



Universidade Federal Rural do Rio de Janeiro
IM/Nova Iguaçu

Relatório de Computação 3

Grupo 5

Alunos: Rafael Kaustchr
Jones Marques
Elaine Tady
Racicley Costa
Thiago Damasceno

1.0 Minimundo

Imagine o contexto de um restaurante universitário onde alunos e funcionários comprem refeições diariamente. O sistema a ser implementado permitirá o controle de alunos, funcionários, departamentos e cursos assim como controlará as refeições disponíveis no restaurante e quais refeições foram consumidas pelos alunos e funcionários.

Um Consumidor apresenta:

- Um nome
- Uma matrícula (por exemplo: 23)
- Um ano de ingresso (por exemplo: 2014).
- Um sexo onde o Sexo pode assumir os valores MASCULINO e FEMININO
- Um título que pode assumir os valores ESPECIALIZACAO, MESTRADO e DOUTORADO.
- Um CPF apresenta dígitos verificadores que devem ser verificados durante a criação do professor.
- Não podem existir dois consumidores com o mesmo CPF.
- O sistema listará todos consumidores disponíveis até o momento apresentando o nome, a matrícula, o ano de ingresso, o sexo, o código do departamento (caso seja funcionário), o código do curso(caso seja aluno) e o CPF.
- O sistema permitirá alterar o nome, a matrícula, o sexo e o ano de ingresso de um consumidor.

O nome, a matrícula e o ano de um novo consumidor devem ser informados no momento de sua criação. Um Consumidor ou é um Aluno ou é um Funcionário. Um Aluno apresenta o Curso em que está inscrito e cada Funcionário apresenta o Departamento em que está lotado. O Curso e o Departamento de um novo Aluno e Funcionário, respectivamente, devem ser informados no momento da sua criação (caso não existam o Aluno ou o Funcionário não podem ser criados, isto é, o Curso e o departamento deve ser criados antes).

Um Departamento:

- Apresenta um nome (por exemplo: Departamento de ciência da computação) e uma sigla (por exemplo: DCC).
- Tem vários Funcionários lotados nele.
- Tem vários Cursos.
- Não podem existir dois departamentos com a mesma sigla.
- O sistema permitirá a criação de novos departamentos

Um Curso:

- Apresenta um nome (por exemplo: Ciência da computação) e uma sigla (por exemplo: CComp).
- Apresenta um Departamento onde, não existe curso sem o seu respectivo Departamento.
- Tem vários Alunos inscritos
- Não podem existir dois cursos com a mesma sigla.
- O sistema permitirá a criação de novos cursos.

Os preços que deverão ser pagos pelo Funcionário ou pelo Aluno estão associados ao Turno da refeição. Por exemplo: de Manhã o aluno pagar 0.5 enquanto o Funcionário paga 3.0

O Turno pode apresentar três valores:

- Manhã: Onde o preço do Funcionário é 3.0 e o preço do Aluno é 0.5
- Tarde: Onde o preço do Funcionário é 6.0 e o preço do Aluno é 1.0
- Noite: Onde o preço do Funcionário é 6.0 e o preço do Aluno é 1.0

Uma Refeição apresenta um Turno uma descrição e uma opção vegetariana. Um exemplo de refeição apresenta uma descrição “Feijão, arroz, bife e batata-frita”, um Turno tarde e uma opção vegetariana “Bolinho de arroz com macarrão integral e soja”

Uma Refeição ou é um Dejejum ou é um Almoço ou é um Jantar. Onde:

- Um Dejejum é sempre do turno Manhã.
- Um Almoço é sempre do Turno Tarde
- Um Jantar é sempre do Turno Noite.
- Após a criação da refeição o seu turno não pode ser alterado!
- O sistema permitirá a criação de novas refeições.
- O sistema listará todas as refeições disponíveis até o momento apresentando o turno, a descrição e a opção vegetariana.
- O sistema permitirá alterar a descrição e a opção vegetariana de qualquer refeição.

