Segurança na prática com ISO 27001 aplicado em Delphi

Embarcadero Conference 2023

{ Palestrante

Gean Carlo Trevizani Nascimento

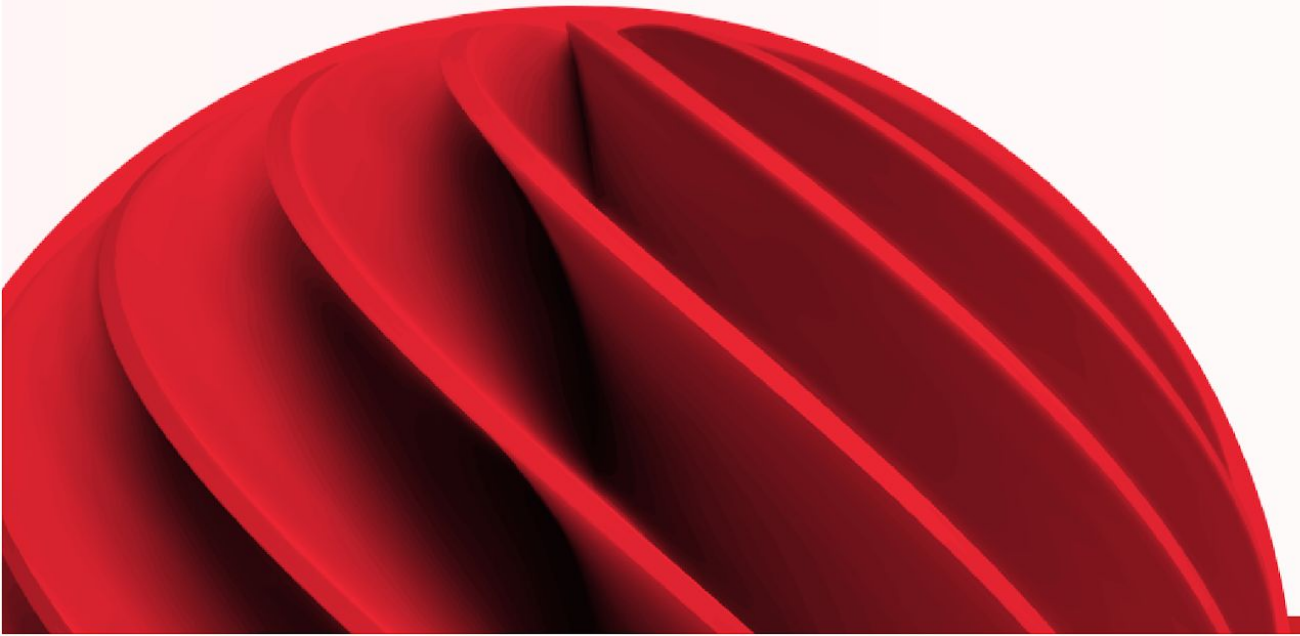
Agenda

- O que é a ISO 27001
- Aplicação da ISO 27001 em:
  - Análises
  - Desenvolvimentos
  - Testes
- Exemplos práticos:
  - SQL Injection
  - Criptografia/HTTP/HTTPS
- Prova Information Security Foundation based on ISO IEC 27001 (ISFS)

O que é a ISO 27001 ?

O que é a ISO 27001

A ISO 27001 é uma norma internacional de segurança da informação que estabelece os requisitos para implementar um Sistema de Gerenciamento de Segurança da Informação (ISMS - Information Security Management System). Seu objetivo é proteger as informações sensíveis e críticas de uma organização, garantindo a confidencialidade, integridade e disponibilidade dos dados.



Aplicação da ISO 27001 como aplicar em:
➔ Análises

Na área de análise de software, a norma é aplicada para identificar riscos de segurança, estabelecer controles de proteção de dados, gerenciar incidentes, promover conscientização e treinamento em segurança, além de realizar auditorias internas para garantir a conformidade e aprimorar continuamente os processos de análise de software. Isso demonstra o comprometimento da organização com a segurança da informação e a proteção de dados sensíveis envolvidos no desenvolvimento e uso do software.

