



Bootcamp – Azure Service Bus





Um pouco sobre mim...

> Lucas Rocco Ferreira



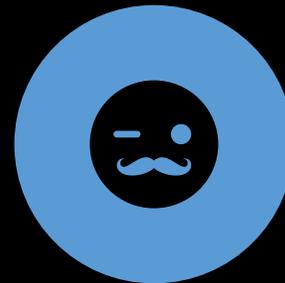
Consultor Cloud & Devops
pela Cloud Motion.



Desenvolvedor há quase 20
anos, tendo como principal
tecnologia o Microsoft .Net,
C# e Asp Net.



Ultimamente, um entusiasta
no desenvolvimento e
arquitetura para a nuvem com
Azure Cloud e Azure Devops.



Músico Guitarrista, Fã de
games e animes. Sem aquela
velha opinião formada sobre
tudo!



Agenda

- Perfil da Apresentação. Qual perfil terá melhor aproveitamento?
- Alguns Cenários Comuns. Porque utilizar o Azure Service Bus?
- Sobre o Azure Service Bus. (Conceito)
 - Namespaces
 - Queues
 - Topics / Subscriptions
 - BrokeredMessages
 - Outros recursos (Sessions(FIFO), Autoforwarding, Dead-letter queue, Scheduled delivery, etc.)
- Lab: Utilizando o Azure Service Bus.
- Algumas Métricas
- Preço (Azure Pricing for Service Bus)
- Referências e Contato.
- Duração estimada em 40 minutos. Questões serão respondidas após a apresentação.



Perfil da Apresentação. Qual perfil terá melhor aproveitamento?

- Desenvolvedores e arquitetos que estão iniciando ou já tem alguma experiência com Broker de mensagens.
- Tenha algum conhecimento em Azure Cloud, Microsoft .Net e C#. Os labs serão demonstrados nessas tecnologias, utilizando o Azure SDK.
- Tenha objetivo de desenvolver e utilizar os serviços na nuvem com o Azure Cloud.
- Para desenvolvedores de outras tecnologias que podem utilizar o protocolo AMQP (Advanced Message Queuing Protocol)



Alguns Cenários Comuns. Porque utilizar o Azure Service Bus?

- Poder escalar a aplicação e evitar *overhead* de processamento.
- Poder ter alta disponibilidade e garantir o envio/recebimento com operações síncronas/assíncronas.
- Poder ser distribuído. Comunicar com outras aplicações.
- Poder ser mais seguro. Definição de políticas de autorização e acesso.
- Com a infraestrutura na nuvem. Não se preocupar com *Backup*, *Disaster Recovery* e disponibilidade do serviço.
- Não depender de um infraestrutura On-Premises que precisa ser gerenciada, se preocupar com gerenciamento de servidores, disponibilidade, backup, etc.



