

Engenharia de Software 2

Projeto – Escalonamento de Enfermeiros por Bloco Operatório

Bruno Santos Nº1012388

Leonardo Lourenço Nº1012395

Luciano Brido Nº 1011881

Índice

Introdução e Descrição do tema do projeto………………………………………………..pag.3

>Atividades e tempos gastos em horas por elemento de grupo(tabela ou gráfico)………………………………………………………………………………………………………pág. 3 a 4

>Diagrama de Contexto……………………………………………………………………………..pag.5

Descrição dos três padrões ……………………………………………….………………………pág.5 a 7

Tabela de Atores, objetivos e respetivos Casos de Uso ………………………..…...pág.8

Descrição dos atores………………………………………………………………....................pág.9

Diagrama de Casos de Uso………………………………………………………………………...pág.10 a 12

Descrição dos Casos de Uso…………………….…..................................................pág.13 a 29

Algoritmo(s) para gerar horários…………………………………………………………………pág.29

Diagramas de Sequência dos Casos de Uso………………………………………………...pág.30 a 32

Diagrama de Classes…………………………………………………………………………………..pág.33

Diagrama de Estados………………………………………………………………………………….pág.34

Semântica de Classes: ……………………………………………….......................pág.35 a 43

* Dicionário de Dados
* Operações
* Algoritmo de validação do Cartão de cidadão ……………………..pág.37 e 38
* Diagramas de sequência a que pertence

Diagrama de Atividades ……………………………………………………………………………..pág.44

Diagrama de Pacotes de Componentes……………………………………………………….pág.45

Diagrama de Instalação……………………………………………………………………………….pág.46

Protótipos …………………………………………………………………………………………………..pág.47 a 60

Conclusão, Anexos…………………………………………………………………………………………pág.61 a 63

Introdução e Descrição do tema do projeto

Neste trabalho proposto nas cadeiras, Engenharia de Software 2 e Programação para a Internet, do curso Engenharia Informática temos como objetivo a criação de um software que realize o escalonamento de enfermeiros por bloco operatório.

Os pontos alvo do nosso projeto são a criação dos horários dos enfermeiros recorrendo a um algoritmo que tem em conta as regras impostas pela legislação e o bloco operatório desejado. Tem também a possibilidade dos enfermeiros poderem realizar trocas.

Atividades e tempos gastos em horas por elemento de grupo <<ADICIONAR GRAFICO COM ESTES VALORES>>

Leonardo Lourenço:

Organização e Planeamento – 44H

Desenvolvimento dos Atores e respetivos casos de uso – 15H

Desenvolvimento da Descrição dos Atores – 2H

Desenvolvimento de Diagramas de caso de uso – 15H

Desenvolvimento das Descrição dos casos de usos – 22H

Desenvolvimento de Diagramas de Sequência – 15H

Desenvolvimento do Diagramas de Classes – 5H

Desenvolvimento do Diagrama de Estados – 8H

Desenvolvimento da Semântica de Classes – 12H

Desenvolvimento do Diagrama de Atividades – 3H

Desenvolvimento do Diagrama de Pacotes de Componentes – 6H

Desenvolvimento do Diagrama de Instalação – 6H

Desenvolvimento de Protótipos – 14H

Padrão do caso de Uso: BreathBeforeDepth – 5H

CRUD Horário – 25H

CRUD Blocos Operatórios – 10H

CRUD Blocos Operatórios - Turnos – 15H

CRUD Turnos – 10H

CRUD Enfermeiros – 10H

CRUD Regras – 10H

CRUD Pedidos de Troca – 40H

Regular Expressions – 15H

Layout das Páginas – 8H

Roles – 3H

Função para a criação Dinâmica dos pedidos de troca – 20H

Mensagens de Alerta e Confirmação – 10H

Autenticação – 10H

Limpeza de Código – 8H

Bruno Santos:

CRUD Horário-10H

CRUD Turnos-10H

CRUD Blocos Operatórios –6H

CRUD Turnos – 2H

CRUD Enfermeiros – 1H

Luciano Brido:

Desenvolvimento de Diagramas de contexto – 2H

Desenvolvimento de Diagramas de caso de uso – 6H  
CRUD Enfermeiros – 15H

CRUD Regras – 2H

Diagrama de Contexto

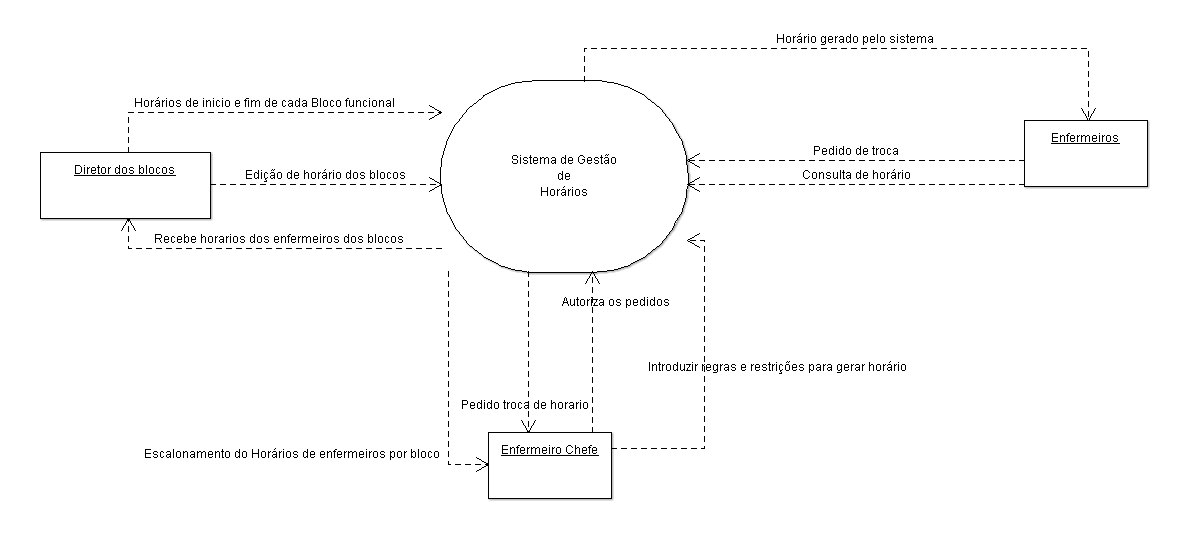


Figura.1

Descrição dos três padrões

BreathBeforeDepth:

O processo de desenvolvimento de casos de uso é um processo em que se está constantemente a aprender e a reavaliar o nosso modelo.

Este processo é muito parecido á forma como antigamente se faziam as casas, não tinham planos detalhados de como seria a casa, iam construído os quartos segundo as necessidades, tanto que nas casas mais antigas existem quartos que apenas podemos entrar neles através de um outro quarto.

É preciso pensar nos casos de uso mas não ao mais afincado pormenor.

A equipa de desenvolvimento esgotar-se-á facilmente se detalharmos as coisas cedo e de forma complexa, depressa perdem o interesse em faze-lo. Se começarmos com algo muito complexo e necessitarmos de mudar algo mais tarde, teremos de mudar mais coisas nos nossos casos de uso. Só quando a nossa visão geral estiver completa e conhecermos mais o sistema que estamos a trabalhar, devemos a começar a aumentar a precisão do delineamento dos casos de uso.

O BreadthBeforeDepth indica a ordem para o processo de escrita dos casos de uso.

Às vezes, é necessário concluir alguns casos de uso antes dos outros. Nesse caso, precisamos de uma abordagem incremental com base na prioridade do projeto. Identificamos os casos de uso mais importantes, como aqueles que definem o projeto ou aqueles que precisam ser feitos primeiro, com base em problemas comerciais ou técnicos. É necessário a explicitação de cada caso de uso para que se este não for realmente necessário eliminá-lo.

O BreadthBeforeDepth, é isso mesmo, como o nome indica Respirar Antes de Aprofundar, ou seja, pensar primeiro de forma calma no aspeto global dos nossos casos de uso e depois adicionar detalhes progressivamente.

A visão geral permite ver o todo sem ficar preso nos detalhes. Trabalhar em alto nível facilita o esboço do sistema e dos seus limites, além de verificar se os seus casos de uso satisfazem todas as funcionalidades do sistema. Esta visão de alto nível também permite que iniciemos o planeamento do projeto, o estabelecimento das prioridades do nosso trabalho, e o desenvolvimento dos casos de uso mais importantes primeiro e os menos importantes depois.

PreciseAndReadable:

Um caso de uso deve ser compreensível para os stakeholders e para os programadores, isto porque o caso de uso pode ser para vários tipos de nível de habilidades.

Um caso de uso descrito muito tecnicamente e ou muito detalhadamente pode ser útil para os programadores, mas os próprios clientes podem não compreender e podem-no descartar por não perceberem do que se trata.

Portanto um caso de uso também pode ser o oposto ser pouco detalhado mas de fácil compreensão, o que pode tornar o caso difícil de implementar, já que o programador não sabe exatamente o que é pedido.

Por estas duas razões, um caso de uso para ser bem escrito tem de satisfaz todas as audiências sem favorecer nenhum dos grupos).

Quando um caso de uso está incompleto para os programadores, eles têm a tendência de adicionar detalhe e soluções, o que ajuda a complicar o caso de uso, mas ao mesmo tempo isto torna-o difícil de entender para o cliente, criando desacordo por não entenderem a racionalidade da solução e ou detalhe.

A solução apresentada pelo autor é que haja o diálogo entre os envolvidos, para que todos compreendem como funciona, pois o objetivo dos casos de uso é determinar o que o sistema deve fazer para o cliente e que como o sistema deve funcionar para os programadores implementarem.

A conclusão a que chega é que um caso de uso preciso e compreensível é um caso em que há um consenso de todos os envolvidos.

Para a construção do próprio caso de uso, o autor propôs que seja escrito com um único objetivo em mente, outra proposta para o escritor do caso é saber a que vai ler e usar a linguagem do leitor. O caso de uso não deve deixar espaço para interpretação, usar vários exemplos de como o sistema deve funcionar.

LeveledSteps:

Etapas de casos de uso excessivamente grandes ou excessivamente pequenas ocultam o objetivo e dificultam a leitura e compreensão do caso de uso. O padrão LeveledSteps descreve como escrever casos de uso equilibrados e razoavelmente dimensionados.

Qualquer coisa pode ser descrita em etapas pequenas que tornam o caso de uso cada vez menor perdendo o sentido e a intenção de detalhe.

O oposto também poderá acontecer qualquer coisa descrita em etapas excessivamente grandes podem ocultar um comportamento importante, omitindo ações chave que os desenvolvedores devem conhecer.

Misturar níveis de detalhe pode levar a uma distração do que é suposto estar a acontecer, e dificulta a interpretação correta das instruções.

Cada cenário deve ter de três a nove etapas. Idealmente, eles estão todos em níveis semelhantes, um nível de abstração logo abaixo do objetivo do caso de uso

Tabela de Atores, objetivos e respetivos Casos de Uso

|  |  |  |
| --- | --- | --- |
| **Ator** | **Casos de Uso** | **Objetivo** |
| **Diretor**  **Dos**  **Blocos** | Gerar Horário | Realizar a criação do horário do bloco x segundo as regras impostas pelo Enfermeiro Chefe. |
| Visualizar horários | Visualizar horários dos blocos operatórios. |
| Gerir enfermeiros | Criar, editar, visualizar e eliminar dados pessoais dos enfermeiros. |
| Gerir Especialidades | Criar, editar, visualizar e eliminar especialidades. |
| Gerir Turnos | Criar, editar, visualizar e eliminar turnos. |
| Gerir Blocos Operatórios | Criar, editar, visualizar e eliminar turnos. |
| Gerir a ligação entre Blocos Operatórios e Turnos | Criar, editar, visualizar e eliminar a ligação entre blocos operatórios e turnos. |
| **Enfermeiro**  **Chefe** | Introduzir regras e restrições | Para gerar horário corretamente. Ex: enfermeiros escolhidos, etc. |
| Pedir trocas de turnos | Realizar o pedido de trocas de turnos no horário x. |
| Validar trocas | Autorizar ou não os pedidos de trocas. |
| Visualizar horários | Visualizar horários dos blocos operatórios. |
| **Enfermeiros** | Pedir trocas de turnos | Realizar o pedido de trocas de turnos no horário x. |
| Visualizar horário | Visualizar horários dos blocos operatórios. |

Descrição dos atores

No nosso software temos 3 tipos de utilizadores:

Diretor dos Blocos: A quem cabe gerar os horários e certificar que as trocas entre enfermeiros são feitas segundo a legislação, nível da hierarquia alto.

Enfermeiro Chefe: A quem cabe introduzir as regras e restrições e validar as trocas, nível de hierarquia médio.

Enfermeiro: Pode visualizar o seu horário e pedir trocas de turnos, nível de hierarquia baixo.

Diagrama de Casos de Uso

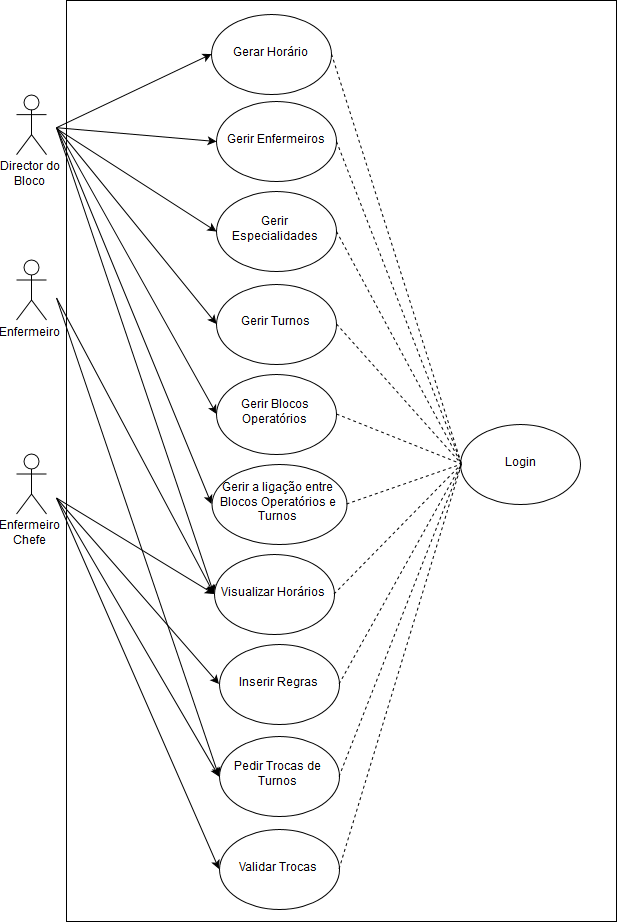


Figura.2

Diagrama de Pacotes de Casos de Uso

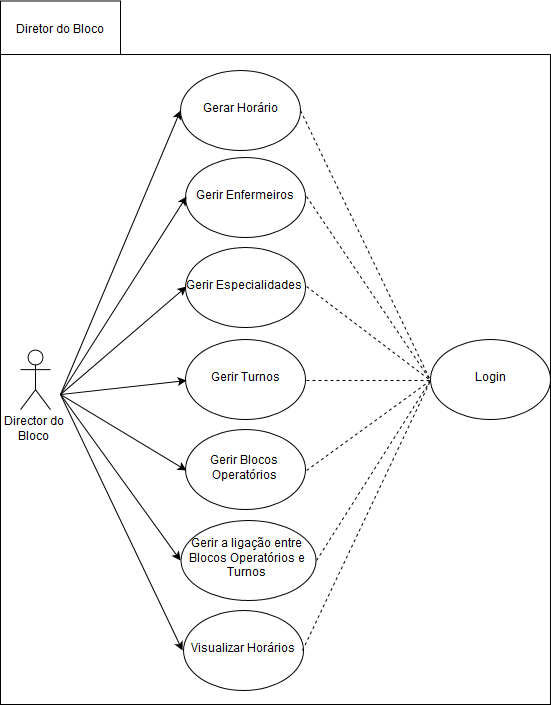


Figura.3

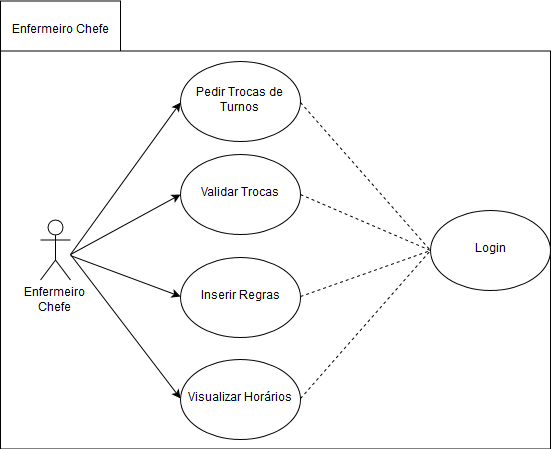


Figura.4

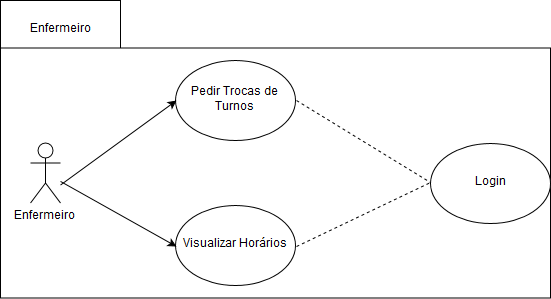


Figura.5

Descrição de todos os Casos de Uso

|  |  |
| --- | --- |
| Nome do caso de uso | RF01, Autenticar Ator (Login) |
| Descrição do requisito | O objetivo deste caso de uso é controlar o acesso ao sistema de software |
| Tamanho | 21 |
| Pré-condição |  |
| Cenário Principal |  |
|  | 1-Acede ao software.  2-Pede a sua identificação (login e password).  3-Introduz login e password.  4-Apresenta o ambiente de trabalho. |
| Cenário Secundário | 4a-Se o email de utilizador estiver errado o sistema volta a pedia a identificação. 4a-Se o email de utilizador estiver errado o sistema volta a pedia a identificação.  4b-Se a password estiver incorreta o sistema volta a pedir a sua password. |
| Requisitos não funcionais |  |

|  |  |
| --- | --- |
| Nome do caso de uso | RF02, Criar Enfermeiro |
| Descrição do requisito | O objetivo deste caso de uso é criar novo enfermeiro |
| Tamanho | 8 |
| Pré-condição | O diretor tem o login válido. |
| Cenário Principal |  |
|  | 1-O diretor seleciona a opção criar novo enfermeiro  2-O sistema abre o formulário com os dados a introduzir  3-O diretor insere os atributos nome, email, especialidade, tipo de enfermeiro, número de telefone, número de identificação de identidade, data de nascimento, data de nascimento do filho mais novo e escolhe a especialidade.  4-O diretor carrega na opção criar.  5-O sistema atribui automaticamente o ID ao novo enfermeiro.  6-O sistema dá sucesso. |
| Cenário Alternativo | 2a-O sistema tem opção de voltar à lista  3a-Se o atributo nome estiver nulo o sistema pede para inserir um nome.  3b- Se o atributo nome estiver incorreto o sistema pede para inserir um nome válido  3c- Se o atributo email estiver nulo o sistema pede para inserir um email.  3d- Se o atributo email estiver incorreto o sistema pede para inserir um email válido.  3e- Se o atributo tipo estiver nulo o sistema pede para inserir um tipo.  3f- Se o atributo tipo estiver incorreto o sistema pede para inserir um número.  3g- Se o atributo número de telefone estiver nulo o sistema pede para inserir número de telefone.  3h- Se o atributo número de telefone estiver incorreto o sistema pede para inserir um número de telefone válido.  3i- Se o atributo número de identificação de identidade estiver nulo o sistema pede para inserir número de identificação de identidade.  3j- Se o atributo número de identificação de identidade estiver incorreto o sistema pede para inserir um número de identificação de identidade válido.  3k- Se o atributo data de nascimento estiver nulo o sistema pede para inserir uma data de nascimento.  3l- Se o atributo data de nascimento estiver incorreta o sistema pede para inserir uma data de nascimento válida.  3m- Se o atributo data de nascimento do filho mais novo estiver incorreta o sistema pede para inserir uma data de nascimento válida. |
| Requisitos não funcionais | * Testar se o sistema deixa registar o enfermeiro sem os campos obrigatórios estarem preenchidos. * Testar campo IDCard através do algoritmo Dígito de controlo. * Teste1-Validar email   Utilizador insere no Email = skgeldgjeç  O sistema notifica – O Email tem de ter um domínio e um subdomínio válido (exemplo [nomedoutilizador@nomedosubdominio.nomedodominio](mailto:nomedoutilizador@nomedosubdominio.nomedodominio));   * Teste2– Validação do Contacto   Utilizador insere – Contacto = 8154866848egege84e  O sistema notifica – O contacto apenas pode conter caracteres numéricos.   * Teste3– Validação do Nome   Utilizador insere – Name = Alfredo012  O sistema notifica – O Nome apenas pode conter caracteres.   * Teste4– Validação do Tipo   Utilizador insere – Type = e  O sistema notifica – Por favor insira um número.   * Teste5– Validação do Número de Telefone   Utilizador insere – CellPhoneNumber = 999888777  O sistema notifica – Por favor insira um número de telefone válido.   * Teste5– Validação da Data de Nascimento   Utilizador insere – BirthDate = “ “  O sistema notifica – Por favor insira uma data de nascimento.   * Teste6– Validação da Data de Nascimento do Filho Mais Novo   Utilizador insere – BirthDate = 1/10/1987  Utilizador insere – YoungestChildBirthDate = 1/10/1986  O sistema notifica – A data de nascimento do filho mais novo não pode anteceder a data de nascimento do progenitor. |

|  |  |
| --- | --- |
| Nome do caso de uso | RF03, Editar perfil dos Enfermeiros |
| Descrição do requisito | O objetivo deste caso de uso é editar perfil do enfermeiro |
| Tamanho | 8 |
| Pré-condição | O diretor tem de ter o login válido. |
| Cenário Principal |  |
|  | 1-O diretor seleciona a opção editar enfermeiros no enfermeiro que pretende editar.  2-O sistema abre o formulário com os dados já preenchidos.  3-O diretor altera os atributos que pretende alterar.  4-O diretor carrega na opção criar.  5-O sistema atribui automaticamente o ID ao novo enfermeiro.  6-O sistema dá sucesso. |
| Cenário Alternativo | 2a-O sistema tem opção de voltar à lista  3a-Se o atributo nome estiver nulo o sistema pede para inserir um nome.  3b- Se o atributo nome estiver incorreto o sistema pede para inserir um nome válido  3c- Se o atributo email estiver nulo o sistema pede para inserir um email.  3d- Se o atributo email estiver incorreto o sistema pede para inserir um email válido.  3e- Se o atributo tipo estiver nulo o sistema pede para inserir um tipo.  3f- Se o atributo tipo estiver incorreto o sistema pede para inserir um número.  3g- Se o atributo número de telefone estiver nulo o sistema pede para inserir número de telefone.  3h- Se o atributo número de telefone estiver incorreto o sistema pede para inserir um número de telefone válido.  3i- Se o atributo número de identificação de identidade estiver nulo o sistema pede para inserir número de identificação de identidade.  3j- Se o atributo número de identificação de identidade estiver incorreto o sistema pede para inserir um número de identificação de identidade válido.  3k- Se o atributo data de nascimento estiver nulo o sistema pede para inserir uma data de nascimento.  3l- Se o atributo data de nascimento estiver incorreta o sistema pede para inserir uma data de nascimento válida.  3m- Se o atributo data de nascimento do filho mais novo estiver incorreta o sistema pede para inserir uma data de nascimento válida. |
| Requisitos não funcionais | * Testar se o sistema deixa registar o enfermeiro sem os campos obrigatórios estarem preenchidos. * Testar campo IDCard através do algoritmo Dígito de controlo. * Teste1-Validar email   Utilizador insere no Email = skgeldgjeç  O sistema notifica – O Email tem de ter um domínio e um subdomínio válido (exemplo [nomedoutilizador@nomedosubdominio.nomedodominio](mailto:nomedoutilizador@nomedosubdominio.nomedodominio));   * Teste2– Validação do Contacto   Utilizador insere – Contacto = 8154866848egege84e  O sistema notifica – O contacto apenas pode conter caracteres numéricos.   * Teste3– Validação do Nome   Utilizador insere – Name = Alfredo012  O sistema notifica – O Nome apenas pode conter caracteres.   * Teste4– Validação do Tipo   Utilizador insere – Type = e  O sistema notifica – Por favor insira um número.   * Teste5– Validação do Número de Telefone   Utilizador insere – CellPhoneNumber = 999888777  O sistema notifica – Por favor insira um número de telefone válido.   * Teste5– Validação da Data de Nascimento   Utilizador insere – BirthDate = “ “  O sistema notifica – Por favor insira uma data de nascimento.   * Teste6– Validação da Data de Nascimento do Filho Mais Novo   Utilizador insere – BirthDate = 1/10/1987  Utilizador insere – YoungestChildBirthDate = 1/10/1986  O sistema notifica – A data de nascimento do filho mais novo não pode anteceder a data de nascimento do progenitor. |

