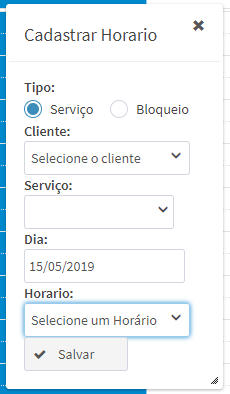
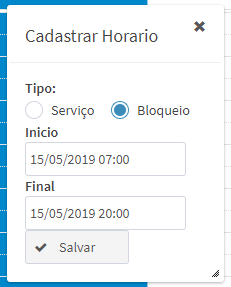
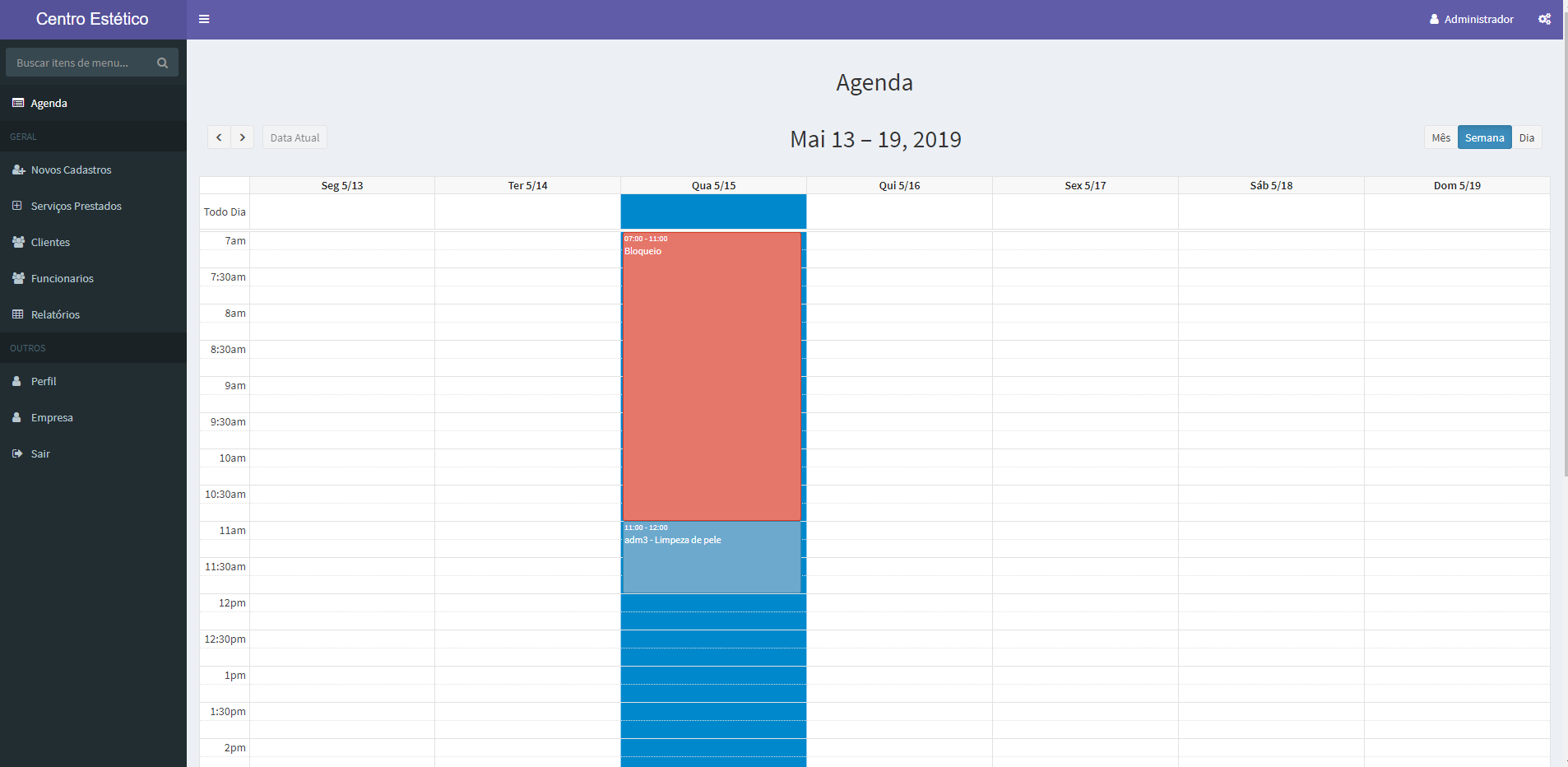
 Essa é a tela da Agenda do administrador, onde aparecerão todos os serviços cadastrados durante a semana, ele pode escolher ver somente o dia, toda a semana ou o mês inteiro. Clicando em algum dos dias abrirá uma modal com as opções de cadastrar um serviço ou um bloqueio de agenda.

