundo, ou ainda menos.

Outra necessidade que pode aparecer é a de atribuir diferentes papéis aos dispositivos, de forma permanente ou somente transitória. O papel de um dispositivo não é mais do que a função que este assume em determinado momento. Um dispositivo poderá assumir mais do que um papel em simultâneo.

Uma grande preocupação a ter em conta é a quantidade de dados a transmitir pelos dispositivos IoT. Assim deve-se tentar diminuir o tamanho da informação, quer eliminando a repetição dos dados como pela sua compactação, tentando ainda diminuir o número de mensagens trocadas, juntando, quando tal for possível, as mensagens em lotes.

Fora do âmbito deste trabalho ficam a integridade e confidencialidade da informação trocada bem como a não repudição da mesma, ou seja, questões relacionadas com segurança.

Emerge do problema um requisito de coordenar as ações dos dispositivos e, dado o elevado número de dispositivos que podem estar envolvidos, identifica-se a necessidade da existência de entidades de coordenação, em que cada coordenador será responsável por um grupo de dispositivos que lhe forem confiados. Assim terão de existir tantos coordenadores quantos os necessários para responder às necessidades de negócio. Os coordenadores serão assim as entidades ao nível do *fog computing* capazes de comunicar de forma eficiente com os dispositivos.

Um coordenador de dispositivos terá de receber de forma organizada a descrição de um plano de execução, que contém informação que permita coordenar a interdependência das ações a executar pelos vários dispositivos.

Os planos de execução terão de ser especificados por especialistas de negócio e ser armazenados numa entidade central alojada na *cloud*. Para tal define-se o conceito de centro de controlo com as seguintes responsabilidades:

- Disponibilizar uma interface de utilizador que permita a definição de planos de execução para diversos cenários de aplicação;

- Armazenar os planos de execução e enviá-los para as entidades coordenador que, de acordo com o plano, coordenam os dispositivos na execução das respectivas ações;

Do ponto de vista dos dispositivos, a estratégia a seguir é, neste sentido, a de definir uma especificação de interface mínima que os dispositivos tenham de suportar, assumindo assim um pressuposto de que alguns dispositivos não possam ser usados.

De seguida são descritos, como exemplo, alguns dos problemas que se tratam neste trabalho:

- A sincronização entre os vários dispositivos, quando é exigido que, por exemplo, todos os dispositivos necessitem executar uma tarefa em simultâneo, ou em horários pré-determinados;
- A diversidade de dispositivos para um mesmo serviço, ou dispositivos que executam tarefas diferenciadas, assumindo diferentes papéis.

Assim identificam-se como grandes desafios a especificação de um plano de execução, em metadados, e de um conjunto de mensagens que permitam a coordenação dos dispositivos. O coordenador recebe um plano que é composto por toda a informação necessária à execução do mesmo, garantindo a execução das ações de forma coordenada entre todos os dispositivos participantes.

O plano de execução deverá ser o conjunto de informação estruturada de forma compreensível por todos os intervenientes que a necessitem conhecer. A totalidade da informação do plano não será importante para todos os intervenientes e poderá ser partilhada apenas aquela que tenha relevância para cada um dos intervenientes, seja o coordenador ou um dispositivo.

Assim, o grande desafio do trabalho pode ser resumido - como desenhar e implementar uma solução assente num modelo, uma arquitetura lógica e um protótipo que possibilite demonstrar e avaliar a execução de planos concretos de diferentes cenários de aplicação, como por exemplo um maestro virtual. Aplicação em que músicos têm um dispositivo que recebe ações para sinalizar o início e o fim da música, a passagem da parte do verso para a parte do refrão, etc.

Em síntese o problema consiste em encontrar um modelo e uma arquitetura o mais genéricos possível que respondam às seguintes dimensões:

- Como definir um plano global de ações a serem realizadas pelos dispositivos?
- Como interpretar um plano e coordenar as ordens a enviar aos dispositivos tendo em conta as limitações de comunicação, latências que possam existir na rede, e a diversidade de capacidades de computação dos vários dispositivos?

## 1.6 Organização do documento

No capítulo seguinte, "2. Trabalho Relacionado", e tal como o nome indica, são abordados trabalhos relacionados com o *Rain Computing*, na tentativa de encontrar algo que possa ajudar ao desenvolvimento deste trabalho.

De seguida, no capítulo "3. Proposta de Solução" é elaborada uma proposta de solução, descrevendo o modelo e a arquitetura pretendida.

No capítulo "4. Implementação do Sistema *Rain Computing*" é descrita a implementação da arquitetura, e indicadas as tecnologias utilizadas.

Já no capítulo "5. Operacionalidade do Sistema *Rain Computing*", são apresentados dois estudos de caso, um mais simples e outro mais elaborado e os resultados obtidos na sua validação.

Por fim, no capítulo "6. Conclusões e Trabalho Futuro" é discutida a aprendizagem e resultados obtidos, indicando possíveis caminhos a seguir no futuro.





## Trabalho Relacionado

Apesar do extenso trabalho na área de IoT e *cloud computing* e da pesquisa por trabalhos relacionados ter sido considerável, não foi encontrada nenhuma metodologia ou arquitetura com todas as características que se pretendem no sistema *Rain Computing*. Existem muitos trabalhos com grande ênfase noutras matérias pouco, ou nada, discutidas neste trabalho, tais como a segurança ou a poupança de energia. Durante a pesquisa, surgiram alguns trabalhos com bastante detalhe no componente de orquestração de dispositivos, mas sempre muito ligados a determinados tipos específicos de dispositivos, muitas vezes dispositivos médicos IoMT (*Internet of Medical Things*) [16].

Os trabalhos apresentados de seguida são os que estão mais próximos dos objetivos pretendidos para o *Rain Computing*.

### 2.1 Agendamento (*scheduling*) de tarefas em dispositivos IoT

O gestor JarvSis [20] permite efetuar o agendamento de tarefas para aplicações IoT.

De todos os trabalhos analisados o JarvSis é, sem dúvida, o trabalho mais parecido com o *Rain Computing*. Se por um lado a primeira leitura revelava grandes semelhanças, ao ponto de fazer levantar a hipótese de não se justificar avançar para o *Rain Computing*, por outro lado acabava por validar a linha de pensamento do *Rain Computing*. Uma análise mais cuidada ao mesmo fez sobressair algumas limitações que justificaram plenamente a continuação do trabalho.

O modelo do JarvSis está dividido em três camadas, uma camada de controlo global presente na *cloud*, uma camada de controlo intermédia ao nível do *fog computing node*, que chamam *node*, e os dispositivos. Na camada intermédia pode existir uma hierarquia de *nodes*, fazendo com que possam existir *nodes* cuja função é apenas a coordenação de outros *nodes* que comunicam com os dispositivos. Tem um modelo extensível, permitindo diversos tipos de comunicação e codificação das mensagens para abranger mais tipos de dispositivos.

Os comandos para os dispositivos são configurados numa consola desenvolvida pelo próprio cliente do JarvSis. Para conseguir carregar os comandos no sistema JarvSis, esta consola do cliente deverá respeitar a interface de carregamento de comandos disponibilizada pelo JarvSis.

Esta plataforma não suporta dispositivos com diferentes papéis na execução de tarefas. Dado que uma tarefa é atribuída a um dispositivo específico, se este estiver com problemas não é possível ser substituído automaticamente por outro dispositivo com as mesmas capacidades. Para além deste importante fator, acresce ainda o facto de que se existirem vários dispositivos a executar uma tarefa semelhante a configuração vai repetir comandos iguais para os diversos dispositivos. Outra funcionalidade que o JarvSis tem em falta é a capacidade de incluir dados predefinidos na execução da tarefa, visto que na configuração dos comandos não apresenta essa possibilidade. Assume-se que esses dados são pré-carregados nos dispositivos, sendo estáticos, ou seja, iguais em todas as execuções.

Concluindo, as funcionalidades em falta, referidas anteriormente, e as limitações na construção de um conjunto de tarefas justificam o interesse na continuação da direção seguida para o sistema *Rain Computing*, como poderemos ver nos capítulos seguintes.

## 2.2 Controlador de dispositivos IoT com decisão em tempo real

Em [19] é apresentado um trabalho com grande foco nas políticas de decisão/controlo em tempo real. Estas políticas são feitas de forma distribuída até ao nível dos próprios dispositivos, com objectivo de reduzir a complexidade e flexibilizar a integração. É uma linha de pensamento interessante, mas que se afasta do que se pretende obter com o sistema *Rain Computing*. O trabalho apresentado em [19] também tem foco na segurança dos sistemas, que é um tópico fora do âmbito do *Rain Computing*.

Em cima da camada dos dispositivos, o sistema apresenta três níveis de controladores:

- Global: é a raiz e, simultâneamente, o motor do modelo, sendo responsável pela decisões mais críticas. controla os nós do tipo Super Controlador;

- Super Controlador: está num nível intermédio, entre o Global e o Controlador Local, e serve para coordenar os Controladores Locais;
- Controlador Local: gere os dispositivos, tomando decisões mais simples que não necessitam de permissão superior.

A arquitetura foi definida desta forma para garantir a qualidade do serviço, flexibilidade e escalabilidade, respondendo de forma célere e confiável ao constrangimento de qualquer peça do sistema. É um sistema destinado a gerir, por exemplo, a condução autónoma.

Este trabalho, apesar de ter preocupações em comum com o *Rain Computing*, acaba por se dedicar muito ao tema da tomada de decisões em tempo real, que passam ao lado do que se pretende obter com o *Rain Computing*. Por outro lado, não define a estrutura dos dados do sistema, nomeadamente do plano de execução e das mensagens para os dispositivos, tal como se pretende fazer com o sistema *Rain Computing*.

## 2.3 Coordenação de aplicações IoT para cidades inteligentes

Em [26] é construído um modelo de coordenação de aplicações IoT para cidades inteligentes que está ligado em alguns pontos ao sistema *Rain Computing*. Mais uma vez, existe uma grande preocupação com a heterogeneidade dos dispositivos existentes, tentando construir um modelo que consiga albergar o maior número possível de dispositivos.

O modelo já demonstra uma preocupação com os dados necessários à execução de tarefas, mas não estrutura a sua organização e refere-a como sendo estática. Este modelo também divide a sua organização em três camadas, mas fá-lo entre dados, computação e comunicações.

A associação do dispositivo em relação ao *broker*, que é a peça que coordena a execução, é feita por proximidade geográfica, sendo automaticamente alterada com a deslocação do dispositivo para junto de outro *broker*.

O trabalho apresentado em [26] partilha alguns objetivos com o *Rain Computing*, mas tem uma arquitetura que lhe dá uma direção diferente, não contemplando todos os desafios a que o Rain Compute se propõe responder.

## 2.4 Coordenação de dispositivos inteligentes em missão

Em [27] é elaborado um sistema sobre coordenação de dispositivos inteligentes em missão que regista várias semelhanças com os temas abordados no *Rain Computing*. Tem foco no IoMT, *Internet-of-Mobile-Things*, não em todo o universo IoT, mais especificamente em dispositivos móveis inteligentes. O trabalho referido em [27] tem uma grande relação com dispositivos que possuem vários sensores e atuadores, como por exemplo drones, quer sejam aéreos ou terrestres. Estes dispositivos possuem vários sensores e atuadores. A precisão temporal e espacial requerida é muito grande.

O plano de execução é chamado de missão e é composto por tarefas. Este sistema organiza os dispositivos por grupos.

O coordenador é um dispositivo eleito pelos seus pares. Os dispositivos recebem as tarefas do coordenador, mas também comunicam entre si, partilhando dados da execução. Os autores dividem o seu trabalho em quatro camadas. Duas delas são a missão e as tarefas, que são um conjunto de comandos para os dispositivos, onde são agregadas todas as mensagens que são necessárias para efetuar uma tarefa. De seguida vem a camada das mensagens que tem correspondência direta com um comando a enviar ao dispositivo. Por último existe a camada de comunicação.

O trabalho apresentado também se foca um pouco nas características de hardware dos dispositivos, que não é um objetivo do sistema *Rain Computing*.

O trabalho aqui analisado tem muitos pontos em comum com o *Rain Computing*, mas está mais centrado na coordenação por localização geográfica de dispositivos em movimento, estando muito comprometido com as tecnologias utilizadas, demonstrando pouca flexibilidade em relação à heterogeneidade dos dispositivos.

# 3

## Proposta de Solução

Este capítulo, apresenta os conceitos e pressupostos tidos em conta na procura de uma solução para a resolução do problema discutido e apresentado nos capítulos anteriores. Assim, a partir da enumeração dos requisitos funcionais e não funcionais considerados, apresentados na secção 3.1, é proposta uma solução que se descreve nas secções 3.2 e 3.3 . Na secção 3.2 é apresentado o modelo de interação entre as partes e os módulos envolvidos. Na secção 3.3 é apresentado o modelo de adesão ao sistema *Rain Computing*. Na secção 3.4 é apresentada a arquitetura do sistema que descreve as funcionalidades de cada nível e as responsabilidades de cada parte e módulos, indicando as soluções para os diversos requisitos envolvidos. O capítulo termina com a secção 3.5 com as conclusões e uma discussão dos pontos fortes e as limitações da solução proposta.

### 3.1 Especificação de requisitos

De forma sintética a proposta de solução permite que múltiplos dispositivos, onde se executam aplicações específicas relacionadas com um cenário de utilização real, sejam capazes de executar ordens emanadas de um módulo coordenador, que por sua vez recebeu um plano de execução a partir de um módulo designado de centro de controlo.

Podem existir múltiplos módulos coordenadores, podendo um dispositivo selecionar qual o coordenador a que se pretende ligar.

Para criar um plano de execução, um utilizador *especialista no negócio* de uma solução concreta, que conhece os detalhes do negócio a implementar no plano,

constrói um plano de execução. O plano contém a informação necessária à execução do negócio, estruturada por forma a obter o resultado pretendido e ao menor custo possível.

Pressupostos:

- Independentemente do negócio, o coordenador será capaz de coordenar a execução de todos os planos de execução concretos;
- O coordenador será capaz de coordenar um grande número de dispositivos.

### 3.1.1 Requisitos funcionais

Requisitos funcionais do ponto de vista do *especialista de negócio* que vai especificar um plano:

- Criar o plano segundo um esquema global a que todos os planos de execução concretos obedecem;
- Validar o plano face a um esquema global;
- Alterar um plano previamente criado.

Requisitos funcionais do ponto de vista do *operador* que inicia a execução do plano:

- Agendar a execução de um plano concreto;
- Abortar a execução de um plano concreto;
- Alterar a prioridade de execução de um plano concreto.

Um plano tem vários tipos de execução:

- *Síncrono*: é executado de forma coordenada no tempo para todos os dispositivos participantes, ligados a um coordenador;
- *Assíncrono*: não existe a necessidade de coordenar temporalmente os vários dispositivos participantes, ligados a um coordenador;
- *Batch*: é executado por um dispositivo de forma autónoma, sem ligação ao coordenador no momento da execução do plano. O plano é fornecido ao dispositivo pelo coordenador, em momento anterior à execução.

### 3.1.2 Requisitos não funcionais

O sistema *Rain Computing* apresenta os seguintes requisitos não funcionais:

- Não é considerada a segurança nos pilares de autenticação, integridade, confidencialidade e autorização;

- O centro de controlo é um processo que se executa num computador de uma *cloud* pública;
- Não são definidas estratégias de manutenção e tolerância a falhas específicas, sendo utilizadas as inerentes à própria *cloud*;
- Os módulos coordenadores são processos que se executam em qualquer computador com ligação à *cloud* num nível designado por *fog computing*.

## 3.2 Modelo de interação entre os módulos envolvidos

A figura 3.1 apresenta um diagrama de interação entre os vários módulos do sistema, a seguir descritos:

- **Centro de controlo:** módulo central do sistema onde se armazenam as informações globais do sistema, nomeadamente as múltiplas instâncias dos planos de execução e os módulos de software que disponibilizam as funcionalidades que permitem a um *especialista de negócio* elaborar planos e posteriormente a um *operador* agendar e iniciar a execução de um plano;
- **Coordenador:** módulo que recebe um plano do centro de controlo e coordena a execução das ações em cada dispositivo. O plano define se a sua execução será síncrona, assíncrona ou *batch* e o coordenador fará a execução do mesmo de acordo com essa indicação;
- **Dispositivos:** possuem uma aplicação que recebe ações para executar, enviadas pelo coordenador, e as executa de acordo com o plano. Os dispositivos assumem uma determinada função durante a execução de um plano. A esta função chamamos de papel, ou seja, os dispositivos desempenham um papel durante a execução de um plano concreto.

Os três módulos acabados de referir e suas interações são apresentados na figura 3.1 e detalhados de seguida:

Uma instância de um plano concreto cumpre sempre o *schema* **A** do *Rain Computing*. Para criar uma instância de plano concreto, com toda a informação necessária, estruturada de forma a se obter o resultado pretendido ao menor custo possível, é necessária a intervenção de um utilizador *especialista de negócio* **B**. O *especialista de negócio* conhece os detalhes do negócio que se pretende implementar no *Rain Computing*. O *especialista de negócio* utiliza uma interface gráfica disponibilizada pelo centro de controlo para construir o plano. Quando é concluída a construção do plano este é armazenado no repositório do centro de controlo.

Após a construção do plano é necessário agendar a execução do mesmo. Esta tarefa é realizada por um utilizador que denominamos de *operador* **C**. A função

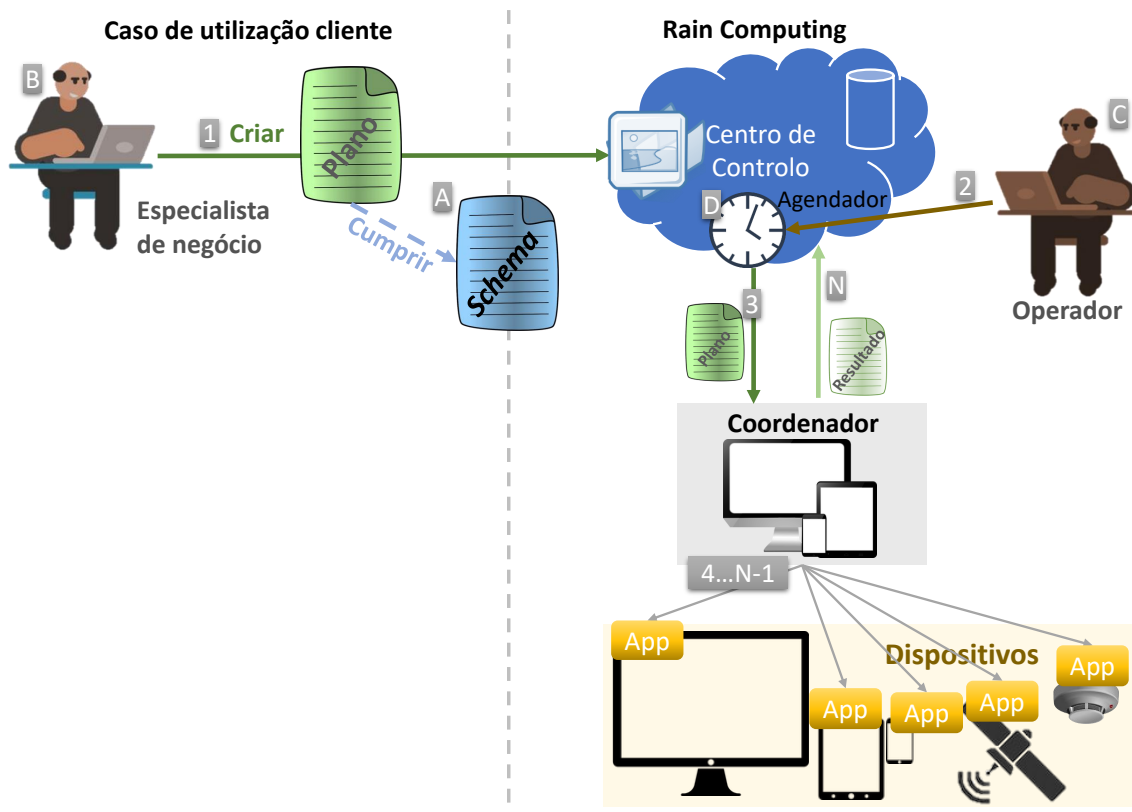


Figura 3.1: Desenho do modelo de processamento de um plano

do *operador* é configurar no *Rain Computing* os horários de execução do plano **2**. Para além dos horários de execução do plano, o *operador* também pode nomear o coordenador responsável pela execução do plano. Essa configuração é gerida por uma funcionalidade do centro de controlo chamada de *agendador* **D**. O *agendador* controla os horários de execução das instâncias de planos concretos. Quando chega à hora de execução de um plano, seleciona um dos coordenadores disponíveis e envia-lhe a instância do plano concreto **3**.

Quando o coordenador recebe um plano para executar, seleciona os dispositivos que melhor se adequam à sua execução. Para tal o coordenador começa por enviar aos dispositivos selecionados uma mensagem com indicação do estado inicial que estes devem assumir. De seguida, o coordenador envia as mensagens que contêm as ações a efetuar pelos dispositivos.

Cada dispositivo processa as mensagens com as ações a executar de acordo com o plano e devolve o resultado obtido ao coordenador. O coordenador filtra e agrega os resultados obtidos pelos dispositivos e envia-os para o centro de controlo **N**. O centro de controlo guarda os resultados que servem como estatísticas do sistema, mas também como medidas que poderão ser utilizadas pelo *especialista de negócio* na afinação de um plano de execução concreto, por exemplo na escolha do coordenador a executar o mesmo plano numa próxima vez.

### 3.3 Modelo de adesão ao sistema *Rain Computing*

Para se obter as interações dos módulos do sistema *Rain Computing* descritas na secção 3.2 é necessário que coordenadores e dispositivos se dêem a conhecer. A adesão de coordenadores e dispositivos ao sistema *Rain Computing* está descrita na figura 3.2 e é uma simples implementação do paradigma de *lookup* muito utilizado pelos sistemas distribuídos.

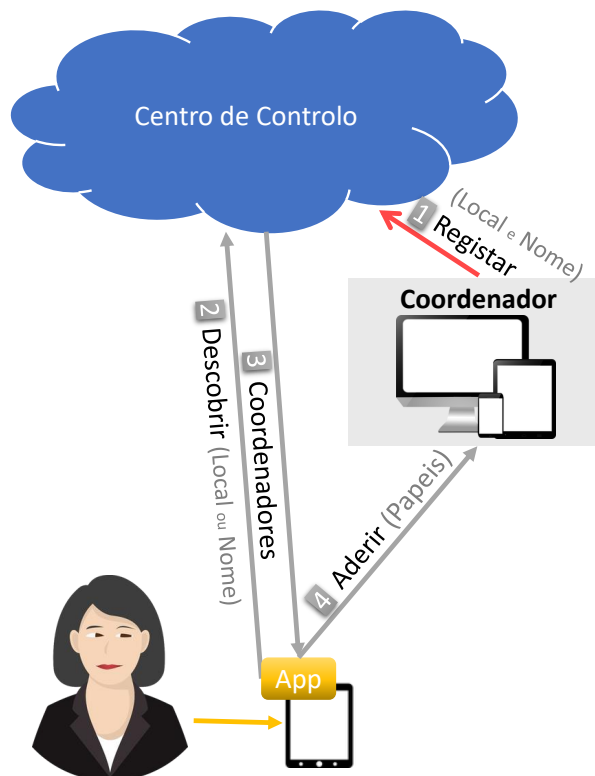


Figura 3.2: Desenho do modelo de adesão ao sistema

No início do sistema o centro de controlo nasce sem qualquer coordenador associado. Os coordenadores que pretendam aderir ao sistema devem registar-se no centro de controlo através do envio de uma mensagem de registo **1**. Nesta mensagem de registo do coordenador é incluído o seu nome bem como informação sobre a sua localização física, que pode ser interessante para alguns cenários de aplicação. Desta forma o centro de controlo consegue construir uma rede de coordenadores que utilizará para a execução de planos.

Quando o utilizador de um dispositivo pretende aderir ao *Rain Computing*, deve dar indicações ao dispositivo para este iniciar o processo de registo. Quando tal acontece, o dispositivo envia um pedido **2** ao centro de controlo com o propósito de obter uma lista de coordenadores disponíveis. Esta mensagem **2** leva informação sobre a localização do dispositivo, eventualmente o nome do coordenador pretendido ou mesmo o número de identificação do coordenador. Na resposta à

mensagem de descoberta enviada pelo dispositivo **3** o centro de controlo devolve uma lista de coordenadores que considera adequados para o dispositivo.

Após a receção da resposta do centro de controlo **3**, o dispositivo escolhe um dos coordenadores da lista e envia, ao coordenador selecionado, uma mensagem de registo **4**. Esta mensagem leva indicação das capacidades do dispositivo, designadas por papéis. Esta informação permitirá ao coordenador selecionar os melhores dispositivos para a posterior execução de planos de determinados cenários de aplicação.

## 3.4 Arquitetura lógica

Nesta secção, é descrita a arquitetura lógica de suporte à implementação do sistema *Rain Computing*. Serão materializados os formatos para as mensagens a trocar entre os três diferentes módulos intervenientes, o centro de controlo, os coordenadores e os dispositivos, bem como a estrutura do plano de execução.

A figura 3.3 representa essa arquitetura. As diferentes cores de planos e de ordens pretendem indicar diferentes instâncias desses componentes. As ordens são as ações que o dispositivo deve executar. É possível verificar a existência dos três módulos: o centro de controlo, que é executado ao nível da *cloud*; os coordenadores, a serem executados ao nível do *fog*; e os dispositivos associados a cada um dos coordenadores. É possível observar os planos a serem enviados a cada um dos coordenadores e estes a devolverem os resultados da execução dos planos ao centro de controlo. Por sua vez os coordenadores enviam as ordens aos dispositivos participantes na execução do plano e estes respondem às ordens com os resultados da sua execução.

### 3.4.1 Representação do plano de execução

O plano contém toda a informação necessária para a execução do negócio pretendido sendo criado e armazenado no centro de controlo. O plano está dividido em 5 partes, como se pode ver na figura 3.4. Nomeadamente a identificação do plano, os responsáveis pelo plano, os papéis envolvidos no plano, os *contextos* do plano (estado inicial dos dispositivos); e as *ordens* (ações) envolvidas na execução do plano. A parte do plano correspondente aos papéis contém apontadores para as partes dos contextos e das ordens. Estas partes são descritas nas subsecções seguintes.

#### 3.4.1.1 Identificação do plano

A identificação do plano, apresentada na tabela 3.1, tal como o nome sugere, serve não só para identificar de forma unívoca um plano, mas também o caracteriza quanto ao seu propósito e à sua forma de funcionamento.

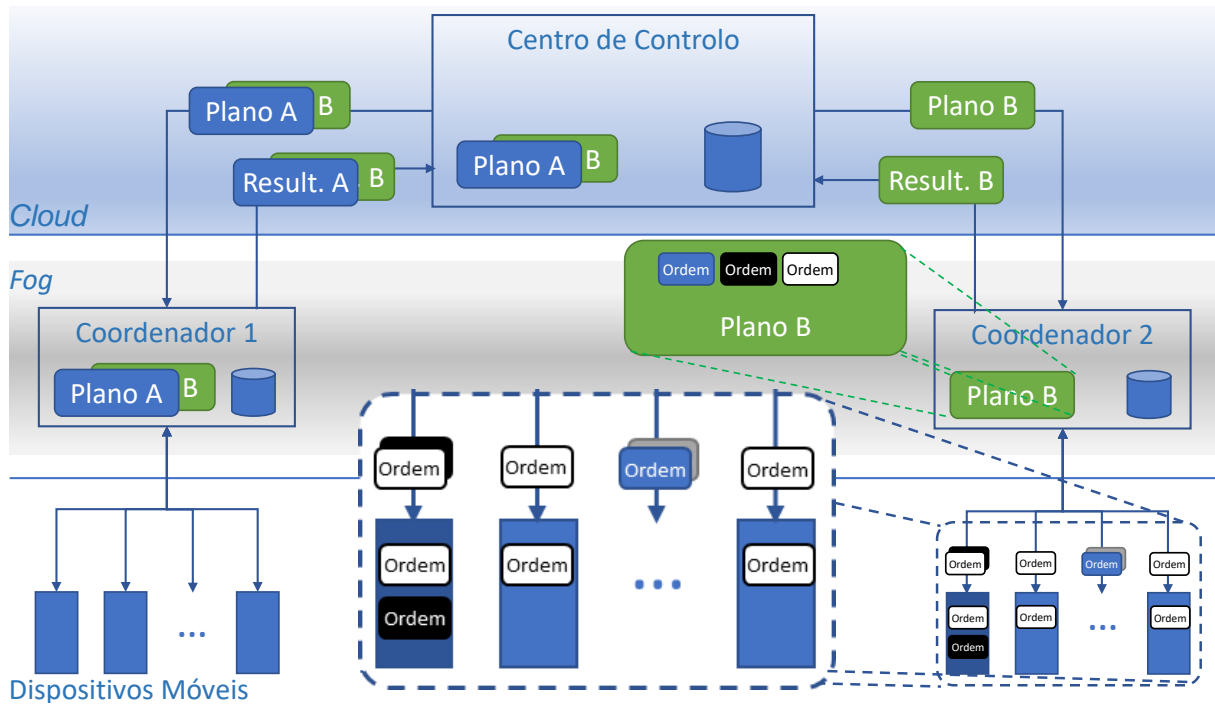


Figura 3.3: Arquitetura Lógica Global

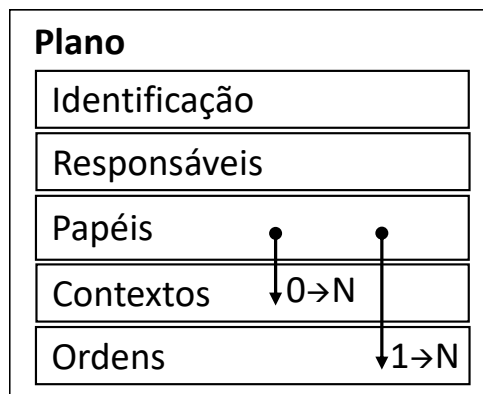


Figura 3.4: Partes do plano de execução

A parte de identificação do plano pode, por sua vez, ser dividida em várias subpartes das quais se destacam as seguintes:

- **ID, Aplicação e Nome:** compõem a identificação do plano. Em primeiro lugar vem o atributo ID, que é um identificador numérico universal, não existindo dois planos com o mesmo identificador; de seguida vem o atributo Aplicação que também se apresenta como um identificador numérico universal, mas neste caso associado à aplicação, não existindo duas aplicações com o mesmo identificador; por último vem o atributo Nome em formato texto que corresponde ao nome amigável do plano de execução concreto. O primeiro e segundo atributos servem para processamento por uma máquina, o terceiro

Plano				
Atributo	Tipo		Valor	Descrição
<b>Identificação</b>				
ID	Número	8		Uma identificação simplificada do plano
Aplicação	Número	8		Uma identificação simplificada da aplicação
Nome	Texto	80		Um nome que permita uma identificação fácil do propósito do plano
Tipo	Texto	1	A: Assíncrono B:Batch S: Síncrono	
Prioridade	Número	3		Prioridade de execução do plano
Categoria	Texto	1	D: Desporto M: Música S: Sensor T: Tráfego V: Vídeo	
Tipo de unidade temporal	Texto	3	ANO, MES, DIA, HOR, MIN, SEG, MIL, MIC	Tipo da unidade temporal de execução
Unidade temporal	Número	8		Número de unidades que ocorrem em cada tipo de unidade temporal
Data de início	DataHora	20		Data de ativação do plano
Data de fim	DataHora	20		Data de inativação do plano
Data de criação do plano	DataHora	20		Momento da criação do plano
Utilizador de criação do plano	Texto	20		Utilizador da criação do plano
Data de alteração do plano	DataHora	20		Data da última alteração ao plano
Utilizador de alteração do plano	Texto	20		Utilizador da alteração do plano

Tabela 3.1: Identificação do plano

atributo destina-se ao processamento humano;

- **Prioridade:** indica a prioridade do plano de execução para, de alguma forma, permitir o desempate em caso de execução simultânea de dois planos distintos, sendo privilegiado o plano com o menor número de prioridade;
- **Tipo de unidade temporal e Unidade temporal:** definem a unidade temporal de execução do plano que é indicada nas ordens (ações) pertencentes ao plano de execução concreto. Por exemplo uma unidade temporal do tipo minuto com um valor de 120 indica que a unidade temporal ocorre 120 vezes num minuto, ou seja, ocorre de meio em meio segundo.