|  |  |
| --- | --- |
| Nome do caso de uso | RF04, Eliminar Enfermeiro |
| Descrição do requisito | O objetivo deste caso de uso é eliminar enfermeiro. |
| Tamanho | 15 |
|  |  |
| Pré-condição | O diretor tem de ter o login válido.  Tem que existir uma lista de enfermeiros no sistema. |
| Cenário Principal |  |
|  | 1-O diretor seleciona eliminar no enfermeiro que pretende apagar.  2-O sistema abre uma janela com os detalhes do enfermeiro.  3-O diretor seleciona o botão delete para eliminar o enfermeiro.  4-O sistema pede para confirmar que pretende eliminar o bloco operatório.  5-O sistema devolve uma mensagem de sucesso. |
| Cenário Alternativo | 2a-O sistema tem um botão de voltar à lista.  3a- Se o enfermeiro tiver um horário não pode ser eliminado, o sistema devolve uma mensagem de erro. |
| Requisitos não funcionais |  |

|  |  |
| --- | --- |
| Nome do caso de uso | RF05, Criar Especialidade |
| Descrição do requisito | O objetivo deste caso de uso é criar uma nova especialidade |
| Tamanho | 8 |
| Pré-condição | O diretor tem o login válido. |
| Cenário Principal |  |
|  | 1-O diretor seleciona a opção criar nova especialidade  2-O sistema abre o formulário com os dados a introduzir  3-O diretor insere o atributo nome.  4-O diretor carrega na opção criar.  5-O sistema atribui automaticamente o ID à nova especialidade e devolve uma mensagem de sucesso. |
| Cenário Alternativo | 2a-O sistema tem opção de voltar à lista  3a-Se o atributo nome estiver nulo o sistema pede para inserir um nome.  3b-Se já houver uma especialidade cujo nome seja igual ao atributo nome o sistema pede para inserir um nome diferente. |
| Requisitos não funcionais | * Testar se o sistema deixa registar a especialidade sem os campos obrigatórios estarem preenchidos. * Teste1– Validação do Nome   Utilizador insere – Name = Infetologia  O sistema notifica – Já existe uma especialidade com esse nome. |

|  |  |
| --- | --- |
| Nome do caso de uso | RF06, Editar Especialidade |
| Descrição do requisito | O objetivo deste caso de uso é editar uma especialidade |
| Tamanho | 8 |
| Pré-condição | O diretor tem o login válido. |
| Cenário Principal |  |
|  | 1-O diretor seleciona a opção editar na especialidade que pretende editar.  2-O sistema abre o formulário com os dados já preenchidos.  3-O diretor altera os atributos que pretende alterar.  4-O diretor carrega na opção guardar.  5-O sistema devolve uma mensagem de sucesso. |
| Cenário Alternativo | 2a-O sistema tem opção de voltar à lista  3a-Se o atributo nome estiver nulo o sistema pede para inserir um nome.  3b-Se já houver uma especialidade cujo nome seja igual ao atributo nome o sistema pede para inserir um nome diferente. |
| Requisitos não funcionais | * Testar se o sistema deixa registar a especialidade sem os campos obrigatórios estarem preenchidos. * Teste1– Validação do Nome   Utilizador insere – Name = Infetologia   * O sistema notifica – Já existe uma especialidade com esse nome. |

|  |  |
| --- | --- |
| Nome do caso de uso | RF07, Eliminar Especialidade |
| Descrição do requisito | O objetivo deste caso de uso é eliminar uma especialidade |
| Tamanho | 8 |
| Pré-condição | O diretor tem o login válido. |
| Cenário Principal |  |
|  | 1-O diretor seleciona eliminar na especialidade que pretende apagar.  2-O sistema abre uma janela com os detalhes da especialidade.  3-O diretor seleciona o botão delete para eliminar a especialidade.  4-O sistema pede para confirmar que pretende eliminar o bloco operatório.  5-O sistema devolve uma mensagem de sucesso. |
| Cenário Alternativo | 2a-O sistema tem um botão de voltar à lista.  3a- Se a especialidade estiver atribuída, a pelo menos um enfermeiro, não pode ser eliminada, o sistema devolve uma mensagem de erro. |
| Requisitos não funcionais |  |

|  |  |
| --- | --- |
| Nome do caso de uso | RF08, Criar Turno |
| Descrição do requisito | O objetivo deste caso de uso é criar um novo turno |
| Tamanho | 8 |
| Pré-condição | O diretor tem o login válido. |
| Cenário Principal |  |
|  | 1-O diretor seleciona a opção criar novo turno.  2-O sistema abre o formulário com os dados a introduzir  3-O diretor insere o atributo nome, hora de início e duração.  4-O diretor carrega na opção criar.  5-O sistema atribui automaticamente o ID ao novo turno e devolve uma mensagem de sucesso. |
| Cenário Alternativo | 2a-O sistema tem opção de voltar à lista  3a-Se o atributo nome estiver nulo o sistema pede para inserir um nome.  3b-Se já houver um turno cujo nome seja igual ao atributo nome o sistema pede para inserir um nome diferente.  3c- Se o atributo hora de início estiver nulo o sistema pede para inserir uma hora de início.  3d- Se o atributo hora de início tiver sido introduzido num formato incorreto o sistema pede para inserir no formato 00:00.  3e- Se o atributo duração estiver nulo o sistema pede para inserir uma duração.  3f- Se o atributo duração tiver sido introduzido num formato incorreto o sistema pede para inserir no formato 00:00. |
| Requisitos não funcionais | * Testar se o sistema deixa registar o turno sem os campos obrigatórios estarem preenchidos. * Teste1– Validação do Nome   Utilizador insere – Name = Manhã  O sistema notifica – Já existe um turno com esse nome.   * Teste2- Validação da hora de início   Utilizador insere – Starting Hour = 989:00  O sistema notifica – Por favor insira a hora em que o turno começa no formato 00:00.   * Teste3- Validação da duração   Utilizador insere – Duration = 8  O sistema notifica – Por favor insira duração do turno no formato 00:00. |

|  |  |
| --- | --- |
| Nome do caso de uso | RF09, Editar Turno |
| Descrição do requisito | O objetivo deste caso de uso é editar um turno |
| Tamanho | 8 |
| Pré-condição | O diretor tem o login válido. |
| Cenário Principal |  |
|  | 1-O diretor seleciona a opção editar no turno que pretende editar.  2-O sistema abre o formulário com os dados já preenchidos.  3-O diretor altera os atributos que pretende alterar.  4-O diretor carrega na opção guardar.  5-O sistema devolve uma mensagem de sucesso. |
| Cenário Alternativo | 2a-O sistema tem opção de voltar à lista  3a-Se o atributo nome estiver nulo o sistema pede para inserir um nome.  3b-Se já houver um turno cujo nome seja igual ao atributo nome o sistema pede para inserir um nome diferente.  3c- Se o atributo hora de início estiver nulo o sistema pede para inserir uma hora de início.  3d- Se o atributo hora de início tiver sido introduzido num formato incorreto o sistema pede para inserir no formato 00:00.  3e- Se o atributo duração estiver nulo o sistema pede para inserir uma duração.  3f- Se o atributo duração tiver sido introduzido num formato incorreto o sistema pede para inserir no formato 00:00. |
| Requisitos não funcionais | * Testar se o sistema deixa registar o turno sem os campos obrigatórios estarem preenchidos. * Teste1– Validação do Nome   Utilizador insere – Name = Manhã  O sistema notifica – Já existe um turno com esse nome.   * Teste2- Validação da hora de início   Utilizador insere – Starting Hour = 989:00  O sistema notifica – Por favor insira a hora em que o turno começa no formato 00:00.   * Teste3- Validação da duração   Utilizador insere – Duration = 8  O sistema notifica – Por favor insira duração do turno no formato 00:00. |

|  |  |
| --- | --- |
| Nome do caso de uso | RF10, Eliminar Turno |
| Descrição do requisito | O objetivo deste caso de uso é eliminar um turno |
| Tamanho | 8 |
| Pré-condição | O diretor tem o login válido. |
| Cenário Principal |  |
|  | 1-O diretor seleciona eliminar no turno que pretende apagar.  2-O sistema abre uma janela com os detalhes do turno.  3-O diretor seleciona o botão delete para eliminar o turno.  4-O sistema pede para confirmar que pretende eliminar o turno.  5-O sistema devolve uma mensagem de sucesso. |
| Cenário Alternativo | 2a-O sistema tem um botão de voltar à lista.  3a- Se o turno estiver atribuído a pelo menos uma ligação entre um bloco operatório e um turno, o turno não pode ser eliminado, o sistema devolve uma mensagem de erro. |
| Requisitos não funcionais |  |

|  |  |
| --- | --- |
| Nome do caso de uso | RF11, Criar Bloco Operatório |
| Descrição do requisito | O objetivo deste caso de uso é criar um novo bloco operatório |
| Tamanho | 8 |
| Pré-condição | O diretor tem o login válido. |
| Cenário Principal |  |
|  | 1-O diretor seleciona a opção criar um novo bloco operatório.  2-O sistema abre o formulário com os dados a introduzir.  3-O diretor insere o atributo nome.  4-O diretor carrega na opção criar.  5-O sistema atribui automaticamente o ID ao novo bloco operatório.  6-O sistema devolve uma mensagem de sucesso. |
| Cenário Alternativo | 2a-O sistema tem opção de voltar à lista  3a-Se o atributo nome estiver nulo o sistema pede para inserir um nome.  3b-Se já houver um bloco operatório cujo nome seja igual ao atributo nome o sistema pede para inserir um nome diferente. |
| Requisitos não funcionais | * Testar se o sistema deixa registar o turno sem os campos obrigatórios estarem preenchidos. * Teste1– Validação do Nome   Utilizador insere – Name = Bloco A  O sistema notifica – Já existe um turno com esse nome. |

|  |  |
| --- | --- |
| Nome do caso de uso | RF12, Editar Bloco Operatório |
| Descrição do requisito | O objetivo deste caso de uso é editar um bloco operatório |
| Tamanho | 8 |
| Pré-condição | O diretor tem o login válido. |
| Cenário Principal |  |
|  | 1-O diretor seleciona a opção editar no bloco operatório que pretende editar.  2-O sistema abre o formulário com os dados já preenchidos.  3-O diretor altera os atributos que pretende alterar.  4-O diretor carrega na opção guardar.  5-O sistema devolve uma mensagem de sucesso. |
| Cenário Alternativo | 2a-O sistema tem opção de voltar à lista  3a-Se o atributo nome estiver nulo o sistema pede para inserir um nome.  3b-Se já houver um bloco operatório cujo nome seja igual ao atributo nome o sistema pede para inserir um nome diferente. |
| Requisitos não funcionais | * Testar se o sistema deixa registar o turno sem os campos obrigatórios estarem preenchidos. * Teste1– Validação do Nome   Utilizador insere – Name = Bloco A   * O sistema notifica – Já existe um turno com esse nome. |

|  |  |
| --- | --- |
| Nome do caso de uso | RF13, Eliminar Bloco Operatório |
| Descrição do requisito | O objetivo deste caso de uso é eliminar um bloco operatório |
| Tamanho | 8 |
| Pré-condição | O diretor tem o login válido. |
| Cenário Principal |  |
|  | 1-O diretor seleciona eliminar no bloco operatório que pretende apagar.  2-O sistema abre uma janela com os detalhes do bloco operatório.  3-O diretor seleciona o botão delete para eliminar o bloco operatório.  4-O sistema pede para confirmar que pretende eliminar o bloco operatório.  5-O sistema devolve uma mensagem de sucesso. |
| Cenário Alternativo | 2a-O sistema tem um botão de voltar à lista.  3a-Se o bloco operatório estiver atribuído a pelo menos uma ligação entre um bloco operatório e um turno, o bloco operatório não pode ser eliminado, o sistema devolve uma mensagem de erro. |
| Requisitos não funcionais |  |

|  |  |
| --- | --- |
| Nome do caso de uso | RF14, Criar ligação entre Bloco Operatório e Turno |
| Descrição do requisito | O objetivo deste caso de uso é criar uma nova ligação entre bloco operatório e turno |
| Tamanho | 8 |
| Pré-condição | O diretor tem o login válido. |
| Cenário Principal |  |
|  | 1-O diretor seleciona a opção criar uma nova ligação entre bloco operatório e turno.  2-O sistema abre o formulário com todos os blocos operatórios e turnos.  3-O diretor escolhe um bloco operatório e um turno.  4-O diretor carrega na opção criar.  5-O sistema atribui automaticamente o ID à troca.  6-O sistema devolve uma mensagem de sucesso. |
| Cenário Alternativo | 2a-O sistema tem opção de voltar à lista.  3a-Se não existir nenhum bloco operatório para fazer ligação o sistema dá erro.  3b-Se não existir nenhum turno para fazer ligação o sistema dá erro.  3c-Se já houver uma ligação entre o bloco operatório escolhido e o turno escolhido o sistema pede para escolher um bloco operatório ou turno diferente. |
| Requisitos não funcionais | * Testar se o sistema deixa registar a ligação entre Bloco Operatório e Turno sem os campos obrigatórios estarem preenchidos. * Teste1– Validação da ligação   Utilizador seleciona – Operation Block = Bloco A  Utilizador seleciona – Shift = Manhã  O sistema notifica – Já existe uma ligação entre o bloco operatório escolhido e o turno escolhido. |

|  |  |
| --- | --- |
| Nome do caso de uso | RF15, Editar ligação entre Bloco Operatório e Turno |
| Descrição do requisito | O objetivo deste caso de uso é editar uma ligação entre bloco operatório e turno |
| Tamanho | 8 |
| Pré-condição | O diretor tem o login válido. |
| Cenário Principal |  |
|  | 1-O diretor seleciona a opção editar na ligação entre bloco operatório e turno que pretende editar.  2-O sistema abre o formulário com o bloco operatório e o turno já selecionados.  3-O diretor escolhe o bloco operatório e/ou o turno que pretende alterar.  4-O diretor carrega na opção guardar.  5-O sistema devolve uma mensagem de sucesso. |
| Cenário Alternativo | 2a-O sistema tem opção de voltar à lista  3a-Se já houver uma ligação entre o bloco operatório escolhido e o turno escolhido o sistema pede para escolher um bloco operatório ou turno diferente. |
| Requisitos não funcionais | * Testar se o sistema deixa registar a ligação entre Bloco Operatório e Turno sem os campos obrigatórios. * Teste1– Validação da ligação   Utilizador seleciona – Operation Block = Bloco A  Utilizador seleciona – Shift = Manhã   * O sistema notifica – Já existe uma ligação entre o bloco operatório escolhido e o turno escolhido. |

|  |  |
| --- | --- |
| Nome do caso de uso | RF16, Eliminar ligação entre Bloco Operatório e Turno |
| Descrição do requisito | O objetivo deste caso de uso é eliminar uma ligação entre bloco operatório e turno |
| Tamanho | 8 |
| Pré-condição | O diretor tem o login válido. |
| Cenário Principal |  |
|  | 1-O diretor seleciona eliminar na ligação entre bloco operatório e turno que pretende apagar.  2-O sistema abre uma janela com os detalhes da ligação entre bloco operatório e turno.  3-O diretor seleciona o botão delete para eliminar a ligação entre bloco operatório e turno.  4-O sistema pede para confirmar que pretende eliminar a ligação entre bloco operatório e turno.  5-O sistema devolve uma mensagem de sucesso. |
| Cenário Alternativo | 2a-O sistema tem um botão de voltar à lista.  3a- Se a ligação entre bloco operatório e turno estiver atribuído a pelo menos um horário não pode ser eliminado, o sistema devolve uma mensagem de erro. |
| Requisitos não funcionais |  |

|  |  |
| --- | --- |
| Nome do caso de uso | RF17, Gerar Horário |
| Descrição do requisito | O objetivo deste caso de uso é gerar um novo horário |
| Tamanho | 8 |
| Pré-condição | O diretor tem o login válido.  Tem de existir enfermeiros e pelo menos uma ligação entre bloco operatório e turno |
| Cenário Principal |  |
|  | 1-O diretor seleciona a opção criar horários.  2-O sistema abre a janela de criação de horários.  3-O diretor insere o atributo data de início, o atributo data de fim e seleciona o atributo bloco operatório.  4-O enfermeiro carrega na opção confirmar.  5-O sistema, segundo o algoritmo, atribui horários aos enfermeiros.  6-O sistema atribui automaticamente os IDs aos horários.  7-O sistema devolve uma mensagem de sucesso. |
| Cenário Alternativo | 2a-O sistema tem um botão de voltar à lista.  4a-Se algum atributo não estiver preenchido, o sistema dá erro. |
| Requisitos não funcionais | * Testar se o sistema deixa gerar horários sem os campos obrigatórios estarem preenchidos. * Teste1– Validação das datas   Utilizador insere – Initial Date = 1/10/2019  Utilizador insere – Final Date = 1/10/2018   * O sistema notifica – A data de fim não pode ser antes da data de inicio. |

|  |  |
| --- | --- |
| Nome do caso de uso | RF18, Visualizar Horários |
| Descrição do requisito | O objetivo deste caso de uso é visualizar horários dos enfermeiros por bloco operatório |
| Tamanho | 21 |
| Pré-condição | O ator tem o login válido. |
| Cenário Principal |  |
|  | 1-O ator seleciona a opção horários.  2-O sistema apresenta uma lista de horários. |
| Cenário Alternativo |  |
| Requisitos não funcionais |  |

|  |  |
| --- | --- |
| Nome do caso de uso | RF19, Criar Pedido de Troca |
| Descrição do requisito | O objetivo deste caso de uso é criar troca entre enfermeiros |
| Tamanho | 17 |
| Pré-condição | O enfermeiro ou enfermeiro chefe tem o login válido. |
| Cenário Principal |  |
|  | 1-O enfermeiro seleciona a opção horários.  2-O sistema apresenta uma lista de horários.  3-O enfermeiro escolhe o horário A no qual está inserido e o horário B que pretende trocar.  4-O enfermeiro seleciona a opção Realizar Pedido de Troca.  5-O sistema atribui automaticamente o ID à troca.  6-O sistema devolve uma mensagem de sucesso. |
| Cenário Alternativo | 2a- O sistema tem a opção de voltar ao menu anterior.  6b – O sistema tem uma opção de cancelar troca. |
| Requisitos não funcionais |  |

|  |  |
| --- | --- |
| Nome do caso de uso | RF20, Visualizar Trocas |
| Descrição do requisito | O objetivo deste caso de uso é, visualizar todas as trocas dos enfermeiros. |
| Tamanho | 5 |
|  |  |
| Pré-condição | O ator tem o login válido. |
| Cenário Principal |  |
|  | 1-O ator seleciona a opção trocas.  2-O sistema apresenta uma lista de trocas. |
| Cenário Alternativo |  |
| Requisitos não funcionais |  |

|  |  |
| --- | --- |
| Nome do caso de uso | RF21, Validar Pedido de Troca |
| Descrição do requisito | O objetivo deste caso de uso é validar todas as trocas dos enfermeiros. |
| Tamanho | 21 |
|  |  |
| Pré-condição | O enfermeiro chefe tem o login válido. |
| Cenário Principal |  |
|  | 1-O enfermeiro chefe seleciona a janela com a lista de trocas.  2-O sistema apresenta uma lista de pedidos de trocas.  3-O enfermeiro chefe seleciona as trocas que pretende aprovar ou não aprovar.  4-O sistema dá uma mensagem da troca validada e reencaminha o ator para os horários. |
| Cenário Alternativo |  |
| Requisitos não funcionais |  |

|  |  |
| --- | --- |
| Nome do caso de uso | RF22, Criar Regras e Restrições |
| Descrição do requisito | O objetivo deste caso de uso é criar novas regras e restrições para o horário. |
| Tamanho | 21 |
| Pré-condição | O diretor tem o login válido. |
| Cenário Principal |  |
|  | 1-O diretor seleciona a opção criar novas regras.  2-O sistema abre o formulário com os dados a introduzir.  3-O diretor insere os atributos horas semanais, idade do enfermeiro, idade do filho mais novo, tempo entre turnos.  4-O diretor carrega na opção confirmar.  5-O sistema atribui automaticamente o ID às Regras.  6-O sistema devolve uma mensagem de sucesso. |
| Cenário Alternativo | 2a-O sistema tem opção de voltar à lista.  3a- Se o atributo horas semanais estiver nulo o sistema pede para inserir as horas semanais.  3b- Se o atributo horas semanais tiver sido introduzido de forma incorreta o sistema pede para inserir um número.  3c- Se o atributo idade do enfermeiro estiver nulo o sistema pede para inserir uma idade do enfermeiro.  3d- Se o atributo idade do enfermeiro tiver sido introduzido de forma incorreta o sistema pede para inserir um número.  3e- Se o atributo idade do filho mais novo estiver nulo o sistema pede para inserir uma idade do filho mais novo.  3f- Se o atributo idade do filho mais novo tiver sido introduzido de forma incorreta o sistema pede para inserir um número.  3g- Se o atributo tempo entre turnos estiver nulo o sistema pede para inserir um tempo entre turnos.  3h- Se o atributo tempo entre turnos tiver sido introduzido num formato incorreto o sistema pede para inserir no formato 00:00. |
| Requisitos não funcionais | * Testar se o sistema deixa registar o turno sem os campos obrigatórios estarem preenchidos. * Teste1– Validação das Horas Semanais   Utilizador insere – Weekly Hours = e  O sistema notifica – Por favor insira um número.   * Teste2– Validação da Idade do Enfermeiro   Utilizador insere – Nurse's Age = e  O sistema notifica – Por favor insira um número.   * Teste3– Validação da Idade do Filho Mais Novo   Utilizador insere – Nurse's Child's Age = e  O sistema notifica – Por favor insira um número.   * Teste4- Validação do Tempo Entre Turnos   Utilizador insere – Time Between Shifts = 8   * O sistema notifica – Por favor insira duração do turno no formato 00:00. |

|  |  |
| --- | --- |
| Nome do caso de uso | RF23, Editar Regras e Restrições |
| Descrição do requisito | O objetivo deste caso de uso é editar regras e restrições |
| Tamanho | 8 |
| Pré-condição | O enfermeiro chefe tem de ter o login válido. |
| Cenário Principal |  |
|  | 1-O diretor seleciona a opção criar novas regras.  2-O sistema abre o formulário com os dados já preenchidos.  3- O diretor altera os atributos que pretende alterar.  4-O enfermeiro carrega na opção confirmar.  5-O sistema devolve uma mensagem de sucesso. |
| Cenário Alternativo | 2a-O sistema tem opção de voltar à lista.  3a- Se o atributo horas semanais estiver nulo o sistema pede para inserir as horas semanais.  3b- Se o atributo horas semanais tiver sido introduzido de forma incorreta o sistema pede para inserir um número.  3c- Se o atributo idade do enfermeiro estiver nulo o sistema pede para inserir uma idade do enfermeiro.  3d- Se o atributo idade do enfermeiro tiver sido introduzido de forma incorreta o sistema pede para inserir um número.  3e- Se o atributo idade do filho mais novo estiver nulo o sistema pede para inserir uma idade do filho mais novo.  3f- Se o atributo idade do filho mais novo tiver sido introduzido de forma incorreta o sistema pede para inserir um número.  3g- Se o atributo tempo entre turnos estiver nulo o sistema pede para inserir um tempo entre turnos.  3h- Se o atributo tempo entre turnos tiver sido introduzido num formato incorreto o sistema pede para inserir no formato 00:00. |
| Requisitos não funcionais | * Testar se o sistema deixa registar o turno sem os campos obrigatórios estarem preenchidos. * Teste1– Validação das Horas Semanais   Utilizador insere – Weekly Hours = e  O sistema notifica – Por favor insira um número.   * Teste2– Validação da Idade do Enfermeiro   Utilizador insere – Nurse's Age = e  O sistema notifica – Por favor insira um número.   * Teste3– Validação da Idade do Filho Mais Novo   Utilizador insere – Nurse's Child's Age = e  O sistema notifica – Por favor insira um número.   * Teste4- Validação do Tempo Entre Turnos   Utilizador insere – Time Between Shifts = 8  O sistema notifica – Por favor insira duração do turno no formato 00:00. |

|  |  |
| --- | --- |
| Nome do caso de uso | RF24, Eliminar Regras e Restrições |
| Descrição do requisito | O objetivo deste caso de uso é eliminar regras e restrições. |
| Tamanho | 15 |
|  |  |
| Pré-condição | O diretor tem de ter o login válido.  Tem que existir regras e restrições no sistema. |
| Cenário Principal |  |
|  | 1-O diretor seleciona eliminar na regra que pretende apagar.  2-O sistema abre uma janela com os detalhes da regra que selecionou.  3-O diretor seleciona o botão delete para eliminar a regra.  4-O sistema pede para confirmar que pretende eliminar a regra.  5-O sistema devolve uma mensagem de sucesso. |
| Cenário Alternativo | 2a-O sistema tem um botão de voltar à lista. |
| Requisitos não funcionais |  |

Algoritmo para gerar horários

1.Obter o nome do bloco a escalonar.

