

Reserva Já: Solução em reservas de Salas

Joice da Cruz Marques¹, Daniela Marques¹

¹Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Campus Hortolândia
CEP: 13183-250 – Hortolândia - SP – Brasil

joicemarques170@gmail.com, marquesdaniela@ifsp.edu.br

Abstract. *In several places, such as companies, schools and colleges, it is necessary to use rooms for events or other activities, there are still few solutions that manage the reserves for the use of these rooms. This paper will present the process of developing a new website, which aims to provide a management for room reservations, following the practices of the SCRUM methodology, and the use of technologies that will be presented.*

Resumo. *Em diversos locais como empresas, escolas e faculdades se faz necessária a utilização de salas para realização de eventos ou outras atividades, atualmente ainda existem poucas soluções que gerenciem completamente as reservas para utilização dessas salas. Nesse artigo será apresentado o processo de desenvolvimento de uma nova aplicação web, que visa fornecer uma gestão para reservas de sala, seguindo as práticas da metodologia SCRUM.*

1. Introdução e Motivação

Em diversos locais, como empresas, escolas e faculdades se faz necessária a utilização de salas para realização de diversas atividades, e após uma pesquisa de mercado sobre aplicações existentes, foi encontrada uma pequena quantidade de aplicações que visam solucionar problemas que essa necessidade acarreta. Aplicações por exemplo, onde é possível escolher salas para realizar seu evento e reservá-las.

Pensando em uma forma de melhorar o processo de gerenciamento de salas, o Reserva Já foi planejado para permitir que qualquer empresa ou instituição interessada em administrar a utilização de suas salas, realize e visualize as alocações das salas, fazendo com que os usuários possam garantir que suas atividades sejam realizadas no dia, horário e local desejado. Além disso, a aplicação *web* permite o cadastro de empresas, usuários, salas, equipamentos, oferecendo uma nova solução para o gerenciamento da utilização das salas.

O presente trabalho terá as seguintes seções: a Seção 2 apresenta a metodologia, materiais e métodos utilizados; na seção 3 é descrito o desenvolvimento do trabalho; e, finalmente, a Seção 4 descreve a conclusão do trabalho com sugestões de continuidade no desenvolvimento da proposta.

2. Materiais e Métodos

A seguir serão apresentadas os materiais e métodos utilizados no desenvolvimento da aplicação Reserva Já.

2.1. Metodologia Ágil - Scrum

"O *Scrum* é uma abordagem de desenvolvimento de *software* iterativa e incremental"[Cohn 2011] promovendo entrega de valor ao final de cada *Sprint*. O *Scrum* visa a redução de riscos, maior qualidade e entrega de valor do produto desenvolvido, aumento da produtividade, entregas incrementais, autogerenciamento das equipes, maior aceitação à mudanças durante o ciclo de desenvolvimento, entre outros.

2.1.1. O ciclo do Scrum

O *Scrum* possui um ciclo, que descreve passo a passo o processo do desenvolvimento de software com essa metodologia, conforme a Figura 1 [Cruz 2018].

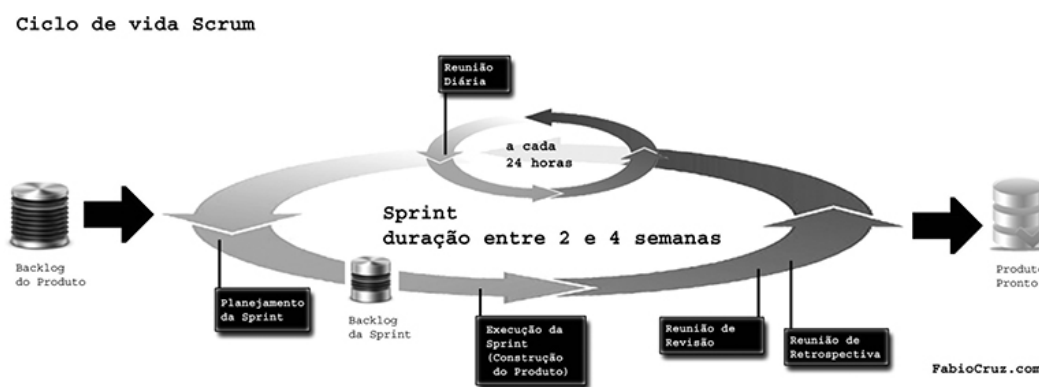


Figura 1. Ciclo de vida do Scrum

O processo se inicia com a definição do *Backlog* do produto que é uma lista com todas as funcionalidades desejadas e priorizadas de acordo com a necessidade do cliente. A partir dessa lista de funcionalidades são escritas as histórias de usuário.

