



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Pedro Destri

**Desenvolvimento de um sistema de Business Intelligence para uma fintech de
Crédito como Serviço**

Florianópolis
2021

Pedro Destri

**Desenvolvimento de um sistema de Business Intelligence para uma fintech de
Crédito como Serviço**

Relatório final da disciplina DAS5511 (Projeto de Fim de Curso) como Trabalho de Conclusão do Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Santa Catarina em Florianópolis.

Orientador: Prof. Ricardo José Rabelo, Dr.

Supervisor: Gustavo Arruda Macedo

Florianópolis

2021

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Destri, Pedro

Desenvolvimento de um sistema de Business Intelligence
para uma fintech de Crédito como Serviço / Pedro Destri ;
orientador, Ricardo José Rabelo, 2021.

90 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Engenharia de Controle e Automação,
Florianópolis, 2021.

Inclui referências.

1. Engenharia de Controle e Automação. 2. Dashboards. 3.
Power BI. 4. Fintech. 5. Inteligência de mercado. I.
Rabelo, Ricardo José. II. Universidade Federal de Santa
Catarina. Graduação em Engenharia de Controle e Automação.
III. Título.

Pedro Destri

**Desenvolvimento de um sistema de Business Intelligence para uma fintech de
Crédito como Serviço**

Esta monografia foi julgada no contexto da disciplina DAS5511 (Projeto de Fim de Curso) e aprovada em sua forma final pelo Curso de Graduação em Engenharia de Controle e Automação

Florianópolis, 27 de Setembro de 2021.

Prof. Hector Bessa Silveira, Dr.
Coordenador do Curso

Banca Examinadora:

Prof. Ricardo José Rabelo, Dr.
Orientador
UFSC/CTC/DAS

Gustavo Arruda Macedo
Supervisor
Flow Finance

Prof. Felipe Gomes de Oliveira Cabral, Dr.
Avaliador
UFSC/CTC/DAS

Prof. Marcelo De Lellis Costa de Oliveira, Dr.
Presidente da Banca
UFSC/CTC/DAS

Dedico esse trabalho a minha mãe Lucia, e ao meu pai
Jorge.

AGRADECIMENTOS

Início está seção de agradecimentos, primeiramente, agradecendo ao meu pai Jorge, por todo apoio que ele me deu durante a graduação, pelos inúmeros almoços que tivemos durante a universidade, e pela ajuda na realização desse trabalho.

Agradeço a minha mãe Lucia, que sempre me deu todo suporte necessário para que eu conseguisse trilhar meu caminho até aqui, que escutou sempre minhas dificuldades e me incentivou a seguir em frente.

Agradeço também a minha irmã caçula Gabriela, que serve de inspiração para mim com todo seu esforço e dedicação.

Agradeço também ao meu amigo e parceiro de trabalho Gustavo, por sempre ter acreditado no meu trabalho e pela confiança depositada em mim.

Agradeço aos meus amigos pelo constante incentivo, e por terem me acompanhado, sempre dando boas risadas, nessa trajetória que agora se encerra.

Agradeço ao restante da minha família, que durante esses quase 6 anos de faculdade, me perguntaram em todo evento de família quando eu iria me formar, com expectativas de que esse dia chegasse um dia, e finalmente o último passo nessa direção chegou.

Por fim, agradeço ao meu orientador, professor Rabelo, por ser paciente comigo e por sempre me ajudar. Sem ele esse trabalho nunca teria sido possível.

*“ We’re witnessing the creative destruction of financial services,
rearranging itself around the consumer.
Who does this in the most relevant,
exciting way using data and digital, wins”
(Sankaran, 2016)*

RESUMO

Este trabalho foi desenvolvido para atender uma necessidade da equipe de negócios de uma fintech de crédito como serviço. A gestão dos dados de empresa era feita de maneira descentralizada, através de planilhas espalhadas. Com o crescimento do volume de dados, tanto a organização quanto a análise e controle desses dados, estavam se tornando inviáveis. A análise desses dados serve como base para as linhas de crédito que os clientes de empresa tomam, e a ausência de uma boa análise compromete as linhas de crédito fornecidas, e por consequência o faturamento. Esse trabalho realizou o processo de ETL com os dados da fintech, organizando a estrutura de dados existente na empresa, armazenando os dados em um banco de dados em nuvem, para que fosse utilizado para análise. Além de organizar e armazenar, também foi desenhado um processo de atualização dos dados no banco, para garantir a continuidade do trabalho. Tendo os dados em nuvem, esses dados serviram como base para construção de dashboards no Power BI, onde posteriormente foi acoplado um sistema que auxilia a encontrar a linha de crédito ideal para cada cliente.

Palavras-chave: ETL. Fintech. Dashboard. Power BI.

ABSTRACT

This project was developed to meet a business team need in a fintech that works with credit as a service. The company's data management was decentralized, and data was spread through electronic spread sheets. With the growth of the data volume, organizing, analyzing and controlling the data became unfeasible. The analysis of these data serves as a basis to set the client's line of credit, and the absence of a good analysis compromises the lines of credit provided, and, consequently, the revenues. This work carried out the ETL process with the fintech data, organizing the company's existing data structure and storing the data in a cloud database. In addition to organizing and storing, a process for updating the data in the database was also designed to ensure the continuity of the work. Having the data in the cloud, this data served as the basis for building dashboards in Power BI, where a system was later added, which helps to find the ideal credit line for each customer.

Keywords: ETL. Fintech. Dashboard. Power BI.

LISTA DE FIGURAS

Figura 1 – Estrutura do DW	24
Figura 2 – Diagrama ER	25
Figura 3 – Modelo Lógico	25
Figura 4 – Modelo físico	26
Figura 5 – Star Schema	27
Figura 6 – Exemplo de um dashboard	29
Figura 7 – O processo de ETL	30
Figura 8 – Conexão do DW com PBI via gateway	33
Figura 9 – Conexão do DB com PBI via VPN	34
Figura 10 – Arquitetura do sistema	44
Figura 11 – Modelo físico do projeto	51
Figura 12 – Query para criar tabela	52
Figura 13 – Fontes de conexão de dado com o Power BI	54
Figura 14 – Modelo dimensional das parcelas	56
Figura 15 – Modelo dimensional do histórico das parcelas	57
Figura 16 – Modelo dimensional dos vencimentos	57
Figura 17 – Painel de operações	60
Figura 18 – Painel de vencimentos	62
Figura 19 – Painel de composição das parcelas mensais	64
Figura 20 – Painel de histórico das parcelas	66
Figura 21 – Painel de cobrança mensal	67
Figura 22 – Painel de auxílio as expansões de crédito	68
Figura 23 – Código em DAX - construção de uma medida de soma	70
Figura 24 – Painel de indicadores da operação	70
Figura 25 – Código em DAX - faturamento acumulado 4 meses	72
Figura 26 – Tabela de <i>output</i> do sistema de auxílio de limite de crédito	73
Figura 27 – Painel de operações	84
Figura 28 – Painel de vencimentos	85
Figura 29 – Painel de composição das parcelas mensais	86
Figura 30 – Painel de histórico das parcelas	87
Figura 31 – Painel de cobrança mensal	88
Figura 32 – Painel de auxílio as expansões de crédito	89
Figura 33 – Painel de indicadores da operação	90

LISTA DE QUADROS

Quadro 1 – Fontes de dado	41
-------------------------------------	----

LISTA DE TABELAS

Tabela 1 – Conteúdo dashboards empresa parceira	45
Tabela 2 – Conteúdo dashboards Flow Finance	46
Tabela 3 – Fontes de dado do projeto	50
Tabela 4 – Conteúdo Painel de operações realizadas	59
Tabela 5 – Conteúdo Painel de vencimentos	61
Tabela 6 – Conteúdo Painel de Composição da parcela mensal	63
Tabela 7 – Conteúdo Painel de Histórico de todas as parcelas	65
Tabela 8 – Conteúdo Painel de Gráficos de controle interno da operação	69

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
BaaS	Bancarização
BD	Banco de Dados
CaaS	Crédito como serviço
csv	<i>Comma separated values</i>
DBT	Data building tool
DW	Data Warehouse
ER	Entidade Relacionamento
ETL	Extração, tratamento e carga de dados
ODBC	<i>Open Database Connectivity</i>
PIB	Produto Interno Bruto
SAD	Sistema de apoio à decisão
SGBD	Sistema gerenciador do banco de dados

SUMÁRIO

1	INTRODUÇÃO	15
1.1	OBJETIVOS	17
1.1.1	Objetivo geral	17
1.1.2	Objetivos específicos	17
1.2	ESTRUTURA DO DOCUMENTO	18
2	DESCRIÇÃO DA EMPRESA	20
3	EMBASAMENTO TEÓRICO	21
3.1	ARQUITETURA DE UM SISTEMA	21
3.2	BANCO DE DADOS RELACIONAL	21
3.3	SISTEMA GERENCIADOR DE BANCO DE DADOS (SGBD)	22
3.4	DATA WAREHOUSES	23
3.5	DIAGRAMA Entidade Relacionamento (ER), MODELO CONCEITUAL E MODELAGEM FÍSICA	24
3.6	MODELAGEM DIMENSIONAL DOS DADOS E O STAR SCHEMA	26
3.7	ARMAZENAMENTO EM NUVEM	27
3.8	DASHBOARDS	28
3.9	METODOLOGIA DE TRABALHO COM POWER BI	28
3.10	ETL	30
3.11	SISTEMAS DE APOIO A TOMADA DE DECISÃO	34
4	ESPECIFICAÇÃO DO PROJETO	36
4.1	ESBOÇO DA SOLUÇÃO	37
4.2	REQUISITOS DO SISTEMA	38
4.3	DEFINIÇÃO DA ARQUITETURA DO SISTEMA	39
4.3.1	Estrutura de dados	39
4.3.2	Armazenamento do dado	42
4.3.3	Software para construção dos dashboards	43
4.3.4	Sistema de auxílio a decisão do limite de crédito	43
4.3.5	Visão geral do sistema	44
4.4	ESPECIFICAÇÃO DO CONTEÚDO DOS <i>DASHBOARDS</i>	45
5	DESENVOLVIMENTO	47
5.1	REUNINDO OS DADOS DA OPERAÇÃO	47
5.2	TRATANDO OS DADOS	47
5.3	MODELANDO OS DADOS	50
5.4	CRIAÇÃO DO BANCO DE DADOS	52
5.5	CARREGANDO OS DADOS NO BANCO	53
5.6	CONECTANDO O BANCO AO POWER BI	53
5.7	TRATAMENTO DO DADO NO POWER BI	55

5.8	DESENVOLVIMENTO DOS DASHBOARDS	58
5.8.1	Painel de operações realizadas	59
5.8.2	Painel do histórico de vencimentos	60
5.8.3	Painel do Composição da parcela mensal	62
5.8.4	Painel do histórico de todas as parcelas	64
5.8.5	Pagamento mensal dos clientes	66
5.8.6	Painel de auxílio nas expansões de crédito	67
5.8.7	Gráficos de controle interno da operação	68
5.9	ESTUDO DAS LINHAS DE CRÉDITO	71
5.10	SISTEMA DE AUXÍLIO PARA LIMITE DE CRÉDITO	71
6	ANÁLISE DOS RESULTADOS	74
6.1	VALIDAÇÃO DOS RESULTADOS PARA EMPRESA PARCEIRA	74
6.2	VALIDAÇÃO DOS RESULTADOS PARA FLOW FINANCE	75
6.2.1	Painel de apoio as expansões de crédito	75
6.2.2	Painel de cobrança mensal	75
6.2.3	Sistema de auxílio a determinação das linhas de crédito	76
6.3	REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS	76
6.4	CONSIDERAÇÕES FINAIS SOBRE OS RESULTADOS	77
7	CONCLUSÃO	78
	REFERÊNCIAS	80
	APÊNDICE A – DASHBOARDS DESENVOLVIDOS	83

1 INTRODUÇÃO

O acesso a crédito para micro e pequenas empresas (MPEs) no Brasil é um processo complicado demorado e caro. Enquanto grandes grupos empresariais tem acesso a crédito com juros baixos, as MPEs sofrem com a dificuldade de ter acesso a crédito e quando tem, muitas vezes esse crédito possui uma alta taxa de juros.

Segundo dados do SEBRAE (2020) "Os pequenos negócios respondem por mais de um quarto do Produto Interno Bruto (PIB) brasileiro. Juntas, as cerca de 9 milhões de micro e pequenas empresas no País representam 27% do PIB ". Hoje os pequenos negócios na economia brasileira representam 27% do PIB do Brasil, geram 52% dos empregos com carteira assinada e somam 40% dos salários pagos no país. Há um total de 10,9 milhões de MPEs no Brasil (GOV.BR, 2021).

O mercado de crédito no Brasil é um mercado de 3,3 trilhões de reais. Desse mercado, 85% do montante está centralizado em cinco instituições financeiras. Essa concentração, dentre outros fatores, resulta no segundo maior *spread* bancário do mundo. Junto a isso, 68% da população não possui acesso a crédito, 59,4 milhões de pessoas (J.COMERCIO, 2021) estão negativadas e 34 milhões de adultos estão desbancarizados (VALOR, 2021). No âmbito empresarial a situação também não é diferente. 90% dos empreendedores também não tem acesso a crédito. Mesmo com programas de auxílio durante a pandemia do *covid-19*, apenas 15% das MPEs conseguiu acesso a crédito (NOTÍCIAS, 2020).

Grande parte da dificuldade do acesso ao crédito no Brasil é resultado de uma grande assimetria de informação. Muitas vezes não é lucrativo para um banco atender as micro e pequenas empresas. A precificação do risco de crédito por parte dos bancos é feita através do histórico de maus pagadores. Por consequência, aqueles que não possuem um histórico costumam ficar de fora do sistema.

Esse problema é complexo, visto que, para ter um histórico, é preciso ter acesso, e para ter acesso, é preciso ter um histórico. A consequência dessa estranha equação é um mercado de crédito proibitivo àqueles que precisam e disponível ao restante.

Hoje há uma nova geração de empresas que começam agregar ao sistema financeiro tradicional e buscam atender parcela ainda pouco explorada do mercado. Utilizando de novas tecnologias e modelos de negócios diferentes do tradicional, essas empresas conseguem ser mais eficientes e ter um menor custo, proporcionando uma experiência singular e atendendo a uma parcela de clientes que antes era marginalizada. As chamadas *fintechs* estão criando uma disrupção no sistema financeiro incumbente.

Embora não exista ainda um consenso em definição, as *fintechs* são geralmente caracterizadas como empresas que redesenham os serviços financeiros, utilizando tecnologia de forma intensiva para reduzir custos e atuar de forma mais eficiente e

abrangente do que as instituições financeiras tradicionais.

O diferencial dessas empresas não se restringe apenas a tecnologia. As *fintechs* atuam na intersecção dos mercados de finanças, tecnologia e regulatório. O grande desafio dessas empresas é utilizar tecnologia, se moldando à regulamentação e suas mudanças, e usar o capital (próprio ou de terceiros) para criar produtos ou serviços financeiros que os consumidores adorem e que coloquem em uso no seu dia-a-dia.

O mundo está cada vez mais conectado através da tecnologia. Ela permite escalar produtos e serviços com custo reduzido e aumentar o alcance das soluções. Novas tecnologias permitem a captura e processamento de informação em escala elevada, o que abre espaço para novas oportunidades. Ao contrário dos bancos tradicionais, as *fintechs* já nasceram adaptadas a essa nova realidade.

Dentro desse contexto surge a Flow Finance (ou apenas Flow), uma *fintech* que trabalha com Crédito como serviço (CaaS) e Bancarização (BaaS). A modalidade de crédito como serviço permite com que qualquer empresa possa distribuir produtos financeiros. Unindo dados transacionais para a criação de algoritmos customizados e a capilaridade necessária para a distribuição, produtos financeiros customizados podem ser lançados por qualquer empresa que queira contratar o serviço da *fintech*.

A Flow Finance acredita na desintermediação da oferta de crédito como guia para redução do *spread* e aumento do acesso a crédito para micro e pequenas empresas. Pensando nisso, a empresa criou uma tecnologia capaz de transformar qualquer um em um distribuidor de crédito. Com uma simples integração de integração de *Application Programming Interface* (API), qualquer empresa pode se tornar uma *fintech* munida de crédito rápido, transparente e justo.

A Flow Finance possui um cliente (que a partir de agora será chamado de **parceiro** ou **empresa parceira** - Por questões de *complicance* com cliente, não será citado) com uma operação na modalidade de crédito como serviço. Esse cliente utiliza da infraestrutura tecnológica da Flow para monetizar a sua própria base de clientes, permitindo assim que os seus clientes tenham acesso a um produto de crédito customizado.

Os dados gerados pelos clientes são de suma importância para o parceiro e para a própria Flow. Esses dados são utilizados para controle da operação, desde visualização dos dados, até gestão de indicadores, como também para análise de limite de crédito dos clientes.

Hoje a estrutura de dados dessa operação está descentralizada e não tratada. Isso implica em um desgaste operacional, elevando o tempo gasto para lidar com as informações da operação. Além disso, devido a desestruturação do dado, se torna difícil realizar análises, novamente elevando o tempo e custo da operação, como também implicando em possíveis perdas, visto que é de suma importância para todas as partes na operação (Flow Finance, parceiro e clientes do parceiro) que o limite de crédito

sendo disponibilizado seja o limite ideal.

Dada a dificuldade em se lidar e utilizar estes dados, surge a motivação e a justificativa para este trabalho. É necessário tornar de fácil acesso e visualização os dados da operação. Além disso, também se faz necessário criar um sistema ou método que auxilie a encontrar a linha de crédito ideal para cada cliente.

1.1 OBJETIVOS

Visando guiar o projeto, traçou-se um objetivo geral e 11 objetivos específicos. O objetivo geral representa o foco do projeto, enquanto os objetivos específicos são atividades que tem de ser realizadas para alcançar de forma satisfatória o que foi proposto no objetivo geral.

1.1.1 Objetivo geral

Este trabalho tem como objetivo geral desenvolver um sistema de *business intelligence* (BI) para uma *fintech* de crédito como serviço (CaaS).

Para isso ser feito, deve-se agrupar todos os dados da operação de CaaS, tratar estes dados, e armazená-los em um local de acesso público para os colaboradores da Flow Finance.

Após a disponibilização do dado, devem ser montados *dashboards* para visualização dos dados da operação, tanto por parte da Flow Finance, quanto por parte da empresa parceira.

Por fim, deve-se desenvolver um sistema ou método que auxilie no processo de análise do limite de crédito dos clientes do parceiro.

1.1.2 Objetivos específicos

Os objetivos específicos do trabalho são a tradução do objetivo geral em ações que, quando completadas, devem resultar na conclusão do objetivo específico.

Esses objetivos podem ser listados nas seguintes partes específicas, que não necessariamente são realizadas nessa ordem:

1. Definição da arquitetura do sistema - contemplando como e onde armazenamento de dado, software para dashboards, conexão entre partes e sistema de apoio a tomada de decisão;
2. Mapear, agrupar, modelar e tratar os dados;
3. Criar estrutura de armazenamento dos dados;
4. Armazenar os dados - seguindo a modelagem já feita;
5. Definição de um processo e cadência de atualização dos dados;

6. Levantar as necessidades da área de negócios e no parceiro (indicadores, painéis e dados para serem visualizados);
7. Conexão dos dados com um software de construção de dashboards;
8. Desenvolver os dashboards - visando atender as necessidades internas e do parceiro;
9. Criar os dashboards;
10. Desenvolver modelo de auxílio a decisão de crédito;
11. Integrar modelo aos dashboards.

Dos objetivos específicos citados acima, o primeiro é a origem de todo o projeto, e onde se desprende um grande tempo. A arquitetura do sistema envolve diversas partes e diversas decisões, que irão afetar o projeto como um todo.

Ali será definido onde os dados serão armazenados. Seja esse armazenamento local ou em nuvem. Qual será o sistema gerenciador do banco de dados (SGBD), qual será o software utilizado para criar os dashboards e como será feita a conexão desse software com o banco de dados. Por fim, aqui também é definido como será o sistema de apoio a tomada de decisão que será acoplado aos painéis desenvolvidos.

Os objetivos 2 à 5 são referentes ao processo de Extração, tratamento e carga de dados (ETL), e é onde mora grande parte da complexidade deste trabalho, como também onde há maior espaço para aplicar diferentes técnicas.

Os objetivos 6 à 9 são focados na área de negócios da empresa e na necessidade do parceiro. Aqui o foco maior é em entender as necessidades e traduzi-las em painéis que atendam as necessidades elencadas.

Por fim, os objetivos 10 e 11 são focados no processo de definição de limite de crédito de um cliente. Aqui será desenvolvido o sistema/método que apoiará a tomada de decisão da *fintech*. Vale ressaltar que esse sistema a ser desenvolvido é um modelo simples para auxiliar no processo de tomada de decisão, e não um Sistema de apoio à decisão (SAD) complexo, com sua própria arquitetura e modelos de tomada de decisão.

Tendo atingido todos esses objetivos específicos o projeto terá obtido sucesso, e os objetivos gerais assim estarão concluídos.