2.Obter o nº de enfermeiros disponíveis.

3.Obter a duração do turno.

4.Obter as horas semanais de cada enfermeiro.

5.Obter o nº de horas entre turnos.

6.Obter a idade limite do filho mais novo.

7.Calcular o nº de turnos(pela divisão inteira de 24h com a duração de cada turno).

8.Fazer se:

8.1.Bloco não estiver cheio e Turno for disponivel e Enfermeiros disponiveis:

8.1.Para cada enfermeiro, fazer:

8.1.1.Obter a idade e a idade do filho mais novo e o cargo//Pelo cálculo da data de nascimento do enfermeiro.

8.1.1.1.Se o enfermeiro tiver mais de x anos e ou se o enfermeiro tiver o filho menor que x anos e ou se o enfermeiro for o enfermeiro-chefe , então:

8.1.1.1.1.Não faz turnos de noite.

8.1.1.2.Se não:

8.1.1.2.1.Escalona o enfermeiro num turno disponível(por ordem da hora do turno).

5 Diagramas de Sequência dos Casos de Uso mais importantes

Diagrama de Sequência Login

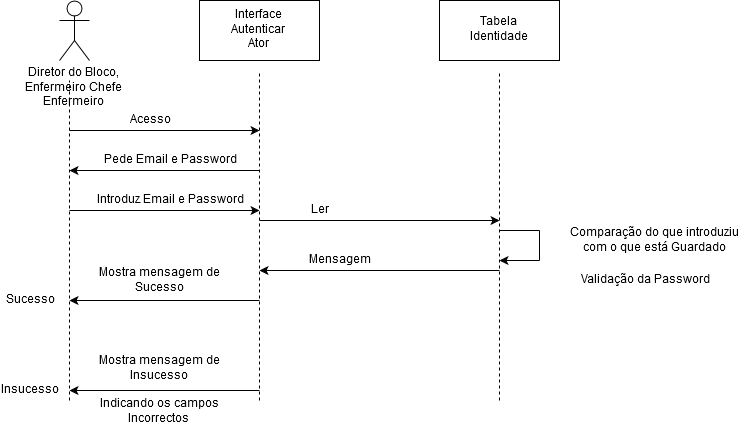


Diagrama de Sequência Criar Enfermeiro

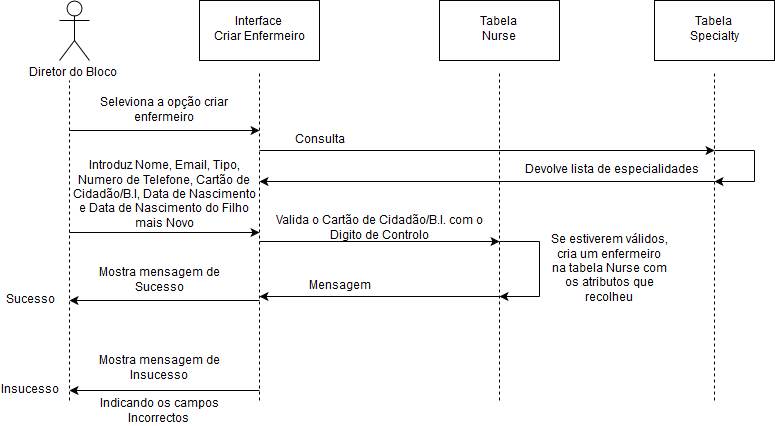


Diagrama de Sequência Criar Ligação Bloco Operatório - Turno

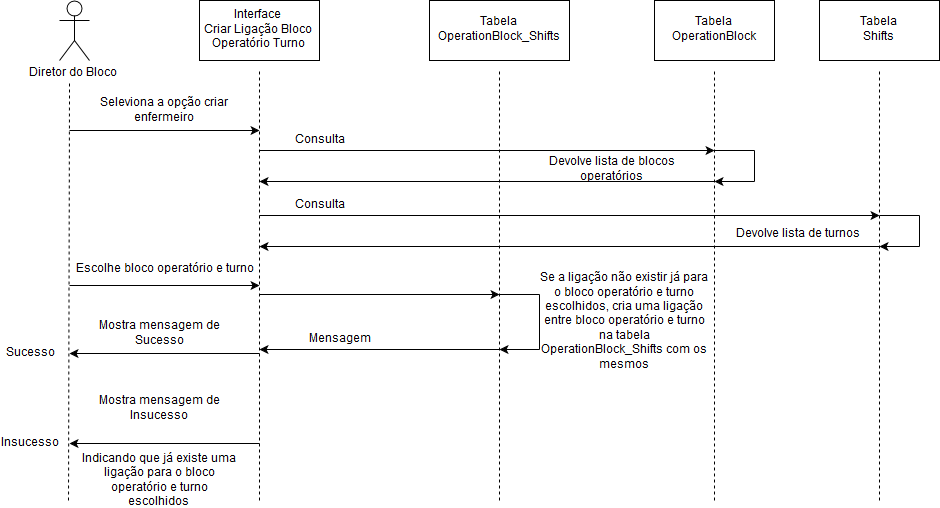


Diagrama de Sequência Gerar Horário

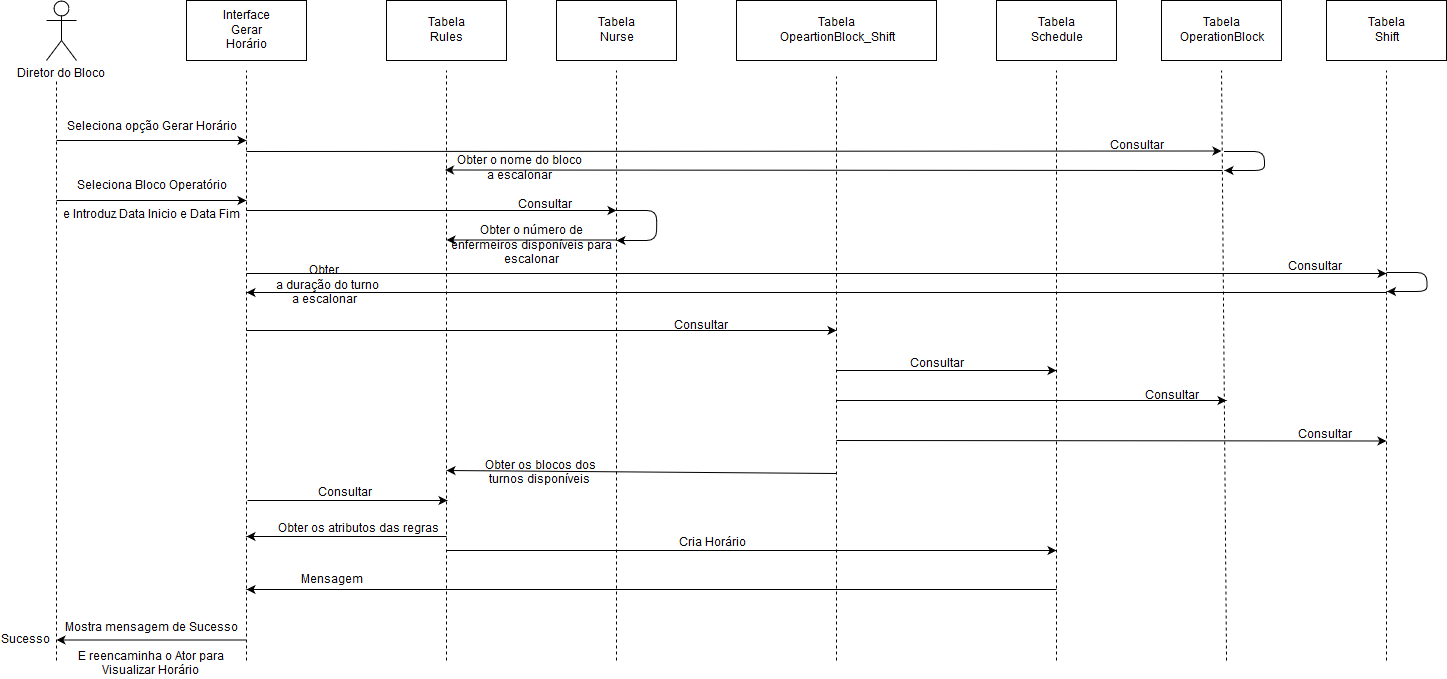


Diagrama de Sequência Pedir Troca de Turnos

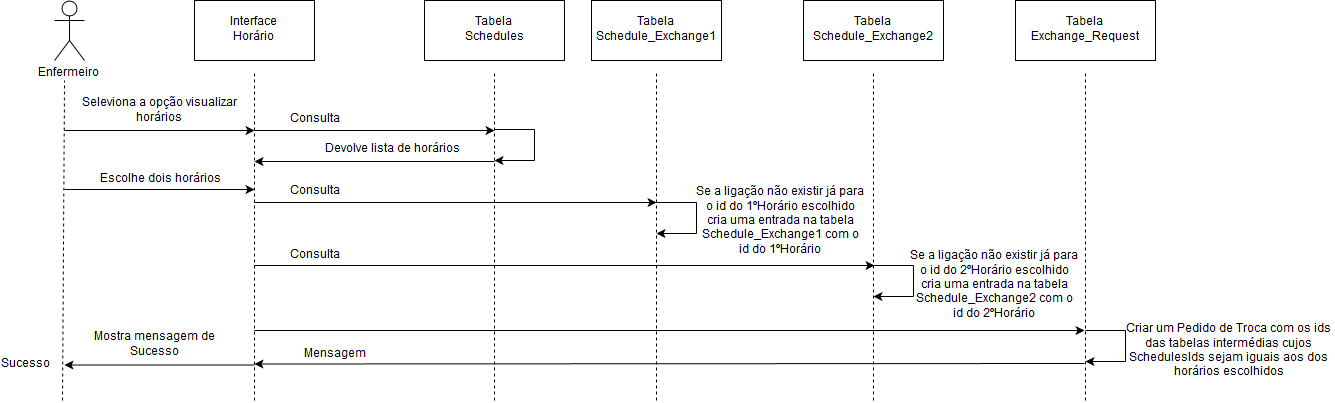


Diagrama de Classes

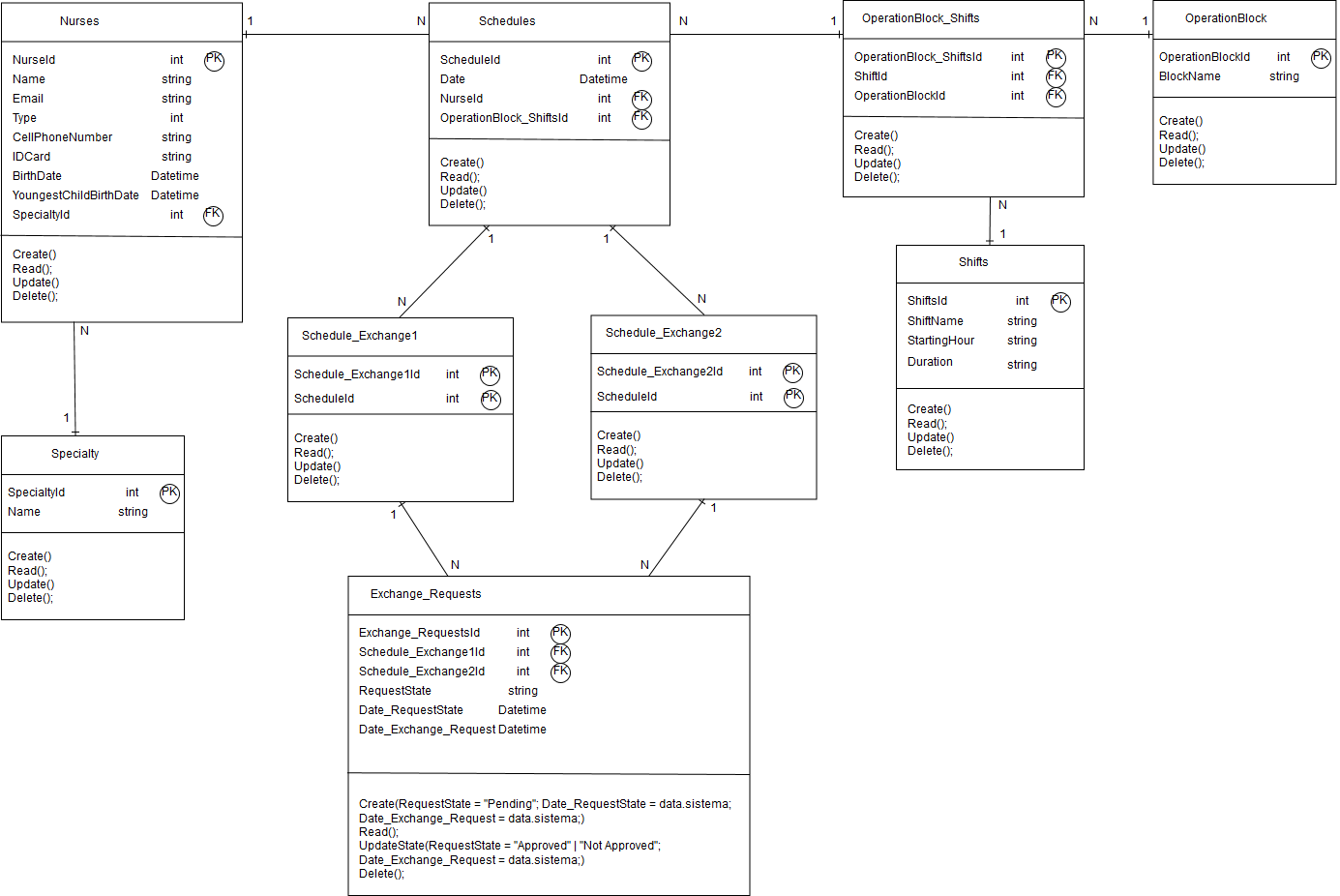
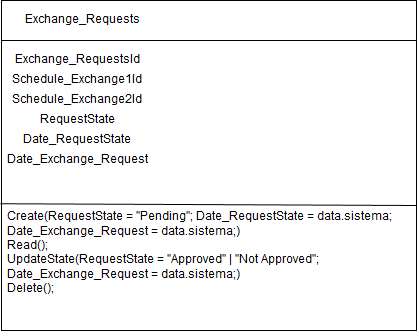
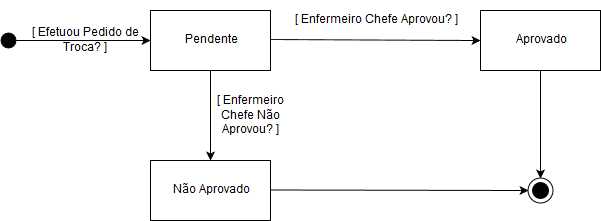


Diagrama de Estados





Semântica de Classes:

## Classe Nurse

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nurse | | | | | |
| Nome do campo | Tipo de Dados | Descrição | Valores Válidos | Formato | Restrições |
| NurseId | Inteiro | Número que identifica de forma única o enfermeiro | Números/Maior que zero | Ate 5 dígitos | Gerado pelo Sistema/Obrigatório/não alterável |
| Name | String | Identifica o nome do enfermeiro. | Caracteres | Até 20 caracteres | Obrigatório/Alterável |
| Email | Sring | Email do enfermeiro. | Caracteres | Até 256 caracteres | Obrigatório/Alterável |
| Type | Inteiro | Indica o tipo de enfermeiro (enfermeiro=0; enfermeiroChefe=1). Deixamos do tipo inteiro para permitir expansibilidade. | Números | Um dígito (0/1) | Obrigatório/Alterável |
| CellPhoneNumber | Inteiro | Indica o contacto do enfermeiro. | Caracteres | 9 caracteres | Obrigatório/Alterável |
| IDCard | String | Identificação do enfermeiro. (Cartão de Cidadão, Bilhete de Identidade) | Carateres | 12 caracteres | Obrigatório/Alterável |
| BirthDate | Date | Data de nascimento do enfermeiro. | Números | dd/MM/yyyy | Obrigatório/Alterável |
| YoungestChildBirthDate | Date | Data de nascimento do filho mais novo do enfermeiro. | Números | dd/MM/yyyy | Alterável |

Esta classe pertence ao diagrama de sequência Criar Enfermeiro

## Operações da classe Nurse

|  |  |
| --- | --- |
| Operações | |
| Nome | Descrição |
| Create() | Operação que permite criar um novo enfermeiro:   1. O sistema gera o NurseId 2. Introduzir o nome do Nurse 3. Introduzir o Email 4. Introduzir o Type 5. Introduzir o Cell Phone Number 6. Introduzir o ID Card 7. Introduzir o Birth Date 8. Introduzir a Youngest Child Birth Date 9. Introduzir Speciality 10. Se o atributo Name estiver nulo o sistema pede para inserir um nome. 11. Se o atributo Name estiver incorreto o sistema pede para inserir um nome válido 12. Se o atributo Email estiver nulo o sistema pede para inserir um email. 13. Se o atributo Email estiver incorreto o sistema pede para inserir um email válido. 14. Se o atributo Type estiver nulo o sistema pede para inserir um Type. 15. Se o atributo Type estiver incorreto o sistema pede para inserir um número. 16. Se o atributo Cell Phone Number estiver nulo o sistema pede para inserir Cell Phone Number. 17. Se o atributo Cell Phone Number estiver incorreto o sistema pede para inserir Cell Phone Number válido. 18. Se o atributo ID Card estiver nulo o sistema pede para inserir um ID Card. 19. Se o atributo ID Card não estiver correcto segundo o algoritmo de Validação do Digito de Controlo o sistema pede para inserir um ID Card válido. 20. Se o atributo Birth Date estiver nulo o sistema pede para inserir um Birth Date. 21. Se o atributo Birth Date estiver incorreta o sistema pede para inserir um Birth Date válido. 22. Se o atributo Youngest Child Birth Date estiver incorreta o sistema pede para inserir um Youngest Child Birth Date válido. 23. Criar novo Nurse |
| Update() | Operação que permite editar um enfermeiro:   1. Alterar o nome do Nurse 2. Alterar o Email 3. Alterar o Type 4. Alterar o Cell Phone Number 5. Alterar o ID Card 6. Alterar o Birth Date 7. Alterar a Youngest Child Birth Date 8. Alterar Speciality 9. Se o atributo Name estiver nulo o sistema pede para inserir um nome. 10. Se o atributo Name estiver incorreto o sistema pede para inserir um nome válido 11. Se o atributo Email estiver nulo o sistema pede para inserir um email. 12. Se o atributo Email estiver incorreto o sistema pede para inserir um email válido. 13. Se o atributo Type estiver nulo o sistema pede para inserir um Type. 14. Se o atributo Type estiver incorreto o sistema pede para inserir um número. 15. Se o atributo Cell Phone Number estiver nulo o sistema pede para inserir Cell Phone Number. 16. Se o atributo Cell Phone Number estiver incorreto o sistema pede para inserir Cell Phone Number válido. 17. Se o atributo ID Card estiver nulo o sistema pede para inserir um ID Card. 18. Se o atributo ID Card não estiver correcto segundo o algoritmo de Validação do Digito de Controlo o sistema pede para inserir um ID Card válido. 19. Se o atributo Birth Date estiver nulo o sistema pede para inserir um Birth Date. 20. Se o atributo Birth Date estiver incorreta o sistema pede para inserir um Birth Date válido. 21. Se o atributo Youngest Child Birth Date estiver incorreta o sistema pede para inserir um Youngest Child Birth Date válido. 22. Alterar Nurse |
| Delete() | Operação que permite eliminar um enfermeiro:   1. O NurseId selecionado corresponde ao NurseId do Nurse a eliminar. 2. Se existir pelo menos um horário onde o seu NurseId seja igual ao NurseId escolhido significa que este enfermeiro já tem uma ligação com a tabela Schedule, logo não pode ser eliminado, reencaminha o utilizador para a pagina de erro. 3. Elimina o Nurse |
| Read() | Operação que permite consultar os detalhes de um enfermeiro:   1. Permitir consultar todos os Nurses através de uma procura pelo nome. 2. O sistema devolve uma view com os detalhes do mesmo |

## Algoritmo de validação do Cartão de cidadão

O Cartão de Cidadão possui um número de documento, constituído por diversos elementos base, chamado de Digito de Controlo e pelo qual se pode validar se o cartão está correto ou não, de modo a evitar falha aquando da recolha do mesmo.

## Composição

O número de documento é constituído por 4 elementos distintos, sendo que se encontra visível na parte frontal do Cartão de Cidadão.

Formato: DDDDDDDD C AAT

D- Número de Identificação Civil [0.. 9]

C – Check Digit do Número de Identificação Civil [0.. 9]

A – Versão [A.. Z, 0.. 9]

T – Check Digit Número de Documento [0.. 9]

## Validação através do Digito de Controlo

A fórmula de validação efetua a verificação usando o número de documento completo. O número de documento tem de passar os seguintes testes:

1. Efetuando uma contagem da direita para a esquerda do número de documento, duplicar o valor de cada 2º elemento, sendo que as letras deverão ser substituídas de acordo com a tabela de conversão;

2. No caso de o resultado da duplicação ser igual ou superior a 10, subtrair 9 ao seu valor;

3. Somar a totalidade dos valores obtidos;

4. Ao valor obtido, deve-se calcular o resto da sua divisão por 10, sendo que se esse valor for 0 o número de documento é válido.

Exemplo CC= 00000000 0 ZZ4,

Iniciamos por retirar os espaços existentes. De seguida, deverão ser substituídas as letras pelo seu respetivo valor, sendo que para o caso concreto, ficaríamos com o valor:

[000000000(35)(35)4]

Passo 1 e 2: (0×2) = 0, (0×2) = 0, (0×2) = 0, (0×2) = 0, (0×2) = 0, ((35×2) = 70 – 9) = 61

Passo 3: (0) + 0 + (0) + 0 + (0) + 0 + (0) + 0 + (0) + 35 + (61) + 4 = 100

Passo 4: Se o resto da divisão inteira do resultado do passo 3, neste caso 100, por 10 for igual a 0 significa que o valor é válido.