O *Sprint Backlog* é lista de priorização das histórias que serão desenvolvidas na *Sprint* que irá ser iniciada.

A *Sprint* é o período de desenvolvimento que pode durar de duas a quatro semanas, de acordo com as horas estimadas para as funcionalidades que foram priorizadas. Diariamente, há um *Daily Meeting* na execução da *Sprint* onde os participantes respondem basicamente a três perguntas: i) O que fiz ontem que ajudou a equipe a atingir a meta da *Sprint*?, ii) O que vou fazer hoje para atingir a meta da equipe? e iii) Existe algum impedimento que não me permite atingir a meta? [Cruz 2018].

E o último passo do ciclo é quando o produto está pronto e é apresentado ao cliente. Nesta etapa (i) o cliente pode fazer considerações sobre a entrega feita e (ii) a equipe pode fazer uma retrospectiva de como foi a *Sprint* em relação às pessoas, relacionamentos, ferramentas e processos e propor um plano de melhoria para o trabalho da equipe para o próximo ciclo de desenvolvimento. Por se tratar de um processo iterativo e incremental, a cada entrega o produto vai adquirindo novas funcionalidades.

2.2. Coleta de Requisitos

Ao iniciar o processo de desenvolvimento de um *software*, é normalmente realizada uma coleta de requisitos. Nesta etapa deve ser feita uma análise com o cliente do produto e a partir dessa análise serão definidos quais os requisitos funcionais e não funcionais que o *software* deverá conter para atender as necessidades do solicitante. Segundo [Sommerville 2011]:

"Na realidade, a distinção entre diferentes tipos de requisitos não é tão clara como sugerem essas definições simples. Um requisito de usuário relacionado com a proteção, tal como uma declaração de limitação de acesso a usuários autorizados, pode parecer um requisito não funcional. No entanto, quando desenvolvido em mais detalhes, esse requisito pode gerar outros requisitos, claramente funcionais, como a necessidade de incluir recursos de autenticação de usuário no sistema."

Após a coleta dos requisitos os mesmos podem ser transcritos para um caso de uso ou para histórias de usuário, como no *Scrum*, onde serão descritos o que cada requisito irá executar no sistema. No caso das histórias de usuário, haverão também os critérios de aceite, que determinarão quais são as necessidades fundamentais para que aquele requisito seja atendido.

2.2.1. Estórias de usuário

As histórias de usuário são a forma de detalhar uma funcionalidade no *Scrum* de forma clara e objetiva. Um exemplo de como uma história deve ser escrita é:

"Como um <autor da ação> eu quero <objetivo> para que <atenda uma necessidade>".

A descrição possui informações do autor da ação, que descreve quem é o responsável por executar a funcionalidade no sistema. O objetivo, descreve o que o autor realizará, e a necessidade, é o que acontecerá ao ter o objetivo atendido.

Logo após a descrição, os critérios de aceite devem ser listados. Esses critérios de aceite são os itens fundamentais para verificar se após implementada a história, a mesma contempla todos os critérios listados. Uma história pode conter um ou mais critérios de aceite, e todos devem ser atendidos na entrega da *Sprint*.

2.3. Diagrama de Navegação

Um diagrama é uma representação visual do sistema a ser desenvolvido, existindo diversos tipos, cada um com sua finalidade específica no desenvolvimento. O diagrama de navegação é um deles, o qual representa graficamente, o fluxo entre as possíveis telas do sistema. A Figura 2 apresenta um exemplo do diagrama que possui retângulos e linhas. O retângulo representa as telas, e as linhas a navegabilidade entre as telas, ou seja, a partir da Tela 1 é possível acessar a Tela 2.



Figura 2. Diagrama de Navegação

2.4. Tecnologias

A seguir, serão apresentadas as tecnologias utilizadas no desenvolvimento da aplicação *web* Reserva Já.

2.4.1. JavaServer Faces

O *JavaServer Faces* (JSF) é uma extensão do Java EE, utilizado para aprimorar e reduzir o tempo de desenvolvimento, sendo uma tecnologia que nos permite criar aplicações Java para *web* utilizando componentes visuais pré-prontos. Existem diversas bibliotecas que são extensões com componentes mais sofisticados do JSF, um deles, o *Primefaces* oferece diversas classes prontas para serem utilizadas no desenvolvimento [Caelum 2018].

2.4.2. Maven

Maven é um *software* de código aberto que é responsável pelo gerenciamento e compreensão de projetos de *software*, utilizando de um modelo de objeto de projeto, denominado POM, que significa *Project Object Model*. No POM são declaradas todas as dependências do projeto e outras configurações [Apache 2018].