1.2 ESTRUTURA DO DOCUMENTO

No capítulo 2 desse documento é feita uma breve descrição da empresa, do seu produto de crédito como serviço, e dos processos de tomada de crédito.

No capítulo 3 é apresentada uma fundamentação teórica do projeto. Ali são expostas as teorias que embasam o desenvolvimento, assim como conceitos essenciais para o entendimento do projeto.

No quarto capítulo são expostos os requisitos gerais, funcionais e não-funcionais a serem considerados no projeto. Além disso, é exposto um esboço inicial da solução e uma visão geral do sistema.

No capítulo 5 é descrito em detalhes o projeto do sistema criado. Para isto a solução é formalizada por meio de diagramas que modelam os dados, além da descrição dos processos que foram utilizados neles. Também é exposto o desenvolvimento dos *dashboards* e do sistema auxiliar. Aqui estão as decisões que foram tomadas e o seu porquê, em alinhamento com os requisitos colocados.

No capítulo 6 é feita uma análise de resultados do projeto na empresa. As vantagens e desvantagens e os impactos reais nos resultados nos processos da empresa

Por fim é feita a conclusão do trabalho. Ela engloba um resumo do que foi feito, retomando a motivação do trabalho, a solução proposta e os principais resultados atingidos. Além disso, é feita uma análise das limitações do projeto, e são apontadas melhorias que podem ser realizadas no trabalho.

2 DESCRIÇÃO DA EMPRESA

A Flow Finance é uma fintech de crédito como serviço que a partir de uma solução “*White Label*”, conecta o investidor ao tomador por meio de plataformas parceiras. A Flow, possui uma tecnologia capaz de transformar qualquer empresa em um distribuidor de crédito. Isso é feito através de uma integração de integração de API. A Flow atua como motor por trás de toda mecânica de oferta de crédito. Uma solução completa com atuação em todas as etapas do ciclo de distribuição de crédito, indo desde o processo de KYC (know your customer) até a etapa de cobrança.

Através de uma plataforma online desenvolvida pela Flow, as empresas parceiras conseguem então ofertar crédito para a sua base de clientes. Essa oferta de crédito pode partir de um *funding* da própria empresa parceira, ou então de uma instituição financeira terceira. Independente de qual é a origem do crédito, o processo é semelhante.

O cliente, ao cadastrar na plataforma online da Flow passa por um processo de análise de crédito, que é feito através de um algoritmo próprio. Se o cliente é aprovado na análise de crédito, ele fica com um limite de crédito disponível online e realiza a tomada de crédito quando desejar. Para tomar o crédito, o cliente simula a linha através da plataforma, e seleciona a opção que mais o agrada.

Tendo selecionado o valor do crédito e a quantidade de parcelas, é gerada uma cédula de crédito bancária, que é renderizada para o cliente visualizá-la e realizar a assinatura de forma online. Na prática portanto, quando se fala em gerar uma operação de crédito, se fala em gerar uma CCB, que é um contrato entre duas ou mais partes, que necessita ser assinado por todos os envolvidos.

A modalidade de crédito adotada nesse produto é o capital de giro com garantia de recebíveis. O processo inicia quando os clientes da empresa parceira tomam crédito através da plataforma da Flow. Após um período de carência, inicia-se a cobrança dessa operação. Como o crédito é com garantia de recebível, a cobrança das parcelas é feita diretamente através da empresa parceira. A Flow Finance envia para a empresa parceira mensalmente um relatório de cobrança e ela, antes de transferir aos seus clientes o que eles faturaram, retém a parcela de seus empréstimos.

Esse é o ciclo padrão de uma operação e crédito como serviço. Esse ciclo pode passar por algumas mudanças, dependendo do tipo de *funding* da operação, ou então da modalidade de crédito adotada, mas de maneira geral, esse é o seu funcionamento.

3 EMBASAMENTO TEÓRICO

Abaixo serão expostas todas teorias e técnicas que embasaram o desenvolvimento deste trabalho. O primeiro passo a ser dado no desenvolvimento deste trabalho é voltado a compreender as necessidades de quem vai consumir o sistema, e traduzir isso em técnicas e tecnologias serão utilizadas. A Arquitetura de um sistema de informação tem foca na análise das necessidades dos usuários dentro de um possível sistema a ser desenvolvido. A mesma não se atem a detalhes tecnológicos, mas sim em entender o que o cliente realmente precisa, levando em conta ainda as características do negócio em que o mesmo está inserido. Esse será o ponto de partida do projeto.

3.1 ARQUITETURA DE UM SISTEMA

O desenvolvimento desse projeto irá iniciar especificando a arquitetura do sistema. Esse também será o ponto de partida para a fundamentação teórica do projeto.

A arquitetura de sistema de significa o estabelecimento de um conjunto de elementos cuja finalidade é proporcionar um mapeamento da organização no tocante aos elementos envolvidos com o processo de desenvolvimento/implantação de um sistema (TAIT; BARCIA; PACHECO, s.d.).

Ou seja, a definição da arquitetura de um sistema consiste em mapear e definir quais os componentes de software e hardware que farão parte do sistema, qual a finalidade de cada um e como eles interagem.

Além de definir quais os componentes fazem parte do sistema, também é definida a arquitetura da informação no sistema. De onde ela irá originar e como que serão originados esses dados que alimentarão o sistema.

Ao definir a arquitetura do sistema, é especificado o tipo da tecnologia que será utilizada. Essa definição não é feita de forma genérica, como quando definem-se os requisitos não-funcionais de um sistema.

3.2 BANCO DE DADOS RELACIONAL

Conforme descrito pela AWS (2021a), : "Um banco de dados relacional é uma coleção de itens de dados com relacionamentos predefinidos entre si. Esses itens são organizados como um conjunto de tabelas com colunas e linhas. As tabelas são usadas para reter informações sobre os objetos a serem representados no banco de dados. Cada coluna da tabela retém um determinado tipo de dado e um campo armazena o valor em si de um atributo. As linhas na tabela representam uma coleção de valores relacionados de um objeto ou de uma entidade. Cada linha em uma tabela pode ser marcada com um único identificador chamado de chave principal. Já as linhas entre as

várias tabelas podem ser associadas usando chaves estrangeiras. Esses dados podem ser acessados de várias maneiras diferentes sem reorganizar as próprias tabelas do banco de dados."

Ou seja, um banco de dados nada mais é do que um conjunto de tabelas, semelhante a uma tabela de Excel, que se relacionam através de chaves. Essas chaves são identificadores únicos de cada entrada de uma tabela.

Todas as transações que ocorrem em um banco de dados relacional devem estar em conformidade com o ACID (**A**tomicidade, **C**onsistência, **I**solamento, **D**urabilidade) ou serem atômicas, uniformes, isoladas e duráveis, garantindo assim a integridade dos dados.

Atomicidade exige que a transação como um todo seja executada com êxito ou, caso parte da transação falhe, que a transação inteira seja invalidada. **Consistência** determina que os dados gravados no banco de dados como parte da transação devem aderir a todas as regras definidas, e restrições incluindo impedimentos, cascatas e *triggers*.

Isolamento é essencial para obter controle da concorrência e garante que cada transação seja independente por si só. **Durabilidade** estabelece que todas as alterações feitas ao banco de dados sejam permanentes depois de a transação ser concluída com êxito (AWS, 2021a).

Um banco de dados pode ser armazenado em diversos locais diferentes. É possível criar um banco de dados em um computador pessoal, em um servidor, seja esse servidor local a empresa ou um servidor online, ou ainda há a opção de armazenar esse banco de dados na nuvem.

Independente do local de armazenamento, um banco deve utilizar um mecanismo de banco de dados relacional (no inglês, Relational Database Engines). Existem diversas opções da qual se escolher, sendo que diversas dessas opções são *open source*, como o *MySQL* e o *PostgreSQL*.

3.3 SISTEMA GERENCIADOR DE BANCO DE DADOS (SGBD)

De acordo com a IBM (s.d.): Um sistema de gerenciamento de banco de dados (SGBD) nada mais é do que um sistema computadorizado de manutenção de dados. Os usuários do sistema têm facilidades para executar vários tipos de operações em tal sistema, seja para manipulação dos dados no banco de dados ou para o gerenciamento da própria estrutura do banco de dados. Os sistemas de gerenciamento de banco de dados são categorizados de acordo com suas estruturas ou tipos de dados.

Ou seja, um SGBD realiza uma série de funções no banco de dados, apoiando a gestão dos dados. Através dele é possível criar e relacionar as tabelas, carregar, alterar e excluir dados, assim como efetuar consultas no banco e em suas tabelas.

Um sistema para gerir bancos de dados pode ser formatado de diferentes manei-

ras, dependendo dos objetivos e dos próprios dados a serem manipulados. Conforme exposto em Acts (2021), existem diversos tipos de SGBD, como:

- Relacionais (SQL): cuja função é conectar dados de origens diferentes;
- Não relacionais (NoSQL): a diferença em relação aos bancos SQL é que, aqui, a estrutura de dados não tem que ser previamente definida;
- Hierárquico: formato mais antigo, no qual os dados se organizam em uma disposição piramidal, onde dados “filhos” são ligados a dados “pai”;
- De rede: nele, cada “filho” pode ter mais de um “pai” e a estrutura dos dados se assemelha mais a uma teia de aranha do que a uma pirâmide;
- Orientado a objetos: modelo mais avançado em que diferentes formatos de dados se mesclam, sendo, por isso, mais caro de se implementar;

A escolha do SGBD varia de acordo com a aplicação que está sendo desenvolvida. No entanto, independente da aplicação, a utilização de um SGBD é imprescindível. Existem diversas opções de SGBDs, sendo que diversas delas são opções *open source*. Para aplicações pequenas, ou para provas de conceito, as opções de código aberto conseguem se mostrar mais do que suficiente.

3.4 DATA WAREHOUSES

Um Banco de Dados (BD) é uma coleção de dados relacionados que representam elementos do mundo real. Ele foi projetado para ser construído e preenchido com dados para uma tarefa específica. Em aplicações maiores, com maior volume de dados e de diversas fontes, a utilização de apenas um banco de dados passa a ser inviável.

Um Data Warehouse (DW) é um sistema de informação que armazena dados históricos de fontes únicas ou múltiplas. Ele é projetado para analisar, relatar e integrar dados de transações de diferentes fontes.

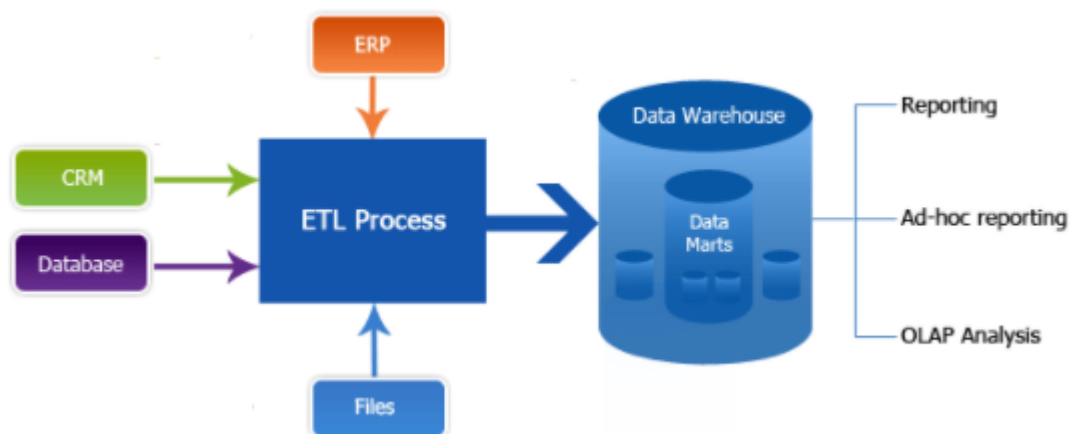
O DW facilita o processo de análise de dados de uma organização. É também uma fonte única da verdade para a organização no processo de tomada de decisão.

Na etapa final no processo de ETL é feito o armazenamento do dado em um *data warehouse* (DW). No entanto, em aplicações com um volume pequeno de dados, ou em provas de conceito, é possível utilizar um banco de dados relacional.

De acordo com a definição da Oracle (2004) : “Um *data warehouse* é um tipo de sistema de gerenciamento de dados projetado para ativar e fornecer suporte às atividades de *business intelligence* (BI), especialmente a análise avançada. Os *data warehouses* destinam-se exclusivamente a realizar consultas e análises avançadas e geralmente contêm grandes quantidades de dados históricos. Os dados em um

data warehouse geralmente são derivados de uma ampla variedade de fontes, como arquivos de *log* de aplicativos e aplicativos de transações.”

Figura 1 – Estrutura do DW



Fonte – Cetax

Ou seja, conforme vemos na figura 1, o DW centraliza os dados retirados de diversas fontes e facilitar a consulta, estas fontes podem conter dados em diversos formatos. Após a extração e tratamento dos dados, é possível armazená-los no Data Warehouse. Com isso, é possível buscar todas as informações importantes em um único lugar organizado e atualizado, criado com foco em facilitar a consulta.

Um ponto importante sobre a utilização dos DW é referente a sua segurança. Suas informações estão disponíveis apenas para leitura. Seus dados não podem ser modificados, apenas em raras exceções. A possibilidade de fazer apenas a leitura das informações assegura a integridade do conteúdo armazenado.

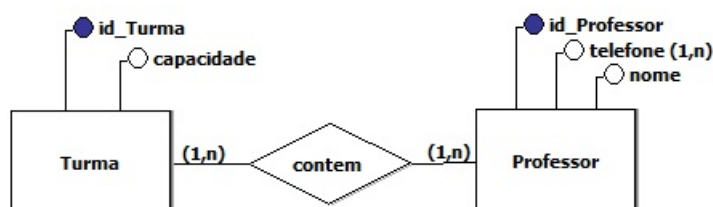
3.5 DIAGRAMA ER, MODELO CONCEITUAL E MODELAGEM FÍSICA

O Modelo Entidade-Relacionamento (ou diagrama ER) é um modelo conceitual que representa a estrutura de informações de um domínio de problema em termos de entidades e relacionamentos. O resultado da modelagem usando o ER é representado graficamente como um diagrama de relacionamento de entidade (SONG; CHEN, 2009).

Um diagrama ER representa a estrutura conceitual do problema que está sendo modelado. Eles são utilizados no projeto de banco de dados e na análise de sistemas para capturar os requisitos de um sistema ou o domínio de problema. Em particular, quando usado para modelagem de dados, o diagrama ER auxilia o desenvolvedor do banco de dados na identificação dos dados, no relacionamento entre eles e das regras que são representadas e usadas em um banco de dados. OS modelos ER

são prontamente traduzidos em esquemas de banco de dados relacionais. A figura 2 mostra o exemplo de diagrama ER.

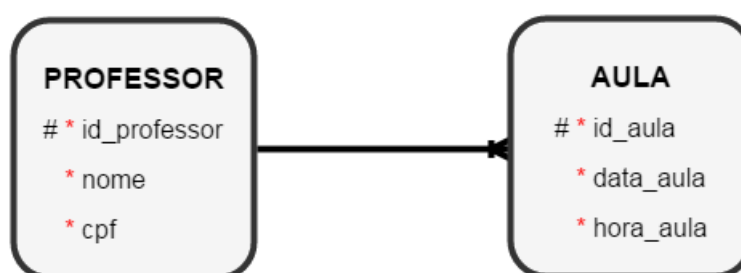
Figura 2 – Diagrama ER



Fonte – Stack Overflow - 2021

Tendo o diagrama ER montado, parte-se para a modelagem lógica. A modelagem lógica é necessária para compilar os requisitos de negócio e representar os requisitos como um modelo. Ela está principalmente associada à coleta de necessidades de negócios, e não ao design do banco de dados. As informações que precisam ser coletadas são sobre unidades organizacionais, entidades de negócios e processos de negócios. Nela descreve-se como os dados serão armazenados no banco e também seus relacionamentos. Esse modelo pode ser: relacional, orientado a objetos, orientado a colunas, entre outros.

Figura 3 – Modelo Lógico



Fonte – Stack Overflow

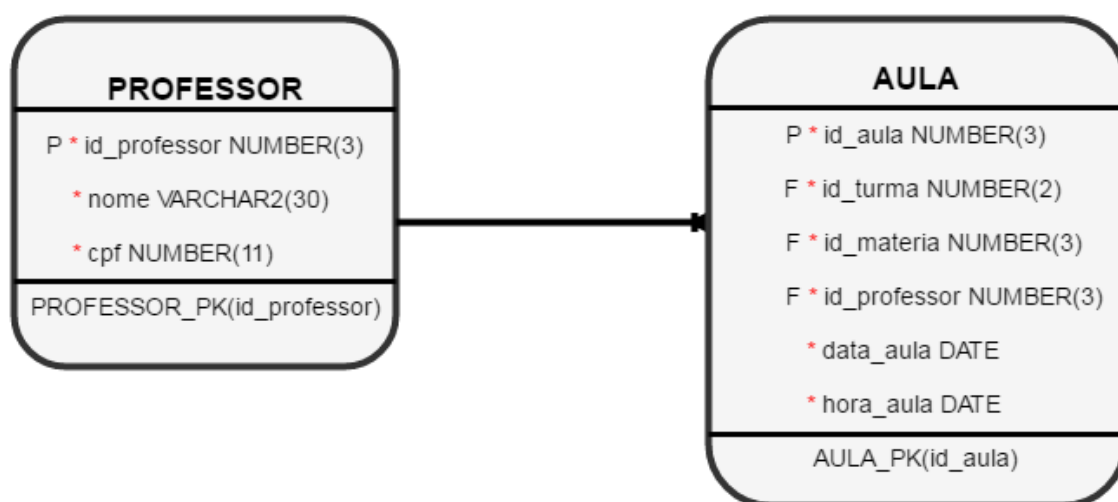
Os modelos lógicos basicamente determinam se todos os requisitos do negócio foram reunidos. O modelo lógico ainda é genérico, e independe do tipo mecanismo de banco de dados que será utilizado. Na figura 3 vemos como ficou o diagrama ER representado na figura 2 após ser transformado no modelo lógico.

Por fim, existe a modelagem física. A modelagem física lida com o design do banco de dados com base nos requisitos reunidos durante a modelagem lógica. To-

das as informações coletadas são convertidas em modelos relacionais e modelos de negócios.

Aqui, tabelas e colunas são feitas de acordo com as informações fornecidas durante a modelagem lógica. Chaves primárias, chaves exclusivas e chaves estrangeiras são definidas para fornecer restrições e relacionar as tabelas. A modelagem física depende do software que já está sendo usado na organização. Na figura 4 vemos como ficou a transformação do modelo lógico visto na figura 3.

Figura 4 – Modelo físico



Fonte – Stack Overflow

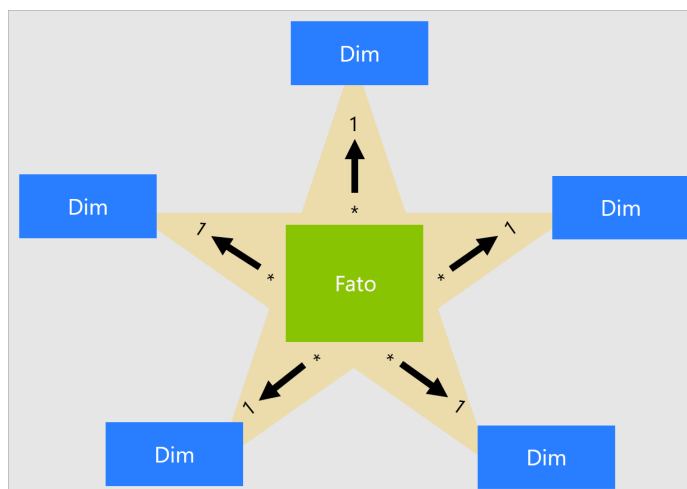
3.6 MODELAGEM DIMENSIONAL DOS DADOS E O STAR SCHEMA

A modelagem dimensional é uma técnica de projeto lógico normalmente usada para em projetos de DW. Ela contrasta com a tradicional modelagem ER. Segundo Rodrigues (2020), a modelagem dimensional é a única técnica viável para bancos de dados que devem responder consultas em um data warehouse.

Star Schema ou esquema estrela (Figura 5), idealizado por Ralph Kimball, é o modelo mais utilizado na modelagem dimensional para dar suporte à tomada de decisão e melhorar a performance de sistemas voltados para consulta. O esquema estrela é composto no centro por uma tabela fato, rodeada por tabelas de dimensão, ficando parecido com a forma de uma estrela, provendo uma visão da modelagem da base de dados para sistemas de apoio à decisão que se aplica ao *Data Warehouse*. Alternativa ao esquema em estrela, é o modelo *snowflake*, que não é tão difundido.

As tabelas dimensão acima, são tabelas que contém as características de um evento. São, na maioria das vezes, dados que não se repetem, dados explicativos. Uma

Figura 5 – Star Schema



Fonte: Microsoft

dimensão, por exemplo, pode armazenar o CNPJ de empresas e sua razão social, já outra tabela dimensão poderia armazenar um CEP e nome de rua.