## Classe OperationBlock\_Shifts

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| OperationBlock\_Shifts | | | | | |
| Nome do campo | Tipo de Dados | Descrição | Valores Válidos | Formato | Restrições |
| OperationBlock\_ShiftsId | Inteiro | Número que identifica de forma única a ligação entre o bloco operatório e o turno. | Números/Maior que zero | Ate 5 dígitos | Gerado pelo Sistema/Obrigatório/não alterável |
| OperationBlockId | String | Identifica o nome do enfermeiro. | Caracteres | Até 20 caracteres | Obrigatório/Alterável |
| ShiftId | Sring | Número que identifica de forma única o turno. | Caracteres | Até 256 caracteres | Obrigatório/Alterável |

Esta classe pertence ao diagrama de sequência Criar Ligação Bloco Operatório - Turno

## Operações da classe OperationBlock\_Shifts

|  |  |
| --- | --- |
| Operações | |
| Nome | Descrição |
| Create() | Operação que permite criar uma ligação entre o bloco operatório e o turno:   1. O sistema gera o OperationBlock\_ShiftsId 2. Escolher o OperationBlock 3. Escolher o Shift 4. Se já existe algum OperationBlock\_ShiftsId com o OperationBlock e o Shift escolhidos então o sistema manda ua mensagem de erro. 5. Criar novo OperationBlock\_Shifts. |
| Update() | Operação que permite editar uma ligação entre o bloco operatório e o turno:   1. Escolher o OperationBlock 2. Escolher o Shift 3. Se já existe algum OperationBlock\_ShiftsId com o OperationBlock e o Shift escolhidos então o sistema manda ua mensagem de erro. 4. Alterar OperationBlock\_Shifts. |
| Delete() | Operação que permite eliminar uma ligação entre o bloco operatório e o turno:   1. O OperationBlock\_ShiftsId selecionado corresponde ao OperationBlock\_ShiftsId do OperationBlock\_Shifts a eliminar. 2. Se existir pelo menos um Schedule onde o seu OperationBlock\_ShiftsId seja igual ao OperationBlock\_ShiftsId escolhido significa que este OperationBlock\_Shifts já tem uma ligação com a tabela Schedule, logo não pode ser eliminado, reencaminha o utilizador para a página de erro. 3. Elimina o OperationBlock\_ShiftsId. |
| Read() | Operação que permite consultar os detalhes de uma ligação entre o bloco operatório e o turno:   1. Permitir consultar todos os OperationBlock\_Shifts através de uma procura pelo nome do OperationBlock. 2. O sistema devolve uma view com os detalhes do mesmo |

## Classe Schedule

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Schedule | | | | | |
| Nome do campo | Tipo de Dados | Descrição | Valores Válidos | Formato | Restrições |
| ScheduleId | Inteiro | Número que identifica de forma única o horário. | Números/Maior que zero | Ate 5 dígitos | Gerado pelo Sistema/Obrigatório/não alterável |
| Date | Date | Identifica a data em que o horário foi criado. | Números | DD/MM/AAAA | Obrigatório/Não alterável |
| NurseId | Inteiro | Número que identifica de forma única o enfermeiro. | Números/Maior que zero | Ate 5 dígitos | Gerado pelo Sistema/Obrigatório/não alterável |
| OperationBlock\_ShiftsId | Inteiro | Chave Primária. | Números/Maior que zero | Ate 5 dígitos | Gerado pelo Sistema/Obrigatório/não alterável |

Esta classe pertence ao diagrama de sequência Criar Horário

## Operações da classe Schedule

|  |  |
| --- | --- |
| Operações | |
| Nome | Descrição |
| Create() | Operação que permite criar um novo Schedule:   1. O sistema gera o ScheduleId 2. Introduzir a data de inicio 3. Introduzir a data de fim 4. Introduzir o OperationBlock 5. Obter o nome do bloco a escalonar. 6. Obter o nº de enfermeiros disponíveis. 7. Obter a duração do turno. 8. Obter as horas semanais de cada enfermeiro. 9. Obter o nº de horas entre turnos. 10. Obter a idade limite do filho mais novo. 11. Calcular o nº de turnos(pela divisão inteira de 24h com a duração de cada turno). 12. Fazer se: 13. Bloco não estiver cheio e Turno for disponivel e Enfermeiros disponiveis: 14. Para cada enfermeiro, fazer: 15. Obter a idade e a idade do filho mais novo e o cargo//Pelo cálculo da data de nascimento do enfermeiro. 16. Se o enfermeiro tiver mais de x anos e ou se o enfermeiro tiver o filho menor que x anos e ou se o enfermeiro for o enfermeiro-chefe , então: 17. Não faz turnos de noite. 18. Se não: 19. Escalona o enfermeiro num turno disponível(por ordem da hora do turno). 20. Criar novo Schedule |
| Update() | Operação que permite efetuar o pedido de troca: (Quando o enfermeiro chefe aprova o Exchange\_Request)   1. O sistema busca os ids dos Schedules que estão referenciados nas tabelas intermédias indicadas pelo Exchange\_Request. 2. Com os ids dos Schedules buscar os ids dos Nurses e troca-os, efectuando assim o pedido de troca. |
| Read() | Operação que permite consultar os detalhes de um Schedule:   1. Permitir consultar todos os Schedules através de uma procura pelo nome do Nurse. 2. O sistema devolve uma view com os detalhes do mesmo. |

## Classe Exchange\_Request

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Exchange\_Request | | | | | |
| Nome do campo | Tipo de Dados | Descrição | Valores Válidos | Formato | Restrições |
| Exchange\_RequestId | Inteiro | Número que identifica de forma única o pedido de troca. | Números/Maior que zero | Até 5 dígitos | Gerado pelo Sistema/Obrigatório/Não Alterável |
| Schedule\_Exchange1Id | Inteiro | Número que identifica de forma única a tabela intermédia1. | Números/Maior que zero | Até 5 dígitos | Gerado pelo Sistema/Obrigatório/Não Alterável |
| Schedule\_Exchange2Id | Inteiro | Número que identifica de forma única a tabela intermédia2. | Números/Maior que zero | Até 5 dígitos | Gerado pelo Sistema/Obrigatório/Não Alterável |
| RequestState | String | Indica o estado do pedido, pode ter os valores de “Pending”, “Not Approved” ou ”Approved”. | “Pending”, “Not Approved” ou ”Approved” | Um dígito (0/1) | Obrigatório/Alterável |
| Date\_RequestState | DateTime | Indica a data e hora em que foi criado o pedido de troca. | De 1 de Janeiro 1 de 1753 00:00 até 31 de Dezembro de 9999 | dd/mm/aaaa hh:mm | Gerado pelo Sistema/Obrigatório/Não Alterável |
| Date\_Exchange\_Request | DateTime | Indica a data e hora em que o pedido de troca foi alterado. | Carateres | 12 caracteres | Gerado pelo Sistema/Obrigatório/Alterável |

Esta classe pertence ao diagrama de sequência Pedir de Troca de Turno

## Operações da classe Exchange\_Request

|  |  |
| --- | --- |
| Operações | |
| Nome | Descrição |
| Create(RequestState = “Pending”, Date\_RequestState = data.sistema, Date\_Exchange\_Request = data.sistema) | Operação que permite criar um pedido de troca:   1. Escolher o 1ºSchedule 2. Escolher o 2ºSchedule 3. Se nas tabelas intermédias(Schedule\_Exchange1 e 2) ainda não existir nenhuma entrada cujo ScheduleId seja igual ao ScheduleId dos Schedules escolhidos, cria entradas nessas tabelas 4. Buscar os ids das tabelas intermédias cujo ScheduleId seja igual ao ScheduleId dos Schedules escolhidos. 5. Com os ids das tabelas intermédias criar o Exchange\_Request com RequestState = “Pending”, Date\_RequestState = data.sistema, Date\_Exchange\_Request = data.sistema, o id do Exchange\_Request é gerado automaticamente. |
| UpdateState() | Operação que permite atualizar o estado do Exchange\_Request:   1. Alterar o Request\_State para “Approved” ou para “Not Approved” 2. Altera o Exchange\_Request 3. Se o Request\_State for alterado para “Approved” executa o Update() da classe Schedule. |
| Read() | Operação que permite consultar os detalhes de uma ligação entre o bloco operatório e o turno:   1. Permitir consultar todos os Exchange\_Request através de uma procura pelo nome do OperationBlock. 2. O sistema devolve uma view com os detalhes do mesmo |

Diagrama de Atividades

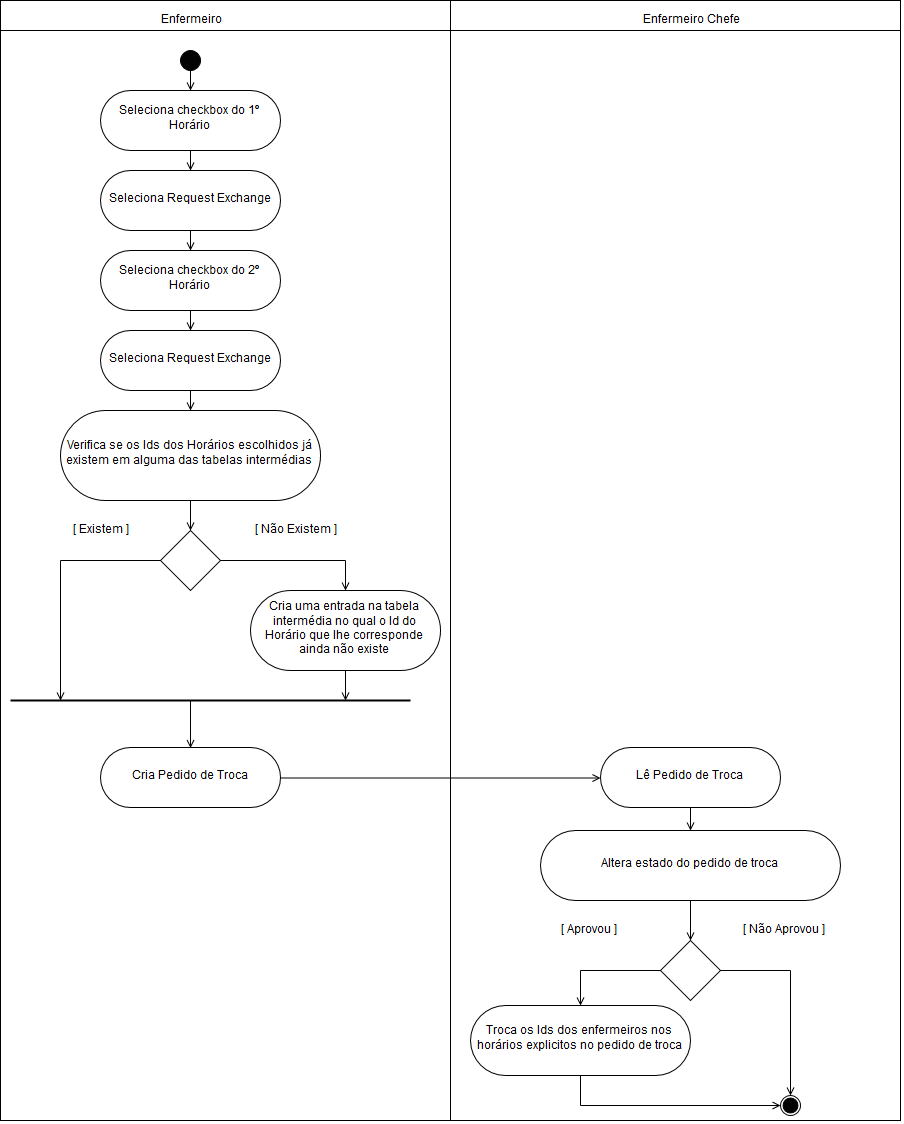


Diagrama de Pacotes de Componentes

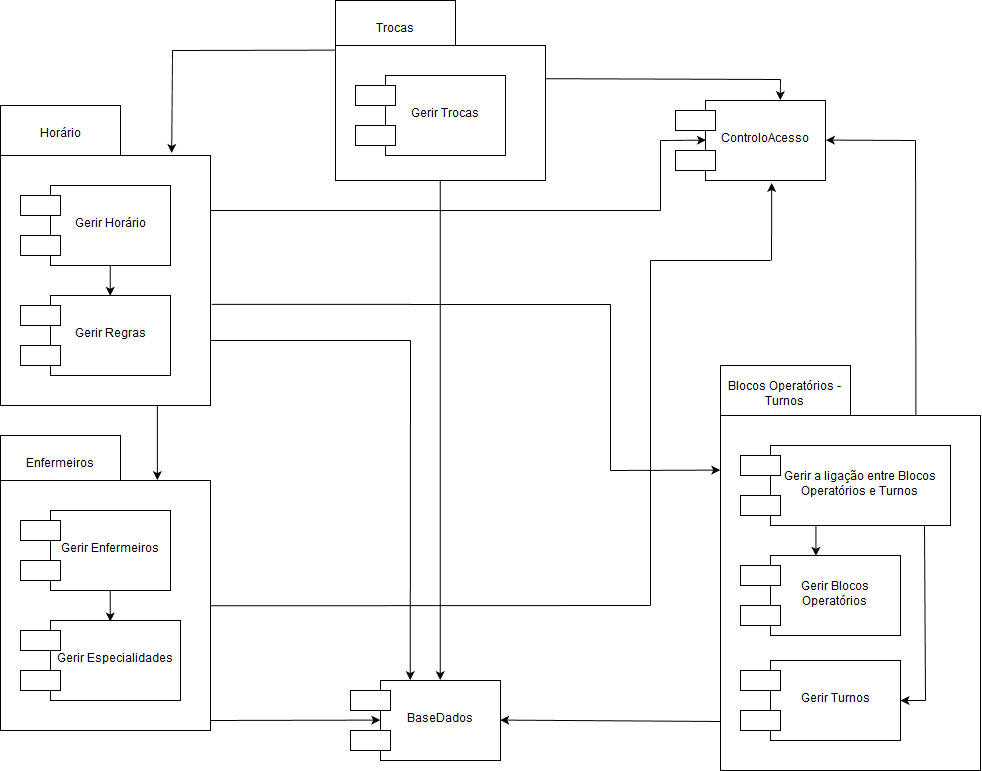
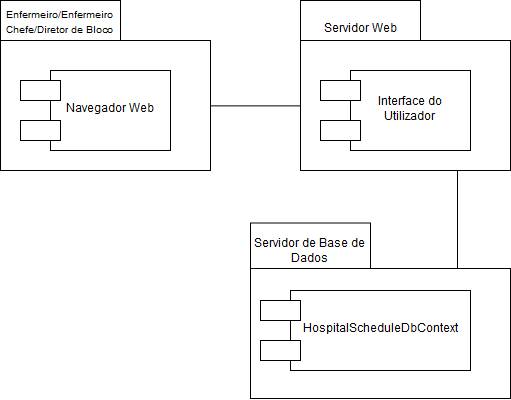


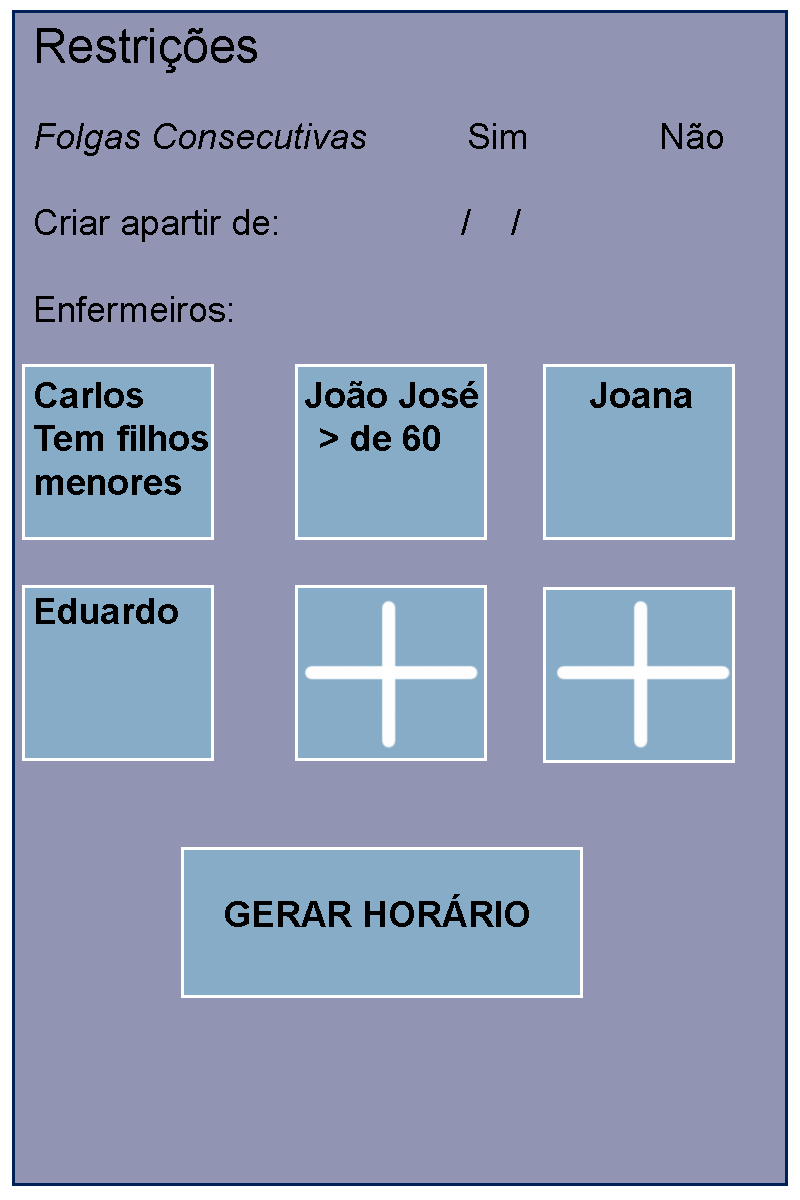
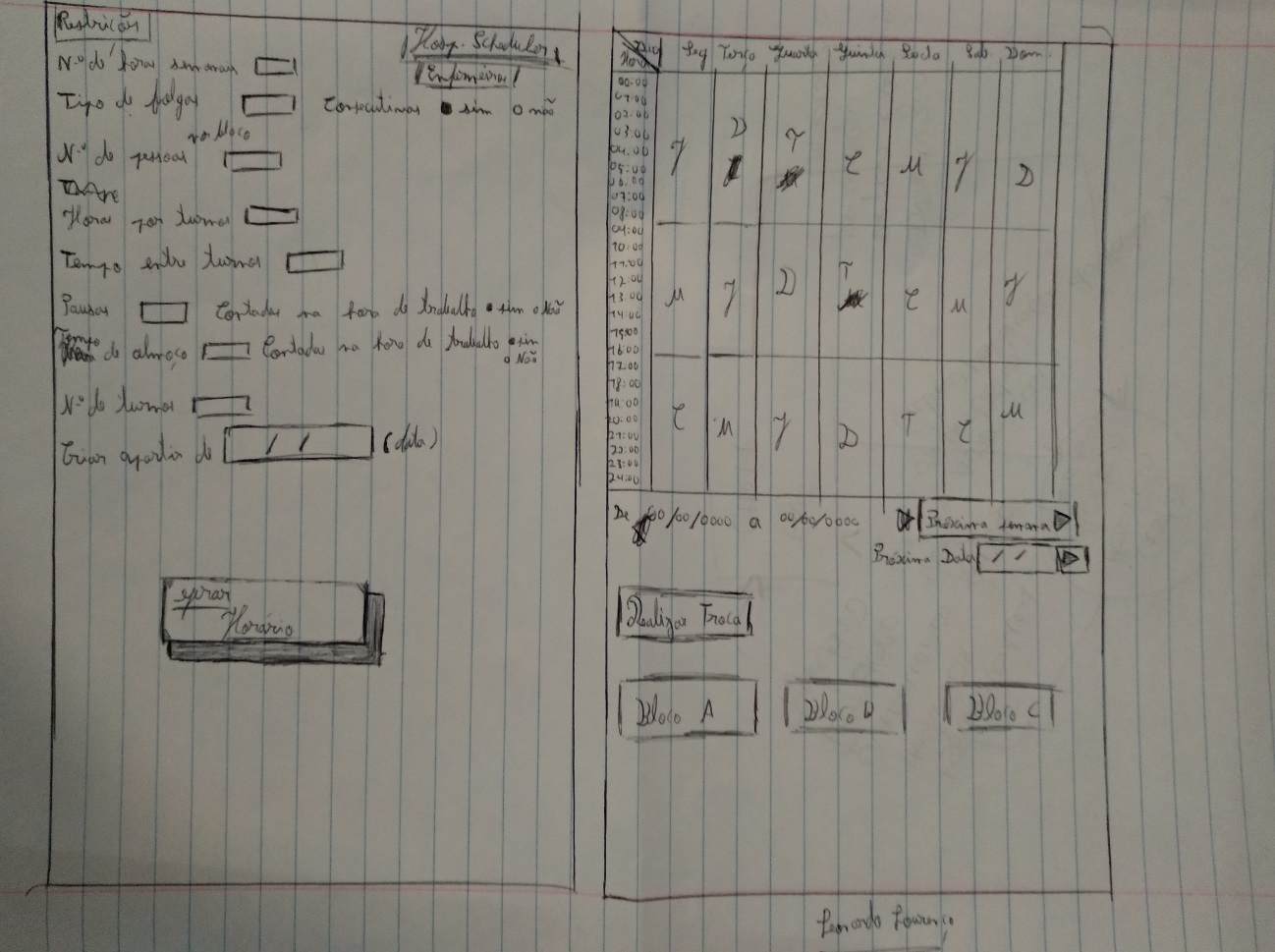
Diagrama de Instalação

Este diagrama descreve as vertentes de software e de hardware do sistema.

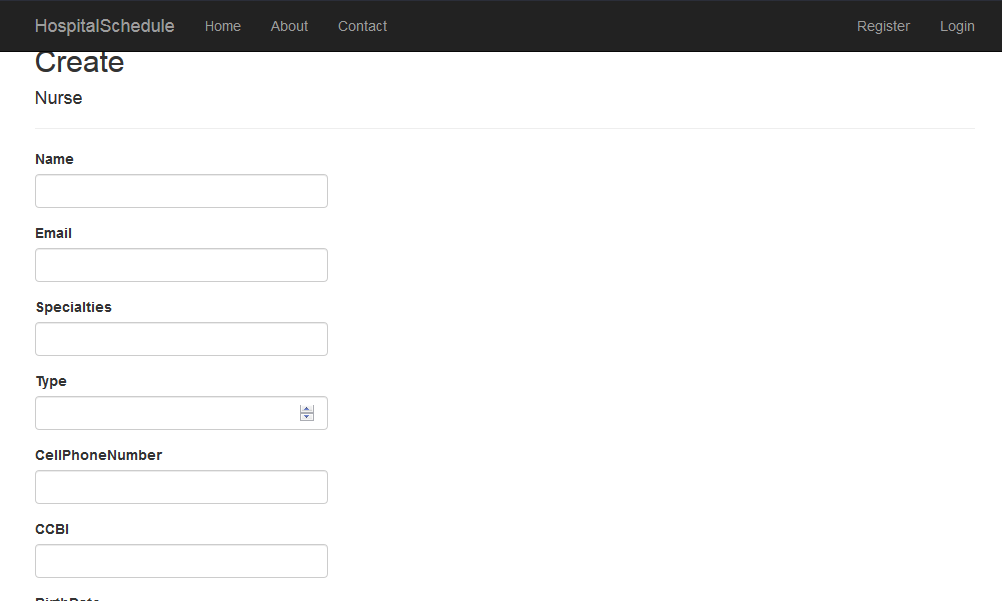


Na Interface do Utilizador estão situados todos os componentes do diagrama de componentes acima.

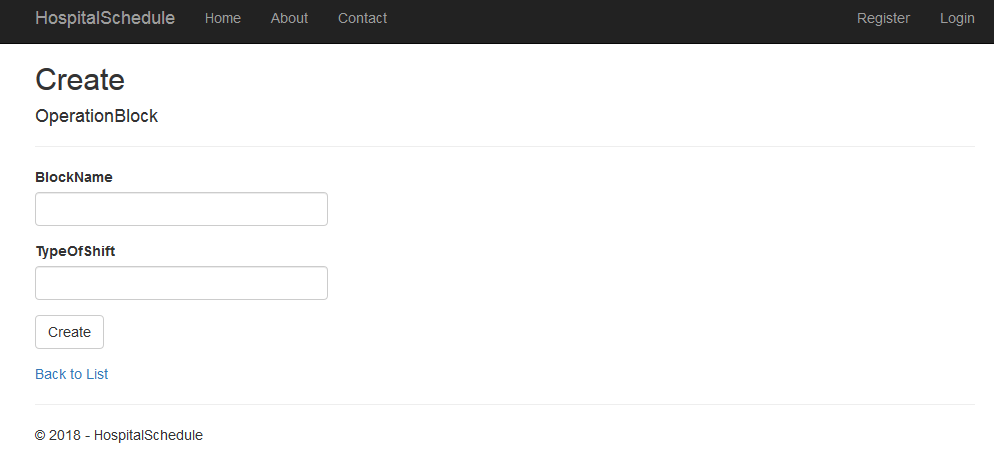
Na Base de Dados ( HospitalScheduleDbContext ) estão situadas as seguintes classes: Schedule, Nurse, Specialty, Operation Block, Shift, Operation Block Shift, Exchange Request, Schedule Exchange1, Schedule Exchange2.

