

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIAS E TECNOLOGIA DE
RONDÔNIA – CAMPUS JI-PARANÁ
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

**VEX BACKUP – SISTEMA DE BACKUP DE BANCO DE DADOS
MYSQL COM INTEGRAÇÃO À NUVEM**

WILLIANS GOMES NUNES

Ji-Paraná – RO

2025

WILLIANS GOMES NUNES

**VEX BACKUP – SISTEMA DE BACKUP DE BANCO DE DADOS
MYSQL COM INTEGRAÇÃO À NUVEM**

Monografia apresentada ao Instituto Federal de Rondônia - Campus Ji-Paraná, como requisito para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Área de Concentração: Ciências Exatas e da Terra.

Orientador (a): Prof. Dr. Jackson Henrique da Silva Bezerra.

Ji-Paraná – RO

2025

FICHA CATALOGRÁFICA

Ficha catalográfica elaborada pelo Sistema Gerador de Ficha Catalográfica do IFRO.

Nunes, Willians Gomes.

Vex Backup – Sistema de backup de banco de dados MySQL com Integração à nuvem / Willians Gomes Nunes. - Ji-Paraná, 2025.
71 f.

Orientador(a): Dr. Jackson Henrique da Silva Bezerra.

Trabalho de Conclusão de Curso (Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO, Ji-Paraná, 2025.

1. Backup. 2. MySQL. 3. Automação. 4. Nuvem. I. Bezerra, Jackson Henrique da Silva (orient.). II. Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO. III. Título.

Bibliotecário(a) Responsável: Cleuza Diogo Antunes, CRB-11/864

BANCA EXAMINADORA

Prof. Dr. Wanderson Roger Azevedo Dias

Avaliador nº 1

Prof. Me. Gleison Guardia

Avaliador nº 2

Prof. Dr. Jackson Henrique da Silva Bezerra

Presidente da Banca Examinadora

Data da Defesa: 20 de agosto de 2025

Nota Final: 98

AGRADECIMENTOS

Agradeço à minha mãe, que sempre esteve ao meu lado me dando apoio e acreditando em mim mesmo quando eu duvidava. Sua presença foi essencial em cada etapa desta caminhada. Agradeço também aos meus amigos, que torceram por mim, me incentivaram e estiveram presentes nos momentos em que mais precisei.

RESUMO

A crescente necessidade de confiabilidade e desempenho em rotinas de backup tem motivado o desenvolvimento de ferramentas que automatizam esse processo, reduzem falhas humanas e oferecem visibilidade sobre as operações realizadas. Este trabalho apresenta o desenvolvimento de uma solução de backup automatizado para bancos de dados MySQL, composta por dois módulos. O primeiro é uma aplicação responsável pela execução do backup, abrangendo as etapas de dump, compactação e envio para um serviço de armazenamento em nuvem. O segundo é uma ferramenta web que permite o acompanhamento em tempo real das execuções, a análise de métricas e o gerenciamento de usuários. Para o desenvolvimento da solução, foi adotada a metodologia ágil Scrum, com ciclos de desenvolvimento curtos e iterativos. Já para a avaliação da ferramenta, utilizou-se uma abordagem de estudo de caso com métodos quantitativos, baseada em testes controlados. Foram realizados testes práticos com bases de dados de 1 GB, 5 GB e 10 GB, com cinco execuções por cenário. Foram coletados os tempos de cada etapa da operação e do processo total, possibilitando a análise estatística por meio da média e do desvio padrão. Cada cenário foi executado cinco vezes, com medições dos tempos de dump, compactação, upload para a nuvem, além do cálculo da média e desvio padrão de cada etapa. A ferramenta foi implantada em ambiente Linux com banco de dados MySQL e integração com serviços de armazenamento em nuvem, utilizando automação em Python e monitoramento via interface web. Os resultados demonstraram que a ferramenta manteve estabilidade frente ao aumento do volume de dados e apresentou desempenho satisfatório em todas as etapas. A interface web ofereceu uma experiência de uso intuitiva e funcional, com painel de visualização dos backups do dia, métricas por hora, minuto e dia, histórico completo de execuções e tela de gerenciamento de usuários. Conclui-se que a solução desenvolvida é viável, eficiente e aplicável a ambientes reais que demandam rotinas de backup seguras, automatizadas e monitoradas de forma centralizada.

Palavras-chave: Backup, MySQL, automação, nuvem.

ABSTRACT

The growing need for reliability and performance in backup routines has driven the development of tools that automate this process, reduce human error, and provide visibility into operations. This work presents the development of an automated backup solution for MySQL databases, composed of two modules. The first is an application responsible for executing the backup, covering the steps of dumping, compression, and uploading to a cloud storage service. The second is a web tool that allows real-time monitoring of executions, metric analysis, and user management. For the development of the solution, the agile Scrum methodology was adopted, with short and iterative development cycles. For the evaluation of the tool, a case study approach with quantitative methods was used, based on controlled tests. Practical tests were conducted with databases of 1 GB, 5 GB, and 10 GB, with five executions per scenario. The execution times for each step and the total process were recorded, enabling statistical analysis through averages and standard deviations. Each scenario was executed five times, with measurements of dump, compression, and cloud upload times, as well as the calculation of the mean and standard deviation for each stage. The tool was deployed in a Linux environment with a MySQL database and integrated with cloud storage services, using Python automation and web interface monitoring. The results showed that the tool maintained stability with increasing data volume and demonstrated satisfactory performance in all stages. The web interface provided an intuitive and functional user experience, with a dashboard displaying daily backups, metrics by hour, minute, and day, a complete execution history, and a user management screen. It is concluded that the developed solution is feasible, efficient, and applicable to real-world environments that require secure, automated, and centrally monitored backup routines.

Keywords: Backup, MySQL, automation, cloud.

LISTA DE FIGURAS

Figura 1 - Fluxograma da ferramenta de backup.....	25
Figura 2 - Tela de Login.....	28
Figura 3 - Dashboard com métricas dos backups.....	29
Figura 4 - Listagem completa de backups.....	29
Figura 5 - Detalhes técnicos da execução de backup.....	30
Figura 6 - Interface de gerenciamento de usuários.....	30

LISTA DE TABELAS

Tabela 1 - Comparação processo antigo x novo	25
Tabela 2 - Tempos Médios por Etapa e Total.....	27
Tabela 3 - Desvios Padrão por Etapa e Total	27

LISTA DE ABREVIATURAS E SIGLAS

AWS	<i>Amazon Web Service</i>
CSS	<i>Cascading Style Sheets</i>
GB	<i>Gigabyte</i>
HTML	Linguagem de Marcação de Hipertexto
IDC	<i>International data corporation</i>
LVM	<i>Logical Volume Manager</i>
MHz	<i>Megahertz</i>
ORM	<i>Object-Relational Mapping</i>
RAM	<i>Random Access Memory</i>
RPO	<i>Recovery Point Objective</i>
RTO	<i>Recovery Time Objective</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
SMTP	<i>Simple Mail Transfer Protocol</i>
SQL	<i>Structured query language</i>
TI	Tecnologia da Informação

SUMÁRIO

1 INTRODUÇÃO	12
1.1 PROBLEMÁTICA	12
1.2 JUSTIFICATIVA	13
1.3 OBJETIVOS	14
1.3.1 Tipos de backup.....	14
1.3.2 Objetivos específicos	14
1.4 ESTRUTURA DO TRABALHO	14
2 REFERENCIAL TEÓRICO	15
2.1 BANCO DE DADOS	15
2.2 SISTEMAS DE BACKUP	16
2.2.1 Tipos de backup.....	17
2.3 FERRAMENTAS DE BACKUP	18
2.4 COMPUTAÇÃO EM NUVEM	18
2.5 AUTOMAÇÃO.....	19
3 MEDODOLOGIA	21
3.1 METODOLOGIA DE PESQUISA.....	21
3.2 METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE	22
4 RESULTADOS E DISCUSSÕES.....	24
4.1 PROCESSO DE BACKUP DESENVOLVIDO.....	24
4.2 COMPARATIVO COM O PROCESSO TRADICIONAL.....	25
4.3 TESTES FERRAMENTA DE BACKUP	26
4.4 FERRAMENTA WEB.....	27
5 CONCLUSÃO.....	31
REFERÊNCIAS	32
APÊNDICE	35

1 INTRODUÇÃO

Com o avanço tecnológico em diferentes áreas, aliado às diferentes fontes e formatos de dados, surge o fenômeno frequentemente chamado de *Big Data*, que se caracteriza pelos chamados “cinco Vs”: Volume, Velocidade, Variedade, Veracidade e Valor. De acordo com as projeções da *International Data Corporation* (IDC, 2021), a quantidade de dados gerados globalmente ultrapassará 157 ZB (*zetabytes*) em 2024, com expectativa de dobrar até 2027.

Um dos conceitos mais presentes na atualidade e que teve origem há muitos anos é o conceito de automação. Segundo o *Oxford Learner’s Dictionaries*, a automação é “o uso de máquinas e computadores para fazer o trabalho que anteriormente era feito por pessoas” (*Automation*, 2025, tradução nossa)¹. Com a automação, foi possível realizar inúmeras contribuições para a melhoria da vida humana. Isso pode ser observado desde a Revolução Industrial, com a invenção da máquina a vapor, e inclui a “progressiva automatização de setores inteiros da economia e da vida social com base na comunicação máquina-máquina” (MAGRANI, 2018, p. 15).

Nesse contexto, a adoção de rotinas de *backup* robustas torna-se essencial para minimizar riscos de perda de dados em incidentes decorrentes de falhas de hardware ou serviços (ou ambos) ou ainda de desastres naturais, e, assim, possibilitar a continuidade do negócio sem maiores encargos. Além disso, as organizações buscam cada vez mais escalabilidade e eficiência no gerenciamento de suas bases de dados, o que leva à procura de métodos automatizados de *backup* e serviços de armazenamento (em inglês, *storage*) em nuvem.

1.1 PROBLEMÁTICA

Apesar de existirem inúmeros métodos de *backup*, a ausência de processos que combinem a execução de *backup* lógicos (*dump*), a compressão dos arquivos para redução do espaço ocupado em disco e o armazenamento seguro e escalável em nuvem de forma automática, pode levar à perda irreversível de bases de dados inteiras em casos de falhas ou incidentes. Ademais, a falta de um *dashboard* que viabilize, em tempo real, o acompanhamento do andamento dos *backups* e o registro histórico das operações representa outro obstáculo para as organizações.

¹ No original: “the use of machines and computers to do work that was previously done by people”.

Diante disso, surge o questionamento de como implementar um sistema capaz de performar de forma integrada o *backup* lógico de bancos de dados MySQL, a compactação e o envio dos arquivos de *backup* a um serviço de armazenamento em nuvem, e permitir a visualização transparente de todo esse processo através de um *dashboard*. A adoção de soluções automatizadas e escaláveis, capazes de simplificar a gestão de recursos de TI e garantir a proteção de dados em cenários de alto crescimento de informações, é essencial para viabilizar ambientes de big data (HURWITZ *et al.*, 2013).

Embora existam soluções de backup mais gerais no mercado, como o *Veeam Backup & Replication*, amplamente utilizado em ambientes empresariais para a proteção de máquinas virtuais e que também oferece suporte a backups de banco de dados MySQL e a ferramenta *Bacula*, que fornece recursos de agendamento e monitoramento de backups, ambas carecem de um dashboard simplificado e em tempo real voltado especificamente para bancos de dados MySQL, agregando compressão, transferência automática para a nuvem e visualização dos resultados em um único sistema. Além disso, essas soluções recorrem à ferramenta *mysqldump*, integrada nativamente ao MySQL, que é lenta para grandes volumes de dados e não oferece compressão nem envio automático para serviços em nuvem.

1.2 JUSTIFICATIVA

De acordo com DB-Engines (2025), o SGBD (Sistema de Gerenciamento de Banco de Dados) o MySQL é classificado com o segundo banco de dados relacional mais utilizado no mundo, fato que demonstra sua importância para diversas organizações que necessitam de soluções confiáveis para a proteção e recuperação de dados. No entanto, apesar dessa popularidade, ainda há carência de ferramentas que ofereçam, em uma única solução, recursos de backup lógico, compressão de arquivos para otimizar custos de armazenamento e envio automático e seguro de dados para a nuvem. A ausência de processos unificados que integrem essas etapas de forma coesa e monitorada eleva o risco de perdas irreversíveis em cenários de falhas ou incidentes.

Soluções como Veeam Backup & Replication ou o Bacula, demonstram avanços na execução de backups e na proteção de ambientes virtuais, mas não softwares de propósito geral, não sendo voltadas para o MySQL e não contam com ferramentas de backup otimizadas como o MyDumper para performar backups lógicos, além de não permitirem compressão nativa e automação de envio para serviços de armazenamento em nuvem. Nesse contexto, torna-se

necessário desenvolver uma solução que supra essas lacunas, agregando praticidade, escalabilidade e transparência na gestão de dados.

1.3 OBJETIVOS

1.3.1 Tipos de backup

Desenvolver uma solução integrada de backup para bancos de dados MySQL, capaz de automatizar rotinas de backup lógico, armazenar os arquivos de forma segura em um serviço de nuvem e disponibilizar, por meio de um dashboard, a visualização dos processos executados.

1.3.2 Objetivos específicos

- I. Desenvolver um processo para gerenciamento e gestão de backup lógico de banco de dados MySQL.
- II. Desenvolver uma ferramenta web do tipo dashboard para gestão do processo de backup.
- III. Realizar uma revisão de literatura acerca de processos de backup lógico no MySQL.

1.4 ESTRUTURA DO TRABALHO

Este trabalho foi estruturado da seguinte forma: no Capítulo 1, foi apresentada a introdução ao tema, destacando sua relevância prática, a justificativa para o desenvolvimento da ferramenta e os objetivos geral e específicos do projeto. O Capítulo 2 contempla o referencial teórico, abordando os principais conceitos relacionados a sistemas de backup, banco de dados MySQL, armazenamento em nuvem e práticas de automação, fundamentais para embasar tecnicamente o desenvolvimento da solução. No Capítulo 3, é descrita a metodologia adotada, incluindo a abordagem de pesquisa e os processos técnicos envolvidos na implementação do sistema. O Capítulo 4 apresenta os resultados obtidos com a ferramenta, tanto na interface web desenvolvida quanto na análise de desempenho dos backups realizados em diferentes cenários, por meio de testes controlados e análise estatística. Por fim, o Capítulo 5 traz as considerações finais, discutindo as observações extraídas ao longo do trabalho.

2 REFERENCIAL TEÓRICO

2.1 BANCO DE DADOS

A Lei Geral de Proteção de Dados (Lei nº 13.709/2018) define banco de dados como um "conjunto estruturado de dados pessoais, estabelecido em um ou em vários locais, em suporte eletrônico ou físico" (Brasil, 2018, art. 5º, inciso IV). Trata-se de qualquer organização sistemática de informações pessoais, sendo essencial para a realização de operações de tratamento de dados, como coleta, armazenamento, compartilhamento e eliminação.

Para que um banco de dados seja gerenciado de maneira eficiente e segura, faz-se necessário o uso de um Sistema Gerenciador de Banco de Dados (SGBD), o qual consiste em um software que facilita os processos de definição, construção, manipulação e compartilhamento de bancos de dados entre diversos usuários e aplicações, garantindo controle e eficiência nas operações (ELMASRI e NAVATHE, 2015). Já para Silberschatz, Korth e Sudarshan (2020), esse sistema consiste em um conjunto de programas que permitem aos usuários criar e manter um banco de dados, além de controlar as operações de acesso, recuperação e modificação de dados.

Para a modelagem de um banco de dados existem os modelos conceitual, lógico e físico. No nível lógico, temos as estruturas do banco de dados conforme observadas pelo usuário do SGBD (HEUSER 2009, p. 6). Quanto ao modelo conceitual, de acordo com Heuser (2009, p.16) "o modelo conceitual registra que dados podem aparecer no banco de dados, mas não registra como estes dados estão armazenados a nível de SGBD. Heuser (2009, p. 17), ao abordar o modelo lógico, afirma que este "[...] é uma descrição de um banco de dados no nível de abstração visto pelo usuário do SGBD. Assim, o modelo lógico é dependente do tipo particular de SGBD que está sendo usado".

Há diversos SGBDs disponíveis no mercado, divididos em várias categorias, cada um destinado a um uso distinto. Para Silberschatz, Korth e Sudarshan (2020), os sistemas de gerenciamento de banco de dados são frequentemente organizados de acordo com o modelo de dados em que se baseiam. Podem ser relacionais, orientados a objetos, orientados a documentos, baseados em grafos, entre outros, cada qual com características e aplicações específicas.

Com o auxílio dos SGBDs - ou em inglês Data Base Management System (DBSM)-, é possível persistirmos os dados que geramos todos os dias. Um SGBD possibilita:

[...] aos usuários criar e manter um banco de dados. O DBMS é um sistema de software de propósito geral que facilita os processos de definição, construção, manipulação e

compartilhamento de bancos de dados entre vários usuários e aplicativos (Elmasri; Navathe, 2015, p. 6, tradução nossa)².

No mercado, há diversos Sistemas de Gerenciamento de Bancos de Dados (SGBDs), cada um com funcionalidades e nível de maturidade distintos. Entre os principais SGBDs relacionais estão, o Oracle Database que é amplamente utilizado em grandes corporações por ser robusto e de recursos avançados; Microsoft SQL Server que se destaca pelas integrações com o sistema operacional Windows e por possuir ferramentas de análise de dados; o MySQL, muito adotado em aplicações web devido ao desempenho e à simplicidade de uso; e o PostgreSQL, um SGBD de código aberto que se diferencia pelo sólido suporte aos padrões SQL e pelo suporte ativo da comunidade.

2.2 SISTEMAS DE BACKUP

Segundo Hey *et al.*, 2009, o grande volume de dados gerados diariamente por meio de instrumentos e modelos computacionais inclina-se a possuir longevidade e permanecer publicamente acessível, de modo a possibilitar análises contínuas. Isso posto, para que possamos assegurar que esses dados possuirão essa longevidade, é necessário a utilização de sistemas de backups para garantir sua segurança.

