

TRABALHO TÓPICOS ESPECIAIS I

A turma desenvolverá **um** sistema de uma pizzeria. Cenário único para todo o período.

Cenário:

Um amigo seu adquiriu recentemente uma pizzeria. O movimento começou devagar, mas, atualmente, pode-se dizer que está “bombando”. Quase toda noite a pizzeria está lotada. Tanto que ele contratou dois novos funcionários: um pizzaiolo e um garçom. Ele atribui esse sucesso a qualidade das pizzas, que são muito saborosas e ao horário de atendimento (veja quadro a seguir). Muitas pessoas que estão voltando para casa, seja do serviço, seja do lazer, frequentam a pizzeria.

Quadro 1: Horário de funcionamento da pizzeria.

domingo	18:00–02:00
segunda-feira	18:00–02:00
terça-feira	18:00–02:00
quarta-feira	18:00–02:00
quinta-feira	18:00–04:00
sexta-feira	18:00–05:00
sábado	18:00–05:00

Fonte: próprio autor

Hoje em dia, a pizzeria não é informatizada, exceto pela emissão do cupom fiscal, que obedece a Legislação de Automação Comercial do Varejo vigente no Espírito Santo (Decreto 1.090-R de 25/10/2002 e alterações posteriores). Além disso, a pizzeria possui as chamadas “maquininhas de cartão” para aceitar pagamentos com cartão de crédito ou débito. O seu amigo ainda está pensando em adotar pagamentos com carteiras digitais como PicPay©.

Seu amigo quer informatizar sua pizzeria. Quando você perguntou o que ele quer informatizar, ele deu a resposta clássica: **“TUDO”**. Então, você explicou como as coisas funcionam e ele decidiu contratá-lo para desenvolver um sistema de informação para gerir a pizzeria dele. Vocês se reuniram e conversaram por 4 horas seguidas. Ele faz questão de remunerar seu serviço, independente do vínculo de amizade de vocês. Você obteve uma série de informações. Dentre elas as seguintes:

1. A pizzeria não possui entrega, atualmente. Os clientes fazem seus pedidos e esses são associados à mesa a que eles estão. No cardápio, há, além das pizzas, vários tipos de bebidas: cerveja, refrigerante, suco de laranja e goiaba (de poupa), água mineral.
2. A pizzeria possui vários sabores de pizza: calabresa, atum, quatro queijos e mais 37 sabores. Para cada sabor, há quatro tipos de tamanho: B – Brotinho, M – Média, G – Grande e GG – Gigante. O cliente pode pedir várias pizzas em um mesmo pedido. Contudo, ele só poderá pedir dois sabores por pizza. Nesse caso, ao contrário de outras pizzarias, não será cobrado o valor do sabor da pizza mais

cara. Será cobrado o valor de cada metade. No mesmo pedido, o cliente também pede a bebida. Em geral, é um pedido por mesa, até a conta ser fechada. Porém, às vezes, há comemorações, e os clientes juntam mesas. Nesse caso, será escolhida **uma** mesa para que seja associada ao pedido e, posteriormente, à conta.

3. Cada pizza, respeitando seu tamanho, consome ingredientes. É fundamental dar baixa na quantidade dos ingredientes assim que a pizza ficar pronta. Isso porque para cada ingrediente há o registro da quantidade (em determinada unidade) e o ponto de reposição. O sistema deve emitir um alerta para o funcionário responsável pela compra de mercadorias assim que a quantidade de um determinado ingrediente atingir o ponto de reposição. O cálculo do ponto de reposição, as compras e as entradas de mercadorias não estão no escopo do sistema que será desenvolvido.

Como não há entrega, não há necessidade de cadastrar dado nenhum dos clientes. Contudo, o seu amigo está pensando em contratar o iFood®. Na verdade, precipitado que é, já contratou, só não está usando ainda. Você observou que o iFood, através do aplicativo “Gestor de Pedidos”, permite alguma integração com o sistema corporativo do restaurante contratante. Seu amigo quer todas. Veja o que ele quer integrar:

1. **Da pizzeria para o iFood:** o cardápio da pizzeria para não ser necessária digitação inicial e nem alteração de produtos e preços.
2. **Do iFood para pizzeria:** Os dados do cliente: nome, telefone e endereço; dados do pedido: identificação do cliente, itens pedidos, com quantidades, valores e detalhes (exemplo: “pizza de atum, *com borda grossa, sem coentro*”) e forma de pagamento. Esses dados devem ser alimentados no sistema a ser desenvolvido para que sejam analisadas informações para tomada de decisão sobre promoções, descontos ou compra de materiais. Dados como bairros dos clientes, preferências por bebida são bastante úteis para definir estratégias de venda.
3. **Do iFood para pizzeria:** Tempos para cada fase do processo do pedido. É interessante armazenar no novo sistema os intervalos de tempo entre as fases de um pedido feito via iFood, até mesmo para reproduzir esse processo com os pedidos presenciais. O iFood trabalha com a data e hora do pedido (quando o pedido foi feito pelo cliente via aplicativo) e com um prazo de entrega (“*Entregar até 12:00*”). Além disso, o software “Gestor de Pedidos” trabalha com 6 situações: (i) Pendente; (ii) Entregador a caminho, chega em **x** minutos; (iii) Em preparo; (iv) Entregador chegou; (v) Despachado; (vi) Concluído. É necessário que o novo sistema armazene o tempo do início do pedido e quando o pedido foi despachado. Obter o tempo médio de preparo de cada pizza é importante.

No final da história, o dono da pizzeria quer que os controles sejam os mesmos, seja via pedido presencial, seja via iFood.

Cada grupo desenvolverá um artefato a cada sprint até o fim do período. Seguem os artefatos obrigatórios:

1. Project Management Canvas (por exemplo: PM Canvas). (Reunião de Sprint: **15/03/2022**)
2. Levantamento de Requisitos. (Reunião de Sprint: **29/03/2022**)
3. Especificação dos Requisitos: diagrama de casos de uso com descrição de casos de uso (englobando desenho das interfaces - telas). Não precisará especificar todos os casos de uso. (Reunião de Sprint: **12/04/2022**)
4. Modelo de Dados: Diagrama de Entidades e Relacionamentos, com dicionário. Gerar script de banco de dados, a partir do diagrama (por exemplo, usando *SQLite* e *DB Designer Fork*). O *SQLite* pode ser uma boa escolha por ter uma interface (*SQLite Studio*) que você não precisa configurar nada para usar. (Reunião de Sprint: **26/04/2022**)
5. Diagrama de Classes, podendo ser a partir do Modelo de Dados (o contrário do ORM - Modelagem Objeto-Relacional). O inverso também pode ser usado, ou seja, desenvolver o diagrama de classes e depois o modelo relacional. (Reunião de Sprint: **10/05/2022**)
6. Diagrama de estados, pelo menos para o requisito de gestão dos pedidos. (Reunião de Sprint: **24/05/2022**)
7. Geração de um CRUD do cardápio, por exemplo, usando Visual Studio, ASP Net Core MVC/C#, com *Scaffolding* e *DB First* e um conjunto de testes automatizados utilizando alguma ferramenta (Selenium, Cucumber, Visual Studio Test Professional, etc). (Reunião de Sprint: **14/06/2022**)
8. Apresentação final do sistema: **28/06/2022**.

Condições de entrega

O trabalho deve ser feito em grupos de no mínimo 3 e no máximo 4 componentes. As ferramentas e linguagens de programação (pilha tecnológica) que irão apoiar o desenvolvimento devem ser registradas na documentação final do projeto, mas são de livre escolha. Ao final de cada aula, o grupo deverá subir a versão do trabalho através da ferramenta de envio de atividades disponibilizada no AVA. O trabalho deve compilar e executar corretamente até o final do semestre.

Critérios de avaliação

Os trabalhos serão avaliados em duas etapas:

- Avaliação objetiva dos artefatos entregues no AVA, de 0 a 10 pontos.
- Avaliação subjetiva (entrevistas), de 0 a 10 pontos.

Observações finais

- Aproveite a oportunidade para praticar, portanto, não copie código da web ou do colega.

- Copiar não é indicado, mas a discussão com outros alunos, a análise de outros códigos e a solicitação de apoio para tirar dúvidas é permitido e incentivado.
- Se houver a identificação de plágio/cópia de código e artefatos de outros grupos ou de trabalhos disponibilizados na Internet, o grupo será penalizado com a perda total dos pontos.
- Quaisquer questões não contempladas por este documento devem ser levadas ao professor para esclarecimentos.