Protótipos

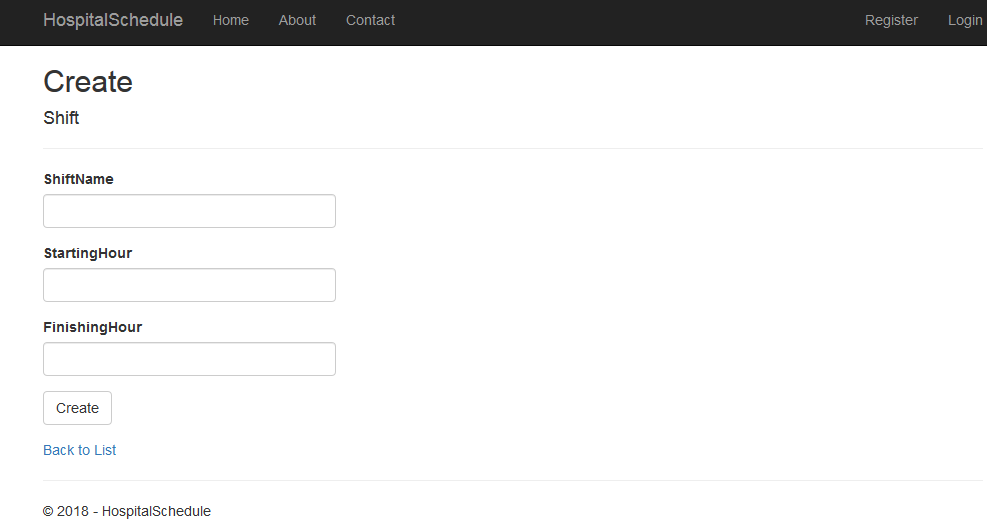
Protótipo Inserir enfermeiros



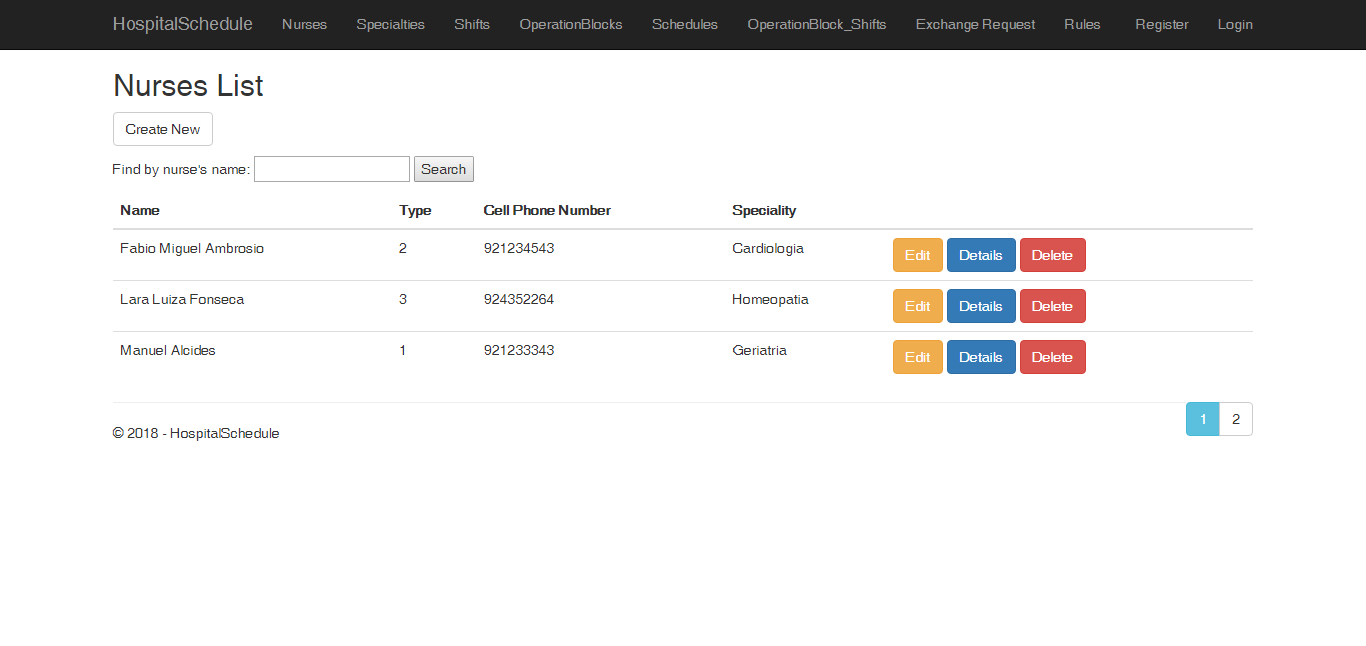
Protótipo Inserir Bloco Operatório

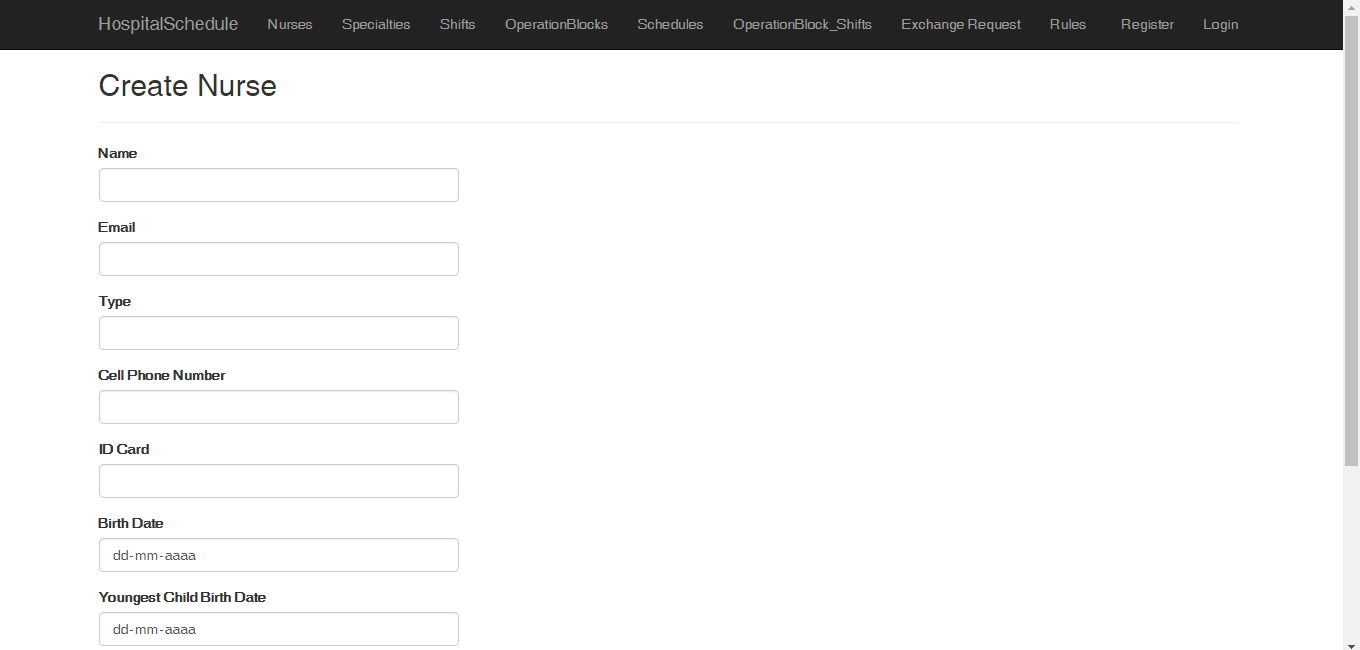


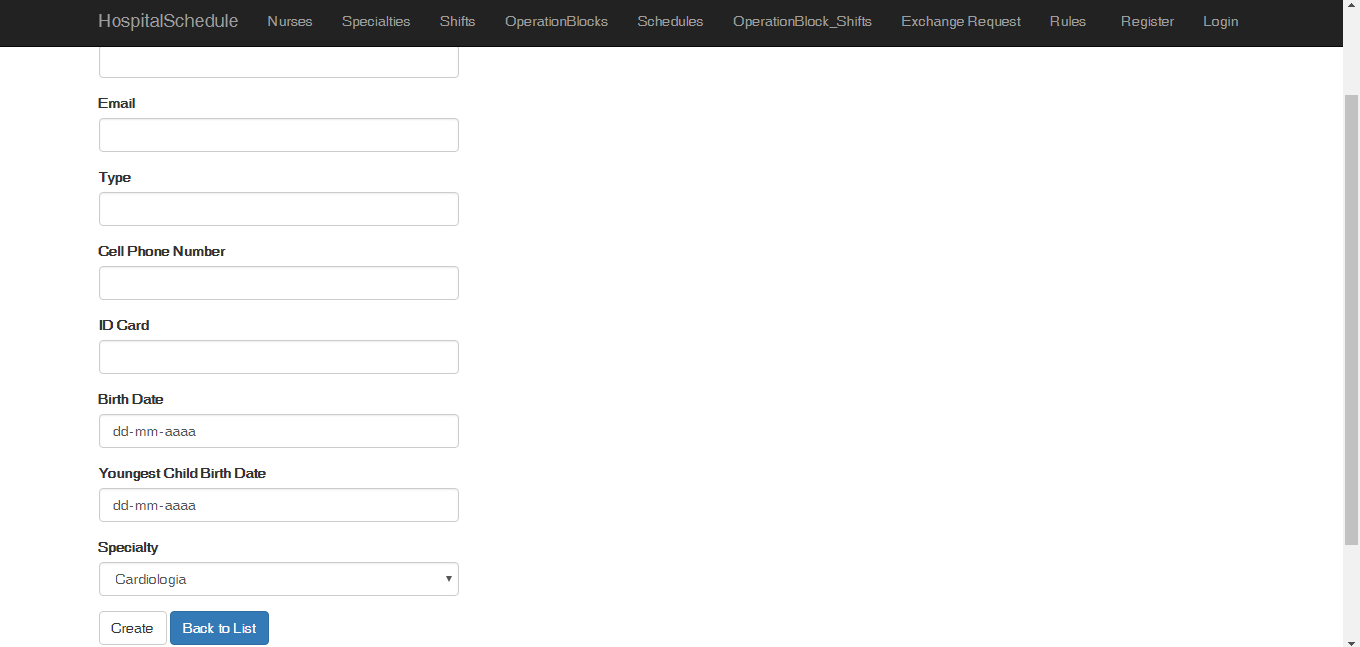
Protótipo Inserir Turno

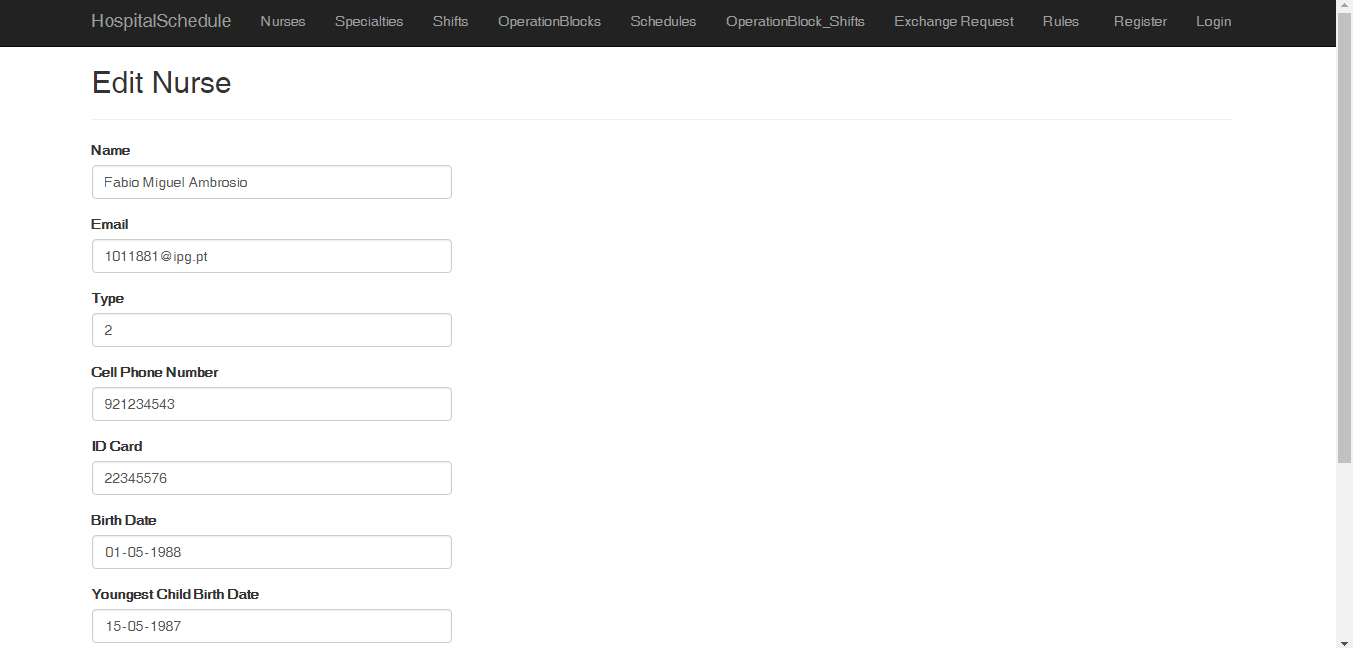
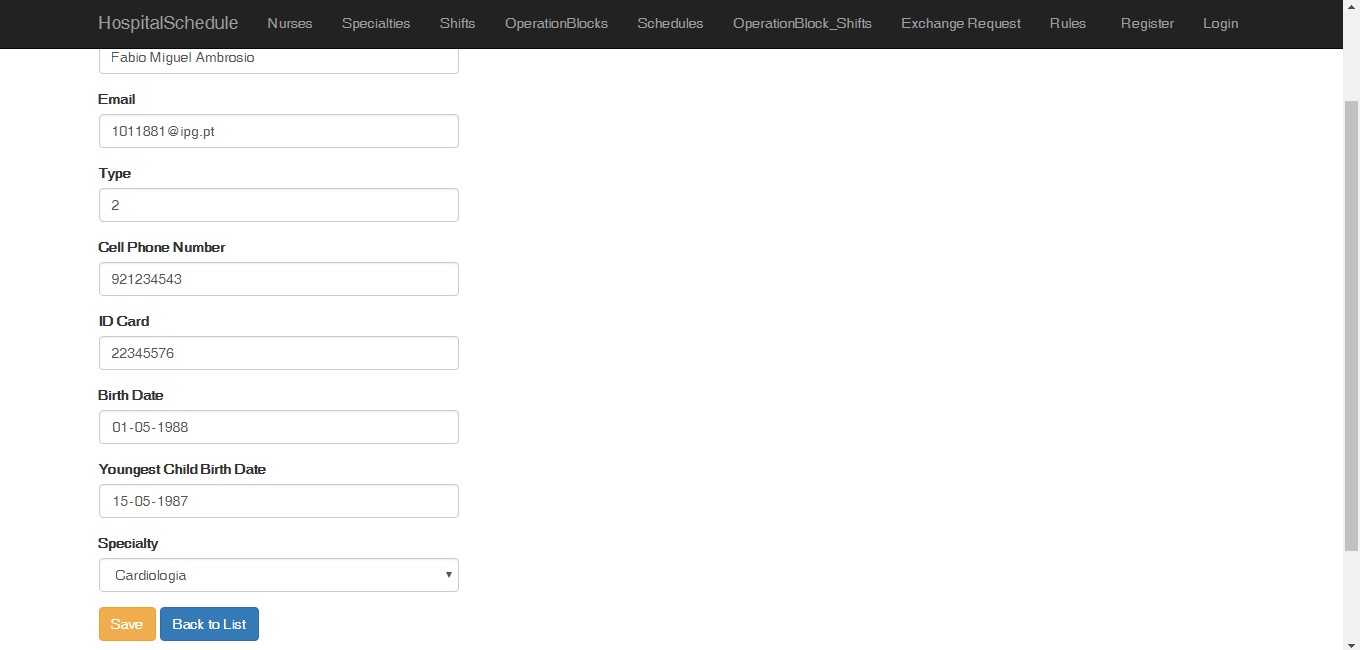


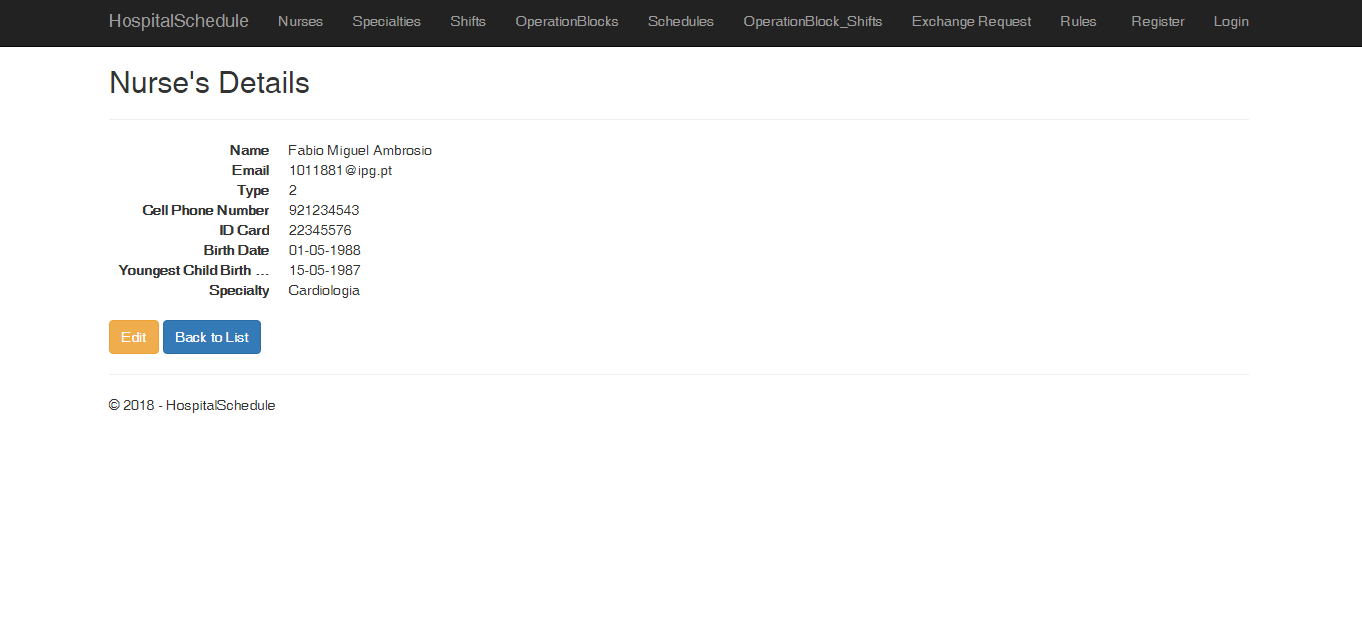
Protótipo de tudo relacionado com os enfermeiros

