

**INSTITUTO INFNET**  
**ESCOLA SUPERIOR DE TECNOLOGIA DA INFORMAÇÃO**  
**GRADUAÇÃO EM ANÁLISE E DESENVOLVIMENTO**  
**DE SISTEMAS**



**Orientação a Objetos com UML e**  
**Modelagem de Dados - AT**

**ALUNO: MAGNO VALDETARO DE OLIVEIRA**  
**E-MAIL: [mvaldetaro@gmail.com](mailto:mvaldetaro@gmail.com)**  
**TURMA: NOITE - LIVE**  
**MATRÍCULA: 10403782775**

# Introdução

Este documento apresenta a modelagem de um sistema de gerenciamento e venda de canais TV a cabo que oferece operações simples para seus usuários. Detalhamos seus casos de uso, cenários e os diagramas relacionados.

## Justificativa

Este trabalho nos permite desenvolver o raciocínio para desenvolvimento de um sistema e aplicar os conhecimentos adquiridos sobre as visões de sistema, orientação a objeto e UML.

## Descrição

Documentação que apresenta a modelagem de um sistema de gerenciamento e venda de canais TV a cabo, capaz de oferecer algumas operações simples para seus usuários. Como exemplo, esses usuários podem ser os clientes (titular e dependentes), atendentes e administradores, têm acesso a ofertas e a lista de canais por categoria. A documentação será composta por elementos textuais e gráficos.

## Objetivo

- Compreender os conceitos de classes, atributos e métodos.
- Compreender os conceitos de instanciação e interação entre objetos.
- Compreender o conceito de encapsulamento.
- Compreender o conceito de herança.
- Compreender o conceito de polimorfismo.
- Compreender as visões de um sistema.
- Conhecer os principais diagramas da UML.
- Conhecer outros diagramas da UML
- Descrever casos de uso.
- Identificar e desenhar casos de uso.
- Identificar e desenhar atores.
- Identificar e desenhar relacionamentos.
- Compreender e criar classes
- Construir associações simples
- Identificar multiplicidades
- Construir classes associativas
- Compreender a diferença entre modelo de domínio e de especificação
- Especificar atributos e operações
- Especificar dependências
- Especificar agregações e composições

- Especificar generalizações
- Compreender e criar os principais elementos do diagrama
- Compreender e utilizar corretamente as interações com mensagens simples, síncronas, assíncronas e reflexivas
- Compreender os conceitos de estado e suas transições
- Construir diagramas com estados e transições simples entre eles
- Definir transições com eventos, parâmetros, guardas e ações
- Construir fluxogramas
- Construir diagramas para fluxos de controle sequenciais
- Construir diagramas para fluxos de controle paralelos
- Agrupar as classes do modelo de classes de especificação em pacotes
- Modelar a arquitetura do sistema em camadas
- Compreender como a abordagem relacional soluciona o problema de redundância dos dados
- Compreender como a abordagem relacional soluciona o problema do desperdício de espaço de armazenamento
- Compreender o que é entidade e relacionamento
- Compreender o que são chaves primária, estrangeira e secundária.
- Compreender o que são relacionamentos um para um, um para muitos e muitos para muitos.
- Construir um modelo conceitual do banco de dados
- Construir um modelo lógico do banco de dados
- Construir um modelo físico do banco de dados
- Compreender as diferenças entre os modelos de entidade relacionamento e o de classes da UML, bem como suas aplicações.

# Documento de Projeto de Sistema

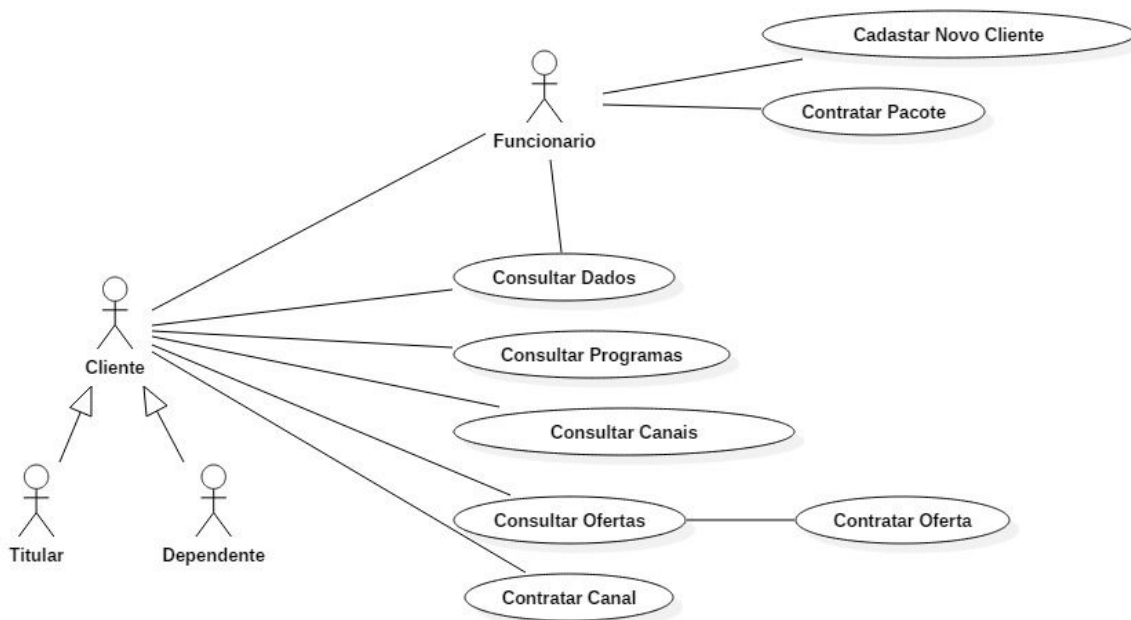
**Projeto:** Sistema de gerenciamento e venda de canais TV a cabo

**Versão:** 1.0

## 1. Casos de uso

### 1.1 Diagrama

**Definição:** Este diagrama descreve o cenário das funcionalidades do sistema de gerenciamento e venda do ponto de vista dos usuários.



