RECAM
Revista Camalotes

***KOALA*: Implementação tecnológica de cibersegurança de um projeto *webapp*.**

***KOALA*: Cybersecurity technological implementation of a webapp project.**

***KOALA*: Implementación tecnológica de ciberseguridad de un proyecto webapp.**

Thales Tayson do Nascimento Vargas[1]

Lucas Almeida Tiburtino da Silva[2]

Matheus Fernandes de Figueiredo[3]

Prof. Me. Ana Claudia De Oliveira Pedro Andrêo [4]

Prof. Me. Edilene Aparecida Veneruchi de Campos [5]

Prof. João Carlos Domingos [6]

RESUMO: O sistema KOALA visa o uso interativo de metodologias ágeis, tendo como principal elemento o *Kanban*, por ser interativo e de fácil manuseio, oferecendo aos usuários a disponibilidade de uso, com o acesso à internet. Nesse sistema, a implementação da cibersegurança é importante, pois sua falta pode ocasionar falhas inesperadas ao serviço, brechas na segurança da transmissão de informações, impedindo o comportamento adequado da plataforma, atrapalhando na confiabilidade, integridade e disponibilidade. Esta obra foi elaborada em Campo Grande, Mato Grosso do Sul, na graduação de Análise e Desenvolvimento de Sistemas. O objetivo deste artigo é de refletir sobre os possíveis usos de implementação da cibersegurança, com o intuito de melhorar a segurança e minimizar as vulnerabilidades, propondo uma visão ampla referente à tecnologia de segurança e as demais estratégias, logo, ajudando o leitor a

[1] Acadêmico do curso de Análise e Desenvolvimento de Sistemas da Faculdade Insted. ORCID iD: https://orcid.org/0000-0001-7570-7367. E-mail: taysonvargas@outlook.com.
[2] Acadêmico do curso de Análise e Desenvolvimento de Sistemas da Faculdade Insted. ORCID iD: https://orcid.org/0000-0001-7347-810X. E-mail: lucas.almida.da.silva@gmail.com.
[3] Acadêmico do curso de Análise e Desenvolvimento de Sistemas da Faculdade Insted. ORCID iD: https://orcid.org/0000-0002-3281-0222. E-mail: matheus.figueiredo1999@yahoo.com.
[4] Ana Claudia De Oliveira Pedro Andrêo é professora da Faculdade Insted. ORCID iD: https://orcid.org/0009-0000-6593-4259. E-mail: anaclaudia.andreo@insted.edu.br.
[5] Edilene Aparecida Veneruchi de Campos é professora da Faculdade Insted. ORCID iD: https://orcid.org/0000-0002-4427-608X. E-mail: edilene.veneruchi@insted.edu.br.
[6] João Carlos Domingos é professor da Faculdade Insted. ORCID iD: https://orcid.org/0009-0009-7364-8127. E-mail: joao.domingos@insted.edu.br.

conhecer algumas estratégias de segurança. A metodologia foi elaborada transversalmente com pesquisas na base de dados científica oficial do Google Acadêmico, buscando por bibliografias e demais artigos, fazendo o uso da leitura objetiva aos temas referentes ao uso de tecnologias de segurança, gêneros de ataques, proteção e vulnerabilidades. Os resultados demonstram que as implementações fizeram com que os códigos melhorassem a blindagem do sistema, impedindo ataques.

PALAVRAS-CHAVE: Cibersegurança. Sistema *online*. Cibersegurança de *sites*. Confiabilidade, integridade e disponibilidade de *sites*.

ABSTRACT: The KOALA system aims at the interactive use of agile methodologies, having Kanban as its main element, as it is interactive and easy to handle, offering users the availability of use with internet access. In this system, the implementation of cybersecurity is important, as its lack can cause unexpected failures in the service, breaches in the security of information transmission, preventing the proper behavior of the platform, disturbing reliability, integrity and availability. This work was elaborated in Campo Grande, Mato Grosso do Sul, in the graduation of Systems Analysis and Development. The purpose of this article is to reflect on the possible uses of cybersecurity implementation, with the aim of improving security and minimizing vulnerabilities, proposing a broad view regarding security technology and other strategies, thus helping the reader to know some security strategies. The methodology was elaborated transversally with searches in the official scientific database of Google Scholar, searching for bibliographies and other articles, making use of objective reading of themes related to the use of security technologies, types of attacks, protection and vulnerabilities. The results demonstrate that the implementations made the codes improve the shielding of the system, preventing attacks.

KEYWORDS: Cybersecurity. Online system. Website cybersecurity. Website reliability, integrity, and availability.

