

Dicinário de Dados		DIAGRAMA de CLASSES								
Título:	Sistema BinaTur Software.						BINATUR			
Descrição:	Sistema para controle e captação de dados referentes ao objetivo comercial da agência de viagens BinaTur.									
Cliente:	BinaTur Ltda.									
CLASSE	ATRIBUTO	VEW	DESCRIÇÃO DO ATRIBUTO	TIPO DE DADO ATRIBUTO	METODO	VEW	DESCRIÇÃO DO METODO	PARÂMETRO	RETORNO	
USUARIO	id	-	O usuário define a corma como será identificado ao criar a conta.	Long	cadastrar()	+	Permite criar um novo cadastro no sistema.	(email: string, senha: string)	Boolean	
	email	-	O emaisl que o usuário usará para gerencias seua conta.	String	logar()	+	Permite acessar o sistema por meio de um cadastro existente.	(email: string, senha: string)	Boolean	
	senha	-	Definição da senha de acesso em sua conta.	String	recuperarSenha()	+	Permite criar uma nova senha em caso de esquecimento, extravio ou afins.	(email: string)	String	
CLIENTE	nome	-	Nome do usuário	String	contratarServico()	+	Permite contratar um dos serviços oferecidos.	(tipoServico: string)	Void	
	sobrenome	-	Sobrenome do usuário	String	cancelarServico()	+	Permite cancelar um ou mais serviços pré-selecionados.	(tipoContratado: string)	Void	
	cpf	-	CPF do usuário	String	pagar()	+	Permite efetuar o pagamento de um serviço escolhido.	(totalAPagar: float)	Void	
	dataNascimento	-	Data de nascimento do usuário	Date	logout()	+	Permite sair do sitema BINATUR.	(:)	Void	
	telefone	-	Telefone do usuário	String	editarDados()	+	Permite editar dados da conta a nível de gerenciamento do cliente.	(nome: string, sobrenome: string, telefone: string, email: string, endereco: string, dataNascimento: date)	Void	
	email	-	E-mail do usuário	String	excluirConta()	+	Permite excluir a conta do sistema BINATUR.	(:)	Void	
	endereco	-	Endereco do usuário	String	-----	---	-----	-----	-----	
PAGAMENTO	valor	-	Total a pagar.	Float	liquidarFatura()	+	Registra no sistema o comprovante do pagamento efetuado pelo cliente.	(:)	Void	
	tipo	-	Tipo de pagamento (a vista, parcelado, cartão, débito, crédito...).	String	parcelarValor()	+	Permite parcelar o valor da compra.	(numeroParcelas: int)	Void	
	descricao	-	Resumo dos serviços que compõem o valor do tapamento.	String	cancelarPagamento()	+	Permite cancelar o pagamento.	(:)	Void	
	numeroParcelas	-	Em caso de parcelamento indica qual o número de parcelas será efetuado o pagamento.	Int	-----	---	-----	-----	-----	
SERVICO	tipoServico	-	Indica qual entre os tres serviços (viagem, hospedagem ou pacote) o cliente escolheu.	String	selecionarTipoServico()	+	Permite o cliente selecionar um ou mais serviços oferecidos na plataforma.	(tipoServico: string)	Void	
	descricao	-	Resumo explicativo sobre um ou mais serviços foram selecionados pelo cliente.	String	adicionarTipoServico()	+	Permite o cliente adicionar mais um serviço ao carrinho de compras.	(tipoServico: string)	Void	

	totalAPagar	-	Apresenta o valor total a ser pago pelo cliente.	Float	alterar()	+	Permite fazer alterações nos detalhes dos serviços selecionados.	(:)	Void
	-----	---	-----	-----	excluir()	+	Permite excluir todos os itens no carrinho de compras.	(:)	Void

HOSPEDAGEM	ativo	-	Assinala em verdadeiro ou falso o status da opção hospedagem em relação aos itens escolhidos pelo cliente.	Boolean	reservarHosp()	+	Permite iniciar o processo de reserva de hospedagem.	(checkin: datetime, checkout: datetime, nomeHotel: string, precoHosp: float)	Void
	idHosp	-	Identificador do serviço de hospedagem escolhido.	Long	alterarHosp()	+	Permite alterar os detalhes da hospedagem escolhida.	(checkin: datetime, checkout: datetime, nomeHotel: string, precoHosp: float)	Void
	nomeHotel	-	Nome do estabelecimento de hospedagem escolhido pelo cliente.	String	cancelarHosp()	+	Permite cancelar uma hospedagem.	(:)	Void
	checkin	-	Data e hora do início da primeira diária.	DateTime	-----	---	-----	-----	-----
	checkout	-	Data e hora do fim do período de hospedagem.	DateTime	-----	---	-----	-----	-----
	precoHosp	-	Total a pagar por este serviço de hospedagem.	Float	-----	---	-----	-----	-----

