



Mecanismo de Sincronização de Dados Online/Offline - App de Registo de Consumos Hospitalares

ANA INÊS NOVAIS DAVID

Outubro de 2019

**Mecanismo de Sincronização de Dados
Online/Offline
App de Registo de Consumos Hospitalares**

Ana Inês Novais David

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Engenharia do Software**

Orientador: Dr. Paulo Baltarejo Sousa

Porto, 13 de Outubro de 2019

Dedicatória

«Aos meus pais que desde sempre me apoiaram
em todas as decisões que tomei ao longo da minha vida. »

Resumo

O rigoroso mercado da saúde impõe soluções adaptadas, completas e atualizadas que possibilitem gerir todas as informações produzidas numa organização hospitalar.

A solução Globalcare desenvolvida pela Glintt, pretende acompanhar a evolução tecnológica. Dentro da solução Globalcare, encontra-se o módulo de farmácia e logística hospitalar que tem como propósito dar suporte a todos os intervenientes que participam no circuito do medicamento. Assegurando assim uma gestão eficiente de todo o fluxo logístico no que respeita a bens e serviços.

Atualmente, na logística hospitalar são utilizados equipamentos, dispositivos móveis do tipo PDA, que em determinados clientes funcionam em modo *online*, noutras situações, é possível que estes dispositivos funcionem em modo *offline*.

Contudo, em ambiente de armazém que muitas vezes se situam em caves ou em locais sem acesso a Wi-Fi, torna-se necessário a criação de uma aplicação móvel que permita o registo de consumos quer em modo *online* ou *offline*. Para que este objetivo seja cumprido, é essencial fazer a sincronização de dados assim que tenha conectividade Wi-Fi, sem que tenha qualquer impacto na sua utilização.

Para a construção da aplicação de registo de consumos, também foi necessário elaborar um estudo sobre *frameworks* multiplataforma móveis, que acabou por eleger React Native como a escolha mais viável.

Através do estudo de diferentes métodos de sincronização, elaborou-se um mecanismo que permitiu a construção de uma aplicação que funciona em modo *online* e *offline*. Este mecanismo segue uma abordagem algorítmica, que trabalha maioritariamente assincronamente e guarda todos os dados necessários no dispositivo móvel.

Finalmente, e com o objetivo de provar a viabilidade da solução desenvolvida, é elaborada uma avaliação, com o objetivo de comprovar que a aplicação construída resolve o problema inicialmente descrito. Ou seja, chegar à conclusão que foi possível construir um mecanismo de sincronização viável, que permite trabalhar em modo *online/offline* no contexto deste problema.

Palavras-chave: *framework* móvel multiplataforma, logística hospitalar, sincronização de dados, registo de consumos.

Abstract

The rigorous healthcare market imposes tailored, complete and up-to-date solutions that allow the use of all information produced in a hospital organization.

The Globalcare solution developed by Glintt aims to keep up with technological development. Within the Globalcare solution, it's possible to find the pharmacy and hospital logistics module, that aims to support all stakeholders who participate in the medicine circuit. This ensures efficient management of the entire logistics flow.

Currently in a hospital, logistics use equipment's, mobile devices (PDA), which sometimes works online, in other situations, it is possible that these devices work on offline mode.

However, in a warehouse environment, that is often in caves or places without Wi-Fi access, arises the need to create a mobile application that allows consumption registration both online and offline. To achieve this goal, it is essential to create a mechanism of data synchronization.

To build this app, it was also necessary to elaborate a study on mobile multiplatform frameworks, which eventually chose React Native as the more viable option.

Through the study of different synchronization methods, a mechanism was developed. This mechanism allows the application to work on online and offline mode. This mechanism follows an algorithmic approach, which works mostly asynchronously and stores all necessary data on the mobile device.

Finally, in order to prove the viability of the developed solution, an evaluation is elaborated in order to prove that the application created solves the described problem. The goal is to conclude that it was possible to create a workable synchronization mechanism that allows working in online/offline mode within the context of this problem.

Keywords: multiplatform mobile framework, hospital logistics, data synchronization, consumption registration.

Agradecimentos

Gostaria de agradecer em primeiro lugar à Instituição Superior de Engenharia do Porto (ISEP), por todo o conhecimento adquirido durante o meu percurso no Mestrado em Engenharia de Software.

Agradeço ao meu orientador, pela orientação prestada, salientando a disponibilidade e flexibilidade para esclarecimento de dúvidas e pelo esforço de revisão da presente dissertação.

Agradeço à Glintt, por facilitar a realização deste projeto e aos meus colegas de trabalho por me incentivarem durante este percurso.

Agradeço aos meus pais, que sempre me suportaram ao longo do meu percurso académico quer em fases negativas como em fases positivas.

Por fim agradeço ao meu namorado, por todas as atitudes de apoio que me levaram a atingir os meus objetivos.

Índice

1	Introdução	1
1.1	Contexto	1
1.1.1	Registo de Consumos Hospitalares.....	1
1.1.2	Glintt - Global Intelligent Technologies	1
1.1.3	SGICM-Mobile	2
1.2	Problema.....	2
1.3	Objetivos.....	3
1.4	Abordagem preconizada	3
1.5	Análise de valor	4
1.6	Estrutura do Documento	4
2	Contexto, Problema e Estado de arte	5
2.1	Contexto	5
2.1.1	Problema.....	6
2.1.2	Propósito.....	7
2.2	Estado de Arte em Soluções Existentes	7
2.2.1	Soluções Glintt	8
2.2.2	Solução BIQ Health Solutions.....	10
2.2.3	SGICM-Mobile vs MAPP	11
2.3	Abordagens Utilizadas na Sincronização de Dados	12
2.3.1	Padrões de Mecanismos de Sincronização de Dados.....	12
2.3.2	Padrões de Armazenamento de Dados	14
2.3.3	Padrões de Transferências de Dados	16
2.4	Abordagens de Implementação em Aplicações Móveis.....	18
2.4.1	Tipos de Desenvolvimento de Aplicações Móveis	19
2.4.2	Tecnologias	21
2.5	Sumário	24
3	Análise de Valor	25
3.1	Modelo New Concept Development	25
3.1.1	Identificação da Oportunidade.....	26
3.1.2	Análise de Oportunidade.....	27
3.1.3	Geração de Ideias.....	27
3.1.4	Seleção de Ideias	28
3.1.5	Definição do Conceito.....	28
3.1.6	Valor, Valor Percecionado e Valor para o Cliente	28
3.1.7	Valor Percebido	29
3.1.8	Valor para o Cliente	30
3.2	Proposta de Valor.....	30
3.3	Modelo de Negócio Canvas	30
3.4	Rede de Valor	33
3.5	AHP (Analytic Hierarchy Process).....	33
3.6	Sumário	37

4	Solução.....	39
4.1	Casos de Uso	39
4.1.1	Descrição de Casos de Uso	40
4.2	Alternativas e Abordagem de Implementação	41
4.3	Visão Lógica	43
4.4	Visão de Processos	44
4.5	Visão Física	45
4.6	Visão de Desenvolvimento.....	46
4.6.1	Registo de Consumos.....	46
4.6.2	Gravação de um Consumo	47
4.6.3	Sincronização de Dados.....	48
4.7	Modelo de Dados.....	49
4.8	Sumário	50
5	Avaliação	51
5.1	Grandezas	51
5.2	Identificação de Hipóteses	52
5.3	Metodologia de avaliação	52
5.4	Testes estatísticos.....	52
5.4.1	Tempo de Sincronização de Dados	52
5.4.2	Taxa de Exatidão na Sincronização de Dados.....	54
5.4.3	Nível de Satisfação da Equipa SGICM com a Nova Solução Desenvolvida.....	56
5.5	Sumário	61
6	Conclusão.....	63
6.1	Síntese	63
6.2	Objetivos Atingidos.....	64
6.2.1	Questão Q1	64
6.2.2	Questão Q2	64
6.2.3	Questão Q3	65
6.2.4	Questão Q4	65
6.2.5	Questão Q5	65
6.3	Limitações.....	65
6.4	Apreciação Final.....	66
	Referências	67
	Anexos	70
	Anexo I - Requisitos	70
	Anexo II - Descrição de Casos de Uso	71
	UC01 - Realizar autenticação	71
	UC02 - Adicionar o local físico onde se encontra	71
	UC03 - Associar um paciente ao registo de consumos.....	72
	UC04 - Registrar artigos manualmente	73
	UC05 - Eliminar previamente artigos adicionados	73
	UC07 - Consultar histórico de dados sincronizados	74

Anexo III - Desenho da Interface.....	75
Ecrã de <i>Login e Home</i>	76
Ecrã de seleção de local e pacientes	76
Ecrã de seleção de artigos	76
Ecrã de sincronização de movimentos.....	76
Anexo IV - Modelo de Dados	77
Anexo V - Funcionalidades desenvolvidas	78
Sincronização de dados	78
Base de Dados Local.....	79
Obter estado da conexão	79

Lista de Figuras

Figura 1 – Diagrama de implantação do sistema de registo de consumos da Glintt	8
Figura 2 – Abordagem assíncrona de sincronização de dados (McCormick & Schmidt, 2012) .	13
Figura 3 – Abordagem síncrona de sincronização de dados (McCormick & Schmidt, 2012)	14
Figura 4 – Armazenamento parcial de dados (McCormick & Schmidt, 2012).....	15
Figura 5 – Armazenamento total de dados (McCormick & Schmidt, 2012)	16
Figura 6 – Transferência total de dados (McCormick & Schmidt, 2012)	16
Figura 7 – Transferência por <i>timestamp</i> de dados (McCormick & Schmidt, 2012)	17
Figura 8 – Transferência matemática de dados (McCormick & Schmidt, 2012)	18
Figura 9 – Partilha de código utilizando Xamarin (Microsoft, 2018)	21
Figura 10 – Estrutura da <i>framework</i> Ionic (Ionic, 2018)	22
Figura 11 – Arquitetura da <i>framework</i> React Native (ICAPPS, 2017)	23
Figura 12 – Modelo NCD (New Concept Development)	26
Figura 13 – Perspetiva longitudinal VC (Woodall, 2003)	29
Figura 14 – Diagrama de análise da rede de valor.	33
Figura 15 – Árvore hierárquica de decisão	35
Figura 16 – Diagrama de Casos de Uso	40
Figura 17 – Diagrama de Componentes SGICM-Mobile	44
Figura 18 – Diagrama de atividade sobre a sincronização de dados.....	45
Figura 19 – Diagrama de Implantação utilizando React Native.....	45
Figura 20 – Diagrama de sequência de Registo de Consumo.....	47
Figura 21 – Diagrama de sequência gravação de consumo	48
Figura 22 – Diagrama de sequência de sincronização de dados	49
Figura 23 – Modelo de dados parcial (BD local)	50
Figura 24 – Inquérito: Grau de satisfação relativo ao tempo da primeira sincronização	57
Figura 25 – Inquérito: Grau de satisfação relativo ao tempo de sincronização	57
Figura 26 – Inquérito: Grau de satisfação relativo à consistência de dados sincronizados	58
Figura 27 – Inquérito: Grau de satisfação relativo à geração de um consumo em <i>offline</i>	58
Figura 28 – Inquérito: Grau de satisfação relativo ao ecrã de estado de sincronização.....	58
Figura 29 – Inquérito: Grau de satisfação relativo às tecnologias escolhidas.....	59
Figura 30 – Inquérito: Grau de satisfação relativo ao projeto global.....	59
Figura 31 – Inquérito: Opinião pessoal sobre continuação do projeto	61
Figura 32 – Ecrã <i>login e home</i>	75
Figura 33 – Ecrã de seleção de local e pacientes.....	75
Figura 34 – Ecrã de seleção de artigos.....	75
Figura 35 – Ecrã de sincronização de movimentos.....	75
Figura 36 – Diagrama de relação entidade	77

Lista de Tabelas

Tabela 1 – Comparação entre a solução SGICM-Mobile e MAPP.	12
Tabela 2 – Comparação entre tipo de desenvolvimentos para aplicações móveis	20
Tabela 3 – Comparação de <i>frameworks</i> multiplataforma	24
Tabela 4 - Tabela de benefícios e sacrifícios	29
Tabela 5 – Modelo de negócio Canvas	32
Tabela 6 – Comparações AHP (Saaty, 2008)	34
Tabela 7 – Tabela de avaliação AHP	35
Tabela 8 – Matriz normalizada do método de avaliação HP	35
Tabela 9 – Comparação paritária dos critérios	36
Tabela 10 – Matriz Normalizada para os critérios e prioridade local.....	36
Tabela 11 – Classificação de <i>frameworks</i>	37
Tabela 12 – UC06 – Gravar o registo de consumos	41
Tabela 13 – Tempos de sincronização	53
Tabela 14 – Tempos de sincronização	55
Tabela 15 – Correspondência entre a escolha do inquirido e o seu valor numérico.....	56
Tabela 16 – Nível de Satisfação: Número de respostas por pergunta	60
Tabela 17 – UC01: Realizar autenticação	71
Tabela 18 – UC02: Adicionar o local físico onde se encontra.....	71
Tabela 19 – UC03: Associar um paciente ao registo de consumos	72
Tabela 20 – UC04: Registrar artigos manualmente	73
Tabela 21 – UC05: Eliminar previamente artigos adicionados	73
Tabela 22 – UC07: Consultar histórico de dados sincronizados	74

Lista de Código

Código 1 – Teste de Shapiro para tempo de sincronização de dados inferior a 300s.....	53
Código 2 – T-test para tempo de sincronização de dados inferior a 300s	54
Código 3 – Teste de Shapiro para taxa de exatidão na sincronização de dados	55
Código 4 – Teste de Wilcoxon para taxa de exatidão na sincronização de dados	56
Código 5 – Excerto sobre o processo de sincronização de dados	78
Código 6 – Inserção de dados na BD local	79
Código 7 – Redux: Implementação de <i>action</i>	80
Código 8 – Implementação do <i>reducer</i> para guardar estado da conectividade	80
Código 9 – NetInfo: Obtenção do estado de conectividade	80

Acrónimos e Símbolos

Lista de Acrónimos

AHP	Analytic Hierarchy Process
AIDC	Identificação automática e captura de dados
API	Interface de programação de aplicações
BD	Base de dados
CSS	Cascading Style Sheets
DB	Database
FFE	Fuzzy Front-end
HTML	Hyper Text Markup Language
HS	Health Solutions
HTTP	Hypertext Transfer Protocol
IDE	Ambiente de Desenvolvimento Integrado
NCD	New Concept Development
NPD	Novo Processo de Desenvolvimento
PDA	Personal digital assistant
SGICM	Sistema de Gestão Integrado do Circuito do Medicamento
SOAP	Simple Object Access Protocol
UC	Use Case
UI	User Interface
UML	Unified Modeling Language
VC	Valor para o Cliente
WMDC	Windows Mobile Device Center

1 Introdução

Este capítulo introduz o contexto do trabalho e assinala o problema a ser resolvido, com o propósito de enquadrar o leitor no tema em questão. Neste capítulo ainda são expostos os objetivos e respetivas abordagens para que estes sejam cumpridos. Por fim, é apresentado um breve resumo sobre a análise de valor e estrutura do documento.

1.1 Contexto

Esta secção apresenta os conceitos importantes associados ao projeto. Inicialmente é introduzido o conceito de registo de consumos hospitalares. De seguida, é elaborada uma pequena apresentação sobre a empresa para a qual a solução será desenvolvida. Por fim, é feita uma breve introdução sobre a solução de registo de consumos já desenvolvida pela empresa.

1.1.1 Registo de Consumos Hospitalares

O registo de consumos hospitalares tem como principal objetivo imputar os custos de um determinado conjunto de artigos/medicamentos e posteriormente realizar os movimentos de saída de *stock* do armazém associado. Em seguida enumera-se algumas das principais vantagens associadas à utilização de um sistema informático no registo de consumos:

- Permite a racionalização de produtos nos serviços;
- Controlo mais rigoroso de todos os custos;
- Reduz desperdícios.

1.1.2 Glintt – Global Intelligent Technologies

A Glintt é uma empresa com mais de 20 anos de experiência, que atua na consultoria e serviços tecnológicos na saúde. Esta empresa dedica-se ao desenvolvimento de soluções tecnológicas e as suas soluções são utilizadas em mais de 200 hospitais e clínicas (GSG 2019).

Dentro da empresa existem diversas soluções que permitem resolver problemas de gestão na área de saúde. Entre estas, a solução Globalcare, abrange áreas tão relevantes como os meios complementares de diagnóstico e terapêutica, a farmácia - considerado o epicentro de ação dos farmacêuticos, e a logística que otimiza os serviços farmacêuticos e de aprovisionamento (Glantt, 2018).

Dentro da solução Globalcare, encontra-se o módulo de Farmácia e Logística Hospitalar que tem como propósito dar suporte a todos os intervenientes que participam no circuito do medicamento. Assegurando assim uma gestão eficiente de todo o fluxo logístico no que respeita a bens e serviços (Glantt, 2018).

1.1.3 SGICM-Mobile

O SGICM-Mobile, disponibilizado pela Glantt, é uma aplicação para Personal Digital Assistant (PDA) desenvolvida em duas vertentes, *online* – significa que o dispositivo deve ter uma conexão Wi-Fi ativa e *offline* – significa que o utilizador trabalha sem acesso ao sistema de gestão e só mais tarde faz a sincronização do seu trabalho no servidor principal.

A aplicação SGICM-Mobile utilizada na logística hospitalar, surge com o objetivo de complementar a aplicação *web*, Sistema de Gestão Integrado do Circuito do Medicamento (SGICM). A aplicação móvel tem como objetivos o registo informático mais perto da ação, encurtar o tempo de resposta a pedidos de serviços clínicos, diminuir a rutura de *stocks*, reduzir os erros humanos e, por fim, intensificar a produtividade dos utilizadores. Esta solução dispõe de vários módulos que apoiam o utilizador no registo de consumos, na elaboração de pedidos de reposição, na contagem de inventários e na satisfação de pedidos.

A principal diferença entre a aplicação que trabalha em modo *online* e a que trabalha em modo *offline* são as funcionalidades disponíveis. Caso o PDA esteja a ser utilizado em modo *offline*, as funcionalidades disponíveis não dependem de dados reais existentes no sistema principal. Por exemplo, na reposição de *stocks* e contagem de inventários, não é necessário verificar a existência de *stock* de um artigo. Mas quando o utilizador está a trabalhar em modo *online*, este pode elaborar um registo de consumos ou uma satisfação de pedidos, em que é necessário dar a saída de *stock* dos artigos utilizados. Caso estes dados não sejam sincronizados num curto espaço de tempo, pode levar à incoerência de dados. Por exemplo, caso um utilizador esteja a trabalhar no PDA, em modo *offline*, e der saída de um artigo, e outro utilizador, que esteja a trabalhar ligado ao sistema principal, fizer um pedido do mesmo artigo, os movimentos podem ficar inconsistentes.

1.2 Problema

O rigoroso mercado da saúde impõe soluções integradas, completas e atualizadas que possibilitem gerir todas as informações produzidas numa organização hospitalar. A Glantt dispõe de duas soluções móveis, que atualmente são suportadas por PDA e são utilizadas pelos clientes em diversas tarefas hospitalares.

Na logística hospitalar, são utilizados atualmente dispositivos móveis, nomeadamente os PDAs que em determinados clientes funcionam em modo *online*. Noutras situações é possível que estes dispositivos funcionem em modo *offline*. Contudo, em ambiente de armazém que muitas

vezes se situam em caves ou em locais sem acesso Wi-Fi, torna-se necessário a criação de uma aplicação móvel que permita o registo de consumos quer em modo *online* como *offline*.

As soluções SGICM-Mobile carecem de suporte tanto a nível tecnológico como aplicacional, sendo assim existem algumas limitações principalmente devido à descontinuação da tecnologia utilizada no projeto. O Windows Mobile é um sistema operativo compacto desenvolvido para dispositivos móveis como PDAs, descontinuados pela Microsoft (Warren, 2011). Por esta razão, torna-se difícil a correção de erros ou implementação de novas funcionalidades pela equipa de desenvolvimento de *software* da Glintt.

1.3 Objetivos

O objetivo deste projeto é o desenvolvimento duma solução que permita elaborar o registo de consumos hospitalares via aplicação móvel. Este projeto tem por missão desenvolver uma aplicação móvel, que permita o registo de consumos, esta deve fazer a sincronização de dados assim que tenha conetividade, sem que tenha qualquer impacto na sua utilização. Tendo em conta esta missão, surgem diferentes questões que serão respondidas ao longo desta dissertação.

Sendo estas questões:

Q1. Qual o mecanismo de sincronização de dados, que permite melhor desempenho e coerência de dados?

Q2. Qual o mecanismo de sincronização que melhor se adequa à solução a ser desenvolvida?

Q2.1. Será que a sincronização de dados deverá ser síncrona ou assíncrona?

Q2.2. Qual o padrão de armazenamento de dados que deverá ser usado?

Q2.3. Qual o padrão de transferência de dados que deve ser implementado?

Q3. Quais as vantagens e desvantagens das diferentes *frameworks* multiplataforma de desenvolvimento de aplicações móveis?

Q4. Qual a *framework* multiplataforma que deve ser adotada pela Glintt?

Chegando assim à questão final e principal:

Q5. Será possível construir um mecanismo de sincronização viável, que permita trabalhar em modo *online/offline* no contexto deste problema?

1.4 Abordagem preconizada

De seguida são apresentadas as diversas tarefas que permitem a concretização da missão do projeto:

- Interpretar o problema de sincronização de dados de acordo com as necessidades da empresa;
- Avaliar diferentes *frameworks* para o desenvolvimento da aplicação;
- Proposta de valor da solução;
- Definição de requisitos;
- Comparar e avaliar a eficácia de mecanismos de sincronização de dados;
- Comparar e avaliar a tecnologia a implementar;
- Desenhar uma solução que permita implementar o registo de consumos hospitalares;
- Implementar um mecanismo que elabore a sincronização de dados no registo de consumos;
- Avaliar os resultados da solução desenhada, mediante os testes efetuados.

1.5 Análise de valor

A realização da análise de valor vai de encontro ao objetivo principal deste trabalho que é a de trazer valor aos serviços prestados pela empresa, neste caso no desenvolvimento de aplicações, tendo em conta os custos associados e sem que haja perda de qualidade dos serviços.

A análise de valor, apresentada no terceiro capítulo, irá aplicar os modelos New Concept Development (NCD) e Analytic Hierarchy Process (AHP). Por fim, é apresentado o Modelo de negócio Canvas que estrutura o plano de negócio. O objetivo primordial desta ferramenta é definir uma proposta de valor através da análise de nove pontos estratégicos interligados.

1.6 Estrutura do Documento

Este documento está dividido em seis capítulos, que se encontram repartidos em secções e subsecções com o objetivo de apresentar o detalhe do conteúdo a ser estudado. Neste presente capítulo é realizada uma introdução ao tema com vista a enquadrar o leitor na compreensão correta do problema e definir os objetivos pretendidos. No segundo capítulo, é descrito o estado de arte de soluções e tecnologias atualmente usadas no mercado. Adicionalmente, no terceiro capítulo é elaborada uma análise de valor, que pretende contextualizar mais detalhadamente o problema e analisar o potencial valor da solução. Posteriormente, no quarto capítulo, é apresentada a análise e *design* da solução desenvolvida. No quinto capítulo, avalia-se a solução desenvolvida através de testes e questionários. Por fim, no sexto capítulo são apresentadas as conclusões e respostas às questões definidas na secção 1.3.

2 Contexto, Problema e Estado de arte

Neste capítulo serão abordados os conceitos teóricos relativos ao tema, com o objetivo de contextualizar o leitor. Ao longo deste capítulo serão apresentados os diferentes mecanismos de sincronização existentes. Durante a apresentação destes conceitos será referenciada a questão **Q1**, em que será possível perceber qual dos mecanismos terá a melhor desempenho e coerência de dados. Conjuntamente serão apresentadas as diferentes tecnologias para desenvolvimento de aplicações móveis a serem estudadas, o que trará resposta à questão **Q3**.

2.1 Contexto

As instituições de saúde prestam cuidados de saúde à população, para que estas tenham uma melhor qualidade de vida. As soluções disponibilizadas pela Glintt permitem que as instituições sejam eficientes ao ponto de oferecerem melhores cuidados de saúde e com uma maior eficiência financeira.

Uma das áreas de suporte à prestação de cuidados é a logística hospitalar, que efetua ações desde a avaliação da necessidade do bem, aquisição, gestão do armazenamento, expedição e consumo. A logística hospitalar efetua toda a gestão de um catálogo de produtos de áreas como medicamentos, reagentes, outros produtos farmacêuticos, produtos de consumo clínico, produtos de hotelaria, produtos administrativos e produtos de suporte à atividade de manutenção das instituições.

A informação do registo de utilização de um bem é assim fulcral para a área de prestação de cuidados como medida de segurança. É necessário ter conhecimento por quem e quando foi utilizado um determinado produto. Por esta razão, o registo de consumos mais próximo do seu ponto de utilização é fulcral. É importante perceber que a exatidão de *stocks* é a base de uma gestão logística otimizada, que consegue obter o bem correto no tempo certo e ao melhor custo. Estes pontos devem ser tidos em conta para que as instituições consigam desempenhar o seu propósito de forma correta.

As instituições hospitalares quer pela sua natureza arquitetónica, quer pelas suas restrições de segurança e infraestrutura de rede Wi-Fi, o registo da informação do consumo de um determinado bem nem sempre ocorre a quando da utilização efetiva. Muitas vezes o registo de

movimentos é efetuado á priori ou á posteriori tendo como base informação transcrita de papel. Esta transcrição leva a erros de interpretação ou erros de digitação tornando a informação desfasada no tempo ou incorreta e de pouca utilidade futura.

Outro fator crítico é o tempo despendido pelos recursos para efetuar o registo da informação, tempo esse que devia ser otimizado para ser utilizado na prestação de cuidados ou em atividades de suporte às mesmas, melhorando assim a eficiência e objetivos das instituições.

