[Introdução](#_nbm0rcxkfhra)

[Caso de Uso 1: CRUD de hóspede](#_kf0p3hummbco)

[Caso de Uso 2: Checkin de hóspedes](#_9yggn9z50vu8)

[Caso de Uso 3: Checkout de hóspedes e histórico de lucros](#_a8ryz6l7t7ys)

[Caso de Uso 4: Restaurante do Hotel](#_208j29o48ak0)

[Caso de Uso 5: Organização do cardápio e pedidos](#_x2m846rh6r5k)

[Caso de Uso 6: Cartão e Pontos de Fidelidade](#_ludjh236kutz)

[Caso de Uso 7: Descontos e benefícios da fidelidade](#_evg4nqn8ka2c)

[Caso de Uso 8: Geração de relatórios do sistema](#_ndjv7m1clrbt)

[Caso de Uso 9: Persistência do sistema](#_3mqx90ppuefj)

# Introdução

O Hotel Urbano de Gotemburgo (HUG) é um dos hotéis mais tradicionais para os visitantes suecos que querem economizar um pouco de dinheiro para aproveitar a Suécia. No entanto, mesmo sendo um hotel acessível, o mesmo apresenta alguns confortos como um restaurante e um cartão de fidelidade para seus clientes.

Claro, por estar em um país de alta tecnologia, o sistema desse hotel é completamente informatizado. Hóspedes do hotel devem ser registrados nesse sistema e, devido às leis de privacidade, facilmente removidos. O hóspede se registra em um quarto específico e pode fazer uso do restaurante do hotel. É preciso controlar tanto o cardápio (do clássico *smörgåsbord* até o apetitoso *gravlax* e a inesquecível *pannkaka* de sobremesa) quanto o preço associado de cada produto.

Os gastos dos hóspedes devem ser registrados, para permitir a gerência financeira do hotel. À medida que os hóspedes gastam dinheiro, isto também será contabilizado no cartão de fidelidade de cada cliente, de forma que os mesmos possam obter acesso a descontos para futuras compras.

É sua responsabilidade desenvolver este sistema para o administrador do HUG. Além do controle básico de hóspedes, estadias, gerência das refeições do restaurante e do cartão de fidelidade, esse sistema deverá armazenar as informações históricas de gastos, bem como ser capaz de salvar e recuperar o estado atual do sistema. Estas funcionalidades estão descritas nos casos de uso apresentados neste documento.

# Caso de Uso 1: CRUD de hóspede

**User Story**: Como administrador eu desejo cadastrar e remover hóspedes do hotel para registrar estadias e seus gastos.

O primeiro passo é realizar o cadastro dos hóspedes do Hotel. Cada hóspede possui um nome, um email para contato e um ano de nascimento. Note que o cadastro do hóspede é usado pelo sistema independente dele(a) estar hospedado(a), pois ao retornar no hotel, os dados do hóspedes estarão ainda no sistema.

Além de cadastrar, você deve permitir a edição/atualização dos dados de um hóspede (nome, email e data de nascimento), e permitir a busca e remoção de hóspedes do sistema através do email.

# Caso de Uso 2: Checkin de hóspedes

**User Story**: Como administrador eu desejo realizar checkin de hóspedes cadastrados no sistema para efetivar a estadia de cada hóspede.

Ao realizar um checkin o hóspede será associado a uma ou mais estadias. Cada estadia é composta por um quarto e uma quantidade de dias. Cada quarto possui um ID que representa o “número do quarto” (e.g., 2A, 15B, 214A) e um valor de diária que varia de acordo com o tipo do quarto que podem ser: Simples, Luxo e Presidencial, com diárias de respectivamente R$100,00; R$250,00; R$450,00.

Cada estadia informa também o total que será pago pelo hóspede de acordo com o número de dias e o tipo do quarto. Um mesmo hóspede pode estar associado a várias estadias. Note que se uma família se hospedar, não é necessário cadastrar todos os membros. O sistema cadastra apenas uma pessoa que será a responsável pelas estadias.

# Caso de Uso 3: Checkout de hóspedes e histórico de lucros

**User Story**: Como administrador desejo realizar o checkout dos meus hóspedes para registrar os lucros de estadias obtidos pelo Hotel.

Antes de sair do hotel o hóspede deve fazer checkout. Nesse momento, o administrador do hotel vai inserir no sistema os dados de checkout do hóspede. O objetivo dessa funcionalidade é registrar o histórico de lucros do Hotel. O registro de checkout deve constar com a data do sistema ao realizar o checkout, o nome do hóspede, o número do quarto associado a sua estadia e o total pago pelo hóspede ao fazer o checkout. Note que esse registro não precisa estar sincronizado com as atualizações do hóspede, pois ele representa a transação realizada no checkout. Se um hóspede está associado a mais de uma estadia, o procedimento de checkout deve ser realizado para cada estadia.

