

CTeSP em Programação de Sistemas de Informação
UC: Acesso Móvel a Sistemas de Informação (AMSI)
2º Ano – 1º Semestre
2020/2021

Relatório do Projeto

“SARAMAGO - Sistema de Gestão de Bibliotecas Académicas”

Elementos do Grupo:

Estudante 1 – André Machado @2180622

Estudante 2 – Gonçalo Rocha @2180659

TP1/PL2

Índice

1. INTRODUÇÃO	3
1.1. MOTIVAÇÃO E ENQUADRAMENTO.....	3
2. ARQUITETURA DA PLATAFORMA.....	4
2.1. ESPECIFICAÇÃO DA SOLUÇÃO EM ANDROID.....	4
3. METODOLOGIA E PLANEAMENTO.....	5
3.1. REQUISITOS FUNCIONAIS BASE (PROPOSTOS)	5
4. TECNOLOGIAS E FERRAMENTAS UTILIZADAS	6
5. TRABALHO REALIZADO	7
5.1. APLICAÇÃO SARAMAGO	7
5.1.1. ESTRUTURAÇÃO	7
5.1.1.1. ADAPTERS	7
5.1.2. LISTENERS	7
5.1.3. MODELS	7
5.1.4. UTILS	8
5.1.5. VIEWS.....	8
6. CONCLUSÃO	9
6.1. RESULTADOS	9
6.2. TRABALHO FUTURO	10

1. Introdução

As bibliotecas académicas têm por objetivo divulgar e permitir o livre acesso à produção científica criada pela comunidade académica, promovendo a integração, partilha e a visibilidade da informação e garantindo a preservação da sua memória intelectual.

Organizado de forma transparente, é armazenado num sistema de base de dados todo o fundo documental, independentemente da sua forma ou suporte, assim permitindo aos seus leitores a rápida consulta e requisição por meio de empréstimo.

Atualmente, as bibliotecas disponibilizam postos de trabalho aos leitores, como salas de salas de trabalho, gabinetes de investigação e até mesmo sala de reuniões, que permite uma procura autónoma.

Existem sistemas, como o Biblio.net, o *Evergreen*, o *PMB*, o *Koha* e o *Aleph* – estes dois últimos, *open-source* e proprietária, respetivamente, os melhores do mercado – que permitem a catalogação detalhada, a manutenção e o desenvolvimento de todo o acervo. Contudo, não possibilita muita das vezes a comunidade académica ser eficiente, tanto na sua gestão, como na sua procura.

Pretende-se implementar um sistema integrado de gestão de bibliotecas, um sistema de base de dados com interface web e ligação via API REST a uma aplicação *android* complementar, onde se faça a gestão de documentação, que permitirá, desde a fácil localização de exemplares de obras nas suas bibliotecas, sejam elas monográficas sejam analíticas, aos empréstimos efetuados pelos seus leitores, que se dividem entre diferentes estatutos personalizáveis, como por exemplo alunos, funcionários (docentes e não docentes), externos (leitores externos), etc.

Este sistema permite ao técnico bibliotecário rever algumas atitudes e posturas profissionais como forma de adaptar os serviços da biblioteca à nova realidade advinda com o desenvolvimento dos sistemas de informação.

1.1. Motivação e enquadramento

O trabalho apresentado neste documento foi elaborado no âmbito de projeto da unidade curricular de Acesso Móvel a Sistemas de Informação, para com a unidade curricular Projeto, de CTeSP em Programação de Sistemas de Informação, na Escola Superior de Tecnologia e Gestão (ESTG) pertencente ao Politécnico de Leiria.

Atualmente, com a presença de tecnologia sofisticada e a diversidade de dispositivos móveis, existe a possibilidade de simplificar as tarefas que envolvem o dia-a-dia do utilizador. Para um acesso mais pratico na plataforma, foi desenvolvida uma aplicação mobile *android* com ligação API REST á base de dados da mesma previamente concebida.

2. Arquitetura da Plataforma

A arquitetura da plataforma é constituída por diferentes componentes responsáveis pelo processo decorrido desde o preenchimento da base de dados até a visualização da informação por diferentes meios.