Um Ticket:

- Armazena a informação de que uma Refeição foi consumida por um Consumidor
- Isto é : Pertence a um Consumidor e Um consumidor tem vários tickets.
- Apresenta um valor (Por exemplo : 0.5, 10.0, etc ...), se foi pago ou não e a Refeição que foi consumida.
- O valor do ticket está associado com o Turno da refeição e qual é o consumidor (aluno ou funcionário). valor e a Refeição devem ser informados no momento da criação do ticket e não podem ser alterados.
- O sistema listará todos os tickets apresentando o a matrícula do consumidor, o valor, se foi pago,
- O turno e a descrição da refeição.
- O sistema permitirá alterar no ticket apenas a informação de pagamento (isto é, se ele foi pago ou não).

2.0 Lista de CRUD

Para o sistema foram levantados os seguintes CRUD's.

- Criar Aluno
- Criar Funcionário
- Criar Curso
- Criar Departamento
- Criar Refeição
- Criar Ticket
- Atualizar Consumidor
- Atualizar Curso
- Atualizar Departamento
- Atualizar Refeição
- Atualizar Ticket
- Ver Consumidor
- Ver Curso
- Ver Departamento
- Ver Refeição
- Ver Ticket

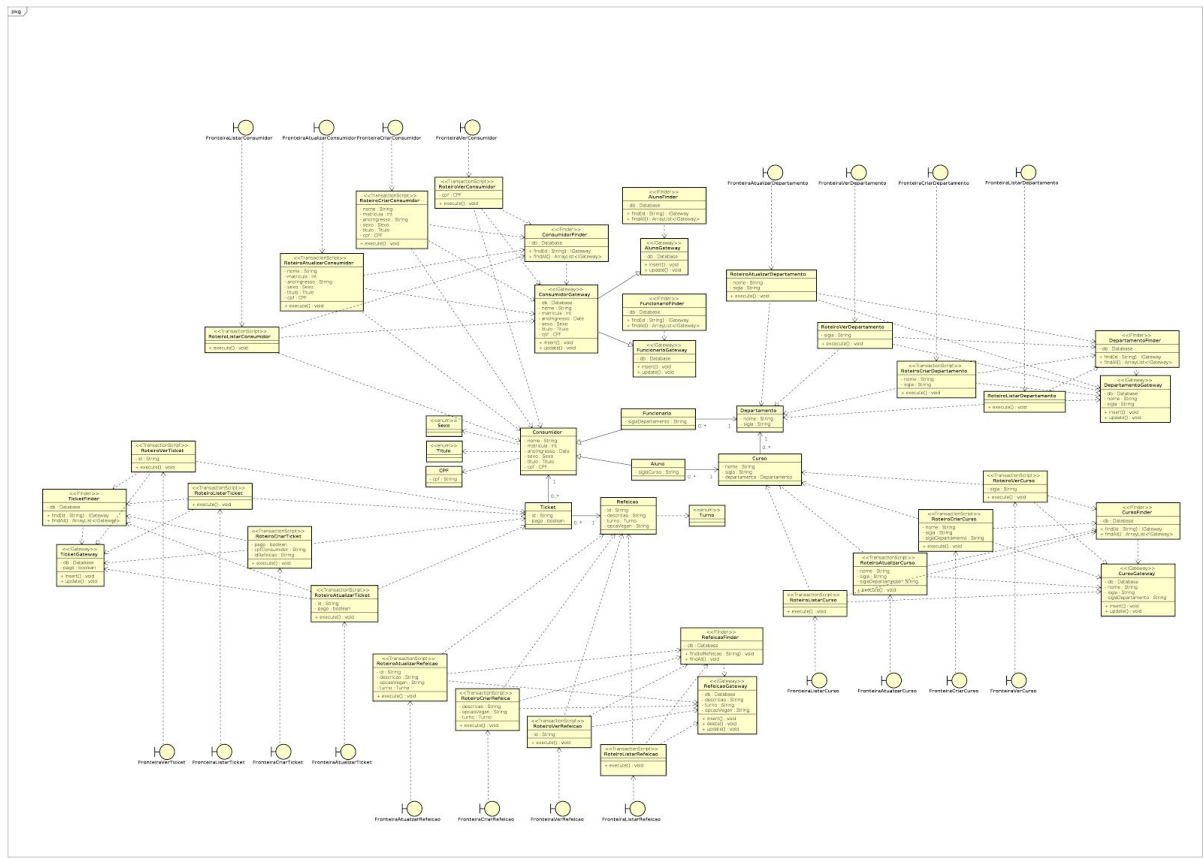
3.0 Requisitos Funcionais

O sistema possui os seguintes requisitos funcionais:

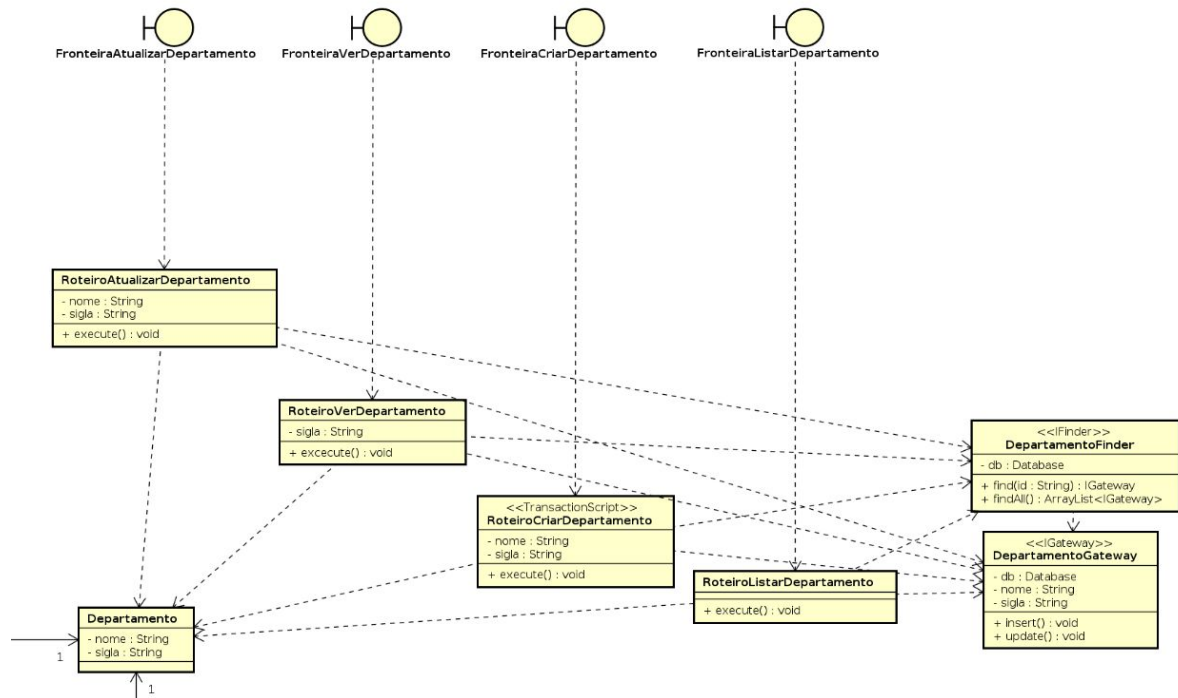