RESUMEN: El sistema KOALA apunta al uso interactivo de metodologías ágiles, teniendo como elemento principal a Kanban, por ser interactivo y de fácil manejo, ofreciendo a los usuarios la disponibilidad de uso con acceso a internet. En este sistema, la implementación de la ciberseguridad es importante, ya que su falta puede provocar fallas inesperadas en el servicio, brechas en la seguridad de la transmisión de la información, impidiendo el correcto comportamiento de la plataforma, perturbando la confiabilidad, integridad y disponibilidad. Este trabajo fue elaborado en Campo Grande, Mato Grosso do Sul, en la graduación de Análisis y Desarrollo de Sistemas. El propósito de este artículo es reflexionar sobre los posibles usos de la implementación de la ciberseguridad, con el objetivo de mejorar la seguridad y minimizar las vulnerabilidades, proponiendo una visión amplia en cuanto a la tecnología de seguridad y otras estrategias, ayudando así al lector a conocer algunas estrategias de seguridad. La metodología se elaboró de manera transversal con búsquedas en la base de datos científica oficial de Google Scholar, búsqueda de bibliografías y otros artículos, haciendo uso de lectura objetiva de temas relacionados con el uso de tecnologías de seguridad, tipos de ataques, protección y vulnerabilidades. Los resultados demuestran que las implementaciones hicieron que los códigos mejoraran el blindaje del sistema, previniendo ataques.

PALABRAS CLAVE: Ciberseguridad. Sistema en línea. Ciberseguridad del sitio web. Confiabilidad, integridad y disponibilidad del sitio web.

INTRODUÇÃO

As evoluções das fábricas trouxeram revoluções, inclusive na área de gerenciamento, as indústrias mostraram que são capazes de criar produtos em linhas de montagem seguindo a tendência do mercado ou de entregar os produtos conforme o pedido dos clientes. Os dois modelos citados são o Fordismo e Toyotismo (BEZERRA, 2020), ambos necessitam de um rígido controle de logística pois, com o controle correto das atividades, o gestor poderá ter a ampla visão de cada etapa da atividade, proporcionando redução no tempo de produção ou de serviços, bem como obter análises comportamentais dos funcionários, resultando em melhora nos produtos ou serviços prestados.

Embora o controle de atividades tenha diversos benefícios, por outro lado o descontrole acarretará perda de novos serviços, vendas sem registros, gastos adicionais com materiais em excesso, falta de visão do progresso das atividades, produção de itens danificados ou defeituosos e horas desperdiçadas em resoluções de problemas que poderiam ter sidos corrigidos durante a produção inicial.

A justificativa encontrada pela equipe que redigiu este texto ressalta na construção de um sistema *online* denominado de KOALA (*Kanban; Organized; Assistance; Less work, more Agility*), que visa o uso interativo de metodologias ágeis, tendo como principal elemento o diagrama de *kanban*, por ser o mais interpretativo e de fácil manuseio, oferecendo aos usuários a disponibilidade de uso em qualquer lugar com acesso à rede mundial de computadores.

A metodologia foi elaborada utilizando pesquisas na base de dados científicos oficial do Google Acadêmico e demais artigos não oficiais de *sites online*, utilizando leitura objetivada e aos temas referentes ao uso de tecnologias de segurança, gêneros de ataques, proteção e vulnerabilidades. A seleção das

ferramentas para o desenvolvimento do projeto ocorreu a partir das análises de necessidades em sessões de reuniões com a equipe.

Assim, com a breve introdução, segue a definição do desenvolvimento, uma abordagem completa da fundamentação teórica ao desenvolvimento.

DESENVOLVIMENTO: Explorando Conceitos Essenciais de Segurança na Web

Antes de tudo, inicia-se com uma dúvida, o que é segurança na *internet*? De acordo com (CLEMENTE, 2019), a segurança do *site* inclui quaisquer ações ou ferramentas adotadas para evitar a exposição de informações ou ataques de criminosos. Essas medidas também ajudam a proteger os usuários, como clientes de comércio eletrônico, leitores de *blogs*, e até mesmo *hosts*.

Existem diversos estilos de ataques, com finalidades idênticas ou distintas. Alguns desses ataques poderão ser de grande ameaça ao sistema KOALA e, portanto, serão retratados nesse campo de estudo, com a finalidade de embarcar possíveis soluções para "barrá-los" e proteger o sistema.

2.1 Estudo sobre gêneros de ataques

Um gênero bem reconhecido pela mídia é a negação de serviço, e a versão mais complexa é conhecida como negação de serviço distribuído. Ambos seguem o intuito de parar um serviço, seja *web* ou *desktop*.

