

 **VEM SER**  
**DBC**

# Imobiliária Shinigamis

Paulo Sergio

Jonas de Lara

Ricardo Langbecker

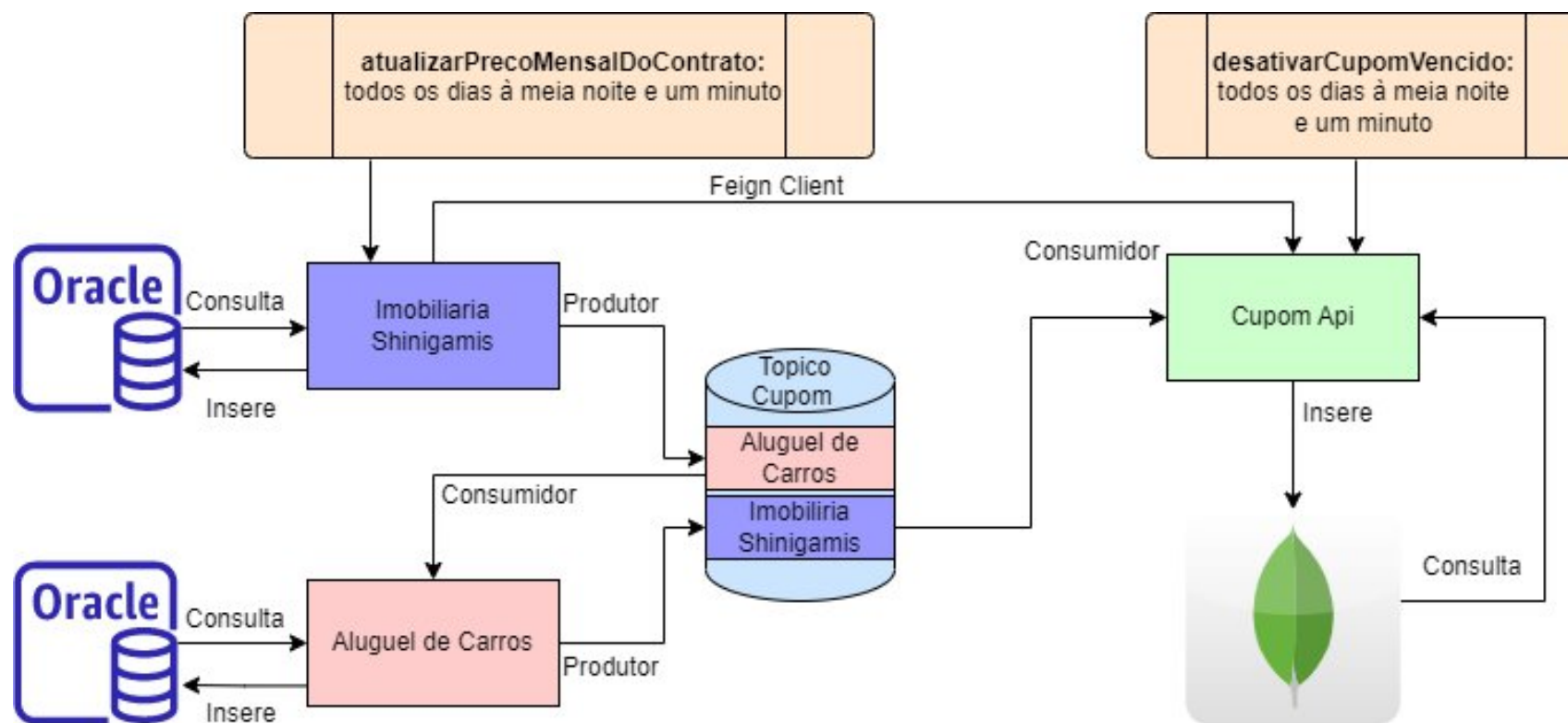
# Sobre a Imobiliária

O nosso projeto foi planejado para facilitar a vida tanto de pessoas que desejam colocar seus imóveis para alugar quanto para quem busca alugar um imóvel, ou seja, com esse sistema que criamos isso tudo se torna muito mais fácil.

# Principais Funcionalidades

- Geração de cupom no Kafka
- Consumo de cupom no Kafka
- Agendamento de tarefas (Scheduler)
- Atualização de valor mensal depois de 30 dias
- Desativar cupons vencidos

# Diagrama de arquitetura



# Diagrama de arquitetura

Temos a API-ImobiliariaShinigamis que produz cupons no tópico para a API-AluguelDeCarros.

