Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Renan Luiz Bezerra da Cunha

Ambiente para Engenharia de Dados

Americana, SP

2022

Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Renan Luiz Bezerra da Cunha

Ambiente para Engenharia de Dados

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, sob a orientação do Prof. Diógenes de Oliveira.
Área de concentração: Engenharia de Dados.

Americana, SP

2022

Renan Luiz Bezerra da Cunha

Ambiente para Engenharia de Dados

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de Americana como parte dos requisitos para obtenção do Título de Tecnólogo em Análise e Desenvolvimento de Sistemas pelo Centro Paula Souza.
Área de Atuação : Sistema de Informação

Americana, 22 de junho de 2022.

Banca Examinadora:

Diogenes de Oliveira
Mestre

Adriano Cilhos Doimo
Especialista

Antonio Alfredo Lacerda
Especialista

RESUMO

O projeto Ambiente para Engenharia de Dados trata-se de uma ferramenta para realizar a transformação de dados brutos e disponibilização para consumo onde pode ser utilizado para a criação de dashboards ou reports de forma prática e com a menor intervenção humana possível, com o objetivo transformar dados em informações concisas. O projeto será construído utilizando ferramentas de Big Data com a linguagem principal Python e a metodologia Scrum para seu desenvolvimento. Os diagramas UML serão utilizados para demonstrar as funcionalidades e compreensão do projeto.

Palavras-Chave: Engenharia de Dados; Ciência de Dados; Business Intelligence.

ABSTRACT

The Environment for Data Engineering Project is a tool to transform raw data and make it available for consumption to create dashboards or reports in a practical way with the least possible human intervention with the objective to transform data into concise information. The project will be built using Big Data tools with the main language Python and the Scrum methodology for development. UML diagrams will be used to demonstrate the functionality and understanding of the project.

Keywords: Data Science; Data Engineer; Business Intelligence.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO

Engenharia de Dados é a área que trata os dados brutos de uma companhia. Essa é a primeira etapa de uma série de atividades que transformam os dados em informações para realização de análises, estudos, comparações, previsões ou qualquer objetivo que o detentor dos dados quiser realizar com as informações.

A partir de linguagens como Java, Scala, Python, tecnologias de Big Data e computação em nuvem é possível a construção da arquitetura para a realização dos processamentos dos dados. A criação das pipelines de dados para a realização dos ETLs (extract, transform, load), ou seja, extração, transformação e carga dos dados para que sejam utilizados nas etapas após todos os tratamentos necessários para melhor utilização dos dados.

Este trabalho tem como objetivo desenvolver um ambiente para Engenharia de Dados utilizando as ferramentas disponibilizadas pela AWS para a construção e disponibilização de um Warehouse contendo dados prontos para o consumo que foram tratados nas etapas descritas no trabalho.

O trabalho foi estruturado em quatro capítulos, sendo que o primeiro conceitua e apresenta o levantamento de requisitos do ambiente, alguns ambientes similares ao utilizado no projeto e os recursos e ferramentas utilizados para a sua construção. O segundo capítulo apresenta os diagramas com o intuito de documentar o ambiente e o plano de teste do projeto. No capítulo seguinte é apresentado o desenvolvimento do ambiente e as interfaces capturadas do ambiente e no último capítulo são apresentadas as considerações finais e os principais desafios encontrados durante o desenvolvimento do projeto.

2 PROJETO DO SISTEMA

2.1 Softwares Similares

Atualmente existem diversas soluções para o desenvolvimento de um ambiente para processamento de dados na nuvem. Cada solução tem as suas similaridades, particularidades, vantagens e desvantagens. O projeto proposto utilizará os serviços disponibilizados pela AWS para a criação de um ambiente na nuvem para o processamento e disponibilização desses dados.

- **Amazon Web Services:** “Também conhecido como AWS, é uma plataforma de serviços de computação em nuvem, que formam uma plataforma de computação na nuvem oferecida pela Amazon.com. Os serviços são oferecidos em várias áreas geográficas distribuídas pelo mundo”. (Wikipédia, 2022).
- **Microsoft Azure:** “O Microsoft Azure é uma plataforma destinada à execução de aplicativos e serviços, baseada nos conceitos da computação em nuvem”. (Wikipédia, 2022).
- **Google Cloud Platform:** “Google Cloud Platform é uma suíte de computação em nuvem oferecida pelo Google, funcionando na mesma infraestrutura que a empresa usa para seus produtos dirigidos aos usuários, dentre eles o Buscador Google e o Youtube”. (Wikipédia, 2022).

Tabela 1 – Comparativo de funcionalidades

COMPARAÇÃO DE FERRAMENTAS SILIMARES			
Tools	**AWS (Amazon Web Services)**	**Microsoft Azure**	**GCP (Google Cloud Platform)**
Basic Compute	X	X	X
Containers	X	X	X
Serverless	X	X	X
App Hosting	X	X	X
Batch Processing	X	X	-
Object Storage	X	X	X
Block Storage	X	-	X
File Storage	X	X	-
Hybrid Storage	X	X	-
Offline Data Transfer	X	-	X
Relational/SQL Database	X	X	X

NoSQL Database	X	X	X
In-Memory Database	X	X	-
Archive/Backup	X	X	-
Disaster Recovery	-	X	-
Machine Learning	X	X	X
Cognitive Services	X	X	X
IoT	X	X	X
Networking	X	X	X
Content Delivery	X	X	X
Big Data Analytics	X	X	X
Authentication and Access Management	X	X	X
Security	X	X	X
Application Lifecycle Management	X	X	-
Cloud Monitoring	X	X	X
Cloud Management	X	X	X
AR & VR	X	-	-
Virtual Private Cloud	X	-	X
Training	X	X	X
Support	X	X	X
3rd Party Software and Services	X	X	X

Fonte: Elaborado pelo autor.

2.2 Levantamento de Requisitos

“[...] a engenharia de requisitos, que tem como premissa, conhecer todas as atividades usadas para a produção da documentação necessária ao entendimento do problema, como também a manutenção desses no decorrer do projeto. Outra forma de definir requisito é a descrição de uma condição ou capacidade necessitada por um usuário para resolver um problema ou alcançar um objetivo. Em outras palavras, é o que o sistema deve fazer para implementar uma necessidade de automação requerida pela solução.” (PINHEIRO, 2015, p. 112-113).

2.2.1 Requisitos Funcionais

“Os requisitos funcionais servem para descrever as funcionalidades que se espera que o sistema tenha, isto é, aquilo que o usuário espera que o sistema ofereça, atendendo aos objetivos requisitados. Esses requisitos especificam ações que um

sistema deve ser capaz de executar, sem levar em consideração restrições físicas, portanto, especificam o comportamento de entrada e saída de um sistema.” (PINHEIRO, 2015, p. 118-119).

Tabela 2 – Requisitos funcionais do projeto

Identificação	Requisito Funcional	Prioridade
RF001	Fazer extração de dados	Essencial
RF002	Fazer transformação de dados	Essencial
RF003	Fazer carga de dados	Essencial
RF004	Fazer backup dos dados	Importante

Fonte: Elaborado pelo autor.

2.2.2 Requisitos Não Funcionais

“Os requisitos não-funcionais servem como restrições e estão relacionados ao uso do sistema em termos de desempenho, usabilidade, confiança, segurança, disponibilidade, manutenibilidade, tecnologias envolvidas, utilidade, suporte e escalabilidade. Normalmente não são obtidas com o usuário, já que são características mínimas de um software de qualidade, ficando a cargo do desenvolvedor optar por atender esses requisitos.” (PINHEIRO, 2015, p. 119).

Tabela 3 – Requisitos não funcionais do projeto

Identificação	Requisito Não Funcional	Categoria	Prioridade
RNF001	Governança de dados	Segurança	Essencial
RNF002	Confiabilidade dos dados	Usabilidade	Essencial
RNF003	Transparência nas políticas	Padrões	Essencial
RNF004	Eficiência de desempenho	Desempenho	Essencial

RNF005	Otimização de custos	Custos	Desejável

Fonte: Elaborado pelo autor.

2.3 Recursos e Ferramentas

Nesta sessão serão apresentados os recursos utilizados para o desenvolvimento do projeto.

- **Python:** “Python é uma linguagem de programação interpretada, orientada a objetos e de alto nível com semântica dinâmica. Suas estruturas de dados integradas de alto nível, combinadas com tipagem dinâmica e vinculação dinâmica, o tornam muito atraente para o desenvolvimento rápido de aplicativos, bem como para uso como uma linguagem de script ou cola para conectar componentes existentes. A sintaxe simples e fácil de aprender do Python enfatiza a legibilidade e, portanto, reduz o custo de manutenção do programa. Python suporta módulos e pacotes, o que incentiva a modularidade do programa e a reutilização de código. O interpretador Python e a extensa biblioteca padrão estão disponíveis em formato fonte ou binário gratuitamente para todas as principais plataformas e podem ser distribuídos gratuitamente.” (PYTHON, 2022).
- **PySpark:** “PySpark é uma interface para Apache Spark em Python. Ele não apenas permite que você escreva aplicativos Spark usando APIs Python, mas também fornece o shell PySpark para analisar interativamente seus dados em um ambiente distribuído. O PySpark suporta a maioria dos recursos do Spark, como Spark SQL, DataFrame, Streaming, MLlib (Machine Learning) e Spark Core.” (PYSPARK, 2022).
- **AWS IAM:** “O AWS Identity and Access Management (IAM) é um serviço da Web que ajuda você a controlar o acesso aos recursos da AWS de forma segura. Você usa o IAM para controlar quem é autenticado (fez login) e autorizado (tem permissões) a usar os recursos.” (AWS IAM, 2022).
- **AWS Lake Formation:** “O AWS Lake Formation é um serviço que facilita a configuração de um data lake seguro em dias. Um data lake é um repositório centralizado, administrado e seguro que armazena todos os seus dados,

tanto em sua forma original quanto preparados para análise. Um data lake permite romper os silos de dados e combinar diferentes tipos de análises para obter insights e orientar as melhores decisões de negócios.” (AWS LAKE FORMATION, 2022).

