

# Trabalho T3

## Engenharia de Software II

Data limite de entrega: 26/06/2024

Organização: grupos de no máximo 5 integrantes

Objetivo:

Implementar e implantar em uma plataforma de nuvem um sistema baseado em microsserviços de acordo com a descrição de negócio fornecida a seguir em uma linguagem e plataforma de escolha do grupo.

Descrição:

Nos dias de hoje é comum o surgimento de startups baseadas em aplicativos para celular. Muitas vezes, uma boa ideia se transforma em um aplicativo de sucesso, alavancando pequenas empresas para o sucesso rápido.

Considere uma startup que trabalha com o modelo de assinaturas. Neste modelo de negócio, o cliente pode baixar os aplicativos gratuitamente na loja, porém eles só funcionam se o cliente tiver uma assinatura paga. Em função disso, todo o mês, é necessário dispor de um sistema para manter o controle das assinaturas.

Neste modelo de negócio, os aplicativos, periodicamente, precisam verificar se a assinatura continua válida para poder continuar funcionando. Em função disso, o sistema tem de ser capaz de responder rapidamente se uma determinada assinatura continua válida. Quando um usuário assina um aplicativo, deve ser gerado um código (código da assinatura). Esse código, juntamente com o código do cliente, deve ser inserido no aplicativo de maneira que o uso do aplicativo seja liberado. De tempos em tempos, o aplicativo deve questionar se o par cliente/assinatura continua válido. Conforme a resposta, o aplicativo continua ativo ou não.

Para fazer toda esta gestão, startups que trabalham nesse modelo de negócio necessitam de um software de apoio que tenha as seguintes funcionalidades:

- Cadastrar/editar/listar a base de aplicativos que disponibiliza
- Cadastrar/editar/listar a base de clientes
- Cadastrar a assinatura de um aplicativo por parte de um cliente
- Atualizar o valor do custo mensal de um aplicativo
- Responder se um par cliente/assinatura continua válido
- Listar as assinaturas de um cliente
- Listar os assinantes de um aplicativo
- Receber a notificação de pagamento de uma assinatura (enviado pelo banco conveniado) e atualizar a data de validade da assinatura

O objetivo deste trabalho é desenvolver o módulo “backend” de um sistema de apoio para empresas que disponibilizem seus aplicativos neste modelo de negócio.

## Entidades de domínio

Considerando a descrição dos objetivos e funcionalidades do sistema foram definidas as seguintes entidades de domínio:

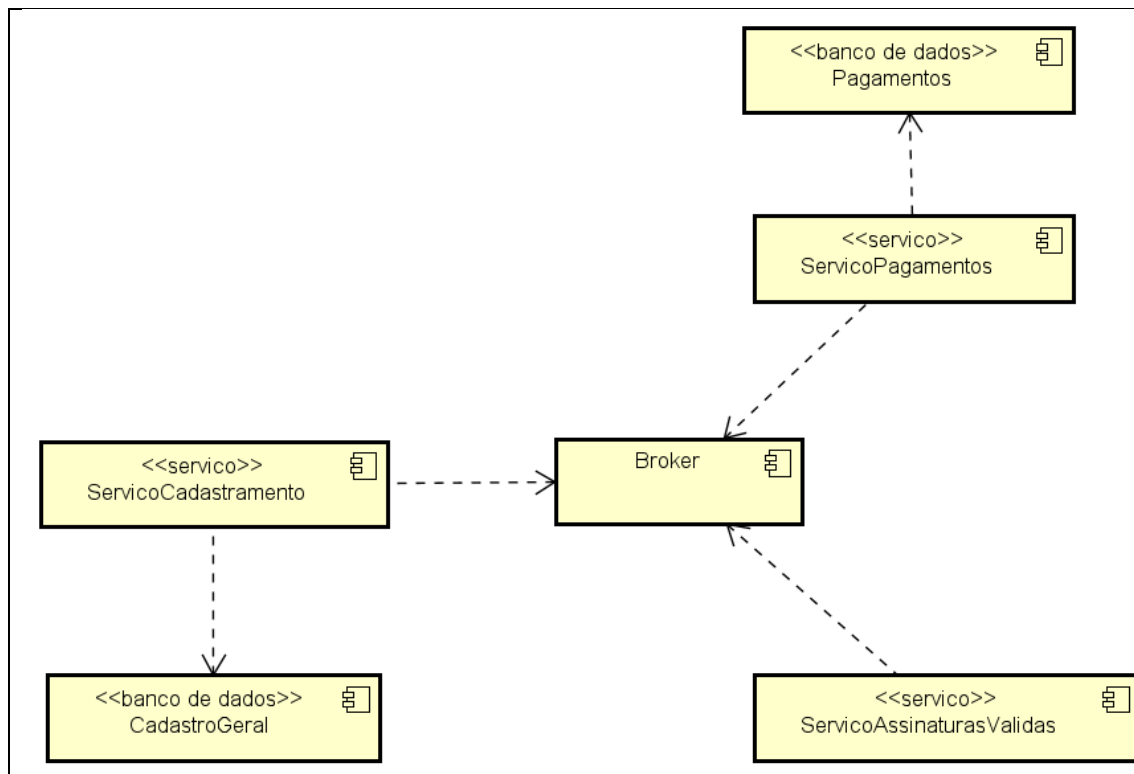
- Aplicativo: modela um aplicativo comercializado pela empresa no sistema de assinatura
- Cliente: pessoa interessada em assinar os aplicativos disponibilizados pela empresa
- Assinatura: modela a relação entre um aplicativo e um cliente
- Pagamento: modela o pagamento de uma determinada assinatura em um determinado mês
- Usuário: o usuário único que está usando o sistema para gerenciar os aplicativos que comercializa.

A tabela que segue detalha os atributos sugeridos de cada uma das entidades de domínio.

Entidade	Atributos		
Aplicativo	Atributo	Descrição	Tipo
	codigo	Código identificador do aplicativo	Inteiro longo
	nome	Nome fantasia pelo qual o aplicativo é conhecido	Texto
	custoMensal	Valor da assinatura mensal	Número de ponto flutuante
Cliente	Atributo	Descrição	Tipo
	codigo	Código identificador do cliente	Inteiro longo
	nome	Nome do cliente	Texto
	email	Email do cliente	Texto
Assinatura	Atributo	Descrição	Tipo
	codigo	Código da assinatura	Inteiro longo
	codApp	Código do aplicativo assinado	Inteiro longo
	codCli	Código do cliente	Inteiro longo
	inicioVigencia	Início da vigência da assinatura	Data
	fimVigencia	Fim da vigência da assinatura	Data
Pagamento	Atributo	Descrição	Tipo
	codigo	Identificador único do pagamento	Inteiro longo
	codAssinatura	Código da assinatura paga	Inteiro longo
	valorPago	Valor pago	Número de ponto flutuante
	dataPagamento	Data em que o pagamento foi efetivado	Data

## Arquitetura de microserviços

Visando explorar os aspectos desenvolvidos ao longo do curso, será proposta uma arquitetura baseada em microserviços. Um microserviço principal será responsável pelas seguintes funcionalidades: cadastramento e manutenção dos dados relativos a clientes, aplicativos e assinaturas. Um microserviço será responsável por manter o registro dos pagamentos efetuados (este serviço será notificado pelos bancos conveniados cada vez que um pagamento é efetuado). Por questões de performance, um terceiro microserviço será capaz de responder rapidamente para os aplicativos se uma determinada assinatura continua ativa ou não. A figura que segue apresenta um diagrama de componentes que procura detalhar essa arquitetura.



Arquitetura do sistema em nível de componentes

O microserviço chamado de “ServicoCadastramento” será o módulo principal do *backend*. Ele deverá ser o responsável por todas as operações de manutenção dos cadastros (clientes, aplicativos e assinaturas) bem como pelas operações relativas à cobrança, tais como, atualizar o preço das assinaturas, atualizar a data de validade, etc. Sempre que uma assinatura for cadastrada, o cliente ganha 7 dias grátis. A extensão do período de validade da assinatura se dá mediante o pagamento da mensalidade dentro desse período.

