

Sistema de Faturamento e Cobrança em Telecomunicações com Tarifários Dinâmicos – Altice Labs

Realizado por:

Rui Pedro Raposo Moniz

Índice

• Classes.....	3
• Fluxo de Funcionamento da Aplicação.....	5
• Manual de Instruções.....	6
• Detalhes de Implementação.....	6
• Relatório de Testes.....	7

Classes

Todas as classes possuem os atributos com acesso privado. Consequentemente, de modo a respeitar o princípio do encapsulamento, possuem os métodos necessários para aceder e modificar estes mesmos atributos, assim como funções que imprimem o conteúdo dos objetos (getters, setters , toString()) Para além disso estas possuem métodos adicionais que permitem realizar as operações necessárias para o funcionamento do sistema.

A Classe Tariff é uma super-classe abstrata que representa um tarifário, existem 6 sub-classes referentes à mesma sendo uma para cada tipo de tarifário (Alpha1Tariff, Alpha2Tariff, Alpha3Tariff, Beta1Tariff, Beta2Tariff, Beta3Tariff).

Tabela 1 - Registos de pedidos de carregamento juntamente com a sua resposta.

CDR	
Atributo	Descrição
timeStamp	Momento da requisição (timestamp)
msisdn	Msisdn do conta que realizou o request (string).
serviceld	Serviço solicitude (A ou B) (char).
chargingRequest	Requisição realizada (ChargingRequest)
chargingReply	Resposta à requisição realizada (ChargingReply)
bucket1Value	Valor do Bucket1 após a transação (float)
bucket2Value	Valor do Bucket2 após a transação (float)
bucket3Value	Valor do Bucket3 após a transação (float)
counterAValue	Valor do CounterA após a transação (float)
counterBValue	Valor do CounterB após a transação (float)
counterCValue	Valor do CounterC após a transação (float)
counterDValue	Valor do CounterD após a transação (timestamp)
appliedTariffId	Tarifário aplicado (Alfa1,Alfa2,Alfa3,Beta1,Beta2,Beta3)(string)

Tabela 2- Pedidos de carregamento.

ChargingRequest	
Atributo	Descrição
requestId	ID da requisição (int).
timeStamp	Momento da requisição (timestamp).
service	Serviço solicitude (A ou B) (char).
roaming	Requisição é por roaming (boolean) (true – roaming false – local)
msisdn	Msisdn do conta que realizou o request (string).
rsu	Unidades requisitadas do serviço(int)

Tabela 3 - Resposta aos pedidos de carregamento

ChargingReply	
Atributo	Descrição
requestId	ID da respetiva requisição (int).
result	Resultado da requisição (string).
gsu	Unidades que foram garantidas pela transação.

Tabela 4 - Contas registradas no sistema.

BillingAccounts	
Atributo	Descrição
msisdn	Msisdn da conta (string).
bucket1	Valor do Bucket1(float)
bucket2	Valor do Bucket2(float)
bucket3	Valor do Bucket3(float)
counterA	Valor do CounterA(float)
counterB	Valor do CounterB (float)
counterC	Valor do CounterC (float)
counterD	Valor do CounterD (float)
tariffServiceA	Tarifário do Serviço A.
tariffServiceB	Tarifário do Serviço B.

Tabela 5 - Interface do sistema de faturação.

SystemInterface	
Atributo	Descrição
billingAccounts	Contas registradas no sistema (ArrayList de BillingAccounts).
cdrs	CDR's registrados no sistema (ArrayList de CDR).
chargingRequests	Pedidos registrados no sistema (ArrayList de ChargingRequest).
chargingReplies	Respostas registradas no sistema (ArrayList de ChargingReply).
Método	Descrição
initializeData()	Inicializa a lista de contas (billingAccounts) com dados de teste.
getRequestIdMax()	Retorna o id do próxima requisição feita.
processRequest()	Método que processo uma requisição.
findBillingAccount()	Encontra uma conta na lista de contas com base no MSISDN.
findBillingAccountCDRs()	Encontra todos os CDR's referentes a um MSISDN.