Já as tabelas fato armazenam algo que ocorreu, é o fato propriamente dito, por isso ela tem esse nome, porque é o fato ocorrido. A tabela fato está sempre ligada a duas ou mais dimensões, não existe tabela fato com menos de duas dimensões. Uma tabela fato, por exemplo, poderia armazenar o crédito tomado por um cliente, e se relacionar com as tabelas acima usando CNPJ e CEP. A tabela fato conecta com as dimensões e isso forma o modelo dimensional. Ela também é a principal tabela no modelo dimensional.

3.7 ARMAZENAMENTO EM NUVEM

Quando se busca um local para armazenar o seu banco de dados, existem diversas opções em questão. O armazenamento do seu banco pode ser feito localmente em um computador pessoal, ou então em um servidor. Esse servidor pode ser local, ou ainda em um servidor online.

Por fim, há também a opção de utilizar o armazenamento em nuvem (no inglês *Cloud*).

De acordo com a definição da IBM (2021): "O armazenamento em nuvem permite que você salve dados e arquivos em um local externo que você acessa por meio da Internet ou de uma conexão de rede privada dedicada. Os dados que você transfere para um local externo para armazenamento tornam-se responsabilidade de um provedor de nuvem terceirizado. O provedor hospeda, protege, gerencia e mantém os servidores e a infraestrutura associada e garante que você tenha acesso aos dados

sempre que precisar."

Embora pareça similar, existe uma grande diferença entre um servidor online dedicado e um servidor em nuvem. Em um servidor dedicado, o cliente pode comprar o servidor completo para si mesmo, onde ele ou ela não precisa compartilhar o espaço do servidor e recursos com outros clientes. Muitas grandes empresas usam servidores dedicados porque podem oferecer um alto nível de segurança. Esta opção não é adequada para pequenas empresas porque requer conhecimento de especialistas e recursos seniores para gerenciar e manter um servidor dedicado.

O servidor em nuvem, em vez de hospedar a página da web em hardware físico, utiliza um ambiente virtual, onde vários servidores são usados para fornecer espaço e recursos para sua aplicação. Em palavras simples, um servidor em nuvem usa vários servidores virtuais e oferece alta escalabilidade e recursos. Ao contrário de um servidor dedicado, a nuvem não salva sua página da web no local específico. Ele usa vários sistemas para gerenciar seu site (SINGH, 2020).

3.8 DASHBOARDS

Sistemas de inteligência de mercado (do inglês *Business Intelligence* ou apenas BI), são sistemas de computador que auxiliam pessoas no processo de tomada de decisão e de visualização de dados. No contexto BI *dashboards* são painéis visuais, que apresentam de maneira centralizada um conjunto de dados. Na figura 6 abaixo vemos um exemplo de dashboard construído no Power BI.

O ponto central da utilização de *dashboards* e sua grande vantagem é o seu dinamismo. Ao invés de construir relatórios com dados estáticos, os *dashboards* permitem que o usuário modifique em tempo real o dado que está sendo visualizado.

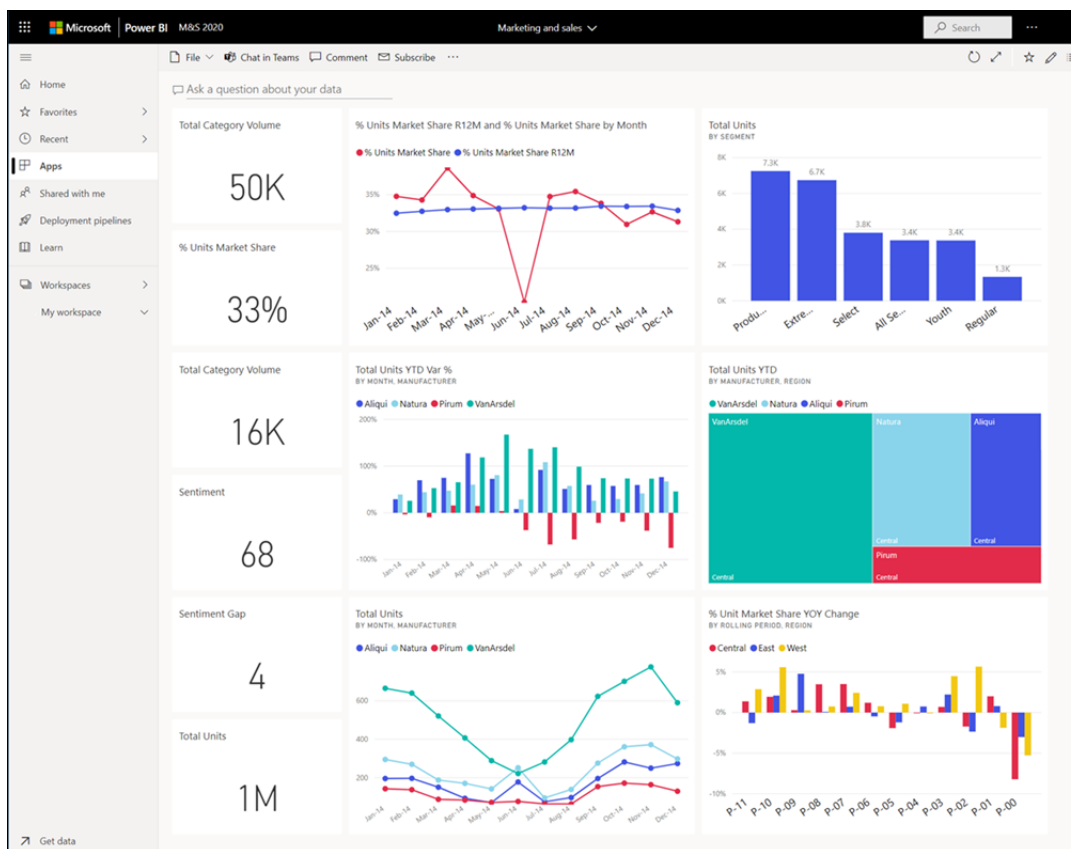
Mesmo que seja utilizado apenas para exibir dados em forma de tabela, um dashboard já traz um dinamismo e agilidade muito elevado. Esses painéis podem ser utilizados até mesmo como ferramenta de consulta em um banco de dados, sem que seja necessário executar uma *query* (consulta) ou fornecer um acesso ao banco de dados.

Sendo assim, além de ser uma ferramenta valiosa para equipes de negócio, dado o dinamismo na sua utilização, o uso dessas ferramentas traz segurança para os dados que estão em um banco, pois através do dashboard não é possível modificar os dados que estão sendo visualizados.

3.9 METODOLOGIA DE TRABALHO COM POWER BI

O Power BI (PBI) é um e um *software* da Microsoft, utilizado para construção de *dashboards*. Através dele é possível relacionar e conectar dados, e posteriormente

Figura 6 – Exemplo de um dashboard



Fonte: Microsoft - 2021

visualizar estes dados. Dentre as vantagens de se utilizar o Power Bi, uma delas é que o usuário consegue se conectar diretamente a um DW ou a um BD.

A partir disso, o usuário consegue utilizar os dados que foram armazenados para construir os seus painéis. A arquitetura de dados do Power BI é baseada em tabelas fato e dimensão, assim como é comumente visto no DW. No entanto, por ter a arquitetura de dados baseada em modelagem dimensional, ao conectar dados de um banco ao BI pode ser necessário tratar o dado, caso o dado esteja modelado de maneira diferente.

O trabalho no PBI, de maneira simplificada, inicia com a conexão com uma fonte de dados, que pode ser desde uma (ou várias) tabela no Excel, um arquivo no formato *Comma separated values* (csv), um BD ou até um DW. Após conectar-se a sua base de dados, é possível selecionar apenas os dados com os quais deseja-se trabalhar, para posteriormente realizar algum eventual tratamento nas suas tabelas. Esse tratamento pode ser tanto nos dados das tabelas, como também no relacionamento entre as tabelas.

Após este eventual tratamento nos dados, devem ser feitas as ligações entre

tabelas dimensão e tabela fato. Essa ligação entre tabela dimensão e tabela fato muitas vezes ocorre de forma automatizada se as tabelas estão bem modeladas e com os nomes corretos em suas chaves. Aqui, se um banco está modelado de forma relacional, podem ocorrer problemas e dificuldades em tratar com as tabelas, pois é necessário trocar a modelagem de relacional para dimensional.

Por fim, a última etapa é a etapa onde são de fato construídos os *dashboards*. Nesta etapa, embora o usuário possa construí-los diretamente é uma prática comum primeiro montar um esboço através do *powerpoint*, para posteriormente passar este esboço para o PBI.

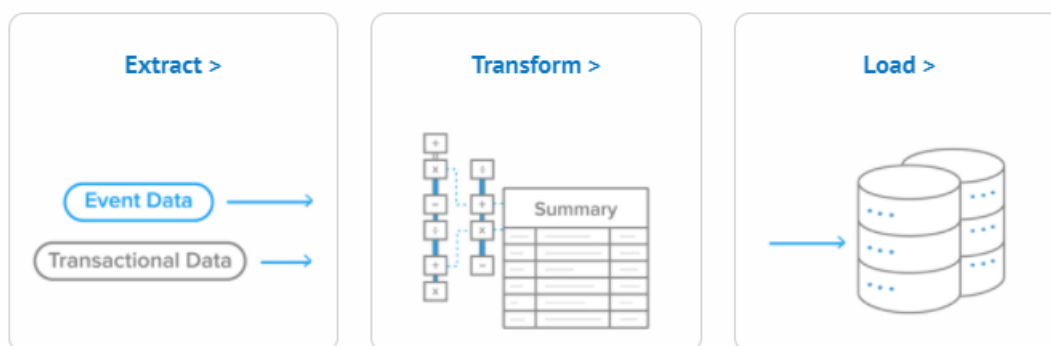
Esta construção do esboço via *powerpoint* permite mapear com o usuário final dos painéis suas necessidades, de forma simples de se construir. Esse *mockup* do que será feito acaba evitando retrabalhos e facilita na hora de construir os *dashboards*.

3.10 ETL

O processo de ETL, do inglês Extraction-Transformation-Loading (Extração-transformação-Carregamento), é um processo que destina-se a extração e transformação de dados de algum lugar, finalizando com a inserção destes dados em um *Data warehouse* (DW), ou ainda, em aplicações mais simples e com um volume baixo de dados, ou para provas de conceito, pode ser utilizado apenas um banco de dados (BD).

Na figura 7 está ilustrado o funcionamento do processo de ETL. ETL é um processo para extrair dados de uma ou mais fontes de dados, sendo estas fontes estruturadas ou não, para posteriormente transformar estes dados (conversão, limpeza, enriquecimento) e carregá-los em uma base dentro de um DW ou BD.

Figura 7 – O processo de ETL



Fonte – Stitch Data - 2021

Detalhando os processos vistos na Figura 7, primeiro o processo de extração (extract), onde são obtidos os dados. Muitas vezes estes dados são coletados de

diversas fontes de dados (tabelas, ou até mesmo bases diferentes) e enviados para a área de transformação, para posteriormente serem trabalhados. Este processo de extração pode ser desenvolvido de diversas maneiras, e suas fontes de dados podem ser diversas, como sites na internet, planilhas em Excel, bases de dados públicas como a da receita federal, ou até mesmo dados de um CRM.

O grande desafio que se encontra na maioria das vezes, no jeito em que o dado está disponível. Muitas vezes os dados não estão disponíveis de maneira estruturada em um banco de dados ou uma planilha única, os dados podem estar muitas vezes disponíveis até em arquivos PDF.

Portanto, para realizar a extração destes dados, o uso de soluções automatizadas se torna essencial. Há diversas soluções disponíveis no mercado, assim como bibliotecas nas mais diversas linguagens de programação. Alguns dos comumente utilizados são os algoritmos de *crawling* e *web scraping* que são capazes de buscar e indexar as informações que encontram em páginas. Muitas vezes, é necessário utilizar algoritmos de aprendizado e/ou de otimização junto, para conseguir realizar essa busca de maneira mais eficiente.

Em algumas situações específicas, quando o volume de dados é baixo ou o gasto de tempo para automatizar é superior ao o tratamento manual, esse dado pode ser extraído manualmente e posteriormente se estrutura um processo de extração automatizado. Ou seja, para dados passados é feita a extração manual, e dados futuros já nascerão no formato e local adequando, sendo necessário apenas um eventual tratamento antes de carregá-lo para o DB ou DW.

A etapa de extração do dado, é onde comumente se encontra a maior dificuldade em ETL, visto que os dados não são estruturados e é improvável que exista uma solução já pronta para ser usada, assim como vai ser exposto a frente neste relatório, este processo geralmente leva sucessivas iterações até que a extração esteja satisfatória. Conforme mencionado acima, a etapa de extração pode servir como base para desenhar um processo para lidar com os dados futuros, garantindo que os novos dados já sejam originados de forma estruturada e organizada.

Na etapa seguinte, ocorre a transformação (*transform*) dos dados. Os dados extraídos de suas fontes são analisados para que então seja possível definir quais atividades deverão ser realizadas, atividades como limpeza, eliminação de campos ou dados que não serão úteis ao DW, combinação de fontes de dados e enriquecimento de base, remoção de duplicatas ou até mesmo normalização dos dados.

Dentro desta etapa de transformação, é necessário ter modelado como será a estrutura dos dados, assim como o formato em que o dado deve ser exposto. Quando transformado, aquele dado que antes estava disponível de uma maneira não estruturada, agora irá ficar disponível em um formato padrão de banco de dados.

Esse formato irá depender da especificação do projeto e de como foi feita a sua

modelagem. A partir desta modelagem são criados os modelos de tabelas utilizados. Novamente, aqui existem diversas técnicas e métodos que podem ser utilizados. Desde a convencional modelagem Entidade-Relacionamento usada em BDs relacionais, até a mais comumente usada em DW, a modelagem dimensional.

Além disso, quando se trabalha com projetos com grande volume de dados, existem ferramentas e bibliotecas que auxiliam na transformação deste dado. Uma comumente utilizada é a Data building tool (DBT), que permite transformar o dado já dentro do DW, elevando a performance do processamento. Visto que DW mais modernos são extremamente performantes e escaláveis, a transformação do dado pode então ser feita diretamente no DW, em um processo conhecido como ELT (Extração, Carga e Transformação), que altera a ordem dos processos de Loading e Extraction padrão de ETL

No último passo, a carga é feita (*load*). Os dados, que já foram extraídos e transformados, agora são carregados no DW OU NO BD (ou ainda, se foi realizado ELT estes dados já estão no DW). Isto requer checagem da integridade e qualidade dos dados (validação), otimização do processo de carga e suporte às necessidades do processo de carga (HOKAMA *et al.*, 2004).

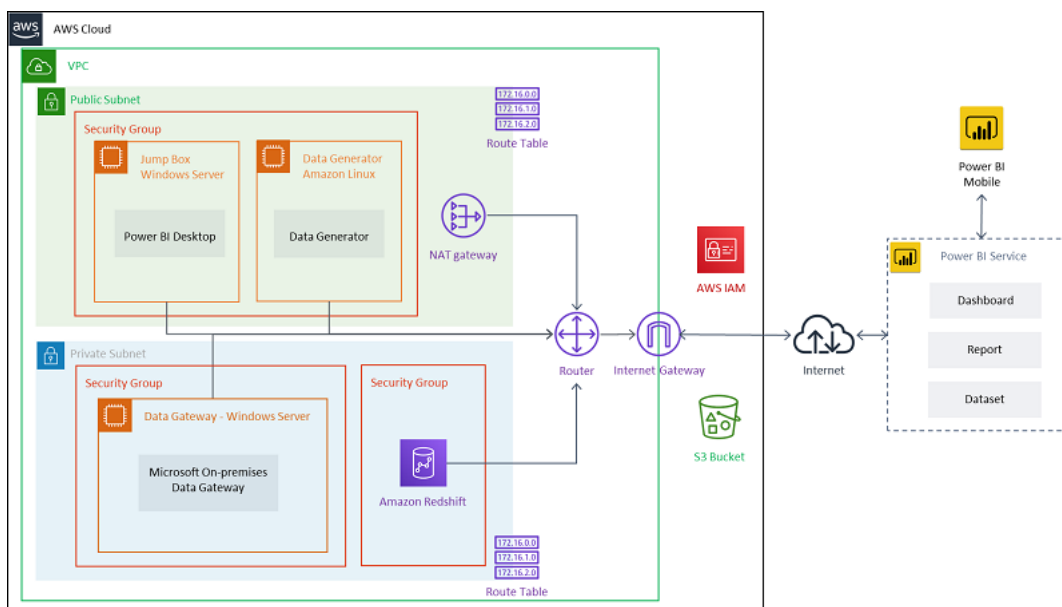
Aqui também existem ferramentas que auxiliam no processo. Muitas das ferramentas que são utilizadas na etapa de transformação, quando escolhidas corretamente, também auxiliam no processo de carga do dado no DW. Nessa etapa de carga do dado, é necessário atentar-se a dados duplicados quando se trata de carga em um banco de dados relacional. Se a etapa de tratamento dos dados não lidou com dados duplicados, pode ser necessário utilizar de alguma ferramenta ou método na hora de carregar o dado para o banco, para garantir a não duplicidade dos dados.

Após o dado estar disponível do DW, não basta simplesmente acessá-lo também. Novamente, vão existir diversas maneiras de se conectar com o seu dado, e depende da aplicação final dele. Quando se deseja conectar o dado ao Power BI, e utiliza-se do serviço da AWS, com o Amazon Redshift (DW nativo em *cloud*), a maneira mais comum e segura de se fazer isso, é através da utilização de um *gateway* (LEE; HO, 2018) . Essa ferramenta permite a conexão entre a rede privada de empresa e a Internet na AWS, permitindo assim o acesso ao dado. Na figura 8, vemos a estrutura do DW (Amazon Redshift) e o *gateway* fazendo a porta com o Power Bi.

Em aplicações pequenas, quando se utiliza um banco de dados relacional ao invés de um DW, há diversas outras maneiras de se conectar com esse dado. Se o banco de dados é um banco local, a conexão com ele pode ser feita de maneira direta utilizando as suas credenciais. No entanto, se o banco de dados está em nuvem (*cloud*), a conexão tem de ser realizada de outras maneiras.

A documentação da AWS recomenda 4 maneiras (AWS, 2021b) diferentes de se conectar um banco de dados em nuvem ao Power Bi. Assim como na utilização de um

Figura 8 – Conexão do DW com PBI via gateway



Fonte – Amazon Web Services

DW também é possível conectar ao DB utilizando um *gateway*. Outra opção é utilizar um *Open Database Connectivity* (ODBC), que é um padrão para acesso a sistemas gerenciadores de bancos de dados. Por fim, provedores de serviço de computação em nuvem também permitem hospedar o Power BI Desktop, facilitando a conexão direta ao banco.

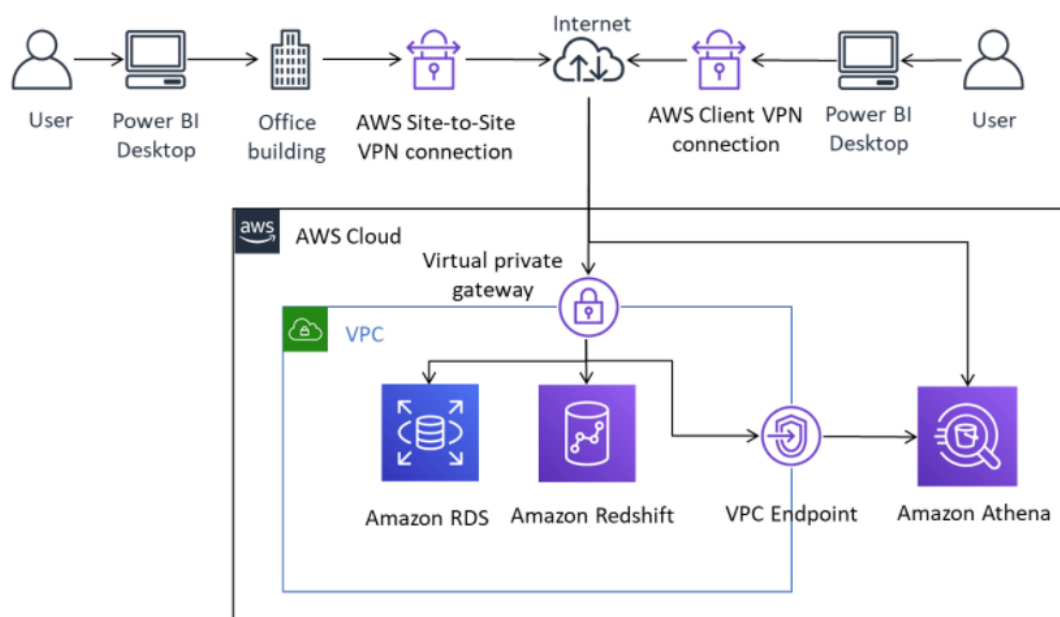
Para conectar o Power BI que está em sua máquina localmente, e os dados em nuvem, é possível utilizar VPN (Virtual Private Network) site-a-site da AWS (AWS Site-to-Site VPN) ou VPN cliente AWS (AWS Client VPN). Cada tipo de conexão oferece uma solução VPN em nuvem elástica, gerenciada e altamente disponível para proteger o tráfego da rede.