Conforme Silva, Neto e Almeida (2015, p. 2), os sistemas de backup têm a missão principal de proteger informações e garantir a recuperação eficaz em casos de desastres. Segundo os autores:

sistemas de *backup/recovery* são implementados primordialmente para salvaguardar informações e fornecer uma recuperação efetiva em caso de desastres. A implementação de sistema de *backup/recovery* resulta em um baixo impacto na continuidade dos negócios.

Além disso, os autores destacam que o backup tem o propósito de armazenar informações críticas de uma instituição ou empresa, especialmente aquelas cujos processos de negócio são baseados em Tecnologia da Informação (TI). Para isso, a cópia fiel dos dados deve ser criada e retida conforme estratégias previamente definidas, garantindo a recuperação em situações de perda ou desastre (Silva; Neto; Almeida, 2015).

Dessa forma, fica claro que a longevidade e acessibilidade contínua dos dados está diretamente associada aos sistemas de backups, já que esses sistemas são fundamentais para

² No original: [...] is a computerized system that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications.

assegurar a continuidade operacional e a minimização de perdas em cenários de falha ou desastres. De acordo com a pesquisa realizada pela empresa especializada em ambientes de infraestrutura de TI Active Solutions (2017), constatou-se que:

[...] 44% das perdas de dados ocorreram por ações sem intenção, como apagar arquivos ou partições erradas, má colocação de cartões de memória ou ataque por vírus. Já 32% ocorrem por ações intencionais (como a formatação do disco rígido) e 21% por falhas (no software, HD ou energia, além de corrupção dos sistemas de arquivo do banco de dados).

De acordo com Jesus e Schimiguel (2018), possuir uma rotina de backup definida é uma estratégia de negócio, já que mesmo que hajam que perdas de dados causados por quaisquer tipos de incidentes, com um backup é possível dar continuidade nas atividades da empresa com impactos mínimos.

2.2.1 Tipos de backup

Existem diversos tipos de backup, cada um mais adequado a diferentes cenários e organizações. Portanto, cabe a cada instituição analisar qual método melhor atende às suas necessidades. Uma forma de mensurar e orientar a escolha do método de backup é por meio das métricas *Recovery Time Objective* (RTO) e *Recovery Point Objective* (RPO). O RTO representa o período máximo de indisponibilidade aceitável de um processo de negócio baseado em TI, enquanto o RPO indica o volume máximo de dados que podem ser perdidos sem causar impactos prejudiciais à instituição (ORACLE, 2020).

O backup total (*full backup*) é caracterizado pela cópia integral de todos os dados especificados em um determinado instante. Esse procedimento apresenta duas desvantagens principais: a operação de leitura e escrita de todo o sistema é mais lenta, e o armazenamento de uma cópia completa consome significativa parte da mídia de *backup* (CHERVENAK; VELLANKI; KURMAS, 1998).

O backup incremental (*incremental backup*) cópia apenas os dados modificados desde o último backup, resultando em backups menores e mais rápidos. Sua principal vantagem está na redução do volume de arquivos armazenados, já que, em geral, apenas uma pequena fração dos dados é alterada em um dia (CHERVENAK; VELLANKI; KURMAS, 1998).

Conforme Schwartz, Zaitsev e Tkachenko (2012), para o SGBD MySQL, existem duas vertentes mais comuns de formas de backup, sendo os backups do tipo Lógico (*dump*), que apresentam os dados em um formato que o MySQL interpreta como um texto com delimitadores

ou um comando SQL, e os backups do tipo físico (*Raw*) consistem em um formato de backup em que os arquivos físicos a nível de sistema operacional são copiados.

2.3 FERRAMENTAS DE BACKUP

Existem diversas ferramentas de backup disponíveis no mercado, algumas excelentes e outras não tão relevantes. Algumas das ferramentas mais conhecidas são: *mylvmbackup*, que foi desenhado para realizar backups do tipo snapshot em *Logical Volume Manager* (LVM); *Percona XtraBackup*, uma opção de ferramenta *open source*; e *MySQL Enterprise Backup*, uma solução proprietária da Oracle destinada ao SGBD MySQL (SCHWARTZ; ZAITSEV; TKACHENKO, 2012).

O *MySQL Enterprise Backup* é a ferramenta ofertada através da subscrição *Enterprise* do MySQL pela empresa Oracle. Esta ferramenta de backup não necessita interromper o funcionamento do serviço do banco de dados do MySQL, através dessa ferramenta é possível realizar backups incrementais, backups completos (SCHWARTZ, ZAITSEV E TKACHENKO, 2012).

A ferramenta *mysqldump* é um dos métodos de backup mais comuns no ecossistema do MySQL, pois acompanha o instalador do SGBD e permite a realização de backups lógicos de forma simples. De acordo com a documentação oficial, o *mysqldump* é um utilitário (ferramenta) que realiza backups do tipo lógico, que “[...] produz um conjunto de instruções SQL que podem ser executadas para reproduzir as definições originais dos objetos de banco de dados e os dados das tabelas” (MySQL, 2025, tradução nossa)³.

O MyDumper é uma ferramenta *open source* que foi desenvolvida através de engenheiros responsáveis pelo MySQL como um substituto à ferramenta de backup *mysqldump*. De acordo com a página do MyDumper no GitHub, essa ferramenta de backup é destinada a realização de backups do SGBD MySQL, possibilitando backups consistentes e permitindo paralelismo para ampliar a performance e a velocidade dos backups, além de permitir operações de backups individuais por objetos de banco de dados, isto é, backups por *database*, *table*, *view*, *event*, *procedure* ou *trigger*.

2.4 COMPUTAÇÃO EM NUVEM

³ No original: [...] *producing a set of SQL statements that can be executed to reproduce the original database object definitions and table data.*

Podemos definir a computação em nuvem (*cloud computing*), como um modelo de fornecimento de recursos voltados para a TI, como servidores, aplicações, banco de dados e armazenamento de forma escalável e sob demanda, disponibilizados através da internet (MELL; GRANCE, 2011).

No contexto de cópias de segurança (*backups*), os serviços em nuvem proporcionam uma proteção adicional contra desastres ou falhas locais de *hardware* e/ou *software*, uma vez que mantém os dados em um ambiente externo e em muitos casos armazena cópias adicionais do mesmo arquivo de forma distribuída geograficamente. Armbrust *et al.* (2010) destacam que a elasticidade intrínseca ao *cloud computing* e o pagamento baseado no consumo facilitam a adoção de soluções de backup para grandes volumes de dados, sem a necessidade de investimentos iniciais consideráveis em infraestrutura de hardware.

Ao utilizar serviços de armazenamento em nuvem para manter cópias de segurança (*backups*), possibilita que em cenários de ataques, falhas ou desastres naturais, a recuperação dos dados seja acelerada ao ter acesso a cópias de segurança em localizações distintas. Segundo Botta *et al.* (2016), ao combinar a escalabilidade da nuvem com processos eficientes de upload e download, é possível restaurar dados críticos de maneira ágil, minimizando paradas de serviço e prejuízos operacionais.

A flexibilidade ofertada pelos serviços de *cloud computing* faz com que a adoção de estratégias de backup que façam uso de plataformas de armazenamento em nuvem sejam cada vez mais atrativas, principalmente em cenários em que o volume de dados cresce de forma imprevisível. Segundo Jouini e Rabai (2014), a capacidade elástica de armazenamento em nuvem contribui para que empresas ajustem seus recursos de backup conforme a demanda, evitando investimentos iniciais onerosos em infraestrutura de data centers.

Ademais, ao utilizar serviços disponibilizados por *cloud computing* permite a redundância geográfica, isto é, possibilita guardar cópias dos dados em múltiplos locais distribuídos ao redor do mundo. De acordo com Subashini e Kavitha (2011), essa prática não apenas melhora a resiliência contra desastres naturais ou falhas regionais, mas também garante que, em situações extremas, haja uma cópia atualizada dos backups em um local remoto, pronta para ser restaurada.

2.5 AUTOMAÇÃO

A automação pode ser definida como o uso de tecnologias, sistemas ou dispositivos que executam tarefas de forma autônoma ou que necessitam de mínima interferência humana

(GROOVER, 2015). A automação possui como principais objetivos o aumento da eficiência, redução de custos e minimização de erros.

Um dos marcos históricos do uso da automação ocorreu há muitos anos atrás, no início do século XX, com o advento da implantação da linha de produção (ou montagem) idealizada pelo estadunidense Henry Ford. Em sua obra, Ford (1926) enfatiza a importância de um fluxo de trabalho contínuo para reduzir o tempo de fabricação e baratear custos de produção.

A partir desse legado, a automação ganhou um novo rumo, se expandindo para além das esteiras de montagem e passando a incorporar áreas como: sistemas de controle, robótica, softwares e sensores. Na década de 1960 iniciou o surgimento dos primeiros robôs capazes de executar tarefas repetitivas com precisão dentro das indústrias (GROOVER, 2015).

Independente do campo de aplicação, a automação parte do mesmo princípio, a busca por otimizar processos, tornando-os mais eficientes, confiáveis e rentáveis. Em áreas de saúde e farmacêutica, sistemas automatizados de checagem de medicamentos e diagnósticos podem elevar significativamente a segurança do paciente (POON *et al.*, 2010).

3 METODOLOGIA

3.1 METODOLOGIA DE PESQUISA

Segundo Pereira et al. (2018, p. 65), “um estudo de caso é uma descrição e análise, tão detalhada quanto possível, de um caso que apresenta particularidades que o tornam especial”. Essa abordagem é reconhecida por sua contribuição significativa à área do conhecimento em que é aplicada, devido à riqueza de dados e informações úteis para a resolução da problemática.

Diante disso, a adoção da metodologia de estudo de caso neste projeto é fundamental, pois permitiu uma análise detalhada do desempenho dos processos de backup e upload de arquivos em um ambiente de banco de dados baseado no SGBD MySQL. A referida escolha possibilitou compreender de forma detalhada o fenômeno, trazendo à luz a influência do tamanho do backup no tempo de envio para o armazenamento em nuvem, bem como a estabilidade do desempenho da ferramenta de backup desenvolvida.

Para a coleta dos dados, foram realizados testes repetidos em três cenários distintos, nos quais o tamanho do banco de dados variou entre 1, 5 e 10 gigabytes (GB). Em cada um dos cenários, foram executados cinco backups, registrando três parâmetros principais: (i) o tempo de execução do dump (início do processo de exportação até sua conclusão), (ii) o tempo de compactação (desde o início da compressão do arquivo de dump até o seu término) e (iii) o tempo de upload (compreendendo o início do envio até a finalização no serviço de nuvem).

Além da análise individual de cada etapa, foi também considerado o tempo total de execução do backup, definido como a soma do tempo de dump e da compactação. Após a coleta dos dados, estes foram analisados estatisticamente, por meio do cálculo da média dos tempos e do desvio padrão, de modo a avaliar a estabilidade, a consistência do processo e a escalabilidade da ferramenta de backup diante do aumento progressivo no volume de dados.

Utilizando esse método, foi possível identificar o comportamento da influência no desempenho conforme tamanho do backup, bem como checar possíveis inconsistências relacionados a realização dos backups e ao processo de envio dos arquivos à nuvem.

Para a realização dos testes e coleta de dados, foi utilizada uma máquina com a seguinte configuração: sistema operacional Oracle Linux 9.5 e Kernel na versão 5.15.0-305.176.4; processador Intel Core i5-12500, com 6 núcleos e 2 threads ativas por núcleo (Hyper Threading); frequência máxima do processador em 4.6 Gigahertz (GHz); 16 GB de Random Access Memory (RAM) com 3200 Megatransfer por segundo (MT/s) de velocidade. Já para o

banco de dados, foi utilizado o MySQL na versão 9.1.0 Commercial e o armazenamento em nuvem AWS contará com a versão S3 Standard.

3.2 METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE

Para o desenvolvimento deste projeto, foi adotada a metodologia ágil de desenvolvimento de software Scrum, reconhecida por gerir projetos complexos de maneira iterativa e adaptativa. “O desenvolvimento incremental envolve a construção de um sistema pedaço a pedaço. Primeiro uma parte é desenvolvida, depois uma próxima parte é adicionada á primeiro e assim por diante” (COHN, 2011). Essa abordagem, além de promover a comunicação e a transparência, facilita o acompanhamento do andamento do projeto por meio de ciclos curtos, as Sprints, que possibilitam a definição de tarefas e prazos, que variam entre duas a quatro semanas de duração. Para este projeto este prazo será adaptado para a duração de 15 dias de duração para cada uma das Sprints.

O Scrum pode ser aplicado a diversos modelos de projetos, não ficando restrito exclusivamente à implementação de software. Com o Scrum é possível assegurar que todas as etapas sejam acompanhadas de forma visível e colaborativa. Essa estrutura é composta por artefatos essenciais, como:

- *Product Backlog*: Lista priorizada de todas as funcionalidades, tarefas e melhorias a serem implementadas.
- *Sprint Planning*: Antes do início de cada sprint é realizada uma reunião para definir quais atividades presentes no *Product backlog* serão trabalhados ao longo da próxima *sprint*.
- *Daily Scrum*: Encontro diário (geralmente de 15 minutos) para que a equipe alinhe o progresso, discuta obstáculos e ajuste o plano do dia.
- *Sprint Review*: Ao final de cada uma das sprints é apresentado o que foi implementado/desenvolvido com sucesso ao longo da *sprint*.
- *Sprint Retrospective*: Após a revisão, é realizada uma reunião destinada à avaliação do processo de trabalho realizado na *Sprint*, abordando os aspectos positivos e negativos percebidos.

Além desses artefatos, o Scrum define papéis específicos que garantem a eficácia do processo. Esses papéis são:

- *Product Owner*: é responsável por definir e priorizar os requisitos do projeto, representando os interesses do cliente e garantindo que o *Product Backlog* esteja sempre atualizado.
- *Scrum Master*: atua como facilitador, removendo impedimentos e assegurando que a equipe siga as práticas e valores do *Scrum*.
- *Development Team*: é o grupo que executa as tarefas planejadas, garantindo a entrega de incrementos de produto em cada sprint.

A aplicação do Scrum neste projeto seguiu um ciclo de vida composto por três fases principais: (i) planejamento, (ii) desenvolvimento e (iii) entrega. Na fase de planejamento, foram levantadas estimativas sobre o tempo e o esforço necessários para o desenvolvimento de cada requisito, definindo-se também a equipe e as ferramentas a serem utilizadas. Durante a fase de desenvolvimento, foi trabalhado a implementação dos requisitos, com a possibilidade de ajustes e incrementos ao longo do processo, conforme identificado nas reuniões de revisão e retrospectiva. Por fim, na fase de entrega, o produto final foi demonstrado e validado, permitindo a coleta de opiniões que orientaram melhorias futuras.

As reuniões periódicas presentes no Scrum foram adaptadas para atender à dinâmica deste projeto. A Sprint Planning definiu as atividades que foram desenvolvidas ao decorrer dos 15 dias de duração de cada uma das Sprints, a Sprint Review e a Sprint Retrospective foram realizadas ao fim de cada Sprint conforme definição do escopo da metodologia Scrum.

A aplicação web de dashboard foi desenvolvida na linguagem de programação Python, utilizando o framework Flask para gerenciar o lado servidor (back-end) e, na camada de apresentação (front-end), foram utilizados HTML, CSS e JavaScript. Já a comunicação entre a aplicação web e o banco de dados MySQL foi realizada através da biblioteca Python SQLAlchemy. A autenticação de login dos usuários foi implementada utilizando a biblioteca flask_login, que também controlou o acesso às áreas administrativas da ferramenta. As senhas dos usuários foram encriptadas através da biblioteca werkzeug.security, garantindo maior segurança na gestão de credenciais.

4 RESULTADOS E DISCUSSÕES

4.1 PROCESSO DE BACKUP DESENVOLVIDO

O processo de backup desenvolvido é totalmente automatizado. A execução se inicia com o carregamento de um arquivo `config.yaml`, o qual centraliza todas as configurações necessárias para a operação. Este arquivo inclui as credenciais de acesso ao banco de dados (host, porta, usuário e senha), a lista de bancos a serem incluídos ou excluídos do backup, os diretórios para saída dos arquivos temporários e compactados, bem como as credenciais de serviços de armazenamento em nuvem como AWS S3. Também estão definidos nesse arquivo parâmetros para o envio de notificações por e-mail, como remetente e destinatário. Além disso, o arquivo permite ao usuário ativar ou desativar funcionalidades específicas, como o envio do backup para a nuvem, a realização da compactação dos arquivos e o disparo de notificações por e-mail, tornando o processo altamente configurável e adaptável às necessidades de diferentes cenários.

