**Unidade Curricular: Integração de Sistemas**

**Professor: Prof. Doutor Ricardo Baptista**

**Trabalho realizado por: Ivo Nunes nº 20232582**

**RELATÓRIO CREDIBANK**

# Contexto

O objetivo deste projeto foi desenvolver uma API de pagamento online para o sistema CrediBank, que simula a emissão de cheques digitais. A API foi implementada utilizando o framework Flask em Python, permitindo que os users obtenham um cheque digital com base na identificação da conta de créditos e no valor a ser debitado. Este cheque digital pode ser utilizado para pagamentos em serviços eletrônicos que estejam aderentes ao sistema CrediBank.

O sistema expõe um endpoint RESTful que, ao ser acedido com a identificação da conta de créditos e o valor, devolve uma resposta no formato JSON com a data de emissão e o número de identificação do cheque digital gerado.

# Especificações

O sistema CrediBank segue as seguintes especificações técnicas:

A URL do endpoint é formatada como:

* https://credibank.intsis.utad.pt:8080/check/{credit\_account\_id}/ammount/{value}/

Método HTTP: GET

Parâmetros:

* credit\_account\_id: A identificação da conta de créditos, representada por um número de 16 dígitos.
* value: O valor a ser debitado da conta, representado por um número inteiro.

Resposta: A resposta do endpoint é um código JSON que contém:

* date: A data e hora de emissão do cheque no formato yyyy-mm-ddThh:mm:ss.
* checkID: A identificação do cheque digital, composta por 16 dígitos aleatórios.

Exemplo de resposta:  
[  
 {  
 "date": "2024-12-15T14:30:15",  
 "checkID": "1234567890123456"  
 }  
]

# Arquitetura e Implementação

A implementação do sistema foi realizada com as seguintes considerações:

Tecnologia Utilizada:

* Flask: Framework web para a criação da API REST.
* Python: Linguagem de programação utilizada para implementar o servidor e a lógica do sistema.

Estrutura do Projeto:  
 A estrutura do projeto é simples, com os principais componentes organizados da seguinte forma:  
A close up of a text

Description automatically generated

Geração do CheckID:  
 A identificação do cheque digital é gerada aleatoriamente, com 16 dígitos numéricos, utilizando a função generate\_check\_id(). Esta abordagem simula a criação de um cheque único para cada solicitação.

Validação da Conta de Créditos:  
 O sistema valida se a conta de créditos fornecida possui exatamente 16 dígitos e se é composta apenas por números. Caso contrário, uma mensagem de erro é retornada ao user.

Data e Hora de Emissão:  
 A data e hora atuais são capturadas e formatadas no formato adequado.

# Fluxo do Sistema

O fluxo do sistema segue estas etapas:

* O user faz uma solicitação GET para o endpoint /check/<credit\_account\_id>/ammount/<value>/, passando o número da conta de créditos e o valor a ser debitado.
* O sistema valida a conta de créditos, verificando se o número possui exatamente 16 dígitos.
* O sistema gera um número aleatório de 16 dígitos para ser usado como a identificação do cheque digital (checkID).
* A data e hora atuais são capturadas e formatadas no formato adequado.
* A resposta em JSON é construída e retornada ao user, contendo o date e o checkID.

# Resultados Obtidos

Ao realizar a solicitação para o endpoint, o sistema retorna a seguinte resposta:

Se a conta de créditos for válida e a solicitação for bem-sucedida, o sistema retornará um JSON com a data de emissão e o ID do cheque digital.

Se a conta de créditos fornecida não for válida (não contiver 16 dígitos ou contiver caracteres não numéricos), o sistema retornará uma mensagem de erro com o código HTTP 400.

Exemplo de chamada válida:  
GET http://localhost:8080/check/1234567890123456/ammount/500/

Exemplo de resposta:  
[  
 {  
 "date": "2024-12-15T14:30:15",  
 "checkID": "1234567890123456"  
 }  
]

# Conclusão

O sistema CrediBank foi implementado com sucesso, utilizando o framework Flask para criar uma API simples e eficiente para a emissão de cheques digitais. O sistema garante a validação da conta de créditos e a geração de um cheque digital único, que pode ser utilizado para pagamentos em outros serviços eletrônicos aderentes ao sistema CrediBank.

O projeto pode ser expandido para incluir funcionalidades adicionais, como autenticação, armazenamento de dados em banco de dados, ou ainda a integração com sistemas de pagamento reais. Além disso, melhorias podem ser feitas para aumentar a segurança, como a criptografia dos dados de contas de créditos e cheques digitais.

Este sistema demonstra a aplicação prática de uma API RESTful em um contexto financeiro e serve como base para sistemas de pagamento online mais complexos.O projeto pode ser usado como ponto de partida para a criação de sistemas de pagamentos eletrônicos seguros e escaláveis.  
Além disso, ao comparar a implementação do CrediBank com o uso de gRPC, podemos observar as principais diferenças entre o modelo RESTful utilizado e o modelo gRPC.

No modelo RESTful, utilizado no CrediBank, a comunicação entre o cliente e o servidor é feita por meio de requisições HTTP, utilizando JSON como formato de troca de dados. Esse modelo é simples de implementar e amplamente utilizado, sendo ideal para sistemas que não exigem comunicação em tempo real ou alta performance.