De acordo com a AWS (2021b): "A VPN site-a-site cria canais criptografados entre sua rede e sua Amazon Virtual Private Cloud (Amazon VPC) ou AWS Transit Gateway. A VPN cliente conecta seus usuários à AWS ou a recursos locais usando um cliente de software VPN gratuito". A Figura 9 detalha como é feita a conexão

Outra opção para conectar o banco ao PBI, é utilizar um *Internet Gateway*, de maneira semelhante ao que é feito com o DW (Amazon Redshift), que é detalhado na Figura 8.

Por fim, este processo, muitas vezes acaba sendo um processo contínuo. Muitas vezes é necessário extrair o dado dada uma certa frequência, portanto, existem também ferramentas, como o *Apache airflow*, que realizam a orquestração destas inúmeras rotinas, criando dependências entre elas, garantindo que um processo ocorra após o outro. Assim, é possível manter o dado final correto e atualizado para quando seu uso for requerido.

Figura 9 – Conexão do DB com PBI via VPN



Fonte – Amazon Web Services

Em aplicações menores, ou ainda em aplicações de custo reduzido, nem sempre é viável utilizar essas ferramentas. Uma alternativa a isso é utilizar ferramentas *Open Source* de *Robotic process automation* (RPA), que podem replicar atividades braçais de carga dos dados. Ou ainda, é possível desenhar um processo manual, porém estruturado, onde é feita a carga no banco de dados dentro de um certo período de tempo.

O processo de ETL é um processo complexo, que exige coordenação entre diversas áreas dentro de um projeto, como também entre diversas ferramentas. O processo de ETL é a base do projeto e sem um processo bem feito, todo restante é inviabilizado. O resultado final do ETL é um dado estruturado em um DW ou BD, para que o usuário, através de alguma ferramenta como um *gateway*, consiga acessá-lo e utilizá-lo de maneira simples.

3.11 SISTEMAS DE APOIO A TOMADA DE DECISÃO

A tomada de decisões é um processo que deve alinhar a melhor possibilidade de solução de problemas com as estratégias da organização visando considerável aprimoramento na eficiência e eficácia empresariais.

Diversos fatores influenciam a tomada de decisão, como o conhecimento e experiência de quem toma essa decisão, informação que essa pessoa possui, o risco envolvido, entre outros.

Pode se dizer que existem três tipos de problemas (PÁDUA, 2020):

- Problemas estruturados: problemas de rotina repetitivos (soluções padronizadas);
- Problemas não estruturados: complexos e imprecisos. A inteligência, o projeto e a escolha não estão organizados em uma sequência específica.
- Problemas semi-estruturados: combinação de solução padrão e julgamento individual.

“Um sistema de apoio a tomada de decisão (SAD) é um sistema de informação computadorizado que combina modelos e dados em uma tentativa de resolver os problemas semi-estruturados e alguns problemas não estruturados, com intenso envolvimento do usuário.” (TURBAN; RAINER; POTTER, 2007)

Portanto, o SAD irá auxiliar ao usuário do sistema, junto de outras informações, ou da experiência da pessoa, a resolver um problema específico. Vale ressaltar que SADs são uma área científica complexa, e neste trabalho será abordado o tema de maneira superficial como prova de conceito.

4 ESPECIFICAÇÃO DO PROJETO

Neste capítulo serão abordados dois temas: descrição, de maneira conceitual, de como foi proposta uma solução para o problema tratado; e, tendo exposto a solução conceitual, será descrita como ficou a solução final do sistema desenvolvido.

Antes de iniciar a exposição do problema, é válido relembrar o problema inicial tratado neste documento. A fintech que trabalha com crédito como serviço. Há uma empresa parceira que utiliza a plataforma CaaS da Flow para prover crédito para sua base de clientes.

Essa operação tem os seus dados armazenados em diversos locais diferentes e essas fontes de dados não estão relacionadas. Para acessar esses dados e realizar consultas entre as fontes, é necessário realizar isto de maneira manual.

Portanto, se torna custoso acompanhar os dados da operação, assim como realizar análises em cima desses dados.

Existem duas partes que precisam ter acessos a estes dados de forma estruturada e fácil. A própria fintech, e a empresa parceira que utiliza do serviço da Flow.

Cada uma das partes possui interesses distintos. A fintech quer utilizar os dados para controlar melhor a sua operação, visualizar dados de forma macro e micro, visualizar indicadores da operação, construir relatórios de cobrança de forma automatizada e, por fim, utilizar os dados para melhorar o processo de análise do limite de crédito que é fornecido para cada cliente da empresa parceira.

A empresa parceira possui um interesse diferente. A parceira necessita de acesso aos dados por questões de *compliance* (conformidade legal). No entanto, não basta simplesmente inserir estes dados em um banco e prover acesso a eles.

Por questões de segurança dos dados, eles não podem ter acesso direto ao banco. Eles também precisam ter acesso de forma fácil e dinâmica aos dados, algo que o banco não fornece. A equipe de negócios da empresa parceira precisa realizar consultas aos dados sem precisar realizar *queries* (consultas). O processo deve ser simples como utilizar um aplicativo.

Contextualizado o problema, pode-se partir para a definição de uma solução que consiga atender a essas demandas. De maneira, sucinta em linguagem natural, o que será preciso fazer é reunir todos os dados em um único local, a partir destes dados, criar algum sistema para visualizar os dados e também um sistema de apoio a decisão.

Além disso, será necessário desenhar um processo de como será feita a atualização periódica dos dados da operação, para garantir que a solução será duradoura.

Por fim, deve-se encontrar uma maneira de compartilhar o sistema desenvolvido com a empresa parceira, para garantir que eles possuam acesso ao dado desejado.

4.1 ESBOÇO DA SOLUÇÃO

O primeiro passo para chegar na concepção de uma solução, antes mesmo de se levantar requisitos funcionais e requisitos não funcionais do projeto, é mapear um esboço do problema e sua solução, possibilitando ai partir para requisitos funcionais e não funcionais do sistema.

Tudo parte de um conjunto de dados que deseja-se utilizar, e está espalhado por diversas tabelas, armazenadas em locais distintos. Aqui, surgem os seguintes questionamentos:

1. É possível obter esses dados sem ser via tabela?
2. Os dados precisarão de que tratamento?
3. Qual o custo e tempo para tratar esse dado manualmente vs. usando uma solução automatizada que precisará ser criada?
4. O dado que temos acesso atenderá as demandas da fintech e da empresa parceira?
5. Como lidar com os dados futuros para evitar novos gastos com tratamento?

Partindo desse conjunto de dados, após reuni-los e tratá-los, será necessário salvá-los em algum lugar. Existem diversas opções de onde e como salvar esse dado, sendo as principais:

1. Armazenamento local em arquivos ".csv";
2. Armazenamento online (Google Drive) em arquivos ".csv";
3. Armazenamento em um banco de dados local;
4. Armazenamento em um banco de dados na nuvem;

Existem diversas outras opções de armazenamento, porém dadas as limitações técnicas, de custo e tempo, essas foram as principais elencadas.

Independente da solução escolhida, sabe-se também que será necessário modelar o relacionamento desse dado, e, no caso de um banco de dados, utilizar mecanismo de banco de dados e um SGBD para gerir o banco e seus dados.

Tendo definido como e onde armazenar esse dado, o próximo passo é voltado à uma solução para expor esse dado. Aqui, embora existam outras opções, o foco converge em softwares de *business intelligence* (BI) já existentes. Nesse caso, foram analisadas duas principais opções, o Power BI da Microsoft, e o Tableau, da Salesforce. Outra alternativa é desenvolver os dashboards via código e isto é impraticável em termos de custo e tempo.

Após escolher uma ferramenta de BI, é preciso conectar essa ferramenta com os dados, realizar alguma eventual modelagem e tratamento nos dados dentro da ferramenta, para ai poder montar os painéis. Ambas as ferramentas, tanto Power BI quanto Tableau possuem maneiras de compartilhar seus painéis online.

No entanto, é necessário mapear as necessidades de quem irá consumir o dashboard, para poder de fato construí-lo. As necessidades internas a fintech já estão mapeadas, e as necessidades da empresa parceira já haviam sido elencadas em uma conversa inicial. Aqui levanta-se o questionamento de que poderia ser necessário mais conversas para chegar em uma solução melhor. No entanto, por uma decisão de negócios, optou-se por realizar o projeto apenas baseado na conversa inicial com a empresa parceira, e posteriormente após apresentar a solução, realizar os devidos ajustes caso seja necessário.

Por fim, a última etapa é a construção do sistema de apoio a decisão. Esse sistema deve auxiliar no processo de definição da linha de crédito que é disponibilizada para um cliente da empresa parceira.

Para chegar nas especificações do sistema, se faz necessário realizar um estudo estatístico da base de dados da operações, focado nos limites de crédito existentes. Esse estudo servirá como base para desenhar o sistema.

Tendo a base estatística para criar o sistema, também se faz necessário encontrar uma maneira de acoplar o sistema ao software de exposição do dashboard. Isso pode ser feito usando linguagens nativas ao Power BI ou Tableau, ou então utilizando alguma linguagem de programação compatível com os softwares, como *Python* ou *R*.

Dada esse esboço do problema, e a visão genérica do que se busca na solução, pode-se partir para criação dos requisitos do sistema.

4.2 REQUISITOS DO SISTEMA

Primeiro, foca-se em mapear os requisitos funcionais do sistema. Os requisitos funcionais são todos os problemas e necessidades que devem ser atendidos e resolvidos por meio de funções ou serviços. Traduzindo o que foi levantado no esboço, e dadas as necessidades tem-se os seguintes requisitos não funcionais:

- consultar operações de crédito realizadas;
- consultar as parcelas do mês de cada cliente;
- consultar o faturamento mensal de cada cliente;
- consultar o histórico de pagamentos;
- gerar relatório de cobrança mensal;
- gerar um número indicando o limite de crédito ideal de cada cliente;

Dessa lista de requisitos, apenas os dois últimos não são requisitos que vão atender demanda da empresa parceira, e sim da fintech. Nota-se que as necessidades de ETL, modelagem dos dados e afins não entram nesses requisitos. Inicialmente grande parte desse processo será feito manualmente, visto que o foco principal do sistema nesse momento é desenvolver uma solução minimamente viável, que atenda todos os requisitos.

Tendo os requisitos funcionais, e o esboço do problema levantado na seção anterior, desenha-se os requisitos não funcionais do sistema. Os requisitos não funcionais são todos aqueles relacionados à forma como o sistema tornará realidade o que está sendo planejado. Os requisitos não funcionais descrevem como serão feitos, sendo eles:

- O sistema deverá ficar disponível para uso online;
- O sistema deverá utilizar um banco de dados *open source*;
- O sistema não deve permitir alterações nos dados do banco por parte dos usuários;
- O sistema deverá comunicar com um banco hospedado na nuvem;
- O sistema só deverá permitir o acesso de usuários autorizados;

Tendo os requisitos funcionais e não funcionais, parte-se para a definição da arquitetura do sistema.

4.3 DEFINIÇÃO DA ARQUITETURA DO SISTEMA

A definição da arquitetura do sistema parte desde o início, de onde surge o dado, até o ponto final que é o dashboard. Ou seja, é o caminho que dado deverá percorrer até chegar ao dashboard.

Nesta etapa estão englobadas as decisões que respondem parte dos questionamentos levantados na seção 4.1. A arquitetura do sistema deve atender aos requisitos levantados previamente também.

4.3.1 Estrutura de dados

Os dados já utilizados hoje são o ponto de partida para entender o que é possível se fazer e o que precisará ser feito em termos de estrutura de dados.

Para entender a estrutura de dados, faz-se necessário recapitular como funciona a operação de crédito como serviço.

Quando um cliente da empresa parceira decide tomar crédito, uma CCB é gerada através da plataforma online da *fintech*. Essa CCB é registrada na instituição financeira responsável pelo *funding* da operação.

Todo mês é informado a empresa parceira qual o valor da parcela que esse cliente deve pagar. Essa parcela pode ser de única, ou então é a composição diversas parcelas caso o cliente tenha tomado crédito mais de uma vez.

Tendo quanto o cliente faturou, a *fintech* informa a empresa parceira quanto que ela deverá reter do faturamento de cada cliente, e quanto que o cliente deverá receber após o desconto das parcelas. Esse processo de cobrança é feito através de planilhas compartilhadas.

Hoje, há uma série de planilhas que compõem a estrutura de dados interna e controlam essa operação:

- Planilha de operações realizadas;
- Planilhas mensais de cobrança e vencimento;
- Planilhas mensais com todas as parcelas;

A planilha de operações é a principal planilha de controle interno. Nela encontram-se todas as operações de crédito realizadas. Além das operações, a partir dela é possível obter a lista de todos os clientes que já tomaram crédito, como também as informações desses clientes, como nome fantasia da empresa, razão social, CNPJ e ID vinculado a empresa parceira.

Toda operação de crédito possui um código. Toda operação é vinculada a um CNPJ. Todo CNPJ é vinculado a um ou mais *IDs* de controle interno da empresa parceira.

Inicialmente supos-se que a partir dessa planilha que seriam originados os dados de cada operação realizada, assim como a relação de cada cliente que já tomou crédito.

No entanto, por não ser a fonte da verdade, originar o registro de cada operação a partir da "planilha de operações realizadas" abriria uma margem muito grande para ocorrerem erros. Essa planilha pode ser utilizada apenas como lista de quais clientes já tomaram crédito. A lista de operações tem de ser originada de maneira diferente.

Como a instituição financeira é responsável pela emissão das CCBs, todos esses dados estão salvos no banco de dados deles. A instituição financeira é quem fornece o *funding* para essa operação.

A partir dos dados da instituição financeira são originadas as planilhas com as parcelas mensais de cada cliente. Essa planilha é tratada, e compartilhada com a empresa parceira. Essa planilha tratada é a planilha de vencimentos mensais. Nela,

ficam os dados de faturamento de cada cliente, assim como quanto foi cobrado de cada cliente cada mês.

Como os dados da instituição financeira são a fonte da verdade. Para garantir a acurácia dos dados das operações originadas, das parcelas do mês e do histórico de parcelas pagas e em aberto, deve-se utilizar os dados que estão armazenados no banco de dados da instituição financeira.

As únicas informações que devem seguir originando de planilhas são a lista de clientes, e a planilha de vencimentos mensais, que é a ferramenta utilizada entre a fintech e a empresa parceira para realizar a cobrança.

Dado o contexto acima, o resultado final da parte de estruturação de dados é apresentado no Quadro 1. Ali, constam todos os dados que serão utilizados e qual será a sua origem.

Quadro 1 – Fontes de dado

Lista de clientes	Planilha interna de cadastro das operações.
Relação dos CNPJs e IDs do parceiro	Planilha interna de cadastro das operações.
Vencimentos mensais	Planilhas de vencimento - há uma por mês
Operações realizadas	Dado fornecido pela instituição financeira em ".csv"
Parcelas do mês	Dado fornecido pela instituição financeira em ".csv"
Histórico de todas as parcelas	Dado fornecido pela instituição financeira em ".csv"

Fonte – Autoria própria

Dadas as fontes de dados citadas no Quadro 1, o próximo passo é ter esse dado tratado para inserir em um banco de dados. Para tratar os dados passados existem duas alternativas: realizar o tratamento usando alguma ferramenta de automação, ou realizar o tratamento manual.

Como o volume de dados é baixo, há um total 12 tabelas de vencimento e a tabela de cadastro das operações é única, optou-se por tratar este dado manualmente, para validar a solução, e posteriormente desenvolver um processo para tratar os dados automaticamente. A relação tempo x resultado não justifica a criação de uma ferramenta automatizada desde o início. Portanto, a única automação utilizada será para tratar duplicidades.

Os dados que originam da instituição financeira, já vem tratados e no formato correto. A única necessidade é, para a tabela de Operações realizadas, inserir apenas as novas no banco de dados, para evitar duplicidades. Todo o tratamento de duplicidades será feito através de um SGBD.

Tendo definido quais serão as fontes dos dados, também será necessário desenhar um processo de atualização destes dados, e com que frequência isso será feito. Este processo será abordado na seção de desenvolvimento.

O passo seguinte da arquitetura é onde este dado será armazenado.

4.3.2 Armazenamento do dado

Conforme mencionado na seção 4.1, existem diversas combinações de como pode ser feito o armazenamento. Os requisitos para o armazenamento do dado são os seguintes:

- a solução de armazenamento deve ser gratuita ou gratuita por um período para validar a solução;
- o local de armazenamento deve ser gratuito ou gratuito por um período de tempo;
- o dado deve ser acessível de qualquer local;

A solução inicialmente pensada era utilizar uma pasta em algum serviço de armazenamento em nuvem, como o Google Drive por exemplo, e lá guardar arquivos ".csv". Para utilizar o dado bastaria conectá-lo em um software de BI. Essa solução foi logo descartada, por não ser escalável, e não permitir o gerenciamento do dado.

A solução para o armazenamento precisa de robustez, e essa robustez só será fornecida por um banco de dados. Para que esse banco seja acessível de qualquer lugar, ele pode ficar em um servidor online, ou em algum serviço de armazenamento em nuvem. Os serviços de armazenamento em nuvem mais populares oferecem 12 meses gratuitos, e a fintech utiliza já um serviço de Cloud. Portanto, a escolha final foi por utilizar um banco de dados na nuvem.

A escolha da *engine* do banco de dados seguiu o critério de busca por soluções *open source* e que fosse compatíveis com as principais plataformas de cloud. Dentre as principais opções, havia MariaDB, MySQL e PostgreSQL. Todas estas são compatíveis tanto com Amazon Web Services quanto com o Azure.

Devido a familiaridade e a existência de um SGBD gratuito (*PgAdmin*), optou-se por utilizar PostgreSQL como *engine* para o banco de dados. Tendo definido isso, e tendo o SGBD, o próximo passo é definir onde esse banco irá ser armazenado.

As duas opções existentes de serviço de nuvem são AWS ou Azure. Ambas as soluções apresentam 12 meses de serviço gratuito para testes. A utilização do Azure traz uma vantagem grande em relação ao AWS caso a ferramenta utilizada para construir os *dashboards* seja o Power BI. Como são feitos pela mesma empresa, a conexão entre o banco na nuvem e o software de BI é facilitada.

Embora a preferência fosse por utilizar o Azure, como o AWS já é utilizado na fintech, optou-se por hospedar o banco no AWS. Essa é uma decisão que dificulta um

pouco em termos de solução, mas ao pensar em escalabilidade e custos futuros, essa é a melhor opção. Inicialmente será criado um novo ambiente na nuvem, e não será utilizado o da fintech. Posteriormente a migração pode ser feita facilmente para o AWS da empresa.

Tendo definido a estrutura dos dados e o armazenamento, o próximo da arquitetura do sistema é definir como serão construídos os painéis e qual será o software utilizado para isso.

4.3.3 Software para construção dos dashboards

O software utilizado para construir os *dashboards* deve atender aos seguintes requisitos:

- possuir maneira de conectar ao banco de dados na nuvem;
- possuir uma plataforma para disponibilizar os *dashboards* online;
- possuir controle de acesso ao *dashboard* por usuário;

Ambas as soluções que atendem aos requisitos (Power BI e Tableau) são disponíveis na versão *desktop* de forma gratuita. No entanto, para publicar os painéis online, em ambos é necessário contratar o serviço (SaaS).

Em termos de custos o Tableau apresenta vantagens, no entanto, dada a familiaridade e a documentação mais ampla do Power BI, optou-se por utilizar este como software de construção dos *dashboards*.

A última etapa de definição da arquitetura do sistema é a definição do sistema de apoio a tomada de decisão, que auxiliará a definir o limite de crédito de cada cliente.

4.3.4 Sistema de auxílio a decisão do limite de crédito

Sistemas de apoio a tomada decisão são um modelo genérico, onde variáveis são analisadas que seja possível o posicionamento a uma determinada questão. SADs possuem uma grande complexidade, e o que deverá ser desenvolvido nesse primeiro momento no trabalho é apenas um sistema para auxiliar as decisões, que é baseado em modelos matemáticos simples.

O sistema deve informar, através dos dados de vencimento (faturamento) de cada cliente e de quanto ele já contratou em crédito, qual o seu limite de crédito ideal.

Esta é a última parte da arquitetura do sistema e inicialmente pretendia-se desenvolver um sistema em Python, visto que esta linguagem de programação é *open source*, consegue integrar com Power BI e Tableau para expor o resultado no próprio *dashboard* e também é de fácil utilização e de ampla documentação.