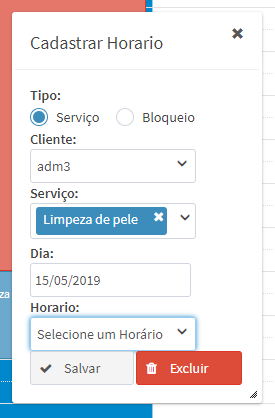


No tipo serviço o administrado tem que escolher qual cliente que deseja realizar o agendamento, escolher quais serviços serão realizados(pode ser mais de um), selecionar o dia do serviço, e por último o horário que será realizado o serviço.

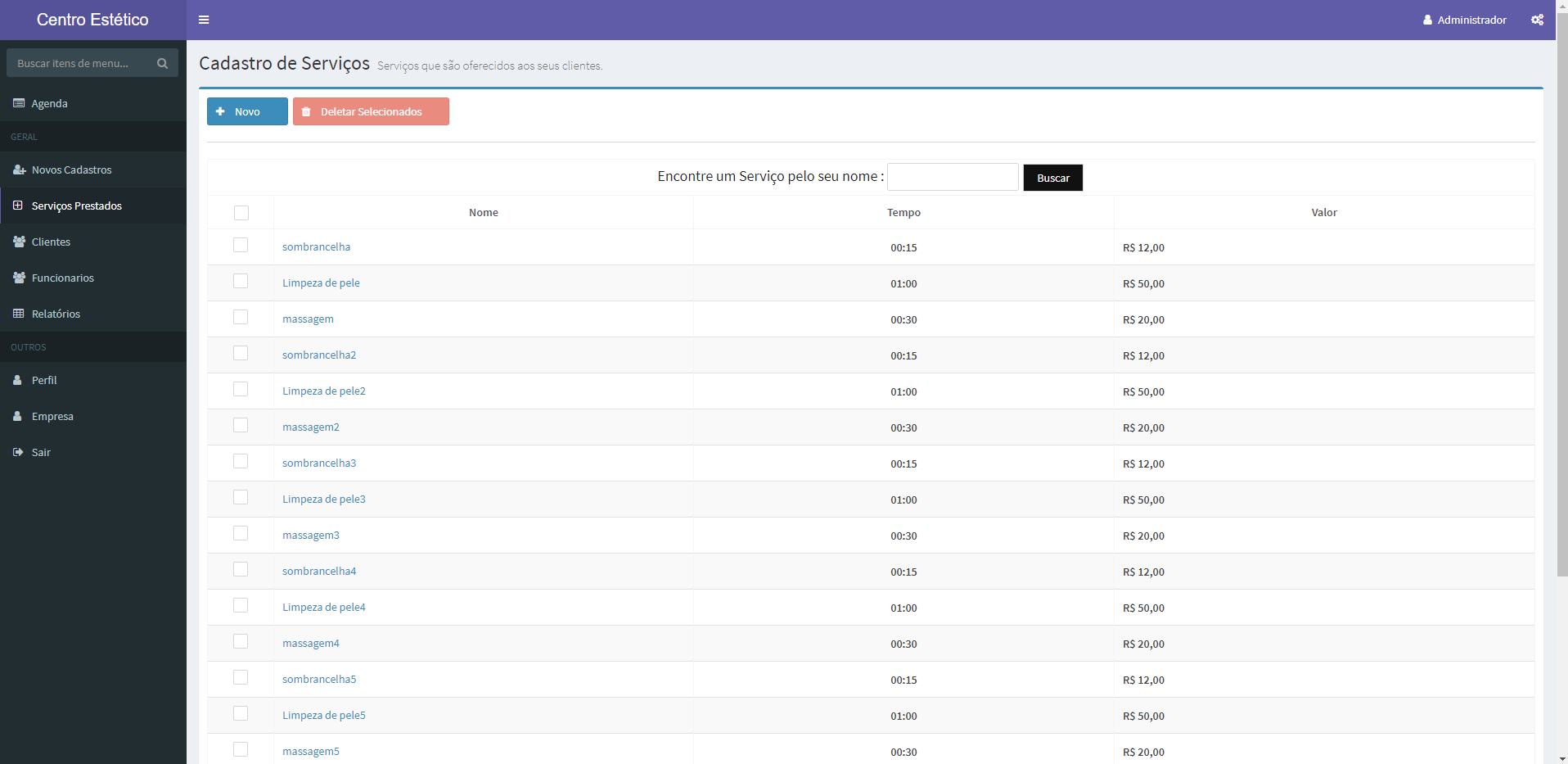


Na aba bloqueio o administrador tem que escolher o incio e fim do bloqueio, sendo que já vem pré-definido como início o primeiro horário do dia e como final o último horário do dia.

 Após o serviço e/ou bloqueio ser cadastrados eles apareceram no calendário, de forma que um serviço tenha a cor azul e um bloqueio tenha a cor vermelha. Clicando no serviço ou bloqueio cadastrado abrirá novamente o modal, mas agora, com os dados do cadastro selecionado e com um botão para excluir o cadastro.



Se o administrador desejar alterar alguma informação do cadastro, basta ele fazer a alteração e salvar.

