Política de Desenvolvimento Seguro

Versão do documento: 1.2

Data de criação: 15/07/2024

Data da próxima revisão: 01/03/2027

Histórico de revisões

Versão	Data	Tipo de alteração	Autor	Revisor	Aprovador
1.2	01/03/2026	Revisão e simplificação da política; adequação da governança, treinamento, testes de segurança, SSDLC, segregação de ambientes,	Rafael Fernandes		

		CI/CD, versionamento e exceções ao porte atual da WeCare			
1.1	04/02/2025	Incluídas diretrizes na utilização do GIthub	Rafael Fernandes	Eduardo Fernandes	Deborah Oliveira
1.0	15/07/2024	Criação	Rafael Fernandes	Eduardo Fernandes	Deborah Oliveira

Sumário

Propósito

Esta política tem como finalidade estabelecer diretrizes que garantam que o software desenvolvido pela WeCare esteja em conformidade com os padrões de segurança, promovendo a minimização de vulnerabilidades em todas as fases do ciclo de vida do desenvolvimento, desde a concepção e design até a implementação, testes e manutenção contínua.

Governança e Responsabilidade

A diretoria da WeCare designará um responsável pela segurança da informação, que será encarregado de manter esta política atualizada, orientar sua aplicação e apoiar a avaliação de riscos e exceções. A equipe de desenvolvimento é responsável por aplicar os requisitos de segurança definidos neste documento ao longo de todo o ciclo de desenvolvimento.

Conformidade Legal e Regulatória

Esta política foi elaborada com base em boas práticas de segurança da informação e desenvolvimento seguro, incluindo referências compatíveis com a ISO 27001, OWASP e requisitos regulatórios aplicáveis à WeCare, conforme sua realidade operacional.

Público-alvo

Esta política aplica-se a desenvolvedores internos, terceiros autorizados e demais pessoas que contribuam com código, configuração, automação, infraestrutura de entrega ou manutenção de aplicações da WeCare.

Descrição da Política

Práticas de Desenvolvimento Seguro

TREINAMENTO

Os desenvolvedores e demais envolvidos no ciclo de desenvolvimento devem manter conhecimento atualizado sobre segurança da informação, codificação segura e tratamento de vulnerabilidades, por meio de treinamentos, reciclagens internas, estudo dirigido ou materiais de referência adotados pela WeCare. A atualização deve ocorrer periodicamente e sempre que houver mudança relevante de tecnologia, arquitetura ou risco.

DEFINIR OS REQUISITOS DE SEGURANÇA

A segurança e a privacidade devem ser consideradas desde o início do desenvolvimento de qualquer aplicação ou funcionalidade. Os requisitos de segurança devem ser definidos de forma proporcional ao risco do serviço, considerando os dados tratados, o tipo de usuário, a exposição da aplicação,

integrações envolvidas e impactos para o negócio. Esses requisitos devem ser revistos sempre que houver mudança relevante de arquitetura, tecnologia ou risco.

- **Identificação de dados críticos**: Determinar quais dados serão coletados, processados e armazenados com base em seu nível de criticidade.
- **Avaliação de riscos**: Identificar potenciais ameaças e vulnerabilidades.
- **Consulta a regulamentações**: Garantir conformidade com legislações relevantes, como a LGPD.

DEFINIR E UTILIZAR PADRÕES DE CRIPTOGRAFIA

Proteger os dados, especialmente as informações confidenciais, contra vazamentos ou alterações não autorizadas durante a transmissão ou armazenamento deve ser uma prioridade. A criptografia de dados é a principal ferramenta para alcançar esse objetivo.

- Utilizar padrões de criptografia seguros e amplamente aceitos pela indústria, compatíveis com a tecnologia adotada pela WeCare;
- Utilizar bibliotecas e serviços confiáveis, oficialmente suportados e mantidos;
- Garantir gestão segura de chaves, segredos e certificados;
- Evitar implementações criptográficas próprias, salvo quando tecnicamente justificado e revisado.