Ao chegar na etapa final do projeto, a decisão de utilizar Python mudou. Dentro do Power BI existe uma linguagem de programação chamada "*DAX*". *DAX* é uma

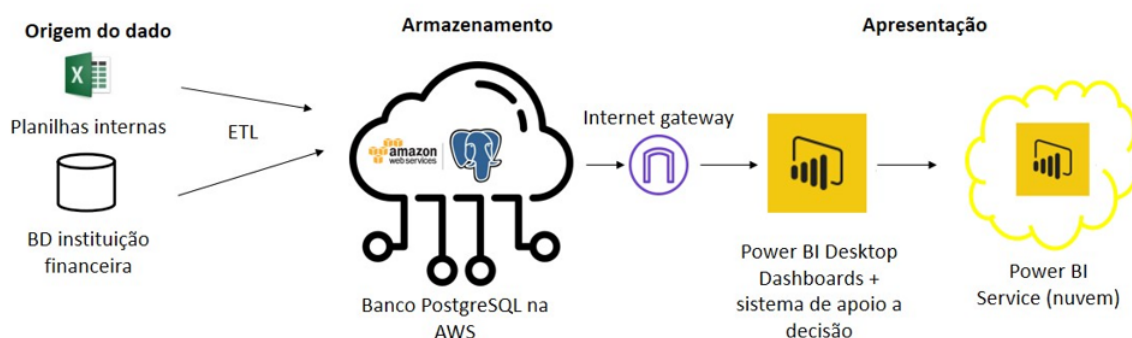
linguagem voltada para *data analytics* que pode ser usada para análises de dados aprofundadas.

Embora seja factível desenvolver o sistema em Python, pivotar para DAX é uma escolha que traz menos complexidade ao sistema, entrega o mesmo resultado e com a mesma robustez. A complexidade do sistema de apoio mora em entender os dados e desenhar uma formula para o calculo do limite de crédito. Esse estudo dos dados e o desenvolvimento serão expostos no próximo capítulo.

4.3.5 Visão geral do sistema

Dadas todas as informações expostas acima, é desenhado o diagrama da figura Figura 10. Neste diagrama é possível visualizar de forma macro, todos os componentes do sistema.

Figura 10 – Arquitetura do sistema



Fonte – Autoria própria

O caminho do dado inicia em três fontes diferentes:

- Planilha de controle de operações (Interna a fintech)
- Planilhas de vencimento mensal (Interna a fintech)
- Base de dados da instituição financeira (Enviada via ".csv")

Esse dado é tratado utilizando excel e *queries* no SGBD, e é inserido no banco de dados PostgreSQL, que está hospedado no AWS. Uma conexão entre o banco de dados e o Power BI é realizada, utilizando uma VPN. Tendo esse dado conectado ao PBI, ele é tratado internamente no próprio software, e o dado alimenta os *dashboards*. Dentro do Power BI é acoplado um sistema de auxílio a decisão do limite de crédito, que foi escrito utilizando DAX.

Os relatórios que tem de ser compartilhados, são compartilhados utilizando o Power Bi Service (Serviço de Nuvem) em uma Workspace online, onde os usuários que possuem autorização conseguem acessar os painéis.

4.4 ESPECIFICAÇÃO DO CONTEÚDO DOS *DASHBOARDS*

A empresa parceira tem foco em *compliance*, portanto o objetivo deles com os *dashboards* é ter acesso aos dados das operações, das parcelas e do pagamento das parcelas mensalmente. Além disso, é necessário que os dados das operações seja filtrado não apenas via CNPJ, que é o identificador padrão de uma operação e dos clientes, mas também utilizando o ID interno dos clientes na empresa parceira.

Na Tabela 1 tem-se esboço do conteúdo dos *dashboards*, e sua origem. A partir desse esboço, será relacionado o dado necessário em cada dashboard com o dado que estará no banco, e serão construídos os painéis.

Tabela 1 – Conteúdo dashboards empresa parceira

Painel	Conteúdo	Fonte
Operações realizadas	Histórico das operações por CNPJ e ID;	Tabela de clientes (Flow), Tabela de operações realizadas (IF) Histórico de vencimentos
Histórico de vencimentos	Compilado de todas as tabelas mensais de cobrança que são enviadas para a empresa parceira	Tabelas de vencimento (Flow)
Composição da parcela mensal	Todas as parcelas em aberto para cada CNPJ. Deve ser filtrada por ID.	Tabela de clientes (Flow), Tabela de parcelas do mês (IF)
Histórico de todas as parcelas	Para cada mês, histórico de todas as parcelas pagas e as em aberto para cada CNPJ. Deve ser filtrada por ID	Tabela de clientes (Flow), tabela de histórico de parcelas (IF)

Fonte – Autoria própria

Os painéis desenvolvidos para a empresa parceira também serão úteis para a fintech, porém há 3 demandas específicas que são apenas para a Flow, que são elas:

- Um painel para construir o relatório de pagamento mensal dos clientes;
- Um painel reproduzindo os gráficos de controle interno da operação;
- Um painel para auxiliar nas solicitações dos clientes na expansão de suas linhas de crédito;

Na Tabela 2 temos a descrição de como serão compostos esses painéis, e quais dados são utilizados em sua composição.

Tabela 2 – Conteúdo dashboards Flow Finance

Painel	Conteúdo	Fonte
Gráficos de controle interno da operação	Evolução dos empréstimos mês a mês (quantidade e volume); Acumulado dos empréstimos no tempo (quantidade e volume); Total de clientes x clientes únicos; Total de operações;	Tabela de clientes (Flow), Tabela de operações realizadas (IF)
Pagamento mensal dos clientes	Somatório das parcelas a serem pagas, tendo o valor total agrupado por CNPJ, e tendo o ID disponível junto.	Tabela de clientes (Flow), Tabela de parcelas do mês (IF)
Painel de auxílio nas expansões de crédito	Histórico de operações por cliente (CNPJ), Histórico de faturamento (4 meses) e Histórico de pagamento (4 meses)	Tabela de clientes (Flow), Tabelas de vencimento (Flow), Tabela de operações realizadas (IF)

Fonte – Autoria própria

Tendo definido a arquitetura do sistema e a especificação dos *dashboards*, é possível prosseguir para a próxima etapa, que é a etapa de desenvolvimento.

5 DESENVOLVIMENTO

O desenvolvimento do projeto inicia no processo de ETL, sendo que antes de carregar o dado no banco é necessário criar o banco na nuvem e suas tabelas. Posteriormente, tendo feita a carga dos dados, é feita a conexão do banco com o BI.

Tendo a conexão do banco com o BI, é feito o tratamento e modelagem do relacionamento das tabelas dentro do BI e então são construídos os dashboards. Por fim, é realizado o estudo que serve de base para desenvolver o sistema de apoio a decisão.

Tendo o estudo, é possível desenhar o sistema, e implementá-lo no Power Bi utilizando DAX. Essas serão as etapas de desenvolvimento abordadas a seguir.

5.1 REUNINDO OS DADOS DA OPERAÇÃO

O primeiro passo a ser dado no processo de ETL é a extração dos dados para um único local. Essa extração pode ser feita de diversas maneiras, e cabe ao desenvolvedor do projeto avaliar os custos e o tempo gasto com cada etapa.

Inicialmente, pretendia-se extrair todos os dados para uma pasta local, utilizando um algoritmo a ser desenvolvido. No entanto, com o volume de dados do projeto, a solução com menor gasto de tempo é exportar manualmente todas as planilhas para um local único.

Além de extrair as planilhas para um local único, também é necessário alinhar com a instituição financeira, que possui os dados das operações de das parcelas, um processo para que eles forneçam os dados contidos no banco de dados deles.

Como é necessário que a atualização dos dados seja mensal apenas, foi definido com a instituição financeira que, no primeiro dia de cada mês, a instituição financeira realizará uma consulta no banco de dados deles, e enviará os arquivos em formato ".csv".

Quanto aos dados internos da empresa, estes já estão reunidos em uma única pasta local, sendo agora necessário apenas tratá-los. Como esses dados são dados históricos, essa extração só precisa ser realizada uma única vez. Uma vez que esses dados forem para o banco, não será mais necessário lidar com eles.

5.2 TRATANDO OS DADOS

A etapa de tratamento dos dados remete a etapa *transform* do processo de ETL. Nesta etapa, é necessário selecionar quais dados irão para o banco, filtrar dados indesejados, formatar dados e afins.

Inicialmente, assim como na etapa de extração de dados, o planejamento do projeto considerava desenvolver um sistema ou algoritmo para realizar o tratamento

dos dados. A primeira carga feita no banco de dados iria exportar dados das seguintes planilhas:

- 14 planilhas de vencimentos mensais (dados históricos);
- 1 planilha com os dados cadastrais dos clientes;
- 1 planilha com a relação entre CNPJs e IDs de cada cliente;
- 1 planilha com o histórico de operações realizadas;
- 1 planilha com as parcelas a serem pagas mês;
- 1 planilha com o histórico de todas as parcelas pagas e em aberto de cada cliente.

As planilhas de vencimento mensal são onde se encontra a maior dificuldade. A decisão de como tratar esse dado considerou tempo necessário para desenvolver um *crawler*, o tempo para tratar esse dado manualmente no Excel, e o fato de que esse seria um esforço único para importar o histórico, uma vez importadas as planilhas históricas de vencimento, seria apenas necessário importar uma nova uma vez ao mês.

Dadas as considerações acima, optou-se por tratar as planilhas manualmente no Excel. O tempo gasto desenvolvendo um sistema para tratá-las de forma automatizada seria muito maior do que o tempo gasto para tratá-las de forma manual. Essa foi uma decisão que vai contrária ao que o estado da arte do processo de ETL prega, no entanto, é uma decisão que reduz significativamente o tempo de projeto.

Caso seja necessário, deve-se desenvolver posteriormente um sistema para tratar os dados da planilha, ou ainda, construí-las no formato onde seria necessário apenas transformá-la em um arquivo ".csv" e exportar para o banco.

O tratamento que essas planilhas exigiu foi a remoção de cabeçalhos, e inserção de uma coluna extra indicando qual a data em que foi realizada aquela cobrança.

A planilha com os dados cadastrais de cada cliente irá originar da planilha de cadastro das operações, conforme exposto no capítulo 4 Quadro 1. Essa planilha contém o CNPJ do cliente, o nome fantasia de sua empresa e a razão social. Para obter a lista de clientes é necessário realizar 3 etapas de tratamento do dado:

- Remover as colunas da planilha, mantendo apenas as colunas D, E e F;
- Remover as "/" e "." (barras e pontos) da coluna D, onde estão os CNPJs.
- Remover as duplicatas;

Para remover as colunas e os pontos e barras foi, novamente utilizado o Excel. Novamente, essa tratativa não é ideal, mas para validar a solução dentro de um tempo hábil, essa alternativa atende aos requisitos.

Por fim, para realizar a remoção de duplicatas, é utilizado um *script* em SQL na hora de inserir o dado no banco. Esse será explicado na seção de carga dos dados no banco. Vale apenas ressaltar que aqui uma duplicata é considerada toda linha que possui um CNPJ repetido.

A planilha com a relação entre CNPJs e IDs de cada cliente parte da mesma planilha mencionada acima. As etapas para realizar o tratamento do dado são semelhantes as anteriores:

- Remover as colunas da planilha, mantendo apenas as colunas C e D;
- Remover as "/" e "." (barras e pontos) da coluna D, onde estão os CNPJ.
- Remover as duplicatas;

O tratamento foi feito da mesma maneira que foi feito o tratamento da planilha anterior. A única diferença é que, para uma entrada ser considerada uma duplicata nesta tabela, deve ser considerada a combinação do CNPJ e do ID do cliente. Ou seja, para uma linha ser a duplicata da outra não basta ela possuir apenas o CNPJ igual. Ela necessita possuir a combinação de CNPJ e ID iguais.

As planilhas com histórico de operações realizadas, com as parcelas a serem pagas mês e com o histórico de todas as parcelas pagas e em aberto de cada cliente são obtidas diretamente da instituição financeira, através de uma consulta no banco de dados deles. Esses dados já vem tratados.

As planilhas de parcelas e histórico de parcelas são planilhas com dados novos todo mês, e são inseridas diretamente no banco. No entanto, na planilha de operações, a cada mês constarão as operações antigas e as que foram feitas no novo mês. Não serão enviadas apenas as novas operações.

Por consequência disso, é necessário realizar o tratamento do dado. Antes de inserir no banco, é preciso remover as operações que já estão salvas no banco de dados, e inserir apenas as novas.

Para realizar o tratamento dessas duplicatas é utilizado novamente um *script* em SQL, que foi detalhado na seção anterior. Vale ressaltar que, para uma operação ser considerada duplicada, ela deve conter o CNPJ e o código da operação iguais. Novamente, é uma entrada única é considerada a partir da combinação de duas colunas.

Na Tabela 3 estão as fontes de dado do projeto, e o tratamento necessário de cada fonte. Vale ressaltar que não estão sendo considerados os dados históricos da planilha de vencimentos.

Tabela 3 – Fontes de dado do projeto

Banco	Nome Quadro 1	Dados (colunas)	fonte	atualização
academias	Lista de clientes	CNPJ, Nome Fantasia, R. Social	Tabela de operações	Manter Coluna D, E e F. Remover "."e "/"e duplicatas
indice-id-cnpj	Relação dos CNPJs e IDs do parceiro	CNPJ, ID no parceiro	Tabela de operações	Manter Coluna C e D. Remover "."e "/"e duplicatas
hst-todas-parcelas	Histórico de todas as parcelas	CNPJ, Data consulta, Data vencimento, N° parcela, Código da CCB, Status, Valor receber, Parcela, Recebido	Fornecido pela IF	Sem tratamento necessário.
operacoes	Operações realizadas	CNPJ, CCB, Data aceite, Qtd. parcelas, Tx. Juros, Valor tomado	Fornecido pela IF	Remover duplicatas.
parcelas-mes-raw	Parcelas do mês	CNPJ, Data, Data vencimento, Atraso, Valor pago, Juros, Multa, N° da parcela, CCB, Parcela do mes, Status do Pgto, Valor face parcela	Fornecido pela IF	Remover duplicatas.
vencimento	Vencimentos mensais	CNPJ, data, ID, pgto., parcela, faturamento, faturamento liquido	Planilhas de vencimento	Remover cabeçalhos e inserir coluna com data da cobrança.

Fonte – Autoria própria

5.3 MODELANDO OS DADOS

A partir dos dados citados acima, é possível partir para a modelagem dos dados no banco de dados. Dado o banco escolhido (PostgreSQL), será realizada a modelagem ER dos dados, o modelo conceitual e a modelagem física.

Para construir a modelagem, é importante entender como que as entidades se relacionam, e qual a cardinalidade dessas relações também. Em linguagem natural abaixo, está exposta a concepção geral de como os dados se relacionam:

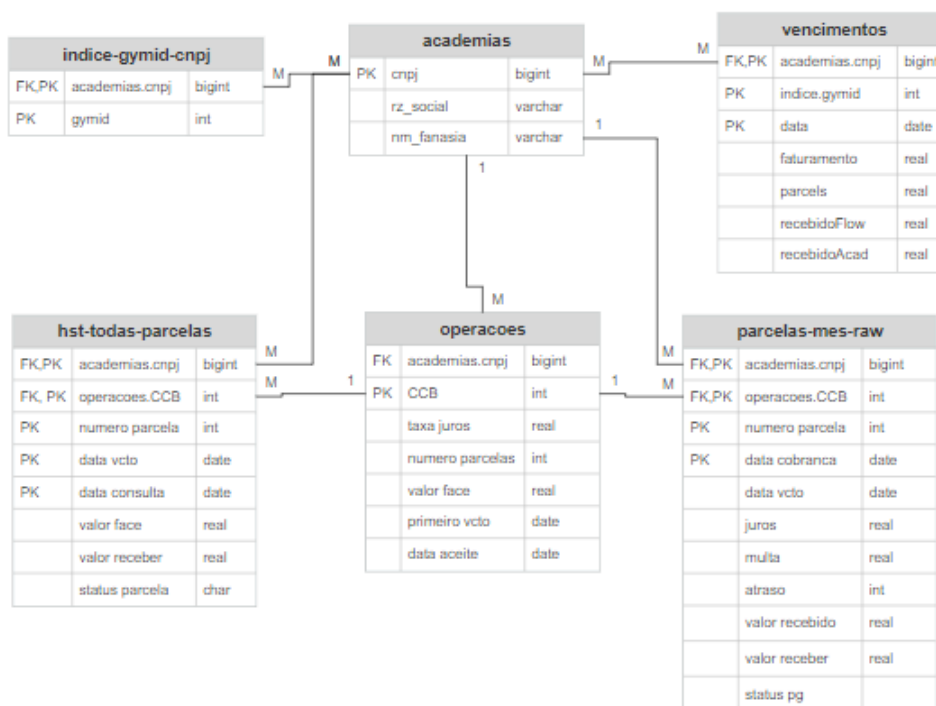
- Todo cliente possui um CNPJ. Esse CNPJ é único. Para cada CNPJ há uma única razão social;
- Um cliente pode originar n operações de crédito (originar uma CCB);

- Cada operação de crédito é identificada por um código **único**, chamado de código da CCB (ou apenas CCB).
- Cada CCB possui n parcelas.
- Cada parcela possui um número único, no entanto, a parcela pode aparecer mais de uma vez, em meses diferentes.
- Cada CNPJ possui 1 ou mais IDs do parceiro;
- Um ID do parceiro pode estar vinculado a mais de um CNPJ.

Essa é a concepção geral de como serão as relações entre as tabelas. Existe um total de 6 tabelas no banco, e a nomenclatura de cada tabela está exposta na seção anterior na coluna "banco", da Quadro 1.

Para modelar o relacionamento entre as tabelas, foi construído um diagrama físico, que está exposto na Figura 11

Figura 11 – Modelo físico do projeto



Fonte – Autoria própria

No diagrama está exposto o relacionamento entre as tabelas. Começando pela tabela "academias", onde o CNPJ é a chave primária desta tabela. Na tabela indice-id-cnpj a chave primária é composta pelo CNPJ e pelo ID, sendo que o ID é uma chave estrangeira também.

Na tabela de operações a chave primária é o código da CCB (código da operação), e a tabela possui o CNPJ como chave estrangeira. Na tabela parcelas-mes-raw, a chave primária é composta pelo número da parcela, código da CCB e data da cobrança. A tabela possui duas chaves estrangeiras, o CNPJ e o código da CCB.

A tabela com o histórico de todas as parcelas, chamada de hst-todas-parcelas tem a chave primária composta pela data de vencimento, numero da parcela, código da CCB e data de consulta. O código da CCB é uma chave estrangeira e o CNPJ também.

Por fim, na tabela de vencimentos a chave primária é composta pelo CNPJ, pelo ID e pela data da cobrança. O CNPJ é chave estrangeira nessa tabela, assim como o ID.

5.4 CRIAÇÃO DO BANCO DE DADOS

Conforme justificado no capítulo anterior, foi criado um banco em PostgreSQL para esse projeto. Para hospedar o banco, é utilizado o Amazon Relational Database Service (RDS), que é um serviço de banco de dados relacional distribuído da Amazon Web Services (AWS). Esse serviço é executado na nuvem, e foi projetado para simplificar a configuração, operação e dimensionamento de um banco de dados relacional.

Para criar o banco de dados, foi utilizada a função de criação de banco de dados da Amazon RDS, utilizando as configurações básicas, que são as que permitem utilizar a versão gratuita do serviço.

Para conectar no banco de dados, foi utilizado o SGBD PgAdmin. Através do PgAdmin é feita a criação das tabelas. Para criar as tabelas é possível utilizar a GUI do PgAdmin, ou escrever diretamente a query em SQL.

Figura 12 – Query para criar tabela

```
CREATE TABLE public.academias
(
  "CNPJ" bigint NOT NULL,
  rz_social text NOT NULL,
  "nome fantasia" text NOT NULL,
  CONSTRAINT "PK_CNPJ" PRIMARY KEY ("CNPJ")
);

ALTER TABLE public.academias
  OWNER to destri;
```

Fonte – Autoria própria

Na figura Figura 12 vemos um exemplo de código utilizado para criar a tabela

"academias". Nele são especificadas as colunas da tabela, o tipo de cada coluna, os seus atributos, e também quais são as chaves da tabela.

Na figura Figura 12 vemos um exemplo de código utilizado para criar a tabela "academias". Nele são especificadas as colunas da tabela, o tipo de cada coluna, os seus atributos, e também quais são as chaves da tabela.

5.5 CARREGANDO OS DADOS NO BANCO

Esta etapa do projeto é a etapa de *loading* do processo de ETL. Nesta etapa será realizada a carga dos dados no banco.

O carregamento dos dados no banco é feito diretamente através do SGBD. Conforme mencionado na seção de tratamento de dados, ao importar alguns dados, se faz necessário tratar a duplicidade deles.

Além disso, é necessário que os dados estejam em conformidade com ACID (Atomicidade, Consistência, Isolamento e Durabilidade).

Ao utilizar um SGBD para realizar a carga dos dados, caso o *payload* esteja fora da conformidade a carga dos dados não é executada.

Todas as tabelas que foram importadas foram importadas no formato ".csv" para o banco. As tabelas onde era necessário lidar com a duplicidade do dado, foi utilizado um *script* em SQL para tratar desse problema. Esse *script* cria uma tabela temporária, para onde os novos dados são movidos. A partir dessa tabela temporária, é consultado se o dado a ser inserido já está naquela tabela.