 Na tela dos serviços temos uma dataTable com todos os serviços cadastrados, com o botão para realizar um novo cadastro, deletar cadastros, um filtro para buscar os cadastros pelo nome e se o administrador clicar no nome do serviço abrirá a página cadastro de serviços com os dados do serviço.

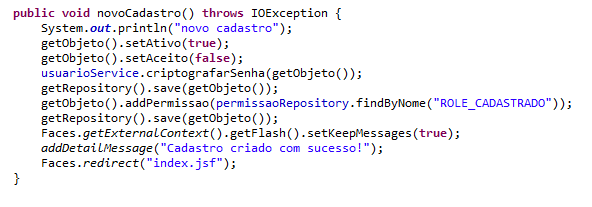


Nessa tela pode ser cadastrado um novo usuário, inserindo os dados nos inputs indicados e criando sua conta.

Código incluir



O código para o cadastro de um novo é simples, utilizamos um form com inputText padrões do primefaces para a inserção de dados do usuário e foi utilizado o componente <p:password /> do primefaces que faz a confirmação da senha automaticamente, e um <p:commandButton /> que faz a requisição ao servidor para salvar o novo cadastro no banco utilizando o spring. Usamos também a tag p:messages para as mensagens de erro e validateRegex para validar um e-mail da forma correta.

 Na parte servidor temos uma função que seta alguns valores automaticamente e faz a inserção no banco utilizando o spring.

