



Universidade Federal de Pernambuco
Centro de Informática

Graduação em Ciência da Computação

**Reviewer: Uma Plataforma Web
Colaborativa para Revisão da
Completeness de Planos Experimentais**

Erick Lucena Palmeira Silva

Trabalho de Graduação

Recife
10 de julho de 2017

Universidade Federal de Pernambuco
Centro de Informática

Erick Lucena Palmeira Silva

**Reviewer: Uma Plataforma Web Colaborativa para Revisão
da Completude de Planos Experimentais**

Trabalho apresentado ao Programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: *Sérgio Castelo Branco Soares*
Co-orientadora: *Liliane Sheyla da Silva Fonseca*

Recife
10 de julho de 2017

*Para meus pais, José Edson Palmeira Silva e Edna Maria
de Lucena Silva, que fizeram tudo que lhes era possível
para que eu chegasse até aqui. Muito obrigado!*

Agradecimentos

Agradeço aos meus pais, José Edson Palmeira Silva e Edna Maria de Lucena Silva, que me deram uma vida inteira de amor e apoio. Que sempre fizeram o possível, e as vezes tentaram o impossível, para que eu tivesse a oportunidade de sempre continuar meus estudos. Não teria conseguido chegar até aqui sem vocês. Muito obrigado!

Aos meus avós Maria do Carmo Palmeira e Edson Araújo Silva, que sempre estiveram presentes em minha vida, e com quem vivi durante meus anos de graduação.

À minha namorada Daniele Passos, pelo amor, paciência, apoio e incentivo, especialmente nesses últimos meses. Tenha certeza que foste essencial nesse grande marco da minha vida acadêmica.

Aos professores e funcionários do Centro de Informática, especialmente a Adriano Sarmiento, com quem pude conviver e aprender através da monitoria de IP-EC, e Sérgio Soares, que me possibilitou grandes oportunidades de aprendizado durante a minha vida acadêmica e aceitou a tarefa de me orientar nessa reta final da graduação.

À Liliane Fonseca, que assumiu o papel de co-orientadora deste trabalho, e que esteve sempre presente e disposta a me ajudar onde fosse possível.

À todos os amigos que tive oportunidade de fazer durante esses anos dentro do Centro de Informática, que tornaram os dias na faculdade bem melhores e divertidos: Alexandre Cisneiros, André Carneiro, Arley Ristar, Caio Leão, Camila Nery, Djeefther Souza, Daniel Oliveira, Deyvson Lázaro, Eduardo Simões, Eduardo Ximenes, Eric Freitas, Ezequiel Matos, Felipe Nunes, Graciliano Galindo, Isabelly Damascena, Joselito Junior, João Oliveira, João Paulo Lins, Luana Martins, Marina Barros, Pedro Dias, Pedro Rossi, Ricardo Robson, Rodrigo Lopes, Rodrigo Oliveira, Victor Pimenta e tantos outros. Alguns de vocês eu tenho a oportunidade de falar quase que diariamente, com outros infelizmente perdi o contato, mas todos tiveram importância e deixaram suas marcas.

Aos amigos que tive a oportunidade de fazer durante o Ciência sem Fronteiras, e que estavam presentes em uma parte muito importante da minha vida: Luciana Veloso, minha irmã maranhense e eterna *beer buddy*; e a união de pessoas mais diferentes que deu mais certo: Amanda Barros, Ana Junqueira, César Goersch, Eduardo Rodrigues, Jéssica Torres, Felipe Luna, Mayara Bárbara, Vanessa Carneiro. O intercâmbio não teria sido o mesmo sem vocês.

E por último, aos meus colegas de trabalho na Suati: André Pimentel, Hélio Brito, Luiz Eduardo, Thayssa Lacerda e Thiago Nascimento, por todo o *bullying* motivacional para que eu me formasse.

A strong man doesn't need to read the future, he makes his own.
—SOLID SNAKE (Metal Gear Solid)

Resumo

É possível dizer que, na área da engenharia de software, planos experimentais bem planejados são a receita para se obter resultados de experimentos bem sucedidos. Embora seja possível encontrar ferramentas que auxiliam pesquisadores a reportarem seus experimentos para publicações científicas, não é comum encontrar estudos cujo objetivo é avaliar os protocolos de estudo em relação à sua qualidade científica e completude. Esse trabalho apresenta o resultado da implementação de um sistema cujo objetivo é auxiliar pesquisadores inexperientes a criarem seus planos experimentais. Este sistema possibilita a revisão de planos experimentais com intuito de avaliar se os mesmos produziram um plano experimental completo, que inclui todos os possíveis fatores para minimizar eventuais problemas.

Palavras-chave: Engenharia de Software, Plano Experimental, Experimentos, Revisão, Completude

Abstract

It is possible to say that well-planned experimental plans are recipes for successful experiments in software engineering. Although it is possible to find tools that help researchers to report their experiments to scientific publications, it is not common to find studies which the main objective is to evaluate study protocols in relation to their scientific quality and completeness. This work presents the result of the implementation of a system which the main goal is to help inexperienced researchers to create their own experimental plans. This system allows reviews of experimental plans with the intention of evaluating whether they have produced a complete experimental plan, which includes all possible factors to minimize possible problems.

Keywords: Software Engineering, Experimental Plan, Experiments, Review, Completeness

Sumário

1	Introdução	1
1.1	Motivação	1
1.2	Objetivo	1
1.3	Estrutura do documento	2
2	Contexto	3
2.1	Engenharia de Software	3
2.2	Engenharia de Software Experimental	3
2.3	Plano experimental	4
3	Visão geral do sistema	5
3.1	Casos de uso	5
3.2	Arquitetura do sistema	6
4	Implementação	9
4.1	Definição de entidades e camada de dados	9
4.2	Camada de negócios e interface com o usuário	12
5	Resultados	15
5.1	Telas de acesso	15
5.2	Telas de administração	16

5.2.1	Criação de usuário	17
5.2.2	Gerenciamento de usuários	17
5.2.3	Criação de grupo	17
5.2.4	Gerenciamento de grupos	18
5.3	Telas de planos experimentais	19
5.3.1	Criação de plano experimental	19
5.3.1.1	Criação de plano com guia	19
5.3.1.2	Criação de plano sem guia	20
5.3.2	Gerenciamento de planos experimentais	21
5.3.3	Solicitação de revisão	23
5.3.4	Visualização do resultado da revisão	23
5.4	Telas de revisão	24
5.4.1	Gerenciamento de revisões	25
5.4.2	Revisão de plano experimental	25
5.5	Telas de grupos	26
5.6	Tela de <i>profile</i>	27
6	Conclusão	29
6.1	Trabalhos futuros	29
A	Lista de casos de uso	31
B	Ferramentas utilizadas	33
B.1	Eclipse	33
B.2	Maven	33
B.3	Git	33

B.4 DbVisualizer

34

Lista de Figuras

3.1	Arquitetura do sistema	6
4.1	Diagrama de classes	9
4.2	Modelo ER	10
4.3	Trecho de código da classe <i>UserController</i>	11
4.4	Trecho de código da classe <i>HibernateUserRepository</i>	11
4.5	Arquitetura do <i>framework</i> JSF	12
4.6	Módulos do sistema	13
5.1	Tela de login	15
5.2	Telas de recuperação de senha	16
5.3	Tela principal de um administrador do sistema	16
5.4	Tela de criação de usuário	17
5.5	Gerenciamento de usuários	18
5.6	Criação de grupos	18
5.7	Gerenciamento de grupos	19
5.8	Tela de criação de planos experimentais	20
5.9	Tela de criação de planos experimentais genéricos	20
5.10	Gerenciamento de planos experimentais	21
5.11	Planos prontos para revisão	21

5.12	Revisores disponíveis	22
5.13	Lista de planos arquivados	22
5.14	Planos em colaboração	23
5.15	Tela de criação de <i>reviews</i>	24
5.16	Tela de relatório da revisão	24
5.17	Gerenciamento de revisões	25
5.18	Tela de gerenciamento de revisões após aceite	26
5.19	Revisão de plano experimental	26
5.20	Tela de grupos	27
5.21	Tela de gerenciamento de um grupo	27
5.22	Tela de informações do usuário	28

CAPÍTULO 1

Introdução

Neste capítulo são apresentados a motivação para a realização deste trabalho, os objetivos, a metodologia e a estrutura desse trabalho, a fim de facilitar o entendimento do leitor.

1.1 Motivação

A engenharia de software é uma área da ciência da computação que estuda os procedimentos que compõe o processo de produção de software (concepção, especificação, desenvolvimento e manutenção), com o objetivo de tornar o processo mais sistemático, científico e quantificável. [Som07].

Em meio a evolução da engenharia de software, surgiu a engenharia de software experimental, que tem como objetivo avaliar aplicações de engenharia de software, utilizando-se de métodos empíricos. Para Travassos et al. [TGA02], uma das melhores abordagens para experimentação na engenharia de software é o método experimental. Isto se deve pelo fato que o mesmo se utiliza de procedimentos empíricos que medem, analisam, avaliam e re-experimentam o software. O principal objetivo do método empírico é a melhoria gradual a cada iteração dos experimentos.

Assim, tendo em vista a importância da execução dos experimentos, é possível dizer que o planejamento dos mesmos tem uma grande importância no aumento das chances de sucesso de todas as etapas envolvidas nesse processo.

1.2 Objetivo

Este trabalho de graduação possui como objetivo principal a modelagem e desenvolvimento de uma plataforma web colaborativa, cuja função é dar suporte aos pesquisadores que planejam experimentos controlados. Esse suporte acontece através de um processo de revisão de completude dos planos experimentais, garantindo que eles contenham todos os fatores possíveis para minimizar os problemas que podem surgir durante a execução do experimento.