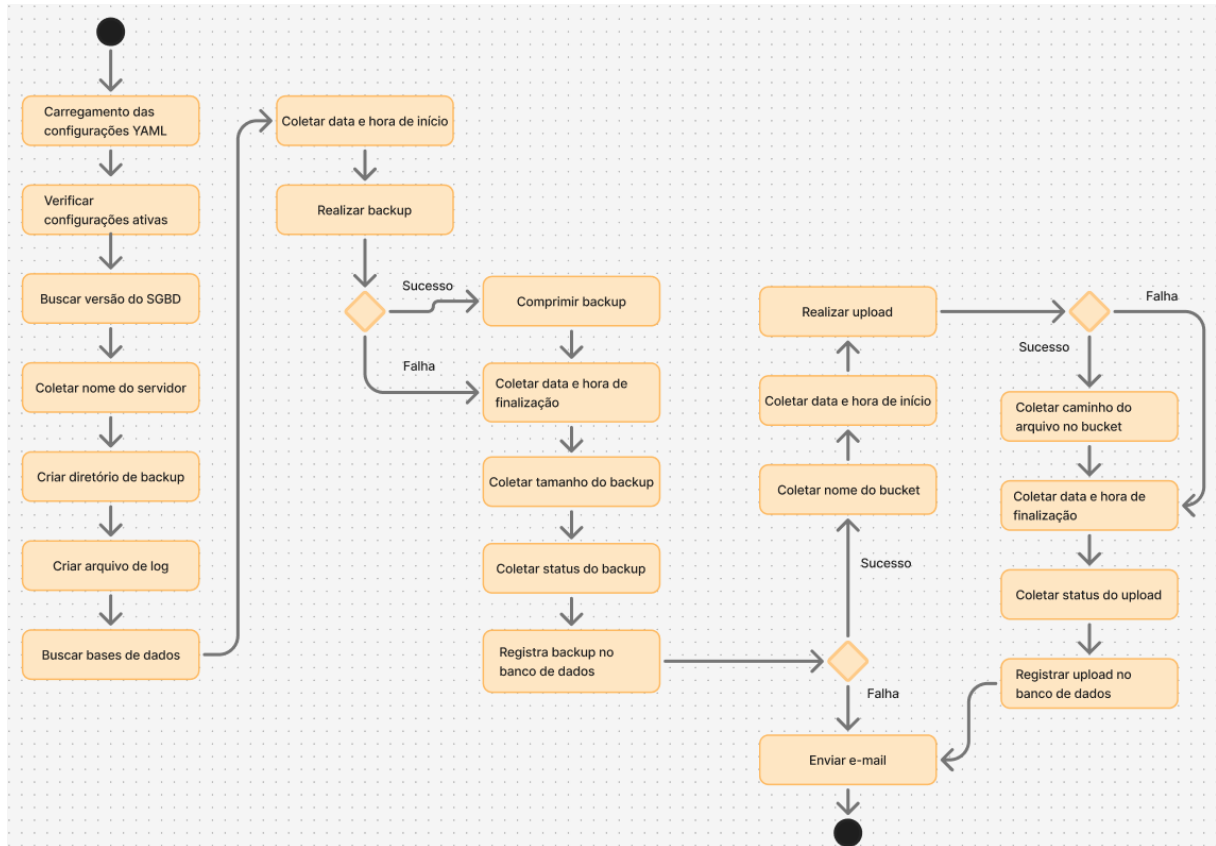
Após o carregamento das configurações, a aplicação se conecta ao banco de dados MySQL, identifica as bases de dados válidas para a operação e realiza o dump utilizando Mydumper, uma ferramenta paralela que evita o bloqueio da base (Locks). Se o backup for concluído com sucesso, o processo segue para a compactação dos arquivos gerados em formato `.zip`, coleta informações como horário de finalização, tamanho do arquivo e status da operação, e registra esses dados em uma tabela de controle no banco de dados. Caso ocorra alguma falha, os metadados são igualmente registrados, mas com status de erro.

Se habilitado, o processo prossegue com o envio do backup compactado para a nuvem. Essa etapa envolve a coleta do nome do bucket, o registro da data de início e, após o upload, o armazenamento do caminho final, a data de conclusão e o status do envio, que também são persistidos no banco de dados. Em caso de falha, o erro é coletado e registrado adequadamente na base de dados e o upload é registrado com status de erro.

Por fim, é disparado um e-mail com o resumo da execução contendo os principais dados da operação, incluindo falhas, se houver. Esse e-mail funciona como uma notificação final ao administrador do sistema, encerrando o ciclo do processo.

Uma representação visual detalhada desse fluxo pode ser observada na imagem abaixo:

Figura 1 - Fluxograma da ferramenta de backup



Fonte: Elaborado pelo autor, 2025.

4.2 COMPARATIVO COM O PROCESSO TRADICIONAL

Apesar da existência de diversas ferramentas voltadas à realização de backups lógicos, observa-se uma escassez de soluções que integrem, de forma unificada, todas as funcionalidades contempladas na ferramenta proposta. Recursos como seleção personalizada de bancos de dados, compactação automática, envio direto para serviços em nuvem, registro estruturado de logs em banco de dados e envio de notificações por e-mail são, em sua maioria, inexistentes ou demandam execução manual nos processos tradicionais de backup lógico.

Abaixo está a Tabela 1, que apresenta um comparativo entre o novo processo de backup desenvolvido e um dos processos mais tradicionais de backup lógico, o mysqldump:

Tabela 1 - Comparação processo antigo x novo

Critério	Processo tradicional	Novo processo
Execução manual	Manual ou semiautomatizada	Automatizado

Impacto na base	Pode causar travamento	Não há com uso do <i>MyDumper</i>
Compactação	Inexistente ou manual	Automática em <i>Zip</i>
Envio para nuvem	Manual ou inexistente	Automático para S3
Registro de logs	Limitado	Registrado em banco de dados
Notificações	Não possui	Envio automático por e-mail
Interface de acompanhamento	Inexistente	Sim, com painel e histórico
Configuração por YAML	Não aplicável	Sim, flexível e centralizada
Tolerância a falhas.	Não gerenciada	Sim, com registros para controle
Registro detalhado de metadados	Limitado	Completo: horários, status, tamanho
Adaptação modular	Inexistente	Sim, ativação opcional de etapas
Paralelismo	Inexistente	Sim, por padrão definido em 4 threads

Fonte: Elaborado pelo autor, 2025.

Para fins comparativos, adotou-se como referência o processo tradicional de backup utilizando a ferramenta *mysqldump*, amplamente empregada em ambientes MySQL por ser uma solução nativa. Apesar de sua popularidade, trata-se de uma ferramenta com desempenho limitado, especialmente quando utilizada de forma manual ou com scripts básicos.

4.3 TESTES FERRAMENTA DE BACKUP

Após a execução dos testes nos três cenários definidos (1GB, 5GB e 10GB), foram coletadas as métricas de desempenho da ferramenta de backup, com foco na análise das três etapas fundamentais do processo: dump, compactação e upload. Cada cenário foi submetido a cinco execuções consecutivas, permitindo o registro do tempo individual de cada etapa, bem como do tempo total de backup, agora definido como a soma dos tempos de dump, compactação e upload.

Os dados obtidos foram analisados estatisticamente, por meio do cálculo da média e do desvio padrão de cada métrica, com o objetivo de avaliar o comportamento da ferramenta quanto à estabilidade, consistência e escalabilidade frente ao crescimento do volume de dados.

Tabela 2 - Tempos Médios por Etapa e Total

Cenário	Dump	Compactação	Upload	Tempo total
1 GB	0,6s	3,6s	0,6s	4,8s
5 GB	4,0s	9,4s	1,0s	14,4s
10 GB	15,8s	19,6s	1,0s	36,4s

Fonte: Elaborado pelo autor, 2025.

Tabela 3 - Desvios Padrão por Etapa e Total

Cenário	Dump	Compactação	Upload	Tempo total
1 GB	0,55s	0,55s	0,55s	0,95s
5 GB	4,64s	0,55s	0,00s	4,97s
10 GB	6,46s	0,55s	0,00s	6,48s

Fonte: Elaborado pelo autor, 2025.

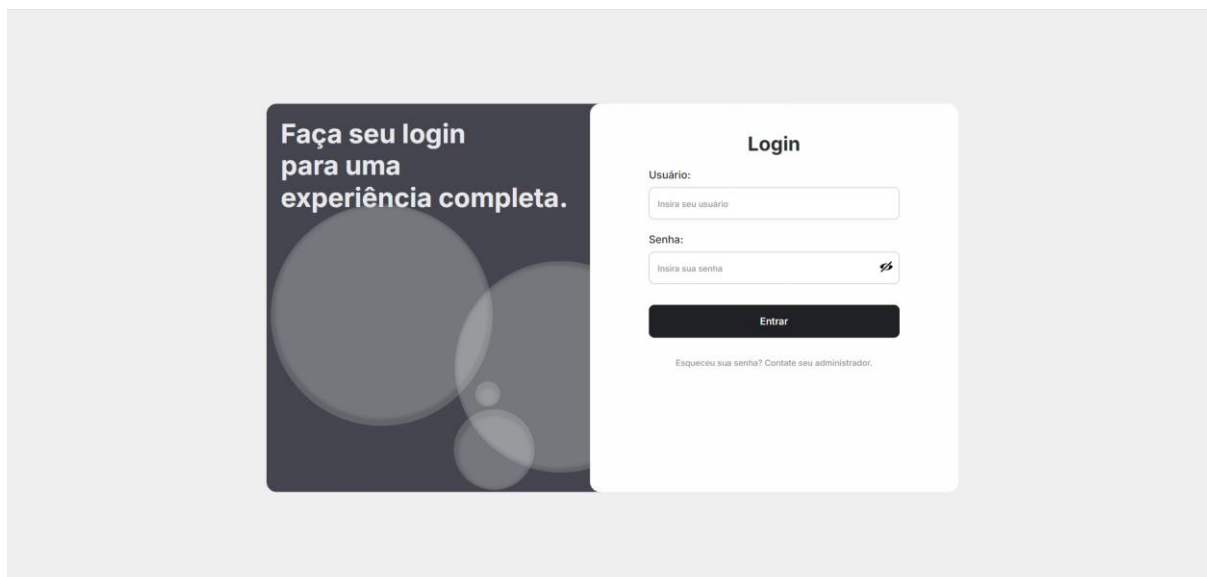
A Tabela 2 apresenta os tempos médios registrados em cada uma das etapas e o total consolidado por cenário, enquanto a Tabela 3 apresenta os respectivos desvios padrão. Os resultados mostram um crescimento progressivo e previsível no tempo total de backup à medida que o volume de dados aumenta, com destaque para a etapa de compactação como a mais custosa em termos de tempo, porém com baixa variabilidade entre as execuções. A etapa de upload, por sua vez, manteve-se constante e eficiente em todos os cenários.

4.4 FERRAMENTA WEB

A ferramenta web desenvolvida foi projetada para complementar a solução de backup, proporcionando uma interface gráfica robusta, intuitiva e responsiva para o monitoramento e administração do sistema. A aplicação foi construída utilizando o microframework Flask (em Python), com comunicação com o banco de dados MySQL realizada por meio da biblioteca SQLAlchemy, responsável pelo mapeamento objeto-relacional (ORM). A interface gráfica foi desenvolvida utilizando HTML5, CSS3, JavaScript e a biblioteca Bootstrap para responsividade e estilo.

A tela de login garante o controle de acesso ao sistema. Ela foi implementada utilizando HTML/CSS no front-end e Flask-Login no back-end, que oferece mecanismos de autenticação e gerenciamento de sessões. As senhas dos usuários são protegidas utilizando hash via Werkzeug Security, garantindo maior segurança das credenciais.

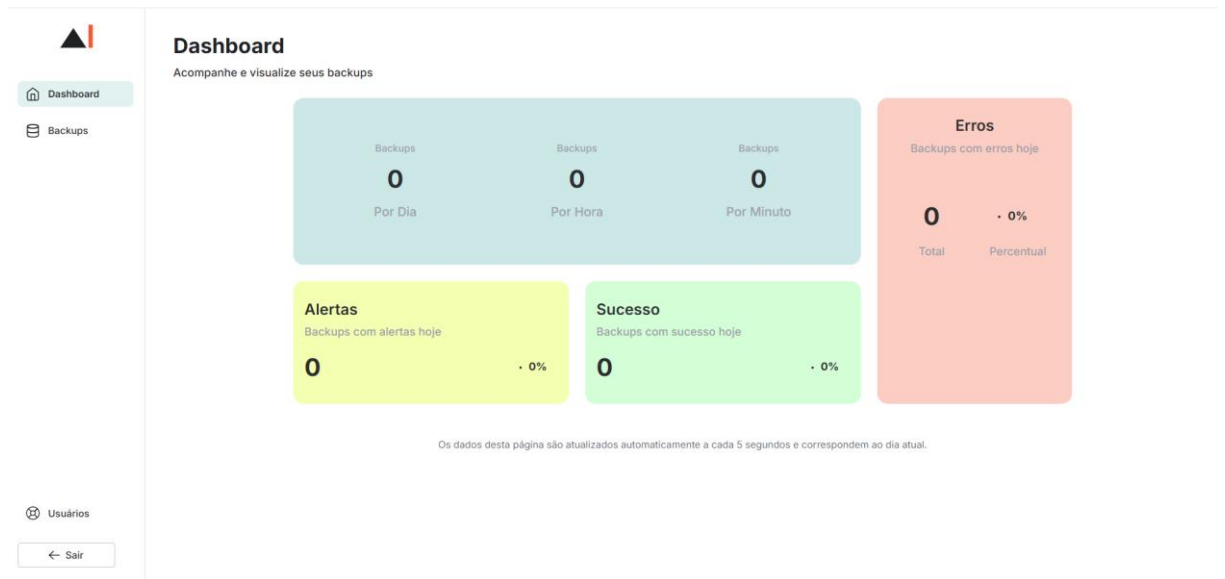
Figura 2 - Tela de Login



Fonte: Elaborado pelo autor, 2025.

Após a autenticação, o usuário é redirecionado para o dashboard principal, onde são apresentadas métricas do dia atual. Essa tela consome dados diretamente do banco MySQL e utiliza *Jinja2* para renderização dinâmica. Os gráficos de frequência por hora, minuto e dia foram implementados com a biblioteca *Chart.js*, embutida via JavaScript no HTML. Além da visualização dos totais por status (sucesso, alerta, erro), o dashboard permite uma visão rápida da performance operacional do sistema de backup.

Figura 3 - Dashboard com métricas dos backups



Fonte: Elaborado pelo autor, 2025.

A aba de "Backups" apresenta todos os registros salvos no banco de dados, ordenados por data. Essa funcionalidade foi construída com o auxílio de Flask, SQLAlchemy e Jinja2, utilizando laços de repetição no template para gerar as linhas da tabela com os dados. O botão de ação “ver detalhes”, utiliza rotas REST do Flask para recuperar dados específicos por ID e renderizar uma nova página com as informações detalhadas.

Figura 4 - Listagem completa de backups

Backups
Veja mais detalhes sobre seus backups

Todos 81

Q Buscar...

#	Hostname	Status	Tamanho	Data de início	Data de encerramento	Status upload
1	servidor02	Sucesso	6.21 GB	20/07/2025 16:15:23	20/07/2025 16:16:04	Sucesso >
2	servidor02	Sucesso	6.21 GB	14/07/2025 09:48:50	14/07/2025 09:49:34	Sucesso >
3	servidor02	Sucesso	6.21 GB	14/07/2025 09:47:35	14/07/2025 09:48:15	Sucesso >
4	servidor02	Sucesso	6.21 GB	14/07/2025 09:46:28	14/07/2025 09:47:02	Sucesso >
5	servidor02	Sucesso	6.21 GB	14/07/2025 09:45:25	14/07/2025 09:45:55	Sucesso >
6	servidor02	Sucesso	6.21 GB	14/07/2025 09:43:30	14/07/2025 09:44:00	Sucesso >
7	servidor02	Sucesso	3.11 GB	14/07/2025 09:37:24	14/07/2025 09:37:35	Sucesso >
8	servidor02	Sucesso	3.11 GB	14/07/2025 09:36:29	14/07/2025 09:36:42	Sucesso >
9	servidor02	Sucesso	3.11 GB	14/07/2025 09:35:33	14/07/2025 09:35:44	Sucesso >
10	servidor02	Sucesso	3.11 GB	14/07/2025 09:34:19	14/07/2025 09:34:30	Sucesso >

« Anterior Página 1 de 9 Próximo »

Usuários

← Sair

Fonte: Elaborado pelo autor, 2025.

Ao clicar em “ver detalhes” na listagem, o sistema exibe uma página com todos os dados da execução: tempos por etapa, tamanhos dos arquivos, status do upload e, se aplicável, erros

ou mensagens de alerta. Essa visualização é gerada dinamicamente com Flask e renderizada por meio do template engine Jinja2, com os dados recuperados via SQLAlchemy.

Figura 5 - Detalhes técnicos da execução de backup

The screenshot shows a web interface for backup details. The main content area is titled 'Detalhes do Backup' and contains two columns of information. The left column lists backup metadata, and the right column lists file and upload details. Below this, there is a section for 'Erros no backup' containing a log of warnings from a MySQL backup tool.

Backup		Backup	
Hostname	servidor02	Arquivo	backup_20250527,225401.zip
Banco de dados	vex_pass,percona,vex_backup,bench_test,auditoria,dba	Tamanho do arquivo	1.08 GB
Tamanho	3.09 GB	Caminho do arquivo	backup_20250527,225401.zip
Data e hora de início	27/05/2025 22:54:01	Bucket	vex-backup
Data e hora de conclusão	27/05/2025 22:55:24	Data e hora de início	27/05/2025 22:55:24
Duração	1 minuto 23 segundos	Data e hora de conclusão	27/05/2025 22:55:40
Status Backup	Alerta	Duração	16 segundos
		Status Upload	Sucesso

```

mydumper:293027: WARNING **: 22:54:08.157: Couldn't get master position: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near
'MASTER STATUS' at line 1
** (mydumper:293027): WARNING **: 22:54:08.180: Executing in no-locks mode, snapshot might not be consistent
** (mydumper:293027): WARNING **: 22:54:08.188: We are not able to determine if the backup will be consistent.
** (mydumper:293027): WARNING **: 22:54:08.188: We are not able to determine if the backup will be consistent.
** (mydumper:293027): WARNING **: 22:54:08.188: We are not able to determine if the backup will be consistent.
** (mydumper:293027): WARNING **: 22:54:08.188: We are not able to determine if the backup will be consistent.
** (mydumper:293027): WARNING **: 22:54:08.188: We are not able to determine if the backup will be consistent.
** (mydumper:293027): WARNING **: 22:54:08.188: We are not able to determine if the backup will be consistent.
** (mydumper:293027): WARNING **: 22:54:08.188: Couldn't get master position: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'MASTER
STATUS' at line 1
  
```

Fonte: Elaborado pelo autor, 2025.

A área administrativa foi construída com um formulário dinâmico em HTML5 e *Bootstrap*, validado com *WTFORMS* no *back-end*. O administrador do sistema pode adicionar, editar e excluir usuários. Os dados são persistidos via *SQLAlchemy*, e as ações são auditadas por meio de registros no banco de dados. Essa tela é fundamental em ambientes com múltiplos usuários, permitindo a gestão de permissões e controle das operações realizadas por cada conta.

Figura 6 - Interface de gerenciamento de usuários

The screenshot shows a web interface for user management. The main content area is titled 'Usuários' and contains a table listing users. The table has columns for user ID, name, privilege, last access, and registration date. There are also search and navigation controls.