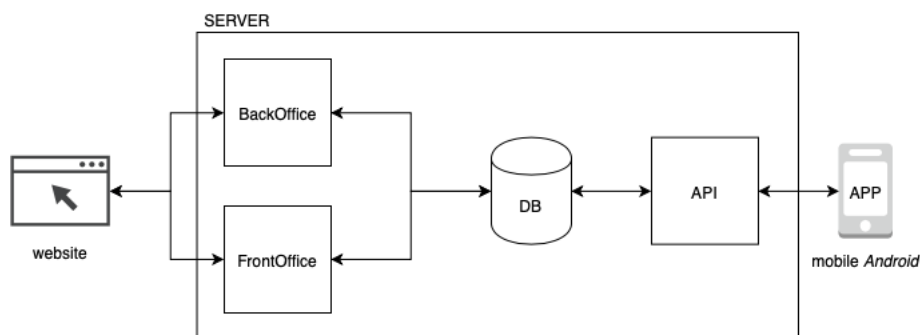


Figura 1 - Diagrama da arquitetura da plataforma

2.1. Especificação da solução em Android

A aplicação Android desenvolvida no contexto da unidade curricular, conecta-se ao SARAMAGO API, uma API REST, que pela qual é responsável pelas operações CRUD na base de dados MySQL.

Tais operações, permitem criar, visualizar, editar e apagar leitores e obras.

Tal como no navegador, o utilizador realiza pedidos HTTP via API. Após a receção do pedido, o servidor utiliza o mecanismo de rotas que associa o caminho do URL num método interno responsável por adquirir/tratar informação da base de dados.

3. Metodologia e planejamento

As metodologias de desenvolvimento de software têm como objetivo estruturar, planejar e controlar o processo de desenvolvimento e o ciclo de vida de um software.

As metodologias ágeis apresentam vários ciclos incrementais na realização do software, apresentando uma menor documentação e um maior foco no desenvolvimento do produto. É importante referir que proporciona uma comunicação mais informal entre os membros, havendo assim uma maior facilidade na estruturação da equipa. Para este trabalho, foi utilizada entre os membros da equipa o método *Ágil*.

3.1. Requisitos funcionais base (propostos)

Na tabela abaixo é apresentado os requisitos funcionais inicialmente propostos.

Módulos		Funcionalidades:	RF
	Menus/Opções		
Leitores	Listagem de Leitores	Visualização, inserção, edição e a eliminação de leitores	01
	Ficha do leitor	Visualização das informações do leitor	02
		Visualização de empréstimos do leitor	03
		Visualização reservas do leitor	04
		Visualização de pedidos reprográficos do leitor	05
	Catalogação	Visualização, inserção, edição e eliminação de obras	06
		Visualização, inserção, edição e eliminação de exemplares	07
Circulação e Empréstimos	Circulação	Visualização rápida dos empréstimos efetuados no dia	08
		Visualização rápida de empréstimos atrasados	09
		Consulta em tempo real	10
		Renovações	11
	Reservas	Inserção rápida de empréstimo a renovar	12
		Devoluções	13
		Fila de Espera	14
	Transferências	Aguarda recolha	15
		Visualização de exemplares a receber	16
Serviços Reprográficos	Visualização, inserção e aceitação de pedidos reprográficos		17
	Visualização de pedidos por levantar		18
Postos de Trabalho	Visualização de reservas diárias de postos de trabalho		19
	Visualização de reservas futuras de postos de trabalho		20
Em arrumação		Visualização e manipulação de exemplares arrumar	

Tabela 1 - Requisitos Funcionais

4. Tecnologias e ferramentas utilizadas

Neste capítulo serão descritas todas as tecnologias desenvolvimento e bibliotecas externas utilizadas no decorrer do projeto. Para o desenvolvimento da aplicação, foi utilizado o Android Studio e a Saramago API.

4.1. Android Studio

O Android Studio é conhecido como o IDE oficial de desenvolvimento nativo de aplicações Android, substituindo o Eclipse. Surgiu em 2013 e está disponível, gratuitamente, para os sistemas operativos Windows, macOS, Linux.

Este IDE ao início foi um pouco intimidante, mas depois de praticar algum tempo pudemos perceber que o seu estilo e tipo de organização eram bastante acessíveis ao utilizador.

4.2. SARAMAGO API

A SARAMAGO API é uma *webservice RESTful*, que tem como principal objetivo servir de suporte à aplicação mobile SARAMAGO.

A API neste projeto foi desenvolvida na linguagem de programação PHP com apoio da Yii2 *Fremework* que disponibilizam o recurso necessário á sua implementação e de uma base dados *MySQL*.

Todos os métodos relativos aos pedidos da API são do tipo GET, POST, PUT e DELETE e devolvem dados com o formato JavaScript Object Notation (JSON).

5. Trabalho Realizado

Este capítulo descreve o desenvolvimento da aplicação móvel SARAMAGO. Para além disso irá ser explicado como se procedeu à implementação da Saramago API e de que modo esta está relacionada com a aplicação. Serão também apresentadas as decisões tomadas no desenvolvimento destes projetos.