Aplicação da ISO 27001 como aplicar em: ➔ Desenvolvimentos

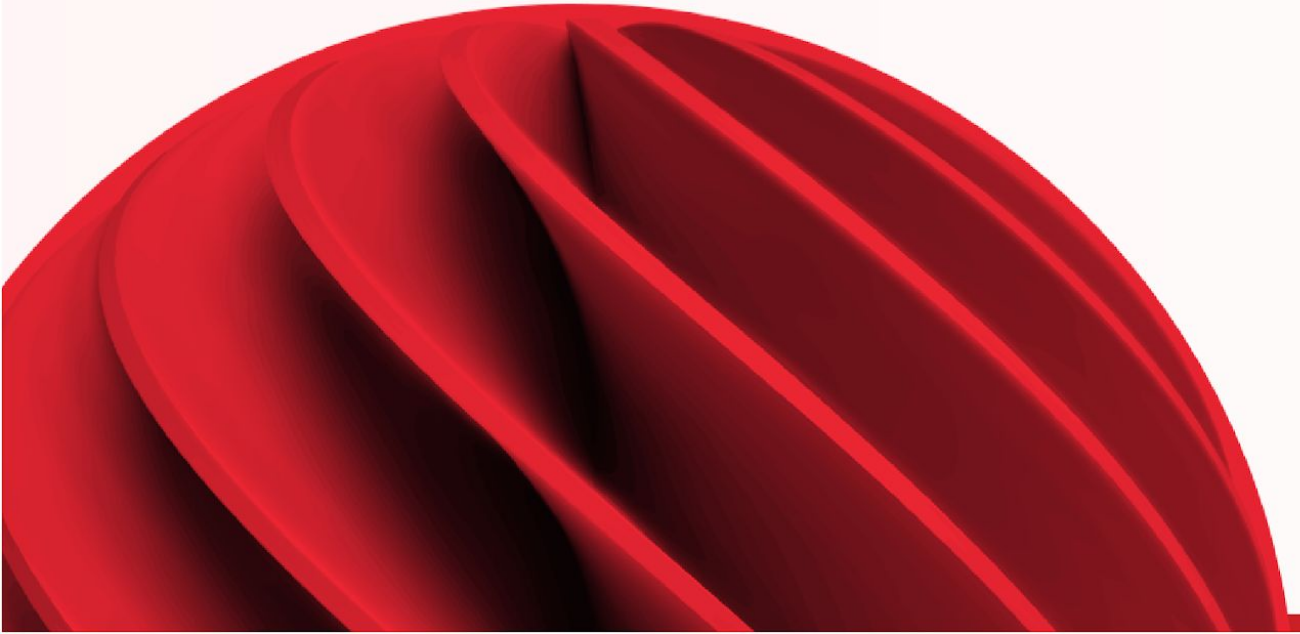
No desenvolvimento de sistemas envolve uma abordagem sistemática para identificar riscos, estabelecer controles de segurança, gerenciar fornecedores, treinar a equipe e monitorar continuamente a conformidade. Isso visa proteger as informações sensíveis durante todo o ciclo de vida do sistema, aumentar a confiança dos clientes, mitigar riscos de segurança e demonstrar um compromisso sério com a proteção dos dados.

Aplicação da ISO 27001 como aplicar em:
➔ Testes

Na área de testes de softwares é aplicada para implementar práticas de segurança da informação durante o ciclo de vida do desenvolvimento e testes. A norma ajuda a identificar e mitigar riscos, estabelecer políticas de segurança, controlar o acesso a dados sensíveis, garantir a segurança da infraestrutura de teste, proteger dados através de criptografia, promover a conscientização dos colaboradores e assegurar a integridade, confidencialidade e disponibilidade das informações envolvidas nos testes de forma segura, gerando confiança junto a clientes e parceiros.