### 1.2 Caso de Uso 1 - Consultar Programas

**Descrição:**

Este caso de uso permite que o cliente realize a consulta da programação de um canal específico.

**Atores:**

Cliente

**Pré-condições:**

Ter os dados no sistema, ter acesso a interface do cliente no sistema, ter um plano contratado, estar autenticado.

**Pós-condições:**

Visualizar a grade de programação.

### Fluxo de Eventos

Cenário Principal	
Atores	Sistema
1 - Cliente acessa a interface gráfica	2 - Exibe menu com as opções e ofertas
3 - Cliente seleciona o item de menu "Consultar programas de um canal"	4 - Solicita o número do canal
	5 - Aguarda resposta
6 - Cliente informa o número do canal	7 - Busca os programas do canal informado
	8 - Exibe opções: "Visualizar toda a programação" e "Definir intervalo"
9 - Cliente seleciona a opção "Visualizar toda a programação"	10 - Retorna programas

Cenários Alternativos	
Atores	Sistema
9 - Cliente seleciona a opção "Definir intervalo"	10 - Solicita "Horário Inicial" e "Horário Final"
11 - Cliente informa o intervalo	12 - Retorna programas dentro do intervalo de horário informado

### 1.3 Caso de Uso 2 - Consultar Oferta

#### Descrição:

Este caso de uso permite que o cliente consulte ofertas de canais e planos caso, as ofertas podem ou não existirem, cada oferta tem sua descrição e preço para contratação.

#### Atores:

Cliente

#### Pré-condições:

Ter os dados no sistema, ter acesso a interface do cliente no sistema, estar autenticado.

#### Pós-condições:

Visualizar Ofertas.

## Fluxo de Eventos

Cenário Principal	
Atores	Sistema
1 - Cliente acessa a interface gráfica	2 - Exibe menu com as opções e ofertas
3 - Cliente seleciona uma oferta	4 - Exibe a descrição, valor da oferta, opção de “Contratar” e “Retornar”
5 - Cliente seleciona a opção “Contratar”	6 - Inicia o caso de uso “Contratar Oferta”

Cenários Alternativos	
Atores	Sistema
5 - Cliente seleciona a opção “Retornar”	6 - Retorna para tela inicial

### 1.3 Caso de Uso 3 - Contratar Canal

#### Descrição:

Este caso de uso permite que o cliente contrate um canal adicional para o seu plano.

#### Atores:

Cliente

#### Pré-condições:

Ter os dados no sistema, ter acesso a interface do cliente no sistema, estar autenticado.

#### Pós-condições:

Novo canal contratado e adicionado ao pacote do cliente.

## Fluxo de Eventos

Cenário Principal	
Atores	Sistema
1 - Cliente acessa a interface gráfica	2 - Exibe menu com as opções e ofertas
3 - Cliente seleciona a opção “Adquirir canal”	4 - Exibe lista de categorias de canais
5 - Cliente seleciona a opção a categoria	5 - Exibe lista de canais da categoria selecionada

6 - Cliente seleciona o canal	7 - Exibe informações sobre o canal e as opções “Contratar”, “Retornar”, “Categorias” e “Retorna para tela inicial”
8 - Cliente seleciona a opção “Contratar”	
	9 - Exibe mensagem sobre a forma de pagamento.
	10 - Solicita confirmação
11 - Confirma a contratação.	12 - Informa que a contratação foi bem sucedida
	13 - Envia informações para o sistema de cobrança
	13 - Retorna para lista de canais

Cenários Alternativos	
Atores	Sistema
8 - Cliente seleciona a opção “Retornar”	9 - Retorna para lista de canais
8 - Cliente seleciona a opção “Categorias”	9 - Retorna para lista de categorias de canais
8 - Cliente seleciona a opção “Retorna para tela inicial”	9 - Retorna para tela inicial do sistema

## 1.4 Caso de Uso 4 - Cadastrar Novo Cliente

### Descrição:

Este caso de uso permite que um funcionário cadastre um novo cliente no sistema.

### Atores:

Cliente, Funcionário

### Pré-condições:

Funcionário deve estar autenticado no sistema.

### Pós-condições:

Um novo cliente cadastrado no sistema.

### Fluxo de Eventos

<b>Cenário Principal</b>
--------------------------

<b>Atores</b>	<b>Sistema</b>
1 - Cliente entra em contato com o Funcionário	
2 - Funcionário acessa o sistema	3 - Exibe menu com as opções “Cadastrar cliente” e “Contratar pacote”
4 - Funcionário seleciona opção “Cadastrar cliente”	5 - Solicita as informações do cliente
6 - Funcionário solicita os dados necessários para o cadastro ao Cliente	
7 - Cliente informa os dados ao Funcionário	
8 - Funcionário informa os dados do Cliente	9 - Solicita confirmação
10 - Funcionário confirma	11 - Retorna que um novo cliente foi cadastrado
12 - Funcionário informa ao Cliente que o cadastro foi efetuado	

<b>Cenários Alternativos</b>	
<b>Atores</b>	<b>Sistema</b>
7 - Cliente encerra o contato	
8 - Funcionário suspende o cadastro	9 - Retorna para tela inicial do sistema
	9 - Informa que CPF é inválido
10 - Funcionário confirma informação com Cliente	
11 - Cliente corrige informação	
12 - Funcionário informa novamente o CPF do Cliente	13 - Solicita confirmação

## 2. Classes

As classes que identificamos neste sistema são: Usuário, Endereço, Cliente, Titular, Dependente, Funcionário, Fatura, Menu, Venda, Plano, Canal, Programa, Grade, Autenticar.

