|  |  |  |  |
| --- | --- | --- | --- |
| 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**: PL1/2-GX | |
| **Nº 2180601** | Jason Mendes |
| **Nº 2180602** | Tiago Antunes |
| **Nº 2180689** | Tiago Ramos |

**Índice**

[1 Definição da Lógica de Negócio 6](#_Toc51941922)

[2 Análise de Impacto 6](#_Toc51941923)

[2.1 Análise Concorrencial 6](#_Toc51941924)

[2.1.1 <Sistema 1> 6](#_Toc51941925)

[2.1.2 <Sistema 2> 6](#_Toc51941926)

[2.1.3 <Sistema *n*> 6](#_Toc51941927)

[2.1.4 Comparação dos Sistemas 6](#_Toc51941928)

[3 Requisitos do Sistema 7](#_Toc51941929)

[4 User Stories 7](#_Toc51941930)

[5 Use Cases 9](#_Toc51941931)

[6 Diagrama de Classes e Modelo de Dados 10](#_Toc51941932)

[7 Wireframes/Mockups 11](#_Toc51941933)

**Índice de figuras**

[Figura 1 – Use Case main diagram 9](#_Toc51941841)

[Figura 2 – Diagrama de classes do projeto 11](#_Toc51941842)

**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. >

A aplicação apresentada será uma aplicação terá como âmbito principal o lado do Fitness mas também terá o lado de Saúde e Bem Estar. Será uma aplicação para pessoas mais viradas para essa área que necessitem de motivação ou que precisem de planos de treino com ou sem acompanhamento ou que precisem de planos de nutrição dependendo do objetivo do cliente.

Será somente para clientes inscritos de um ginásio “fictício” com uma subscrição mensal com vários contratos como por exemplo se é com fidelização ou não tendo toda a gestão da mesma no Backend.

Teremos 3 cargos administrativos que serão o ‘*Admin’* que gerirá todos os utilizadores registados na plataforma e de fazer alterações quer no Backend Quer no Frontend. O ‘*Personal Trainer’* terá a função de poder criar planos de treino caso o cliente queira ou não, aumentando o valor da subscrição mensal. Não será obrigatório, poderá ser com acompanhamento durante o treino atribuindo um ‘*PT’ a um* ‘*Cliente*’. Teremos também o ‘*Nutricionista*’ que terá como objetivo de gerir planos nutricionais que incluirá o tipo de alimento que o cliente deverá comer, a que horas, dependendo do que o cliente quererá alcançar estando no ginásio, se é por questões de saúde etc. Terão acesso também em gerir uma tabela de alimentos com seus carboidratos para o cliente ter acesso completamente grátis na aplicação. Depois teremos um cargo de ‘*Cliente’* que são os utilizadores que se registarem no website e que façam a subscrição do mesmo dependendo do que quer incluir no seu orçamento ou não, podendo também mudar o mesmo a qualquer momento. Terá acesso aos seus planos de treino e aos seus planos nutricionais (se tiver). Terá acesso também a páginas informativas sobre a mesma área. O cliente pode ter um Nutricionista e não ter um *Personal Trainer*, ou vice-versa.

A app móvel será algo mais prático para se usar pré-treino, durante ou até mesmo pós-treino podendo consultar o seu plano de treino, como fazer os exercícios tendo imagens ilustrativas e tendo também o número de cada máquina para o cliente concluir o seu plano de treino com mais facilidade.

Na aplicação web terá no Backend a gestão da aplicação toda em geral, podendo fazer alteração a todos os cargos, adicionar e remover funcionários, alteração de orçamentos, etc… Os PT’s poderão gerir os seus planos de treino lá, os Nutricionistas poderão também gerir os planos nutricionais. Na parte do Frontend o cliente terá acesso também aos seus planos de treino e nutricionais (se tiver), ter acesso a calendários de aulas de grupo e horário dos mesmos podendo fazer pré-inscrição do mesmo. Poderá ver também a lista de alimentos e seus Carboidratos, páginas informativas grátis sobre o que é melhor, suplementação etc… e a lista de exercícios é explicada mais detalhadamente.

# Análise de Impacto

<Impactos positivos e negativos do sistema a desenvolver>

## Análise Concorrencial

### <Sistema 1>

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

### <Sistema 2>

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

### <Sistema *n*>

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

### Comparação dos Sistemas

<De seguida...>

Tabela 1 – Comparação entre os sistemas concorrenciais

|  |  |  |  |
| --- | --- | --- | --- |
| Funcionalidades | Sistema 1 | Sistema 2 | Sistema 3 |
| Func. 1 | x | - | x |
| Func. 2 |  |  |  |
| Func. 3 |  |  |  |
| Func. 4 |  |  |  |
| ... |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Requisitos do Sistema

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

Tabela 2 – Requisitos Não Funcionais

|  |  |
| --- | --- |
| Requisitos Funcionais - Backend | Prioridade |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Requisitos Funcionais - Frontend | Prioridade |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Requisitos não funcionais | Prioridade |
| 1. (Segurança) Proteção contra SQL Injection. | 10 |
| 1. Um sistema de encriptar todas as passwords através do algoritmo sha256. | 10 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# 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 1 – 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]* | | |
| Story Points: |  | | |

# 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 2 – Diagrama de classes do projeto

# Wireframes/Mockups

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