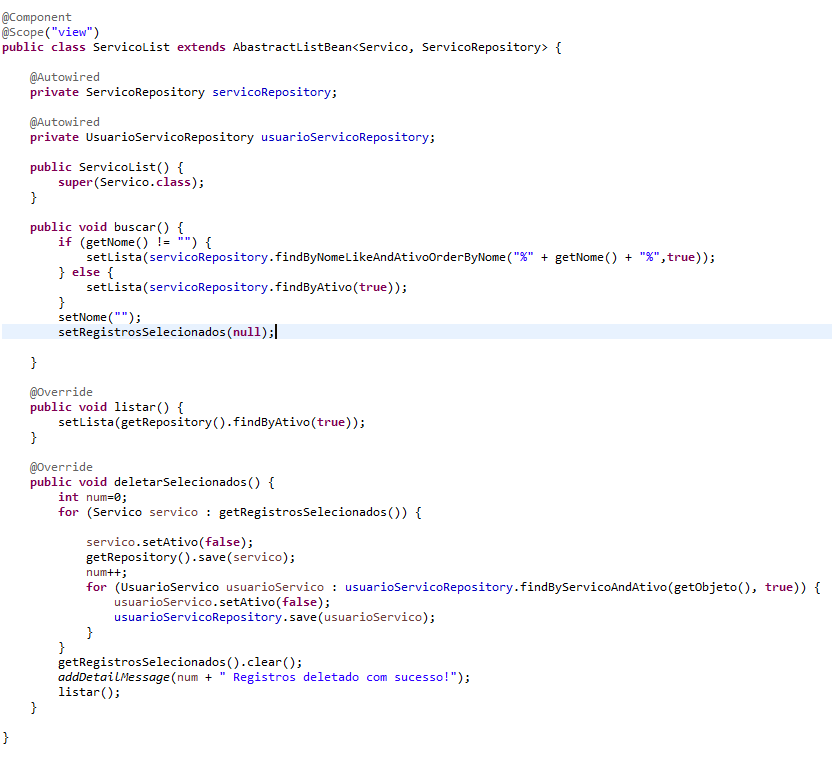


Foi utilizado o datatable selection padrão do prime faces, onde foi feito somente a configuração do mesmo, como no value foi colocado o List de serviços, o var é o nome da variável para utilizarmos, emptyMessage é a mensagem quando estiver vazio, rowKey é a chave no model, no selection informamos a variável onde será armazenada os serviços selecionados na tabela, e os demais configurações são relacionados aos botões e para deixar a tabela como paginator. Nesse DataTable temos cinco eventos ajax para quando uma linha/row da tabela seja selecionada, esse model seja enviado para o servidor e armazenado no nosso List de registros selecionados, o mesmo vale para quando o registro é disselecionado. Além disso é necessário que na coluna dos checkBox sele configurada como selectionMode= "multiple" para que possa selecionar mais de um checkBox.

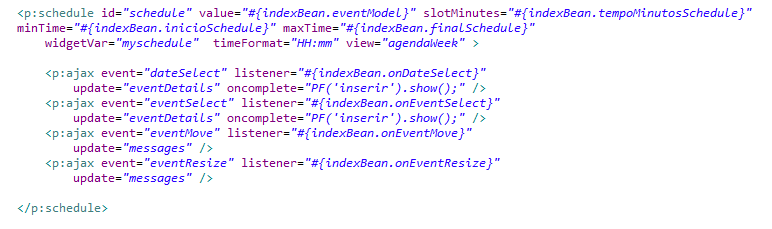
Também foi utilizado o <p:link para que clicando no nome do serviço o administrador seja redirecionado para a pagina de alteração, com o com o <f:param será enviado o id do serviço como parametro para que o objeto ja estaja corretamente selecionado.



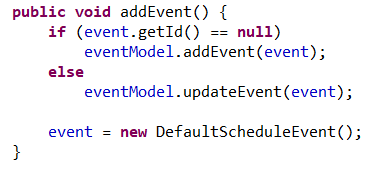
Na parte do servidor eu utilizo um AbastractListBean onde tem as funções padrões para o crud.

Além dele temos o ServicoList que extende o Abastract onde fazemos todas as operações utilizando o Spring, mas a diferença desse classe comparando com o abastract é que no hora de deletar um serviço, ele só desativa o mesmo, não apaga no banco.

Schedule/calendario



No calendário nos temos eventos ricos que ajudam no crud de um agendamento, no schedule o value recebe um ScheduleModel como parâmetro, que terá nossos agendamentos, além de algumas tags para a configuração visual do calendário.