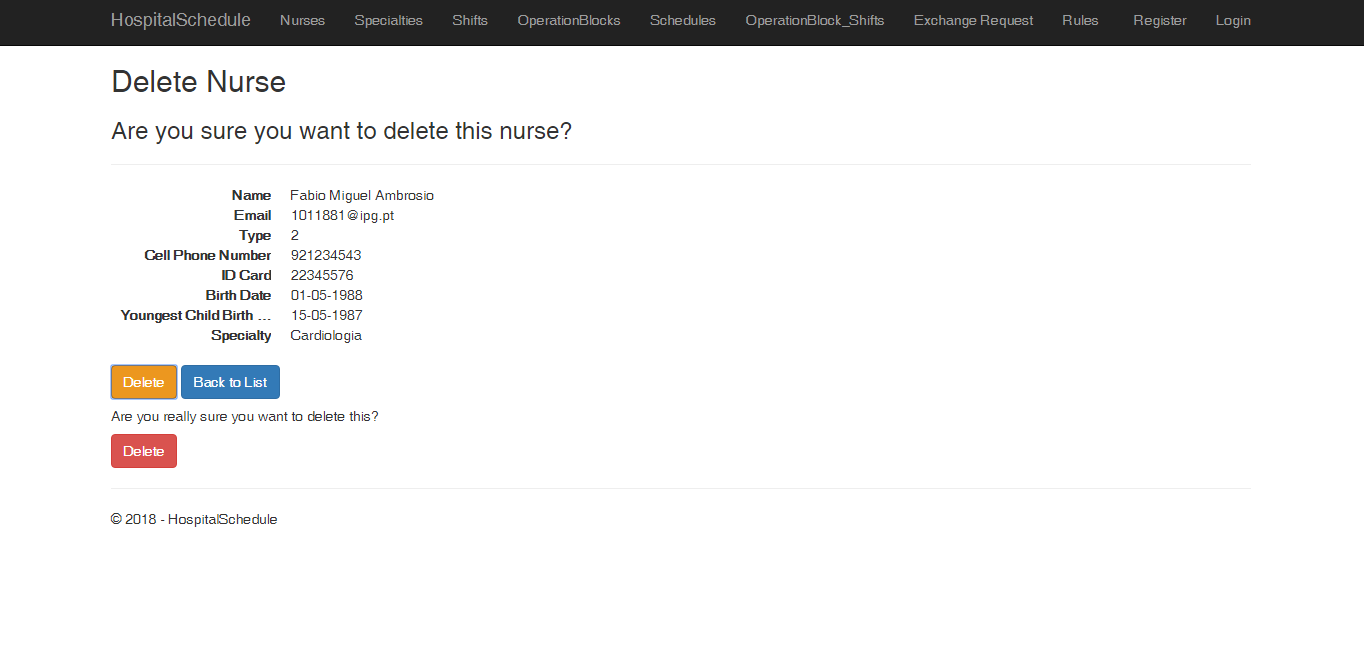




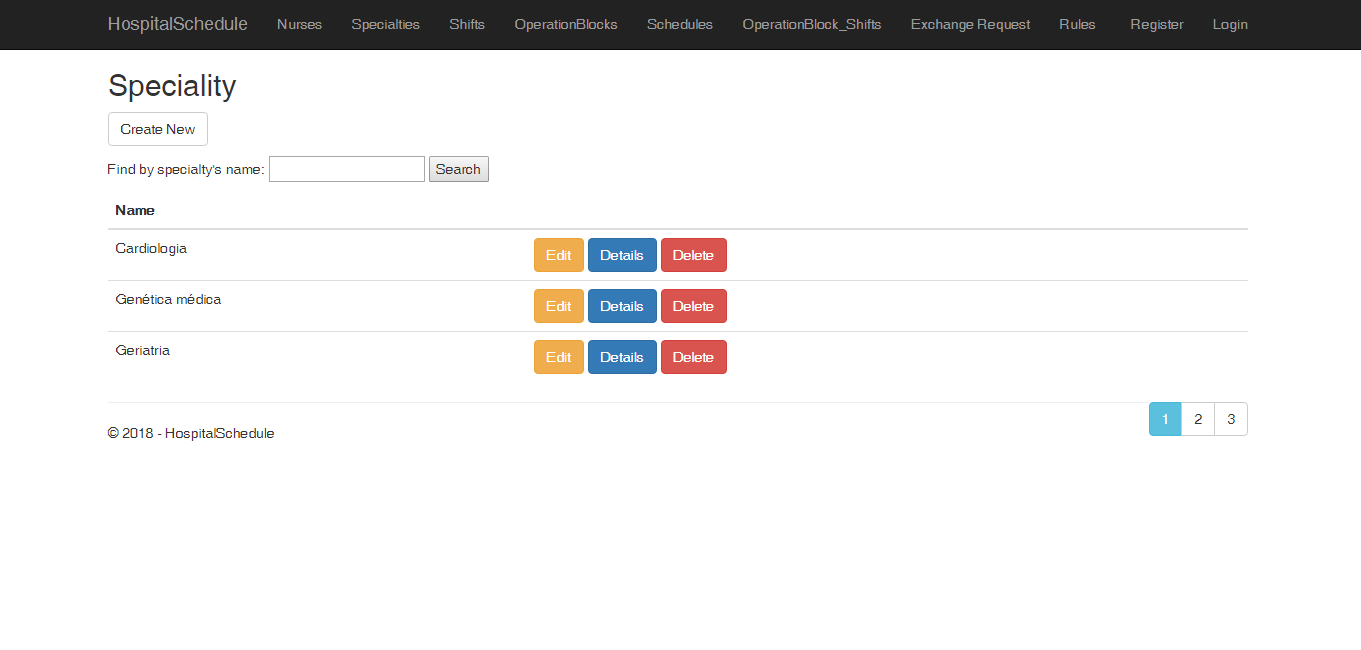


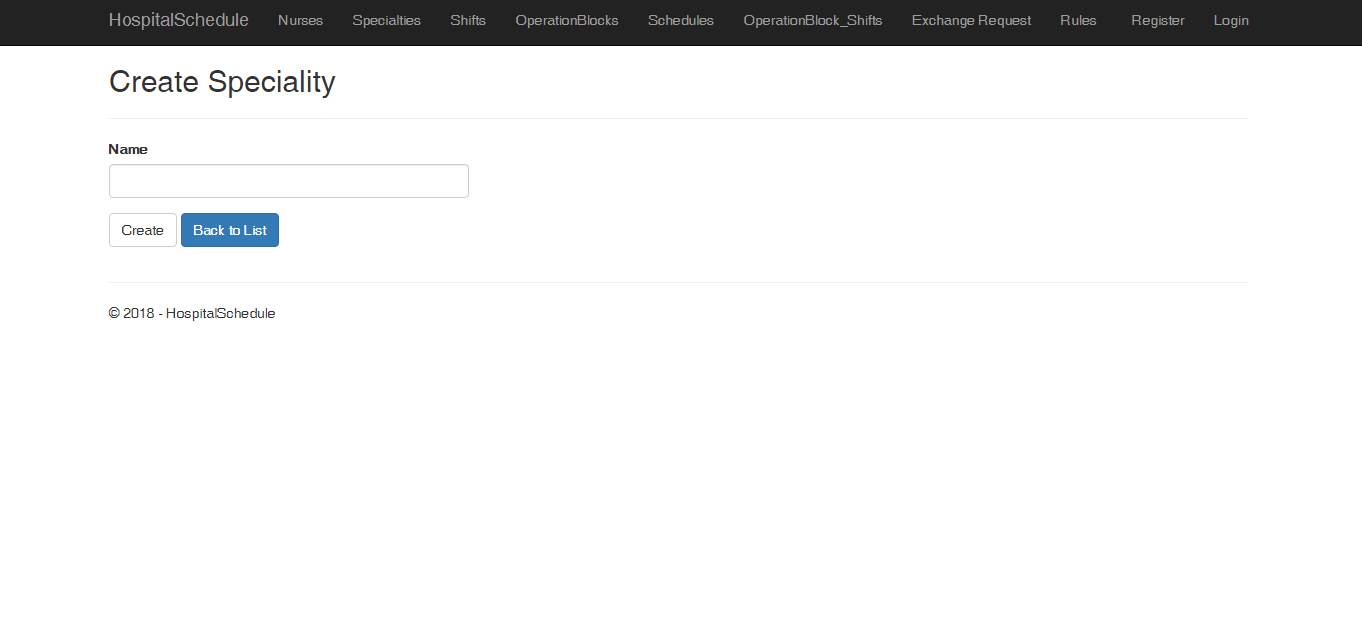
 

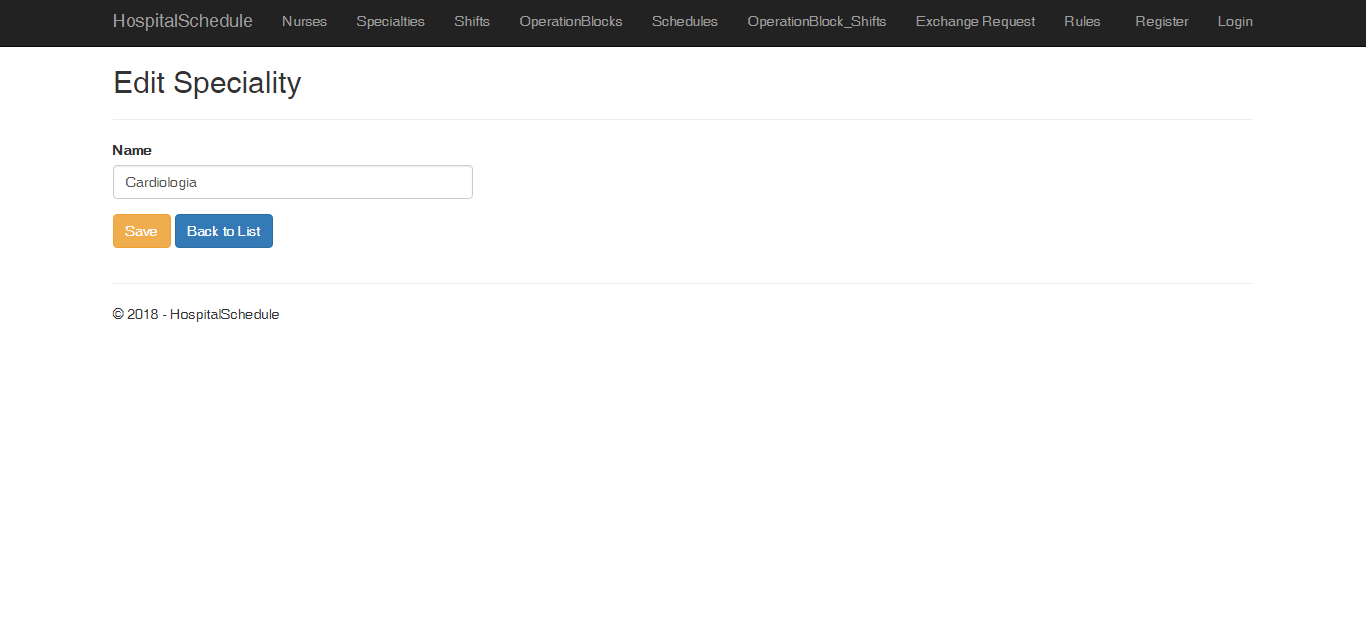


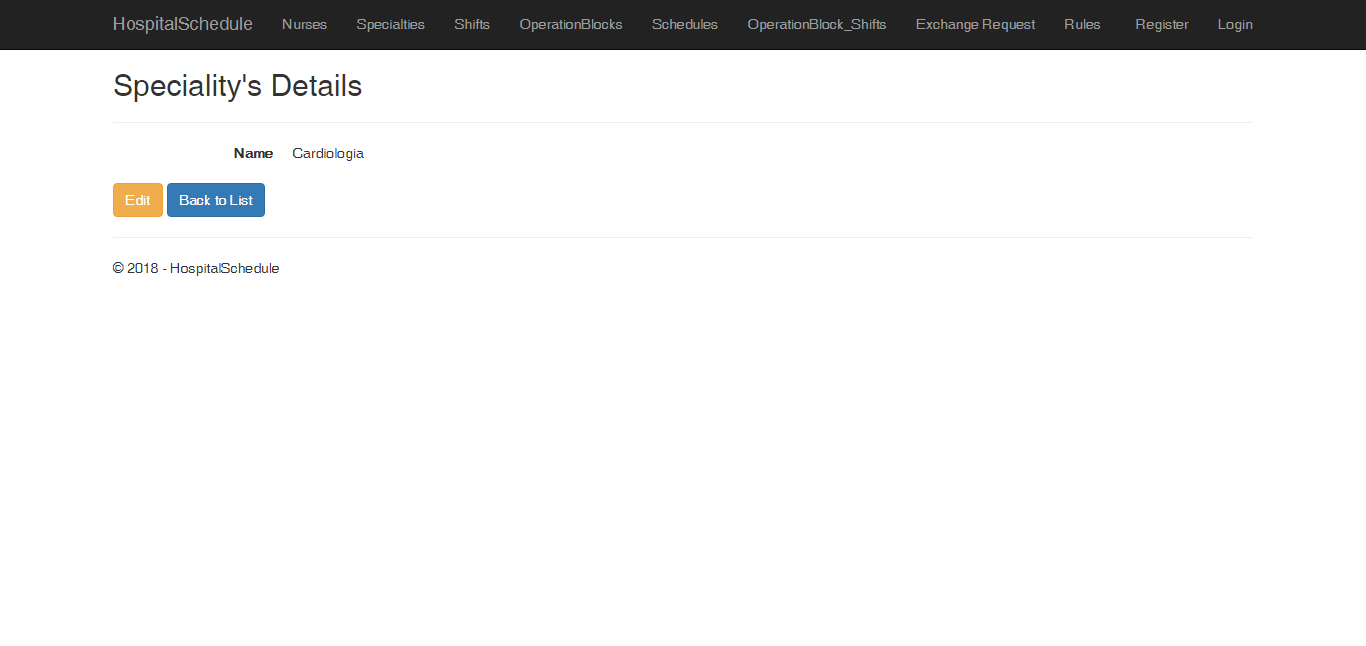


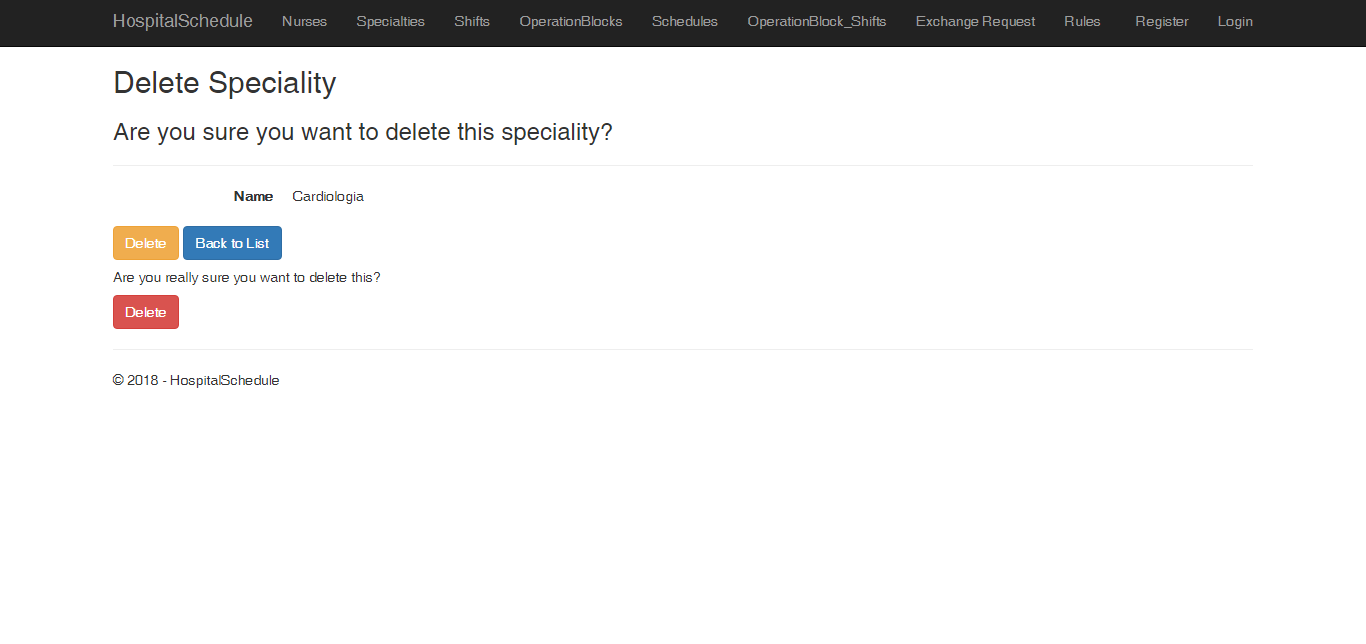
Protótipo de tudo relacionado com as especialidades



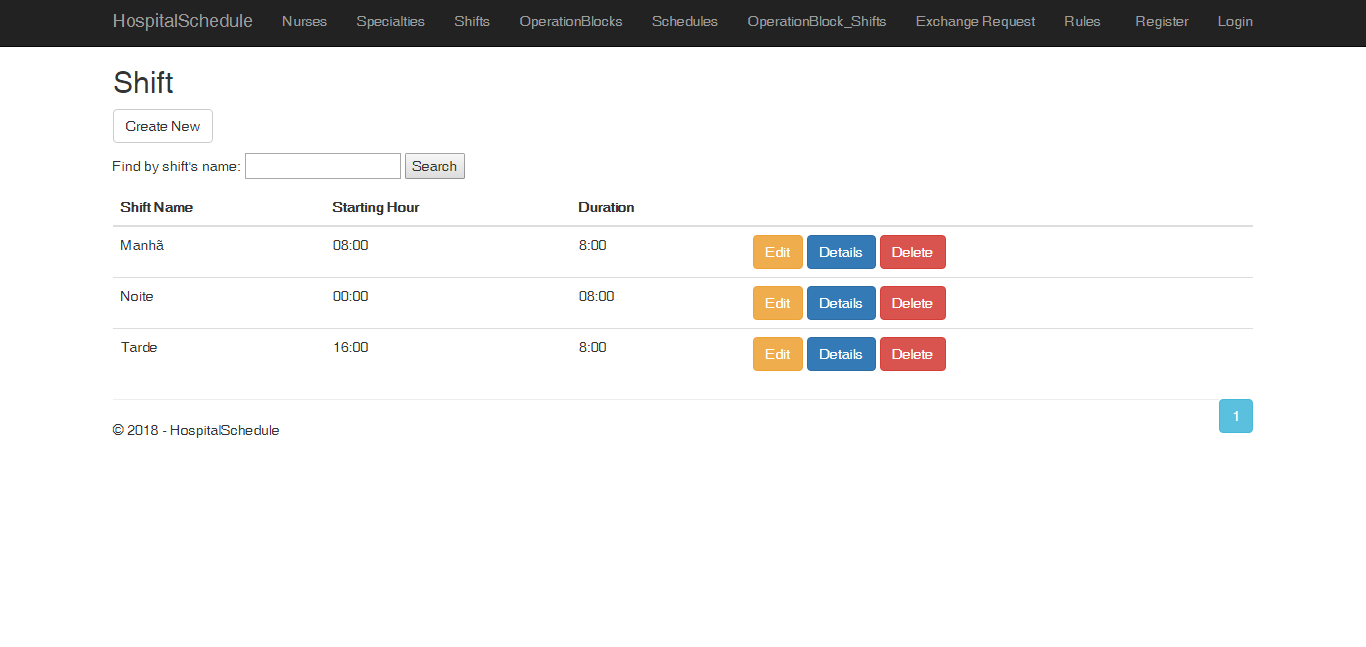


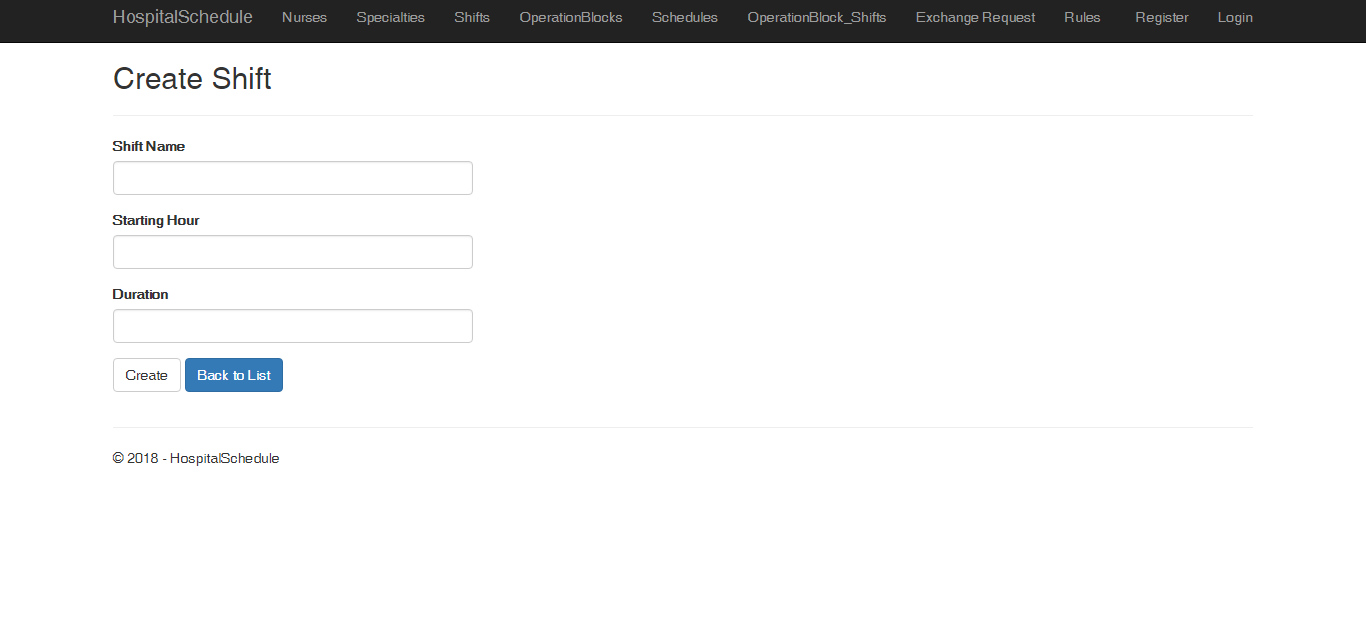


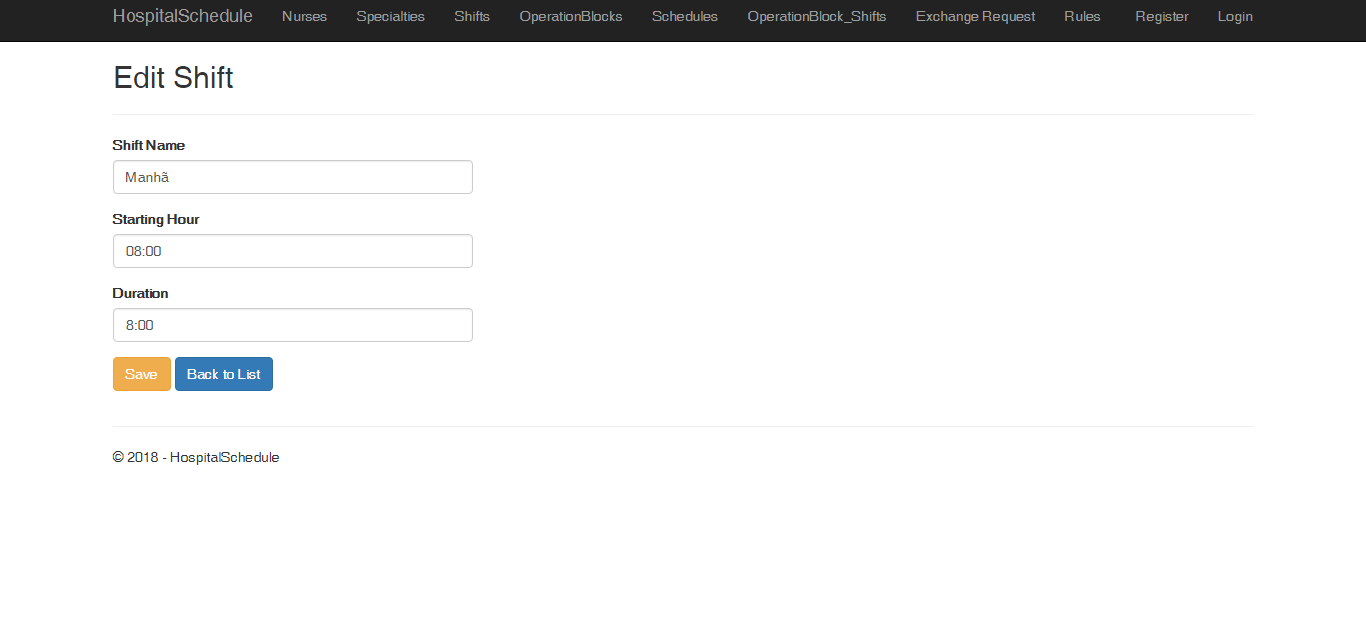


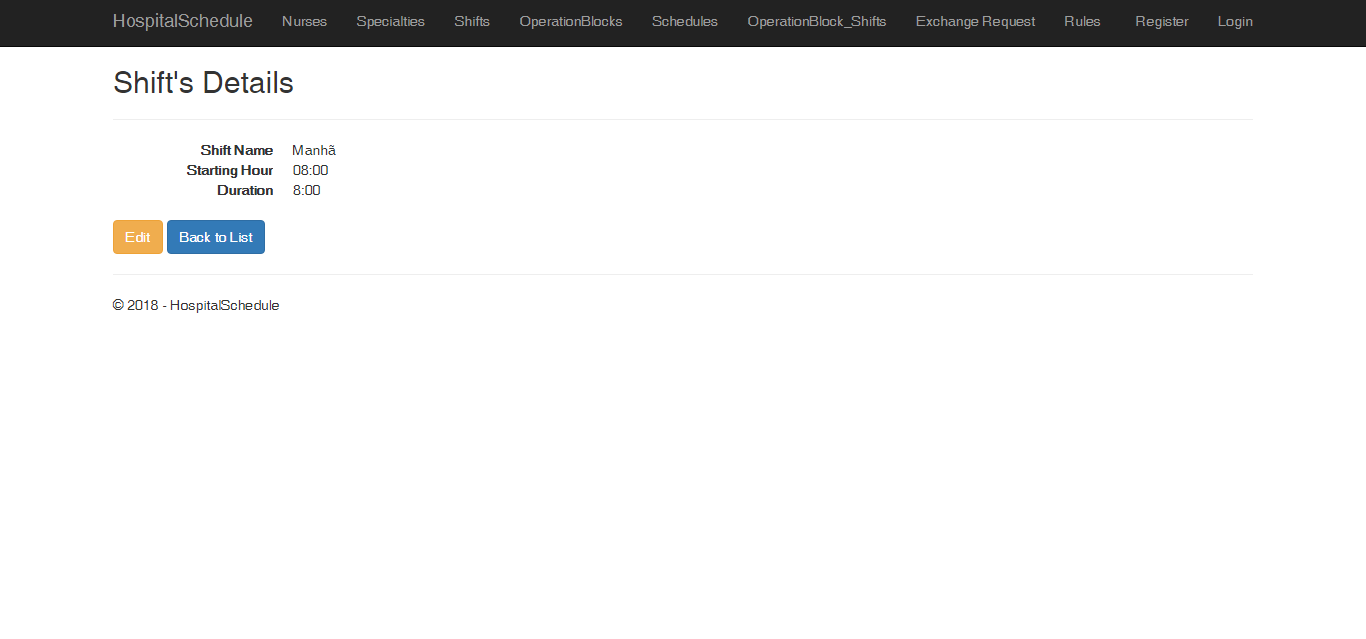


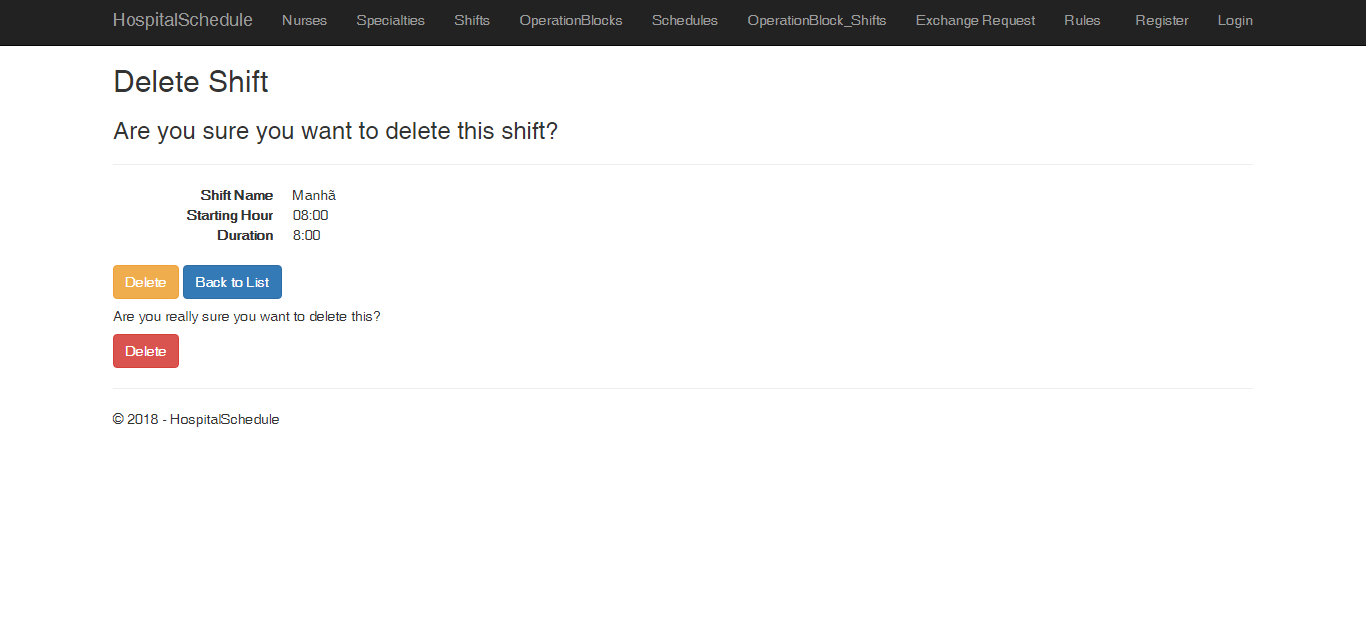
Protótipo de tudo relacionado com os turnos