### 3.4.1.2 Responsáveis pelo plano

A parte dos responsáveis do plano, representada na tabela 3.2, agrega informação sobre os responsáveis pelo plano de execução. Para além do nome do responsável, constam ainda dados sobre a localização mais habitual do mesmo, bem como os seus dados de contacto: telefone e email.

Esta informação pode ser utilizada para solicitar ajuda com qualquer problema que surja durante a execução do plano.

Plano				
Atributo	Tipo		Valor	Descrição
Responsável []				
Nome	Texto	80		Nome de responsável do plano
Telefone	Número	15		Telefone
Email	Texto	40		Conta de correio eletrônico
País	Texto	24		País de residência do responsável
Cidade	Texto	24		Cidade de residência do responsável

Tabela 3.2: Responsáveis pelo plano

### 3.4.1.3 Papéis envolvidos no plano

Na parte dos papéis envolvidos no plano, tal como se encontra ilustrado na tabela 3.3, são indicados os papéis que são necessários para assegurar a completa execução do plano.

Plano				
Atributo	Tipo		Valor	Descrição
Papéis []				
ID	Número	2		Uma identificação simplificada do papel
Nome	Texto	24		Nome do papel a quem se destinam as instruções que se seguem
Descrição	Texto	256		Descrição do papel
Contexto	Número	2		O contexto
Ordens	Número []	4x...		Um conjunto de ordens a efetuar em sequência

Tabela 3.3: Papéis envolvidos no plano

Um papel não é mais do que uma função, caracterizada por um conjunto de ações, que denominamos de ordens, que um dispositivo assume durante a execução do plano. Pode ser, por exemplo, algo como "sensor de temperatura" ou "sensor de humidade".

A identificação do papel é feita pelo nome do papel ou por um identificador (ID) numérico único. Existe também uma descrição sobre o papel.

O papel pode ter um contexto inicial associado, este contexto não é mais do que o estado inicial dos dispositivos que irão executar este papel. Ver detalhes do contexto na parte 3.4.1.4. O atributo contexto é apenas um identificador do contexto e não o próprio conteúdo do contexto. Desta forma, é possível que vários papéis partilhem o mesmo contexto evitando repetições de informação e consequentemente o tamanho total de um plano.

Por fim, existe um vetor de identificadores de ordens. As ordens são as ações que o coordenador enviará para os dispositivos executarem de acordo com o papel de cada um, ver a parte 3.4.1.5. O vetor de identificadores de ordens permite identificar que ordens estão associadas ao papel em causa. Mais uma vez, tal como para o contexto, apenas os identificadores estão presentes, permitindo que diferentes papéis partilhem a mesma ação. As ordens vêm indicadas na última parte do plano.

### 3.4.1.4 Contextos do plano

Um plano pode estar relacionado com múltiplos papéis e cada um dos papéis pode ter associado um contexto, que consiste nos dados do estado inicial para esse papel.

A estrutura do contexto está representada na tabela 3.4.

Plano				
Atributo	Tipo		Valor	Descrição
Contexto []				
ID	Número	2		Uma identificação simplificada do contexto
Nome	Texto	24		Breve nome do contexto
Descrição	Texto	256		Descrição do contexto
Tempo de espera	Número	8		Tempo de espera pela resposta (mls)
Gestão de erro	Texto	1	A: Abortar C: Continuar	Indica o comportamento em caso de não chegarem todas as respostas
Tipo de Codificação	Texto	1	J: JSON P: Proprietary T: Plain Text X: XML	Indica o formato dos dados da ordem
Dados	Texto	...		Dados do contexto, específico do negócio

Tabela 3.4: Contextos do plano

Esta parte contém os contextos iniciais de processamento do plano.

A identificação do contexto é feita pelo nome do contexto ou por um identificador (ID) numérico único. Existe ainda uma descrição sobre o contexto.

Existe um atributo (Tempo de espera) para indicar o tempo que o coordenador deve esperar pelas respostas dos dispositivos ao envio dos contextos. O valor deste atributo é expresso em milissegundos.

Existe também um atributo (Gestão de erro) que indica ao coordenador o que deverá fazer em caso de erro na execução de um contexto. O erro pode ter origem na ausência de resposta, por parte de um dos dispositivos, ao contexto enviado, ou quando as respostas de um dos dispositivos indicam a ocorrência de um erro. Este atributo pode assumir os seguintes valores:

- **Abortar:** cancelamento da execução do plano;
- **Continuar:** a execução do plano continua como se o processamento do contexto, por parte dos dispositivos, tivesse sido efetuado com sucesso.

O formato de codificação da mensagem é representado por um atributo (Tipo de Codificação) que pode assumir valores tais como: JSON, XML, *Plain Text*, *Proprietary*, etc.. Este formato de codificação refere-se à mensagem do contexto inicial de acordo com o indicado neste atributo e indica em que formato a mensagem está codificada no plano.

O atributo Dados contém o contexto a ser enviado a um dispositivo. Este será composto pelos estados iniciais do papel a ser executado por um dispositivo.

### 3.4.1.5 Ordens (ações) envolvidas na execução do plano

Na tabela 3.5 é possível observar a estrutura das ordens existentes num plano.

Plano				
Atributo	Tipo		Valor	Descrição
Ordens []				
ID	Número	4		Uma identificação simplificada da ordem
Nome	Texto	24		Breve nome da ordem
Descrição	Texto	256		Uma descrição do significado da ordem
Momento	Número	8		Momento da execução da ordem
Gestão de erro	Texto	1	A: Abortar C: Continuar E: Executar em erro U: Erro na ordem anterior	Indica o comportamento da ordem em caso de erro
Tipo de Codificação	Texto	1	J: JSON P: Proprietary T: Plain Text X: XML	Indica o formato dos dados da ordem
Requer Resposta	Booliano	1		Indica se a ordem requer resposta do dispositivo
Dados	Texto	...		Dados da ordem, específica dos serviços

Tabela 3.5: Ordens (ações) envolvidas na execução do plano

As ordens são as ações enviadas pelos coordenadores aos dispositivos de acordo com o papel de cada um.

A identificação da ordem é feita pelo seu nome ou por um identificador (ID) numérico único.

Existe também um atributo (Gestão de Erro) que indica ao coordenador o que deverá fazer em caso de erro na execução de uma ordem. Este atributo pode assumir os seguintes valores: abortar, continuar, erro (ordem que só é executada em caso de erro) e última (só é executada em caso da ordem imediatamente anterior ser concluída com erro).

- **Abortar:** cancelamento do processo quando ocorre um erro na ordem em causa;
- **Continuar:** o processamento continua, como se não tivesse ocorrido qualquer erro;
- **Erro:** executar apenas em caso de ocorrência de erro. Estas ordens são sempre executadas independentemente da ordem onde ocorreu o erro;
- **Última:** ordem a executar caso a ordem executada imediatamente antes tenha dado erro. Estas ordens podem propagar o erro da anterior, ou limpar o erro, permitindo que o processamento prossiga.

O formato de codificação da mensagem é representado pelo atributo Tipo de Codificação que já foi descrito na subsecção anterior e pode assumir valores tais como: JSON, XML, *Plain Text*, *Proprietary*, etc..

Existe um atributo (*Requer Resposta*) que indica se o coordenador necessita de esperar pela resposta do dispositivo, após o envio da ordem ao dispositivo.

O atributo *Dados* de uma ordem contém um conteúdo capaz de ser interpretado e executado pelos dispositivos que cumpram os respectivos papéis do plano.

### 3.4.2 Fluxos de comunicação

Para efetuar a operacionalidade do sistema *Rain Computing* é necessário um fluxo de comunicação por mensagens, fluxo esse que é apresentado na figura 3.5. Existem diversos cenários possíveis para a execução do plano, as setas a tracejado representam as mensagens opcionais. As únicas mensagens obrigatórias são as mensagens *Planear*, *Ordenar* e *Resultado*. As mensagens *Ordenar* e *Sincronizar* são enviadas tantas vezes quantas as necessárias para cumprir o seu propósito.

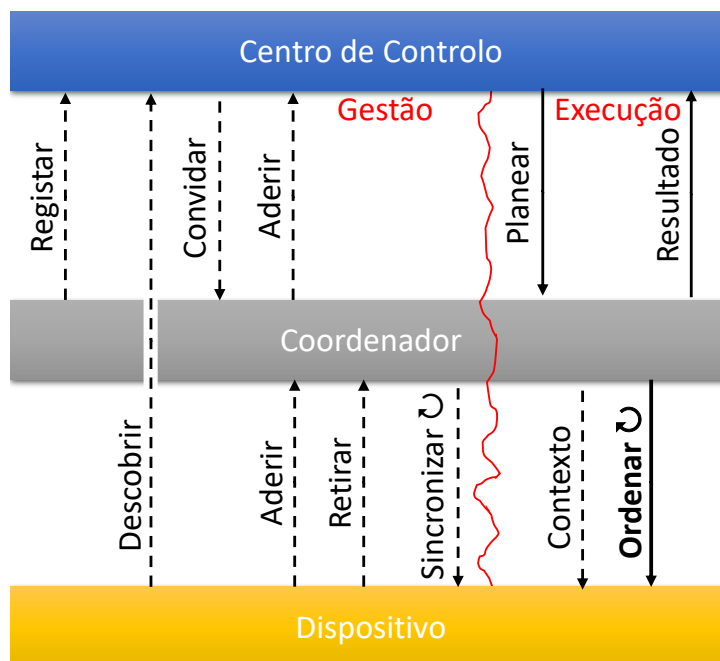


Figura 3.5: Fluxo Global de Comunicação

Os dados de negócio que viajam nas mensagens *Contexto* e *Ordenar* são específicos de cada negócio, pelo que a sua estrutura deverá ser definida pelo *especialista de negócio* utilizando a interface gráfica disponível.

As mensagens podem ser trocadas em vários formatos, tais como XML, JSON, texto simples, etc., mas o seu conteúdo será sempre o mesmo.

A estrutura das mensagens é indicada nas próximas secções. Antes da descrição das mensagens será apresentada a estrutura do cabeçalho comum a todas elas.

O cabeçalho está presente em todas as mensagens, tanto nos pedidos, como nas respostas. Tal como é possível observar na tabela 3.6, permite identificar o tipo de mensagem, a sua origem, o momento do seu envio e a sua sequência.

Cabeçalho				
Nome	Tipo		Valor	Descrição
<b>Pedido</b>				
Tipo mensagem	Texto	1	A: Aderir R: Retirar P: Alterar Papel	Tipo de mensagem
Pedido-Resposta	Texto	1	P: Pedido	Sub tipo de mensagem
Versão	Número	1	1-9	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	24		Momento da formatação da mensagem
Número de sequência	Número	4		
<b>Resposta</b>				
Tipo de mensagem	Texto	1	Igual ao pedido	Tipo de mensagem
Pedido-Resposta	Texto	1	R: Resposta	Sub tipo de mensagem
Versão	Número	1	Igual ao pedido	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	24		Momento da formatação da mensagem
Marca temporal do pedido	DataHora	24		Retirado do pedido
Número de sequência	Número	4		Retirado do pedido

Tabela 3.6: Cabeçalho de Mensagem.

O cabeçalho de uma resposta inclui ainda a identificação do pedido a que diz respeito.

O cabeçalho do pedido é composto pelo *Tipo de Mensagem*, que identifica o objetivo da mensagem. Existem os seguintes tipos de mensagem:

- **Registrar:** mensagem que permite que um coordenador se dê a conhecer ao centro de controlo;
- **Descobrir:** mensagem que permite que um dispositivo descubra, por intermédio do centro de controlo, o coordenador que deverá utilizar no processamento;
- **Convidar:** mensagem que permite que o centro de controlo apresente um dispositivo a um coordenador. Esta mesma mensagem permite também que, posteriormente, o coordenador solicite a adesão de um dispositivo ao sistema;
- **Aderir:** mensagem que permite a adesão de um dispositivo ao serviço, feito através de um coordenador;
- **Retirar:** Permite que um dispositivo cancele a sua subscrição do serviço;
- **Planear:** transmissão de um plano a um coordenador, por parte do centro de controlo;
- **Contexto:** transmissão do contexto inicial a um dispositivo, por parte do coordenador;
- **Ordenar:** transmissão de uma ordem a um dispositivo, por parte do coordenador;
- **Resultado:** transmissão do resultado da execução de um plano, feita por um coordenador ao centro de controlo;
- **Sincronizar:** mensagem utilizada para sincronizar um dispositivo, medindo a latência das comunicações. Também utilizada como *heartbeat*;

- **Erro:** mensagem de resposta a qualquer uma das mensagens anteriores sempre que ocorra um erro não controlado no processamento da mensagem pedido.

O cabeçalho tem um identificador para distinguir se é um pedido ou uma resposta, e uma versão de mensagem.

Com estes 3 campos, o Tipo de Mensagem, o indicador de Pedido-Resposta e a Versão da mensagem é possível identificar a mensagem, e conseqüentemente a sua estrutura de dados. Para além dos campos anteriormente mencionados, temos a identificação da origem da mensagem, o momento do envio da mensagem, denominada de marca temporal, e um número de seqüência. A marca temporal e o número de seqüência devem permitir obter uma chave única para a mensagem, ou seja, permitir distinguir qualquer mensagem das restantes.

O cabeçalho da resposta é muito semelhante ao do pedido, mas não tem um número de seqüência próprio. Transposta a marca temporal e o número de seqüência do pedido a que diz respeito, para ser possível realizar o seu emparelhamento com o pedido.

### 3.4.2.1 Mensagem de erro

Aderir				
Nome	Tipo		Valor	Descrição
Resposta				
Cabeçalho				
Tipo de mensagem	Texto	1	Igual ao pedido	Tipo de mensagem
Pedido-Resposta	Texto	1	E: Erro	Sub tipo de mensagem
Versão	Número	1	Igual ao pedido	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	20		Momento da formatação da mensagem
Marca temporal do pedido	DataHora	20		Retirado do pedido
Número de seqüência	Número	4		Retirado do pedido
Corpo				
Código de erro	Número	8		Número sequencial único por erro
Descrição do erro	Texto	80		Descrição técnica do erro
Texto utilizador	Texto	80		Mensagem para o utilizador

Tabela 3.7: Mensagem de erro

A tabela 3.7 descreve a estrutura da mensagem de erro. Quando existe uma falha no processamento do pedido, independentemente da origem do problema, será enviada uma mensagem de erro, identificada como tal, que apenas pretende indicar de forma clara o problema ocorrido. Existe uma mensagem técnica, que contém informação técnica sobre o erro ocorrido e se destina a permitir que um técnico perceba em detalhe o erro ocorrido. Existe ainda uma mensagem que se destina ao utilizador para permitir que este saiba como proceder perante a situação de erro.

Registrar				
Nome	Tipo		Valor	Descrição
Pedido				
cabecalho				
Tipo de mensagem	Texto	1	G: Registrar	Tipo de mensagem
Pedido-Resposta	Texto	1	P: Pedido	Sub tipo de mensagem
Versão	Número	1	1-9	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	20		Momento da formatação da mensagem
Número de sequência	Número	4		
Corpo				
Coordenador				Dados do coordenador
ID	Número	11		Identificação única
Nome	Texto	40		Nome
Descrição	Texto	80		Descrição
Código Postal	Número	7		
Estado	Texto	64	A: Ativo I: Inativo	Estado de funcionamento
Ligação	Texto	256		Dados da ligação
Resposta				
cabecalho				
Tipo de mensagem	Texto	1	Igual ao pedido	Tipo de mensagem
Pedido-Resposta	Texto	1	R: Resposta	Sub tipo de mensagem
Versão	Número	1	Igual ao pedido	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	20		Momento da formatação da mensagem
Marca temporal do pedido	DataHora	20		Retirado do pedido
Número de sequência	Número	4		Retirado do pedido

Tabela 3.8: Mensagem registrar

### 3.4.2.2 Mensagem registrar

A mensagem Registrar é enviada por um coordenador ao centro de controlo para se dar a conhecer. A Tabela 3.8 descreve a estrutura desta mensagem.

### 3.4.2.3 Mensagem descobrir

A mensagem Descobrir é enviada por um dispositivo ao centro de controlo para seleccionar um coordenador a utilizar no seu processamento. A Tabela 3.9 descreve a estrutura desta mensagem.

A descoberta de um coordenador pode ser feita pelo nome ou pela localização do dispositivo, utilizando o código postal como localização, ou ainda pelo identificador único de um coordenador.

A mensagem Descobrir tem ainda a data de início e de fim em que um dispositivo está disponível para permanecer em serviço.

Na resposta é enviada uma lista de coordenadores a que um dispositivo pode aderir. A escolha será feita pelo dispositivo, enviando a mensagem de adesão ao dispositivo de sua escolha.

Descobrir				
Nome	Tipo		Valor	Descrição
<b>Pedido</b>				
cabecalho				
Tipo de mensagem	Texto	1	D: Descobrir	Tipo de mensagem
Pedido-Resposta	Texto	1	P: Pedido	Sub tipo de mensagem
Versão	Número	1	1-9	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	20		Momento da formatação da mensagem
Número de sequência	Número	4		
<b>Corpo</b>				
Tipo de pesquisa	Texto	1	C: Código Postal I: Identificação única N: Nome (parcial)	
Valor de pesquisa	Texto	40		Código postal, ID ou Nome
<b>Resposta</b>				
cabecalho				
Tipo de mensagem	Texto	1	Igual ao pedido	Tipo de mensagem
Pedido-Resposta	Texto	1	R: Resposta	Sub tipo de mensagem
Versão	Número	1	Igual ao pedido	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	20		Momento da formatação da mensagem
Marca temporal do pedido	DataHora	20		Retirado do pedido
Número de sequência	Número	4		Retirado do pedido
<b>Corpo</b>				
Coordenadores[]				Resultado da pesquisa
ID	Número	11		Identificação única
Nome	Texto	40		Nome
Descrição	Texto	80		Descrição
Código Postal	Número	7		
Estado	Texto	64	A: Ativo I: Inativo	Estado de funcionamento
Ligação	Texto	256		Dados da ligação

Tabela 3.9: Mensagem descobrir

#### 3.4.2.4 Mensagem aderir

A mensagem aderir é enviada pelo dispositivo ao coordenador para solicitar a adesão ao serviço. A Tabela 3.10 descreve a estrutura desta mensagem.

O pedido leva a indicação dos papéis que o dispositivo sabe realizar.

Tem ainda a data de início e de fim em que está disponível para permanecer em serviço.

#### 3.4.2.5 Mensagem convidar

A mensagem Convidar existe para adicionar dispositivos sem capacidades de implementação da interface do sistema *Rain Computing*. O dispositivo é inserido no centro de controlo que envia esta mensagem a um coordenador.

Serve ainda para redirecionar um dispositivo associado a um coordenador para outro, quando este necessita de entrar em manutenção, ou tem um problema grave. A mensagem é enviada pelo centro de controlo, via um coordenador, ao dispositivo para o convidar a aderir ao serviço. O dispositivo pode ignorar a mensagem, continuando associado ao seu coordenador. Caso aceite a mudança de coordenador o

Aderir				
Nome	Tipo		Valor	Descrição
Pedido				
cabecalho				
Tipo de mensagem	Texto	1	A: Aderir	Tipo de mensagem
Pedido-Resposta	Texto	1	P: Pedido	Sub tipo de mensagem
Versão	Número	1	1-9	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	20		Momento da formatação da mensagem
Número de sequência	Número	4		
Corpo				
Nome	Texto	40		Nome
Descrição	Texto	80		Descrição
Ligação	Texto	256		Dados da ligação
Papéis[]	Texto	16		Papéis disponíveis no Dispositivo
Data de início	Data	14		Data de início de serviço
Data de fim	Data	14		Data de fim de serviço
Resposta				
cabecalho				
Tipo de mensagem	Texto	1	Igual ao pedido	Tipo de mensagem
Pedido-Resposta	Texto	1	R: Resposta	Sub tipo de mensagem
Versão	Número	1	Igual ao pedido	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	20		Momento da formatação da mensagem
Marca temporal do pedido	DataHora	20		
Número de sequência	Número	4		
Corpo				
ID	Número	8		Identificação única atribuída
Data de inicio	Data	14		Data de início de serviço
Data de fim	Data	14		Data de fim de serviço

Tabela 3.10: Mensagem aderir

dispositivo deverá enviar a mensagem Aderir. A Tabela 3.11 descreve a estrutura desta mensagem.

#### 3.4.2.6 Mensagem retirar

A mensagem Retirar é enviada pelo dispositivo ao coordenador para solicitar o cancelamento da adesão ao serviço. Esta mensagem apenas apresenta os cabeçalhos, tal como é possível observar na tabela 3.12.

#### 3.4.2.7 Mensagem planear

A mensagem Planear é utilizada pelo centro de controlo para solicitar a execução de um plano a um coordenador.

A mensagem transporta o plano de execução completo bem como a indicação do momento da sua execução.

A Tabela 3.13 descreve a estrutura desta mensagem.

Convidar				
Nome	Tipo		Valor	Descrição
Pedido				
cabecalho				
Tipo de mensagem	Texto	1	C: Convidar	Tipo de mensagem
Pedido-Resposta	Texto	1	P: Pedido	Sub tipo de mensagem
Versão	Número	1	1-9	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	20		Momento da formatação da mensagem
Número de sequência	Número	4		
Corpo				
ID	Texto	40		Identificação única
Nome	Texto	40		Nome
Descrição	Texto	80		Descrição
Tipo de Codificação	Texto	1	J: JSON P: Proprietary T: Plain Text X: XML	Indica o formato dos dados da ordem
Tipo de Comunicação	Texto	1	H: HTTP P: PUB/SUB T: TCP/IP X: Proprietary	Indica o tipo de comunicação
Papéis[]	Texto	16		papéis disponíveis no Coordenador
Ligação	Texto	256		Dados da ligação
Plugin	Texto	128		Classe que implementa o <i>plugin</i>
Resposta				
cabecalho				
Tipo de mensagem	Texto	1	Igual ao pedido	Tipo de mensagem
Pedido-Resposta	Texto	1	R: Resposta	Sub tipo de mensagem
Versão	Número	1	Igual ao pedido	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	20		Momento da formatação da mensagem
Marca temporal do pedido	DataHora	20		Retirado do pedido
Número de sequência	Número	4		Retirado do pedido

Tabela 3.11: Mensagem convidar

Retirar				
Nome	Tipo		Valor	Descrição
Pedido				
cabecalho				
Tipo de mensagem	Texto	1	R: Retirar	Tipo de mensagem
Pedido-Resposta	Texto	1	P: Pedido	Sub tipo de mensagem
Versão	Número	1	1-9	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	20		Momento da formatação da mensagem
Número de sequência	Número	4		
Resposta				
cabecalho				
Tipo de mensagem	Texto	1	Igual ao pedido	Tipo de mensagem
Pedido-Resposta	Texto	1	R: Resposta	Sub tipo de mensagem
Versão	Número	1	Igual ao pedido	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	20		Momento da formatação da mensagem
Marca temporal do pedido	DataHora	20		Retirado do pedido
Número de sequência	Número	4		Retirado do pedido

Tabela 3.12: Mensagem retirar

### 3.4.2.8 Mensagem de contexto

Mensagem enviada pelo coordenador ao dispositivo para posicionar o contexto de processamento inicial do dispositivo referente ao plano de execução.

Planear				
Nome	Tipo		Valor	Descrição
<b>Pedido</b>				
cabecalho				
Tipo de mensagem	Texto	1	P: Planear	Tipo de mensagem
Pedido-Resposta	Texto	1	P: Pedido	Sub tipo de mensagem
Versão	Número	1	1-9	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	20		Momento da formatação da mensagem
Número de sequência	Número	4		
<b>Corpo</b>				
Momento	DataHora	20		Momento de execução do plano
Plano	Texto	...		Plano de execução completo
<b>Resposta</b>				
cabecalho				
Tipo de mensagem	Texto	1	Igual ao pedido	Tipo de mensagem
Pedido-Resposta	Texto	1	R: Resposta	Sub tipo de mensagem
Versão	Número	1	Igual ao pedido	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	20		Momento da formatação da mensagem
Marca temporal do pedido	DataHora	20		Retirado do pedido
Número de sequência	Número	4		Retirado do pedido

Tabela 3.13: Mensagem planear

Contexto				
Nome	Tipo		Valor	Descrição
<b>Pedido</b>				
cabecalho				
Tipo de mensagem	Texto	1	X: Contexto	Tipo de mensagem
Pedido-Resposta	Texto	1	P: Pedido	Sub tipo de mensagem
Versão	Número	1	1-9	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	20		Momento da formatação da mensagem
Número de sequência	Número	4		
<b>Corpo</b>				
Nome papel	Número	24		Nome do papel
Tipo de Codificação	Texto	1	J: JSON P: Proprietary T: Plain Text X: XML	Indica o formato dos dados da ordem
Dados	Texto	...		Dados específicos por serviço/papel
<b>Resposta</b>				
cabecalho				
Tipo de mensagem	Texto	1	Igual ao pedido	Tipo de mensagem
Pedido-Resposta	Texto	1	R: Resposta	Sub tipo de mensagem
Versão	Número	1	Igual ao pedido	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	20		Momento da formatação da mensagem
Marca temporal do pedido	DataHora	20		Retirado do pedido
Número de sequência	Número	4		Retirado do pedido

Tabela 3.14: Mensagem de contexto

A mensagem de Contexto transporta no atributo Dados informação que permite preparar o dispositivo para a execução do plano. Poderão estar incluídos neste conjunto de informação caminhos para ficheiros que o dispositivo tenha que descarregar diretamente.

A tabela 3.14 descreve a estrutura desta mensagem.

### 3.4.2.9 Mensagem ordenar

Ordenar				
Nome	Tipo		Valor	Descrição
Pedido				
cabecalho				
Tipo de mensagem	Texto	1	O: Ordenar	Tipo de mensagem
Pedido-Resposta	Texto	1	P: Pedido	Sub tipo de mensagem
Versão	Número	1	1-9	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	20		Momento da formatação da mensagem
Número de sequência	Número	4		
Corpo				
Nome papel	Número	24		Nome do papel
Momento	Número	8		Momento da execução da ordem
Gestão de erro	Texto	1	A: Abortar C: Continuar E: Executar em erro U: Erro na ordem anterior	Indica o comportamento da ordem em caso de erro
Dados	Texto	...		Dados específicos por serviço/papel
Resposta				
cabecalho				
Tipo de mensagem	Texto	1	Igual ao pedido	Tipo de mensagem
Pedido-Resposta	Texto	1	R: Resposta	Sub tipo de mensagem
Versão	Número	1	Igual ao pedido	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	20		Momento da formatação da mensagem
Marca temporal do pedido	DataHora	20		Retirado do pedido
Número de sequência	Número	4		Retirado do pedido
Corpo				
Dados	Texto	...		Dados específicos por cenário de aplicação/papel

Tabela 3.15: Mensagem ordenar

A mensagem Ordenar é enviada pelo coordenador ao dispositivo para dar uma ordem. A tabela 3.15 descreve a estrutura desta mensagem.

O pedido da mensagem Ordenar também tem um indicador (Gestão de erro) para o dispositivo saber como proceder em caso de ocorrência de erro. Este atributo já foi definido na estrutura do plano, secção 3.4.1.5 e pode assumir os seguintes valores: abortar, continuar, erro e última.

Na resposta da mensagem Ordenar vêm os dados específicos do processamento do pedido, ou seja os resultados da sua execução.

### 3.4.2.10 Mensagem sincronizar

A mensagem Sincronizar é utilizada para sincronizar os relógios dos dispositivos. Também é utilizada para verificar se os dispositivos estão ativos (*heartbeat*). A Tabela 3.16 descreve a estrutura desta mensagem.

A mensagem Sincronizar não necessita de mais informação para além dos cabeçalhos, pois a informação aí presente é suficiente, exceto nos casos onde é necessário acertar o relógio. Sempre que não seja necessário acertar o relógio este campo

Sincronizar				
Nome	Tipo		Valor	Descrição
<b>Pedido</b>				
cabecalho				
Tipo de mensagem	Texto	1	H: Sincronizar	Tipo de mensagem
Pedido-Resposta	Texto	1	P: Pedido	Sub tipo de mensagem
Versão	Número	1	1-9	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	20		Momento da formatação da mensagem
Número de sequência	Número	4		
<b>Corpo</b>				
Sub Tipo de Mensagem	Texto	1	R: Lêr U: Atualizar	Define a operação a efetuar
Diferença	Número	8		Número de milissegundos a somar ao relógio. Pode assumir valores negativos. Apenas para o Sub Tipo "U"
<b>Resposta</b>				
cabecalho				
Tipo de mensagem	Texto	1	Igual ao pedido	Tipo de mensagem
Pedido-Resposta	Texto	1	R: resposta	Sub tipo de mensagem
Versão	Número	1	Igual ao pedido	Versão da mensagem
ID da origem	Número	8		Identificação única da origem
Marca temporal	DataHora	20		Momento da formatação da mensagem
Marca temporal do pedido	DataHora	20		Retirado do pedido
Número de sequência	Número	4		Retirado do pedido
<b>Corpo</b>				
Sub Tipo de Mensagem	Texto	1	R: Lêr U: Atualizar	Define a operação a efetuar. Igual ao pedido

Tabela 3.16: Mensagem sincronizar

deverá ter o valor 0. Quando é necessário avançar com o tempo do relógio do dispositivo este campo assume um valor positivo. Quando é necessário atrasar o relógio do dispositivo este campo assume um valor negativo.

## 3.5 Sumário

Neste capítulo apresentámos uma solução para dar resposta ao problema e desafios descritos no capítulo 1.

A solução é descrita como um modelo de interação entre as partes, com identificação das mensagens trocadas e com uma arquitetura lógica que serviu para implementar o sistema *Rain Computing*, conforme descrito no capítulo 4.

# 4

## Implementação do Sistema *Rain Computing*

Este capítulo descreve a implementação do sistema *Rain Computing*, indicando também as tecnologias utilizadas. O sistema *Rain Computing* é constituído pelos módulos centro de controlo, coordenadores e por uma biblioteca que permite acelerar o desenvolvimento de aplicações *Rain*. Esta biblioteca, para além de poder ser utilizada no desenvolvimento de aplicações concretas, nomeadamente no componente que se executa nos dispositivos envolvidos nessas aplicações, também é utilizada nos restantes módulos do sistema *Rain Computing*, isto é, no centro de controlo e nos coordenadores.

O sistema *Rain Computing* foi desenvolvido em Java, mas cada módulo utilizou técnicas distintas para resolver os desafios encontrados durante a implementação. As técnicas utilizadas são enumeradas na secção correspondente a cada módulo.

Os clientes do sistema *Rain Computing* podem ser desenvolvidos em qualquer linguagem de programação, desde que respeitem o formato das mensagens do sistema.