O ataque de negação de serviço DoS (*Denial of Service*), tem como foco tornar recursos do sistema indisponíveis para os usuários, manipulando pacotes de rede, programação, lógica ou recursos de manipulação de vulnerabilidades. Se um serviço receber um número alto e indeterminado de solicitações (de modo comum, causadas por usuários falsos - *bots*), ele poderá deixar de estar disponível para usuários legítimos.

Da mesma forma, um serviço pode ser interrompido se uma vulnerabilidade de programação for explorada ou a maneira como o serviço trabalha com os recursos for acessada por diversos usuários simultaneamente

excedendo o limite de acessos na margem de capacidade dos servidores. Sendo assim, há o risco de o invasor injetar e executar código arbitrário enquanto realiza o ataque DoS para acessar informações confidenciais ou executar comandos no servidor. Os ataques de negação de serviço degradam de forma significativa a qualidade do serviço experimentada por usuários autênticos. Esses ataques introduzem atrasos de resposta, perdas excessivas e interrupções de serviço, resultando em impacto direto na disponibilidade (OWASP, 2021a).

Em resumo, esse gênero de ataque poderá travar ou deixar inativo serviços vitais do sistema KOALA, distribuindo uma série de sobrecarga de requisições aos serviços *web,* impedindo que outros usuários possam acessar o sistema.

> Outro modelo de ataque, é o Script entre Sites (XSS, também conhecido por Cross-site Scripting). Essa vulnerabilidade permite que o criminoso adicione scripts em páginas webs – geralmente na linguagem JavaScript. Quando outros usuários carregarem as páginas afetadas, os scripts do invasor serão executados, permitindo que o invasor roube cookie-tokens de sessão e altere o conteúdo da página por meio da manipulação do DOM (Document Object Model) ou redirecione o navegador para páginas de terceiros. As vulnerabilidades XSS geralmente ocorrem quando um aplicativo utiliza a entrada de dados do usuário sem passar por um processo de validação – filtro que irá verificar a consistência dos dados (ANDESON, 2022).

Portanto, ataques do tipo XSS, podem fazer com que o criminoso tenha acesso aos *cookies* de autorização do usuário. Assim, o invasor poderá acessar o *site* se passando pelo usuário e, portanto, acessar detalhes do cartão de crédito, ver detalhes de contato, alterar senhas entre outras funções cruciais.

> Um modelo parecido com o XSS é a injeção SQL (Structured Query Language). Esse ataque consiste na adição ou "injeção" de uma consulta SQL através dos dados de entrada do cliente (normalmente em barras de pesquisa de consultas, como buscadores de nomes). Uma exploração de injeção SQL bem planejada pode criar, ler, modificar ou excluir dados confidenciais. Os ataques de injeção de SQL permitem que os invasores falsifiquem a identidade, adulterem os dados existentes, causem problemas de repúdio, como anulação de transações ou alteração de saldos, permitam a divulgação completa de todos os dados no sistema, destruam os dados ou os tornem indisponíveis e, no pior dos casos, tornem-se administradores do servidor de banco de dados (OWASP, 2021b).

É comum que *Frameworks webs* cuidem do caractere que está sendo consultado. O *Django* garante que todos os dados do usuário, passados para os conjuntos de consultas sejam escapados – técnica que adiciona barra invertida na frente do caractere de consulta " \' ".

Um dos ataques mais temidos é a Falsificação de Solicitação Entre *Sites* (CSRF). Esse ataque permite aos criminosos acesso a transferências de formulários (métodos *GET* e *POST*). Esse processo é exemplificado como:

> Um E-mail, ou página web com uma notícia falsa ou imitação de uma página, assim que o usuário a acessa, poderá ser enviado um formulário ao site real. E se o usuário tiver acesso token salvo ao site real, os cookies permitirão que o formulário seja autenticado com os dados do usuário (OWASP, 2021c).

Esse ataque permite ao criminoso acesso ou modificação de dados, sem a necessidade de ter, de forma direta, as credenciais do usuário. Nota-se, o formulário é preenchido com dados estratégicos voltados para manipulação indevida do sistema, sem que o usuário perceba. Algo similar poderá ocorrer com APIs (*Application Programming Interface*) desprotegidas ou mal projetadas.

O *Django* utiliza CSRF *token*, um código que é inserido junto ao formulário no momento em que a página HTML é requisitada, processada no servidor e enviada ao cliente, fazendo com que o formulário sem a adição deste código seja rejeitado pelo servidor. Uma estratégia simples, que evita o processamento de formulários não autorizados.