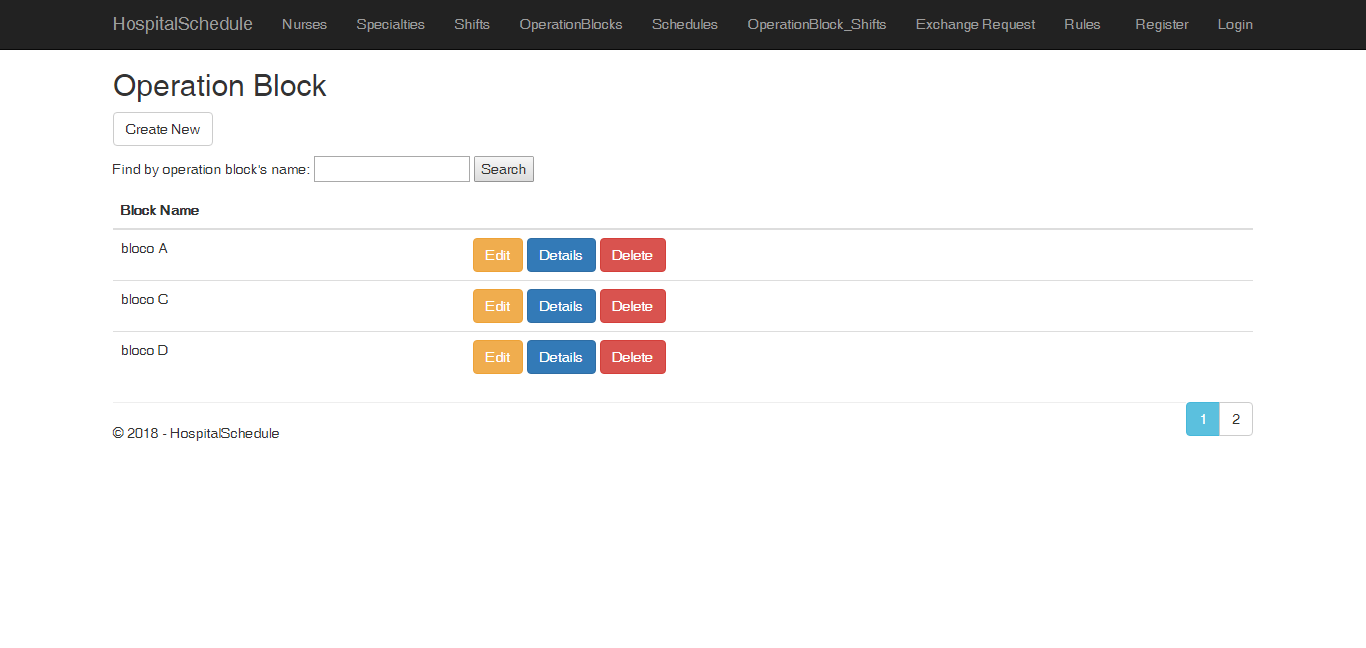


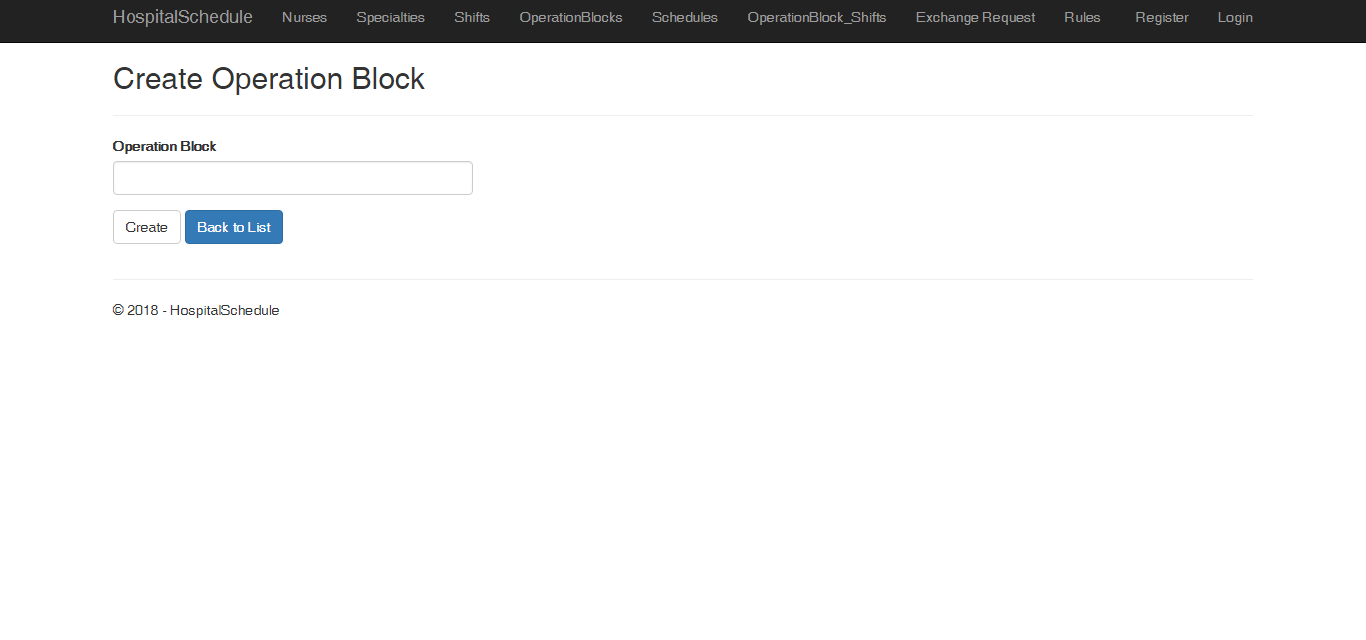


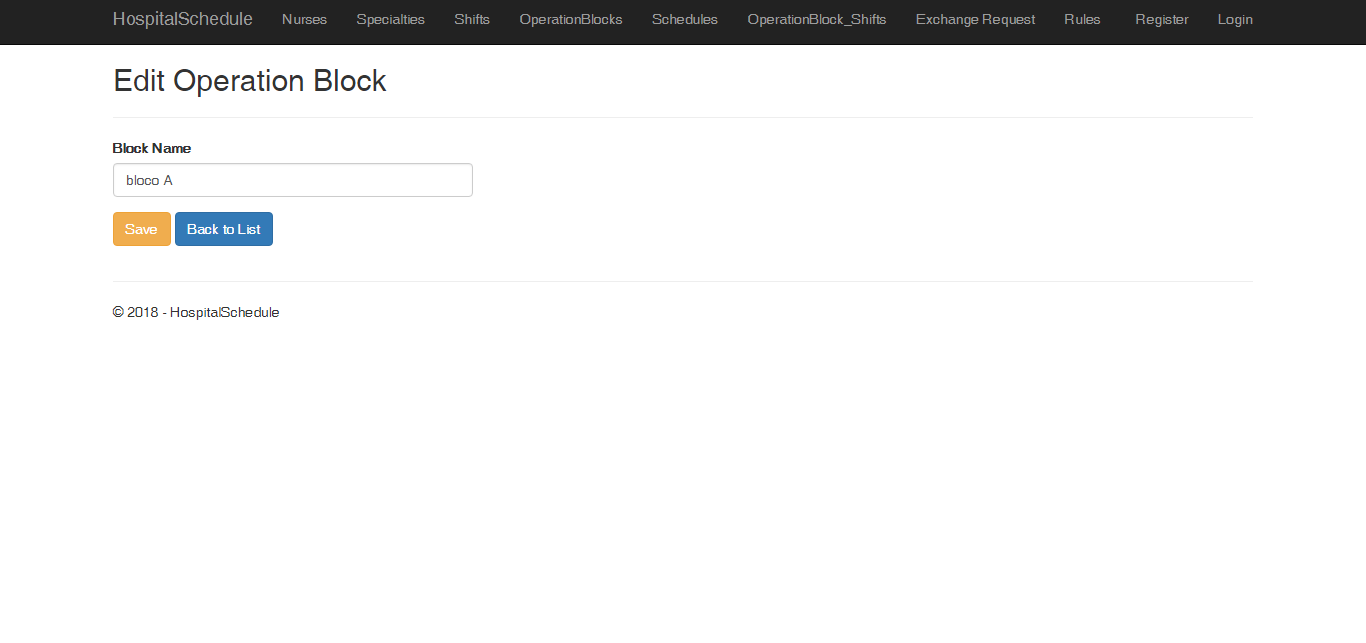


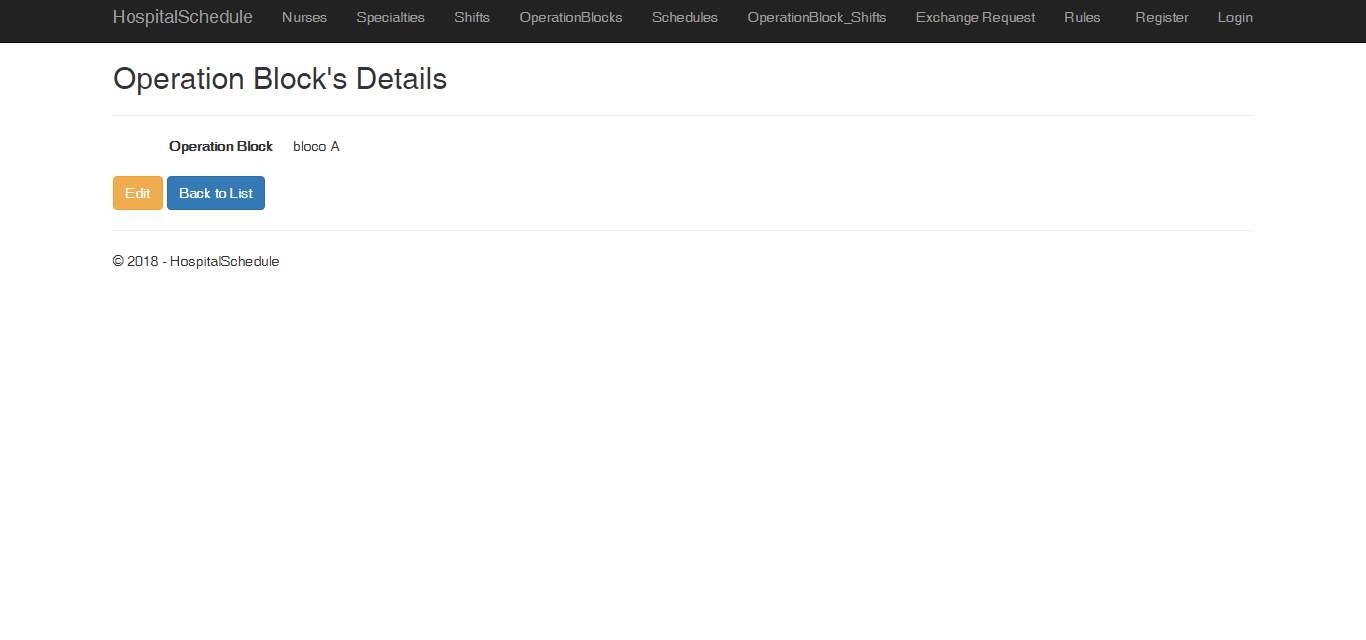


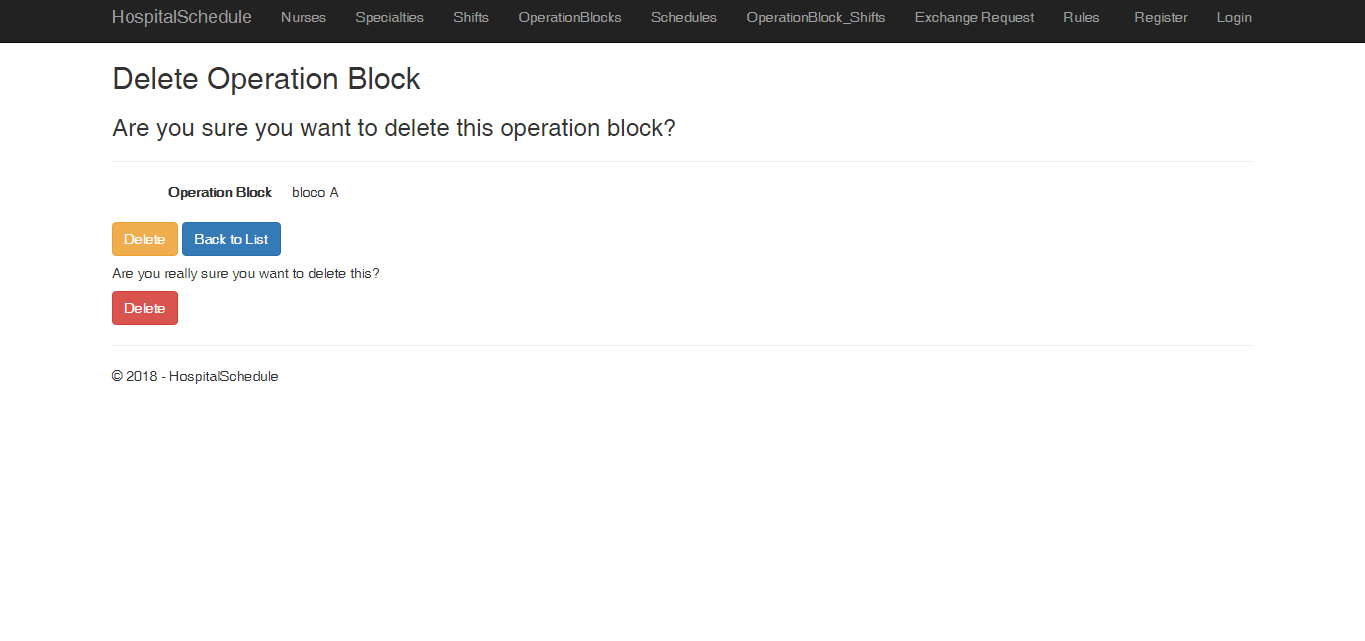
Protótipo de tudo relacionado com os blocos operatórios



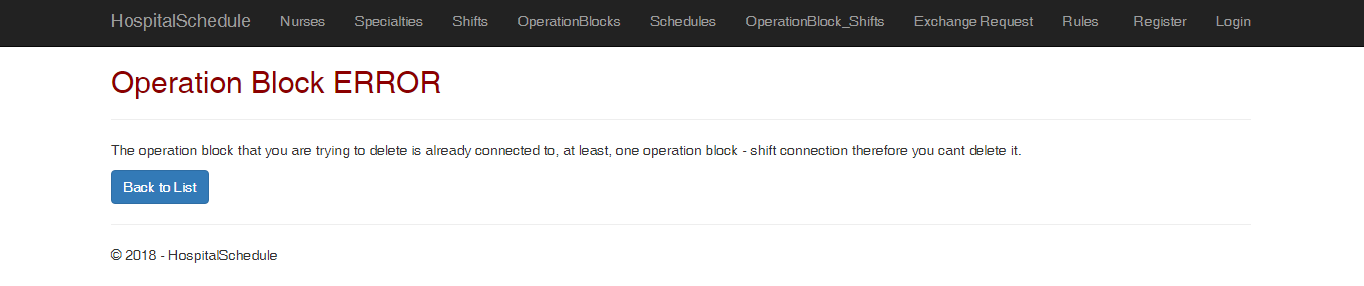




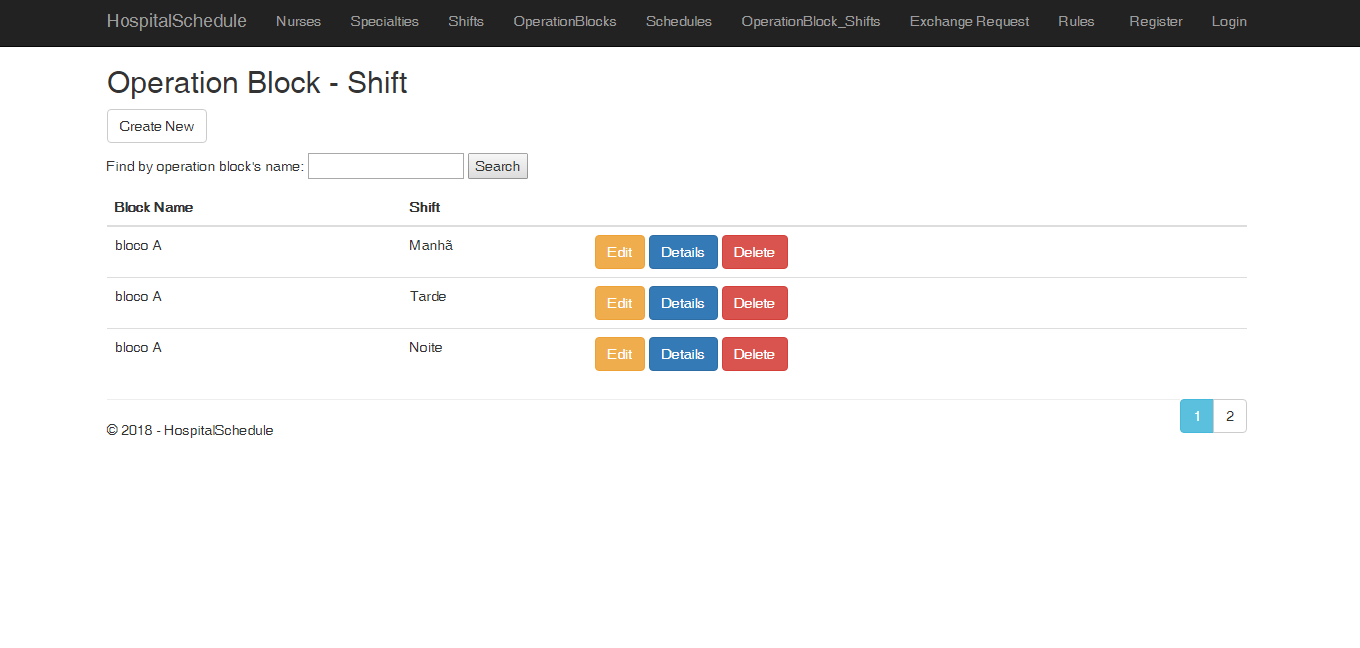


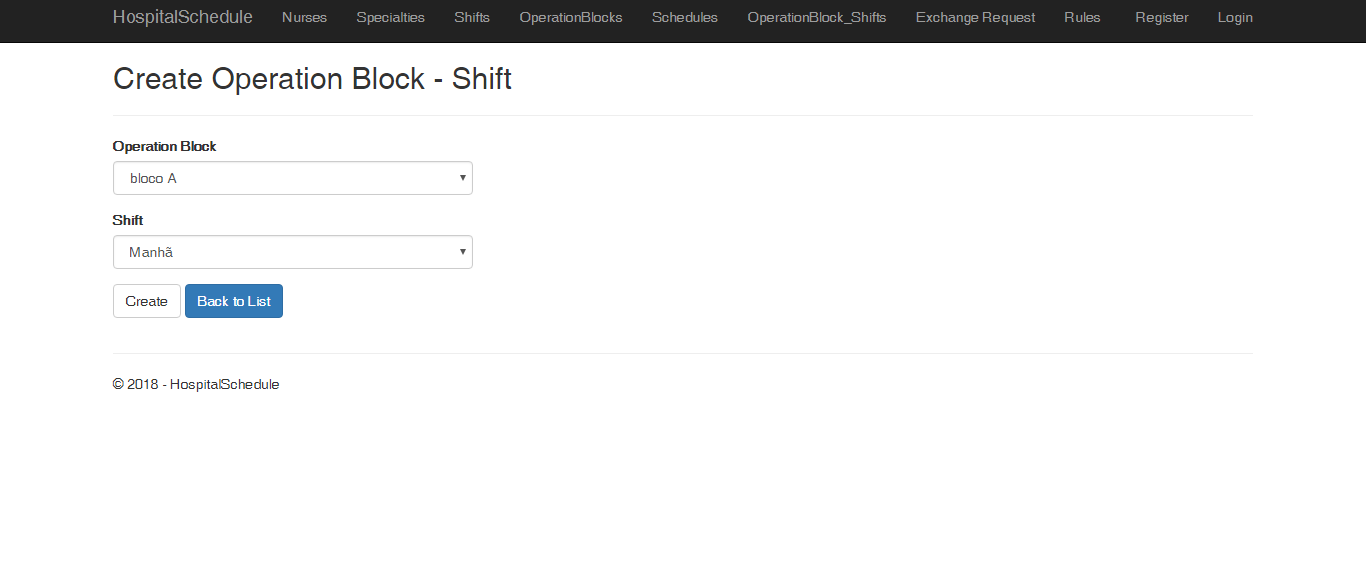


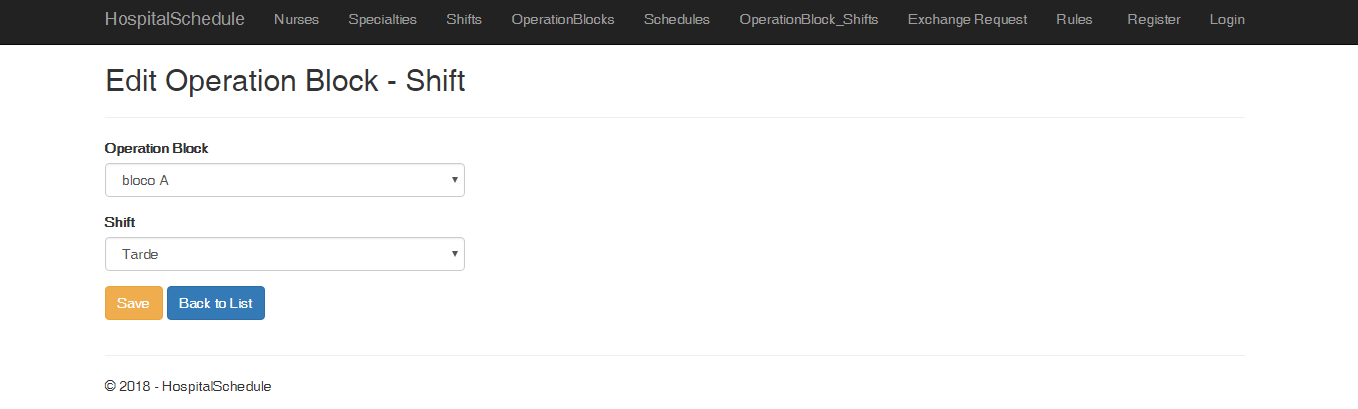
Exemplo de uma Mensagem de Erro

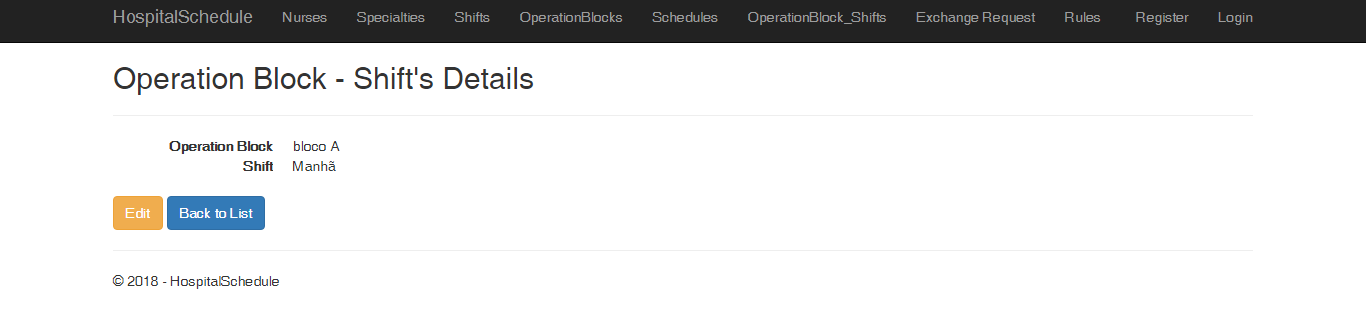


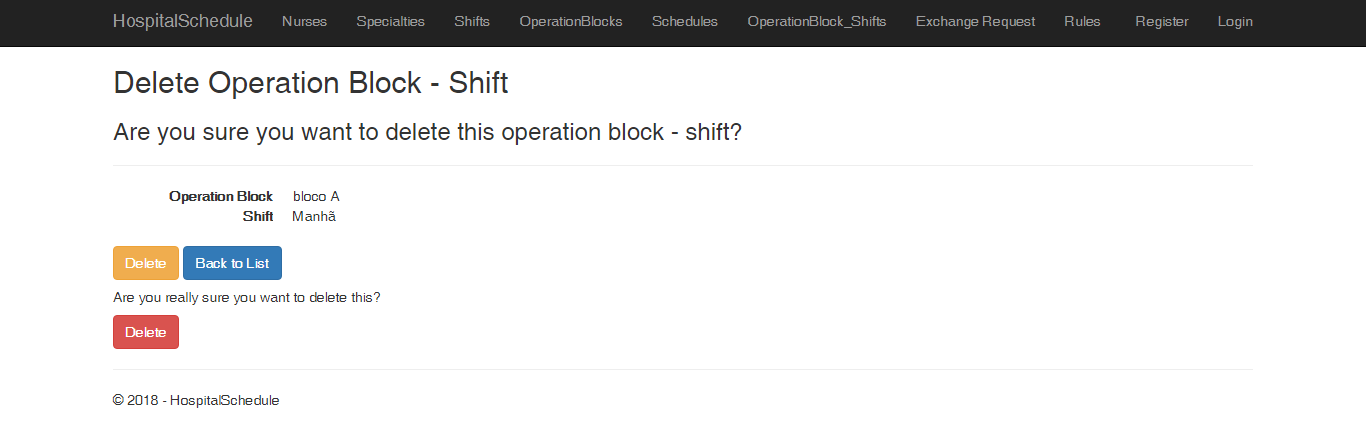
Protótipo de tudo relacionado com a ligação entre os blocos operatórios e os turnos