2.4.3. Hibernate

O *Hibernate* é um *framework* utilizado para mapeamento objeto-relacional para ambientes Java. Segundo [Izalmo 2018], "mapeamento objeto-relacional (ORM) refere-se à técnica de mapeamento de uma representação de dados em um modelo de objetos para um modelo de dados relacional". Esse mapeamento é feito à partir de anotações específicas do *framework*, diretamente nas classes do projeto. O *Hibernate*, além de permitir o mapeamento, possibilita a criação de consultas e persistência dos dados, e tem como objetivo reduzir a complexidade e o tempo de mapeamento das tabelas implementando a especificação JPA que significa *Java Persistence API* através das anotações.

2.4.4. Materialize

Materialize é um *framework* para desenvolvimento *web*, utilizado para facilitar a criação do *layout* de aplicações para torná-los mais simplistas e sofisticados, oferecendo componentes prontos em sua biblioteca que podem ser utilizados em seu código HTML [Materialize 2018].

2.5. Segurança da informação

A área da segurança da informação, segundo [Stallings 2014] consiste em fornecer medidas de prevenir violações da segurança, garantindo formas de manter a confidencialidade, integridade e disponibilidade, que são os três conceitos principais sobre a segurança. A confidencialidade assegura que os dados serão mantidos sob sigilo para indivíduos não autorizados. Integridade diz respeito à preservação dos dados, garantindo que os mesmos não sofram alterações sem autorização, e que sejam manipulados somente com autorização específica. E disponibilidade assegura que os serviços não fiquem indisponíveis durante sua utilização.

Dentre os mecanismos de segurança, existe a codificação que é uma forma de utilizar-se de algoritmos matemáticos para criptografar mensagens. Um desses algoritmos é a função de *hash*, que criptografa uma mensagem recebida de tamanho variável, e produz um *hash* de tamanho fixo.

```
public String encryptPassword(String senha) throws NoSuchAlgorithmException, UnsupportedEncodingException {  
    MessageDigest algorithm = MessageDigest.getInstance("SHA-256");  
    byte messageDigest[] = algorithm.digest(senha.getBytes("UTF-8"));  
  
    StringBuilder hexString = new StringBuilder();  
    for (byte b : messageDigest) {  
        hexString.append(String.format("%02X", 0xFF & b));  
    }  
  
    return hexString.toString();  
}
```

Figura 3. Método Hash Criptográfica

Na Figura 3, a classe *MessageDigest* do *javax.security* é utilizado para a geração de um *array* de *bytes*, onde é feita a criptografia, os hexadecimais dos *bytes* são transformados em uma *string*, e preparados para serem salvos no banco de dados.

Para validar a senha do usuário posteriormente, na realização do login por exemplo, o método pode também ser implementado, e ao receber a senha a *string* gerada ao final do método é comparada com as strings dos hexadecimais do *hash* que foi salvo, para o usuário que está tentando realizar o *login* na aplicação.

3. Desenvolvimento

Nesta seção é apresentado o ciclo de desenvolvimento deste trabalho. Será mostrado a coleta de requisitos, criação das estórias de usuário, o diagrama de navegação, apresentação do DER, as execuções das *Sprints* e, ao final, os desafios encontrados no durante o processo de desenvolvimento.

3.1. Levantamento de requisitos e pesquisa de mercado sobre aplicações semelhantes

A coleta dos requisitos foi iniciada analisando a necessidade de um sistema de reserva de salas para o Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, tendo como resultado as funcionalidades apresentadas na Tabela 1. Após o levantamento dos requisitos foi constatado que o aplicativo poderia ser algo mais abrangente e não apenas

para a instituição em questão, mas sim, para qualquer empresa ou instituição que necessite reservar salas.

Baseado nos requisitos levantados, foi feita a pesquisa de aplicações correlatas, visando verificar se algum software já possuía todos os requisitos desejados. A Tabela 1 apresenta os requisitos levantados para a aplicação Reserva Já em comparação com os demais aplicativos pesquisados.

Tabela 1. Tabela Comparativa

Funcionalidades	Sistemas			
				
Login no Sistema	✓	✓	✓	✓
Cadastrar Empresa	✓	✗	✓	✓
Cadastro de Salas	✓	✓	✓	✓
Cadastro de Equipamentos	✓	✗	✗	✗
Envio de Email	✓	✗	✗	✗
Realizar Reserva	✓	✓	✓	✗
Visualizar Reserva	✓	✓	✓	✗
Gerar Relatório	✗	✗	✓	✗

SUAP é uma aplicação, utilizada pelo Instituto Federal, que permite a realização das reservas das salas disponíveis para os professores, e posteriormente, os alunos podem visualizar onde o professor reservou a sala para a sua aula [SUAP 2018].

