
Desenvolvimento de Sistemas Software

2008/2009

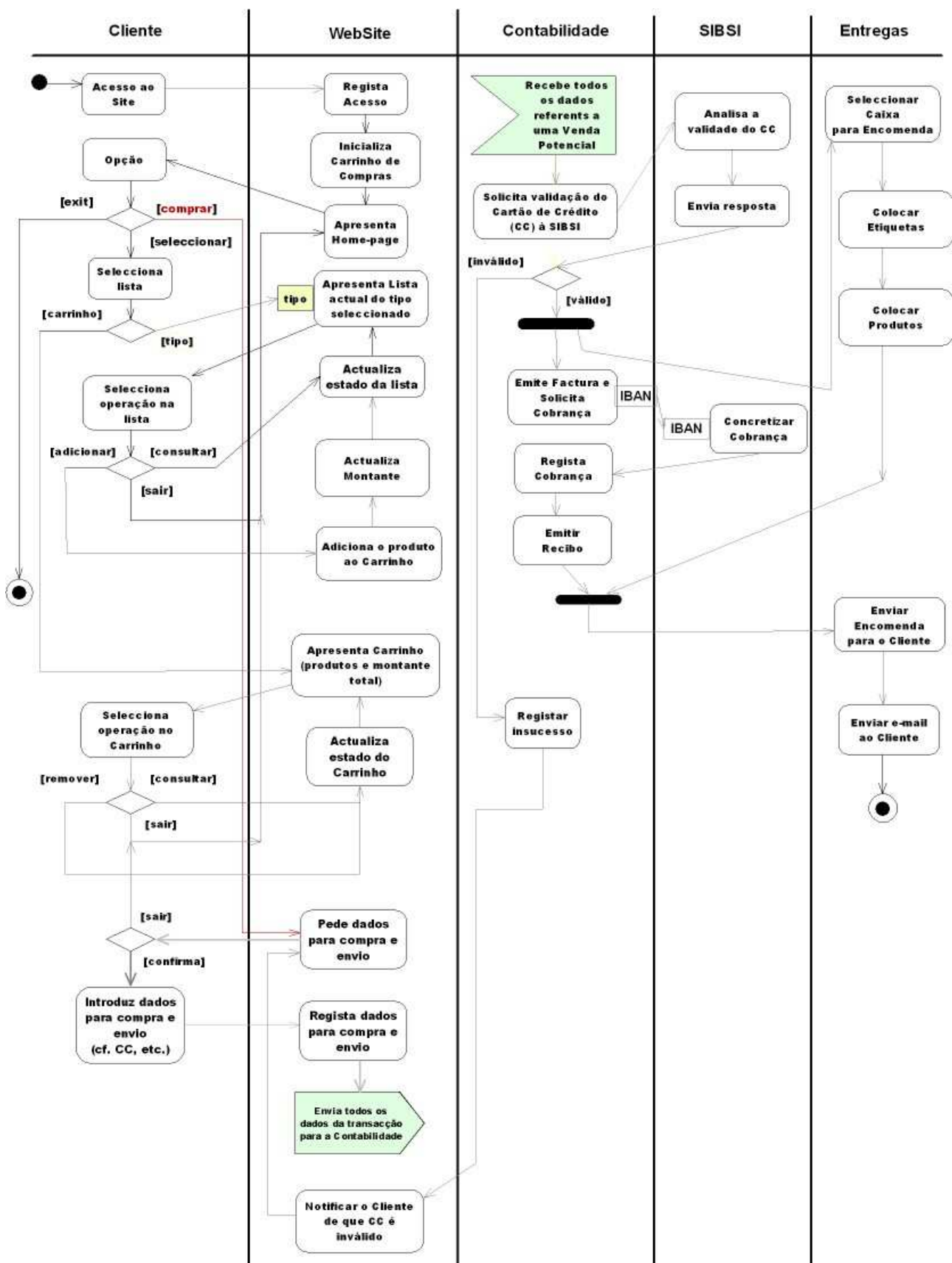
3º ano - LEI

