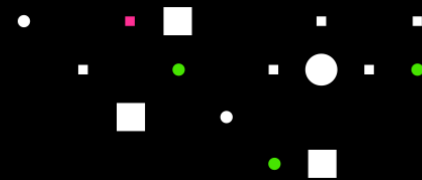


treinadev

# Projeto final

Etapa 1 – Turma 6

CAMPUS  
CODE



## Introdução

O projeto final desta etapa visa replicar em uma aplicação real os conhecimentos adquiridos no treinamento. Durante a execução do projeto, além da revisão de conteúdo, novas dicas e materiais serão enviados pela Campus Code, via Slack, para que você possa incrementar seu código.

Para tornar essa experiência mais fluida, o projeto será apresentado de forma geral no tópico abaixo e, em seguida, detalharemos os fluxos internos que devem ser codificados. Você pode (e deve) organizar seu trabalho em alguma plataforma como Trello ou GitHub Projects, assim ficará mais fácil acompanharmos sua linha de raciocínio. Fique à vontade para utilizar qualquer outro recurso como fluxogramas, *mocks* de telas etc.

O projeto deve ser construído de forma incremental, ou seja, esperamos que você atualize seu código a cada nova funcionalidade finalizada. Dessa forma, conseguimos acompanhar e proporcionar feedbacks mais pontuais que podem te ajudar na continuidade do trabalho. Então, ao **começar o projeto**, esperamos que você crie um repositório Git e compartilhe conosco desde o início para acompanharmos a evolução *commit a commit*.

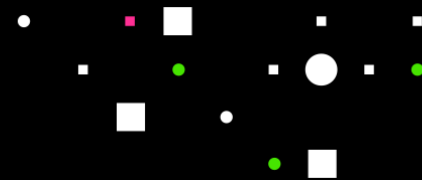
Ao final desse documento, você vai encontrar um tópico explicando as regras e restrições com mais detalhes.

## Prazo e entrega

O projeto deve ser concluído e entregue **até o dia 20 de junho de 2021, às 23h59**. Será considerado todo código submetido via Git até este horário, mas incentivamos que você continue evoluindo esse projeto como forma de desafio e reforço nos estudos.

O repositório Git deve ser criado logo no começo do seu trabalho, preferencialmente no primeiro dia, e deve ser compartilhado **via e-mail**. Envie o endereço do repositório na ferramenta que preferir (GitHub, GitLab, Bitbucket etc.) para o [treinadev@campuscode.com.br](mailto:treinadev@campuscode.com.br) com o assunto “Projeto Final – TreinaDev 06 - Etapa 01”

Se você criou um quadro no Trello ou utilizou qualquer outra ferramenta para organizar o projeto e deseja compartilhar conosco, envie o endereço na mesma mensagem.



## Sobre o projeto

Uma escola de programação, a CodePlay, decidiu lançar uma plataforma de cursos online de programação. Você já está trabalhando nesse projeto e agora vamos começar uma nova etapa: uma ferramenta de pagamentos capaz de configurar os meios de pagamentos e registrar as cobranças referentes a cada venda de curso na CodePlay. O objetivo deste projeto é construir o mínimo produto viável (MVP) dessa plataforma de pagamentos.

Na plataforma de pagamentos temos dois perfis de usuários: os administradores da plataforma e os donos de negócios que querem vender seus produtos por meio da plataforma, como as pessoas da CodePlay, por exemplo. Os administradores devem cadastrar os meios de pagamento disponíveis, como boletos bancários, cartões de crédito, PIX etc, especificando detalhes de cada formato. Administradores também podem consultar os clientes da plataforma, consultar e avaliar solicitações de reembolso, bloquear compras por suspeita de fraudes etc.

Já os donos de negócios devem ser capazes de cadastrar suas empresas e ativar uma conta escolhendo quais meios de pagamento serão utilizados. Devem ser cadastrados também os planos disponíveis para venda, incluindo seus valores e condições de desconto de acordo com o meio de pagamento. E a cada nova venda realizada, devem ser armazenados dados do cliente, do produto selecionado e do meio de pagamento escolhido. Um recibo deve ser emitido para cada pagamento e esse recibo deve ser acessível para os clientes finais, alunos da CodePlay no nosso contexto.

