

EP 2 POO

Integrantes:

Bruno Henrique Yoshimura #USP: 5122944

Felipe Carvalho Perestrelo #USP: 3728320

Gustavo Andrade Nery #USP: 5123389

Arquitetura do Sistema:

A arquitetura da fase 1 continuou praticamente a mesma, com apenas algumas modificações para adaptar a interface Web.

Já para a fase 2, implementamos algumas classes:

Main

interface Web principal. Controla o style das páginas e o menu principal

Main Gerente

interface para as funcionalidades do Gerente:

- 1) Ver Comentários deixados pelos clientes sobre os serviços do restaurante e sobre os pratos do cardápio.
- 2) Ver as mensagens do robô.

herda Main

Main Garcom

interface para as funcionalidades do Garcom:

- 1) Filtrar o Cardapio (por Tempo de preparo, por Preço e por Ingredientes).
- 2) Gerenciar o Cardapio (editar, adicionar e remover pratos).
- 3) Gerenciar Pedidos (fazer/cancelar pedidos).

herda Main

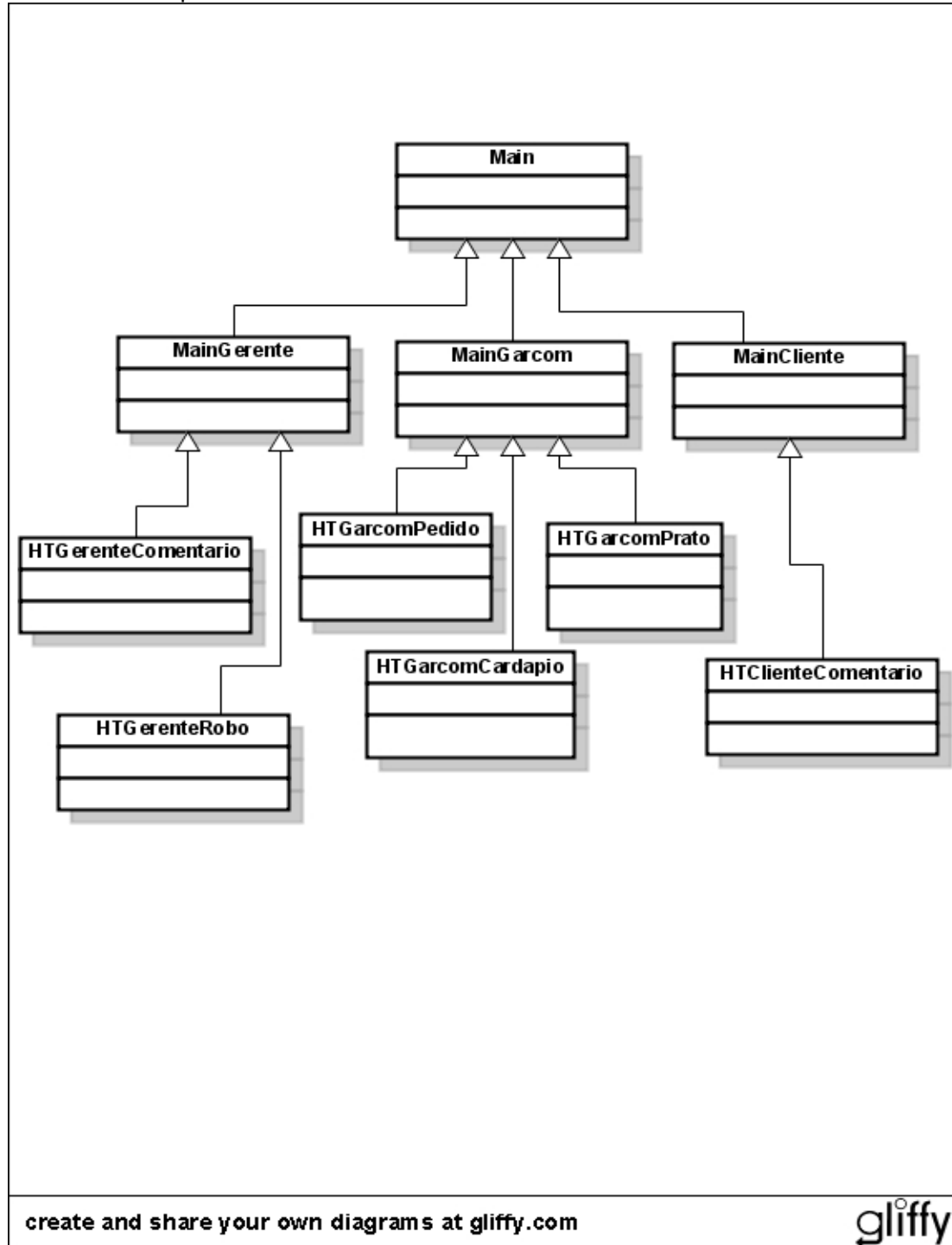
Main Cliente

interface para as funcionalidades do Cliente:

- 1) Inserir Comentários sobre os pratos do cardapio e serviços do restaurante.

herda Main

Abaixo um simples UML das classes:



Bruno

Funcionalidades Novas

1) Comentários do Cliente (1 ponto)

Os clientes podem avaliar qualquer *prato* ou *serviço* do sistema, atribuindo uma nota de 1 a 5 e escrevendo um comentário. Fixamos alguns serviços que são fornecidos por restaurantes, tais como *Banheiro*, *Manobrista*, *Atendimento*, *Ambiente*, etc.

2) Filtro de Cardapio (3 pontos)

a) Filtro por preço mínimo e filtro por preço máximo.

- b) Filtro por tempo de preparo mínimo e tempo de preparo máximo.
c) Filtro de prato com ingrediente e sem ingrediente.

3) Comprar Ingrediente (??? ponto)

É possível comprar ingredientes novos, além daqueles já setados. Isso é interessante caso o garçom queira adicionar um prato novo, com ingredientes novos.

4) Montagem de Pratos(3 pontos)

A Montagem de pratos ficou incompleta, podendo o garçom apenas montar um prato novo ao cardápio com os ingredientes que o cliente desejar. Se os ingredientes não estiverem na lista atual de ingredientes, o garçom pode adicionar o ingrediente (utilizando o item 3 acima) para conseguir montar o novo prato e assim definir seu preço, tempo de preparo e nome.

Organização das Classes

HTClienteComentario

adicionaComentario

Dado: umTipo, umNome, umComentario, umaNota

Ação: Adiciona um novo comentário.

Detalhes:

- tipo:servico, prato
- nome: é o nome do prato ou servico
- nota: de 1 a 5

HTGarcomCardapio

busca: umTipo

Dado: umTipo, umFiltro, umValor

Ação: Faz a busca do cardapio com um filtro.

Detalhes:

- tipo:tempoMaior, tempoMenor, valorMaior, valorMenor, comIngrediente, semIngrediente
- valor: Caso seja um filtro de tempo ou preco, recebe um inteiro. Caso seja um filtro de ingrediente, recebe um nome de ingrediente.

HTGarcomPedido

adicionaPrato: umPrato

Dado: umPrato

Ação: Adiciona o prato ao pedido do cliente

fechaPagamento

Dado: pagamento, pedido

Ação: Cuida do fechamento e pagamento.

Detalhes: Recebe o dinheiro e verifica se é suficiente. Se precisar, devolve o troco. Uma consequencia do pedido é enviar uma 'mensagem' para o cozinheiro, que vai pedir para o robo preparar.

removePrato:umPrato

Dado: prato

Ação: Remove um prato do pedido

HTGarcomPrato

adicionaPrato: umPrato

Dado: umPrato

Ação: Adiciona o prato no cardápio do restaurante.

Detalhes: Verifica se existe um prato com o mesmo nome. A comparação ignora espaços em branco e letras maiúsculas, para garantir que a unicidade dos pratos.

comprarIngrediente

Dado: ingrediente

Ação: Adiciona um ingrediente, caso ainda não existe.

Detalhes: Verifica se já existe o ingrediente no restaurante. Essa funcionalidade pode ser útil quando o garçom quer criar um novo prato.

editarPrato:umPrato

Dado: prato

Ação: Edita um prato do cardápio.

Detalhes: É possível alterar: valor, tempo de preparação, lista de ingredientes.

removePrato: umPrato

Dado: umPrato

Ação: Remove um prato do cardápio.

Detalhes: Essa remoção não afeta os comentários já postados sobre o dado prato. Poderíamos remover esses comentários, mas achamos mais interessante manter.

HTGerenteComentario

renderContentOn: html

Detalhes: Apenas serve para mostrar os comentários para o gerente. Além de poder ver a lista de todos pratos e serviços que foram comentados, ele mostra um relatório individual de comentários com a média de avaliação do mesmo

HTGerenteRobo

renderContentOn: html

Detalhes: Serve para mostrar o log de ações que o robô fez.

HTGerenteRobo

renderContentOn: html

Detalhes: Serve para mostrar o log de ações que o robô fez.