Neste capítulo começará por ser exposta a arquitetura de implementação do centro de controlo, dos coordenadores e da biblioteca *Rain*. De seguida será discutida a disponibilidade e sincronismo entre dispositivos e coordenadores. Por fim, será apresentado um cenário possível de troca de mensagens.

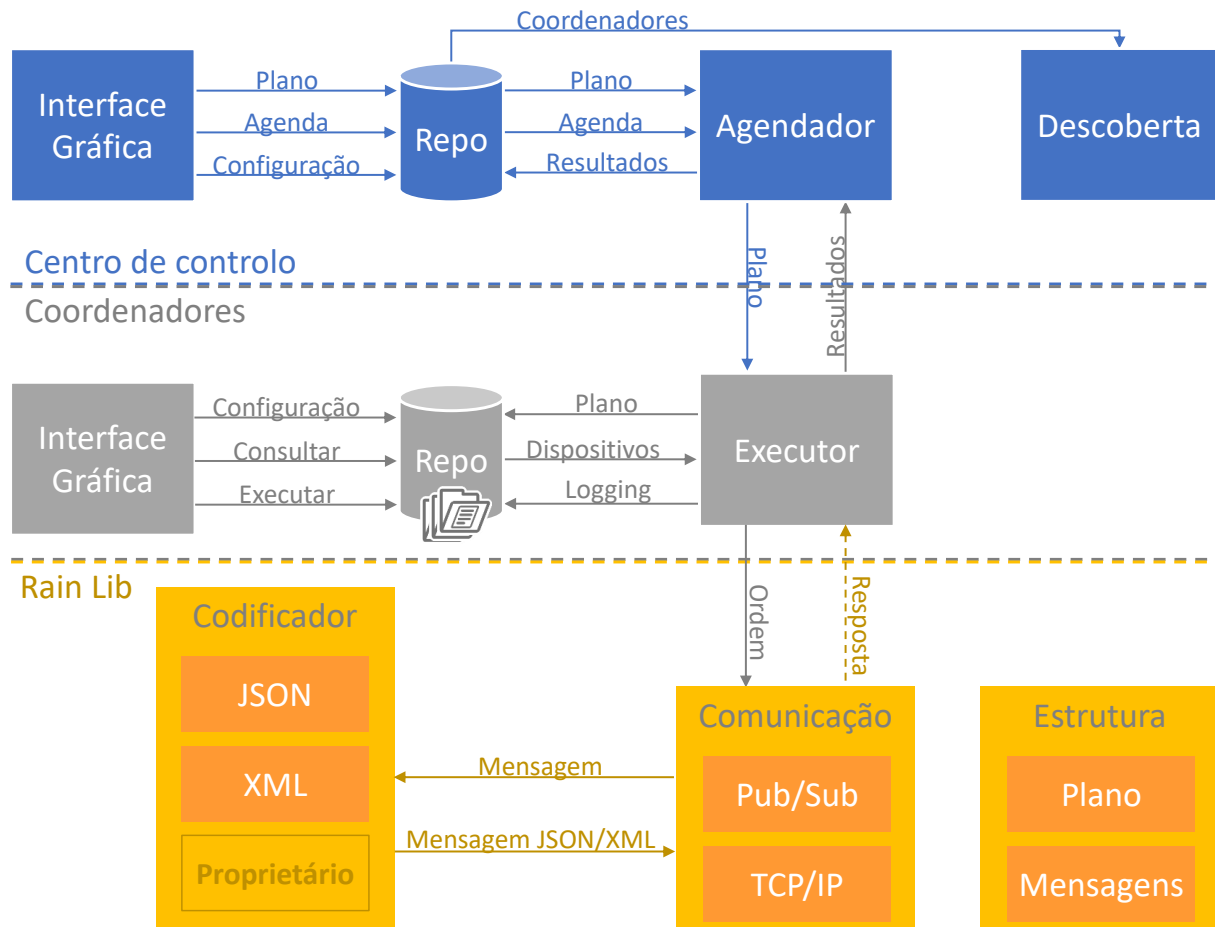


Figura 4.1: Arquitetura geral da implementação do sistema *Rain Computing*

## 4.1 Arquitetura da implementação

Na figura 4.1 ilustra-se a arquitetura global da implementação do sistema *Rain Computing*.

O módulo centro de controlo é composto por um repositório de dados onde é armazenada toda a informação relativa ao centro de controlo, nomeadamente os planos de execução. Este módulo disponibiliza uma interface gráfica que permite gerir as informações no repositório, nomeadamente os planos de execução, os agendamentos e os dados de configuração. No repositório existe também informação relativa aos resultados da execução de planos que o componente *Agendador* recebe dos coordenadores após estes terminarem a execução de planos.

O componente chamado *Agendador* é responsável pelo agendamento dos planos de execução e, conseqüentemente, por desencadear a execução dos mesmos junto dos coordenadores. O centro de controlo também disponibiliza um serviço *REST* de descoberta de coordenadores existentes no sistema *Rain Computing*.

O módulo coordenadores é formado pelos múltiplos coordenadores. Cada coordenador tem um repositório de dados onde guarda a informação sobre os dispositivos com quem interage e sobre os planos a executar. Os planos são-lhe enviados pelo centro de controlo. Existe ainda informação de configuração do coordenador que é criada e mantida através de uma interface de utilizador. Cada coordenador é ainda composto por um componente designado de *Executor* que controla a execução dos planos. O *Executor* obtém no seu repositório o plano e os dispositivos, a utilizar na execução desse plano. O *Executor* comunica com os dispositivos de acordo com as instruções constantes do plano e do papel de cada dispositivo na execução do plano.

No módulo *Rain Lib*, referente aos dispositivos, é disponibilizada a biblioteca *Rain* para ser utilizada pelos dispositivos na implementação das suas aplicações locais. A biblioteca *Rain* está organizada em três grandes componentes: *Estrutura*, *Codificador* e *Comunicação*. O componente *Estrutura* conhece o formato dos planos e de todas as mensagens do sistema. O componente *Codificador* sabe codificar as mensagens no formato adequado para a transmissão. O componente responsável pela *Comunicação* é capaz de proceder à transmissão da informação de acordo com o protocolo de comunicação desejado.

#### 4.1.1 Arquitetura do centro de controlo

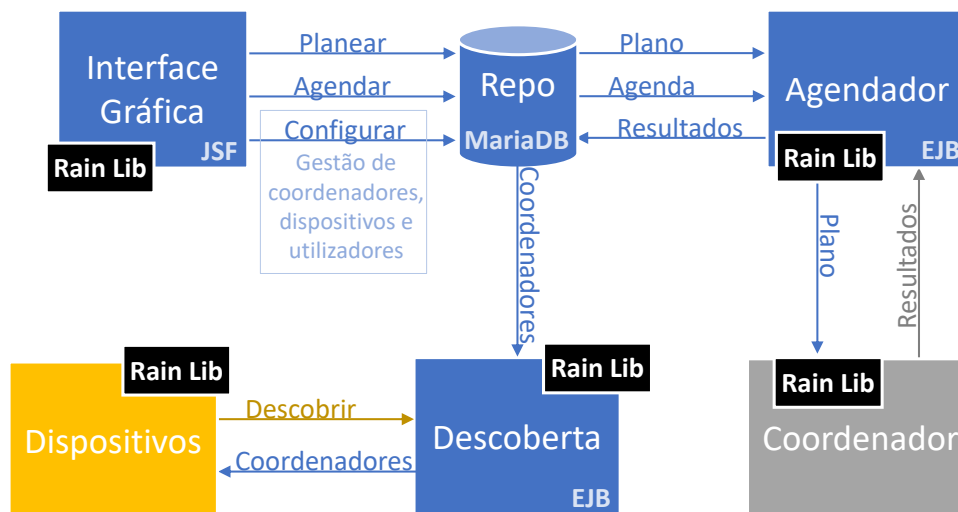


Figura 4.2: Arquitetura do centro de controlo

Na figura 4.2 descreve-se em detalhe a arquitetura da implementação do centro de controlo.

Levando em consideração os requisitos do sistema *Rain Computing*, a implementação do centro de controlo foi feita utilizando a plataforma JavaEE. O JavaEE disponibiliza um ambiente completo, com várias ferramentas, para desenvolvimento de

aplicações, facilitando o desenvolvimento das mesmas, nomeadamente no acesso a bases de dados, serviços *Web*, gestão de concorrência, etc. Outra razão que pesou nesta escolha foi o Java permitir execuções em ambiente heterogéneos, por exemplo em máquinas Linux, Windows, Android, etc.

A interface gráfica *Web* do centro de controlo foi desenvolvida em JSF - Java Server Faces.

Em relação às comunicações com o centro de controlo, tanto em relação aos dispositivos como aos coordenadores, foi implementada uma interface baseada em *sockets* TCP/IP e também uma interface *Publish/Subscriber* suportada no *middleware* Pub/Sub da Google [14]. Os dispositivos comunicam com o centro de controlo unicamente para efetuar a descoberta de um coordenador, já a comunicação entre o centro de controlo e os seus coordenadores é efetuada de uma forma mais frequente.

Em relação aos *sockets* TCP/IP a escolha é bastante imediata pois permite um excelente desempenho nas comunicações entre as partes e são utilizados, desde há muitos anos, em muitas implementações nos mais diversos ambientes. Não se fizeram implementações específicas do JavaEE visto que se utilizou a biblioteca *Rain*, que opera em vários ambientes e não faz uso das facilidades do JavaEE.

Por forma a experimentar um maior desacoplamento entre os módulos do sistema *Rain Computing* e introduzir o processamento assíncrono foi implementada uma interface de comunicações baseado no modelo *publish/subscriber*. Este modelo permite que as mensagens sejam enviadas para um tópico sem saber antecipadamente quem as irá processar. Os subscritores de determinado tópico devem saber processar as mensagens que são colocadas nesse mesmo tópico. Na implementação do modelo Pub/Sub foi escolhido o serviço *cloud*, o Google Pub/Sub. Este serviço disponibiliza bibliotecas para múltiplas linguagens, com transparência à localização, isto é sem necessidade de especificar, de forma explícita, endereço IP e portos. O processamento assíncrono que é permitido pela interface que utiliza o modelo de filas também ajuda na implementação do código. Contrariamente, quando se utiliza os *sockets* TCP/IP o componente de processamento assíncrono teve que ser desenvolvido, no âmbito deste trabalho. Quando é utilizado o Google Pub/Sub, o centro de controlo assume tanto o papel de publicador como de subscritor de mensagens, dependendo da funcionalidade a processar. Tal como aconteceu com os *sockets* TCP/IP, não se fizeram implementações específicas do JavaEE visto que se utilizou a biblioteca *Rain*.

A figura 4.3 ilustra os tópicos do Google Pub/Sub utilizados pelo centro de controlo, no diálogo com os coordenadores e dispositivos. Estes irão ser detalhados nas próximas subsecções.

O repositório de dados do centro de controlo foi implementado num sistema de gestão de base de dados relacional, MariaDB versão 10.4.12. Esta decisão foi feita para permitir a experimentação de um sistema de gestão de base de dados não utilizado antes da realização do trabalho. O modelo relacional é descrito na figura

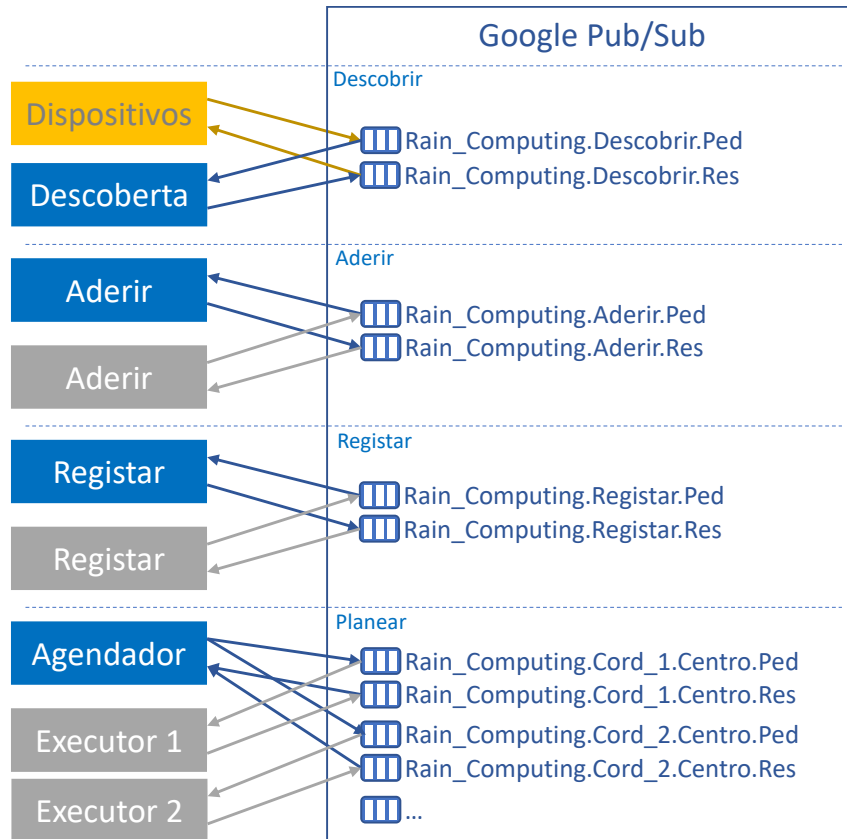


Figura 4.3: Tópicos Pub/Sub do centro de controle

## 4.4.

A tabela  $T\_PLAN$  tem a informação de todos os planos. A coluna  $PLAN\_DATA$  contém o plano, que respeita a estrutura de um plano do sistema *Rain Computing*. A tabela também contém estatísticas sobre a execução dos planos. A tabela  $T\_SCHEDULE$  tem a informação sobre o agendamento da execução dos planos, incluindo a data da próxima execução e o intervalo entre execuções, no caso de se tratar de um agendamento recorrente. Pode ainda ter indicação do coordenador responsável pela execução do plano. Podem existir várias execuções agendadas para um mesmo plano. Por sua vez, a tabela  $T\_PLAN\_RESULTS$  contém informação sobre cada uma das execuções concretas do plano, nomeadamente os resultados obtidos, indicador de sucesso da execução e o coordenador responsável pela execução do plano.

A tabela  $T\_COORDINATOR$  contém informação sobre os coordenadores do sistema *Rain Computing*.

A tabela  $T\_DEVICE$  contém informação sobre os dispositivos do sistema *Rain Computing*.

A tabela  $T\_USER$  contém informação sobre os utilizadores do sistema *Rain Computing*, sejam eles os *especialistas de negócio*, os operadores ou os administradores. A tabela  $T\_USER\_SUBSCRIPTION$  contém informação temporária necessária apenas

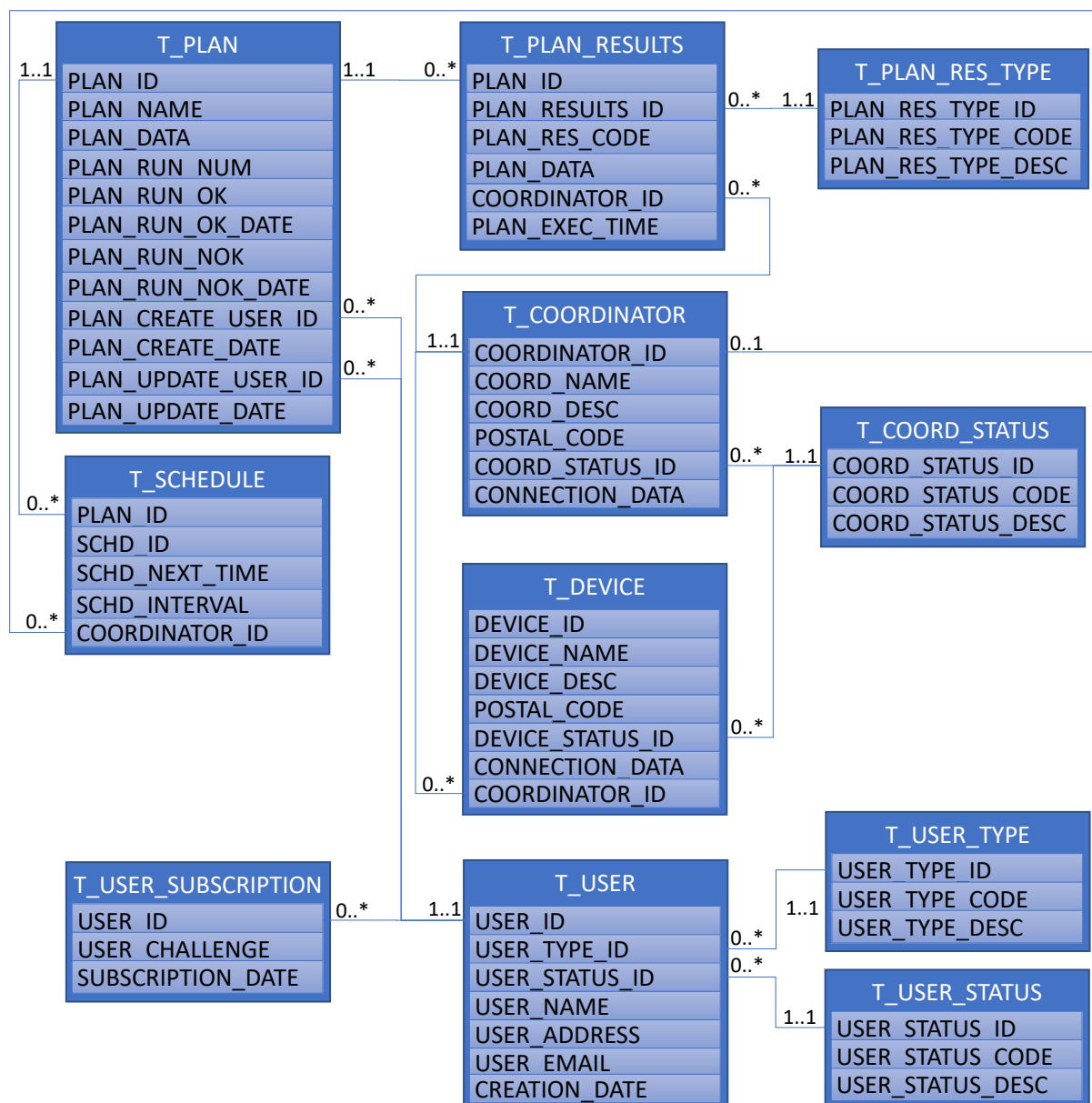


Figura 4.4: Modelo de dados do centro de controlo

na fase de adesão do utilizador ao sistema.

As tabelas *T\_PLAN\_RES\_TYPE*, *T\_COORD\_STATUS*, *T\_USER\_TYPE* e *T\_USER\_STATUS* são tabelas de domínio. As tabelas de domínio suportam dados que raramente são alterados, como por exemplo uma tabela de países.

O centro de controlo disponibiliza as seguintes funcionalidades, que serão descritas nas próximas subsecções:

- Permitir a um *especialista de negócio* a criação de planos de execução;
- Permitir a um *operador* o agendamento da execução de planos de execução;

- Consultar estatísticas/resultados da execução dos planos;
- Permitir a *operadores* a gestão de coordenadores;
- Possibilitar a descoberta de coordenadores;
- Permitir aos operadores a gestão de dispositivos;
- Permitir aos administradores a gestão de utilizadores;
- Desencadear a execução de planos.

#### 4.1.1.1 Permitir a criação de plano de execução concreto

The screenshot shows a web interface for creating a plan. It is divided into several sections:

- Main Vocals:** A form with a text input for 'MainVocals', a dropdown for 'Pauta do voz principal', a number input for '1', and a list of indices: '[1, 10, 21, 31, 43, 61, 67]'. Below it is a 'Lyrics' section with a text input for 'Letra da música', a dropdown for '1', and a list of indices: '[1, 11, 12, 13, 14, 15, 16, 22, 23, 24, 25, 26, 32, 33, 34, 35, 36, 37, 38, 44, 45, 46]'. A blue '+' button is below.
- Recursos:** A section with a 'Nome' label. Below it is a list item 'GNR - Dunas.mp3' with an 'Eliminar' button. A '+ Choose' button is below.
- Contextos:** A table with columns: Nome, Descrição, Espera (mls), Gestao de erro, Codificacao, and Dados. A row is shown with 'Letras e audio', 'Letra e audio desta música', '7000', 'Continua', 'Json', and '{"banda": "GNR", "nome'.
- Ordens:** A table with columns: Nome, Descrição, Momento, Gestao de erro, Codificacao, and Resp. Dados. Three rows are shown:
 

Nome	Descrição	Momento	Gestao de erro	Codificacao	Resp. Dados
Pré-aviso de início de música	São 4 toques para alertar para o eminente começo da m	0	Continua	Json	{'seq': 1, 'tipo': 'Start'}
Início da bateria	São 4 toques que indicam o começo da bateria	0	Continua	Json	{'seq': 1, 'tipo': 'Start'}
Início do baixo	São 4 toques que indicam o começo da baixo	5	Continua	Json	{'seq': 1, 'tipo': 'Start'}

Figura 4.5: Permitir a criação de plano no centro de controlo

Para criar um novo plano no sistema *Rain Computing* é necessário que um utilizador do tipo *especialista de negócio* aceda ao sistema *Rain Computing* através da interface de utilizador disponibilizada pelo centro de controlo. No ecrã apresentado na figura 4.5, o utilizador poderá preencher todos os atributos referentes ao plano que pretende criar, ou simplesmente, alterar.

Chave	Tipo	Valor
band	Simple	GNR
musicName	Simple	Dunas
lyrics	Lista	["Dunas, são como divãs", "Bior"]
audioPath	Lista	["file://10.0.2.2/D\$/Escola/Mest"]

Figura 4.6: Editar atributo *dados*

Para editar o atributo *dados* das partes do plano *contexto*, ou *ordens*, é necessário utilizar o botão que está junto a esses atributos e que apresentará um novo ecrã para edição detalhada do conteúdo desse atributo, que se encontra exemplificado na figura 4.6. Caso o valor do atributo *dados* não seja um texto, encontrará de novo um botão que permite a edição desse atributo, de forma recorrente, até se alcançar o nível onde o valor do atributo é texto. Existem 3 tipos de dados: texto, lista ou mapa.

#### 4.1.1.2 Permitir o agendamento da execução de planos

Aplicacao	Nome
Maestro Digital	GNR - Dunas
Stage Light	GNR - Dunas

Figura 4.7: Permitir o agendamento da execução de planos

Nome do plano	ID	Proxima Execucao	Intervalo de execucao	Coordenador
Maestro Digital - GNR - Dunas	1	2020-05-06 12:13:14.0	1 Dia	TONY2
Maestro Digital - GNR - Dunas	2	2020-05-08 22:23:24.0	2 Dia	TONY2
Maestro Digital - GNR - Dunas	3	2020-05-09 17:03:14.0	1 Dia	TONY2

Figura 4.8: Agendar a execução de um plano

Para agendar, ou reagendar, um plano no sistema *Rain Computing* é necessário que um utilizador do tipo operador acesse ao sistema *Rain Computing* através da interface de utilizador disponibilizada pelo centro de controlo, descrita na figura 4.7.

O operador seleciona o plano pretendido e é apresentado um ecrã, como o da figura 4.8 onde é possível indicar o calendário de execução do plano, incluindo dias, horas e periodicidade de execução.

#### 4.1.1.3 Consultar estatísticas/resultados da execução dos planos

É apresentada, ao utilizador do tipo operador, uma lista dos planos conhecidos. Depois de selecionar um plano é-lhe apresentado um ecrã onde é possível visualizar as estatísticas/resultados globais do plano. É ainda apresentada uma lista com as diversas execuções do plano permitindo selecionar uma execução concreta do plano e visualizar as estatísticas/resultados da execução concreta.

#### 4.1.1.4 Permitir a gestão de coordenadores

Aos utilizadores do tipo operador é-lhes permitido gerir os coordenadores que existem no sistema, eliminando, inativando ou ativando coordenadores.

#### 4.1.1.5 Possibilitar a descoberta de coordenadores

O centro de controlo do sistema *Rain Computing* disponibiliza um serviço *REST* para que os dispositivos possam encontrar um coordenador.

Quando os dispositivos utilizam a biblioteca *Rain* sabem chegar ao centro de controlo, pois esta biblioteca conhece a localização do centro de controlo.

O centro de controlo disponibiliza para este efeito um serviço *REST* e um tópico do Google Pub/Sub. A aplicação presente no dispositivo decide o método a utilizar na comunicação com o centro de controlo.

As rotas do serviço *REST* são:

- GET <http://raincmp.com/descoberta/coordenadores?codigoPostal=1234123>
- GET <http://raincmp.com/descoberta/coordenadoresPorNome?nome=Bar>
- GET <http://raincmp.com/descoberta/coordenador?id=2>

Estes serviços geram conteúdo do tipo "*Content-Type:application/json*".

Quando é utilizado o Google Pub/Sub, o centro de controlo assume o papel de subscritor de mensagens, encontrando-se à escuta da mensagem de *Descobrir*, a ser enviada pelo dispositivo, no tópico *Rain\_Computing.Descobrir.Ped*. A resposta é enviada para o tópico *Rain\_Computing.Descobrir.Res*, assumindo o centro de controlo neste caso o papel de publicador de mensagens.

#### 4.1.1.6 Permitir a gestão de dispositivos

Quando um dispositivo não tem capacidades para efetuar o registo através do serviço de descoberta de coordenadores, pois não conhece o sistema *Rain Computing*, é então necessário registar o dispositivo no centro de controlo de forma manual.

Um utilizador do tipo operador, acedendo através da interface gráfica disponibilizada pelo centro de controlo, dá entrada às características do dispositivo, incluindo as configurações de rede. Nesta fase é possível selecionar o coordenador desejado.

Por forma a dar a conhecer ao coordenador o novo dispositivo, o centro de controlo tem que informar um coordenador sobre a adesão do dispositivo.

Quando é utilizado o Google Pub/Sub na comunicação entre o centro de controlo e o coordenador, o centro de controlo assume o papel de publicador de mensagens, publicando a mensagem *Aderir* no tópico *Rain\_Computing.Coord.9.Ped* (onde o 9 representa a identificação do coordenador) para ser consumida pelo respetivo coordenador do dispositivo. A resposta do coordenador é recebida no tópico *Rain\_Computing.Coord.9.Res*.

#### 4.1.1.7 Permitir a gestão de utilizadores

Já nos deparamos com os três tipos de utilizadores do sistema: administrador, operador e *especialista de negócio*. Para gerir as contas de utilizador existe uma interface de utilizador que é acedida pelos administradores do sistema, para que seja possível gerir utilizadores. É possível criar novos utilizadores, eliminar utilizadores, alterar o papel do utilizador ou suspender o utilizador.

#### 4.1.1.8 Desencadear a execução de planos

A execução de planos é gerida por um componente do centro de controlo chamado *Agendador*. Este tem uma tarefa muito simples: a construção de uma coleção de temporizadores Java (*Timer*) para desencadear a execução de cada um dos planos. O *Timer* é calendarizado tendo em conta o tempo de *setup* do mesmo, indicado nos contextos do plano. Este tempo é multiplicado por dois, para permitir que o dispositivo e o coordenador se preparem em tempo útil para a execução do plano.

Para obter informação necessária à execução desta tarefa, consulta a tabela de planos e o calendário de execução de cada um dos planos.

Do ponto de vista do centro de controlo, a execução de um plano consiste na seleção de um coordenador adequado, e com disponibilidade, para a execução do plano e o envio do plano a esse mesmo coordenador. O coordenador pode ser atribuído manualmente no momento da criação da calendarização da execução do plano.

O coordenador adequado à execução do plano é aquele que, possuindo dispositivos capazes de interpretar todos os papéis necessários à execução do plano, apresenta o melhor desempenho na execução do plano, ou seja, tem a taxa de erros mais baixa. Este desempenho é medido através das estatísticas de execução do plano, obtidas através da interpretação dos resultados da execução do plano.

O pseudo-código que se apresenta de seguida descreve o funcionamento do *Agendador*:

1. Verificar se o plano tem um coordenador fixo para execução
  - 1.1 Caso tenha, utiliza esse coordenador
2. Obtém uma lista dos coordenadores que têm dispositivos capazes de interpretar todos os papéis do plano
3. Seleciona o coordenador que tem o melhor rácio entre o número de execuções do plano e o número de sucessos na execução do plano
  - 3.1 Em caso de empate utiliza o coordenador com o menor número de execuções do plano
    - 3.1.1 Em caso de empate utiliza o coordenador com a identificação mais baixa

Quando o coordenador termina a execução do plano envia para o centro de controlo os resultados de execução do mesmo.

#### 4.1.1.9 Disponibilizar recursos necessários à execução de planos

A execução de alguns dos planos requer a existência de recursos que são guardados fora do plano de execução, tais como: fotos vídeos, áudio, etc... Quando um

coordenador recebe um plano para executar, verifica se este necessita de recursos externos ao plano e solicita-os ao centro de controlo. Para esse efeito o centro de controlo disponibiliza um serviço *REST*. A rota desse serviço é:

- GET `http://raincmp.com/ficheiro/obtem?aplicacao=Maestro%20Digital&nomePlano=aaaaaa&nome=musicax.mp3`

O serviço gera uma resposta com o formato *Content-Disposition: attachment; filename="musicax.mp3"*, contendo o recurso representado numa cadeia de bytes.

### 4.1.2 Arquitetura do coordenador

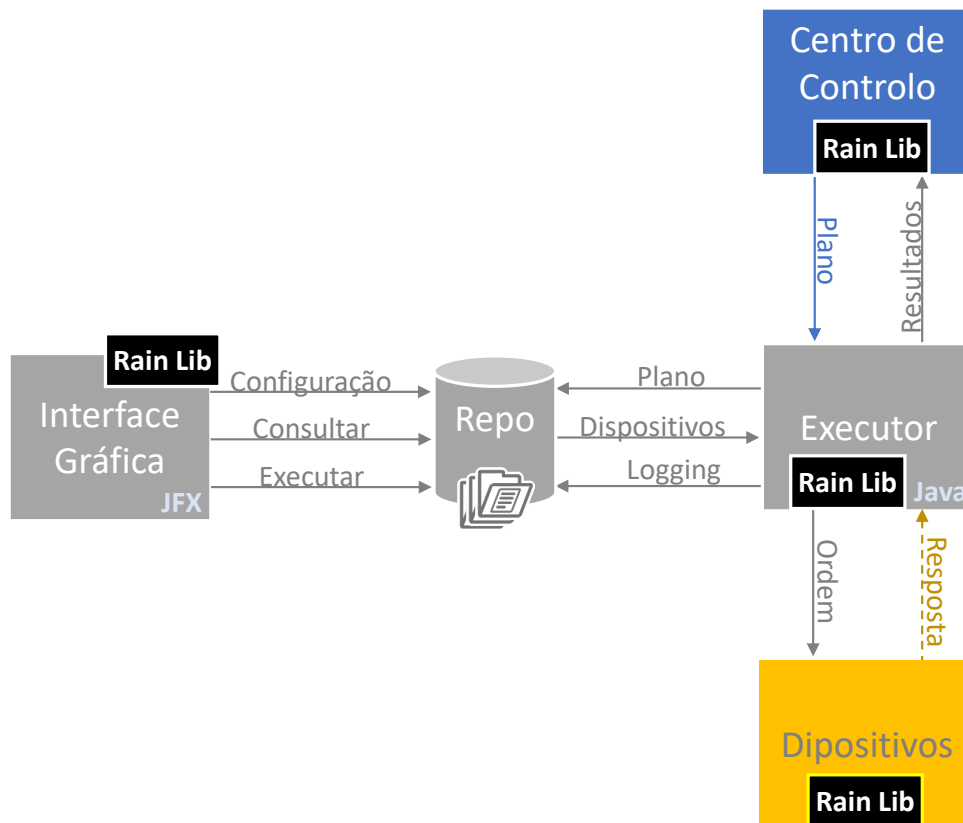


Figura 4.9: Arquitetura do coordenador

Na figura 4.9 descreve-se a arquitetura da implementação do coordenador.