A seguir, estão detalhadas as funcionalidades básicas para o funcionamento da plataforma. Logo depois, são apresentadas funcionalidades extras que você pode codificar, caso tenha tempo ou até mesmo após o fim do prazo estabelecido para entrega do projeto, como forma de desafio.

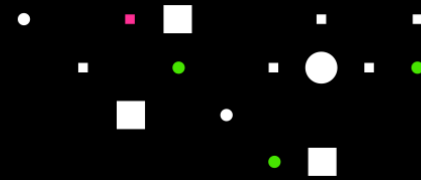
## Funcionalidades básicas

### Acesso de Administradores

Usuários de e-mail com domínio “@paynow.com.br” devem ser reconhecidos automaticamente como usuários com perfil administrativo. Os demais usuários devem ser considerados clientes e, com isto, devem possuir sempre um perfil da sua empresa configurado.

### Cadastro de Meios de Pagamento

Um usuário administrador autenticado pode gerenciar os meios de pagamento disponíveis na plataforma. Os meios de pagamento devem ser do tipo boleto bancário, PIX ou cartão de crédito. Cada meio de pagamento deve ter um nome, um ícone, uma taxa por cobrança em % e



uma taxa máxima em reais. Exemplos de meio de pagamentos: boleto bancário do banco laranja, com taxa de 5%; PIX do banco roxinho com taxa de 4,5%; cartão de crédito PISA com taxa de 6%.

Um meio de pagamento pode ser desativado por um administrador. Meios de pagamento desativados não devem ser disponibilizados para novas compras realizadas por qualquer cliente da plataforma.

## Acesso de Clientes

Um usuário pode criar sua conta como cliente usando um e-mail de sua empresa. Contas de e-mails públicos como Gmail, Hotmail e Yahoo não podem ser utilizadas nesta plataforma. No nosso contexto, um funcionário da CodePlay deve criar sua conta usando um e-mail com domínio “@codeplay.com.br”.

Ao criar sua conta, o cliente deve configurar os dados de sua empresa informando: CNPJ, razão social, endereço de faturamento e e-mail para faturamento. Um cliente não deve conseguir prosseguir com o acesso à plataforma enquanto não concluir o cadastro de todos os dados de sua empresa.

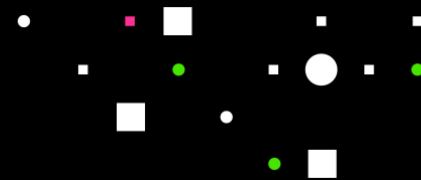
Após a conclusão do cadastro da empresa, outras pessoas da mesma empresa podem criar contas com o mesmo domínio de e-mail e ter acesso aos dados da empresa. O primeiro usuário, no entanto, deve ser identificado como administrador da empresa dentro da plataforma, com permissão para bloquear o acesso de outras pessoas da empresa.

## Token de Integração

Ao cadastrar uma empresa, os usuários desta empresa terão acesso a um token único alfanumérico de 20 caracteres que será utilizado para integração com a API de pagamentos. Este token deve ser disponibilizado no perfil da empresa, mas não deve ser passível de edição. O administrador da empresa pode solicitar um novo token que deve ser gerado automaticamente.

## Administração de Clientes

Um usuário administrador da PayNow autenticado na plataforma deve ser capaz de visualizar e alterar dados de clientes (as empresas), inclusive solicitar a renovação do token. Todas alterações de dados devem ser registradas na plataforma criando um histórico de modificações para fins de auditoria.



## Suspensão de Contas de Clientes

Um usuário administrador da PayNow autenticado na plataforma deve ser capaz de solicitar a suspensão de uma conta de cliente, impossibilitando tanto o acesso à plataforma como a emissão de novas cobranças para este cliente. A suspensão deve ser confirmada por outro usuário administrador da PayNow antes de entrar em vigor.

## Gestão de Meios de Pagamento (Cliente)

Um usuário de uma empresa na plataforma deve configurar os meios de pagamento que deseja disponibilizar para os seus clientes. Exemplo: uma empresa de e-commerce de sapatos pode criar sua conta na plataforma e habilitar somente pagamentos via cartão de crédito, enquanto nossa empresa de cursos de programação pode habilitar pagamentos via PIX e boleto bancário, excluindo a opção de cartão de crédito.