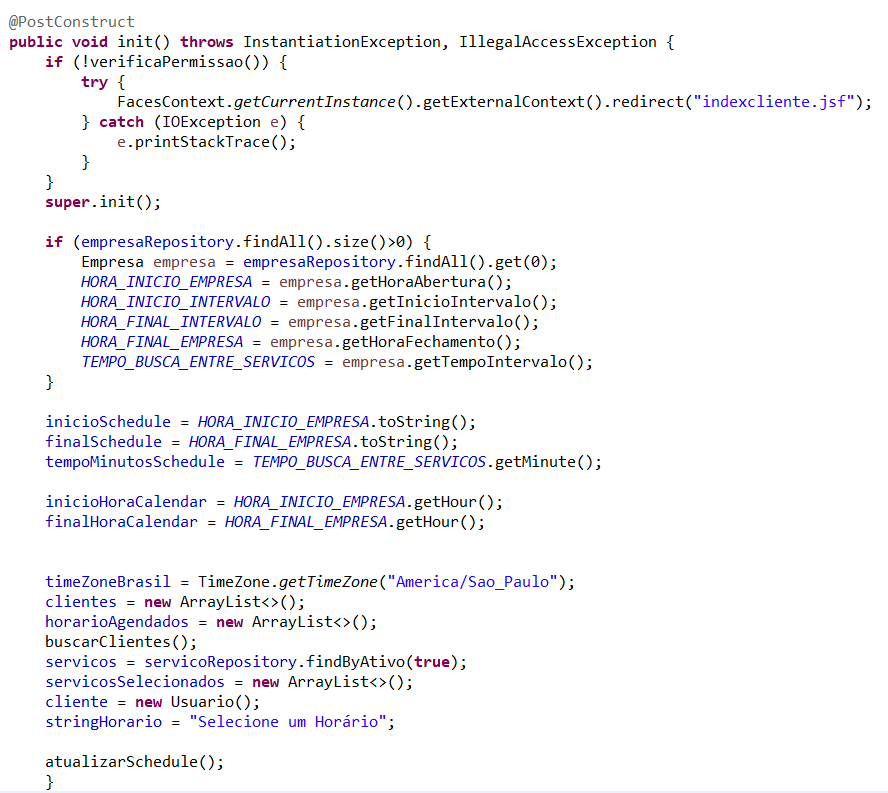




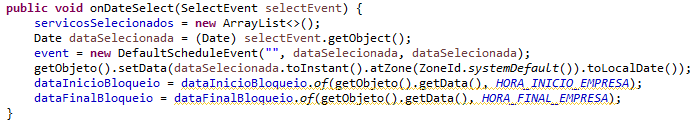
Primeiramente para setarmos os nossos agendamentos cadastrados, nós temos essa função que pega os dados do banco e transformamos esses dados em um DefaultScheduleEvent, esses eventos serão atribuídos a uma variavel ScheduleModel que está sendo envocada ao schedule. Aqui também setamos a cor diferenciada entre um agendamento e um bloqueio, colocando o StyleClass de um bloqueio como "btn-danger", com essa tag que vem do bootstrap deixamos ele com a cor vermelha. E no sistema oque difere um agendamento de um bloqueio, é que no bloqueio não temos nenhum prestador de serviço cadastrado.



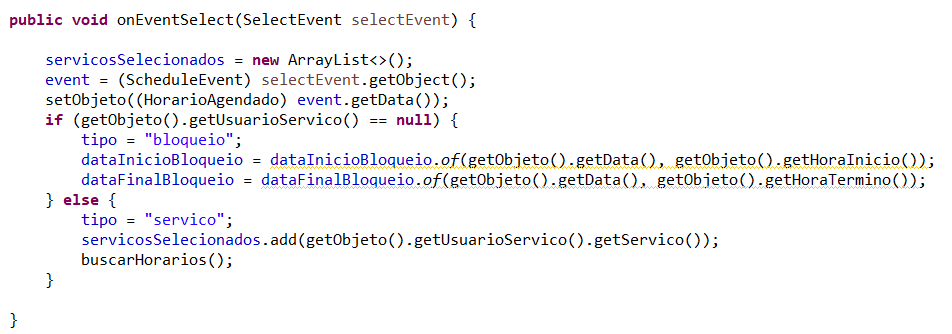
No nosso modal todos os inputs são dos padrões do primefaces e utilizamos um SelectOneRadio para alterar nosso formulário conforme um agendamento ou um bloqueio, cada um está em um outputPanel e conforme o radio selecionado é fica visível ou o outro. Temos um selectOneMenu para selecionar o cliente e o horário, um selectCheckboxMenu para os serviços com filter e multiple como true, para fazer a filtragem e poder marcar mais de um serviço. Temos também um calendar para selecionar o dia. Além disso temos o botão para salvar ou alterar um HorarioAgendado selecionado e um botão para excluir que só estará visível se estiver algum serviço selecionado. Também podemos ver que quando acontece uma alteração nos serviços ou na data é acionado o método buscar horários que fará a verificação da existência de horários Livres nessa data com esses serviços