Criamos a API-Cupom que consome os cupons no tópico criados pela a API-AluguelDeCarros e salvamos as informações no MongoDB.

A API-ImobiliariaShinigamis acessa a API-Cupom via Feign-Client para verificar se o cliente tem um cupom ativo para receber desconto no seu primeiro mês de contrato.

# Principais códigos

Método para verificar se já passou dos 30 dias aplicados pelo cupom e atualizar o valor mensal do aluguel para o valor sem desconto.

```
@Scheduled(cron = "0 1 * * * *")
public void atualizarPrecoMensualDoContrato(){
    List<ContratoEntity> contratos = contratoRepository.findAllByDataEntrada(LocalDate.now().minusMonths(1))
        .stream()
        .map(contrato -> {
            contrato.setValorAluguel(contrato.getImovel().getValorMensual() + contrato.getImovel().getCondominio());
            return contrato;
        }).toList();
}
```



# Principais códigos

Método para criar contrato onde verifica se o locatário tem um cupom disponível para gerar um desconto no primeiro mês de aluguel e desativa o cupom que foi utilizado.

```
@Override
public ContratoDTO create(ContratoCreateDTO contrato) throws RegraDeNegocioException, JsonProcessingException {
    ContratoEntity contratoEntityNovo = objectMapper.convertValue(contrato, ContratoEntity.class);
    ImovelEntity imovelEntity = imovelService.findById(contrato.getIdImovel());
    ClienteEntity clienteLocatario = clienteService.findById(contrato.getIdLocatario());

    double valorAluguel = imovelEntity.getValorMensal() + imovelEntity.getCondominio();

    CupomDTO cupomDTO = cupomApiClient.findByEmail(clienteLocatario.getEmail());
    if (cupomDTO != null && cupomDTO.isAtivo()) {
        contratoEntityNovo.setValorAluguel(valorAluguel - (valorAluguel * (cupomDTO.getDesconto() / 100)));
        cupomApiClient.desativarCupom(clienteLocatario.getEmail());
    } else {
        contratoEntityNovo.setValorAluguel(valorAluguel);
    }
}
```

# Principais códigos

Método para consumir os cupons no tópico do Kafka na partição 1 geradas pela AluguelDeCarros-API e salvamos estes cupons na nossa collection no MongoDB.

```
@KafkaListener(
    clientIdPrefix = "${spring.kafka.consumer.group-id}",
    groupId = "${spring.kafka.consumer.group-id}",
    topicPartitions = {@TopicPartition(topic = "${kafka.topic}", partitions = {"1"})}
)
public void consumirCupom(@Payload String mensagem) throws JsonProcessingException {
    CupomDTO cupomDTO = objectMapper.readValue(mensagem, CupomDTO.class);
    CupomEntity cupom = objectMapper.convertValue(cupomDTO, CupomEntity.class);
    cupomRepository.save(cupom);

    log.info("{}: {} {} {}", cupomDTO.getEmail(), cupomDTO.getDataCriacao().format(DateTimeFormatter.ofPattern("dd/MM/yyyy")),
        cupomDTO.getDataVencimento().format(DateTimeFormatter.ofPattern("dd/MM/yyyy")), cupomDTO.getDesconto());
}
```



# Principais códigos

Método para desativar os cupons agendados pelo Scheduler para todos os dias a meia noite e um minuto.

```
@Scheduled(cron = "0 1 * * * *")
public void desativarCupomVencido(){
    List<CupomEntity> cupons = cupomRepository.findAllByDataVencimento(LocalDate.now().minusDays( daysToSubtract: 1))
        .stream() Stream<CupomEntity>
        .map(cupom -> {
            cupom.setAtivo(false);
            return cupom;
        }).toList();
    cupomRepository.saveAll(cupons);
}
```

# Principais dificuldades

- Fazer a conexão via Feign-Client entre API-ImobiliariaShinigamis e API-Cupom
- Testar o Scheduler
- Testar os cupons criados pela equipe API-AluguelDeCarros
- Criar o diagrama de arquitetura com as corretas conexões



# Equipe

Paulo Sergio  
Jonas de Lara  
Ricardo Langbecker



# Obrigado!

# DBC

DIGITAL BUSINESS COMPANY®



 /dbc.company

 /dbccompany.com.br

 /company/dbc-company