

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE
MINAS GERAIS – CAMPUS V

SimSUS

Álvaro Martins Espíndola
Helise de Assis Almeida
Maria Teresa Menezes Costa

Divinópolis - MG

2015

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE
MINAS GERAIS – CAMPUS V

SimSUS

Álvaro Martins Espíndola

Helise de Assis Almeida

Maria Teresa Menezes Costa

Orientador: Daniel Morais dos Reis

Orientadora: Dra. Cláudia Di Lorenzo Oliveira Mendes

Coorientador: Eduardo Habib Bechelane Maia

Trabalho de Conclusão de Curso
apresentado ao Curso Técnico em
Informática do Centro Federal de
Educação Tecnológica de Minas
Gerais – Campus V como requisito
parcial para a obtenção do título de
Técnico em Informática.

Divinópolis - MG

2015

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE
MINAS GERAIS – CAMPUS V

Trabalho de Conclusão de Curso julgado adequado para obtenção do título de Técnico em Informática e aprovado pela banca composta pelos seguintes professores.

Prof. Daniel Morais dos Reis – CEFET – MG (Orientador)

Dra. Cláudia Di Lorenzo Oliveira Mendes – UFSJ (Orientadora)

Prof. Eduardo Habib Bechelane Maia - CEFET-MG (Coorientador)

Prof. Luís Augusto Mattos Mendes
Coordenador do Curso Técnico em Informática

Data de aprovação: Divinópolis, ___ de _____ de 2015

SUMÁRIO

| | |
|--|-----------|
| 1. Introdução | 6 |
| 1.1. Definição da Empresa | 6 |
| 1.2. Definição do Escopo | 7 |
| 1.3. Descrição Detalhada das Funcionalidades | 8 |
| 1.3.1. Login..... | 8 |
| 1.3.2. Cadastro de usuário | 9 |
| 1.3.3. Cadastro de cenário | 9 |
| 1.3.4. Jogar | 10 |
| 1.4. Referencial Teórico..... | 11 |
| 1.4.1. Java | 12 |
| 1.4.2. MVC (Model View Controller)..... | 12 |
| 1.4.3. DAO (Data Access Object) | 13 |
| 1.4.4. JSP (Java Server Pages)..... | 13 |
| 1.4.5. MySQL | 13 |
| 1.4.6. JavaScript | 13 |
| 1.4.7. JSON..... | 13 |
| 2. Projeto conceitual | 13 |
| 2.1. Diagrama de Contexto UML | 14 |
| 2.2. Documentação dos Atores..... | 15 |
| 2.2.1. Ator 01 – Aluno..... | 15 |
| 2.2.2. Ator 02 - Professor | 16 |
| 2.2.3. Ator 03 – Administrador | 16 |
| 3. Projeto Físico..... | 16 |
| 3.1. DER – Diagrama de Entidade e Relacionamento..... | 16 |
| 3.2. Diagrama de Classes..... | 17 |

| | |
|---|-----------|
| 3.3. Dicionário de Dados | 19 |
| 4. Resultados | 20 |
| 5. Cronograma | 21 |
| 6. Considerações Finais | 21 |
| 7. Referências Bibliográficas | 22 |
| 8. Anexos..... | 23 |
| 8.1. Anexo A: Documentação detalhada dos Casos de Uso..... | 23 |
| 8.2. Anexo B: DTR – Diagrama das Tabelas Relacionais | 31 |
| 8.3. Anexo C: Classes detalhadas | 32 |
| 8.3.1. Model..... | 32 |
| 8.3.2. Controller..... | 37 |
| 8.3.3. DAO | 39 |
| 8.3.4. Exception | 43 |

1. Introdução

O Sistema Único de Saúde (SUS) é o sistema público de saúde brasileiro que engloba grande parte dos atendimentos médicos existentes, garantindo acesso gratuito para toda a população do país. Tendo em vista o tamanho do programa e a quantidade de pessoas a serem beneficiadas por seus serviços, é necessário o gerenciamento dos recursos direcionados a ele. Para isso, os cursos da área da saúde apresentam a disciplina teórica de Gestão de Saúde, que é lecionada no Campus Centro Oeste da Universidade Federal de São João Del Rei (UFSJ) pela Dra. Cláudia Di Lorenzo Oliveira. Segundo a professora, o conteúdo pode parecer para o aluno desnecessário e cansativo, porém é tão importante quanto o das demais estudadas ao longo do curso, já que a administração adequada de recursos do SUS pode poupar gasto de um procedimento e investir o dinheiro economizado em outro.

Uma maneira de atrair a atenção do estudante para a disciplina é mostrá-lo a aplicação em prática. No caso da Gestão de Saúde, uma alternativa para tornar a aula mais interativa consiste na utilização de tecnologias. Pensando nisso, será desenvolvido um programa para estudantes da área da saúde que permita testar seu conhecimento acerca de administração de recursos. O objetivo do software em questão é tornar a aula de Gestão de Saúde Pública mais interativa, possibilitando a aplicação de seus estudos em cenários que possivelmente serão encontrados pelos alunos no decorrer de suas carreiras.

1.1. Definição da Empresa

O ensino da Gestão de Saúde é obrigatório em qualquer curso de graduação brasileiro na área da saúde. Embora existam grandes variações nos diferentes cursos da área e nos diversos projetos pedagógicos, todos incluem o ensino da epidemiologia, políticas de saúde e gestão em saúde. O estudo dessas disciplinas historicamente foi relegado a segundo plano, mas nas últimas Diretrizes Curriculares Nacionais dos cursos de Graduação em Medicina a relevância fica muito clara, conforme aponta a Resolução N° 3 de 20 de junho de 2014, Art 4º, transcrito a seguir:

Art. 4º - Dada a necessária articulação entre conhecimentos, habilidades e atitudes requeridas do egresso, para o futuro exercício profissional do médico, a formação do graduado em Medicina desdobrar-se-á em nas seguintes áreas:

I - Atenção à Saúde;

II - Gestão em Saúde;

III - Educação na Saúde.

Por outro lado, as aulas teóricas, muitas vezes distantes da realidade dos alunos, padecem de uma falta de interesse, pois, como foi percebido pela professora Dra. Cláudia Di Lorenzo Oliveira, eles não conseguem perceber a utilidade da matéria na sua futura vida profissional. Para aproximar o conteúdo assimilado na teoria à prática, a incorporação de novas tecnologias e metodologias computacionais pode ser de grande importância.

Adicionalmente, a educação moderna busca o desenvolvimento de metodologias ativas de aprendizado e novos cenários de prática. No ensino clínico, isso já vem sendo incorporado com o uso de ambientes simulados e robôs que permitem um ensino seguro de procedimentos invasivos. Por outro lado, no caso dos conteúdos teóricos de saúde coletiva e gestão em saúde, não existem softwares no mercado que possam ser adotados pelos professores e que permitam simular um ensino prático. A criação de um software com esse objetivo teria uma enorme utilidade na formação de futuros profissionais de saúde, em particular médicos e enfermeiros.

1.2. Definição do Escopo

O software consiste em um jogo que apresenta situações-problema, as quais são compostas por um conjunto de indicadores e fatores que definem a condição em que se encontra o sistema, com finalidade de simular a realidade do SUS. Nesse contexto, os indicadores de saúde podem ser caracterizados como:

Indicadores de saúde são parâmetros utilizados internacionalmente com o objetivo de avaliar, sob o ponto de vista sanitário, a higidez de agregados humanos, bem como fornecer subsídios aos planejamentos de saúde, permitindo o acompanhamento das flutuações e tendências históricas do padrão sanitário de diferentes coletividades consideradas à mesma época ou da mesma coletividade em diversos períodos de tempo. (Rouquayrol, 1993).

Já os fatores são considerados determinantes sociais da saúde que variam de acordo com as condições da realidade abordada, sendo elas sociais, econômicas, culturais, psicológicas e comportamentais, que influenciam a ocorrência de problemas de saúde e fatores de risco à população. Dessa forma, fatores e indicadores apresentam uma relação de

interferência direta que exige a presença de soluções. Sendo assim, a função do jogador é administrar a quantia disponibilizada de recursos, de modo que escolha opções com melhor custo benefício.

Uma das limitações apresentadas pelo programa é que ele não trata da Gestão de Hospitais no geral, apenas da Gestão de Saúde Pública. Os dados utilizados no projeto são fictícios e fornecidos pela Dra. Cláudia Di Lorenzo Oliveira.

1.3. Descrição Detalhada das Funcionalidades

Serão apresentadas, a seguir, as funcionalidades de forma detalhada do sistema SimSUS. Estas consistem em login, cadastros de usuários e cenários e relações entre fatores, indicadores e soluções.

1.3.1. Login

A tela de login, mostrada na Figura 1, é composta pelos campos de nome e senha do usuário. Para entrar no sistema, é necessário também escolher o nível do usuário, podendo ser ele aluno, professor ou administrador. Além disso, caso o indivíduo ainda não esteja cadastrado, existe a opção “Cadastrar-se” nessa tela, que direciona o usuário ao formulário de cadastro.

SimSUS Sobre

Login

Nome de usuário:

Senha:

Aluno

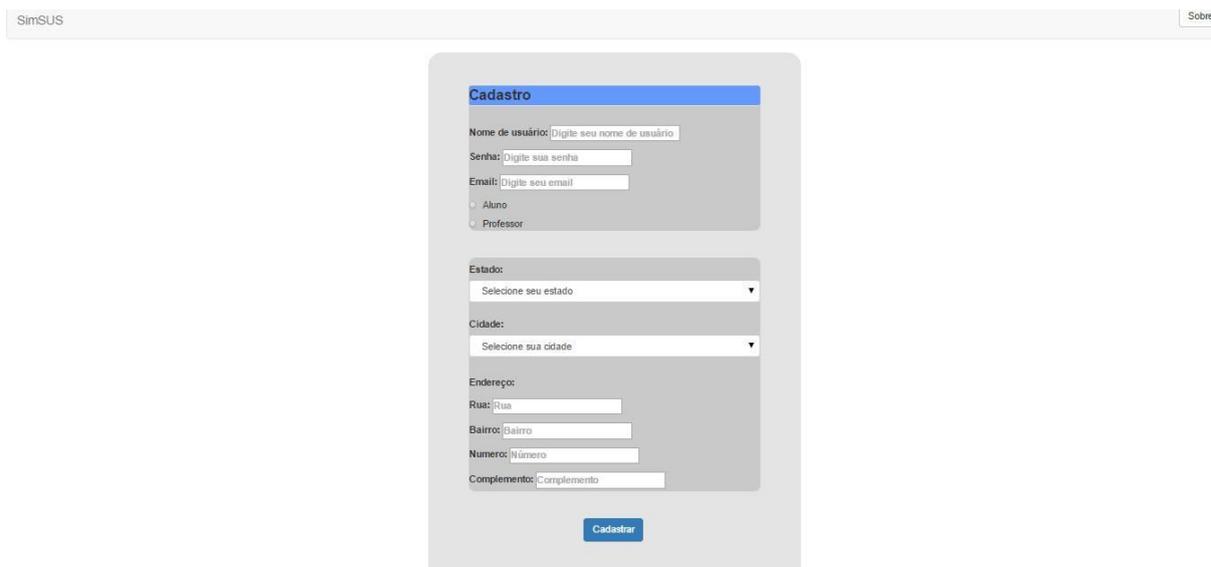
Professor

Administrador

FIGURA 1 – Tela de Login

1.3.2. Cadastro de usuário

Nesse tópico, apresenta-se na Figura 2 a tela de cadastro. Essa funcionalidade é essencial para o acesso ao simulador e requer informações pessoais do usuário como nome, senha, email, nível de usuário e endereço. Tanta importância é dada devido ao fato de que apenas o indivíduo cadastrado possui acesso ao sistema SimSUS, seja ele como professor ou aluno.