BIZLY é uma aplicação americano que facilita a reserva de diversos locais para realização de reuniões e eventos [Bizly 2018].

Deskoverly é também uma aplicação que facilita a utilização de salas de ambientes *Coworking*. Ele facilita a busca de locais disponíveis como, por exemplo, escritórios [Deskoverly 2018].

Conforme nota-se pela Tabela 1, o Reserva Já se destaca por ser uma aplicação que fornece mais funcionalidades aos seus usuários. O aplicativo *web* fornece cadastro de empresa, salas e equipamentos, permite que na reserva da sala sejam selecionados equipamentos para uso e, além disso, a visualização da reserva facilita a compreensão dos usuários quanto a quais salas estão sendo usadas no momento, ou que vão ser usadas.

3.2. Priorização do *Backlog*

Após a coleta dos requisitos e a pesquisa sobre os aplicativos similares, o *Backlog* de Produto foi priorizado e cada estória foi designada à uma *Sprint*, conforme Tabela 2.

Cada *Sprint* possui suas estórias com devidas descrições e seus critérios de aceite e sua duração foi de uma semana.

3.3. Criação do diagrama de navegação

A aplicação possuirá dez páginas (Figura 4). A tela inicial ao acessar o aplicativo *web* será a tela de Login, nela o usuário poderá escolher entre realizar o login no sistema, ou acessar o cadastro de empresa.

Tabela 2. Resultado da Priorização do Backlog

Backlog	
Login no Sistema	Sprint 1
Cadastrar Empresa	Sprint 2
Cadastrar Usuário	Sprint 3
Cadastrar Salas da Empresa	Sprint 4
Cadastrar Equipamentos	Sprint 5
Reservar sala	Sprint 6
Visualizar Reservas	Sprint 7

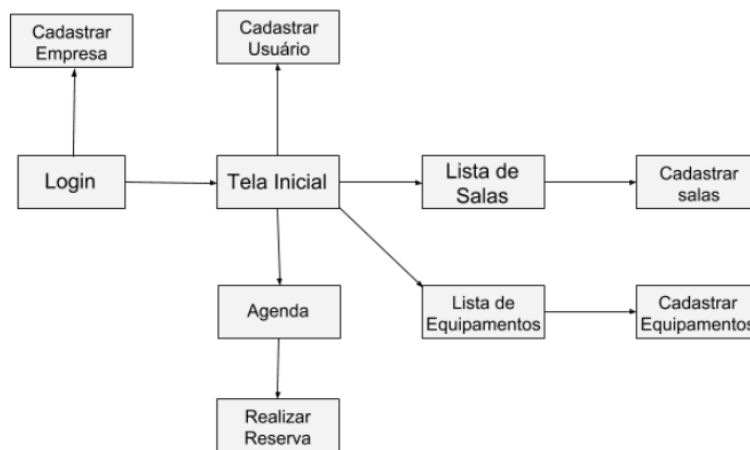


Figura 4. Diagrama de Navegação

Após efetuado o login, a página inicial da aplicação será exibida, contendo um menu fixo, que possui telas que podem ser acessadas e utilizadas, são elas: "Salas", "Equipamentos", "Agenda" e "Cadastrar Usuário".

A tela "Cadastrar Usuário" é a opção do menu que permite a criação de um novo usuário do sistema.

Ao acessar a opção "Salas", a lista com todas as salas cadastradas é disponibilizada para visualização e uma nova sala poderá ser cadastrada através do botão "Nova Sala".

A opção "Equipamentos", assim como "Salas" fornece uma lista com todos os equipamentos cadastrados e um botão para redirecionar ao cadastro de um novo equipamento.

A tela "Agenda" permite a visualização de todas as reservas feitas da data atual em diante. A partir da agenda, ao clicar no botão "Realizar Reserva", é possível selecionar uma nova sala e seus equipamentos para a data escolhida.

3.4. Ciclo de desenvolvimento das Sprints

3.4.1. Sprint 1 - Login no sistema

Na *Sprint 1* foi desenvolvida a história "Login no Sistema" (Figura 5). O login é a primeira tela do sistema, onde o usuário poderá acessar o sistema, ou escolher a opção "Cadastre

sua Empresa", que deverá ser usado para direcionar o usuário para a página na qual poderá cadastrar uma empresa (Figura 6).