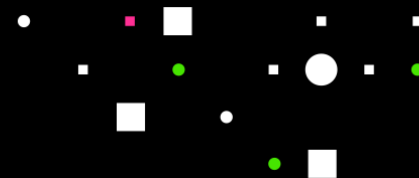
Para cada meio de pagamento escolhido, a empresa deve ter dados de configuração próprios conforme a tabela abaixo:

Meio de Pagamento	Dados Configurados	<i>Observação: os códigos de bancos devem ser os mesmos indicados pela FEBRABAN.</i>  <a href="https://portal.febraban.org.br/pagina/3164/12/pt-br/associados">https://portal.febraban.org.br/pagina/3164/12/pt-br/associados</a>
PIX	Chave PIX de 20 caracteres alfanuméricos e o código do banco	
Cartão de Crédito	Código alfanumérico de até 20 caracteres referente à conta da empresa junto à operadora de cartão	
Boleto bancário	Código do banco, número da agência e da conta bancária	

Então, caso a CodePlay deseje receber pagamentos via cartão de crédito PISA, ao ativar este meio de pagamento deverá informar o código de 20 caracteres. Para habilitar outro tipo de cartão, por exemplo o MESTRECARD, deverá informar outro código de 20 caracteres. Estes códigos representam identificadores únicos dos contratos da CodePlay com as operadoras PISA e MESTRECARD, estes contratos não estão dentro do escopo deste projeto.

## Cadastro de Produtos (Cliente)

Um usuário de uma empresa pode cadastrar produtos informando seus valores para venda. Os produtos devem possuir um nome, seu valor e possíveis descontos de acordo com o meio de pagamento escolhido. Em caso de mudança nos valores, deve ser armazenado um histórico com os preços anteriores para efeito de auditoria. Cada produto terá um token gerado



automaticamente após o cadastro. Este token deve ser utilizado para efetivar contratações através da API.

## API de Criação de Token de Cliente Final (Cliente do cliente)

A plataforma de pagamento deve armazenar os dados dos clientes finais de seus clientes. Exemplo: Eu, João, ao criar minha conta na plataforma de cursos online devo ter um token único criado pela plataforma de meios de pagamentos PayNow. Este token será utilizado para referenciar todas as cobranças emitidas para este cliente.

O token do cliente final deve ser vinculado ao nome completo e CPF de cada cliente e seu formato é alfanumérico com 20 caracteres e único. Caso a mesma pessoa crie contas em diversos clientes da PayNow, seu token deve ser sempre o mesmo, mas deve ser registrado que este token foi vinculado com uma nova empresa para efeito de auditoria.

A criação deste token deve ser feita através de uma API e o token da empresa cliente deve ser utilizado para identificar a qual empresa pertencem os dados do cliente final.

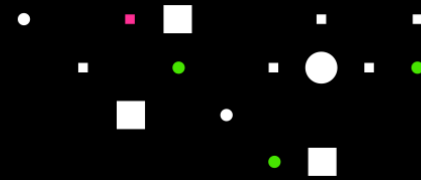
## API para emissão de cobrança

A plataforma de pagamento pode receber uma solicitação para emitir uma nova cobrança avulsa via API. Essa solicitação deve incluir o token da empresa, o token do produto envolvido na transação, o meio de pagamento escolhido e os dados do cliente final para emissão da cobrança. Cada transação deve conter somente 1 token de produto, ou seja, a plataforma não suporta contratações de múltiplos produtos simultaneamente.

De acordo com o meio de pagamento escolhido para emissão da cobrança devem ser recebidos parâmetros adicionais conforme a tabela abaixo:

Meio de Pagamento	Parâmetros Adicionais
Cartão de Crédito	Número do cartão, nome impresso no cartão, código de verificação
Boleto	Endereço completo para envio do boleto
PIX	Nada

Os parâmetros adicionais, caso existam, são sempre obrigatórios.



Durante a emissão da cobrança, deve ser aplicado o desconto de acordo com o meio de pagamento escolhido, caso tenha sido configurado no cadastro do produto. A cobrança deve registrar o valor original do pedido e o valor com desconto. Cada cobrança emitida deve possuir um token de identificação alfanumérico único de 20 caracteres e o status inicial da cobrança é pendente.