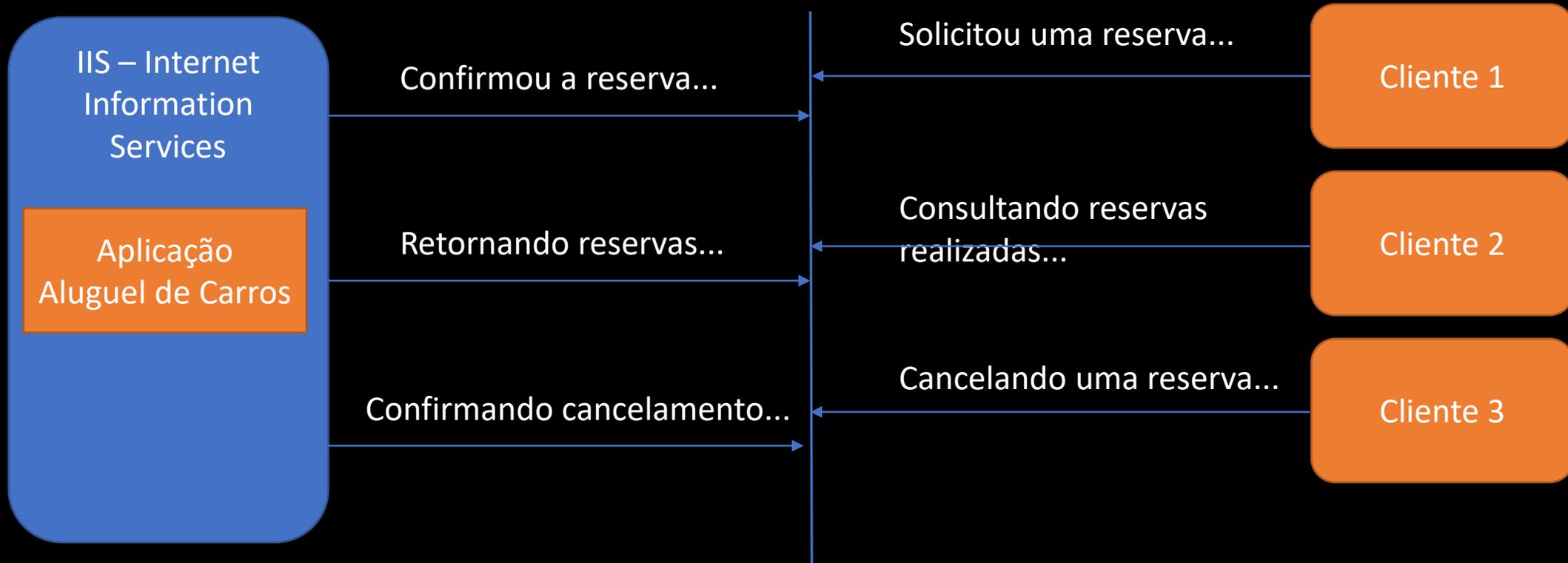
Alguns Cenários Comuns. Porque utilizar o Azure Service Bus?

- Cenário comum. Sistema de gerenciamento de reservas para automóveis.
- Precisa receber os pedidos do cliente que efetua a reserva do veículo.
- Precisa notificar o cliente sobre os veículos disponíveis, reservas efetuadas ou não com sucesso. Outras mensagens: Promoções, ofertas, multas, etc.
- Não depender do pool de conexões do gerenciador do site. (IIS, Apache, etc).
- O cliente escolhe quais serviços deseja se inscrever para recebimento de informações.



