BullkApp

Especificação de Requisitos de Software

Versão <1.0>

Histórico da Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 18/03/2023 | 1.0 | Estruturação do Documento | Leonardo Almeida |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Índice

[1. Introdução 4](#_Toc130069429)

[1.1 Objetivo 4](#_Toc130069430)

[1.2 Escopo 4](#_Toc130069431)

[1.3 Definições, Acrônimos e Abreviações 4](#_Toc130069432)

[1.4 Referências 4](#_Toc130069433)

[1.5 Visão Geral 4](#_Toc130069434)

[2. Descrição Geral 4](#_Toc130069435)

[3. Requisitos Específicos 5](#_Toc130069436)

[3.1 Funcionalidade 5](#_Toc130069437)

[3.1.1 <Requisito Funcional Um> 6](#_Toc130069438)

[3.2 Utilidade 6](#_Toc130069439)

[3.2.1 <Requisito de Utilidade Um> 7](#_Toc130069440)

[3.3 Confiabilidade 7](#_Toc130069441)

[3.3.1 <Requisito de Confiabilidade Um> 7](#_Toc130069442)

[3.4 Desempenho 7](#_Toc130069443)

[3.4.1 <Requisito de Desempenho Um> 8](#_Toc130069444)

[3.5 Suportabilidade 8](#_Toc130069445)

[3.5.1 <Requisito de Suportabilidade Um> 8](#_Toc130069446)

[3.6 Restrições de Design 9](#_Toc130069447)

[3.6.1 <Restrição de Design Um> 9](#_Toc130069448)

[3.7 Documentação do Usuário On-line e Requisitos do Sistema de Ajuda 9](#_Toc130069449)

[3.8 Componentes Comprados 9](#_Toc130069450)

[3.9 Interfaces 9](#_Toc130069451)

[3.9.1 Interfaces com o Usuário 9](#_Toc130069452)

[3.9.2 Interfaces de Hardware 11](#_Toc130069453)

[3.9.3 Interfaces de Software 11](#_Toc130069454)

[3.9.4 Interfaces de Comunicações 12](#_Toc130069455)

[3.10 Padrões Aplicáveis 12](#_Toc130069456)

Especificação de Requisitos de Software

# Introdução

## Objetivo

Esta documentação de Especificação de Requisito de Software tem como finalidade, descrever a estruturação e comportamento do software a ser desenvolvido, intitulado como BullkApp.

A partir dela, será possível identificar os pontos de atenção da aplicação e as necessidades que precisam ser supridas, esses pontos serão elencados através da sessão de Requisitos.

## Escopo

A aplicação abordada neste documento, tem como objetivo permitir uma melhor experiência para os frequentadores de academias de musculação. Com a utilização do software, os alunos poderão acompanhar sua evolução e demonstração do treino.  
O BullkApp será uma aplicação multiplataforma sendo dividido em dois segmentos. Usuário aluno, que terá uma aplicação mobile na qual permitirá acessar as funções destacadas anteriormente e a área para administradores/professores, os quais serão responsáveis por lançarem os treinos e toda a parte de administração do App.

## Definições, Acrônimos e Abreviações

CRUD – Esse termo é utilizado para referenciar as operações básicas de uma aplicação, C - Create, R -Read, U - Update e D - Delete. Ou seja, criar, ler, atualizar e apagar, respectivamente.

## Referências

*[Esta subseção fornece uma lista completa de todos os documentos mencionados em outra parte na* ***SRS****. Identifique cada documento por título, número do relatório se aplicável, data e organização da publicação. Especifique as origens a partir das quais as referências podem ser obtidas. Essas informações podem ser fornecidas por um anexo ou outro documento.]*

## Visão Geral

*[Esta subseção descreve o que o restante da* ***SRS*** *contém e explica como o documento é organizado.]*

# Descrição Geral

O BullkApp será desenvolvido para suprir a demanda de aplicações voltadas para o público de academias de musculação. Nos cenários analisados, percebeu que quando há a utilização de aplicativos para intermediar essa rotina de treinos, eles não proporcionam uma boa experiência para o aluno. O método comumente utilizado é a impressão dos treinos em papel, o que é um problema, porque durante o treino o aluno acaba suando e inevitavelmente acaba lhe danificando, assim será necessário realizar outra impressão, resultando em um consumo de papel desnecessário.