GERENCIAR RISCOS DE SEGURANÇA EM COMPONENTES DE TERCEIROS

Ao utilizar componentes de terceiros, deve-se considerar o impacto de possíveis vulnerabilidades de segurança. Para mitigar riscos, é necessário manter um inventário completo de todos os componentes de terceiros e ter um plano de resposta para lidar com novas vulnerabilidades.

- Manter visibilidade sobre os principais componentes de terceiros utilizados, preferencialmente por meio dos arquivos de dependência, repositórios, ferramentas automatizadas ou documentação técnica mínima adotada pela WeCare.
- **Plano de respostas a vulnerabilidades:** desenvolver um plano para responder a novas vulnerabilidades, incluindo avaliação de impacto e implementação de correções.
- **Varreduras automatizadas:** utilizar ferramentas utilizadas para realizar varreduras periódicas de vulnerabilidades em bibliotecas, dependências, imagens, pacotes ou componentes utilizados pelas aplicações.
- **Tratamento de vulnerabilidades:** atualizar prontamente os componentes para versões não impactadas ou corrigidas e documentar todas as ações tomadas.

UTILIZAR FERRAMENTAS APROVADAS

A WeCare deve adotar e comunicar as ferramentas padrão de desenvolvimento, versionamento, automação, testes e deploy utilizadas pela

equipe. Sempre que possível, devem ser priorizadas ferramentas já aprovadas internamente e com suporte adequado ao processo de desenvolvimento seguro.

Os desenvolvedores precisam:

- **Utilizar ferramentas aprovadas:** priorizar o uso das versões mais recentes das ferramentas aprovadas.
- **Manter-se atualizado:** acompanhar novas análises de segurança e medidas de proteção disponíveis para essas ferramentas.

Essas práticas garantem a consistência e eficácia dos processos de desenvolvimento, fortalecendo a segurança do software contra ameaças cibernéticas.

TESTES DE SEGURANÇA

Análise estática (SAST)

- Utilizar, sempre que viável técnica e operacionalmente, ferramentas de análise estática (SAST) ou verificações equivalentes para identificar vulnerabilidades no código-fonte durante o desenvolvimento.
- Essa técnica automatizada examina o código em repouso, procurando por falhas de segurança e violações das melhores práticas de programação.

- Ao integrar SAST no ciclo de desenvolvimento, a equipe de desenvolvimento pode identificar e corrigir problemas de segurança precocemente.

Análise dinâmica (DAST)

- Utilizar, quando aplicável à arquitetura e ao tipo de aplicação, ferramentas de análise dinâmica (DAST) ou verificações equivalentes para identificar vulnerabilidades em execução.
- Esta técnica simula ataques reais, explorando a aplicação em funcionamento para descobrir falhas de segurança, como *SQL Injection,* e *Cross-Site Scripting*.
- Ao integrar DAST no ciclo de desenvolvimento, ajuda a identificar e corrigir vulnerabilidades antes que elas se tornem alvos de ataques reais.

Teste de penetração (Pentest)

- Testes de penetração devem ser realizados periodicamente, de acordo com o risco, a exposição da aplicação, a criticidade do serviço, mudanças relevantes de arquitetura ou exigências contratuais/regulatórias.
- Sempre que adotados, esses testes devem ser conduzidos por profissionais qualificados, internos ou externos, com tratamento adequado de confidencialidade e registro dos planos de correção.

Ciclo de Vida de Desenvolvimento Seguro (SSDLC)

O ciclo de desenvolvimento da WeCare adota práticas ágeis, com etapas de descoberta, análise de requisitos e design, desenvolvimento, testes, homologação e publicação em produção. A forma de organização da equipe pode variar conforme o porte do time, a natureza das demandas e a maturidade operacional, devendo sempre preservar os controles mínimos de segurança descritos nesta política.

Em cada etapa, há práticas para assegurar a segurança no desenvolvimento de nossos sistemas, softwares e aplicações.

DESCOBERTA, ANÁLISE DE REQUISITOS E DESIGN

Nessa etapa, as demandas já priorizadas pelas áreas comercial e sucesso do cliente, são discutidas para compreender as necessidades dos clientes e usuários. Fluxogramas de uso, *wireframes* e documentação de casos de uso e regras de negócio são elaborados. Também são definidas a arquitetura e tecnologias que serão utilizadas.