O coordenador é o módulo do sistema *Rain Computing* responsável pela execução dos planos concretos. O seu repositório de dados está assente no sistema de diretórios e ficheiros de configuração no formato *json* e está descrito na figura 4.10. O repositório do coordenador é persistido na própria máquina onde é executado o coordenador, existindo um repositório por cada coordenador.

Existem três tipos de ficheiros de configuração no coordenador. Os parâmetros gerais do coordenador são guardados no ficheiro *CoordenadorConfig*. O ficheiro

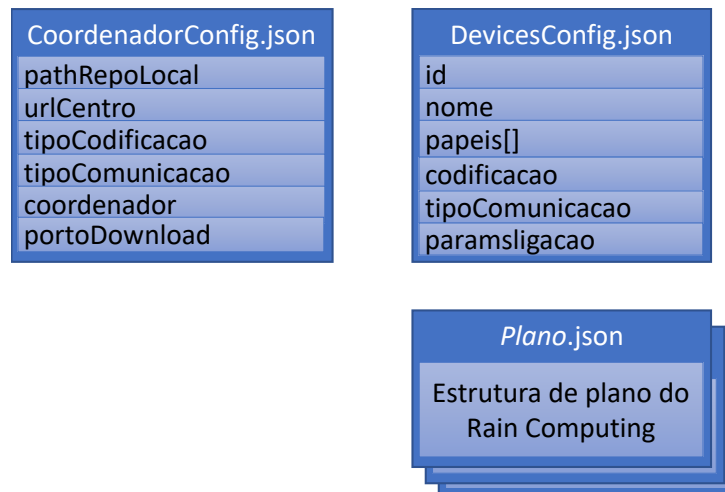


Figura 4.10: Modelo de dados do coordenador

*DevicesConfig* mantém uma lista dos dispositivos que lhe enviaram a mensagem *Aderir* ou que foram inseridos na interface de utilizador do centro de controlo e lhe foram posteriormente atribuídos. O coordenador também guarda os planos de execução em ficheiros específicos por plano, cujo nome do ficheiro é o nome do próprio plano de execução.

Em relação às comunicações, dos coordenadores com o centro de controlo e com os dispositivos, foram utilizadas as mesmas interfaces já referidas na arquitetura do centro de controlo: *sockets* TCP/IP e o Google Pub/Sub.

A figura 4.11 ilustra os tópicos do Google Pub/Sub utilizados pelo centro de controlo, no diálogo com os coordenadores e dispositivos. Estes irão ser detalhados nas próximas subsecções.

O coordenador disponibiliza as funcionalidades cuja implementação é descrita nas próximas subsecções:

- Permitir a adesão de dispositivos;
- Permitir a receção de planos do centro de controlo;
- Executar planos;
- Disponibilizar uma interface de utilizador para consulta de dispositivos e de planos;
- Registrar informação de *logging* durante a execução de planos;
- Disponibilizar recursos necessários à execução de planos.

#### 4.1.2.1 Permitir a adesão de dispositivos

O coordenador tem um serviço, sempre ativo, que sabe processar os pedidos de adesão dos dispositivos. Os pedidos de adesão dos dispositivos trazem indicação

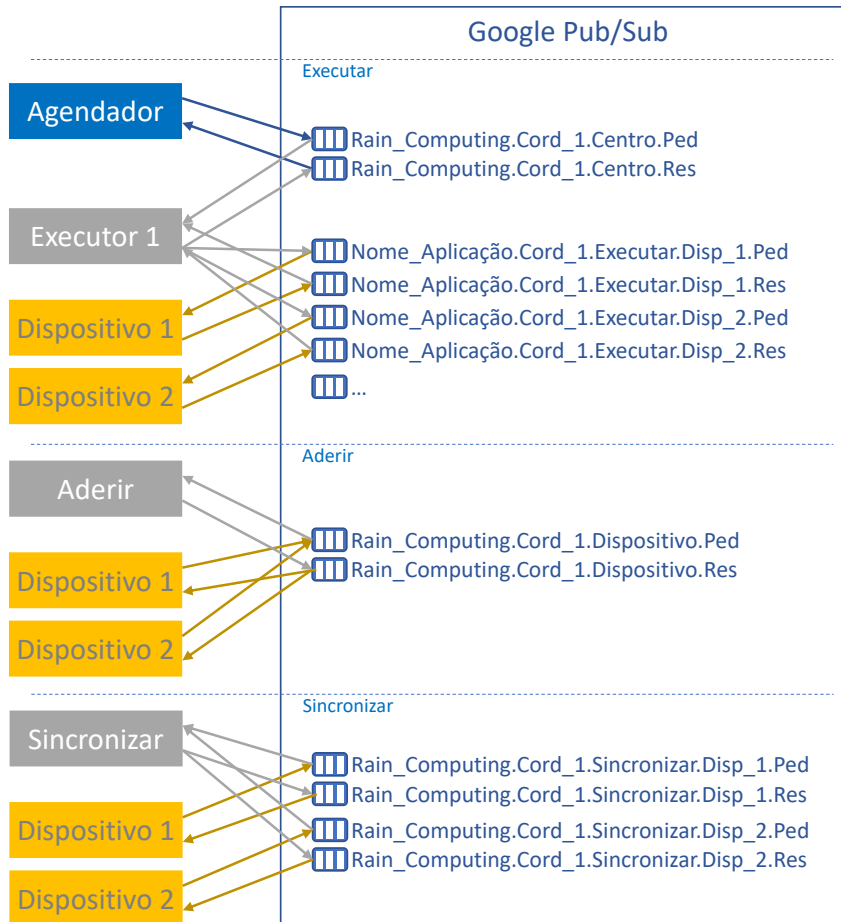


Figura 4.11: Tópicos Pub/Sub do coordenador

das características dos mesmos, incluindo as suas capacidades de processamento e a sua configuração na rede de comunicações.

O coordenador quando recebe o pedido de adesão por parte de um dispositivo regista-o na sua lista de dispositivos e responde ao dispositivo confirmando a sua adesão.

#### 4.1.2.2 Permitir a receção de planos do centro de controlo

Quando o coordenador recebe um plano do centro de controlo, armazena-o no repositório de dados, passando de seguida à execução do plano.

#### 4.1.2.3 Executar planos

Para executar planos o coordenador verifica que papéis existem no plano e seleciona os dispositivos que sabem executar cada um dos papéis do plano.

A lista de dispositivos selecionados é entregue, juntamente com o plano, a um componente do coordenador chamada *executor*, que é quem tem a responsabilidade de executar o plano.

O *executor* estabelece a comunicação com os dispositivos, consoante as características de cada um. Podem existir dispositivos que assumam mais do que um papel na execução do plano, mas as ligações são criadas por dispositivo e não por papel do plano.

Depois de estabelecida a ligação aos dispositivos, o *executor*, começa por enviar o *contexto* aos mesmos. O *executor* fica à espera das respostas dos dispositivos até ter todas as respostas ou ter sido atingido o intervalo para estabelecimento do *contexto*. Caso esse intervalo seja atingido sem que todas as respostas tenham chegado, procede de acordo com o indicado no *contexto* em relação à gestão de erro.

Após ter sido concluído o estabelecimento do *contexto*, o *executor* passa ao envio das *ordens*. O *executor* cria uma lista ordenada pelo momento de envio das *ordens*, calculando o intervalo de tempo entre o envio de cada uma das *ordens* tomando em consideração as latências de cada dispositivo. Após terminada esta preparação da execução do plano o *executor* começa a enviar as mensagens de *Ordem* para os dispositivos.

As respostas dos dispositivos, no caso de existirem, são processadas de forma assíncrona, não interferindo com o envio das *ordens* aos dispositivos.

O pseudo-código que se apresenta de seguida descreve o funcionamento do *executor*:

1. Para cada papel do plano seleciona um dispositivo para o executar
  - 1.1 Caso não encontre um dispositivo termina a execução
2. Por cada um dos papéis do plano
  - 2.1 Por cada um dos dispositivos que o vai executar esse papel
    - 2.1.1 Enviar o contexto para o dispositivo
3. Espera pelas respostas de todos os dispositivos até atingir o *Tempo de espera*, ou receber todas as respostas
  - 3.1 Caso não tenha recebido todas as respostas, procede de acordo com o indicado no atributo *Gestão de erro*
4. Ordena as ordens pelo momento de envio ao dispositivo
5. Ciclo até enviar todas as ordens
  - 5.1 Espera
    - 5.1.1  $\text{tempoEspera} = \text{Ordem}(i).\text{tempoEnvio} - \text{Ordem}(i-1).\text{tempoEnvio}$
  - 5.2 Envia ordem (a resposta é processada de forma assíncrona)

No final da execução de todas as ordens, ou quando o processo é cancelado devido a um erro de processamento, o coordenador envia os resultados consolidados da execução do plano ao centro de controlo.

Quando é utilizado o Google Pub/Sub na comunicação entre o coordenador e o dispositivo, o coordenador assume o papel de publicador de mensagens, publicando a mensagem *Ordem*, ou *Contexto*, no tópico *Rain\_Computing.Coord\_9.Executar.Disp\_1.Ped* (onde o 9 representa a identificação do coordenador e o 1 a identificação do dispositivo) para ser consumida pelo respetivo dispositivo. A resposta do dispositivo é recebida no tópico *Rain\_Computing.Coord\_9.Executar.Disp\_1.Res*.

#### 4.1.2.4 Disponibilizar uma interface de utilizador para consulta de dispositivos e de planos

Figura 4.12: Configuração geral do coordenador

Figura 4.13: Gestão de dispositivos no coordenador

O coordenador do sistema *Rain Computing* também disponibiliza uma interface de utilizador, onde é possível gerir configurações locais do coordenador, figura 4.12, gerir os dispositivos, figura 4.13, consultar os planos armazenados localmente e no centro de controlo.

A interface gráfica do coordenador foi implementada em JFX, o sucessor do *swing* do Java, que apresenta uma plataforma de implementação bastante amigável.

#### 4.1.2.5 Registrar a execução de planos *Logging*

Todas as atividades do coordenador, nomeadamente quando incluem a troca de mensagens, são registadas em log. Foi utilizada a *framework* de *logging* slf4j e a sua implementação log4j2.

Este registo serve para o apoio na resolução de problemas, mas também como meio de validação dos resultados obtidos.

#### 4.1.3 Biblioteca *Rain*

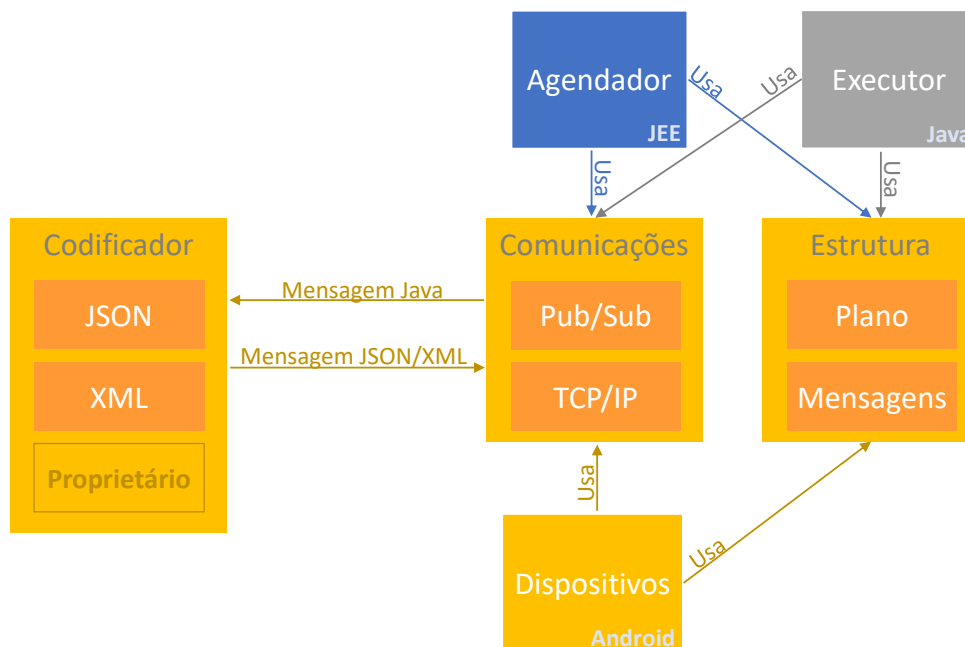


Figura 4.14: Composição da biblioteca *Rain*

Na figura 4.14 descreve-se a composição da biblioteca *Rain*.

O sistema *Rain Computing* disponibiliza uma biblioteca, conhecida como biblioteca *Rain*, ou abreviadamente por *Rain Lib*, onde estão definidas a estrutura do plano de execução e as estruturas das mensagens trocadas entre os três módulos do sistema: centro de controlo, coordenador e dispositivos.

A biblioteca *Rain* disponibiliza ainda classes auxiliares de comunicação e codificadores de mensagens, que são utilizados pelos módulos do sistema *Rain Computing* e que também podem ser utilizadas pelas aplicações instaladas nos dispositivos. A biblioteca *Rain* disponibiliza os componentes descritas nas próximas subsecções: estrutura do plano; estrutura das mensagens; comunicação; interfaces de extensão (codificação de mensagens; comunicação).

#### 4.1.3.1 Estrutura do plano

A estrutura do plano está definida num conjunto de classes Java que podem ser utilizadas para criar um objeto Java a partir de um plano representado em texto, seja ele codificado em JSON ou XML, ou obter a representação em texto de um plano materializado num objeto Java.

A classe que representa o plano é a classe *RainPlano*. Esta contém toda a informação do plano, tal como indicado na secção 3.4.1.

#### 4.1.3.2 Estrutura das mensagens

A estrutura de todas as mensagens do serviço *Rain Computing* também está definida num conjunto de classes Java que oferecem as mesmas facilidades que a da estrutura do plano. As classes que representam as mensagens têm seu nome definido pela estrutura *RainMsg*<Nome da Mensagem><Ped/Res>, onde <Ped/Res> indica se é um pedido ou uma resposta e o <Nome da Mensagem> assume um dos valores da seguinte lista:

- Aderir;
- Contexto;
- Convidar;
- Descobrir;
- Erro;
- Ordenar;
- Planear;
- Registrar;
- Resultado;
- Retirar;
- Sincronizar.

Estas classes, ao contrário da classe principal do plano, são apenas contentores de dados, é o codificador existente no componente responsável pela comunicação que as transforma no texto da mensagem a enviar, ou num objeto Java "serializado" a partir da mensagem recebida.

#### 4.1.3.3 Comunicação

Para facilitar a troca de mensagens do sistema *Rain Computing*, a biblioteca *Rain* disponibiliza classes Java que permitem efetuar o envio e receção de mensagens com um mínimo esforço de desenvolvimento. A biblioteca *Rain* disponibiliza os seguintes métodos de comunicação: *sockets* TCP/IP e Google Pub/Sub.

A figura 4.15 descreve a utilização da biblioteca *Rain*.

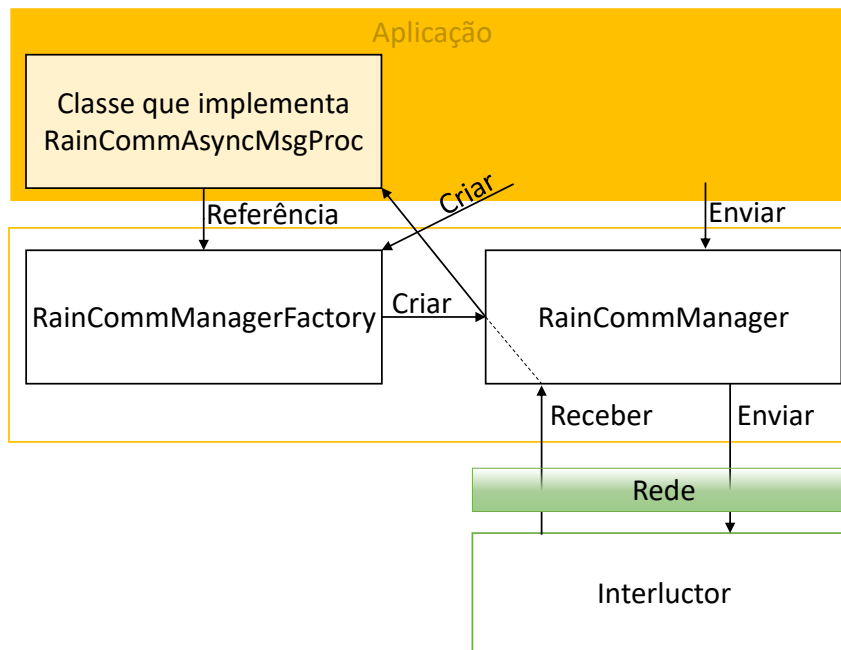


Figura 4.15: Utilização da biblioteca *Rain*

Existe uma fábrica de gestores de ligações chamada *RainCommManagerFactory* que recebe informação sobre o tipo de comunicação a efetuar (TCP/IP, Google Pub/Sub), o tipo de codificação das mensagens (*JSON*, *XML*), a identificação local (quem está a criar a fábrica), a identificação remota (do interlocutor remoto da comunicação) e, caso exista, a classe de negócio que irá processar as mensagens recebidas. Com esta informação a fábrica de gestores de comunicações é capaz de criar um gestor de comunicações. O gestor de comunicações, por sua vez, encarrega-se de gerir a comunicação com o interlocutor remoto, convertendo as mensagens de, e para, o formato desejado, construindo os cabeçalhos das mensagens no momento do envio. Quando recebe uma mensagem, invoca a classe de negócio que lhe foi previamente indicada para processar a mensagem. Esta classe de negócio tem que implementar a interface *RainCommAsyncMsgProc* para permitir a interação com o *Rain Computing*.

O componente de auxílio à comunicação utiliza um outro componente da biblioteca *Rain*, chamada de "codificador" que sabe traduzir as classes que representam as mensagens nos formatos JSON ou XML.

#### 4.1.3.4 Interfaces de extensão de protocolos

Dado que existe uma grande multiplicidade de dispositivos e que cada um deles poderá ter um formato de mensagens específicas e mesmo utilizar diferentes técnicas de comunicação, foi implementado um mecanismo que permite acrescentar novos protocolos aos componentes de codificação de mensagens e de comunicação. Com esta abertura, pretende-se abarcar o maior número possível de dispositivos.

Quando um novo dispositivo que implementa um protocolo diferente dos disponibilizados pelo *Rain Computing* quer participar no sistema *Rain Computing*, o fabricante do dispositivo deverá desenvolver um conjunto de classes Java que implementem as interfaces criadas para a extensão de protocolos. Esta implementação é consolidada num JAR que será entregue ao sistema *Rain Computing* e, após validação, instalado na *classpath* dos coordenadores, permitindo-lhes interagir com o dispositivo. A validação do JAR é feita através de análise ao código e de testes em ambiente não produtivo.

#### 4.1.3.5 Codificação de mensagens

Para acrescentar um novo protocolo ao componente de codificação de mensagens é necessário criar uma classe que implemente a interface *RainCodificador* e que seja anotada com a anotação Java *RainCodificadorParser*.

A anotação serve para que o coordenador consiga encontrar durante o arranque todos os "codificadores" existentes.

Os "codificadores" JSON e XML, que estão disponíveis por omissão na biblioteca *Rain*, respeitam esta interface e têm a anotação *RainCodificadorParser*, tal como os "codificadores" proprietários.

#### 4.1.3.6 Comunicação

Para estender o componente de comunicação da biblioteca *Rain* é necessário criar uma classe que implemente a interface *RainCommManager* e que seja anotada com a anotação Java *RainCommManagerAnnot*. Por omissão são disponibilizados gestores de comunicação para *sockets TCP/IP* e *Google Pub/Sub*.

A interface tem diversos métodos que permitem enviar uma mensagem, receber uma mensagem, conversar, ou seja, enviar uma mensagem e receber a sua resposta, e um método para terminar a ligação.

A anotação serve para que o coordenador consiga encontrar durante o arranque todos os gestores de comunicações existentes.

Os gestores de comunicações que estão disponíveis por omissão respeitam esta interface e têm a anotação *RainCommManagerAnnot*, tal como os gestores de comunicações proprietários.

#### 4.1.3.7 Exemplo de utilização da biblioteca *Rain*

A aplicação DevApp dos dispositivos que pretende utilizar a biblioteca *Rain* têm que efetuar os seguintes passos:

- Estender a classe gestora das ligações *RainCommPubSubManager*:

Apenas para aplicações Android. Esta extensão é necessária para que as bibliotecas de acesso ao serviço Google Pub/Sub utilizadas na implementação consigam encontrar o seu ficheiro de configuração, processo que é feito de outra forma no Android;

- Criar uma instância da classe gestora das ligações *RainCommPubSubAndroidManager*:

Apenas é necessário criar uma instância para toda a aplicação. Apesar disso, por motivos de gestão de recursos da aplicação, essa instância apenas é criada quando se vai realmente comunicar, quer seja no momento de adesão ou de execução de um plano;

```
RainCommPubSubAndroidManager comm = new
RainCommPubSubAndroidManager(AppConfigEnt.SUBSCRIPTION_SESSION_ID,
prefEnt.getCoordinator().getId(), 0, this, keys, getApplicationContext(),
Codificacao.valueOf(prefEnt.getCodificacao().toString()));
```

- Criar a mensagem a enviar:

A classe que representa a mensagem é disponibilizada pela biblioteca *Rain*. No caso da adesão a classe *RainMsgAderirPed* representa a mensagem a enviar, pelo dispositivo ao coordenador, para efetuar a adesão ao coordenador;

```
RainMsgAderirPed aderirPedido = new RainMsgAderirPed(prefEnt.getName(),
prefEnt.getDesc(), papeis, new Date(), RainCommPubSubAndroidManager.getMaxDate());
```

- Enviar mensagem ao coordenador:

Para enviar a mensagem basta invocar um método da classe gestora de comunicações. Caso se pretenda aguardar pela resposta utiliza-se o método *converse*, caso contrário o *send*;

```
RainMsgAderirRes aderirResposta = (RainMsgAderirRes) comm.converse(aderirPedido,
null, 5000);
```

- Estender a interface *RainCommAsyncMsgProc*:

Para receber mensagens do coordenador é necessário implementar um processador de pedidos que estenda a interface *RainCommAsyncMsgProc*. Esta interface disponibiliza o método *nova* que é invocado para entregar as mensagens recebidas do coordenador. O processador de pedidos é associado ao gestor de ligações logo na instanciação da classe gestor de ligações;

```
@Override
public boolean nova(String idMensagem, Map<String, String> atribs, RainMensagem
mensagem) { ... }
```

- Enviar resposta ao pedido do coordenador:

Para enviar uma resposta a um pedido recebido do coordenador basta invocar o método *send* da classe gestora de comunicações, passando como parâmetro o objecto Java que representa a mensagem de resposta;

```
RainMsgContextualizarRes msg = new RainMsgContextualizarRes();  
comm.send(msg, null);
```

- Terminar a comunicação:

Para terminar a ligação ao coordenador é necessário invocar o método *close* da classe gestora de comunicações.

```
comm.close();
```

## 4.2 Disponibilidade e sincronismo dos dispositivos e coordenadores

Existiram vários desafios na implementação do trabalho, mas os dois mais relevantes: i) foram garantir que todos os dispositivos necessários à execução de um plano irão estar disponíveis no momento da sua execução; ii) que efetuam a execução de forma sincronizada.

O primeiro desafio foi ultrapassado com recurso a mensagens de *heartbeat*, enviadas do coordenador para os dispositivos. Na implementação da mensagem de *heartbeat* foi utilizada a mensagem *Sincronizar*, que também é utilizada na sincronização dos relógios dos dispositivos.

O segundo desafio envolveu uma técnica de ajuste de relógio dos dispositivos, efetuada de forma regular para garantir a continuidade de sincronismo. Para o efeito, a metodologia descrita no *standard* PTP - *Precision Time Protocol* [25] serviu de inspiração. Muito genericamente, esta metodologia consiste em ter um dispositivo que funciona como relógio mestre e envia mensagens de sincronismo (*Sync*), tipicamente de dois em dois segundos, aos seus relógios subordinados. Estas mensagens funcionam como *heartbeat* para que os relógios subordinados saibam que o relógio mestre está operacional. Estes respondem-lhe, por norma, apenas de um em um minuto, com a mensagem de intervalo (*Delay\_Req*) para calcular a diferença dos relógios. Estas duas mensagens levam quatro *timestamps* que são utilizados para calcular as diferenças dos relógios. Após os cálculos das latências, o relógio mestre responde à mensagem de intervalo com uma mensagem (*Delay\_Resp*) onde é enviada a diferença dos relógios calculada.

Neste caso o relógio mestre do protocolo PTP será o relógio do coordenador e os relógios subordinados os relógios dos dispositivos ligados ao coordenador. Ao contrário, não sendo efetuada a eleição do relógio mestre, ao contrário do que acontece

no PTP. O coordenador começa por enviar uma mensagem *Sincronizar* ao dispositivo e o dispositivo responde de imediato. É assumido que o tempo de receção do pedido é muito próximo do tempo de envio da resposta, visto que esta mensagem não requer qualquer processamento aplicacional. O coordenador calcula o tempo decorrido desde o envio da pedido até à receção da resposta e assume que a latência entre o coordenador e o dispositivo é metade desse valor. Esse valor é muito importante pois será utilizado no envio das *ordens* tentando obter o maior sincronismo possível na execução das mesmas. É então comparado o tempo do envio da resposta do dispositivo com o tempo da receção da mesma pelo coordenador e, caso os relógios estejam certos, a latência calculada será igual à diferença dos dois tempos comparados.

Dado que as respostas dos dispositivos são processadas de forma assíncrona não é necessário calcular a latência da comunicação na direcção dos dispositivos para o coordenador.

### 4.3 Exemplo de um cenário de troca de mensagens

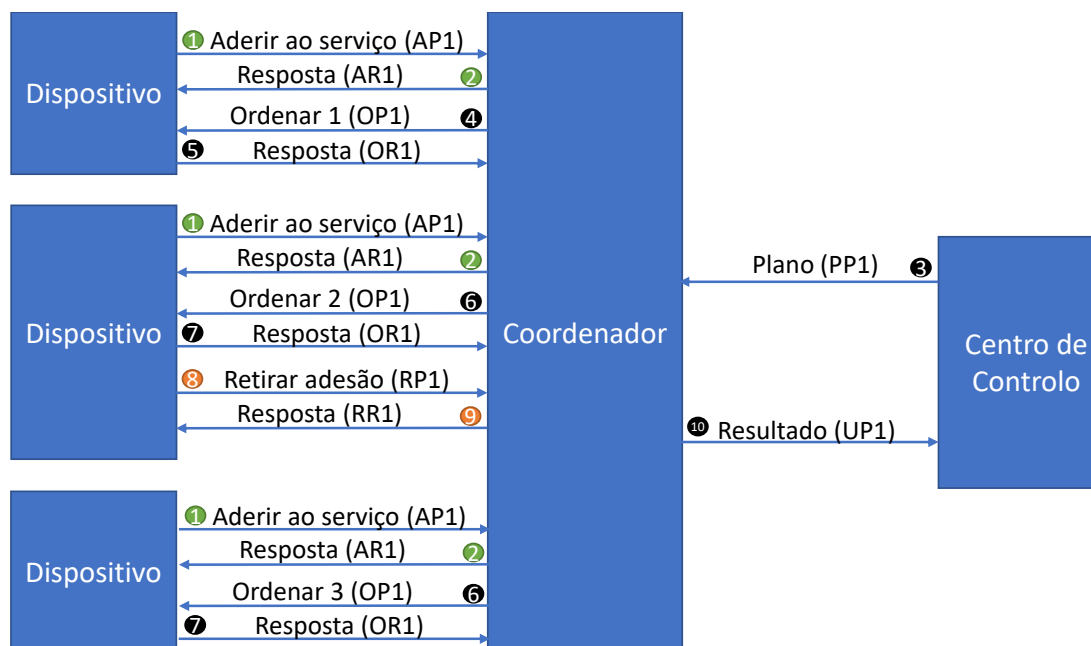


Figura 4.16: Exemplo de troca de mensagens *Rain*

Existem múltiplos cenários possíveis de interação entre as várias partes do sistema. Na figura 4.16 são mencionadas apenas algumas mensagens, que se consideram relevantes para exemplificar o cenário.

Para melhor interpretação da figura 4.16 é necessário ter presente como é identificado o tipo de mensagem. Este é composto pelos seguintes três campos:

- Tipo de mensagem:

- **A:** Aderir;
- **R:** Retirar;
- **P:** Planear;
- **O:** Ordenar;
- **U:** Resultado;
- Pedido/Resposta/Erro:
  - **P:** Pedido;
  - **R:** Resposta;
- Versão: um algarismo com início em 1;

Por exemplo, uma mensagem *Ordem* enviada pelo coordenador ao dispositivo é identificada por "OP1".

Descrição da figura 4.16:

- 1: em primeiro lugar os dispositivos enviam mensagens de adesão a um coordenador, indicando que capacidades têm, ou seja, que papéis são capazes de realizar;
- 2: o coordenador responde aos dispositivos aceitando os pedidos de adesão;
- 3: o centro de controlo faz chegar aos coordenadores os planos de execução que eles necessitam de executar;
- 4, 6: quando, no âmbito da execução de um plano, é necessário a realização de uma determinada ação, o coordenador envia uma mensagem *Ordem* para um dispositivo;
- 5, 7: os dispositivos realizam a ação e respondem ao coordenador indicando o resultado da mesma, desde que tal seja necessário;
- 8: os dispositivos podem, a qualquer momento, solicitar a revogação da adesão a um coordenador. A partir desse momento o coordenador deixará de enviar mensagens a esse dispositivo;
- 9: o coordenador responde aos dispositivos aceitando a revogação da adesão;
- 10: no final do processamento de um plano, quer este tenha tido sucesso ou não, o coordenador dá conhecimento ao centro de controlo dessa ocorrência.

## 4.4 Sumário

Em resumo, neste capítulo descrevemos a arquitetura de implementação do sistema *Rain Computing*, bem como dos seus módulos constituintes, detalhando os aspetos mais relevantes e os mecanismos de interação entre os componentes de cada módulo.

A implementação desta arquitetura resultou num protótipo operacional do sistema *Rain Computing* que permitiu validar o sistema e utilizá-lo no desenvolvimento de aplicações concretas, como se apresenta no capítulo 5.

Na implementação da componente de comunicação baseada em *sockets* TCP/IP e para respeitar a arquitetura do JavaEE seria necessário implementar um conetor java *inbound* para receber as mensagens dos coordenadores e dispositivos. Este conetor seria também responsável por transformar as mensagens recebidas no formato do *Rain Computing* em objetos Java. Seria ainda necessário implementar um conetor java *outbound* para enviar as mensagens para os coordenadores.

Em relação ao desenvolvimento do *Publish/Subscriber* e seguindo a arquitetura Java EE, utilizaríamos a *API JMS (Java Message Services)* e teríamos um *MDB (Message-Driven Bean)* associado ao serviço Google Pub/Sub capaz de processar as mensagens recebidas.

Tanto no caso da implementação dos *sockets* TCP/IP, como no caso do *Publish/Subscriber* não se fizeram implementações específicas do JavaEE visto que se utilizou a biblioteca *Rain*.