## Confirmação Manual de Pagamentos

Na plataforma de pagamentos, um usuário administrador pode consultar todas cobranças com status pendente e modificar este status para “aprovada” ou “rejeitada”. No caso de aprovada deve ser informada a data efetiva do pagamento. No caso de rejeitada, devem ser informados: a data da tentativa do pagamento, o código de retorno informado pela operadora/banco (tabela disponível ao final do documento).

Uma cobrança rejeitada deve seguir no status “pendente”, para que novas tentativas de cobrança sejam realizadas, mas mesmo assim devem ser armazenadas as informações da tentativa de cobrança e todas estas tentativas devem ser visíveis ao consultar uma cobrança no sistema.

## Emissão de Recibos

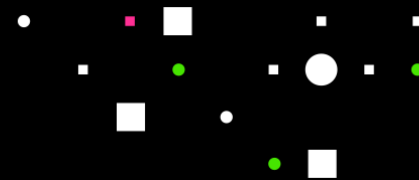
Toda cobrança deve emitir um recibo quando seu status for alterado para “aprovada”. O recibo deve ser acessível mesmo por usuários externos não autenticados no sistema. O recibo deve conter a data de vencimento da cobrança, a data efetiva do pagamento e um código de autorização enviado pela empresa/banco responsável pela confirmação de pagamento.

## API para consulta de cobranças

O sistema de pagamento deve possuir uma API para consulta de cobranças permitindo filtrar por data de vencimento e por forma de pagamento. A API deve permitir também a atualização de status de uma determinada cobrança. Esta API será utilizada pelo sistema de execução de cobranças e todas as requisições devem ser autorizadas por um token único de autenticação.

## Cliente consulta cobranças

As empresas cadastradas na plataforma de pagamentos podem acessar suas contas e verificar o status das cobranças relacionadas à sua empresa. A tela ‘inicial’ desta funcionalidade deve trazer as cobranças referentes aos últimos 30 dias e devem existir opções para visualizar os últimos 90 dias e o histórico completo.



## Criptografia de dados de cartão (bônus)

Utilizando alguma estratégia de criptografia, a aplicação deve armazenar os dados de cartão de crédito dos clientes finais no banco de dados de forma mais segura. Cada cartão de crédito deve possuir um token de identificação e este token deve ser utilizado para referência nas cobranças emitidas para ele.

## Relatório de cobranças (bônus)

As empresas cadastradas na plataforma de pagamentos podem acessar um relatório com todas as cobranças de um determinado intervalo de tempo. Este relatório deve conter, ao final, o montante total das cobranças realizadas, o total de taxas pagas (de acordo com o meio de pagamento de cada cobrança), o valor total a receber (descontando as taxas) e o total dos descontos concedidos aos clientes.

## Emissão de um arquivo de cobrança (bônus)

Os administradores da plataforma de pagamentos devem ser capazes de efetivar a cobrança com os bancos e operadoras. Para isto é preciso gerar um arquivo com todas as cobranças pendentes de cada banco ou operadora. O arquivo deve conter todas cobranças pendentes de um determinado meio de pagamento e deve conter cobranças de todos clientes, sem necessidade de separar estas cobranças em arquivos diferentes.

Para cada forma de pagamento diferente (boleto, cartão de crédito ou PIX) deve ser gerado um arquivo com os dados para cobrança. Os arquivos devem seguir a nomenclatura: `TIMESTAMP_MEIO_PAGAMENTO_EMISSAO.TXT`.

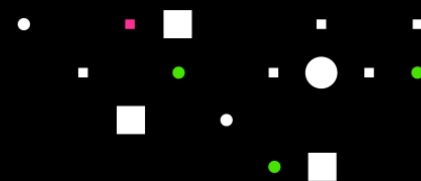
Exemplo: `20210310_PIX_EMISSAO.TXT`. Todos arquivos gerados devem ser armazenados em um diretório a seu critério.

Deve ser lançado no sistema que uma determinada cobrança foi registrada em um arquivo. Devem ser armazenados ao menos o token referente à cobrança, o nome do arquivo gerado e a data da emissão do arquivo. A mesma cobrança pode ser registrada em mais de um arquivo, no caso de ser rejeitada e acontecer uma nova tentativa de cobrança, por exemplo.

Quando a cobrança é lançada em um arquivo, o status da cobrança deve mudar de “pendente” para “pendente de processamento”. Todos os arquivos devem conter 3 blocos: cabeçalho, corpo e rodapé. Cada linha do arquivo começa sempre com um caracter que indica se aquela linha faz parte do cabeçalho, do corpo ou do rodapé.