#	Usuário	Privilegio	Último acesso	Data de cadastro	
1	vitoria.rodrigues	Cliente	21/07/2025	21/07/2025	
2	maria.silva	Administrador	21/07/2025	21/07/2025	
3	mateus.alves	Cliente	21/07/2025	21/07/2025	
4	willians.nunes	Administrador	10/06/2025	10/06/2025	
5	a	Administrador	31/05/2025	31/05/2025	
6	vex.admin	Administrador	14/02/2025	14/02/2025	

Fonte: Elaborado pelo autor, 2025.

5 CONCLUSÃO

A presente pesquisa apresentou o desenvolvimento e a análise de desempenho de uma ferramenta de backup automatizada, integrada a uma interface web, com foco na realização de backups de bancos de dados MySQL e no envio dos dados compactados para a nuvem. A proposta visou atender a requisitos de eficiência, escalabilidade e usabilidade, frequentemente demandados em ambientes corporativos que lidam com volumes crescentes de dados.

Por meio da execução de testes em três cenários distintos (1 GB, 5 GB e 10 GB), foi possível mensurar os tempos de execução das etapas críticas do processo, dump, compactação e upload e verificar, por meio de estatísticas como média e desvio padrão, a estabilidade e o comportamento da ferramenta frente ao aumento do volume de dados. Os resultados obtidos demonstraram que o sistema manteve consistência nos tempos de execução, mesmo diante de diferentes tamanhos de banco de dados, e que o desempenho foi proporcional ao volume de dados tratado, o que indica um excelente grau de escalabilidade e previsibilidade da performance da ferramenta em diferentes volumes de dados.

A interface web desenvolvida se mostrou eficaz ao permitir o acompanhamento em tempo real dos backups realizados, fornecendo ao usuário um panorama completo sobre as operações executadas, incluindo status, métricas por intervalo de tempo, histórico detalhado e gerenciamento de usuários. Esta camada de visualização e controle contribui diretamente para a adoção prática da ferramenta em cenários reais de produção.

Entre as melhorias previstas para versões futuras da aplicação, destaca-se a possibilidade de expandir o suporte para outras plataformas de armazenamento em nuvem, como Oracle Cloud, Google Cloud e Microsoft Azure, aumentando assim a flexibilidade da ferramenta. Além disso, planeja-se implementar funcionalidades de agendamento e execução manual de backups diretamente pela interface web, ampliando a autonomia dos usuários e oferecendo maior controle operacional sobre os processos de cópia e recuperação de dados.

REFERÊNCIAS

A guide to the Scrum Body of knowledge (SBOK Guide). Phoenix, Arizona: Scrumstudy, A Brand Of Vmedu, Inc, 2013.

ACTIVE SOLUTIONS. **A importância de fazer backup de dados em sua empresa | Active Solutions.** Disponível em: <https://www.activesolutions.com.br/blog/a-importancia-de-fazer-backup-de-dados-em-sua-empresa/>. Acesso em: 13 mar. 2025.

AMAZON WEB SERVICES. **Amazon S3 Documentation.** Seattle, 2024. Disponível em: <https://aws.amazon.com/pt/s3/>. Acesso em: 10 fev. 2025.

ARMBRUST, M. et al. **A view of cloud computing.** Communications of the ACM, n. 4, p. 50-58, 2010.

AUTOMATION. *In:* OXFORD UNIVERSITY PRESS. **Oxford Learner's Dictionaries.** Oxford, 2025. Disponível em: <https://www.oxfordlearnersdictionaries.com>. Acesso em: 2 mar. 2025.

BOTTA, A. et al. **Integration of Cloud computing and Internet of Things: A survey.** Future Generation Computer Systems, n. 2, p. 684–700, mar. 2016.

BRASIL. Lei nº 13.709, de 14 de agosto de 2018. **Lei Geral de Proteção de Dados Pessoais (LGPD).** Brasília: Presidência da República, 2018. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/113709.htm. Acesso em: 3 mar. 2025.

CHERVENAK, A. VELLANKI, V.; KURMAS, Z. **Protecting file systems: a survey of backup techniques.** 1998. Disponível em: https://www.researchgate.net/publication/2431989_Protecting_File_Systems_A_Survey_of_Backup_Techniques. Acesso em: 9 mar. 2025.

COHN, M. **Desenvolvimento de Software com Scrum.** Porto Alegre: Bookman, 2011.

DB-ENGINES. **DB-Engines Ranking.** 2025. Disponível em: <https://db-engines.com/en/ranking>. Acesso em: 2 mar. 2025.

ELMASRI, R.; NAVATHE, S. B. **Fundamentals of Database Systems.** 7. ed. Boston: Pearson Education, 2015.

FORD, H.; CROWTHER, S. **Today and Tomorrow.** [s.l.]: Doubleday, 1926.

GROOVER, M. P. **Automation, production systems, and computer-integrated manufacturing.** Nova York: Pearson Education, 2019.

HEUSER, CARLOS ALBERTO. **Projeto de Banco de Dados.** 6. ed. Porto Alegre: Bookman, 2009.

HEY, A. J. G. **The fourth paradigm: data-intensive scientific discovery**. Redmond, Washington: Microsoft Research, 2009.

HURWITZ, J. et al. **Big data for dummies**. Hoboken, N.J.: Wiley, 2013.

INTERNATIONAL DATA CORPORATION (IDC). **IDC's Global DataSphere Forecast**. Framingham, 2021. Disponível em: <https://www.idc.com/>. Acesso em: 10 fev. 2025.

ISMAIL, B. I.; MYDIN, M.; NIZAM, M.; KHALID, M. F. **Architecture of scalable backup service for private cloud**. In Open Systems (ICOS), 2013.

JESUS, G. B. A.; SCHIMIGUEL, J. **Implementação de Backup como processo de segurança da informação**. Jundiaí: Revista Atlante, 2018.

JOUNI, M.; RABAI, L. B. A. **Surveying and Analyzing Security Problems in Cloud Computing Environments**. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7016985>. Acesso em: 9 mar. 2025.

MAGRANI, Eduardo. **A Internet das coisas**. Rio de Janeiro: FGV Editora, 2018.

MELL, P.; GRANCE, T. **The NIST Definition of Cloud Computing**. Gaithersburg: National Institute of Standards and Technology, 2011.

MYDUMPER. **GitHub - mydumper/mydumper: Official MyDumper Project**. Disponível em: <https://github.com/mydumper/mydumper>. Acesso em: 7 mar. 2025.

MYSQL. **MySQL 9.1 Reference Manual 6.5.4 mysqldump - A Database Backup Program**. Disponível em: <https://dev.mysql.com/doc/refman/9.1/en/mysqldump.html>. Acesso em: 7 mar. 2025.

NIAZKHANI, Z. et al. **Barriers to patient, provider, and caregiver adoption and use of electronic personal health records in chronic care: a systematic review**. BMC Medical Informatics and Decision Making, n. 1, 2020.

ORACLE. **Architecture Principles and Implementation Practices for Remote Replication Using Oracle ZFS Storage Appliance An Oracle ZFS Storage Appliance White Paper**. [s.l.: s.n.]. Disponível em: <https://www.oracle.com/a/otn/docs/zfssa-genreplication-v30.pdf>. Acesso em: 9 mar. 2025.

PEREIRA, Adriana Soares; SHITSUKA, Dorlivete Moreira; PARREIRA, Fabio José; SHITSUKA, Ricardo. **Metodologia da pesquisa científica**. 1. ed. Santa Maria, RS: UFSM, NTE, 2018. Disponível em: https://repositorio.ufsm.br/bitstream/handle/1/15824/Lic_Computacao_Metodologia-Pesquisa-Cientifica.pdf. Acesso em: 9 mar. 2025.

POON, E. G. et al. **Effect of Bar-Code Technology on the Safety of Medication Administration**. New England Journal of Medicine, n. 18, p. 1698–1707, 2010.

RAVI, V.; ARUN KUMAR SAMAYAM; NARESH KUMAR MIRYALA. **MySQL Logical Backup**. Apress eBooks, p. 431–482, 2024.

SCHWARTZ, B.; ZAITSEV, P.; TKACHENKO, V. **High Performance MySQL**. [s.l.] O'Reilly Media, 2012.

SILBERSCHATZ, A.; KORTH, H. F.; S SUDARSHAN. **Database system concepts**. New York, Ny: Mcgraw-Hill Education, 2020.

SILVA, J. M. A.; NETO, S. V. B.; ALMEIDA, E. S. **Análise de um sistema de backup/recovery para grandes volumes de dados**. Resende: Associação Educacional Dom Bosco (AEDB), 2015. Disponível em: <https://www.aedb.br/seget/arquivos15/11822197.pdf>. Acesso em: 6 mar. 2025.

SUBASHINI, S.; KAVITHA, V. **A Survey on Security Issues in Service Delivery Models of Cloud Computing**. Journal of Network and Computer Applications, n. 1, 2011.

APÊNDICE

APÊNDICE I – PROJETO DE SOFTWARE.

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
RONDÔNIA – IFRO CAMPUS JI-PARANÁ**
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

WILLIANS GOMES NUNES

PROJETO DO SISTEMA VEX BACKUP 1.0

Ji-Paraná – RO

2025

LISTA DE FIGURAS

Figura 1 - Protótipo Realizar Login	17
Figura 2 - Protótipo Mensagens de Login	18
Figura 3 - Protótipo Exibir Dashboard	19
Figura 4 - Protótipo Listar Backup	19
Figura 5 - Protótipo Detalhes do Backup	20
Figura 6 - Protótipo Listar Usuário	21
Figura 7 - Protótipo Mensagens Listar Usuário	21
Figura 8 - Protótipo Cadastrar usuário	22
Figura 9 - Protótipo Mensagens Cadastrar Usuário	23
Figura 10 - Diagrama de atividades.....	24
Figura 11 - Diagrama de Caso de Uso	28
Figura 12 - Diagrama de Entidade-Relacionamento	33

LISTA DE QUADROS

Quadro 1 - Arquitetura do Sistema	7
Quadro 2 - Configuração Mínima dos Servidores	7
Quadro 3 - Equipe de Desenvolvimento	8
Quadro 4 - Estrutura do Scrum e seus Artefatos	9
Quadro 5 - Papéis na metodologia Scrum.....	9
Quadro 6 - Product Backlog.....	10
Quadro 7 - Cronograma de Sprints	11
Quadro 8 - Realizar <i>Login</i>	12
Quadro 9 - Exibir <i>Dashboard</i>	12
Quadro 10 - Listar Backup.....	13
Quadro 11 - Listar Usuário	14
Quadro 12 - Cadastrar Usuário.....	14
Quadro 13 - Realizar Backup.....	15
Quadro 14 - Catálogo de dados da tabela usuario.....	34
Quadro 15 - Catálogo de dados da tabela usuario.....	34
Quadro 16 - Catálogo de dados da tabela usuario_privilegio	34
Quadro 17 - Catálogo de dados da tabela backup_log.....	35
Quadro 18 - Catálogo de dados da tabela upload_s3_log.....	36

SUMÁRIO

1	DOCUMENTO DE ESPECIFICAÇÃO DO PROJETO	6
1.1	SOBRE O PRODUTO DE SOFTWARE.....	6
1.2	ASPECTOS TÉCNICOS DO PRODUTO	6
1.2.1	Tecnologias Front End	6
1.2.2	Tecnologias Back End	6
1.3	EQUIPE DE DESENVOLVIMENTO	8
1.4	METODOLOGIA DE DESENVOLVIMENTO.....	8
1.4.1	Product Backlog	10
1.4.2	Sprints Backlog	10
2	DOCUMENTO DE REQUISITOS	12
2.1	REALIZAR <i>LOGIN</i>	12
2.2	EXIBIR DASHBOARD	12
2.3	LISTAR BACKUP	13
2.4	LISTAR USUÁRIO	13
2.5	CADASTRAR USUÁRIO	14
2.6	REALIZAR BACKUP	15
3	PROTÓTIPO DO SISTEMA	17
3.1	RF1 – Realizar Login	17
3.1.1	Mensagens de Login	17
3.2	RF2 – Exibir Dashboard	18
3.3	RF3 – LISTAR BACKUP	19
3.3.1	Detalhes do Backup	20
3.4	RF4 – LISTAR USUÁRIO	20
3.4.1	Mensagens Listar Usuário	21
3.5	RF5 – CADASTRAR USUÁRIO	22

3.5.1	Mensagens Cadastrar Usuário	22
4	FERRAMENTA DE BACKUP.....	24
4.1	DIAGRAMA DE ATIVIDADES.....	24
4.2	ETAPA DE BACKUP.....	25
4.3	ETAPA DE UPLOAD.....	26
4.4	NOTIFICAÇÕES E TRATAMENTO DE FALHAS	27
5	MODELO DE CASO DE USO.....	28
5.1	DIAGRAMA DE CASO DE USO	28
5.2	CASO DE USO EXPANDIDO.....	28
5.2.1	Caso de Uso – Realizar login	28
5.2.2	Caso de Uso – Exibir dashboard	29
5.2.3	Caso de Uso – Listar backup	29
5.2.4	Caso de Uso – Realizar backup	30
5.2.5	Caso de Uso – Listar usuário	30
5.2.6	Caso de Uso – Cadastrar usuário.....	31
6	DIAGRAMA DE ENTIDADE-RELACIONAMENTO	33
7	DICIONÁRIO DE DADOS.....	34
7.1	TABELA USUÁRIO.....	34
7.2	TABELA PRIVILEGIO	34
7.3	TABELA USUARIO_PRIVILEGIO	34
7.4	TABELA BACKUP_LOG	35
7.5	TABELA UPLOAD_S3_LOG.....	36

1 DOCUMENTO DE ESPECIFICAÇÃO DO PROJETO

1.1 SOBRE O PRODUTO DE SOFTWARE

Será desenvolvido um software de tecnologia desktop para a realização e gestão de backups de bancos de dados do SGBD (*Sistema de Gerenciamento de Banco de Dados*) MySQL. Para primeira versão, o software terá o nome de *Vex Backup 1.0*. O software possui a missão de realizar os processos de backup das bases de dados e o upload do backup para a nuvem de forma automática, desse modo, reduzindo o tempo gasto realizando estes processos de forma manual, além de possibilitar a visualização e acompanhamento do status de cada backup e upload realizado através de um dashboard.

1.2 ASPECTOS TÉCNICOS DO PRODUTO

1.2.1 Tecnologias Front End

O software será construído utilizando a Linguagem de Programação Python, com *HTML*, *CSS* e *JavaScript* estruturados por meio do *Framework Flask*, em conjunto com a *Engine Jinja2* para renderização dinâmica das páginas.

Durante a fase de análise e projeto, será utilizado a ferramenta Figma para elaborar o protótipo do sistema, e a ferramenta *Atlassian Work Management* para organização e controle das atividades no quadro *Kanban*.

1.2.2 Tecnologias Back End

O software será desenvolvido na linguagem de programação Python, que será responsável por realizar a automação dos backups do banco de dados MySQL, compactação e envio dos arquivos compactados ao serviço de armazenamento em nuvem *Bucket S3* na AWS (*Amazon Web Service*).

Para viabilizar todas as integrações, serão utilizadas as seguintes bibliotecas e ferramentas:

- Biblioteca *PyMySQL* para a comunicação com o banco de dados MySQL;

- Biblioteca *Boto3* para envio dos arquivos ao serviço S3, realizando tanto o upload quanto consultas;
- Biblioteca *Subprocess* para a execução do *MyDumper* e do *7-Zip* na geração e compressão dos *Backups Dump*.
- Ferramenta *MyDumper* para realização dos backups do tipo *Dump*.
- Ferramenta *7-Zip* para compactação dos arquivos *Dump*, reduzindo o tempo de upload para o *Bucket S3*.
- Ferramenta *Bucket S3* para armazenamento em nuvem dos backups realizados.

Considerando que o projeto está dividido em uma aplicação Web e uma ferramenta de backup, será utilizando uma arquitetura dividida em dois servidores, sendo que, a aplicação Web ficará instalada em um servidor exclusivo, assim como o banco de dados MySQL. Deste modo, temos as seguintes arquiteturas:

Quadro 1 - Arquitetura do Sistema

Componente	Descrição
(i) Servidor de Aplicação (<i>Flask</i>)	Hospeda a interface Web desenvolvida através do Framework Flask.
	Disponibiliza o acesso à interface de Dashboard aos usuários por meio do navegador Web.
	Se conecta ao servidor de banco de dados para consultar os dados de backups.
(ii) Servidor de banco de dados (MySQL) e execução de Backup	Armazena os metadados referentes às rotinas de backups.
	Executa os scripts de backup em Python, responsáveis por realizar o backup, compactação e envio para o Bucket S3.

Fonte: Elaborado pelo autor, 2025.

No que tange aos servidores de Aplicação e de Banco de Dados, abaixo estão dispostas as configurações mínimas para o seu regular funcionamento:

Quadro 2 - Configuração Mínima dos Servidores

Parâmetro	Servidor de Aplicação	Servidor de Banco de Dados
Processador	2 núcleos (ou superior)	4 núcleos (ou superior)
Memória RAM	4 GB (ou superior)	8 GB (ou superior)
Armazenamento	SSD com 20 GB (ou superior)	SSD com 100 GB (ou superior)

Sistema Operacional	Oracle Linux 9.4 (ou superior)	Oracle Linux 9.4 (ou superior)
Conexão de Rede	30 MB/s (ou superior)	100 MB/s (ou superior)

Fonte: Elaborado pelo autor, 2025.

Para maior clareza, apresenta-se no item 4 deste documento um diagrama de atividades, bem como uma descrição detalhada das atividades realizadas pela ferramenta de backup.

1.3 EQUIPE DE DESENVOLVIMENTO

A equipe de desenvolvimento será composta pelos seguintes papéis e atribuições, de acordo com a metodologia ágil *Scrum* estabelecida para este projeto:

Quadro 3 - Equipe de Desenvolvimento

ID	Nome	Função	Atribuições
01	Willians Gomes Nunes	<i>Scrum Master</i>	Liderar a equipe durante o andamento do projeto.
02	Willians Gomes Nunes	Desenvolvedor	Desenvolver o projeto de software.
03	Willians Gomes Nunes	<i>Product Owner</i>	Representante do cliente que ajuda no entendimento do negócio.
04	Willians Gomes Nunes	Especialista	Usuário especializado no domínio da informação que ajuda no entendimento do negócio.