Ou seja, caso a chave que identifica uma linha da tabela como única na tabela destino possua um valor semelhante na tabela temporária, essa linha da tabela temporária é ignorada e o valor não é carregado para a tabela.

Para as tabelas onde não é necessário tratar a duplicidade, basta fazer a importação direta copiando, através de uma *query* (consulta) em SQL, os dados do arquivo ".csv" para a tabela no banco. Caso ocorra algum erro, e um dado estiver duplicado, o SGBD irá suspender a carga e apontará onde está o erro para que seja investigado. Garantindo assim que não irá ocorrer a carga de dados repetidos no banco.

5.6 CONECTANDO O BANCO AO POWER BI

Conforme apresentado ao final da seção Seção 3.10, existem diversas maneiras de realizar a conexão de um banco de dados com o Power BI. A maneira que essa conexão vai ser feita depende, principalmente, de onde esse banco está hospedado. Quando o banco está hospedado localmente, na mesma máquina em que está sendo desenvolvido o projeto no Power BI, basta utilizar a conexão direta do BI com o banco, no entanto, quando o banco está hospedado em nuvem, é necessário realizar a conexão garantindo a segurança dos dados.

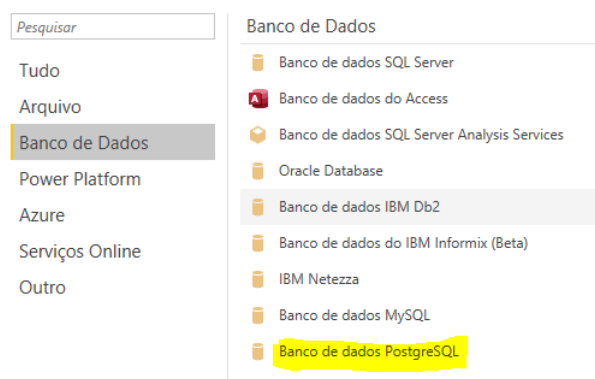
Neste trabalho foram consideradas duas maneiras de realizar a conexão com o BI, a conexão utilizando ODBC , a conexão utilizando AWS Client VPN e a conexão utilizando um *internet gateway*. Foram aplicadas as conexões via *internet gateway* e via ODBC, e ambas funcionam, no entanto, como existe uma documentação maior para realizar a conexão utilizando *gateway*, e também há mais segurança com os dados, optou-se por utilizar essa solução.

Para realizar a conexão do banco com o BI é criado um *internet gateway* para o banco desejado, e dentro das configurações dos grupos de segurança da VPC (*Virtual private cloud*) são criadas regras de conexão para IPs específicos.

Nessas regras de conexão é necessário especificar quais IPs estarão realizando a conexão com o banco. Posteriormente, é necessário obter *Access keys* para o usuário que deseja realizar a conexão com o banco.

Vale ressaltar que, caso o banco de dados houvesse sido hospedado utilizando o serviço da Azure, a conexão teria sido feita diretamente utilizando o conector específico do PBI para bancos de dados que estão hospedados na Azure, o que torna a configuração da aplicação mais simples.

Figura 13 – Fontes de conexão de dado com o Power BI



Fonte – Autoria própria

Tendo realizado os passos mencionados acima, é possível agora utilizar o conector padrão do Power BI, exposto na Figura 13. Utilizando a conexão com o PostgreSQL, basta inserir o nome que foi fornecido ao banco no campo "Database", no campo "Server" é inserido o *endpoint*, no campo "port" a porta utilizada (o padrão é 5432), e por fim, o usuário e senha do banco. Vale ressaltar que para esse usuário e senha que devem ter sido criadas as *Access Keys*.

Tendo feito isso, é possível importar os dados para o Power BI. Como o volume de dados é baixo, optou-se por carregar todos esses dados para o modelo, ao invés de utilizar a opção de "*direct queries*".

5.7 TRATAMENTO DO DADO NO POWER BI

Com os dados do banco carregados no BI, é possível iniciar o tratamento de dados no BI para a aplicação. Dois tratamentos de dados são necessários:

- Modelar a relação entre as tabelas;
- Alterar o formato dos dados para maior usabilidade;

A alteração do formato dos dados para garantir maior usabilidade consiste em modificações simples nos dados para garantir que a experiência do usuário seja boa. As principais alterações feitas são no formato das datas exibidas, para garantir que estão no formato padrão do Brasil, ou então, em casos específicos estão apenas exibindo o mês e ano, ao invés da data completa.

Além disso, é necessário configurar as colunas de dados que são dados monetários (moeda) para exibirem os valores com duas casas decimais e com o símbolo monetário correto (R\$). Essas alterações são simples de serem feitas, mas garantem uma melhor experiência do usuário.

A dificuldade desta etapa mora em modelar o relacionamento dos dados. A modelagem dos dados no banco foi feita considerando o modelo relacional dos dados, no entanto, dentro do Power BI, a modelagem dos dados é feita utilizando o modelo dimensional.

Ou seja, são utilizadas tabelas fato e tabelas dimensão para modelar os dados. Tabela fato é a tabela com uma grande quantidade de linhas, composta pelas métricas, que são tudo aquilo que se quer medir, junto com as chaves que ligam às tabelas dimensões, que descrevem essas métricas. A tabela dimensão auxilia a tabela fato com dados complementares ou explicativos, e possui informações que não se repetem.

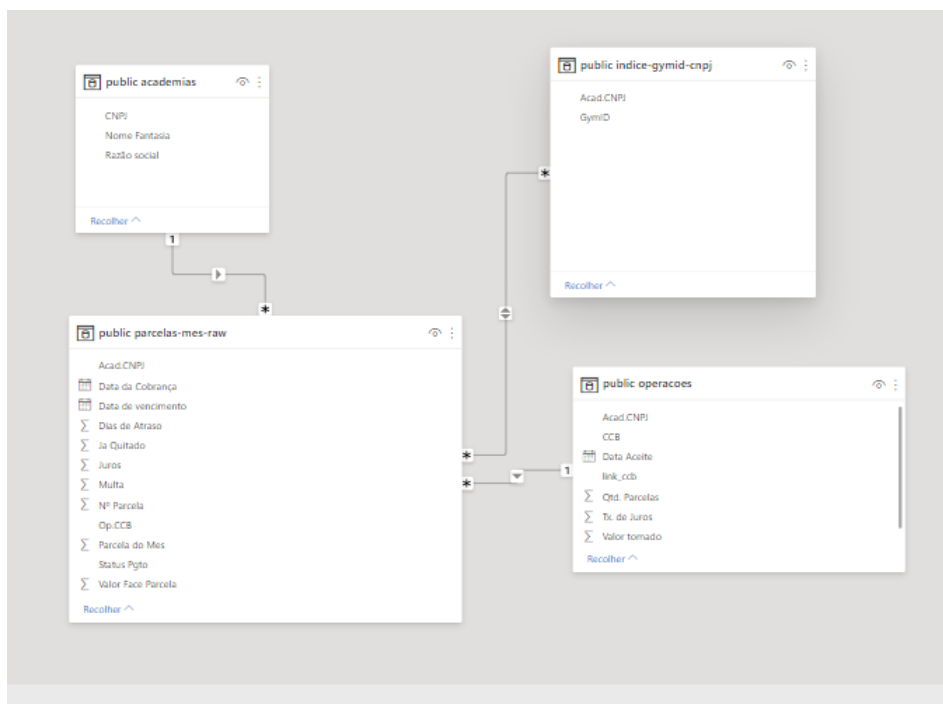
Como o banco de dados já estava carregado, e as tabelas estavam modeladas de forma relacional, optou-se por modelar dentro do BI as tabelas de forma dimensional, e não alterar o banco.

Para modelos maiores de dados a complexidade pode começar a gerar problemas. Conforme a aplicação for escalando, pode-se revisitar esse problema e alterar o modelo no banco, no entanto, como prova de conceito, a solução atual funciona.

Para adequar o modelo, ao modelo dimensional, as tabelas foram modeladas conforme exibido nas figuras a seguir. Na Figura 14 está modelada a relação da tabela "parcelas-mes-raw "(fato) com as tabelas dimensão "operacoes ", "academias" e "indice-id-cnpj ".

A tabela "operacoes "é ligada na tabela fato através do código da operação (CCB), a tabela "academias"é ligada pelo CNPJ, e a tabela "indice-id-cnpj "também é ligada pelo CNPJ.

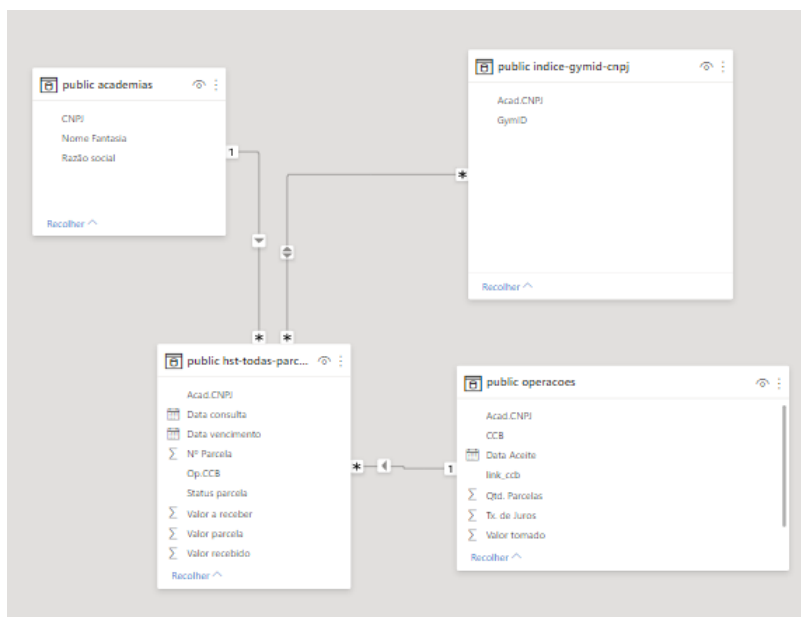
Figura 14 – Modelo dimensional das parcelas



Fonte – Autoria própria

A modelagem das relações com a tabela de "hst-todas-parcelas" é semelhante a relação da tabela "parcelas-mes-raw". Essa modelagem é exposta na Figura 15.

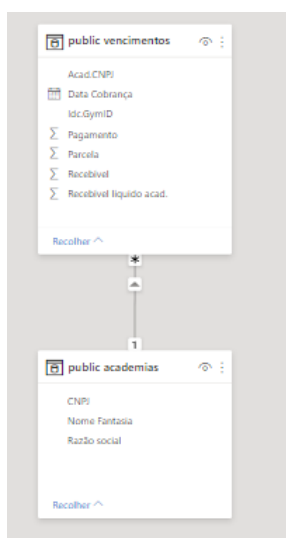
Figura 15 – Modelo dimensional do histórico das parcelas



Fonte – Autoria própria

Por fim, é modelado o relacionamento da tabela "vencimentos"(fato). Essa tabela relaciona apenas com a tabela "academias"através do CNPJ, como é exposto na figura Figura 16.

Figura 16 – Modelo dimensional dos vencimentos



Fonte – Autoria própria

Ao modelar o relacionamento da tabela de vencimentos encontrou-se um futuro melhoramento que pode ser aplicado na importação dos dados e na modelagem das

tabelas. Novamente, como prova de conceito o modelo atende aos seus requisitos, no entanto, o modelo pode ser melhorado.

Tendo modelado o relacionamento entre as tabelas, é feita a validação dos dados. Para validar o dado, foram recriadas réplicas das planilhas que alimentam o banco de dados, utilizando o Power BI. Em cima dessas tabelas, dois métodos foram utilizados para validar os dados:

- Comparação dos totais mensais;
- Cruzamento dos dados via Excel;

Para comparar os totais, foi utilizada a função de somar totais do Power BI. Somando valor total por mês das parcelas e os valores encontrados na tabela de vencimentos (parcela, recebível e recebível líquido e pagamento), e comparando esses valores com os totais que estão nas planilhas do excel que são a base para o projeto, é possível realizar a primeira validação do dado.

Se os totais encontrados mês a mês são semelhantes aos valores encontrados nas planilhas, pode-se tomar como primeiro indicativo de que o dado está correto.

A segunda validação é feita utilizando as réplica das planilhas criadas foi exportada para um modelo em Excel. A partir desse modelo do Excel, é utilizada uma função de busca ("PROC") para verificar, através das chaves das tabelas, se todos os dados que estão nas planilhas estão nas tabelas.

Caso os valores não estejam faltando, e os totais estejam corretos, pode-se afirmar que os dados estão corretos. Com os dados corretos é possível iniciar a construção dos dashboards.

5.8 DESENVOLVIMENTO DOS DASHBOARDS

Para iniciar a construção dos *dashboards*, inicialmente deve-se ter a relação entre os dados e o que eles representam em linguagem natural, e quais as colunas que contem esses dados e em que tabelas. Essa relação será feita considerando os dados das fontes originais dos dados, e das tabelas existentes no BI.

Cada *dashboard* desenvolvido irá atender ao que foi previsto na seção Seção 4.4. O layout de fundo dos dashboards é o mesmo, e foi desenvolvido através do Power Point. A disponibilização dos dashboards será feita através de uma workspace online do Power BI. Esse é um serviço de nuvem da PBI onde é possível publicar o relatório online. O próprio serviço do PBI faz a gestão de acesso, e basta adicionar o email dos usuários que desejam acessar o painel online.

5.8.1 Painel de operações realizadas

O painel de operações deve conter o histórico de todas as operações de crédito realizadas. Essas operações devem ser expostas sendo relacionadas com o CNPJ que originou essa operação. No entanto, deve ser possível filtrar a tabela usando o ID na empresa parceira.

Esse dashboard será composto por duas tabelas. Uma tabela com os dados de cada operação de crédito, e uma tabela auxiliar, onde estará exposta a relação entre o ID no cliente e o CNPJ. Essa tabela se faz necessária pois para cada CNPJ há mais de um ID, e isso iria dificultar muito a visualização de dados na primeira tabela, implicando em uma usabilidade ruim.

A composição de cada tabela se encontra na Tabela 4

Tabela 4 – Conteúdo Painel de operações realizadas

Objeto	Conteúdo	Fonte do dado
Tabela operações	CNPJ	public.parcelas-mes-raw -> Acad.CNPJ
	Razão social	public.academias -> Razão social
	Código da operação (CCB)	public.operacoes -> CCB
	Data da operação	public.operacoes -> Data de aceite
	Quantidade de parcelas	public.operacoes -> Qtd. de parcelas
	Taxa de juros	public.operacoes -> Tx. de juros
	Valor tomado	public.operacoes -> Valor tomado
	1º Vencimento	public.operacoes -> Vcto. 1ª parcela
Tabela auxiliar	ID	public.indice-id-cnpj -> id
	CNPJ	public.academias -> CNPJ
	Nome fantasia	public.academias -> Nome fantasia
	Razão social	public.academias -> Razão social
	Código da operação (CCB)	public.operacoes -> CCB
	Data da operação	public.operacoes -> Data de aceite

Fonte – Autoria própria

Além das tabelas, o painel deve conter filtros pelos seguintes critérios:

- Data de contratação;
- CNPJ;
- ID;
- Razão social;
- Código da CCB.

Utilizando a visualização "tabela" do Power BI, são construídas ambas as tabelas com o conteúdo mencionado acima. Todos os dados devem ser exibidos no formato "não resumir". Para construir os filtros foi utilizada a visualização "Segmentação de

dados". Vale ressaltar ainda que as tabelas são relacionadas, portanto, ao clicar em uma linha da tabela auxiliar por exemplo, a entrada que possui o mesmo CNPJ na tabela de operações será filtrada.

O painel desenvolvido pode ser visto na Figura 17. Nele, vemos a esquerda do painel os filtros que podem ser aplicados, e na parte central, há as duas tabelas que foram mencionadas nos parágrafos acima. No apêndice desse documento também há uma versão maior da figura para melhor visualização.

Figura 17 – Painel de operações



Fonte – Autoria própria

5.8.2 Painel do histórico de vencimentos

O painel do histórico de vencimentos contém o compilado de todas as planilhas de cobrança desde o início da operação de crédito como serviço. Esse painel deve conter a réplica exata da planilha. Ou seja, os dados são segmentados pelo ID, ao lado do ID é exposto qual CNPJ relacionado a esse ID.

Após o ID e CNPJ, é exposta a razão social do cliente, quanto o cliente faturou, quanto foi a parcela do mês, qual foi o pagamento efetuado daquela parcela e quanto a academia recebeu após ter a parcela descontada. Por fim, é acrescentada uma coluna

extra indicando qual foi a data da cobrança. Aqui, todos os dados também devem ser exibidos no formato "não resumir".

Além dessa tabela, é colocada uma tabela auxiliar semelhante a tabela que foi mencionada na seção anterior. O conteúdo utilizado para construir o painel está exposto na Tabela 5

Tabela 5 – Conteúdo Painel de vencimentos

Objeto	Conteúdo	Fonte do dado
Tabela vencimen- tos	ID	public.vencimentos -> Idc.id
	CNPJ	public.academias -> CNPJ
	Razão social	public.academias -> Razão social
	Faturamento bruto	public.vencimentos -> Recebível
	Parcela do mês	public.vencimentos -> Parcela
	Pagamento efetuado	public.vencimentos -> Pagamento
	Faturamento líquido	public.vencimentos -> Recebível líquido
Data cobrança	public.vencimentos -> Data cobrança	
Tabela auxiliar	ID	public.vencimentos -> Idc.id
	CNPJ	public.academias -> CNPJ
	Nome fantasia	public.academias -> Nome fantasia
	Razão social	public.academias -> Razão social
	Data cobrança	public.vencimentos -> Data cobrança

Fonte – Autoria própria

Além das tabelas, o painel deve conter filtros pelos seguintes critérios:

- Data de cobrança;
- CNPJ;
- ID;
- Razão social;

As tabelas e os filtros foram construídos de maneira semelhante ao que é mencionado na Seção 5.8.1. O resultado do painel pode ser visto na Figura 18 ou então no apêndice desse documento.

Figura 18 – Painel de vencimentos



Fonte – Autoria própria

5.8.3 Painel do Composição da parcela mensal

Neste painel devem estar contidas todas as parcelas que um cliente deve pagar em cada mês. A composição da parcela que o cliente paga no mês vem do somatório do valor de todas as parcelas que um cliente tem em aberto aqui. Um cliente pode possuir mais de uma valor compondo a sua parcela quando ele possui mais de uma operação de crédito, ou quando possui parcelas em atraso. Aqui, novamente, todos os dados também devem ser exibidos no formato "não resumir".

Segmentado por CNPJ (e razão social ao lado), devem constar o código da operação que está sendo cobrada, o número da parcela em aberto, a quantidade total de parcelas daquele empréstimo, a data original de vencimento da parcela, o valor devido e a data em que se está sendo efetuado aquela cobrança. Neste painel também deve ser inserida uma tabela auxiliar semelhante a exposta em Seção 5.8.1.

Ao expor os dados em uma tabela, de forma semelhante ao que vem sendo feito, temos o resultado exposto em Tabela 6

Além das tabelas, o painel deve conter filtros pelos seguintes critérios:

Tabela 6 – Conteúdo Painel de Composição da parcela mensal

Objeto	Conteúdo	Fonte do dado
Tabela parcelas	CNPJ	academias -> CNPJ
	Razão social	public.academias -> Razão social
	Código da operação (CCB)	public.operacoes -> CCB
	Nº parcela	public.parcelas-mes-raw -> Nº parcela
	Quantidade de parcelas	public.operacoes -> Qtd. de parcelas
	Data de vencimento	public.parcelas-mes-raw -> Data de vencimento
	Valor devido	public.parcelas-mes-raw -> parcela do mes
	Data cobrança	public.parcelas-mes-raw -> Data cobrança
Tabela auxiliar	ID	public.indice-id-cnpj -> id
	CNPJ	public.hst-todas-parcelas -> Acad.CNPJ
	Nome fantasia	public.academias -> Nome fantasia
	Razão social	public.academias -> Razão social
	Código da operação (CCB)	public.operacoes -> CCB
	Data da operação	public.operacoes -> Data de aceite

Fonte – Autoria própria

- Data de contratação;
- CNPJ;
- ID;
- Razão social;
- Código da CCB.

Por fim, a construção desta tabelas é feita de forma semelhante ao que vem sido mencionado nas subseções anteriores, e o resultado está na figura Figura 19 como também em tamanho maior no apêndice do documento.

Figura 19 – Painel de composição das parcelas mensais

The image shows a dashboard titled "PAINEL DA COMPOSIÇÃO DAS PARCELAS MENSAS". On the left side, there are five filter panels, each with a dropdown menu set to "Todos":

- Data da Cobrança
- CNPJ
- Gym ID
- Razão Social
- Código da CCB

The main area contains a table titled "Composição das parcelas mensais". The table has the following columns: CNPJ, Razão social, Código da CCB, Nº Parcela, Qtd. Parcelas, Data Vcto., and Composição do Valor da parcela. The table displays several rows of data, with some rows highlighted in blue. The bottom of the table shows a summary section with columns for "Total", "Valor", "Data", and "Código da CCB".

