SARCTD

Supporting Requirements

# System Qualities

O produto desenvolvido nessecita uma confiabilidade muito alta na execução de seus algoritmos de balanceamento, devido a falta de validação humana sobre o processo. Além disso o produto será responsável por manter uma divisão coerente para seus utilizadores.

Sua usabilidade não pode ser interrompida devido a erros de implementação durante a execução do algoritmo de aplicação, não podendo impedir a concretização do objetivo.

## Usability

Sua aplicação será disponibilizada em um domínio online, para que possa ser acessada a qualquer momento. Sendo que o gerenciamento de recursos deverá estar centralizado somente através do Sistema, sem interação humana(com excessão do suporte). Ela facilita a organização e o controle dos recursos, podendo ser acessível aos usuários para realizarem trocas durante o semestre. A aplicação deve reduzir o tempo de resposta do processo em 50% e aumentar sua confiabilidade para 95% em relação ao antigo método utilizado.

## Reliability

O Sistema será alojado em um servidor flexível e distríbuido. Por ser muito varíavel sua utilização, com picos extremos de acesso, a aplicação deverá utilizar o azure para que possa ser escalável em relação aos acessos simultâneos e distribuído para se manter acessível com menos chances de falhas.

## Performance

A performace do produto não será seu ponto crítico, pois sua usabilidade será pra funções muito simples e rápidas, independentemente da capacidade do servidor ou da otimização do código. Sendo utilização um padrão de qualidade e homologação de código, para que não seja tolerado inconsistências de implementação.

## Supportability

A aplicação deverá ser utilizada por qualquer dispositivo capaz de acessar a web, para que sua usabilidade possa ser aplicada de qualquer lugar. Por ser utilizado em aplicações móveis com internet de baixa banda, a App também necessitará ter uma comunicação baixa entre servidor e cliente, executando seu processamento todo no cliente, sem afetar sua segurança.

# System Interfaces

## User Interfaces

### Look & Feel

O design da Interfaces do usuário deverão manter um padrão simples e fácil de interpretar. Os fluxos serão gerados pelo sistema, no qual o usuário somente solicita o tipo de interação com o projeto(Cadastrar, solicitar equipamento, realizar troca, etc.). Com o padrão do Bootstrap de design a interação do usuário com o sistema não pode ser considerado um caso crítico, devido a patente e nível de conhecimento dos usuários.

### Layout and Navigation Requirements

Os fluxos acessados por usários cadastrados do sistema variam entre qual tipo de ação desejam realizar. Partindo do principio que o usuário já está logado, sua tela inicial será os tipos de operações desejadas, na qual ele seleciona o que deseja fazer no sistema. Após isso o sistema será responsável por direcionar seu fluxo a lugares respectivos para a concretização de seu objetivo. Aonde partiremos de uma premissa que o sistema possuí tais operações: Cadastro, solicitação de recursos, solicitação de troca e listagem de resultados.

### Consistency

A Aplicação será multi-plataforma, podendo ser acessada de qualquer dispositivo capaz de acessar a internet via navegadores. Com isso serão criados alguns tipos de padronização de dados visíveis ao usuário. Em dispositivos móveis que o acesso é mais precário, o sistema utilizará uma inteligência de auto-complete e de busca de dados distribuídos, facilitando seu uso.

### User Personalization & Customization Requirements

Existem dois tipos de permissão na aplicação: a de usuário simples e o Administrador. Essas duas possíbilidades possuem funções, totalmente distintas, pois o admin não pode acessar o fluxo do usuário e vice-versa. Os usuários simples não poderão executar fluxos que não fazem parte da sua necessidade ao usar o sistema. O Administrador é utilizado para teste e carga de dados, ou alteração de produtos já cadastrados.

## Interfaces to External Systems or Devices

### Software Interfaces

A aplicação necessita fazer alguns WebServices para se conectar com o banco da PUCRS onde ela disponibilizará dados fundamentais para seu funcionamento, como usuários, turmas, quantidade alunos, etc. para a carga de dados no sistema e atualização de outros. Os serviços devem retornar em formato JSON para sua manipulação.

### Hardware Interfaces

As condições de uso a nível de hardware de usuário não são aplicadas, porém o servidor deve ser distríbuido e possuir uma alta escalabilidade e tolerância a erros.

# Business Rules

Todos os Usuários conectados ao sistema, deverão logar com a conta da pucrs, caso o contrário não poderá se cadastrar no sistema; Usuários deverão poder;

Cada usuário poderá solicitar recursos computacionais de acordo com seus horários de aula;

A solicitação de troca de recurso deverá ser através de um calendário, notificando os horários correspondentes ao de sua grade e os recursos devem corresponder aos solicitados;

O algoritmo de balanceamento deve ser executado em um prazo pré-definido pelo Administrador com no minimo uma semana antes das aulas começarem;

Recursos computacionais podem ser cadastrados a qualquer instante e sua solicitação deverá ser disponibilizada aos professores que menos possuirem recursos, caso não seja escolhido por ninguém ele disponibilizará a todos;

# System Documentation

O sistema deve possuir uma página de ajuda, na qual o usuário solicita o fluxo desejado e o sistema irá ensiná-lo a concretizar o processo. Caso não tenha sido útil será fornecido o contato do suporte ao final do tutorial. Para o desenvolvimento do processo de ajuda será necessário a utilidade do Gerente de Projetos.