Fonte: Elaborado pelo autor, 2025.

1.4 METODOLOGIA DE DESENVOLVIMENTO

Para o desenvolvimento deste projeto, será adotada a metodologia ágil de desenvolvimento de software *Scrum*, reconhecida por gerir projetos complexos de maneira iterativa e adaptativa. Com o auxílio dessa metodologia, além do *Scrum* promover a comunicação e a transparência, facilita o acompanhamento do andamento do projeto por meio de ciclos curtos, as *Sprints*, que possibilitam a definição de tarefas e prazos, que variam entre duas a quatro semanas de duração. Para este projeto este prazo será adaptado para a duração de 15 dias de duração para cada uma das *Sprints*.

O *Scrum* pode ser aplicado a diversos modelos de projetos, não ficando restrito exclusivamente à implementação de software. Com o *Scrum* é possível assegurar que todas as

etapas sejam acompanhadas de forma visível e colaborativa. Essa estrutura é composta por artefatos essenciais, como:

Quadro 4 - Estrutura do Scrum e seus Artefatos

Artefato	Atribuições
(i) <i>Product Backlog</i>	Lista priorizada de todas as funcionalidades, tarefas e melhorias a serem implementadas.
(ii) <i>Sprint Planning</i>	Reunião realizada antes do início de cada sprint para definir quais atividades do <i>Product Backlog</i> serão trabalhadas na próxima iteração.
(iii) <i>Daily Scrum</i>	Encontro diário (geralmente de 15 minutos) para alinhar o progresso, discutir obstáculos e ajustar o plano de trabalho do dia.
(iv) <i>Sprint Review</i>	Reunião realizada ao final de cada sprint para apresentar e validar o que foi desenvolvido com sucesso durante o ciclo.
(v) <i>Sprint Retrospective</i>	Reunião realizada após a <i>Sprint Review</i> para avaliar o processo de trabalho, identificando os pontos positivos e negativos e promovendo melhorias.

Fonte: Elaborado pelo autor, 2025.

Além desses artefatos, o *Scrum* define papéis específicos que garantem a eficácia do processo. Esses papéis são:

Quadro 5 - Papéis na metodologia Scrum

Papel	Descrição
(i) <i>Product Owner</i>	Responsável por definir e priorizar os requisitos do projeto, representando os interesses do cliente e garantindo que o <i>Product Backlog</i> esteja sempre atualizado.
(ii) <i>Scrum Master</i>	Atua como facilitador, removendo impedimentos e assegurando que a equipe siga as práticas e os valores do <i>Scrum</i> .
(iii) <i>Development Team</i>	Grupo multifuncional que executa as tarefas planejadas, garantindo a entrega de incrementos de produto a cada <i>sprint</i> .

Fonte: Elaborado pelo autor, 2025.

A aplicação do *Scrum* neste projeto seguirá um ciclo de vida composto por três fases principais: (i) planejamento, (ii) desenvolvimento e (iii) entrega. Na fase de planejamento, serão levantadas estimativas sobre o tempo e o esforço necessários para o desenvolvimento de cada requisito, definindo-se também a equipe e as ferramentas a serem utilizadas. Durante a fase de desenvolvimento, será trabalhado a implementação dos requisitos, com a possibilidade de

ajustes e incrementos ao longo do processo, conforme identificado nas reuniões de revisão e retrospectiva. Por fim, na fase de entrega, o produto final será demonstrado e validado, permitindo a coleta de opiniões que orientarão melhorias futuras.

As reuniões periódicas presentes no *Scrum* serão adaptadas para atender à dinâmica deste projeto. A *Sprint Planning* irá definir as atividades que serão desenvolvidas ao decorrer dos 15 dias de duração de cada uma das *Sprints*, a *Sprint Review* e a *Sprint Retrospective* serão realizadas ao fim de cada *Sprint* conforme definição do escopo da metodologia *Scrum*.

1.4.1 Product Backlog

Neste tópico serão documentados todos os itens do **Product Backlog** criados ao longo do desenvolvimento do projeto.

Quadro 6 - Product Backlog

Item	Descrição	Estimativa de Esforço	Prioridade
1	Controlar o processo de backup das bases de dados.	8 dias	1°
2	Controlar o processo de compactação dos backups.	7 dias	2°
3	Controlar o processo de envio dos backups para o Bucket S3.	15 dias	3°
4	Controlar o processo de envio de e-mail com os detalhes dos backups performados.	7 dias	4°
5	Gerir os usuários e os níveis de acesso ao sistema e menus.	15 dias	5°
6	Desenvolver o dashboard para gestão de backups e o relatório de metadados dos backups.	15 dias	6°

Fonte: Elaborado pelo autor, 2025.

1.4.2 Sprints Backlog

Neste tópico serão documentados todos os itens das *Sprints Backlog* criados ao longo do desenvolvimento do projeto. Todas as tarefas do projeto foram gerenciadas com a ferramenta *Atlassian Work Management*.

Quadro 7 - Cronograma de Sprints

Item do Product Backlog	Sprint	Período	Descrição da Tarefa	Responsável	Esforço
1 e 2	1	24/03/2025 a 09/04/2025	Realizar a implementação da função responsáveis por realizar o processo de backup das bases de dados do MySQL.	Willians G. Nunes	20h
			Construir a função destinada a compactação dos arquivos de backup utilizando a ferramenta 7-Zip.	Willians G. Nunes	20h
			Construir registro de log execução para envio dos metadados por e-mail.	Willians G. Nunes	10h
3 e 4	2	10/04/2025 a 25/04/2025	Construir o Bucket na plataforma da AWS para que os backups possam ser enviados ao Bucket.	Willians G. Nunes	25h
			Desenvolver função responsável por realizar o envio dos dados para o Bucket S3.	Willians G. Nunes	10h
			Desenvolver função responsável por realizar o envio dos e-mails contendo os metadados do processo de backup e upload.	Willians G. Nunes	25h
5 e 6	3	26/04/2025 a 09/05/2025	Desenvolver autenticação dos usuários no sistema.	Willians G. Nunes	10h
			Desenvolver controle de usuários no sistema.	Willians G. Nunes	10h
			Desenvolver Dashboard de visualização de backups.	Willians G. Nunes	10h
			Construir interface de visualização de metadados dos backups.	Willians G. Nunes	10h

Fonte: Elaborado pelo autor, 2025.

2 DOCUMENTO DE REQUISITOS

Após o levantamento de requisitos, foram identificados e documentados os seguintes elementos essenciais que orientarão o desenvolvimento do software:

2.1 REALIZAR LOGIN

Quadro 8 - Realizar Login

Requisito funcional		
Nome: Realizar Login		Código: RF1
Descrição: O usuário deve preencher os dados para entrar no sistema.		
Estimativa de Esforço: 1 dia.		Prioridade: Alta
Requisitos não funcionais		
ID NF	Descrição	Categoria
1.1	A função deverá solicitar autenticação para habilitar o acesso às informações do sistema por meio dos campos: usuário e senha.	Interface
1.2	A função deverá conter a mensagem “Senha ou login inválidos”, a mensagem deverá conter um botão ‘Fechar’.	Interface
1.3	A função deverá conter uma mensagem indicativa “Esqueceu sua senha? Contate seu administrador”.	Interface

Fonte: Elaborado pelo autor, 2025.

2.2 EXIBIR DASHBOARD

Quadro 9 - Exibir Dashboard

Requisito funcional		
Nome: Exibir dashboard		Código: RF2
Descrição: O sistema deverá exibir as principais estatísticas dos backups realizados.		
Estimativa de Esforço: 1 dia.		Prioridade: Alta
Requisitos não funcionais		
ID NF	Descrição	Categoria

1.1	A função deverá exibir os dados quantitativos de total de backups por dia, por hora e por minuto.	Interface
1.2	A função deverá realizar uma consulta à base e dados a cada 1 minuto para atualização dos dados exibidos.	Especificação
1.3	A função deverá conter um gráfico indicando o total de backups com alertas durante o dia atual.	Interface
1.4	A função deverá possuir um gráfico exibindo o total de backups realizados com sucesso durante o dia atual.	Interface
1.5	A função deverá conter a mensagem “Acompanhe e visualize seus backups” ao topo da interface.	Interface
1.6	A função deverá conter um gráfico indicando o total de backups com erros durante o dia atual.	Interface

Fonte: Elaborado pelo autor, 2025.

2.3 LISTAR BACKUP

Quadro 10 - Listar Backup

Requisito funcional		
Nome: Listar backup		Código: RF3
Descrição: O sistema deverá permitir que o usuário visualize os metadados dos backups realizados.		
Estimativa de Esforço: 2 dias.		Prioridade: Média
Requisitos não funcionais		
ID NF	Descrição	Categoria
1.1	A função deverá conter a mensagem “Veja mais detalhes sobre seus backups” ao topo da interface.	Interface
1.2	A função deverá conter um campo de busca contendo um ícone de lupa e a mensagem “Buscar...” dentro do campo.	Interface
1.3	A função deverá permitir que o usuário busque por dados de <i>Hostname</i> , Status, Tamanho, Data de início, Data de encerramento.	Interface
1.4	A função deverá listar em uma tabela os dados: <i>Hostname</i> , Status, Tamanho, Data de início, Data de encerramento e Status do Upload.	Interface

Fonte: Elaborado pelo autor, 2025.

2.4 LISTAR USUÁRIO

Quadro 11 - Listar Usuário

Requisito Funcional		
Nome: Listar usuário		Código: RF4
Descrição: O sistema deverá exibir os usuários cadastrados no sistema.		
Estimativa de Esforço: 1 dia.		Prioridade: Alta
Requisitos não funcionais		
ID NF	Descrição	Categoria
1.1	A função deverá possuir a mensagem “Crie, gerencie e altere as credenciais de acesso ao sistema” ao topo da interface.	Interface
1.2	A função deverá conter um campo de busca que permita que o usuário busque por logins criados no sistema.	Interface
1.3	A função deverá conter um botão chamado “Cadastrar” que permita que o usuário cadastre uma nova conta de usuário para acessar o sistema.	Interface
1.4	A função deverá conter um ícone de lixeira na cor vermelha ao lado de cada registro de usuário que permita a exclusão do registro.	Interface
1.5	A função deverá conter um ícone de lápis na cor verde ao lado de cada registro de usuário que permita a edição do registro.	Interface
1.6	A função deverá conter uma mensagem de confirmação ao clicar no ícone de lixeira para excluir o registro contendo a seguinte mensagem “Deseja realmente excluir o acesso do usuário?” e contendo os botões “Cancelar” e “Excluir”.	Interface
1.7	A função deverá exibir, em formato de tabela, a lista dos usuários cadastrados no sistema, obedecendo à seguinte ordem de colunas: "Usuário", "Privilégio", "Último acesso" e "Data de cadastro".	Interface

Fonte: Elaborado pelo autor, 2025.

2.5 CADASTRAR USUÁRIO

Quadro 12 - Cadastrar Usuário

Requisito funcional		
Nome: Cadastrar usuário		Código: RF5
Descrição: O usuário deve preencher os campos para adicionar um novo usuário ao sistema.		
Estimativa de Esforço: 2 dias		Prioridade: Alta
Requisitos não funcionais		
ID NF	Descrição	Categoria

1.1	A função deverá possuir a mensagem “Adicione um novo usuário ao sistema” ao topo da interface.	Interface
1.2	A função deverá conter os seguintes campos de preenchimento obrigatório: "Nome do usuário", "Login", "Senha", "Repetir Senha", "Permissão de acesso" e "Data de cadastro".	Interface
1.3	Os dados que deverão ser armazenados no bando de dados são: nome, login, senha, permissão de acesso (Privilégio) e data de cadastro.	Especificação
1.4	A função deverá apresentar botões para as seguintes ações: Salvar e Cancelar.	Interface
1.5	A função deverá conter a mensagem “A data de cadastro será atribuída automaticamente pelo sistema.” logo abaixo do campo “Data de cadastro”.	Interface
1.6	A função não deverá permitir que o usuário interaja com o campo “Data de cadastro”, deixando-o levemente acinzentado.	Especificação
1.7	O campo “Permissão de acesso” deverá ser uma caixa de seleção e deverá buscar os dados dos níveis de permissão no banco de dados.	Especificação
1.8	A função deverá conter exibir uma mensagem de erro contendo a mensagem: “Erro ao cadastrar o usuário. Todos os campos devem ser preenchidos para que o usuário seja cadastrado.” e um botão com a mensagem “Ok!”.	Interface
1.9	Os dados coletados dos campos de “Senha” deverão ser armazenados criptografados no banco de dados.	Segurança
1.10	Ao clicar em salvar a função deverá exibir uma mensagem de confirmação contendo a mensagem: “Novo usuário cadastrado!” contendo um botão escrito “Ok!”.	Interface

Fonte: Elaborado pelo autor, 2025.

2.6 REALIZAR BACKUP

Quadro 13 - Realizar Backup

Requisito funcional		
Nome: Realizar backup		Código: RF6
Descrição: O sistema deverá realizar o backup das bases de dados do MySQL.		
Estimativa de Esforço: 1 dia.		Prioridade: Alta
Requisitos não funcionais		
ID NF	Descrição	Categoria
1.1	A função deverá coletar o nome do servidor em que está sendo executado automaticamente sempre que for executada.	Especificação
1.2	A função deverá coletar a data e hora de início sempre que for executada.	Especificação

1.3	A função deverá criar um novo diretório seguindo a padronização: backup_ + data + hora que a função realizou a criação do diretório.	Especificação
1.4	A função deverá realizar a ferramenta <i>Mydumper</i> para realizar o backup das bases de dados individualmente.	Especificação
1.5	A função deverá analisar o log de execução do <i>Mydumper</i> para avaliar se o backup foi performedado sem erros ou alertas.	Especificação
1.6	A função deverá compactar o backup utilizando a ferramenta 7-zip.	Especificação
1.7	A função deverá coletar o tamanho do arquivo de backup após a compactação.	Especificação
1.8	A função deverá coletar a versão do SGBD do servidor em que está sendo executada.	Especificação
1.9	A função deverá inserir os metadados coletados durante a execução do backup no banco de dados, sendo eles: "hostname", "databases", "tamanho do arquivo", "status do backup", "versão do SGBD", "data de início" e "data de encerramento".	Especificação
1.10	A função deverá realizar o upload do arquivo de backup compactado para o serviço de armazenamento em nuvem da AWS Bucket S3.	Especificação
1.11	A função deverá realizar o envio do log de execução do backup no formato de anexo, seguindo a padronização: backup_script.log e mydumper_log.log	Especificação
1.12	A função deverá enviar o e-mail com o título seguindo padrão: Rotina de backup + servidor + Data + hora.	Especificação
1.13	A função deverá inserir os metadados coletados durante a execução do upload do arquivo de backup compactado, sendo eles: tamanho do arquivo de backup, nome do arquivo, caminho do arquivo no Bucket, nome do Bucket, status do upload, data e hora de início e data e hora de encerramento.	Especificação

Fonte: Elaborado pelo autor, 2025.

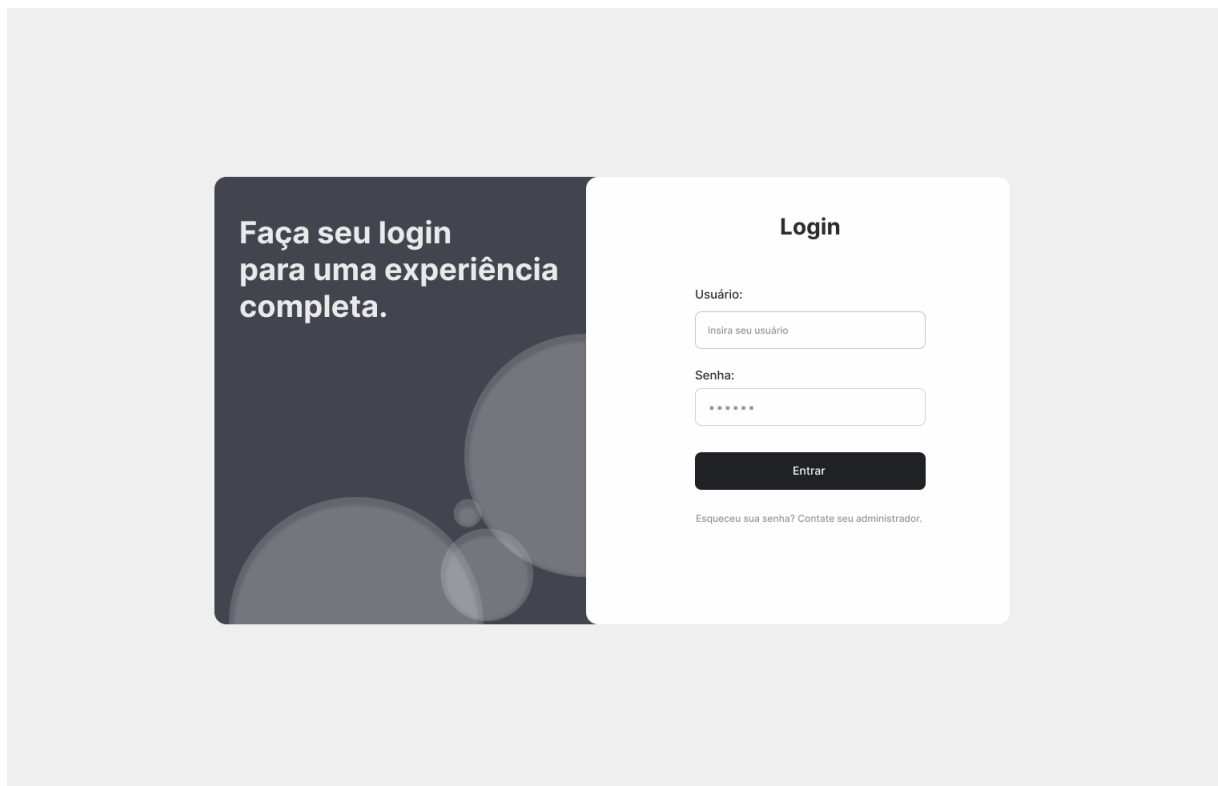
3 PROTÓTIPO DO SISTEMA

O protótipo do sistema *Vex Backup* foi desenvolvido utilizando a ferramenta Figma, permitindo a criação de interfaces interativas e facilitando a validação do design antes da implementação final. O protótipo pode ser acessado por meio do seguinte endereço eletrônico: <<https://www.figma.com/design/dGZnpWTDSfGpNVrMnDZZyh/VexBackup?node-id=0-1&m=dev&t=-uvYDTzclX6DQLN80-1>>.