# Operacionalidade do Sistema *Rain Computing*

Neste capítulo descreve-se a instalação e configuração do sistema *Rain Computing*, cujas funcionalidades e arquitetura foram apresentadas nos capítulos anteriores, bem como a sua utilização no desenvolvimento de estudos de caso que demonstram a operacionalidade do sistema e a sua flexibilidade para executar casos concretos de aplicação.

## 5.1 Instalação e configuração

Nesta secção são descritos todos os passos necessário para colocar o sistema *Rain Computing* em funcionamento.

### 5.1.1 Centro de controlo

O centro de controlo é uma aplicação JEE que está empacotada num ficheiro WAR. Foi utilizada uma máquina virtual Linux, com o sistema operativo CentOS 7, para alojar o servidor aplicacional JEE Wildfly. Para estabelecer sessões de terminal com a máquina virtual e, conseqüentemente, com o servidor wildFly é utilizado um túnel SSH. As características da máquina virtual onde é executada a aplicação do centro de controlo são as seguintes: 10 GB de espaço em disco; 1 processador; 3,75 GB de memória e uma rede com 20GB de largura de banda.

Para aceder ao centro de controlo deve ser utilizado o domínio `raincmp.com`.

### 5.1.2 Coordenador

O coordenador ficou instalado numa máquina com o sistema operativo Windows 10 de 64 bits, 4 cores com *hyperthreading* a uma velocidade de 2,2 GHz, 8 GB de memória e 2 discos SSD. A instalação é feita como qualquer outra aplicação *Windows*, através do instalador disponibilizado.

O coordenador arranca quando é executada a sua interface gráfica, onde, na primeira execução, se configuram os parâmetros indicados no seu modelo de dados, ver figura 4.10. Fazem parte dos parâmetros o tipo de codificação das mensagens (JSON, XML) e o tipo de comunicação (TCP/IP, Google Pub/Sub).

### 5.1.3 Dispositivos

A instalação a efetuar nos dispositivos varia consoante o tipo de dispositivo, até se pode dar o caso de não ser necessária qualquer instalação, como por exemplo acontece com o estudo de caso das lâmpada *LED RGB Wi-Fi*.

Na secção 5.2 serão indicados detalhes de instalação e configuração para dispositivos específicos utilizados em dois estudos de caso: um relógio Inteligente e uma lâmpada *LED RGB Wi-Fi*.

## 5.2 Validação e estudos de caso

Para validar o modelo e respetiva implementação do sistema *Rain Computing* foram efetuados dois estudos de caso. Um mais complexo, devido à maior exigência para os dispositivos utilizados, que podem ser designados por dispositivos IoT inteligentes, que é o caso do *Maestro Digital*, e outro estudo de caso com dispositivos mais básicos, que é o caso do *Luzes de Palco*.

Os dispositivos utilizados pelo *Maestro Digital* executam uma aplicação, denominada de DevApp, que utiliza a interface de mensagens do sistema *Rain Computing*, quer seja pela utilização da biblioteca *Rain*, como por uma implementação própria desta interface. Assim sendo, os dispositivos, vão conseguir extrair todo o valor que a interface do sistema *Rain Computing* é capaz de fornecer, mas necessitando de desenvolvimentos específicos.

Os dispositivos utilizados pelo *Luzes de Palco* são dispositivos completamente independentes do sistema *Rain Computing*, não tendo a perceção de que estão a ser coordenados por esse sistema e conseqüentemente não é necessário qualquer esforço de implementação extra por parte deste tipo de dispositivos. Este dispositivos não conseguem ver os seus relógios sincronizados pelo coordenador.

### 5.2.1 *Maestro Digital*

O *Maestro Digital* é um cenário que se propõe coordenar uma orquestra, substituindo o maestro, ou, como no caso concreto deste trabalho, coordenar uma banda *rock*.

No caso do *Maestro Digital* os papéis do plano de execução correspondem aos vários instrumentos da banda.

O *Maestro Digital* permite que os vários músicos (dispositivos) ligados ao maestro (coordenador) através de relógios inteligentes recebam *ordens* tais como: início da música, fim da música, começo do verso, começo do refrão e outras *ordens* personalizadas pelo utilizador *especialista de negócio* que criou o plano. As ordens dadas aos músicos, através de vibrações ou da luz do visor do relógio, são diferenciadas por instrumento/dispositivo. O *Maestro Digital* também pode disponibilizar as letras das músicas, caso o dispositivo o solicite. As letras são processadas como se de um instrumento se tratasse. Cada verso que é apresentado no ecrã do dispositivo é ativado por uma *ordem*, mas neste caso o dispositivo não vibra limitando-se a exibir a letra no ecrã do dispositivo.

O *Maestro Digital* pode ainda fornecer o áudio da música, caso seja indicado no plano. O músico pode controlar a execução do áudio no dispositivo. Um ficheiro de áudio pode conter todos os instrumentos, permitindo uma experiência do tipo *karaoke*. O áudio pode também ser representado em vários ficheiros em que cada um deles é um instrumento diferente, sendo possível selecionar os instrumentos pretendidos, permitindo uma experiência tipo *karaoke* de um qualquer instrumento.

Na primeira execução do *Maestro Digital* cada dispositivo necessita proceder à descoberta de um coordenador, por intermédio do centro de controlo, e efetuar a adesão ao coordenador. Nesta adesão o dispositivo indica o instrumento que irá tocar, se vai cantar e se pretende receber a letra e/ou áudio da música.

O coordenador do *Rain Computing* consegue efetuar o papel do maestro sem necessidade de efetuar o desenvolvimento de um novo *plugin* para dialogar com os dispositivos.

A aplicação DevApp do *Maestro Digital*, que é executada nos dispositivos IoT inteligentes - neste caso relógios inteligentes - utiliza o sistema operativo WearOS da Google, que é em tudo semelhante ao Android. A aplicação DevApp do *Maestro Digital* utiliza a biblioteca *Rain* na implementação do seu serviço, abstraindo-se de quase tudo o que tem a ver com a gestão do serviço do *Rain Computing*, permitindo que se foque na implementação do seu negócio. Na implementação do *Maestro Digital* foi utilizado um relógio TicWatch E.

A instalação da aplicação no relógio foi feita a partir do IDE onde foi desenvolvida a aplicação Android, o Android Studio.

Neste estudo de caso foi utilizado o serviço Pub/Sub da Google, dada a facilidade de gestão da configuração das comunicações dos dispositivos, coordenador e centro de controlo.

### 5.2.1.1 Plano de execução do *Maestro Digital*

Na verificação da correta operacionalidade do estudo de caso do *Maestro Digital*, foi utilizada a música *Dunas dos GNR*. É uma música relativamente simples, mas com a complexidade suficiente para permitir validar a execução do plano de forma efetiva.

O plano concreto de execução descrito nesta seção representa essa música.

- Identificação do plano

Plano	
Atributo	Valor
<b>Identificação</b>	
ID	1
Aplicação	<i>Maestro Digital</i>
Nome	GNR - Dunas
Tipo	S: Síncrono
Prioridade	5
Categoria	M: Música
Tipo de unidade temporal	MIN
Unidade temporal	120
Data de início	2020-01-31T10:12:13+00:00
Data de fim	2021-01-31T10:12:12+00:00
Data de criação do plano	2020-01-11T11:02:17+00:00
Utilizador de criação do plano	Filipe Santos
Data de alteração do plano	2020-01-22T15:22:15+00:00
Utilizador de alteração do plano	Filipe Santos

Tabela 5.1: *Maestro*: Identificação do plano

A identificação do plano é descrita na tabela 5.1.

Os parâmetros são na sua grande maioria apenas descritivos, mas os parâmetros *Tipo*, *Tipo de unidade temporal* e *Unidade temporal* vão influenciar a forma como o plano é executado.

- Responsáveis pelo plano

Plano	
Atributo	Valor
<b>Responsável [1]</b>	
Nome	Rui Reininho
Telefone	919293945
Email	rei.do.rock@gmail.com
País	PT
Cidade	Porto
<b>Responsável [2]</b>	
Nome	Toli
Telefone	967293945
Email	to.ali@gmail.com
País	PT
Cidade	Porto

Tabela 5.2: *Maestro*: Responsáveis pelo plano

Os responsáveis do plano são descritos na tabela 5.2.

Neste caso temos dois responsáveis pelo plano.

- Papéis envolvidos no plano

Plano	
Atributo	Valor
<b>Papéis [1]</b>	
ID	1
Nome	Drums
Descrição	Pauta da bateria
Contexto	1
Ordens	[1, 2, 6, 17, 27, 39, 63, 67]
<b>Papéis [2]</b>	
ID	1
Nome	Bass
Descrição	Pauta do baixo
Contexto	1
Ordens	[1, 3, 7, 18, 28, 40, 64, 67]
<b>Papéis [3]</b>	
ID	1
Nome	LeadGuitar
Descrição	Pauta da guitarra principal
Contexto	1
Ordens	[1, 4, 8, 19, 29, 41, 49, 65, 67]
<b>Papéis [4]</b>	
ID	1
Nome	Accordion
Descrição	Pauta do acordeão
Contexto	1
Ordens	[1, 5, 9, 20, 30, 42, 66, 67]
<b>Papéis [5]</b>	
ID	1
Nome	MainVocals
Descrição	Pauta da bateria
Contexto	1
Ordens	[1, 10, 21, 31, 43, 61, 67]
<b>Papéis [6]</b>	
ID	1
Nome	Lyrics
Descrição	Letra da música
Contexto	1
Ordens	[1, 11, 12, 13, 14, 15, 16, 22, 23, 24, 25, 26, 32, 33, 34, 35, 36, 37, 38, 44, 45, 46, 47, 48, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 62, 67]

Tabela 5.3: Maestro: Papéis envolvidos no plano

Os papéis existentes no plano são descritos na tabela 5.3.

Os papéis contidos neste plano correspondem aos instrumentos que tocam a música, à voz e à letra. No caso desta música específica, temos os seguintes instrumentos: bateria, baixo, guitarra e acordeão.

- *Contextos* do plano

Os *contextos* do plano são descritos na tabela 5.4.

O *contexto* tem um nome e uma descrição associada e o tipo de codificação dos *dados* do *contexto*.

O atributo *dados* do *contexto* deste estudo de caso consiste na letra e na localização do ficheiro de áudio da música, bem como o nome da banda e o nome da música.

Plano	
Atributo	Valor
Contexto []	
ID	1
Nome	Letras e audio
Descrição	Letra e áudio desta música
Tempo de espera	7000
Gestão de erro	C
Tipo de Codificação	J
Dados	{ "banda": "GNR", "nomeMusica": "Dunas", "bpm": 120, "ultimoMomento": 396, "letra": ["Dunas, são como divãs", "Biombos indiscretos de alcatrão sujo", "Rasgados por cactos e hortelãs", "Deitados nas dunas, alheios a tudo", "Olhos penetrantes", "Pensamentos lavados", "Bebemos dos lábios, refrescos gelados", "Selamos segredos", "Saltamos rochedos", "Em camara lenta como na TV", "Palavras a mais na idade dos: porque", "Dunas, como que são divãs", "Quem nos visse deitados", "De cabelos molhados, bastante enrolados", "Sacos camas salgados", "Nas dunas, roendo maçãs", "A ver garrafas de óleo", "Boiando vazias nas ondas da manhã", "Bebemos dos lábios, refrescos gelados", "Nas dunas", "Em camara lenta como na TV", "Nas dunas", "Nas dunas", "Nas dunas", "Nas dunas", "Refrescos gelados", "Como na TV", "Nas dunas"], "audioPath": ["[RainResource:GNR - Dunas.mp3]"] }

Tabela 5.4: Maestro: *Contextos* do plano

O formato destes dados é específico do *Maestro Digital* e encontra-se definido em 5.2.1.2.

Neste caso concreto o *contexto* é igual para todos os papéis.

- *Ordens* (ações) envolvidas na execução do plano

Dado ao elevado número de *ordens* envolvidas no presente caso, apresentam-se apenas algumas *ordens*, como exemplo, na tabela 5.5. Em anexo A.1 encontra-se o plano completo utilizado neste cenário de execução, com codificação JSON. O formato destes dados das *ordens* é específico do *Maestro Digital* e encontra-se definido em 5.2.1.2.

### 5.2.1.2 Conteúdos específicos de mensagens para o *Maestro Digital*

Uma das tarefas de desenvolvimento da aplicação do *Maestro Digital* passa por definir a estrutura do conteúdo, do atributo *Dados*, da mensagem de *ordem* e da mensagem de *contexto*. Esta é a forma de garantir que o *especialista de negócio* consegue criar o plano de acordo com a estrutura esperada pela aplicação dos dispositivos.

- Atributo *dados* da mensagem de *Contexto* para a música

Nesta subsecção é definida a estrutura de *contexto* inicial do *Maestro Digital*. Esta encontra-se ilustrada na tabela 5.6.

A mensagem de *contexto* serve para identificar a música a executar e posicionar alguns parâmetros iniciais, como a letra e o áudio da música.

O atributo *banda* contém o nome da banda e o atributo *nomeMusica* o nome da música. Existe um atributo *bpm* que indica o intervalo de execução da música, materializada na medida batidas por minuto. O atributo *ultimoMomento*

Plano	
Atributo	Valor
<b>Ordens [2]</b>	
ID	2
Nome	Início da bateria
Descrição	São 4 toques que indicam o começo da bateria
Momento	0
Gestão de erro	C
Tipo de Codificação	J
Requer resposta	False
Dados	{"sequence": 1, "type": "Start"}
<b>Ordens [3]</b>	
ID	3
Nome	Início do baixo
Descrição	São 4 toques que indicam o começo do baixo
Momento	5
Gestão de erro	C
Tipo de Codificação	J
Requer resposta	False
Dados	{"sequence": 1, "type": "Start"}
<b>Ordens [6]</b>	
ID	6
Nome	Verso 1 bateria
Descrição	Verso 1 bateria
Momento	69
Gestão de erro	C
Tipo de Codificação	J
Requer resposta	False
Dados	{"sequence": 2, "type": "Verse"}
<b>Ordens [18]</b>	
ID	18
Nome	Refrão 1 do baixo
Descrição	Refrão 1 do baixo
Momento	132
Gestão de erro	C
Tipo de Codificação	J
Requer resposta	False
Dados	{"sequence": 3, "type": "Chorus"}
<b>Ordens [63]</b>	
ID	63
Nome	Final da bateria
Descrição	Final da bateria
Momento	396
Gestão de erro	C
Tipo de Codificação	J
Requer resposta	False
Dados	{"sequence": 7, "type": "Stop"}

Tabela 5.5: Maestro: *Ordens* (ações) envolvidas na execução do plano

Contexto				
Nome	Tipo		Valor	Descrição
<b>Pedido</b>				
banda	Texto	32		Nome da banda
nomeMusica	Texto	40		Nome da música
bpm	Número	3		Batidas Por Minuto
ultimoMomento	Número	8		Último momento da música
letra[]	Texto	9999		Letra da música
audioPath []	Texto	999		Caminhos para os ficheiros de audio
<b>Resposta</b>				
N.A.				Sem dados

Tabela 5.6: Maestro: *Contexto* inicial da música

indica a duração da música. Existe mais um atributo, com o nome *letra*, que

é um vetor de textos onde é transportada a letra da música. Por fim, existe o atributo *audioPath* que é um vetor de textos com indicação da localização dos ficheiros de audio, para que o dispositivo possa efetuar o *download* dos mesmos com a antecedência necessária para estar pronto antes do início da execução da música. Os atributos *letra* e *audioPath* são opcionais.

- Atributo *dados* da mensagem de *Ordem* para a música

Nesta subsecção é definida a estrutura de *ordem* do *Maestro Digital*. Esta encontra-se ilustrada na tabela 5.7.

A mensagem de *ordem* é utilizada para dar indicações ao dispositivo, para este enviar determinados avisos ao executante do instrumento.

Aviso				
Nome	Tipo		Valor	Descrição
Pedido				
sequence	Número	4		Sequência da ordem para o papel/instrumento
type	Texto	v13	Start Stop Change Chorus Solo Verse UserDefined_1-9	Tipo de aviso  Costumizado

Tabela 5.7: Maestro: *Ordem* de música

A estrutura do atributo *Dados* da mensagem de *ordem* do *Maestro Digital* contém um subatributo chamado *sequence* que identifica a sequência da *ordem* dentro de um papel. Este subatributo não é essencial ao processamento, mas serve como auxiliar de verificação da execução, nesta fase de verificação da correta operacionalidade do presente caso, facilitando a verificação da ordem das mensagens recebidas. O subatributo *sequence* contribui ainda para acrescentar mais complexidade aos dados da *ordem*, podendo permitir uma melhor compreensão do processamento que é feito sobre o atributo *Dados*. O outro subatributo, presente no atributo *Dados*, o *type*, identifica o tipo de aviso a dar ao executante do instrumento.

### 5.2.1.3 Fluxo de troca de mensagens

É possível ter uma ideia do fluxo de mensagens do *Maestro Digital* consultando a figura 5.1, que se passa a descrever:

- 1: o plano é enviado do centro de controlo para o coordenador selecionado para a execução do plano concreto;
- 2: o *contexto* é enviado para os 3 dispositivos;
- 3: os dispositivos informam o coordenador da correta aplicação do *contexto*;



Figura 5.1: *Maestro Digital*: Exemplo de troca de mensagens

- 4: o coordenador envia a primeira *ordem* a todos os dispositivos, com indicação de início de música;
- 5: o coordenador envia uma *ordem* específica ao dispositivo *MainVocals* com indicação de início de verso;
- 6: O coordenador envia uma *ordem* específica ao dispositivo *MainVocals* com indicação de início de refrão;
- 7: O coordenador envia a última *ordem* a todos os dispositivos, com indicação de fim de música;
- 8: o coordenador informa o centro de controlo do resultado da execução do plano.

#### 5.2.1.4 Aspetos da utilização da *Rain lib*

A aplicação DevApp dos dispositivos deste estudo de caso - relógios inteligentes - usam a biblioteca *Rain*. Assim, para desenvolver a DevApp tiveram que ser efetuados os passos descritos na subsecção 4.1.3.7.

#### 5.2.1.5 Confirmação dos resultados

Após várias execuções do estudo de caso foi possível comprovar que as mensagens são entregues aos vários dispositivos/instrumentos de acordo com os seus papéis,

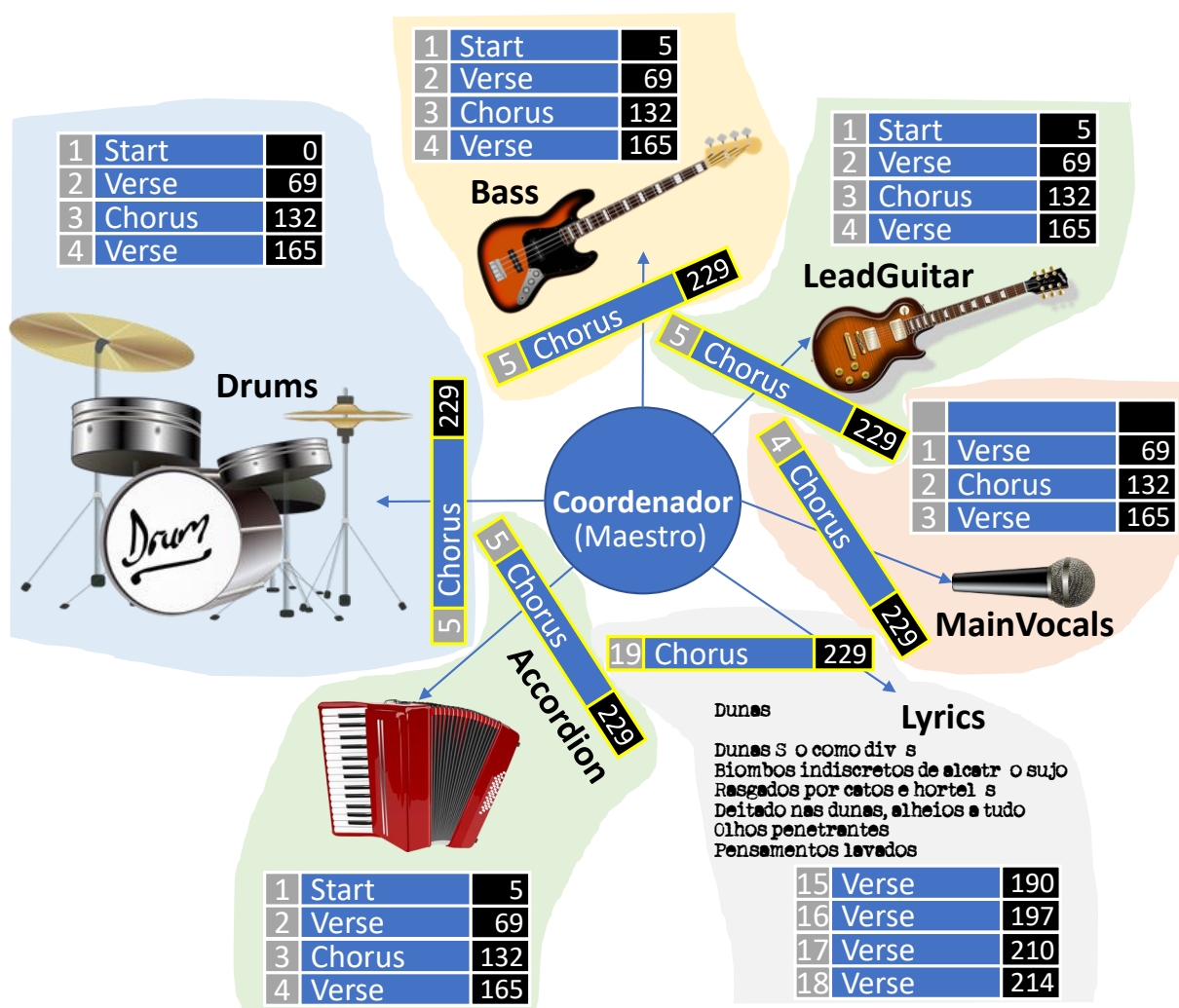


Figura 5.2: *Maestro Digital*: Mensagens do estudo de caso

pela ordem esperada, sem perdas de mensagens e no momento certo.

Um exemplo do fluxo de entrega de mensagens é apresentado na figura 5.2. Esta figura está parada no momento 229, onde todos os instrumentos vão receber nova *ordem* que se encontram a caminho, posicionadas em cima das setas com origem no coordenador e destacadas com o rebordo em amarelo.

O momento 229 que corresponde a 3m58s, dado que esta música tem um ritmo de 120 batidas por minuto. A tabela que está associada a cada um dos instrumentos indica a sequência das *ordens* dentro de cada papel, na coluna mais à esquerda, o tipo de *ordem*, na coluna do meio e o momento de execução da *ordem*, na coluna da direita. Até ao momento 229 a bateria, o baixo, a guitarra e o acordeão já processaram, cada um deles, quatro *ordens*. A voz processou três *ordens* e a letra já processou dezoito *ordens*, das quais apenas as quatro últimas fazem parte da figura.

Entrando mais em detalhe do que está representado na figura 5.2, podemos observar que:

- Momento 0: a bateria faz sozinha uma curta introdução da música;
- Momento 5: o baixo, a guitarra, e o acordeão começam a sua atuação, juntando-se à bateria, numa parte instrumental;
- Momento 69: é assinalado o começo do primeiro verso da música para todos os instrumentos, juntando-se a voz à música. Neste momento começa também a ser apresentada a letra, mas, dado que cada linha da letra corresponde a uma *ordem*, a ordem do momento 69 da letra não é apresentada na figura;
- Momento 132: começa o primeiro refrão sendo enviadas ordens a todos os instrumentos, incluindo voz e letra;
- Momento 165: assinala-se o segundo verso da música sendo enviadas *ordens* a todos os instrumentos, incluindo voz e letra;
- Momentos 190, 197, 210 e 214: são dadas *ordens* para mudança da linha da letra;
- Momento 229: é o momento captado na figura. Tem início o segundo refrão sendo enviadas *ordens* a todos os instrumentos, incluindo voz e letra;

Os resultados foram validados através de logs aplicativos escritos tanto pelo computador como pelos dispositivos. Excertos dos logs aplicativos podem ser encontrados no anexo A.2.

### 5.2.2 **Luzes de Palco**

O *Luzes de Palco* é um aplicação que se propõe coordenar luzes. A aplicação foi pensada para utilizar as lâmpadas coloridas inteligentes que hoje em dia se encontram à venda por menos de 10 euros. As lâmpadas são os dispositivos do sistema *Rain Computing*. Para efeitos de caso de estudo foi implementado em Java JFX um simulador de lâmpada inteligente, que é referido simplesmente por lâmpada.

A interface externa das lâmpadas foi previamente definida com total independência. A implementação da comunicação da lâmpada foi feita utilizando a biblioteca de *sockets* TCP/IP.

O *Luzes de Palco* consegue acender e desligar luzes. Consegue selecionar a cor da luz de cada lâmpada e colocar a lâmpada a piscar. Estão disponíveis efeitos de *fade-in*, *fade-out* e passagem de uma cor para outra de forma suave.

A configuração de um plano concreto de execução terá que conter no atributo *Dados* dos *contextos* e das *ordens*, a representação exata da interface das lâmpadas, para que estas reconheçam as *ordens* emanadas pelo coordenador do sistema *Rain Computing*.

Na criação do plano o *especialista de negócio* tem que seguir a estrutura da interface com as lâmpadas, para conseguir criar um plano de forma a que os dispositivos façam o que é desejado.

Neste caso, em que os dispositivos não conhecem o *Rain Computing*, a adesão é feita manualmente no centro de controlo, que por sua vez informará o coordenador da existência do dispositivo e suas características. Dado que os dispositivos utilizados neste estudo de caso têm uma interface com uma estrutura proprietária, é necessário indicar essa característica no momento de adesão. No caso do *Luzes de Palco* a interface proprietária consegue ser completamente descrita no atributo *Dados*, do *contexto* e da *ordem*, pelo que pode ser utilizado o tipo de interface proprietário que envia as mensagens de *contexto* e *ordem* apenas com o conteúdo do atributo *Dados*. Esta interface é implementada na classe *RainCodificadorProprietary* da biblioteca *Rain*.

Os dispositivos utilizados não vão responder em caso algum ao coordenador, pelo que este não saberá se estes executaram com sucesso as suas indicações, pelo que responderá ao centro de controlo somente com a indicação de ter executado o plano com sucesso.

Foi criada de raiz uma aplicação que simula uma lâmpada inteligente, desenvolvida em Java JFX. No estudo de caso *Luzes de Palco* a aplicação foi instalada em três executáveis independentes entre si, correspondendo cada uma das instalações a uma diferente lâmpada. A instalação da aplicação consiste numa directoria comum que tem todos os JAR necessários à sua execução. É também necessário criar uma directoria por cada lâmpada que se pretende utilizar, com os seguintes ficheiros:

- Configuração da aplicação, de seu nome *config.properties*, onde é configurado o nome da lâmpada, o IP e porto por onde recebe as mensagens.

```
name = Lamp1
ip = 127.0.0.1
port = 10001
showComments = True
```

- Configuração do log4j, chamado *log4j2.xml*, que permite indicar o nome do ficheiro onde serão registados os eventos de *logging* e o nível de *logging*, permitindo incluir mais ou menos informação na execução da aplicação;
- *Windows Batch File* de execução da aplicação, que refere a directoria comum onde se encontram os JAR necessário à execução da aplicação;
- *Log* da aplicação gerado na primeira execução da aplicação. Este ficheiro tem todos os detalhes da execução da aplicação. É um dos ficheiros utilizado para validar o correto funcionamento da aplicação.

Interface nativa da lâmpada				
Atributo	Tipo		Valor	Descrição
CommandType	Enum	-	TurnOn TurnOff SetColour TravelColour SetIntensity Blink FadeIn FadeOut	Nome do comando da lâmpada
Colour	Colour	-		Cor a colocar na lâmpada. Ver tabela 5.9
Intensity	Número	10		Intensidade da luz da lâmpada
Interval	Número	10		Intervalo de tempo do comando. Utilizado nos comandos "TravelColour", "Blink", "FadeIn" e "FadeOut"
StartColour	Colour	-		Cor de início no caso do comando "TravelColour". Ver tabela 5.9
Comment	Texto	64		Comentário

Tabela 5.8: *Luzes de Palco*: Interface nativa da lâmpada

Interface nativa da lâmpada: Cor				
Atributo	Tipo		Valor	Descrição
Red	Número	3		The red component. A number between 0 and 255
Green	Número	3		The green component. A number between 0 and 255
Blue	Número	3		The blue component. A number between 0 and 255

Tabela 5.9: *Luzes de Palco*: Interface nativa da lâmpada: Cor

### 5.2.2.1 Interface nativa das lâmpadas

O protocolo de mensagem da interface das lâmpadas consiste em ter pares de chave valor, separados pelo sinal de igualdade e empacotados em parêntesis retos. A interface está descrita nas tabelas 5.8 e 5.9. Exemplos:

- Ligar a luz suavemente para a cor amarela em 1 segundo:

```
[CommandType=FadeIn;Colour=[Red=255;Green=255;Blue=0];Interval=1000]
```

- Mudar cor para vermelho:

```
[CommandType=SetColour;Colour=[Red=255;Green=0;Blue=0]]
```

- Colocar a lâmpada a piscar na cor amarela com o intervalo de 1 segundo:

```
[CommandType=Blink;Colour=[Red=255;Green=255;Blue=0];Interval=1000]
```

- Alterar a cor suavemente de vermelho para amarelo em 10 segundos:

```
[CommandType=TravelColour;Colour=[Red=255;Green=255;Blue=0];Intensity=58;Interval=10000;StartColour=[Red=255;Green=0;Blue=0]]
```

- Alterar a intensidade da luz da lâmpada:

```
[CommandType=SetIntensity;Intensity=50]
```

- Desligar a luz suavemente em 4 segundos:

```
[CommandType=FadeOut;Interval=4000]
```

### 5.2.2.2 Plano de execução do *Luzes de Palco*

Nesta subsecção é dado um exemplo concreto de um plano de execução. Foi maximizada a interação dos dois estudo de caso efetuados e, tal como na fase de verificação da correta operacionalidade do *Maestro Digital*, a coreografia de luzes de palco criada utilizou a música *Dunas* dos GNR. Desta forma conseguimos verificar se a execução dos dois planos concretos é coordenada eficazmente.

- Identificação do plano

Plano	
Atributo	Valor
<b>Identificação</b>	
ID	2
Aplicação	Luzes de Palco
Nome	GNR - Dunas
Tipo	S: Síncrono
Prioridade	6
Categoria	O: Outros
Tipo de unidade temporal	MIN
Unidade temporal	120
Data de início	2020-01-31T10:12:13+00:00
Data de fim	2021-01-31T10:12:12+00:00
Data de criação do plano	2020-01-11T11:02:17+00:00
Utilizador de criação do plano	Filipe Santos
Data de alteração do plano	2020-01-22T15:22:15+00:00
Utilizador de alteração do plano	Filipe Santos

Tabela 5.10: *Luzes de Palco*: Identificação do plano

A identificação do plano está descrita na tabela 5.10.

Os parâmetros *Tipo*, *MomentoUnidade* e *MomentoNum* vão influenciar a forma como o plano é executado. Os restantes são apenas descritivos.

- Responsáveis pelo plano

Os responsáveis do plano estão descritos na tabela 5.11.

Neste caso temos dois responsáveis pelo plano.