The screenshot shows the SimSUS user registration interface. At the top, there is a header with 'SimSUS' on the left and 'Sobre' on the right. The main content area is a light gray box with a blue header labeled 'Cadastro'. Below the header, there are several input fields: 'Nome de usuário:' with a placeholder 'Digite seu nome de usuário', 'Senha:' with a placeholder 'Digite sua senha', and 'Email:' with a placeholder 'Digite seu email'. There are two radio buttons for user type: 'Aluno' and 'Professor'. Below these are two dropdown menus for 'Estado:' and 'Cidade:', both with the placeholder 'Selecione seu estado/cidade'. The 'Endereço:' section includes four input fields: 'Rua:' (placeholder 'Rua'), 'Bairro:' (placeholder 'Bairro'), 'Número:' (placeholder 'Número'), and 'Complemento:' (placeholder 'Complemento'). At the bottom of the form is a blue button labeled 'Cadastrar'.

FIGURA 2 – Tela de Cadastro de Usuário

1.3.3. Cadastro de cenário

A funcionalidade de cadastrar cenários, ilustrada na Figura 3, é exclusividade do professor. Nela, são definidos os indicadores, os fatores e o capital disponível para o gerenciamento do cenário de simulação do Sistema de Saúde.



The screenshot shows the SimSUS scenario creation interface. At the top, there is a header with 'SimSUS' on the left and 'Log Out' on the right. Below the header is a large blue banner with the text 'Crie seu cenário!'. The main content area is a light gray box with a blue header labeled 'Fatores'. Below the header, there are two columns of checkboxes for factors and indicators. The 'Fatores' section includes: 'Baixa escolaridade da mãe', 'Falta de saneamento', 'Baixa cobertura vacinal', 'Desnutrição', 'Baixa cobertura de exames preventivos', 'Tratamento de todos os casos', 'Parto hospitalar', and 'Vacinação'. The 'Indicadores' section includes: 'Taxa de mortalidade infantil tardia', 'Teste2', 'Taxa de mortalidade na infância (<5 anos)', 'Taxa de mortalidade por câncer de mama em mulheres de 30 a 69 anos', 'Taxa de incidência de hanseníase', 'Proporção de partos assistidos por profissional de saúde qualificado', 'Número de intimações por aborto no SUS, por grandes regiões', and 'Taxa de intimação dos portadores de diabetes e suas complicações'. The 'Renda:' field is set to '10000'. At the bottom of the form is a blue button labeled 'Criar'.

FIGURA 3 - Tela de Criação de Cenário

1.3.4. Jogar

A tela disponibilizada para o usuário jogar, mostrada na Figura 4, representa a maior parte do sistema. Tal relevância é dada devido à abrangência total das possibilidades de jogadas do aluno em uma só tela. Nela, estão disponíveis as situações-problema, as interferências, as soluções, a renda e as relações do conjunto.



FIGURA 4 - Tela de Simulação

No centro da tela, estão dispostos os indicadores e suas respectivas taxas de satisfação, que indicam a conjuntura em que os parâmetros de avaliação dos níveis de saúde se encontram. Elas são representadas por barras preenchidas e coloridas de acordo com a condição dos níveis. Quanto maior o contentamento, maior o preenchimento da barra em tom de verde, entretanto, quando esse nível estiver baixo, a cor será alterada para vermelho e em nível intermediário para amarelo.

Ao redor dos indicadores, estão os fatores representados por círculos ilustrados conforme suas características. Além disso, no momento em que o usuário clicar em um indicador ou fator, os fatores influenciados por ele ficarão destacados com um contorno na cor verde. Já quando isso ocorrer com os indicadores, um pequeno círculo ao lado deles ficará colorido de verde.

Ainda na tela principal, está presente uma barra superior, a qual é composta pelas possibilidades de visualizar as soluções, as opções do jogo, o capital disponível para resolver os problemas e ainda sair do sistema. Ao escolher a opção “Soluções”, o jogador terá visibilidade de todas as possíveis ações que resolvam os problemas e seus respectivos custos, mostrado na Figura 5. Após isso, ele deverá implementar a que julgar mais eficiente, ou seja, aquela que proporciona maiores benefícios ao simulador com menores custos.

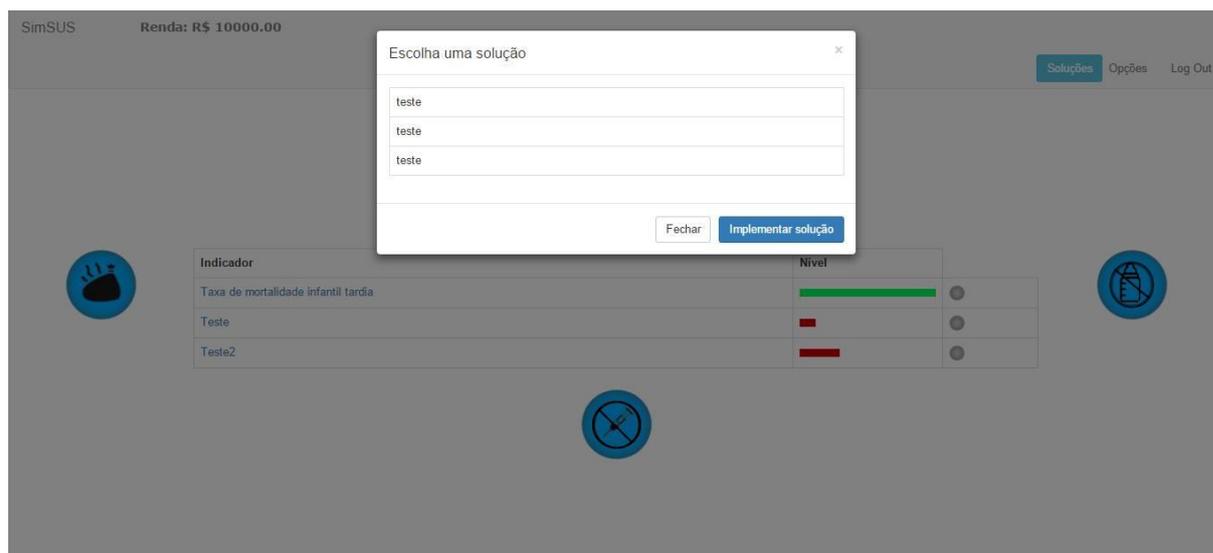


FIGURA 5 – Soluções

1.4.Referencial Teórico

Para realizar o projeto, as informações que geram as situações-problema foram fornecidas pela professora da Universidade Federal de São João Del-Rei (UFSJ) Cláudia Di Lorenzo Oliveira.

Como exemplo de jogabilidade, foi usado o jogo *Democracy 3*, mostrado na Figura 6, produzido pela *Positech Games* em 2013. Seu funcionamento consiste em manter os níveis de satisfação dos eleitores, de modo que o governo do atual líder, que é o próprio jogador, agrade as diversas etnias da população. Como é um jogo de simulação, ele exibe diversas variáveis que consideram a felicidade de diferentes grupos de eleitores e a forma do governante de lidar com crime, desemprego, terrorismo, alterações climáticas e educação, por exemplo. As semelhanças entre o SimSUS e o *Democracy 3* são, além da jogabilidade, o layout e o estilo.

Já as diferenças entre os dois se referem ao conteúdo. Enquanto o *Democracy 3* preza a satisfação dos eleitores considerando os indicadores sociais, o SimSUS se preocupa com o bem-estar da população de acordo com os indicadores de saúde.

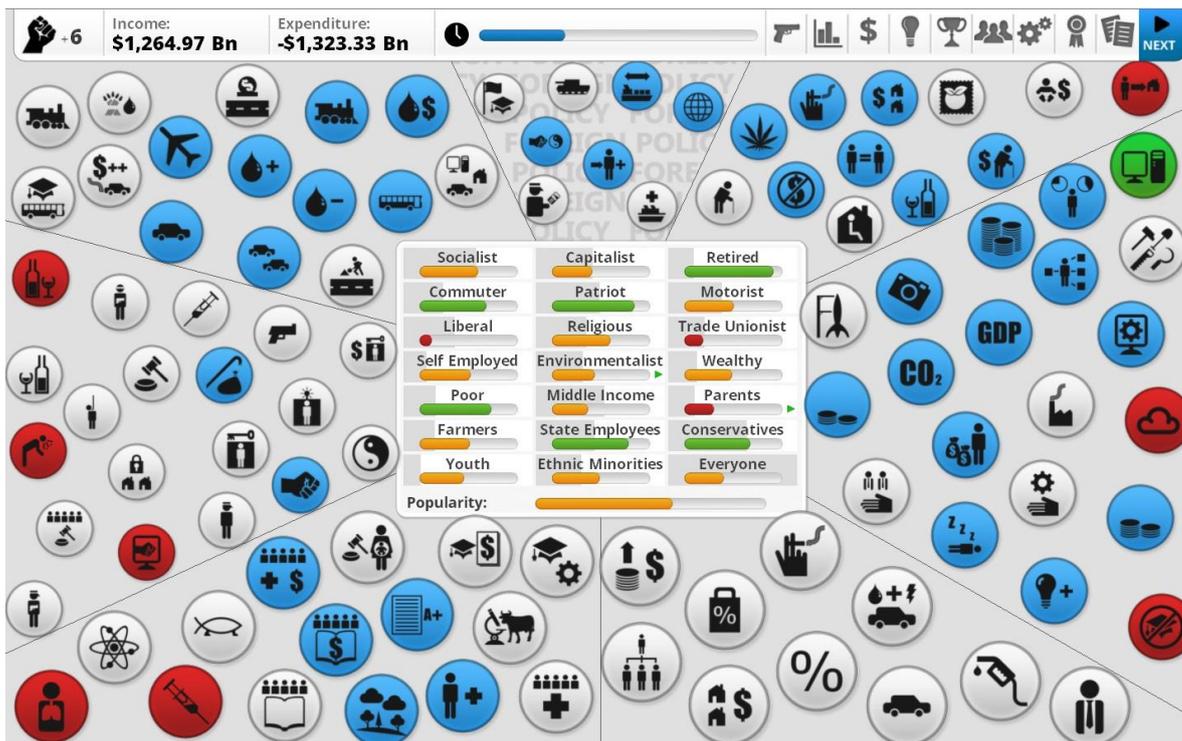


FIGURA 6 - Democracy 3

A seguir serão apresentadas as ferramentas e linguagens escolhidas para o desenvolvimento do software.

1.4.1. Java

Java é uma linguagem de programação orientada a objeto criada na década de 90, com finalidade de criação de programas confiáveis, isentos de situações de erro e universais. Suas principais propriedades são segurança e robustez e ela foi utilizada para o desenvolvimento do sistema de cadastramento e do sistema do jogo em si. Além disso, foram seguidos os padrões *MVC* e *DAO*.

1.4.2. MVC (Model View Controller)

MVC é um padrão de arquitetura de software que separa a representação da informação da interação do usuário com ele. Nele, são separadas três partes: *model*, *view* e *controller*. Na primeira delas estão todos os modelos de dados, lógicas e funções que serão aplicadas. Já a segunda representa toda forma de saída e/ou impressão desses dados, seja por forma de diagramas ou tabelas e na terceira se localizam tudo o que fará com que os dados cheguem nas outras duas partes.

1.4.3. DAO (Data Access Object)

DAO é um padrão para persistência de dados que permite separar regras de negócio das regras de acesso a banco de dados. Numa aplicação que utilize a arquitetura *MVC*, todas as funcionalidades de bancos de dados, tais como obter as conexões, mapear objetos Java para tipos de dados SQL ou executar comandos SQL, devem ser feitas por classes *DAO*.