Com exceção do requisito funcional "2.6 Realizar backup" que é uma funcionalidade exclusiva do *Back-End* do projeto e, portanto, não possuem interface vinculada, apresentam-se a seguir as imagens do protótipo validado e aprovado para os demais requisitos funcionais, seguindo a mesma ordem estabelecida no "Documento de Requisitos", para fins de documentação do projeto:

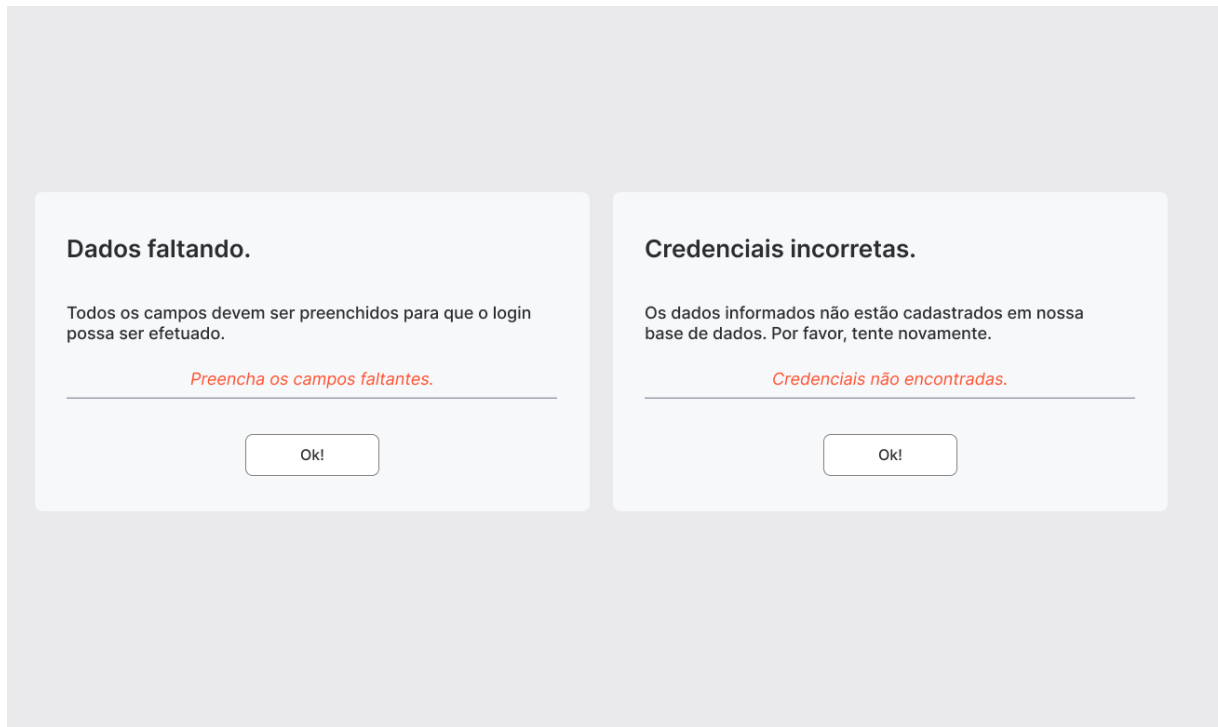
3.1 RF1 – Realizar Login

Figura 1 - Protótipo Realizar Login



Fonte: Elaborado pelo autor, 2025.

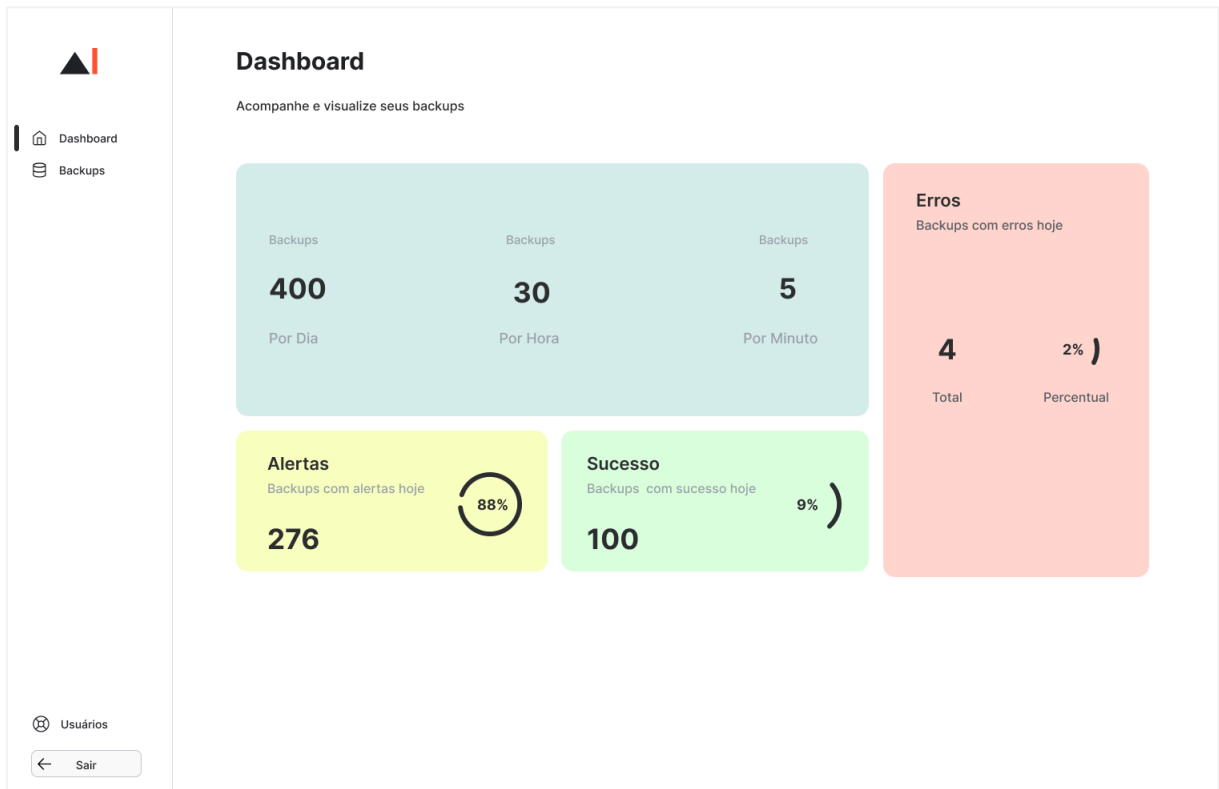
3.1.1 Mensagens de Login

Figura 2 - Protótipo Mensagens de Login

Fonte: Elaborado pelo autor, 2025.

3.2 RF2 – Exibir Dashboard

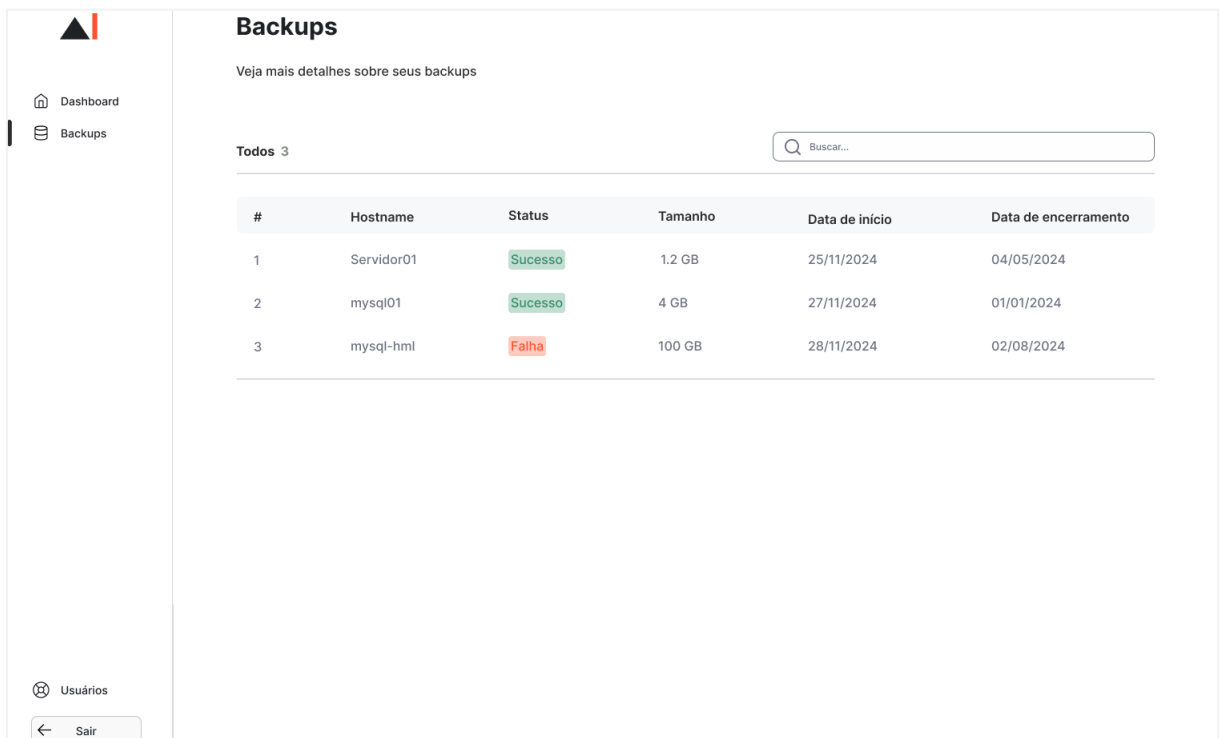
Figura 3 - Protótipo Exibir Dashboard



Fonte: Elaborado pelo autor, 2025.

3.3 RF3 – LISTAR BACKUP

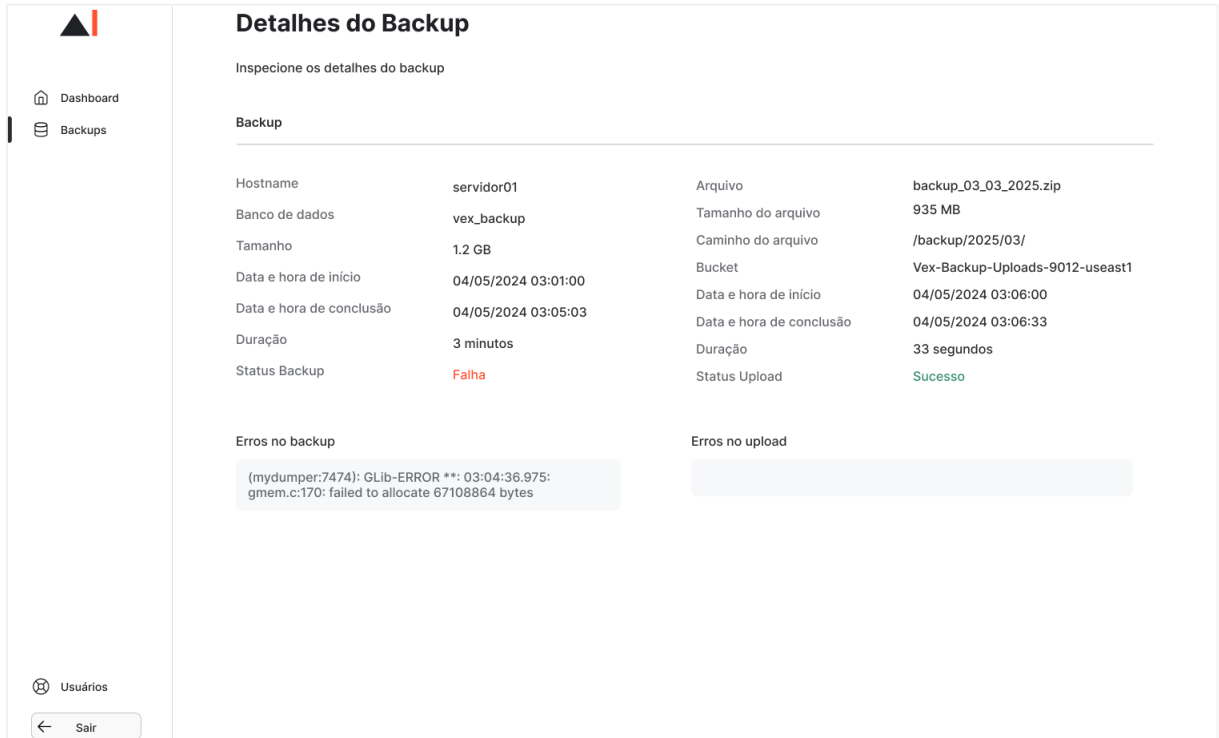
Figura 4 - Protótipo Listar Backup



Fonte: Elaborado pelo autor, 2025.

3.3.1 Detalhes do Backup

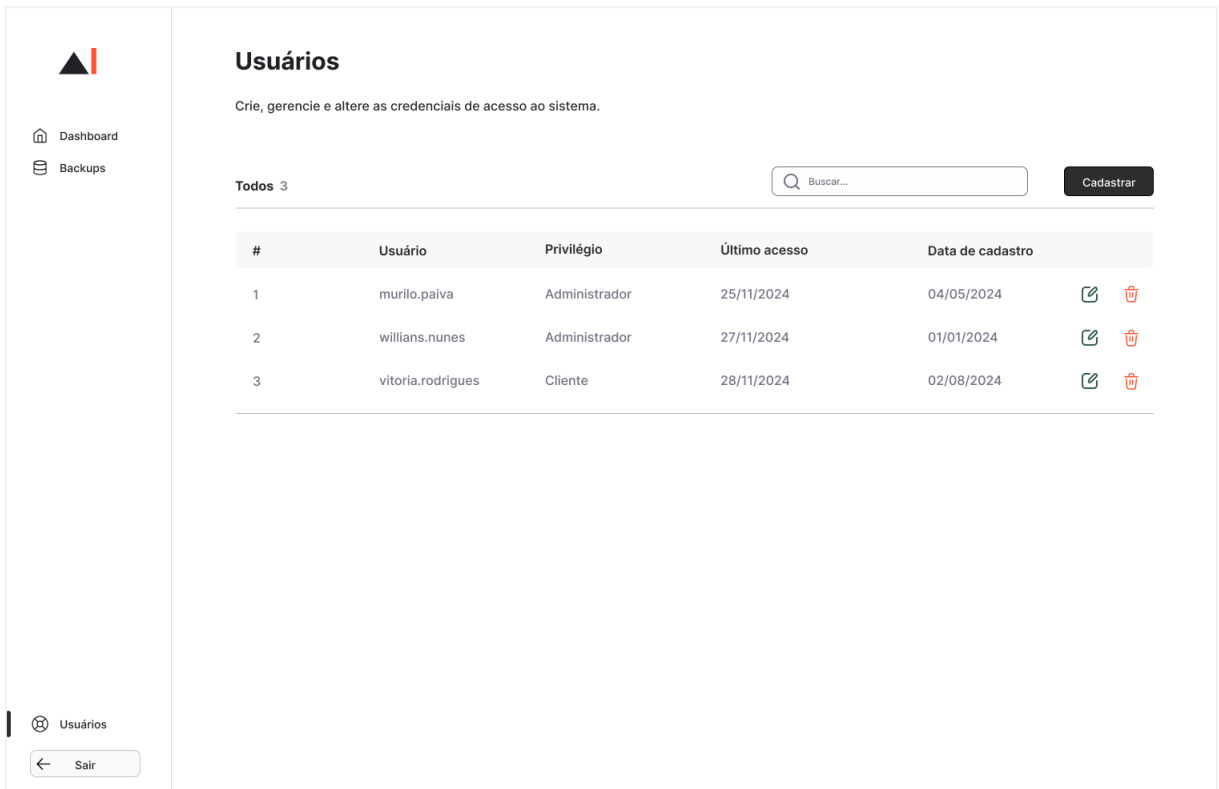
Figura 5 - Protótipo Detalhes do Backup



Fonte: Elaborado pelo autor, 2025.

3.4 RF4 – LISTAR USUÁRIO

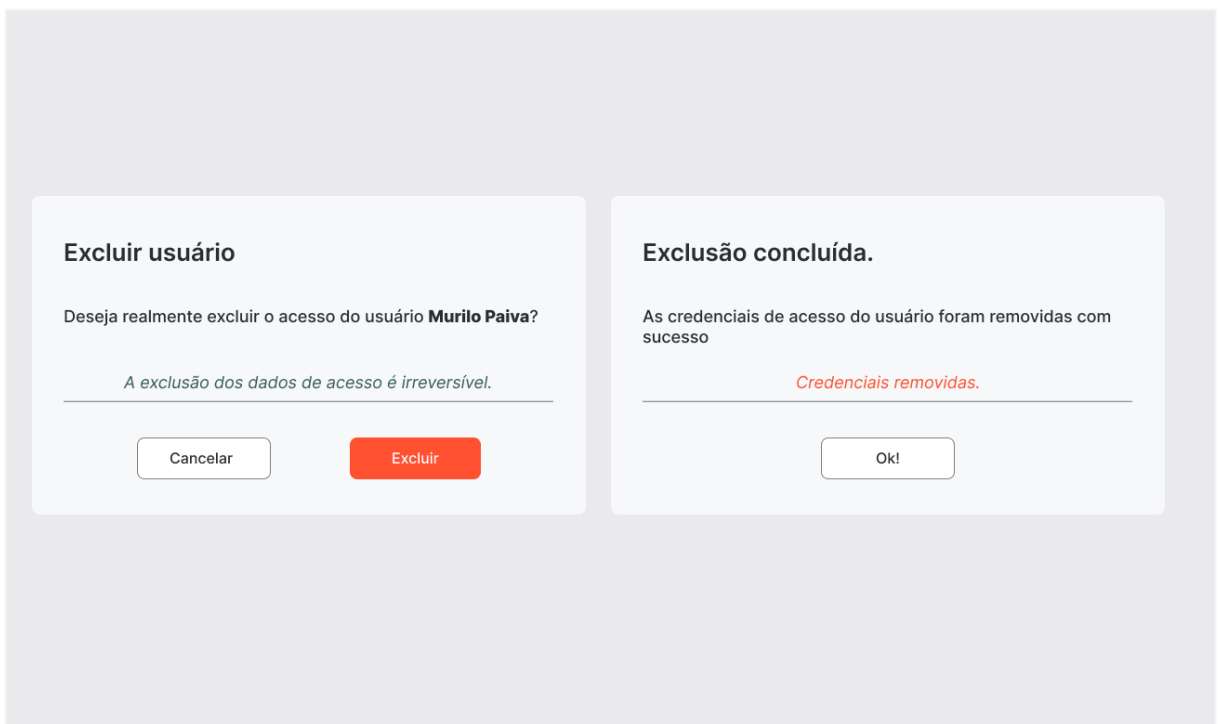
Figura 6 - Protótipo Listar Usuário



Fonte: Elaborado pelo autor, 2025.