Dica: Para pegar a data do sistema use: LocalDate.now();

# Caso de Uso 4: Restaurante do Hotel

**User Story**: Como administrador desejo cadastrar, atualizar e remover pratos e refeições para oferece-los aos hóspedes do Hotel.

Um dos grandes destaques do HUG é o seu restaurante exclusivo para hóspedes. Esse restaurante possui pratos e refeições. Um prato possui um nome, um preço e uma descrição, por exemplo: (*ärtsoppa com panquecas, R$18,50, Deliciosa sopa de ervilhas com carne. Essa comida tradicional da Suecia é saboreada principalmente nas quintas-feiras.*); (*Kötbullar com purê de batatas e lingon, R$ 38,50, Almondegas de carne feito no tradicional estilo sueco.*). Além de pratos a la carte como os exemplos anteriores, existe no restaurante o conceito de uma refeição completa.

As refeições seguem um cardápio internacionalizado. Uma refeição completa possui um nome, uma descrição própria e é composta por uma sequência de pratos que o restaurante oferece podendo englobar: entrada, prato principal, sobremesa e *petit four*, sendo a sobremesa ou petit four opcionais. (Em resumo a refeição pode ser composta por 3 pratos ou por 4 pratos, não menos que isso). Por exemplo, podemos ter uma Refeição Italiana composta por: (Bruschetta, R$ 10,50); (Penne ao Pesto, R$ 34,50); (Tiramisu, R$ 12,00); (Espresso, R$ 2,50). Ou então uma Refeição Sueca com (ärtsoppa com panquecas, R$ 18,50); (Plankstek com purê de batatas gratinado, R$ 35,60); (Kanelbullar com Kaffe, R$19,00). Note que a ordem dos pratos é importante em uma refeição.

Enquanto que um hóspede paga o preço integral de cada prato, o preço da refeição completa será a soma dos preços de seus pratos com um desconto de 10% nessa soma total. Toda refeição e prato cadastrados devem fazer parte do mesmo cardápio do restaurante. Além disso, faça com que seja possível apenas cadastrar refeições com pratos presentes no cardápio.

# Caso de Uso 5: Organização do cardápio e pedidos

**User Story**: Como administrador desejo disponibilizar um cardápio com todos os pratos e refeições para que os hóspedes possam realizar pedidos.

Para facilitar o uso do cardápio pelos administradores do sistema, permita que o cardápio possa ser ordenado utilizando:

· Ordem crescente alfabética pelo nome do prato/refeição completa;

· Ordem crescente de preço de cada prato/refeição completa;

Além disso, faça com que o sistema registre os pedidos dos hóspedes. Para isso, o hóspede deve informar o nome do prato ou refeição completa, e a cobrança é registrada diretamente no histórico de transações do sistema, seguindo a mesma lógica definida no checkout, porém no lugar do nome do quarto, será registrado o nome da refeição/prato pedido.

# Caso de Uso 6: Cartão e Pontos de Fidelidade

**User Story**: Como sistema desejo pontuar meus hóspedes de acordo com seus gastos para oferecer diferentes tipos de descontos pelo cartão de fidelidade.

Agora que seu sistema está em pleno funcionamento você precisa oferecer algumas bonificações para os hóspedes. Todo hóspede agora possui um cartão de fidelidade para acumular pontos (inteiros). O cartão possui três tipos de fidelidade: Padrão, Premium e VIP. As responsabilidades do cartão são: adicionar pontos, aplicar desconto em gastos e pagar dívidas com gastos.

Ao adicionar pontos o cartão adiciona também uma quantidade de pontos bônus que varia de acordo com o tipo da fidelidade. De forma semelhante, cada tipo de fidelidade aplica um desconto distinto em uma cobrança realizada ao hóspede. Note que os hóspedes devem trocar dinamicamente entre os tipos de fidelidade de acordo com a quantidade de pontos que acumulam. Os limites de pontos para definir os tipos de Fidelidade são:

* Padrão: Menor que 350 pontos
* Premium: Entre 350 e 1000 pontos (ambos inclusos)
* VIP: Maior que 1000 pontos.

Os pontos são recompensados ao registrar um gasto (e.g., checkout e pedidos no restaurante). Cada fidelidade recompensar uma quantidade de pontos de acordo com o valor que foi gasto. Esse valor de pontos é calculado em cima do valor original do gasto, independente do desconto que o hóspede possa obter (veja proximo caso de uso).