2.2 Entendimento Sobre Metodologias Para Gestão de Riscos

A fundamentação de segurança de serviços *webs* são de impedir esses gêneros conhecidos e desconhecidos de ataques (assim como visto no tópico 2.1). Uma definição mais formal de segurança de sites é o ato ou prática de proteger um serviço seja ele *web*, *software* ou *mobile* de acesso, uso, modificação ou destruição não autorizados (MDN, 2022).

E, para isso, a segurança eficaz desses sistemas requerem trabalho de design completo, tendo o planejamento sempre focado no usuário e em desenvolver um ambiente seguro para o serviço.

Sendo assim, uma das formas de planejar a lógica por trás da segurança é a utilização de diagramas de sequência.

> O diagrama de sequência, é uma ferramenta UML (Unified Modeling Language, do inglês, Linguagem de Modelagem Unificada). O principal objetivo é demonstrar em uma linha do tempo as interações entre os objetos de uma determinada cena representada por um diagrama. É comum que os diagramas de sequência sejam criados a partir de diagramas de caso de uso para descrever como serão as interações ou mensagens entre cada objeto ou elemento do sistema. O gráfico possui dois eixos: o eixo vertical representa a ordem das mensagens e o ciclo de vida dos objetos, e o eixo horizontal representa quais objetos participam do gráfico (VIEIRA, 2013).

2.3 Estratégias e Implementações

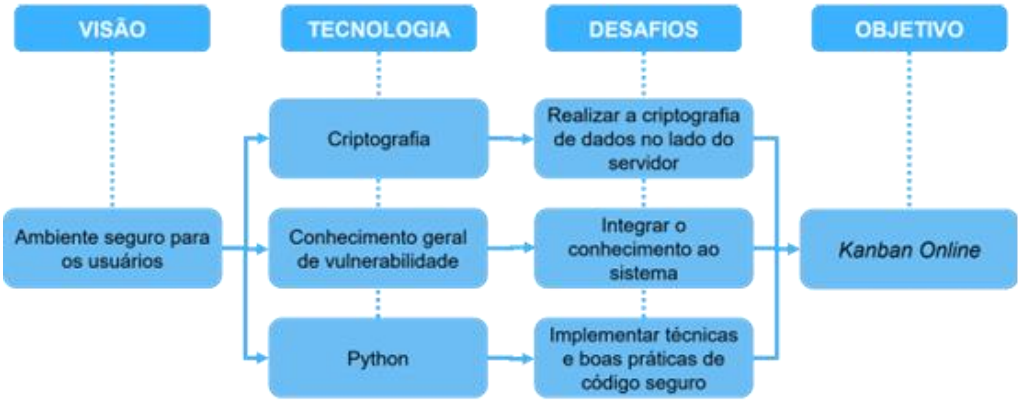
Esta seção foi destinada para revelar como será trabalhada a segurança e a privacidade da informação no projeto, seguindo os moldes dos tópicos anteriores. E, de forma geral, a motivação que levou ao uso dos processos que fazem parte da infraestrutura de segurança.

Para o desenvolvimento do objetivo geral de privacidade de dados e o gerenciamento de riscos cibernéticos – e, de forma posterior, para a implementação, necessitou-se encontrar respostas diante de três questionamentos:

- O que se planeja atingir com a privacidade de dados e o gerenciamento de riscos cibernéticos?
- Quais meios tecnológicos necessários para o desenvolvimento da privacidade de dados e o gerenciamento de riscos cibernéticos?
- Quais são os desafios da arquitetura da privacidade de dados e o gerenciamento de riscos cibernéticos no sistema KOALA?

Para responder os questionamentos e melhor esclarecimento, foi desenvolvido um diagrama, colocando as principais hipóteses obtidas através das pesquisas e conversação durante as sessões de reuniões.

Figura 1 - Diagrama de Objetivo Geral



Fonte: Elaborado pelos autores.