Login no sistema

Eu como usuário do sistema, desejo realizar o login no sistema Reserva Já para poder realizar o cadastro de salas, equipamentos, e reservar salas.

Critério de Aceite:

1. Tela inicial do sistema, contendo campos Usuário e Senha;
2. Botão de Entrar;
3. Manter usuário logado para realizar as demais funções do sistema;

Figura 5. Estória "Login no Sistema"

Como forma de manter a segurança a senha informada passa por um método *hash*, que encriptografa a senha fornecida e, então, valida se é compatível com a senha salva no banco para o usuário informado.



Figura 6. Telas referente a estória "Login no Sistema"

3.4.2. Sprints de Cadastros do sistema

Para parametrização e realização das reservas, alguns cadastros são necessários para fornecer as informações que serão utilizadas.

O *wireframe* da Figura 7 foi criado para padronizar todos os cadastros.

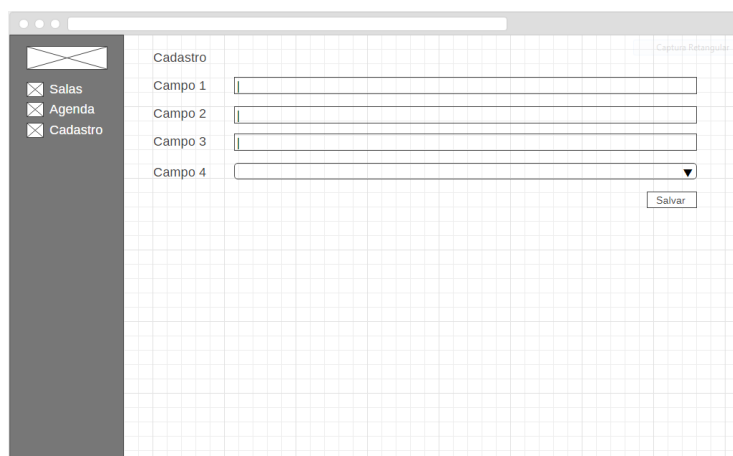


Figura 7. Wireframe utilizado para os Cadastros existentes no Sistema

Cada uma das telas possuem seus campos específicos, todas elas foram criadas utilizando componentes do *Primefaces*, como campos de texto, senha, combos, entre outros, junto aos componentes do *Primefaces*, o *Materialize* também foi utilizado para fornecer todo o estilo e *layout* das páginas. As cores dos cadastros foram padronizadas, sendo todos os campos pretos, e títulos e botões na cor escolhida para a aplicação.

As telas que foram desenvolvidas utilizando o *wireframe* são os cadastro de: Empresa, Salas, Equipamentos e Usuários.

3.4.3. *Sprint 2* - Cadastrar Empresa

Na segunda *Sprint*, a estória "Cadastrar Empresa" foi implementada, com a finalidade de cadastrar novas empresas no sistema. No cadastro da empresa é necessária a criação de um usuário administrador responsável pela mesma (Figura 8). Após a empresa ser cadastrada e fornecido um usuário e senha, o mesmo será utilizado para efetuar login no sistema posteriormente. Na criação do usuário a senha informada é criptografada.

Figura 8. Tela de Inclusão de Empresa

3.4.4. Sprint 3 - Cadastrar Usuário

O cadastro de usuários pode ser acessado pelo usuário administrador do sistema para permitir criação de novos usuários que realizarão as reservas de salas.

Ao salvar, a senha informada também é submetida ao método de criptografia, já utilizado em outras telas, como a do Cadastro de Empresa.

Ao salvar o usuário e sua senha, um método de geração de *hash* é executado, recebendo a senha como parâmetro para ser salva criptografada. O mesmo método é utilizado ao realizar o login, o *hash* gerado é comparado com o *hash* salvo, para verificar se são compatíveis, e consequentemente, se a senha está correta. Se forem iguais, o método retorna verdadeiro, e valida o acesso a aplicação.

Cadastrar Usuário

Eu como administrador quero cadastrar um usuário para permitir que ele realize reservas de salas.

