

SISTEMAS DE INFORMAÇÃO

Gabriel Vasconcelos

**Tópicos em Desenvolvimento de sistemas**

SALVADOR

2020

Gabriel Vasconcelos

**Tópicos em Desenvolvimento de sistemas**

Trabalho sobre desenvolvimento de sistemas, com o foco na documentação de uma aplicação, objetivo complementar e aperfeiçoar os conhecimentos adquiridos na sala de aula da disciplina de tópicos em desenvolvimento de sistemas.

Orientador: Heleno Filho

SALVADOR

2020

**SUMÁRIO**

[1. OBJETIVO 4](#_Toc37141775)

[2. INTRODUÇÃO 4](#_Toc37141776)

[3. DESENVOLVIMENTO 4](#_Toc37141777)

[5. ATORES 5](#_Toc37141778)

[6. REQUISITOS FUNCIONAIS 5](#_Toc37141779)

[7. REQUISITOS NÃO-FUNCIONAIS 6](#_Toc37141780)

[8. DIAGRAMA DE CASOS DE USO 6](#_Toc37141781)

[9. REQUISITOS 8](#_Toc37141782)

[10. CLASSMATCH 8](#_Toc37141783)

# 1. OBJETIVO

Documentação dos processos de desenvolvimento de uma ferramenta de software com base nos conteúdos exibidos e aprendidos em sala de aula. Por sua vez o programa será responsável pelo gerenciamento e controle de todas as dependências de um local, para que seja fluido e não haja conflitos relacionados as salas de uma organização.

# 2. INTRODUÇÃO

O aplicativo ClassMatch foi desenvolvido para suprir a necessidade do gerenciamento das salas de um empreendimento, atualmente é uma ferramenta essencial para o funcionamento de qualquer empresa que queira manter o controle e a organização de suas dependências, como por exemplo salas de reuniões, salas de aulas de universidades ou escolas. O aplicativo irá fazer a reserva da sala a partir da solicitação de um requerente cadastrado, verificando a sua disponibilidade de acordo com a data e hora informada pelo solicitante, efetuando assim a reserva caso a mesma esteja livre.

# 3. DESENVOLVIMENTO

Para chegar às funcionalidades finais do sistema, foram utilizados HTML5 e Javascript . Na estilização, foi utilizado CSS3e para o banco de dados o MySQL. O método de desenvolvimento de software utilizado foi o SCRUM.

**HTML5:** Hypertext Markup Language 5 (HTML5) é uma linguagem utilizada para desenvolvimento de páginas web. Trata-se da versão mais recente da linguagem, trazendo diversos novos atributos em relação às versões anteriores. Alguns destaques das funcionalidades utilizadas no projeto são as tags “date”, que abre um calendário para que o usuário escolha a data de forma interativa, e “time”, que permite que o usuário insira um horário, dentre várias outras.

**Javascript:** É uma linguagem que, em conjunto com o HTML, permite que funções sejam desenvolvidas para que a página web se torne dinâmica, permitindo o envio e recebimento de informações pelo sistema para que as funcionalidades de fato sejam incorporadas.

**CSS3:** Cascading Style Sheets 3 (CSS3) permite que sejam adicionados às páginas web todos os estilos necessários, como, por exemplo, cores, espaçamentos, tamanhos e tipos de fontes, entre outros. Trata-se da versão mais recente da linguagem.

**SCRUM:** Éuma metodologia ágil para gestão e planejamento de projetos de software, os projetos são divididos em ciclos chamados de Sprints. Os Sprints representam um Time Box no qual um conjunto de atividades deve ser executado dentro deles.

**MySQL:** O MySQL é um sistema de gerenciamento de banco de dados, que utiliza a linguagem SQL como interface.

# 5. ATORES

Os seguintes atores possuíram o acesso ao software, são eles:

**Funcionário**: Um funcionário da instituição será responsável pelo cadastro, alterações das informações e exclusões de uma devida sala.

**Palestrantes**: O palestrando solicitará por meio do aplicativo o uso de uma sala informando a data, horário e sala para a reunião de acordo a disponibilidade da sala.

**Público**: O público que recebera o conteúdo do expositor poderá ver qual dependência será alocada a sala a qual ocorrerá a reunião mediante as informações do palestrante, ou podendo enxergar o calendário e ver a relação de palestras marcadas no período escolhido.