3.4.1 Mensagens Listar Usuário

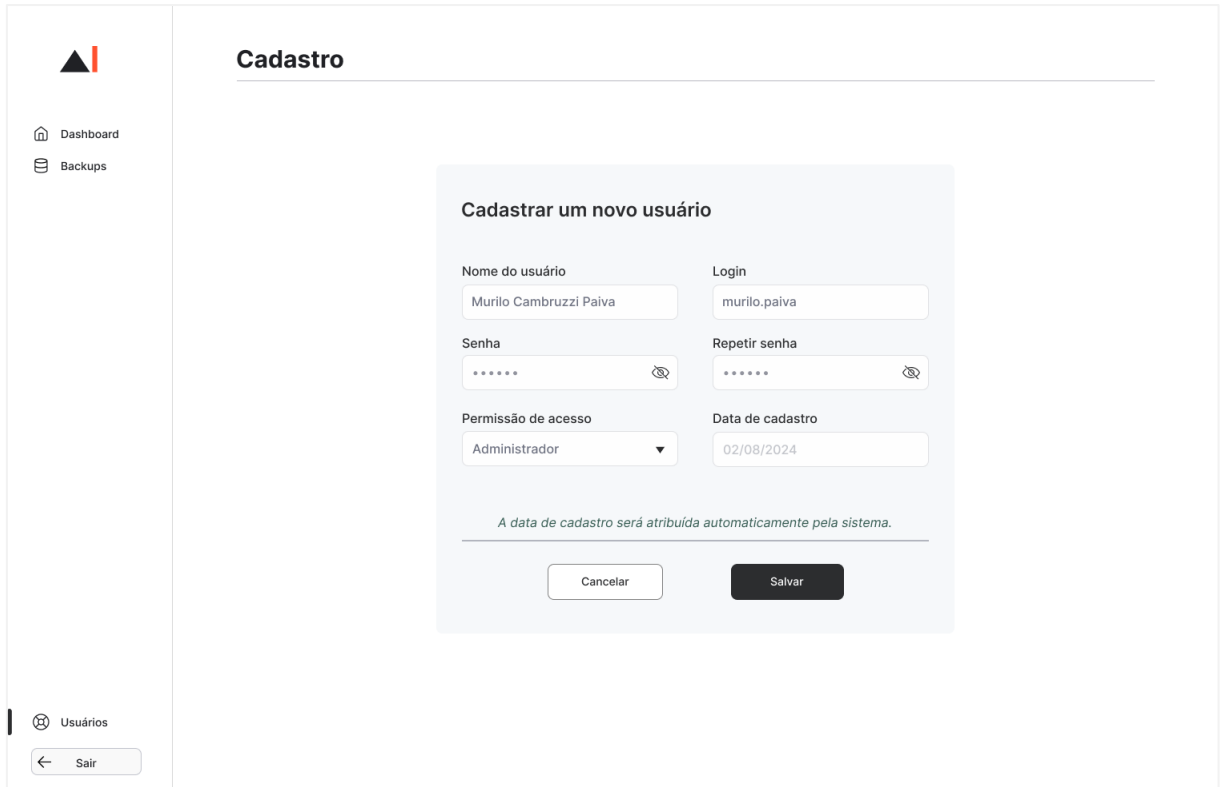
Figura 7 - Protótipo Mensagens Listar Usuário



Fonte: Elaborado pelo autor, 2025.

3.5 RF5 – CADASTRAR USUÁRIO

Figura 8 - Protótipo Cadastrar usuário



O protótipo apresenta uma interface de usuário para o cadastro de um novo usuário. A página é intitulada "Cadastro" e possui um menu lateral com opções para "Dashboard", "Backups", "Usuários" e "Sair". O formulário principal, intitulado "Cadastrar um novo usuário", contém os seguintes campos:

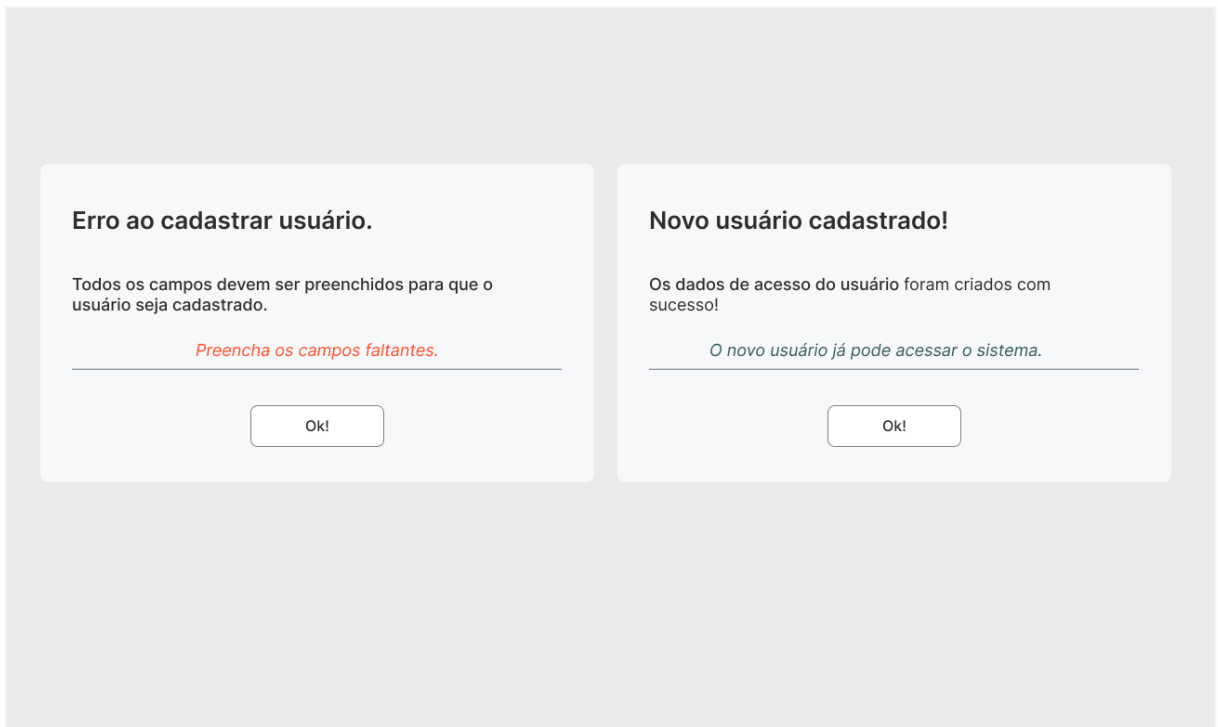
- Nome do usuário:** Campo de texto com o valor "Murilo Cambuzzi Paiva".
- Login:** Campo de texto com o valor "murilo.paiva".
- Senha:** Campo de texto com pontos para ocultar o conteúdo e ícone para alternar visibilidade.
- Repetir senha:** Campo de texto com pontos para ocultar o conteúdo e ícone para alternar visibilidade.
- Permissão de acesso:** Menu suspenso com o valor "Administrador".
- Data de cadastro:** Campo de texto com o valor "02/08/2024".

Abaixo dos campos, há uma mensagem de informação: "A data de cadastro será atribuída automaticamente pela sistema." Na base do formulário, existem dois botões: "Cancelar" e "Salvar".

Fonte: Elaborado pelo autor, 2025.

3.5.1 Mensagens Cadastrar Usuário

Figura 9 - Protótipo Mensagens Cadastrar Usuário



Fonte: Elaborado pelo autor, 2025.

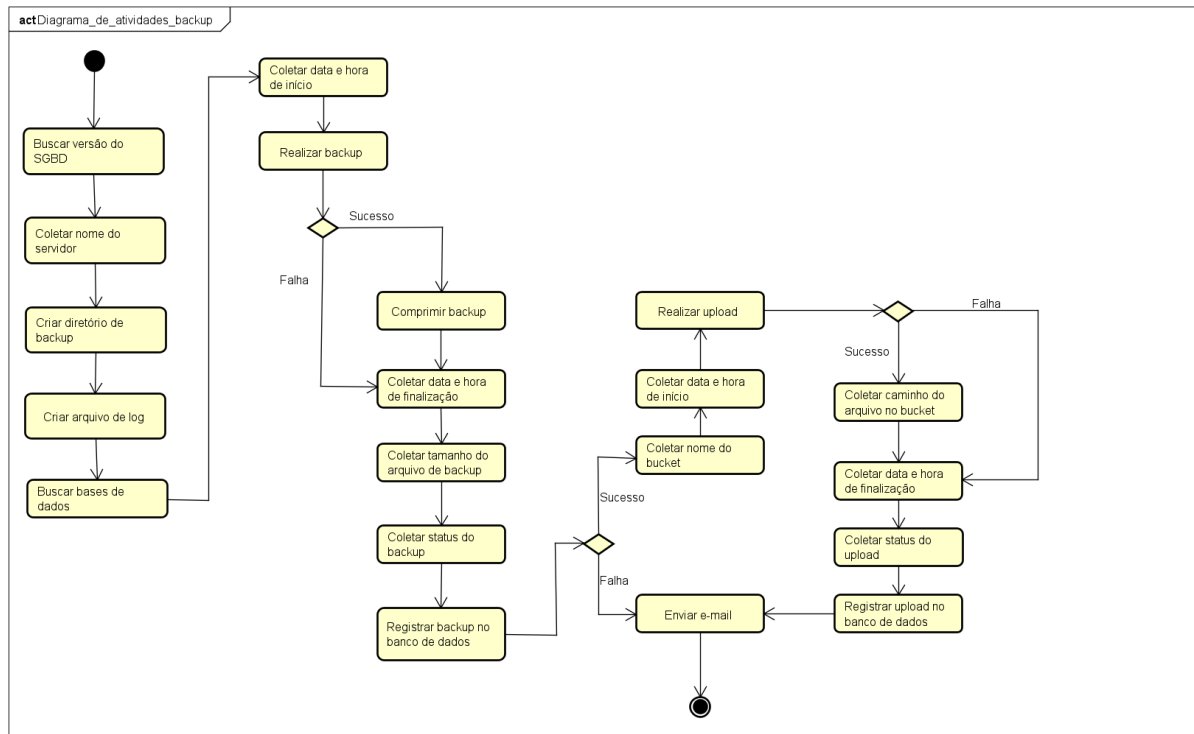
4 FERRAMENTA DE BACKUP

A ferramenta de backup do projeto consistirá em um mecanismo automatizado, construído inteiramente em linguagem de programação Python, sendo capaz de realizar a coleta de informações iniciais (versão do SGBD, nome do servidor, bases de dados e diretório de backup), realizar *backups* lógicos de bases de dados MySQL, comprimir os arquivos resultantes do backup, efetuar o *upload* para um Bucket S3 na AWS. Além disso, seja em caso de falha ou sucesso, a ferramenta dispara notificações por e-mail, assegurando uma maior visibilidade e controle sobre as rotinas, além de manter um registro no banco de dados de cada uma das etapas realizadas.

Essa ferramenta de *backup* poderá ser utilizada de forma pontual, isto é, atividade manualmente sempre que o usuário necessitar ou poderá ser adicionada ao agendador *Cron* de atividades/tarefas presentes em distribuições Linux, viabilizando sua execução de forma recorrente e automatizada.

4.1 DIAGRAMA DE ATIVIDADES

Figura 10 - Diagrama de atividades



Fonte: Elaborado pelo autor, 2025.

4.2 ETAPA DE BACKUP

Abaixo estão listados e descritos os passos que serão realizados na etapa de backup com base no diagrama de atividades do item “4.1 Diagrama de atividades”:

1. **Buscar versão do SGBD:** Neste passo será realizado a coleta e identificação da versão do MySQL em que a ferramenta de backup será executada, os dados serão coletados através da biblioteca *pymysql*.
2. **Coletar o nome do servidor:** Será realizada a identificação do *host* onde foi realizado o backup, através da biblioteca *socket*.
3. **Criar diretório de backup:** Será feita a criação do diretório que receberá os dados dos backups lógicos realizados, por meio da biblioteca *os*.
4. **Criar arquivo de log:** Será criado um arquivo de log para registrar todo o histórico de execução do *backup*, possibilitando a identificação de quaisquer erros durante a execução.
5. **Buscar bases de dados:** Nesta etapa será realizada uma listagem das bases de dados presentes no servidor que não sejam bases de dados de sistema, como (*sys*, *mysql*, *performance_schema*, *information_schema* e *innodb_cluster_metadata*), por meio da biblioteca *pymysql*.
6. **Coletar data e hora de início:** Neste passo, será coletado a data e hora de início do backup para fins de auditoria e controle de tempo de execução, sendo que essa coleta será realizada através da biblioteca *datetime*.
7. **Realizar backup:** A ferramenta de *backup* irá realizar o processo de *backup* lógico das bases de dados listadas anteriormente utilizando a ferramenta Mydumper, esta etapa será orquestrada através da biblioteca *subprocess*.
8. **Coletar data e hora de finalização:** Neste passo, será coletado a data e hora de finalização do backup para complementar o controle de tempo de execução, essa coleta será realizada por intermédio da biblioteca *datetime*.
9. **Realizar compressão:** Após completar o processo de *backup*, os arquivos gerados serão compactados através da ferramenta 7-Zip para reduzir o espaço e disco por meio da biblioteca *subprocess*.
10. **Coletar tamanho do backup:** Com o arquivo de *backup* compactado, será coletado o tamanho do arquivo de gerado por meio da biblioteca *subprocess*.

11. **Coletar status de backup:** Será analisado o log de execução do backup para identificação de erros presentes no backup, esse processo de análise será realizado através da biblioteca *os*.
12. **Registrar backup no banco de dados:** Serão registrados no banco de dados os dados coletados durante a execução do backup, utilizando a biblioteca *pymysql*.

4.3 ETAPA DE UPLOAD

Após o término bem-sucedido do backup, inicia-se a fase de upload para o serviço de armazenamento em nuvem (S3):

1. **Coletar o nome do Bucket:** A ferramenta irá coletar o nome do Bucket S3 que será o destino do arquivo de backup com o auxílio da biblioteca *boto3*.
2. **Coletar data e hora de início:** Neste passo, será coletado a data e hora de início do upload para fins de auditoria e controle de tempo de execução, sendo que essa coleta será realizada através da biblioteca *datetime*.
3. **Realizar upload:** O arquivo compactado será transferido para o bucket S3. Caso haja sucesso, prossegue-se para a próxima etapa; se houver falha, um e-mail de notificação será disparado para o responsável, indicando o status de erro, e o processo é encerrado.
4. **Coletar caminho do arquivo no Bucket:** A ferramenta irá coletar com o auxílio da biblioteca *boto3*, o caminho no Bucket onde o arquivo ficou armazenado.
5. **Coletar data e hora de finalização:** Neste passo, será coletada a data e hora de finalização do upload para complementar o controle de tempo de execução, essa coleta será realizada por intermédio da biblioteca *datetime*.
6. **Coletar status de upload:** Será analisado o log de execução do upload para identificação de erros presentes na transferência do arquivo, esse processo de realizado por intermédio da biblioteca *os*.
7. **Registrar upload no banco de dados:** As informações referentes ao envio (nome do arquivo, caminho no *bucket*, horários de início/fim, status) serão inseridas em uma tabela de logs de upload (*upload_s3_log*), vinculada ao registro de backup que originou o upload por meio de um identificador único., utilizando a biblioteca *pymysql*.

4.4 NOTIFICAÇÕES E TRATAMENTO DE FALHAS

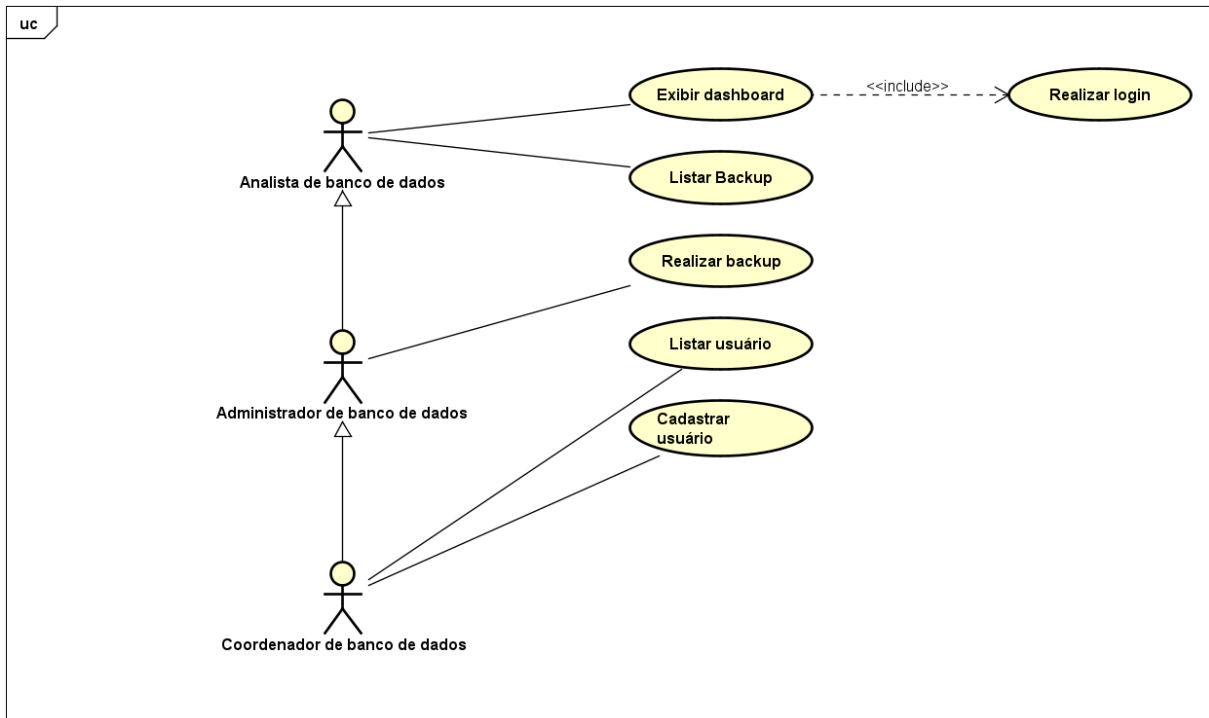
Em qualquer etapa dos processos realizados pela ferramenta em que seja detectado uma falha, a ferramenta irá registrar o problema no arquivo de log, irá atribuir o status de erro (Código 3, conforme item 7.4 e 7.5 do Dicionário de Dados presente neste documento) e irá realizar o envio de uma notificação por e-mail contendo os logs de execução da ferramenta contendo os erros encontrados ao decorrer da execução.