- RF01** - O sistema permitirá ao usuário criar um departamento.
- RF02** - O sistema permitirá ao usuário visualizar um departamento.
- RF03** - O sistema permitirá ao usuário alterar um departamento.
- RF04** - O sistema permitirá ao usuário listar os departamentos.
- RF05** - O sistema permitirá ao usuário criar um curso.
- RF06** - O sistema permitirá ao usuário visualizar um curso.
- RF07** - O sistema permitirá ao usuário alterar um curso.
- RF08** - O sistema permitirá ao usuário listar os cursos.
- RF09** - O sistema permitirá ao usuário criar um consumidor.
- RF10** - O sistema permitirá ao usuário visualizar um consumidor.
- RF11** - O sistema permitirá ao usuário alterar um consumidor.
- RF12** - O sistema permitirá ao usuário listar os consumidores.
- RF13** - O sistema permitirá ao usuário criar um ticket.
- RF14** - O sistema permitirá ao usuário visualizar um ticket.
- RF15** - O sistema permitirá ao usuário alterar um ticket.
- RF16** - O sistema permitirá ao usuário listar os ticket.
- RF13** - O sistema permitirá ao usuário criar uma refeição.
- RF14** - O sistema permitirá ao usuário visualizar uma refeição.
- RF15** - O sistema permitirá ao usuário alterar uma refeição.
- RF16** - O sistema permitirá ao usuário listar as refeições.