# 6. REQUISITOS FUNCIONAIS

Os requisitos funcionais descrevem as funcionalidades que cada tipo de usuário deve ser capaz de acessar no sistema (SOMMERVILLE, 2011). Esses requisitos estão relacionados às atividades que o sistema realiza.

**Cadastro de usuário (RF1):** O sistema irá disponibilizar uma área para que o funcionário da empresa possa cadastrar novos usuários, sendo eles: palestrante ou outro funcionário que manipulará o sistema.

**Autenticação (RF2):** O sistema vai disponibilizar uma tela de login na qual o usuário digitará suas credenciais e terá acesso às funcionalidades de acordo com sua função no sistema (funcionário ou palestrante).

**Redefinição de senha (RF3):** O sistema possuirá capacidade de enviar um link para o e- -mail do usuário para a redefinição de senha, caso esse tenha perdido a anterior.

**Cadastro de dependências (RF4):** O sistema será capaz de ter uma área privada onde um funcionário da instituição fara as alterações necessárias do sistema como sala, palestrante, data e horário.

**Agendamento de reunião (RF5):** Através da aplicação após efetuar o login de expositor poderá ser feito um agendamento de uma sala de acordo a disponibilidade da mesma no momento solicitado.

**Calendário (RF6):** O software possuirá um calendário onde os usuários poderão enxergar os dias do período e as reuniões e informações de cada reunião seja ela data e horário, palestrante e localização da dependência.

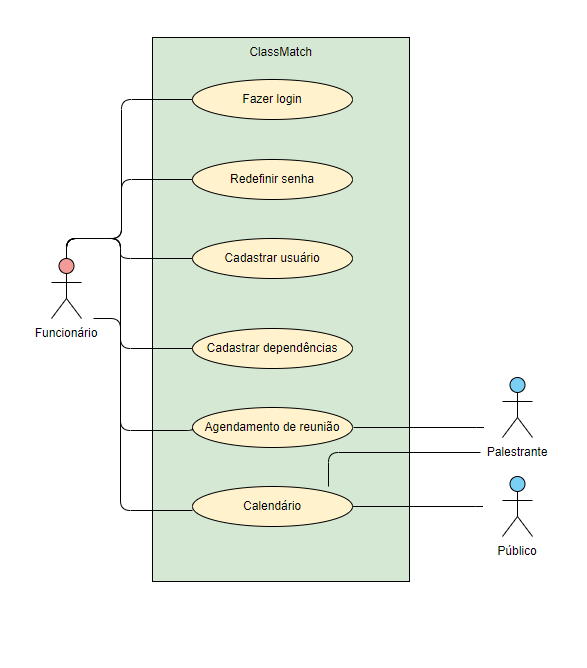
# 7. REQUISITOS NÃO-FUNCIONAIS

Os requisitos não-funcionais são aqueles que não se relacionam diretamente com as funções do sistema; no entanto, são restrições impostas aos serviços oferecidos pelo sistema (SOMMERVILLE, 2011).

**Estado da dependência (RFN1):** O sistema apresentará o estado e condições físicas detalhadas de cada dependência para o funcionário de acordo com as informações coletadas pelos usuários palestrantes.