- **Amazon S3:** “O Amazon Simple Storage Service (Amazon S3) é um serviço de armazenamento de objetos que oferece escalabilidade, disponibilidade de dados, segurança e performance líderes do setor. Clientes de todos os portes e setores podem armazenar e proteger qualquer quantidade de dados de praticamente qualquer caso de uso, como data lakes, aplicações nativas da nuvem e aplicações móveis. Com classes de armazenamento econômicas e recursos de gerenciamento fáceis de usar, você pode otimizar custos, organizar dados e configurar controles de acesso ajustados para atender a requisitos específicos de negócios, organizacionais e de conformidade.” (AMAZON S3, 2022).
- **AWS Glue:** “O AWS Glue é um serviço de ETL (extração, transformação e carregamento) totalmente gerenciado que torna mais fácil e econômico o processo de categorizar dados, limpá-los, aprimorá-los e movê-los de modo confiável entre vários armazenamentos e streams de dados.” (AWS GLUE, 2022).
- **Amazon Athena:** “O Amazon Athena é um serviço de consultas interativas que facilita a análise de dados no Amazon S3 usando SQL padrão. O Athena não precisa de servidor. Portanto, não há infraestrutura para gerenciar e você paga apenas pelas consultas executadas.” (AMAZON ATHENA, 2022).
- **Amazon RDS:** “O Amazon Relational Database Service (RDS) é uma coleção de serviços gerenciados que facilita a configuração, operação e escalabilidade de bancos de dados na nuvem.” (AMAZON RDS, 2022).
- **AWS Glue DataBrew:** “O AWS Glue DataBrew é uma ferramenta visual de preparação de dados que permite aos usuários limpar e normalizar dados sem escrever nenhum código. O uso do DataBrew ajuda a reduzir o tempo necessário para preparar dados para análise e machine learning (ML) em até 80% em comparação com a preparação de dados desenvolvida sob medida. Você pode escolher entre mais de 250 transformações prontas para

automatizar tarefas de preparação de dados, como filtrar anomalias, converter dados em formatos padrão e corrigir valores inválidos.” (AWS DATABREW, 2022).

- **Amazon VPC:** “A Amazon Virtual Private Cloud (Amazon VPC) permite executar recursos da Amazon Web Services em uma rede virtual definida por você. Essa rede virtual se assemelha a uma rede tradicional que você operaria no seu datacenter, com os benefícios de usar a infraestrutura dimensionável do AWS.” (AMAZON VPC, 2022).
- **Amazon Kinesis:** “O Amazon Kinesis oferece recursos essenciais para processar dados de streaming em qualquer escala de forma econômica, além da flexibilidade de escolher as ferramentas mais adequadas aos requisitos dos aplicativos. Com o Amazon Kinesis, você pode consumir dados em tempo real como vídeo, áudio, logs de aplicativos, clickstreams de sites e dados de telemetria de IoT para machine learning, análises e outros aplicativos. O Amazon Kinesis permite processar e analisar dados assim que são recebidos e responder instantaneamente, em vez de aguardar a conclusão da coleta de dados para poder iniciar o processamento.” (AMAZON KINESIS, 2022).
- **Amazon CloudTrail:** “AWS CloudTrail é um serviço AWS que lhe permite administrar, manter-se compatível e realizar auditorias operacionais e de risco na sua conta AWS. As ações realizadas por um usuário, uma função ou um serviço da AWS são registradas como eventos no CloudTrail. Os eventos incluem ações realizadas em AWS Management Console, AWS Command Line Interface, e AWS SDKs e APIs.” (AMAZON CLOUDTRAIL, 2022).
- **Amazon DynamoDB:** “O Amazon DynamoDB é um serviço de banco de dados NoSQL totalmente gerenciado que fornece uma performance rápida e previsível com escalabilidade integrada. O DynamoDB permite que você transfira os encargos administrativos de operação e escalabilidade de um banco de dados distribuído.” (AMAZON DYNAMODB, 2022).
- **Amazon Redshift:** “O Amazon Redshift usa SQL para analisar dados estruturados e semiestruturados em data warehouses, bancos de dados operacionais e data lakes, usando hardware e machine learning projetados

pela AWS para oferecer a melhor performance de preço em qualquer escala.” (AMAZON REDSHIFT, 2022).

- **Amazon Macie:** “O Amazon Macie é um serviço de segurança e privacidade de dados totalmente gerenciado que usa machine learning e correspondência de padrões para descobrir e proteger seus dados confidenciais na AWS.” (AMAZON MACIE, 2022).

3 MODELAGEM

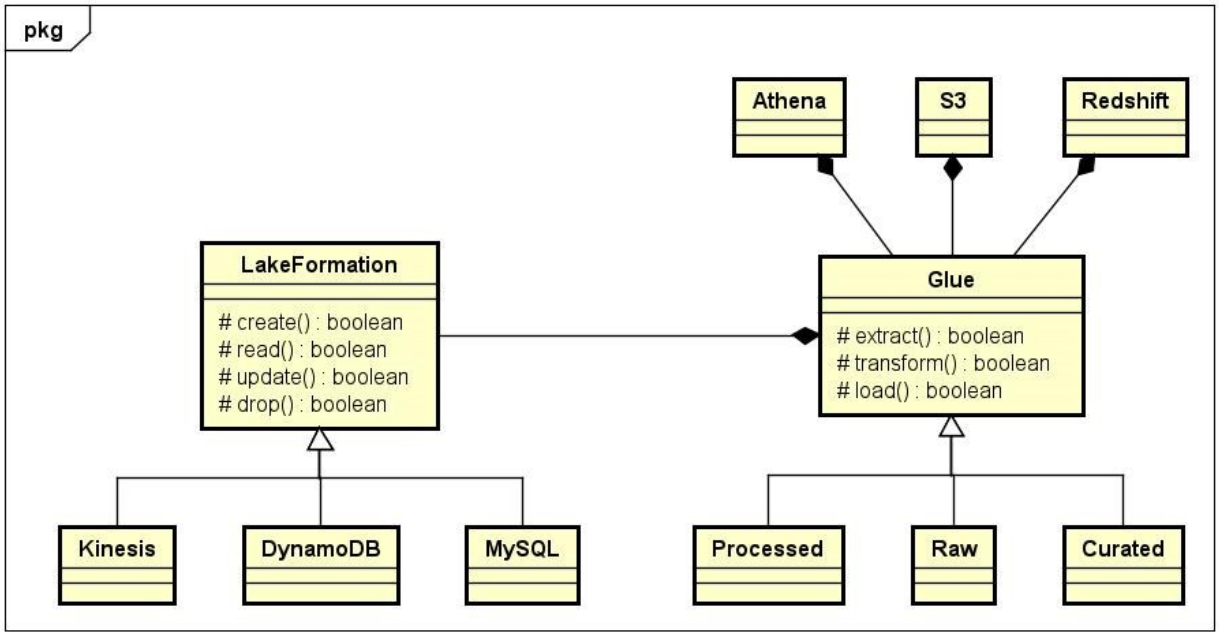
A documentação deste projeto utilizará a linguagem de modelagem Unified Modeling Language (UML) para realizar a modelagem do diagrama de classes, casos de uso e diagrama de estados.

“A UML representa símbolos, esses usados em diagramas que assim representam uma linguagem simbólica com regras claras e precisas para utilização desses símbolos nos diversos diagramas. O objetivo dos diagramas é apresentar múltiplas visões do sistema chamado de modelo. Assim, um modelo UML é um conjunto de diagramas que servem para compreender e desenvolver um projeto de software, descrevendo o que o software deve fazer”. (PINHEIRO, 2015, p. 145).

3.1 Diagrama de Classes

“Os diagramas de classe registram atributos e operações de uma classe e as restrições de como os objetos podem ser conectados, descrevendo também os tipos de objetos no sistema e os relacionamentos entre eles e esses podem ser associações e abstrações.” (PINHEIRO, 2015, p. 154).

Figura 1 – Diagrama de classes



Fonte: Elaborado pelo autor.

A figura 1 representam os jobs em que o ambiente utiliza para a criação dos bancos estruturados ou não estruturados e as funcionalidades para a execução dos ETLs (Extract, Transform e Load).

- **LakeFormation –** Utilizado para a gerenciamento dos bancos de dados desde a criação de databases e tabelas, execução de queries, atualização e exclusão das tabelas.
- **Glue –** Utilizado para a movimentação, limpeza e inserção dos dados entre os buckets e ambientes.
- **Kinesis –** Ferramenta utilizada para a inserção de dados stream (em tempo real), serve tanto para dados estruturados, não estruturados e semiestruturados.
- **DynamoDB –** Banco utilizado para armazenamento de dados noSQL (não-relacional).
- **MySQL –** Banco utilizado para o armazenamento de dados relacionais.
- **Raw –** Bucket S3 utilizado para o armazenamento de dados em parquets extraídos do LakeFormation.

- **Processed –** Bucket S3 utilizado para o armazenamento de dados após limpeza e transformações dos dados extraídos de Raw.
- **Curated –** Bucket S3 utilizado para o armazenamento de dados prontos para armazenamento e consumo do Warehouse e Reports que foram extraídos de Processed.
- **Athena –** Ferramenta utilizada para realizar consulta nos dados armazenados nos Buckets S3.
- **S3 –** Utilizado para armazenamento dos dados em Buckets e divisão entre ambientes para o processamento dos dados.
- **Redshift –** Utilizado para criação de cluster para o consumo de dados do Warehouse do ambiente.