Assim com vista a ajudar a mitigar e muitas vezes a eliminar as situações previamente identificadas, surge a necessidade de existir uma solução de suporte ao registo da utilização de bens. Esta nova solução deve permitir que seja efetuado o registo digital, o mais próximo da sua utilização, mesmo em locais críticos ou sem cobertura de rede.

2.1.1 Problema

Conforme referido na introdução da secção 2.1, em ambiente hospitalar nem sempre é possível usufruir de uma conexão à Wi-Fi, sendo que a Glintt pretende construir uma solução que funcione em modo *offline* e *online*. Um dos seus objetivos é que os seus utilizadores tenham uma experiência flexível quando a conexão estiver lenta, oscilante ou mesmo quando esta é inexistente.

Este problema reflete-se tanto na aplicação SGICM-Mobile *online* como na versão *offline*, pois a existência de duas aplicações com funcionalidades diferentes torna a experiência do utilizador um pouco ambígua. Ou seja, caso o utilizador disponha das duas aplicações no seu PDA, estes têm de conhecer inteiramente a aplicação para saber qual pretendem abrir para começar a trabalhar. Para além disto, o utilizador pode estar a trabalhar no SGICM-Mobile *offline* num local em que têm conetividade, e os seus registos não são automaticamente atualizados para o servidor principal. Nestes casos existe a necessidade do utilizador, ligar o PDA a um computador e só depois elaborar manualmente a sincronização de dados. Quanto à aplicação SGICM-Mobile *online*, o utilizador pode estar numa zona em que não existe conetividade, então este fica imediatamente excluído de poder elaborar o seu trabalho. Sendo então necessário dirigir-se a uma zona com acesso a Wi-Fi, ou registar em papel os seus passos e posteriormente transcrever o seu trabalho.

Outro problema, que afeta principalmente a equipa de desenvolvimento e manutenção de soluções da Glintt, é a descontinuação do sistema operativo Windows Mobile. As aplicações SGICM-Mobile foram implementadas para serem utilizadas neste sistema operativo nas versões Windows Mobile CE e na versão Windows Mobile 6.5. No entanto, o programa ActiveSync, que permite a sincronização e conexão do dispositivo móvel com um computador, também foi descontinuado pela Microsoft (Microsoft, 2018). Tendo sido lançado o seu substituto, o Windows Mobile Device Center (WMDC), que permite a sincronização da mesma forma que o ActiveSync. No entanto, este programa apresenta problemas de compatibilidade quando o sistema operativo utilizado no computador é Windows 10.

Por vezes, chega mesmo a ser necessário completar uma sequência de passos até conseguir a conetividade entre o PDA e o computador. Este processo moroso, leva a equipa de desenvolvimento a gastar tempos elevados, sempre que é necessária a correção ou alteração de um comportamento, tendo assim custos acrescidos associados a esta necessidade.

2.1.2 Propósito

O propósito deste trabalho é desenvolver uma aplicação móvel que trabalhe em modo *online* e *offline*. A Glintt pretende implementar esta solução, com a intenção de proporcionar aos seus clientes o registo de consumos sem se terem de preocupar se existe uma conexão Wi-Fi em certos pontos das suas instalações. O objetivo final será implementar o método de sincronização de dados que melhor se adequa à logística hospitalar. Este mecanismo de sincronização deverá poder ser implementado em diferentes módulos logísticos para além do registo de consumos hospitalar.

Consequentemente, é identificada a necessidade de implementar a solução utilizando uma *framework* multiplataforma móvel. A utilização de uma tecnologia recente permite um ambiente de desenvolvimento ágil e de fácil manutenção da nova solução.

Os clientes da Glintt estão no segmento do mercado da área da saúde, nomeadamente serão os centros hospitalares. Mais concretamente, poderão ser considerados os hospitais, que já utilizam a solução disponível pela Glintt para PDA, ou os hospitais que ainda não aderiram a esta funcionalidade.

De seguida, são enumeradas as quatro partes interessadas no desenvolvimento deste projeto:

- Centro Hospitalar – Ou hospitais, são os clientes diretos da Glintt que já utilizam algumas das soluções fornecidas. Estes irão beneficiar desta solução, pois permitirá ter uma fase da logística hospitalar computadorizada com tecnologias recentes que admitem utilizar outros dispositivos móveis para além dos PDAs.
- Utilizador – Que pretende ter acesso a uma aplicação móvel, no seu dispositivo. Este pretende fazer registos de consumos em locais sem conectividade Wi-Fi, e que todo o seu trabalho seja sincronizado assim que atingir um local com conexão.
- Empresa – A Glintt, que pretende perceber as vantagens de elaborar a conversão tecnológica da aplicação já existente.
- Equipa de desenvolvimento – A equipa de desenvolvimento irá incluir tecnologias inovadoras e reutilizar algoritmos de modo a não alterar a lógica de negócio do módulo de consumos hospitalares.

2.2 Estado de Arte em Soluções Existentes

Atualmente a Glintt oferece quatro soluções que permitem elaborar o registo de consumos hospitalares em diferentes situações. Nesta secção, será apresentada uma breve descrição sobre as mesmas, dando ênfase às aplicações SGICM-Mobile.

Por fim, será exposto um produto concorrente da Glintt, apresentada a empresa que o desenvolveu e efetuada uma breve análise comparativa com a solução da Glintt.

2.2.1 Soluções Glintt

Nesta subsecção pretende-se ilustrar através da Figura 1, como é elaborada a interação entre as diferentes soluções da Glintt, que permitem o registo de consumos. Através do diagrama de implantação da Figura 1, é possível verificar que o registo de consumos pode ser elaborado a partir de quatro pontos de entrada. No entanto todos os registos são guardados na BD principal, que se encontra num servidor geral.

A Glintt disponibiliza quatro aplicações que permitem o registo de consumos hospitalares. A solução SGICM, que é utilizada via *desktop* através de um *web browser*, é considerada a aplicação principal pois permite a gestão de toda a logística hospitalar. No entanto, a Glintt também disponibiliza a solução Tátil, que permite o registo de consumos via *tablet* através de um *browser*. Por fim, existem as soluções SGICM-Mobile *online* e *offline*, que surgiram para dar resposta à necessidade de elaborar registos via PDA.

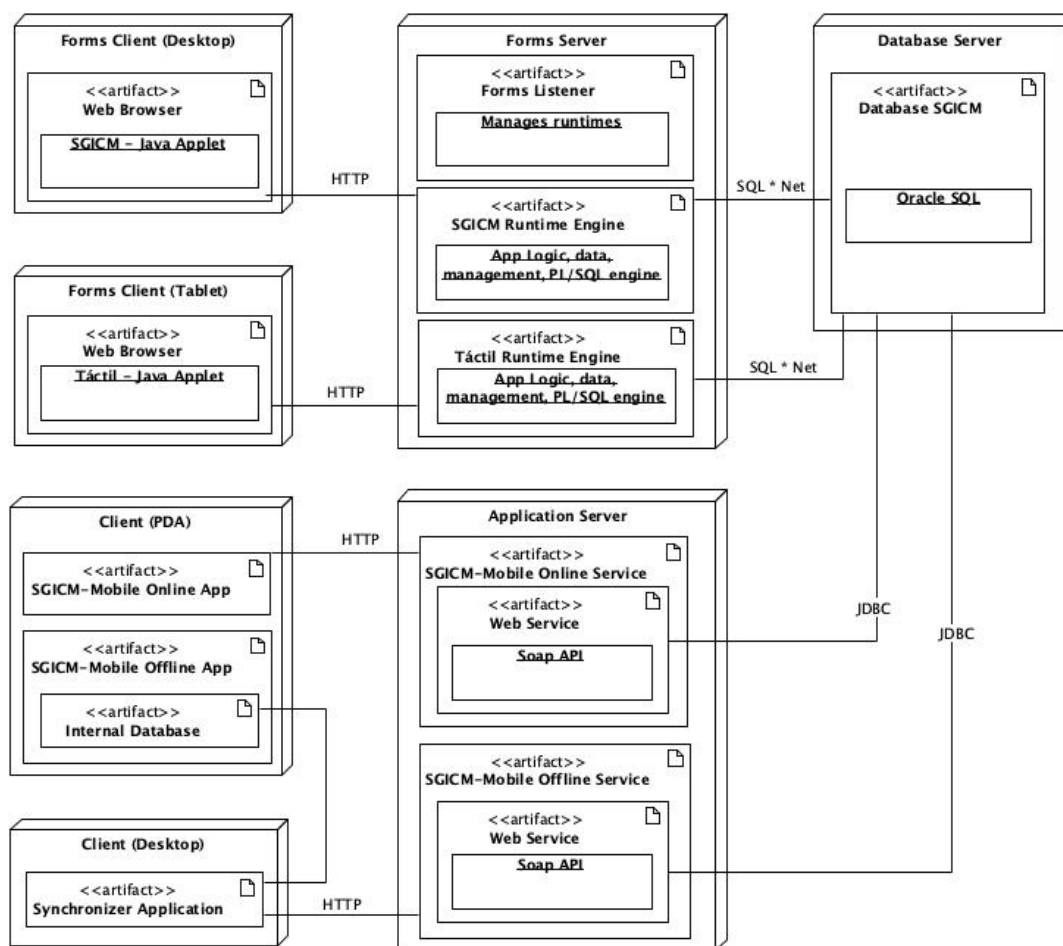


Figura 1 – Diagrama de implantação do sistema de registo de consumos da Glintt

Solução SGICM

Conforme mostra a Figura 1, o SGICM é uma aplicação *web* desenhada para ser utilizada via *desktop* através de um *web browser*. Este sistema foi desenvolvido utilizando a tecnologia Oracle Forms, que utiliza uma base de dados Oracle (*Database SGICM*). A aplicação é bastante

complexa, pois permite a gestão do fluxo logístico hospitalar. Deste a criação de encomendas até à sua receção e todos os seus processos adjacentes, sendo o registo de consumos um desses processos.

Solução Tátil

Esta solução, concebida para ser utilizada em dispositivos móveis, tem como objetivo simplificar e agilizar a introdução de consumos em conta corrente no bloco operatório, internamento e urgência. Esta solução tem a particularidade de associar os consumos diretamente aos doentes. Conforme mostra a Figura 1, são utilizados *tablets* para aceder à aplicação que foi desenvolvida utilizando a tecnologia Oracle Forms. O enfermeiro que assiste no bloco operatório vai colocando os materiais/medicamentos ou *kits* que estão a ser gastos à medida que o procedimento está a ocorrer.

Solução SGICM-Mobile Online

Esta versão do SGICM-Mobile, permite o registo de consumos através de um PDA, o que se torna bastante conveniente. Esta conveniência também advém destes dispositivos permitirem a leitura ótica, o que torna o registo de artigos segura e rápida.

Esta versão da aplicação SGICM-Mobile, é totalmente *online*, sem nenhum tipo de sincronismo. A ligação à base de dados SGICM é feita através de um *web service*, conforme a Figura 1. Esta solução permite utilizar todas as parametrizações do utilizador elaboradas a partir da aplicação SGICM. A principal vantagem desta solução é não existir perda de dados, pois todos os dados são guardados diretamente na base de dados SGICM, devido a apenas ser possível registar movimentos quando existe conectividade com o servidor. Como não é necessário qualquer sincronismo, no final da confirmação é gerado um consumo. Ou seja, é devolvido ao utilizador o número do documento gerado, que é o identificador único daquele registo.

O SGICM-Mobile foi criado segundo as seguintes diretivas totalmente *online*, sem qualquer tipo de sincronismo:

- Ligação à base de dados SGICM;
- Mecanismos de recuperação de informação em caso de falha da aplicação;
- Devolução da referência do documento gerado;
- O acesso a aplicação é parametrizado;
- Extração de mais informação do código de barras 2D (caso o dispositivo permita a leitura de código 2D), códigos GS1.

De seguida, enumera-se as principais funcionalidades do SGIM-Mobile *online*, que é composto pelos módulos:

- Registo de consumos ao serviço e/ou doente;
- Satisfação de pedidos;

- Pedidos de reposição de *stock* (níveis);
- Pedidos de reposição Standard;
- Pedidos por localização (dupla gaveta);
- Contagem de Inventários;
- Devolução ao Serviço e/ou doente;
- Registo de entradas;
- Transferências entre armazéns;
- Satisfação de pedidos através de lista de *picking*.

Também é importante referir que tecnologicamente esta aplicação foi desenvolvida utilizando C# Windows Forms, da Microsoft. Para a implementação do seu *web service*, que fornece os dados à aplicação, foi usado o protocolo SOAP, utilizando a linguagem C#.

Solução SGICM-Mobile Offline

Esta solução permite trabalhar através de um PDA, o que se torna bastante conveniente por estes permitirem a leitura ótica para o registo de artigos de uma forma segura e rápida. Esta versão da aplicação SGICM-Mobile, é totalmente *offline*, o que significa que as ações elaboradas durante o trabalho de um utilizador têm de ser posteriormente sincronizadas.

Conforme mostra a Figura 1, esta solução inclui uma aplicação *desktop*, chamada Synchronizer, que sincroniza os dados da BD (Base de dados) interna do PDA com os dados existente na BD principal do hospital (*Database* SGICM). No final do dia, quando os utilizadores terminam o seu trabalho, o PDA é ligado a um computador e mediante uma ação do utilizador irá ser feita a sincronização de dados.

Também é importante referenciar que tecnologicamente esta aplicação móvel foi desenvolvida utilizando C# Windows Forms. A aplicação Synchronizer foi desenvolvida em Windows Forms e utiliza um *web service* para efetuar a sincronização de dados. Este *web service* implementa o protocolo SOAP, utilizando a linguagem C#.

2.2.2 Solução BIQ Health Solutions

De seguida irá ser apresentado um estudo sobre um dos principais concorrentes da Glintt, a BIQ Health Solutions, que dispõe de uma solução móvel para o registo de consumo no âmbito da logística hospitalar.

Atualmente, a BIQ Health Solutions possui no seu portfólio de clientes, hospitais do país, incluindo as ilhas, tendo uma abrangência total da confirmação do seu "*know-how*" a nível nacional. A BIQ Health Solutions tem por visão ser uma empresa que fornece soluções de redução de custos e segurança do paciente na indústria da saúde. Esta tem por missão apoiar os seus distribuidores, parceiros e clientes finais na comercialização e implementação das suas soluções, recorrendo às melhores práticas de forma a alcançar o maior ganho possível para

todos os *stakeholders*, em termos de qualidade, segurança e redução de custos na indústria da saúde (BIQHS, 2018). De seguida é apresentada a solução MAPP, disponibilizada pela BIQ Health Solutions que permite o registo de consumos a partir de um *smartphone*.

MAPP – Mobile App Platform

Nesta aplicação, são fornecidas todas as soluções BIQ HS, que estão disponíveis para a plataforma Android. Neste posto de trabalho móvel para enfermeiros é possível realizar todos os registos de forma segura junto ao doente, promovendo a proximidade entre os doentes e os enfermeiros e aumentando a qualidade dos registos de enfermagem (BIQHS, 2018).

A aplicação MAPP disponibiliza os módulos:

- Recolha de amostras para estudo laboratorial;
- Registo de administração terapêutica;
- Registo de movimentação logística;
- Registo de sinais vitais;
- Registo de atos de enfermagem.

O módulo de registo de movimentação logística permite o controlo de existências através do registo de consumos, introduz uma série de mecanismos que ajudam a combater as ruturas de stocks, e promovem a diminuição do desperdício. Esta funcionalidade permite gerir, de forma automatizada, toda a movimentação dos artigos, desde que entram no armazém central, até que são realmente consumidos, garantindo sempre a existência de um inventário permanente (BIQHS, 2018).

2.2.3 SGICM-Mobile vs MAPP

Por fim, irá ser elaborada uma pequena comparação entre a solução disponibilizada pela Glintt e a solução concorrente, com objetivo de perceber as vantagens e desvantagem entre as aplicações disponibilizadas por estas instituições.

Antes de iniciar esta análise é importante referir que tanto as soluções da Glintt, como a solução disponibilizada pela BIQ Health Solutions, são soluções vendidas diretamente aos clientes, neste caso aos hospitais, o que torna ligeiramente difícil a análise das soluções pois não existe acesso à aplicação MAPP. Será então apresentada na Tabela 1, uma comparação baseada na informação recolhida, com o objetivo de analisar as características diferenciadoras.

Na Tabela 1, é possível verificar que as principais diferenças entre a aplicação SGICM-Mobile e a MAPP, são os sistemas operativos que suportam a aplicação, o tipo de dispositivo e a sincronização de dados assim que existe conectividade. Através desta comparação percebemos que disponibilizar a aplicação SGICM para o uso em *smartphone* e existir um mecanismo de sincronização de dados, é essencial para continuar a ter a preferência dos clientes.

Tabela 1 – Comparação entre a solução SGICM-Mobile e MAPP.

Caraterísticas	SGICM-Mobile	MAPP
Sistema operativo que suporta a aplicação	Windows Mobile 6.5, Windows Mobile CE	Android
Tipo de dispositivo móvel	PDA	<i>Smartphone</i>
Modulo de registo de consumos	Sim	Sim
Leitura Ótica	Sim	Sim
Sincronização de dados assim que existe conetividade	Não	Sim
Capacidade de integração com outras aplicações	Sim	Sim

2.3 Abordagens Utilizadas na Sincronização de Dados

Nesta secção irão ser dados a conhecer diferentes mecanismos de sincronização de dados, os padrões de armazenamento e, por fim, os padrões de transferência de dados.

Os dispositivos móveis nem sempre têm uma conexão Wi-Fi ativa, por isso, algumas aplicações necessitam de funcionar em modo *offline*, para que os utilizadores possam continuar o seu trabalho mesmo quando atingem zonas sem conetividade. Para solucionar este problema é necessário o armazenamento de dados local, mas também é essencial uma forma de atualizar o servidor de BD principal com as alterações efetuadas em modo *offline*.

2.3.1 Padrões de Mecanismos de Sincronização de Dados

Nesta secção é abordado o tema de padrões de mecanismos de sincronização de dados, sendo que são apresentadas duas abordagens uma assíncrona e outra síncrona.

Sincronização de Dados Assíncrona

O objetivo principal deste método é que a sincronização de dados não seja percebida pelo utilizador, ou seja, que não seja perceptível via interface gráfica que está a existir uma atualização de dados.

No entanto, é necessário definir duas alturas essenciais na sincronização de dados, o descarregamento de dados para o dispositivo móvel e o carregamento dos dados para o servidor de BD principal. O carregamento de dados para o servidor principal, pode ser despoletado quando o dispositivo encontra uma rede Wi-Fi, ou podem ser definidos eventos que levem ao carregamento dos dados (McCormick & Schmidt, 2012).

Quanto ao descarregamento de dados, as preocupações podem ser outras, nomeadamente a performance da aplicação pode ser prejudicada devido à quantidade de dados que estão a ser descarregados para o dispositivo. A quantidade de dados existentes num servidor principal irá ser imensas vezes maior que os dados criados ou alterados num dispositivo. Por exemplo, nesta situação uma abordagem poderia ser o descarregamento de dados ser feito quando o utilizador abre a aplicação. Mas existem outras opções, como fazer a sincronização quando a aplicação não está a correr em primeiro plano (McCormick & Schmidt, 2012).

A Figura 2, apresenta como a sincronização de dados é processada assincronamente. Assume-se que existem três estados principais: antes da sincronização, quando a sincronização é iniciada e a ação de sincronizar. Consegue-se também observar que quando é lançado um evento de sincronização, este lança um evento de sincronização assíncrona, o que permite voltar imediatamente ao estado de usabilidade da aplicação, não existindo impacto na interface gráfica para o utilizador.

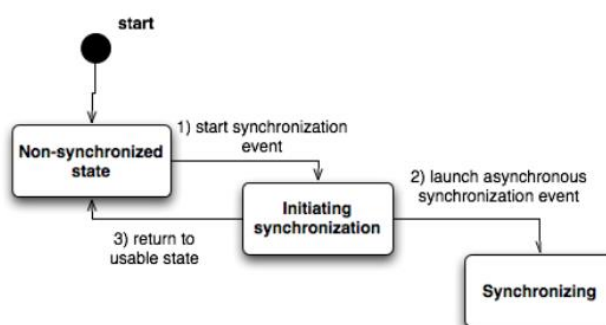


Figura 2 – Abordagem assíncrona de sincronização de dados (McCormick & Schmidt, 2012)

Alguns exemplos reais, são as aplicações Facebook e Twitter, que permitem aos utilizadores consultar informação enquanto é elaborada a sincronização de dados, aplicando assim este padrão de sincronização de dados (McCormick & Schmidt, 2012).

Sincronização de Dados Síncrona

Este método de sincronização de dados é síncrono, o que significa que a interface do utilizador irá ficar bloqueada enquanto a sincronização de dados acontece.

Este método é normalmente utilizado quando as aplicações dependem de dados, ou seja, necessitam que os dados sejam reais. Este tipo de aplicações não deixa o utilizador avançar até que os dados sejam sincronizados, pois deixar que este avanço aconteça pode significar que os dados entrem em estados de erro e de mal funcionamento (McCormick & Schmidt, 2012).

Esta perspetiva é mais utilizada em aplicações que trabalham maioritariamente em modo *online*, pois o trabalho do utilizador é sequencial. Ou seja, se as tarefas têm de ser elaboradas segundo uma ordem, significa que o utilizador necessita de uma resposta a uma tarefa antes de elaborar a próxima. Neste caso, a sincronização não deveria ser assíncrona pois poderia levar a erros ou perda de informação (McCormick & Schmidt, 2012).

A Figura 3, apresenta como a sincronização de dados é processada sincronamente. Assume-se que existem três estados principais: estado de usabilidade da aplicação, quando a sincronização é iniciada e a espera pela sincronização. Consegue-se também observar que quando é lançado um evento de início de sincronização, este lança um evento síncrono, o que apenas permite voltar ao estado de usabilidade da aplicação quando a sincronização é terminada.

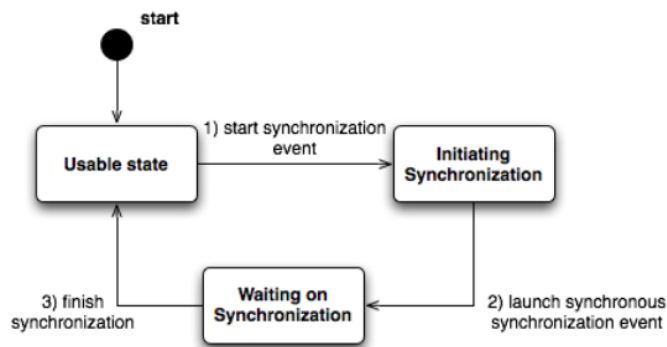


Figura 3 – Abordagem síncrona de sincronização de dados (McCormick & Schmidt, 2012)

A aplicação Spotify armazena a data do último pagamento, depois desse dia verifica de forma síncrona o estado do pagamento, para alertar o utilizador de que a conta expirou. Este exemplo usa uma abordagem síncrona para garantir que o utilizador não usufrui de serviços que não foram pagos (McCormick & Schmidt, 2012).

2.3.2 Padrões de Armazenamento de Dados

Muitas vezes, as aplicações móveis têm restrições como a velocidade de rede e a largura de banda. Da mesma forma, a capacidade de armazenamento local é muitas vezes limitada em relação ao conjunto de dados necessários a sincronizar. Nesta secção são abordados os temas armazenamento parcial e total de dados.

Armazenamento Parcial de Dados

Este tipo de armazenamento de dados apenas sincroniza os dados necessários para uma ação, ou seja, sempre que o utilizador demonstra intenção de elaborar uma ação são descarregados os dados para o dispositivo. Este tipo de sincronização minimiza o armazenamento de dados, o que melhora a performance da aplicação pois a sincronização de uma base de dados completa pode ter um peso grande e comprometer a usabilidade da aplicação. No entanto, sempre que o utilizador realizar uma ação irá ser despoletada um pedido de dados, o que também se poderá traduzir numa má experiência para utilizador, pois se a conexão à rede estiver lenta este terá de esperar que os dados sejam carregados (McCormick & Schmidt, 2012).

As vantagens deste método podem ser a redução da necessidade de espaço de armazenamento e a possibilidade de definição de diferentes níveis de sincronização, o que significa melhor performance. As desvantagens aparecem quando existe uma má conexão à rede, pois a interface pode bloquear enquanto os dados são carregados (McCormick & Schmidt, 2012).

A Figura 4, apresenta os dois casos que abrangem o armazenamento parcial. A primeira sequência acontece quando é necessário acesso a dados e estes não estão armazenados no dispositivo. É possível observar que existe um pedido de dados (*network request*), que tenta obter os dados necessários e só continua as suas ações quando os recebe. A segunda sequência mostra que quando os dados já estão armazenados no dispositivo, não necessita de elaborar uma *network request*.