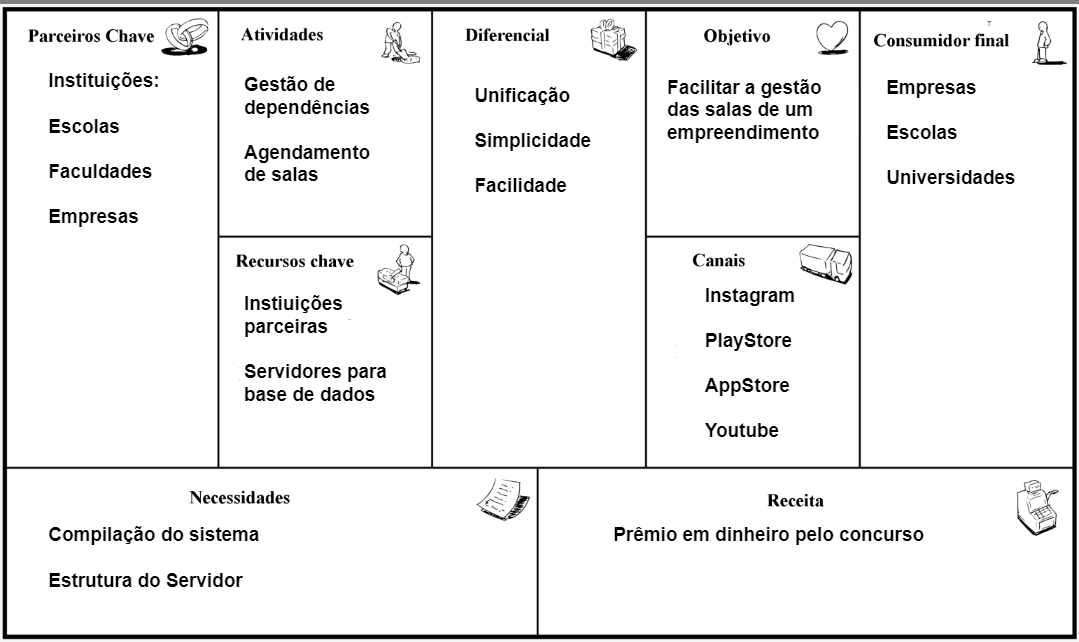
**Chamado para cada dependência (RFN2):**  A aplicação possuirá um campo onde o responsável pela reunião poderá solicitar um chamado imediato se algo estiver impedindo a fluidez da reunião que notificará um funcionário responsável para que seja dado o suporte requerido urgentemente.

# 8. DIAGRAMA DE CASOS DE USO

Considerando os requisitos descritos anteriormente, é possível modelar as funcionalidades do sistema. Para isso, inicialmente, é apresentado na figura 1 o diagrama de casos de uso do sistema. Dos três atores do sistema (funcionário, palestrante e público), apenas o funcionário tem acesso a funcionalidade geral do sistema. Os demais usuários são restritos das outras funcionalidades do sistema, o usuário palestrante poderá somente agendar a reunião e visualizar o calendário o usuário público somente o calendário. Figura 1 - Diagrama de casos de uso da aplicação

# 9. REQUISITOS

A análise de requisitos como na figura 1 abaixo, é um conjunto de técnicas que visa levantar, classificar e adequar as necessidades reais de um software em conjunto com cliente ou consumidor final, visando focar melhor os esforços relacionados à sua composição e compreender o que realmente deve ser feito durante a execução do projeto.

Figura 1 - Levantamento de requisitos do ClassMatch – Modelo CANVAS

# 10. CLASSMATCH

Na tela mostrada na figura 1 os usuários públicos e todos outros conseguem visualizar o calendário e suas informações, logo em seguida caso o usuário queira marcar uma reunião, clicando no dia desejado e preenchendo as informações solicitadas na figura 3 o usuário será direcionado a página de login figura 4, posteriormente digita as credenciais e obtém acesso ao sistema. Se o usuário for um funcionário, ele é direcionado para a tela mostrada na figura 2, onde consegue visualizar todas as funcionalidades do sistema. Se o usuário for um palestrante, ele é direcionado para a tela mostrada na figura 4, caso o palestrante não seja cadastrador ele irá solicitará para o funcionário cadastra-lo como pode ver na figura 4 onde consegue visualizar apenas o calendário e agendar uma reunião.