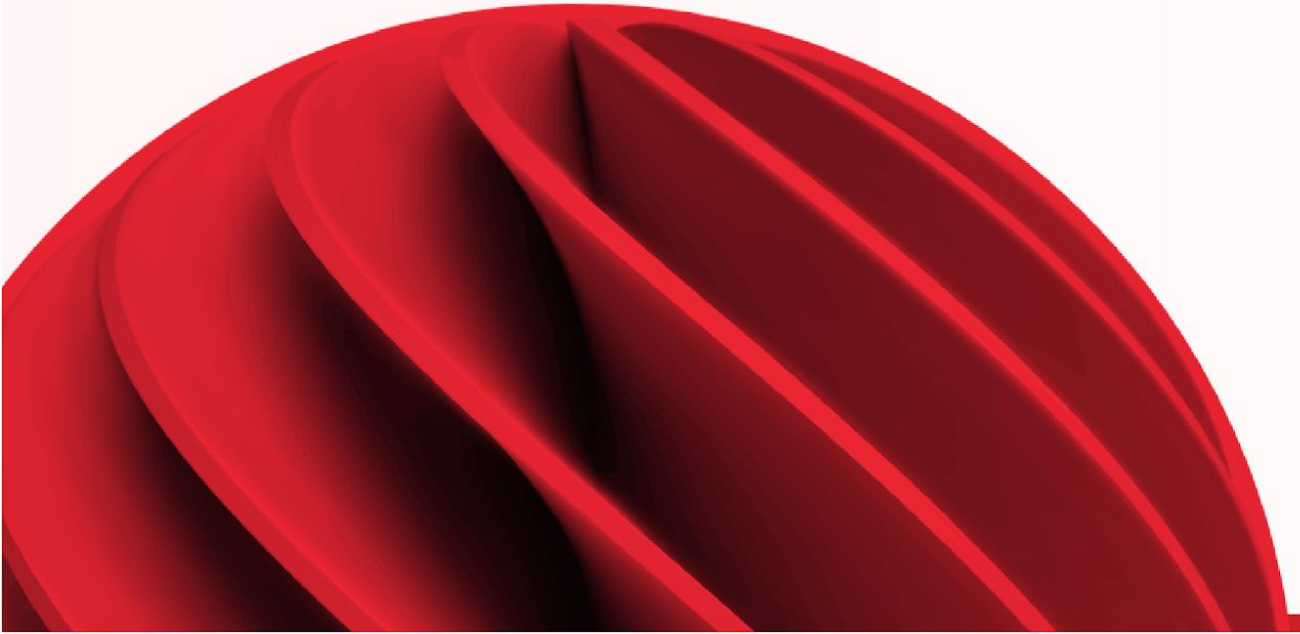
Exemplos práticos:

- SQL Injection



Exemplos práticos:

- Criptografia/HTTP/HTTPS

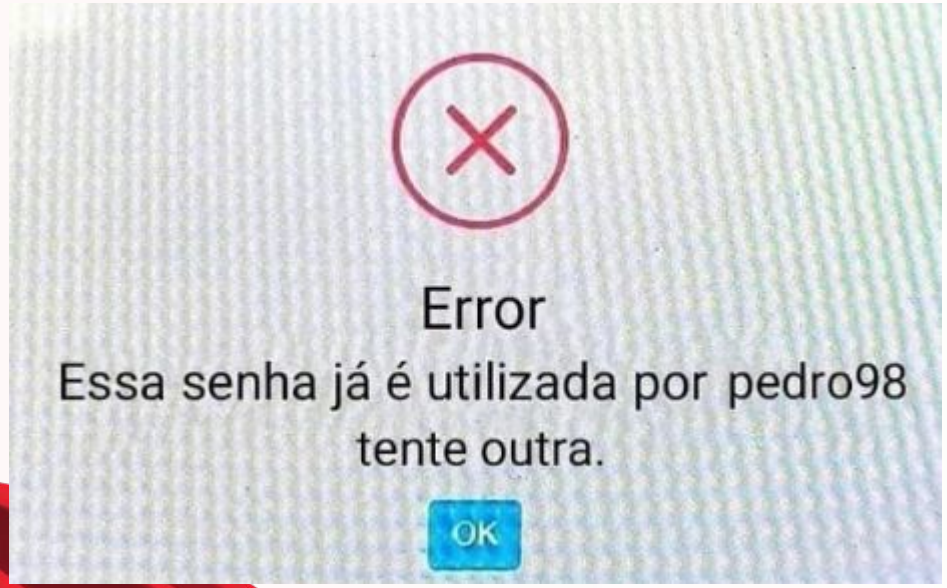



Pontos de Atenção
Embarcadero Conference 2023

```
query := TMyQuery.Create(nil);
query.Connection := MyConnection;

Write('Digite o nome do usuário: ');
ReadLn(userInput);

query.SQL.Text := 'SELECT * FROM Usuarios WHERE Nome = ''' + userInput + '''';
query.Open;

while not query.Eof do
begin
  Writeln('Nome: ', query.FieldByName('Nome').AsString);
  query.Next;
end;
```

Validação de Entrada: Sempre valide as entradas de todas as fontes de dados não confiáveis. A validação adequada das entradas pode eliminar a grande maioria das vulnerabilidades de software. Desconfie da maioria das fontes externas de dados, incluindo argumentos de linha de comando, interfaces de rede, variáveis de ambiente e arquivos controlados pelo usuário.

```
var
  senha: string;

begin
  try
    // Suponha que ocorreu um erro na autenticação.
    senha := 'senha_incorreta';

    Writeln('Erro de autenticação: A senha fornecida é inválida.');

  except
    on E: Exception do
      Writeln('Erro: ', E.ClassName, ' - ', E.Message);
  end;
end.
```