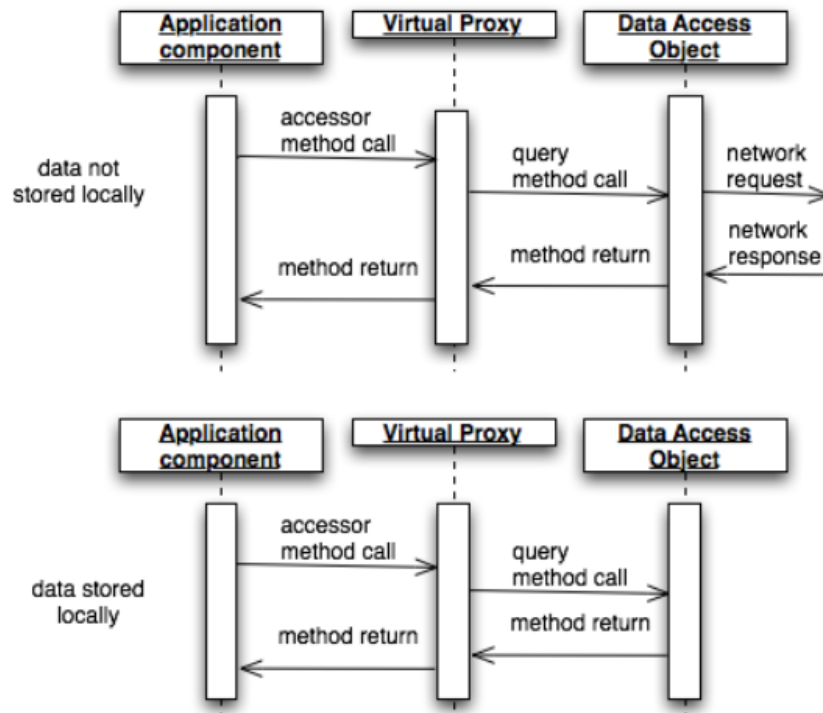


Figura 4 – Armazenamento parcial de dados (McCormick & Schmidt, 2012)

A aplicação Google Maps exibe um mapa relevante num determinado nível de granularidade e faz o *download* de novos blocos conforme estes sejam necessários. Por exemplo, quando o mapa é ampliado ou reduzido. Esta abordagem elimina a necessidade de armazenar o mapa inteiro, mas requer conectividade de rede (McCormick & Schmidt, 2012).

Armazenamento Total de Dados

Neste caso a abordagem é sincronizar e guardar todos os dados necessários, o que faz com que a aplicação tenha um tempo de resposta mais rápido. Este tipo de abordagem é usado quando a conexão Wi-Fi possa estar comprometida. A não obtenção de dados pode ser um problema na interatividade do utilizador com a aplicação, logo surge esta abordagem (McCormick & Schmidt, 2012).

A principal vantagem desta abordagem é, caso não exista conexão com Wi-Fi, o utilizador pode continuar o seu trabalho. A desvantagem deste método é a capacidade de armazenamento do dispositivo, pois esta pode ficar comprometida devido à quantidade de dados sincronizados (McCormick & Schmidt, 2012).

A Figura 5, apresenta a abordagem de armazenamento total fazendo uma distinção entre duas ações diferentes, a sincronização e obtenção de dados. A ação de sincronização causa uma solicitação de rede e a obtenção retorna os dados armazenados no dispositivo.

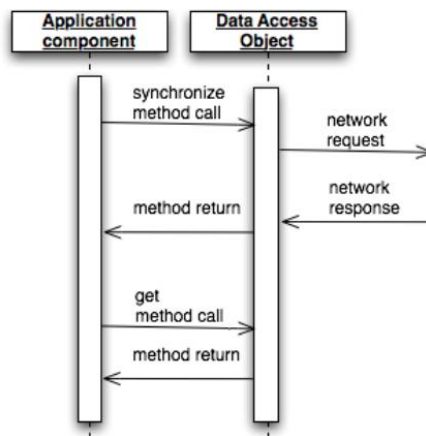


Figura 5 – Armazenamento total de dados (McCormick & Schmidt, 2012)

A aplicação Dropbox usa este padrão para permitir que um utilizador aceda a ficheiros quando a largura de banda é baixa ou quando não existe uma conexão à rede (McCormick & Schmidt, 2012).

2.3.3 Padrões de Transferências de Dados

Nesta secção irão ser apresentados os diferentes padrões de transferências de dados, nomeadamente os métodos de transferência de dados total, de transferência por *timestamp* e transferência matemática ou algorítmica de dados. A largura de banda de rede muitas vezes é uma preocupação a ter no desenvolvimento de aplicações móveis. Portanto, os desenvolvedores devem implementar as aplicações com mecanismos para minimizar os recursos da tarefa de sincronizar dados.

Transferência Total de Dados

Neste caso, quando acontece um evento de sincronização são sincronizados todos os dados entre o dispositivo móvel e o servidor principal. Esta técnica de sincronização de dados é a mais simples de implementar o que pode ser considerada uma vantagem. No entanto, esta abordagem de transferência de dados têm uma desvantagem, pode causar redundância nos dados, pois se apenas certos dados foram alterados estar a sincronizar dados que não foram alterados é um desperdício de recursos (McCormick & Schmidt, 2012).

A Figura 6 apresenta como é efetuada uma transferência total de dados, em que apenas existem o estado normal da aplicação e é elaborado um pedido para obtenção de todos os dados.



Figura 6 – Transferência total de dados (McCormick & Schmidt, 2012)

Num contexto real esta abordagem é usada quando é detetado um erro num conjunto de dados. Em vez de usar um esquema de reconciliação complexo, é utilizada uma transferência total para substituir facilmente o conjunto de dados.

Transferência por *Timestamp* de Dados

Neste método de transferência de dados, somente as partes do conjunto de dados alteradas desde a última sincronização são transferidas entre o dispositivo móvel e o servidor principal. Este tem em conta o registo de data e hora da última alteração. Com a questão da velocidade da rede e largura de banda em dispositivos móveis, a quantidade de dados transferidos para reconciliar conjuntos de dados entre um dispositivo e um sistema remoto deve ser minimizado.

A vantagem é a menor utilização de largura de banda que o método de transferência total. Quanto às desvantagens deve ser dada atenção à fonte dos *timestamps*, pois é importante manter a origem dos *timestamps* consistentes. A sincronização pode se tornar inconsistente caso sejam usados diferentes *timestamps*. Um registo com *timestamp* também tem dificuldades caso os dados tenham sido eliminados do sistema remoto, quando o dispositivo tenta transferir os dados excluídos estes não existem para uma comparação de registo de data e hora (McCormick & Schmidt, 2012).

A Figura 7 apresenta a transferência por *timestamp*, em que inicialmente existe um pedido que tem anexado o *timestamp* da solicitação. De seguida, o servidor irá avaliar se existe algum conjunto de dados a ser sincronizado. Caso existam dados recentes e que ainda não foram sincronizados, o servidor envia somente os dados que foram adicionados ou alterados para o dispositivo, desde a data da última sincronização bem-sucedida.

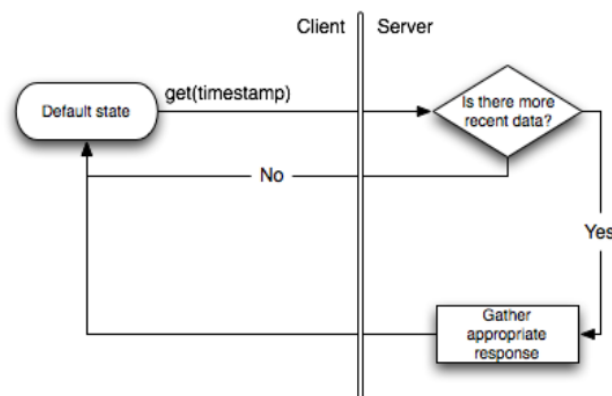


Figura 7 – Transferência por *timestamp* de dados (McCormick & Schmidt, 2012)

Alguns exemplos reais, são as APIs públicas do Twitter e do Facebook, que oferecem aos desenvolvedores a capacidade de recuperar os *posts* ao escolher uma data e hora, suportando assim a transferência por *timestamp* (McCormick & Schmidt, 2012).

Transferência Matemática ou Algorítmica de Dados

Na transferência matemática ou algorítmica de dados, apenas existe sincronização das partes do conjunto de dados alteradas desde a última sincronização. A largura de banda e a velocidade da rede são preocupações, portanto, a transferência total não pode ser usada para sincronizar um conjunto de dados entre um dispositivo e um sistema remoto. Neste tipo de abordagem pode ser desenvolvido um algoritmo ou pode ser utilizado um algoritmo matemático já existente (McCormick & Schmidt, 2012).

A vantagem deste método é usar potencialmente a menor largura de banda em comparação com os métodos *timestamp*. Com um método matemático ou algorítmico, é possível dividir os dados em blocos, antes de os transferir, reduzindo a largura de banda usada (McCormick & Schmidt, 2012).

Uma das desvantagens dos métodos matemáticos é que são altamente dependentes do contexto. A reutilização de código pode não ser aplicável, pois o método de sincronização é diferente para diferentes tipos de dados. Os métodos matemáticos ou algorítmicos geralmente também requerem mais tempo para serem desenvolvidos.

A Figura 8, mostra que o método de transferência matemática ou algorítmica de dados é idêntico ao método de transferência de dados por *timestamp*. A principal diferença existente, é um algoritmo que calcula as diferenças nos conjuntos de dados, determinando assim o esquema de dados ideal da sincronização. Conforme referido, este tipo de transferência de dados tem melhor desempenho que a transferência de dados por *timestamp*, pois calcula o esquema de dados ideal da sincronização, respondendo assim à **Q1**.

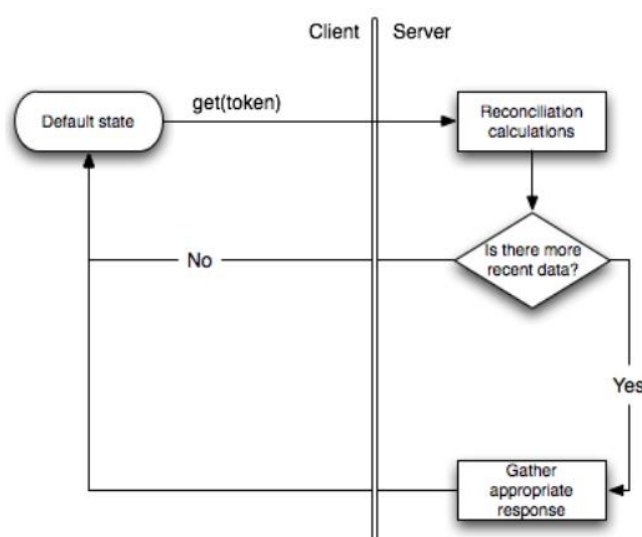


Figura 8 – Transferência matemática de dados (McCormick & Schmidt, 2012)

Em contexto real, esta abordagem é utilizada por aplicações de videoconferência como o Skype. Esta aplicação usa compactação de vídeo baseada em diferenças perceptíveis pelo Homem. O sistema remoto armazena os dados e usa as metodologias matemáticas para determinar o esquema de codificação ideal (McCormick & Schmidt, 2012).

2.4 Abordagens de Implementação em Aplicações Móveis

Nesta secção, pretende-se elaborar um estudo dos diferentes tipos de aplicações móveis existentes e de *frameworks* multiplataforma. Entre estas Xamarin, React Native e Ionic, com o objetivo de analisar as possibilidades tecnológicas para a realização do projeto.

2.4.1 Tipos de Desenvolvimento de Aplicações Móveis

Para entender o desenvolvimento de aplicações móveis, é necessário compreender que existem diferentes tipologias de aplicações, nomeadamente estas podem ser nativas, *web* e híbridas. É imprescindível uma avaliação cuidada das várias opções tecnológicas, com o propósito de escolher qual se adequa melhor ao problema. A escolha da tipologia pode ter em conta fatores como o sistema operativo, funcionalidades, custo e tempo de desenvolvimento.

Aplicações Nativas

As aplicações nativas são desenvolvidas unicamente para um único sistema operativo móvel, portanto, são nativas para uma determinada plataforma ou dispositivo. Uma aplicação construída para sistemas como iOS, Android e Windows Phone não podem ser usadas numa plataforma diferente da que foi construída (M2Mobi, 2018).

A principal vantagem das aplicações nativas é o alto desempenho e a garantia de uma boa experiência do utilizador. Além de que, garantem o acesso a uma ampla gama de APIs (Interface de programação de aplicações) que melhora o uso da aplicação (Armour, 2018).

Um dos pontos negativos das aplicações nativas, é um custo mais alto quando é necessário o desenvolvimento de réplicas da aplicação para diferentes sistemas operativos. Consequentemente, o suporte separado e manutenção para diferentes tipos de aplicações, resulta num preço superior do produto (M2Mobi, 2018).

Dentro das aplicações nativas, existe o desenvolvimento de aplicações multiplataforma que são escritas numa única linguagem de programação, mas estão disponíveis em várias plataformas. Estas podem ser implementadas utilizando uma *framework*, como Ionic, Xamarin, React Native, etc. Estes são apenas alguns exemplos das muitas *frameworks* disponíveis. Depois do código estar desenvolvido, este é encaminhado para um *middleware*. O *middleware* traduz o código para as APIs nativas de plataformas como iOS e Android (M2Mobi, 2018).

Devido ao facto das aplicações multiplataforma serem escritas num único código, estas soluções são económicas e fáceis de implementar. Uma desvantagem das aplicações multiplataforma é que dependem do *middleware* que traduz o código escrito para um ambiente nativo. Em alguns casos, uma atualização do *middleware* é necessária para que um erro seja corrigido, dependendo assim de outras entidades que controlam o *middleware* (M2Mobi, 2018).

Aplicações Híbridas

Uma aplicação híbrida é a combinação de uma aplicação nativa e de um *website* para dispositivos móveis. Geralmente, uma aplicação híbrida contém uma estrutura nativa, mas também possui páginas de um *website* que são carregadas na aplicação. Por outras palavras, as *webpages* são empacotadas numa aplicação nativa (M2Mobi, 2018).

Por conseguinte, as aplicações híbridas são consideradas relativamente simples e rápidas de desenvolver, sendo uma clara vantagem. É possível desenvolver as páginas principais de uma aplicação nativamente e implementar as páginas adicionais como *webpages*. Por fim, uma aplicação híbrida tem acesso às funções nativas de um *smartphone*, como a câmara (M2Mobi, 2018). Por outro lado, as aplicações híbridas não têm o desempenho, a velocidade e a

otimização igual quando comparadas com aplicações nativas. Também há uma grande diferença na experiência do utilizador.

Aplicações Web

As aplicações *web* são geralmente escritas em tecnologias como HyperText Markup Language (HTML), JavaScript e Cascading Style Sheets (CSS). Estas utilizam um *browser* para serem executadas, o que pode ser considerado uma vantagem pois, caso seja exigido, permite o acesso em diversas plataformas. Para além disso, como todos os dados são guardados num servidor, os utilizadores têm acesso a partir de qualquer dispositivo sempre que existir conexão com a Internet. É por isso que o uso de aplicações *web* com uma conexão fraca resultaria em uma experiência insatisfatória do utilizador. Outra desvantagem é não poderem aceder a APIs nativas do dispositivo (Glas, 2018).

Comparação de Tipo de Desenvolvimento para Aplicações Móveis

Após terem sido explicados os diferentes tipos de desenvolvimentos para aplicações móveis, pretende-se fazer uma análise que vista comparar os diferentes tipos de desenvolvimento. A Tabela 2, apresenta como os diferentes tipos de desenvolvimento móvel respondem a diferentes características (PINTERAC, 2018). Tendo em conta a informação recolhida, é perceptível que o desenvolvimento nativo permite as aplicações terem uma performance elevada, no entanto com um custo elevado caso seja necessário implementar a mesma aplicação para diferentes plataformas. Aqui surge a hipótese de utilizar uma *framework* multiplataforma, que reduz os custos e mantém o comportamento nativo da aplicação.

Tabela 2 – Comparação entre tipo de desenvolvimentos para aplicações móveis

Caraterísticas	Nativo	Híbrido	Web
Conetividade	Online/Offline	Online/Offline	Maioritariamente Online
Custo	Grande custo, caso seja desenvolvido para diferentes plataformas.	Custo parecido com o tipo Web, mas são necessárias alterações para acesso a APIs nativas.	Baixo custo, devido a existir uma única base de código.
Portabilidade	O código desenvolvido apenas funciona numa plataforma.	Permite a portabilidade para diferentes plataformas.	Apenas existem problemas com a compatibilidade dos navegadores.
Acesso a funções do dispositivo	Acesso total a todas as APIs.	Poucas APIs não podem ser usadas.	Apenas algumas APIs podem ser usadas.
Interface do Utilizador (UI)	Apresenta componentes nativos familiares.	UI <i>frameworks</i> não apresentam um visual nativo.	UI <i>frameworks</i> não apresentam um visual nativo.
Performance	Apresenta acesso direto às funcionalidades da plataforma, logo boa performance.	Apresenta uma performance inferior a aplicações nativas.	A performance é baseada no navegador que está a ser usado e na conetividade à rede de trabalho.

2.4.2 Tecnologias

Nesta subsecção, são apresentadas as *frameworks* tidas em conta para o desenvolvimento do trabalho realizado. As *frameworks* Xamarin, Ionic e React Native permitem a criação de aplicações nativas, pois possibilitam o desenvolvimento multiplataforma de aplicações.

Xamarin

O Xamarin, fundado em 2011 é uma das *frameworks* multiplataforma disponíveis. O projeto foi adquirido pela Microsoft em 2016 e tornou-se parte do ambiente de desenvolvimento integrado (IDE) do Visual Studio. Com o Xamarin é possível desenvolver aplicações móveis utilizando C#. Simplificando, o que Xamarin faz é utilizar C# para fazer chamadas nativas do sistema operacional nas plataformas móveis. C#, é uma linguagem de programação, desenvolvida pela Microsoft para a *framework* .NET. (Microsoft, 2018)

O desenvolvimento na plataforma Xamarin pode ser feito utilizando três hipóteses:

- Xamarin.Forms - O Xamarin.Forms fornece um *kit* de ferramentas de interface multiplataforma completo para desenvolvedores .NET. Permite criar aplicações móveis para Android, iOS usando C# no Visual Studio;
- Xamarin.iOS - O Xamarin.iOS permite criar aplicações em iOS utilizando C# no Visual Studio;
- Xamarin.Android – Assim como no Xamarin.iOS, permite desenvolver aplicações Android utilizando C#.

A grande diferença de desenvolver com Xamarin é a facilidade que este dá para o desenvolvimento multiplataforma. Com Xamarin é possível utilizar apenas uma linguagem e desenvolver aplicações para iOS, Android e Windows Phone. Isto significa que com Xamarin existe grande reaproveitamento da base de código quando se está a desenvolver com o foco em mais de uma plataforma. Mais precisamente, é possível reaproveitar toda a camada de negócios (acesso a dados, classes de domínio, chamadas a serviços), sendo apenas necessário especificar a interface do utilizador (UI) e as chamadas à APIs específicas de cada plataforma (Microsoft, 2018).

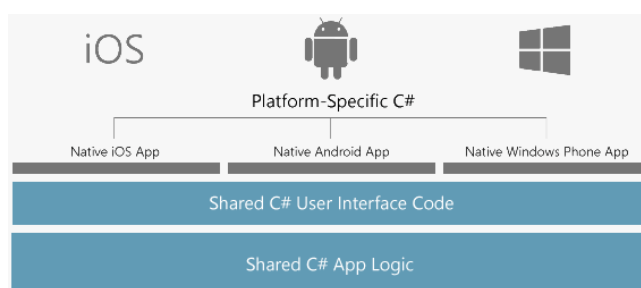


Figura 9 – Partilha de código utilizando Xamarin (Microsoft, 2018)

A Figura 9, ilustra como o Xamarin permite partilhar o código, ou seja, utilizar a mesma linguagem em APIs e estruturas de dados para mais de 96% do código da aplicação em todas as plataformas de desenvolvimento móvel.

Ionic

Ionic, lançado em 2013, é uma *framework* gratuita que permite o desenvolvimento de aplicações móveis e de *desktop*, que usa as tecnologias *web* (HTML, CSS e JavaScript). A Ionic *framework* é focada na experiência do utilizador com a aplicação (controlos, interações, gestos, animações). É fácil de aprender e integra-se muito bem com outras bibliotecas ou estruturas, como o Angular. O Ionic disponibilizou o ambiente de desenvolvimento (IDE), o Ionic Studio que proporciona uma experiência de desenvolvimento suave para equipas que implementam soluções com Ionic (Ionic, 2018).

De seguida, são listadas algumas das vantagens que esta *framework* tem para oferecer:

- Fonte livre e aberta, o Ionic *framework* é um projeto 100% gratuito e de código aberto;
- Plataforma completa, permite criar aplicações *web* progressivos e móveis, com apenas uma base de código;
- *Plugins* nativos, permite a utilização de recursos nativos dos dispositivos, como Bluetooth, HealthKit, Finger Print Auth e muito mais;
- Documentação, esta é construída com exemplos de aplicações reais, demonstrações de componentes, guias e instruções para colocá-lo em prática.

A Figura 10, ilustra como o Ionic permite implementar tecnologias *web* numa só *framework*.



Figura 10 – Estrutura da *framework* Ionic (Ionic, 2018)

React Native

O React Native é uma estrutura de Javascript construída sobre a *framework* React, ambas criadas pelo Facebook, que permite criar aplicações iOS e Android com uma única base de código. É utilizado principalmente pelo Facebook, Instagram, entre muitos outros. O React Native estreou-se como um projeto interno de *hackathon* no Facebook em 2013 e, em 2015, foi lançado ao público (ICAPPS, 2017).

De seguida são listadas algumas das vantagens do React Native, que provam que esta é uma *framework* confiável:

- O React Native acelera o desenvolvimento e poupa custos. Um dos lados mais atraentes do React Native é que apenas é preciso criar a aplicação uma vez usando JavaScript. Com apenas um código escrito, o React Native significa menores custos de construção e manutenção e apenas uma equipe de desenvolvedores (ICAPPS, 2017);
- React é *open-source*. O React Native é totalmente aberto. Isso significa que os desenvolvedores podem utilizar as *frameworks* e suas bibliotecas sem custos (ICAPPS, 2017);
- As grandes empresas estão a utilizar React Native. O React Native foi desenvolvido pelo próprio Facebook em 2015. Três anos depois, a ferramenta já conquistou o reconhecimento de algumas das principais empresas do mundo, como Instagram, Tesla, Skype, Pinterest e muito mais (ICAPPS, 2017).

O Facebook mostra um grande comprometimento com o projeto e investe tempo e recursos consideráveis no aperfeiçoamento da estrutura e na libertação de atualizações regularmente. O suporte do Facebook faz do React Native uma opção estável.

A Figura 11 mostra como a *framework* React Native permite a reutilização de código quer seja desejado que a aplicação seja criada para Android ou iOS.



Figura 11 – Arquitetura da *framework* React Native (ICAPPS, 2017)

Comparação de *Frameworks* Multiplataforma

Após terem sido descritas as *frameworks* Xamarin, React Native e Ionic que são utilizadas em desenvolvimento de aplicações móveis multiplataforma, é elaborada uma análise comparativa na Tabela 3. Esta análise pretende mostrar as diferentes características de cada uma destas (Sharma, 2018) (Sinhal, 2017). Inicialmente foi pesquisado qual destas *frameworks* apresenta um melhor desempenho. Chegando-se à conclusão de que o melhor desempenho é apresentado pela *framework* React Native, seguida por Xamarin e de seguida por Ionic. Outra característica importante é a reutilização de código. Foi possível perceber que a *framework* que melhor satisfaz esta condição é Ionic, seguida por Xamarin e por último React Native. Seguidamente é apresentado o fator de custo, em que a única *framework* que não apresenta custos de desenvolvimento é React Native. Ionic e Xamarin podem apresentar custos em

determinadas funcionalidades. Na Tabela 3 é possível verificar as vantagens e desvantagens de cada *framework* respondendo então à pergunta **Q3**.

Tabela 3 – Comparação de *frameworks* multiplataforma

Caraterísticas	React Native	Xamarin Forms	Ionic
Linguagem de Programação	JavaScript	C#	HTML5, CSS, JavaScript + Typescript
Desempenho	Boa, perto do desempenho nativa	Menor que React Native	Menor que Xamarin
Interface gráfica do utilizador (GUI)	Utiliza controlos de UI nativa	Utiliza controlos de UI nativa	HTML e CSS
Suporte da comunidade	Muito forte	Forte	Forte
Utilizado em aplicações simples ou complexas	Todo o tipo de aplicações	Aplicações simples	Aplicações simples
Aplicações populares	Facebook, Instagram	Slack, Pinterest	MarketWatch, Pacifica
Reutilização de código	90%	96%	98%
Custos	Sem custos	Sem custos, mas com custos para determinadas funcionalidades	Sem custos, mas com custos para determinadas funcionalidades

2.5 Sumário

Neste capítulo foram apresentados todos os conceitos teóricos sobre o tema em desenvolvimento, desde o estado de arte de soluções existentes até as tecnologias de aplicações móveis.

Assim sendo foi possível obter resposta à **Q1** – “Qual o mecanismo de sincronização de dados, que permite melhor desempenho e coerência de dados?”, ou seja, o mecanismo de sincronização/transferência de dados que apresenta melhor desempenho é a transferência matemática ou algorítmica de dados.

Por fim, foi obtida resposta à **Q3** – “Quais as vantagens e desvantagens das diferentes *frameworks* multiplataforma de desenvolvimento de aplicações móveis?”. Em que através de uma comparação entres as *frameworks* estudadas, foi possível observar as vantagens e desvantagens de cada uma destas, em relação às outras.

3 Análise de Valor

Esta análise de valor pretende estudar os requisitos e o custo de cada um, sempre avaliando a viabilidade das mesmas e a possibilidade de reduzir os custos para a realização. Sendo assim, inicialmente será utilizado o modelo New Concept Development. Também será apresentada uma proposta de valor, o respetivo modelo Canvas e a rede de valor.

Por fim será utilizado o método AHP onde se pretende responder à questão **Q4**, ou seja, perceber que framework multiplataforma deverá ser adotada pela Glintt.

3.1 Modelo New Concept Development

O modelo NCD (*New Concept Development*) está dividido em três partes principais como é exibido na Figura 7, que são:

- O motor é a liderança, cultura e estratégia de negócios da organização que impulsiona os cinco principais elementos que são controláveis pela corporação;
- A área do raio interno define os cinco elementos de atividade controláveis do Fuzzy *front-end* (FFE);
- Os fatores de influência consistem em fatores externos que influenciam a organização como por exemplo, canais de distribuição, lei, política do governo, clientes, concorrentes, clima político e econômico, etc.

O modelo NCD aclara cinco elementos de atividade:

- Identificação da oportunidade;
- Análise da oportunidade;
- Geração de ideias;

- Seleção de ideias;
- Desenvolvimento do conceito.