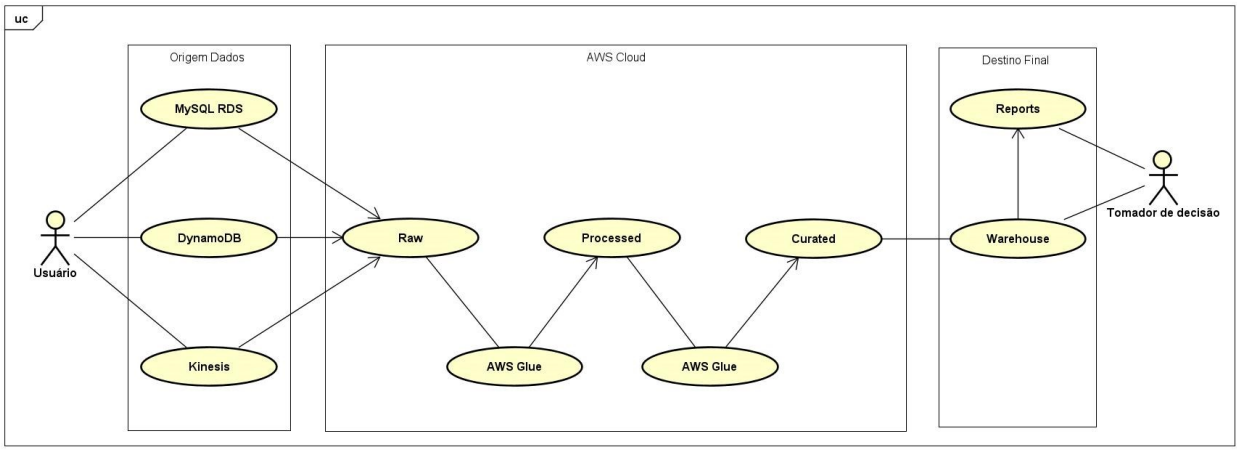
3.2 Casos de Uso

“Um caso de uso descreve quais comportamentos o sistema deverá responder para cada um dos usuários do mesmo, servindo de formalização das ações que precisarão ser desenvolvidas. Retratando uma lista de eventos entre os usuários e o sistema em uma visão abstrata, onde essa lista de eventos relatados abstratamente descreve as interações desde o início da atividade até o fim da mesma.” (PINHEIRO, 2015, p. 148-149).

Os atores que interagem com o sistema são: o usuário por si que realizará a inserção dos dados brutos e o tomador de decisão que poderá consumir os dados tratados.

- **Usuário:** é o ator que realiza a inserção dos dados brutos que podem partir de uma planilha do excel, uma tabela, arquivos JSON, entre outros tipos de dados.
- **Tomador de decisão:** esse ator pode ser uma outra aplicação que realizará o consumo dos dados, pode ser utilizado para a criação de dashboards e reports, apenas ser armazenado para históricos, pode também ser consumidos por cientistas de dados, engenheiros de dados ou mesmo ser realizadas novas transformações para outras áreas ou propósitos.

Figura 2 – Diagrama de casos de uso



Fonte: Elaborado pelo autor.

No subcapítulo 3.2.1 será apresentada a documentação dos casos de uso do projeto.

3.2.1 Documentação dos Casos de Uso

As funcionalidades dos diagramas de casos de uso serão descritas entre a tabela 4 até a tabela 8.

Tabela 4 – Caso de uso “Origem dados”

Nome do caso de uso	Origem dados
Atores envolvidos	Usuário, Ambiente
Objetivo	Inserir os dados brutos no ambiente
Ações do Ator	**Ações do Sistema**
1. Importar os dados para dentro do ambiente respeitando as regras: - MySQL RDS: dado relacional - DynamoDB: dado não-relacional - Kinesis: stream (caso a leitura tenha que ocorrer em tempo real)	
	2. Armazenar os dados de maneira otimizada

Fonte: Elaborado pelo autor.

Tabela 5 – Caso de uso “Raw”

Nome do caso de uso	Raw
Atores envolvidos	Usuário, Ambiente
Objetivo	Extrair dados brutos para o Bucket S3 para salvá-los em Parquets para otimizar os demais passos a serem executados na sequência
Ações do Ator	**Ações do Sistema**
1. Criar o Bucket S3	
	2. Extrair os dados brutos
	3. Armazenar os dados em Parquets no Bucket S3 criado pelo usuário

Fonte: Elaborado pelo autor.

Tabela 6 – Caso de uso “AWS Glue”

Nome do caso de uso	AWS Glue
Atores envolvidos	Usuário, Ambiente
Objetivo	Realizar o ETL para entrega dos dados para a próxima etapa do ambiente
Ações do ator	**Ações do Sistema**
1. Configurar o ETL conforme a necessidade de tratativa dos dados	
	2. Extrair os dados do Bucket S3
	3. Aplicar as transformações configuradas pelo usuário
	4. Fazer a carga dos dados no local configurado pelo usuário

Fonte: Elaborado pelo autor.

Tabela 7 – Caso de uso “Processed” e “Curated”

Nome do caso de uso	Processed, Curated
Atores envolvidos	Usuário, Ambiente
Objetivo	Extrair dados do Bucket S3 da etapa anterior para realizar as transformações

	necessárias e disponibilizar para os demais passos a serem executados na sequência
Ações do Ator	**Ações do Sistema**
1. Criar Bucket S3	
2. Configurar o ETL conforme a necessidade de tratativa dos dados	
	3. Extrair os dados do Bucket S3
	4. Aplicar as transformações configuradas pelo usuário

Fonte: Elaborado pelo autor.

Tabela 8 – Caso de uso “Warehouse” e “Reports”

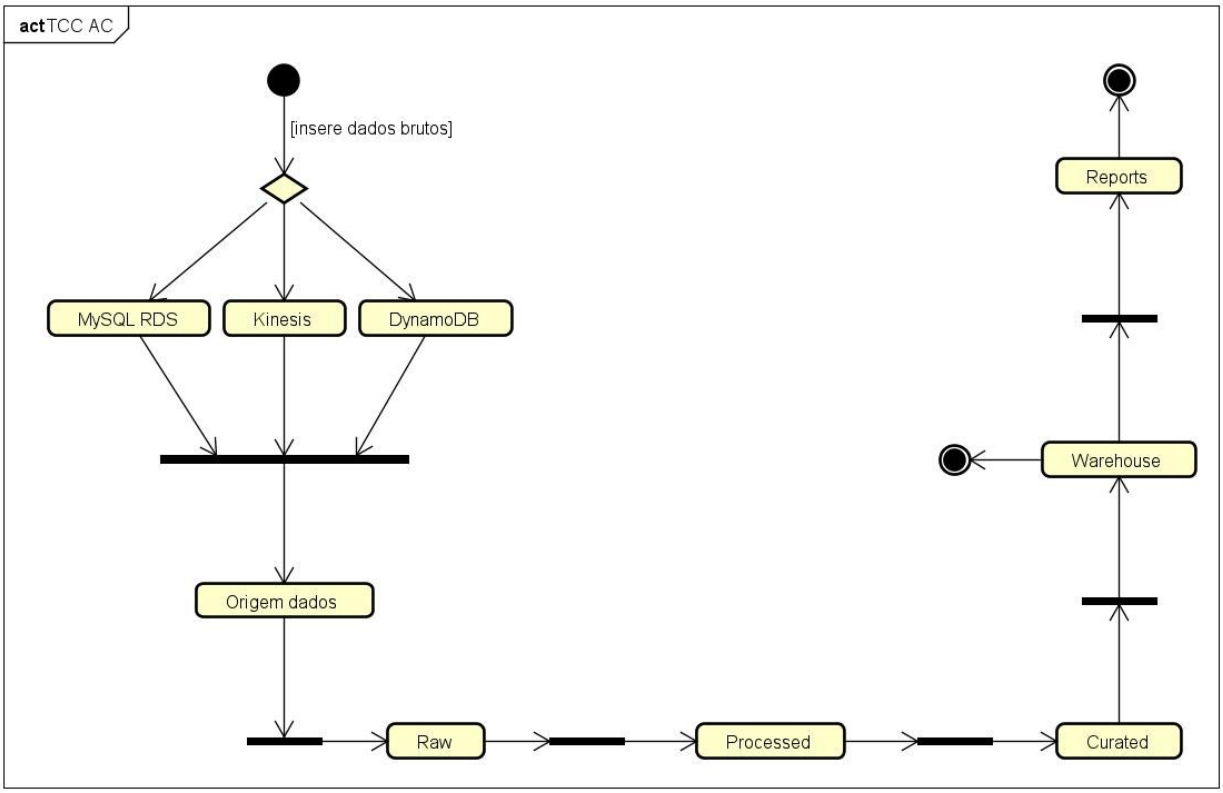
Nome do caso de uso	Warehouse, Reports
Atores envolvidos	Usuário, Ambiente
Objetivo	Disponibilizar para consumo do usuário final
Ações do Ator	**Ações do Sistema**
	1. Manter a disponibilidade dos dados para realização de queries
2. Consumir os dados conforme a necessidade	

Fonte: Elaborado pelo autor.

3.3 Diagrama de Estados

“O diagrama de estados tem a finalidade de exibir como um objeto realiza uma determinada operação num determinado momento da execução, representando um estado particular.” (PINHEIRO, 2015, p. 162).

Figura 3 – Diagrama de estados



Fonte: Elaborado pelo autor.

Os estados representados mostram:

- **Insere dados brutos:** A partir do ponto inicial do ambiente o usuário deve realizar a importação para o banco adequado seguindo a seguinte regra: caso o dado seja relacional, ele deve ser armazenado no MySQL RDS. Caso o dado seja não-relacional, deve ser armazenado no DynamoDB. Caso a leitura tenha que ocorrer em tempo real (streaming-data), independente do tipo de dado que esteja sendo importado, deve ser armazenado no Kinesis.
- **Origem dados:** É realizada a leitura de dados a partir de seus bancos de origem.
- **Raw:** Após os dados serem importados inicia-se a fase de transformações de dados. Nessa fase os dados são lidos de sua origem e serão armazenados em formato parquet, para otimizar as consultas e serão armazenados em um Bucket S3 para futuros tratamentos.