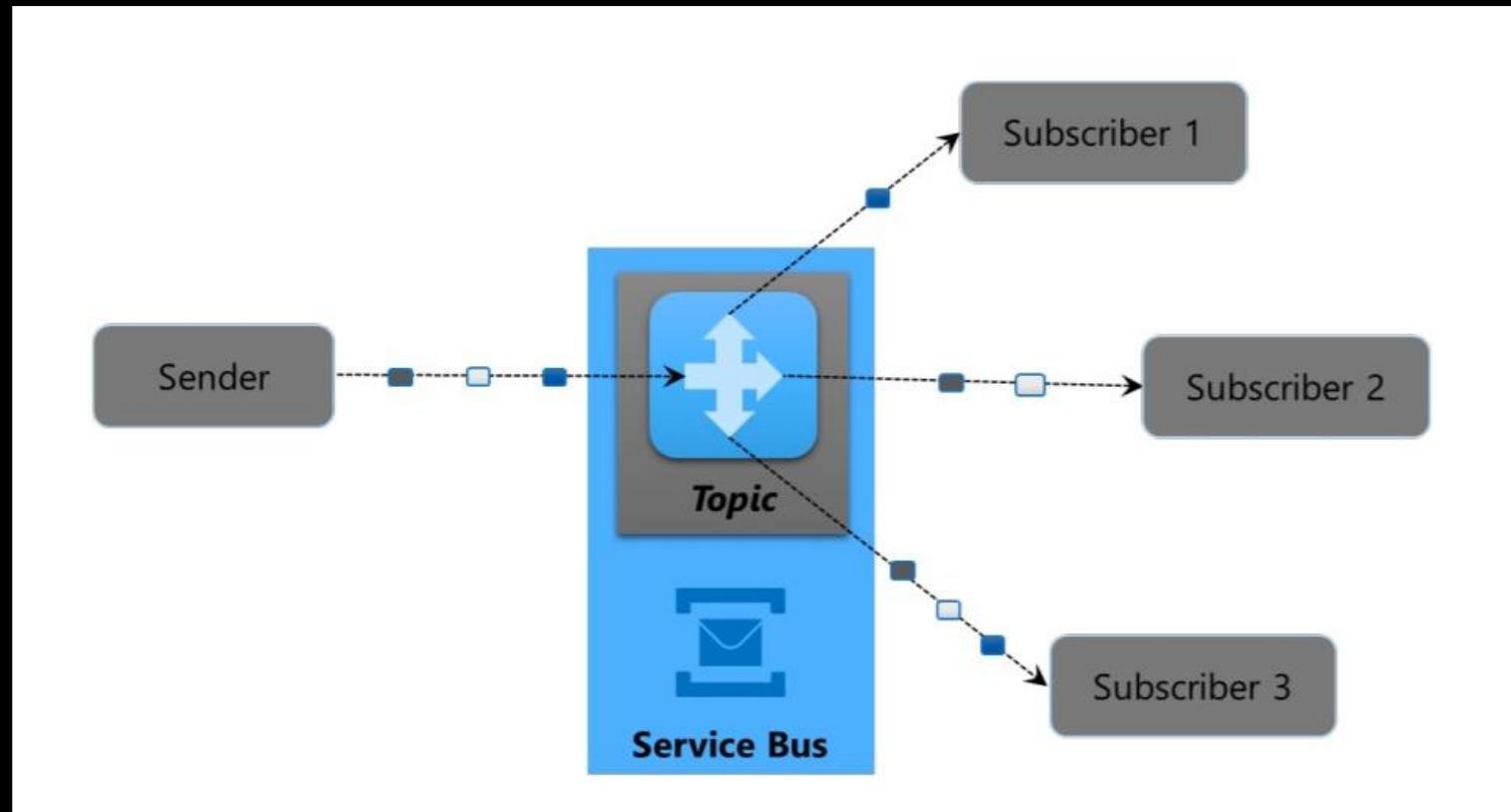
Alguns Cenários Comuns. Porque utilizar o Azure Service Bus? Modelo Tradicional – On-Premises

I/O – Input/Output





Alguns Cenários Comuns. Porque utilizar o Azure Service Bus? Modelo Service Bus – Azure Integration Services