PROPOSTA DE RESOLUÇÃO

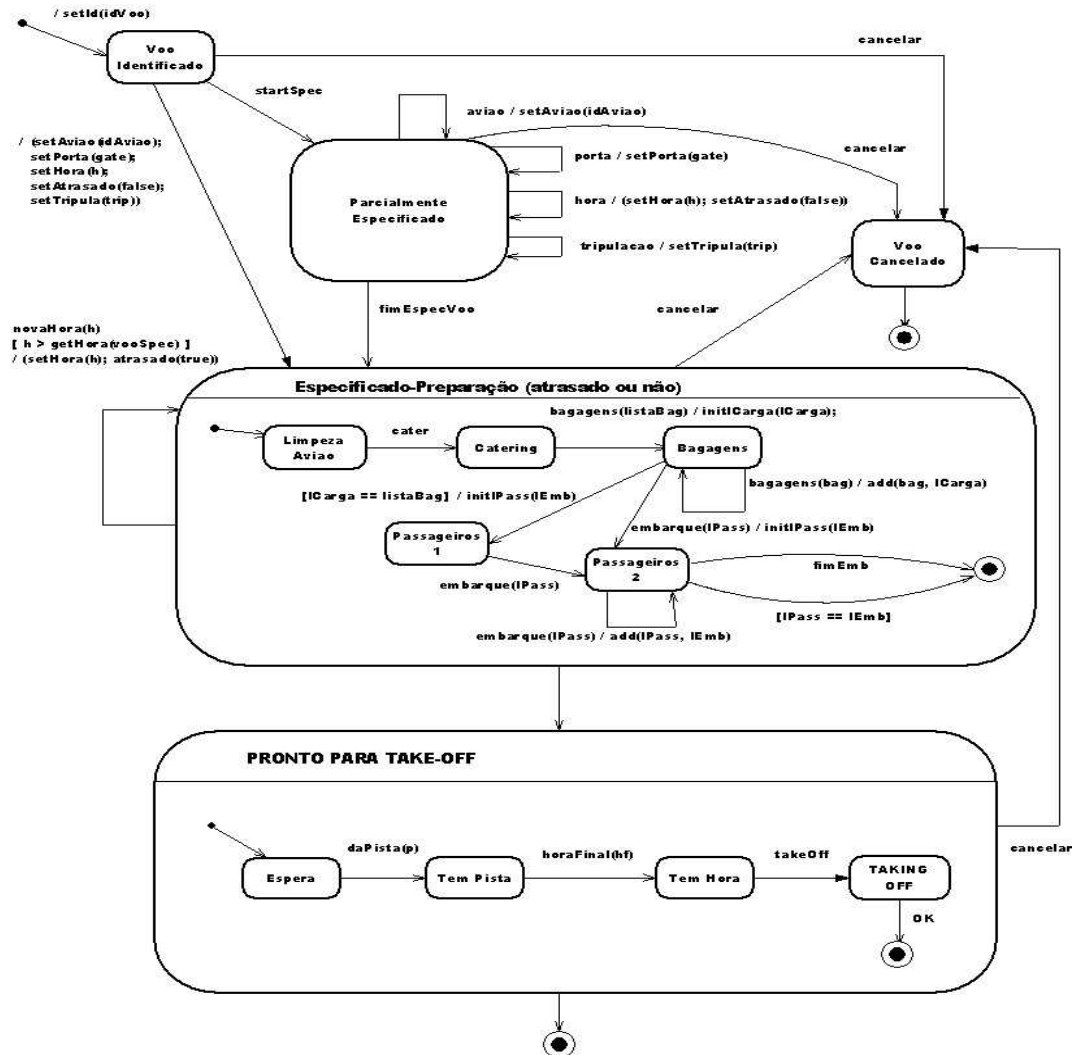
F. Mário Martins

GRUPO I (8 valores)