Esta plataforma combina a automatização do instrumento de revisão de planos experimentais e sugestões de mecanismos que dão suporte a atividades experimentais desenvolvidos recentemente em uma pesquisa de doutorado [Fon16]. Além disso a plataforma permite a construção de um repositório de planos experimentais permitindo o acesso de pesquisadores colaboradores a visualização de planos experimentais construídos anteriormente por colaboradores parceiros favorecendo assim o amadurecimento da área de engenharia de software experimental.

1.3 Estrutura do documento

O restante desse trabalho é dividido nos seguintes capítulos:

- Capítulo 2: Contexto
- Capítulo 3: Visão geral do sistema
- Capítulo 4: Implementação
- Capítulo 5: Resultados
- Capítulo 6: Conclusão

CAPÍTULO 2

Contexto

Neste capítulo são apresentados alguns conceitos pertinentes ao sistema criado, de forma a contextualizar o leitor e dar significado a termos recorrentes neste trabalho.

2.1 Engenharia de Software

O termo surgiu na década de 1960, juntamente com os sistemas de tempo compartilhado. Durante o desenvolvimento dos primeiros mainframes a utilizarem esse conceito, notou-se que a dificuldade de implementar o sistema era bem mais difícil que o previsto, dada a complexidade do problema em comparação aos sistemas de processamento em lote. Essas dificuldades levaram à organização de uma conferência pela OTAN em 1968, onde essas dificuldades foram discutidas e surgiu o termo 'engenharia de software' [Wir08].

A engenharia de software se preocupa com todos os passos do desenvolvimento de software, desde os primeiros estágios da modelagem de um sistema, seu desenvolvimento, testes e até o processo de manutenção, que acontece após o sistema entrar em produção. Isso implica em não só se preocupar com a parte técnica do processo, mas também com atividades de gerenciamento de projeto e o desenvolvimento de teorias, ferramentas e metodologias que possam auxiliar no decorrer do processo.

2.2 Engenharia de Software Experimental

Em meio a evolução da engenharia de software, surgiu a engenharia de software empírica, que possui como objetivo estudar como os engenheiros de software executam o seu trabalho, e aprender acerca da realização de estudos experimentais. É possível destacar quatro métodos relevantes para a condução de estudos empíricos dentro da Engenharia de Software : científico, de engenharia, empírica e analítica. O método empírico, que tradicionalmente é mais utilizado nas ciências sociais e na psicologia, mostrou-se adequado para o uso na engenharia de software, cujos procedimentos são influenciados pelo comportamento das pessoas que participam do processo de desenvolvimento [WRH⁺12].

2.3 Plano experimental

Um plano experimental é um documento que contém um guia para a condução de um experimento. É importante salientar que o conteúdo de um plano experimental não é um relatório acerca da sua execução, mas sim o passo a passo de como o experimento descrito por ele deve ser executado. Nele devem estar contidas todas as informações necessárias para a preparação de um experimento, o que deve ser feito durante sua execução e como deve se suceder o processo de análise de dados e definição de resultado.

Dado o número de variáveis envolvidas, pode ser difícil para um pesquisador da área de engenharia de software preparar um plano experimental de qualidade, especialmente se o mesmo for um pesquisador com pouca experiência. Por isso, a existência de ferramentas que possam auxiliar neste processo é de extrema importância.

Visão geral do sistema

A *Reviewer* é uma plataforma web que tem como objetivo auxiliar pesquisadores nas atividades de planejamento e revisão de planos experimentais. O sistema será especialmente útil para pesquisadores que não possuem muita experiência no processo, provendo um guia para a construção de planos e a oportunidade de que pesquisadores mais experientes possam revisar e fornecer *feedbacks* sobre os planos criados e submetidos para revisão.

O instrumento utilizado como base para a criação dessa plataforma foi desenvolvido recentemente em uma pesquisa de doutorado [Fon16], e tem como objetivo dar suporte aos pesquisadores que planejam experimentos controlados, auxiliando os mesmos tanto no processo de concepção dos planos, bem como durante a revisão dos mesmos.

3.1 Casos de uso

Durante a etapa de modelagem do sistema, o primeiro passo foi a definição dos casos de uso. Ter uma lista dos casos de uso do sistema é necessário para se ter uma visão geral do que se espera da ferramenta, podendo assim avaliar se o que está sendo implementado atende de forma satisfatória as necessidades previstas.

Entre os casos de uso levantados nesta etapa, dois deles podem ser destacados como sendo o *core* da sistema, e são detalhados logo abaixo. Os demais casos de uso podem ser encontrados no apêndice A deste documento.

Nome	Manter planos experimentais
Ator	Autor
Descrição	Permite ao autor incluir, excluir, alterar informações dos planos experimentais.

Esse caso de uso define as principais ações de um autor dentro do sistema. Um autor deve ter em mãos funcionalidades que permitam a ele criar planos experimentais, editar planos previamente criados e submeter os mesmos para revisão. Além disso, o gerenciamento dessas opções deve ser de fácil acesso ao autor.

Nome	Revisar Planos experimentais
Ator	Revisor
Descrição	A plataforma deve exibir para o revisor o conjunto de questões e itens descritos na ferramenta de revisão de completude de planos experimentais.

Este caso de uso representa a principal funcionalidade do sistema. O sistema deve permitir que usuários possam revisar os planos criados. Para isso, deve ser utilizada a ferramenta de revisão de completude definida por Fonseca [Fon16]. Essa ferramenta possui 31 itens, divididos em 9 categorias, que fornecem aos revisores um guia de revisão, facilitando seu trabalho e tornando o processo mais ágil.

3.2 Arquitetura do sistema

A plataforma implementada durante esse trabalho é um sistema web, cuja arquitetura está disponível na Figura 3.1.

A plataforma é composta por quatro componentes principais: interface com o usuário, servidor web, *framework* ORM¹ e banco de dados. Foi escolhido usar um *framework* para possibilitar uma abstração maior da camada de dados, possibilitando uma maior agilidade durante o desenvolvimento, manutenção e refatoração de código, além de tornar o sistema independente da tecnologia de SGBD² utilizada.

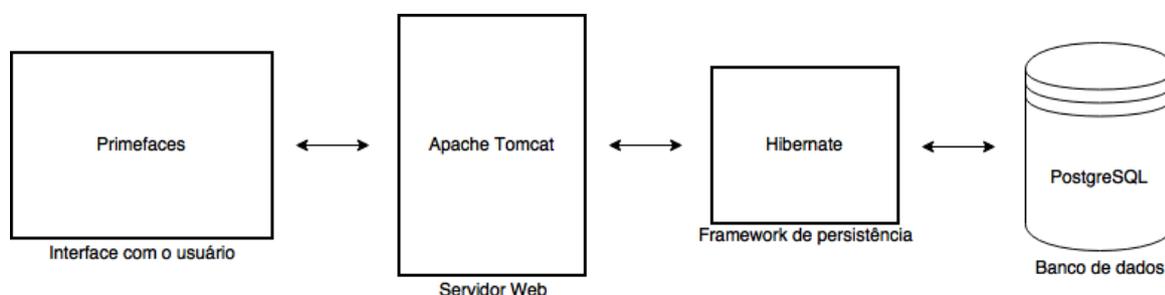


Figura 3.1 Arquitetura do sistema

As tecnologias utilizadas durante o desenvolvimento foram escolhidas baseados em parte na familiaridade do autor como as mesmas, parte nos benefícios trazidos pelas tecnologias que podem ser vistos a seguir.

É desejável que o sistema seja de fácil instalação, *deployment*, manutenção e evolução. Java se enquadra nesse perfil, sendo amplamente usada para aplicações web, móveis e *desktop*. A implementação foi feita voltada a ser executada em Apache Tomcat, servidor de código-aberto

¹*Object-Relational Mapping*, ou em português, Mapeamento Objeto-Relacional

²Sistema de Gerenciamento de Banco de Dados

que está a mais de 18 anos no mercado. A camada de dados utiliza-se de JPA³, *Hibernate* e *PostgreSQL*⁴. A utilização de JPA em conjunto com *Hibernate* foi feita buscando uma maior abstração da comunicação da aplicação e o SGBD, provendo agilidade e estabilidade no desenvolvimento, manutenção e refatoração da camada de dados. Na interface com o usuário, foi escolhido o uso de *PrimeFaces*⁵, um *framework* gráfico baseado em JSF⁶, de forma a facilitar a implementação das telas do sistema.

As ferramentas utilizadas no desenvolvimento em conjunto as tecnologias previamente citadas podem ser encontradas no apêndice B.

³Java Persistence API, interface para manipulação de dados relacionais em Java

⁴<https://www.postgresql.org>

⁵<https://www.primefaces.org>

⁶JavaServer Faces

CAPÍTULO 4

Implementação

Após a definição dos casos de uso e escolha das tecnologias, a implementação da plataforma web é apresentada. Neste capítulo são vistas as principais etapas do processo de desenvolvimento do sistema e seus resultados.

4.1 Definição de entidades e camada de dados

Levando-se em consideração os requisitos obtidos através dos casos de uso, primeiramente, as entidades manipuladas pelo sistema foram definidas e a camada de dados da aplicação implementada. Nesta etapa foram criadas as classes que representam todas as entidades manipuláveis dentro da aplicação. O diagrama dessas classes pode ser visto na Figura 4.1.