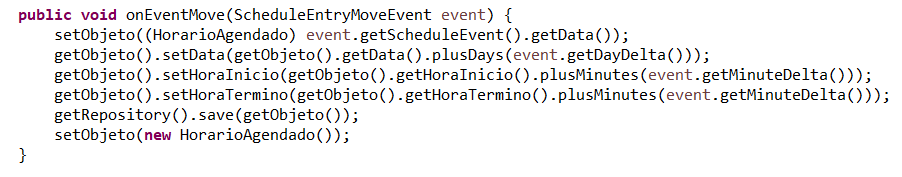
Primeiramente ao acessar a página index temos o método init que verifica se o usuário que esta logado tem persão e se não tiver o redireciona para o index do cliente, após isso setamos todas as variáveis necessárias e atualizamos o schedule.



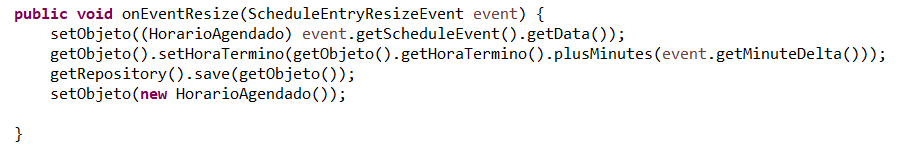
O evento dateSelect é quando clicamos em alguma data do nosso schedule e nesse evento ele abrirá o nosso modal para cadastro de um agendamento ou bloqueio. Por padrão ele abre o modal com o data que foi clicada no schedule selecionada no modal e tambem com o data e hora inicio do bloqueio como o dia selecionado e primeiro horário do dia selecionado, e como data e hora final de bloqueio o primeiro dia selecionado com o ultimo horário do dia selecionado. Pode Observar também que não obtemos a data que foi selecionada no schedule pelo selectEvent.



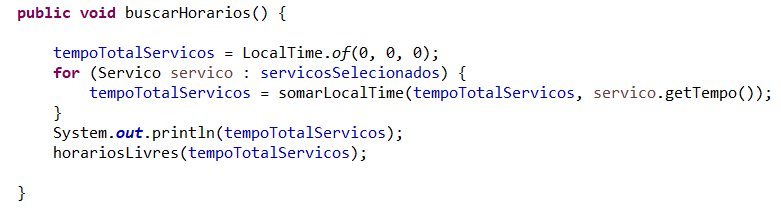
No evento onEventSelect ocorre quando selecionamos um horarioAgendado, Nós também pegamos os dados do objeto pelo selectEvent mas dessa vez pegamos com o getData, Além disso verifica se é bloqueio ou serviço pra mostrar a parte do modal correto.