1.- Desenvolver um Diagrama de Actividades



2.- Desenvolver um Diagrama de Estados



NOTA: Não existe realmente um estado **PREPARAÇÃO COM ATRASO** após alterar a hora ao estado **ESPECIFICADO-PREPARAÇÃO** por várias razões. A primeira é que tal estado tem apenas um atributo diferente do estado **ESPECIFICADO-PREPARAÇÃO**, o valor do atributo booleano *atrasado*. Se tal evento de alteração de hora conduziisse a um estado seguinte de nome **PREPARAÇÃO COM ATRASO** haveria agora que assegurar que o comportamento dentro deste novo estado seria igual, ou seja, o restante do estado donde partimos. Ainda que alguns alunos tivessem usado H (aliás de forma inteligente, o H não funciona porque o H apenas repõe o estado anterior ao nele se reentrar e não, como gostaríamos neste caso, o estado anterior **DENTRO DO NOVO ESTADO**. A etiqueta "(atrasado ou não)" significa que um método *atrasado?* desencadeado por um evento atrasado?, poderá sempre ser usado para consultar o "verdadeiro" estado interno do estado que se designou por **ESPECIFICADO-PREPARAÇÃO (atrasado ou não)**. Foram consideradas válidas diferentes ideias apresentadas pelos alunos à volta desta questão. Esta justificação visa esclarecer a solução apresentada, que é a mais simples, e até possível, em função da semântica UML para Diagramas de Estados.

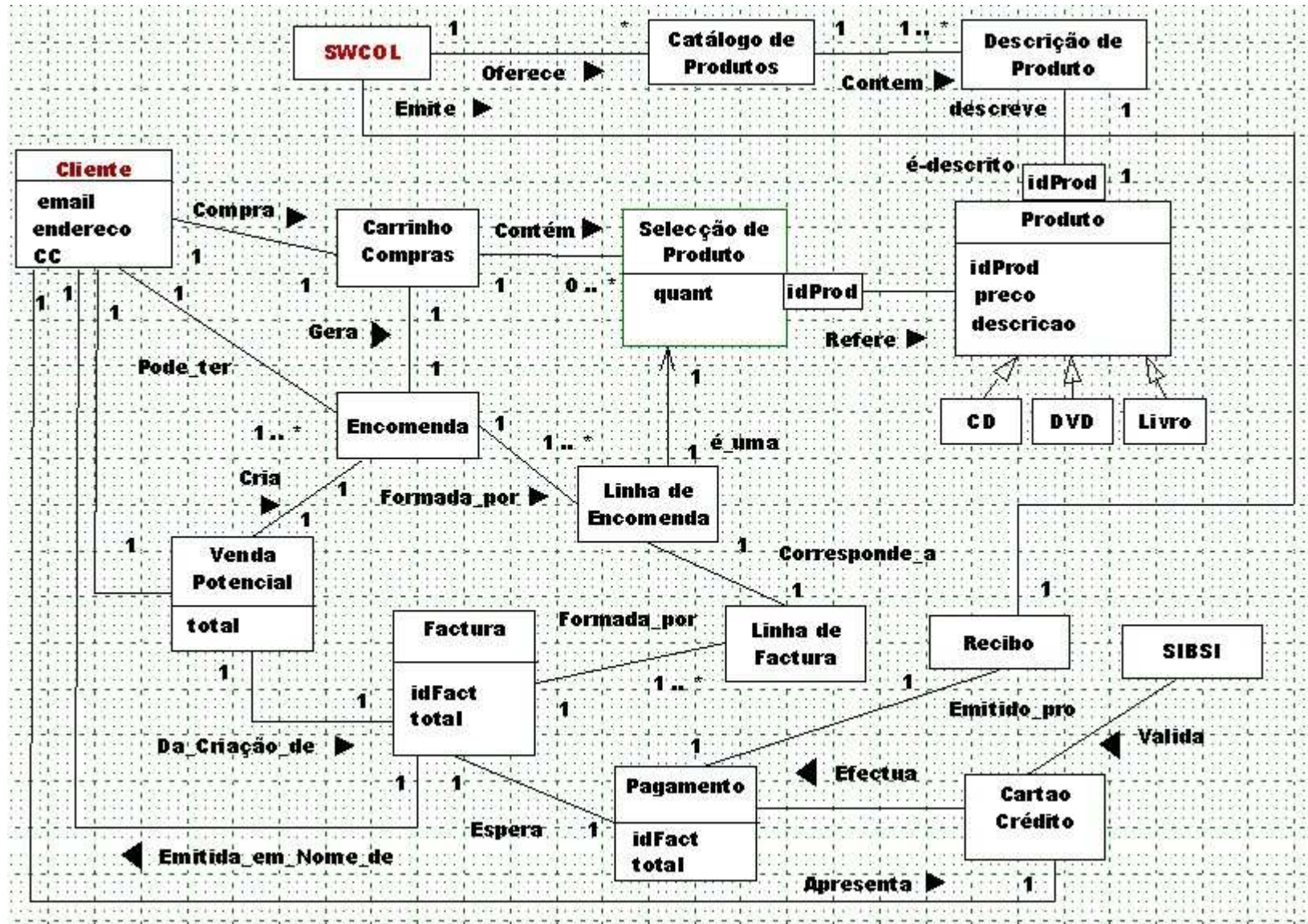
O que gostaríamos de poder ter:



Esta semântica indicaria que no estado **ESP_PREP_ATRASO** deveríamos continuar a realizar todas as tarefas ainda por realizar (especificadas em **ESP_PREP**), mas tendo um atributo alterado. Não possuindo esta capacidade notacional em UML2.0, deve-se usar um evento externo a todo o estado composto, o que significa na semântica de UML que pode ocorrer a qualquer momento da execução do estado composto, e que altera atributos internos a tal estado, que podem ser consultados. Assim, não há uma verdadeira mudança de estado, apenas uma mudança dos valores internos de tal estado.

GRUPO II (12 valores)

1.- Modelo do Domínio – Diagrama de Classes Conceptual



2.- Use Case Textual (Comprar Carrinho de Compras – Actor: Cliente)

Use Case: Comprar Carrinho de Compras			
Actor: Cliente			
Desc: Cliente pretende comprar os produtos que seleccionou para o seu carrinho de compras			
Pré: Carrinho é válido (todos os produtos têm existência em stock)			
Pós: Cada produto comprado (na quantidade comprada) deverá ser abatido no stock;			
Os dados do Cliente são registados na BD Clientes; A Contabilidade regista a entrada de um pagamento igual ao total da compra			
		Cliente	Sistema
	1	Notifica que pretende comprar o Carrinho actual	
	2	Calcula valor total da Venda Potencial (Carrinho)	
	3	Pede dados do Cliente	
	4	Apresenta os dados pedidos	
	5	Regista os dados do Cliente	
FLUXO SUCESSO	6	Pede dados de Cartão de crédito para Pagamento da Venda	
	7	Apresenta informação do Cartão de Crédito	
	8	Confirma validade do Cartão	
	9	Regista informação do Cartão do Cliente	
	10	Verifica se a conta do Cartão tem saldo suficiente (saldo > total)	
	11	Processar Transferência Bancária (pagamento do total)	
	12	Processa Encomenda	
	13	Emite Factura	
	14	Emite Recibo	
	15	Apresenta Home-Page	
ALTERNATIVA 8	1	8: Dados de Cartão inválidos (engano?)	
	2	Informa Cliente da invalidade do Cartão apresentado	
	3	Voltar ao passo 6 do Main Flow	
ALTERNATIVA 10	1	10: Saldo da conta associado ao Cartão é insuficiente	
	2	Informa Cliente da inexistência de saldo no Cartão	
	3	Pede dados de Cartão de crédito para Pagamento da Venda	
	4	Voltar ao passo 6 do Main Flow	
EXCEPÇÃO 8	1	8: Sistema internacional de validação está OFF	
	2	Informa Cliente do problema	
	3	Regista ocorrência na BD de falhas	
	4	STOP	

3.- Diagrama de Sequência (com Subsistemas)