Plano	
Atributo	Valor
<b>Responsável [1]</b>	
Nome	Zé das Luzes
Telefone	919123945
Email	rei.das.luzes@gmail.com
País	PT
Cidade	Porto
<b>Responsável [2]</b>	
Nome	Foco
Telefone	913493945
Email	foco@gmail.com
País	PT
Cidade	Porto

Tabela 5.11: *Luzes de Palco*: Responsáveis pelo plano

Plano	
Atributo	Valor
<b>Papéis [1]</b>	
ID	1
Nome	Right-lamp
Descrição	Lâmpada do lado direito do palco
Contexto	1
Ordens	[1, 2, 5, 8, 11, 14, 17, 20, 23, 24, 26, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 63, 66, 67]
<b>Papéis [2]</b>	
ID	1
Nome	Centre-lamp
Descrição	Lâmpada do centro do palco
Contexto	1
Ordens	[1, 3, 6, 9, 12, 15, 18, 21, 23, 25, 26, 28, 31, 34, 37, 40, 43, 46, 49, 52, 55, 58, 61, 64, 66, 67]
<b>Papéis [3]</b>	
ID	1
Nome	Left-lamp
Descrição	Lâmpada do lado esquerdo do palco
Contexto	1
Ordens	[1, 4, 7, 10, 13, 16, 19, 22, 23, 25, 26, 29, 32, 35, 38, 41, 44, 47, 50, 53, 56, 59, 62, 65, 66, 67]

Tabela 5.12: *Luzes de Palco*: Papéis envolvidos no plano

- Papéis envolvidos no plano

Os papéis do plano estão descritos na tabela 5.12.

Para demonstração foram utilizadas 3 lâmpadas, que receberam os nomes de *Left-Lamp*, *Centre-Lamp* e *Right-Lamp*, correspondendo o nome à sua localização em palco.

- *Contextos* do plano

Os *contextos* do plano estão descritos na tabela 5.13.

O *contexto* do plano é codificado com uma mensagem proprietária do *Luzes de Palco* que permite desligar a lâmpada. Desta forma é conhecido o estado inicial das lâmpadas aquando do início da execução plano.

- *Ordens* (ações) envolvidas na execução do plano

As *ordens* do plano estão descritas na tabela 5.14.

Plano	
Atributo	Valor
Contexto [ ]	
ID	1
Nome	Luzes desligadas
Descrição	Desliga as luzes
Tempo de espera	1000
Gestão de erro	C
Tipo de Codificação	P
Dados	[CommandType=TurnOff]

Tabela 5.13: *Luzes de Palco: Contextos* do plano

Plano	
Atributo	Valor
Ordens [1]	
ID	1
Nome	Fade In amarelo
Descrição	Ligar suavemente todas as luzes
Momento	0
Gestão de erro	C
Tipo de Codificação	P
Requer resposta	False
Dados	[CommandType=FadeIn;Colour=[Red=255;Green=255;Blue=0];Interval=1000]
Ordens [2]	
ID	2
Nome	Direita vermelho
Descrição	Coloca a luz da direita com a cor vermelha
Momento	5
Gestão de erro	C
Tipo de Codificação	P
Requer resposta	False
Dados	[CommandType=SetColour;Colour=[Red=255;Green=0;Blue=0]]
Ordens [3]	
ID	15
Nome	Centro a piscar
Descrição	Coloca a luz do centro a piscar com intervalo de 1 segundo
Momento	69
Gestão de erro	C
Tipo de Codificação	P
Requer resposta	False
Dados	[CommandType=Blink;Colour=[Red=255;Green=255;Blue=0];Interval=1000]
Ordens [4]	
ID	67
Nome	Fadeout a todas as luzes
Descrição	Desliga todas as luzes suavemente em 2 segundos
Momento	386
Gestão de erro	C
Tipo de Codificação	P
Requer resposta	False
Dados	[CommandType=FadeOut;Interval=10000;]

Tabela 5.14: *Luzes de Palco: Ordens* (ações) envolvidas na execução do plano

Apenas é indicada uma pequena amostra das *ordens* do plano, encontrando-se em anexo B.1 a totalidade do plano utilizado neste cenário de execução, com codificação JSON.

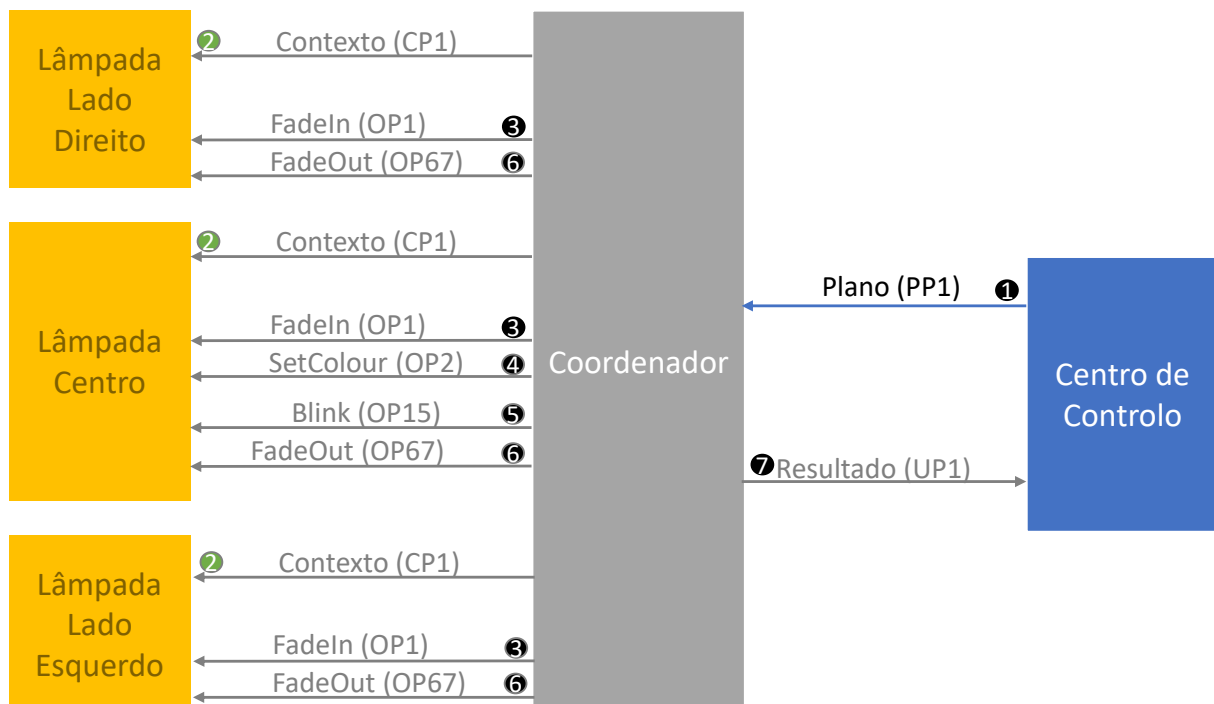


Figura 5.3: *Luzes de Palco*: Exemplo de troca de mensagens

### 5.2.2.3 Fluxo de troca de mensagens

É possível ter uma ideia do fluxo de mensagens do *Luzes de Palco* consultando a figura 5.3. Apenas são mencionadas algumas mensagens como exemplo, que se passam a descrever:

- 1: o plano é enviado do centro de controlo para o coordenador selecionado para a execução do plano concreto;
- 2: o *contexto* é enviado para os 3 dispositivos;
- 3: o coordenador envia a primeira *ordem* a todos os dispositivos, com indicação para efetuarem um *FadeIn*;
- 4: o coordenador envia uma *ordem* específica ao dispositivo *Centre-Lamp* com indicação de mudança de cor;
- 5: o coordenador envia uma *ordem* específica ao dispositivo *Centre-Lamp* com indicação para esta piscar;
- 6: o coordenador envia a última *ordem* a todos os dispositivos, com indicação para efetuarem um *FadeOut*;
- 7: o coordenador informa o centro de controlo do resultado da execução do plano

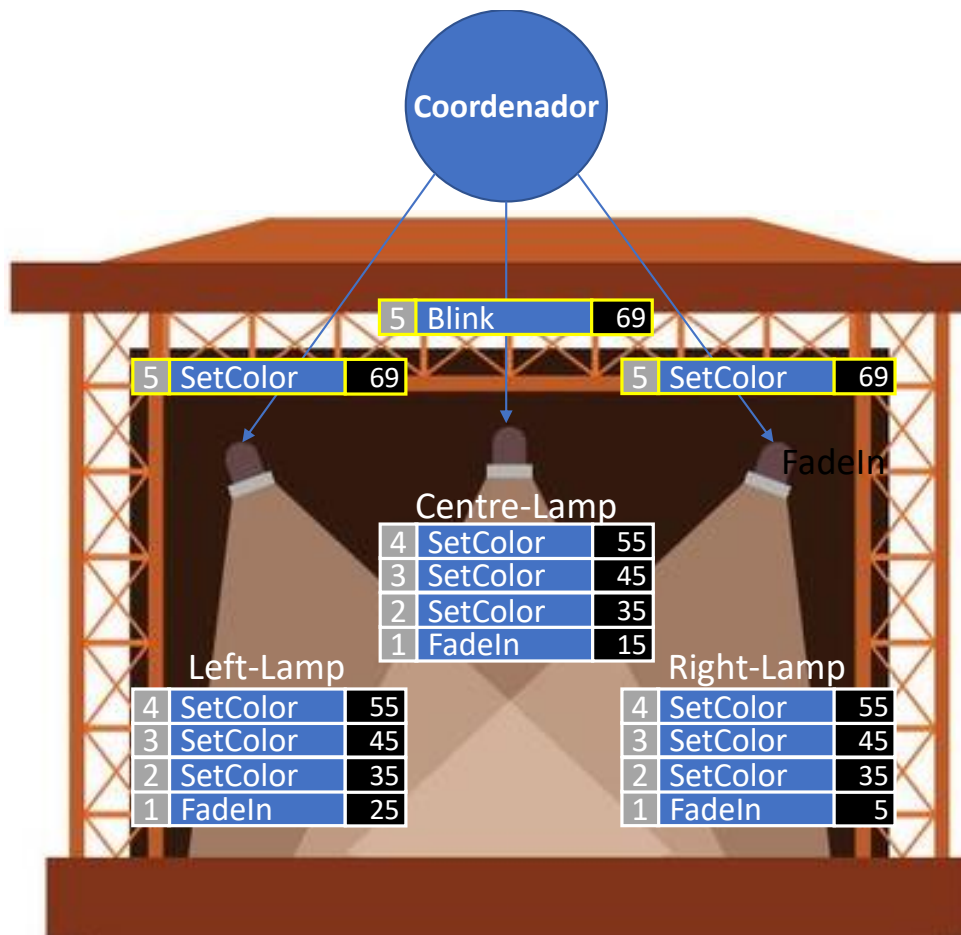


Figura 5.4: *Luzes de Palco*: Mensagens do estudo de caso

#### 5.2.2.4 Confirmação dos resultados

Após várias execuções do estudo de caso é possível comprovar que as mensagens são entregues aos vários dispositivos/lâmpadas de acordo com os seus papéis, pela ordem esperada, sem perdas de mensagens e no momento certo.

Um exemplo do fluxo de entrega de mensagens é apresentado na figura 5.4. Esta figura está parada no momento 69, que corresponde a 35 segundos, dado que esta música tem um ritmo de 120 batidas por minuto. A tabela que está associada a cada uma das lâmpadas indica a sequência da ordem, na coluna mais à esquerda, o tipo de ordem, na coluna do meio e o momento de execução da ordem, na coluna mais à direita. Até ao momento cada uma das lâmpadas já processou quatro ordens. No momento 69, que corresponde ao começo do verso da música, todas as lâmpadas vão receber nova ordem que se encontra a caminho e destacada com um rebordo em amarelo, e se encontram sobre as setas provenientes do coordenador.

Entrando mais em detalhe do que está representado na figura 5.4, podemos observar que:

- Momento 5: a luz do lado direito efetua um *FadeIn*, ou seja liga-se lentamente;

- Momento 15: a luz do centro efetua um *FadeIn*;
- Momento 25: a luz da esquerda efetua um *FadeIn*;
- Momento 35: todas as lâmpadas mudam de cor;
- Momento 45: todas as lâmpadas mudam de cor;
- Momento 55: todas as lâmpadas mudam de cor;
- Momento 69: é o momento captado na figura. Tem início o primeiro verso da música. São enviadas ordens às lâmpadas laterais para mudarem de cor e à central para começar a piscar;

Os resultados foram validados através de logs aplicativos escritos tanto pelo coordenador como pelos dispositivos. Excertos dos logs aplicativos podem ser encontrados no anexo B.2.

### 5.2.3 Avaliação dos resultados

Os resultados dos testes apresentados nos anexos A e B, foram validados através de *logs* aplicativos escritos tanto pelo coordenador como pelos dispositivos.

Analisando os vários *logs* dos diferentes dispositivos, e das várias execuções efetuadas, comprovou-se a efetiva capacidade do sistema *Rain Computing* de coordenar diversos dispositivos de forma sincronizada, enquanto estes desempenham ações distintas, de acordo com o papel que assumem durante a execução do plano. No caso do *Maestro Digital* a validação da correta execução do sistema foi feita sentindo as vibrações do relógio inteligente e pela luz emitida pelos visores dos telefones utilizados em substituição dos restantes relógios inteligentes.

#### 5.2.3.1 Conclusões

Foram apresentados neste capítulo dois estudos de casos diferentes, um sem integração da biblioteca *Rain* pela aplicação do dispositivo, o *Luzes de Palco*, e outro com a utilização da biblioteca *Rain*, o *Maestro Digital*.

Também na forma de comunicar, os dois estudos de caso funcionam de forma distinta, o *Maestro Digital* utiliza o Google Pub/Sub e o *Luzes de Palco* utiliza os *sockets* TCP/IP. Esta diferença serviu para comprovar que a biblioteca *Rain*, e restantes módulos do sistema conseguem interatuar com os dispositivos utilizando protocolos de comunicação distintos.

O caso do *Luzes de Palco* tem uma maior complicação no momento de adesão ao sistema *Rain Computing*, pois esta é feita de forma manual. Já no que diz respeito à construção do plano não foi encontrada qualquer dificuldade na sua construção ou execução, tendo o coordenador sido capaz de comunicar com os dispositivos/lâmpadas sem qualquer problema.

No caso do *Maestro Digital*, a adesão ao sistema *Rain Computing* é feita de forma automática, com uma pequena intervenção no dispositivo que pretende aderir. A construção e execução do plano decorrem de forma idêntica ao caso *Luzes de Palco*.

Apesar de os planos terem sido corretamente construídos no centro de controlo, poderia sugerir-se o desenvolvimento de uma interface com utilizador, específica para cada um dos estudos de caso, permitindo agilizar a criação dos planos. É um exemplo muito claro esta necessidade a codificação da cor no estudo de caso *Luzes de Palco*, onde a mesma foi codificada por três números indicando os componentes RGB da cor, quando seria muito mais eficaz a exibição de uma paleta com as diversas cores para seleção pelo utilizador.

# 6

## Conclusões e Trabalho Futuro

Neste capítulo são apresentadas as considerações finais sobre a realização do trabalho, com a indicação de alguns pontos conclusivos sobre o mesmo.

Como em qualquer trabalho, existem sempre alguns pontos em aberto que consideramos importantes e possíveis de realizar no futuro.

### 6.1 Conclusões

O propósito principal deste trabalho foi encontrar uma arquitetura que permita agilizar a coordenação de dispositivos IoT, a partir da *cloud*. Essa arquitetura inclui a definição de uma estrutura de dados que possibilita o registo de toda a informação necessária à coordenação dos dispositivos e à execução das ações pelos dispositivos. Esta estrutura de dados é designada por plano de execução. Inclui ainda a definição das mensagens a serem trocadas entre todas as partes do sistema, para que este consiga operar eficazmente. O outro objetivo foi a implementação do sistema *Rain Computing* por forma a ser possível comprovar que a arquitetura pensada responde às necessidades encontradas no mundo real.

Dado o elevado número de dispositivos que podem estar envolvidos, foi necessário criar o conceito de coordenadores de dispositivos. Esta camada do sistema *Rain Computing* fica entre a camada *cloud*, que chamamos de centro de controlo, e os dispositivos. Cada coordenador terá vários dispositivos sobre sua alçada, sendo o responsável direto pela sua coordenação.

Na definição da estrutura do plano de execução foi necessário pensar em como dar resposta a todas as diferenças de funcionamento existentes entre os dispositivos

móveis. A maior diferença entre os dispositivos móveis reside em conseguir implementar a interface do sistema *Rain Computing*. Os dispositivos que o conseguem fazer são chamados de inteligentes, os outros são identificados como básicos. O *Rain Computing* consegue coordenar ambos os tipos de dispositivos, apesar dos dispositivos básicos não saberem que estão a ser coordenados pelo *Rain Computing*. Os dispositivos, cada um com as suas características próprias, assumem diferentes funções na execução dos plano de execução, que se denominam por papéis.

Uma vertente importante do trabalho foi garantir o sincronismo entre os diversos dispositivos. Este desafio do sincronismo foi resolvido com a implementação de um protocolo de acerto de relógios, em que o coordenador funciona como relógio mestre dos dispositivos a si associados. A implementação deste protocolo foi validada no estudo de caso *Maestro Digital*

Foi criada uma interface gráfica no centro de controlo para que possam ser adicionados planos de execução ao sistema *Rain Computing*. Os planos de execução são criados por utilizadores do tipo *especialista de negócio*, que conseguem indicar de forma assertiva todos os dados necessário à execução das tarefas associadas ao plano. Os utilizadores do tipo *operador* agendam a execução dos planos de acordo com as necessidades, podendo escolher qual o coordenador responsável pela sua execução. O centro de controlo seleciona o coordenador, caso este não tenha sido indicado pelo *operador*, e envia-lhe o plano para que este o possa executar. O coordenador seleciona os dispositivos para a execução do plano, entre os que tem disponíveis e de acordo com os papéis que sabem executar, e envia-lhes as ordens de forma sincronizada, respeitando o plano de execução.

Este trabalho apresenta algumas limitações, nomeadamente a pouca diversidade dos dispositivos utilizados na experimentação e a dificuldade em validar com total precisão a execução dos planos. Para resolver a primeira limitação seria necessário adquirir vários tipos de dispositivos, o que implicaria maiores custos financeiros para a realização do trabalho.

Os resultados obtidos indicaram que todas as ordens foram entregues aos dispositivos quase sempre com uma precisão ao milissegundo, tendo ocorrido nos piores casos - exceção feita aos *outliers* - um desvio máximo na casa da dezena de milissegundos. A precisão na entrega das ordens aos dispositivos é semelhante tanto para os dispositivos inteligentes como para os básicos.

Apesar dos resultados obtidos não terem atingido uma precisão total, ainda assim, demonstram que o sistema *Rain Computing* funciona com uma precisão suficiente para resolver grande parte dos cenários existentes no mundo real. A facilidade da criação dos planos de execução é uma mais valia na utilização do sistema *Rain Computing*, mas não é de descartar a possibilidade de criação de interfaces gráficas dedicadas à criação de planos de execução para determinadas aplicações específicas.

Este trabalho elaborou um conjunto de regras que permitem acelerar a construção

de sistemas de dispositivos, onde a precisão de execução das acções pelos dispositivos é relevante. Estas regras permitem ainda a utilização de uma grande diversidade de dispositivos, todos a trabalhar em conjunto para o objetivo global. As regras definidas não limitam a criatividade de quem pretende construir o sistema de dispositivos, sendo possível a utilização *Rain Computing* nas mais diversas áreas de negócio.

A experimentação em torno dos dois estudo de caso permite afirmar que o sistema *Rain Computing* está operacional para o desenvolvimento de outros cenários de aplicação.

## 6.2 Trabalho futuro

Este trabalho deixa uma grande margem para desenvolvimentos sobre ele próprio. Aqui ficam algumas ideias:

- **Localização:** permitir a gestão dos dispositivos de acordo com a sua localização, permitindo a troca automática de coordenador de acordo com a localização dos dispositivo. Esta evolução permitiria alargar o âmbito da utilização desta plataforma;
- **Segurança:** implementação de políticas de autenticação dos intervenientes no sistema, validação da integridade dos dados e confidencialidade dos dados, por forma a elevar a qualidade do sistema;
- **Validação da consistência do plano:** aquando da criação do plano ser possível validar a sua consistência ao nível do negócio interagindo com o coordenador e seus dispositivos;
- **Criação de bibliotecas em C# e Python:** para alargar o leque de potenciais utilizadores do sistema.
- **Execução de um plano por iniciativa de outro plano:** permitir a reutilização de planos. Eventualmente criar o conceito de macro ordem que contém várias ordens.



# Referências

- [1] Wikipedia, *Artificial intelligence @ONLINE*, 2020. URL: [https://en.wikipedia.org/wiki/Artificial\\_intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence).
- [2] —, *Big data @ONLINE*, 2020. URL: [https://en.wikipedia.org/wiki/Big\\_data](https://en.wikipedia.org/wiki/Big_data).
- [3] —, *Cloud computing @ONLINE*, 2020. URL: [https://en.wikipedia.org/wiki/Cloud\\_computing](https://en.wikipedia.org/wiki/Cloud_computing).
- [4] Dan C. Marinescu, *Cloud Computing – Theory and Practice*, 2<sup>a</sup> ed., sér. International series of monographs on physics. Morgan Kaufmann, 2017.
- [5] Simpler Media Group, *Edge computing vs. fog computing: What’s the difference? @ONLINE*, 2014. URL: <https://www.cmswire.com/information-management/edge-computing-vs-fog-computing-whats-the-difference/>.
- [6] WEISONG SHI, GEORGE PALLIS & ZHIWEI XU, “Edge computing”, *Computing Journal*, 2019.
- [7] J. Hong, Y-G. Hong X. de Foy, M. Kovatsch, E. Schooler, D. Kutscher & Emden/Leer, “Iot edge computing challenges and functions”, *Draft*, 2019.
- [8] Kewei Sha, T. Andrew Yang, Wei Wei & Sadegh Davari, “A survey of edge computing based designs for iot security”, *Journal Pre-proof*, 2019.
- [9] Wikipedia, *Elasticity (cloud computing) @ONLINE*, 2020. URL: [https://en.wikipedia.org/wiki/Elasticity\\_\(cloud\\_computing\)](https://en.wikipedia.org/wiki/Elasticity_(cloud_computing)).
- [10] —, *Fog computing @ONLINE*, 2020. URL: [https://en.wikipedia.org/wiki/Fog\\_computing](https://en.wikipedia.org/wiki/Fog_computing).
- [11] Ellis H. Butterfield, “Fog computing with go: A comparative study”, *Claremont Colleges*, 2016.
- [12] Flavio Bonomi, Rodolfo Milito & Sateesh Addepalli Jiang Zhu, “Fog computing and its role in the internet of things”, *Cisco Systems*, 2012.
- [13] Admir Monteiro, Harishchandra Dubey1, Leslie Mahler, Qing Yang & Kunal Mankodiya1, “Fit: A fog computing device for speech tele-treatments”, *Smart Computing*, 2017.

- [14] Google, *Google pub/subs @ONLINE*, 2020. URL: <https://cloud.google.com/pubsub/docs>.
- [15] —, *Vision ai @ONLINE*, 2020. URL: <https://cloud.google.com/vision>.
- [16] Fandom, *Iomt (internet of medical things) @ONLINE*, 2020. URL: [https://internet-of-medical-things.fandom.com/wiki/Internet\\_of\\_Medical\\_Things\\_Wiki](https://internet-of-medical-things.fandom.com/wiki/Internet_of_Medical_Things_Wiki).
- [17] Wikipedia, *Internet of things @ONLINE*, 2020. URL: [https://en.wikipedia.org/wiki/Internet\\_of\\_things](https://en.wikipedia.org/wiki/Internet_of_things).
- [18] Radia Belkeziz, “Iot coordination: Designing a context-driven architecture”, *Signal-Image Technology and Internet-Based Systems*, 2017.
- [19] Ala Darabseh & Nikolaos M. Freris, “A software-defined architecture for control of iot cyberphysical systems”, *Cluster Comput*, 2018.
- [20] M. De Benedetti, F. Messina, G. Pappalardo & C. Santoro, “Jarvsi: A distributed scheduler for iot applications”, *Cluster Comput* 20, 2017.
- [21] Muhammad Alam & João Rufino, “Orchestration of microservices for iot using docker and edge computing”, *IEEE Communications Magazine*, 2018.
- [22] Andy Stanford-Clark & Hong Linh Truong, “Mqtt for sensor networks (mqtt-sn)”, *IBM*, 2013.
- [23] Anatolijs Zabasta, “Mqtt service broker for enabling the interoperability of smart city systems”, *IEEE*, 2018.
- [24] Trevor Harwood, *Iot standards and protocols @ONLINE*, 2020. URL: <https://www.postscapes.com/internet-of-things-protocols/>.
- [25] John Eidson & other, “Precision time protocol”, *IEEE*, 2016.
- [26] Nam K Giang, Rodger Lea, Michael Blackstock & Victor C.M. Leung, “On building smart city iot applications: A coordination-based perspective”, *SmartCities*, 2016.
- [27] Pasquale Pace, Gianluca Aloï, Giuseppe Caliciuri & Giancarlo Fortino, “A mission-oriented coordination framework for teams of mobile aerial and terrestrial smart objects”, *Mobile Netw Appl*, 2016.
- [28] Carla Mouradian, Diala Naboulsi, Sami Yangui, Roch H. Glitho, Monique J. Morrow & Paul A. Polakos, “A comprehensive survey on fog computing: State-of-the-art and research challenges”, *IEEE Communications Surveys and Tutorials*, 2017.
- [29] Lindong Liu, Deyu Qi, Naqin Zhou & YilinWu, “A task scheduling algorithm based on classification mining in fog computing environment”, *Wireless Communications and Mobile Computing*, 2018.



# Maestro Digital

## A.1 Plano de execução

Este foi o plano utilizado nos testes no estudo de caso *Maestro Digital*

---

```
{
  "plano":
  {
    "id":
    {
      "id": 1,
      "aplicacao": "Maestro Digital",
      "nome": "GNR - Dunas",
      "tipo": "S",
      "prioridade": 5,
      "categoria": "M",
      "momentoUnidade": "MIN",
      "momentoNum": 120,
      "dataInicio": "2020-01-31T10:12:13+00:00",
      "dataFim": "2021-01-31T10:12:12+00:00",
      "dataCriacao": "2020-01-11T11:02:17+00:00",
      "utilizadorCriacao": "Filipe Santos",
      "dataAlteracao": "2020-01-22T15:22:15+00:00",
      "utilizadorAlteracao": "Filipe Santos"
    },
    "responsaveis":
    [
      {
        "nome": "Rui Reininho",
        "telefone": 919293945,
        "email": "rei.do.rock@gmail.com",
        "pais": "PT",
        "cidade": "Porto"
      },
      {
        "nome": "Toli",
        "telefone": 967293945,
        "email": "to.ali@gmail.com",
        "pais": "PT",
      }
    ]
  }
}
```

```

        "cidade": "Porto"
    }
],
"papeis":
[
    {
        "id": 1,
        "nome": "Drums",
        "descricao": "Pauta da bateria",
        "contextoId": 1,
        "ordensId": [ 1, 2, 6, 17, 27, 39, 63, 67 ]
    },
    {
        "id": 2,
        "nome": "Bass",
        "descricao": "Pauta do baixo",
        "contextoId": 1,
        "ordensId": [ 1, 3, 7, 18, 28, 40, 64, 67 ]
    },
    {
        "id": 3,
        "nome": "LeadGuitar",
        "descricao": "Pauta da guitarra principal",
        "contextoId": 1,
        "ordensId": [ 1, 4, 8, 19, 29, 41, 49, 65, 67 ]
    },
    {
        "id": 4,
        "nome": "Accordion",
        "descricao": "Pauta do acordeao",
        "contextoId": 1,
        "ordensId": [ 1, 5, 9, 20, 30, 42, 66, 67 ]
    },
    {
        "id": 5,
        "nome": "MainVocals",
        "descricao": "Pauta do voz principal",
        "contextoId": 1,
        "ordensId": [ 1, 10, 21, 31, 43, 61, 67 ]
    },
    {
        "id": 6,
        "nome": "Lyrics",
        "descricao": "Letra da musica",
        "contextoId": 1,
        "ordensId": [ 1, 11, 12, 13, 14, 15, 16, 22, 23, 24, 25, 26, 32, 33, 34, 35, 36, 37, 38,
44, 45, 46, 47, 48, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 62, 67 ]
    }
],
"contextos":
[
    {
        "id": 1,
        "nome": "Letras e audio",
        "descricao": "Letra e audio desta musica",
        "tempoEspera": 7000,
        "codificacao": "J",
        "dados": "{\"banda\": \"GNR\", \"nomeMusica\": \"Dunas\", \"bpm\": 120, \"ultimoMomento\": 396, \"letra\": [\"Dunas, sao como divas\", \"Bimbos indiscretos de alcatrao sujo\", \"Rasgados por cactos e hortelas\", \"Deitados nas dunas, alheios a tudo\", \"Olhos penetrantes\", \"Pensamentos lavados\", \"Bebemos dos labios, refrescos gelados\", \"Selamos segredos\", \"Saltamos rochedos\", \"Em camara lenta como na TV\", \"Palavras a mais na idade dos: porque\", \"Dunas, como que sao divas\", \"Quem nos visse deitados\", \"De cabelos molhados, bastante enrolados\", \"Sacos camas salgados\", \"Nas dunas, roendo macas\", \"A ver garrafas de oleo\", \"Boiando vazias nas ondas da manha\", \"Bebemos dos labios, refrescos gelados\", \"Nas dunas\", \"Em camara lenta como na TV\", \"Nas dunas\", \"Nas dunas\", \"Nas dunas\", \"Nas dunas\", \"Refrescos gelados\", \"Como na TV\", \"Nas dunas\"]\", \"audioPath\": [\"[RainResource:GNR - Dunas.mp3]\"]}"
    }
]