Codificação de Saída: Garanta que a saída seja codificada de forma a não ser processada de maneira não intencional. Isso é particularmente importante ao retornar mensagens de erro, pois mensagens de erro detalhadas podem revelar informações sobre o sistema que poderiam ser usadas em um ataque.

```
begin
  try
    Write('Digite sua nova senha: ');
    ReadLn(senha);

    // Verificar se a senha atende a requisitos de segurança (exemplo simples).
    if Length(senha) >= 8 then
      Writeln('Senha definida com sucesso.')
    else
      Writeln('A senha deve ter pelo menos 8 caracteres.');

  except
    on E: Exception do
      Writeln('Erro: ', E.ClassName, ' - ', E.Message);
  end;
end.
```

Autenticação e Gerenciamento de Senhas: Certifique-se de que as senhas sejam armazenadas de forma segura e que os dados de autenticação não sejam vazados em mensagens de erro ou logs. Implemente políticas de bloqueio de conta para proteger contra ataques de força bruta.

```
begin
  try
    // Gera um token de sessão aleatório (exemplo simples).
    tokenSessao := IntToStr(Random(1000000));

    // Valida o token de sessão.
    if not sessoesAtivas.Contains(tokenSessao) then
    begin
      sessoesAtivas.Add(tokenSessao);
      Writeln('Sessão iniciada com sucesso.');
    end
    else
      Writeln('Token de sessão inválido.');

  except
    on E: Exception do
      Writeln('Erro: ', E.ClassName, ' - ', E.Message);
  end;
```

Gerenciamento de Sessão: Implemente um gerenciamento de sessão seguro, incluindo o uso de cookies, e certifique-se de que os tokens de sessão sejam gerados com segurança e invalidados quando não forem mais necessários.

```
begin
  try
    // Verificar se o usuário está autenticado (exemplo simples).
    usuarioAutenticado := AutenticarUsuario(usuario);

    if usuarioAutenticado then
      Writeln('Acesso concedido à funcionalidade.')
    else
      Writeln('Acesso negado.');

  except
    on E: Exception do
      Writeln('Erro: ', E.ClassName, ' - ', E.Message);
  end;
end.
```

Controles de Acesso: Implemente controles de acesso adequados para evitar acesso não autorizado a dados e funcionalidades sensíveis. Isso inclui garantir que os controles de acesso sejam aplicados no lado do servidor e estejam configurados corretamente.

```
begin
  try
    log := TStringList.Create;
    senha := 'senha_incorreta';

    // Lida com o erro sem divulgar informações sensíveis.
    Writeln('Erro de autenticação: A senha fornecida é inválida.');

    // Registra o evento em um arquivo de log.
    log.Add(FormatDateTime('yyyy-mm-dd hh:nn:ss', Now) + ' - Erro de autenticação');

    // Salva o log em um arquivo.
    log.SaveToFile('log.txt');

  except
    on E: Exception do
      Writeln('Erro: ', E.ClassName, ' - ', E.Message);
  end;
  log.Free;
end.
```