* Padrão: 10% do valor gasto é convertido automaticamente em pontos. Exemplo:
  + Tapioca: R$3,00 = 0 pontos;
  + Checkout: R$1535,89 = 153 pontos;
* Premium: A recompensa sobe para 30% dos pontos gastos. Porém, se o gasto for maior que R$100,00, o hóspede ganha 10 pontos adicionais para cada R$100,00 gastos acima dos 100.
  + Pudim de leite: R$6,50 = 1 ponto;
  + Svensk mat: R$97,65 = 29 pontos;
  + Checkout: R$2534,85 = 760 pontos + 250 pontos = 1015 pontos;
* VIP: A recompensa sobe para 50% dos pontos gastos. Por exemplo:
  + Tapioca: R$ 3,00 = 1 ponto;
  + Svensk mat: R$97,65 = 48 pontos;
  + Checkout: 1805,44 = 902 pontos;

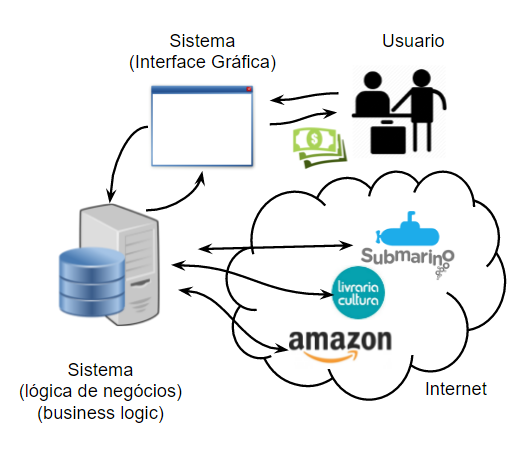
Os tipos de fidelidade também permitem diminuir o valor de uma cobrança feita ao cliente. Se após o desconto, o valor tiver mais que duas casas decimais, o sistema deve sempre arredondar o valor para cima. Os seguintes descontos são aplicados:

* Padrão: Não é aplicado desconto. É retornado o próprio valor da cobrança
  + Tapioca: R$3,00 (original) = R$3,00 (com desconto);
  + Checkout: R$1535,89 (original) = R$1535,89 (com desconto);
* Premium: É aplicado um desconto de 10% no valor da cobrança.
  + Pudim de leite: R$6,50 (original) = R$5,85 (com desconto);
  + Svensk mat: R$97,65 (original) = R$87,885 ~ R$87,89 (com desconto);
  + Checkout: R$2534,85 (original) = R$2281,365 ~ R$2281,37 (com desconto);
* VIP: Essa categoria oferece o melhor desconto, que é de 15%. Só que a cada R$100,00 em gastos, o hóspede recebe um desconto de R$10,00 dos R$100,00 pagos. Por exemplo:
  + Tapioca: R$ 3,00 (sem desconto) = R$2,55 (com desconto)
  + Svensk mat: R$97,65 (sem desconto) = R$83,0025 ~ R$83,01 (com desconto)
  + Checkout: 1805,44 = R$1534,63 - R$180,00 = R$1354,63 (com desconto);

# Caso de Uso 7: Descontos e benefícios da fidelidade

**User Story**: Como sistema desejo utilizar os pontos dos usuários para fornecer operações de crédito.

Por fim, um hóspede pode usar seu cartão de fidelidade para converter pontos em dinheiro. Essas conversões funcionam como um saque no sistema do Hotel ou feita por requisição online por outros sites de compras (e.g., Submarino, Livraria Cultura, Mercado Livre, Amazon, etc.). Note que esse dinheiro convertido não precisa ser armazenado no nosso sistema, mas apenas retornado pelo sistema.



* Padrão: Para cada ponto de fidelidade, o hóspede consegue sacar R$0,10;
  + Ex: Pontos: 158 pontos = R$15,80
* Premium: Cada ponto de fidelidade vale R$0,30 e a cada 10 pontos convertidos, o hóspede ganha R$0,20 adicionais;
  + Ex: 5 pontos = R$1,00
  + Ex: 50 pontos = R$10,00 + R$1,00 = R$11,00
* VIP: Cada ponto de fidelidade vale R$0,70 e a cada 10 pontos convertidos, o hóspede ganha R$0,50 adicionais. Por exemplo:
  + Ex: 5 pontos = R$3,50
  + Ex: 50 pontos = R$35,00 + R$2,50 = R$37,50.

É importante, no entanto, atualizar a quantidade de pontos presentes no cartão do respectivo hóspede que fez a conversão. Ou seja, os pontos convertidos serão debitados do cartão de fidelidade do hóspede impactando, ou não no tipo de fidelidade que o hóspede irá possuir (veja os limiares no caso de uso 6).

# Caso de Uso 8: Geração de relatórios do sistema

**User Story**: Como administrador desejo criar relatórios com os dados do sistema para ter uma persistência em texto dos dados do sistema.