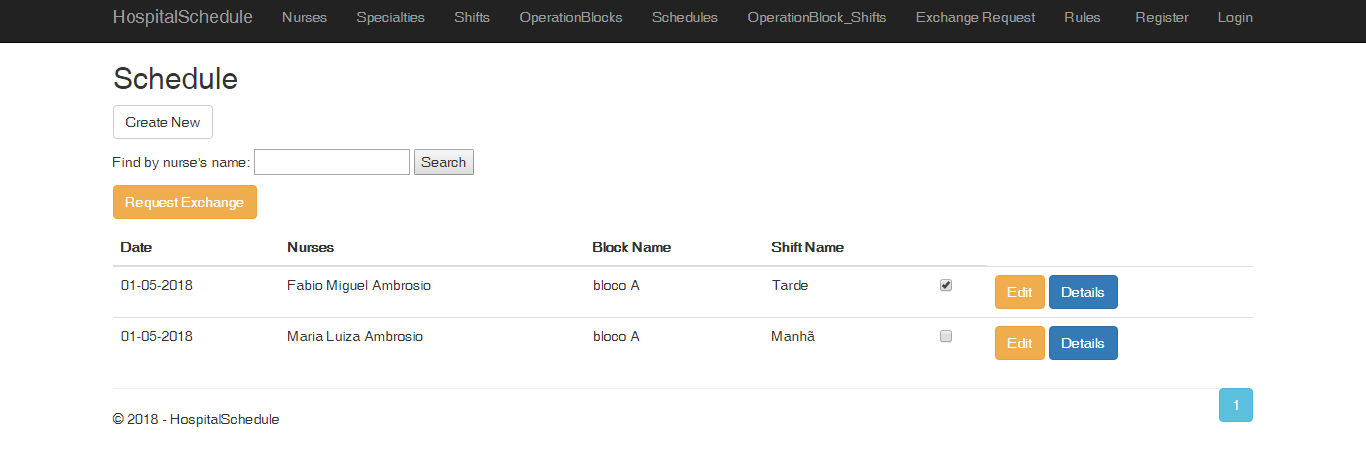


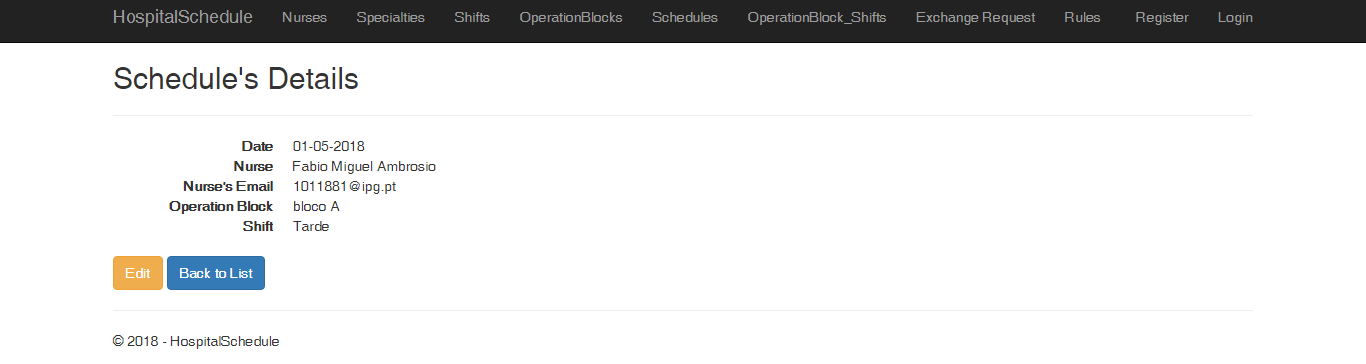




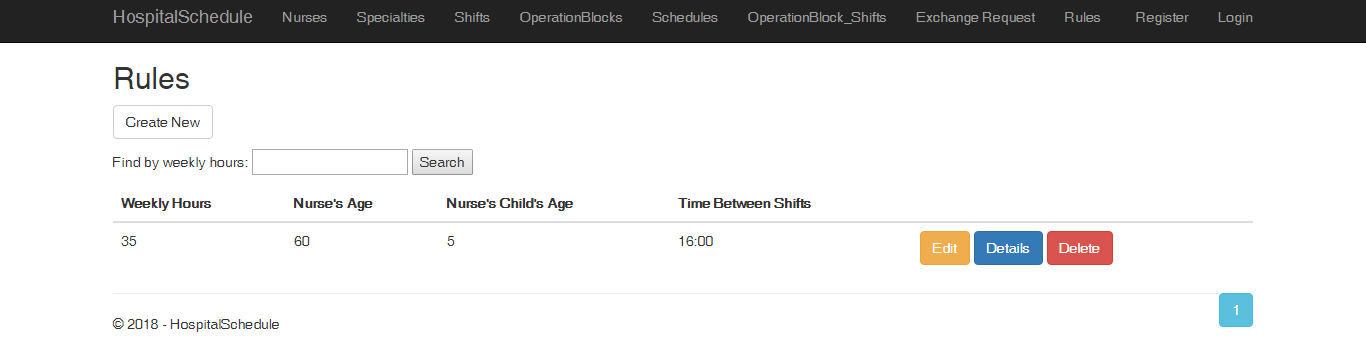


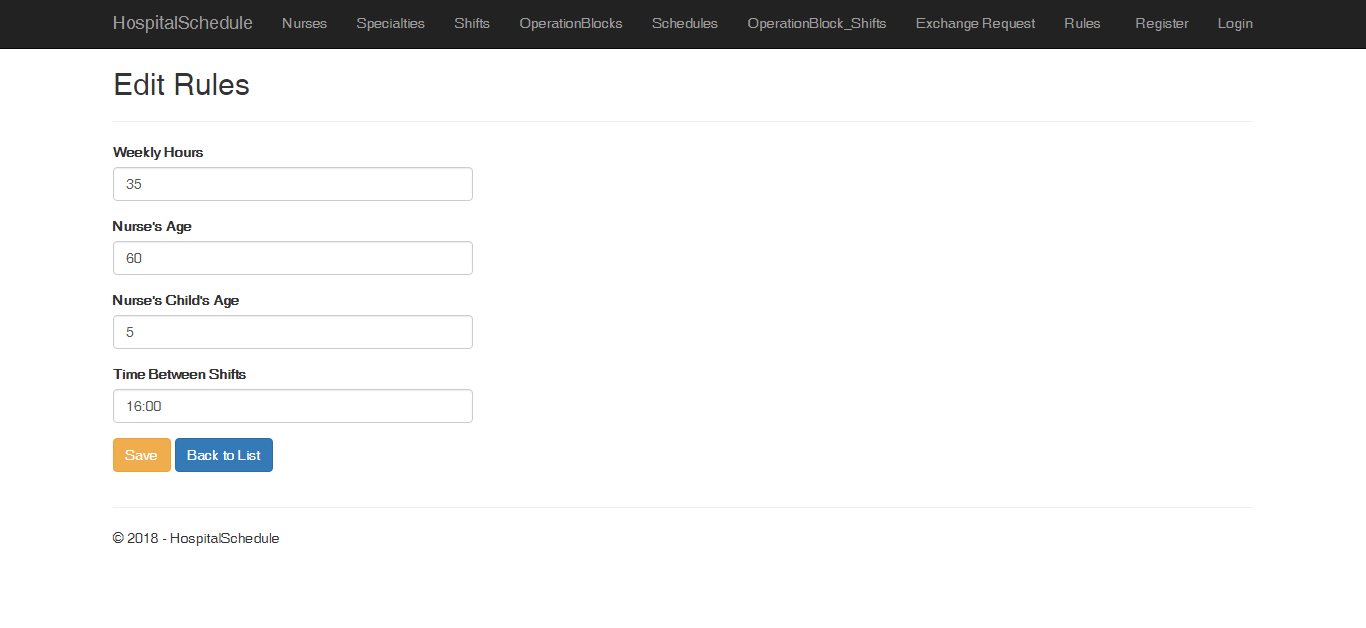
Protótipo de tudo relacionado com os horários

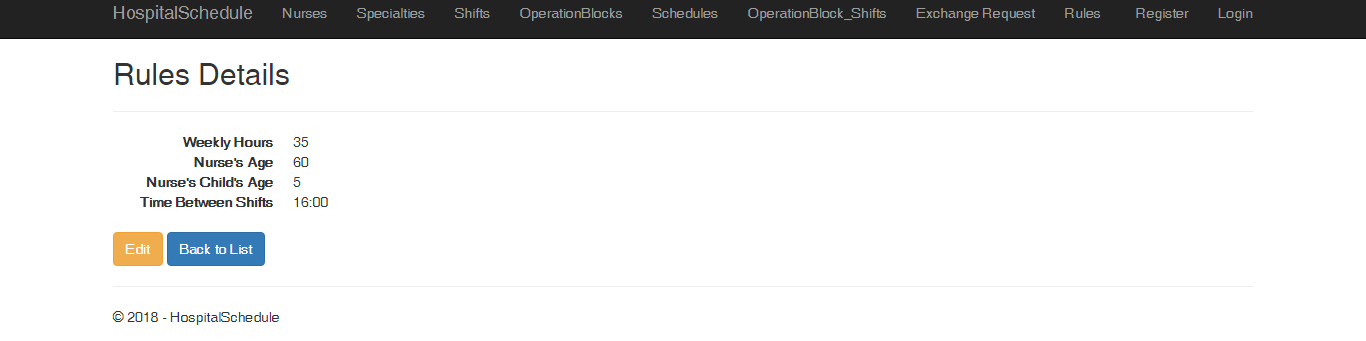


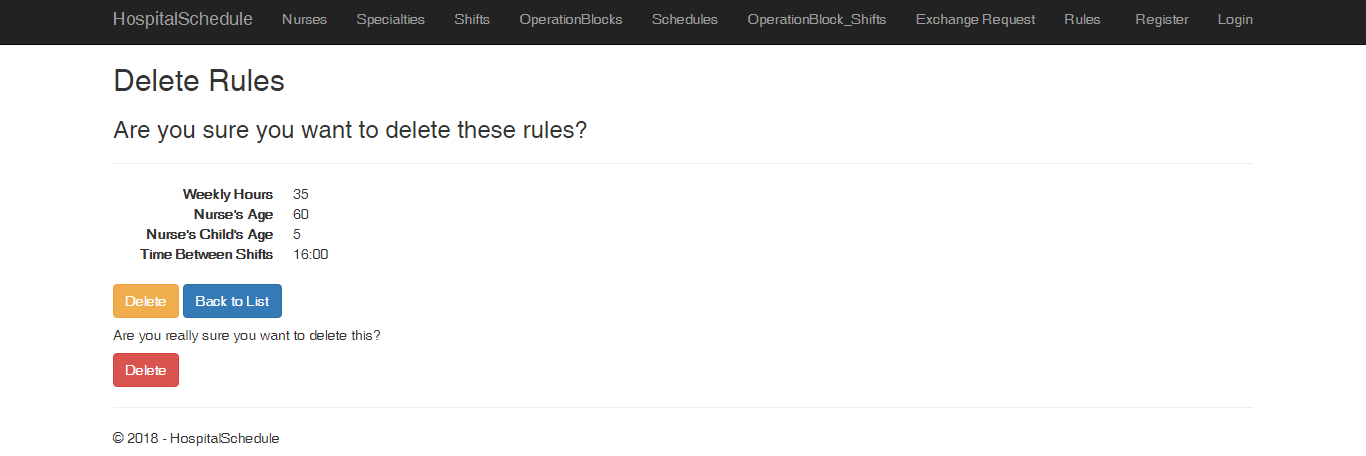


Protótipo de tudo relacionado com as rules

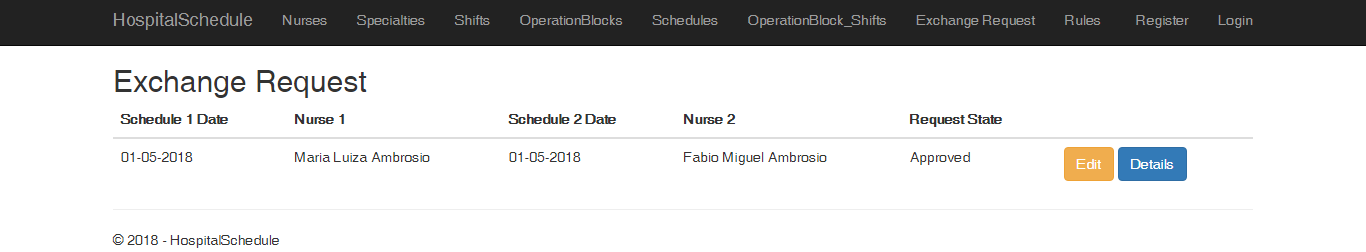


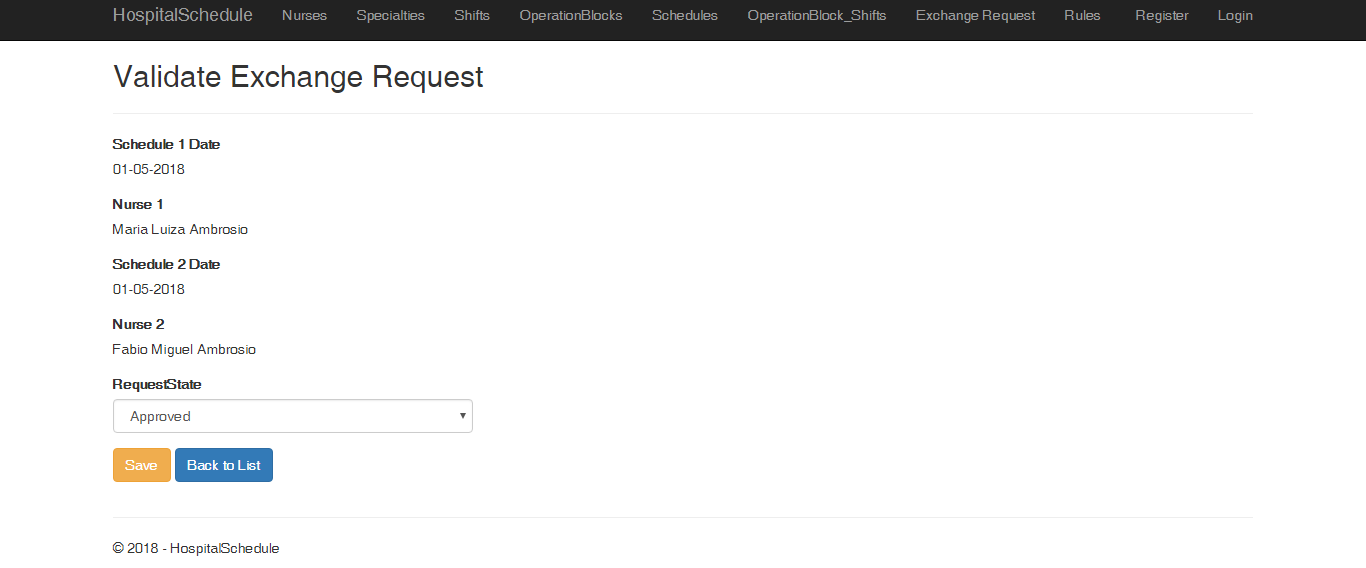


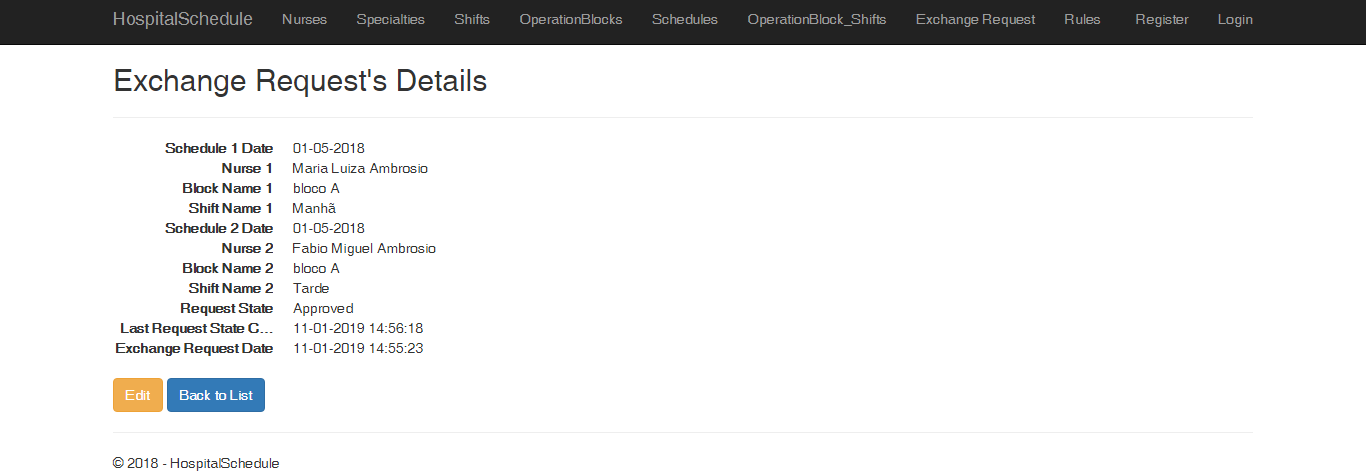




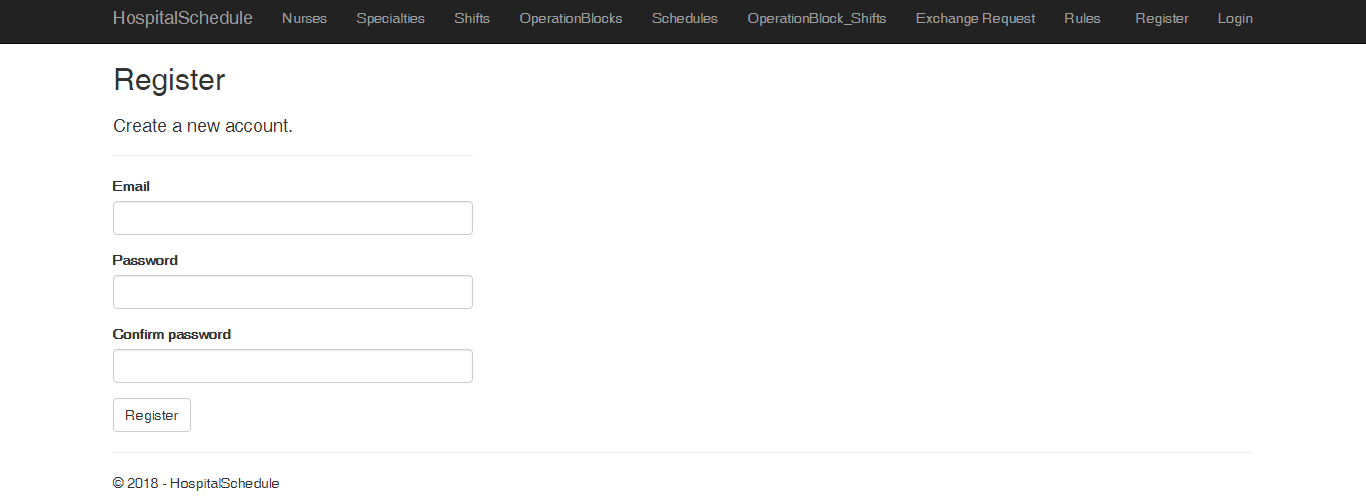
Protótipo de tudo relacionado com os pedidos de troca



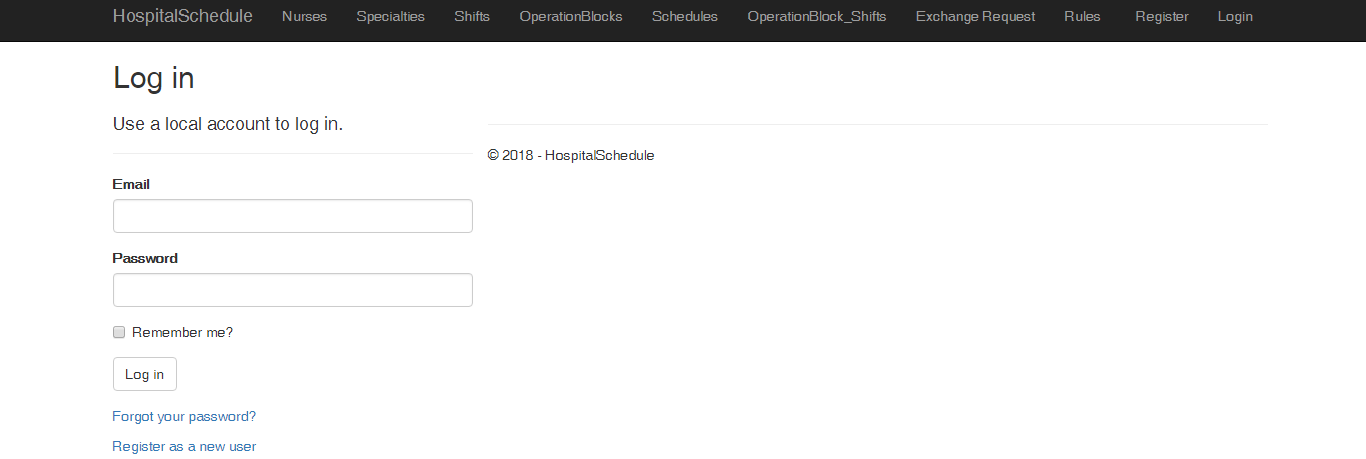




Protótipo de registar



Protótipo de login



Conclusão

Apesar das diversas impasses no progresso do trabalho como por exemplo a curva de dificuldade do software utilizado, neste caso o Visual Studio com o Asp.Net MVC, conseguimos cumprir os nossos objetivos.

Conseguimos criar um software que realiza o escalonamento de enfermeiros por bloco operatório, os pontos alvo do nosso projeto também foram cumpridos, a criação dos horários dos enfermeiros recorrendo a um algoritmo que tem em conta as regras impostas pela legislação e o bloco operatório desejado e a realização de trocas de turnos entre enfermeiros.

## Anexos

Implementação (código) da validação dígito de controlo do CC

public bool ValidateDocumentNumber(string DocumentNumber)

{

if (DocumentNumber == null)

{

return false;

}

int sum = 0;

bool secondDigit = false;

if (DocumentNumber.Length != 12)

throw new ArgumentException("Tamanho inválido para número de documento.");

for (int i = DocumentNumber.Length - 1; i >= 0; --i)

{

//string upper = numeroDocumento.ToUpper();

int valor = GetNumberFromChar(DocumentNumber[i]);

if (secondDigit)

{

valor \*= 2;

if (valor > 9)

valor -= 9;

}

sum += valor;

secondDigit = !secondDigit;

}

return (sum % 10) == 0;

}

public int GetNumberFromChar(char letter)

{

switch (letter)

{

case '0': return 0;

case '1': return 1;

case '2': return 2;

case '3': return 3;

case '4': return 4;

case '5': return 5;

case '6': return 6;

case '7': return 7;

case '8': return 8;

case '9': return 9;

case 'A': return 10;

case 'B': return 11;

case 'C': return 12;

case 'D': return 13;

case 'E': return 14;

case 'F': return 15;

case 'G': return 16;

case 'H': return 17;

case 'I': return 18;

case 'J': return 19;

case 'K': return 20;

case 'L': return 21;

case 'M': return 22;

case 'N': return 23;

case 'O': return 24;

case 'P': return 25;

case 'Q': return 26;

case 'R': return 27;

case 'S': return 28;

case 'T': return 29;

case 'U': return 30;

case 'V': return 31;

case 'W': return 32;

case 'X': return 33;

case 'Y': return 34;

case 'Z': return 35;

}

throw new ArgumentException("Valor inválido no número de documento.");

}

}