O microserviço “ServicoPagamentos” mantém uma base dados com todos os pagamentos efetuados. Sempre que receber uma notificação de pagamento, a mesma deve ser armazenada no banco e um evento assíncrono deve ser gerado para notificar os interessados de que um pagamento foi efetuado.

O microserviço “ServicoCadastramento” já é capaz de responder se uma determinada assinatura é válida ou não. Entretanto, como a demanda por esta informação é muito grande (todos os aplicativos periodicamente necessitam fazer essa consulta), foi projetado um microserviço adicional visando garantir a performance do sistema. Desta forma o microserviço “ServicoAssinaturasValidas” será responsável por responder, de forma rápida, se uma determinada assinatura é válida ou não. Ele será demandado, tipicamente pelos aplicativos, sempre que estes tiverem necessidade de confirmar se devem continuar respondendo ou se devem bloquear por falta de pagamento da assinatura. Cada vez que for demandado, este microserviço deve consultar sua “cache” interna verificando se já possui a informação relativa àquela assinatura. Caso não disponha, deverá perguntar para o “ServicoCadastramento” e, então, registrar em sua “cache” a informação para consultas futuras.

Tanto o “ServicoCadastramento” quanto o “ServicoAssinaturasValidas” consomem o evento que notifica que um pagamento foi efetuado. Com essa informação o “ServicoCadastramento”, usando suas regras de negócio, deverá atualizar a validade da assinatura. Já o “Servico AssinaturasValidas” deverá remover da sua “cache” a entrada correspondente à assinatura paga.

Desta forma, da próxima vez que for solicitado a respeito dessa assinatura, irá solicitar para o “ServicoCadastramento” a informação atualizada evitando inconsistências.

A seguir, iremos detalhar todos os endpoints sugeridos de cada um dos serviços listados anteriormente. Para cada endpoint será detalhada a rota, os parâmetros recebidos e o JSON resposta.

#### *ServicoCadastramento*

Por uma questão de dimensionamento de escopo, iremos omitir desse trabalho todos os endpoints relativos ao cadastramento/manutenção de clientes e cadastramento/manutenção de aplicativos. Para que o sistema possa ser testado a contento, deve ser previsto um script de inicialização capaz de popular o banco de dados com pelo menos 10 clientes e 5 aplicativos diferentes. Embora o sistema deva incluir os endpoints relativos ao gerenciamento de assinaturas, como será visto na sequência, o mesmo script deverá prever o cadastro prévio de pelo menos 5 assinaturas.

*Endpoint:* GET /servcad/clientes

Descrição	Lista com todos os clientes cadastrados
Parâmetros de entrada	Nenhum
JSON resposta	[{codigo,nome,email}, ... ]

*Endpoint:* GET /servcad/aplicativos

Descrição	Lista com todos os aplicativos cadastrados
Parâmetros de entrada	Nenhum
JSON resposta	[{codigo,nome,custo}, ... ]

*Endpoint:* POST /servcad/assinaturas

Descrição	Cria uma assinatura
Corpo da requisição	{código do cliente, código do aplicativo}
JSON resposta	Retorna o cadastro completo da assinatura {código assinatura, código cliente, código aplicativo, data de início, data de encerramento}

*Endpoint:* PATCH /servcad/aplicativos/:idAplicativo

Descrição	Atualizar o custo mensal do aplicativo
Corpo da requisição	{custo}
JSON resposta	Retorna o cadastro completo do aplicativo {código aplicativo, nome, custo mensal}

*Endpoint:* GET /servcad/assinaturas/{tipo}

Descrição	Retorna a lista com todas as assinaturas vigentes de acordo com o tipo indicado
Parâmetros de entrada	{tipo} = [TODAS ATIVAS CANCELADAS]
JSON resposta	[{código assinatura, código cliente, código aplicativo, data de início, data de fim, status}, ... ] Se a assinatura está ativa, a data de encerramento é posterior a data atual Status = [ATIVA CANCELADA]