Conforme demonstrado na figura 1 (Diagrama de Objetivo Geral), a coluna "visão" revela o esperado ao concluir a implementação de segurança, e, portanto, um sistema seguro e ideal para usuários, com a intenção de impedir ataques ao ambiente. Já a coluna "objetivo" reflete o 'alvo' que desejasse alcançar, no caso, preparar o ambiente para proteger o sistema e os dados dos usuários do KOALA *online*. Entretanto, para sair do ponto de visão e atingir os objetivos, a coluna "tecnologia" serve para apoiar a lista tecnológica que será implementada no sistema KOALA, isto é: criptografia, uma forma ideal para impedir que terceiros não sejam capazes de ler informações sigilosas, como senhas salvas no banco de dados, além de permitir a comunicação segura de um ponto 'A' ao ponto 'B' através de mensagens cifradas; Conhecimento geral de vulnerabilidade, essencial para fazer análises do projeto por procuras de brechas na segurança, sejam elas óbvias ou não. E por fim, *Python*, como principal linguagem de programação do sistema, sendo o ponto onde todo o controle e gerenciamento da plataforma estará concentrado, e, portanto, deve-se seguir normas e boas práticas por parte dos desenvolvedores para impedir que ocorram erros de lógica capazes de serem usados futuramente por usuários mal-intencionados, e assim, evitar

vulnerabilidades desconhecidas pelo fabricante, porém conhecidas por invasores (vulnerabilidades de dia zero).

Para a resolução dos desafios, existem três possíveis soluções, seguindo a ordem crescente da coluna "desafios":

a) Utilizar o entendimento da infraestrutura do banco de dados, para criar um modelo de criptografia;

b) Usufruir do conhecimento de vulnerabilidade para desenvolver novas estratégias que servirão de blindagem ao sistema;

c) Analisar códigos já desenvolvidos, implementar segurança nas boas práticas de codificação e desenvolver novos meios que possam servir como chaves de segurança.

2.3.1 Implementação do Sistema de Autenticidade do Usuário

A figura do apêndice A (Diagrama de sequência de *login*) revela o diagrama de sequência de autenticidade para acessar o sistema KOALA.

A primeira coluna da figura mostra o controle do usuário, e todas as possíveis tomadas de decisão que o mesmo fará para acessar a plataforma, a segunda coluna, trata-se da interface humano-computador. A terceira coluna, o controle, ou seja, o sistema KOALA. A última coluna refere-se ao banco de dados.

Portanto, a decisão por parte da equipe KOALA e seguindo os critérios do levantamento de requisitos (documento criado inicialmente no procedimento de arquitetura do KOALA), o processo de entrar no sistema deverá primeiro iniciar com o usuário inserindo os dados "nome" e "senha", e concluir com o clicar do botão "*logar*". Assim o sistema da página irá enviar as informações junto com o código CSRF ao controle.

O controle irá validar o código, permitindo a continuação do processo ou impedindo. Após isso, irá buscar no banco de dados pela existência do usuário, e, portanto, retornado uma mensagem em forma booleana. Caso o usuário exista, o controle irá validar a senha, se retornar senha invalida o mesmo enviará

uma mensagem de erro à interface humano-computador. Note que a mensagem é a mesma tanto para o usuário inexistente quanto para senha incorreta, isso é uma boa prática de segurança, pois irá informar ao usuário apenas que há algo de errado.

Figura 2 – Fragmento de código, *backend* e *frontend.*

```python
def entrar(request):
    email = request.POST['email']
    senha = request.POST['password']

    try:
        user = Usuario.objects.get(username=email)
    except:
        return JsonResponse(data={
            'Autenticado': False,
            'motivo': 'Usuário não encontrado'
        })

    if (user.password == senha):
        login(request, user)
        return JsonResponse(data={'Autenticado': True})
    else:
        return JsonResponse(data={
            'Autenticado': False,
            'motivo': 'Senha inválida'
        })
```

(A)

```javascript
function logar() {
    var form = new FormData(document.getElementById("Logar"));
    fetch("/login/atentica/user", { method: 'POST', body: form })
    .then(response => response.json())
    .then(response => {
        console.log(response["Autenticado"]);
        if(response["Autenticado"]){
            window.location.href = "workspace";
        }else{
            alert(response["motivo"]);
        }
    })
    .catch(error => {
        console.log("#Error: "+error);
    });

}
```

(B)

Fonte: Elaborado pelos autores.

A figura 2 (Fragmento de código, *backend* e *frontend*) revela os trechos em que o diagrama de sequências foi implementado. O lado (A) da figura é o controle (escrito em *python* com o sistema *Django*), o sistema de autenticação CSRF é automatizado assim que o controle chama o método POST. Note que, ao invés de utilizar um comparador lógico para determinar a existência do usuário, foi implementada uma condição de tentativa, pois, se o usuário não existir, ele irá retornar um erro, impedindo o progresso no sistema de entrada. Enviando a mensagem de "usuário não encontrado" – a mensagem deverá ser mudada para "Usuário ou Senha incorreto", como forma de boas práticas de codificação e segurança.

Isso mantém a disponibilidade dos dados, caso o usuário exista, e prossegue com a análise da senha, identificando se a senha digitada é a mesma armazenada no banco de dados.

