|  |  |  |  |
| --- | --- | --- | --- |
| simbolo_vertical_sem_fundo_cores |  | | |
| **TeSP - PSI** | **2º Ano - 1º Semestre** | **Ano letivo: 2020/21** |
| **Projeto de Sistemas de Informação**  **Data: 26 de Outubro de 2020** | | |

|  |
| --- |
| **<nome do sistema>** |

**Relatório de**

**Especificação do Sistema de Informação**

|  |  |
| --- | --- |
| **Grupo**: PL2-G04 | |
| **Nº 2180622** | **André Filipe Andrade Machado** |
| **Nº 2180659** | **Gonçalo Bertão Coelho da Rocha** |
| **Nº 2180696** | **Rui Nuno de Almeida Pereira** |

**Índice**

[1 Definição da Lógica de Negócio 6](#_Toc52545112)

[2 Análise de Impacto 6](#_Toc52545113)

[2.1 Análise Concorrencial 7](#_Toc52545114)

[2.1.1 Aleph 7](#_Toc52545115)

[2.1.2 Bibliosoft 8](#_Toc52545116)

[2.1.3 Koha 9](#_Toc52545117)

[2.1.4 Comparação dos Sistemas 10](#_Toc52545118)

[3 Requisitos do Sistema 11](#_Toc52545119)

[4 User Stories 11](#_Toc52545120)

[5 Use Cases 13](#_Toc52545121)

[6 Diagrama de Classes e Modelo de Dados 14](#_Toc52545122)

[7 Wireframes/Mockups 15](#_Toc52545123)

**Índice de figuras**

[Figura 1 - Screenshot de Aleph 7](#_Toc52545201)

[Figura 2 - Screenshot de bibliosoft 9](#_Toc52545202)

[Figura 3 - Screenshot de Koha 10](#_Toc52545203)

[Figura 4 – Use Case main diagram 13](#_Toc52545204)

[Figura 5 – Diagrama de classes do projeto 14](#_Toc52545205)

**Índice de tabelas**

[Tabela 1 – Comparação entre os sistemas concorrenciais 6](#_Toc51941862)

[Tabela 2 – Requisitos Não Funcionais 7](#_Toc51941863)

[Tabela 3 – User Stories e respetivos critérios de aceitação 8](#_Toc51941864)

# Definição da Lógica de Negócio

<Contextualizar o SI a implementar, temática do projeto, indicando objetivos gerais, descrição sumária das necessidades do utilizador. Deve ficar-se com uma ideia clara do âmbito do projeto. >

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 Alexandria, o Pergamum, o Mandarin, 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 uma base de dados com interface web e complementar onde se faça a gestão de centros de documentação, que permitirá, desde a fácil localização de exemplares de obras nas suas sub-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. Contará com gestão de postos de trabalho personalizáveis, a titulo de exemplo como salas de trabalho em grupo, gabinetes de investigação individual, etc. Esta base de dados ainda contará com algumas funcionalidades padronizadas da indústria, como a Classificação Decimal Universal (CDU), que permite a uniformização dos critérios de classificação de documentos ou o controlo de autoridade que permite e

Este sistema permitirá 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

# Análise de Impacto

<Impactos positivos e negativos do sistema a desenvolver>

O nosso sistema apresenta diversas características únicas direcionadas ao apoio dos funcionários. A utilização da aplicação móvel irá facilitar consideravelmente o trabalho de cada funcionário e a ajuda mutua entre os mesmos, devido ao sistema de notificações que será implementado, assim como o acesso básico aos detalhes de cada leitor, porém, a nossa aplicação sendo apenas destinada a esses mesmos funcionários, poderá trazer complicações a um leitor que não possua um computador ou um *tablet* e que se tenha de deslocar à biblioteca, para reservar ou renovar um livro, requisitar um posto de trabalho ou utilizar um serviço de reprografia.

## Análise Concorrencial

### Aleph

<Nome, site (*screenshot*), descrição detalhada, vantagens, desvantagens, o que falta>



Figura 1 - Screenshot de Aleph

O Aleph (figura 1) é um sistema integrado de bibliotecas. Pode ser utilizado em empresas de qualquer dimensão e fornece bibliotecas de pesquisa e académicas eficientes, interfaces gráficas intuitivas e *User-Friendly* que são simples de manusear. O sistema por si também é descrito como escalável, ou seja, é possível ao cliente recriar ou corrigir o seu ambiente de trabalho.

Aleph assegura uma medida *future proof* para que as estruturas de cliente/servidor garantam as necessidades ao cliente no futuro.

### Bibliosoft

<Nome, site (*screenshot*), descrição detalhada, vantagens, desvantagens, o que falta>



Figura 2 - Screenshot de bibliosoft

Bibliosoft é uma empresa portuguesa que desenvolve produtos para gestão de bibliotecas, consistindo no “apoio técnico aos produtos que desenvolve e distribui”, “conversão de catálogos manuais”, “organização e produção de bibliografias” e “conversão de bases de dados para formatos normalizados”.

O seu produto biblio.NET tem um sistema normalizado cuja estrutura está de acordo com o formato UNIMARC. Tem obviamente um sistema *web based* centralizado num único servidor. A biblio.NET permite na mesma instalação a existência de várias bibliotecas com bases de dados, todas elas com a possibilidade de incluir múltiplos formatos.