1.4.4. JSP (Java Server Pages)

Java Server Pages foi utilizado para gerar páginas web dinamicamente baseadas em HTML 5. Para isso, foi usado o Glassfish como servidor web, pois esse é compatível com um container servlet. Com a utilização dessa tecnologia, é possível que o sistema SimSUS acesse o banco de dados, manipule arquivos de texto e capture informações a partir de formulários.

1.4.5. MySQL

MySQL é um sistema de gerenciamento de banco de dados (SGBD) que utiliza a linguagem SQL. A *Structured Query Language* (Linguagem Estruturada de Pesquisa) é uma forma de consulta e manipulação de dados, desenvolvida especialmente para o ambiente relacional. Os principais benefícios dessa ferramenta são a simplicidade, a facilidade e a portabilidade entre computadores. Isso, porque o usuário tem que lidar com comandos em inglês, de fácil entendimento e a consulta aos dados ocorre de maneira rápida.

1.4.6. JavaScript

A comunicação entre servidor e cliente foi dinamizada com a utilização do JavaScript. Essa linguagem de programação utiliza scripts que funcionam do lado do usuário, ou seja, são executados diretamente no navegador.

1.4.7. JSON

Visando a integração das duas linguagens já citadas (Java e JavaScript), foi utilizado o *JavaScript Object Notation* (JSON), o qual é um formato leve de intercâmbio de dados computacionais.

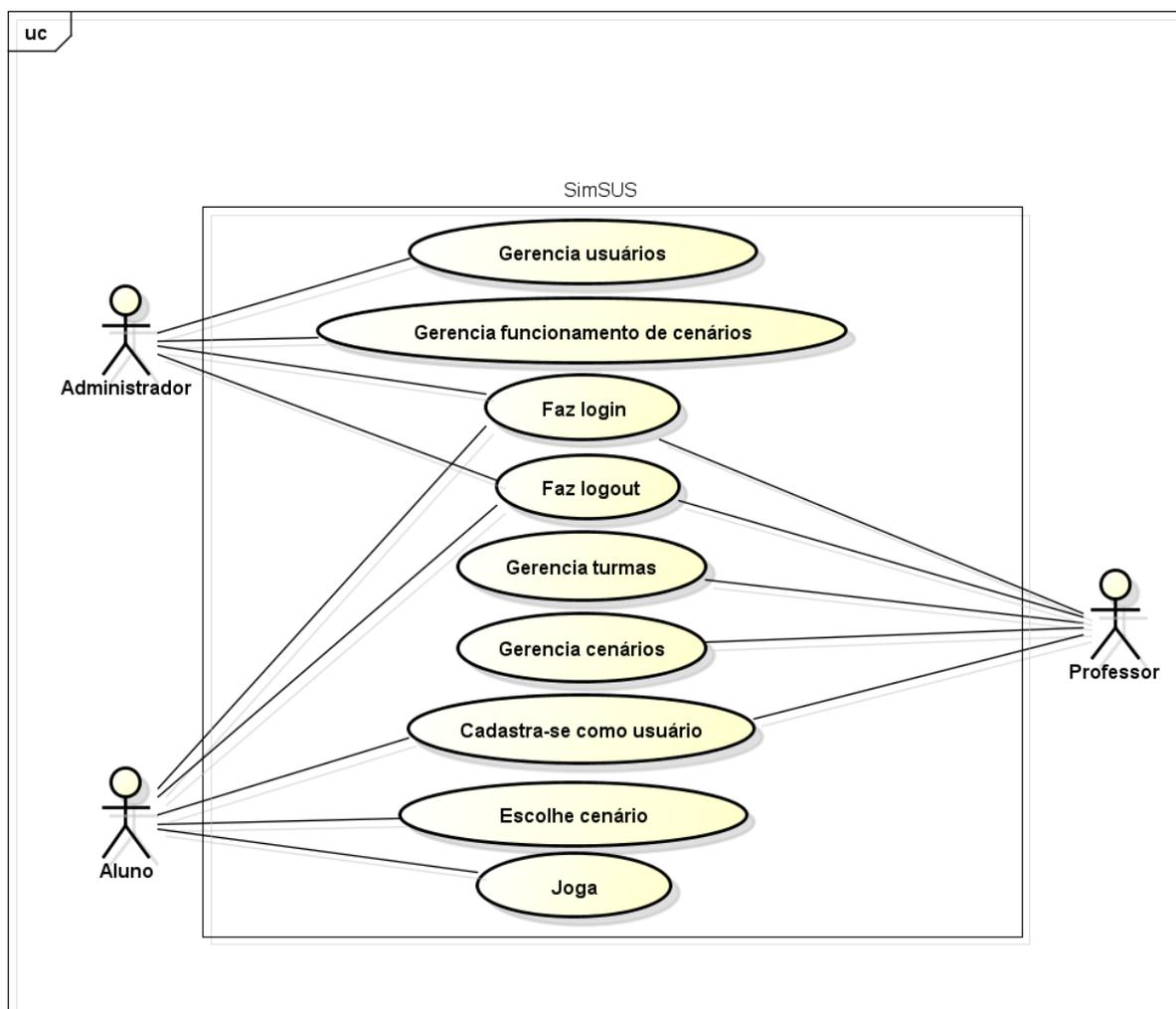
2. Projeto conceitual

O projeto conceitual consiste na descrição do problema para análise das possíveis soluções, de forma que consiga atender as necessidades do software. A sessão 2.1 contém o

diagrama de Caso de Uso, que descreve um ambiente que mostra as funcionalidades do sistema e a sessão 2.2 detalha as ações dos atores.

2.1.Diagrama de Contexto UML

A Figura 7 representa o diagrama de contexto UML, no qual estão contidos os atores e as permissões executadas por eles. No total são três atores, aluno, professor e administrador, cada qual com suas próprias funcionalidades e cadastros. Ao aluno é concedido as ações de se cadastrar, fazer login e logout, escolher o cenário a ser jogado e jogar. Os cenários utilizados para que o estudante jogue são criados e disponibilizados pelo professor, o qual especifica o capital a ser gasto e as condições do SUS como um todo. O professor também gerencia as turmas e cenários, cadastra-se e faz login e logout. Além disso, há o administrador, ator que tem como função verificar constantemente o funcionamento dos cenários e manuseamento por parte dos usuários. A documentação detalhada desses casos de uso é apresentada no Anexo A.



powered by Astah

FIGURA 7 - Diagrama de Contexto UML

2.2. Documentação dos Atores

Nos subtópicos abaixo serão tratadas detalhadamente as ações de cada usuário do sistema.

2.2.1. Ator 01 – Aluno

O aluno possui apenas as ações de cadastrar-se como usuário, realizar login e logout, escolher o cenário em que deseja jogar, dentre os que forem disponibilizados pelo professor, e jogar, apenas buscando soluções para problemas existentes.

2.2.2. Ator 02 - Professor

O professor, além de possuir todas as ações do aluno, exceto jogar, tem autonomia de gerenciar suas turmas e cenários. Na funcionalidade de cadastros, este ator pode se registrar no sistema e criar salas de aula para seus alunos. Após seu cadastro, o professor também tem a possibilidade de criar cenários, especificando características como população, recursos disponíveis, taxas dos indicadores, número de leitos, entre outros, alterar cenários criados anteriormente, excluí-los e ainda liberá-los para uso dos alunos.

2.2.3. Ator 03 – Administrador

O cadastro do administrador do sistema é feito durante a implementação do software para que também possa realizar login e logout. Esse ator pode ainda gerenciar cadastros e o funcionamento, realizando alterações e exclusões conforme julgar necessário, além de supervisionar a atuação dos alunos e professores nestes ambientes, respectivamente.

3. Projeto Físico

Nesta seção, será apresentada a documentação do modelo do banco de dados e o diagrama de classes e o dicionário de dados, com o objetivo de propiciar uma visão mais ampla da parte física do sistema.

3.1.DER – Diagrama de Entidade e Relacionamento

A Figura 8 mostra o DER, diagrama de entidade e relacionamento, o qual ilustra o modelo conceitual que descreve as entidades envolvidas na composição do banco de dados, suas características e o tipo de relacionamento entre elas. As tabelas do banco serão detalhadas no Anexo B.

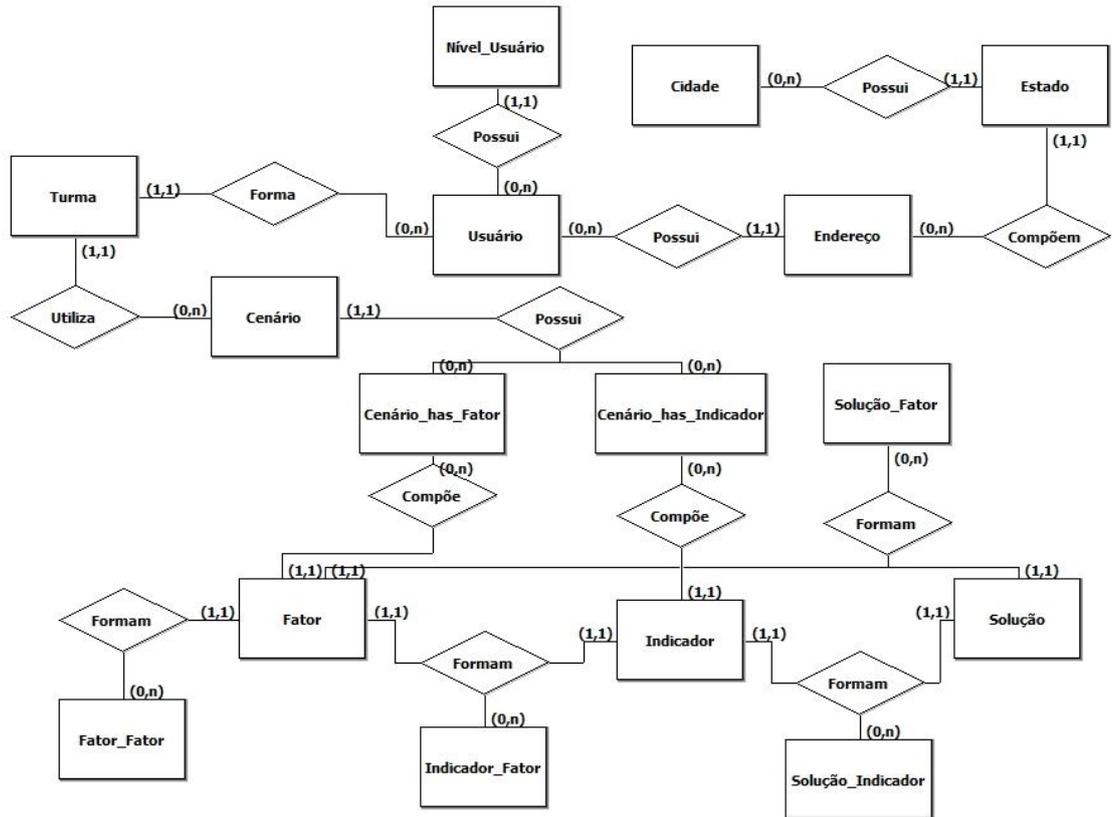


FIGURA 8 – DER – Diagrama de Entidade e Relacionamento

3.2. Diagrama de Classes

O diagrama de classes, ilustrado na Figura 9 contém as principais classes do sistema, neste caso incluindo as referentes ao padrão MVC e seus relacionamentos. No diagrama a seguir estão contidos apenas os nomes das classes e, devido à sua extensão, não está propício para leitura. Para facilitar a compreensão, estas são detalhadas, separadamente, no Anexo C.