```

```
    }
  ],
  "ordens":
  [
    {
      "id": 1,
      "nome": "Pre-aviso de inicio de musica",
      "descricao": "Sao 4 toques para alertar para o eminente comeco da musica",
      "momento": 0,
      "gestaoErro": "C",
      "codificacao": "J",
      "dados": "{\"seq\": 1, \"tipo\": \"Start\"}"
    },
    {
      "id": 2,
      "nome": "Inicio da bateria",
      "descricao": "Sao 4 toques que indicam o comeco da bateria",
      "momento": 0,
      "gestaoErro": "C",
      "codificacao": "J",
      "dados": "{\"seq\": 1, \"tipo\": \"Start\"}"
    },
    {
      "id": 3,
      "nome": "Inicio do baixo",
      "descricao": "Sao 4 toques que indicam o comeco da baixo",
      "momento": 5,
      "gestaoErro": "C",
      "codificacao": "J",
      "dados": "{\"seq\": 1, \"tipo\": \"Start\"}"
    },
    {
      "id": 4,
      "nome": "Inicio da guitarra",
      "descricao": "Sao 4 toques que indicam o comeco da guitarra",
      "momento": 5,
      "gestaoErro": "C",
      "codificacao": "J",
      "dados": "{\"seq\": 1, \"tipo\": \"Start\"}"
    },
    {
      "id": 5,
      "nome": "Inicio do acordeao",
      "descricao": "Sao 4 toques que indicam o comeco da acordeao",
      "momento": 5,
      "gestaoErro": "C",
      "codificacao": "J",
      "dados": "{\"seq\": 1, \"tipo\": \"Start\"}"
    },
    {
      "id": 6,
      "nome": "Verso 1 bateria",
      "descricao": "Verso 1",
      "momento": 69,
      "gestaoErro": "C",
      "codificacao": "J",
      "dados": "{\"seq\": 2, \"tipo\": \"Verse\"}"
    },
    {
      "id": 7,
      "nome": "Verso 1 baixo",
      "descricao": "Verso 1",
      "momento": 69,
      "gestaoErro": "C",
      "codificacao": "J",
      "dados": "{\"seq\": 2, \"tipo\": \"Verse\"}"
    }
  ],
}
```

```
{
  "id": 8,
  "nome": "Verso 1 LeadGuitar",
  "descricao": "Verso 1",
  "momento": 69,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 2, \"tipo\": \"Verse\"}"
},
{
  "id": 9,
  "nome": "Verso 1 Accordion",
  "descricao": "Verso 1",
  "momento": 69,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 2, \"tipo\": \"Verse\"}"
},
{
  "id": 10,
  "nome": "Verso 1 MainVocals",
  "descricao": "Verso 1",
  "momento": 69,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 1, \"tipo\": \"Verse\"}"
},
{
  "id": 11,
  "nome": "Verso 1 Letra",
  "descricao": "Verso 1",
  "momento": 69,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 1, \"tipo\": \"Verse\"}"
},
{
  "id": 12,
  "nome": "Verso 1 Letra",
  "descricao": "Verso 1",
  "momento": 82,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 2, \"tipo\": \"Verse\"}"
},
{
  "id": 13,
  "nome": "Verso 1 Letra",
  "descricao": "Verso 1",
  "momento": 92,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 3, \"tipo\": \"Verse\"}"
},
{
  "id": 14,
  "nome": "Verso 1 Letra",
  "descricao": "Verso 1",
  "momento": 100,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 4, \"tipo\": \"Verse\"}"
},
{
  "id": 15,
  "nome": "Verso 1 Letra",
  "descricao": "Verso 1",
```

```
"momento": 114,
"gestaoErro": "C",
"codificacao": "J",
"dados": "{\"seq\": 5, \"tipo\": \"Verse\"}"
},
{
  "id": 16,
  "nome": "Verso 1 Letra",
  "descricao": "Verso 1",
  "momento": 122,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 6, \"tipo\": \"Verse\"}"
},
{
  "id": 17,
  "nome": "Chorus 1 Drums",
  "descricao": "Chorus 1",
  "momento": 132,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 3, \"tipo\": \"Chorus\"}"
},
{
  "id": 18,
  "nome": "Chorus 1 Bass",
  "descricao": "Chorus 1",
  "momento": 132,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 3, \"tipo\": \"Chorus\"}"
},
{
  "id": 19,
  "nome": "Chorus 1 LeadGuitar",
  "descricao": "Chorus 1",
  "momento": 132,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 3, \"tipo\": \"Chorus\"}"
},
{
  "id": 20,
  "nome": "Chorus 1 Accordion",
  "descricao": "Chorus 1",
  "momento": 132,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 3, \"tipo\": \"Chorus\"}"
},
{
  "id": 21,
  "nome": "Chorus 1 MainVocals",
  "descricao": "Chorus 1",
  "momento": 132,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 2, \"tipo\": \"Chorus\"}"
},
{
  "id": 22,
  "nome": "Chorus 1 Letra",
  "descricao": "Chorus 1",
  "momento": 132,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 7, \"tipo\": \"Chorus\"}"
}
```

```
},
{
  "id": 23,
  "nome": "Chorus 1 Letra",
  "descricao": "Chorus 1",
  "momento": 140,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 8, \"tipo\": \"Chorus\"}"
},
{
  "id": 24,
  "nome": "Chorus 1 Letra",
  "descricao": "Chorus 1",
  "momento": 144,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 9, \"tipo\": \"Chorus\"}"
},
{
  "id": 25,
  "nome": "Chorus 1 Letra",
  "descricao": "Chorus 1",
  "momento": 148,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 10, \"tipo\": \"Chorus\"}"
},
{
  "id": 26,
  "nome": "Chorus 1 Letra",
  "descricao": "Chorus 1",
  "momento": 156,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 11, \"tipo\": \"Chorus\"}"
},
{
  "id": 27,
  "nome": "Verse 2 Drums",
  "descricao": "Verso 2",
  "momento": 165,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 4, \"tipo\": \"Verse\"}"
},
{
  "id": 28,
  "nome": "Verse 2 Bass",
  "descricao": "Verso 2",
  "momento": 165,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 4, \"tipo\": \"Verse\"}"
},
{
  "id": 29,
  "nome": "Verse 2 LeadGuitar",
  "descricao": "Verso 2",
  "momento": 165,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": "{\"seq\": 4, \"tipo\": \"Verse\"}"
},
{
  "id": 30,
  "nome": "Verse 2 Accordion",
```

```
    "descricao": "Verso 2",
    "momento": 165,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 4, \"tipo\": \"Verse\"}"
  },
  {
    "id": 31,
    "nome": "Verse 2 MainVocals",
    "descricao": "Verso 2",
    "momento": 165,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 3, \"tipo\": \"Verse\"}"
  },
  {
    "id": 32,
    "nome": "Verso 2 Letra",
    "descricao": "Verso 2",
    "momento": 165,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 12, \"tipo\": \"Verse\"}"
  },
  {
    "id": 33,
    "nome": "Verso 2 Letra",
    "descricao": "Verso 2",
    "momento": 178,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 13, \"tipo\": \"Verse\"}"
  },
  {
    "id": 34,
    "nome": "Verso 2 Letra",
    "descricao": "Verso 2",
    "momento": 182,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 14, \"tipo\": \"Verse\"}"
  },
  {
    "id": 35,
    "nome": "Verso 2 Letra",
    "descricao": "Verso 2",
    "momento": 190,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 15, \"tipo\": \"Verse\"}"
  },
  {
    "id": 36,
    "nome": "Verso 2 Letra",
    "descricao": "Verso 2",
    "momento": 197,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 16, \"tipo\": \"Verse\"}"
  },
  {
    "id": 37,
    "nome": "Verso 2 Letra",
    "descricao": "Verso 2",
    "momento": 210,
    "gestaoErro": "C",
    "codificacao": "J",
```

```
"dados": {"seq": 17, "tipo": "Verse"}
},
{
  "id": 38,
  "nome": "Verso 2 Letra",
  "descricao": "Verso 2",
  "momento": 214,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": {"seq": 18, "tipo": "Verse"}
},
{
  "id": 39,
  "nome": "Chorus 2 Letra",
  "descricao": "Chorus 2",
  "momento": 229,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": {"seq": 5, "tipo": "Chorus"}
},
{
  "id": 40,
  "nome": "Chorus 2 Bass",
  "descricao": "Chorus 2",
  "momento": 229,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": {"seq": 5, "tipo": "Chorus"}
},
{
  "id": 41,
  "nome": "Chorus 2 LeadGuitar",
  "descricao": "Chorus 2",
  "momento": 229,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": {"seq": 5, "tipo": "Chorus"}
},
{
  "id": 42,
  "nome": "Chorus 2 Accordion",
  "descricao": "Chorus 2",
  "momento": 229,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": {"seq": 5, "tipo": "Chorus"}
},
{
  "id": 43,
  "nome": "Chorus 2 MainVocals",
  "descricao": "Chorus 2",
  "momento": 229,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": {"seq": 4, "tipo": "Chorus"}
},
{
  "id": 44,
  "nome": "Chorus 2 Letra",
  "descricao": "Chorus 2",
  "momento": 229,
  "gestaoErro": "C",
  "codificacao": "J",
  "dados": {"seq": 19, "tipo": "Chorus"}
},
{
  "id": 45,
```

```
    "nome": "Chorus 2 Letra",
    "descricao": "Chorus 2",
    "momento": 240,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 20,\"tipo\": \"Chorus\"}"
  },
  {
    "id": 46,
    "nome": "Chorus 2 Letra",
    "descricao": "Chorus 2",
    "momento": 244,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 21,\"tipo\": \"Chorus\"}"
  },
  {
    "id": 47,
    "nome": "Chorus 2 Letra",
    "descricao": "Chorus 2",
    "momento": 257,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 22,\"tipo\": \"Chorus\"}"
  },
  {
    "id": 48,
    "nome": "Chorus 2 Letra",
    "descricao": "Chorus 2",
    "momento": 261,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 23,\"tipo\": \"Chorus\"}"
  },
  {
    "id": 49,
    "nome": "Solo Guitarra",
    "descricao": "Solo de guitarra",
    "momento": 261,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 6,\"tipo\": \"Solo\"}"
  },
  {
    "id": 50,
    "nome": "Verso solto 3 Letra",
    "descricao": "Chorus 2",
    "momento": 275,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 24,\"tipo\": \"Change\"}"
  },
  {
    "id": 51,
    "nome": "Verso solto 3 Letra",
    "descricao": "Chorus 2",
    "momento": 282,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 25,\"tipo\": \"Stop\"}"
  },
  {
    "id": 52,
    "nome": "Verso solto 3 Letra",
    "descricao": "Chorus 2",
    "momento": 292,
    "gestaoErro": "C",
```

```
    "codificacao": "J",
    "dados": "{\"seq\": 26,\"tipo\": \"Change\"}"
  },
  {
    "id": 53,
    "nome": "Verso solto 3 Letra",
    "descricao": "Chorus 2",
    "momento": 298,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 27,\"tipo\": \"Stop\"}"
  },
  {
    "id": 54,
    "nome": "Verso solto 3 Letra",
    "descricao": "Chorus 2",
    "momento": 324,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 28,\"tipo\": \"Change\"}"
  },
  {
    "id": 55,
    "nome": "Verso solto 3 Letra",
    "descricao": "Chorus 2",
    "momento": 331,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 29,\"tipo\": \"Stop\"}"
  },
  {
    "id": 56,
    "nome": "Verso solto 3 Letra",
    "descricao": "Chorus 2",
    "momento": 332,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 30,\"tipo\": \"Change\"}"
  },
  {
    "id": 57,
    "nome": "Verso solto 3 Letra",
    "descricao": "Chorus 2",
    "momento": 337,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 31,\"tipo\": \"Stop\"}"
  },
  {
    "id": 58,
    "nome": "Verso solto 3 Letra",
    "descricao": "Chorus 2",
    "momento": 348,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 32,\"tipo\": \"Change\"}"
  },
  {
    "id": 59,
    "nome": "Verso solto 3 Letra",
    "descricao": "Chorus 2",
    "momento": 354,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 33,\"tipo\": \"Stop\"}"
  },
  {
```

```
    "id": 60,
    "nome": "Verso solto 3 Letra",
    "descricao": "Chorus 2",
    "momento": 356,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 33, \"tipo\": \"Change\"}"
  },
  {
    "id": 61,
    "nome": "Verso solto 3 MainVocals",
    "descricao": "Chorus 2",
    "momento": 358,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 6, \"tipo\": \"Stop\"}"
  },
  {
    "id": 62,
    "nome": "Verso solto 3 Letra",
    "descricao": "Chorus 2",
    "momento": 360,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 34, \"tipo\": \"Stop\"}"
  },
  {
    "id": 63,
    "nome": "Final Drums",
    "descricao": "Final Drums",
    "momento": 396,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 7, \"tipo\": \"Stop\"}"
  },
  {
    "id": 64,
    "nome": "Final Bass",
    "descricao": "Final Bass",
    "momento": 396,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 6, \"tipo\": \"Stop\"}"
  },
  {
    "id": 65,
    "nome": "Final LeadGuitar",
    "descricao": "Final LeadGuitar",
    "momento": 396,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 6, \"tipo\": \"Stop\"}"
  },
  {
    "id": 66,
    "nome": "Final Accordion",
    "descricao": "Final Accordion",
    "momento": 396,
    "gestaoErro": "C",
    "codificacao": "J",
    "dados": "{\"seq\": 6, \"tipo\": \"Stop\"}"
  },
  {
    "id": 67,
    "nome": "Final All",
    "descricao": "Music Final",
    "momento": 396,
```

```
        "gestaoErro": "C",  
        "codificacao": "J",  
        "dados": "{\\"seq\": 2,\\"tipo\": \\"Stop\"}"  
    }  
} ]  
}
```

---

## A.2 Registos de testes

Excertos do *log* de execução do estudo de caso.

Exemplo de um registo de *log*, neste caso corresponde ao envio de uma ordem do coordenador para o centro de controlo:

```
2020-09-14 20:13:10,105 TRACE RainLigacaoGestorPubSubMsgAsincr [220] > Sent Message: Maestro_Digital.5.2.Ped  
-> {"cabecalho":{"tipoMensagem":"O", "pedRes":"P","versao":1,"idOrigem":5,"instante":"2020-09-14 20:13:10.104Z",  
"seq":156}, "nomePapel":"MainVocals","momento":396,"gestaoErro":"C", "dados":{"seq": 2,"tipo": "Stop"}}
```

O exemplo apresentado não será dos mais verbosos, ainda assim a sua leitura não é a mais apropriada para permitir uma fácil compreensão do seu conteúdo. Dada essa limitação, os excertos de *log* serão apresentados parcialmente, para tentar melhorar a sua compreensão.

### A.2.1 Centro de controlo

Alguns exemplos de mensagens registadas no *log* do centro de controlo.

#### A.2.1.1 Arranque da aplicação no servidor aplicacional

Mensagens que permitem aferir o correto arranque da aplicação do centro de controlo.

```
2020-10-19 19:34:28,387 Servico Pubsub: Inicializacao iniciada _____  
2020-10-19 19:34:31,303 Escuta em: Rain_Computing.Registar.Ped  
2020-10-19 19:34:32,005 Escuta em: Rain_Computing.Aderir.Ped  
2020-10-19 19:34:32,717 Escuta em: Rain_Computing.Descobrir.Ped  
2020-10-19 19:34:33,437 Escuta em: Rain_Computing.Resultado.Ped  
2020-10-19 19:34:34,308 Escuta em: Rain_Computing.Cord_1.Centro.Res  
2020-10-19 19:34:34,970 Escuta em: Rain_Computing.Cord_2.Centro.Res  
2020-10-19 19:34:35,682 Escuta em: Rain_Computing.Cord_5.Centro.Res  
2020-10-19 19:34:35,698 Servico Pubsub: Inicializacao terminada _____  
2020-10-19 19:34:35,713 Agendador: Inicializacao iniciada _____  
2020-10-19 19:34:49,812 Agendamento: Plano=[1-Maestro Digital - GNR - Dunas], Data=2020-10-19T19:43:04.000+0100,  
Intervalo=10 Minuto, Intervalo de Contexto=00:05.000.
```

2020-10-19 19:35:24,543 Agendamento: Plano=[1-Maestro Digital - GNR - Dunas], Data=2021-05-08T22:23:14.000+0100, Intervalo=2 Dia, Intervalo de Contexto=00:05.000.

2020-10-19 19:35:28,485 Agendamento: Plano=[1-Maestro Digital - GNR - Dunas], Data=2021-05-09T17:03:04.000+0100, Intervalo=1 Dia, Intervalo de Contexto=00:05.000.

2020-10-19 19:36:09,752 Agendamento: Plano=[2-Luzes de Palco - GNR - Dunas], Data=2020-10-19T19:43:12.000+0100, Intervalo=10 Minuto, Intervalo de Contexto=00:01.000.

2020-10-19 19:41:18,904 Agendador: Inicializacao concluída \_\_\_\_\_

### **A.2.1.2 Encerramento da aplicação no servidor aplicacional**

Registo do encerramento da aplicação do centro de controlo.

2020-10-19 19:51:22,120 Servico Pubsub: Finalização concluída \_\_\_\_\_

### **A.2.1.3 Registo de coordenadores no sistema *Rain Computing***

O serviço de registo de coordenadores no centro de controlo recebe uma mensagem de um coordenador a solicitar a entrada no sistema *Rain Computing*.

```
2020-10-09 17:30:14,277 Received message: Rain_Computing.Registar.Ped -> {"cabecalho":{"tipoMensagem":"G",
"pedRes":"P","versao":1,"idOrigem":0,"instante":"2020-10-09 17:30:13.613Z","seq":1},"coordenador":{"id":5,
"nome":"Coordenador Rain de Lisboa","descricao":"Coordena dispositivos na grande Lisboa","codigoPostal":1234,"estado":"A",
"ligacao":"PubSub"}}
```

```
2020-10-09 17:30:14,368 Sent Message: Rain_Computing.Registar.Res -> {"cabecalho":{"tipoMensagem":"G","pedRes":"R",
"versao":1,"idOrigem":0,"instante":"2020-10-09 17:30:14.358Z","instantePedido":"2020-10-09 17:30:14.358Z","seq":1},"id":5}
```

### **A.2.1.4 Descoberta de coordenadores**

O serviço de descoberta de coordenadores no centro de controlo recebe uma mensagem a solicitar coordenadores de uma determinada região, definida pelo seu código postal. Em resposta a essa solicitação envia os dados dos coordenadores mais adequados.

```
2020-09-24 16:24:54,304 Received message: Rain_Computing.Descobrir.Ped -> {"cabecalho":{"idOrigem":0,"instante":"2020-09-24 15:24:50.277Z",
"pedRes":"P","seq":1,"tipoMensagem":"D","versao":1,"tipoPesquisa":"C","valor":"2655678"}}
```

```
2020-09-24 16:24:54,775 Sent Message: Rain_Computing.Descobrir.Res -> {"cabecalho":{"tipoMensagem":"D","pedRes":"R",
"versao":1,"idOrigem":0,"instante":"2020-09-24 16:24:54.746Z","instantePedido":"2020-09-24 16:24:54.746Z","seq":9},
"coordenadores":[{"id":1,"nome":"Bar do Joy","descricao":"BAR com música ao vivo e Karaouke","codigoPostal":2655095,
"estado":"A","ligacao":["TCP/IP:IP03d10.0.2.2, Porto:8080"]}, {"id":5,"nome":"Coordenador Rain de Lisboa",
"descricao":"Coordena dispositivos na grande Lisboa","codigoPostal":2655097,"estado":"A","ligacao":"PubSub"}]}
```

### A.2.1.5 Registo (adesão) de dispositivos

O serviço de adesão de dispositivos no centro de controlo recebe uma mensagem a com os dados do dispositivo que acabou de aderir a um coordenador. Na resposta a esta mensagem envia o número de dispositivo que lhe atribuiu.

```
2020-10-09 18:34:12,622 Received message: Rain_Computing.Aderir.Ped -> {"cabecalho":{"tipoMensagem":"A","pedRes":"P","versao":1,"idOrigem":5,"instante":"2020-10-09 18:34:11.180Z","seq":3},"name":"Name 2","desc":"Desc 2","papeis":["Accordion","Accordion"],"ligacao":"Pubsub","dataInicio":"2020-10-09 17:34:05.623Z","dataFim":"292278994-08-17 07:12:55.807Z"}
```

```
2020-10-09 18:34:12,627 Sent Message: Rain_Computing.Aderir.Res -> {"cabecalho":{"tipoMensagem":"A","pedRes":"R","versao":1,"idOrigem":0,"instante":"2020-10-09 18:34:12.626Z","instantePedido":"2020-10-09 18:34:12.626Z","seq":2},"id":5,"dataInicio":"2020-10-09 17:34:05.623Z","dataFim":"292278994-08-17 07:12:55.807Z"}
```

### A.2.1.6 Registo (convite) de dispositivos limitados

Quando um dispositivo não tem capacidades para efetuar o registo através do serviço de descoberta de coordenadores, pois não conhece o sistema *Rain Computing*, é registado por um *Operador* na interface de gráfica do centro de controlo. Quando assim é, o centro de controlo informa o coordenador selecionado do novo dispositivo.

```
2020-10-08 19:56:10,512 Received message: Rain_Computing.Cord_5.Centro.Ped -> {"cabecalho":{"tipoMensagem":"C","pedRes":"P","versao":1,"idOrigem":0,"instante":"2020-10-08 19:44:01.542Z","seq":4,"id":6,"nome":"Novo Despositivo 33","desc":"aaa","codificacao":"J","tipoCom":"P","papeis":["Geral","Backup"],"ligacao":["PUB/SUB:Dados:Default"]}
```

```
2020-10-08 19:56:46,534 Received message: Rain_Computing.Cord_5.Centro.Res -> {"cabecalho":{"tipoMensagem":"C","pedRes":"R","versao":1,"idOrigem":5,"instante":"2020-10-08 19:56:44.622Z","instantePedido":"2020-10-08 19:56:44.622Z","seq":1}}
```

### A.2.1.7 Receção dos resultados da execução de planos

O serviço de receção de resultados de planos recebe uma mensagem com os resultados da execução do plano.

```
2020-10-14 19:14:16,434 Received message: Rain_Computing.Resultado.Ped -> [Keys: {idOrigem=5, pedRes=P, seq=37, tipoMensagem=U, versao=1}] {"cabecalho":{"tipoMensagem":"U","pedRes":"P","versao":1,"idOrigem":5,"instante":"2020-10-14 19:14:15.590Z","seq":37},"planoId":1,"codigoErro":1,"resultado":"Executado com sucesso"}
```

```
2020-10-14 19:14:16,462 Sent Message: Rain_Computing.Resultado.Res -> {"cabecalho":{"tipoMensagem":"U","pedRes":"R","versao":1,"idOrigem":0,"instante":"2020-10-14 19:14:16.460Z","instantePedido":"2020-10-14 19:14:16.460Z","seq":12}}
```

### A.2.1.8 Execução de planos

Quando o *Agendador* decide executar um plano, ou quando um *Operador* force a execução de um plano, é enviada uma mensagem para o coordenador.

```
2020-10-11 15:01:14,015 Sent Message: Rain_Computing.Cord_5.Centro.Ped -> {"cabecalho":{"tipoMensagem":"P",
"pedRes":"P","versao":1,"idOrigem":0,"instante":"2020-10-11 15:01:14.013Z","seq":8},"momento":"2020-10-11 15:03:14.013Z",
"plano":{"plano":{"id": {"id": 1,"aplicacao":"Maestro Digital","nome": "GNR - Dunas", "tipo": "S","prioridade": 5,"catego-
ria": "M",
...
{"id": 67,"nome": "Final All","descricao": "Music Final","momento": 396, "gestaoErro": "C","codificacao": "J","dados":
{"seq": 2,"tipo": "Stop"}}}}}}
2020-10-11 15:01:26,233 Received message: Rain_Computing.Cord_5.Centro.Res -> {"cabecalho":{"tipoMensagem":"P",
"pedRes":"R","versao":1,"idOrigem":5,"instante":"2020-10-11 15:01:24.427Z","instantePedido":"2020-10-11 15:01:24.427Z",
"seq":1}}
```

## A.2.2 Coordenador

### A.2.2.1 Registo de coordenador no sistema *Rain Computing*

A primeira mensagem a enviar por parte de um coordenador é a de registo no centro de controlo a solicitar a entrada no sistema *Rain Computing*.

```
2020-10-09 17:30:13,645 Sent Message: Rain_Computing.Registar.Ped -> {"cabecalho":{"tipoMensagem":"G","pedRes":"P",
"versao":1,"idOrigem":0,"instante":"2020-10-09 17:30:13.613Z","seq":1},"coordenador":{"id":5,"nome":"Coordenador Rain
de Lisboa","descricao":"Coordena dispositivos na grande Lisboa","codigoPostal":1234,"estado":"A","ligacao":"PubSub"}}
2020-10-09 17:30:15,926 Received message: Rain_Computing.Registar.Res -> {"cabecalho":{"tipoMensagem":"G",
"pedRes":"R","versao":1,"idOrigem":0,"instante":"2020-10-09 17:30:14.358Z","instantePedido":"2020-10-09 17:30:14.358Z",
"seq":1,"id":5}}
```

### A.2.2.2 Registo (convite) de dispositivos limitados

Quando um dispositivo não tem capacidades para efetuar o registo através do serviço de descoberta de coordenadores, pois não conhece o sistema *Rain Computing*, é registado por um *Operador* na interface de gráfica do centro de controlo. Quando assim é, o centro de controlo informa o coordenador selecionado do novo dispositivo.

```
2020-10-08 19:56:44,576 > Received message: Rain_Computing.Cord_5.Centro.Ped -> {"cabecalho":{"tipoMensagem":"C",
"pedRes":"P","versao":1,"idOrigem":0,"instante":"2020-10-08 19:56:08.241Z","seq":1},"id":6,"nome":"Novo Despositivo
33","desc":"aaa","codificacao":"J","tipoCom":"P","papeis":["Geral","Backup"],"ligacao":["PUB/SUB:Dados:Default"]}
2020-10-08 19:56:44,636 Sent Message: Rain_Computing.Cord_5.Centro.Res -> {"cabecalho":{"tipoMensagem":"C",
"pedRes":"R","versao":1,"idOrigem":5,"instante":"2020-10-08 19:56:44.622Z","instantePedido":"2020-10-08 19:56:44.622Z",
"seq":1}}
```

### A.2.2.3 Registo (adesão) de dispositivos no centro de controlo

Após receber um registo de adesão por parte de um dispositivo o coordenador envia uma mensagem de adesão ao centro de controlo, para obter a identificação única para o dispositivo e informar o centro de controlo da sua existência. Na resposta o coordenador recebe a identificação do novo dispositivo.

```
2020-10-09 18:34:11,182 Sent Message: Rain_Computing.Aderir.Ped -> {"cabecalho":{"tipoMensagem":"A","pedRes":"P",
"versao":1,"idOrigem":5,"instante":"2020-10-09 18:34:11.180Z","seq":3},"name":"Name 2","desc":"Desc 2",
"papeis":["Accordion","Accordion"],"ligacao":"Pubsub","dataInicio":"2020-10-09 17:34:05.623Z","dataFim":"292278994-08-
17 07:12:55.807Z"} 2020-10-09 18:34:12,868 Received message: Rain_Computing.Aderir.Res -> {"cabecalho":
{"tipoMensagem":"A","pedRes":"R","versao":1,"idOrigem":0,"instante":"2020-10-09 18:34:12.626Z","instantePedido":"2020-
10-09 18:34:12.626Z","seq":2},"id":5,"dataInicio":"2020-10-09 17:34:05.623Z",
"dataFim":"292278994-08-17 07:12:55.807Z"}
```

### A.2.2.4 Execução de plano

Quando o centro de controlo decide executar um plano envia uma mensagem para o coordenador com o próprio plano a executar.

```
2020-09-14 20:13:10,105 Received message: Rain_Computing.Cord_5.Centro.Ped -> {"cabecalho":{"tipoMensagem":"P",
"pedRes":"P","versao":1,"idOrigem":0,"instante":"2020-10-11 15:01:14.013Z",
"seq":8},"momento":"2020-10-11 15:03:14.013Z","plano":{"plano":{"id":{"id": 1,"aplicacao":"Maestro Digital","nome":
"GNR - Dunas","tipo": "S","prioridade": 5,"categoria": "M",
...
{"id": 67,"nome": "Final All","descricao": "Music Final","momento": 396,"gestaoErro": "C","codificacao": "J","dados":
{"seq": 2,"tipo": "Stop"}}}}}} 2020-09-14 20:13:104,170 Sent Message: Rain_Computing.Cord_5.Centro.Res ->
{"cabecalho":{"tipoMensagem":"P",
"pedRes":"R","versao":1,"idOrigem":5,"instante":"2020-10-11 15:01:24.427Z","instantePedido":"2020-10-11 15:01:24.427Z",
"seq":1}}
```

### A.2.2.5 Envio dos resultados da execução de plano

O serviço de receção de resultados de planos recebe uma mensagem com os resultados da execução do plano.

```
2020-10-14 19:14:15,591 Sent Message: Rain_Computing.Resultado.Ped -> {"cabecalho":{"tipoMensagem":"U","pedRes":"P",
"versao":1,"idOrigem":5,"instante":"2020-10-14 19:14:15.590Z","seq":37},"planoId":1,"codigoErro":1,"resultado":"Executado
com sucesso"}
2020-10-14 19:14:17,930 Received message: Rain_Computing.Resultado.Res -> {"cabecalho":{"tipoMensagem":"U",
"pedRes":"R","versao":1,"idOrigem":0,"instante":"2020-10-14 19:14:16.460Z","instantePedido":"2020-10-14 19:14:16.460Z",
"seq":12}}
```

### A.2.2.6 Registo (adesão) de dispositivos

Recebe um pedido de registo de adesão por parte de um dispositivo. Na resposta envia a identificação do dispositivo.

```
2020-10-09 18:34:10,241 Received message: Rain_Computing.Cord_5.Dispositivo.Ped -> {"cabecalho":{"idOrigem":1,"instante":"2020-10-09 17:34:05.623Z","pedRes":"P","seq":3,"tipoMensagem":"A","versao":1,"dataFim":"292278994-08-17 07:12:55.807Z","dataInicio":"2020-10-09 17:34:05.623Z","desc":"Desc 2","ligacao":"Pubsub","name":"Name 2","papeis":["Accordion","Accordion"]}  
2020-10-09 18:34:12,900 Sent Message: Rain_Computing.Cord_5.Dispositivo.Res -> {"cabecalho":{"tipoMensagem":"A","pedRes":"R","versao":1,"idOrigem":5,"instante":"2020-10-09 18:34:12.898Z","instantePedido":"2020-10-09 18:34:12.898Z","seq":4,"id":5,"dataInicio":"2020-10-09 17:34:05.623Z","dataFim":"292278994-08-17 07:12:55.807Z"}}
```

### A.2.2.7 Revogação (retirada) de dispositivos

Recebe um pedido de registo de cancelamento da adesão por parte de um dispositivo.

```
2020-10-09 19:18:04,592 Received message: Rain_Computing.Cord_5.Dispositivo.Ped -> {"cabecalho":{"idOrigem":5,"instante":"2020-10-09 19:18:04.592Z","pedRes":"P","seq":3,"tipoMensagem":"R","versao":1}}  
2020-10-09 19:18:04,673 Sent Message: Rain_Computing.Cord_5.Dispositivo.Res -> {"cabecalho":{"tipoMensagem":"R","pedRes":"R","versao":1,"idOrigem":5,"instante":"2020-10-09 19:18:04.673Z","instantePedido":"2020-10-09 18:34:12.898Z","seq":4}}
```

### A.2.2.8 Sincronização de dispositivos

O coordenador envia, em intervalos regulares, mensagens de sincronização com o objectivo de verificar a sincronização dos relógios dos dispositivos e, em caso de necessidade, solicitar o acerto dos mesmos.

