

Trabalho 1

Sistema de Controle de Assinaturas de Aplicativos

Enunciado geral

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 e alavancando pequenas empresas para o sucesso rápido.

Considere que 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.



Figura 1 – Ícones de aplicativos

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.

Considerando o prazo disponível para o desenvolvimento desta tarefa faremos uma simplificação importante: cada instância deste sistema irá atender uma única empresa e, por esta razão, ele terá apenas um único usuário cadastrado. Isso significa que todos os aplicativos e clientes cadastrados no sistema pertencem a uma única empresa/usuário.

Requisitos não funcionais

Como o objetivo deste trabalho é avaliar a capacidade do aluno de desenvolver um “backend” usando a arquitetura CLEAN, iremos definir na sequência o conjunto de entidades de domínio a serem utilizadas. Note que neste trabalho o “backend” deverá ser implementado como um monolito. No trabalho final da disciplina este mesmo código será refatorado para uma arquitetura de microserviços.

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 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
	aplicativo	Aplicativo de que trata a assinatura	Aplicativo
	cliente	Cliente de que trata a assinatura	Cliente
	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
	assinatura	Assinatura paga	Assinatura
	valorPago	Valor pago	Número de ponto flutuante
	dataPagamento	Data em que o pagamento foi efetivado	Data
	promocao	Código correspondente a promoção usada no pagamento. Pode ser um “none” no caso de não ser usado código nenhum.	String
Usuario	Atributo	Descrição	Tipo
	usuario	Identificador do usuário para login	Texto
	Senha	Senha de acesso do usuário	Texto

Regras adicionais

O sistema 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. Cada mensalidade paga estende a assinatura por mais 30 dias. A extensão do prazo se dá somando-se mais 30 dias ao prazo de validade atual ou mais 30 dias a partir da data de pagamento no caso de uma assinatura “cancelada” (com prazo de validade anterior a data corrente). Isso significa que para reativar uma assinatura basta pagar uma mensalidade.

Cada pagamento deve corresponder a uma mensalidade por vez. Se o valor do pagamento estiver correto, deve-se retornar a nova data de validade e o valor a ser estornado é zero. Se o valor estiver incorreto ou se a promoção informada for inexistente ou não for aplicável, a data

de validade retornada deve ser a mesma que já estava registrada e o valor pago deve ser retornado como sendo o valor que será estornado (devolvido).

O sistema deve manter, também, o registro de todos os pagamentos efetuados. Sempre que o sistema receber uma notificação de pagamento, o mesmo deve ser armazenado e a assinatura atualizada conforme o caso.

Note que as promoções podem ser tanto relativas ao prazo ganho a cada pagamento (tipo pague uma e ganhe 45 dias etc) ou quanto ao valor pago pela mensalidade (descontos etc).

Requisitos funcionais

Nesta seção iremos detalhar todos os endpoints correspondentes as funcionalidades indicadas anteriormente. Para cada endpoint será detalhada a rota, os parâmetros recebidos e o JSON resposta.

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 de inicialização do banco de dados deverá prever pelo menos 5 assinaturas ativas iniciais.

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: POST /servcad/aplicativos/atualizacusto/: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 confirme o tipo
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]

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	{status,data,valor estornado} Status = [PAGAMENTO_OK PROMOCAO_INVALIDA VALOR_INCORRETO]

Endpoint: GET /assinvalida/:codass

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