VIAGEM	ativo	-	Assinala em verdadeiro ou falso o status da opção de viagem em relação aos itens escolhidos pelo cliente.	Boolean	agendarViagem()	+	Permite inserir os dados necessários para efetuar o agendamento da viagem escolhida.	(embarque: String, desembarque: string, dataEmbarque: date, horaEmbarque: time, dataDesembarque: date, horaDesembarque: time, precoViagem: float)	Void
	idViagem	-	Identificador do serviço de viagem escolhido.	Long	alterarViagem	+	Permite editar os dados necessários para um agendamento de viagem escolhido anteriormente.	(embarque: String, desembarque: string, dataEmbarque: date, horaEmbarque: time, dataDesembarque: date, horaDesembarque: time, precoViagem: float)	Void
	embarque	-	Local de embarque.	String	cancelarViagem()	+		(:)	Void
	dataEmbarque	-	Data marcada para o início da viagem.	Date	-----	---	-----	-----	-----
	horaEmbarque	-	Hora escolhida para o início da viagem.	time	'-----	---	-----	-----	-----
	desembarque	-	Local do desembarque.	String	'-----	---	-----	-----	-----
	dataDesembarque	-	Data marcada para o fim da viagem.	Date	'-----	---	-----	-----	-----
	horaDesembarque	-	Hora escolhida para o fim da viagem.	Time	'-----	---	-----	-----	-----
	precoViagem	-	Total a pagar por este serviço de viagem.	Float	-----	---	-----	-----	-----

PACOTE	ativo	-	Assinala em verdadeiro ou falso o status da opção de pacote (viagem + hospedagem) em relação aos itens escolhidos pelo cliente.	Boolean	abaterValor()	+	Permite aplicar o desconto, em forma de voucher, oferecido pelo sistema promocional da BINATUR.	(desconto: float)	Void
	idPacote	-	Identificador do serviço de pacote (viagem + hospedagem) escolhido.	Long	alterarPac()	+	Permite alterar dados do pacote.	(:)	Void
	idViagem	-	Dados da viagem que integra o pacote.	Long	cancelarPac()	+	Permite cancelar um pacote escolhido.	(:)	Void
	idHosp	-	Dados dahospedagem que integra o pacote.	Long	-----	---	-----	-----	-----
	desconto	-	Percentual de abatimento promocional por adquirir um pacote.	Float	-----	---	-----	-----	-----
	precoPacote	-	Total a pagar pelo serviço do pacote (valor da viagem + valor da hospedagem - o desconto dado pelo sistema).	Float	-----	---	-----	-----	-----

QUARTO	qtdQuarto	-	Sinaliza a quantidade de quartos o cliente escolheu em seu serviço de hospedagem.	Int	alterar()	+	Permite efetuar alterações nos quesitos do quarto selecionado.	(qtdQuarto: int, idQuarto: string, categoria: string, qtdQuarto: int)	Void
	idQuarto	-	Numeração específica do quarto ao qual o id da reserva faz menção (1-A, 2-c, 01,32...).	String	-----	---	-----	-----	-----
	categoria	-	Sinaliza a ou as categorias dos quartos escolhidas pelo cliente (standard, Luxo ou premium).	String	-----	---	-----	-----	-----
	qtdHospedes	-	Quantidade de hospedes cadastrados por quarto (obedecendo as diretrizes do hotel selecionado).	Int	-----	---	-----	-----	-----

TRANSPORTE	qtdTransporte	-	Sinaliza a possibilidade de mais de um transporte, ou ponte aérea e/ou baldeação durante o trajeto da viagem.	Int	alterar()	+		(qtdTransporte: int, tipoTransporte: string, idTransporte: long, idAgendamento: long, qtdPassageiros: int, assento: string)	void
	tipoTransporte	-	Sinaliza qual ou quais veículos serão utilizados (avião, navio, ônibus ou trem)	String	-----	---	-----	-----	-----
	idTransporte	-	Placa, nome, serial, código de chasi... Que defina o transporte selecionado.	Long	-----	---	-----	-----	-----
	idAgendamento	-	Código do agendamento da viagem.	Long	-----	---	-----	-----	-----
	qtdPassageiros	-	Quantidade de passageiros e consequentemente passagens para a realização da viagem.	Int	-----	---	-----	-----	-----
	assento	-	Numeração específica do assento ao qual o id da reserva faz menção (1-A, 2-c, 01,32...)	String	-----	---	-----	-----	-----