A seguir a descrição das interfaces e interações dos objetos de algumas das classes identificadas acima .



## 2.1 Classe - Usuário

<b>Características</b>	nome, sobrenome, cpf, genero, dtNascimento, telefone
<b>Comportamentos</b>	Não há comportamento para esta classe.
<b>Descrição</b>	<p>Esta classe encapsula como protegidas todas as principais características dos usuários do sistema, sendo estes Clientes (Titulares ou Dependentes) e Funcionários.</p> <p>Se relaciona por composição com as características da classe Endereco.</p>

## 2.2 Classe - Cliente

<b>Características</b>	numeroContrato, situacaoContrato, plano
<b>Comportamentos</b>	acessarSistema()
<b>Descrição</b>	<p>Esta classe encapsula como privado todas as características do cliente no sistema.</p> <p>Herda as características da classe Usuário.</p> <p>O comportamento acessarSistema() carrega a interface do sistema e interage com um objeto da classe Menu exibindo o menu principal e as ofertas. Também interage com a classe Autenticar para verificar se o cliente existe ou não no sistema</p> <p>O comportamento acessarSistema() não tem retorno, no entanto aguarda o retorno do comportamento do objeto de Autenticar para exibir o menu principal e as ofertas. Se o comportamento do objeto da classe Autenticar retornar "true" menu e ofertas são exibidos, caso retorne "false" nada acontece.</p>

## 2.3 Classe - Grade

<b>Características</b>	canal, programa, programacao
<b>Comportamentos</b>	setGradeCanal(), getGradeCanal(), programacao()
<b>Descrição</b>	<p>Esta é uma classe Associativa, suas características são encapsuladas de forma privada.</p> <p>O comportamento setGradeCanal() define um objeto do tipo grade com as características canal do tipo Canal e programa do tipo Programa.</p>

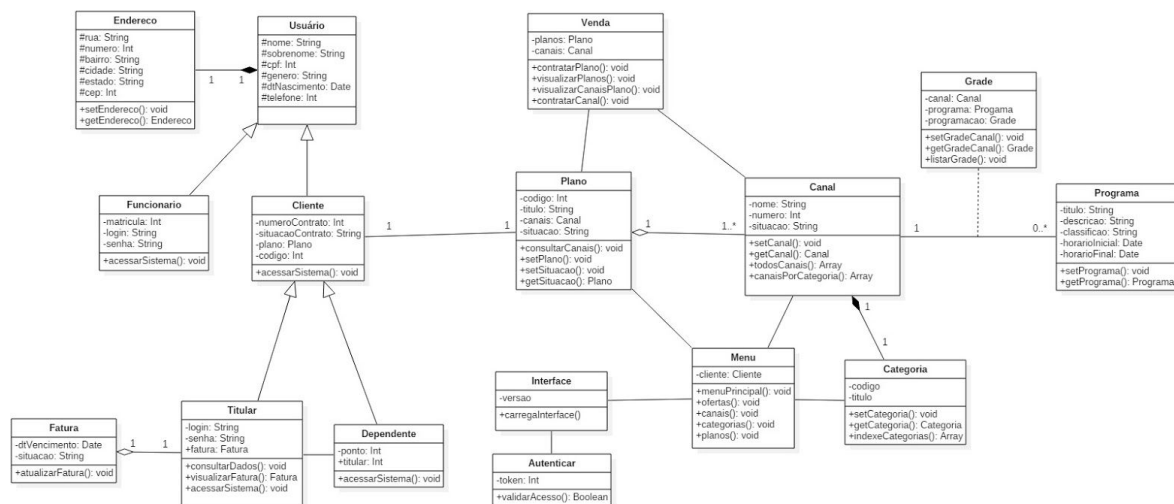
	<p>getGradeCanal() retorna a grade de um canal da característica programacao do Tipo Grade.</p> <p>listarGrade() lista todos os objetos do tipo Grade retornados da característica programacao</p>
--	--

## 2.4 Classe - Canal

<b>Características</b>	canal, numero, categoria, situacao
<b>Comportamentos</b>	setCanal(), getCanal()
<b>Descrição</b>	<p>Esta é uma classe agrega informações a classe Plano, sua características são encapsuladas de forma privada.</p> <p>O comportamento setCanal() por parâmetros define um objeto Canal com todas as suas características.</p> <p>O comportamento getCanal() por parâmetro associado ao canal retorna um objeto Canal.</p>

## 2.5 Diagrama de classe de especificação

**Definição:** este diagrama define as visão no sistema das classes assim com seus objetos compostos por características e comportamentos.



## 3. Diagramas de Sequência

Este diagrama define as sequências das trocas de mensagens entre objetos no sistema.