3.3.Dicionário de Dados

O dicionário de dados descreve as tabelas do banco de dados e exibe todos os atributos presentes nessas.

Indicador = * Indicador de problema no simulador *

id_Indicador + inome + idesc

Fator = * Fator de interfere nos indicadores *

id_Fator + fnome + fdesc + peso

Solucao = * Solução para os indicadores de problemas *

id_Solucao + snome + sdesc + custo

Estado = * Estado cadastrado no sistema *

id_Estado + nome_Estado

Endereco = * Endereço inserido pelo usuário*

id_Endereço + (id_Estado) + rua + bairro + numero + complemento

Nível_Usuario = * Nível de acesso de usuário *

id_Nível_Usuario + usuario

Turma = * Turma cadastrada no sistema*

id_Turma + nome_Turma

Usuario = * Usuário cadastrado no sistema *

id_Usuario + nome + (nivel_Usuario) + (id_Endereco) + (id_Turma) + senha + email

Cenario = * Cenario cadastrado no sistema*

id_Cenario + renda + (id_Turma)

Cidade = * Cidade cadastrada no sistema *

id_Cidade + nome_cidade + (Estado_id_Estado)

Indicador_Fator = * Relação entre indicador e fator*

id_Indicador_Fator + (id_Indicador) + (id_Fator) + taxa_Interferencia

Fator_Fator = * Relação entre fatores*

id_Fator_Fator + (id_Fator) + (id_Fator_Interferido) + taxa_Interferencia

Solucao_Fator = * Relação entre solução e fator*

id_Solucao_Fator + (id_Solucao) + (id_Fator) + taxa_Interferencia

Indicador_Solucao = * Relação entre indicador e solução*

id_Indicador_Solucao + (id_Indicador) + (id_Solucao) + taxa_Interferencia

Cenario_has_Fator = * Relação entre cenário e fator*

(id_Cenario) + (id_Fator)

Indicador_has_Cenario = * Relação entre indicador e cenário*

(id_Cenario) + (id_Indicador)

4. Resultados

A primeira etapa do projeto centralizou-se na compreensão da proposta fornecida pela professora Dra. Cláudia Di Lorenzo Oliveira. Essa fase foi a mais demorada do trabalho, tendo em vista que a análise de requisitos dependia da organização das informações, que, por sua vez, eram complexas e numerosas.

Após a definição do escopo, foi possível trabalhar no desenvolvimento dos diagramas e dos modelos para o banco de dados. Essa documentação auxiliou no encerramento da primeira etapa, possibilitando o entendimento das funcionalidades do sistema.

A partir do escopo e da diagramação, foi permitido o início da etapa seguinte do projeto. Esta consistiu na codificação e, apesar de extensa, foi facilitada pelo padrão MVC-DAO adotado, o qual melhorou a organização e o entendimento das classes e ações do sistema. Nesse momento, houve um atraso relacionado às ferramentas escolhidas inicialmente, já que o *framework Phaser* mostrou-se pouco utilizado no ambiente acadêmico, fato que acarretou dificuldades na busca por auxílio. Diante disso, a maneira encontrada para superar a questão foi a inutilização desse instrumento de apoio e o desenvolvimento do código manualmente através do JSP.

Em suma, a divisão de etapas do projeto contou com a participação do orientador e dos coorientadores, de modo que a integração do todo dependeu da disponibilidade de cada integrante. Os resultados foram dentro do esperado e em breve estarão disponíveis para os alunos da professora Dra. Cláudia testarem.

5. Cronograma

| |  | Task Name | % concluída | Duração | Início | Término |
|----|---|---|-------------|-----------|--------------|--------------|
| 1 | ✓ | ▶ Planejamento | 100% | 50 dias? | Seg 09/03/15 | Sex 15/05/15 |
| 2 | ✓ | Definição de Empresa | 100% | 50 dias | Seg 09/03/15 | Sex 15/05/15 |
| 3 | ✓ | Definição de Escopo | 100% | 50 dias? | Seg 09/03/15 | Sex 15/05/15 |
| 4 | ✓ | Definição de Funcionalidades | 100% | 25 dias | Seg 13/04/15 | Sex 15/05/15 |
| 5 | ✓ | ▶ Desenvolvimento | 100% | 155 dias? | Seg 23/03/15 | Sex 23/10/15 |
| 6 | ✓ | Aprender Java | 100% | 145 dias? | Seg 23/03/15 | Sex 09/10/15 |
| 7 | ✓ | Aprender MVC | 100% | 10 dias? | Seg 06/07/15 | Sex 17/07/15 |
| 8 | ✓ | Aprender DAO | 100% | 10 dias? | Seg 06/07/15 | Sex 17/07/15 |
| 9 | ✓ | Aprender JSP | 100% | 115 dias? | Seg 18/05/15 | Sex 23/10/15 |
| 10 | ✓ | Aprender MySQL | 100% | 145 dias? | Seg 23/03/15 | Sex 09/10/15 |
| 11 | ✓ | Aprender JavaScript | 100% | 50 dias | Seg 18/05/15 | Sex 24/07/15 |
| 12 | ✓ | Aprender JSON | 100% | 30 dias? | Seg 20/07/15 | Sex 28/08/15 |
| 13 | ✓ | Adquirir Banco de Dados | 100% | 7 dias | Seg 13/07/15 | Ter 21/07/15 |
| 14 | ✓ | ▶ Diagramação | 100% | 120 dias? | Seg 23/03/15 | Sex 04/09/15 |
| 15 | ✓ | Diagrama de Contexto UML | 100% | 20 dias? | Seg 25/05/15 | Sex 19/06/15 |
| 16 | ✓ | Diagrama de Caso de Uso | 100% | 6 dias | Seg 04/05/15 | Seg 11/05/15 |
| 17 | ✓ | Diagrama de Classes | 100% | 20 dias? | Seg 10/08/15 | Sex 04/09/15 |
| 18 | ✓ | Diagrama de Estado de Máquina e Atividade | 100% | 7 dias? | Seg 23/03/15 | Ter 31/03/15 |
| 19 | ✓ | Diagrama de Sequência | 100% | 20 dias? | Seg 03/08/15 | Sex 28/08/15 |
| 20 | ✓ | ▶ Documentação | 100% | 168 dias? | Seg 23/03/15 | Qua 11/11/15 |
| 21 | ✓ | Relatório Parcial | 100% | 61 dias? | Seg 23/03/15 | Seg 15/06/15 |
| 22 | ✓ | Relatório Final | 100% | 73 dias? | Seg 03/08/15 | Qua 11/11/15 |
| 23 | | ▶ Implementação | 81% | 80 dias? | Seg 24/08/15 | Sex 11/12/15 |
| 24 |  | Interface gráfica | 80% | 45 dias? | Seg 24/08/15 | Sex 23/10/15 |
| 25 |  | Interface Web | 80% | 45 dias? | Seg 24/08/15 | Sex 23/10/15 |
| 26 |  | Povoar Banco de Dados | 70% | 60 dias? | Seg 24/08/15 | Sex 13/11/15 |
| 27 | ✓ | Elaborar sprites | 100% | 10 dias? | Seg 05/10/15 | Sex 16/10/15 |
| 28 | ✓ | Retoques gráficos | 100% | 5,6 dias | Seg 05/10/15 | Sex 30/10/15 |
| 29 | ✓ | Junção do jogo com banco de da | 100% | 8 dias? | Dom 01/11/15 | Qua 11/11/15 |
| 30 | | ▶ Finalização | 0% | 16 dias? | Sex 20/11/15 | Sex 11/12/15 |
| 31 |  | Hospedagem Web | 0% | 16 dias? | Sex 20/11/15 | Sex 11/12/15 |

FIGURA 10 - Cronograma final do projeto

6. Considerações Finais

O projeto objetivou a elaboração de um *software* voltado para a área de saúde com finalidade acadêmica. Esse simulador representa a realidade do sistema público de saúde de forma que o aluno possa aplicar os conceitos aprendidos em aula e desenvolva a habilidade de lidar com problemas frequentes em seu futuro ambiente de trabalho. O jogo é caracterizado

por ser dinâmico e criativo, além de englobar métodos que ampliem a capacidade dos alunos de administrar um sistema de saúde.

Diante do que foi proposto, o jogo ficará hospedado em um site que poderá ser fornecido pelo Governo Federal. No momento, ele possui uma interface simples para que qualquer usuário se familiarize com o sistema.

O progresso do projeto ocorreu de acordo com o cronograma apresentado na Figura 10, no tópico anterior. Atualmente, o projeto se encontra na fase de Finalização, na qual está ocorrendo a divulgação e a apresentação final do Trabalho de Conclusão de Curso.

Assim, consideramos que o jogo foi finalizado conforme o tempo concedido, todavia, devido aos atrasos no início do projeto, não foi possível implementar as soluções de maneira completa e salvar os cenários até a presente data. O funcionamento estará completo e disponível para testes de usabilidade até março de 2016, prazo em que o Projeto de Iniciação Científica concomitante com esse trabalho expira.

7. Referências Bibliográficas

Apostila Java para desenvolvimento web. Disponível em: <<https://www.caelum.com.br/apostila-java-web/mvc-model-view-controller/#9-1-servlet-ou-jsp>> Acesso em 09 de junho 2015.

ARANTES, André L.. **SQL – Apostila.** Disponível em: <http://www.andrearantes.eti.br/icesp/bd/firebird/sql_apostila_conceitos.pdf> Acesso em 14 de maio 2015.

Democracy 3. Disponível em: <<http://www.positech.co.uk/democracy3/>> Acesso em 13 de maio 2015.

GUDWIN, Ricardo R.. **Introdução à Linguagem UML.** Departamento de Engenharia de Computação e Automação Industrial da Faculdade de Engenharia Elétrica e de Computação da UNICAMP; 2010.

Java. Disponível em: <https://www.java.com/pt_BR/about/> Acesso em 14 de maio 2015.

MINISTÉRIO DA EDUCAÇÃO CONSELHO NACIONAL DE EDUCAÇÃO CÂMARA DE EDUCAÇÃO SUPERIOR. **Institui Diretrizes Curriculares Nacionais do Curso de Graduação em Medicina e dá outras providências.** Resolução nº 3, de 20 de junho de 2014.

Ministério da Saúde - Portal da Saúde. **Entenda o SUS.** Disponível em: <<http://portalsaude.saude.gov.br/>> Acesso em 23 agosto 2015.

ROUQUAYROL, M. Z. Epidemiologia & Saúde. Rio de Janeiro, MEDSI, 1993.

The JavaScript Source is your resource for thousands of free JavaScripts for cutting and pasting into your Web pages. 2015. Disponível em: <<http://www.javascriptsource.com/>> Acesso em 20 julho 2015.