- **Processed:** Os dados serão lidos do Bucket S3 anterior e através de um ETL configurado via AWS Glue os dados serão tratados e armazenados no Bucket S3 para a próxima etapa do processo.
- **Curated:** Após os dados passarem por tratamento nas etapas anteriores é possível realizar joins entre as tabelas armazenadas e armazená-las de forma que os dados estejam prontos para o consumo, conforme o usuário final desejar. Esses dados são finalizados via ETL configurado via AWS Glue e serão carregados no Warehouse.
- **Warehouse:** Dados ficam armazenados após os tratamentos realizados nas etapas anteriores e estão prontos para o consumo, após essa etapa o processo pode ser finalizado ou seguir para a parte de Reports.
- **Reports:** Os dados serão consumidos do Warehouse e conforme o usuário final configurar podem ser utilizados para a criação de dashboards ou reports.

3.4 Plano de Testes

“São sequencias de ações, operações que são executadas com o objetivo de encontrar problemas no software e encontrá-los o mais cedo possível. Aumentando a percepção de qualidade geral do software e garantindo que o usuário final tenha um software que atende suas necessidades. De forma geral, o processo de teste pode ser separado em 4 grandes ações: Planejar – entender o que precisa ser testado e definir como. Preparar – selecionar e montar todo o ambiente para execução. Executar – executar os testes e coletar o resultado. Avaliar – verificar os resultados e métricas para melhorar os testes.” (PINHEIRO, 2015, p. 463-464).

O plano de teste do ambiente ocorre em tempo real de execução. Quando ocorre um erro durante a criação de um bucket, um database ou uma tabela o erro é mostrado no momento da tentativa. Quando o erro ocorre em algum processo do ETL o dado não chega ao seu destino e o processo acusa que finalizou com erro. Pode ser utilizado o Cloudtrail para a checagem dos logs e o próprio processo acusa o status que foi finalizado.

4 DESENVOLVIMENTO

A metodologia ágil adotada para o desenvolvimento do projeto foi a metodologia Scrum.

“Scrum é o processo ágil que tem como objetivo focar na entrega dos requisitos de maior valor agregado ao cliente no menor intervalo de tempo possível. Permitindo assim que, adequações sejam realizadas no software de maneira rápida e contínua, fazendo com que o produto final esteja em conformidade com as necessidades do referido cliente.” (PINHEIRO, 2015, p. 107).

Nesse modelo os ciclos são divididos em 2 semanas, no qual um conjunto de atividades devem ser executadas. Esse trabalho é dividido em iterações.

Ao início de uma Sprint é realizada uma Planning Meeting, nessa reunião são priorizados os itens que devem ser implementados nessa próxima Sprint e são selecionadas as atividades que serão implementadas durante a Sprint que está para iniciar. Essas tarefas então são transferidas do Backlog para o Sprint Backlog.

Daily é o nome da reunião que ocorre diariamente para divulgar o que foi realizado no dia anterior, identificar possíveis impedimentos que podem atrasar o trabalho e o que será priorizado no dia que irá iniciar.

Ao final de cada Sprint é apresentada as funcionalidades implementadas em uma Retrospective celebrando os pontos altos naquela Sprint, quais foram pontos de atenção, o que pode ser melhorado, dúvidas, entre outros pontos. Junto a Retrospective, ou mesmo em uma reunião separada, a Planning é onde é realizado o planejamento para a próxima Sprint.

4.1 Planejamento

Foi utilizado um cronograma com as atividades essenciais para o desenvolvimento do ambiente. A tabela 9 indica o cronograma com as principais entregas realizadas, nela são descritas as atividades de forma simples, a data de início, a data de término, a quantidade de horas esperadas para a realização da atividade e as horas gastas durante o processo.

Tabela 9 – Cronograma de atividades de desenvolvimento do projeto

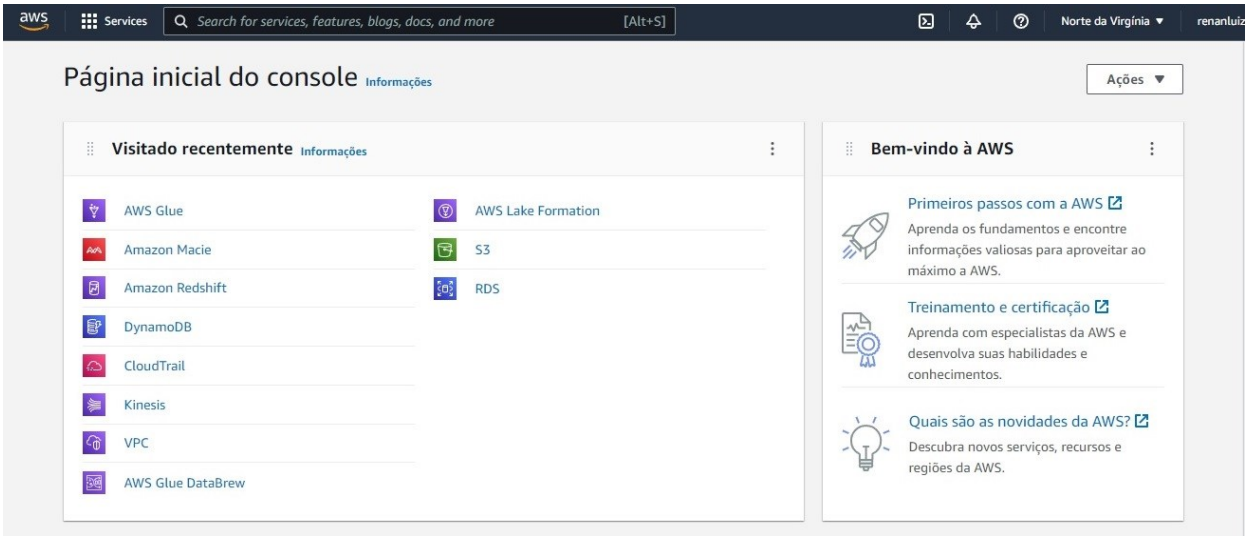
Atividades		Início	Término	Horas	
				Esperadas	Gastas
A1	Definir projeto	19/04/2022	19/04/2022	4h	5h
A2	Realizar esboço da solução	20/04/2022	20/04/2022	3h	3h
A3	Criar conta AWS	21/04/2022	21/04/2022	1h	30m
A4	Configurar IAM	21/04/2022	21/04/2022	1h	1h30m
A5	Criar buckets S3 no Lake Formation	22/04/2022	22/04/2022	4h	3h
A6	Importar dados brutos	23/04/2022	23/04/2022	5h	5h
A7	Configurar ETLs no Glue	25/04/2022	03/05/2022	20h	22h
A8	Realizar carga dados nos ambientes	25/04/2022	03/05/2022	4h	4h
A9	Criar cluster para carga final dos dados	03/05/2022	03/05/2022	4h	4h
A10	Configuração Macie para dados vulneráveis	04/05/2022	04/05/2022	2h	1h
A11	Verificação final do ambiente	04/05/2022	05/05/2022	3h	3h30m

Fonte: Elaborado pelo autor.

4.2 Interfaces de Usuário

As principais telas utilizadas para realizar o projeto de criação de ambiente para engenharia de dados serão apresentados entre a figura 4 a figura 19.

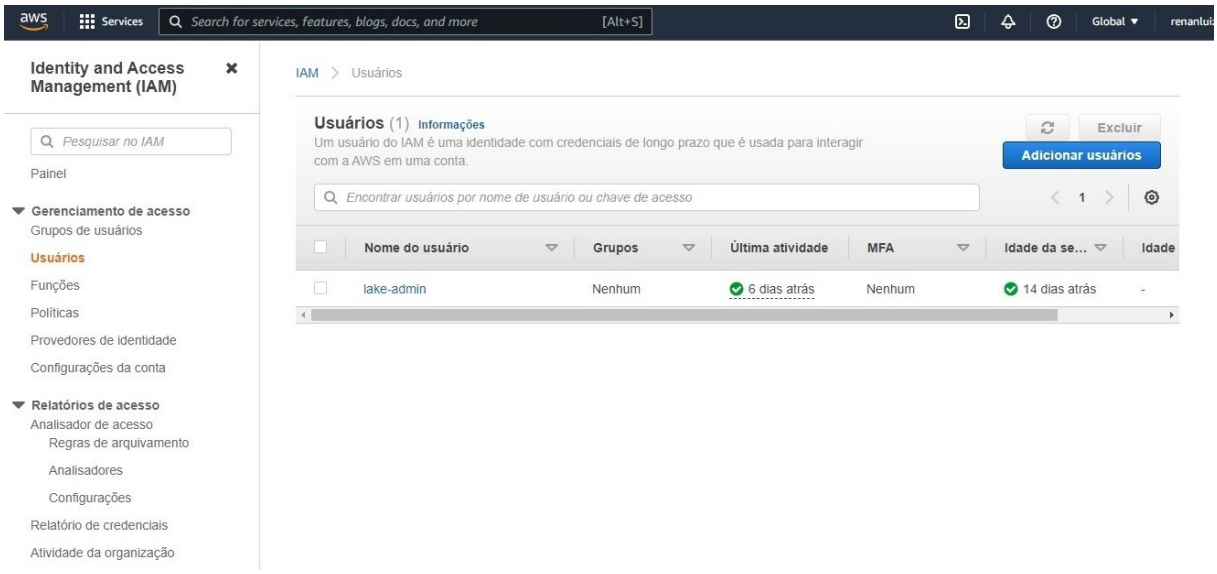
Figura 4 – Captura de tela home plataforma AWS



Fonte: Elaborado pelo autor.

A figura 4 apresenta a tela inicial do ambiente possui basicamente um header presente em todas as telas do ambiente com uma barra de pesquisas, a localização do ambiente que o ambiente está sendo configurado, o menu do usuário, as últimas ferramentas utilizadas, acesso as documentações e a alguns treinamentos disponibilizados pela AWS.