4.0 Diagrama de Classe

Para a construção do sistema foi utilizado como modelo da camada de domínio o Roteiro de Transações e como modelo da camada de dados o Portão de acesso aos dados em linhas.

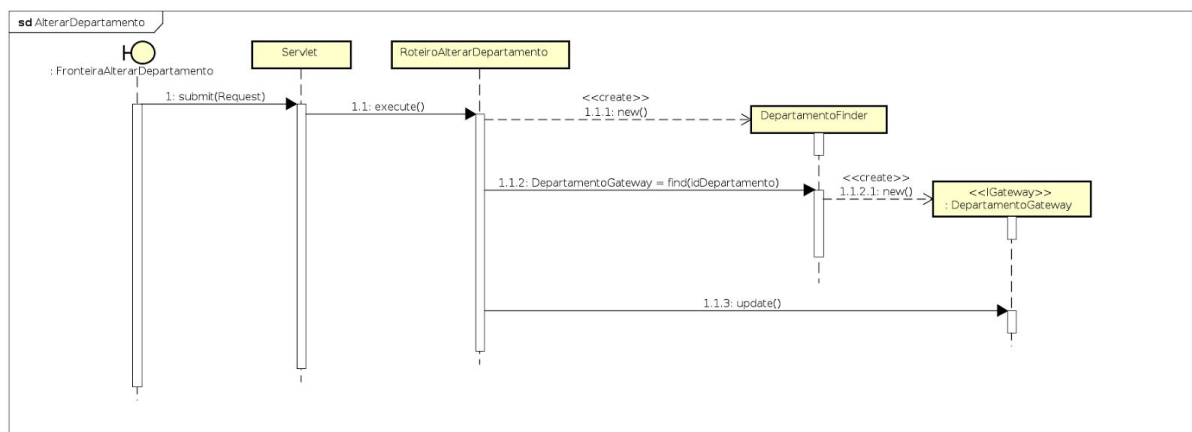


Aproximação da parte de departamento do diagrama de classes:

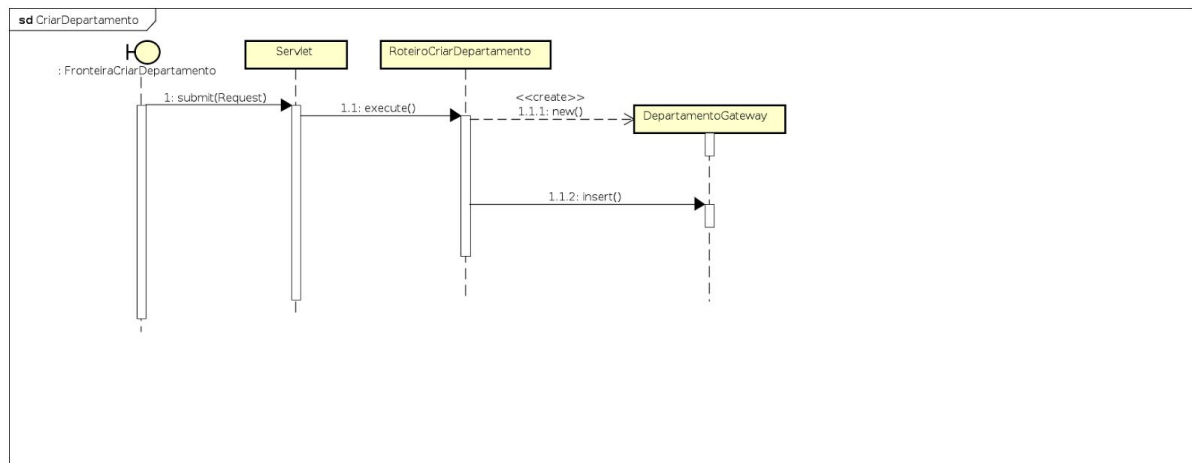


5.0 Diagramas de Sequência

5.1 Alterar Departamento

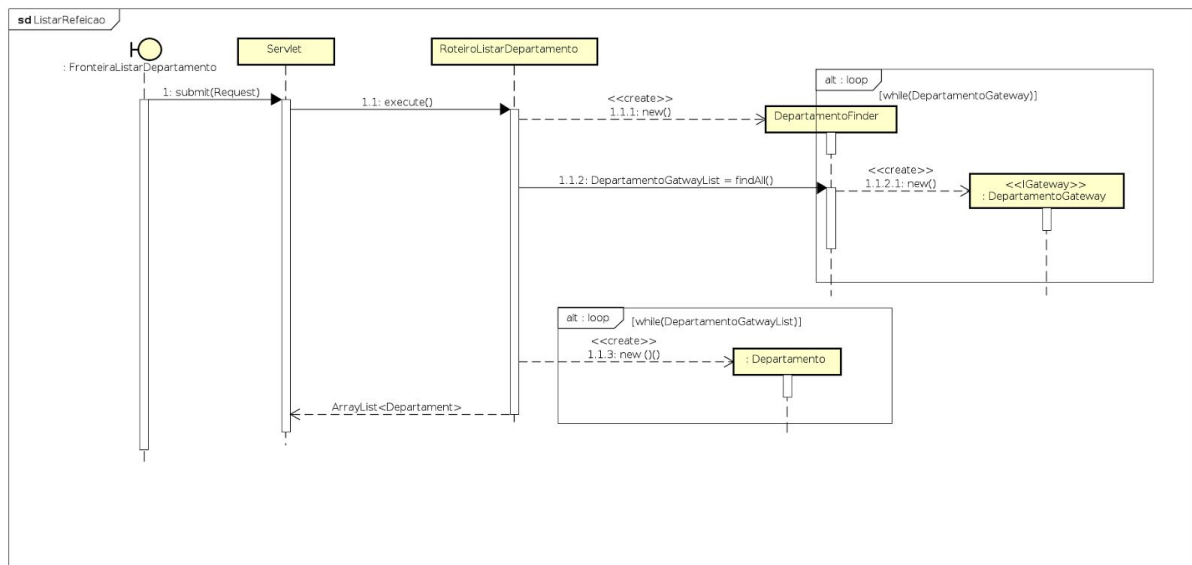


5.2 Criar Departamento



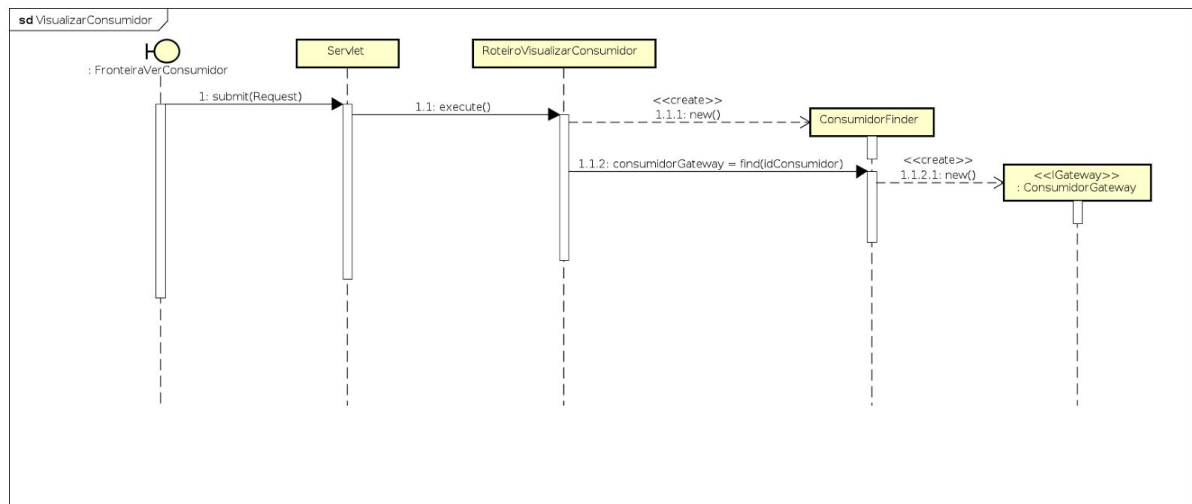
powered by Astah

5.3 Listar Departamento



powered by Astah

5.4 Visualizar Departamento



6.0 Testes

6.1 Testes Unitários

Entidade Aluno:

- testGetCurso()

Entidade Curso:

- testGetNome()
- testGetSigla()
- testGetDepartamento()
- testSetNome()

Entidade Departamento:

- testGetNome()
- testGetSigla()
- testSetNome()

Entidade Funcionario:

- testGetDepartamento()

Entidade Ticket:

- testGetRefeicao()
- testGetConsumidor()

Entidade Refeição:

- testGetDescricao()
- testSetDescricao()
- testGetTurno()
- testSetTurno()
- testGetOpcaoVegan()
- testSetOpcaoVegan()

6.2 Testes Funcionais

- TesteFuncionalAtualizarConsumidor
- TesteFuncionalAtualizarCurso
- TesteFuncionalAtualizarDepartamento
- TesteFuncionalAtualizarRefeicao
- TesteFuncionalAtualizarTicket
- TesteFuncionalCriarAluno
- TesteFuncionalCriarCurso
- TesteFuncionalCriarDepartamento
- TesteFuncionalCriarFuncionario
- TesteFuncionalCriarRefeicao
- TesteFuncionalCriarTicket
- TesteFuncionalSiglaExistenteCurso
- TesteFuncionalSiglaExistenteDepartamento
- TesteFuncionalVerCurso
- TesteFuncionalVerDepartamento
- TesteFuncionalVerFuncionario
- TesteFuncionalVerRefeicao
- TesteFuncionalVerTicket