O aplicativo apresentará dois tipos de usuários, o aluno, que terá condições de consultar seus treinos no padrão ABC, verificando suas descrições (peso, repetições e tempo de descanso) e vídeo de demonstração de cada exercício; solicitar suas avaliações já realizadas anteriormente; Dica de substituição de exercício para quando o aparelho listado no treino já se encontra em uso. E o outro tipo de usuário é o professor/administrador, que será o responsável pelo gerenciamento de treinos (lançamento, atualizações e disponibilização das avaliações físicas) e configurações do aplicativo.

O BullkApp será uma aplicação multiplataforma, para dispositivos mobiles e Web, o acesso a cada plataforma será respeitado as premissas do tipo de usuário, aluno acessará via aplicação mobile e professor/administrador através da Web.

# Requisitos Específicos

## Funcionalidade

*[Esta seção descreve os requisitos funcionais do sistema para os requisitos que são expressos no estilo do idioma nativo. Para muitos aplicativos, isso pode constituir o bulk do pacote* ***SRS*** *e um conceito deve ser fornecido à organização desta seção. Esta seção é, geralmente, organizada por recurso, mas métodos de organização alternativos também podem ser apropriados; por exemplo, organização por usuário ou organização por subsistema. Requisitos funcionais podem incluir conjuntos, recursos e segurança do recurso.*

*Onde as ferramentas de desenvolvimento de aplicativo, como ferramentas de requisitos, ferramentas de modelagem e similares, são empregadas para capturar a funcionalidade, esta seção do documento pode se referir à disponibilidade desses dados, indicando o local e o nome da ferramenta utilizada para capturar os dados.*

*Auditoria: Existe a necessidade de rastrear quem usou o sistema e quando foi usado? Declare requisitos para fornecer trilhas de auditoria quando da execução do sistema.*

*Autenticação: O acesso ao sistema será controlado? Declare requisitos de autenticação.*

*Licenciamento: O sistema ou partes do sistema serão licenciados? Caso seja usado algum software de código livre no sistema, todos os acordos de código livre foram respeitados? Declare requisitos para adquirir, instalar, rastrear e monitorar licenças.*

*Impressão: A capacidade de impressão será necessária? Declare requisitos para impressão.*

*Relatórios: A capacidade de geração de informes será necessária? Declare requisitos para Relatórios.*

*Agendamento: A execução de alguma ação no sistema necessita ser agendada? Declare requisitos para capacidade de agendamento.*

*Segurança: Os elementos ou os dados do sistema necessitam estar seguros? Declare requisitos de proteção de acesso para determinados recursos ou informações.*

*]*

### **RQF001 – Manter Usuário:**

O Sistema deverá suportar as operações CRUD de Usuários. Que por sua vez, possibilitará fazer a distinção entre os usuários do tipo:

- Aluno: Este usuário terá condições de consultar seus treinos, assistir os vídeos de demonstração e acessar suas avaliações já realizadas;

- Professor/Administrador: Este usuário será o responsável por lançar os treinos dos alunos e suas avaliações;

Para realizar o cadastro dos usuários, será necessário apresentar as seguintes informações:

- Nome;

- Idade;

- Sexo;

- Telefone;

- E-mail;

- URL Foto/Avatar;

### **RQF002 – Manter Treino:**

O Sistema deverá suportar as operações CRUD de Treino. Esse requisito permitirá que o professor/administrador faça a definição dos treinos para que os alunos possam seguir nas suas rotinas;

### **RQF003 – Manter Exercício:**

O Sistema deverá suportar as operações CRUD de Exercícios.

### **RQF004 – Manter Aparelho:**

O Sistema deverá suportar as operações CRUD deAparelho.