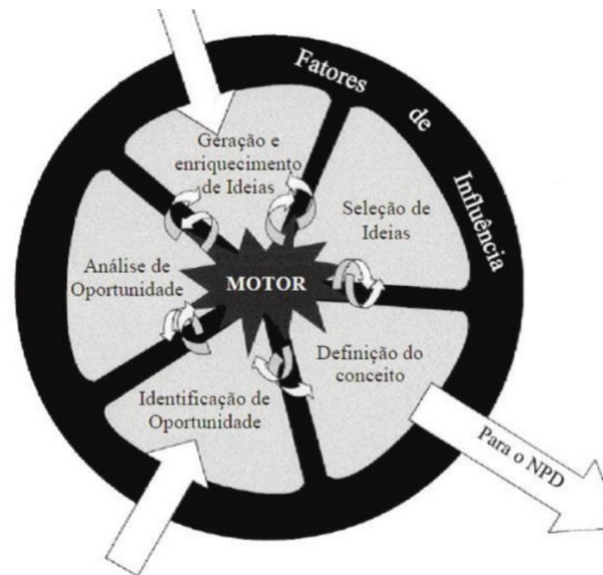


Figura 12 – Modelo NCD (New Concept Development)

A Figura 12 ilustra o motor e os cinco elementos do modelo NCD que são colocados no topo dos fatores de influência. A forma circular do modelo pretende sugerir que as ideias e os conceitos sejam iterados nos cinco elementos. As setas que apontam para o modelo representam os pontos de partida e indicam que os projetos começam na identificação de oportunidades ou geração e enriquecimento de ideias. A seta que sai mostra como os conceitos saem do modelo e entram num novo desenvolvimento de produto (NPD). (A.Koen, 2001)

3.1.1 Identificação da Oportunidade

Durante os últimos anos, os *smartphones* evoluíram em *hardware*, tamanho e eficiência, transformaram-se em pequenos dispositivos de conectividade. Apesar de os *smartphones* assumirem em grande parte o mercado da computação portátil, os PDA não estão totalmente extintos.

Algumas instituições ainda utilizam os PDA em uso laboral, como é o caso de alguns clientes da Glintt. No entanto, os PDA utilizados na logística hospitalar por vezes têm um custo bastante elevado, o que em alguns casos pode ser a razão pela qual nem todos os clientes aderirem à solução SGICM-Mobile. Sabendo que a maior parte das instituições, disponibilizam um *smartphone* para utilização no trabalho, considera-se mais acessível adquirir um destes dispositivos, até porque o custo não necessita de ser muito elevado.

O telemóvel tem vindo a ser cada vez mais usado para aceder à Internet e tornou-se agora o dispositivo preferido dos utilizadores em Portugal, de acordo com um estudo da Marktest, uma empresa de análise de mercado. Segundo os números da Marktest, o telemóvel é usado para aceder à Internet por 58% dos residentes em Portugal com 15 ou mais anos, o que fica acima

dos 55% que usam computador (este é um valor que tem estado em queda ao longo dos últimos quatro anos). Num afastado terceiro lugar surgem os *tablets*, que são usados por 19% das pessoas. A popularidade destes aparelhos tem estado a decrescer à medida que os fabricantes lançam telemóveis com ecrãs cada vez maiores (PÚBLICO, 2018).

A identificação da oportunidade para desenvolver este projeto surge do aumento da utilização de dispositivos móveis, sendo assim fará toda a lógica a Glintt adaptar as suas soluções para este tipo dispositivo.

3.1.2 Análise de Oportunidade

Para suportar a oportunidade identificada, foram recolhidos dados estatísticos sobre a utilização de *smartphones* em Portugal. O facto de os dispositivos móveis se terem tornado os preferidos dos portugueses, suporta que a ideia da conversão tecnológica da aplicação SGICM-Mobile é um importante passo no negócio da Glintt.

Tendo em conta estas informações, este rumo parece o correto. No entanto, também é necessário ter em conta o mercado de desenvolvimento de aplicações móveis. A maioria das empresas está a migrar sites de *desktop* para aplicações móveis, o que é uma passagem rápida para atingir os públicos-alvo.

Ao longo dos anos, tem havido um aumento constante no número de linguagens de programação para sistemas operacionais Android e iOS. Na verdade, o desenvolvimento de aplicações multiplataforma também ganhou impulso certamente devido às *frameworks*.

3.1.3 Geração de Ideias

Quando a necessidade de transformar as aplicações SGICM-Mobile *online* e *offline* surgiu, certamente surgiram várias ideias de implementação, no entanto a Glintt escolheu apenas uma solução.

Como a solução escolhida foi juntar estas duas aplicações numa só, sendo necessário desenvolver um mecanismo de sincronização de dados, as primeiras ideias geradas foram sobre a tipologia da aplicação móvel. Após existir a decisão de que a nova solução deve ser desenvolvida em tecnologia multiplataforma, surgiu a questão **Q5**, – Qual a *framework* que mais benefícios trará neste contexto de negócio?”. Por fim, foram escolhidas três *frameworks*, que resultaram nas ideias abaixo.

- Ideia 1 – Desenvolver a aplicação com a *framework* Xamarin;
- Ideia 2 – Desenvolver a aplicação com a *framework* React Native;
- Ideia 3 – Desenvolver a aplicação com a *framework* Ionic;

3.1.4 Seleção de Ideias

Com o intuito de selecionar uma ideia das propostas (secção 3.1.3), irá ser utilizada a análise hierárquica (AHP), com o objetivo de analisar qual das ideias trará mais benefícios para a empresa. Nesta avaliação serão tidos em consideração diferentes fatores (eficiência, eficácia, custo e usabilidade) com o objetivo de escolher a opção que melhor se adapta ao problema.

O sistema móvel a ser desenvolvido pode escolher diferentes caminhos na sua implementação nomeadamente a nível de tecnologias, no entanto, é importante escolher a ideia que mais vantagem traz para o cliente e tem um menor custo de desenvolvimento.

3.1.5 Definição do Conceito

Neste ponto deve ser apresentado um caso convincente para investimento no negócio ou proposta de tecnologia. A maioria das empresas especifica diretrizes, que tomam decisões no início do processo de desenvolvimento. Estes podem abordar os objetivos da oportunidade, ou necessidades e benefícios do cliente, os participantes da cadeia de valor, etc (FEMTT 2019).

O presente projeto deve perceber como os utilizadores irão beneficiar do uso de uma aplicação móvel para elaborar o registo de consumos. Um dos principais benefícios será a diminuição do tempo perdido durante a sincronização de dados e a possibilidade de trabalhar *offline*.

Existe então, um conjunto de passos que devem ser implementados:

- Entrar na aplicação utilizando um dispositivo móvel;
- Selecionar o módulo de registo de consumos;
- Escolher vários artigos necessários;
- O utilizador dirige-se a uma área em que não existe Wi-Fi;
- O utilizador adiciona mais artigos à sua lista;
- O utilizador grava o consumo hospitalar;
- Dirige-se para uma área com Wi-Fi e seu registo é sincronizado e persistido.

Os requisitos apresentados acima, são requisitos mínimos do projeto que foram identificados para demonstrar o conceito principal do projeto.

3.1.6 Valor, Valor Percecionado e Valor para o Cliente

“Definimos valor como a razão entre o que o cliente recebe e o que ele dá. O cliente recebe benefícios e assume custos. Os benefícios incluem benefícios funcionais e emocionais. Os custos incluem custos monetários, de tempo, de energia e psicológicos.” (Kotler, 2000)

Tendo em conta a definição acima apresentada apercebemo-nos que grande parte do valor de uma solução está no conhecimento dos benefícios por parte do cliente.

Para o valor da solução ser bem definido, tem de ser possível identificar o que realmente fazemos e quem estará interessado nesta abordagem. No contexto deste trabalho, podemos identificar que Glintt pretende construir uma aplicação móvel que assiste os utilizadores no registo de consumos, sendo os centros hospitalares os principais interessados.

3.1.7 Valor Percebido

O cliente nem sempre tem a perceção do valor do produto/serviço que está a adquirir, ou seja, não conhece todos os benefícios que está a comprar, apesar de saber exatamente o preço que está a pagar. A perceção que um cliente terá da nova solução desenvolvida, é muito importante e deve fazer parte da estratégia de comunicação da Glintt. A perceção de valor que o cliente terá da solução será o que destacará a solução em relação à concorrência, e não só o preço.

Para ajudar a demonstrar a perceção de valor, foi construída a Tabela 4, que ilustra a relação de benefício e sacrifício em relação ao serviço a ser desenvolvido.

Tabela 4 - Tabela de benefícios e sacrifícios

Domínio/ âmbito	Serviço	Relação
Benefícios	Permite trabalhar em locais sem acesso a Wi-Fi; Maximizar a produtividade; Reduzir os custos administrativos; Reduzir erros; Simplificar os processos; Eliminação do recurso à informação baseada em papel; Acesso à informação exata, em tempo real; Aumenta a segurança; Permite a racionalização dos diferentes <i>stocks</i> nos serviços; Controlo mais rigoroso de todos os custos; Reduz desperdícios.	Confiança Parceria Inclusão Inovação
Sacrifícios	Investimento inicial; Contrato de manutenção; Investimento em novos dispositivos; Custo do serviço.	Formação de utilizadores

A Figura 13 apresenta a perspetiva longitudinal de valor que segundo Woodall (2003) divide o valor para o cliente (VC) em quatro posições temporais de valor, nomeadamente: antes da compra, no ponto de compra, depois da compra e depois da experiência do utilizador.

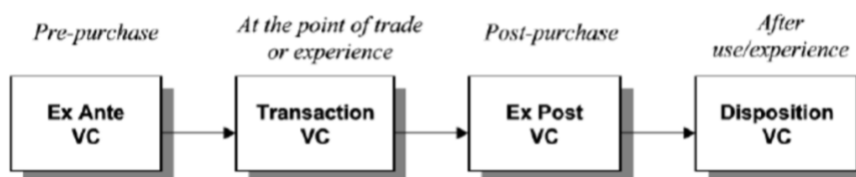


Figura 13 – Perspetiva longitudinal VC (Woodall, 2003)

- *Ex Ante VC* - Nesta fase os centros hospitalares têm em conta o valor pretendido e valor expectável. O cliente consegue perceber a qualidade técnica e funcional e as características da nova solução móvel da Glintt. No entanto, os clientes também avaliam os custos de aquisição desta solução.
- *Transaction VC* – Neste momento os clientes têm em conta o valor de transação, aquisição e de troca. Este avalia o preço da solução, os custos de instalação, os custos de manutenção.
- *Ex Post VC* – Neste momento o cliente avalia o valor recebido, a performance e o valor das entregas. Este consegue identificar os benefícios funcionais, a segurança e conveniência que obtém da nova solução.
- *Disposition VC* - Por fim o cliente consegue avaliar a apreciação dos utilizadores, os benefícios operacionais e os custos de utilização. Depois da experiência com nova solução da Glintt, o utilizador percebe a utilidade do registo de consumos através de *smartphones*.

3.1.8 Valor para o Cliente

A solução idealizada apresenta um elevado valor para o mercado da logística hospitalar, principalmente ao nível da usabilidade. Já existe interesse dos clientes em ver esta solução implementada, pelo que estes anseiam a atualização do sistema atual com o objetivo de causar inovação nos seus processos do dia-a-dia. Na Tabela 4 podemos verificar os benefícios que a aplicação móvel trará, que se traduzem em valor para os clientes da Glintt.

3.2 Proposta de Valor

A aplicação móvel SGICM-Mobile agora disponível para Android, permite o registo de consumo hospitalares em qualquer parte do hospital, sem que o utilizador tenha de se preocupar se existe conectividade à rede. Os centros hospitalares poderão adotar esta solução e permitir que os seus colaboradores realizem esta tarefa diária utilizando um *smartphone*, e ver todo o seu trabalho sincronizado assim que atinjam uma zona com rede. Esta solução estará integrada com a solução *web* SGICM, que permite a consulta e controlo de todos os movimentos elaborados via dispositivos móveis, complementando assim mais um ciclo na logística hospitalar.

3.3 Modelo de Negócio Canvas

A utilização do modelo Canvas permite estruturar de maneira mais formal o negócio e criar mecanismos que permitam maior controlo. O modelo ajuda a ter uma visualização ampla do negócio, de maneira simples. (Viswanathan, 2012)

Esta ferramenta consiste num quadro que apresenta em blocos os nove fatores-chave de um empreendimento que são:

- Parcerias-chave (*Key Partners*), são os relacionamentos que a Glintt tem com outras entidades comerciais, governamentais ou não-consumidoras que ajudam o seu modelo de negócios a funcionar.
- Atividades-chave (*Key Activities*), são as atividades mais importantes na execução da proposta de valor da empresa.
- Recursos-chave (*Key Resources*), descreve os ativos mais importantes necessários para que o modelo de negócios funcione.
- Proposta de valor (*Value Propositions*), é uma inovação, serviço ou recurso destinado a tornar o produto atraente para os clientes.
- Relacionamento (*Customers Relationships*), descreve o tipo de relacionamento que a empresa estabelece com seus segmentos de clientes específicos.
- Canais (*Channels*), descreve como a empresa comunica e atinge os seus segmentos de clientes para entregar a sua proposta de valor.
- Segmentos de clientes (*Customers Segments*), descreve o tipo de relacionamento que uma empresa estabelece com os segmentos de clientes específicos.
- Estrutura de custos (*Cost Structure*), define todos os custos e despesas que a empresa incorrerá ao operar o modelo de negócios.
- Fontes de renda (*Revenue Streams*), descreve como a Glintt irá ganhar dinheiro com a nova solução.

De seguida, na Tabela 5 é apresentado o modelo Canvas, que pretende estruturar de maneira mais formal o negócio.

Tabela 5 – Modelo de negócio Canvas

Key Partners Equipa de Logística e Farmácia; Projetos internos da equipa de Logística e Farmácia;	Key Activities Estudar as <i>frameworks</i> disponíveis para desenvolvimento de aplicações móveis; Aplicar um mecanismo de sincronização de dados;	Value Propositions Para facilitar o trabalho na logística hospitalar, é proposto um sistema móvel que permitirá o registo de consumos; O sistema poderá ser utilizado em qualquer lugar do hospital, mesmo em locais sem acesso à Internet, sendo todo o trabalho do utilizador sincronizado quando o sistema tiver ligado à rede.	Customer Relationships Equipa de Suporte; Equipa de comercialização;	Customer Segments Clientes da Glintt, nomeadamente que pretendam utilizar dispositivos móveis para o registo de consumos hospitalares;
	Key Resources Aplicação móvel; Dispositivos móveis;		Channels Demonstração em clientes; Correio eletrónico; Disponibilização de informação sobre o produto no <i>website</i> da Glintt;	
Cost Structure Desenvolvedores/programadores; Dispositivos móveis; Publicidade/Marketing;		Revenue Streams Contratos de manutenção; Custo de Upgrade para nova versão do SGICM-Mobile;		

3.4 Rede de Valor

“People naturally network as they work so why not model itself as networks”, esta citação de Verna Allee pretende referir que uma forte rede de conhecimentos entre várias pessoas está diretamente relacionada com a criação de valor. A autora do modelo Verna Allee assume que uma rede de valor é composta por valor tangível, valor intangível e atores.

Neste sentido, é utilizado o modelo de Verna Allee para definir a rede de valor e ter uma melhor perceção do valor de negócio da Glintt, no contexto deste trabalho.

Tendo em conta a Figura 14, que faz uma análise à rede de valor da Glintt, em alto nível, consegue-se identificar os principais atores (Glintt, fornecedores de tecnologia, universidades, parceiros de negócio e os centros hospitalares) e percebe-se que todos têm um papel importante na construção de uma solução tecnológica, neste caso do SGICM-Mobile. Também é possível identificar os valores intangíveis desta rede (credibilidade, lealdade, expansão do produto, reconhecimento) e os valores tangíveis (contratos de manutenção, pagamentos de soluções, pagamento de licenças).

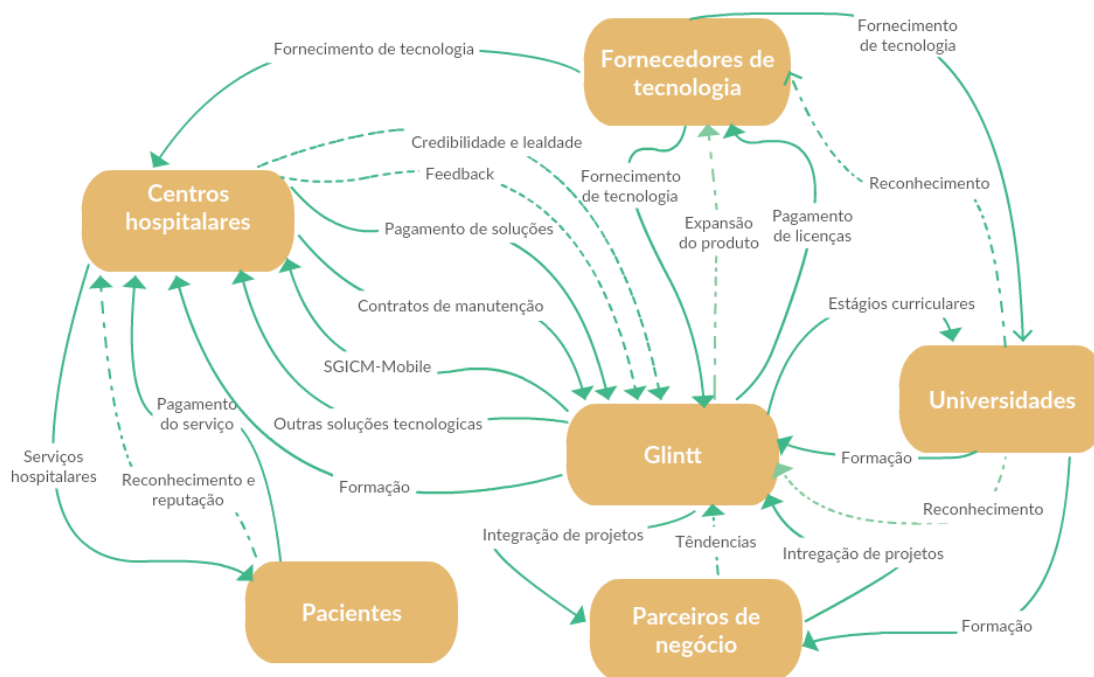


Figura 14 – Diagrama de análise da rede de valor.

3.5 AHP (Analytic Hierarchy Process)

Das ideias refletidas e com base nos estudos do mercado, foi decidido realizar esse mesmo estudo incidindo sobre a *framework* Xamarin, React Native e Ionic.

Dependendo do resultado do mesmo será analisada a possibilidade de executar a ideia principal. Assim sendo, nesta secção pretende-se responder à questão **Q4**, ou seja, perceber que *framework* deverá ser adotada pela Glintt.

O método AHP (Analytic Hierarchy Process) permite tomar uma decisão de maneira organizada para gerar prioridades. É possível decompor as decisões nos seguintes passos:

- Definir o problema e determinar o tipo de conhecimento procurado;
- Estruturar a hierarquia de decisão a partir do topo com o objetivo da decisão e, em seguida, os objetivos de uma perspectiva ampla;
- Construir um conjunto de matrizes de comparação entre pares. Cada elemento em uma parte superior nível é usado para comparar os elementos no nível imediatamente abaixo com respeito a ele.
- Usar as prioridades obtidas nas comparações para ponderar as prioridades no nível imediatamente abaixo.

Para se fazer bom uso da escala de prioridades, é preciso compreender o que são os julgamentos no método criado por Saaty. Um julgamento ou comparação é a representação numérica de uma relação entre dois elementos que possuem o mesmo pai. O grupo de todos esses julgamentos pode ser representado numa matriz quadrada, na qual os elementos são comparados com eles mesmos. Cada julgamento representa a dominância de um elemento da coluna à esquerda sobre um elemento na linha do topo (Saaty, 2008).

A escala recomendada por Saaty, é mostrada na Tabela 6 irá ser utilizada durante a análise hierárquica.

Tabela 6 – Comparações AHP (Saaty, 2008)

Intensidade de Importância	Definição	Explicação
1	Mesma importância	As duas atividades contribuem igualmente para o objetivo.
3	Importância pequena de uma sobre a outra	A experiência e o julgamento favorecem levemente uma atividade em relação à outra
5	Importância grande ou essencial	A experiência e o julgamento favorecem fortemente uma atividade em relação à outra.
7	Importância muito grande ou demonstrada	Uma atividade é muito fortemente favorecida em relação à outra.
9	Importância absoluta	A evidência favorece uma atividade em relação à outra com o mais alto grau de certeza.
2,4,6,8	Valores intermediários entre os valores adjacentes	Quando se procura uma condição de compromisso entre duas definições.

De seguida, na Figura 15 é ilustrada a árvore de decisão criada para a análise hierárquica, em que são identificados os objetivos, critérios e por fim as diferentes alternativas.

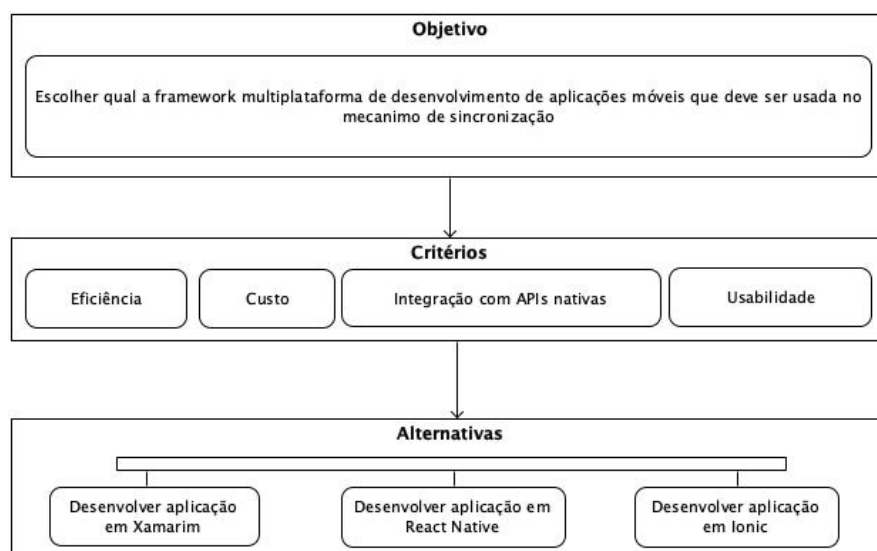


Figura 15 – Árvore hierárquica de decisão

Com recurso a estes critérios é possível avaliar qual é a melhor ideia/solução para atingir o principal objetivo. A Tabela 7 inclui a atribuição de pesos para cada critério, segundo a escala de AHP.

Tabela 7 – Tabela de avaliação AHP

Critérios de avaliação	Eficiência	Custo	Integração com APIs nativas	Usabilidade
Eficiência	1	3	4	7
Custo	1/3	1	1/3	2
Integração com APIs nativas	1/4	3	1	7
Usabilidade	1/7	1/2	1/7	1
Total	1 61/84	7 1/2	5 10/21	17

De seguida, define-se a classificação dos critérios segundo um vetor de prioridade, também conhecido por Eigen Vector X, cujos resultados são apresentados na Tabela 8. Após a análise dos resultados, o critério de “eficiência” (0,53), é considerado o mais importante e a “usabilidade” (0,06), o menos importante.

Tabela 8 – Matriz normalizada do método de avaliação HP

Critérios de avaliação	Eficiência	Custo	Integração com APIs nativas	Usabilidade	Vetor próprio
Eficiência	0,5793	0,4000	0,7304	0,4118	0,5304
Custo	0,1931	0,1333	0,0609	0,1176	0,1262
Integração com APIs nativas	0,1448	0,4000	0,1826	0,4118	0,2848
Usabilidade	0,0828	0,0667	0,0261	0,0588	0,0586
Soma	1,0000	1,0000	1,0000	1,0000	1,0000

Como indica a Tabela 9, o próximo passo é para cada critério de avaliação comparar as *frameworks* Xamarin, Ionic e React Native. A atribuição de níveis de importância, foi baseada

nos dados apresentados na Tabela 3, que apresenta a comparação de *frameworks* multiplataforma.

Tabela 9 – Comparação paritária dos critérios

Eficiência			
Critérios de avaliação	Xamarin	Ionic	React Native
Xamarin	1	3	1/3
Ionic	1/3	1	1/5
React Native	3	5	1
Total	4 1/3	9	1 8/15
Custo			
Xamarin	1	1/2	1/5
Ionic	2	1	1/5
React Native	5	5	1
Total	8	6 1/2	1 2/5
Integração com APIs nativas			
Xamarin	1	7	5
Ionic	1/7	1	1/3
React Native	1/5	3	1
Total	1 12/35	11	6 1/3
Usabilidade			
Xamarin	1	1/3	1/9
Ionic	3	1	1/5
React Native	9	5	1
Total	13	6 1/3	1 14/45

De seguida, define-se a classificação dos critérios segundo um vetor de prioridade, também conhecido por Eigen Vector X, cujos resultados são apresentados na Tabela 10.

Tabela 10 – Matriz Normalizada para os critérios e prioridade local

Eficiência				
Critérios de avaliação	Xamarin	Ionic	React Native	Prioridade
Xamarin	0,2308	0,3333	0,2174	0,2605
Ionic	0,0769	0,1111	0,1304	0,1061
React Native	0,6923	0,5556	0,6522	0,6334
Custo				
Xamarin	0,1250	0,0769	0,1429	0,1149
Ionic	0,2500	0,1539	0,1429	0,1823
React Native	0,6250	0,7692	0,7142	0,7028
Integração com APIs nativas				
Xamarin	0,7447	0,6364	0,7895	0,7235
Ionic	0,1064	0,0909	0,0526	0,0833
React Native	0,1489	0,2727	0,1579	0,1932
Usabilidade				
Xamarin	0,0769	0,0526	0,0848	0,0714
Ionic	0,2308	0,1579	0,1525	0,1804
React Native	0,6923	0,7895	0,7627	0,7482

Por fim, a Tabela 11 apresenta a classificação das *frameworks*. Este cálculo é realizado multiplicando cada elemento da matriz de prioridades com o vetor próprio calculado e somando a linha de maneira a obter o resultado. Como se pode verificar a *framework* React Native possui um resultado de 52%, Xamarin possui o resultado de 36% e Ionic o resultado de 12%.