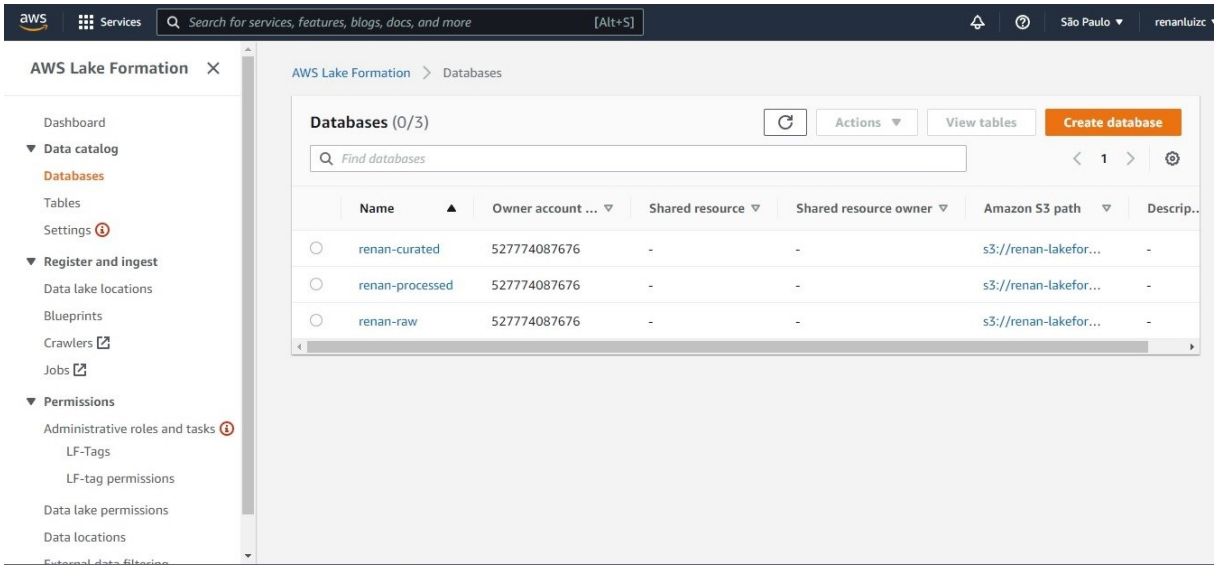
Figura 5 – Captura de tela IAM plataforma AWS



Fonte: Elaborado pelo autor.

A figura 5 apresenta o IAM, onde é realizado o cadastro dos usuários, adicionadas as permissões e políticas que cada usuário terá acesso. Aqui é possível realizar definir e configurar grupos de acesso, se o usuário terá permissão para ler, criar ou atualizar as tabelas, entre todas as permissões que são possíveis configurar no ambiente.

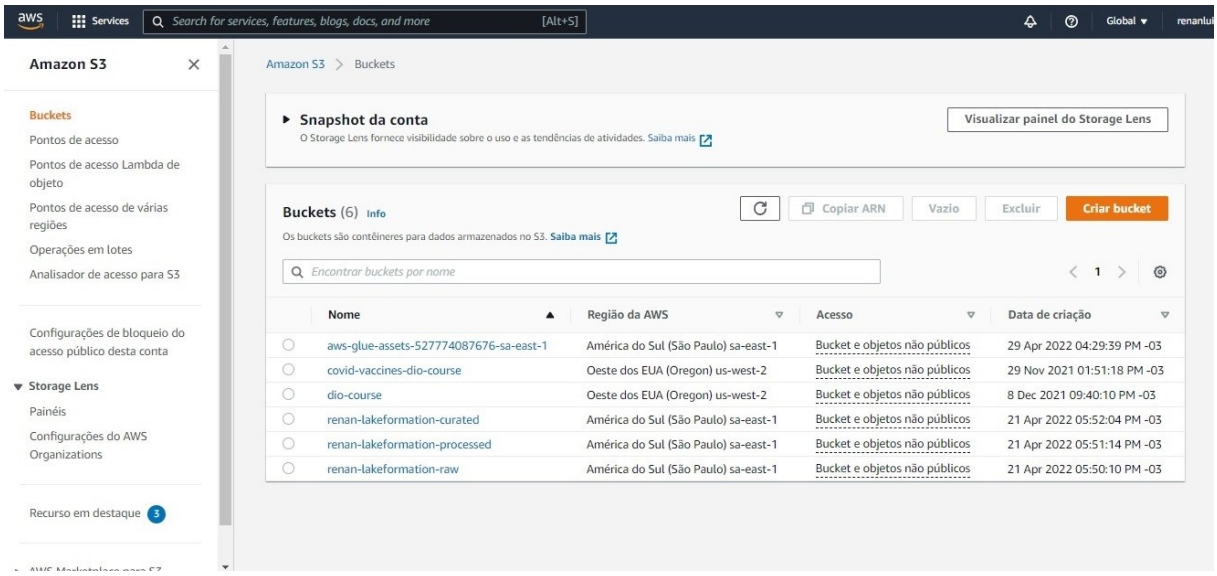
Figura 6 – Captura de tela Lake Formation plataforma AWS



Fonte: Elaborado pelo autor.

A figura 6 apresenta o Lake Formation do ambiente, onde são criados os bancos de dados onde serão armazenados os dados durante os ETLs que serão realizados. Nessa parte do sistema é possível realizar a criação dos Buckets S3 e as tabelas que forem sendo criadas conforme forem sendo realizadas as cargas de dados nos ambientes.

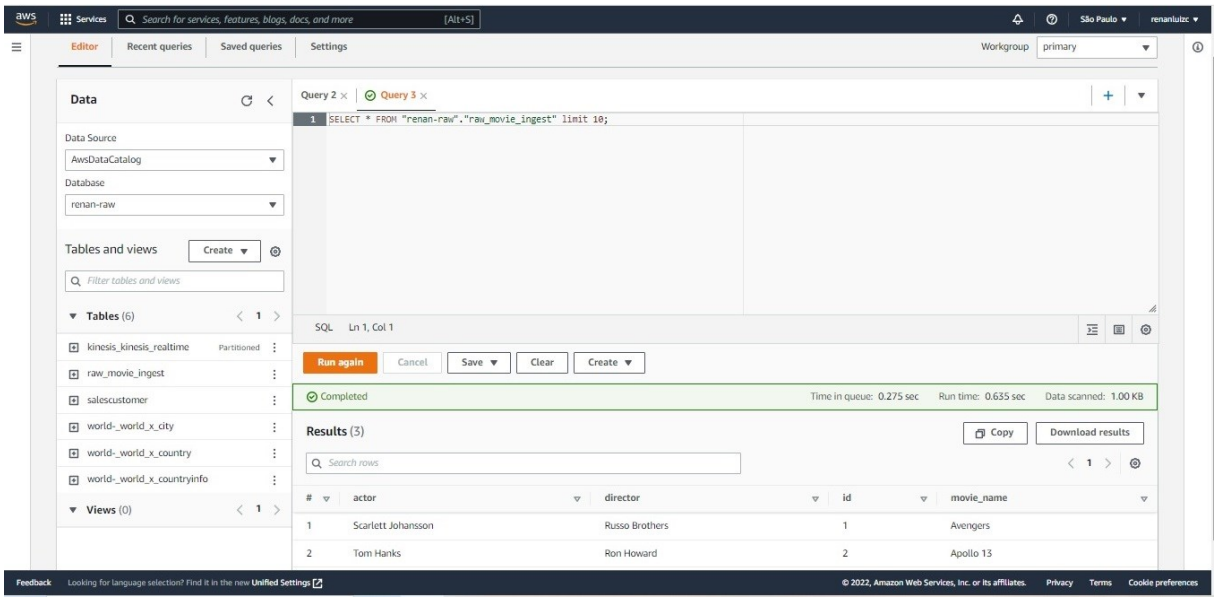
Figura 7 – Captura de tela S3 plataforma AWS



Fonte: Elaborado pelo autor.

A figura 7 apresenta o Amazon S3 onde é possível gerenciar os Buckets criados no ambiente.

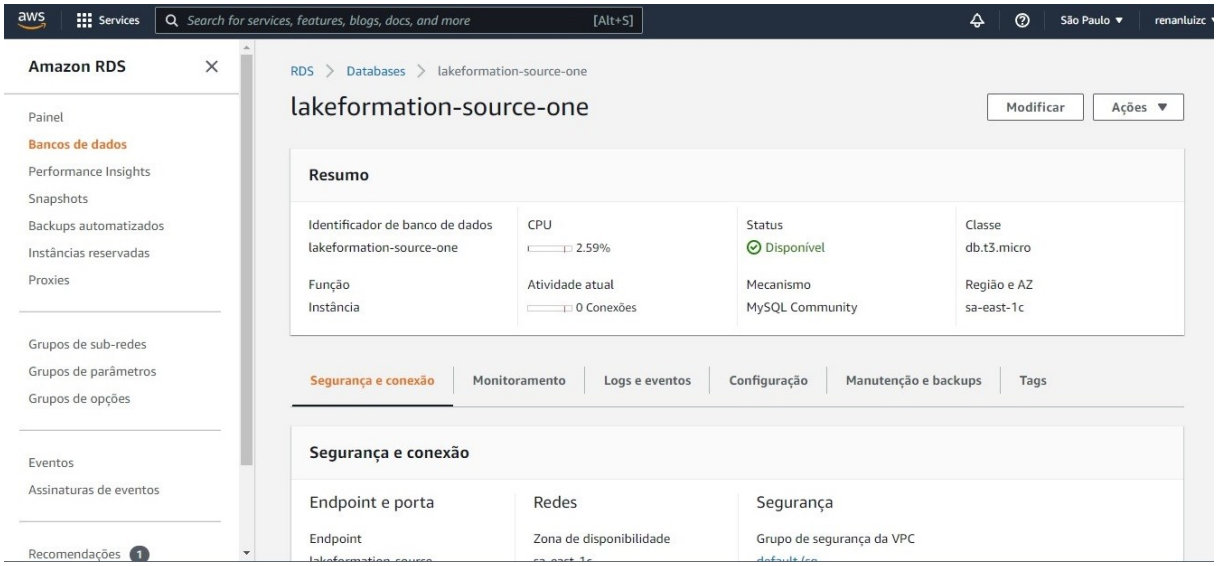
Figura 8 – Captura de tela Athena plataforma AWS



Fonte: Elaborado pelo autor.

A figura 8 possui a captura de tela do AWS Athena, onde são realizadas as queries nos dados armazenados via linguagem SQL.