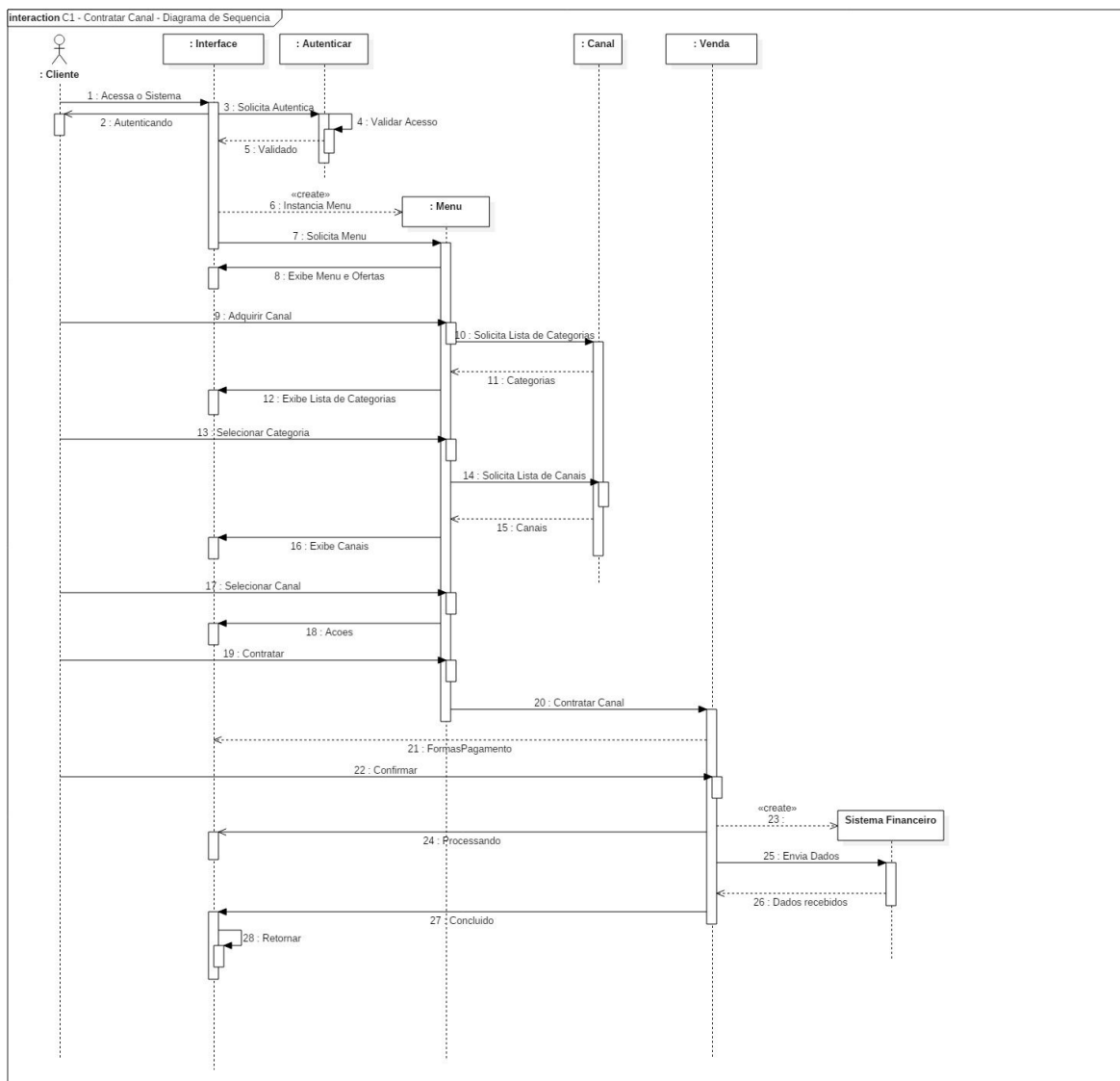
### 3.1 Diagrama de Sequência - Contratar Canal

**Nota:** entendo que o objetos devem ser apresentados com um sublinhado, mas não encontrei na ferramenta starUML está opção. Sendo assim descrevo a seguir quais os objetos e classes.

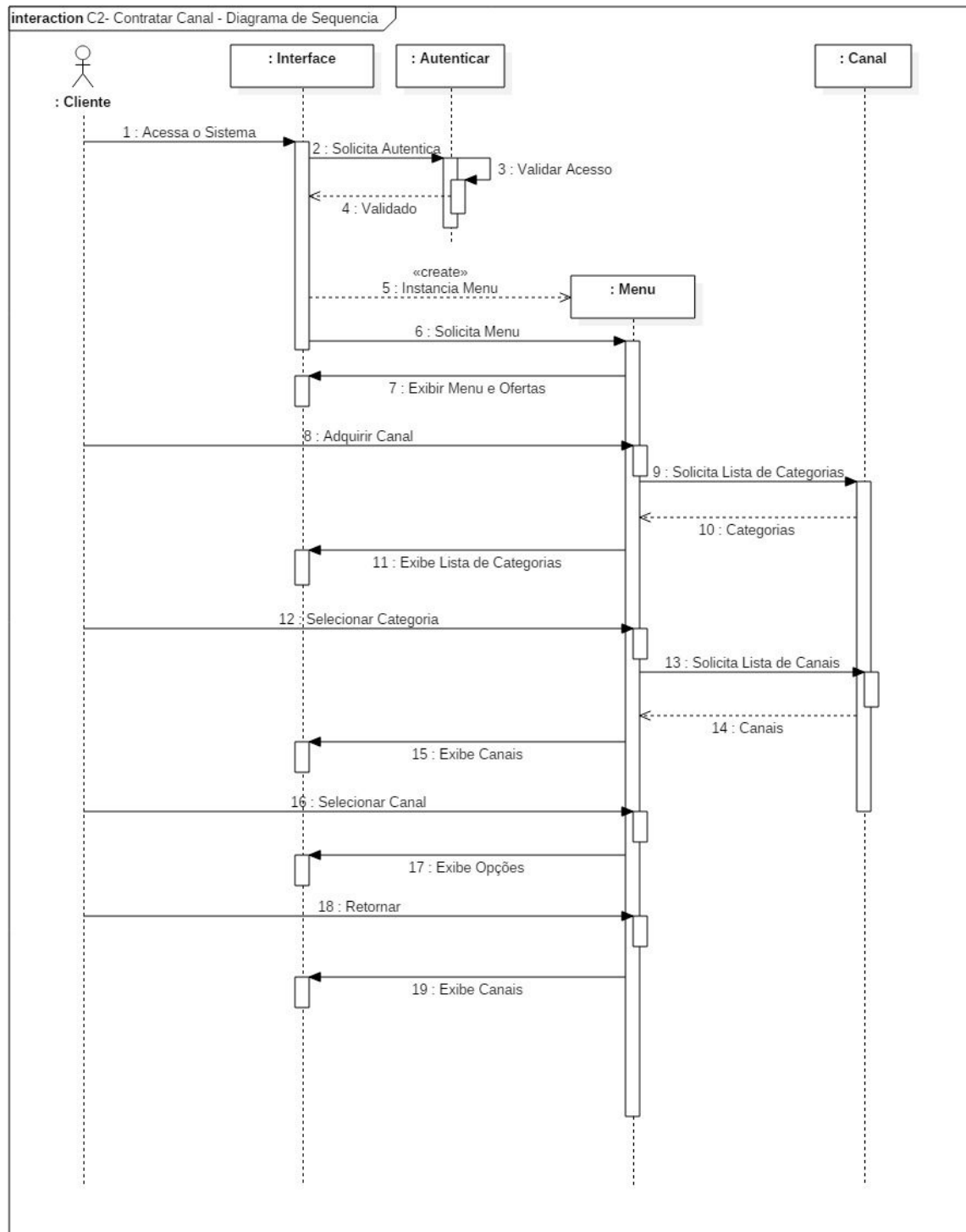
Interface é classe.