Tabela 11 – Classificação de *frameworks*

Crítérios de avaliação	Eficiência	Custo	Integração com APIs nativas	Usabilidade	Resultado
Xamarin	0,1382	0,0145	0,2061	0,0042	0,3630
Ionic	0,0563	0,0230	0,0237	0,01057	0,1138
React Native	0,3359	0,0887	0,0550	0,0438	0,5232

3.6 Sumário

Neste capítulo foi elaborada uma análise de valor, onde foi possível perceber o valor do negócio utilizando os modelos NCD, AHP. Também foram elaboradas uma proposta de valor e a rede valor, que permitiu aprofundar o conhecimento sobre o empreendimento da Glintt.

Este capítulo respondeu à **Q4** – “Qual a *framework* multiplataforma que deve ser adotada pela Glintt?”, através do método AHP, em que é possível perceber que a *framework* React Native é realmente a *framework* que mais se adequa ao problema proposto.

4 Solução

Este capítulo tem como objetivo apresentar a análise e *design* da solução a desenvolver, que irá abordar a escolha do mecanismo de sincronização a implementar.

Para a descrição da arquitetura irá ser utilizado o modelo 4+1, com o objetivo de descrever o funcionamento de sistemas de software. As visões do modelo são: visão de casos de uso, visão lógica, visão de processos, visão física e visão de desenvolvimento (Kruchten, 1995).

Sendo assim, na visão de casos de uso foram identificados os requisitos funcionais e não funcionais da aplicação móvel a ser desenvolvida. Estes requisitos recolhidos encontram-se especificados no Anexo I. Após a recolha de requisitos, são identificados os casos de uso a serem implementados nesta solução.

De seguida, e mediante os requisitos e casos de uso apresentados são expostas duas alternativas de implementação. Estas alternativas focam-se nos padrões de transferência de dados. Após a análise das diferentes alternativas é explicada a abordagem de implementação de forma a responder à questão **Q2** e sublíneas **Q2.1**, **Q2.2** e **Q2.3**.

Posteriormente são apresentadas as restantes visões do modelo 4+1, em que estas são ilustradas utilizando a construção de diagramas UML.

Por fim, é apresentado um modelo de dados, que pretende ilustrar a construção das principais tabelas necessárias para o mecanismo de sincronização.

4.1 Casos de Uso

Os casos de uso (UC) descrevem as funcionalidades propostas para o novo sistema. É uma ferramenta utilizada no levantamento dos requisitos funcionais do sistema. De seguida, na Figura 16 é possível observar o diagrama de UC, construído após o levantamento de requisitos (ver Anexo I) junto da empresa.

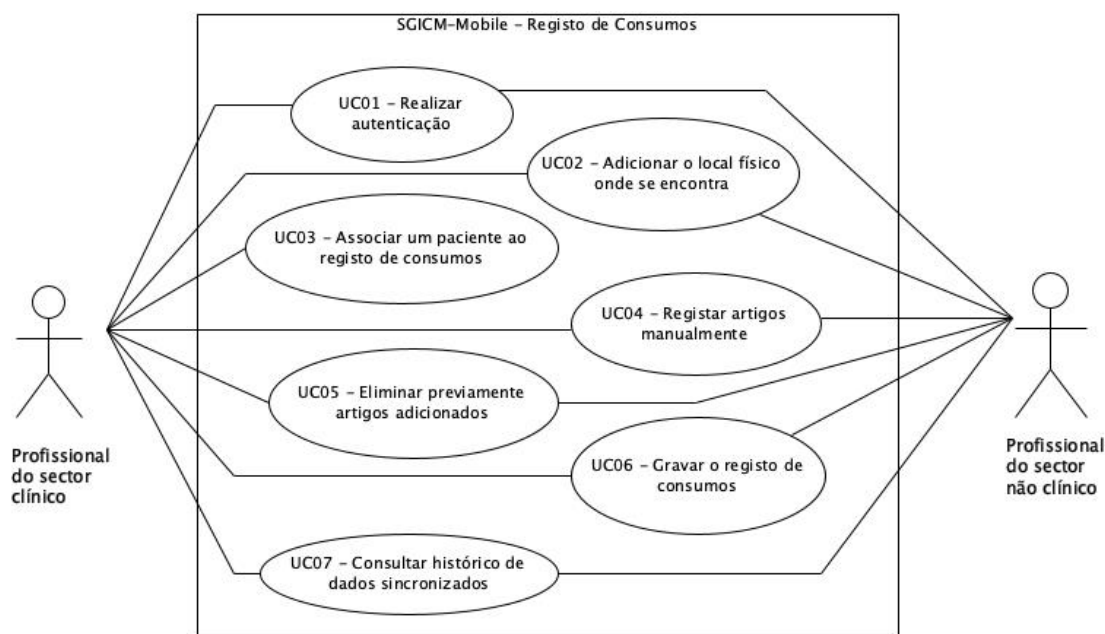


Figura 16 – Diagrama de Casos de Uso

4.1.1 Descrição de Casos de Uso

Nesta subsecção pretende-se listar os UCs identificados na Figura 16, que se tornarão em funcionalidades da aplicação móvel a ser desenvolvida.

Como tal, para a resolução deste problema identificaram-se os seguintes UCs:

- UC01 – Realizar autenticação;
- UC02 – Adicionar o local físico onde se encontra;
- UC03 – Associar um paciente ao registo de consumos;
- UC04 – Registrar artigos manualmente;
- UC05 – Eliminar previamente artigos adicionados;
- UC06 – Gravar o registo de consumos;
- UC07 – Consultar histórico de dados sincronizados.

Nesta subsecção apenas será detalhado o UC06, pois considera-se uma funcionalidade que irá ter mais relevância na sincronização de dados.

A descrição dos UCs irá ser elaborada em forma de tabela, em que a Tabela 12, representa a descrição da gravação do registo de consumos. A descrição dos restantes UC, encontram-se no Anexos II, também representados em forma de tabela.

Tabela 12 – UC06 – Gravar o registo de consumos

Descrição	
Objetivo	O utilizador grava o registo de consumos com os artigos que adicionou à sua lista.
Atores	Profissional do sector clínico (médico, enfermeiro, farmacêutico, técnico de diagnóstico, terapêutica), profissional do setor não clínico (responsável de logística, técnico de logística)
Pré-condições	Existe pelo menos um artigo na lista de registos.
Pós-condições	O utilizador é reencaminhado para o menu principal.
Prioridade	Alta
Frequência de uso	Alta
Campos	Não aplicável.
Requisitos	RF006 – Gravar o registo de consumos. RF008 – Aplicação deve funcionar com e sem rede Wi-Fi; NF01 – Portabilidade NF02 – Usabilidade
Fluxo principal	<ol style="list-style-type: none"> 1. O utilizador clica para gravar o consumo. 2. O sistema faz as validações necessárias. 3. O sistema grava o consumo e gera um número de consumo temporário. 4. O sistema mostra ao utilizador o número de documento gerado. 5. O sistema verifica se o utilizador está em modo <i>online</i>. 6. Caso esteja <i>online</i>, inicia uma sincronização. 7. O sistema reencaminha o utilizador para o menu principal.

4.2 Alternativas e Abordagem de Implementação

Conforme referido no início deste capítulo é necessário definir o mecanismo de sincronização que melhor se adapta à solução a ser desenvolvida. Para resolver o problema existem duas abordagens de transferência de dados que poderiam ser aplicadas:

- Utilizar o padrão de transferência de dados por *timestamp*;
- Utilizar o padrão de transferência de dados algorítmico.

A transferência de dados por *timestamp* permite sincronizar todos os dados necessários para a elaboração de um registo de consumos. Para a implementação deste mecanismo, seria necessário elaborar um método de sincronização, que iria receber o tipo de dados a sincronizar. Ou seja, o tipo de dados seria por exemplo, armazéns, pacientes, etc. Posteriormente, seriam obtidos todos os dados a partir de uma data. Esta data corresponde à data da última sincronização. De forma, a que este método seja aplicado é necessário que todas as tabelas, implementem uma coluna em que guardam a data da última alteração dos registos.

Esta abordagem permitiria que posteriormente, caso fossem implementados novos módulos de logística hospitalar para além do registo de consumo, a lógica de obtenção de dados fosse reaproveitada para novos tipos de dados.

No entanto esta abordagem, sincroniza todos os dados a partir de uma data. Ou seja, estariam a ser sincronizados dados que um utilizador não necessita, ou até que não teria acesso.

Ao implementar um mecanismo de transferência de dados algorítmico, para além de utilizar a transferências de dados por *timestamp*, podem ser implementados pequenos algoritmos que calculem exatamente quais os dados que um utilizador necessita e têm acesso.

Para que os dados a sincronizar sejam relevantes foram reaproveitadas algumas configurações já existentes na versão *web* da aplicação SGICM. Na versão *web* da aplicação é possível fazer as seguintes configurações:

- Configurar os armazéns ao qual um utilizador tem acesso;
- Configurar a que serviços um utilizador tem acesso;
- Os produtos têm armazéns associados;
- Os lotes têm produtos associados;
- Os pacientes têm um serviço associado.

No entanto para além das validações apresentadas acima, são acrescentadas ainda algumas regras de negócio, como por exemplo:

- Apenas são sincronizados lotes com prazos de validades superiores ao dia de hoje;
- Apenas são sincronizados pacientes que estejam no hospital em consultas, internamentos, etc.

Ou seja, apenas devem ser obtidos os dados que têm relevância para cada utilizador. Este tipo de mecanismo têm a vantagem de ter em conta os fatores como a memória do dispositivo, mas também o desempenho do mecanismo seria melhorado.

No entanto, a implementação deste mecanismo levaria mais tempo pois cada conjunto de dados teria uma lógica para ser obtido. Caso futuramente fossem implementados novos módulos logísticos, teria de ser elaborada uma nova lógica de sincronização para cada um destes.

Após conhecer as alternativas de implementação é possível começar a definir a abordagem de implementação da solução. Deste modo, para a implementação do mecanismo de sincronização, foram elaboradas as seguintes escolhas:

- Mecanismo de sincronização assíncrona e síncrona (subsecção 2.4.1) – Na maior parte das vezes, é utilizado um mecanismo de sincronização assíncrono, para que o utilizador não tenha de esperar até que todos os dados sejam sincronizados. No contexto deste problema prevê-se que a melhor abordagem é iniciar a sincronização sempre que o dispositivo tenha uma conexão Wi-Fi de forma assíncrona. Desta forma, o utilizador pode continuar o seu trabalho. Exceto na primeira sincronização, em que a sincronização é elaborada de forma síncrona, pois o utilizador não pode iniciar o seu trabalho sem que os dados sejam carregados para o seu dispositivo móvel. É assim obtida a resposta à **Q2.1**;

- Armazenamento total de dados (subsecção 2.4.2) – é escolhido este tipo de armazenamento devido à aplicação trabalhar em modo *offline*. No entanto o armazenamento total de dados, terá de ser minimizado dependendo das funcionalidades que o cliente utiliza e de modo a que apenas os dados essenciais sejam sincronizados para a BD local do dispositivo móvel (secção 4.3.1). Deste modo é obtida a resposta à **Q2.2**;
- Mecanismo de sincronização algorítmica (subsecção 2.4.3) – Quanto aos padrões de transferência de dados, é possível definir que irá ser necessário utilizar o método de sincronização algorítmica. Este método é escolhido, pois uma simples sincronização de dados por *timestamp* não chegaria para resolver o problema. Ao elaborar um algoritmo de sincronização, é possível que apenas sejam sincronizados os dados que realmente interessam a cada utilizador, respondendo à **Q2.3**.

Também é necessário escolher a tecnologia que melhor se adapta a este problema. Para isso foi definido pela empresa, que deveria ser utilizada uma *framework* que permita à solução ser disponibilizada em multiplataforma. Previamente, foram estudadas as *frameworks* Ionic, Xamarin e React Native. Após o estudo apresentado na secção 3.5 foi decidido implementar a solução utilizando a *framework* de desenvolvimento móvel React Native.

Para além destas restrições, este novo desenvolvimento deverá utilizar o *schema* de base de dados já existente e já utilizada pelas outras soluções da Glintt, nomeadamente em Oracle.

4.3 Visão Lógica

A Figura 17, apresenta os principais componentes existentes no desenvolvimento deste projeto. Ou seja, esta é constituída pelo componente principal SGICM-Mobile que contém:

- Registo de Consumo – Neste componente encontra-se toda a lógica necessária para elaborar um registo de consumos. Este componente utiliza o mecanismo de sincronização;
- Mecanismo de sincronização – Este mecanismo de sincronização comunica com a BD Local, com o principal objetivo de inserir, atualizar e selecionar dados. Este também utiliza a API REST, para guardar e atualizar os dados no BD principal. Por fim, também é responsável por fornecer os dados da BD local a outros componentes;
- Autenticação – A autenticação ou *login* permite apenas a entrada de utilizadores com autorização na aplicação. Este é importante pois guarda o utilizador, que será usado durante a sincronização de dados;
- Histórico de dados sincronizados – Este histórico mostra a data da última sincronização e o estado da conexão Wi-Fi. Também apresenta os movimentos temporários gerados no registo de consumo e o documento definitivo que foi gerado após a sincronização;
- BD local – Esta BD guarda todos os dados que o utilizador necessita para efetuar o seu trabalho, principalmente quando o dispositivo não tem conexão Wi-Fi.

Por fim, existem ainda dois componentes principais:

- SGICM-Mobile REST API – Desenhada especialmente para permitir a comunicação entre a aplicação móvel e a BD principal.
- BD principal – Esta BD contém todos os registos necessários para a elaboração de um registo de consumos. É aqui que devem ser guardados todos os dados sincronizados a partir da aplicação móvel.

Se a aplicação estiver em modo *online*, as solicitações HTTP (GET, POST, PUT, DELETE) de sincronização serão enviadas atualizando a BD principal.

Se a aplicação estiver em modo *offline*, as alterações realizadas nos dados da aplicação serão primeiro refletidas na BD local. Quando existe uma transição de estado da aplicação de *offline* para *online*, deve ser solicitada uma sincronização de dados despoletada através do estado.

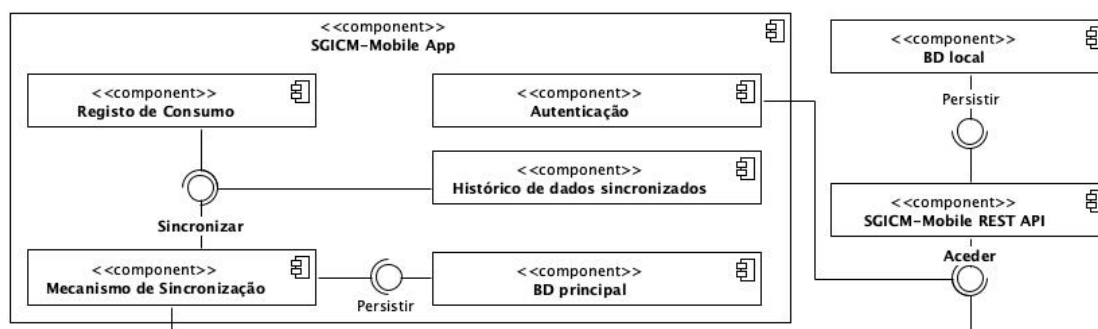


Figura 17 – Diagrama de Componentes SGICM-Mobile

4.4 Visão de Processos

A Figura 18, representa o fluxo de sincronização de dados após um utilizador elaborar *login* com sucesso. É importante referir que este diagrama não representa a primeira vez que o utilizador faz *login* na aplicação móvel. Pois, conforme explicado na subsecção 4.1.3, o processo de sincronização durante o primeiro *login* é síncrono.

O diagrama identificado na Figura 18 pretende demonstrar que os processos de sincronização são assíncronos, podendo o utilizador continuar a navegar na aplicação.

Desta forma, após o *login*, caso o dispositivo móvel tenha uma conexão à rede, é iniciado o processo de sincronização. Este processo irá obter os dados, que o utilizador tem acesso e são relevantes para o registo de consumos desde a última data de sincronização. Durante a sincronização também são sincronizados os dados que foram criados na BD local. Por fim, também demonstra o que acontece quando o processo de sincronização falha. Ou seja, quando existe uma falha, a informação de inexatidão é guardada localmente.

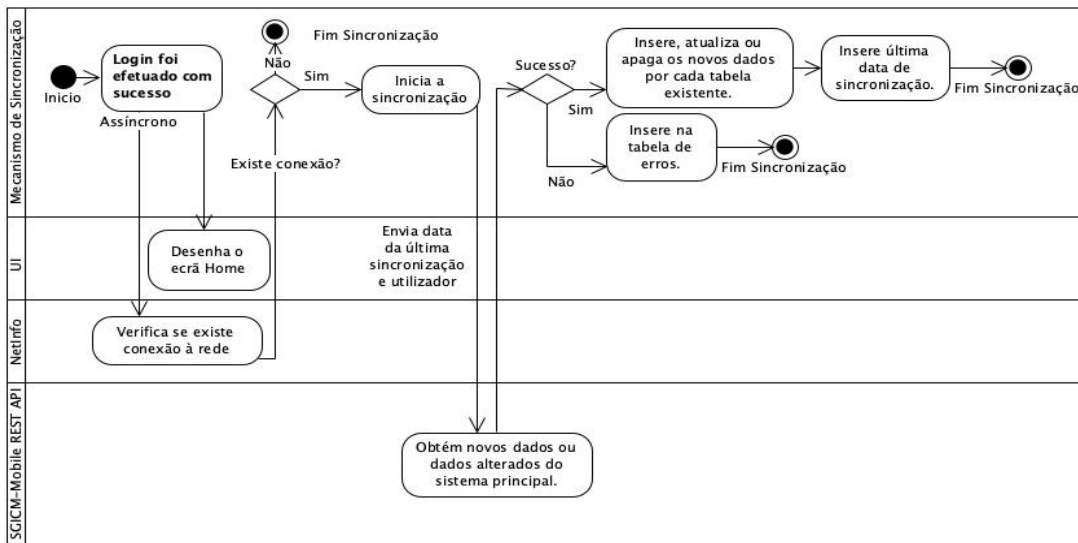


Figura 18 – Diagrama de atividade sobre a sincronização de dados

4.5 Visão Física

Nesta secção é descrita a implementação física das entidades geradas no desenvolvimento deste projeto. A Figura 19 apresenta uma possível implementação da solução a ser desenvolvida utilizando a *framework* React Native. Desta forma existe:

- *Smartphone* – O dispositivo móvel aloja a aplicação SGICM-Mobile, que permite o registo de consumo. No entanto também aloja a BD local, que permite trabalhar em modo *offline*;
- *Servidor Web* – Este servidor aloja a REST API desenvolvida para efetuar a sincronização de dados entre a BD principal e a BD local. Também aloja a aplicação *web* SGICM, ou seja, a aplicação de farmácia logística que também permite elaborar consumos;
- *Servidor da Base de Dados* – Contém a BD principal que fornece e guarda dados para a aplicação móvel e SGICM *Web*.

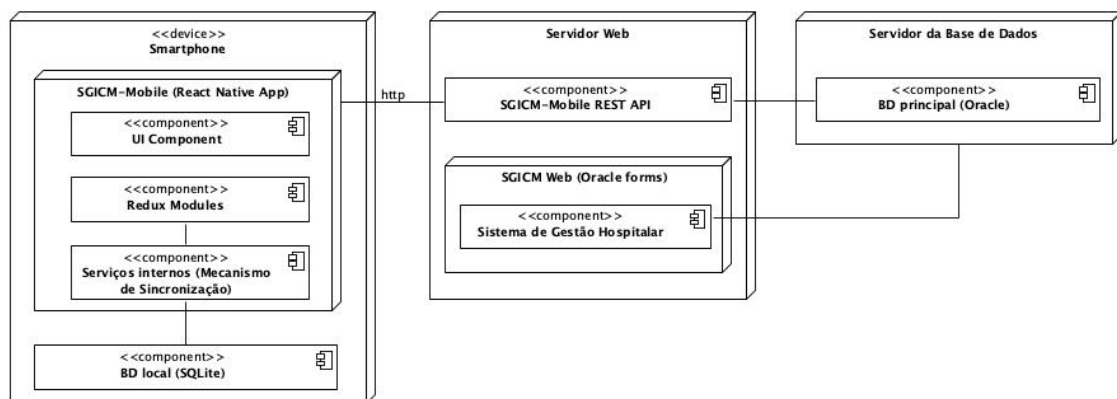


Figura 19 – Diagrama de Implantação utilizando React Native

4.6 Visão de Desenvolvimento

Inicialmente é apresentado um diagrama de sequência que descreve o processo de criação de um consumo. De seguida e com mais detalhe são apresentados os diagramas que demonstram a gravação de um registo de consumos e posteriormente a elaboração de uma sincronização.

4.6.1 Registo de Consumos

O diagrama de sequência ilustrado na Figura 20, têm como objetivo demonstrar todos os passos necessários até à gravação do consumo.

Na Figura 20, são identificados os principais componentes da aplicação, que são: o registo de consumos, mecanismo de sincronização, BD local e Redux Modules. O Redux Modules, no contexto deste problema têm o objetivo de armazenar dados à medida que o utilizador avança no registo de consumos, garantindo assim um estado único da aplicação.

De seguida são explicadas as ações identificadas na Figura 20:

- Ações 1 e 2 – Inicialmente o utilizador terá um menu, onde irá escolher a opção de registo de consumos. Quando isto acontece (ação 1), é iniciado o registo sendo o utilizador encaminhado para um ecrã em que começa o seu trabalho (ação 2);
- Ações 3 a 11 – Após ser iniciado o registo de consumos é apresentado um conjunto de listas que permitem ao utilizador escolher o local físico onde se encontra (UC02) e o paciente ao qual vai ser imputado o consumo (UC03). Este bloco *opt* demonstra como são obtidos os dados apresentados nas listas de serviço, armazém e paciente. Ou seja, assim que utilizador informa o sistema que pretende escolher um armazém (ação 3), é apresentada uma lista pesquisável (ações 4-7). Esta lista obtém os seus dados através do mecanismo de sincronização (ação 4) que apenas têm o objetivo de aceder à BD local (ação 5) e fornecer os dados necessários;
- Ações 12 e 13 – Após o utilizador selecionar os dados previamente falados, este é reencaminhado para um ecrã de seleção de artigos (ação 12). A partir deste ecrã são adicionados todos os artigos a serem consumidos (*ref* adicionar artigos), ilustrando assim o UC04;
- Ações 14 a 16 – Por fim, quando todo o registo está completo, o utilizador grava o registo de consumos (ação 14 e ação 15). De seguida, é mostrada uma mensagem com o número de documento temporário gerado e o utilizador é reencaminhado para um ecrã inicial, ilustrando assim o UC06.

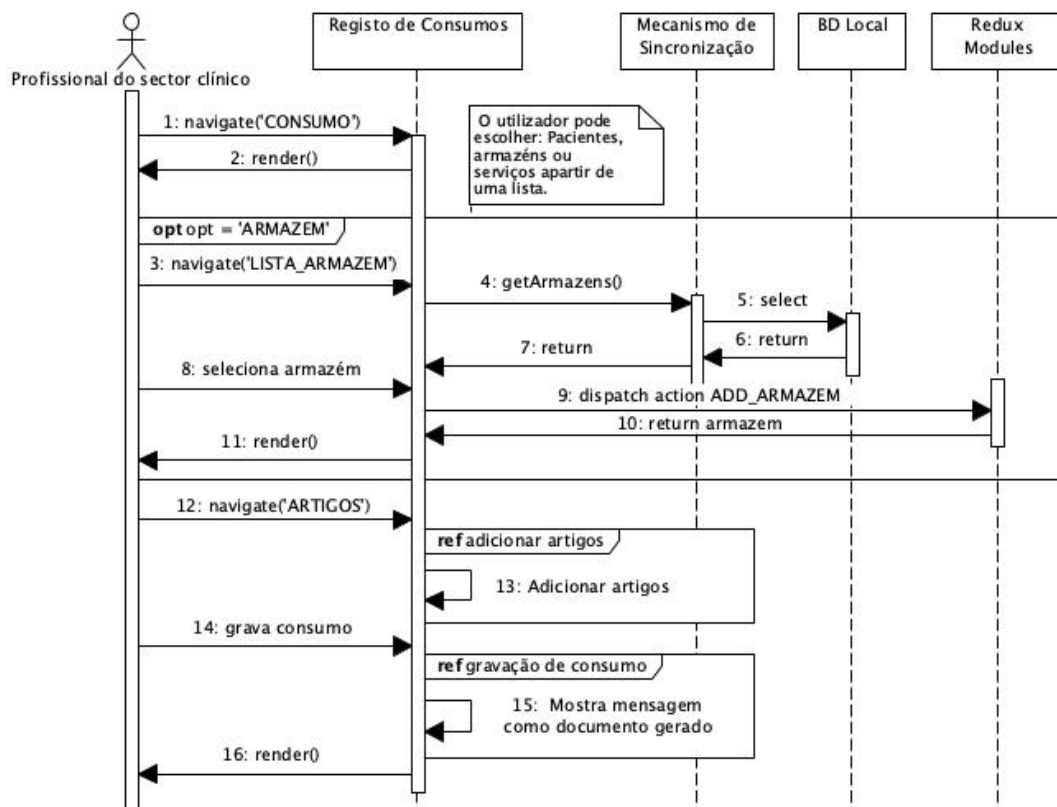


Figura 20 – Diagrama de seqüência de Registo de Consumo

4.6.2 Gravação de um Consumo

A Figura 21 têm como objetivo detalhar como é processada a gravação de um consumo. Nesta figura são identificados os principais componentes da aplicação, já utilizados na Figura 20. Este diagrama de seqüência, faz referência à Figura 20 (*ref* gravação de consumo).