### Koha

<Nome, site (*screenshot*), descrição detalhada, vantagens, desvantagens, o que falta>



Figura 3 - Screenshot de Koha

O Koha é um software de gestão integrada de bibliotecas que gere os serviços da biblioteca quer sejam administrativos ou interagindo com os leitores da mesma. Serviços tais como a catalogação, gestão de autoridades, gestão de periódicos, aquisições, relatórios e estatísticas, etc, são acessíveis através do OPAC. Sendo um software baseado em tecnologias *web* é também compatível com os variados formatos tais como UNIMARC, MARC21, Z39.50, MARCXML, ISO2709, SRU/SRW, SIP2, RSS, etc.

O Koha não possui custos de licenciamento, pois funciona sobre um sistema operativo open-source.

### Comparação dos Sistemas

<De seguida...>

Tabela 1 – Comparação entre os sistemas concorrenciais

|  |  |  |  |
| --- | --- | --- | --- |
| Funcionalidades | Aleph | Bibliosoft | Koha |
| Sistema Próprio | x | x | x |
| OPAC | x | x | x |
| Future Proof | x | - | - |
| Func. 4 |  |  |  |
| ... |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Requisitos do Sistema

## Requisitos Funcionais

O sistema irá conter uma base de dados onde irá armazenar todos os carros do stand, os mesmo podem ser para venda ou para alugar. Ainda terá uma oficina onde os clientes poderão levar os seus carros para serem reparados.

<Requisitos funcionais base, requisitos funcionais extra, requisitos não funcionais (segurança, usabilidade, desenvolvimento, interoperabilidade, portabilidade, ...)>

Tabela 2 – Requisitos Não Funcionais

|  |  |
| --- | --- |
| Requisito | Prioridade |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# User Stories

<Devem ser especificados os requisitos funcionais do ponto de vista do utilizador sob forma de User Stories As issues devem ser estimadas em Story Points utilizando a sequência de Fibonacci: 1, 2, 3, 5, 8, 13, 20, 40 (máx).>

“**Quem**, precisa de fazer **O quê** e **Porquê**”

* Quem – é o utilizador (que pode ser humano ou um sistema)
* O quê – quais os ados que são tratados e necessários
* Porquê – segue o “so that” numa user story.

Focar nestes 3 e evitar o Como (excluir aspetos de desenho, interface design statements).

“Como [persona],” – Para quem estamos a construir isto? Não estamos atrás de um título de trabalho, estamos atrás da persona da pessoa. Entendemos como a pessoa funciona, como pensa e como se sente.

“eu [quero/gostaria que],” – Aqui descrevemos o intuito, mas não as funcionalidades que eles usam. O que é que eles estão a tentar atingir? Deve ser livre da implementação – se estivermos a descrever algo específico da UI, e não o objetivo do utilizador, estamos a perder o objetivo.

“[para que].” – como é que o desejo imediato deles para fazer algo se integra na big picture? Qual o objetivo geral que estão a tentar atingir? Qual o principal problema que precisa de ser resolvido?

Tabela 3 – User Stories e respetivos critérios de aceitação

|  |  |
| --- | --- |
|  |  |
| Título:  US1 – Efetuar lançamento de dados | **Story Points**:  3 |
| Descrição: Como jogador quero ser capaz de lançar os dados de forma a poder fazer a jogada  Critérios de Aceitação:   * Só é possível efetuar uma jogada de cada vez * As faces dos dados têm de ser visíveis após o lançamento | |
| Título:  US2 – Adicionar cliente | **Story Points**:  5 |
| Descrição: Como utilizador quero poder adicionar um cliente de modo a ficar registado  Critérios de Aceitação:   * Os campos nome, morada, contato e NIF têm de ser obrigatoriamente preenchidos * Não pode haver um cliente com o mesmo NIF | |
| Título:  US | **Story Points**: |
| Descrição:  Critérios de Aceitação: | |
| Título:  US | **Story Points**: |
| Descrição:  Critérios de Aceitação: | |
| Título:  US | **Story Points**: |
| Descrição:  Critérios de Aceitação: | |
|  |  |
|  |  |

# Use Cases

<Devem ser especificados os requisitos funcionais sob a forma de casos de uso. Determinação das funções específicas que o sistema realiza e dos dados sobre as quais as funções operam. >

[Use Cases diagram(s)]

Figura 4 – Use Case main diagram

|  |  |  |  |
| --- | --- | --- | --- |
| Title: | *[enter the goal of the UC – short, active verb phrase]* | | |
| ID: | UC-00X | **Priority:** | *[Low, Medium, High]* |
| Actor: | *[a person or a software/hardware system that interacts with your system to achieve the goal of this UC]* | | |
| Preconditions: | *[describe the state the system is in before the first event in this use case]* | | |
| Postconditions: | *[describe the state the system is in after all the events in this UC have taken place]* | | |
| Main Success Scenario: | *[describe the flow of events from preconditions to postconditions, when nothing goes wrong. This is the meat of the use case]*   1. ... 2. ... 3. ... | | |
| Extensions: | *[describe all other scenarios for this use case – including exceptions and error cases]* | | |

# Diagrama de Classes e Modelo de Dados

<Diagrama de classes/modelo de dados do projeto e respetiva explicação>

A close up of a map

Description automatically generated

Figura 5 – Diagrama de classes do projeto

# Wireframes/Mockups

<Wireframes/Mockups desenvolvidos para auxiliar o desenvolvimento da interface>