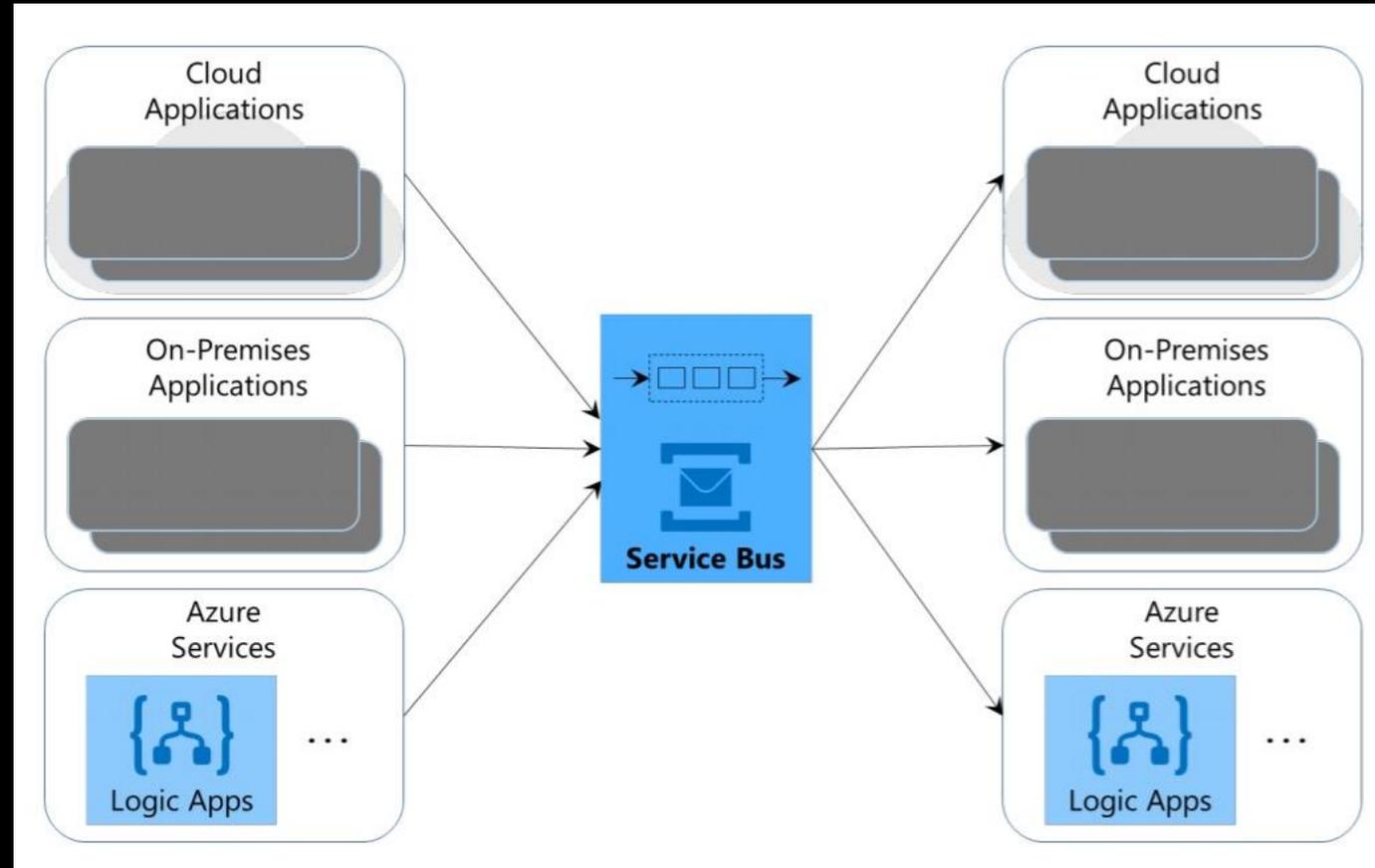




Sobre o Azure Service Bus

Modelo Service Bus – Azure Integration Services

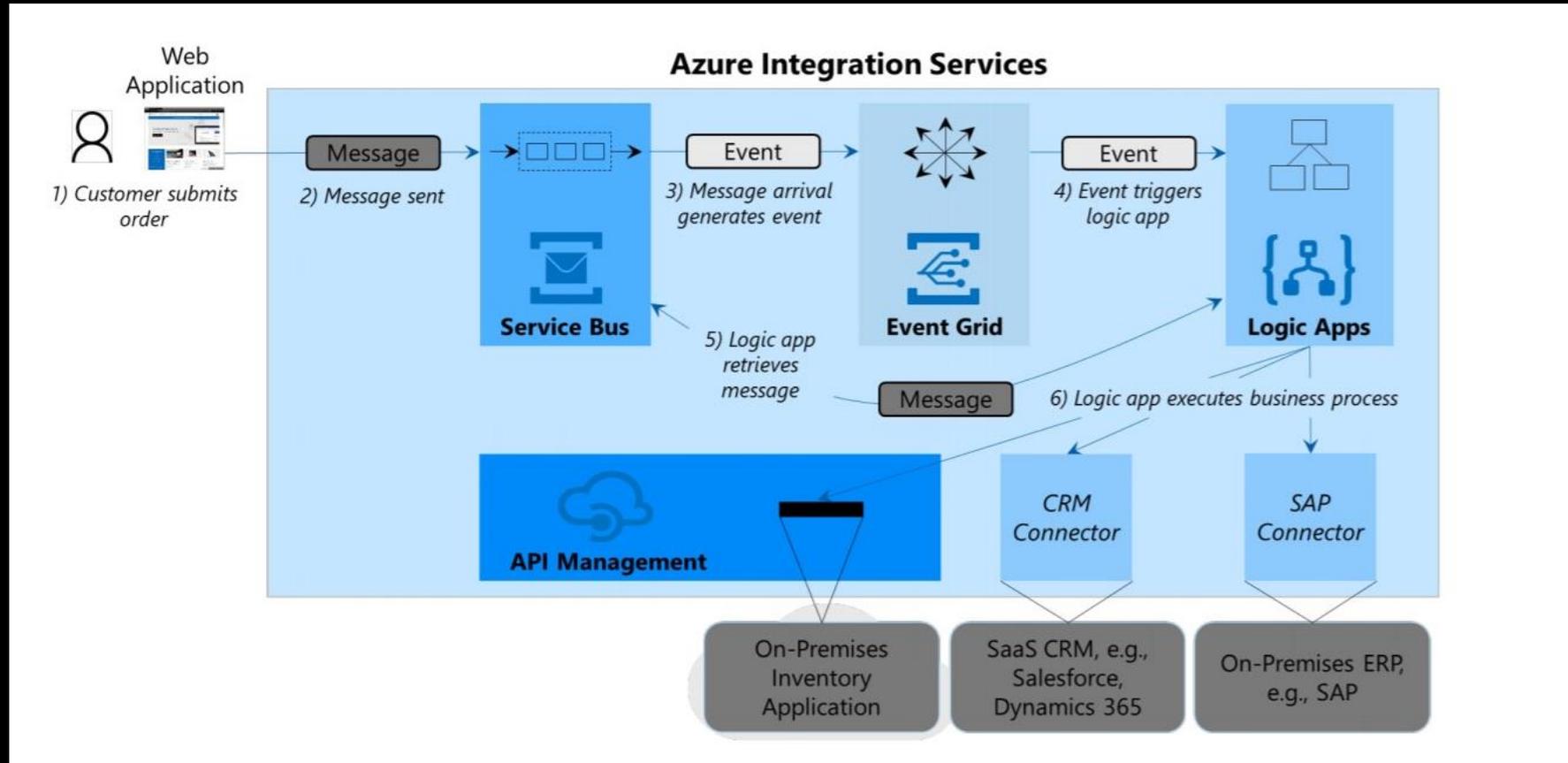
- Namespaces
- Queues
- Topics
- Subscriptions



Sobre o Azure Service Bus

Modelo Service Bus – Azure Integration Services

Message:
- JSON Document



Sobre o Azure Service Bus Namespaces

- Um namespace é um contêiner para todos os componentes de mensagem. Vários tópicos e filas podem estar em um único namespace, e os namespaces geralmente servem como contêineres de aplicativos.
 - Queues: As mensagens são enviadas e recebidas a partir de *filas*. As filas armazenam mensagens até que o aplicativo de recebimento esteja disponível para recebê-las e processá-las. As mensagens em filas são ordenadas e recebem carimbo de data/hora na chegada. Quando aceita, a mensagem é mantida protegida em armazenamento com redundância. As mensagens são entregues em modo *pull*, e somente quando solicitado.



Sobre o Azure Service Bus Namespaces

- Topics: Também é possível usar *tópicos* para enviar e receber mensagens. Enquanto uma fila é frequentemente usada para comunicação ponto a ponto, os tópicos são úteis em cenários de publicação/assinatura.





Sobre o Azure Service Bus Namespaces

- **BrokeredMessage:** Estrutura que possui propriedades e definem a característica da mensagem a ser transmitida.
- Principais propriedades:
 - Body: Corpo da mensagem. Seu conteúdo.
 - ContentType: Tipo do conteúdo. Exemplo: application/json
 - Label: Uma forma amigável de identificação da mensagem
 - LockToken: Token associado a mensagem.
 - MessageId: Id da mensagem definida pelo usuário
 - State: Identifica qual o estado da mensagem
 - TimeToLive: Quanto tempo a mensagem irá ficar armazenada após expirar. Após esse tempo, ela não poderá ser recuperada.
- Principais métodos:
 - CompleteAsync() – Indica que a mensagem foi recebida e a marca para exclusão
 - AbandonAsync() – Se houver uma operação cancelada, utilize para cancelar o lock da mensagem.
 - DeadeLetterAsync() – Move a mensagem para um outra fila de mensagens descartadas.
 - Clone() – Copia a mensagem, permitindo enviar como uma nova mensagem.