*Endpoint:* GET /servcad/asscli/:codcli

Descrição	Retorna a lista das assinaturas do cliente informado
Parâmetros de entrada	codcli: código do cliente
JSON resposta	[{código assinatura, código cliente, código aplicativo, data de início, data de fim, status}, ... ] Status = [ATIVA CANCELADA]

*Endpoint:* GET /servcad/assapp/:codapp

Descrição	Retorna a lista de assinaturas de um aplicativo
Parâmetros de entrada	codapp: código do aplicativo
JSON resposta	[{código assinatura, código cliente, código aplicativo, data de início, data de fim, status}, ... ] Status = [ATIVA CANCELADA]

*Evento (observar):* PagamentoServicoCadastramento

Descrição	Observador do evento de pagamento
Corpo do evento	dia, mês e ano: compõem a data do pagamento codass: identificador da assinatura a que corresponde o pagamento valorPago: valor que foi pago pela assinatura na data indicada

### ServicoPagamentos

Este serviço possui uma base de dados própria usada única e exclusivamente para armazenar o registro de todos os pagamentos efetuados.

*Endpoint:* POST /registrarpagamento

Descrição	Solicita o registro de um pagamento
Parâmetros de entrada	dia, mês e ano: compõem a data do pagamento codass: identificador da assinatura a que corresponde o pagamento valorPago: valor que foi pago pela assinatura na data indicada
JSON resposta	Nenhum

*Evento (gerar):* PagamentoServicoCadastramento

Descrição	Evento de notificação de um pagamento efetuado para o ServicoCadastramento
Corpo do evento	dia, mês e ano: compõem a data do pagamento codass: identificador da assinatura a que corresponde o pagamento valorPago: valor que foi pago pela assinatura na data indicada

*Evento (gerar):* PagamentoServicoAssinaturaValida

Descrição	Evento de notificação de um pagamento efetuado para o ServicoAssinaturaValida
Corpo do evento	dia, mês e ano: compõem a data do pagamento codass: identificador da assinatura a que corresponde o pagamento valorPago: valor que foi pago pela assinatura na data indicada

*ServicoAssinaturasValidas*

Este microserviço deve ser capaz de manter uma “cache” de assinaturas válidas. Deve ser capaz de responder de forma síncrona se uma determinada assinatura permanece válida ou não. Como este serviço será demandado pelas instancias ativas dos aplicativos, espera-se que este módulo tenha alta demanda. Por essa razão deve ser programado como um microserviço do qual podemos ter várias instâncias.

*Endpoint:* GET /assinvalidas/:codass

Descrição	Retorna se a assinatura questionada permanece ativa
Parâmetros de entrada	codass: código da assinatura
JSON resposta	Booleano: true ou false

*Evento (observar):* PagamentoServicoAssinaturaValida

Descrição	Observador do evento de pagamento
Corpo do evento	dia, mês e ano: compõem a data do pagamento codass: identificador da assinatura a que corresponde o pagamento valorPago: valor que foi pago pela assinatura na data indicada

Entrega e Apresentação:

A entrega consiste nos seguintes elementos:

- Documento de texto contendo diagramas adequados sobre a modelagem realizada (diagrama de entidade-relacionamento para os bancos de dados, diagramas de componentes, diagramas de implantação);
- Documento de texto informando todos os passos necessários para implantação do sistema;
- Link para vídeo na plataforma Youtube (sugere-se um vídeo do tipo não-listado) demonstrando a execução do sistema em uma plataforma de nuvem;

- Código-fonte completo da solução final (IMPORTANTE: remova qualquer diretório que contenha os códigos binários para diminuir o tamanho do arquivo ZIP).

Observações:

- LEMBRETE: cópia de trabalhos é plágio, sujeito a processo disciplinar. Os trabalhos envolvidos em fraudes receberão nota 0,0 (zero).
- Dúvidas devem ser esclarecidas com o professor.
- Não serão aceitos trabalhos entregues além da data limite.
- Não serão aceitos trabalho entregues via correio eletrônico.