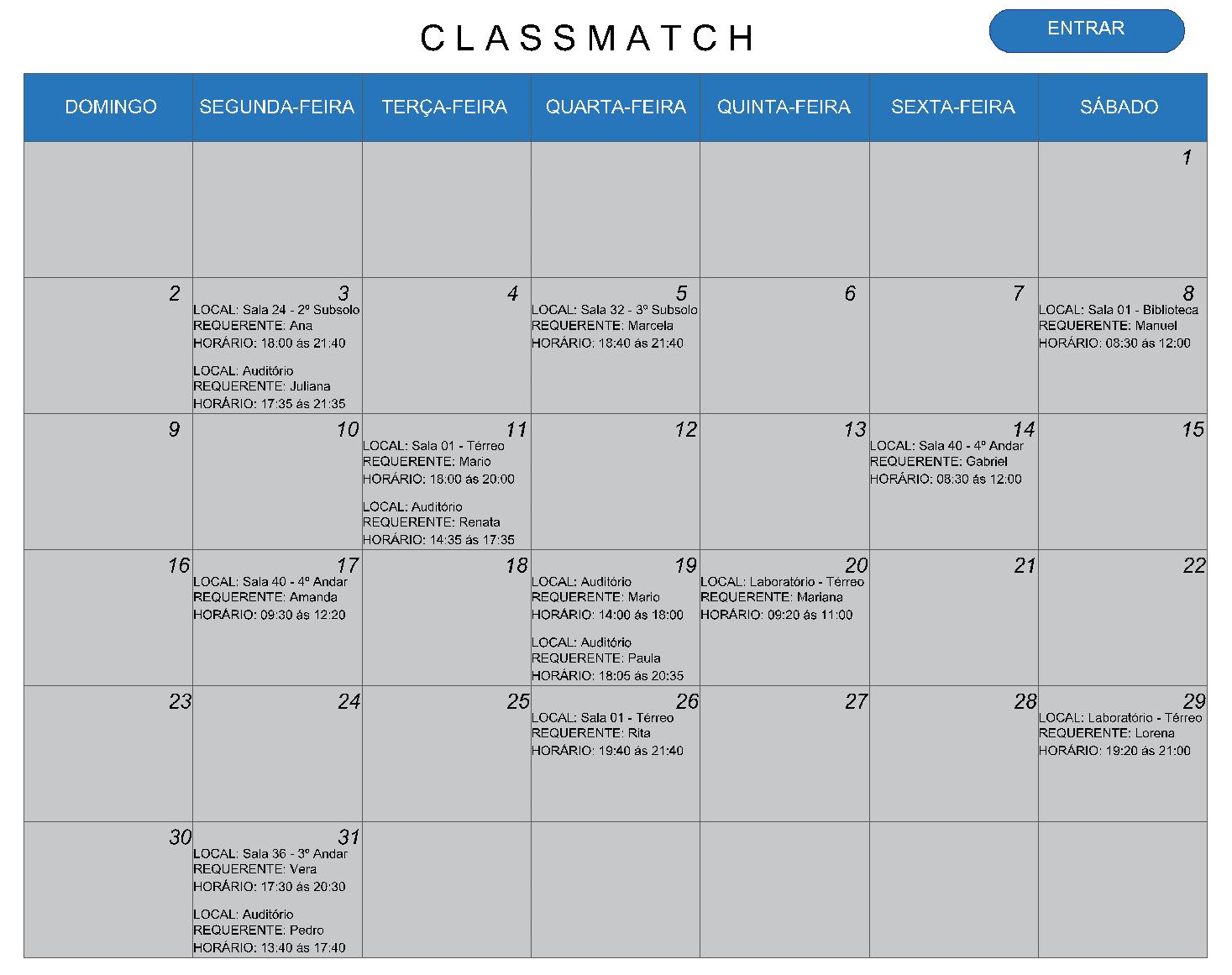
Figura 1 - Tela de cadastro de ClassMatch



Figura 2 - Tela de cadastro do ClassMatch



Figura 3 - Tela de agendamento do ClassMatch



Figura 4 - Tela de cadastro de ClassMatch

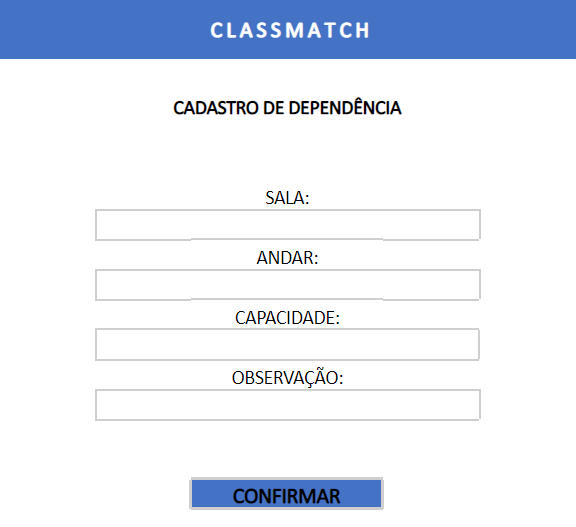


Figura 5 - Tela de cadastro de dependências do ClassMatch