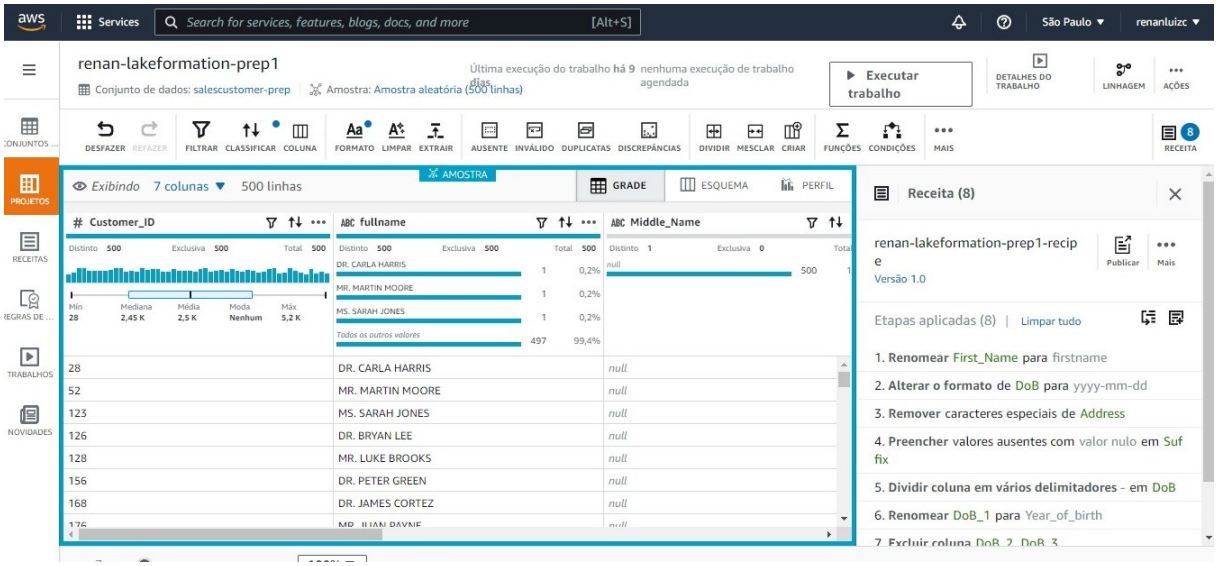
Figura 9 – Captura de tela RDS plataforma AWS



Fonte: Elaborado pelo autor.

Na figura 9 apresenta a tela do Amazon RDS, onde é realizada a criação de uma instância para a importação dos dados para o ambiente.

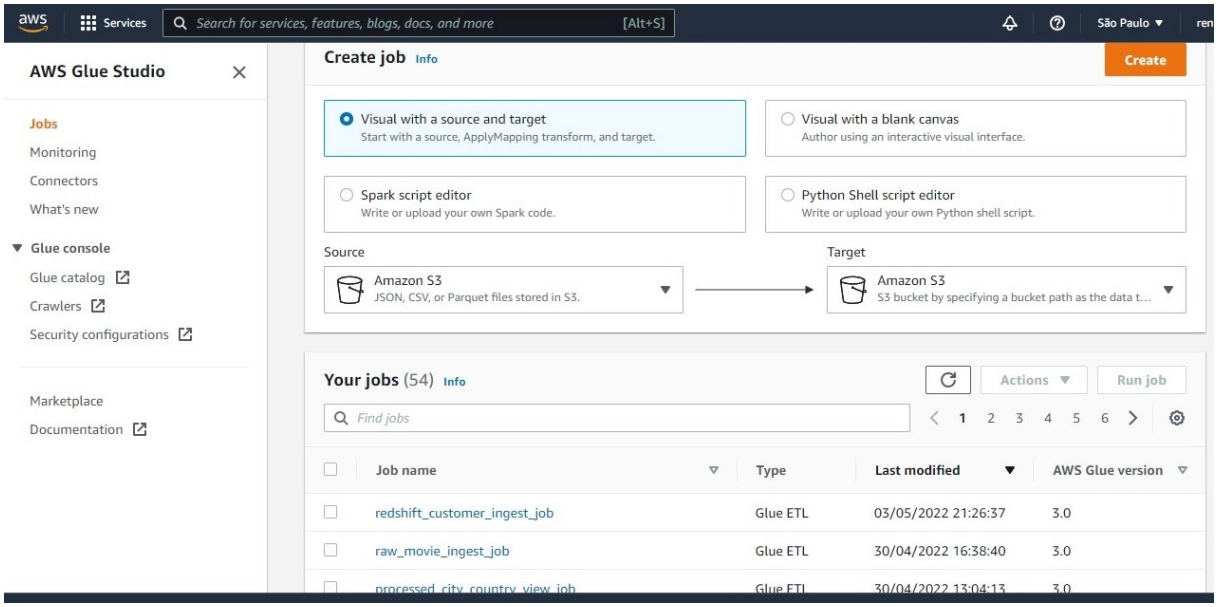
Figura 10 – Captura de tela Glue Databrew plataforma AWS



Fonte: Elaborado pelo autor.

A figura 10 apresenta o Glue Databrew onde é possível realizar o ETL nos dados brutos, nesse passo você cria uma “receita” utilizando as ferramentas visuais do ambiente e publicá-las para serem rodadas.

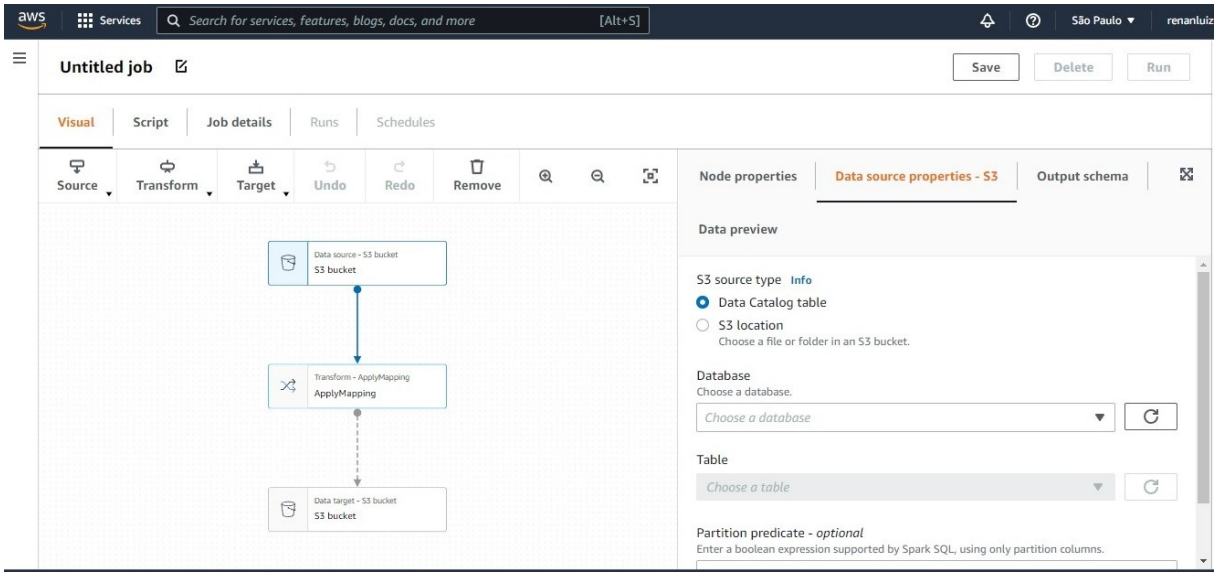
Figura 11 – Captura de tela home Glue Studio plataforma AWS



Fonte: Elaborado pelo autor.

Na figura 11 é apresentado o Glue Studio, onde é possível realizar a criação e monitoramento dos ETLs.

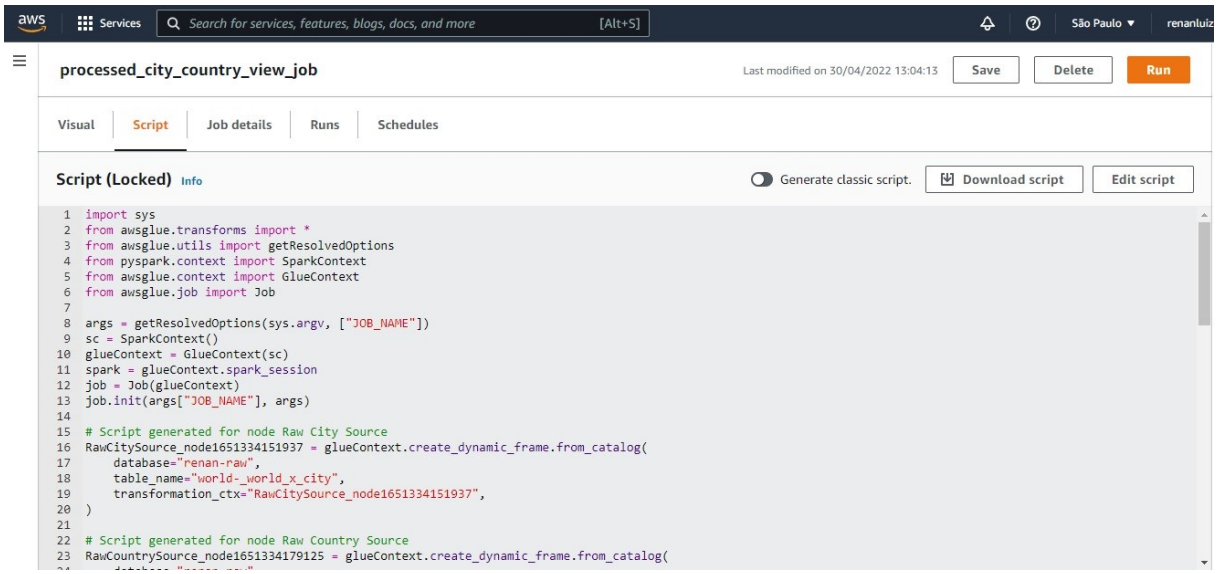
Figura 12 – Captura de tela Glue Studio ETL plataforma AWS



Fonte: Elaborado pelo autor.