Todas as outras representações de linha de tempo são de objetos.

#### Cenário Principal



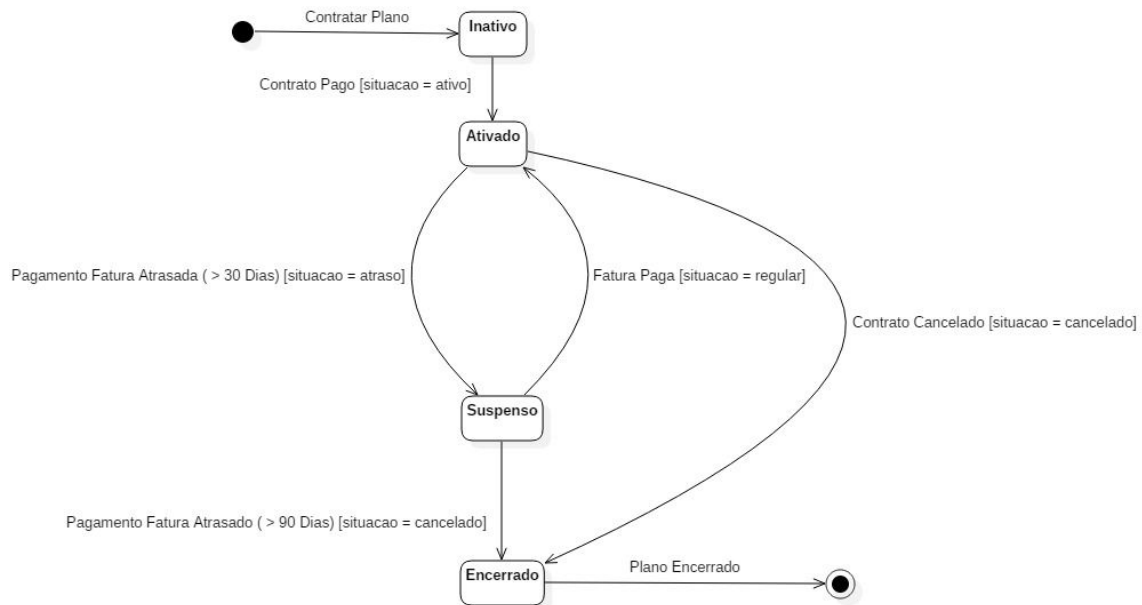
## Cenário Alternativo



## 4. Diagrama de Estado

**Definição:** o diagrama de estado representa os estados e eventos de um objeto no sistema.

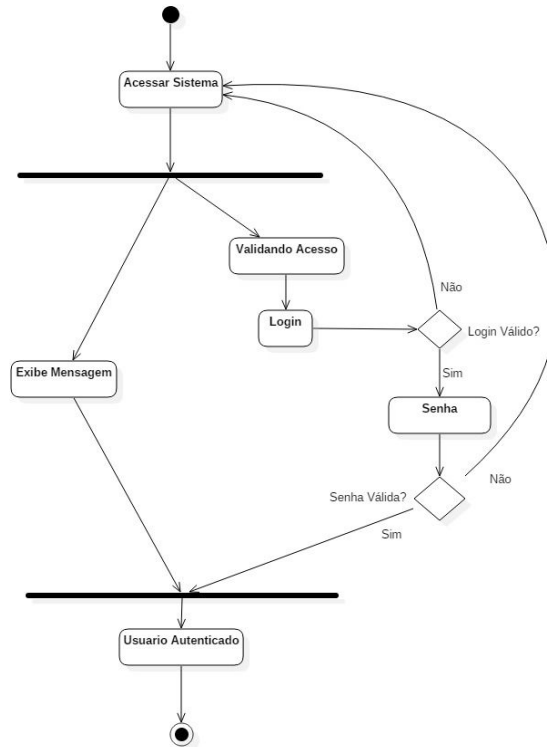
### Representação de um objeto do tipo Plano