De seguida são apresentadas as ações identificadas:

- Ações 1 a 5 – É possível verificar que após o utilizador expressar a sua vontade de gravar o registo (ação 1), é desencadeada a gravação deste registo na BD local (ações 2-5). A gravação gera um número de documento temporário, até que exista uma sincronização;
- Ações 6 e 7 – Para obter o estado da conexão Wi-Fi é utilizada a biblioteca NetInfo. Esta biblioteca permite saber se o dispositivo tem uma conexão Wi-Fi ativa;
- Ações 8 e 9 – De seguida e de forma assíncrona é iniciada a sincronização de dados (*ref iniciaSincronização*), sem que o utilizador se aperceba. Isto acontece, pois este é reencaminhado para o menu inicial e é informado do número temporário que foi gerado para o seu consumo (ação 9).

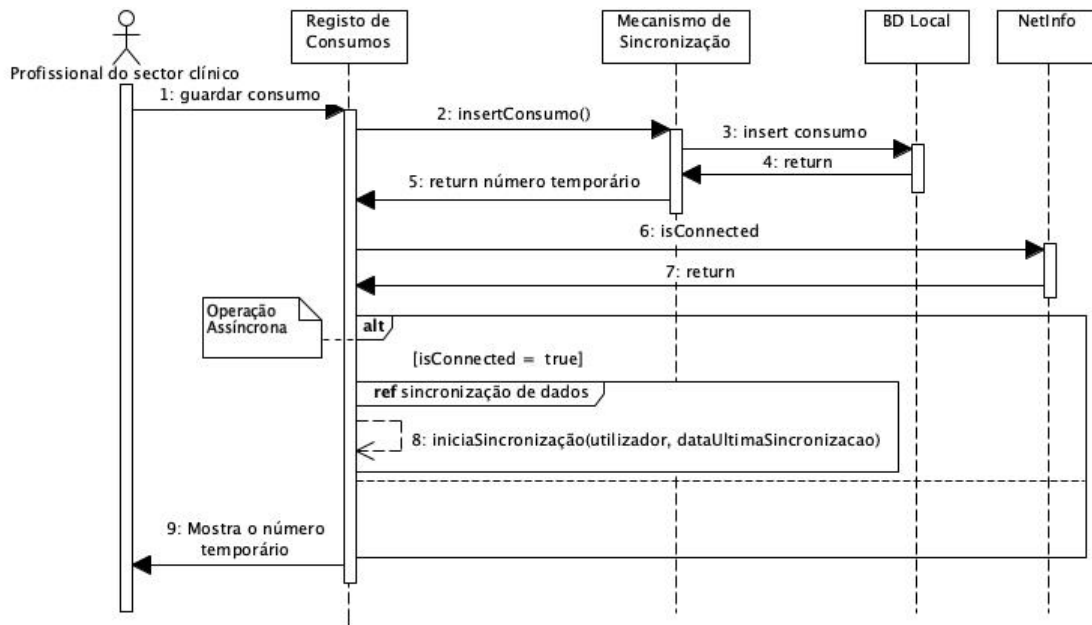


Figura 21 – Diagrama de sequência gravação de consumo

4.6.3 Sincronização de Dados

Por fim, a Figura 22 mostra como é elaborada uma sincronização de dados, com o objetivo de explicar como é feita esta interação e perceber o mecanismo desenvolvido.

De seguida são explicadas as ações, sendo estas:

- Ações 1 a 3 – Após existir a informação sobre o estado da conexão com a rede, é possível despoletar uma ação de sincronização de dados, ou caso não exista conectividade apenas guardar o estado desta no Redux Modules;
- Ações 4 a 7 – Antes de ser iniciada a sincronização deve ser obtida a última data de sincronização para aquele utilizador e dispositivo móvel;
- Ações 8 a 13 – Após ser obtida a data da última sincronização, é então iniciado o processo de sincronização. Este processo tem sempre em conta o utilizador e a data da última sincronização. Para que isto aconteça, é elaborada uma chamada à SGICM-Mobile REST API por cada conjunto de dados que se pretende que seja sincronizado (ação 8). O mecanismo de sincronização tem o objetivo de fazer a sincronização dos dados novos da BD local, mas também de guardar os dados novos que foram inseridos na BD principal desde a última sincronização (ações 9-12). Então, inicia-se a sincronização pelos consumos para que os *stocks* dos produtos sejam atualizados na BD principal e posteriormente na BD local;
- Ações 14 a 20 – Após todos os conjuntos de dados serem atualizados ou inseridos é necessário guardar a data de fim desta sincronização.

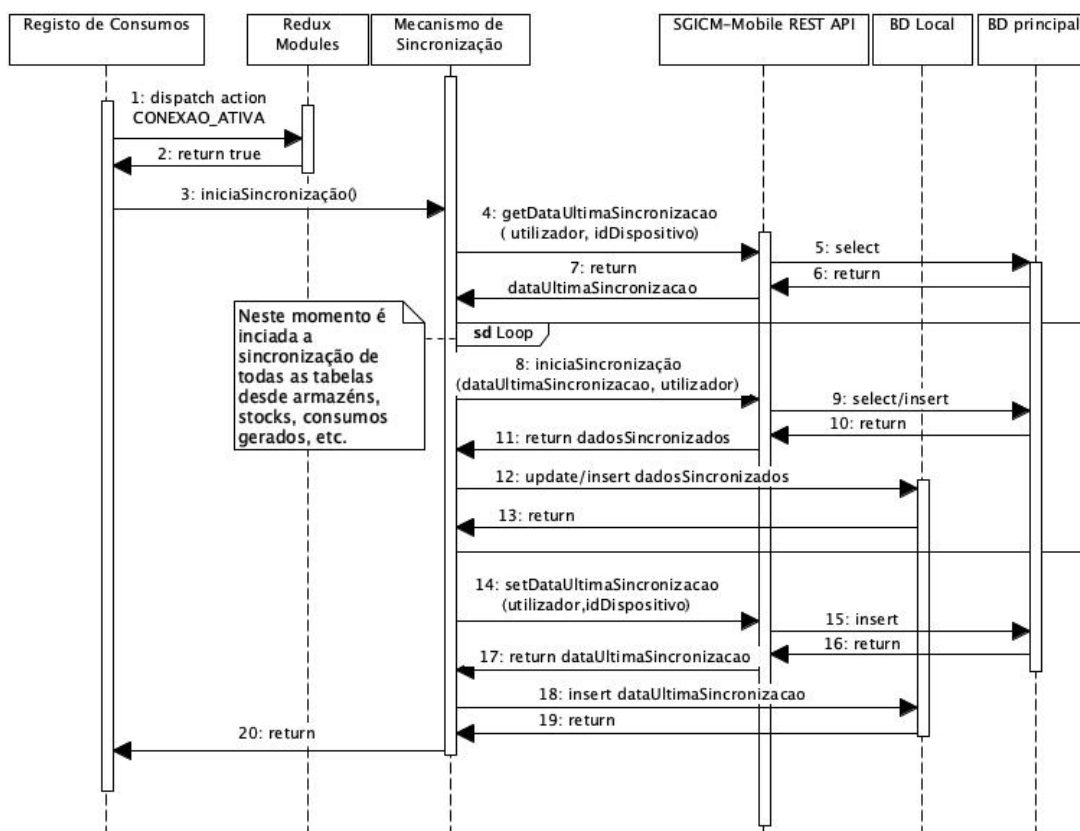


Figura 22 – Diagrama de sequência de sincronização de dados

4.7 Modelo de Dados

Conforme é referido na subsecção 4.1.3, deve existir uma BD local no *smartphone*. Esta BD têm o objetivo de guardar todos os movimentos elaborados pelo utilizador tanto em modo *online* como *offline*.

A Figura 23 mostra três tabelas, que são usadas na gravação de registos de consumos e que são:

- Tabela *Consumo_Temporário* – Assim que é gravado um consumo, estes dados são inseridos nesta tabela. Um consumo pode ter vários artigos, isto significa que na tabela podem existir diversas linhas com o mesmo número temporário (*num_doc_temp*). Para sabermos quais os consumos que já foram sincronizados, é possível olhar para a *flag_proc*, que guarda se a coluna já foi ou não sincronizada;
- Tabela *Consumo* – Após ser feita a sincronização de um consumo, é inserida uma linha nesta tabela que faz corresponder o número de documento temporário ao número de documento real gerado no sistema principal do SGICM;
- Tabela *Sincronizacao_Logs* – Antes de iniciar uma sincronização, é inserida nesta tabela o utilizador, o identificador único do dispositivo, a data de início da nova sincronização e a data da última sincronização. Assim que for terminada a sincronização é inserida a data de fim e uma mensagem de erro, caso algo tenha corrido mal.

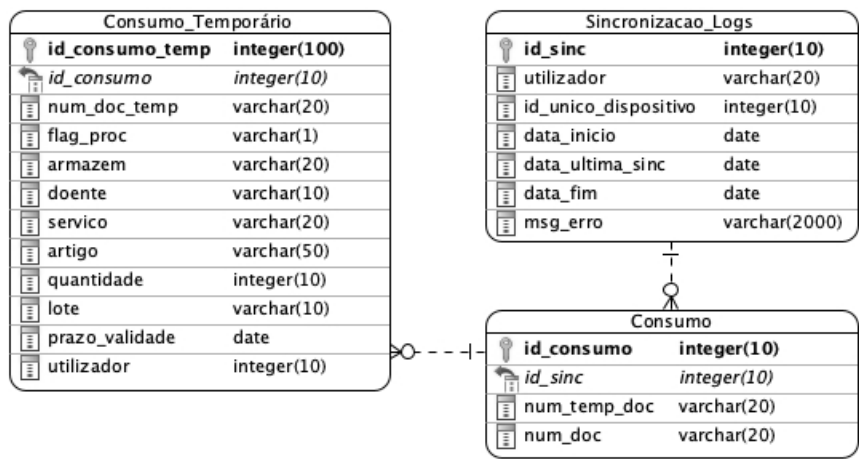


Figura 23 – Modelo de dados parcial (BD local)

Nesta BD local também existem tabelas que guardam os dados referentes aos armazéns, permissões dos utilizadores, artigos, serviços, etc. De forma a representar a estrutura dos dados, as suas características e relações, foi elaborado o diagrama de entidade relação (ver Figura 36) que se encontra no Anexo IV.

No contexto deste problema a eliminação de registos apenas pode ser elaborada via a aplicação *web* SGICM, e mesmo esta eliminação transmite-se na inativação de um campo. Este campo permite verificar, por exemplo, se um armazém está ativo ou não. Durante a construção do mecanismo de sincronização, na BD local foi utilizada a mesma ideia, utilizando um campo para controlar se os dados podem ser visualizados ou não. Para poder aplicar o padrão de transferência de dados algorítmico, que utiliza a *timestamp*, é necessário que todas as tabelas da BD principal incluam colunas com a data de criação e a data de alteração dos dados. É importante referir que o *schema* da BD principal, já adota estas práticas, permitindo assim a obtenção dos dados que foram alterados desde a última data de sincronização.

4.8 Sumário

Neste capítulo é descrita a análise sobre a solução desenvolvida. Na secção 4.1 foram apresentados os casos de uso e requisitos levantados junto da empresa. Na secção 4.2 é apresentada a abordagem de implementação, em que obtemos a resposta à **Q2** – “Qual o mecanismo de sincronização que melhor se adequa à solução a ser desenvolvida?”. Para responder a esta questão é necessário responder às suas alíneas. Portanto, quanto à **Q2.1** – “Será que a sincronização e dados deverá ser síncrona ou assíncrona?”, foi decidido que maioritariamente todas as operações de sincronização são assíncronas, com a exceção da primeira sincronização. Na **Q2.2** – “Qual o padrão de armazenamento de dados que deverá ser usado?”, foi escolhido o armazenamento total de dados, mas apenas são sincronizados os dados que realmente podem ser necessários. Por fim, é obtida a resposta à **Q2.3** – “Qual o padrão de transferência de dados que deve ser implementado?”, em que ficou decidido implementar um mecanismo de sincronização algorítmico. Além disto, foi apresentado o *design* idealizado para solucionar o problema de sincronização de dados. Para mais informações sobre a implementação, pode ser consultado o Anexo III, que demonstra o desenho da interface gráfica da aplicação. O Anexo V pretende detalhar algumas das funcionalidades implementadas.

5 Avaliação

Neste capítulo, são descritas as experiências e avaliações à solução preconizada. Inicialmente, são abordadas as grandezas a utilizar para a avaliação da solução. De seguida são descritas as hipóteses que irão ser testadas para suportar os resultados do trabalho desenvolvido. Por fim, serão apresentados os resultados dos testes efetuados onde se pretende responder à questão Q5, ou seja, perceber se o mecanismo de sincronização construído é viável.

5.1 Grandezas

Nesta secção são apresentadas as grandezas a avaliar, com o objetivo de descrever como será avaliado o trabalho desenvolvido. Sendo estas grandezas:

- Tempo de sincronização de dados – esta grandeza irá avaliar, se o mecanismo de sincronização desenvolvido, têm um menor tempo de execução que a solução anterior SGICM-Mobile *offline*.
- Exatidão na sincronização de dados – o objetivo da utilização desta grandeza é perceber se o trabalho de um utilizador, num ambiente sem conexão ativa à Wi-Fi, é sincronizado de maneira exata para o servidor local quando atingido um local com conexão ativa. Também serão elaborados testes para verificar se os dados da BD principal são sincronizados com o *smartphone*. Deverá também ser analisada a consistências e integridade dos dados sincronizados, com o objetivo de comprovar que a sincronização foi bem-sucedida.
- Nível de Satisfação da Empresa – com a análise desta grandeza, pretende-se perceber qual o nível de satisfação dos colaboradores da Glintt, com a implementação desta nova solução. Esta grandeza deverá ser avaliada dentro da equipa SGICM, que é constituída por uma subequipa de desenvolvedores de *software* e uma equipa de qualidade.

5.2 Identificação de Hipóteses

Com o objetivo de suportar os resultados do trabalho desenvolvido são identificadas as seguintes hipóteses:

- Taxa de exatidão na sincronização de dados superior a 90%;
- Tempo de sincronização de dados inferior a 300 segundos;
- Nível de satisfação da equipa SGICM com a nova solução desenvolvida.

5.3 Metodologia de avaliação

Para avaliar o tempo de sincronização de dados, deverá ser cronometrado o tempo que cada sincronização demora até a sua conclusão.

Quanto à avaliação da exatidão dos dados sincronizados, deverá ser definido um conjunto de dados de teste que irão ser sincronizados. Também deverá ser definido qual o estado dos dados após a sincronização para percebermos se a sincronização foi bem-sucedida.

Por fim, para medir o nível de satisfação da equipa SGICM com a nova solução, serão elaborados inquéritos. As perguntas definidas serão de escolha múltipla e cuidadosamente elaboradas para prever o grau de satisfação da solução implementada.

5.4 Testes estatísticos

Um teste de hipótese analisa duas hipóteses opostas, a hipótese nula (H_0) e a hipótese alternativa (H_1). A hipótese nula é a declaração que está a ser testada. Normalmente, a hipótese nula é uma declaração de nenhum efeito. A hipótese alternativa é a declaração que se pretende ser capaz de concluir, que é verdadeira com base em evidências fornecidas pelos dados da amostra (Minitab, 2017).

Com base nos dados amostrais, o teste determina se devemos rejeitar a hipótese nula. Para fazer esta determinação é usado um *p-value* que faz essa determinação. Se o valor de *p* for menor que o nível de significância, então é possível rejeitar a hipótese nula (Minitab, 2017).

Quando se escreve um teste de hipótese, a hipótese nula deverá ser a excluída. Como o nível de significância é retificado para ser pequeno antes da análise, quando é rejeitada a hipótese nula, existe a prova estatística de que a alternativa é verdadeira.

5.4.1 Tempo de Sincronização de Dados

Para avaliar o tempo de sincronização de dados, foram elaboradas 20 sincronizações. Destas 20 sincronizações, foram elaboradas 10 utilizando um dispositivo Android e as restantes 10 num dispositivo iOS.

Cada sincronização elaborada, enviou cerca de 9266 registos para o dispositivo móvel. Em que foram sincronizados 3352 artigos, 5829 permissões de acesso, 11 armazéns, 70 doentes e 4 serviços. É importante referir que estes testes foram elaborados num ambiente de testes da Glintt, em que foi replicado um cenário com dados reais hospitalares.

O objetivo deste teste é avaliar o tempo que demora uma sincronização, quando o utilizador faz *login* pela primeira vez na aplicação.

Tabela 13 – Tempos de sincronização

Tentativa	Android (tempo em segundos)	iOS (tempo em segundos)	Estado
1	231	231	Sucesso
2	230	226	Sucesso
3	229	221	Sucesso
4	226	226	Sucesso
5	227	219	Sucesso
6	225	230	Sucesso
7	230	228	Sucesso
8	223	223	Sucesso
9	220	221	Sucesso
10	226	225	Sucesso

Após analisar a Tabela 13, é possível verificar que a média de tempos obtidos numa sincronização por tipo de sistema operativo foi para Android 226.7 segundos e de 225.0 segundos para iOS. Tendo sido, a média total de 225.85 segundos.

De modo a confirmar, que os tempos de sincronização são inferiores a 300 segundos, será elaborado um teste de hipóteses. Inicialmente começa-se por verificar se os dados têm uma distribuição normal, sendo as hipóteses:

H_0 : segue a distribuição normal

H_1 : não segue a distribuição normal

```

1. > teste_1 <- c(231,231,230,226,229,221,226,226,227,219,225,230,230,228,223,22
  3,220,221,226,225)
2. > shapiro.test(teste_1)
3.
4.      Shapiro-Wilk normality test
5.
6. data:  teste_1
7. W = 0.93943, p-value = 0.2339

```

Código 1 – Teste de Shapiro para tempo de sincronização de dados inferior a 300s

Utilizando o RStudio foi possível aplicar o teste de Shapiro, em que se obtém o *p-value* 0.2339, conforme mostra o Código 1. Como o *p-value* é superior ao nível de significância de 0.05, não se rejeita a hipótese nula, H_0 , pelo que se pode afirmar com um grau de confiança de 95% que a amostra segue uma distribuição normal.

De seguida define-se as hipóteses para o tempo de sincronização:

$$H_0: \mu \geq 300 \text{ s}$$

$$H_1: \mu < 300 \text{ s}$$

A análise de tempo de sincronização irá utilizar o método *one sample t-test*, pois pode-se afirmar que a amostra segue uma distribuição normal.

```
1. > t.test(teste_1, mu = 300, alternative = "less")
2.
3.      One Sample t-test
4.
5. data: teste_1
6. t = -88.535, df = 19, p-value < 2.2e-16
7. alternative hypothesis: true mean is less than 300
8. 95 percent confidence interval:
9.      -Inf 227.2982
10. sample estimates:
11. mean of x
12.      225.85
```

Código 2 – T-test para tempo de sincronização de dados inferior a 300s

Conforme demonstra o Código 2, verifica-se que o *p-value* é de 2.2e-16. Como *p-value* é menor que 0.05, rejeita-se a hipótese nula. Desta forma pode-se afirmar com um grau de confiança de 95% que o tempo de sincronização é inferior a 300 segundos.

5.4.2 Taxa de Exatidão na Sincronização de Dados

Para avaliar a taxa de exatidão na sincronização de dados, foram elaboradas 10 sincronizações. Para cada sincronização foram monitorizados os dados inseridos ou atualizados na BD local do dispositivo móvel e a BD principal.

Durantes estas sincronizações foram efetuadas as seguintes ações, algumas repetidas:

- Sincronização de todos os dados (primeira sincronização);
- Registo de ofertas via aplicação SGICM *web*, o registo de oferta deu entrada de alguns medicamentos e lotes em diferentes armazéns;
- Criação de consumos via SGICM-Mobile;
- Facultado acesso de um armazém a um utilizador, isto faz com que para além de sincronizar o novo armazém tem de sincronizar dados de medicamentos, lotes e permissões;
- Criação de um novo medicamento e registo de oferta deste num armazém;
- Dado acesso de um serviço ao utilizador;

- Não foi efetuada nenhuma ação.

O objetivo deste teste é avaliar a taxa de exatidão dos dados sincronizados. Estes foram preparados antecipadamente onde foram definidos os conjuntos de dados a sincronizar e os resultados que seriam esperados. É importante referir que estes testes foram elaborados num ambiente de testes da Glintt, em que foi replicado um cenário com dados reais hospitalares.

Tabela 14 – Tempos de sincronização

Tentativa	Nº de registos a sincronizar	Nº de registos sincronizados	Taxa de Sucesso
1	7271	7267	99.944%
2	12	14	85.714%
3	27	27	100%
4	0	0	100%
5	5	5	100%
6	69	69	100%
7	4	4	100%
8	102	102	100%
9	6	6	100%
10	13	13	100%

Após analisar a Tabela 14, é possível verificar que existiram duas sincronizações com taxa de sucesso inferior a 100%. De modo a confirmar que a taxa de exatidão na sincronização de dados é superior a 90%, será elaborado um teste de hipóteses. Inicialmente começa-se por verificar se os dados têm uma distribuição normal, sendo as hipóteses:

H_0 : segue a distribuição normal

H_1 : não segue a distribuição normal

```

1. > teste_1 <- c(99.944,85.714,100,100,100,100,100,100,100,100)
2. > shapiro.test(teste_1)
3.
4.      Shapiro-Wilk normality test
5.
6. data:  teste_1
7. W = 0.36768, p-value = 1.057e-07

```

Código 3 – Teste de Shapiro para taxa de exatidão na sincronização de dados

Utilizando o RStudio foi possível aplicar o teste de Shapiro, em que se pode verificar que o *p-value* é 1.057e-07, conforme o Código 3. Como o *p-value* é inferior ao nível de significância de 0.05, rejeita-se a hipótese nula H_0 , pelo que se pode afirmar com um grau de confiança de 95% que a amostra não segue uma distribuição normal.

Como a amostra não segue uma distribuição normal, é necessário utilizar o teste de Wilcoxon. De seguida define-se as hipóteses para a taxa de exatidão:

$H_0: \mu \leq 90\%$

$H_1: \mu > 90\%$

```
1. > wilcox.test(teste_1, mu = 90, alternative = "greater")
2.
3.      Wilcoxon signed rank test with continuity correction
4.
5. data:  teste_1
6. V = 54, p-value = 0.002495
7. alternative hypothesis: true location is greater than 90
```

Código 4 – Teste de Wilcoxon para taxa de exatidão na sincronização de dados

Conforme mostra o Código 4, verifica-se que o *p-value* é de 0.002495. Como *p-value* é menor que 0.05, rejeita-se a hipótese nula. Desta forma pode-se afirmar com um grau de confiança de 95% que taxa de exatidão na sincronização de dados é superior a 90%.

5.4.3 Nível de Satisfação da Equipa SGICM com a Nova Solução Desenvolvida

De forma a ser possível analisar esta hipótese será elaborado um inquérito que irá medir o grau de satisfação na Glintt, relativamente à nova solução implementada.

Neste inquérito, irão ser elaboradas um conjunto de questões com o objetivo de averiguar a usabilidade, desempenho, tolerância a falhas e coexistência com as outras soluções na área de farmácia e logística hospitalar.

Para ser possível a análise dos resultados dos inquéritos, serão elaboradas questões de dois tipos, aberta e escolha múltipla. Para o tipo de questões fechada irá ser utilizada a escala apresentada na Tabela 15.

Tabela 15 – Correspondência entre a escolha do inquérito e o seu valor numérico

Grau de concordância/discordância	Valor da pontuação
Muito satisfeito	5
Satisfeito	4
Razoavelmente satisfeito	3
Pouco satisfeito	2
Nada satisfeito	1

De forma a recolher a informação necessária, foi elaborado um inquérito via Google Forms e enviado para 10 colaboradores. A quantidade de inquiridos é pequena, pois foi a possível mediante a quantidade de elementos da equipa de desenvolvimento e qualidade de Farmácia e Logística.

Antes do preenchimento deste questionário foi apresentado o projeto e uma demonstração da nova aplicação móvel. Após a demonstração, os colaboradores experimentaram a aplicação na qual tentaram criar um registo de consumos. É importante referir que estes testes foram elaborados num ambiente de testes da Glintt, em que foi replicado um cenário com dados reais hospitalares.

O questionário apresenta 10 perguntas, em que 7 perguntas pretendem avaliar o grau de satisfação da empresa. As restantes 3 perguntas, foram elaboradas para obter a opinião dos inquiridos sobre o projeto e sugestões de melhorias.

Resultados do Inquérito de Satisfação

De seguida são apresentados os resultados, em forma de gráfico com o intuito de promover uma melhor compreensão dos resultados obtidos.

1. Qual foi o nível de satisfação relativo ao tempo de espera durante a primeira sincronização de dados?

10 respostas

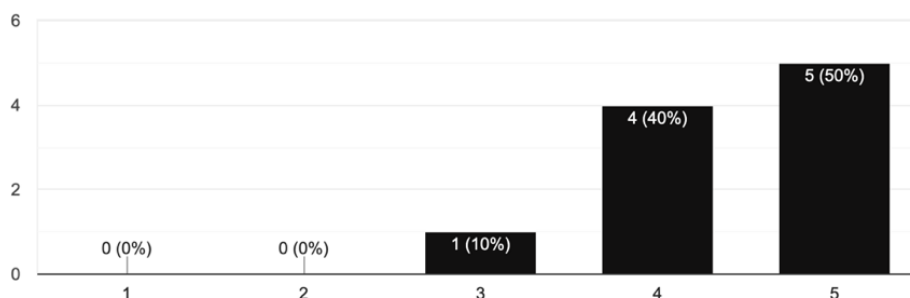


Figura 24 – Inquérito: Grau de satisfação relativo ao tempo da primeira sincronização

O tempo da primeira sincronização é mais longo, pois são carregados milhares de registos para o dispositivo móvel. Como este processo é síncrono, significa que os utilizadores não podem efetuar mais nenhuma tarefa na aplicação móvel, até que a sincronização termine. Mesmo assim, a Figura 24 demonstra que 5 inquiridos (50%), afirmam estar muito satisfeitos, 4 inquiridos (40%) satisfeitos e 1 razoavelmente satisfeito.