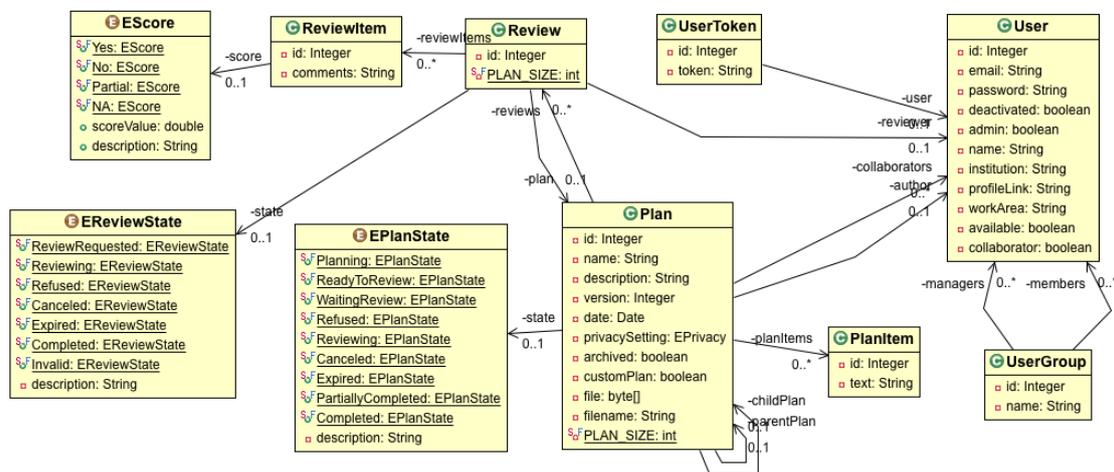


Figura 4.1 Diagrama de classes

Em sequência à criação das classes, foi feita a integração com o *Hibernate* e *JPA*. Através de *Java Annotations*¹, foram definidas as classes, atributos e relacionamentos que devem ser persistidos pelo *framework*. Devido a compatibilidade do *Hibernate* com uma gama de SGBDs²,

¹https://en.wikipedia.org/wiki/Java_annotation

²https://developer.jboss.org/wiki/SupportedDatabases2?_sscc=t

essa etapa torna possível a configuração do sistema em desenvolvimento para trabalhar com qualquer um dos SGBDs suportados. Terminadas as anotações, DAOs³ foram criados através da utilização de JPA.

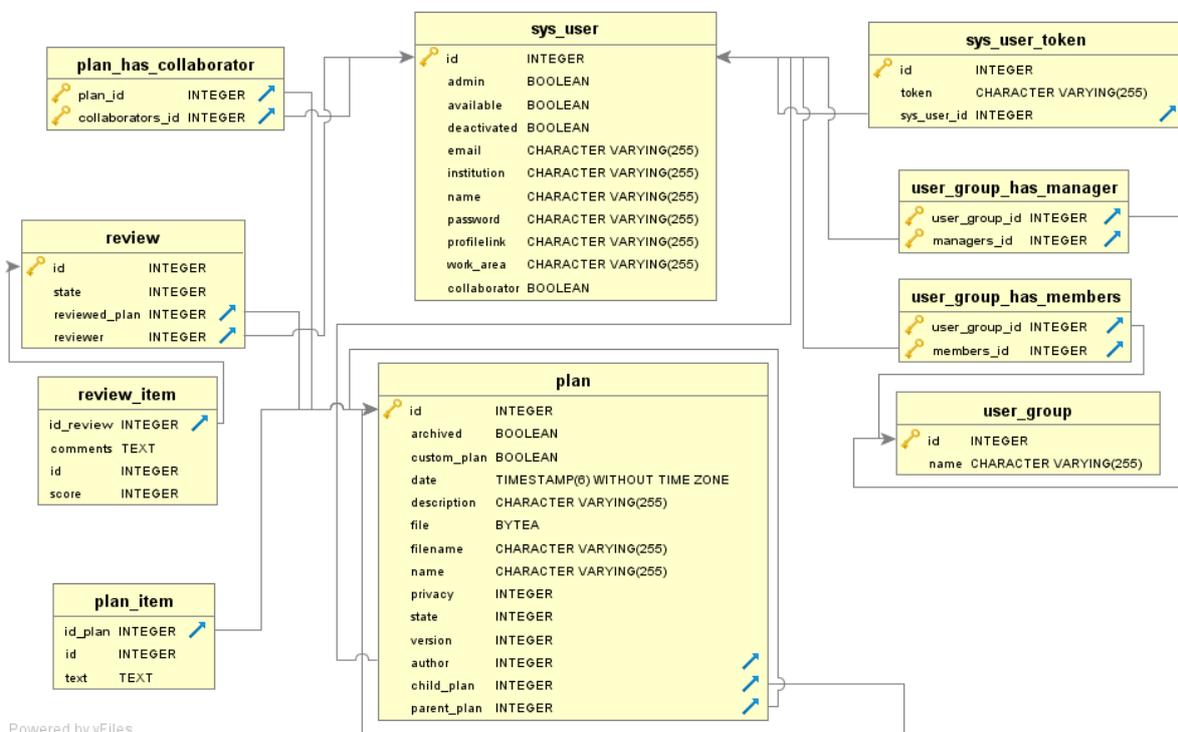


Figura 4.2 Modelo ER

Após finalizada a integração e configurada a conexão com PostgreSQL (SGBD escolhido para este trabalho), o *Hibernate* será capaz de analisar as classes e gerar o modelo ER adequado para o SGBD utilizado. O resultado dessa etapa pode ser visto na Figura 4.2.

Adicionalmente foram criadas interfaces de repositório com o intuito de abstrair a conexão entre negócio-dados, de forma que as regras de negócio do sistema não fiquem acopladas com o provedor de dados escolhido.

Para exemplificar o comportamento da camada de dados, podemos usar como exemplo a criação de um novo usuário dentro do sistema. Na camada de negócios, a classe *UserController* (Figura 4.3) possui um *IUserRepository*, interface que modela as ações desempenhadas pelo repositório de usuários.

Para inserir um usuário esse *controller* recebe um objeto da classe *User*. Após executar as validações necessárias, o objeto é então repassado a interface de repositório que foi instanciada anteriormente pela classe *RepositoryFactory*.

³Data Access Object, ou em português: objeto de acesso a dados. Esse padrão tem como objetivo isolar a lógica de acesso ao SGBD da camada de negócios

```
public class UserController {  
  
    private IUserRepository repository;  
    private EmailController emailController;  
  
    protected UserController() {  
        repository = RepositoryFactory.createUserRepository();  
        emailController = ControllerFactory.createEmailController();  
    }  
  
    public User insert(User user) throws EmailAlreadyInUseException {  
        validateUserDuplicity(user);  
        String password = PasswordGenerator.createPassword();  
        user.setPassword(CryptAgent.encryptPassword(password));  
        user = repository.insert(user);  
        emailController.sendAccountCreatedEmail(user, password);  
        return user;  
    }  
}
```

Figura 4.3 Trecho de código da classe *UserController*

Quem recebe o objeto é a classe *HibernateUserRepository* (Figura 4.4) que utiliza o DAO de usuário, que fica responsável pela comunicação com o SGBD, para executar a persistência do objeto.

Como o *controller* não está diretamente conectado com a implementação do repositório, é obtido um maior nível de desacoplamento entre as camadas. Dessa forma, caso seja necessário utilizar uma nova tecnologia em detrimento do *Hibernate*, desde que o novo repositório implemente a interface definida, não serão necessárias modificações na camada de negócio.

```
public class HibernateUserRepository implements IUserRepository {  
  
    private UserDAO dao;  
  
    public HibernateUserRepository() {  
        dao = new UserDAO();  
    }  
  
    @Override  
    public User insert(User user) {  
        JPAHelper.getInstance().beginTransaction();  
        user = dao.create(user);  
        JPAHelper.getInstance().commit();  
        return user;  
    }  
}
```

Figura 4.4 Trecho de código da classe *HibernateUserRepository*

4.2 Camada de negócios e interface com o usuário

Após a camada de dados estar implementada e funcional, o desenvolvimento da aplicação foi verticalizado. Ao invés de construir toda a camada de negócios / controle, escolheu-se dividir o desenvolvimento entre as macro-entidades do sistema (usuário, plano e *review*) de forma que, tão logo os *controllers* de uma entidade estivessem prontos, fosse iniciada a construção da interface com o usuário em sequência. Sendo o gerenciamento de usuários a parte mais simples e direta do sistema, escolhe-se por atacá-la primeiro, seguida de planos e *reviews*, nessa ordem.

Para atingir o objetivo, foi criado inicialmente um *controller*, que agrega as funcionalidades propostas pelo sistema para cada uma das macro entidades. De posse do *controller*, foram criadas as telas que possibilitavam o uso das funcionalidades implementadas.

Cada tela em JSF segue o padrão MVC⁴. A tela em .xhtml representando o *view*, o *managed bean* e os dados que o mesmo acessa representando o *model*, e o *controller* é implicitamente representado pelo próprio framework.