A figura 12 apresenta a parte do Glue Studio onde é configurado o ETL, é possível criar o ETL utilizando as ferramentas visuais do ambiente extraindo o dado apontando para a sua origem, aplicar as transformações (Join, Select, Drop, Mapping, entre outros) e realizar a carga no destino final.

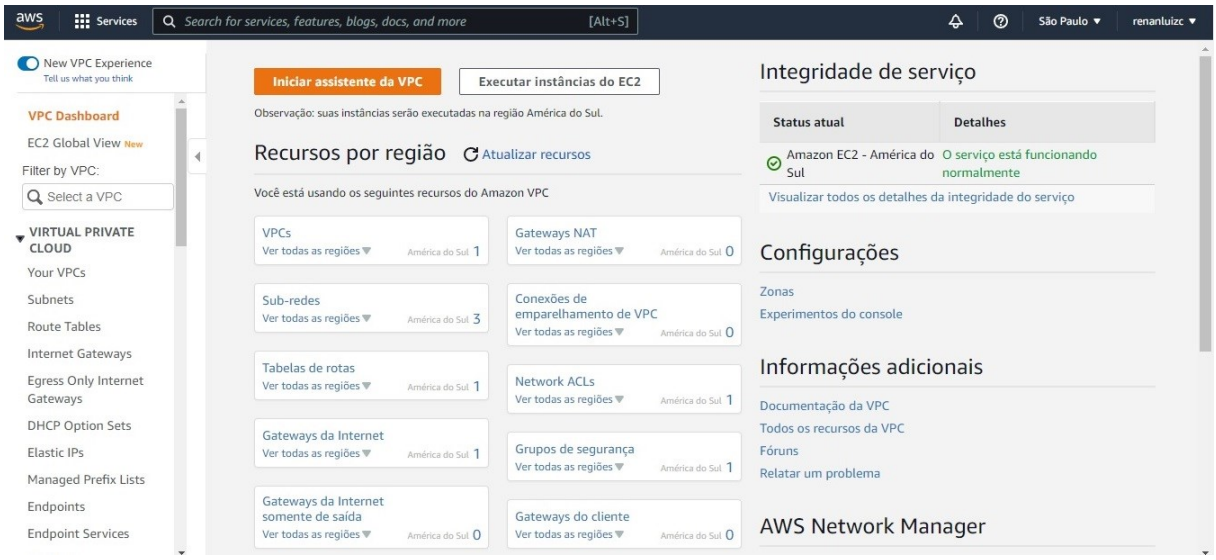
Figura 13 – Captura de tela Glue Studio script plataforma AWS



Fonte: Elaborado pelo autor.

Na figura 13 ainda no Glue Studio, é possível realizar a programação do ETL utilizando linguagem Python.

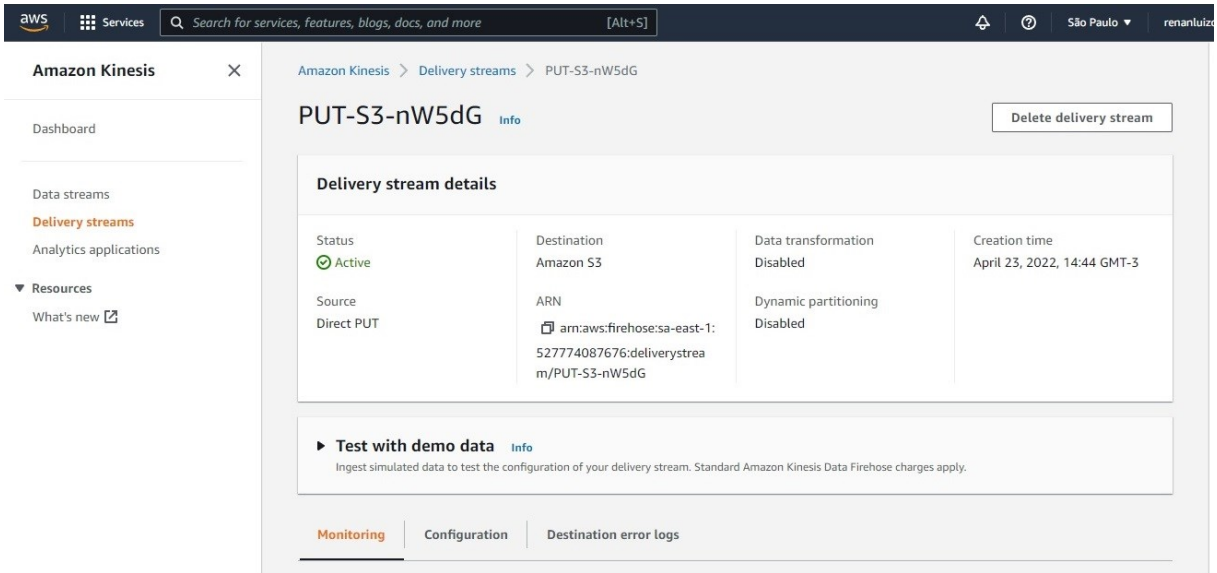
Figura 14 – Captura de tela VPC plataforma AWS



Fonte: Elaborado pelo autor.

Na figura 14 é apresentado o dashboard da VPC (Virtual Private Cloud), onde é realizado o provisionamento do serviço de cloud.

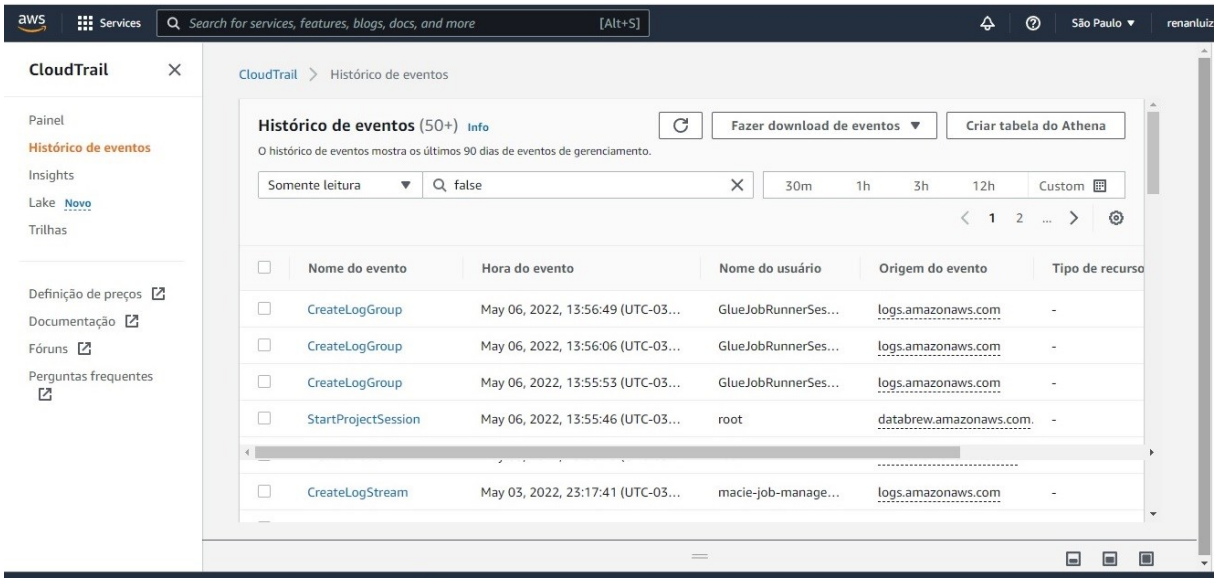
Figura 15 – Captura de tela Kinesis plataforma AWS



Fonte: Elaborado pelo autor.

A figura 15 mostra a tela do Kinesis, ferramenta onde são gerenciados os dados realizados em tempo real (stream), aqui é possível configurar a origem dos dados e a periodicidade em que os dados serão lidos.

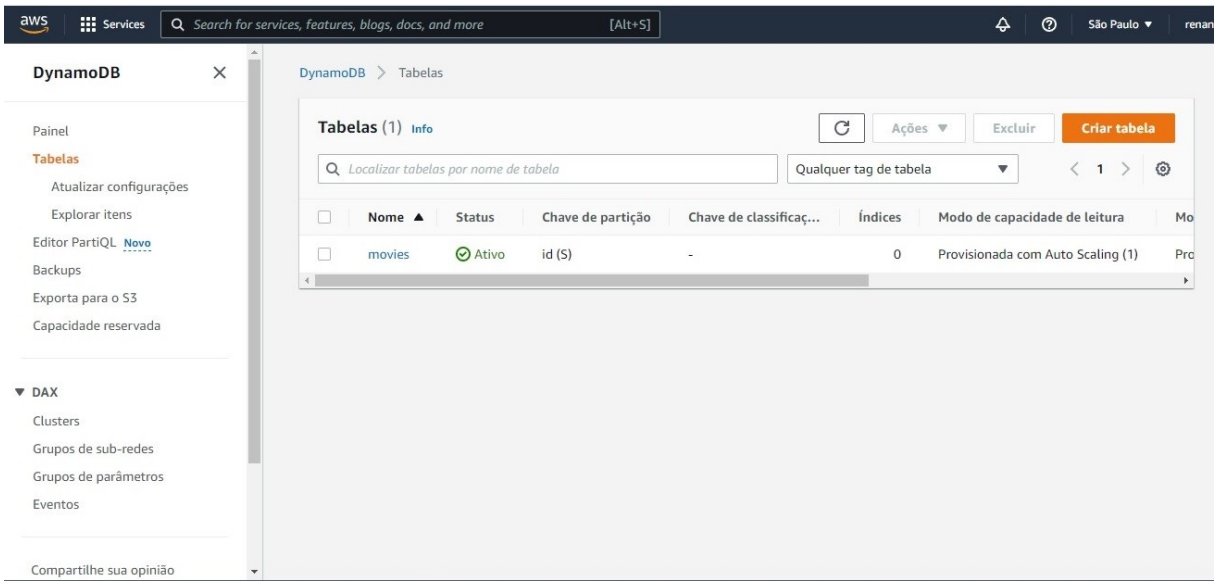
Figura 16 – Captura de tela Cloudtrail plataforma AWS



Fonte: Elaborado pelo autor.