Diante de tantas estratégias sofisticadas de cálculo, recompensa e operações realizadas no seu sistema, é importante registrar em arquivos de texto o histórico de transações, os pratos e refeições cadastradas no seu restaurante, bem como o cadastro completo dos hóspedes cadastrados. Diante disso, deve existir três formas de salvar o arquivo texto.

**Cadastro de hóspedes:**

|  |
| --- |
| Cadastro de Hospedes: <total\_hospedes> hospedes registrados  ==> Hospede 1:  Email: <email\_do\_hospede>  Nome: <nome\_do\_hospede>  Data de nascimento: aaaa-mm-dd  ==> Hospede 2:  Email: <email\_do\_hospede>  Nome: <nome\_do\_hospede>  Data de nascimento: aaaa-mm-dd  … |

**Cadastro de pratos e refeições completas:**

|  |
| --- |
| Menu do Restaurante: <total\_itens> itens no cardapio  ==> Item 1:  Nome: <nome> Preco: R$<preco>  Descricao: <descricao>  ==> Item 2:  Nome: <nome> Preco: R$<preco>  Descricao: <descricao>  Pratos: <nome\_prato1>, <nome\_prato2>, ... |

**Histórico de transações realizadas no hotel**

|  |
| --- |
| Historico de Transacoes:  ==> Nome: <nome\_hospede> Gasto: R$<total\_gasto> Detalhes: <detalhe>  ==> Nome: <nome\_hospede> Gasto: R$<total\_gasto> Detalhes: <detalhe>  ==> Nome: <nome\_hospede> Gasto: R$<total\_gasto> Detalhes: <detalhe>  ...  ===== Resumo de transacoes =====  Lucro total:R$<soma\_lucros>  Total de transacoes:<quantidade\_transacoes>  Lucro medio por transacao: R$<soma\_lucro / quantidade\_transacoes> |

**Resumo completo do estado do hotel**

|  |
| --- |
| ======================================================  Cadastro de Hospedes: <total\_hospedes> hospedes registrados  ==> Hospede 1:  Email: <email\_do\_hospede>  Nome: <nome\_do\_hospede>  Data de nascimento: aaaa-mm-dd  ==> Hospede 2:  Email: <email\_do\_hospede>  Nome: <nome\_do\_hospede>  Data de nascimento: aaaa-mm-dd  …  ======================================================  Menu do Restaurante: <total\_itens> itens no cardapio  ==> Item 1:  Nome: <nome> Preco: R$<preco>  Descricao: <descricao>  ==> Item 2:  Nome: <nome> Preco: R$<preco>  Descricao: <descricao>  Pratos: <nome\_prato1>, <nome\_prato2>, …, <ultimo\_prato>  ======================================================  Historico de Transacoes:  ==> Nome: <nome\_hospede> Gasto:R$<total\_gasto> Detalhes:<detalhe>  ==> Nome: <nome\_hospede> Gasto:R$<total\_gasto> Detalhes:<detalhe>  ==> Nome: <nome\_hospede> Gasto:R$<total\_gasto> Detalhes:<detalhe>  ...  ===== Resumo de transacoes =====  Lucro total:R$<soma\_lucros>  Total de transacoes:<quantidade\_transacoes>  Lucro medio por transacao: R$<soma\_lucro / quantidade\_transacoes> |

**OBS: Esteja atento aos seguintes aspectos de seu projeto:**

* O restaurante deve fornecer seu menu utilizando a útilma ordenação definida pelo administrador do sistema.
* Qualquer operação para imprimir datas deve seguir o formato: aaaa-mm-dd, por exemplo:
  + Ex: 2016-09-13
* Crie um diretorio no root de seu sistema para organizar os arquivos gerados pelo sistema.
  + Ex: arquivos\_sistema/
* Organice os relatórios produzidos em só um diretório:
  + Ex: arquivos\_sistema/relatorios/
* Use caminhos relativos, independente do seu sistema
* Use a seguinte padronização para os arquivos gerados:
  + cad\_hospedes.txt
  + cad\_restaurante.txt
  + cad\_transacoes.txt
  + hotel\_principal.txt

# Caso de Uso 9: Persistência do sistema

**User Story**: Como sistema desejo armazenar o estado atual do sistema para poder recuperar este estado caso o sistema seja desligado

Para essa funcionalidade o sistema deve **salvar e carregar os seus dados** para garantir a persistência dos dados ao finalizar o programa. Diante disso, implemente a lógica dos métodos iniciaSistema e fechaSistema na Façade para salvar o seu sistema em um arquivo nomeado **“hug.dat”**. Esse arquivo deve estar em um diretório chamado **“arquivos\_sistema/”.** Utilize os conceitos de fluxos para salvar e carregar os objetos do seu sistema e garantir que eles são restaurados toda vez que o programa é, respectivamente, iniciado e finalizado.