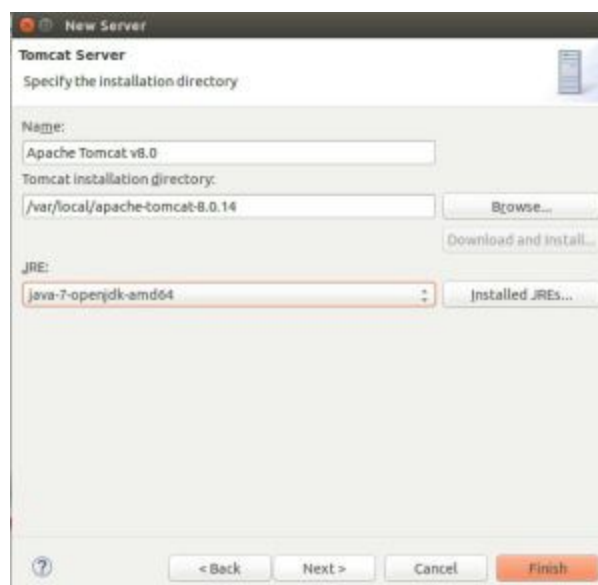
7.0 Tomcat

7.1 Configurando o Tomcat

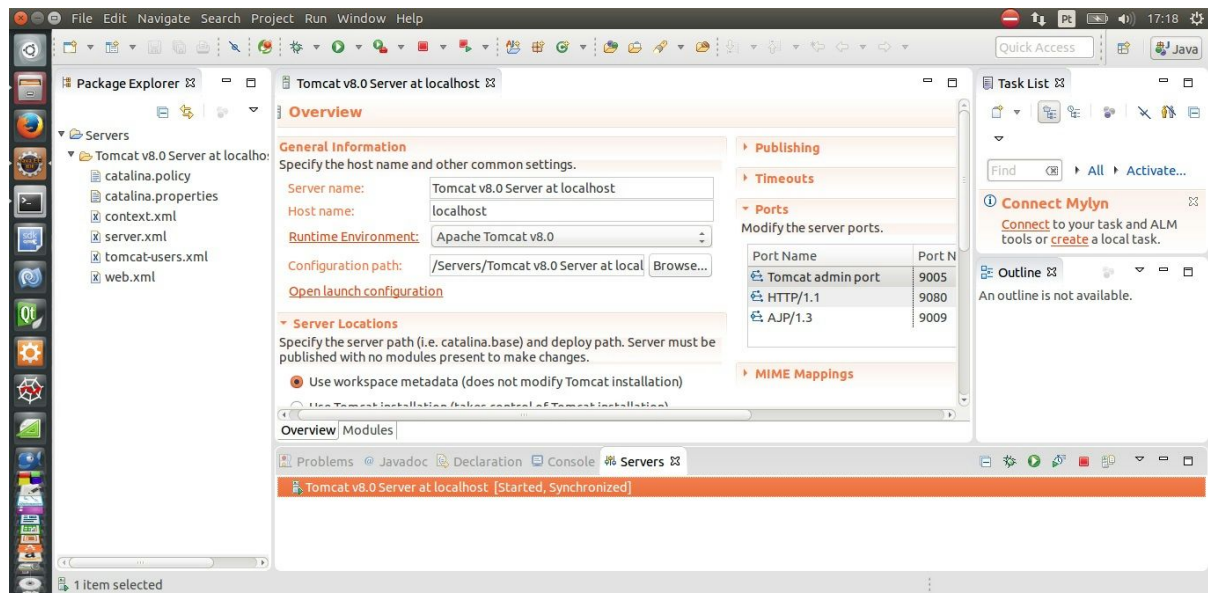
1 – Abra a IDE Eclipse e no menu Window, na opção Show View e escolha Servers. A janela de servidores aparecerá no canto inferior do Eclipse dizendo que não há nenhum servidor disponível, clique em novo servidor e expanda a pasta Apache e selecione a opção Apache Tomcat V8.0 Server, em seguida clique em Next:



2 – Na janela seguinte escolha a pasta onde se encontra a instalação do Tomcat, em seguida o JRE de sua máquina e clique em Finish:

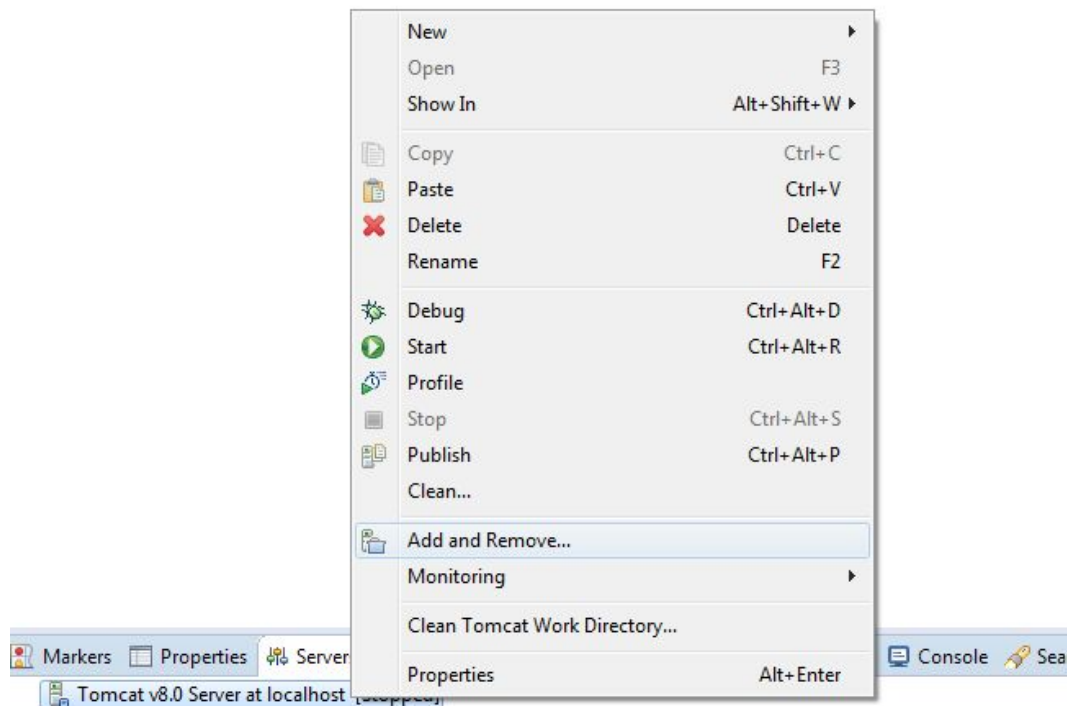


3 – Com o servidor já criado precisamos inicia-lo, porem o Tomcat roda por instancia e se a instancia principal do servidor já estiver rodando em seu computador encontraremos um conflito nas portas de comunicação do servidor, você pode parar (Stop) a instancia principal, ou modificar as portas da sua instancia dinâmica clicando duas vezes no seu servidor, na aba Servers, localizado no painel inferior do Eclipse, e mudar as portas para que não exista conflito entre instancias do Tomcat:

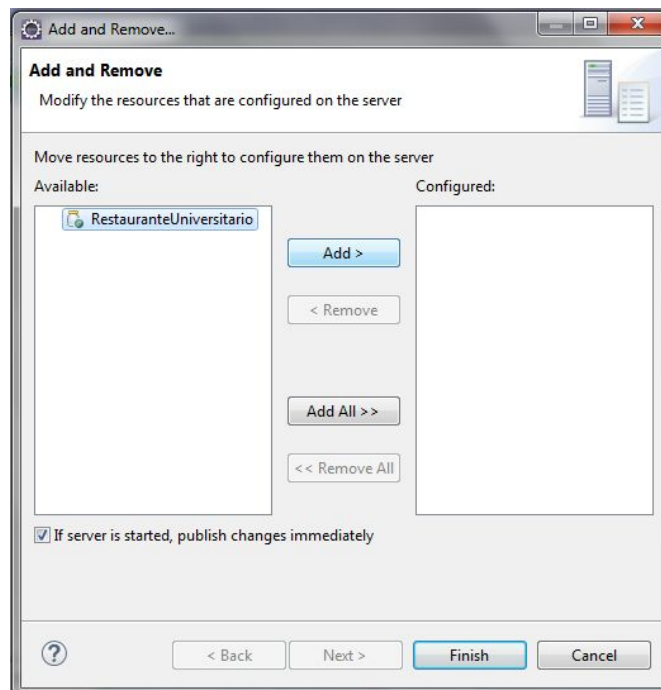


7.2 Adicionando o projeto no Tomcat

Para adicionar o projeto no Tomcat, na aba Server clique com o botão direito sobre o Tomcat e depois vá em “Add and Remove”.



Na janela que abrir, selecione o projeto e depois clique em “Add”.



Quando o projeto passar para o lado direito, é só ir em “Finish”.