## Utilidade

*[Esta seção inclui todos os requisitos que afetam a utilidade. Por exemplo,*

* *especifique o tempo de treinamento requerido para um usuário normal e um usuário potente para que se torne produtivo em operações particulares*
* *especifique tempos de tarefa mensuráveis para tarefas típicas ou baseie os novos requisitos de utilidade do sistema em outros sistemas que os usuários conheçam e gostem*
* *especifique requisito para conformidade com padrões de utilidade comuns, como padrões GUI da Microsoft padrões CUA da IBM*

*Os requisitos de usabilidade são críticos para o sucesso de qualquer sistema. Infelizmente, os requisitos de usabilidade são normalmente os mais mal especificados. Considere este simples requisito: O sistema deve ser fácil de usar. Ele não ajuda muito, porque não pode ser verificado.*

*Ao capturar requisitos de usabilidade, é uma boa idéia identificar primeiro as questões e preocupações, e então refiná-las em requisitos verificáveis posteriormente. De acordo com uma definição tradicional, a usabilidade consiste de cinco fatores:*

*Facilidade de Aprendizagem: Um usuário com um nível especificado de experiência deve aprende como usar o sistema em um determinado prazo especificado.*

*Eficiência da Tarefa: Um usuário deve poder terminar uma determinada tarefa em um prazo especificado (ou em uma quantidade de cliques do mouse).*

*Facilidade de Recordação: Um usuário deve poder recordar como se realizam determinadas tarefas, após um prazo especificado de não utilização do sistema.*

*Entendimento: Um usuário deve entender as mensagens e os alertas do sistema e o que o sistema faz.*

*Satisfação Subjetiva: Uma porcentagem especificada da comunidade de usuários deve expressar a satisfação de usar o sistema.*

*Você pode querer usar o seguinte método para identificar e especificar requisitos de usabilidade:*

*Identifique as principais questões de usabilidade observando as tarefas críticas, perfis de usuário, metas do sistema e problemas prévios de usabilidade.*

*Escolha um estilo apropriado para expressar os requisitos:*

*Estilo de Desempenho: Especifica a velocidade que os usuários podem aprender várias tarefas e a velocidade que eles podem executar as tarefas após treinamento.*

*Estilo de Defeito: Melhor do que medir os tempos da tarefa, identifique os defeitos de usabilidade e especifique a freqüência com que eles ocorrem.*

*Estilo de Diretriz: Especifica a aparência geral e o tempo de resposta da interface de usuário pela referência a um padrão aceito e bem definido.*

*Escreva requisitos reais, incluindo critérios de desempenho.]*

### <Requisito de Utilidade Um>

*[A descrição do requisito aparece aqui.]*

## Confiabilidade

*[Os requisitos de confiabilidade do sistema devem ser especificados aqui. Algumas sugestões são:*

* *Disponibilidade—especifique a porcentagem de tempo disponível ( xx.xx%), as horas de utilização, o acesso de manutenção, as operações de modo degradado e assim por diante.*
* *MTBF (Mean Time Between Failures) — este é, geralmente, especificado em horas, mas pode também ser especificado em dias, meses ou anos.*
* *MTTR (Mean Time To Repair)—por quanto tempo o sistema tem permissão para ficar fora de operação após ter falhado?*
* *Exatidão—especifica a precisão (resolução) e a exatidão (por algum padrão conhecido) requeridas na saída do sistema.*
* *Taxa Máxima de Erros ou Defeitos—geralmente expressa em termos de erros por mil linhas de código (erros/KLOC) ou erros por ponto de função (erros/ponto de função).*
* *Taxa de Erros ou Defeitos—categorizada em termos de erros menores, significativos e críticos: o(s) requisito(s) deve(m) definir o que quer dizer um erro “crítico”; por exemplo, perda completa de dados ou uma inabilidade completa para utilizar determinadas partes da funcionalidade do sistema.*

*Aqui devem ser registrados requisitos de disponibilidade do software(SLA), armazenamento de dados, segurança]*

### <Requisito de Confiabilidade Um>

*[A descrição do requisito.]*

## Desempenho

*[As características de desempenho do sistema são esboçadas nesta seção. Inclua tempos de resposta específicos. Onde aplicável, faça referência a Casos de Uso relacionados por nome.*