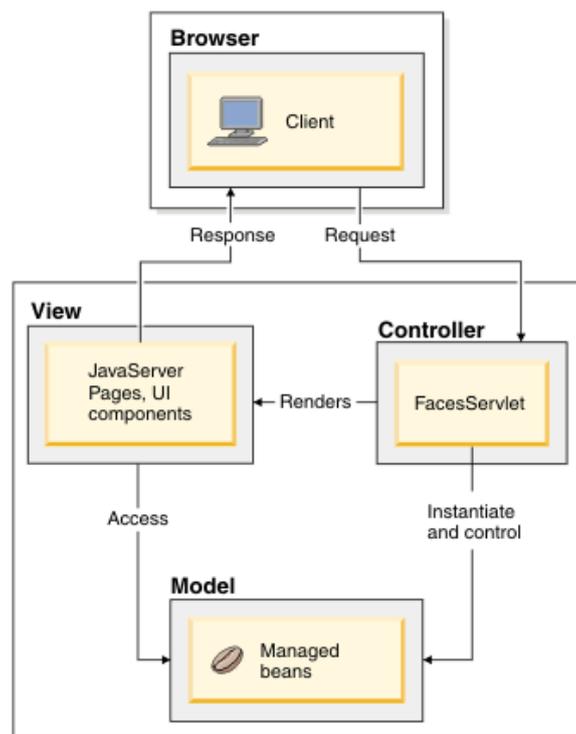


Figura 4.5 Arquitetura do *framework* JSF
Fonte: IBM Knowledge Center

⁴*Model-View-Controller* ou em português: Modelo-Visão-Controlador. Padrão de arquitetura de software que busca isolar a representação interna da informação da maneira como o usuário interage com a mesma

Essa abordagem verticalizada possibilitou um *feedback* mais rápido do funcionamento do sistema e contribuiu para o entendimento de como as duas camadas se comunicam, sendo esse aprendizado aproveitado durante o desenvolvimento de negócios/interface das entidades restantes. A estrutura resultante pode ser vista na Figura 4.6

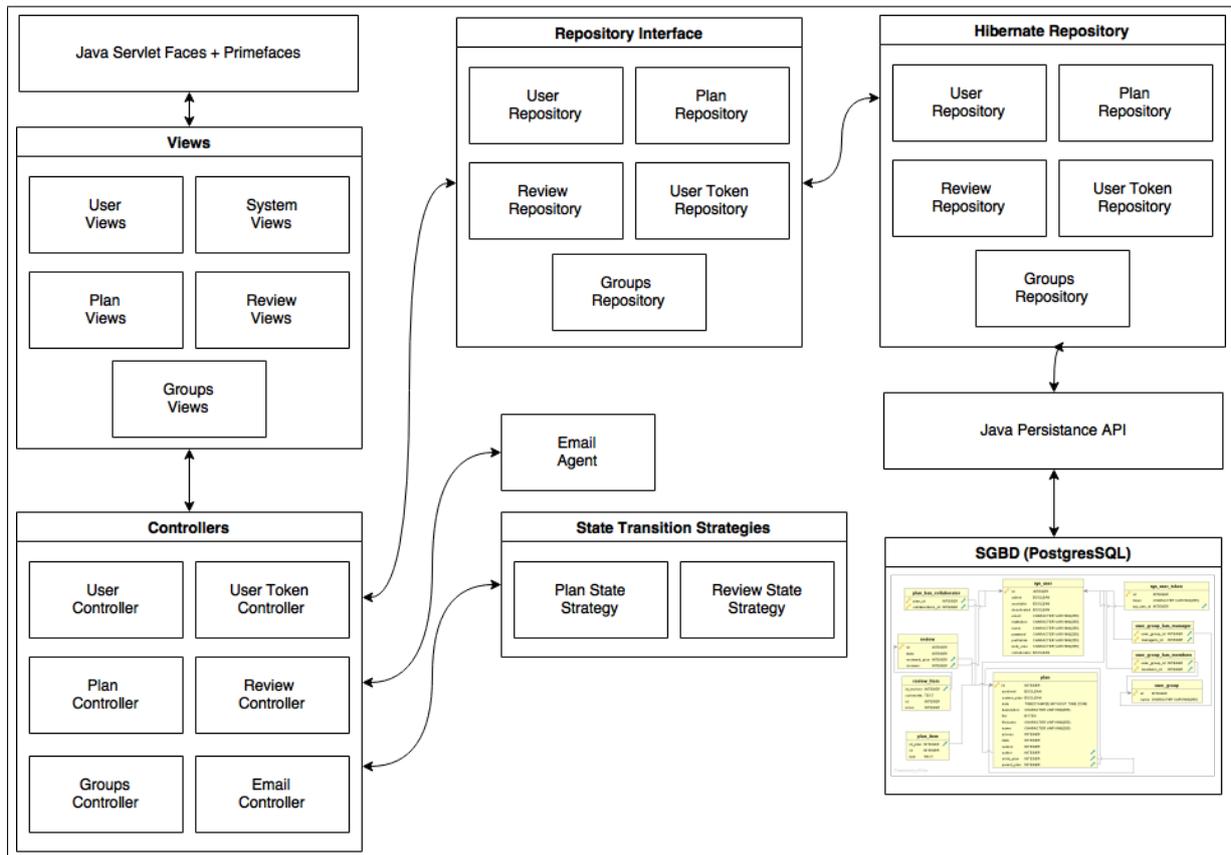


Figura 4.6 Módulos do sistema

CAPÍTULO 5

Resultados

Neste capítulo, os resultados da etapa de desenvolvimento são apresentados. Adicionalmente, as telas do sistema com suas respectivas funcionalidade são exibidas.

5.1 Telas de acesso

Os usuários do sistema serão previamente cadastrados por um administrador. Ao acessar a URL do sistema, a tela de login (Figura 5.1) será exibida ao usuário.



Figura 5.1 Tela de login

Para efetuar o *login*, o usuário deverá inserir o *email* cadastrado e senha, que serão validados pelo servidor

Caso o mesmo venha a esquecer a senha utilizada, é dada a opção de recuperação de senha (Figura 5.2). Essa recuperação é feita em duas etapas: solicitação de *token* e redefinição da senha.

Primeiro o usuário deverá inserir o email cadastrado no sistema e solicitar um *token* de recuperação. Após isso, será enviado um email ao usuário com um *token*, que poderá ser

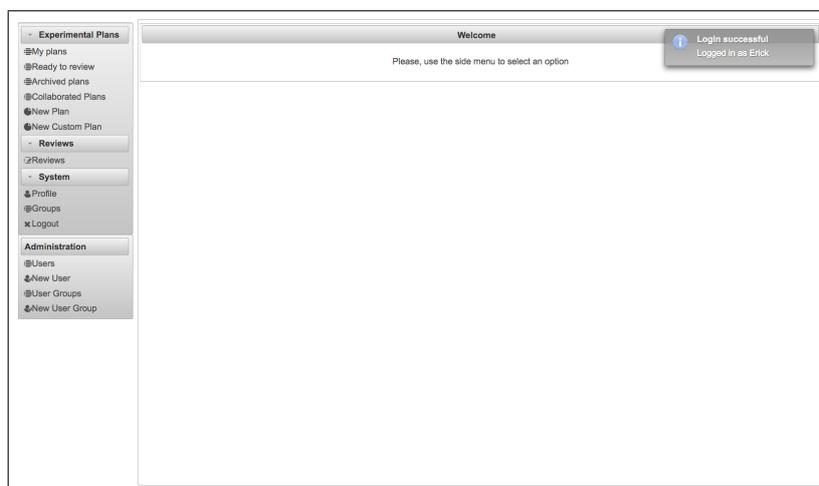
utilizado para criar uma nova senha no sistema. Através do email recebido, o usuário tem acesso a um link que o redirecionará para a página onde o mesmo pode criar uma nova senha utilizando o *token* recebido. Após criar a nova senha, o usuário pode proceder com o *login* normalmente.



The image shows two side-by-side screenshots of web forms. The left screenshot is titled 'Generate Token' and contains a single text input field labeled 'Email' and a 'Recover' button below it. The right screenshot is titled 'Recover Password' and contains three text input fields: 'Token *', 'New password *', and 'Confirm password *'. Below these fields is a 'Recover' button and a link that says 'Generate a recovery token'.

Figura 5.2 Telas de recuperação de senha

Após efetuar *login* com sucesso, o usuário será redirecionado para a *home page* do sistema, e terá acesso as funcionalidades disponibilizadas.



The image is a screenshot of a web application's administrative interface. On the left is a vertical sidebar menu with several expandable sections: 'Experimental Plans' (containing 'My plans', 'Ready to review', 'Archived plans', 'Collaborated Plans', 'New Plan', and 'New Custom Plan'), 'Reviews', 'System', 'Profile', 'Groups', and 'Logout'. Below these is an 'Administration' section with 'Users', 'New User', 'User Groups', and 'New User Group'. The main content area has a header with 'Welcome' and a sub-header 'Please, use the side menu to select an option'. A notification box in the top right corner displays a blue information icon and the text 'Login successful Logged in as Erick'.

Figura 5.3 Tela principal de um administrador do sistema

5.2 Telas de administração

Os usuários com privilégio de administrador tem acesso a funcionalidades específicas dentro do sistema, de forma que assim possam realizar o gerenciamento dos usuários cadastrados. As telas relacionadas a essas funcionalidades serão vistas a seguir.

5.2.1 Criação de usuário

Através do menu lateral, o administrador pode acessar a opção '*New User*', e será levado para a página de criação de usuário (Figura 5.4).

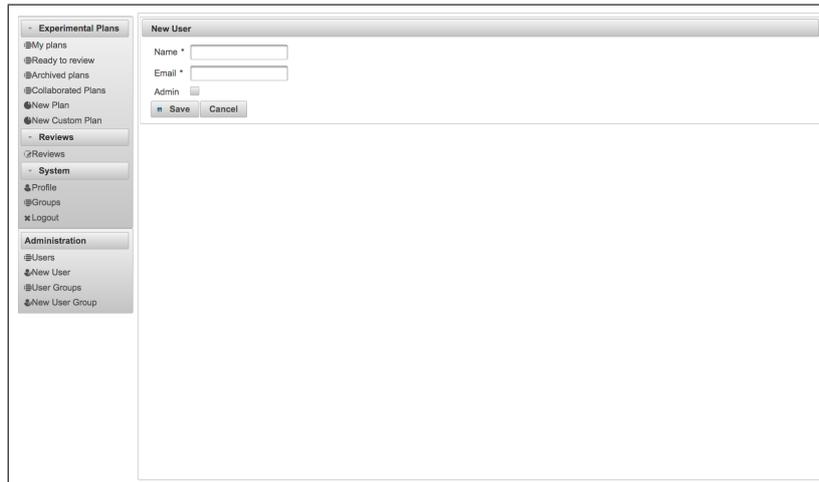
The image shows a web application interface for creating a new user. On the left is a sidebar menu with categories like 'Experimental Plans', 'Reviews', 'System', 'Profile', and 'Administration'. Under 'Administration', 'New User' is selected. The main content area is titled 'New User' and contains a form with two text input fields labeled 'Name' and 'Email', an 'Admin' checkbox, and 'Save' and 'Cancel' buttons.