Para além disso esta classe também possui métodos de inserção de novos elementos nas suas listas (ex: addBillingAccount) e de imprimir o conteúdo das mesmas (ex: printBillingAccounts).

Tabela 6 - Tarifário.

Tariff	
Método	Descrição
processRequest()	Método que processo um pedido e fornece a resposta ao mesmo.
verifyEligibility()	Método abstrato que verifica a elegibilidade do pedido.
calculateRating()	Método abstrato que calcula o custo por unidade do pedido.
doCharging()	Método abstrato faz a cobrança do valor correto.

Esta classe possui métodos abstratos que se comportam de maneira diferente conforme a sua sub-classe, de modo a respeitar o princípio da abstração.

A classe **Main()** é responsável por instanciar a interface do sistema e possui um consola para o utilizador comunicar com o sistema fazendo pedidos através de comandos (Ver Manual de Instruções).

Fluxo de Funcionamento da Aplicação

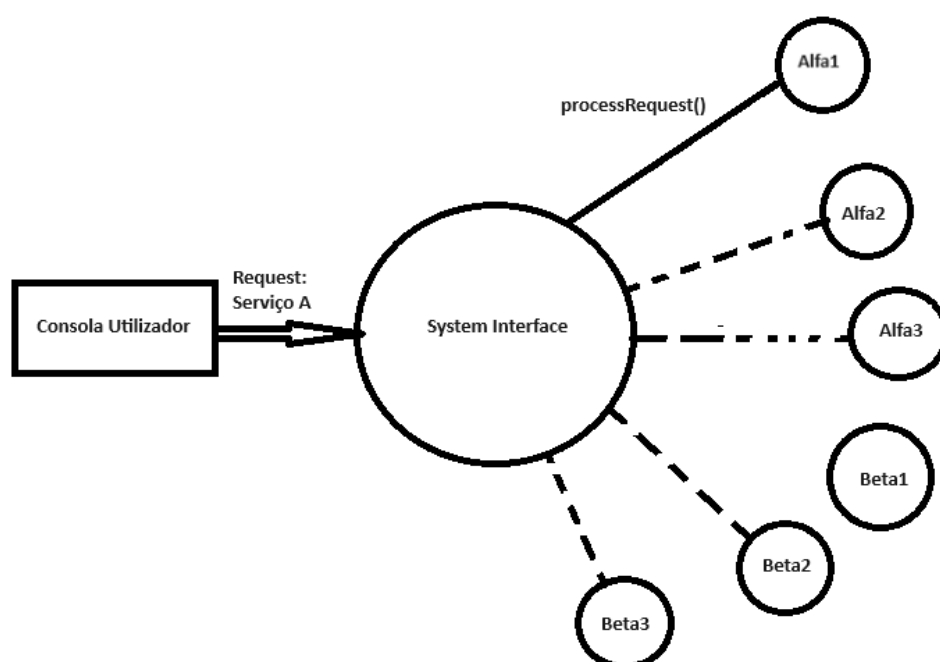


Figura 1 - Diagrama do fluxo de funcionamento do sistema.

O utilizador executa comandos através da consola do utilizador, estes comandos são interpretados e é feito o pedido correspondente à interface de sistema.

No caso de ser realizado um pedido de carregamento, a interface do sistema verifica sobre qual o tarifário o pedido está a ser feito instancia um objeto da classe referente e faz o pedido carregamento à mesma. Interpreta o resultado obtido e regista os dados referentes à operação.

Manual de Instruções

Preparação:

1. Instale o Java no seu computador.
2. Utilizando o seu IDE, abra um terminal.
3. Navegue até à diretoria da raiz do projeto ("/altice").
4. Execute o programa utilizando o comando "java Main".