* *Tempo de Resposta Especifica a quantidade de tempo disponível para o sistema para terminar tarefas e transações especificadas (médio, máximo). Use unidades de medida. Exemplos:*

*Toda interface entre um usuário e o sistema deverá terá um tempo de resposta máximo de 2 segundos.*

*O produto deve carregar os novos parâmetros de status após 5 minutos de uma mudança.*

* *Taxa de Transferência: Especifica a capacidade do sistema de suportar um determinado fluxo de informações (por exemplo, transações por segundo).*
* *Capacidade: Especifica os volumes que o produto deve tratar e as quantidades de dados armazenados pelo produto. Certifique-se que a descrição do requisito é quantificável, podendo assim ser testado. Use uma unidade de medida tal como: a quantidade de clientes ou transações que o sistema pode acomodar, uso dos recursos (memória, disco, etc) ou modos de degradação (qual é o modo de operação aceitável quando o sistema estiver degradado de alguma forma) Exemplos:*

*O produto pode atender a 300 usuários simultâneos no período de 9:00h as 11:00h.*

*A carga máxima em outros períodos será de 150.*

* *Partida: O tempo necessário para o sistema entrar em funcionamento.*
* *Parada: O tempo necessário para o sistema parar de funcionar.*

*.*

### <Requisito de Desempenho Um>

*[A descrição do requisito aparece aqui.]*

## Suportabilidade

*[Esta seção indica os requisitos que aprimorarão a suportabilidade ou a capacidade de manutenção do sistema que está sendo construído, incluindo padrões de codificação, convenções de nomenclatura, bibliotecas de classe, acesso de manutenção e utilitários de manutenção. Adaptabilidade: Existe algum requisito especial que considere a adaptação do software (incluindo atualizações)? Liste os requisitos para facilidade com que o sistema se adapte a novos ambientes.*

*Compatibilidade: Existe algum requisito que considere este sistema e sua compatibilidade com versões anteriores deste sistema ou de sistemas legados que fornecem a mesma capacidade?*

*Configurabilidade: o produto será configurado após ter sido implantado? De que forma o sistema será configurado?*

*Instalação: Declare qualquer requisito especial a respeito da instalação do sistema.*

*Nível de Suporte: Qual é o nível de suporte que o produto necessita? Isto é feito normalmente usando um "Help-desk". Se for necessário existirem pessoas que forneçam suporte ao produto, esse suporte é considerado como parte do que você está fornecendo ao cliente? Existe algum requisito para esse suporte? Você pôde também construir o suporte no próprio produto, neste caso este é o lugar para escrever esses requisitos. Considere o nível de suporte que você deseja fornecer e de que forma ele pode ser obtido.*

*Manutenção: Existe algum requisito especial que considere a manutenção do sistema? Quais são os requisitos para o ciclo de liberação desejado para o produto e a forma que a liberação irá acontecer? Quantifique o tempo necessário para fazer as mudanças especificadas para o produto. Podem também existir requisitos especiais para a manutenção, tal como um requisito onde o produto deva ser mantido por seus usuários finais ou desenvolvedores que não são da sua equipe de desenvolvimento. Isto tem um efeito na forma com que o produto é desenvolvido, e podem existir requisitos adicionais para documentação ou treinamento. Descreva o tipo de manutenção e a quantidade de esforço necessária. Exemplos:*

*Uma nova estação de tempo deve ser adicionada ao sistema durante a noite.*

*As liberações de manutenção serão oferecidas aos usuários finais uma vez ao ano.*

*Escalabilidade: Qual o volume de usuários e dados o sistema irá suportar? Especifica o crescimento previsto que o produto deve suportar à medida que os negócios cresçam (ou que se espera que cresçam), os produtos de software devem aumentar suas capacidades para lidar com novos volumes. Isto pode ser expresso como uma tendência no tempo.*

*Testabilidade: Existe algum requisito especial a respeito da testabilidade do sistema?]*