Figura 5.4 Tela de criação de usuário

Após preencher os dados e confirmar a ação, o novo usuário é criado no sistema, que enviará um *email* informando o mesmo sobre sua inclusão na plataforma, como acessá-la e qual a sua senha inicial.

5.2.2 Gerenciamento de usuários

Através do menu lateral, o administrador pode acessar a opção '*Users*', e será levado para a página de gerenciamento de usuários (Figura 5.5).

Nesta tela o administrador tem acesso a listagem de usuários do sistema, e pode executar ações sobre cada um deles, como editar suas informações e desabilitar seu acesso ao sistema.

5.2.3 Criação de grupo

Através do menu lateral, o administrador pode acessar a opção '*New User Group*', e será levado para a página de criação de grupos (Figura 5.6).

Após preencher o nome do grupo, o administrador pode escolher os responsáveis pelo gerenciamento do mesmo (ao menos um deve ser selecionado) e, opcionalmente, escolher os usuários existentes que devem ser incluídos inicialmente no novo grupo.

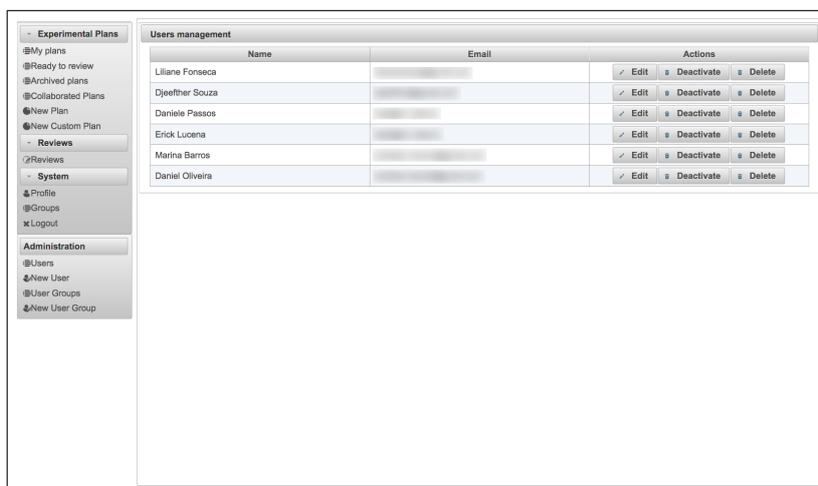


Figura 5.5 Gerenciamento de usuários

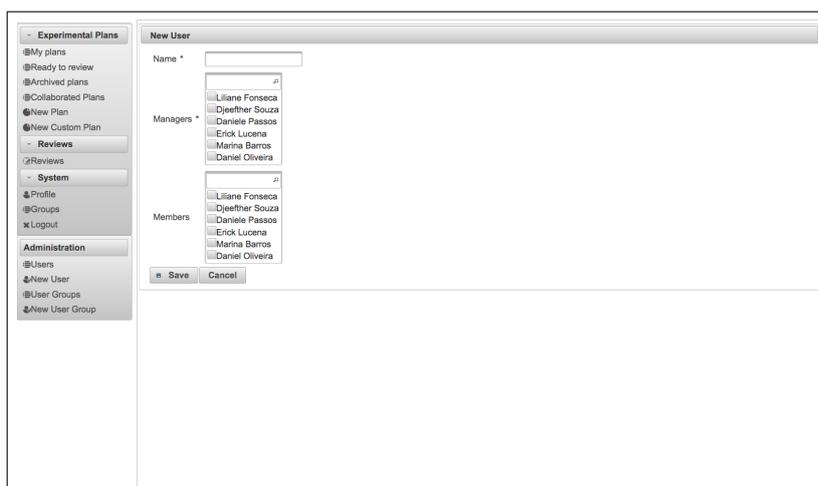


Figura 5.6 Criação de grupos

5.2.4 Gerenciamento de grupos

Através do menu lateral, o administrador pode acessar a opção *'User Groups'*, e será levado para a página de gerenciamento de grupos (Figura 5.7).

Nesta tela o administrador tem acesso a listagem dos grupos de usuários criados no sistema, e pode executar ações sobre cada um deles, como editar suas informações e fazer a gerência de seus usuários. É importante salientar que esse acesso difere da gerência de grupos por usuários sem privilégio de administração, que será visto posteriormente nesse trabalho.

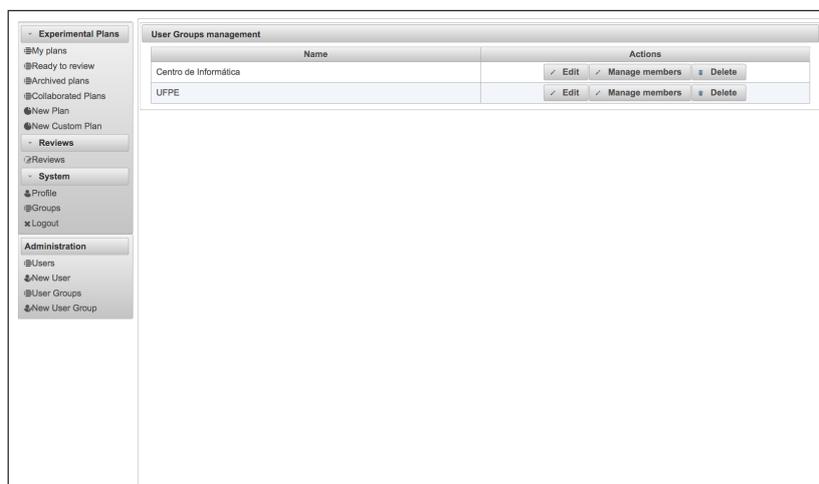


Figura 5.7 Gerenciamento de grupos

5.3 Telas de planos experimentais

Nesta seção serão exibidas as telas que permitem o uso das funcionalidades relacionadas a produção e gerenciamento de planos experimentais.

5.3.1 Criação de plano experimental

A criação de planos pode ocorrer em duas formas diferentes: utilizando os itens de revisão como guia ou fazendo o *upload* de um arquivo com o plano experimental criado. Será vista, em sequência, a criação de ambos.

5.3.1.1 Criação de plano com guia

Através do menu lateral, o usuário pode acessar a opção '*New Plan*', e será levado a tela de criação de plano experimental (Figura 5.8).

Durante a criação do plano, o usuário pode preencher os campos descritivos do plano, como nome e descrição. Neste momento ele também tem a oportunidade de escolher outros usuários da plataforma como colaboradores.

Logo abaixo, o usuário tem a disposição os itens de revisão do sistema, que nesse momento pode ser utilizado para guiá-lo na produção do plano experimental. A qualquer momento o usuário pode salvar o plano e retomar posteriormente de onde parou. Essa funcionalidade descrita nas próximas seções desse trabalho.

The screenshot shows a web interface for creating a new experimental plan. On the left is a sidebar menu with options like 'My plans', 'Ready to review', 'Archived plans', 'Collaborated Plans', 'New Plan', 'New Custom Plan', 'Reviews', 'System', 'Profile', 'Groups', and 'Logout'. The main area is titled 'New Plan' and contains several input fields: 'Title' (filled with 'Primeiro plano experimental'), 'Description' (filled with 'Breve descrição acerca do plano'), and 'Collaborators' (a list with 'Daniela Passos', 'Marina Barros', and 'Daniel Oliveira'). Below the input fields are several tabs: 'Stating the goals', 'Hypotheses, Variables, and Measurements', 'Participants', 'Experimental Materials and Tasks', and 'Experimental Design'. The 'Stating the goals' tab is selected and shows a question: '1. Are the aims clearly and precisely stated?' followed by a text input field. Below the input field, there is a list of considerations for the experiment's goals: 'A clear purpose.', 'Specific Objectives.', and 'The reasons for undertaking the experiment, clearly and explicitly stated.' A 'Hint' button is also visible.

Figura 5.8 Tela de criação de planos experimentais

5.3.1.2 Criação de plano sem guia

Através do menu lateral, o usuário pode acessar a opção 'New Custom Plan', e será levado a tela de criação de plano experimental (Figura 5.9).

The screenshot shows a web interface for creating a new experimental plan. On the left is a sidebar menu with options like 'My plans', 'Ready to review', 'Archived plans', 'Collaborated Plans', 'New Plan', 'New Custom Plan', 'Reviews', 'System', 'Profile', 'Groups', and 'Logout'. The main area is titled 'New Plan' and contains several input fields: 'Title' (filled with 'Primeiro Plano'), 'Description' (filled with 'Breve descrição acerca do plano'), and 'Collaborators' (a list with 'Daniela Passos', 'Marina Barros', and 'Daniel Oliveira'). Below the input fields, there is a 'File' section with a 'Choose' button. At the bottom, there are 'Save' and 'Cancel' buttons.

Figura 5.9 Tela de criação de planos experimentais genéricos

Durante a criação desse tipo de plano o usuário, além de preencher os campos descritivos do plano como já visto na seção anterior, deve fazer o *upload* de um documento contendo o seu plano experimental.

5.3.2 Gerenciamento de planos experimentais

O gerenciamento dos planos experimentais é feito através de quatro telas diferentes, com o intuito de distribuir melhor as ações que podem ser executadas sobre os planos e permitindo um melhor controle por parte do usuário.

A primeira tela de gerenciamento (Figura 5.10) é acessada através da opção *'My Plans'* no menu lateral. Essa tela exibe todos os planos (não arquivados) criados pelo usuário. Através dela é possível ver o estado atual de cada plan, e executar ações de gerenciamento sobre os mesmos (e.g. editar, excluir, arquivar).