Em casos onde não sejam encontrados erros durante a execução, a notificação por e-mail ainda será enviada, mas com o status de sucesso (código 1, conforme item 7.4 e 7.5 do Dicionário de Dados presente neste documento) e o arquivo de log evidenciando as etapas cumpridas sem incidentes. Deste modo, a ferramenta manterá um histórico transparente de execuções acessível para análise e/ou auditoria.

5 MODELO DE CASO DE USO

5.1 DIAGRAMA DE CASO DE USO

Figura 11 - Diagrama de Caso de Uso



Fonte: Elaborado pelo autor, 2025.

5.2 CASO DE USO EXPANDIDO

5.2.1 Caso de Uso – Realizar login

Caso de Uso: Realizar login (CSU01)

Descrição: O usuário deve preencher os dados para entrar no sistema.

Ator Primário: Analista de banco de dados.

Atores Secundários: Administrador de banco de dados e Coordenador de banco de dados.

Precondições: O usuário precisa acessar a página web do Vex Backup em um navegador web em seu computador.

Fluxo Principal

1. O usuário informar o seu usuário de login no campo “Usuário” e senha no campo “Senha” na interface.
2. O usuário clica no botão “Entrar”.

3. O sistema abre a função “Exibir dashboard”.

Fluxo Alternativo 1:

1. Após o passo 2 o sistema identifica que os dados de acesso estão incorretos (Usuário e Senha) e retorna a mensagem “Os dados informados não estão cadastrados em nossa base de dados. Por favor, tente novamente.”
2. O usuário clica no botão “Ok!”.
3. O sistema retorna para a tela de login.

Fluxo Alternativo 2:

1. Após o passo 2 o sistema identifica que o usuário não preencheu todos os campos, usuário ou senha ou ambos e retorna a mensagem “Todos os campos devem ser preenchidos para que o login possa ser efetuado.”
2. O usuário clica no botão “Ok!”.
3. O sistema retorna para a tela de login.

5.2.2 Caso de Uso – Exibir dashboard

Caso de Uso: Exibir dashboard (CSU02)

Descrição: O sistema irá consultar na base de dados de backups e uploads e irá exibir os dados em um dashboard.

Ator Primário: Analista de banco de dados.

Atores Secundários: Administrador de banco de dados e Coordenador de banco de dados.

Precondições: O usuário precisa realizar o login na ferramenta.

Fluxo Principal

4. O sistema consulta o total de backups realizados e os separa por status e exibe no dashboard.
5. O usuário visualiza dos dados exibidos no dashboard.

5.2.3 Caso de Uso – Listar backup

Caso de Uso: Listar Backup (CSU03)

Descrição: O sistema deverá consultar os dados dos backups e uploads e exibir os metadados em formato de tabela na interface.

Ator Primário: Analista de banco de dados.

Atores Secundários: Administrador de banco de dados e Coordenador de banco de dados.

Precondições: O usuário precisa realizar o login na ferramenta.

Fluxo Principal

1. O usuário clica no botão “Backups” presente no menu lateral esquerdo do sistema.
2. O sistema abre a função listar backups.
3. O usuário informa os dados que deseja buscar na caixa de busca presente na interface.
4. O sistema realiza a consulta nos registros de backups e uploads e exibe os dados em formato de tabela, com as seguintes colunas: Hostname, Status, Tamanho, Data de início, data de encerramento e Status upload.

5.2.4 Caso de Uso – Realizar backup

Caso de Uso: Realizar Backup (CSU04)

Descrição: O sistema deverá realizar o backup das bases de dados do MySQL presentes no servidor em que está sendo executado.

Ator Primário: Analista de banco de dados.

Atores Secundários: Administrador de banco de dados e Coordenador de banco de dados.

Precondições: O usuário precisar ter acesso ao servidor onde está instalado o banco de dados que deseja realizar o backup.

Fluxo Principal

1. O usuário executa o comando “python3 backup.py”.
2. O sistema inicia o processo de backup das bases de dados presentes o ambiente, realizando a inserção dos metadados na base de dados, compactando o backup com 7-zip, realiza o upload do arquivo de backup para o Bucket S3 na AWS e por fim envia um e-mail contendo os logs de execução.

Fluxo Alternativo 1:

1. O usuário realiza o agendamento de uma atividade no agendado de tarefas do sistema operacional para que execute o comando “python3 backup.py” no(s) horário(s) que desejar.
2. O sistema segue o fluxo descrito no passo 2 de forma automática.

5.2.5 Caso de Uso – Listar usuário

Caso de Uso: Listar usuário (CSU05)

Descrição: O sistema deverá consultar os dados dos usuários cadastrados no sistema.

Ator Primário: Coordenador de banco de dados.

Atores Secundários: Não há.

Precondições: O usuário precisa realizar o login na ferramenta e possuir o privilégio de administrador da ferramenta.

Fluxo Principal

1. O usuário clica no botão “Usuários” presente no menu lateral esquerdo do sistema.
2. O sistema abre a função listar usuários.
3. O usuário informa os dados que deseja buscar na caixa de busca presente na interface.
4. O sistema realiza a consulta nos registros de usuários e exibe os dados em formato de tabela, com as seguintes colunas: Usuário, Privilégio, Último acesso, Data de cadastro.

Fluxo Alternativo 1:

1. Após o passo 4, o usuário clica no botão com ícone de lixeira posicionado ao lado da coluna “Data de cadastro”.
2. O sistema retorna a mensagem “Deseja realmente excluir o acesso do usuário?”.
3. O usuário clica em “Sim” na mensagem.
4. O sistema remove os dados do usuário na base de dados e retorna a mensagem “As credenciais de acesso do usuário foram removidas com sucesso”.
5. O usuário clica no botão “Ok!”.
6. O sistema retorna ao passo 4.

Fluxo Alternativo 2:

1. Após o passo 4, o usuário clica no botão com ícone de lápis posicionado ao lado da coluna “Data de cadastro”.
2. O sistema abre a funcionalidade de cadastro de usuário com os dados do usuário selecionado preenchido e liberados para edição.
3. O usuário clica no botão “Salvar”.
4. O sistema exibe a mensagem “Os dados do usuário foram alterados com sucesso.”.
5. O usuário clica no botão “Ok!”.
6. O sistema retorna ao passo 4.

5.2.6 Caso de Uso – Cadastrar usuário

Caso de Uso: Listar usuário (CSU06)

Descrição: O usuário deve preencher os campos para adicionar um novo usuário ao sistema.

Ator Primário: Coordenador de banco de dados.

Atores Secundários: Não há.

Precondições: O usuário precisa realizar o login na ferramenta e possuir o privilégio de administrador da ferramenta.

Fluxo Principal

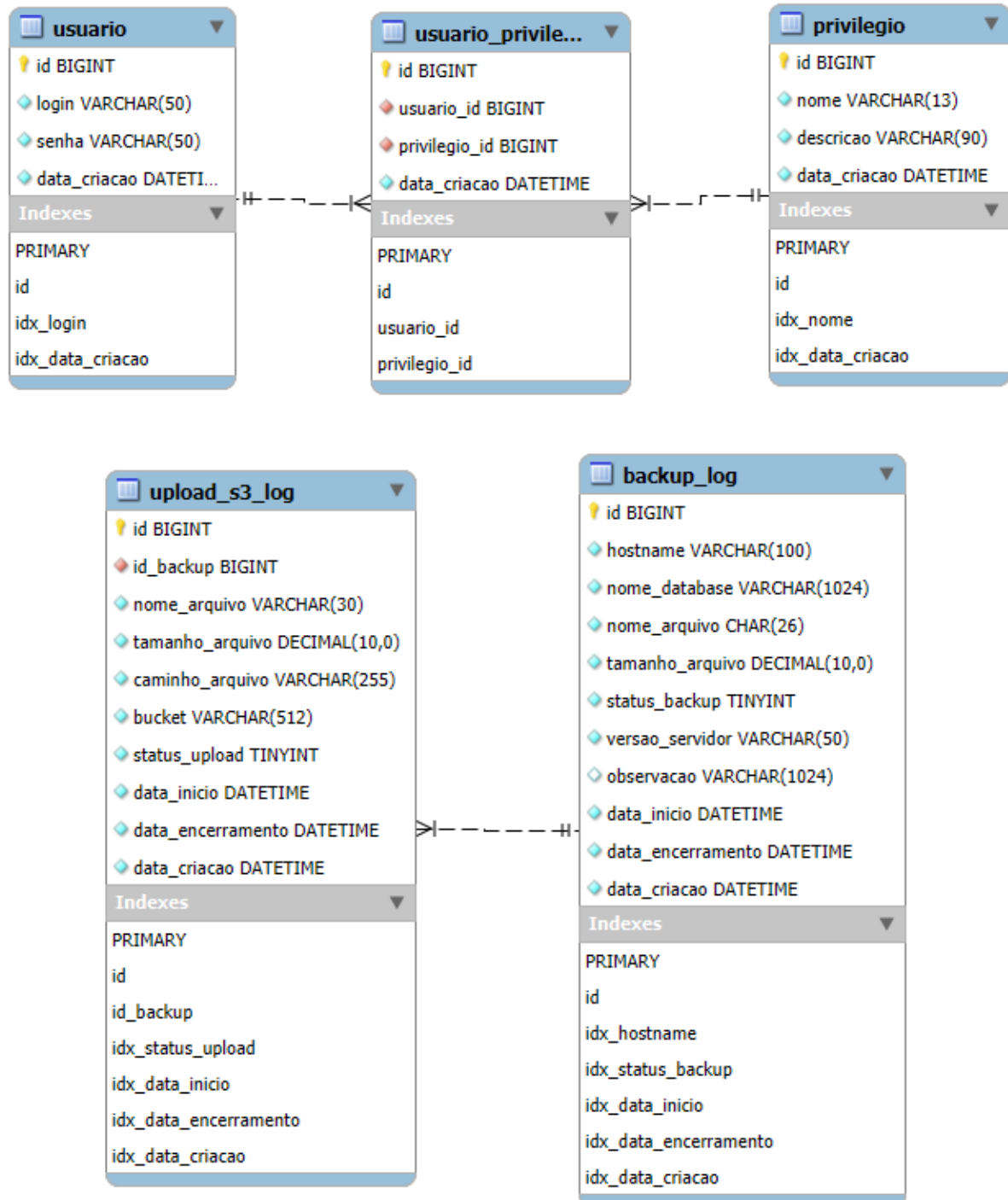
1. O usuário clica no botão “Cadastrar”.
2. O sistema abre a funcionalidade “Cadastra usuário” com os campos habilitados para edição.
3. O usuário preenche os campos: Nome do usuário, Login, Senha, Repetir Senha e Permissão de acesso e clica no botão “Salvar”.
4. O sistema realizar a inserção dos dados do novo usuário no banco de dados e exibe a mensagem “Os dados de acesso do usuário foram criados com sucesso!”.
5. O usuário clica no botão “Ok!”.
6. O sistema abre a funcionalidade de “Listar usuário”.

Fluxo Alternativo 1:

1. Após o passo 3, o sistema identifica que o não foram preenchidos todos os campos e exibe a mensagem “Todos os campos devem ser preenchidos para que o usuário seja cadastrado.”.
2. O usuário clica no botão “Ok!”.
3. O sistema retorna ao passo 3.

6 DIAGRAMA DE ENTIDADE-RELACIONAMENTO

Figura 12 - Diagrama de Entidade-Relacionamento



Fonte: Elaborado pelo autor, 2025.

7 DICIONÁRIO DE DADOS

7.1 TABELA USUÁRIO

Quadro 14 - Catálogo de dados da tabela usuario

Campo	Tipo	Descrição	Restrições
id	<i>bigint unsigned</i>	Identificador único do registro.	<i>primary key, not null, unique key (id)</i>
login	<i>varchar(50)</i>	Login do usuário.	<i>not null, key (idx_login)</i>
senha	<i>varchar(50)</i>	Senha do usuário.	<i>not null</i>
data_criacao	<i>datetime</i>	Data de criação do registro.	<i>not null, default current_timestamp, key (idx_data_criacao)</i>
Descrição da tabela: Armazena dados dos usuários que utilizam a aplicação.			

Fonte: Elaborado pelo autor, 2025.

7.2 TABELA PRIVILEGIO

Quadro 15 - Catálogo de dados da tabela usuario

Campo	Tipo	Descrição	Restrições
id	<i>bigint unsigned</i>	Identificador único do registro.	<i>primary key, not null, unique key (id)</i>
nome	<i>varchar(13)</i>	Código do privilégio.	<i>not null, key (idx_nome)</i>
descricao	<i>varchar(90)</i>	Descrição do privilégio.	<i>not null</i>
data_criacao	<i>datetime</i>	Data de criação do registro.	<i>not null, default current_timestamp, key (idx_data_criacao)</i>
Descrição da tabela: Armazena dados dos privilégios de acesso da aplicação.			

Fonte: Elaborado pelo autor, 2025.

7.3 TABELA USUARIO_PRIVILEGIO

Quadro 16 - Catálogo de dados da tabela usuario_privilegio

Campo	Tipo	Descrição	Restrições
-------	------	-----------	------------

id	<i>bigint unsigned</i>	Identificador único do registro.	<i>primary key, not null, unique key (id)</i>
usuario_id	<i>bigint unsigned</i>	ID do usuário.	<i>not null, foreign key (usuario_id) references usuario (id)</i>
privilegio_id	<i>bigint unsigned</i>	ID do privilégio.	<i>not null, foreign key (privilegio_id) references privilegio (id)</i>
data_criacao	<i>datetime</i>	Data de criação do registro.	<i>not null, default current_timestamp, key (idx_data_criacao)</i>
Descrição da tabela: Relaciona usuários com seus privilégios.			

Fonte: Elaborado pelo autor, 2025.

7.4 TABELA BACKUP_LOG

Quadro 17 - Catálogo de dados da tabela backup_log

Campo	Tipo	Descrição	Restrições
id	<i>bigint unsigned</i>	Identificador único do registro.	primary key, not null, unique key (id)
hostname	<i>varchar(100)</i>	Nome do host.	not null, key (idx_hostname)
nome_database	<i>varchar(1024)</i>	Database(s) presentes no backup.	not null
nome_arquivo	<i>char(26)</i>	Nome do arquivo de backup.	not null
tamanho_arquivo	<i>decimal(18,2)</i>	Tamanho do backup em MB.	not null
status_backup	<i>tinyint unsigned</i>	Status do backup (1 = Sucesso, 2 = Warning, 3 = Erro).	not null, key (idx_status_backup)
versao_servidor	<i>varchar(50)</i>	Versão do servidor onde foi retirado o backup.	not null
observacao	<i>varchar(1024) (NULL)</i>	Observações relacionadas ao backup.	default null
data_inicio	<i>datetime</i>	Data e hora de início do backup.	not null, key (idx_data_inicio)
data_encerramento	<i>datetime</i>	Data e hora da finalização do backup.	not null, key (idx_data_encerramento)

data_criacao	<i>datetime</i>	Data de criação do registro.	not null, default <i>current_timestamp</i> , key (idx_data_criacao)
Descrição da tabela: Armazena dados dos backups realizados.			

Fonte: Elaborado pelo autor, 2025.

7.5 TABELA UPLOAD_S3_LOG

Quadro 18 - Catálogo de dados da tabela upload_s3_log

Campo	Tipo	Descrição	Restrições
id	<i>bigint unsigned</i>	Identificador único do registro.	<i>primary key, not null, unique key (id)</i>
backup_id	<i>bigint unsigned</i>	ID do backup realizado, referenciando backup_log (id).	<i>not null, foreign key (backup_id) references backup (id)</i>
nome_arquivo	<i>varchar(30)</i>	Nome do arquivo de backup compactado.	<i>not null</i>
tamanho_arquivo	<i>decimal(18,2)</i>	Tamanho do arquivo de backup compactado em MB.	<i>not null</i>
caminho_arquivo	<i>varchar(255)</i>	Caminho do arquivo dentro do <i>Bucket S3</i> .	<i>not null</i>
bucket	<i>varchar(512)</i>	Nome do <i>Bucket S3</i> .	<i>not null</i>
status_upload	<i>tinyint</i>	<i>Status do upload</i> (1 = Sucesso, 2 = Alerta, 3 = Erro).	<i>not null, key (idx_status_upload)</i>
data_inicio	<i>datetime</i>	Data e hora de início do upload.	<i>not null, key (idx_data_inicio)</i>
data_encerramento	<i>datetime</i>	Data e hora da finalização do upload.	<i>not null, key (idx_data_encerramento)</i>
data_criacao	<i>datetime</i>	Data de criação do registro.	<i>not null, default current_timestamp, key (idx_data_criacao)</i>
Descrição da tabela: Armazena dados dos uploads dos backups para o <i>Bucket S3</i> .			

Fonte: Elaborado pelo autor, 2025.