Comandos da Consola do Utilizador:

- **exit** – encerra o sistema.
- **list_cdr** – lista todos os CDR's presentes no sistema.
- **list_requests** - lista todos os Charging Requests presentes no sistema.
- **list_replies** - lista todos os Charging Replies presentes no sistema.
- **list_accounts** - lista todos os Billing Accounts presentes no sistema.
- **search_cdr {msisdn}** – lista todos os CDR's relativos a um msisdn.
 - **msisdn** – MSISDN do conta que pretende procurar.
- **search {msisdn}** – encontra um Billing Account no Sistema com base no MSISDN
 - **msisdn** – MSISDN do conta que pretende encontrar
- **request {service} {roaming} {msisdn} {rsu}**
 - **service** – serviço que pretende usufruir
 - **roaming** - se a requisição necessita de roaming ("true" – sim | "false" – não)
 - **msisdn** – MSISDN do conta que pretende efetuar a requisição.
 - **rsu** – quantidade de unidades do serviço requisitadas

Detalhes da Implementação

Devido á existência de uma incoerência na designação dos Buckets acabei por designar-lhos por Bucker1, Bucket2 e Bucker3.

Caso seja feito um pedido de um serviço e a conta não possua créditos suficientes para cobrir o custo total, é feita a transação das unidade que o saldo da conta consiga cobrir.

Relatório de Testes

Durante os desenvolvimento do sistema foram desenvolvidos vários testes de modo a validar a corretude do código. Após a finalização foram mais uma série de testes unitários (Tabela 7) e um teste que abrangia todo o fluxo do sistema (Tabela 8). No entanto, por motivos de falta de tempo não foram realizados testes suficientes para abranger todas as possibilidades de acontecimentos, pelo que seria necessário realizar os mesmos.

Tabela 7 - Tabela de Relatório de Testes

Tarifário	Teste	Resultado
	Listagem de todos os CDR's.	X
	Listagem de todos os Charging Requests.	X
	Listagem de todos os Charging Replies.	X
	Listagem de todos as Billing Accounts.	X
	Procurar por um a conta.	X
	Procurar por uma conta que não existe.	X
	Procurar por os CDR's de uma conta.	X
	Procurar por os CDR's de uma conta que não existe.	X
	Inserir parâmetros não válidos ao fazer um request.	X
	Atualização do counter A.	X
	Atualização do counter B.	X
	Atualização do counter C.	X
	Atualização do counter D.	X
	Inserir comandos inexistentes.	X
Alfa1	Fazer request bem sucedido.	X
	Fazer request mal sucedido.	X
	Determinação do custo.	X
	Realização da Cobrança.	X
Alfa2	Fazer request bem sucedido.	X
	Fazer request mal sucedido.	X
	Determinação do custo.	X
	Realização da Cobrança.	X
Alfa3	Fazer request bem sucedido.	X
	Fazer request mal sucedido.	X
	Determinação do custo.	X
	Realização da Cobrança.	X
Beta1	Fazer request bem sucedido.	X
	Fazer request mal sucedido.	X
	Determinação do custo.	X
	Realização da Cobrança.	X
Beta2	Fazer request bem sucedido.	X
	Fazer request mal sucedido.	X
	Determinação do custo.	X

	Realização da Cobrança.	X
Beta3	Fazer request bem sucedido.	X
	Fazer request mal sucedido.	X
	Determinação do custo.	X
	Realização da Cobrança.	X

Tabela 8 - Utilizador utilizado para testar fluxo inteiro do sistema.

Campo	Valor
msisdn	987654321
bucket1	150
bucket2	9
bucket3	350
counterA	15
counterB	25
counterC	35
counterD	18-9-2023 12:00:00
tariffServiceA	Alfa2
tariffServiceB	Beta2

Ao executar o comando "request A false 987654321 10", isto é, ao efetuar uma requisição local para o serviço A (tarifário Alfa2) com o MSISDN 987654321 para 10 unidades, o sistema valida o pedido. Esta validação ocorre devido à natureza local da requisição e ao fato de que o valor presente no bucket2 do utilizador ser inferior a 10.

O sistema então calcula um custo por unidade de 0.3 cêntimos, considerando que a requisição está a ser feita durante o dia. Adicionalmente, é aplicado um desconto de 0.2 cêntimos devido ao facto do utilizador possuir um valor de counterB superior a 10.

Posteriormente, a transação é efetuada no bucket2 e todos os dados relativos à execução (CDR, Request, Reply) são registados de forma apropriada.