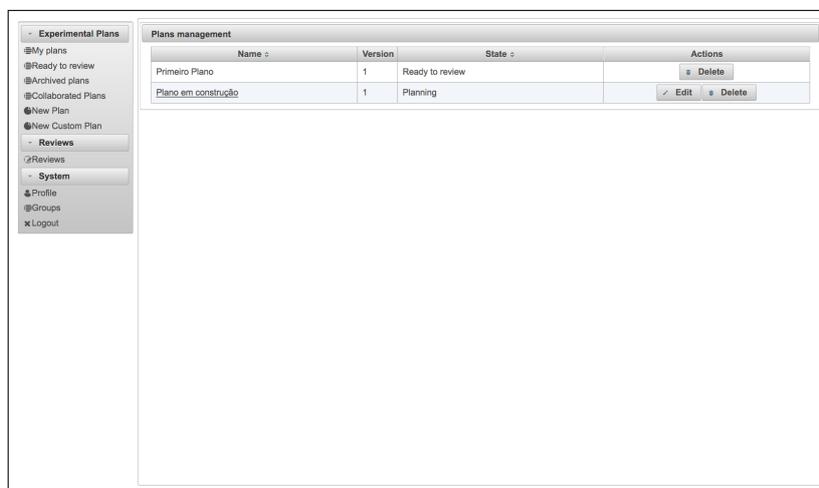


Figura 5.10 Gerenciamento de planos experimentais

Quando o usuário escolhe a opção *'Ready to Review'* no menu lateral, ele tem acesso a tela (Figura 5.11) que lista dos seus planos que se encontram prontos para serem revisados.

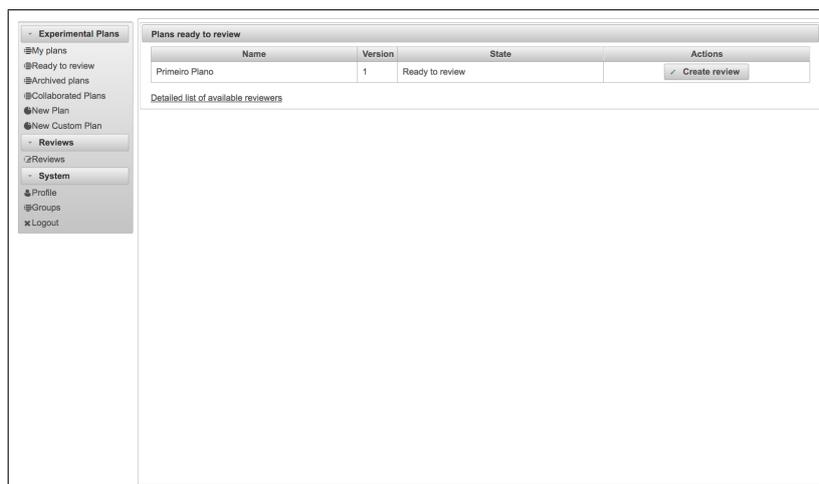
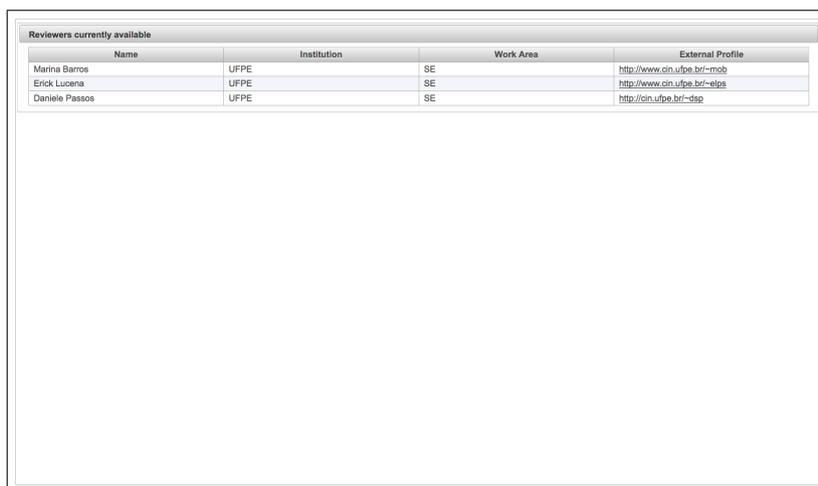


Figura 5.11 Planos prontos para revisão

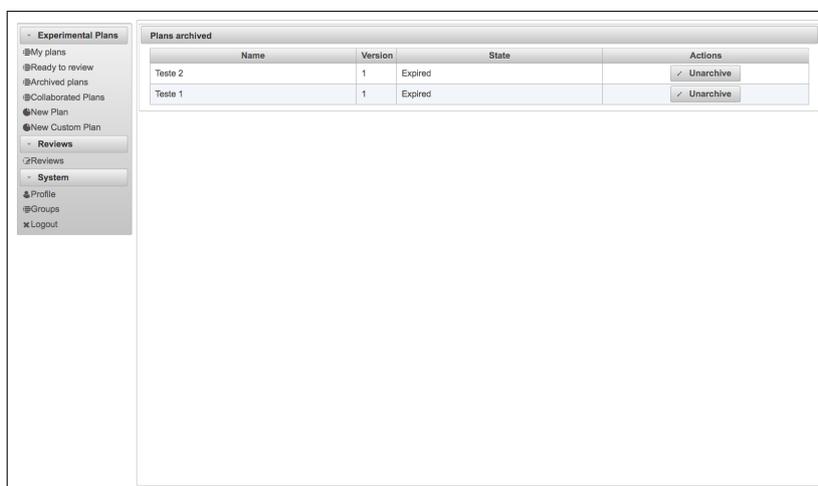
Nesta tela ele também tem acesso a uma listagem dos usuários da plataforma que estão disponíveis para revisão no momento (Figura 5.12) clicando na opção '*Detailed list of available reviewers*'. Essa lista contém informações como área de trabalho e perfil do revisor, possibilitando ao usuário escolher qual o revisor mais adequado para o plano experimental a ser revisado.



Name	Institution	Work Area	External Profile
Marina Barros	UFPE	SE	http://www.cin.ufpe.br/~mob
Erick Luciana	UFPE	SE	http://www.cin.ufpe.br/~elips
Daniele Passos	UFPE	SE	http://cin.ufpe.br/~dps

Figura 5.12 Revisores disponíveis

Por último, o usuário tem disponível uma tela que lista os planos arquivados (Figura 5.13), que pode ser acessada através da opção '*Archived plans*' no menu lateral.

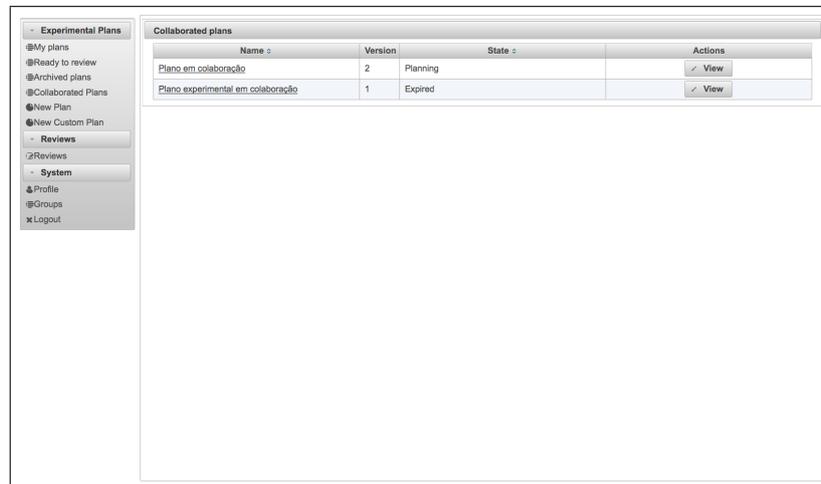


Name	Version	State	Actions
Teste 2	1	Expired	<input type="button" value="Unarchive"/>
Teste 1	1	Expired	<input type="button" value="Unarchive"/>

Figura 5.13 Lista de planos arquivados

O arquivamento de um plano só pode ocorrer quando o mesmo se encontra num estado final, e tem valor puramente organizacional. Se o usuário necessitar executar uma nova ação sobre esse plano, pode desarquivar o mesmo, e voltar a manipulá-lo através da tela principal de gerenciamento.

Por último, quando o usuário escolhe a opção '*Collaborated plans*' no menu lateral, ele tem acesso a tela (Figura 5.14) que lista os planos nos quais o usuário foi marcado como colaborador.



Name :	Version	State :	Actions
Plano em colaboração	2	Planning	<input type="button" value="View"/>
Plano experimental em colaboração	1	Expired	<input type="button" value="View"/>

Figura 5.14 Planos em colaboração

Nesta tela é dada ao usuário a opção de visualizar o progresso da criação do plano experimental e, posteriormente, acompanhar o progresso de revisão através da tela de *report*.

5.3.3 Solicitação de revisão

Após completar a criação de um plano experimental, o autor pode solicitar a revisão do mesmo para um dos revisores disponíveis no sistema. Para isso, ele deve usar a opção '*Create review*', disponível na tela de planos prontos para revisão (Figura 5.11). Ao fazê-lo, o usuário será redirecionado para a tela de criação de reviews (Figura 5.15).

Para criar o review, o autor será solicitado a escolher um ou mais revisores dentre a lista de disponíveis, bem como definir uma data máxima para a revisão do plano experimental. Após a solicitação da revisão, os revisores selecionados pelo autor serão notificados via email, podendo então acessar o sistema para aceitar ou rejeitar a solicitação.

5.3.4 Visualização do resultado da revisão

Durante e após o processo de revisão, o autor pode acessar seu progresso, visualizar comentários e *score* atual. Isso pode ser feito através da tela de gerenciamento de planos experimentais (Figura 5.10), escolhendo a opção '*Report*'. O autor será redirecionado para uma tela contendo um relatório sobre a revisão do plano experimental selecionado (Figura 5.16).