Dicinário de Dados		CLASSES e RELACIONAMENTOS		
Título:	Sistema BinaTur Software.			BINATUR
Descrição:	Sistema para controle e captação de dados referentes ao objetivo comercial da agência de viagens BinaTur.			
Cliente:	BinaTur Ltda.			
NOME DA CLASSE	DESCRIÇÃO DA CLASSE	TIPO DE RELACIONAMENTO	CLASSES RELACIONADAS	MULTIPLICIDADE
USUARIO	Esta classe representa os serviços oferecidos pela agência de viagens.	<u>U - C</u> : este é um relacionamento de Composição . Um Cliente é composto por um Usuário, o que significa que a existência de um Cliente depende de um Usuário. É uma relação do tipo "parte-todo".	Relacionamento com as classes Viagem, Hospedagem e Pacote: Cada Serviço pode ser uma Viagem, Hospedagem ou Pacote.	1 Usuário pode estar associado a 0 ou 1 Cliente (relacionamento 1-0..1).
CLIENTE	Esta classe representa os clientes da agência de viagens.	<u>C - S</u> : este é um relacionamento de Agregação . Um Cliente pode agregar (ou seja, contratar) vários Serviços, como Viagens, Hospedagens ou Pacotes. É uma relação do tipo "todo-parte", em que o Cliente é o "todo" e os Serviços são as "partes".	<u>Relacionamento com a classe Serviço</u> : Um Cliente pode contratar vários Serviços; <u>Relacionamento com a classe Pagamento</u> : Um Cliente faz N Pagamentos.	Cliente pode contratar N Serviços (relacionamento 1-N); 1 Cliente pode fazer N Pagamentos (relacionamento 1-N).
PAGAMENTO	Esta classe representa os pagamentos feitos pelos clientes.	<u>C - PG</u> : este é um relacionamento de Agregação . Um Cliente pode fazer vários Pagamentos, e cada Pagamento está associado a um Cliente; <u>PG - S</u> : este é um relacionamento de Dependência . Um Pagamento depende das informações da classe servico para fazer sentido.	<u>Relacionamento com a classe Cliente</u> : Cada Pagamento está associado a um Cliente; <u>Relacionamento com a classe Serviço</u> : O pagamento depende de informações da tabela serviço.	1 Pagamento pertence a 1 Cliente (relacionamento 1-1); 1 Pagamento depende da informação da tabela serviço.
SERVICO	Esta classe representa os serviços oferecidos pela agência de viagens.	<u>S - V, S - H, S - PC</u> : Aqui, temos um relacionamento de Generalização . A classe Serviço é generalizada por três subclasses: Viagem, Hospedagem e Pacote. Isso reflete a ideia de que cada Serviço é um tipo específico, como Viagem ou Hospedagem.	Relacionamento com as classes Viagem, Hospedagem e Pacote: Cada Serviço pode ser uma Viagem, Hospedagem ou Pacote.	1 Serviço é uma Viagem ou 1 Serviço é uma Hospedagem ou 1 Serviço é um Pacote (relacionamento 1-1..1).
VIAGEM	Esta classe representa os serviços de viagem, incluindo o transporte.	<u>V - T</u> : este é um relacionamento de Associação . Uma Viagem está associada a um Transporte, mas a existência de um não depende do outro.	Relacionamento com a classe Transporte: Cada Viagem usa 1 Transporte.	1 Viagem usa 1 Transporte (relacionamento 1-1).
HOSPEDAGEM	Esta classe representa os serviços de hospedagem.	<u>H - Q</u> : este é um relacionamento de Composição . Uma Hospedagem é composta por um ou mais Quartos. A existência de um Quarto depende de uma Hospedagem.	Relacionamento com a classe Quarto: Cada Hospedagem usa N Quartos.	1 Hospedagem usa N Quartos (relacionamento 1-N).
PACOTE	Esta classe representa a compra de um pacote que inclui Viagem e Hospedagem.	<u>PC - V, PC - H</u> : aqui temos um relacionamento de Generalização . A classe Pacote é generalizada pelas classes Viagem e Hospedagem, refletindo a ideia de que um Pacote pode incluir uma Viagem e uma Hospedagem.	Relacionamento com as classes Viagem e Hospedagem: Cada Pacote inclui 1 Viagem e 1 Hospedagem.	1 Pacote inclui 1 Viagem e 1 Hospedagem (relacionamento 1-1).
TRANSPORTE	Esta classe representa as opções de transporte para a viagem.	Explicado em VIAGEM.	Relacionamento com a classe Viagem: Cada Transporte é usado em 1 Viagem.	1 Transporte é usado em 1 Viagem (relacionamento 1-1).
QUARTO	Esta classe representa os quartos disponíveis para hospedagem.	Explicado em HOSPEDAGEM.	Relacionamento com a classe Hospedagem: Cada Quarto é usado em 1 Hospedagem.	1 Quarto é usado em 1 Hospedagem (relacionamento 1-1).