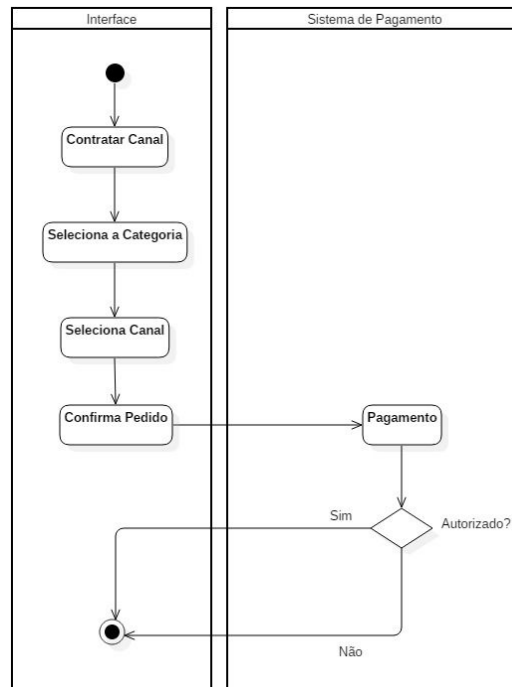
## 5. Diagrama de Atividades

**Definição:** o diagrama de Atividades demonstra o fluxo de controle uma atividade para outra.

### 5.1 Atividade - Acessar Sistema



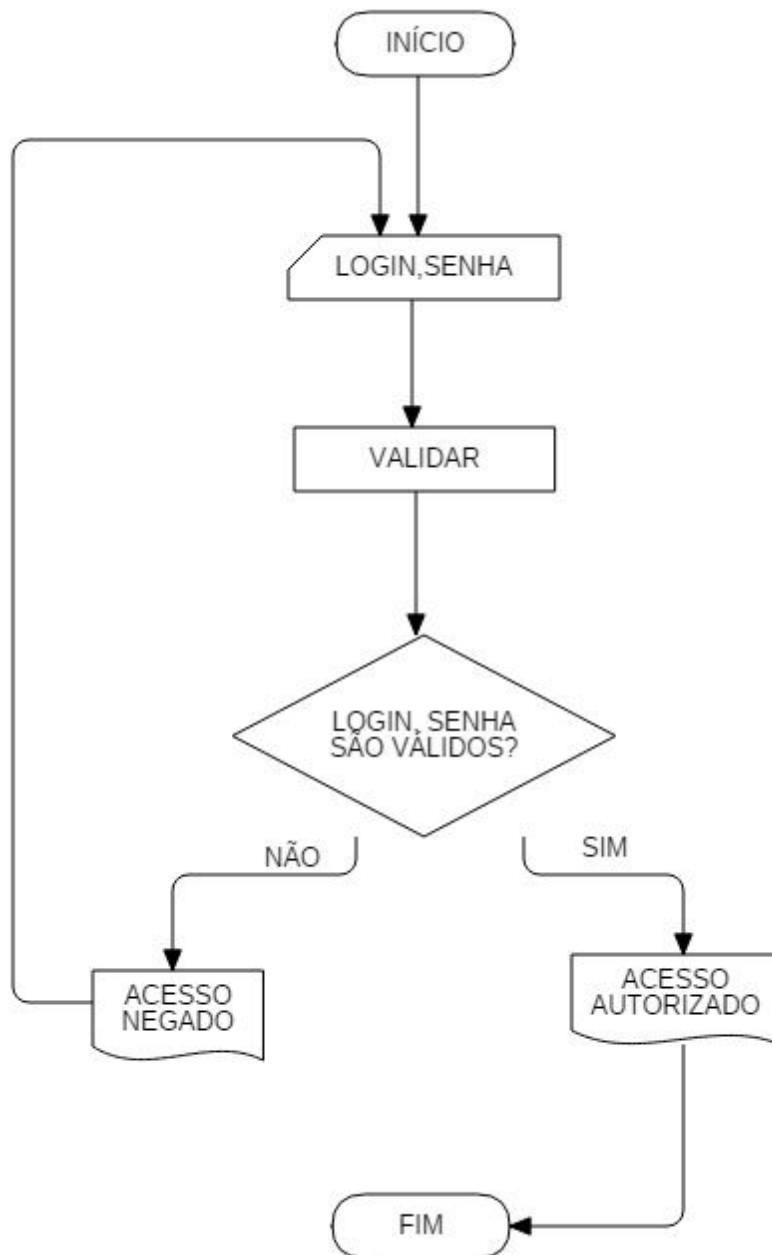
### 5.1 Atividade - Contratar Canal



## 6. Fluxogramas

**Definição:** este diagrama representa uma sequência