A figura 16 apresenta o Cloudtrail, onde são armazenados os logs dos processos realizados em todo o ambiente.

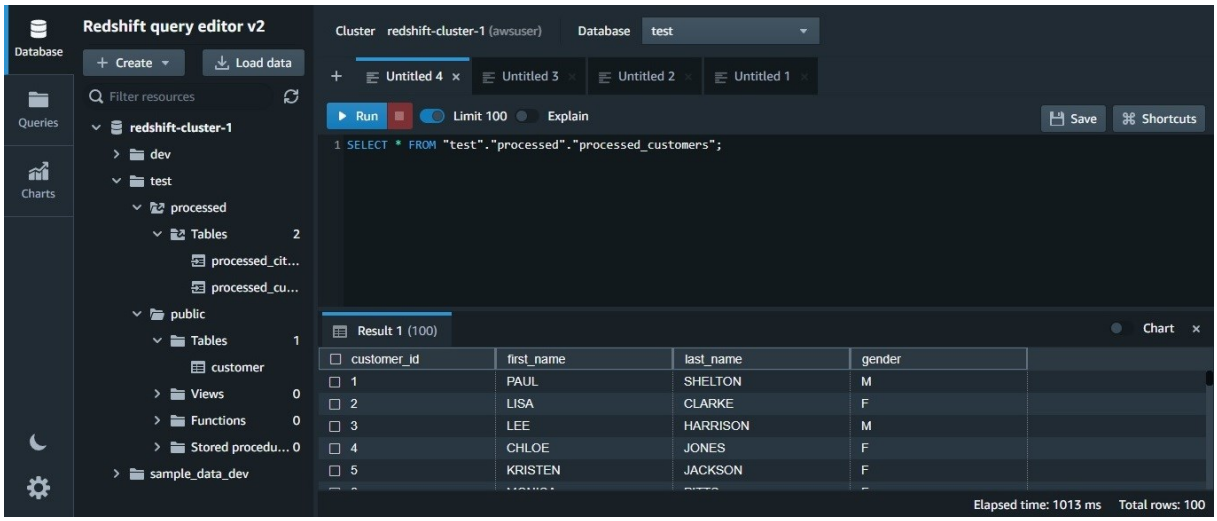
Figura 17 – Captura de tela DynamoDB plataforma AWS



Fonte: Elaborado pelo autor.

A figura 17 mostra o DynamoDB, onde são armazenados os dados não-relacionais do ambiente.

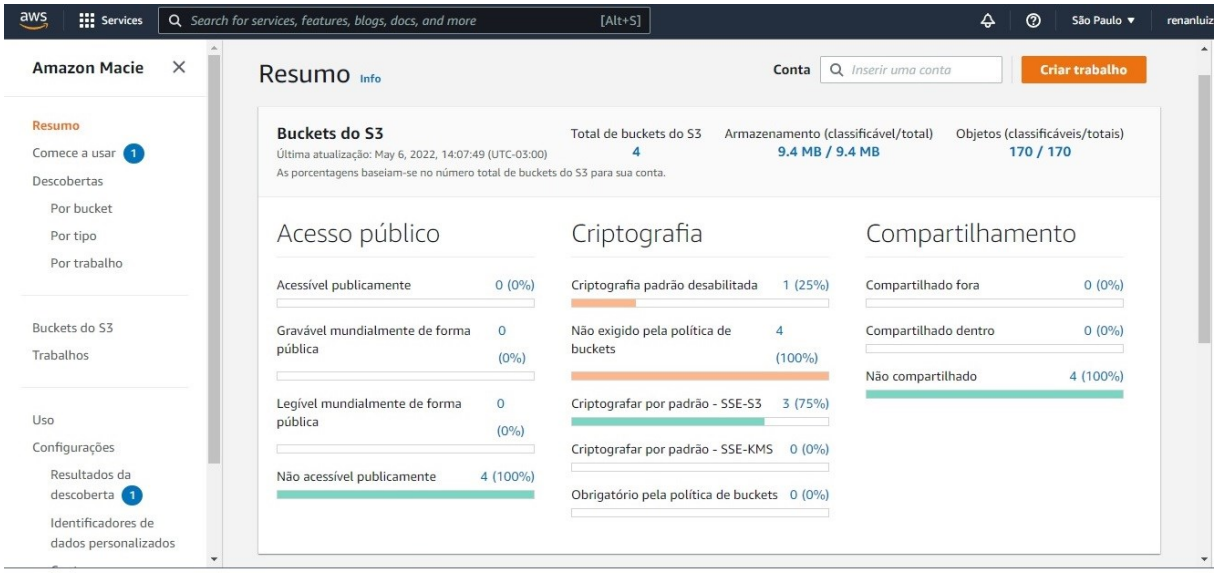
Figura 18 – Captura de tela Redshift plataforma AWS



Fonte: Elaborado pelo autor.

A figura 18 apresenta o Redshift, local do ambiente onde é possível consultar os dados prontos para consumo do Warehouse ou para os reports. Aqui é criado um cluster para que os dados estejam disponíveis de forma otimizada.

Figura 19 – Captura de tela Macie plataforma AWS



Fonte: Elaborado pelo autor.

A figura 19 apresenta o Amazon Macie, ferramenta que disponibiliza uma análise de dados sensíveis, dados comuns como endereços, telefones, documentos podem ser detectados de forma automática ou podem ser configurados conforme a necessidade do negócio.

5 CONSIDERAÇÕES FINAIS

Este trabalho tem como objetivo final proporcional um ambiente AWS para armazenamento e processamento de dados, no momento, termos como Business Intelligence, Ciências de Dados, Engenharia de Dados, “dados é o novo ouro”, entre outros termos estão sendo muito ouvidos em pautas de tecnologia, mas por ser um campo de estudo novo, ainda não temos todas as informações muito claras e de fácil acesso para poder realizar o desenvolvimento de um ambiente. Esse projeto veio como uma grande ferramenta para me desenvolver como engenheiro de dados e conseguir colocar a mão na massa utilizando as diversas ferramentas disponibilizadas pela AWS.

O ambiente se mostrou intuitivo para a criação dos ambientes e os buckets para o armazenamento, a aplicação de transformações também é prática tanto utilizando as ferramentas gráficas, quanto aplicando os códigos utilizando linguagens de programação (Python e PySpark), também existem várias opções para acompanhar o andamento dos processos e logs, além de fácil customização atendendo a necessidade do usuário.

Os principais desafios do ambiente é a aplicação das políticas para os grupos de usuários que terão privilégios de acesso para leitura, escrita e consultas dos dados, nessa configuração é possível realizar a aplicação da LGPD ou outras políticas de tratativas de dados. Outro desafio é o controle dos gastos que do ambiente AWS, existem alguns avisos durante o desenvolvimento referente aos possíveis gastos envolvidos, porém não é muito clara as questões dos valores de cobrança.

Com relação ao ambiente se mostrou funcional e pode ser utilizado para realizar o ETL em diversas fontes de dados e disponibilizá-los prontos para o consumo em diversas APIs, criação de dashboards e reports ou, até mesmo, a aplicação de Machine Learning.

6 REFERÊNCIAS

AMAZON ATHENA. Disponível em: https://aws.amazon.com/pt/athena/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc. Acesso em 27/04/2022.

AMAZON DYNAMODB. Disponível em: https://docs.aws.amazon.com/pt_br/amazondynamodb/latest/developerguide/Introduction.html. Acesso em 30/04/2022.

AMAZON KINESIS. Disponível em: https://aws.amazon.com/pt/kinesis/. Acesso em 27/04/2022.

AMAZON MACIE. Disponível em: https://aws.amazon.com/pt/macie/. Acesso em 03/05/2022.

AMAZON RDS. Disponível em: https://aws.amazon.com/pt/rds/. Acesso em 27/04/2022.

AMAZON REDSHIFT. Disponível em: https://aws.amazon.com/pt/redshift/. Acesso em 30/04/2022.

AMAZON S3. Disponível em: https://aws.amazon.com/pt/s3/. Acesso em 27/04/2022.

AMAZON VPC. Disponível em: https://docs.aws.amazon.com/pt_br/toolkit-for-visual-studio/latest/user-guide/vpc-tkv.html. Acesso em 27/04/2022.

AMAZON WEB SERIES. Disponível em: https://pt.wikipedia.org/wiki/Amazon_Web_Services. Acesso em 23/04/2022.

AWS CLOUDTRAIL. Disponível em: https://docs.aws.amazon.com/pt_br/awscloudtrail/latest/userguide/cloudtrail-user-guide.html. Acesso em 23/04/2022.

AWS GLUE. Disponível em: https://docs.aws.amazon.com/pt_br/glue/latest/dg/what-is-glue.html. Acesso em 27/04/2022.

AWS GLUEDATABREW. Disponível em: https://docs.aws.amazon.com/databrew/latest/dg/what-is.html. Acesso em 27/04/2022.

AWS IAM. Disponível em: https://docs.aws.amazon.com/pt_br/IAM/latest/UserGuide/introduction.html. Acesso em 27/04/2022.

AWS LAKE FORMATION. Disponível em: https://aws.amazon.com/pt/lake-formation/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc. Acesso em 27/04/2022.

GOOGLE CLOUD PLATFORM. Disponível em: https://pt.wikipedia.org/wiki/Google_Cloud_Platform. Acesso em 23/04/2022.

MICROSOFT AZURE. Disponível em: https://pt.wikipedia.org/wiki/Microsoft_Azure. Acesso em 23/04/2022.

PINHEIRO, Álvaro Farias. **Fundamento da Engenharia de Software:** Conceitos Básicos: 1. ed. – Volume I. Recife: Publicação Independente, 2015.

PYSPARK. Disponível em: https://spark.apache.org/docs/latest/api/python/. Acesso em 05/05/2022.

PYTHON. Disponível em: https://www.python.org/doc/essays/blurb/. Acesso em 05/05/2022.