2. Após a primeira sincronização, qual foi o nível de satisfação relativo ao tempo de sincronização de dados?

10 respostas

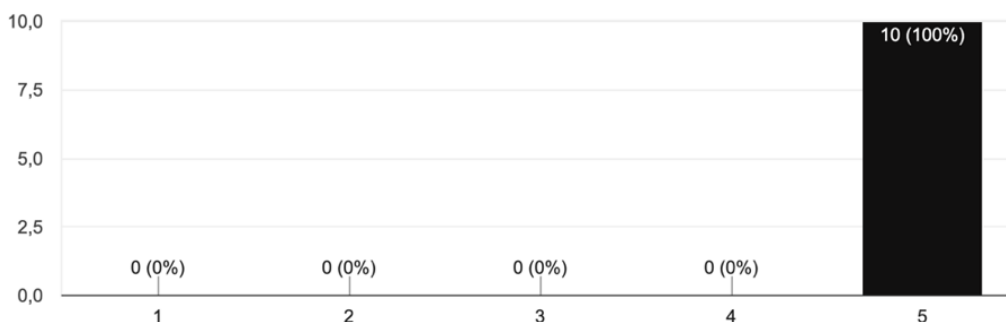


Figura 25 – Inquérito: Grau de satisfação relativo ao tempo de sincronização

Após a primeira sincronização, os tempos de sincronização são bastante menores, principalmente pois o volume de dados é bastante reduzido. Para além disto, o processo é assíncrono o que leva o utilizador a não ser impactado e a não ter de ficar à espera. Segundo a Figura 25, todos os inquiridos (100%) ficaram muito satisfeitos com o tempo de sincronização.

3. Qual foi o nível de satisfação relativo à consistência dos dados sincronizados?

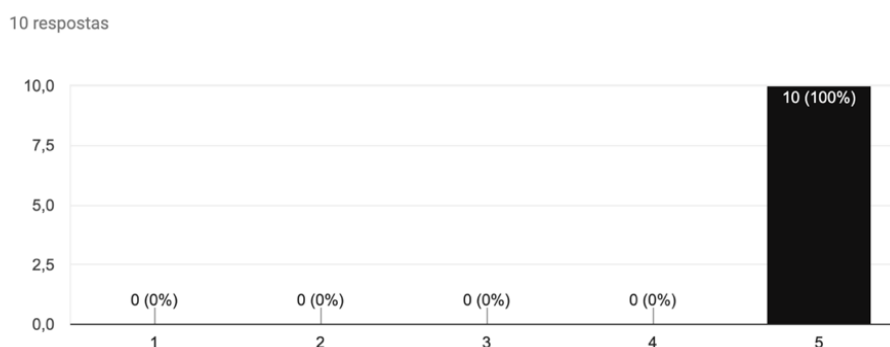


Figura 26 – Inquérito: Grau de satisfação relativo à consistência de dados sincronizados

Esta questão pretende medir se os dados sincronizados, foram corretos. Apesar de ser impossível verificar se todos os dados foram sincronizados corretamente, durante a apresentação e experimentação, não foram identificados dados errados. Como tal, isto resultou que todos os inquiridos (100%) ficassem muito satisfeitos, conforme demonstra a Figura 26.

4. Qual o grau de satisfação relativo ao processo de registo de consumos em modo *offline*?

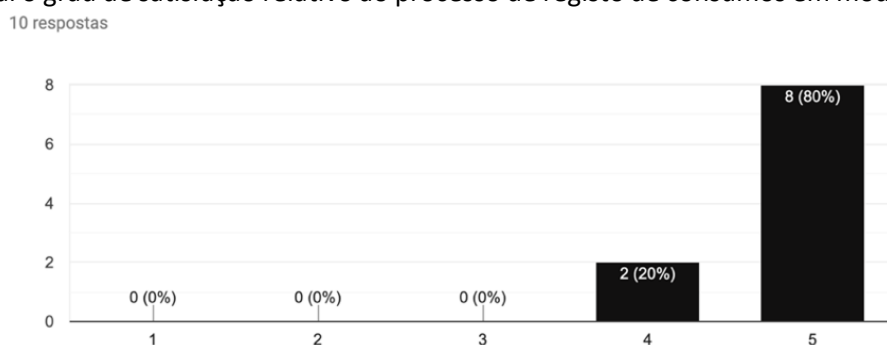


Figura 27 – Inquérito: Grau de satisfação relativo à geração de um consumo em *offline*

A pergunta 4 pretende recolher informação sobre o processo de registo de consumos *offline*. O processo envolveu escolher o local físico, um armazém e adicionar alguns artigos ao consumo. Por fim foi gerado um documento temporário. A Figura 27, mostra que 8 inquiridos (80%) afirmam estar muito satisfeitos, enquanto que os restantes 2 se encontram satisfeitos.

5. Qual o grau de satisfação relativo ao ecrã que mostra os estados de sincronização?

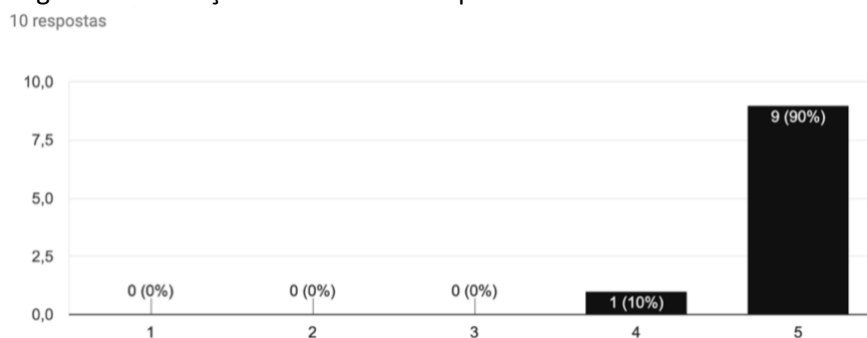


Figura 28 – Inquérito: Grau de satisfação relativo ao ecrã de estado de sincronização

Após ser gerado o consumo em modo *offline*, é possível navegar até o ecrã de estado de sincronização. Aqui é apresentado uma listagem dos documentos a sincronizar e sincronizados. Neste momento, a conexão Wi-Fi foi ativada e foram sincronizados os consumos pendentes. A Figura 28, mostra que 9 inquiridos (90%) ficaram muito satisfeitos e 1 inquirido ficou satisfeito.

6. Qual foi o nível de satisfação relativo às tecnologias escolhidas para este projeto?

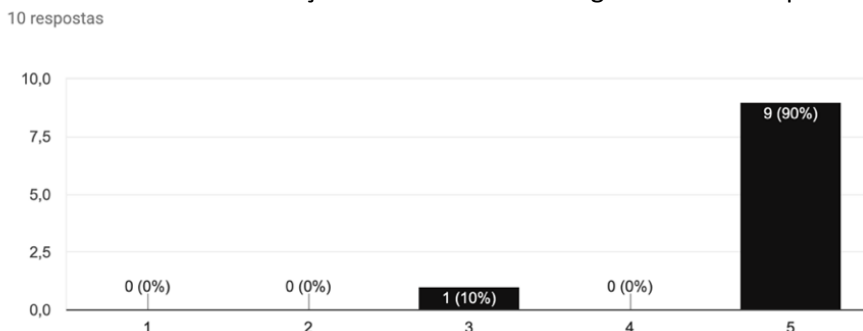


Figura 29 – Inquérito: Grau de satisfação relativo às tecnologias escolhidas

O objetivo principal desta questão é perceber se os inquiridos acham apropriadas as tecnologias escolhidas para resolver o problema de sincronização de dados. Durante a apresentação, foram faladas sobre as tecnologias escolhidas. A Figura 29 mostra que 9 inquiridos (90%) estão muito satisfeitos, enquanto que 1 inquirido se encontra razoavelmente satisfeito.

7. Qual o seu grau de satisfação relativo ao projeto?

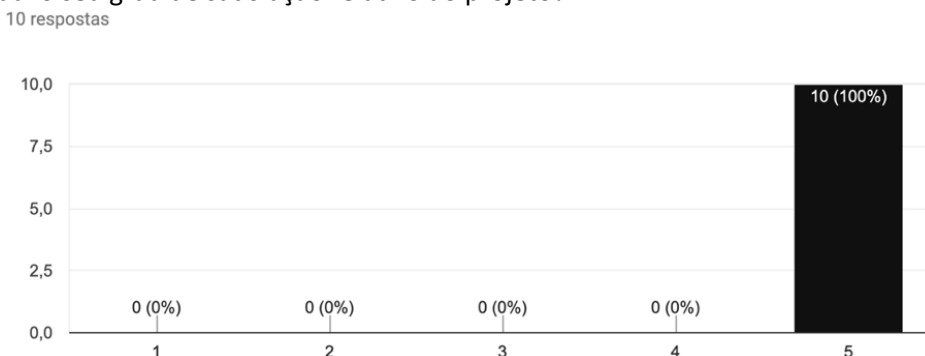


Figura 30 – Inquérito: Grau de satisfação relativo ao projeto global

Por fim, os inquiridos foram questionados sobre o nível geral de satisfação relativo ao projeto realizado. Segundo a Figura 30, os 10 inquiridos (100%) ficaram muito satisfeitos com os resultados.

Análise dos Resultados do Inquérito de Satisfação

Normalmente, ao analisar uma amostra com 30 ou mais observações considera-se a distribuição normal, logo como existem 80 respostas sobre o grau de satisfação considera-se que existe uma distribuição normal na amostra. Para avaliar os resultados do inquérito, define-se que a média deve ser superior a 4, ou seja, satisfeito.

Na Tabela 16, é apresentado o número de respostas por pergunta do inquérito.

Tabela 16 – Nível de Satisfação: Número de respostas por pergunta

Grau de concordância/discordância	Número de repostas
5 - Muito satisfeito	61
4 - Satisfeito	7
3 - Razoavelmente satisfeito	2
2 - Pouco satisfeito	0
1 - Nada satisfeito	0

Como tal define-se as hipóteses para avaliar a satisfação na empresa:

$$H_0: \mu \leq 4$$

$$H_1: \mu > 4$$

Sendo μ é a média das respostas relativas à nova solução, pretende-se provar que a satisfação foi superior a 4. Para tal irá ser utilizado o método *one sample t-test*, pois pode-se afirmar que a amostra segue uma distribuição normal.

Após realizar o teste verifica-se que o *p-value* é de 2.2e-16. Como *p-value* é menor que 0.05, rejeita-se a hipótese nula. Desta forma pode-se afirmar com um grau de confiança de 95% que o nível de satisfação é superior a 4 (satisfeito).

Opinião e Sugestões de Melhoria

As últimas 3 perguntas do inquérito pretendiam recolher sugestões de melhoria e a opinião geral sobre o projeto desenvolvido. Como tal, de seguida são apresentadas as respostas recolhidas.

8. Sugestões de melhoria?

- Alargar a solução a outros módulos de logística;
- Alterações no *layout* de maneira a se tornar mais apelativo visualmente;
- Possibilidade de escolha de sincronização de apenas alguns consumos ou de todos de uma vez;

Na pergunta 8 os inquiridos poderiam dar sugestões de melhorias sobre a aplicação desenvolvida. Em geral cerca de 4 dos inquiridos responderam que uma melhoria seria a implementação de novos módulos logísticos, outros 5 responderam que poderia existir melhorias a nível do *layout* da aplicação. Por fim, existiu 1 inquirido que sugeriu a possibilidade de escolha de movimentos a sincronizar.

Todas as sugestões de melhoria fornecidas ao longo das respostas constituem novas funcionalidades, que devem ser consideradas para um trabalho futuro. Assim sendo, todas as sugestões de melhoria são consideradas uma grande mais valia para o projeto pois permitem perceber o que pode ser melhorado.

9. No futuro, gostarias de ver esta solução melhorada e continuada?

10 respostas

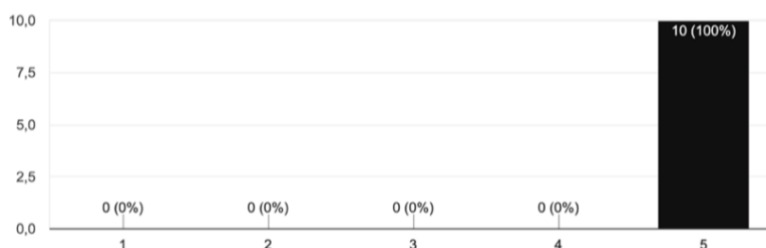


Figura 31 – Inquérito: Opinião pessoal sobre continuação do projeto

10. Opinião geral do projeto, a sua importância, impactos positivos e negativos.

- Uma boa iniciativa de uso mais fácil/intuitivo do que a maior parte das soluções presentes no mercado;
- É uma evolução do produto atual, aonde terá uma importância bastante grande na evolução da aplicação já existente;
- Vai permitir expandir o leque de aparelhos que os clientes podem utilizar assim bem como a usabilidade da aplicação e melhorar o tempo de desenvolvimento;
- Solução muito bem pensada e estruturada e que após terminada e alargada será uma grande mais valia nos nossos clientes;
- De uma forma geral a implementação deste projeto nos clientes vai ter melhorias significativas no seu processo de trabalho. A aplicação está bastante *user-friendly* o que torna o processo de aprendizagem e usabilidade por parte do utilizador mais fácil.
- Comparado com o produto anterior (para PDA), verifica-se um tempo de resposta muito mais rápido em todos os aspetos. Visual e ecrã intuitivo, atraente e "amigo do utilizador". Importação dos dados muito mais fácil para o utilizador.
- Importante para a prática quotidiana em meio hospitalar em que os consumos são elevados e constantes.

As perguntas 9 e 10 do questionário têm um objetivo de obter a opinião dos inquiridos, sobre a solução final desenvolvida. Em modo geral as respostas recebidas foram bastante positivas.

5.5 Sumário

Neste capítulo obteve-se a resposta à **Q5** – “Será possível construir um mecanismo de sincronização viável, que permita trabalhar em modo *online/offline* no contexto deste problema?”. Ao analisar os resultados obtidos nas avaliações deste capítulo, é possível considerar o mecanismo desenvolvido viável, pois as hipóteses inicialmente definidas foram cumpridas com sucesso. Os resultados do questionário de satisfação também foram bastante positivos.

6 Conclusão

Neste capítulo serão apresentadas as conclusões finais do trabalho desenvolvido. Inicialmente neste capítulo será lembrado o contexto e problema desta dissertação. Também, serão revistas todas as perguntas de pesquisa apresentadas na secção 1.3 e respostas obtidas ao longo deste trabalho.

De seguida, serão apresentadas as limitações encontradas durante o desenvolvimento do trabalho. Por fim será apresentada uma apreciação final que pretende resumir todos os objetivos alcançados e conhecimentos adquiridos.

6.1 Síntese

A informação do registo de utilização de um bem é fulcral para a área de prestação de cuidados como medida de segurança. É necessário ter conhecimento por quem e quando foi utilizado um determinado produto. Por esta razão, o registo de consumos mais próximo do seu ponto de utilização é fulcral.

As instituições hospitalares quer pela sua natureza arquitetónica quer pelas suas restrições de segurança e infraestrutura de rede Wi-Fi, o registo da informação do consumo de um determinado bem nem sempre ocorre a quando da utilização efetiva. Muitas vezes o registo de movimentos é efetuado á priori ou á posteriori tendo como base informação transcrita de papel.

Assim com vista a ajudar a mitigar e muitas vezes a eliminar as situações previamente identificadas surge a necessidade de existir uma solução de suporte ao registo da utilização de bens. Esta nova solução deve permitir que seja efetuado o registo digital, o mais próximo da sua utilização, mesmo em locais críticos ou sem cobertura de rede.

Em ambiente hospitalar nem sempre é possível usufruir de uma conexão à Wi-Fi, sendo que a Glintt pretende construir uma solução que funcione em modo *offline* e *online*. Um dos seus

objetivos é que os seus utilizadores tenham uma experiência flexível quando a conexão estiver lenta, oscilante ou mesmo quando esta é inexistente.

Outro problema, que afeta principalmente a equipa de desenvolvimento e manutenção de soluções da Glintt, é a descontinuação do sistema operativo Windows Mobile. As aplicações SGICM-Mobile foram implementadas para serem utilizadas neste sistema operativo nas versões Windows Mobile CE e na versão Windows Mobile 6.5. Este problema leva a equipa de desenvolvimento a gastar tempos elevados, sempre que é necessária a correção ou alteração de um comportamento, tendo assim custos acrescidos associados a esta necessidade.

6.2 Objetivos Atingidos

Nesta secção serão revistas todas as questões de pesquisa apresentadas na secção 1.3 e respostas obtidas ao longo deste trabalho.

6.2.1 Questão Q1

A questão **Q1** – “Qual o mecanismo de sincronização de dados, que permite o melhor desempenho e coerência de dados?”, foi respondida na secção 2.3.3. O melhor mecanismo de sincronização/transferência de dados é a transferência matemática ou algorítmica de dados.

Para obter esta resposta foram estudados três diferentes mecanismos de transferência de dados em que foi possível perceber que este permite o melhor desempenho e coerência de dados.

6.2.2 Questão Q2

Para obter a resposta à **Q2** e suas alíneas foi necessário levantamento prévio de requisitos junto da empresa. Após a análise dos requisitos foi possível identificar duas abordagens diferentes para a construção do mecanismo de sincronização. Após a análise das alternativas foi definida a abordagem de implementação.

Para responder à **Q2** – “Qual o mecanismo de sincronização que melhor se adequa à solução a ser desenvolvida?”, é necessário responder às suas alíneas.

Portanto, quanto à **Q2.1** – “Será que a sincronização de dados deverá ser síncrona ou assíncrona?”, foi decidido que maioritariamente todas as operações de sincronização são assíncronas, com a exceção da primeira sincronização.

Na **Q2.2** – “Qual o padrão de armazenamento de dados que deverá ser usado?”, foi escolhido o armazenamento total de dados, mas apenas são sincronizados os dados que realmente podem ser necessários.

Por fim é obtida a resposta à **Q2.3** – “Qual o padrão de transferência de dados que deve ser implementado?”, em que ficou decidido implementar um mecanismo de sincronização algorítmico.

Tendo sido então implementado um mecanismo de sincronização algorítmico, que trabalha maioritariamente de forma assíncrona e faz um armazenamento total de dados. A resposta à **Q2** foi obtida na secção 4.2.

6.2.3 Questão Q3

A **Q3** – “Quais as vantagens e desvantagens das diferentes *frameworks* multiplataforma de desenvolvimento de aplicações móveis?”, foi respondida na subsecção 2.4.3.

Para obter esta resposta foi elaborada uma comparação entres as *frameworks* estudadas. Deste modo, foi possível observar as vantagens e desvantagens de cada uma destas, em relação às outras.

6.2.4 Questão Q4

A **Q4** – “Qual a *framework* multiplataforma que deve ser adotada pela Glintt?”, foi respondida na secção 3.5.

Através do método AHP, foi possível escolher a *framework* multiplataforma que melhor se adapta à empresa mediante um conjunto de fatores previamente definidos. A *framework* escolhida foi React Native, que foi utilizada na implementação da solução de registos de consumos.

6.2.5 Questão Q5

A resposta à **Q5** – “Será possível construir um mecanismo de sincronização viável, que permita trabalhar em modo *online/offline* no contexto deste problema?”, foi respondida ao longo do quinto capítulo.

Ao analisar os resultados obtidos nas avaliações, é possível considerar o mecanismo desenvolvido viável, pois as hipóteses inicialmente definidas foram cumpridas com sucesso. Os resultados do questionário de satisfação também tiveram uma resposta bastante positiva.

6.3 Limitações

As limitações deste projeto são essencialmente o facto de todo o processo de registo de consumos já existir e não poder existir muita inovação neste desenvolvimento.

Como já existe uma base de código e dados muito grande na aplicação de logística e farmácia hospitalar (SGICM *web*) não foi possível implementar algumas das funcionalidades e validações que deveriam existir no ato de um registo de consumos.

6.4 Apreciação Final

Após a conclusão do projeto atual, considera-se que os objetivos definidos inicialmente foram cumpridos, tendo sido obtida respostas para todas as questões propostas na secção 1.3.

Durante a fase de pesquisa deste projeto, foi possível analisar o valor da solução a ser desenvolvida e o estado de arte de diferentes tecnologias e soluções. Foram adquiridos conhecimentos sobre o desenvolvimento móvel e estudadas diferentes *frameworks* multiplataformas. Este estudo elaborado foi bastante importante pois permitiu escolher a *framework* que melhor se adapta à Glintt.

Durante o desenvolvimento deste trabalho foi possível aprofundar o conhecimento sobre o funcionamento de aplicações em modo *offline*. Foram estudados diferentes mecanismos de sincronização de dados e foi possível compreender em que situações podem e devem ser utilizados.

Tendo isto em conta, foi construído um mecanismo de sincronização de dados, que permite trabalhar em modo *online* e *offline*, sem que o utilizador se aperceba que o dispositivo móvel ficou sem uma conexão Wi-Fi ativa.

Durante a escrita desta dissertação, teve-se em conta as boas práticas de engenharia. Foram tidos em conta os módulos lecionados no contexto da unidade curricular TMDEI. Nomeadamente, a pesquisa e escrita técnico-científica, análise de valor de negócio e experimentação e avaliação, que contribuíram para uma maior qualidade de escrita deste trabalho.

Em conclusão, considero que este projeto foi um contributo bastante positivo para mim, e para a empresa pois permitiu aprofundar os conhecimentos sobre mecanismos de sincronização e *frameworks* multiplataforma.

Referências

A.Koen, P., 2001. *Fuzzy Front End: Effective Methods, Tools, and Techniques*.

[Online]

Available at: http://www.stevens-tech.edu/cce/NEW/PDFs/FuzzyFrontEnd_Old.pdf

[Acedido em 01 02 2019].

Armour, 2018. *5 Key Benefits of Native Mobile App Development*. [Online]

Available at: <https://clearbridgemoible.com/benefits-of-native-mobile-app-development/>

[Acedido em 28 01 2019].

BIQHS, 2018. *EKBAN - Modelo Logístico Hospitalar*. [Online]

Available at:

<https://www.biqhs.com/Solu%C3%A7%C3%B5esIT/Redu%C3%A7%C3%A3odeCustos/Ekanban%C2%AE-ModeloLog%C3%ADsticoHospitalar.aspx>

[Acedido em 21 01 2018].

BIQHS, 2018. *BIQ Health Solutions*. [Online]

Available at: <http://biq.pwm.pt/biq-health-solutions/>

[Acedido em 21 02 2019].

Fernandes, 2018. *Integrating semantic-ui modal with Redux*. [Online]

Available at: <https://itnext.io/integrating-semantic-ui-modal-with-redux-4df36abb755c>

[Acedido em 29 01 2019].

Glas, G., 2018. *Web app vs. Native app*. [Online]

Available at: <https://www.app-press.com/blog/web-app-vs-native-app>

[Acedido em 27 01 2019].

Glantt, 2018. *Glantt*. [Online]

Available at: <https://www.glantt.com/pt/o-que-fazemos/mercados/healthcare/Paginas/Home.aspx>

[Acedido em 20 01 2019].

ICAPPS, 2017. *5 key advantages of React Native*. [Online]

Available at: <https://www.icapps.com/blog/5-key-advantages-react-native>

[Acedido em 29 01 2019].

Ionic, 2018. *An app development platform*. [Online]

Available at: <https://ionicframework.com/>

[Acedido em 28 01 2019].

Kotler, P., 2000. *Marketing Management, Millenium Edition*. [Online]

Available at: <https://www.perspectiva.md/ro/files/biblioteca/Kotler->

Marketing%20Management%20Millenium%20Edition.pdf
[Acedido em 20 02 2019].

M2Mobi, 2018. *medium*. [Online]
Available at: <https://medium.com/m2mobi/web-cross-platform-hybrid-or-native-whats-the-difference-866ca81cc7e0>
[Acedido em 01 02 2019].

M2Mobi, 2018. *Web, Cross-platform, Hybrid or Native? What's the difference?*. [Online]
Available at: <https://medium.com/m2mobi/web-cross-platform-hybrid-or-native-whats-the-difference-866ca81cc7e0>
[Acedido em 27 01 2019].

Maruta, 2018. *Iniciando com Redux em 9 passos*. [Online]
Available at: <https://medium.com/reactbrasil/iniciando-com-redux-c14ca7b7dcf>
[Acedido em 29 01 2018].

McCormick & Schmidt, 2012. *DATA SYNCHRONIZATION PATTERNS IN MOBILE APPLICATION DESIGN*. [Online]
Available at: <https://www.dre.vanderbilt.edu/~schmidt/PDF/PatternPaperv11.pdf>

Microsoft, 2018. *Visual Studio Tools for Xamarin*. [Online]
Available at: <https://visualstudio.microsoft.com/pt-br/xamarin/?rr=https%3A%2F%2Fwww.google.com%2F>
[Acedido em 20 01 2019].

Minitab, 2017. *O que é um teste de hipótese?*. [Online]
Available at: <https://support.minitab.com/pt-br/minitab/18/help-and-how-to/statistics/basic-statistics/supporting-topics/basics/what-is-a-hypothesis-test/>
[Acedido em 10 02 2018].

PÚBLICO, 2018. [Online]
Available at: <https://www.publico.pt/2018/09/12/tecnologia/noticia/telemovel-supera-computador-no-acesso-a-internet-1843867#gs.cRNhex5J>
[Acedido em 02 02 2019].