8. Anexos

8.1. Anexo A: Documentação detalhada dos Casos de Uso

| | |
|---|---|
| Nome do Caso de Uso | |
| Gerencia usuários | |
| Ator Primário | |
| Administrador | |
| Atores Secundários | |
| Professor e Aluno | |
| Resumo | |
| Este caso de uso controla as atividades realizadas pelos usuários relacionadas ao cadastro e ao login | |
| Pré-condições | |
| Realizar login | |
| Pós-condições | |
| Emissão de erro caso não haja usuário cadastrado | |
| Fluxo Principal | |
| Ações do Ator | Ações do Sistema |
| 1 – Verificar se as informações dos usuários estão completas e corretas | |
| | 2 – Informar se há erro nas informações dos usuários e, se houver, indicar onde ele se encontra |
| Restrições e Validações | |
| Não é permitido haver mais de um usuário com mesmo login | |
| Fluxo Alternativo I – Usuário não logado | |
| Ações do Ator | Ações do Sistema |
| | Informar ao usuário que é necessária a |

| | |
|--|--|
| | realização do login |
| Fluxo de Exceção I – Falha no login | |
| Ações do Ator | Ações do Sistema |
| | Informar ao usuário que houve erro no login. |
| Responsável pela definição | Data da criação |
| Álvaro Martins Espíndola | 11/06/2015 |

TABELA 1 – Caso de Uso Gerencia Usuários

| | |
|--|--|
| Nome do Caso de Uso | |
| Gerencia funcionamento de cenários | |
| Ator Primário | |
| Administrador | |
| Atores Secundários | |
| Professor | |
| Resumo | |
| Este caso de uso verifica se as informações necessárias para o funcionamento dos cenários estão corretas | |
| Pré-condições | |
| Os cenários devem ser criados anteriormente pelo professor | |
| Pós-condições | |
| Os cenários devem atender os requisitos necessários para funcionarem | |
| Fluxo Principal | |
| Ações do Ator | Ações do Sistema |
| 1 – Verificar se as informações dos cenários estão completas e corretas | |
| | 2 – Informar se há erro nas informações dos cenários |
| | 3 – Informar onde se encontra o erro |
| 4 – Verificar os valores disponibilizados no cenário para serem administrados | |
| | 5 – Comparar e informar se os valores definidos pelo professor se encontram no intervalo definido pelo administrador |
| Restrições e Validações | |
| As informações do cenário devem estar dentro das normas pré-definidas. | |
| Fluxo de Exceção I – Cenário defeituoso | |
| Ações do Ator | Ações do Sistema |
| | Informar ao professor que há necessidade de alteração do cenário. |
| Responsável pela definição | Data da criação |
| Maria Teresa Menezes Costa | 11/06/2015 |

TABELA 2 – Caso de Uso Gerencia Funcionamento de Cenários

| | |
|--|---|
| Nome do Caso de Uso | |
| Faz login | |
| Ator Primário | |
| Administrador, professor e aluno | |
| Atores Secundários | |
| Não tem | |
| Resumo | |
| Este caso de uso controla o login dos usuários | |
| Pré-condições | |
| Possuir cadastro | |
| Pós-condições | |
| Não tem | |
| Fluxo Principal | |
| Ações do Ator | Ações do Sistema |
| 1 – Informar usuário e senha | |
| 2 – Confirmar dados inseridos | |
| | 3 – Verificar se todos os campos estão preenchidos |
| | 4 – Realizar login caso haja compatibilidade |
| Restrições e Validações | |
| O usuário não pode inserir caracter especial | |
| Fluxo Alternativo I – Erro no login | |
| Ações do Ator | Ações do Sistema |
| | Informar que o usuário e a senha não combinam e pedir para inserir novamente. |
| Responsável pela definição | Data da criação |
| Helise de Assis Almeida | 11/06/2015 |

TABELA 3 – Caso de Uso Faz Login

| |
|---|
| Nome do Caso de Uso |
| Faz logout |
| Ator Primário |
| Administrador, professor e aluno |
| Atores Secundários |
| Não tem |
| Resumo |
| Este caso de uso controla o logout dos usuários |
| Pré-condições |
| Estar logado |
| Pós-condições |
| Não tem |

| Fluxo Principal | |
|-----------------------------------|--------------------------------|
| Ações do Ator | Ações do Sistema |
| 1 – Informar desejo de sair | 3 – Desconecta usuário do jogo |
| Restrições e Validações | |
| Não possui | |
| Responsável pela definição | Data da criação |
| Helise de Assis Almeida | 11/06/2015 |

TABELA 4 – Caso de Uso Faz Logout

| Nome do Caso de Uso | |
|--|--|
| Gerencia turmas | |
| Ator Primário | |
| Professor | |
| Atores Secundários | |
| Não tem | |
| Resumo | |
| Este caso de uso controla a criação, exclusão e alteração das turmas. | |
| Pré-condições | |
| Possuir professor cadastrado | |
| Pós-condições | |
| Turmas devem estar disponíveis, alterados ou excluídos, dependendo da opção escolhida pelo professor | |
| Fluxo Principal | |
| Ações do Ator | Ações do Sistema |
| 1 – Informar se deseja criar uma turma | 2 – Disponibilizar formulário para ser preenchido com dados a respeito da turma |
| 3 – Preencher com informações referentes à turma | 4 – Verificar se todos os campos estão completos e corretos |
| 5 – Informar se deseja excluir uma turma | 6 – Exibir mensagem perguntando para o usuário se ele deseja realmente excluir a turma |
| 7 – Confirmar se deseja excluir uma turma | 8 – Caso a resposta do usuário seja positiva, excluir a turma |
| 9 – Informar se deseja alterar uma turma | 10 – Exibir mensagem para o usuário perguntando qual campo ele deseja alterar |
| 11 – Informar qual campo deseja alterar na turma | 12 – Exibir mensagem para o usuário preencher com o novo dado |

| | |
|---|---|
| 13 – Alterar e confirmar a alteração | 14 – Salvar a alteração |
| Restrições e Validações | |
| Nos casos de exclusão e alteração, é necessário existir uma turma previamente cadastrada. | |
| Fluxo Alternativo I – Dados incorretos | |
| Ações do Ator | Ações do Sistema |
| | Informar ao usuário que o dado inserido está incorreto. Requisitar preenchimento de dados. Revalidar dados. |
| Fluxo de Exceção I – Inexistência de turma | |
| Ações do Ator | Ações do Sistema |
| | Informar ao usuário que não há turma para ser excluída e/ou alterada. |
| Responsável pela definição | Data da criação |
| Maria Teresa Menezes Costa | 16/06/2015 |

TABELA 5 – Caso de Uso Gerencia Turmas

| | |
|---|--|
| Nome do Caso de Uso | |
| Gerencia cenários | |
| Ator Primário | |
| Professor | |
| Atores Secundários | |
| Não tem | |
| Resumo | |
| Este caso de uso controla a criação, exclusão e alteração dos cenários de cada turma referente ao ator primário | |
| Pré-condições | |
| Possuir professor cadastrado | |
| Pós-condições | |
| Cenários devem estar disponíveis, alterados ou excluídos, dependendo da opção escolhida pelo professor | |
| Fluxo Principal | |
| Ações do Ator | Ações do Sistema |
| 1 – Informar se deseja criar um cenário | 2 – Disponibilizar formulário para ser preenchido com dados a respeito do cenário |
| 3 – Preencher com informações referentes ao cenário | 4 – Verificar se todos os campos estão completos e corretos |
| 5 – Informar se deseja excluir um cenário | 6 – Exibir mensagem perguntando para o usuário se ele deseja realmente excluir o cenário |

| | |
|--|---|
| 7 – Confirmar se deseja excluir um cenário | 8 – Caso a resposta do usuário seja positiva, excluir o cenário |
| 9 – Informar se deseja alterar um cenário | 10 – Exibir mensagem para o usuário perguntando qual campo ele deseja alterar |
| 11 – Informar qual campo deseja alterar no cenário | 12 – Exibir mensagem para o usuário preencher com o novo dado |
| 13 – Alterar e confirmar a alteração | 14 – Salvar a alteração |
| Restrições e Validações | |
| Os valores para criação do cenário devem estar contidos no intervalo definido pelo administrador. No caso da exclusão e alteração, é necessário existir um cenário previamente cadastrado. | |
| Fluxo Alternativo I – Valores incorretos | |
| Ações do Ator | Ações do Sistema |
| | Informar ao usuário que o valor inserido não está no intervalo definido pelo administrador. Requisitar preenchimento de dados. Revalidar dados. |
| Fluxo de Exceção I – Inexistência de cenário | |
| Ações do Ator | Ações do Sistema |
| | Informar ao usuário que não há cenário para ser excluído e/ou alterado. |
| Responsável pela definição | Data da criação |
| Maria Teresa Menezes Costa | 16/06/2015 |

TABELA 6 – Caso de Uso Gerencia Cenários

| | |
|--|-------------------------|
| Nome do Caso de Uso | |
| Cadastra-se como usuário | |
| Ator Primário | |
| Aluno, Professor | |
| Atores Secundários | |
| Não tem | |
| Resumo | |
| Este caso de uso controla o cadastramento dos usuários | |
| Pré-condições | |
| Não tem | |
| Pós-condições | |
| Informar se o cadastro foi efetuado | |
| Fluxo Principal | |
| Ações do Ator | Ações do Sistema |
| 1 – Inserir dados | |
| 2 – Confirmar dados | |

| | |
|--|---|
| | 3 – Validar dados |
| | 4 – Requisitar preenchimento e exibir mensagem de erro caso algum campo não foi preenchido ou foi preenchido incorretamente |
| | 5 – Salvar o cadastro do usuário caso todos os dados estejam corretos |
| Restrições e Validações | |
| Não é permitido haver mais de um usuário com mesmo login | |
| Fluxo Alternativo I – Falha no cadastro | |
| Ações do Ator | Ações do Sistema |
| | Informar ao usuário que houve erro no cadastro. Requisitar preenchimento de dados. Revalidar dados. |
| Fluxo de Exceção I – Cadastro já existente | |
| Ações do Ator | Ações do Sistema |
| | Exibir mensagem informando que o usuário já existe. |
| Responsável pela definição | Data da criação |
| Álvaro Martins Espíndola | 16/06/2015 |

TABELA 7 – Caso de Uso Cadastra-se como Usuário

| | |
|---|--|
| Nome do Caso de Uso | |
| Escolhe cenário | |
| Ator Primário | |
| Aluno | |
| Atores Secundários | |
| Não tem | |
| Resumo | |
| Este caso de uso controla a escolha de cenários | |
| Pré-condições | |
| Possuir cenário disponível e funcionando corretamente | |
| Pós-condições | |
| Não tem | |
| Fluxo Principal | |
| Ações do Ator | Ações do Sistema |
| 1 – Escolher cenário | 2 – Verificar se o cenário está disponível e funcionando corretamente 3 – Exibir mensagem de erro caso algo esteja incorreto no cenário escolhido |
| | 4 – Iniciar jogo caso o cenário esteja correto e disponível |

| | |
|---|---|
| Restrições e Validações | |
| Não tem | |
| Fluxo Alternativo I – Escolher outro cenário | |
| Ações do Ator | Ações do Sistema |
| | Exibir mensagem para o usuário pedindo para escolher novo cenário caso haja erro com o cenário escolhido anteriormente. |
| Fluxo de Exceção I – Cenário Indisponível | |
| Ações do Ator | Ações do Sistema |
| | Exibir mensagem informando inexistência de cenário disponível. |
| Responsável pela definição | Data da criação |
| Helise de Assis Almeida | 17/06/2015 |

TABELA 8 – Caso de Uso Escolhe Cenário

| | |
|---|--|
| Nome do Caso de Uso | |
| Joga | |
| Ator Primário | |
| Aluno | |
| Atores Secundários | |
| Não tem | |
| Resumo | |
| Este caso de uso controla a jogabilidade | |
| Pré-condições | |
| Possuir cadastro | |
| Pós-condições | |
| Não tem | |
| Fluxo Principal | |
| Ações do Ator | Ações do Sistema |
| 1 – Escolher o cenário a ser jogado | |
| | 2 – Verificar se o cenário existe e está funcionando corretamente |
| | 3 – Iniciar jogo caso funcione |
| 4 – Examinar indicadores com nível baixo | |
| 5 – Escolher projetos/ações que irão estabilizar a situação | |
| | 6 – Controlar os níveis e as interferências dos fatores e dos indicadores. |
| Restrições e Validações | |
| Um cenário, no mínimo, deve estar disponível para o aluno | |
| Fluxo de Exceção I – Cenário Indisponível | |
| Ações do Ator | Ações do Sistema |

8.3. Anexo C: Classes detalhadas

Neste anexo, serão apresentadas com seus atributos e métodos as classes detalhadas, ilustradas nas figuras Figura 12, Figura 13, Figura 14, Figura 15, Figura 16, Figura 17, Figura 18, Figura 19, Figura 20, Figura 21, Figura 22, Figura 23, Figura 24, Figura 25, Figura 26, Figura 27, Figura 28, Figura 29, Figura 30, Figura 31, Figura 32, Figura 33, Figura 34, Figura 35, Figura 36, Figura 37, Figura 38, Figura 39, Figura 40, Figura 41, Figura 42, Figura 43, Figura 44 e Figura 45 citadas no diagrama de classes, que foi abordado anteriormente na Figura 9.