Uma forma de validar a senha, seria comparar a versão criptografada da senha recebida do *frontend* com a senha armazenada no *backend*, para isso, utiliza-se uma criptografia baseada em chaves (chave secreta do servidor + senha do usuário). Não devem ser utilizados valores mutáveis, como data e hora na chave secreta, pois isso poderá comprometer a validação de senhas.

O lado (B) da figura 2, revela o código implementado do diagrama de sequências no lado da interface humano-computador (escrito em *javascript*). Como esse código é visível ao público, o mesmo foi pensado para ser simples, o *frontend* não valida senhas, apenas transmite a informação pura para o controle, mantendo, assim, a confiabilidade e a integridade dos dados. Uma adição pode ser feita para validar informações como correção da escrita, um exemplo disso seria a formatação de CPFs, E-mails, dentre outras modais, entretendo é de suma importancia ter os mesmos validadores no *backend*.

2.3.2 Implementação do Sistema de Criação de Conta

O apêndice B (Diagrama de Sequência de Criação de Conta) revela o diagrama de sequência de Criação de Conta no sistema KOALA.

O processo de Criação de Conta é ainda mais complexo. Esse depende de várias interações do usuário com o sistema. Primeiro o usuário solicita a criação de conta, passando as informações bases, como "nome" (ou caso queira, nome social ou apelido), E-mail, senha e telefone.

O sistema então solicita ao usuário a confirmação do E-mail. O usuário acessa a nova tela de confirmação e finalização do processo de criação de conta através do *link* com um *Token* limitado por tempo. Esse procedimento serve para validar se o E-mail que o usuário usou é realmente dele, impedindo a criação de contas falsas, já que o sistema permite o relacionamento de uma conta por E-mail. Existem outras formas de validar se um E-mail é real, mas sem a interação do usuário, essas formas podem apenas dizer que o E-mail existe, mas não podem dizer que aquele E-mail pertence ao usuário.

Assim, fecha-se o tópico de desenvolvimento, onde foram revelados os princípios e implementos do conhecimento sobre segurança cibernética. Passa-se, agora, para o tópico de conclusão, onde será abordada a finalização dessa obra.

CONCLUSÃO: Revelação dos resultados

Nesta obra, foi abordado o assunto de privacidade de dados e demais estruturas relacionadas ao gerenciamento de riscos cibernéticos e do ambiente que foi utilizado no sistema para a execução dos testes.

A definição é que a criptografia é a principal forma para realizar a proteção dos dados, porém, outras estratégias simples, puderam auxiliar na proteção da segurança.

As funções que foram implementadas no sistema serviram de base para a criação de camadas de segurança, criando, assim, uma blindagem. Além disso, foram utilizadas funções do tipo CSRF *token* que funcionam como chaves de autenticidade do sistema, aumentado a segurança e diminuição no tamanho do código complexo, mantendo a filosofia do projeto de código limpo, facilitando as transações de informação e impedindo que terceiros mal-intencionados possam executar ações usando credenciais de outro usuário.

A visão no início do projeto de ter um sistema protegido atendeu com os objetivos propostos. A importância da elaboração de privacidade, gestão de riscos e as preocupações em desenvolver um sistema seguro e saudável para o cliente foi atendida com o estudo focado em análise de erros, cibersegurança e defesa cibernética. Não foram utilizados testes de penetração (*pentest*), apenas testes de lógicas.

Este trabalho faz-se importante para melhorar o aprofundamento deste tema pois, visto que a privacidade de dados e gestão de riscos cibernéticos estejam implementados, permitiram compreender melhor o sistema de

segurança, além de desenvolver e aperfeiçoar competências de investigação, seleção e organização.

REFERÊNCIAS BIBLIOGRÁFICAS

ANDESON, Rick. *Impedir o XSS (script entre sites) no ASP.NET Core*. 2022. In: Microsoft docs. Disponível em: https://docs.microsoft.com/pt-br/aspnet/core/security/cross-site-scripting?view=aspnetcore-6.0. Acesso em: 09 setembro. 2022.

BEZERRA, Juliana. *Taylorismo, fordismo e toyotismo*: Qual a diferença entre taylorismo, fordismo e toyotismo? 2020. In: Significados. Disponível em: https://www.diferenca.com/taylorismo-fordismo-e-toyotismo/ Acesso em: 29 abril. 2022.

CLEMENTE, Matheus. *Proteja-se*: 5 dicas essenciais de como manter a segurança do site. 2019. In: blog rockcontent. Disponível em: https://rockcontent.com/br/blog/seguranca-de-site/. Acesso em: 09 setembro. 2022.