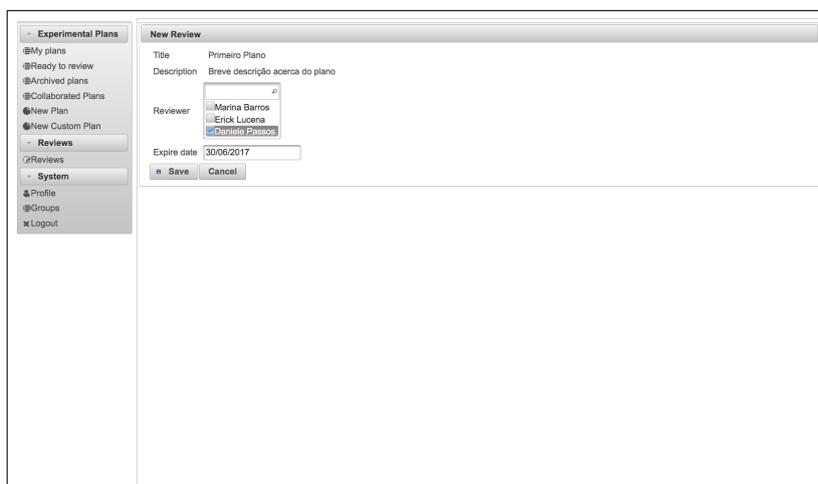


Figura 5.15 Tela de criação de *reviews*

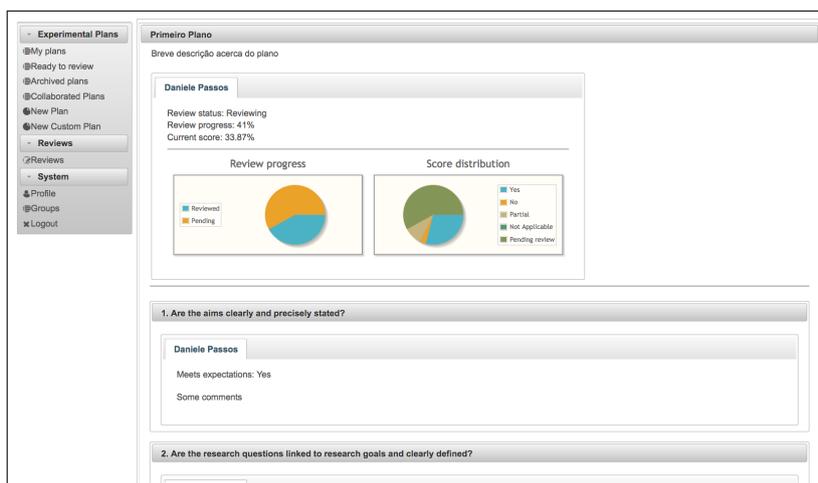


Figura 5.16 Tela de relatório da revisão

No relatório o usuário encontrará detalhes sobre o progresso da revisão (e.g. número de itens revisados, score atual), a distribuição de *score* nos itens já revisados, *score* individual e comentários de cada item de revisão. Caso o plano esteja sendo revisado por mais de uma pessoa, é possível visualizar cada informação separada por revisor.

5.4 Telas de revisão

Nesta seção serão apresentadas as telas que permitem o uso das funcionalidades relacionada ao processo de revisão de planos experimentais.

5.4.1 Gerenciamento de revisões

Os usuários do sistema pode gerenciar revisões que lhe foram solicitadas através da opção 'Reviews' no menu lateral. O usuário será redirecionado para a tela que lista todas as revisões solicitadas ao mesmo (Figura 5.17), e onde ele pode aceitar, rejeitar ou iniciar o processo de revisão.

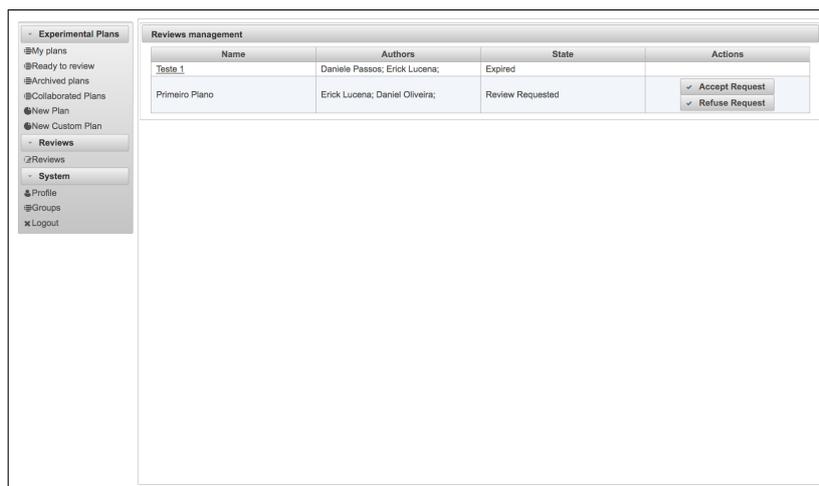


Figura 5.17 Gerenciamento de revisões

5.4.2 Revisão de plano experimental

Para efetuar a revisão de um plano experimental, o usuário deve primeiramente aceitar a revisão solicitada por um autor. Após o aceite, o usuário que solicitou a revisão será notificado via *email* e a opção de revisão será habilitada.

Após isso, o usuário pode iniciar a revisão do plano escolhendo a opção 'Review' no menu de ações, sendo então redirecionado para a tela de revisão (Figura 5.19).

Nesta tela o usuário pode ter acesso ao plano experimental a ser revisado de duas maneiras diferentes, a depender do tipo de plano criado pelo autor. Caso o plano experimental tenha sido criado fora a ferramenta, o arquivo submetido pelo autor será disponibilizado através de um link de *download*. Se o autor criou o plano experimental através da ferramenta guia, os trechos do mesmo são exibidos junto aos itens de revisão ao qual se relacionam.

Para cada item de revisão, o revisor deve escolher o *score* a ser atribuído por aquele item e pode escrever comentários que serão repassados para o autor. É importante destacar que o revisor pode salvar parcialmente o seu trabalho a qualquer hora, para retomar a revisão em outro momento.

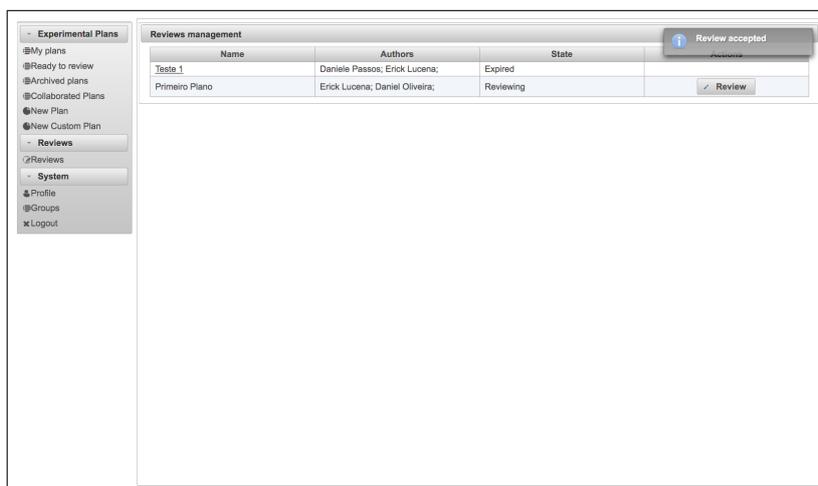


Figura 5.18 Tela de gerenciamento de revisões após aceite

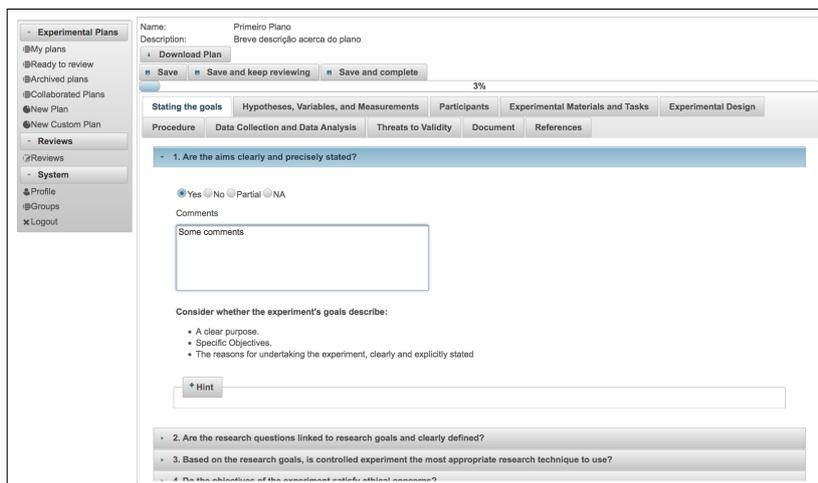


Figura 5.19 Revisão de plano experimental

5.5 Telas de grupos

Os usuários do sistema podem estar incluídos dentro de grupos, seja apenas como membro, ou como um gerente do mesmo. Através da opção 'Groups' no menu lateral, o usuário pode acessar a tela que lista os grupos aos quais o mesmo está relacionado (Figura 5.20)

Nesta tela o usuário tem acesso a duas informações: grupos dos quais o mesmo é membro, e grupos gerenciados por ele.

Neste segundo caso, o usuário pode clicar em 'Manage' para ser levado a tela de gerenciamento (Figura 5.21).