As práticas de segurança nessa etapa devem se atentar a:

- Definir requisitos claros para controle de acesso, incluindo autenticação e autorização e especificar quem tem acesso a quais dados e funcionalidades e sob quais condições;
- Garantir que os dados dos sistemas tenham o mínimo de impacto pelas novas demandas;

- Identificar dados sensíveis e definir requisitos para a sua proteção, como criptografia e anonimização;
- Definir política de retenção e descarte de dados;
- Especificar requisitos para auditoria para garantir que todas as atividades relevantes sejam registradas;
- Especificar procedimentos de backup e restauração de dados;
- Realizar revisão dos requisitos para garantir que as principais preocupações de segurança sejam consideradas antes do desenvolvimento.

DESENVOLVIMENTO

Nessa etapa, o código é produzido e revisado para garantir que os requisitos funcionais e não funcionais sejam cumpridos. O código passa por uma análise para verificar a legibilidade, modularidade, manutenibilidade, documentação e performance.

As práticas de segurança nessa etapa devem se atentar a:

- Desenvolver com testes compatíveis com o risco e a criticidade da funcionalidade, incluindo testes unitários e, quando aplicável, testes funcionais;
- Utilizar práticas de codificação segura com base em referências como OWASP;

- Garantir que somente o código necessário à funcionalidade seja incorporado;
- Submeter alterações relevantes à revisão por outro desenvolvedor sempre que houver disponibilidade operacional.

TESTES E HOMOLOGAÇÃO

Nessa etapa, a solução é testada com o objetivo de se encontrar falhas de funcionamento. A homologação é realizada em um ambiente controlado em nuvem para garantir a qualidade do software. Durante esta fase, são realizadas os seguintes testes

- **Testes Funcionais:** verificar se todas as funcionalidades do software estão operando conforme o esperado.
- **Testes de Integração:** garantir que diferentes módulos e componentes do sistema funcionem bem juntos.
- **Testes de Performance:** avaliar o desempenho do software sob diferentes cargas e condições.
- **Testes de Segurança:** identificar e corrigir possíveis vulnerabilidades de segurança.
- **Testes de Usabilidade:** assegurar que a interface do usuário seja intuitiva e fácil de usar.

As práticas de segurança nessa etapa devem se atentar a:

- Garantir que o sistema e a nova funcionalidade funcionem;
- Garantir que o sistema funcione corretamente em diferentes fluxos e cargas;
- Utilizar ferramentas como SAST, DAST ou verificações equivalentes de forma periódica e compatível com o risco da aplicação, priorizando alterações relevantes, novas funcionalidades, integrações críticas e publicações em produção.

PUBLICAÇÃO EM PRODUÇÃO

Nesta etapa, a solução é publicada no ambiente de produção. As seguintes atividades são realizadas para garantir uma transição suave e segura:

- **Preparação do Ambiente:** configurar e preparar o ambiente de produção para receber a nova versão do software.
- **Deploy:** realizar a implantação do software em produção, seguindo um processo controlado e automatizado.
- **Verificação Pós-Deploy:** executar verificações para assegurar que a implantação foi bem-sucedida e que o sistema está funcionando corretamente.
- **Monitoramento:** iniciar o monitoramento contínuo do sistema para detectar e responder rapidamente a qualquer problema ou anomalia.

- **Comunicação:** informar as partes interessadas sobre a nova versão e quaisquer mudanças ou melhorias significativas.

As práticas de segurança nessa etapa devem se atentar a:

- Garantir que o sistema continue disponível;
- O processo de publicação deve ser o mais automatizado possível, de acordo com a maturidade operacional da WeCare;
- Sempre que viável, a publicação deve minimizar indisponibilidade para os usuários;
- Deve existir procedimento de reversão, mitigação ou correção rápida em caso de falha relevante;
- As publicações devem ser planejadas de modo a reduzir impacto para os usuários e para a operação;
- Após a publicação, a aplicação deve ser acompanhada por meio de monitoramento, logs e alertas compatíveis com a criticidade do serviço.