A linha do cabeçalho do arquivo deve sempre começar com o caracter H, as linhas com as cobranças devem começar com a letra B e a linha do rodapé deve começar com a letra F.





A linha do cabeçalho é única e deve conter a seguinte sequência de informações:

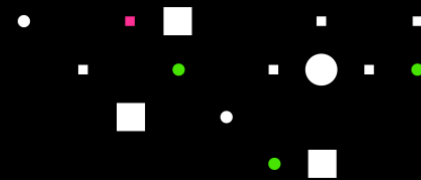
Conteúdo	Descrição
Letra H	Identificador do cabeçalho
Sequência numérica	5 caracteres contendo o total de cobranças presentes no arquivo, com zeros à esquerda. Ex: 00075 para um arquivo com 75 cobranças

As linhas do corpo do arquivo são múltiplas, uma para cada cobrança, e devem conter a seguinte sequência de informações:

Conteúdo	Descrição
Letra B	Identificador de linha do corpo do arquivo
Sequência alfanumérica	Identificador único da cobrança (token da cobrança)
Sequência numérica	Data de vencimento no formato: AAAAMMDD
Sequência vazia	Data efetiva do retorno da cobrança no formato AAAAMMDD. No arquivo de emissão deve ser mantido em branco.
Sequência numérica	10 caracteres com o valor da cobrança, contendo 2 casas decimais e preenchendo com zeros à esquerda caso necessário. Ex: para uma cobrança de R\$ 95,90 a linha deve conter: 0000009590
Sequência numérica	2 caracteres que identificam o status da cobrança (tabela adicional abaixo)

A linha do rodapé é única e deve conter a seguinte sequência de informações:

Conteúdo	Descrição
Letra F	Identificador do rodapé
Sequência numérica	15 caracteres contendo o montante total de todas as cobranças presentes no arquivo, preenchendo com zeros à esquerda caso



	necessário.
--	-------------

Tabela Adicional - Status de Cobrança

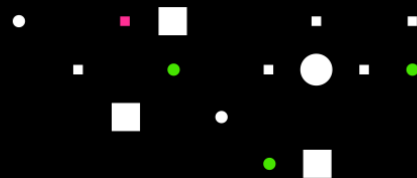
Código de Status	Descrição
01	Pendente de cobrança
05	Cobrança efetivada com sucesso
09	Cobrança recusada por falta de créditos
10	Cobrança recusada por dados incorretos para cobrança
11	Cobrança recusada sem motivo especificado

## Regras e Recomendações

- Evite ficar muitos dias sem enviar *commits* para seu projeto. Com pequenas evoluções constantes, fica mais fácil avaliar e ajudar com feedbacks pontuais durante a execução.
- O sistema deve ser construído usando Ruby on Rails e usando testes com as *gems* RSpec e Capybara devidamente configuradas. O banco de dados deve ser o SQLite e todas dependências adicionais devem ser instaladas automaticamente com os comandos *bundle install* ou *bin/setup*.
- O arquivo README.md deve conter uma descrição do projeto, instruções para executar o projeto e os testes, *gems* adicionais utilizadas (com justificativa do seu uso) etc.
- Para construir as interfaces, é recomendado, mas não obrigatório, o uso do Bootstrap. Você pode utilizar outras bibliotecas ou frameworks, assim como construir seu próprio CSS, mas entenda que essa não é uma prioridade no projeto.
- Ao criar modelos, telas e formulários você deve escolher os campos utilizados, assim como seus tipos de dados. Você pode usar sua criatividade para adicionar outras funcionalidades e informações ao sistema, mas mantenha o foco em concluir as funcionalidades básicas primeiro.

\* Os usuários e/ou e-mails dos avaliadores podem ser obtidos via Slack e devem ser autorizados a acessar qualquer ferramenta privada que você deseje utilizar.

# CAMPUS CODE



# treinadev

é um programa gratuito de  
formação de devs da Campus Code

Patrocínio:

VINDI

R E  
B A  
S E

PORTAL  
solar

smartfit

konduto

CAMPUS  
CODE

Al. Santos, 1293, conj. 73  
Cerqueira César, São Paulo, SP  
Telefone/Whatsapp: [\(11\) 2369-3476](tel:(11)2369-3476)