8.3.1. Model

Nesta seção são mostradas as figuras Figura 12, Figura 13, Figura 14, Figura 15, Figura 16, Figura 17, Figura 18, Figura 19, Figura 20, Figura 21, Figura 22, Figura 23, Figura 24 e Figura 25 que exibem as classes do projeto relacionadas ao módulo Model.

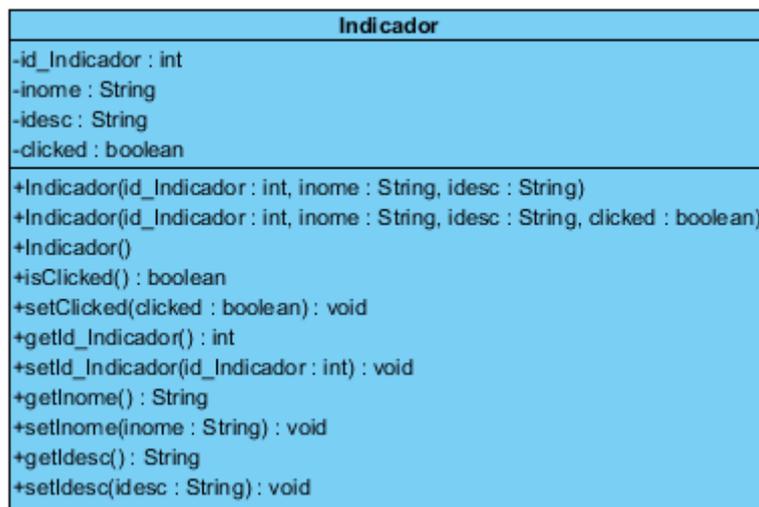


FIGURA 12 – Classe Indicador do software SimSUS

| Fator |
|---|
| -id_Fator : int -peso : int -fnome : String -fdesc : String -clicked : boolean |
| +Fator() +Fator(id_fator : int, peso : int, fnome : String, fdesc : String, clicked : boolean) +Fator(id_fator : int, fnome : String, fdesc : String, peso : int) +Fator(fnome : String, fdesc : String) +getId_Fator() : int +getFnome() : String +setFnome(fnome : String) : void +getFdesc() : String +setFdesc(fdesc : String) : void +isClicked() : boolean +setClicked(clicked : boolean) : void +getPeso() : int +setPeso(peso : int) : void |

FIGURA 13 – Classe Fator do software SimSUS

| Cenario |
|---|
| -id_Cenario : int -id_Turma : int -renda : float |
| +Cenario(id_Cenario : int, id_Turma : int, renda : float) +Cenario() +getId_Cenario() : int +setId_Cenario(id_Cenario : int) : void +getId_Turma() : int +setId_Turma(id_Turma : int) : void +getRenda() : float +setRenda(renda : float) : void |

FIGURA 14 – Classe Cenario do software SimSUS

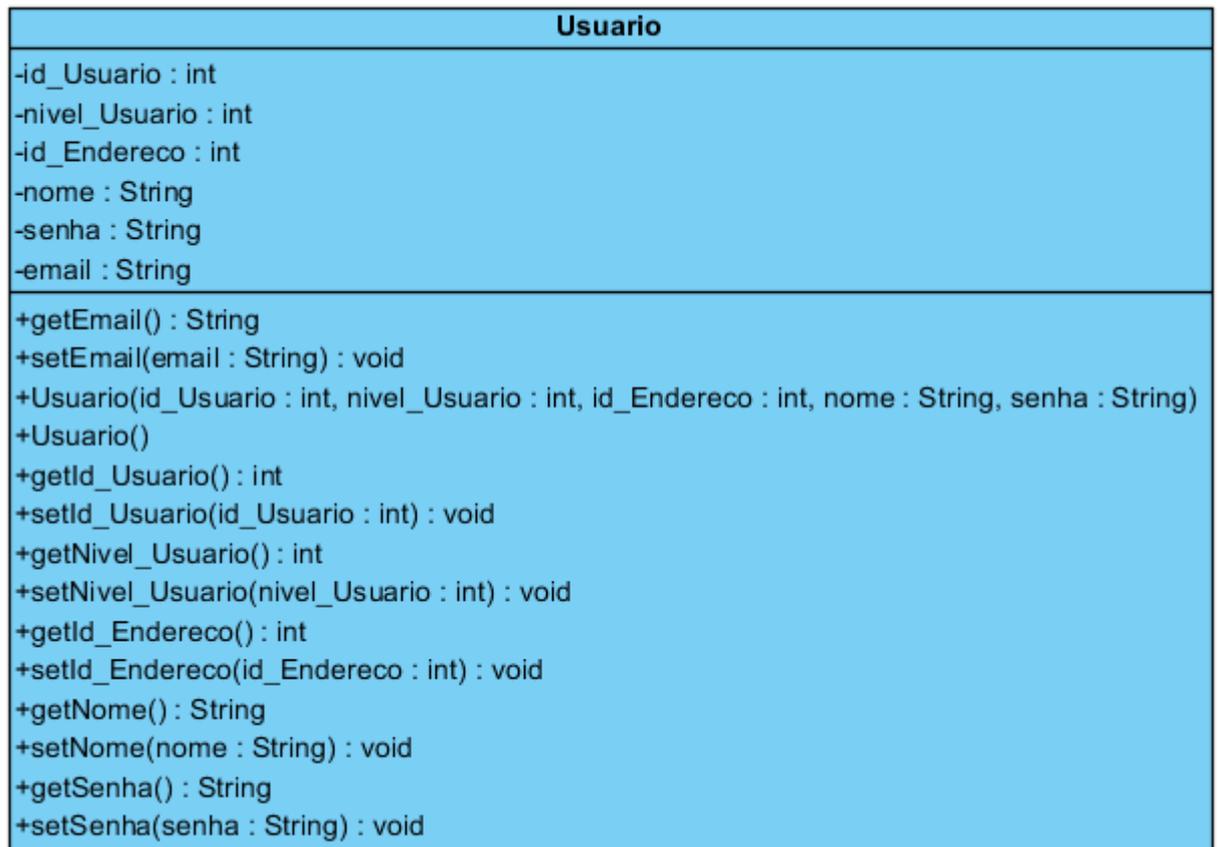


FIGURA 15 – Classe Usuario do software SimSUS



FIGURA 16 – Classe Nivel_Usuario do software SimSUS

| Solucao |
|--|
| -id_Solucao : int -dnome : String -ddesc : String -custo : float |
| +Solucao(id_Solucao : int, dnome : String, ddesc : String, custo : float) +Solucao() +getId_Solucao() : int +setId_Solucao(id_Solucao : int) : void +getDnome() : String +setDnome(dnome : String) : void +getDdesc() : String +setDdesc(ddesc : String) : void +getCusto() : float +setCusto(custo : float) : void |

FIGURA 17 – Classe Solucao do software SimSUS

| Turma |
|---|
| -id_Turma : int -nome_Turma : String |
| +Turma(id_Turma : int, nome_Turma : String) +Turma() +getId_Turma() : int +setId_Turma(id_Turma : int) : void +getNome_Turma() : String +setNome_Turma(nome_Turma : String) : void |

FIGURA 18 – Classe Turma do software SimSUS

| Cenario_has_Fator |
|--|
| -id_Cenario : int -id_Fator : int |
| +Cenario_has_Fator(id_Cenario : int, id_Fator : int) +Cenario_has_Fator() +getId_Cenario() : int +setId_Cenario(id_Cenario : int) : void +getId_Fator() : int +setId_Fator(id_Fator : int) : void |

FIGURA 19 – Classe Cenario_has_Fator do software SimSUS

| Cenario_has_Indicador |
|--|
| -id_Indicador : int -id_Cenario : int |
| +Cenario_has_Indicador(id_Indicador : int, id_Cenario : int) +Cenario_has_Indicador() +getId_Indicador() : int +setId_Indicador(id_Indicador : int) : void +getId_Cenario() : int +setId_Cenario(id_Cenario : int) : void |

FIGURA 20 – Classe Cenario_has_Indicador do software SimSUS

| Fator_Fator |
|---|
| -id_Fator_Fator : int -id_Fator : int -id_Fator_Interferido : int -taxa_int : float |
| +Fator_Fator(id_Fator_Fator : int, id_Fator : int, id_Fator_Interferido : int, taxa_int : float) +Fator_Fator(id_Fator : int, id_Fator_Interferido : int, taxa_int : float) +getId_Fator_Fator() : int +getId_Fator() : int +setId_Fator(id_Fator : int) : void +getId_Fator_Interferido() : int +setId_Fator_Interferido(id_Fator_Interferido : int) : void +getTaxa_int() : float +setTaxa_int(taxa_int : float) : void |

FIGURA 21 – Classe Fator_Fator do software SimSUS

| Indicador_Fator |
|---|
| -id_Indicador : int -id_Fator : int -taxa_Interferencia : float |
| +Indicador_Fator(id_Indicador : int, id_Fator : int, taxa_Interferencia : float) +Indicador_Fator() +getId_Indicador() : int +setId_Indicador(id_Indicador : int) : void +getId_Fator() : int +setId_Fator(id_Fator : int) : void +getTaxa_Interferencia() : float +setTaxa_Interferencia(taxa_Interferencia : float) : void |

FIGURA 22 – Classe Indicador_Fator do software SimSUS

| Indicador_Solucao |
|---|
| -id_Indicador : int -id_Solucao : int -taxa_Interferencia : float |
| +Indicador_Solucao(id_Indicador : int, id_Solucao : int, taxa_Interferencia : float) +Indicador_Solucao() +getId_Indicador() : int +setId_Indicador(id_Indicador : int) : void +getId_Solucao() : int +setId_Solucao(id_Solucao : int) : void +getTaxa_interferencia() : float +setTaxa_interferencia(taxa_Interferencia : float) : void |

FIGURA 23 – Classe Indicador_Solucao do software SimSUS

| Solucao_Fator |
|---|
| -id_Solucao : int -id_Fator : int -taxa_interferencia : float |
| +Solucao_Fator(id_Solucao : int, id_Fator : int, taxa_interferencia : float) +Solucao_Fator() +getId_Solucao() : int +setId_Solucao(id_Solucao : int) : void +getId_Fator() : int +setId_Fator(id_Fator : int) : void +getTaxa_interferencia() : float +setTaxa_interferencia(taxa_interferencia : float) : void |

FIGURA 24 – Classe Solucao_Fator do software SimSUS

| Usuario_has_Turma |
|---|
| -id_Usuario_has_turma : int -id_Usuario : int -id_Turma : int |
| +Usuario_has_Turma(id_Usuario_has_turma : int, id_Usuario : int, id_Turma : int) +Usuario_has_Turma() +getId_Usuario_has_turma() : int +setId_Usuario_has_turma(id_Usuario_has_turma : int) : void +getId_Usuario() : int +setId_Usuario(id_Usuario : int) : void +getId_Turma() : int +setId_Turma(id_Turma : int) : void |

FIGURA 25 – Classe Usuario_has_Turma do software SimSUS

8.3.2. Controller

As figuras Figura 26, Figura 27, Figura 28, Figura 29 e Figura 30, apresentadas nesta seção, evidenciam as classes do SimSUS referente ao módulo Controller.