MDN, Mozilla. *Website security*. 2022. In: developer mozilla. Disponível em: https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Website_security#what_is_website_security Acesso em: 09 setembro. 2022.

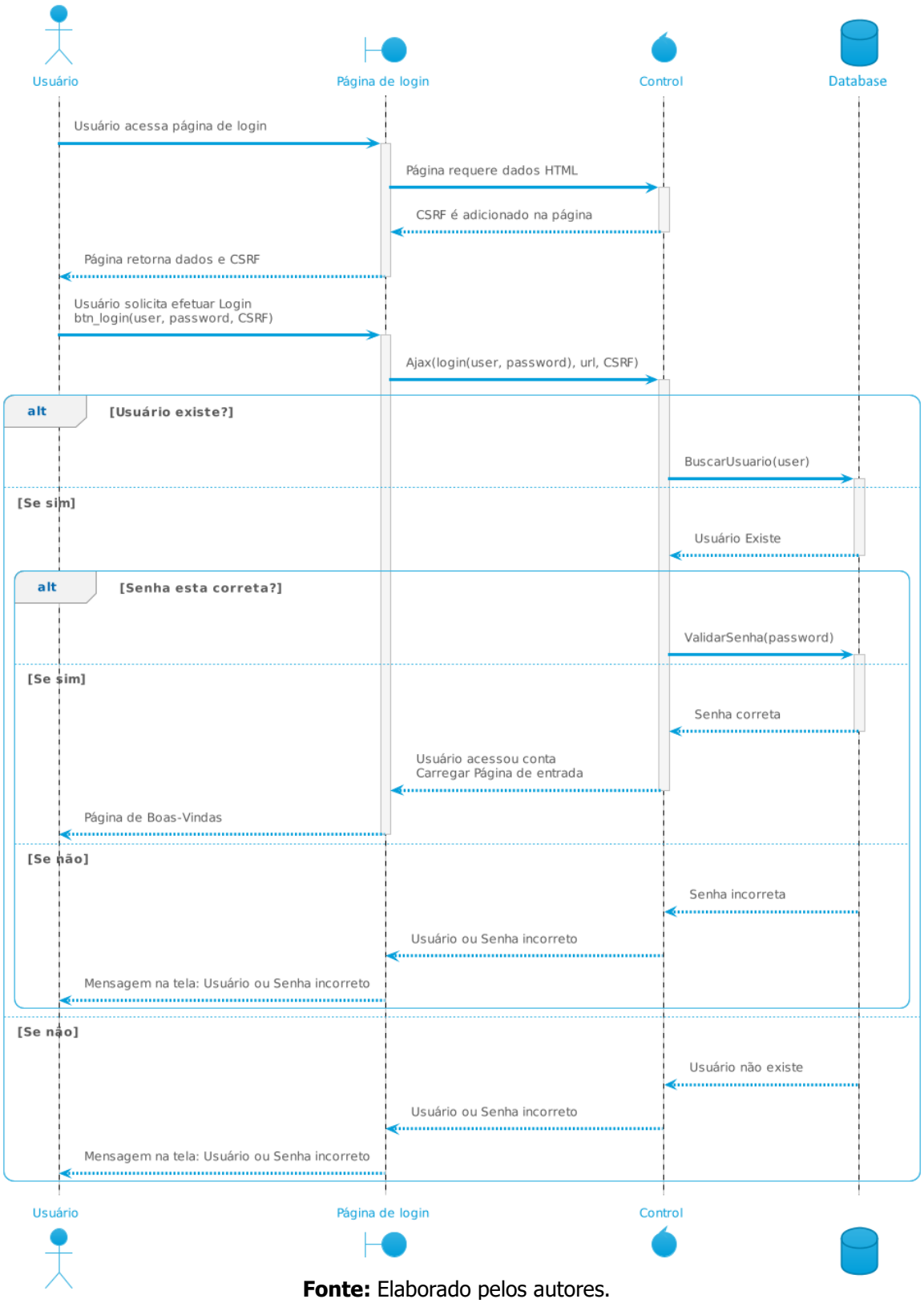
OWASP. *Cross Site Request Forgery* (CSRF). 2021c. In: Owasp. Disponível em: https://owasp.org/www-community/attacks/csrf. Acesso em: 09 setembro. 2022.

OWASP. *Denial of Service*. 2021a. In: Owasp. disponível em: https://owasp.org/www-community/attacks/Denial_of_Service#. Acesso em: 09 setembro. 2022.