PINTERAC, 2018. *Cross-platform iOS / Android development, hybrid and native*. [Online]
Available at: <https://pinterac.net/wp-content/uploads/2017/07/app-differences.png>
[Acedido em 30 01 2019].

Saaty, T., 2008. *Decision making with the analytic hierarchy process*. s.l., Int. J. Services Sciences.

Sharma, N., 2018. *React Native Vs. Xamarin Vs. Ionic Vs. Flutter: Which Is Best For Cross-Platform Mobile App Development?*. [Online]
Available at: <https://www.apptunix.com/blog/frameworks-cross-platform-mobile-app-development/>
[Acedido em 29 01 2019].

Sinhal, A., 2017. *MVC, MVP and MVVM Design Pattern*. [Online]
Available at: <https://medium.com/@ankit.sinhal/mvc-mvp-and-mvvm-design-pattern-6e169567bbad>
[Acedido em 30 01 2019].

Viswanathan, B., 2012. *What should everyone know about the Business Model Canvas?*. [Online]
Available at: <https://www.quora.com/What-should-everyone-know-about-the-Business-Model-Canvas>
[Acedido em 14 01 2019].

Woodall, 2003. *Conceptualising 'Value for the Customer': An Attributional, Structural and Dispositional Analysis*. s.l., s.n.

Anexos

Anexo I - Requisitos

De seguida, são identificados os requisitos funcionais e não funcionais do sistema, que irão abordar aspetos de qualidade importantes.

- RF001 – Realizar autenticação;
- RF002 – Adicionar o local físico onde se encontra;
- RF003 – Associar um paciente ao registo de consumos;
- RF004 – Registrar artigos;
- RF005 – Eliminar artigos previamente adicionados;
- RF006 – Gravar o registo de consumos;
- RF007 - Consultar histórico de dados sincronizados;
- RF008 – Aplicação deve funcionar com e sem rede Wi-Fi;
- RF009 – Efetuar sincronização dos dados assim que possuir rede Wi-Fi;
- RF010 – A correção de erros ou alterações devem ser elaboradas no sistema desktop da aplicação;
- NF01 – Portabilidade, o sistema deve ser executado em ambientes distintos, neste caso Android e iOS;
- NF02 – Usabilidade, o sistema deve ser simples de aprender e de utilizar, tornando-se agradável para a realização de tarefas;
- NF03 – Desempenho, este requisito pode afetar a usabilidade do sistema, por isso deve-se ter em conta o tempo de resposta e o espaço ocupado;
- NF04 – Segurança, o sistema não deve permitir acessos não autorizados ao sistema.

Anexo II – Descrição de Casos de Uso

A descrição de UCs é elaborada em forma de tabela. O objetivo desta descrição é que o leitor perceba mais facilmente os UCs e fluxo de ações.

UC01 – Realizar autenticação

Tabela 17 – UC01: Realizar autenticação

Descrição	
Objetivo	Permite ao utilizador autenticar-se na sua conta para começar a trabalhar.
Atores	Profissional do sector clínico (médico, enfermeiro, farmacêutico), profissional do setor não clínico (responsável e técnico de logística)
Pré-condições	Já possuir uma conta de utilizador, criada através da versão <i>web</i> da aplicação.
Pós-condições	O utilizador é reencaminhado para o menu principal da aplicação.
Prioridade	Média
Frequência de uso	Alta
Campos	Nome do utilizador, palavra-passe.
Requisitos	RF001 – Realizar autenticação RF008 – Aplicação deve funcionar com e sem rede Wi-Fi. NF001 – Portabilidade NF002 – Usabilidade NF004 – Segurança
Fluxo principal	<ol style="list-style-type: none">1. O utilizador entra na aplicação.2. O sistema apresenta uma página de autenticação.3. O utilizador introduz nome do utilizador.4. O utilizador introduz a palavra-passe.5. O utilizador clica no botão para fazer a autenticação.6. O sistema verifica com sucesso que o utilizador existe e têm permissões para a ação.7. O sistema reencaminha o utilizador para o menu principal.

UC02 – Adicionar o local físico onde se encontra

Tabela 18 – UC02: Adicionar o local físico onde se encontra

Descrição	
Objetivo	O utilizador poderá selecionar o local físico onde se encontra. O local físico é constituído pelo armazém e serviço.
Atores	Profissional do sector clínico (médico, enfermeiro, farmacêutico), profissional do setor não clínico (responsável e técnico de logística)
Pré-condições	Deve ter existido uma operação de sincronização de dados nomeadamente dos locais.
Pós-condições	O utilizador pode continuar o registo de consumos.
Prioridade	Alta

Frequência de uso	Alta
Campos	Armazém, serviço.
Requisitos	RF002 – Adicionar o local físico onde se encontra. RF008 – Aplicação deve funcionar com e sem rede Wi-Fi. NF001 – Portabilidade NF002 – Usabilidade
Fluxo principal	<ol style="list-style-type: none"> 1. O utilizador clica para adicionar um armazém. 2. O sistema apresenta uma página de seleção de armazéns. 3. O utilizador escolhe o armazém. 4. O utilizador clica para adicionar um serviço. 5. O sistema apresenta uma página de seleção de serviços. 6. O utilizador escolhe o serviço. 7. O sistema elabora as validações de necessárias. 8. O sistema adiciona o local ao consumo. 9. O utilizador vê o local adicionado.

UC03 – Associar um paciente ao registo de consumos

Tabela 19 – UC03: Associar um paciente ao registo de consumos

Descrição	
Objetivo	Por vezes o registo de consumos pode ter pacientes associados. O utilizador pretende associar um paciente a um registo de consumos.
Atores	Profissional do sector clínico (médico, enfermeiro, farmacêutico)
Pré-condições	Deve ter existido uma operação de sincronização de dados nomeadamente dos pacientes.
Pós-condições	O utilizador vê o doente selecionado.
Prioridade	Média
Frequência de uso	Média
Campos	Paciente, episódio.
Requisitos	RF003 – Associar um paciente ao registo de consumos RF008 – Aplicação deve funcionar com e sem rede Wi-Fi NF001 – Portabilidade NF002 – Usabilidade
Fluxo principal	<ol style="list-style-type: none"> 1. O utilizador clica para adicionar um paciente. 2. O sistema apresenta uma página de seleção de pacientes. 3. O utilizador escolhe o paciente. 4. O sistema verifica se o paciente pode ser adicionado. 5. O sistema adiciona o paciente com sucesso. 6. O sistema reencaminha o utilizador para página e registo de consumos. 7. O utilizador vê o paciente associado ao seu consumo.

UC04 – Registrar artigos manualmente

Tabela 20 – UC04: Registrar artigos manualmente

Descrição	
Objetivo	O utilizador poderá selecionar artigos de uma lista fornecida pelo sistema. Após esta seleção deverá identificar a quantidade, o lote e respetivo prazo de validade.
Atores	Profissional do sector clínico (médico, enfermeiro, farmacêutico), profissional do setor não clínico (responsável e técnico de logística)
Pré-condições	Deve ter existido uma operação de sincronização de dados nomeadamente dos artigos, lotes e <i>stocks</i> . Existe <i>stock</i> disponível artigo.
Pós-condições	O utilizador pode continuar a adicionar artigos à sua lista.
Prioridade	Alta
Frequência de uso	Média
Campos	Artigo, lote, <i>stock</i> , referência, número de série, quantidade.
Requisitos	RF004 – Registrar artigos RF008 – Aplicação deve funcionar com e sem rede Wi-Fi. NF001 – Portabilidade NF002 – Usabilidade
Fluxo principal	<ol style="list-style-type: none">1. O utilizador clica para adicionar um artigo.2. O sistema apresenta uma página de seleção de artigos.3. O utilizador escolhe o artigo.4. O sistema retorna o artigo.5. O utilizador adiciona um lote, quantidade.6. O sistema elabora as validações de <i>stock</i>.7. O sistema adiciona o artigo à lista de artigos registados.8. O utilizador vê a sua lista de artigos registados atualizada.

UC05 – Eliminar previamente artigos adicionados

Tabela 21 – UC05: Eliminar previamente artigos adicionados

Descrição	
Objetivo	O utilizador poderá eliminar os artigos que registou na sua lista.
Atores	Profissional do sector clínico (médico, enfermeiro, farmacêutico), profissional do setor não clínico (responsável e técnico de logística)
Pré-condições	Devem existir artigos na listagem.
Pós-condições	O utilizador pode continuar a adicionar ou eliminar artigos à sua lista.
Prioridade	Alta
Frequência de uso	Alta
Campos	Artigo, lote.
Requisitos	RF005 – Eliminar artigos previamente adicionados RF008 – Aplicação deve funcionar com e sem rede Wi-Fi NF001 – Portabilidade NF002 – Usabilidade
Fluxo principal	<ol style="list-style-type: none">1. O utilizador apaga o artigo.2. O sistema remove o artigo da lista de artigos registados.

UC07 – Consultar histórico de dados sincronizados

Tabela 22 – UC07: Consultar histórico de dados sincronizados

Descrição	
Objetivo	O utilizador pode consultar o seu histórico de movimentos e verificar quais foram sincronizados e os números de documentos associados.
Atores	Profissional do sector clínico (médico, enfermeiro, farmacêutico), profissional do setor não clínico (responsável e técnico de logística)
Pré-condições	O utilizador dever ter elaborado pelo menos um registo de consumos.
Pós-condições	O utilizador é reencaminhado para o menu principal.
Prioridade	Alta
Frequência de uso	Média
Campos	Número de documento do consumo.
Requisitos não funcionais	RF007 – Consultar histórico de dados sincronizados. RF008 – Aplicação deve funcionar com e sem rede Wi-Fi NF001 – Portabilidade NF002 – Usabilidade
Fluxo principal	<ol style="list-style-type: none">1. O utilizador clica para entrar no ecrã de estados de sincronização.2. O sistema apresenta os documentos temporários.3. O utilizador sincroniza dos documentos.4. O sistema retorna os documentos definitivos ou erros de sincronização.

Anexo III - Desenho da Interface

Neste anexo são apresentados os ecrãs desenvolvidos para a solução de registo de consumos hospitalares da Glintt. De seguida, são apresentadas as Figuras 32 a 35 que demonstram visualmente a implementação dos ecrãs. Nas seguintes subsecções são explicadas as funcionalidades de cada uma destas figuras.

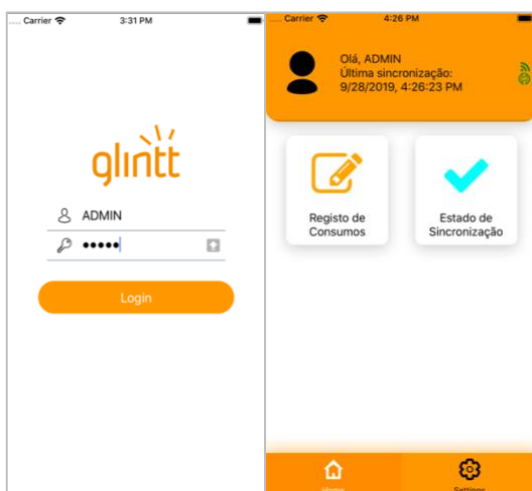


Figura 32 – Ecrã login e home

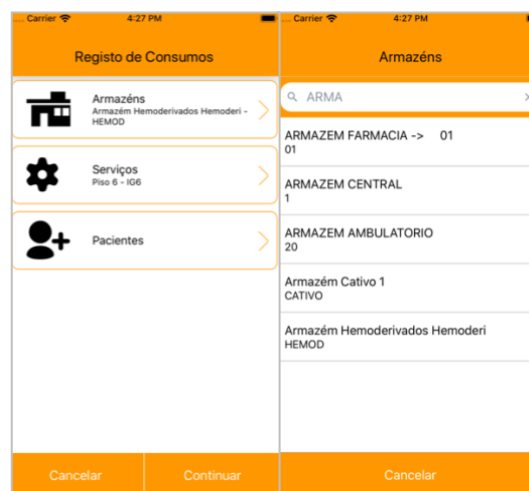


Figura 33 – Ecrã de seleção de local e pacientes

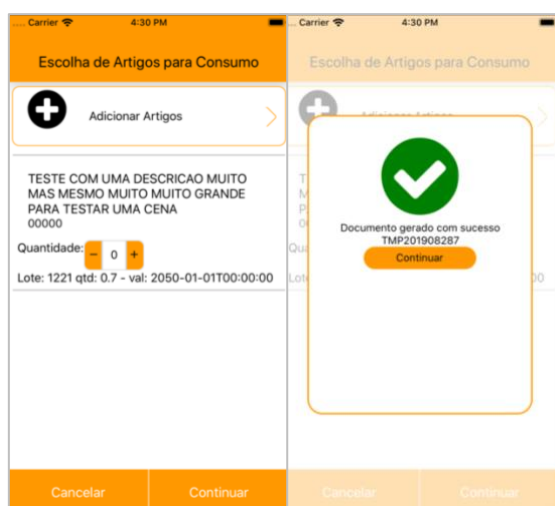


Figura 34 – Ecrã de seleção de artigos

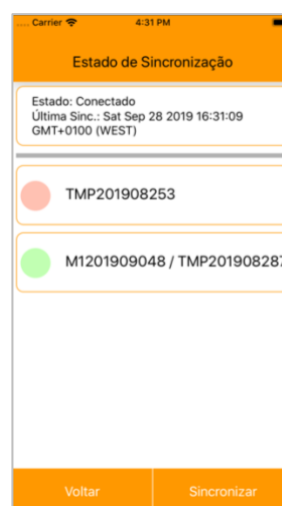


Figura 35 – Ecrã de sincronização de movimentos

Ecrã de *Login* e Home

O ecrã de *login*, será apresentado sempre que o utilizador iniciar a aplicação no seu dispositivo móvel. Assim sendo, tem como objetivo controlar o acesso de utilizadores aos módulos de logística hospitalar.

No âmbito da sincronização de dados, este ecrã tem um comportamento diferente de todos os outros que serão apresentados. Isto acontece, pois, as credenciais de acesso não deverão ser sincronizadas para o dispositivo móvel. Caso estas fossem sincronizadas, poderia comprometer a segurança da aplicação. Isto acontece, pois seria fácil ter acesso à BD local e conseguir todas as palavras secretas de cada utilizador. Como tal foi decidido, que o *login* deve ser feito com o acesso ao servidor principal, necessitando então que o dispositivo tenha uma conexão á rede do hospital.

No entanto, este ecrã também será o primeiro a despoletar uma ação de sincronização de dados, conforme documentado na subsecção 5.1.1. Após o utilizador se autenticar com sucesso, serão carregados todos os dados aos quais estes têm permissão de acesso e que necessita para começar o seu trabalho. Na Figura 32 é possível ver o especto visual do ecrã de *login* implementado. Após o utilizador efetuar o *login*, será apresentada uma lista com os módulos disponíveis. Neste momento o utilizador apenas tem acesso ao módulo de registo de consumos e ao ecrã que mostra os estados de sincronização.

Ecrã de seleção de local e pacientes

Após o utilizador escolher a opção de registo de consumos, este é reencaminhado para um ecrã onde poderá escolher o seu local e paciente. É importante referir que o local é constituído pelo serviço e armazém em que este está a trabalhar. Quando o utilizador escolher uma das opções, é reencaminhado para um ecrã onde pode escolher através de uma lista pesquisável uma opção que foi previamente sincronizada para o seu dispositivo móvel. Na Figura 33 é possível ver o especto visual dos ecrãs de seleção de local e pacientes e respetiva lista pesquisável.

Ecrã de seleção de artigos

Após o utilizador escolher o local e um paciente, este é reencaminhado para um ecrã em que poderá adicionar artigos. Este ecrã permite escolher artigos a partir de uma lista de produtos. De seguida o utilizador pode adicionar um lote e respetiva data de validade, também pode alterar a quantidade e eliminar artigos (Figura 34).

Quando o utilizador, clicar em continuar será então gerado o consumo. Ou seja, será gerado um número temporário. Este documento temporário será então apresentado no ecrã de sincronização de movimentos, permitindo assim ao utilizador perceber quando os seus movimentos são sincronizados com o sistema.

Ecrã de sincronização de movimentos

O utilizador pode consultar este ecrã a partir do ecrã *home*, este ecrã tem como principal objetivo mostrar os estados de sincronização de documentos.

Neste ecrã é possível ver todos os documentos temporários que ainda não foram sincronizados com o sistema principal. No entanto, após a sincronização ser efetuada, é possível ver o número de documento real gerado. Caso ao sincronizar o consumo, aconteça algum erro, como a falta de *stock* informático de um artigo, o utilizador é informado neste ecrã do problema. Na Figura 35 é possível ver o especto visual do ecrã de estados de sincronização.

Anexo IV - Modelo de Dados

De forma a representar a estrutura dos dados, as suas características e relações, foi elaborado o diagrama de entidade relação (ver Figura 36) para que se tenha uma visão detalhada de todas as entidades deste projeto.

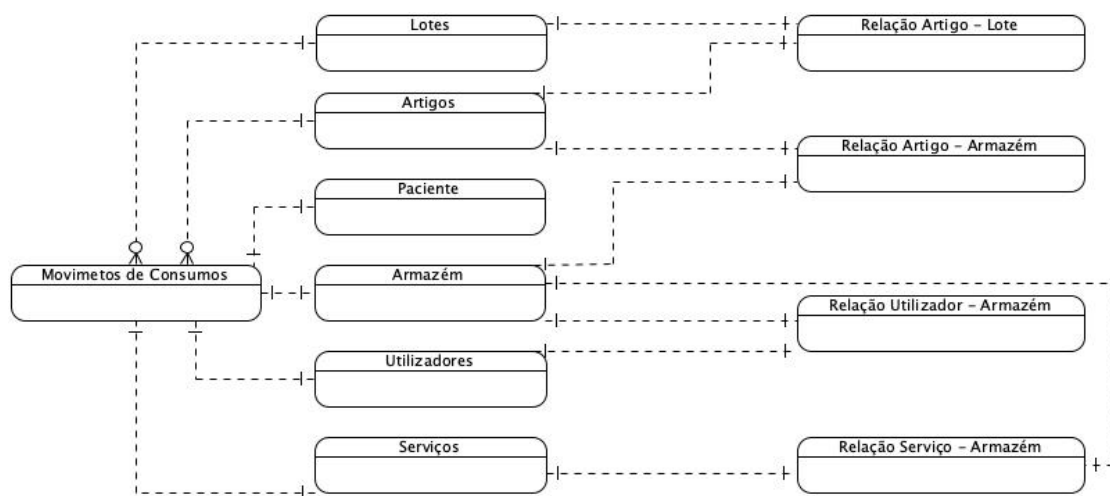


Figura 36 – Diagrama de relação entidade

A Figura 36 descreve todas as entidades e relações existentes durante o registo de um consumo. Como tal é possível verificar que um movimento de consumo tem vários artigos para o mesmo número de documento. Por outro lado, também é possível visualizar que existem tabelas de relação entre as diferentes entidades. Estas tabelas podem ser consideradas como tabelas de permissão. Por exemplo, a tabela *Relação Utilizador – Armazém* contém quais os armazéns que cada utilizador tem acesso.

Anexo V - Funcionalidades desenvolvidas

Este anexo pretende explicar as funcionalidades mais importantes que foram implementadas no âmbito deste projeto. Estas funcionalidades poderão ser reutilizadas caso sejam implementados novos módulos de logística hospitalar. Nomeadamente, poderá ser reutilizada a sincronização de dados e a implementação de componentes visuais.

Sincronização de dados

Esta subsecção pretende demonstrar como foi implementada a sincronização de dados do dispositivo móvel com a BD principal e vice-versa. Na aplicação desenvolvida a sincronização de dados pode ser despoletada de três formas:

- Através do início de sessão;
- Quando o utilizador gravar um registo de consumos;
- O utilizador também pode forçar uma sincronização, manualmente caso tenha rede.

Conforme mostra o Código 5, para iniciar uma sincronização, é necessário enviar o utilizador que efetuou *login*, o identificador único do dispositivo móvel e a última data de sincronização (linha 1). Posteriormente serão enviados pedidos utilizando uma API REST ao servidor, que irá obter os dados novos ou que foram atualizados desde a última data de sincronização (linha 8-17).

```
1. export default async (user: string, deviceInfo: string) =>
2.   Promise.all([
3.     createTables(),
4.     saveSyncProcess(user, deviceInfo),
5.   ])
6.   .then(...);
7.   return Promise.all([
8.     insertSyncLog,
9.     getAllArmazem(user, result.date_last_sync, db),
10.    getAllDoente(user, result.date_last_sync, db),
11.    getAllMedArm(user, result.date_last_sync, db),
12.    getAllMedLote(user, result.date_last_sync, db),
13.    getAllProducts(user, result.date_last_sync, db),
14.    getAllServArm(user, result.date_last_sync, db),
15.    getAllService(user, result.date_last_sync, db),
16.    getAllUsers(user, result.date_last_sync, db),
17.    getAllUtilArm(user, result.date_last_sync, db),
18.  ]).then(() =>
19.    Promise.all([
20.      saveSyncProcess(user, deviceInfo).then((resultss: any) => {...}),
21.    ]),
22.  );
23. })
24. .catch(error => error);
```

Código 5 – Excerto sobre o processo de sincronização de dados

Após a sincronização ser concluída, será gravada a data de sincronização correspondente ao utilizador e identificador único do dispositivo (linha 20). Esta data e hora é obtida a partir do servidor, para evitar mudanças de fusos horários que por vezes acontecem nos dispositivos móveis.

Base de Dados Local

O SQLite é uma base de dados SQL de código aberto que guarda dados num ficheiro de texto no dispositivo móvel. É possível realizar todas as transações SQL do CRUD (*create, replace, update e delete*), este é especialmente utilizado no desenvolvimento móvel.

```
1. const insertNewArmazem = (armazem: any, desArm: any, db: any) =>
2.   new Promise((resolve, reject) =>
3.     db.transaction(
4.       (tx: any) => {
5.         tx.executeSql(
6.           'insert or replace into armazen (armazem, des_arm) values (?, ?)',
7.           [armazem, desArm],
8.           () => {
9.             resolve();
10.          },
11.        );
12.      },
13.      (error: string) => {
14.        ...
15.      },
16.    ),
17.  );
```

Código 6 – Inserção de dados na BD local

No Código 6 é apresentado um exemplo de operações na BD local do dispositivo móvel, nomeadamente a inserção dos novos dados (linha 6), ou dados alterados na tabela local. Desta forma é garantido o armazenamento local de dados, permitindo assim a seleção deste quando o dispositivo não tiver conexão Wi-Fi

Obter estado da conexão

O Redux é utilizado em conjunto com a *framework* React Native. O Redux simplifica a progresso de estados da aplicação quando existem múltiplos estados para controlar e muitos componentes que o necessitam, tirando a responsabilidade de cada componente de guardar o estado e passando para uma centralizada e única *store* (Maruta, 2018).

Para realizar o fluxo descrito, o Redux depende de três partes:

- *Store*, que armazena e centraliza o estado geral da aplicação. Este é imutável, ou seja, nunca se altera, apenas evolui. Podemos ter apenas uma *store* por aplicação, ou seja, ela é a “Única Fonte de Verdade” (*Single Source of Truth*). (Maruta, 2018)

- *Actions*, são fontes de informações que são enviadas da aplicação para a *store*. São disparadas pelas “*action creators*”, que são simples funções que, ao serem executadas, ativam os *reducers*. (Maruta, 2018)
- *Reducers*: recebem e tratam as informações para que sejam ou não enviadas à *store*.

O padrão Redux permite controlar o estado da conexão com a rede. Para atualizar o estado a partir de qualquer componente é necessário definir as ações, conforme no Código 7. Sendo a ação definida UPDATE_SYNC.

```

1. export const UPDATE_SYNC = 'UPDATE_SYNC';
2.
3. export const updateSync = (payload: any) => ({
4.   type: UPDATE_SYNC,
5.   payload
6. });

```

Código 7 – Redux: Implementação de *action*

Assim, a partir de qualquer componente podemos atualizar o estado da conexão à rede, e mesmo usá-lo para começar uma sincronização de dados.

No Código 8 é possível ver a implementação de um *reducer*, em que caso a *action* seja do tipo UPDATE_SYNC (linha 15-16), é atualizado o novo estado de conexão (linha 17).

```

1. export const defaultState: IStateSync = {
2.   isConnected: true,
3.   user: '',
4.   lastSync: '',
5.   deviceInfo: Constants.deviceId,
6.   hasPendingDocs: false,
7. };
8.
9. const updateSyncAction = (newState: IStateSync, payload: any): IStateSync => (
10.  {
11.    ...newState,
12.    ...payload,
13.  });
14. const syncReducer = (state = defaultState, action: any): IStateSync => {
15.   switch (action.type) {
16.     case UPDATE_SYNC:
17.       return updateSyncAction({ ...state }, action.payload);
18.   }
19.   return state;
20. };

```

Código 8 – Implementação do *reducer* para guardar estado da conectividade

Para saber o estado da conexão apenas é necessário utilizar a biblioteca NetInfo disponível pelo React Native. Esta biblioteca informa quando o dispositivo tem acesso à rede e a partir da chamada da ação `handleConnectionChange` (linha 15), a *store* é atualizada e guardado o novo estado (linha 9), conforme demonstrado no Código 9.

```
1. public componentWillMount() {
2.     NetInfo.isConnected.removeListener(
3.         'connectionChange',
4.         this._handleConnectionChange,
5.     );
6. }
7.
8. public _handleConnectionChange = (isConnected: boolean) => {
9.     reduxStore.dispatch(updateSync({ isConnected }));
10. }
11.
12. public async componentDidMount() {
13.     NetInfo.isConnected.addEventListener(
14.         'connectionChange',
15.         this._handleConnectionChange,
16.     );
17. }
```

Código 9 – NetInfo: Obtenção do estado de conectividade