Fonte – Autoria própria

5.8.4 Painel do histórico de todas as parcelas

O último painel a ser desenvolvido para a empresa parceira é o painel que contém o histórico de todas as parcelas de um cliente. Esse painel irá exibir em uma tabela, com uma granularidade mensal, todas as parcelas que o cliente já pagou, e todas que estão em aberto para ele em um dado mês.

Segmentado por CNPJ, devem ser exibidos os códigos das operações, a quantidade de parcelas que essas operações possuem, a data original de vencimento de cada parcela, o número daquela parcela, o valor que já foi recebido para pagá-la, o valor original da parcela e o valor que deve ser recebido para quitar aquela parcela.

O valor que deve ser recebido pode ser diferente do valor original pois o cliente pode ter realizado um pagamento parcial da parcela, ou ainda ter atrasado o seu pagamento, o que implicou no acréscimo de juros e multa. Na tabela Tabela 7 podemos ver a composição do dashboard. Esse painel também deve conter uma planilha auxiliar.

Os filtros e a construção dessa tabela semelhantes aos mencionados na Seção 5.8.1. O resultado pode ser visto na Figura 20 como também no apêndice do

Tabela 7 – Conteúdo Painel de Histórico de todas as parcelas

Objeto	Conteúdo	Fonte do dado
Tabela histórico de parcelas	CNPJ	hst-todas-parcelas -> Acad.CNPJ
	Razão social	public.academias -> Razão social
	Código da operação (CCB)	public.operacoes -> CCB
	Quantidade de parcelas	public.operacoes -> Qtd. de parcelas
	Data de vencimento	public.hst-todas-parcelas -> Data de vencimento
	Nº parcela	public.hst-todas-parcelas -> Nº parcela
	Valor recebido	public.hst-todas-parcelas -> valor recebido
	Valor parcela	public.hst-todas-parcelas -> Valor parcela
Tabela auxiliar	Valor a receber	public.hst-todas-parcelas -> Valor a receber
	CNPJ	public.indice-id-cnpj -> id
	CNPJ	public.hst-todas-parcelas -> Acad.CNPJ
	Nome fantasia	public.academias -> Nome fantasia
	Razão social	public.academias -> Razão social
	Código da operação (CCB)	public.operacoes -> CCB
Data da operação	public.operacoes -> Data de aceite	

Fonte – Autoria própria

documento.

Figura 20 – Painel de histórico das parcelas



Fonte – Autoria própria

Esse painel conclui os painéis requisitados pelo cliente. Os próximos painéis expostos são painéis para atender demandas internas da Flow Finace.

5.8.5 Pagamento mensal dos clientes

Este é o primeiro painel que foi desenvolvido para atender uma demanda interna da Flow. O painel de pagamento mensal do cliente tem como objetivo suprir uma demanda interna na construção da planilha de vencimentos. Com a criação dele, o tempo e gasto para construir a planilha de vencimentos será significativamente reduzido.

Esse painel é praticamente uma réplica do painel exposto na Seção 5.8.3 e para construí-lo, foram reaproveitados os conteúdos do dashboard. A diferença é que, a tabela de parcelas que é referenciada na Tabela 6 teve as colunas "Código da operação", "Nº parcela", "Quantidade de parcelas" e "Data de cobrança".

Além da remoção dessas colunas, foi inserido antes da coluna do "CNPJ" uma coluna contendo o ID do parceiro (public.indice-id-cnpj -> id). Por fim, a principal diferença é que antes eram exibidos os valores de todas as parcelas abertas por academia,

pois o dado estava no formato "não resumir". Agora, nessa tabela, o dado de valor devido (public.parcelas-mes-raw -> parcela do mes) deve ser exibido no formato "soma".

O grande ponto positivo deste painel é que para construí-lo foi gasto muito pouco tempo, visto que grande parte de seus dados foram reaproveitados de outro painel. O resultado pode ser visto no apêndice do documento em tamanho aumentado, como também na Figura 21.

Figura 21 – Painel de cobrança mensal



Fonte – Autoria própria

Através desse painel, agora será possível abandonar o processo de construção manual do tabela utilizada para realizar a cobrança, e será possível utilizá-lo para realizar a construção da tabela de forma automática. O próprio PBI possui uma função para exportar a tabela ali construída para um arquivo ".csv"que será enviado para a empresa parceira.

5.8.6 Painel de auxílio nas expansões de crédito

O painel de auxílio nas expansões de crédito necessita de duas tabelas. Uma tabela com o histórico de vencimento dos últimos 4 meses para um CNPJ, ou seja,

quanto foram as parcelas e quanto ele pagou, e uma tabela com todas as operações já originadas.

Ambas essas tabelas já foram criadas para os dashboards da empresa parceira, e para criar esse novo painel, bastou combinar em um único painel a tabela de operações, descrita na Tabela 4 e a tabela de vencimentos, descrita em Tabela 5. Foram necessárias apenas uma modificação. Foi necessário filtrar a tabela de vencimentos para que ela só mostre os vencimentos dos últimos quatro meses.

O resultado desse painel pode ser visto na Figura 22 ou então em anexo a este documento. Este é mais um exemplo de uma situação onde a reutilização de objetos de outros painéis acelerou o desenvolvimento.

Figura 22 – Painel de auxílio as expansões de crédito



Fonte – Autoria própria

5.8.7 Gráficos de controle interno da operação

O último painel desenvolvido para a Flow Finance foi o painel com os indicadores da operação. Esse painel replica gráficos que antes eram confeccionados via excel. Esse painel deve exibir diversas informações, são elas:

- Evolução dos empréstimos mês a mês (quantidade e volume);

- Acumulado dos empréstimos no tempo (quantidade e volume);
- Total de clientes x clientes únicos;
- Total de operações;

Na tabela Tabela 8 está evidenciado como será construído cada um dos objetos do painel.

Tabela 8 – Conteúdo Painel de Gráficos de controle interno da operação

Objeto	Conteúdo	Fonte do dado
Valor tomado por mês	Data da operação	public.operacoes -> Data de aceite
	Valor tomado	Soma de public.operacoes -> Valor tomado
Quantidade de CCBs por mês	Data da operação	public.operacoes -> Data de aceite
	Quantidade	Contagem de public.operacoes -> CCB
Acumulado dos empréstimos por mês	Data da operação	public.operacoes -> Data de aceite
	Acumulado do valor tomado	Medida usando public.operacoes -> Valor tomado
Quantidade acumulada de CCBs por mês	Data da operação	public.operacoes -> Data de aceite
	Quantidade acumulada de empréstimos	Medida usando public.operacoes -> CCB
Total de clientes	Quantidade total de clientes	Contagem de public.academias -> CNPJ
Total de clientes únicos	Quantidade total de clientes	Contagem distinta de public.academias -> CNPJ
Total de operações	Quantidade total de operações	Contagem de public.operacoes -> CCB

Fonte – Autoria própria

Para expor os dados que variam no tempo foram utilizados gráficos no Power Bi, utilizando as datas no eixo e os valores sendo compostos pelos respectivos valores que desejava-se exibir. Os valores únicos foram exibidos em cartões.

Vale ressaltar que para obter os valores de total acumulado de empréstimos e quantidade acumulada de empréstimos foi necessário realizar a construção de medidas. Essas medidas foram construídas utilizando DAX. Um exemplo de código em DAX de como foi construída a medida do valor total acumulado dos empréstimos pode ser vista na Figura 23

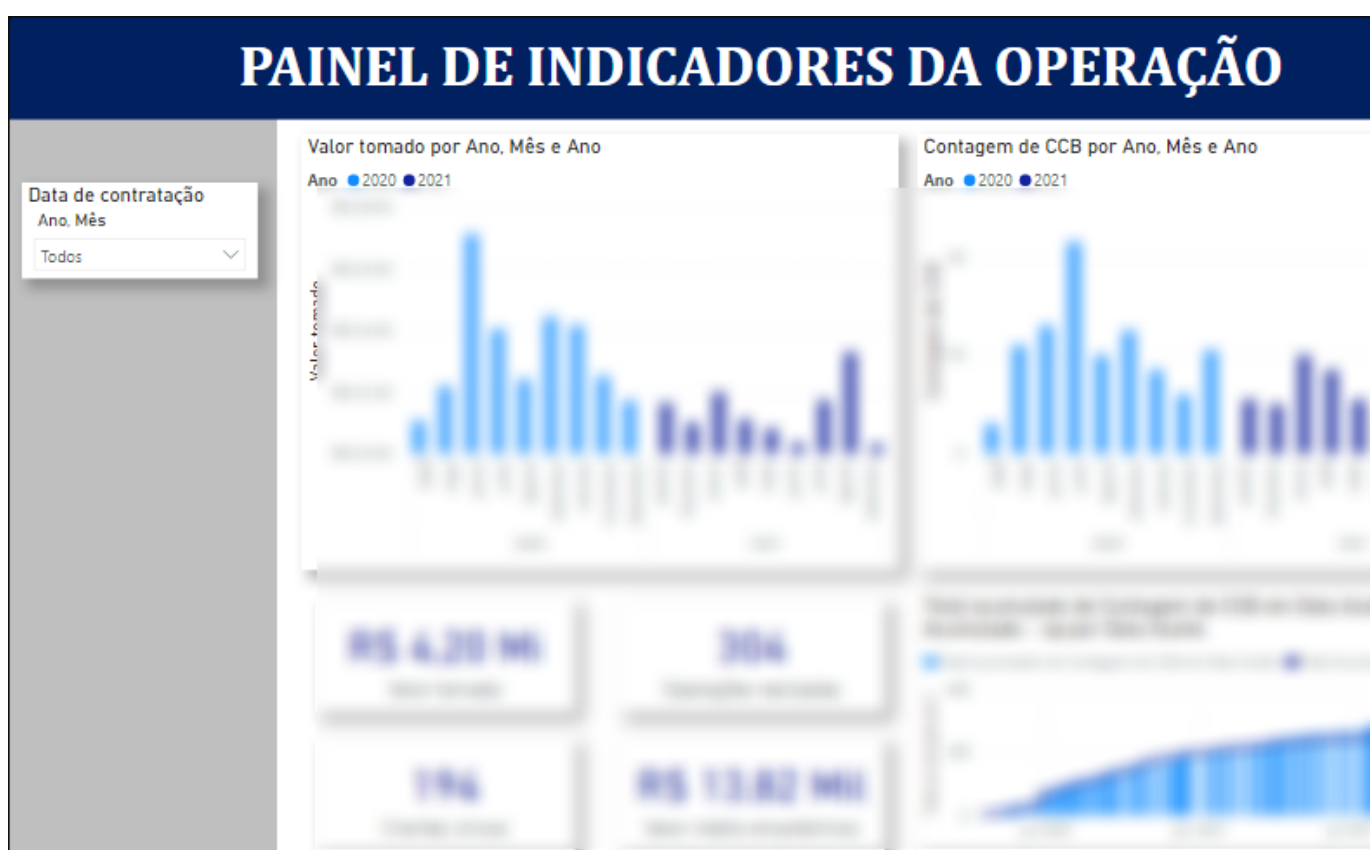
Figura 23 – Código em DAX - construção de uma medida de soma

```
1 Total Acumulado - op =  
2 CALCULATE (  
3     SUM ( 'public operacoes'[Valor tomado]),  
4     FILTER (  
5         ALL ( 'public operacoes' ),  
6         'public operacoes'[Data Aceite] <= MAX ( 'public operacoes'[Data Aceite] )  
7     )  
8 )
```

Fonte – Autoria própria

Esse dashboard conclui a construção dos painéis que a Flow Finance necessitava. Na Tabela 8 está o resultado da construção desse painel. O painel pode também em ser visto em tamanho ampliado no apêndice desse documento.

Figura 24 – Painel de indicadores da operação



Fonte – Autoria própria

Esse painel conclui a construção para atender as demandas internas da Flow Finance, restando agora a construção do sistema de apoio a tomada de decisão.

5.9 ESTUDO DAS LINHAS DE CRÉDITO

A última tarefa restante no projeto é realizar a construção do sistema que auxilia a encontrar as linhas de crédito ideal para cada cliente. Para construir o modelo matemático por traz dele, é necessário entender melhor o perfil das operações de crédito que estão sendo realizadas.

As operações de crédito que são realizadas podem ter de 'x' até 'y' parcelas (dado confidencial) para que seja realizado o pagamento. Para analisar como tem sido a tomada de crédito, foi compilada uma tabela com todas as operações realizadas, e em qual prazo foi realizada a tomada de crédito.

Analisando o prazo de tomada de crédito, observou-se que 95% das operações de crédito realizadas eram em 'y/2 ou mais parcelas. Além disso, observou-se que a moda das quantidade de parcelas das operações realizadas era de y parcelas.

O segundo fator que foi analisado foi o efeito da pandemia no faturamento das empresas clientes. Ao estudar o faturamento dos últimos 12 meses, observa-se que, em média n meses desse faturamento são nulos, devido a restrições de funcionamento que foram impostas as empresas.

Hoje, o cálculo do valor máximo que pode ser tomado em crédito por um cliente é feito tomando $n\%$ do total do faturamento dos últimos 12 meses. O problema que observa-se aqui, é que, devido ao período fechado, o total de faturamento acaba sendo distorcido.

Como a moda da quantidade de parcelas dos empréstimos é de y meses, e nos últimos 12 meses de faturamento há meses nulos que impactam no total, foi definido que a nova regra de crédito irá utilizar o faturamento dos últimos y/6 meses para projetar quanto será o faturamento total em y meses. Será em cima desse faturamento total que passarão agora a serem calculados os $n\%$.

Ou seja, o novo limite de crédito é de $n\%$ do faturamento projetado de y meses, e essa projeção é feita baseada no faturamento dos últimos y/6 meses. Em cima dessa regra que deve ser construído o sistema de auxílio que será acoplado ao BI.

O sistema deve realizar o cálculo para todos os clientes (CNPJs) e expor qual o valor ideal de linha de crédito para cada cliente baseado na nova regra e quanto que esse cliente tem de crédito tomado com a empresa.

5.10 SISTEMA DE AUXÍLIO PARA LIMITE DE CRÉDITO

O sistema de apoio a decisão deve auxiliar a estabelecer o limite máximo de crédito para os clientes. Ele será acoplado ao BI, permitindo que qualquer colaborador da empresa tenha acesso ao sistema e consiga utilizá-lo de forma simples.

No desenvolvimento foi utilizada a linguagem de programação DAX. A construção do sistema não possui grande complexidade. Inicialmente foi construída uma

função que calcula o total do faturamento dos últimos $y/6$ meses dos clientes. O faturamento dos clientes está armazenado na tabela 'public.vencimentos', na variável "Recebível". A construção dessa função pode ser vista na Figura 25.

Figura 25 – Código em DAX - faturamento acumulado 4 meses

```
1 Total Acumulado - op =
2 CALCULATE (
3     SUM ( 'public operacoes'[Valor tomado]),
4     FILTER (
5         ALL ( 'public operacoes' ),
6         'public operacoes'[Data Aceite] <= MAX ( 'public operacoes'[Data Aceite] )
7     )
8 )
```

Fonte – Autoria própria

Tendo o faturamento acumulado de $y/6$ meses, é obtida a média desse faturamento, posteriormente a média é multiplicada pelo novo total de meses que será utilizado como base de faturamento. Ou seja, o valor é multiplicado por y . Agora há uma projeção do faturamento para y meses.

Com o faturamento projetado, basta multiplicar esse valor pelo percentual desejado para a linha de crédito, que são $n\%$.

Novamente, para realizar esse cálculo foi criada uma função em DAX. Tendo criado a função em DAX, para visualizar o resultado, foi desenvolvido um novo painel. O principal componente do painel é a tabela que entrega o valor da linha de crédito, que pode ser visto na Figura 26.

O sistema, embora tenha um simples desenvolvimento, calcula de maneira praticamente instantânea o limite de crédito ideal para cada cliente. Um cálculo que antes teria de ser feito manualmente para cada cliente. Isso traz agilidade para a operação e também robustez. Uma vez que se comprova que o as contas são realizadas corretamente, sabe-se que não ocorrerão mais erros ao calcular a linha de crédito, e pode-se escalar o sistema.

Para validar o cálculo da linha de crédito basta aplicar o faturamento dos últimos $y/6$ meses de qualquer cliente na equação, realizar os cálculos manualmente e comparar com o *output* que foi calculado automaticamente. Caso o valor esteja correto, o sistema está validado.

Portanto, o modelo matemático que foi desenvolvido e aplicado ao BI, auxilia na resolução de um problema semi-estruturado, onde o algoritmo realiza cálculos matemáticos baseados em um série de dados, expões junto ao *dashboard* esse resultado para auxiliar na tomada de uma decisão.

Figura 26 – Tabela de *output* do sistema de auxílio de limite de crédito

Linha de crédito ideal		
CNPJ	Razão social	Linha de Crédito ideal
		R\$ 331.654,76
		R\$ 304.441,34
		R\$ 290.917,44
		R\$ 285.225,11
		R\$ 251.743,11
		R\$ 223.891,20
		R\$ 190.704,96
		R\$ 174.219,84
		R\$ 170.795,51
		R\$ 164.994,24
		R\$ 163.361,29
		R\$ 162.541,44
		R\$ 156.520,71
		R\$ 153.296,35
		R\$ 149.199,57

Fonte – Autoria própria

Além do resultado exposto pelo modelo, são analisados dados de diversas outras fontes, como birôs de crédito, receita federal e banco central. Esses dados apoiam a instituição financeira, que para tomar suas decisões utiliza do julgamento de seus analistas de crédito e também utiliza o resultado do modelo para determinar o limite do cliente.

Ou seja, o modelo que foi criado, por si só não pode ser utilizado para tomar uma decisão, mas auxilia nesse processo.

Novamente, vale ressaltar que o que foi desenvolvido é um modelo matemático simples para auxiliar no processo de definição da linha de crédito. Ele é um sistema auxiliar e não um SAD, que é complexo, com sua própria arquitetura e modelos de tomada de decisão.

Tendo concluído o que foi exposto acima, a etapa de desenvolvimento do projeto está concluída. Podendo agora seguir para a etapa de avaliação dos resultados.

6 ANÁLISE DOS RESULTADOS

Para analisar os resultados do projeto, vale resgatar o problema inicial e o que foi proposto como solução. Conforme mencionado no início do documento, existia a necessidade de estruturar os dados de uma operação de crédito como serviço para uma empresa parceira da Flow Finance.

Esses dados estavam espalhados em diversas planilhas, e utilizá-los para análises era um processo que demandava muito tempo. Além disso, a empresa parceira precisava ter acesso aos dados da operação, principalmente por questões de *compliance*.

Por fim, uma das principais análises que é feita com os dados da operação é a análise do limite de crédito dos clientes, e essa análise precisava ser feita de forma automatizada agora, com o uso de um sistema. Para construir esse sistema, era necessário estudar o perfil do crédito sendo tomado, e desenhar uma nova regra de crédito caso necessário.

Para atender a demanda, os dados foram reunidos em um banco de dados PostgreSQL, hospedado utilizando o serviço de nuvem da AWS. Esse dado foi conectado a um PBI, onde foram construídos painéis para atender as demandas da Flow Finance e da empresa parceira. Dentro desse BI foi acoplado o que realiza o cálculo das linhas de crédito.

6.1 VALIDAÇÃO DOS RESULTADOS PARA EMPRESA PARCEIRA

Para realizar a validação dos resultados para a empresa parceira foram apresentados os painéis desenvolvidos em BI para eles, explicando como que cada painel funciona.

A empresa parceira está satisfeita com o resultado, e já pode começar a utilizar os seus painéis para realizar o monitoramento da operação.

Por parte deles, foram solicitadas apenas duas melhorias no projeto. Uma documentação de como utilizar os painéis, e também solicitaram que fosse acrescentada uma nova coluna na tabela de operações (mencionada em Tabela 4). Essa nova coluna deve apresentar um link para o arquivo do contrato assinado pelo cliente que tomou o crédito.

Essa nova coluna já foi acrescentada no banco de dados, assim como no painel no BI. A coluna está sendo alimentada aos poucos com os arquivos no formato "pdf", que contém os contratos. Como esses arquivos não estão localizados em um único local, e não estão indexados, esse processo irá demandar um pouco mais de tempo para ser realizado.

No entanto, como isso não constava no escopo inicial, esta não é uma demanda prioritária. Portanto, conclui-se que o resultado final para a empresa parceira foi satis-

fatório e atendeu as suas necessidades.

6.2 VALIDAÇÃO DOS RESULTADOS PARA FLOW FINANCE

Para a Flow Finance foram desenvolvidos 4 painéis diferentes. Dois com o intuito de trazer agilidade para operação, um para abrigar o sistema de apoio a decisão, e outro para replicar os gráficos de controle de indicadores.