5.1. Aplicação SARAMAGO

Esta aplicação consome a informação da SARAMAGO API e transforma-a em interface gráfica para o utilizador. Assim, a API é a responsável pela troca de informações entre o servidor da aplicação Web e a base de dados.

Para o respetivo acesso foi implementado a biblioteca Volley, sendo uma biblioteca HTTP que facilita a criação de redes para apps para Android de maneira mais rápida.

5.1.1. Estruturação

Este capítulo descreve a estruturação da aplicação.

5.1.1.1. Adapters

- Catálogo
 - ListaObrasAdaptador
 - TabsObraAdaptador
- Leitores
 - ListaLeitoresAdaptador
 - TabsLeitorAdaptador
- ImageSlider_Details
- ListaAlugueresAdaptor

5.1.2. Listeners

- LeitoresListener
- LoginListener
- ConfigListener
- ObrasListener
- UserListener

5.1.3. Models

- Autor
- Biblioteca
- Cdu
- Config
- ConsultaTReal

- EstatutoExemplar
- Irregularidade
- Leitor
- Obra
- Obra_Autor
- PostoTrabalho
- Reprografia
- SaramagoDBHelper
- SingletonGestorBiblioteca
- TipoLeitor
- User

5.1.4. Utils

- CatalogoJsonParser
- ConfigJsonParser
- LeitoresJsonParser
- LoginJsonParser
- ObrasJsonParser
- UserJsonParser

5.1.5. Views

- Catalogo
 - Tabs:
 - FichaObraFragment
 - ListaExemplarFragment
 - AddObraActivity
 - DetalhesObraActivity
 - EditObraActivity
 - ListaObraActivity
- Leitores
 - Tabs:
 - EmprestimosLeitorFragment
 - FichaLeitorFragment
 - ReprografiaLeitorFragment
 - ReservasLeitorFragment
 - AddLeitorActivity
 - DetalhesLeitorActivity
 - EditLeitorActivity
 - ListaLeitoresActivity
- Dashboard
- LoginActivity
- MenuMainActivity

6. Conclusão

Com este projeto foi possível entender meramente como conceber uma aplicação Android. Contudo, é de referir que é preciso que os autores do projeto tivessem gerido mais o seu tempo na realização das tarefas definidas.

Os objetivos inicialmente definidos não foram cumpridos. Foi criada uma aplicação funcional à medida do utilizador que permitisse cumprir com os requisitos propostos. Durante a implementação foram realizados testes funcionais, de integração e de usabilidade.

6.1. Resultados

Alguns dos requisitos propostos foram implementados com sucesso, já outras não, de seguida é apresentada uma tabela com todas as funcionalidades e o seu estado final.

RF	Funcionalidades:	Estado Final
01	Visualização, inserção, edição e a eliminação de leitores	Implementado
02	Visualização das informações do leitor	Implementado
03	Visualização de empréstimos do leitor	Não implementado
04	Visualização reservas do leitor	Não implementado
05	Visualização de pedidos reprográficos do leitor	Não implementado
06	Visualização, inserção, edição e eliminação de obras	Implementado
07	Visualização, inserção, edição e eliminação de exemplares	Não implementado
08	Visualização rápida dos empréstimos efetuados no dia	Não implementado
09	Visualização rápida de empréstimos atrasados	Não implementado
10	Visualização, inserção e eliminação de consultas em tempo real de fundo especiais	Não implementado
11	Inserção rápida de empréstimo a renovar	Não implementado
12	Visualização rápida das devoluções efetuadas no dia	Não implementado
13	Visualização e cancelamento de reservas em fila de espera	Não implementado
14	Visualização e cancelamento de reservas que aguardam recolha	Não implementado
15	Visualização de exemplares a receber	Não implementado
16	Visualização, inserção e aceitação de pedidos reprográficos	Não implementado
17	Visualização de pedidos por levantar	Não implementado
18	Visualização de reservas diárias de postos de trabalho	Não implementado
19	Visualização de reservas futuras de postos de trabalho	Não implementado
20	Visualização e manipulação de exemplares arrumar	Não implementado

Tabela 2 - Requisitos Finais (implementados)

6.2. Trabalho Futuro

Apesar das falhas relatadas e esta etapa do projeto meramente concluída, existe funcionalidades que poderiam ter sido concebidas e/ou melhoradas. É importante e necessário que exista uma evolução e otimização da aplicação. Para a aplicação, é apresentado como trabalho futuro:

- Melhor implementação no uso da API externa;
- Implementação de mais funcionalidades que possam auxiliar o utilizador à utilização da aplicação;