### <Requisito de Suportabilidade Um>

*[A descrição do requisito aparece aqui.]*

## Restrições de Design

*[Esta seção indica as restrições de design no sistema que está sendo construído. As restrições de design representam decisões de design que foram obrigatórias e às quais deve-se aderir. Os exemplos incluem linguagens de software, requisitos de processo de software, utilização prescrita de ferramentas de desenvolvimento, restrições de arquitetura e design, componentes comprados, bibliotecas de classes e assim por diante.*

*Deve-se utilizar para adicionar requisitos relativos a arquitetura da aplicação que será desenvolvida, processo de software que será utilizado, linguagens de programação que serão utilizadas, requisitos de design da aplicação, ferramentas e ides de desenvolvimento*

*Restrições de Design: Existe alguma decisão de design obrigatória que o produto tenha que aderir?*

*Componentes de Terceiros: Especifica qualquer legado, COTS ou componentes de código livre que tenha sido exigido seu uso com o sistema.*

*Linguagens de implementação: Especifica os requisitos sobre as linguagens de implementação a serem usadas*

*Suporte a Plataformas: Especifica os requisitos sobre as plataformas que o sistema suportará*

*limites de Recursos: Especifica os requisitos sobre o uso de recursos de sistema, tais como memória e espaço de disco rígido*

*Restrições Físicas: Especifica requisitos sobre forma, tamanho e peso do hardware para abrigar o sistema.]*

### <Restrição de Design Um>

*[A descrição do requisito aparece aqui.]*

## Documentação do Usuário On-line e Requisitos do Sistema de Ajuda

*[Descreve os requisitos, se houver, para a documentação do usuário on-line, sistemas de ajuda, ajuda sobre observações e assim por diante.]*

## Componentes Comprados

*[Esta seção descreve os componentes comprados a serem utilizados com o sistema, as restrições aplicáveis de licença ou uso e os padrões associados de compatibilidade, interoperabilidade ou interface.]*

## Interfaces

*[Esta seção define as interfaces que devem ser suportadas pelo aplicativo. Ela deve conter especificidade adequada, protocolos, portas, endereços lógicos e similares para que o software possa ser desenvolvido e verificado em comparação com os requisitos da interface.]*

### Interfaces com o Usuário

*[Descreva as interfaces com o usuário que devem ser implementadas pelo software.*

*Nesta seção devem ser descritas todas as interfaces que serão criadas com o usuário, exemplo:*

*“Para o usuário será criado, as telas de cadastro de cliente, software, prioridade, cliente, usuário e chamado, e a tela de gerenciamento de chamados.”*

*Ou deve se adotar o padrão abaixo:*

|  |  |  |
| --- | --- | --- |
| *Identificador* | *Nome* | *Descrição* |
| *UI.0001* | *Tela de usuários* | *Interface on-line para inclusão, consulta, alteração e exclusão de usuários. Contendo os campos X, Y, Z.* |
| *UI.0002* | *Tela de Conciliação de Estoque* | *Interface on-line para conciliação entre estoque registrado e estoque real. Contendo os campos X, Y, Z* |
| *UI.0003* | *Tela de Impressão de Pedidos* | *Interface on-line para impressão dos pedidos de venda e compra feitos. . Contendo os campos X, Y, Z* |

*Descreve os requisitos relacionados às interfaces de usuário que devem ser implementadas pelo software. A intenção desta seção é declarar os requisitos, mas não descrever a própria interface de usuário, porque o design da interface pode sobrepor o processo de obtenção dos requisitos. Isto é particularmente verdadeiro se você estiver usando a prototipagem como parte de seu processo de coleta de requisitos. À medida que você desenvolver os protótipos, é importante capturar os requisitos que se relacionam aos aspectos visuais da interface de usuário. Ou seja, esteja certo que você compreende as intenções do seu cliente para os aspectos visuais do produto. Registre-os como requisitos, ao invés de meramente usar um protótipo para aprovação.*

*Aspectos Visuais: Uma descrição da aparência e da disposição estética da interface. Seu cliente pode ter lhe solicitado demandas específicas, tais como estilo, cores, grau de interação, etc. Esta seção captura os requisitos para a interface, e não o design da interface. A motivação é capturar as expectativas, as restrições e as demandas do cliente para a interface antes de projetá-la. Exemplos:*