Os 3 primeiros *dashboards* tem como foco principal reduzir o tempo gasto para realizar estudos e acompanhamento da operação e também trazer agilidade para operação. A maneira encontrada de se medir o impacto que a construção desses painéis trouxe para operação é medindo o tempo gasto para realizar tarefas com e sem a existência desse painéis. Já o *dashboard* que replica os gráficos e indicadores da operação, esse pode ser validado comparando com a solução já existente.

6.2.1 Painel de apoio as expansões de crédito

Quando um cliente solicita uma expansão de sua linha de crédito, é necessário reunir os dados de faturamento dos seus últimos 4 meses, como também os dados de todas as suas operações de crédito realizadas e se elas já foram encerradas.

Para reunir esses dados, é necessário acessar, no mínimo, 6 planilhas diferentes. 1 planilha abriga as operações realizadas, 4 delas são referentes ao faturamento dos 4 meses que passaram, e uma última é de quais parcelas estão em aberto, permitindo assim saber quais operações ainda tem parcelas a vencer

Para reunir esses dados e dar o devido tratamento a ele se gasta em média 12 minutos por cliente. Isso se deve principalmente ao fato do dado estar espalhado em diversas fontes.

Com a utilização do painel que foi desenvolvido, todas essas informações hoje se encontram no mesmo lugar, em uma única página. Basta filtrar pelo CNPJ do cliente e o usuário tem acesso a tudo que necessita.

Para consultar e agrupar toda a informação em texto, agora leva-se menos de 3 minutos. Considerando que em média são realizadas 15 solicitações de expansão de linha de crédito por mês, são 3 horas gastas todo mês para que essas operações sejam realizadas. Agora, é possível realizar o mesmo processo em 45 minutos.

6.2.2 Painel de cobrança mensal

O painel de cobrança mensal auxilia a montar a planilha de vencimentos (cobrança) dos clientes. O processo de montagem da planilha antes do auxílio do painel era realizado utilizando uma planilha de excel, e continha 7 etapas até chegar no resultado final. O resultado final da planilha era o compilado de quanto cada cliente devia pagar de parcela mensalmente, acrescido de algumas informações.

Para construir essa planilha levava-se um tempo muito alto. Eram gastas ao menos duas horas por mês para construí-la. Além disso, por ser construída manualmente, ocasionalmente ocorriam erros no processo, que poderiam se tornar erros na construção das parcelas e, conseqüentemente, erros no processo de cobrança.

Com a utilização do painel, gasta-se apenas o tempo de carregar as informações para o banco de dados e atualizar o Power BI. Para atualizar todas as tabelas no banco de dados são gastos menos de 15 minutos. Além disso, como a tabela de vencimentos (cobrança) agora é construída de forma automática, a chance de ocorrerem erros no processo passa a ser quase nula.

Para que ocorra um erro é necessário que o dado que alimentou o banco esteja errado, e isso não é oriundo do processo de construção da tabela em si. Embora o ganho em agilidade já seja significativo, a robustez que utilizar o *dashboard* traz ao processo é a sua principal vantagem.

6.2.3 Sistema de auxílio a determinação das linhas de crédito

Conforme já mencionado anteriormente na seção de desenvolvimento, acoplado ao *dashboard* há um sistema que auxilia a encontrar a linha de crédito ideal (máxima) para um cliente.

Quando se está analisando o limite de crédito de um cliente, são consultadas informações internas dos clientes, informações de outras fontes de dado que somente a instituição financeira concedente do crédito tem acesso e também o *output* do sistema de apoio.

Para realizar manualmente a análise do limite de crédito e reunir os dados necessários, são gastos em média 20 minutos por cliente, visto que se gasta tempo reunindo os dados, assim como realizando os devidos cálculos e expondo essa informação.

Hoje, como o sistema já foi desenvolvido, essa análise pode ser feita de forma instantânea. Além disso, caso uma nova regra de crédito queira ser aplicada, basta alterar a formula desenvolvida, um esforço que pode ser realizado em poucos minutos.

Embora de simples formulação, o sistema traz robustez e agilidade por não gastar tempo para realizar os cálculos, como também fornece flexibilidade por ser facilmente alterável.

6.3 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

Uma parte importante da avaliação dos resultados obtidos pelo sistema está em resgatar os requisitos funcionais e não funcionais do projeto, e verificar se esses requisitos foram atendidos.

Relembrando o leitor, os requisitos funcionais do projeto são os problemas que o sistema deve resolver.

Não será entrado no mérito de qual painel resolve cada problema, apenas afirma-se que todos os requisitos funcionais de fato foram atendidos com o desenvolvimento do projeto. Pode-se até afirmar que os painéis desenvolvidos atendem mais do que foi solicitado nos requisitos, o que aponta uma falha no planejamento.

Quanto aos requisitos não-funcionais do sistema, 5 foram listados:

1. O sistema deverá ficar disponível para uso online;
2. O sistema deverá utilizar um banco de dados *open source*;
3. O sistema não deve permitir alterações nos dados do banco por parte dos usuários;
4. O sistema deverá comunicar com um banco hospedado na nuvem;
5. O sistema só deverá permitir o acesso de usuários autorizados;

Aqui vale explicar que os requisitos 1, 3 e 5 são garantidos pelo próprio Power BI. Mais especificamente, o serviço do Power BI online, que permite publicar relatórios em uma *workspace* online, e ali já fornece o controle de acesso aos seus usuários.

Além disso, pelos dados estarem sendo disponibilizados em um BI, da natureza da aplicação já se impossibilita alterar os dados no banco. Por fim, os outros dois requisitos já são bem detalhados no relatório, sendo suficiente apenas afirmar que estes requisitos também são atendidos.

6.4 CONSIDERAÇÕES FINAIS SOBRE OS RESULTADOS

Considerando os requisitos do sistema (funcionais e não-funcionais) e o sistema desenvolvido, pode-se considerar que os resultados do projeto foram alcançados. Todos os requisitos foram cumpridos, e os painéis trouxeram mais agilidade para a operação.

Com o uso deles, foi possível também transparecer todos os dados da operação para a empresa parceira, garantindo que os requisitos de ter acesso aos dados para o *compliance* deles foi atendido. Por fim, tendo concluído a análise dos resultados, é possível passar para seção final deste documento, que é a conclusão.

Vale ressaltar que todos os dados utilizados no projeto não são dados fictícios ou normalizados, mas sim os dados reais da empresa. A aplicação que foi desenvolvida neste trabalho é utilizada tanto pela empresa, quanto pelo parceiro, para realizar o acompanhamento da operação de CaaS.

Nenhuma alteração foi feita no projeto para que ele fosse utilizado pelas partes. O resultado do projeto é a aplicação real utilizada pela Flow Finance e pela empresa parceira, o que ressalta a importância deste projeto dentro da *fintech*, assim como expõe o atingimento de seus objetivos.

7 CONCLUSÃO

Este projeto foi desenvolvido para atender uma demanda da Flow Finance, uma fintech de crédito como serviço. Na operação de crédito como serviço, a empresa parceira utiliza da infraestrutura da Flow Finance para fornecer crédito para a sua base de clientes.

Os dados de uma dessas operações de CaaS estavam espalhados por diversas planilhas, o que dificultava o controle da operação, assim como a realização de análises com esses dados. Além disso, por não estarem disponíveis online, a empresa parceira que utiliza do serviço da Flow não conseguia ter acesso facilmente aos dados da operação.

O projeto surge então com duas frentes principais. Uma focada em atender as demandas da Flow Finance, e outra da empresa parceira. O ponto de partida de ambas é o mesmo, é necessário organizar os dados da empresa, tratar esses dados, e armazená-los em um local acessível.

O primeiro passo dado, antes de iniciar o trabalho com os dados, foi a definição da arquitetura do sistema. Aqui definiu-se como realizar o ETL, onde armazenar o dado e como seriam construídos os dashboards e o modelo de auxílio na definição do limite de crédito. Essa realização concluiu o primeiro objetivo específico do projeto.

Seguinte a isso, foi realizado o processo de ETL (*Extract, transform, load*) com os dados da operação. Esses dados da operação foram carregados em um banco de dados PostgreSQL, armazenado em um serviço de nuvem da AWS. Feito isso, os objetivos específicos 2, 3 4 e 5 foram atendidos.

Tendo os dados disponíveis na nuvem, o banco foi conectado ao Power Bi, onde foram desenvolvidos dashboards para atender as demandas da Flow Finance e da empresa parceira. Desta forma, os objetivos específicos 6, 7, 8 e 9 foram atendidos.

Além dos dashboards desenvolvidos, foi acoplado um sistema auxiliar à um painel do Power Bi. Esse sistema foi baseado em um estudo estatístico realizado sobre as linhas de crédito fornecidas, e ele auxilia a encontrar a linha de crédito ideal para cada cliente. Isso conclui os últimos dois objetivos específicos (10 e 11) que restavam.

Quando avaliados os resultados do projeto, pode-se afirmar que os requisitos funcionais e não funcionais foram cumpridos, e que os painéis desenvolvidos atendem as demandas, tanto da empresa parceira, quanto da Flow Finance.

Dado o objetivo geral do trabalho de desenvolver um sistema de *business intelligence* para uma fintech de crédito como serviço (CaaS), pode se afirmar que este objetivo foi atingido, visto que os objetivos específicos foram atingidos, e a solução desenvolvida foi validada com todas as partes envolvidas no processo.

Além disso, o trabalho desenvolvido utiliza os dados reais da operação, e já está sendo posto em uso pela empresa. O projeto não é apenas uma prova de conceito,

mas sim uma aplicação final utilizada pela *fintech* em seu dia-a-dia.

Embora o trabalho tenha atingido o seu objetivo, existem ainda diversas melhorias que podem ser feitas nele. A primeira delas surgiu como demanda da própria empresa parceira, e já está sendo implementada. A solicitação deles foi de que fosse disponibilizado em forma de link nos dashboards o arquivo que dá acesso a uma cédula de crédito bancária.

Além da solicitação do parceiro, melhorias podem ser feitas no processo de extração e tratamento dos dados, onde seria de grande valia desenvolver um único algoritmo que conseguisse realizar o processo de extrair os dados de suas fontes, tratá-los e carregá-los no banco.

Ainda olhando para os dados da operação, seria uma adição boa ao projeto passar a puxar os dados da instituição financeira a partir de uma consulta direta no banco de dados deles, e não a partir de um arquivo ".csv".

Quanto ao formato das tabelas no banco, após utilizar os dados da tabela de vencimentos no BI, observou-se que ela pode ser estruturada de uma maneira diferente, e que uma nova tabela pode ser criada no banco para melhorar a usabilidade dos dashboards.

Ademais, a modelagem do banco, que hoje está estruturada utilizando modelagem relacional, pode ser trocada para modelagem dimensional, para se adequar ao formato que é utilizado dentro do Power BI.

Além disso, uma adição ao dashboard que seria de grande valia, seria acrescentar ao painel de auxílio as expansões de crédito um sistema que redigisse um e-mail automaticamente com as informações que estão ali. Visto que os e-mails possuem sempre o mesmo formato, e isso traria ainda mais agilidade para a operação.

Embora o projeto tenha atendido ao que foi proposto, há também essas diversas melhorias que podem ser feitas. Hoje, do jeito que está, o projeto já é uma ótima prova de conceito, e traz um ganho de escala também para a operação. Visto que, diversos processos que eram antes manuais, agora funcionam de forma automática. Não importa se há 1 cliente ou 10.000, o custo operacional será o mesmo.

Por fim, tendo desenvolvido este projeto, ele passa a servir de exemplo para construir painéis de acompanhamento das operações de crédito como serviço que a Flow Finance tem com outras empresas parceiras. Sendo assim, o projeto dá o passo inicial para tornar a Flow Finance uma empresa "*data driven*", e agrega valor ao negócio.

REFERÊNCIAS

ACTS, Equipe de Especialistas Five. **SGBD: O Que É, PARA Que Serve, Tipos E Principais Benefícios**. [S.l.]: Five Acts, jan. 2021. Disponível em:

<https://www.fiveacts.com.br/sghbd/>.

AWS. **O que é banco de dados relacional?** [S.l.]: Filbert Pub., 2021a. Disponível em:

<https://aws.amazon.com/pt/relational-database/>.

CARNEIRO, Luciane. 34 milhões de brasileiros ainda não têm acesso a bancos no país. **Valor Investe**, mar. 2021. Disponível em:

<https://valorinveste.globo.com/produtos/servicos-financeiros/noticia/2021/04/27/34-milhoes-de-brasileiros-ainda-nao-tem-acesso-a-bancos-no-pais.ghtml>.

CONNECTING to data sources via AWS VPN. [S.l.]: AWS Whitepaper, 2021b.

Disponível em: <https://docs.aws.amazon.com/whitepapers/latest/using-power-bi-with-aws-cloud/connecting-to-data-sources-via-aws-vpn.html>.

EDUCATION, IBM Cloud. **cloud-storage**. [S.l.: s.n.], 2021. Disponível em:

<https://www.ibm.com/cloud/learn/cloud-storage>.

GOVERNO FEDERAL. **Estatísticas - Governo Federal, portal do empreendedor**.

[S.l.: s.n.], 2021. Disponível em:

<https://www.gov.br/empresas-e-negocios/pt-br/empreendedor>.

HOKAMA, Daniel Del Bianco; CAMARGO, Denis; FUJITA, Francine;

FOGLINE, João Luiz Valentim. A MODELAGEM DE DADOS NO AMBIENTE DATA WAREHOUSE. **UNIVERSIDADE PRESBITERIANA MACKENZIE**, 2004.

JORNAL DO COMERCIO. No Brasil 59.4 milhões de consumidores estão negativados.

Jornal do Comercio, ago. 2021. Disponível em:

https://www.jornaldocomercio.com/_conteudo/2017/08/economia/578672-no-brasil-59-4-milhoes-de-consumidores-estao-negativados.html.

LEE, Vu; HO, Pong. **Integrate Power BI with Amazon Redshift for insights and analytics**. [S.l.]: Filbert Pub., 2018. Disponível em:

<https://aws.amazon.com/pt/blogs/big-data/integrate-power-bi-with-amazon-redshift-for-insights-and-analytics/>.

MICRO e pequenas empresas geram 27% do PIB do Brasil. [S.l.: s.n.], 2020.

Disponível em:

<https://www.sebrae.com.br/sites/PortalSebrae/ufs/mt/noticias/micro-e-pequenas-empresas-geram-27-do-pib-do-brasil,ad0fc70646467410VgnVCM2000003c74010aRCRD>.

ORACLE. **O que é um Data Warehouse?** [S.l.: s.n.], 2004. Disponível em:

<https://www.oracle.com/br/database/what-is-a-data-warehouse/>.

PÁDUA, Silvia Inês Dallavalle de. **Sistemas de apoio à tomada de decisão.**

[S.l.: s.n.], 2020. Disponível em: https://edisciplinas.usp.br/pluginfile.php/4115468/mod_resource/content/1/Aula%5C%201%5C%20SAD%5C%202017.pdf.

RODRIGUES, Yuri Antonchuk. **Conceitos de Tabelas de Dimensão e Fato na Construção do Data Warehouse.** [S.l.]: MentorsTec, fev. 2020. Disponível em:

<https://www.mentorstec.com.br/post/conceitos-de-tabelas-dimensao-e-fato-na-construcao-do-data-warehouse>.

SENADO NOTÍCIAS. Ajuda do governo só chegou a 15% das MPEs. **Senado notícias**, Agência Senado, out. 2020. Disponível em:

<https://www12.senado.leg.br/noticias/materias/2020/10/13/ajuda-do-governo-so-chegou-a-15-das-micros-e-pequenas-empresas>.

SINGH, VINAY. **Difference Between Cloud Servers and Dedicated Servers.**

[S.l.: s.n.], mar. 2020. Disponível em: <https://cloudacademy.com/blog/difference-between-cloud-servers-dedicated-servers/>.

SONG, Il-Yeol; CHEN, Peter P. Entity Relationship Model. *In: Encyclopedia of Database Systems*. Edição: LING LIU e M. TAMER ÖZSU. Boston, MA: Springer US, 2009. P. 1003–1009. ISBN 978-0-387-39940-9. DOI:

10.1007/978-0-387-39940-9_148. Disponível em:

https://doi.org/10.1007/978-0-387-39940-9_148.

TAIT, Tania; BARCIA, Ricardo; PACHECO, Roberto. UM MODELO DE ARQUITETURA DE SISTEMAS DE INFORMAÇÃO PARA INTEGRAR ASPECTOS TÉCNICOS E ORGANIZACIONAIS NOS SISTEMAS DE INFORMAÇÃO. **PPGEP - USFC**.

TURBAN, Efraim; RAINER, Rexz Kelly; POTTER, Richard E. **Introdução a sistemas De INFORMAÇÃO uma Abordagem gerencial.** [S.l.]: Campus, 2007.

WHAT is a database management system? [S.l.: s.n.]. Disponível em:

<https://www.ibm.com/docs/en/zos-basic-skills?topic=zos-what-is-database-management-system>.

APÊNDICE A – DASHBOARDS DESENVOLVIDOS

Figura 27 – Painel de operações



Fonte – Autoria própria

Figura 29 – Painel de composição das parcelas mensais



Fonte – Autoria própria

Figura 30 – Painel de histórico das parcelas



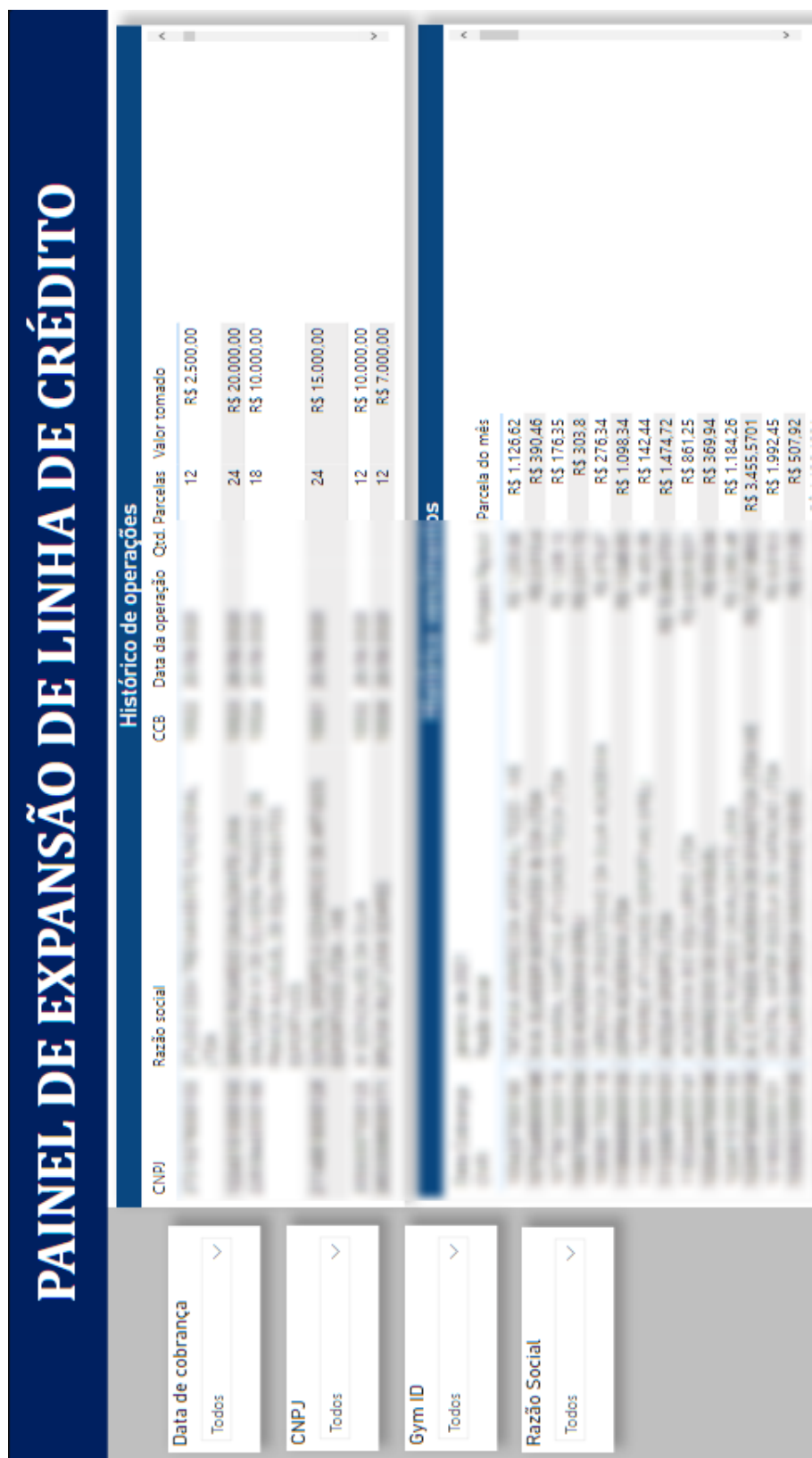
Fonte – Autoria própria

Figura 31 – Painel de cobrança mensal



Fonte – Autoria própria

Figura 32 – Painel de auxílio as expansões de crédito



Fonte – Autoria própria

Figura 33 – Painel de indicadores da operação



Fonte – Autoria própria