#### Mensagem de verificação

```
2020-09-25 14:38:55,943 Sent Message: Rain_Computing.Cord_5.Sincronizar.Disp_2.Ped -> {"cabecalho":{"tipoMensagem":"H","pedRes":"P","versao":1,"idOrigem":5,"instante":"2020-09-25 14:38:55.941Z","seq":2},"subTipo":"R","diferenca":0}  
2020-09-25 14:38:56,013 Sent Message: Rain_Computing.Cord_5.Dispositivo.Res -> {"cabecalho":{"tipoMensagem":"H","pedRes":"R","versao":1,"idOrigem":5,"instante":"2020-09-25 14:38:56.013Z","instantePedido":"2020-09-25 14:38:55.941Z","seq":4},"subTipo":"R"}}
```

#### Mensagem de atualização de relógio

```
2020-09-25 14:38:56,096 Sent Message: Rain_Computing.Cord_5.Sincronizar.Disp_2.Ped -> {"cabecalho":{"tipoMensagem":"H","pedRes":"P","versao":1,"idOrigem":5,"instante":"2020-09-25 14:38:55.096Z","seq":3},"subTipo":"U","diferenca":12}  
2020-09-25 14:38:56,139 Sent Message: Rain_Computing.Cord_5.Dispositivo.Res -> {"cabecalho":{"tipoMensagem":"H","pedRes":"R","versao":1,"idOrigem":5,"instante":"2020-09-25 14:38:56.139Z","instantePedido":"2020-09-25 14:38:56.096Z","seq":6},"subTipo":"U"}}
```

### A.2.2.9 Contextualizar

A execução de um plano pode necessitar do envio de um contexto inicial à aplicação. Não recebe resposta.

```
2020-10-17 18:31:07,392 Sent Message: Maestro_Digital.Cord_5.Executar.Disp_2.Ped -> {"cabecalho":{"tipoMensagem":"X",
"pedRes":"P","versao":1,"idOrigem":5,"instante":"2020-10-17 18:31:07.388Z","seq":14},"codificacao":"J","dados":{"banda":
"GNR", "nomeMusica": "Dunas", "bpm":120, "ultimoMomento": 396, "letra": ["Dunas, são como divãs","Biombos indiscretos
de alcatrão sujo","Rasgados por cactos e hortelãs","Deitados nas dunas, alheios a tudo","Olhos penetrantes","Pensamentos
lavados","Bebemos dos lábios, refrescos gelados","Selamos segredos","Saltamos rochedos","Em camara lenta como na
TV","Palavras a mais na idade dos: porque","Dunas, como que são divãs","Quem nos visse deitados","De cabelos molhados,
bastante enrolados","Sacos camas salgados","Nas dunas, roendo maçãs","A ver garrafas de óleo","Boiando vazias nas
ondas da manhã","Bebemos dos lábios, refrescos gelados","Nas dunas","Em camara lenta como na TV","Nas dunas","Nas
dunas","Nas dunas","Nas dunas","Refrescos gelados","Como na TV","Nas dunas"],"audioPath": ["[RainResource:GNR -
Dunas.mp3]"]}}
```

### A.2.2.10 Ordenar

As ordens enviadas ao dispositivo. Não recebe resposta.

```
2020-10-17 18:32:07,412 Sent Message: Maestro_Digital.Cord_5.Executar.Disp_2.Ped -> {"cabecalho":{"tipoMensagem":"O",
"pedRes":"P","versao":1,"idOrigem":5,"instante":"2020-10-17 18:32:07.411Z","seq":18},"nomePapel":"Bass","momento":0,
"gestaoErro":"C","dados":{"seq": 1,"tipo": "Start"}}
```

## A.2.3 Dispositivos

A título de exemplo, alguns registo de um dos dispositivos.

### A.2.3.1 Descoberta de coordenadores

É enviada uma mensagem ao centro de controlo a solicitar coordenadores de uma determinada região, definida pelo código postal associado ao dispositivo. Em resposta recebe os dados dos coordenadores para seleccionar um deles.

```
2020-09-24 16:24:54.275 TRACE [451] Sent message: Rain_Computing.Descobrir.Ped -> {"cabecalho":{"idOrigem":0,"instante":"2020-
09-24 15:24:50.277Z","pedRes":"P","seq":1,"tipoMensagem":"D","versao":1,"tipoPesquisa":"C","valor":"2655678"}
2020-09-24 16:24:54.827 TRACE [451] Received Message: Rain_Computing.Descobrir.Res -> {"cabecalho":{"tipoMensagem":"D","pedRes":"R",
"versao":1,"idOrigem":0,"instante":"2020-09-24 16:24:54.746Z","instantePedido":"2020-09-24 16:24:54.746Z","seq":9},
"coordenadores":[{"id":1,"nome":"Bar do Joy","descricao":"BAR com música ao vivo e Karaouke","codigoPostal":2655095,
"estado":"A","ligacao":["TCP/IP:IP03d10.0.2.2, Porto:8080"]}, {"id":5,"nome":"Coordenador Rain de Lisboa",
"descricao":"Coordena dispositivos na grande Lisboa","codigoPostal":2655097,"estado":"A","ligacao":"PubSub"}]}
```

### A.2.3.2 Registo (adesão) de dispositivos

Envia um pedido de registo de adesão ao coordenador. Na resposta recebe a sua identificação no sistemas.

```
2020-10-09_18:34:05.623 TRACE [416] Sent message: Rain_Computing.Cord_5.Dispositivo.Ped -> {"cabecalho":{"idOrigem":1,"instante":"2020-10-09 17:34:05.623Z","pedRes":"P","seq":3,"tipoMensagem":"A","versao":1},"dataFim":"292278994-08-17 07:12:55.807Z","dataInicio":"2020-10-09 17:34:05.623Z","desc":"Desc 2","ligacao":"Pubsub","name":"Name 2","papeis":["Accordion","Accordion"]}  
2020-10-09_18:34:12.962 TRACE [416] Received Message: Rain_Computing.Cord_5.Dispositivo.Res -> {"cabecalho":{"tipoMensagem":"A","pedRes":"R","versao":1,"idOrigem":5,"instante":"2020-10-09 18:34:12.898Z","instantePedido":"2020-10-09 18:34:12.898Z","seq":4},"id":5,"dataInicio":"2020-10-09 17:34:05.623Z","dataFim":"292278994-08-17 07:12:55.807Z"}
```

### A.2.3.3 Revogação (retirada) de dispositivos

Envia um pedido de registo de cancelamento ao seu coordenador.

```
2020-10-09_19:18:04.592 TRACE [411] Sent message: Rain_Computing.Cord_5.Dispositivo.Ped -> {"cabecalho":{"idOrigem":5,"instante":"2020-10-09 19:18:04.592Z","pedRes":"P","seq":3,"tipoMensagem":"R","versao":1}}  
2020-10-09_19:18:04.706 TRACE [411] Received Message: Rain_Computing.Cord_5.Dispositivo.Res -> {"cabecalho":{"tipoMensagem":"R","pedRes":"R","versao":1,"idOrigem":5,"instante":"2020-10-09 19:18:04.673Z","instantePedido":"2020-10-09 18:34:12.898Z","seq":4}}
```

### A.2.3.4 Sincronização de dispositivos

Recebe do coordenador, em intervalos regulares, mensagens de sincronização com o objectivo de verificar a sincronização do seu relógio e, em caso de necessidade, o acerto dos mesmos.

Mensagem de verificação

```
2020-09-25_14:38:55.945 TRACE [416] Received Message: Rain_Computing.Cord_5.Sincronizar.Disp_2.Ped -> {"cabecalho":{"tipoMensagem":"H","pedRes":"P","versao":1,"idOrigem":5,"instante":"2020-09-25 14:38:55.941Z","seq":2},"subTipo":"R","diferenca":0}  
2020-09-25_14:38:56.019 TRACE [416] Sent Message: Rain_Computing.Cord_5.Dispositivo.Res -> {"cabecalho":{"tipoMensagem":"H","pedRes":"R","versao":1,"idOrigem":5,"instante":"2020-09-25 14:38:56.013Z","instantePedido":"2020-09-25 14:38:55.941Z","seq":4},"subTipo":"R"}
```

Mensagem de atualização de relógio

```
2020-09-25_14:38:56.101 TRACE [411] Received Message: Rain_Computing.Cord_5.Sincronizar.Disp_2.Ped -> {"cabecalho":{"tipoMensagem":"H","pedRes":"P","versao":1,"idOrigem":5,"instante":"2020-09-25 14:38:55.096Z","seq":3},"subTipo":"U","diferenca":12}  
2020-09-25_14:38:56.147 TRACE [411] Sent Message: Rain_Computing.Cord_5.Dispositivo.Res -> {"cabecalho":{"tipoMensagem":"H","pedRes":"R","versao":1,"idOrigem":5,"instante":"2020-09-25 14:38:56.139Z","instantePedido":"2020-09-25 14:38:56.096Z","seq":6},"subTipo":"U"}
```

### A.2.3.5 Contextualizar

A receção do contexto inicial relativo ao plano em execução. Não envia resposta.

```
2020-10-17_18:39:32.380 TRACE [416] Received message: Maestro_Digital.Cord_5.Executar.Disp_2.Ped -> {"cabecalho": {"tipoMensagem": "X", "pedRes": "P", "versao": 1, "idOrigem": 5, "instante": "2020-10-17 19:39:31.920Z", "seq": 2}, "codificacao": "J", "dados": {"banda": "GNR", "nomeMusica": "Dunas", "bpm": 120, "ultimoMomento": 396, "letra": ["Dunas, são como divãs", "Biombo indiscretos de alcatrão sujo", "Rasgados por cactos e hortelãs", "Deitados nas dunas, alheios a tudo", "Olhos penetrantes", "Pensamentos lavados", "Bebemos dos lábios, refrescos gelados", "Selamos segredos", "Saltamos rochedos", "Em camara lenta como na TV", "Palavras a mais na idade dos: porque", "Dunas, como que são divãs", "Quem nos visse deitados", "De cabelos molhados, bastante enrolados", "Sacos camas salgados", "Nas dunas, roendo maçãs", "A ver garrafas de óleo", "Boiando vazias nas ondas da manhã", "Bebemos dos lábios, refrescos gelados", "Nas dunas", "Em camara lenta como na TV", "Nas dunas", "Nas dunas", "Nas dunas", "Nas dunas", "Refrescos gelados", "Como na TV", "Nas dunas"], "audioPath": [{"RainResource": "GNR - Dunas.mp3"}]}}
```

### A.2.3.6 Ordenar

As ordens recebidas no dispositivo. Não envia resposta.

```
2020-10-17_18:39:40.607 TRACE [431] Received message: Maestro_Digital.Cord_5.Executar.Disp_2.Ped -> {"cabecalho": {"tipoMensagem": "O", "pedRes": "P", "versao": 1, "idOrigem": 5, "instante": "2020-10-17 19:39:41.933Z", "seq": 6}, "nomePapel": "Bass", "momento": 0, "gestaoErro": "C", "dados": {"seq": 1, "tipo": "Start"}}
```



# Luzes de Palco

## B.1 Plano de execução

Este foi o plano utilizado nos testes no estudo de caso *Maestro Digital*

---

```
{
  "plano":
  {
    "id":
    {
      "id": 2,
      "aplicacao": "Stage Light",
      "nome": "GNR - Dunas",
      "tipo": "S",
      "prioridade": 6,
      "categoria": "0",
      "momentoUnidade": "MIN",
      "momentoNum": 120,
      "dataInicio": "2020-01-31T10:12:13+00:00",
      "dataFim": "2021-01-31T10:12:12+00:00",
      "dataCriacao": "2020-01-11T11:02:17+00:00",
      "utilizadorCriacao": "Filipe Santos",
      "dataAlteracao": "2020-01-22T15:22:15+00:00"
    },
    "responsaveis":
    [
      {
        "nome": "Ze das Luzes",
        "telefone": 919293945,
        "email": "rei.das.luzes@gmail.com",
        "pais": "PT",
        "cidade": "Porto"
      },
      {
        "nome": "Foco",
        "telefone": 967293945,
        "email": "foco@gmail.com",
        "pais": "PT",
        "cidade": "Porto"
      }
    ]
  }
}
```

## B. LUZES DE PALCO

---

```
    }
  ],
  "papeis":
  [
    {
      "id": 1,
      "nome": "Right-lamp",
      "descricao": "Lampada do lado direito do palco",
      "codificacao": "P",
      "contextoId": 1,
      "ordensId": [ 1, 2, 5, 8, 11, 14, 17, 20, 23, 24, 26, 27, 30, 33, 36, 39, 42, 45, 48, 51,
54, 57, 60, 63, 66, 67 ]
    },
    {
      "id": 2,
      "nome": "Centre-lamp",
      "descricao": "Lampada do centro do palco",
      "codificacao": "P",
      "contextoId": 1,
      "ordensId": [ 1, 3, 6, 9, 12, 15, 18, 21, 23, 25, 26, 28, 31, 34, 37, 40, 43, 46, 49, 52,
55, 58, 61, 64, 66, 67 ]
    },
    {
      "id": 3,
      "nome": "Left-lamp",
      "descricao": "Lampada do lado esquerdo do palco",
      "codificacao": "P",
      "contextoId": 1,
      "ordensId": [ 1, 4, 7, 10, 13, 16, 19, 22, 23, 25, 26, 29, 32, 35, 38, 41, 44, 47, 50, 53
, 56, 59, 62, 65, 66, 67 ]
    }
  ],
  "contextos":
  [
    {
      "id": 1,
      "nome": "Luzes desligadas",
      "descricao": "Desliga as luzes",
      "codificacao": "P",
      "tempoEspera": 1000,
      "gestaoErro": "C",
      "dados": "[CommandType=TurnOff]"
    }
  ],
  "ordens":
  [
    {
      "id": 1,
      "nome": "Inicio da musica com a bateria",
      "descricao": "Ligar suavemente todas as luzes com a cor amarela",
      "momento": 0,
      "gestaoErro": "C",
      "codificacao": "P",
      "dados": "[CommandType=FadeIn;Colour=[Red=255;Green=255;Blue=0];Interval=4000;Comment=0 -
Todos - Amarelo - FadeIn]"
    }, {
      "id": 2,
      "nome": "Entram os restantes instrumentos",
      "descricao": "Direita cor vermelha",
      "momento": 5,
      "gestaoErro": "C",
      "codificacao": "P",
      "dados": "[CommandType=SetColour;Colour=[Red=255;Green=0;Blue=0];Comment=5 - Direita -
Vermelho]"
    }, {
      "id": 3,
      "nome": "Laranja",
```

```

        "descricao": "Centro cor laranja",
        "momento": 15,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=178;Blue=102];Comment=15 - Centro
- Laranja]"
    }, {
        "id": 4,
        "nome": "Amarelo",
        "descricao": "Esquerda cor amarelo",
        "momento": 25,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=255;Blue=0];Comment=25 - Esquerda
- Amarelo]"
    }, {
        "id": 5,
        "nome": "Amarelo",
        "descricao": "Direita cor Amarelo",
        "momento": 35,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=255;Blue=0];Comment=35 - Direita -
Amarelo]"
    }, {
        "id": 6,
        "nome": "Laranja",
        "descricao": "Centro cor laranja",
        "momento": 35,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=178;Blue=102];Comment=35 - Centro
- Laranja]"
    }, {
        "id": 7,
        "nome": "Vermelho",
        "descricao": "Esquerda cor vermelho",
        "momento": 35,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=0;Blue=0];Comment=35 - Esquerda -
Vermelho]"
    }, {
        "id": 8,
        "nome": "Laranja",
        "descricao": "Direita cor laranja",
        "momento": 45,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=178;Blue=102];Comment=45 - Direita
- Laranja]"
    }, {
        "id": 9,
        "nome": "Vermelho",
        "descricao": "Centro vermelho",
        "momento": 45,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=0;Blue=0];Comment=45 - Centro -
Vermelho]"
    }, {
        "id": 10,
        "nome": "Amarelo",
        "descricao": "Esquerda cor amarelo",
        "momento": 45,
        "gestaoErro": "C",
        "codificacao": "P",

```

```

        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=255;Blue=0];Comment=45 - Esquerda
- Amarelo]"
    }, {
        "id": 11,
        "nome": "Amarelo",
        "descricao": "Direita cor amarelo",
        "momento": 55,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=255;Blue=0];Comment=55 - Direita -
Amarelo]"
    }, {
        "id": 12,
        "nome": "Laranja",
        "descricao": "Centro Laranja",
        "momento": 55,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=178;Blue=102];Comment=55 - Centro
- Laranja]"
    }, {
        "id": 13,
        "nome": "Amarelo",
        "descricao": "Esquerda cor amarelo",
        "momento": 55,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=255;Blue=0];Comment=55 - Esquerda
- Amarelo]"
    }, {
        "id": 14,
        "nome": "Verso",
        "descricao": "Direita cor laranja",
        "momento": 69,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=178;Blue=102];Comment=69 - Direita
- Laranja]"
    }, {
        "id": 15,
        "nome": "Amarelo a piscar",
        "descricao": "Centro cor amarelo a piscar",
        "momento": 69,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=Blink;Colour=[Red=255;Green=255;Blue=0];Interval=1000;Comment=69 -
Centro - Amarelo - Piscar]"
    }, {
        "id": 16,
        "nome": "Laranja",
        "descricao": "Esquerda cor laranja",
        "momento": 69,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=178;Blue=102];Comment=69 -
Esquerda - Laranja]"
    }, {
        "id": 17,
        "nome": "Chorus",
        "descricao": "Direita cor azul",
        "momento": 132,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=0;Green=0;Blue=255];Comment=132 - Direita -
Azul]"
    }, {
        "id": 18,

```

```

        "nome": "Azul",
        "descricao": "Centro cor azul",
        "momento": 132,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=0;Green=0;Blue=255];Comment=132 - Centro -
Azul]"
    }, {
        "id": 19,
        "nome": "Azul",
        "descricao": "Esquerda cor azul",
        "momento": 132,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=0;Green=0;Blue=255];Comment=132 - Esquerda -
Azul]"
    }, {
        "id": 20,
        "nome": "Chorus 1",
        "descricao": "Direita verde",
        "momento": 146,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=0;Green=255;Blue=0];Comment=146 - Direita -
Verde]"
    }, {
        "id": 21,
        "nome": "Vermelho a piscar",
        "descricao": "Centro cor vermelha a piscar",
        "momento": 146,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=Blink;Colour=[Red=255;Green=0;Blue=0];Interval=1000;Comment=146 -
Centro - Vermelho - Piscar]"
    }, {
        "id": 22,
        "nome": "Verde",
        "descricao": "Esquerda verde",
        "momento": 146,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=0;Green=255;Blue=0];Comment=146 - Esquerda -
Verde]"
    }, {
        "id": 23,
        "nome": "Laranja Piscar",
        "descricao": "Todos cor laranja a piscar",
        "momento": 150,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=Blink;Colour=[Red=255;Green=178;Blue=102];Interval=1000;Comment=15
0 - Todos - Laranja - Piscar]"
    }, {
        "id": 24,
        "nome": "Verde",
        "descricao": "Centro Verde",
        "momento": 160,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=0;Green=255;Blue=0];Comment=160 - Centro -
Verde]"
    }, {
        "id": 25,
        "nome": "Azul",
        "descricao": "Esquerda e Direita azul",
        "momento": 160,
        "gestaoErro": "C",

```

```

        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=0;Green=0;Blue=255];Comment=160 - Esquerda e
Direita - Azul]"
    }, {
        "id": 26,
        "nome": "Lilas",
        "descricao": "Todos lilas",
        "momento": 163,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=0;Blue=255];Comment=163 - Todos -
Lilas]"
    }, {
        "id": 27,
        "nome": "Amarelo",
        "descricao": "Direita amarelo",
        "momento": 165,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=255;Blue=0];Comment=165 - Direita
- Amarelo]"
    }, {
        "id": 28,
        "nome": "Laranja",
        "descricao": "Centro cor Laranja",
        "momento": 165,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=178;Blue=102];Comment=165 - Centro
- Laranja]"
    }, {
        "id": 29,
        "nome": "Vermelho",
        "descricao": "Esquerda cor vermelho",
        "momento": 165,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=0;Blue=0];Comment=165 - Esquerda -
Vermelho]"
    }, {
        "id": 30,
        "nome": "Laranja",
        "descricao": "Direita Laranja",
        "momento": 176,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=178;Blue=102];Comment=176 -
Direita - Laranja]"
    }, {
        "id": 31,
        "nome": "Vermelho",
        "descricao": "Centro vermelho",
        "momento": 176,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=0;Blue=0];Comment=176 - Centro -
Vermelho]"
    }, {
        "id": 32,
        "nome": "Amarelo",
        "descricao": "Esquerda amarelo",
        "momento": 176,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=255;Blue=0];Comment=176 - Esquerda
- Amarelo]"
    }, {

```

```

    "id": 33,
    "nome": "Vermelho",
    "descricao": "Direita vermelho",
    "momento": 184,
    "gestaoErro": "C",
    "codificacao": "P",
    "dados": "[CommandType=SetColour;Colour=[Red=255;Green=0;Blue=0];Comment=184 - Direita -
Vermelho]"
  }, {
    "id": 34,
    "nome": "Vermelho",
    "descricao": "Centro vermelho",
    "momento": 189,
    "gestaoErro": "C",
    "codificacao": "P",
    "dados": "[CommandType=SetColour;Colour=[Red=255;Green=0;Blue=0];Comment=189 - Centro -
Vermelho]"
  }, {
    "id": 35,
    "nome": "Vermelho",
    "descricao": "Esquerda vermelho",
    "momento": 196,
    "gestaoErro": "C",
    "codificacao": "P",
    "dados": "[CommandType=SetColour;Colour=[Red=255;Green=0;Blue=0];Comment=196 - Esquerda -
Vermelho]"
  }, {
    "id": 36,
    "nome": "Amarelo",
    "descricao": "Direita amarelo",
    "momento": 203,
    "gestaoErro": "C",
    "codificacao": "P",
    "dados": "[CommandType=SetColour;Colour=[Red=255;Green=255;Blue=0];Comment=203 - Direita
- Amarelo]"
  }, {
    "id": 37,
    "nome": "Amarelo",
    "descricao": "Centro amarelo",
    "momento": 216,
    "gestaoErro": "C",
    "codificacao": "P",
    "dados": "[CommandType=SetColour;Colour=[Red=255;Green=255;Blue=0];Comment=216 - Centro -
Amarelo]"
  }, {
    "id": 38,
    "nome": "Amarelo",
    "descricao": "Esquerda amarelo",
    "momento": 221,
    "gestaoErro": "C",
    "codificacao": "P",
    "dados": "[CommandType=SetColour;Colour=[Red=255;Green=255;Blue=0];Comment=221 - Esquerda
- Amarelo]"
  }, {
    "id": 39,
    "nome": "Chorus 2",
    "descricao": "Direita Laranja",
    "momento": 229,
    "gestaoErro": "C",
    "codificacao": "P",
    "dados": "[CommandType=SetColour;Colour=[Red=255;Green=178;Blue=102];Comment=229 -
Direita - Laranja]"
  }, {
    "id": 40,
    "nome": "Laranja",
    "descricao": "Centro Laranja",
    "momento": 229,

```

```

        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=178;Blue=102];Comment=229 - Centro
- Laranja]"
    }, {
        "id": 41,
        "nome": "Laranja",
        "descricao": "Esquerda Laranja",
        "momento": 229,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=178;Blue=102];Comment=229 -
Esquerda - Laranja]"
    }, {
        "id": 42,
        "nome": "Vermelho",
        "descricao": "Direita vermelho",
        "momento": 240,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=0;Blue=0];Comment=240 - Direita -
Vermelho]"
    }, {
        "id": 43,
        "nome": "Amarelo",
        "descricao": "Centro amarelo",
        "momento": 240,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=255;Blue=0];Comment=240 - Centro -
Amarelo]"
    }, {
        "id": 44,
        "nome": "Azul",
        "descricao": "Esquerda azul",
        "momento": 240,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=0;Green=0;Blue=255];Comment=240 - Esquerda -
Azul]"
    }, {
        "id": 45,
        "nome": "Verde",
        "descricao": "Direita verde",
        "momento": 246,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=0;Green=255;Blue=0];Comment=246 - Direita -
Verde]"
    }, {
        "id": 46,
        "nome": "Lilas",
        "descricao": "Centro Lilas",
        "momento": 251,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=0;Blue=255];Comment=251 - Centro -
Lilas]"
    }, {
        "id": 47,
        "nome": "Azul",
        "descricao": "Esquerda Azul",
        "momento": 261,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=0;Green=0;Blue=255];Comment=261 - Esquerda -
Azul]"

```

```

    }, {
      "id": 48,
      "nome": "Azul",
      "descricao": "Direita Azul",
      "momento": 268,
      "gestaoErro": "C",
      "codificacao": "P",
      "dados": "[CommandType=SetColour;Colour=[Red=0;Green=0;Blue=255];Comment=268 - Direita -
Azul]"
    }, {
      "id": 49,
      "nome": "Verde",
      "descricao": "centro verde",
      "momento": 274,
      "gestaoErro": "C",
      "codificacao": "P",
      "dados": "[CommandType=SetColour;Colour=[Red=0;Green=255;Blue=0];Comment=274 - Centro -
Verde]"
    }, {
      "id": 50,
      "nome": "Lilas",
      "descricao": "Esquerda Lilas",
      "momento": 282,
      "gestaoErro": "C",
      "codificacao": "P",
      "dados": "[CommandType=SetColour;Colour=[Red=255;Green=0;Blue=255];Comment=282 - Esquerda -
Lilas]"
    }, {
      "id": 51,
      "nome": "Verde",
      "descricao": "Direita verde",
      "momento": 287,
      "gestaoErro": "C",
      "codificacao": "P",
      "dados": "[CommandType=SetColour;Colour=[Red=0;Green=255;Blue=0];Comment=287 - Direita -
Verde]"
    }, {
      "id": 52,
      "nome": "Lilas",
      "descricao": "Centro Lilas",
      "momento": 296,
      "gestaoErro": "C",
      "codificacao": "P",
      "dados": "[CommandType=SetColour;Colour=[Red=255;Green=0;Blue=255];Comment=296 - Centro -
Lilas]"
    }, {
      "id": 53,
      "nome": "Azul",
      "descricao": "Esquerda azul",
      "momento": 302,
      "gestaoErro": "C",
      "codificacao": "P",
      "dados": "[CommandType=SetColour;Colour=[Red=0;Green=0;Blue=255];Comment=302 - Esquerda -
Azul]"
    }, {
      "id": 54,
      "nome": "Lilas",
      "descricao": "Direita Lilas",
      "momento": 329,
      "gestaoErro": "C",
      "codificacao": "P",
      "dados": "[CommandType=SetColour;Colour=[Red=255;Green=0;Blue=255];Comment=329 - Direita -
Lilas]"
    }, {
      "id": 55,
      "nome": "Azul",
      "descricao": "Centro azul",

```

```

        "momento": 335,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=0;Green=0;Blue=255];Comment=335 - Centro -
Azul]"
    }, {
        "id": 56,
        "nome": "Verde",
        "descricao": "Esquerda verde",
        "momento": 348,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=0;Green=255;Blue=0];Comment=348 - Esquerda -
Verde]"
    }, {
        "id": 57,
        "nome": "Amarelo",
        "descricao": "Direita amarelo",
        "momento": 354,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=255;Blue=0];Comment=354 - Direita
- Amarelo]"
    }, {
        "id": 58,
        "nome": "Laranja",
        "descricao": "Centro laranja",
        "momento": 355,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=178;Blue=102];Comment=355 - Centro
- Laranja]"
    }, {
        "id": 59,
        "nome": "Vermelho",
        "descricao": "Esquerda vermelho",
        "momento": 356,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=0;Blue=0];Comment=356 - Esquerda -
Vermelho]"
    }, {
        "id": 60,
        "nome": "desligar",
        "descricao": "Direita Desligar suavemente",
        "momento": 362,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=FadeOut;Interval=4000;Comment=362 - Direita - FadeOut]"
    }, {
        "id": 61,
        "nome": "desligar",
        "descricao": "Centro Desligar suavemente",
        "momento": 362,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=FadeOut;Interval=4000;Comment=362 - Centro - FadeOut]"
    }, {
        "id": 62,
        "nome": "desligar",
        "descricao": "Esquerda Desligar suavemente",
        "momento": 362,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=FadeOut;Interval=4000;Comment=362 - Esquerda - FadeOut]"
    }, {
        "id": 63,

```

```

        "nome": "Azul a piscar",
        "descricao": "Direita azul a piscar",
        "momento": 372,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=Blink;Colour=[Red=0;Green=0;Blue=255];Interval=1000;Comment=372 -
Direita - Azul - Piscar]"
    }, {
        "id": 64,
        "nome": "Vermelho",
        "descricao": "Centro vermelho a piscar",
        "momento": 372,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=Blink;Colour=[Red=255;Green=0;Blue=0];Interval=1000;Comment=372 -
Centro - Vermelho - Piscar]"
    }, {
        "id": 65,
        "nome": "Verde a piscar",
        "descricao": "Esquerda verde a piscar",
        "momento": 372,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=Blink;Colour=[Red=0;Green=255;Blue=0];Interval=1000;Comment=372 -
Esquerda - Verde - Piscar]"
    }, {
        "id": 66,
        "nome": "Final 2",
        "descricao": "Todos Amarelo",
        "momento": 380,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=SetColour;Colour=[Red=255;Green=255;Blue=0];Comment=380 - Todos -
Amarelo]"
    }, {
        "id": 67,
        "nome": "Final 3",
        "descricao": "Desligar suavemente",
        "momento": 386,
        "gestaoErro": "C",
        "codificacao": "P",
        "dados": "[CommandType=FadeOut;Interval=10000;Comment=386 - Todos - Preto]"
    }
}
]
}
}

```

## B.2 Registos de testes

Excertos do *log* de execução do estudo de caso.

Exemplo de um registo de *log*, neste caso corresponde ao envio de uma ordem do coordenador para o centro de controlo:

```
2020-10-21 19:07:18,327 TRACE TcpServerComm [19] > [10001, 127.0.0.1:61812] -> (105)[CommandType=FadeIn;Colour=[Red=255;Green=255;Blue=0];Interval=10000;Comment=386 - Todos - Amarelo - FadeIn]
```

Por modo a facilitar a leitura os excertos de *log* serão apresentados parcialmente.

## B.2.1 Centro de controlo

### B.2.1.1 Registo (convite) de dispositivos limitados

Os dispositivos utilizados neste estudo de caso não conseguem efetuar o registo através do serviço de descoberta de coordenadores, pois não conhecem o sistema *Rain Computing*. Os dispositivos são registados por um *Operador* na interface de gráfica do centro de controlo. Por sua vez, o centro de controlo informa o coordenador selecionado do novo dispositivo.

```
2020-10-05 18:16:23,156 Sent message: Rain_Computing.Cord_5.Centro.Ped -> {"cabecalho":{"tipoMensagem":"C",
"pedRes":"P","versao":1,"idOrigem":0,"instante":"2020-10-05 18:16:23.156Z","seq":7},"id":6,"nome":"Lamp_1",
"desc":"Lampada 1",
"codificacao":"P","tipoCom":"T","papeis":["Right-lamp"],"ligacao":["TCP/IP:IP=127.0.0.1, Porto=10001"]}
```

```
2020-10-05 18:16:23,813 Received message: Rain_Computing.Cord_5.Centro.Res -> {"cabecalho":{"tipoMensagem":"C",
"pedRes":"R","versao":1,"idOrigem":5,"instante":"2020-10-05 18:16:23.813Z","instantePedido":"2020-10-05 18:16:23.156Z",
"seq":1}}
```

## B.2.2 Coordenador

### B.2.2.1 Comandos às lâmpadas (Contexto e Ordens)

O coordenador envia comandos aos dispositivos, estes não respondem.

```
2020-10-21 19:07:18,326 [61813, 127.0.0.1:10003] <- (105)[CommandType=FadeIn;Colour=[Red=255;Green=255;Blue=0];
Interval=4000;Comment=0 - Todos - Amarelo - FadeIn]
```

## B.2.3 Dispositivos

A título de exemplo, alguns registo de um dos dispositivos.

### B.2.3.1 Comandos às lâmpadas (Contexto e Ordens)

O coordenador envia comandos aos dispositivos, estes não respondem.

```
2020-10-21 19:07:18,326 [61813, 127.0.0.1:10003] <- [CommandType=FadeIn;Colour=[Red=255;Green=255;Blue=0];
Interval=4000;Comment=0 - Todos - Amarelo - FadeIn]
```

```
2020-10-21 19:07:53,268 [61813, 127.0.0.1:10003] <- [CommandType=SetColour;Colour=[Red=255;Green=178;Blue=102];
Comment=70 - Direita - Laranja]
```

```
2020-10-21 19:07:53,268 [61811, 127.0.0.1:10002] <- (106)[CommandType=Blink;Colour=[Red=255;Green=255;Blue=0];
Interval=1000;Comment=70 - Centro - Amarelo - Piscar]
```