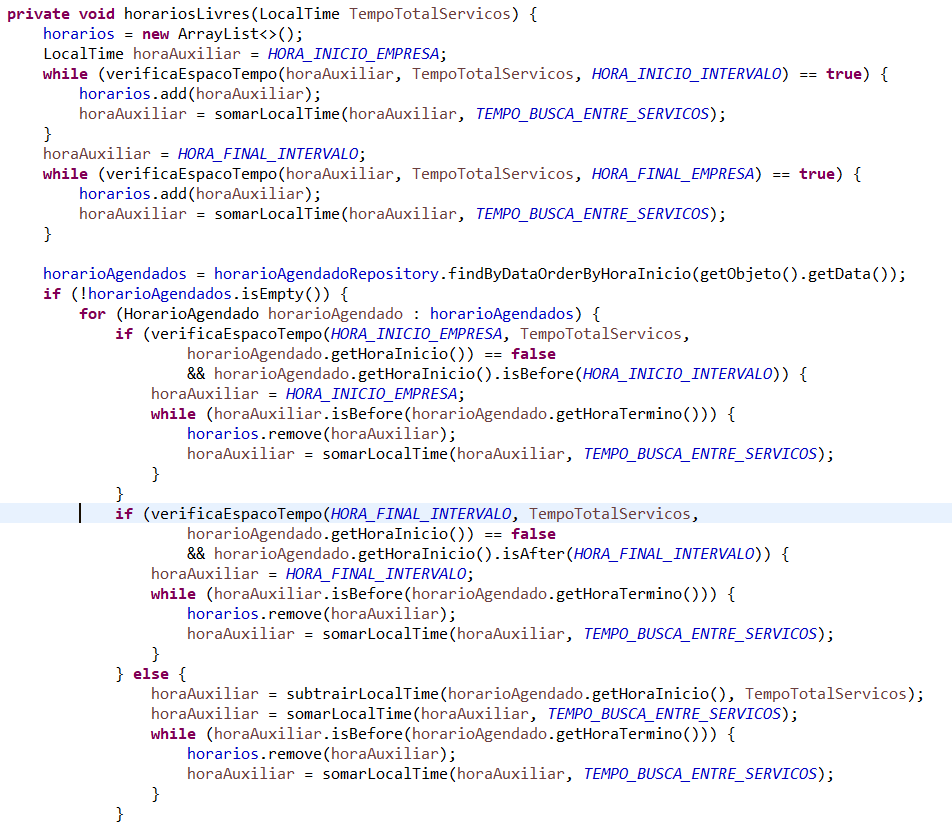
No onEventMove é quando o horarioAgendado é arrastado para outra data ou dia, basicamente após pegar objeto no event, eu faço uma soma da data, hora inicio e termino, com as alterações que são passadas pelo event, se um agendamento foi para uma data mais cedo que era antes, o event me trará um valor negativo, assim quando for feito a soma, como é um valor negativo, ele se transformará em uma subtração automaticamente.



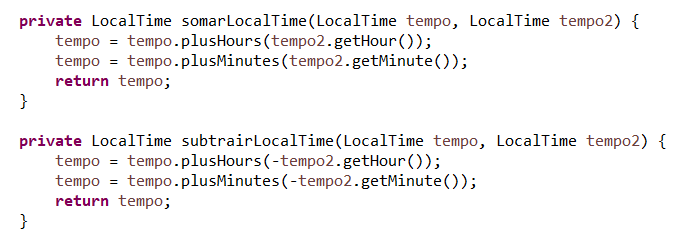
O onEventResize é quando é alterado o horário final de um agendamento, o código é igual do onEventMove a diferença é que como só estamos alterando a data final, só vamos alterar isso no código também.



Nesta primeira parte é feito somente a soma do tempo total dos serviços. E é envocado o método horários livres.



A função que busca os horários Livres funciona da seguinte maneira primeiramente ele vai colocar todos os horários possíveis na variável horários colocando baseado no tempo\_busca\_entre\_horários pulando também o horário de almoço da empresa, a partir disso se não tiver nenhum horário agendado nesta data a função termina, se tiver algum horário agendado ele verifica primeiramente se existe um tempo livre para cadastrar um novo horário entre o início da empresa até o horário já cadastrado, isso é feito tanto no horário de manhã quanto a tarde e por último ele utiliza a função de subtrair o local time para retirar os horários que não são possíveis de agendamento, como por exemplo, tendo como o tempo\_busca\_entre\_horários valendo 15 minutos e nós temos um horário agendado às 8 horas da manhã e queremos agendar um novo horário de manhã em que o serviço demore uma hora só será possível agendar às 7 horas da manhã ou posteriormente ao serviço já cadastrado, então ele irá pagar os horários das 7:15, 7:30, 7:45, 8 e até o serviço já cadastrado terminar.



Temos duas funções uma para somar e outra para subtrair o time