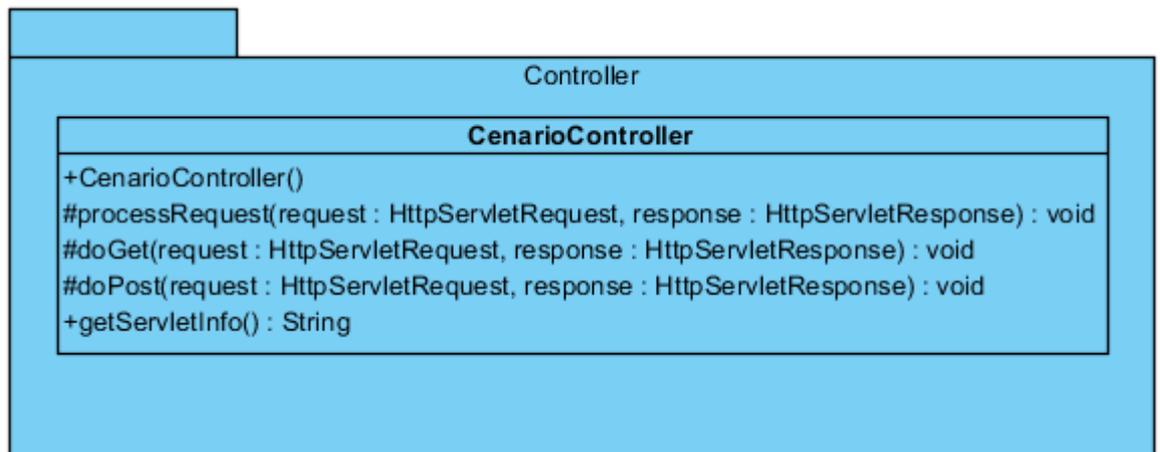


FIGURA 26 – Classe CenarioController do software SimSUS

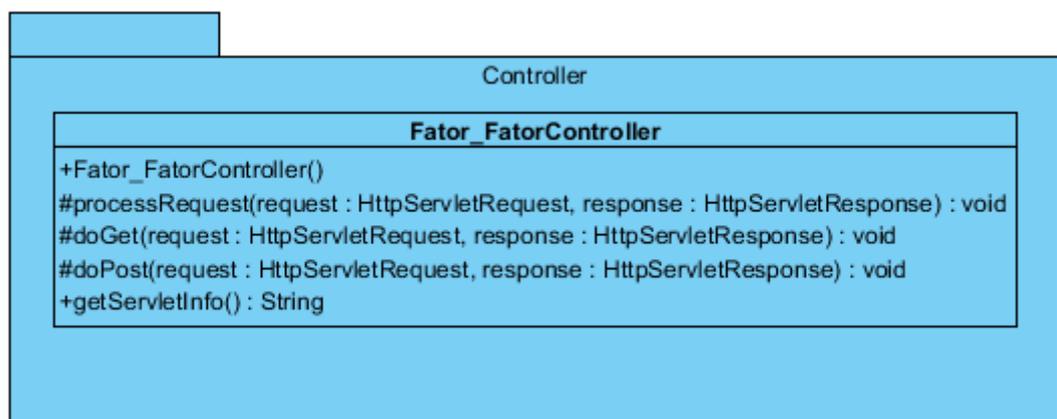


FIGURA 27 – Classe Fator_FatorController do software SimSUS

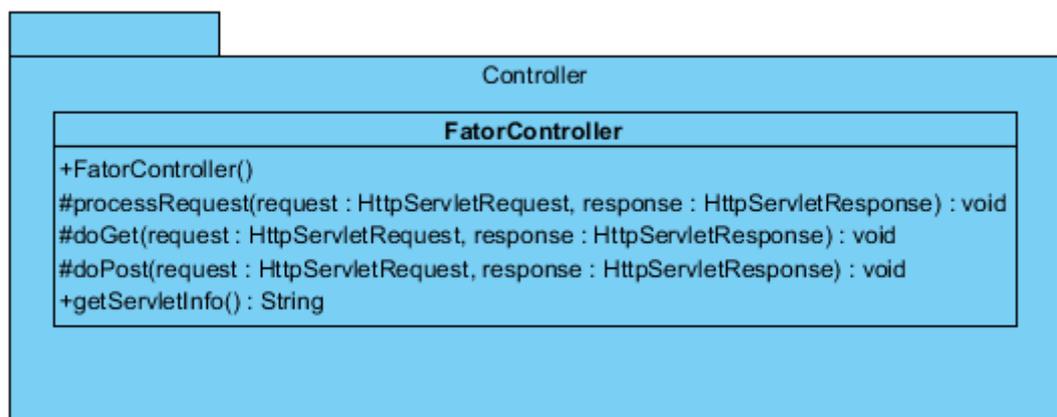


FIGURA 28 – Classe CenarioController do software SimSUS

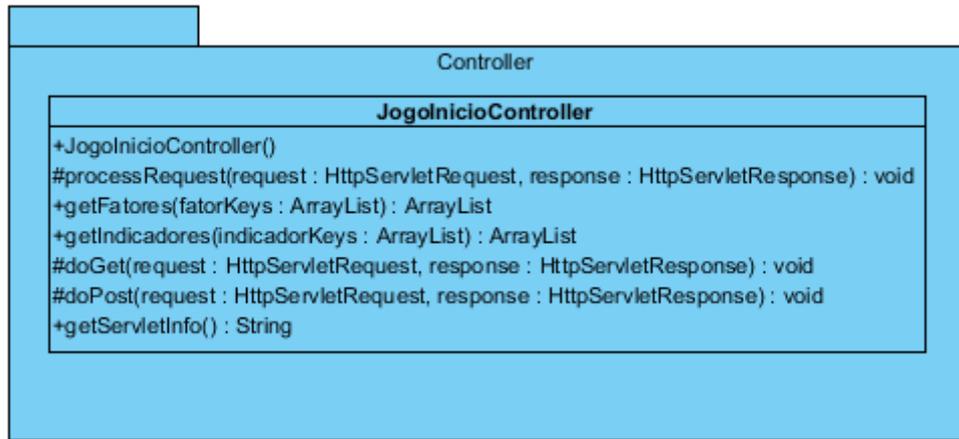


FIGURA 29 – Classe JogInicioController do software SimSUS

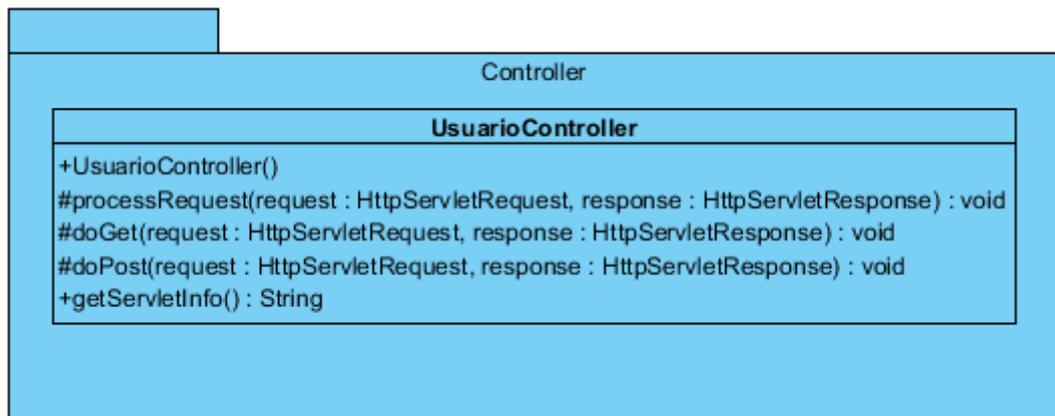


FIGURA 30 – Classe UsuarioController do software SimSUS

8.3.3. DAO

Neste tópico, estão presentes as classes do sistema relativas ao módulo DAO ilustradas pelas figuras Figura 31, Figura 32, Figura 33, Figura 34, Figura 35, Figura 36, Figura 37, Figura 38, Figura 39 e Figura 40.