*O produto terá a mesma disposição dos mapas do distrito para o departamento de engenharia.*

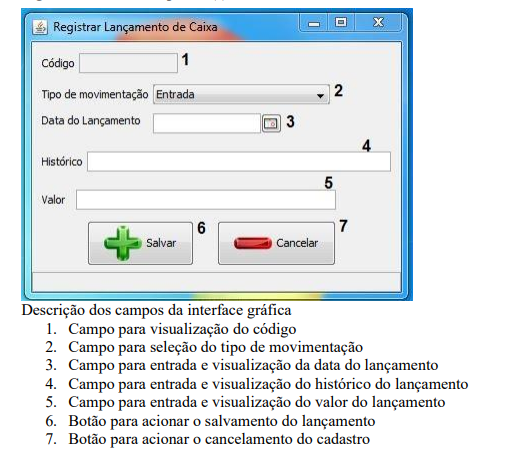
*O produto usará a cor da empresa.*

*Requisitos de layout e de navegação: Especifica os requisitos das principais áreas de tela e como devem ser agrupadas.*

*Consistência: A consistência na interface de usuário permite aos usuários predizer o que irá acontecer. Esta seção declara os requisitos sobre o uso dos mecanismos a serem empregados na interface de usuário. Isto se aplica ao sistema, e a outros sistemas e pode ser aplicado em diferentes níveis: controles de navegação, formas e tamanhos das áreas de tela, locais para entrada ou apresentação de dados e terminologia.*

*Requisitos de personalização do usuário: Requisitos sobre o conteúdo que deve ser automaticamente exibido aos usuários ou estar disponível baseado nos atributos de usuário. Às vezes os usuários autorizados a personalizar o conteúdo exibido.*

*Também é uma área importante onde podem ser adicionados os protótipos de telas. Como por exemplo:*



*]*

### Interfaces de Hardware

*[Esta seção define as interfaces de hardware que devem ser suportadas pelo software, incluindo estrutura lógica, endereços físicos, comportamento esperado e assim por diante.*

*Interfaces de Hardware: Define qualquer interface de hardware que deve ser suportada pelo software, incluindo estrutura lógica, endereços físicos, comportamento previsto, etc.]*

### Interfaces de Software

*[Esta seção descreve as interfaces de software para outros componentes do sistema de software. Estes podem ser componentes comprados, componentes reutilizados de outro aplicativo ou componentes que estão sendo desenvolvidos para subsistemas fora do escopo desta* ***SRS****, mas com os quais este aplicativo de software deve interagir.*

*Se houver integrações com outros sistemas deve-se apontar aqui as interfaces de integração, componentes que de outros sistemas que vão ser reutilizados.*

*Interfaces de Software: Existe algum sistema externo com o qual este sistema deve se conectar? Existe alguma restrição na natureza da interface entre este sistema e algum sistema externo, tal como o formato dos dados passados entre estes sistemas? Eles usam algum protocolo em particular? Descreva as interfaces de software com outros componentes. Podendo ser componentes comprados, componentes reutilizados de uma outra aplicação, ou componentes que estão sendo desenvolvidos para subsistemas fora do escopo do sistema em questão, mas com o qual ele deve interagir. Para cada sistema, considere as interfaces fornecidas e requeridas.*

*]*

### Interfaces de Comunicações

*[Descreva as interfaces de comunicações para outros sistemas ou dispositivos como redes locais, dispositivos seriais remotos e assim por diante.*

*Interfaces de Comunicação: Descreve todas as interfaces de comunicação com outros sistemas ou dispositivos, tais como redes de área local (LANs), dispositivos seriais remotos, etc]*

## Padrões Aplicáveis

*[Esta seção descreve, por referência, o padrão aplicável e as seções específicas desses padrões que se aplicam ao sistema que está sendo descrito. Por exemplo, isso pode incluir padrões jurídicos, de qualidade e reguladores, padrões de mercado para utilidade, interoperabilidade, internacionalização, conformidade com o sistema operacional e assim por diante.]*