Critério de Aceite:

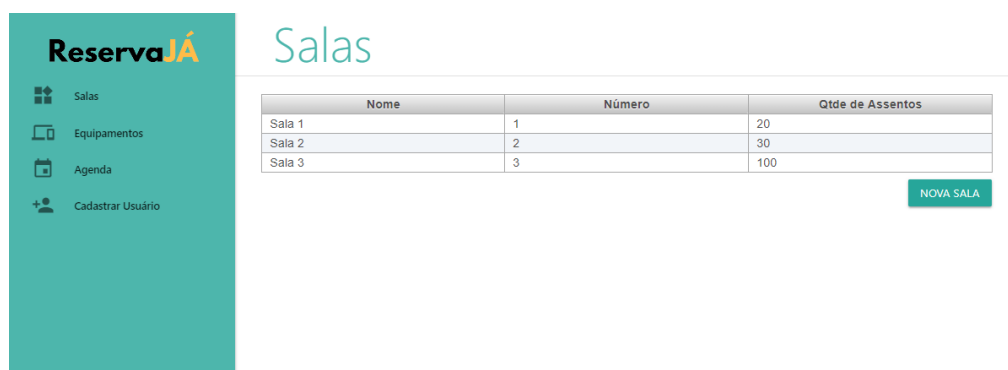
1. Campos: Nome, tipo de usuário (administrador ou funcionário), email, nome de usuário e senha.
2. A senha deve ser criptografada.
3. O usuário poderá realizar reservas das salas disponíveis.

Figura 9. Estória "Cadastrar Usuário"

Ao final da *Sprint* todos os critérios da estória da Figura 9 foram atendidos.

3.4.5. Sprint 4 - Cadastrar Sala da Empresa

Na quarta *Sprint* a estória referente ao "Cadastrar Salas da Empresa" foi implementada e entregue. Nesta estória era possível visualizar uma listagem das salas já cadastradas (Figura 10) e a opção de incluir uma nova sala no sistema (Figura 11).



Nome	Número	Qtde de Assentos
Sala 1	1	20
Sala 2	2	30
Sala 3	3	100

Figura 10. Lista das Salas Existentes

Figura 11. Tela de Inclusão de Sala

3.4.6. *Sprint 5* - Cadastrar Equipamentos

Na quinta *Sprint* a estória "Cadastrar Equipamentos" foi desenvolvida. Os equipamentos adicionados ao sistema podem ser utilizados no momento da reserva da sala.

Na Figura 12 é possível visualizar a listagem dos equipamentos cadastrados no sistema e o botão "Novo Equipamento", que ao ser clicado redireciona o usuário para a inclusão do equipamento, tela apresentada na Figura 13.

Nome	Situação	Descrição
Projektor	Ativo	Projektor e controle
TV 50"	Ativo	TV
Radio	Ativo	Radio para eventos

Figura 12. Lista dos Equipamentos Existentes

Figura 13. Tela de Inclusão de Equipamentos

3.4.7. Sprint 6 - Reservar Sala

Na sexta *Sprint*, a estória "Reservar salas"(Figura 14) foi desenvolvida permitindo que os usuários da aplicação reservem as salas desejadas, informando a data, hora de início e fim, a sala escolhida, e os equipamentos que precisarão ser utilizados durante a reserva.

Reserva sala

Eu como usuário quero reservar uma sala da empresa para poder realizar minhas atividades ou eventos.

Critério de Aceite:

1. Campos: Data, hora início, hora fim, combo com as salas, seleção de um ou mais equipamentos para a reserva.
2. Listar apenas as salas disponíveis na data e horário selecionados.
3. Caso mude a data e horário selecionados a lista de salas deve ser recarregada e caso já tenha uma sala selecionada e essa sala não estiver disponível na nova data e horário o usuário deve ser informado.

Figura 14. Estória Reservar Sala

O *wireframe* da Figura 15 foi criado pensando nos campos descritos na estória. A reserva poderá ser acessada a partir da tela "Agenda". Essa tela permitirá a visualização de todas as reservas realizadas para todas as salas.

O wireframe mostra uma interface de usuário para a reserva de uma sala. À esquerda, há um menu lateral com ícones e os itens "Salas", "Agenda" e "Cadastro". O formulário principal, intitulado "Reserva", contém os seguintes elementos:

- Campos de entrada para "Data:", "Hora início" e "Hora fim".
- Um menu suspenso rotulado "Salas disponíveis".
- Dois grupos de checkboxes sob o título "Equipamentos", cada grupo com cinco opções.
- Uma área de texto rotulada "Observações".
- Um botão "Salvar" no canto inferior direito.

Figura 15. Wireframe usado para a estória "Reservar Sala"

Um dos critérios de aceite definidos ao escrever essa estória foi a validação sobre a disponibilidade da sala selecionada, garantindo que nenhuma sala seja reservada por duas pessoas ao mesmo tempo.

Para os campos sala e equipamentos foram criados combo e *check box*, que listam todas as salas e equipamentos disponíveis (Figura 16). Para a opção de combo da sala é permitido a seleção de uma única sala. Para a opção de *check box* de equipamentos é possível selecionar nenhum ou vários equipamentos desejados.