de processos permitindo uma melhor visualização da atividade.

### 6.1 Fluxograma - Acessar Sistema

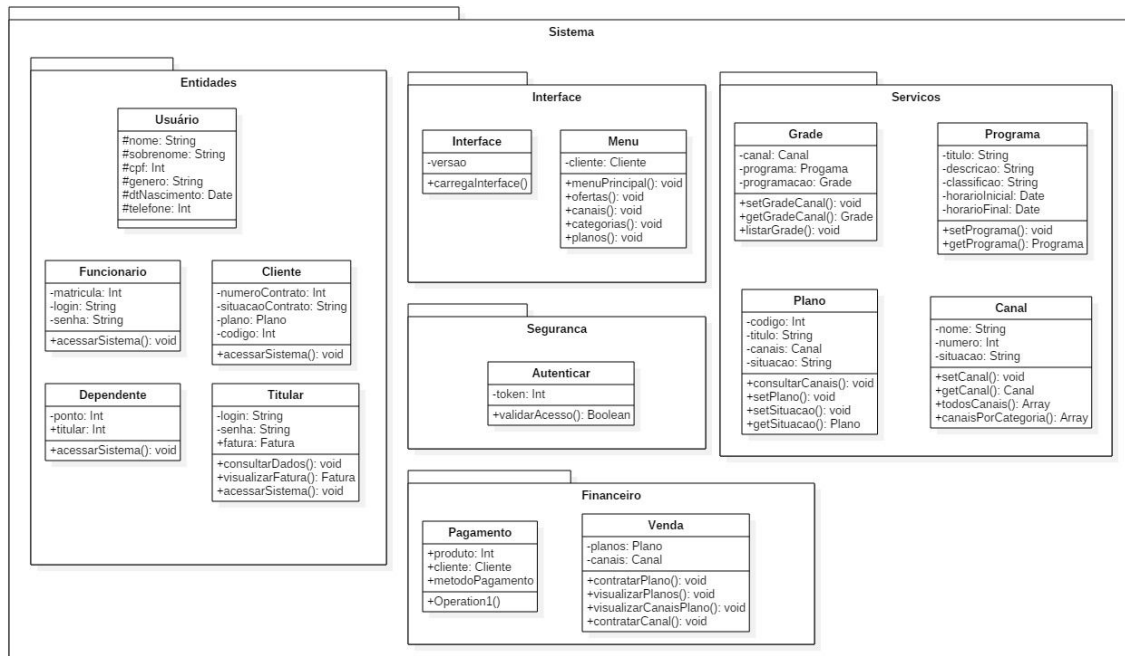


## 7. Diagrama de Pacotes

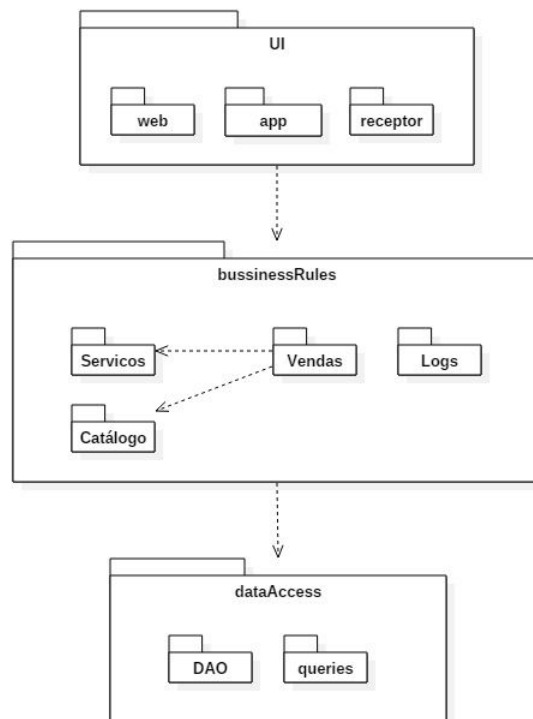
**Definição:** o diagrama de pacotes ilustra um conjunto de elementos agrupados, estes elementos podem ser pacotes, diagramas ou classes.