OWASP. *SQL Injection*. 2021b. In: Owasp. Disponível em: https://github.com/OWASP/www-community/blob/master/pages/attacks/SQL_Injection.md. Acesso em: 09 setembro. 2022.

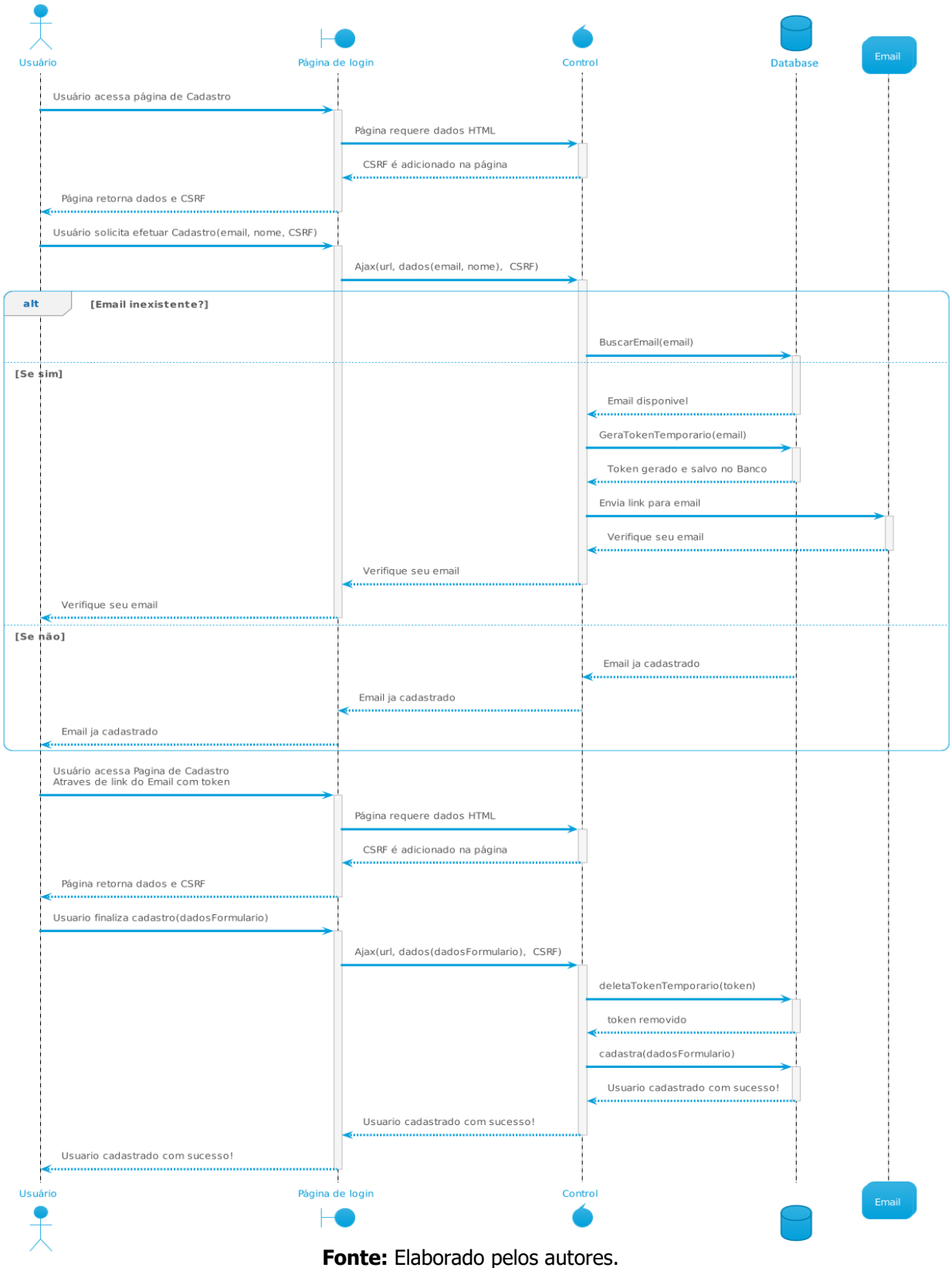
VIEIRA, Lucas. *UML – Diagramas de Sequência*. 2013. In: theclub. Disponível em: http://www.theclub.com.br/restrito/revistas/201308/umld1308.aspx. Acesso em: 09 setembro. 2022.

APÊNDICE A - Diagrama de sequência de *Login*.



Fonte: Elaborado pelos autores.

APÊNDICE B - Diagrama de sequência de criação de conta.



Fonte: Elaborado pelos autores.