Nova Reserva

The form includes the following fields:

- Data:** A date selection field.
- Hora início:** A time selection field.
- Hora fim:** A time selection field.
- Salas:** A dropdown menu with options: Sala 1, Sala 2, and Sala 3.
- Equipamentos:** A list of checkboxes for equipment selection:
 - Projetor
 - Radio
 - TV 50"

Figura 16. Exemplo de Combo para Salas e *Check Box* para Equipamentos

A tela de nova reserva (Figura 17) contém todos os campos necessários para sua inclusão.

The screenshot shows the 'Nova Reserva' form with a sidebar on the left containing the following menu items: Salas, Equipamentos, Agenda, and Cadastrar Usuário. The form fields are: Data, Hora início, Hora fim, Salas (dropdown showing 'Sala 1'), and Equipamentos (dropdown). A green 'SALVAR' button is located at the bottom right of the form.

Figura 17. Tela de Inclusão de Nova Reserva de Sala

3.4.8. *Sprint 7* - Visualizar Reservas

A sétima *Sprint* foi dedicada à criação de uma agenda onde é possível visualizar todas as reservas ordenadas pela data da mesma. Cada coluna disponível lista a sala, data, e horário de início e fim da reserva, ao passar o mouse sobre a coluna, ele gira e apresenta uma lista com os equipamentos selecionados para essa reserva (Figura 18).

The screenshot shows the 'Agenda' screen with a sidebar on the left containing the following menu items: Salas, Equipamentos, Agenda, and Cadastrar Usuário. The main content area displays a table with reservation details:

Agenda	
Projetor	Sala: 28
TV 50"	Data: 13/08/2018
Radio	Hora início: 12:46
	Hora fim: 13:46

A green 'RESERVAR SALA' button is located at the bottom right of the screen.

Figura 18. Tela da Agenda

3.4.9. Diagrama de entidade e Relacionamento

O mapeamento relacional de todas as entidades do sistema foi feito utilizando o *Hibernate* através das anotações realizadas nas entidades do projeto. Ao fim do desenvolvimento de todas as entidades e mapeamento das mesmas, o Diagrama Entidade-Relacionamento (DER) representado na Figura 19 foi criado para apresentar todas as tabelas do projeto e os relacionamentos entre cada uma delas.

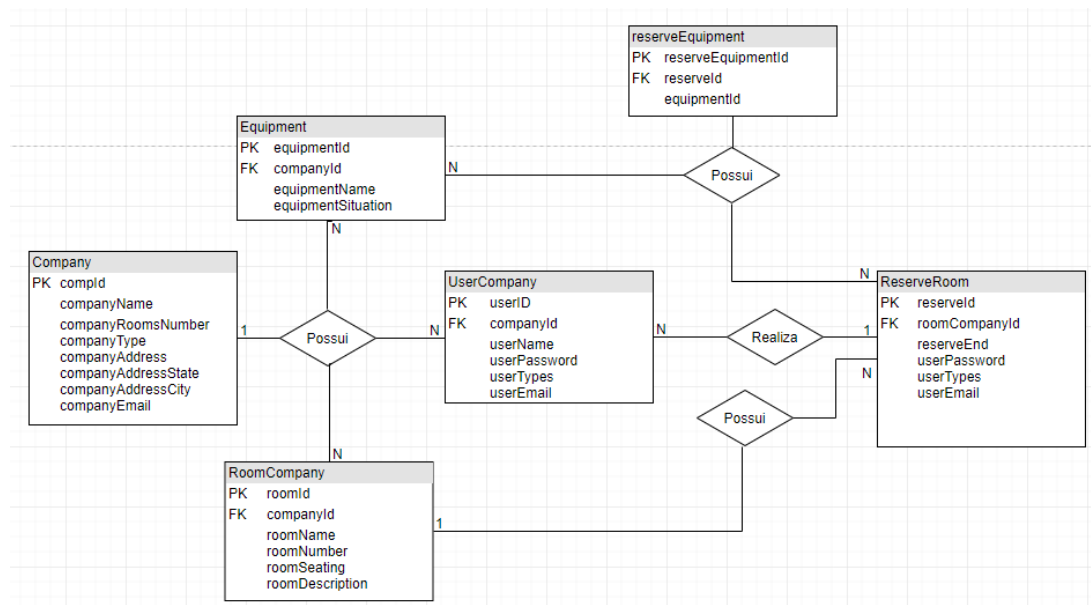


Figura 19. Diagrama de Entidade e Relacionamento

4. Conclusão

A proposta inicial do desenvolvimento de uma aplicação de gerenciamento de reservas com mais funcionalidades e de simples manuseio foi atendida. Todas as *Sprints* planejadas foram concluídas, todas tiveram sucesso em seu desenvolvimento, teste e entrega contínua, conforme o que é proposto pela metodologia SCRUM.