Sobre o Azure Service Bus Segurança

- **Authentication / Authorization:**
 - SAS Token: Define as permissões para a entidade. (Namespace, Queue, Topics)
 - Azure AD (RBAC – Role based access control). Sem necessidade de armazenar credenciais na aplicação. Até mesmo as permissões SAS podem ser gerenciadas pelo Azure AD.
 - Oauth 2.0. Acesso por token. Precisa registrar a aplicação com Azure AD tenant e ter o id de cliente.
 - Client Secrets: A aplicação precisa ter uma secret para poder solicitar o token.
 - Exemplos: GitHub – Role based access sample
- **Geo Outage/Disaster Recovery:** Quando um datacenter ou região apresentar instabilidade, o Azure replica para outra, para manter os serviços funcionando. Em casos de desastre natural, pode ser que nem todos os dados possam ser recuperados.
- **Storage:** Dados armazenados no Azure Storage e SQL Azure.



Sobre o Azure Service Bus Recursos Avançados

- Message Sessions: Usa o modelo FIFO(First-in, First-out) para garantir a entrega ordenada das mensagens.
- Autoforwarding: Encaminha mensagens para outras filas e tópicos.
- Dead-letter: Mensagens que não puderem ser processadas, devido a uma exceção de aplicação ou serviços. São armazenadas numa sub-queue (DLQ) para inspeção do ocorrido.
- Scheduled delivery: Podemos dizer em que momento a mensagem será processada. Podem ser canceladas antes de serem enviadas e a mensagem é removida.
- Message deferral: Adiar o processamento da mensagem por algum tempo devido uma necessidade da aplicação. Como workflows e dependências de regras de negócio. A mensagem é posta em processo paralelo até sua execução.
- Transactions: Suporta a execução de operações dentro de um escopo. Suporta agrupamento dessas operações, como se fosse uma única mensagem. Pode ser uma queue, topic ou subscription.
 - `using (var ts = new TransactionScope(TransactionScopeAsyncFlowOption.Enabled))`
- Filtering and actions: Nas subscrições, podemos filtrar o que queremos ver. Filtros booleanos, correlacionados (olhando propriedades da mensagem) e Notação SQL.
 - Exemplo: `sys.To NOT IN ('Store1','Store2')`
- Duplicate detection: Detecta mensagens duplicadas (MessageId) e as descarta, enviando somente uma ao solicitante.



Lab: Utilizando o Azure Service Bus

- Queues:
 - Criar mensagens e enviar para uma fila.
 - Receber mensagens da fila e ler seu conteúdo.
- Topics:
 - Criar mensagens e enviar para um tópico.
 - Receber mensagens do tópico e de subscrições diferentes.



Algumas Métricas

Premium	Standard
High throughput	Variable throughput
Predictable performance	Variable latency
Fixed pricing	Pay as you go variable pricing
Ability to scale workload up and down	N/A
Message size up to 1 MB	Message size up to 256 KB



Preço (Azure Pricing for Service Bus)

- No portal de preços existe tabela explicativa para poder ter noções de gastos conforme a necessidade da sua empresa
- Basic: Apenas para efeito de testes de desenvolvimento
- Standard: Para empresas de pequeno e médio porte.
- Premium: Para empresas que querem um serviço altamente dedicado e recuperação de desastres. (Geo-Disaster Recovery).



Referências e Contato

- E-Mail: lucas.rocco@cloudmotion.com.br
- Lab Github: <https://github.com/lucasroccoCM/bootcamp-servicebus>
- **What is Azure Service Bus?**
 - De <<https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-messaging-overview>>
- **Filas, tópicos e assinaturas do Barramento de Serviço**
 - De <<https://docs.microsoft.com/pt-br/azure/service-bus-messaging/service-bus-queues-topics-subscriptions>>
- **Azure Service Bus client library for .NET**
 - De <<https://github.com/Azure/azure-sdk-for-net/blob/7793c0ed902c9e57b3e78a00af2332d200f11756/sdk/servicebus/Microsoft.Azure.ServiceBus/README.md>>
- **Azure Service Bus Samples**
 - De <<https://github.com/Azure/azure-service-bus/tree/master/samples/>>