Segregação de Ambientes

A WeCare adota a prática de segregar os ambientes. Essa prática assegura a integridade e disponibilidade nos ambientes e os dados que nele residem, de desenvolvimento, homologação e produção, proteger contra acesso não autorizado, alterações e outros impactos negativos.

O objetivo da segregação de ambientes é:

- Evitar acesso indesejado a servidores críticos para o negócio;
- Realizar testes de aceitação antes da implantação;
- Realizar auditorias de conformidade;
- Isolar vulnerabilidades de segurança;
- Restringir o acesso ao ambiente de produção ao mínimo necessário, com concessão controlada, rastreável e compatível com a necessidade operacional; e
- Definir procedimentos para a promoção do software do ambiente de homologação para o ambiente de produção.

AMBIENTES

A WeCare adota ambientes segregados de desenvolvimento, homologação e produção, conforme aplicável à sua arquitetura. A infraestrutura utilizada deverá observar controles compatíveis de segurança, disponibilidade, acesso e rastreabilidade.

Integração Contínua e Entrega Contínua (CI/CD)

A WeCare implementa a Integração Contínua (CI) e Entrega Contínua (CD) para manter a eficiência e a qualidade ao longo do processo de desenvolvimento de software.

INTEGRAÇÃO CONTÍNUA (CI)

Sempre que tecnicamente viável, alterações de código devem acionar verificações automatizadas, como build, testes e validações de segurança, com retorno rápido para correção de falhas antes da promoção entre ambientes.

ENTREGA CONTÍNUA (CD)

Os deploys devem ser preferencialmente automatizados, especialmente para homologação e produção, conforme a capacidade operacional e a maturidade da esteira adotada pela WeCare.

Versionamento de Código e Acesso ao Código Fonte

O acesso ao código fonte é restrito, com base no princípio do menor privilégio. O controle de versionamento utilizado pela WeCare é baseado em Git, utilizando a plataforma adotada oficialmente pela empresa. Todos os desenvolvedores devem seguir os seguintes requisitos destinados a proteger o código fonte contra acesso não autorizado, roubo, alteração e perda.

- O código fonte só deve ser transferido de forma criptografada;
- As senhas e outras credenciais utilizadas para acessar os repositórios e ferramentas de desenvolvimento devem ser armazenadas em solução de cofre de senhas, gestão de identidades ou segredos aprovada pela WeCare;

- As contas de acesso ao código fonte devem ser protegidas com MFA (Múltiplo Fator de Autenticação);
- Alterações na branch principal devem, preferencialmente, passar por revisão de código via Pull Request (PR) antes da fusão. Em situações excepcionais de urgência operacional, a alteração poderá ser aplicada com registro da justificativa e revisão posterior, quando cabível;
- Commits não devem conter credenciais, chaves de API, tokens de acesso ou outras informações sensíveis. O uso de ferramentas como Git-secrets ou TruffleHog deve ser considerado para evitar exposições acidentais;
- O uso de branches deve seguir um fluxo de trabalho padronizado pela equipe, suficiente para manter organização, rastreabilidade e segurança das alterações;
- Apenas repositórios privados devem ser utilizados para armazenar código fonte interno da WeCare;
- Dependências de terceiros devem ser analisadas regularmente quanto a vulnerabilidades conhecidas, utilizando ferramentas como Dependabot ou similares;
- A remoção de acessos ao repositório deve ocorrer imediatamente quando um colaborador deixar a empresa ou quando não precisar mais do acesso para suas atividades;

- Logs, históricos de alterações e registros disponíveis na plataforma de versionamento devem ser preservados e consultados sempre que necessário para investigação, rastreabilidade ou resposta a incidentes.

Exceções

Qualquer pedido de exceção a esta política deve ser submetido por escrito e será analisado com base em risco. Exceções somente poderão ser concedidas mediante aprovação documentada da diretoria da WeCare ou do responsável formalmente designado pela segurança da informação.

Revisão do Documento

Este documento será revisado, no mínimo, uma vez por ano, ou sempre que houver mudança relevante na arquitetura, na equipe, nas ferramentas, nos riscos ou no processo de desenvolvimento da WeCare.

** FIM DO DOCUMENTO **