Figura 5.20 Tela de grupos

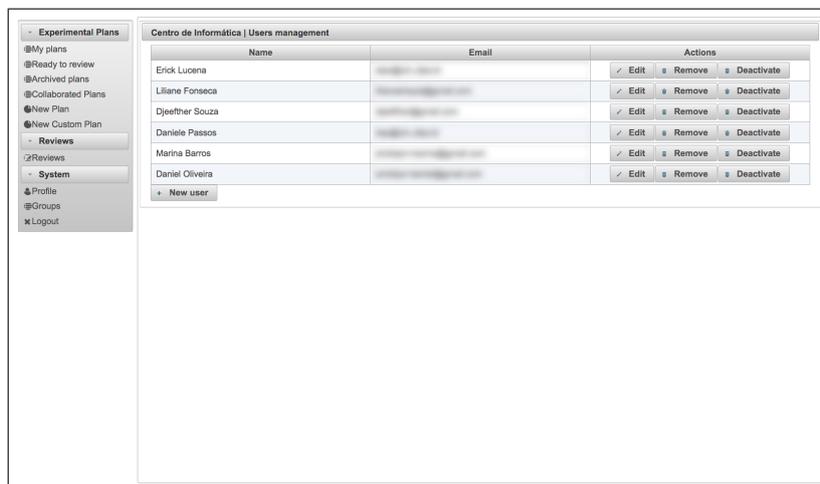


Figura 5.21 Tela de gerenciamento de um grupo

Nela, o usuário pode gerenciar os membros do grupo ou adicionar novos usuários ao mesmo.

5.6 Tela de *profile*

Através da opção '*Profile*' o usuário tem acesso a uma tela que condensa suas informações dentro do sistema (Figura 5.22).

Além de informações básicas como nome e *email*, nesta tela o usuário pode editar sua área de trabalho e fornecer um link para uma *profile* externo (eg. Lates, LinkedIn, página

The screenshot shows a web application interface for editing a user profile. On the left is a sidebar menu with categories like 'Experimental Plans', 'Reviews', and 'System'. The main content area is titled 'Edit Profile' and contains several form fields: 'Name *' (Erick Lucena), 'Institution *' (UFPE), 'Email *' (elpe@cin.ufpe.br), 'Profile Link *' (http://www.cin.ufpe.br/), and 'Work Area *' (SE). There are also two checkboxes: 'Available for reviewing' (checked) and 'Available for collaboration' (checked). Below these are 'Save' and 'Cancel' buttons. A second section titled 'Change password' has two input fields for 'New password *' and 'Confirm password *', with a 'Save' button below them.

Figura 5.22 Tela de informações do usuário

peçoal). Essas informações serão úteis para autores durante a busca de revisores para seus planos experimentais. Nesta tela também é dada ao usuário de se marcar como disponível para revisão e/ou colaboração. Esses *checkboxes* indicam ao sistema se o usuário deve ser exibido nas listagens correspondentes quando algum autor está criando um plano ou solicitando uma nova revisão.

Além de editar suas informações de perfil, o usuário pode também trocar sua senha nesta tela.

CAPÍTULO 6

Conclusão

O Reviewer é um sistema que auxilia pesquisadores, especialmente os inexperientes, no processo de criação de planos experimentais, permitindo uma maior facilidade de acesso a pesquisadores mais experientes, além de fornecer um guia de revisão para auxiliar no processo tanto de criação quanto no processo de revisão de completude.

Além de guiar o processo de criação e revisão de planos, o sistema implementado nesse trabalho fornece ferramentas que possibilitam a criação de diferentes versões de planos experimentais, dando aos autores a possibilidade de melhorar planos experimentais de maneira incremental através de múltiplas revisões. O sistema fornece também ferramentas para criação e gerenciamento de grupos de usuários, facilitando o uso do Reviewer por diferentes grupos de pesquisadores.

É importante destacar que o Reviewer é provavelmente único no que se propõe, dada a ausência de sistemas de suporte a revisão de planos experimentais e o fato da ferramenta de revisão utilizada ter sido recentemente criada.

Durante o segundo semestre de 2017, serão feitos experimentos controlados com disciplinas de engenharia de software experimental de pós-graduação, com intuito de verificar a eficácia do sistema no auxílio ao planejamento de planos experimentais. Através dos experimentos também será feita uma análise qualitativa para verificar a percepção dos participantes inexperientes em relação a ferramenta desenvolvida. Após essa fase de experimentação e avaliação, o sistema será ajustado baseado nas informações obtidas.

6.1 Trabalhos futuros

Para que o sistema evolua em sua completude e usabilidade, algumas funcionalidades podem ser adicionadas ao sistema implementado.

- **Análise da interface do sistema por um especialista em UX**

O sistema pode evoluir em usabilidade com uma análise da sua interface feita por um especialista da área de *user experience*, revisando fluxos, textos e disposição de informações dentro do sistema.

- **Adicionar um sistema de notificação**

A adição de um sistema de notificação possibilita ao usuário ter informações sobre acontecimentos pertinentes que aconteceram desde seu último acesso, como revisões finalizadas ou requisições de revisão.

- **Adicionar uma *dashboard***

Juntamente ao sistema de notificações, uma *dashboard* contendo os últimos itens e ações feitas pelo usuário daria agilidade na utilização do sistema e tem potencial para melhorar a experiência de uso.

- **Tutorial de utilização**

Embora a utilização do sistema seja intuitiva, a criação de um breve tutorial com o passo a passo das funcionalidades mais importantes do sistema durante o primeiro acesso pode facilitar o usuário a ter conhecimento mais eficiente da ferramenta.

APÊNDICE A

Lista de casos de uso

A seguir, a lista de casos de uso da plataforma web é descrita. Nela são apresentados o nome de cada caso de uso com seus respectivos atores e descrição.

Nome	Acesso a plataforma
Ator	Autor, Revisor, Colaborador, Administrador
Descrição	O usuário deve obter acesso a plataforma através de login utilizando-se de usuário e senha válidos.
Nome	Manter cadastro de usuários
Ator	Administrador
Descrição	Permite ao administrador incluir, excluir e alterar dados dos usuários e permissões.
Nome	Visualizar planos experimentais relacionados com os atores
Ator	Autor, Colaborador, Revisor
Descrição	Permite aos atores visualizarem os planos experimentais relacionados a eles.
Nome	Visualizar Status da revisão dos planos experimentais relacionados com os atores
Ator	Autor, Colaborador, Revisor
Descrição	Os atores devem ter a opção de visualizar informações como: estado atual de uma revisão, seu score atual, percentual atual da revisão, etc.
Nome	Manter planos experimentais
Ator	Autor
Descrição	Permite ao autor incluir, excluir, alterar informações dos planos experimentais.
Nome	Dar acesso a colaboradores do plano
Ator	Autor
Descrição	Permite ao autor incluir colaboradores ao plano, possibilitando o acompanhamento de sua construção

Nome	Produzir planos
Ator	Autor
Descrição	O autor poderá criar seus planos experimentais baseado no instrumento de revisão.
Nome	Acessar catálogo web externo de mecanismos de suporte para planejamento de experimentos.
Ator	Autor
Descrição	A plataforma deve permitir acesso ao catálogo externo de mecanismos de suporte através de link.
Nome	Fazer <i>Upload/Download</i> dos planos experimentais
Ator	Autor
Descrição	Deve ser permitido o <i>upload</i> de planos experimentais criados fora do sistema.
Nome	Fazer <i>download</i> dos planos experimentais
Ator	Revisor
Descrição	Durante a revisão, deve ser possível fazer o <i>download</i> de um plano experimental que foi previamente inserido por <i>upload</i> na plataforma.
Nome	Revisar Planos experimentais
Ator	Revisor
Descrição	A plataforma deve exibir para o revisor o conjunto de questões e itens descritos na ferramenta de revisão de completude de planos experimentais.

Ferramentas utilizadas

Segue abaixo as ferramentas utilizadas no processo de desenvolvimento.

B.1 Eclipse

O Eclipse¹ foi escolhido como ambiente de desenvolvimento(IDE²) por ser um software amplamente utilizado para desenvolvimento em java. O Eclipse possui integração nativa com Maven, além de permitir o *deploy*, execução e depuração de código quando adequadamente configurado com o Tomcat. Esse fatores contribuíram para uma melhor experiência durante o desenvolvimento do sistema.

B.2 Maven

Responsável pelo controle de dependências e automação da compilação, o Maven³ foi utilizado para facilitar e abstrair o processo de *build* e **packaging** do sistema.

B.3 Git

O Git⁴ é um sistema de controle de versão (VCS⁵), software utilizado para monitorar mudanças em arquivos digitais cuja utilização principal é para o gerenciamento de código fonte durante o desenvolvimento de software.

A ferramenta foi utilizada durante todo o desenvolvimento para controlar a produção de

¹<https://www.eclipse.org>

²Integrated Development Environment

³<https://maven.apache.org/>

⁴<https://git-scm.com>

⁵Version Control System

código, desde a criação das primeiras entidades. Além de possibilitar versionamento local, foi utilizado o serviço do *Bitbucket*⁶ como repositório remoto, provendo backup de todo o código fonte.

B.4 DbVisualizer

O DbVisualizer⁷ é uma ferramenta para gerenciamento de SGBDs que foi utilizada durante o desenvolvimento e testes da camada de dados.

⁶<https://bitbucket.org>

⁷<https://www.dbvis.com>

Referências Bibliográficas

- [Bas07] Victor R Basili. The role of controlled experiments in software engineering research. In *Empirical Software Engineering Issues. Critical Assessment and Future Directions*, pages 33–37. Springer, 2007.
- [Fon16] L.S.S Fonseca. *An Instrument for reviewing for the completeness of experimental plans for Controlled Experiments using Human Subjects in Software Engineering*. PhD thesis, 2016.
- [JM13] Natalia Juristo and Ana M Moreno. *Basics of software engineering experimentation*. Springer Science & Business Media, 2013.
- [Som07] Ian Sommerville. *Software Engineering*. Addison-Wesley Publishing Company, USA, 9th edition, 2007.
- [TGA02] Guilherme Horta Travassos, Dmytro Gurov, and EAGG Amaral. *Introdução à engenharia de software experimental*. UFRJ, 2002.
- [Wir08] Niklaus Wirth. A brief history of software engineering. *IEEE Annals of the History of Computing*, 30(3):32–39, 2008.
- [WRH⁺12] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012.