### 7.1 Pacotes de Classes





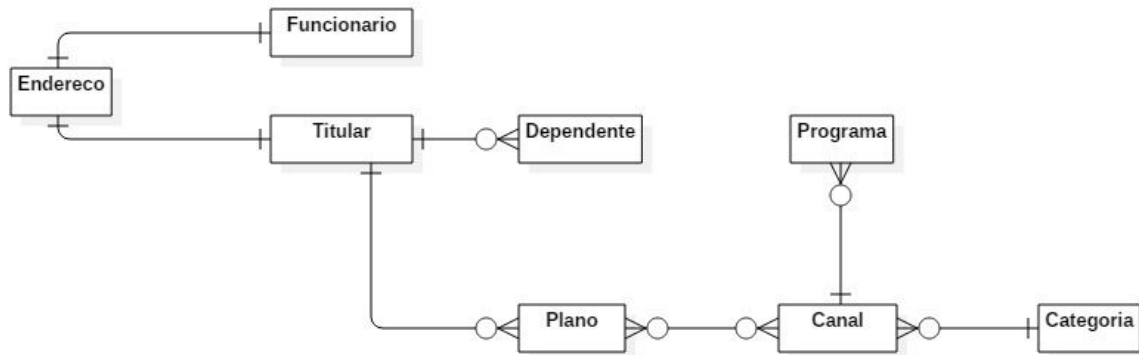
## 7.2 Pacotes em Camadas



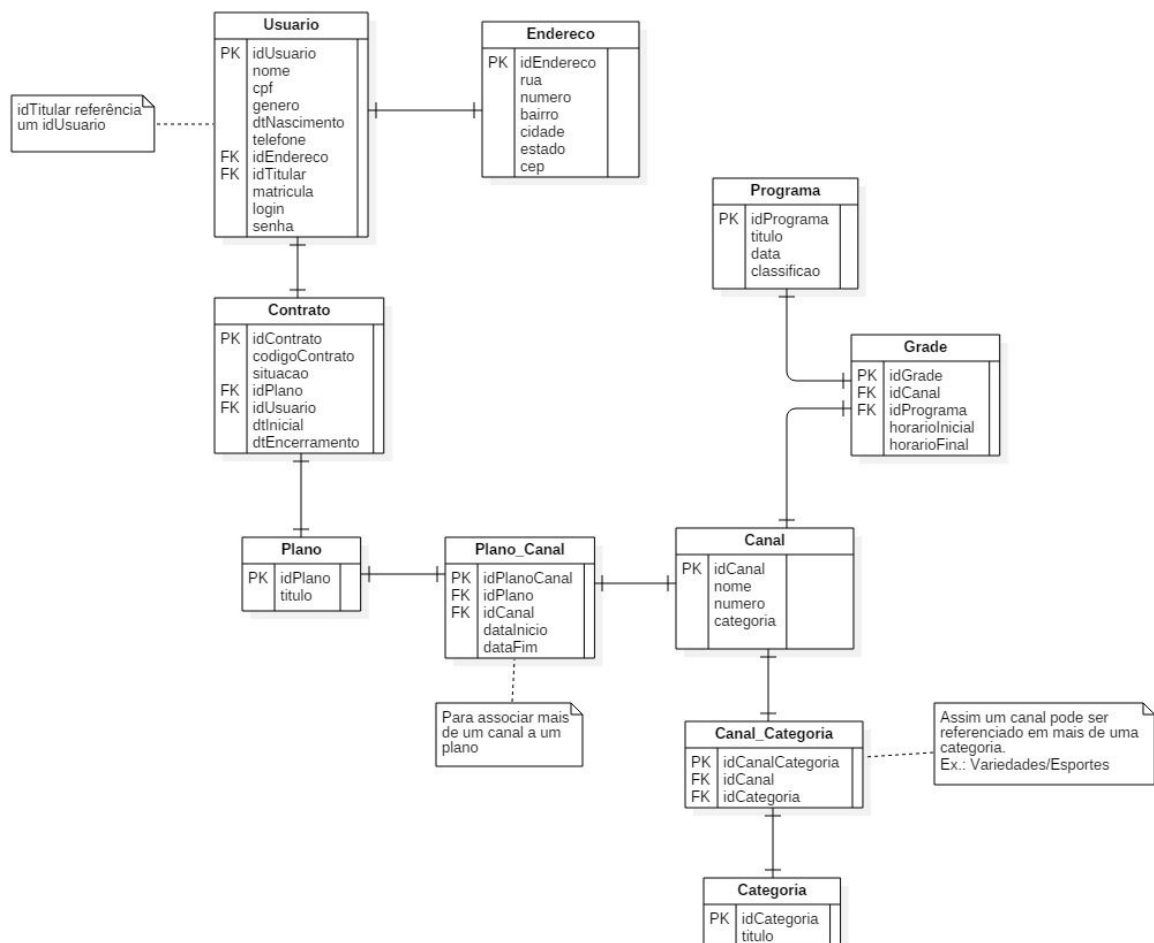
## 8. Modelo de Entidade x Relacionamento

**Definição:** este diagrama ilustra as entidades, atributos e relacionamentos.

### 8.1 Modelo Conceitual



## 8.2 Modelo Lógico



## 8.3 Projeto Físico

O código de criação de tabelas localiza-se em “Assessment > projeto-fisico-dados.sql”.

## 9. Mapeamento objeto relacional

Objetos do tipo “Endereco” se relacionam com a tabela “Endereco” seus valores de atributos estão diretamente relacionados aos campos da entidade. Pode-se persistir dados e requisitá-los por meios dos métodos deste objeto.

Objetos dos tipos “Funcionario”, “Cliente”, “Titular” e “Dependente”, possuem os principais valores de seus atributos associados com a tabela usuário. Pode-se persistir dados e requisitá-los por meios dos métodos deste objeto. Por sua vez objetos do tipo “Cliente” tem seus atributos associados a tabela “Contrato”.

Objetos do tipo “Plano” se relacionam com as tabelas “Plano” e “Plano\_Canal”. Persistindo dados em plano e requisitando dados de “Plano\_Canal”.

Objetos do tipo “Canal” se relacionam com as tabelas “Canal”. Pode-se persistir dados e requisitá-los por meios dos métodos deste objeto.

Objetos do tipo “Categoria” se relacionam com as tabelas “Categoria”. Pode-se persistir dados e requisitá-los por meios dos métodos deste objeto.

Objetos do tipo “Grade” se relacionam com as tabelas “Grade”, “Programa” e “Canal”, seus métodos não persistem dados.

## Visões x Diagramas

Foram desenvolvidas as seguintes visões do sistema:

- visão de casos de uso;
- visão lógica;
- visão de desenvolvimento;
- visão de processos.

<b>Visão de Casos de Uso</b>	Nesta visão utilizamos o diagrama de casos de uso e diagrama de sequência
<b>Visão Lógica</b>	Nesta visão utilizamos o diagrama de classes, diagrama de estado e diagrama de pacotes
<b>Visão de Processos</b>	Nesta visão utilizamos o diagrama de atividades e diagrama de estado
<b>Visão de Desenvolvimento</b>	Nesta visão utilizamos o diagrama de pacotes

Outros diagramas que poderiam ter sido aplicados: diagrama de interação, diagrama de objetos, diagrama de componentes.

# Conclusão

Nesta tarefa foi desenvolvida a documentação de um sistema de gerenciamento e venda de canais TV a cabo, aplicando os diagramas da UML apresentados ao decorrer da disciplina.

# Referências

Roteiros de Aprendizagem de 1 a 9 e os recursos nestes disponíveis  
Disponível em: <http://lms.infnet.edu.br/moodle/course/view.php?id=594>  
Acesso em: 22 de outubro 2016.

As 4+ 1 Visões  
Disponível em: <https://www.dimap.ufrn.br/~jair/as/slides/Visoes4+1.pdf>  
Acesso em: 22 de outubro 2016.