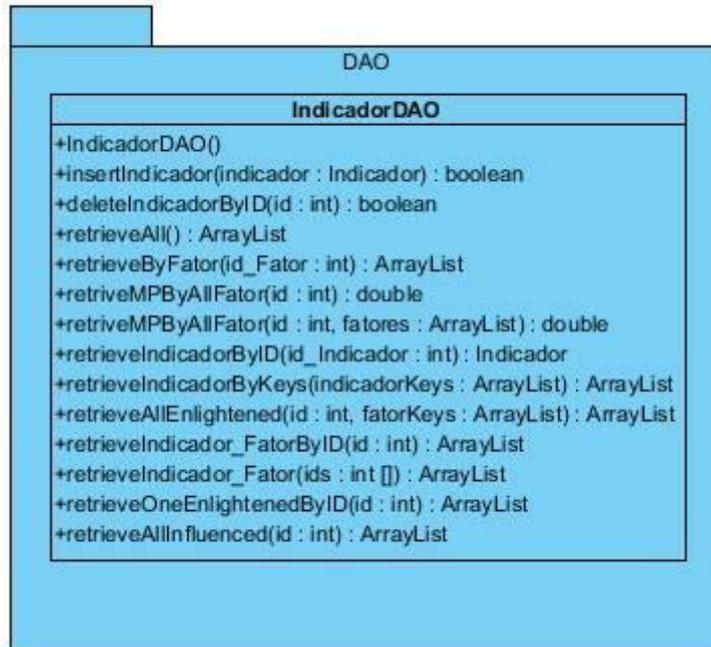


FIGURA 31 – Classe IndicadorDAO do software SimSUS

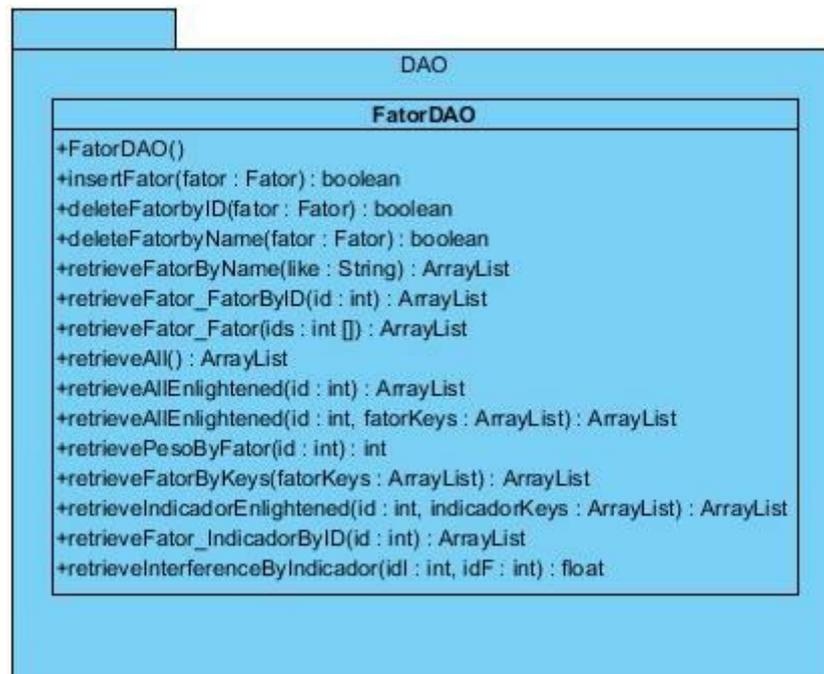


FIGURA 32 – Classe FatorDAO do software SimSUS



FIGURA 33 – Classe CenarioDAO do software SimSUS

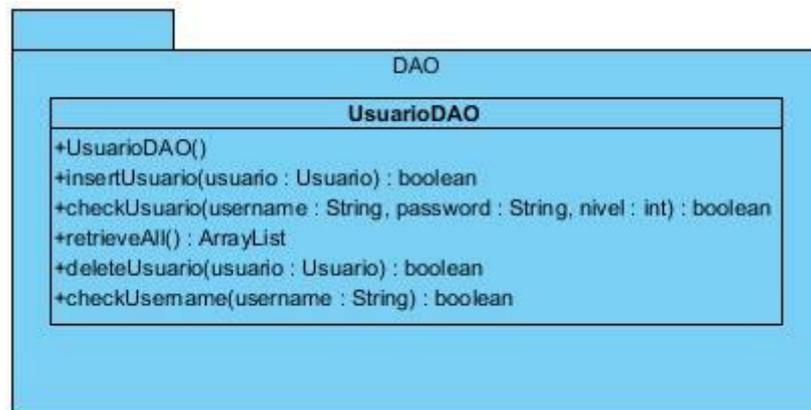


FIGURA 34 – Classe UsuarioDAO do software SimSUS



FIGURA 35 – Classe Fator_FatorDAO do software SimSUS

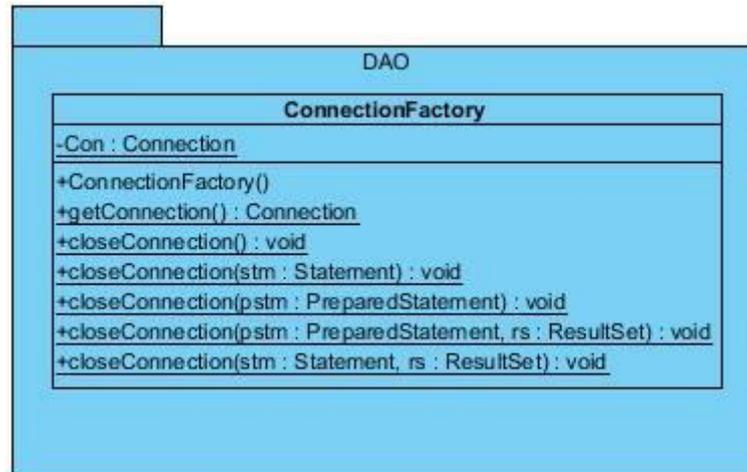


FIGURA 36 – Classe ConnectionFactory do software SimSUS

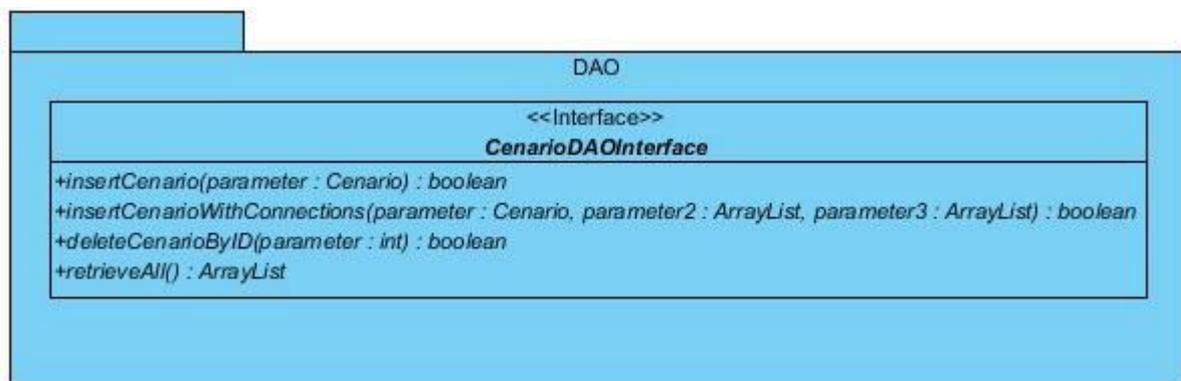


FIGURA 37 – Classe CenarioDAOInterface do software SimSUS



FIGURA 38 – Classe Fator_FatorDAOInterface do software SimSUS

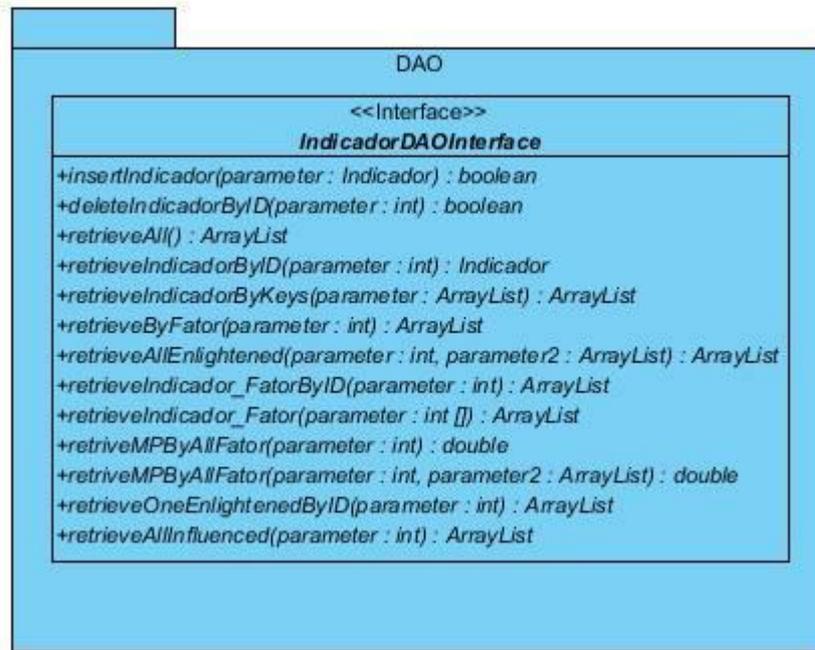


FIGURA 39 – Classe IndicadorDAOInterface do software SimSUS

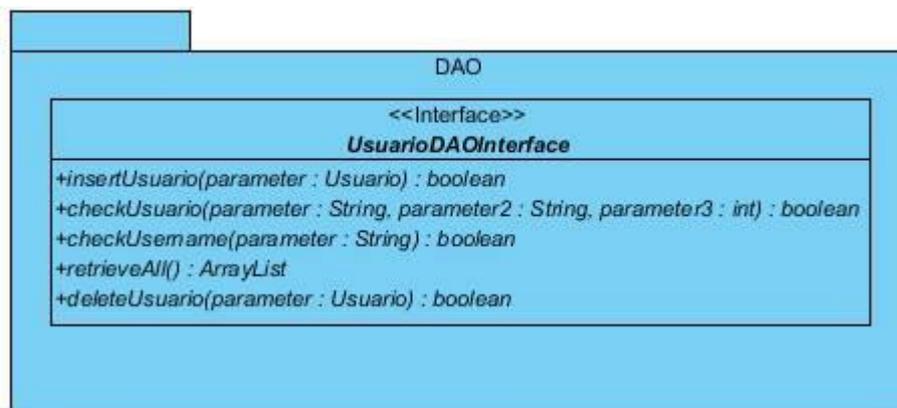


FIGURA 40 – Classe UsuarioDAOInterface do software SimSUS

8.3.4. Exception

Nesta seção, serão apresentadas, por meio das Figuras 41, 42, 43 44 e 45, as classes do *software* SimSUS correspondentes ao módulo *Exception*.

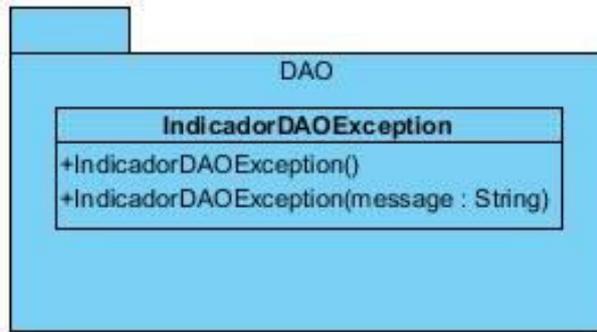


FIGURA 41 – Classe IndicadorDAOException do software SimSUS

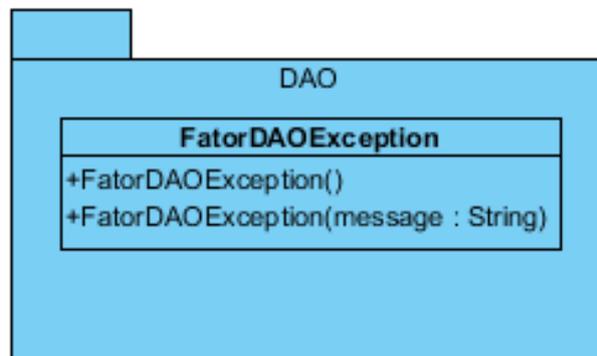


FIGURA 42 – Classe FatorDAOException do software SimSUS

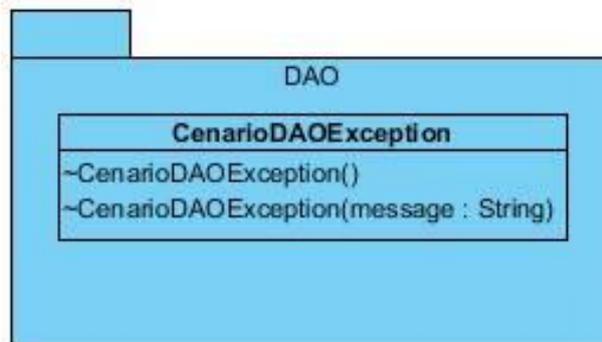


FIGURA 43 – Classe CenarioDAOException do software SimSUS

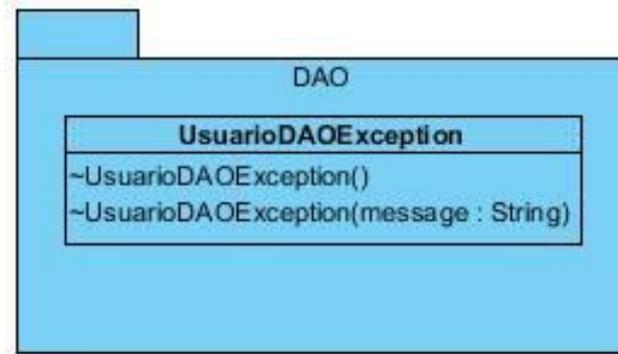


FIGURA 44 – Classe UsuarioDAOException do software SimSUS

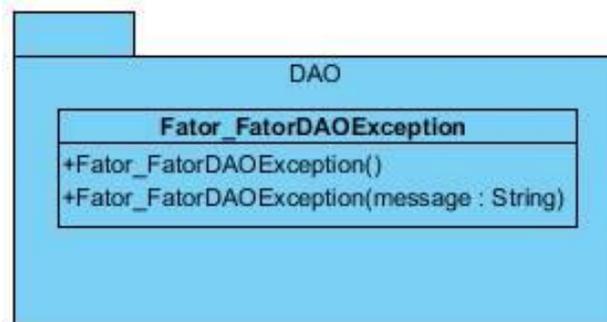


FIGURA 45 – Classe Fator_FatorDAOException do software SimSUS