Um dos desafios foi a otimização do tempo para que cada história fosse fielmente entregue no tempo proposto, pois em alguns momentos surgiram erros que utilizavam um tempo excessivo para resolução, comprometendo parte dos dias planejados para a entrega de uma determinada história.

A utilização de algumas das tecnologias, como *Hibernate*, *Materialize*, *Maven* e *Primefaces* foram aprendidas em paralelo ao desenvolvimento e de acordo com a necessidade.

As matérias que tiveram maior influência e agregaram maior conhecimento para o processo de desenvolvimento de todo o processo de criação da aplicação, foram:

- **Linguagem de Programação**: foi possível aplicar os conceitos e técnicas da programação utilizando a linguagem Java, juntamente com Desenvolvimento de Sistemas *Web*, que aprofundou conhecimentos da linguagem especificamente para

o desenvolvimento *web*, juntamente com outras linguagens e tecnologias como Javascript, JSF, HTML e CSS.

- Engenharia de Software: possibilitou todo o planejamento das funcionalidades que poderiam ser desenvolvidas, utilizando artefatos como diagramas e técnicas para coleta de requisitos. Também ajudou ao empregar o uso de Metodologias Ágeis, especificamente o uso do SCRUM como modelo de desenvolvimento.
- Banco de Dados: utilizado para a criação de toda estrutura de dados da aplicação e diagrama de relacionamento.
- Segurança da Informação: destacou a importância de garantir a segurança e privacidade das informações dos usuários assim como a validação de dados prevenindo erros de digitação do usuário.

Trabalhos Futuros

Quanto a melhorias na aplicação algumas funcionalidades foram pensadas para torná-lo mais completo, útil e visando sempre proporcionar satisfação do usuário, seja na experiência ao utilizar a aplicação, ou na eficiência do serviço prestado.

Algumas das funcionalidades são:

- Alteração da senha ou envio de uma nova senha caso o usuário tenha esquecido a anterior.
- Relatórios sobre as reservas, que apresentariam detalhadamente o padrão de reservas, como por exemplo, quais salas mais reservadas, quais equipamentos mais utilizados, para fornecimento de dados aos administradores das empresas, para garantirem também que a utilização da reserva e da sala seja a melhor possível.
- Mapa das salas, ao ser visualizado, apresenta quais salas existem e qual local do prédio as mesmas estão localizadas.
- Definição da quantidade de equipamentos, inclusão de um novo campo no cadastro de equipamentos, que informa a quantidade que a empresa possui.
- Mostrar as salas disponíveis em determinada data pré-selecionada pelo usuário.

Referências

- Apache (2018). Welcome to apache maven. Disponível em: <<https://maven.apache.org/>> Acesso em 08 de Julho de 2018.
- Bizly (2018). Disponível em: <<https://www.bizly.com/>> Acesso em 08 de Agosto de 2018.
- Caelum (2018). Lab. java com testes, jsf e design patterns - capítulo 7.2 - características do jsf. Disponível em: <<https://www.caelum.com.br/apostila-java-testes-jsf-web-services-design-patterns/introducao-ao-jsf-e-primefaces/7-2-caracteristicas-do-jsf>> Acesso em 08 de Julho de 2018.
- Cohn, M. (2011). *Desenvolvimento de Software com Scrum. Aplicando Métodos Ágeis com Sucesso (Em Portuguese do Brasil)*. Bookman - Grupo A.
- Cruz, F. (2018). *Scrum e Agile em projetos: guia completo*. Brasport, 2a ed.
- Deskovery (2018). Disponível em: <<https://coworkingbrasil.org/news/aplicativo-reservar-salas-coworking/>> Acesso em 20 de Junho de 2018.

Izalmo (2018). Artigo desenvolvendo com hibernate. Disponível em: <<https://www.devmedia.com.br/desenvolvendo-com-hibernate/14756>> Acesso em 08 de Julho de 2018.

Materialize (2018). Disponível em: <<https://materializecss.com/about.html>> Acesso em: 05 de Agosto de 2018.

Sommerville, I. (2011). *Engenharia de Software*. Pearson, 9a ed.

Stallings, W. (2014). *Criptografia E Segurança De Redes*. Pearson Brasil, 6a ed.

SUAP (2018). Disponível em: <<https://suap.ifsp.edu.br>> Acesso em 08 de Maio de 2018.