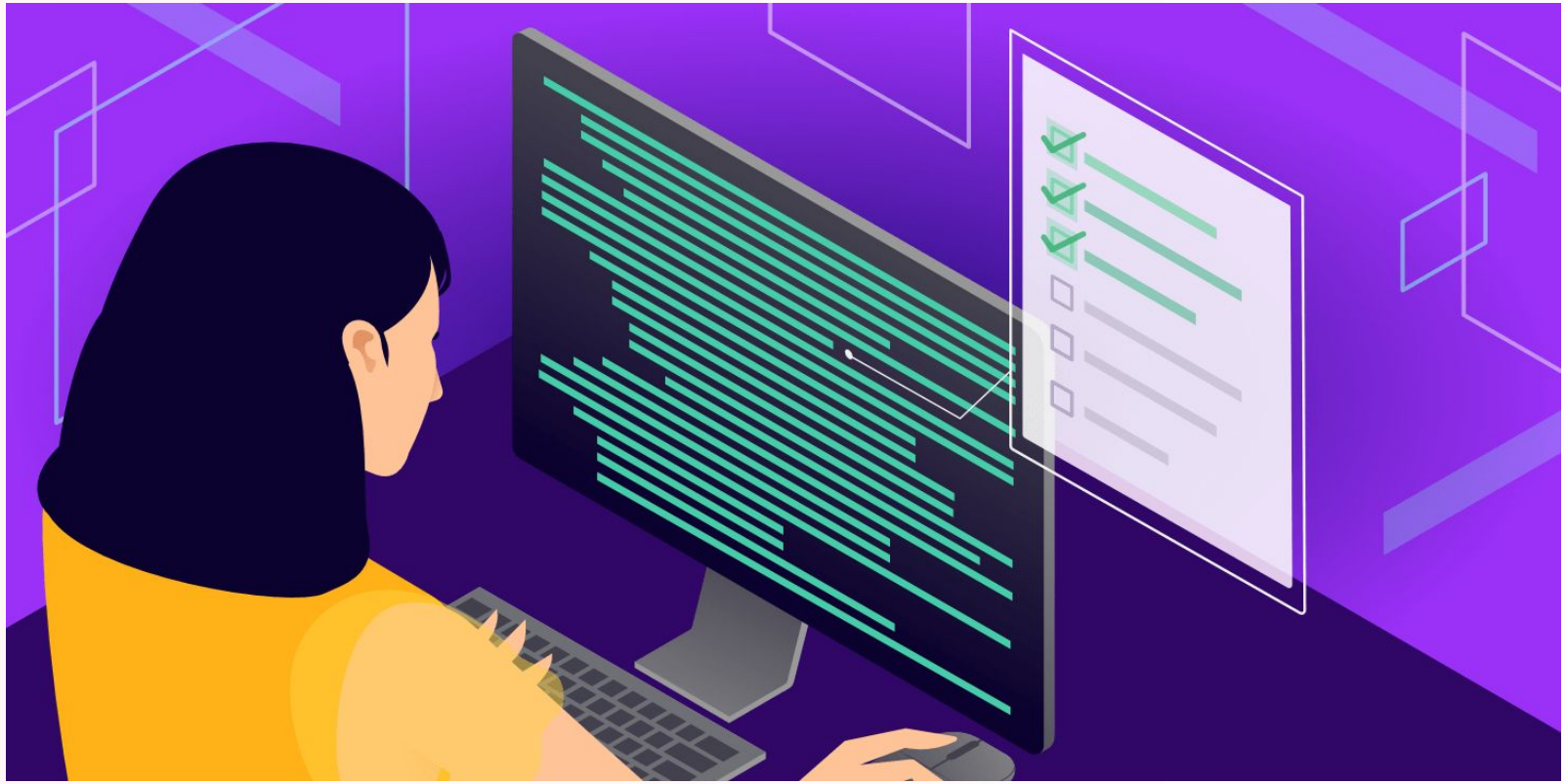
Tratamento de Erros e Registro: Garanta que as mensagens de erro não divulguem informações sensíveis e que os logs não contenham dados confidenciais. Os logs devem ser armazenados e revisados regularmente.

```
procedure CriptografarArquivo(const nomeArquivo: string; const chave: string);
var
  arquivoEntrada, arquivoSaida: TFileStream;
  criptoAES: TIdBlockCipherIO;
begin
  arquivoEntrada := TFileStream.Create(nomeArquivo, fmOpenRead);
  try
    arquivoSaida := TFileStream.Create(nomeArquivo + '.cripto', fmCreate);
    try
      criptoAES := TIdBlockCipherIO.Create(nil);
      try
        criptoAES.Cipher := 'aes-256-cbc';
        criptoAES.Key := ToBytes(chave);
        criptoAES.OpenWrite;
        criptoAES.CopyFrom(arquivoEntrada, 0);
      finally
        criptoAES.Free;
      end;
    finally
      arquivoSaida.Free;
    end;
  finally
    arquivoEntrada.Free;
  end;
end;
```

Proteção de Dados: Certifique-se de que os dados sensíveis sejam criptografados. Use algoritmos seguros e chaves fortes.

```
begin
  try
    httpClient := TIdHTTP.Create;
    try
      // Configuração de SSL para comunicação segura.
      httpClient.IOHandler := TIdSSLIOHandlerSocketOpenSSL.Create(httpClient);

      mensagem := 'Mensagem confidencial';

      // Envia a mensagem criptografada usando HTTPS.
      resposta := httpClient.Post('https: //www.exemplo.com', mensagem);

      Writeln('Resposta do servidor:', resposta);
    finally
      httpClient.Free;
    end;

  except
    on E: Exception do
      Writeln('Erro: ', E.ClassName, ' - ', E.Message);
  end;
end.
```

Segurança na Comunicação: Use protocolos seguros para a comunicação, como HTTPS em vez de HTTP, e certifique-se de que estejam configurados adequadamente.



Revisões de Código e Testes: Faça revisões regulares de código em busca de vulnerabilidades de segurança e escreva testes de unidade para garantir que as alterações não introduzam novas vulnerabilidades.

Gerenciamento de Patches e Vulnerabilidades: Mantenha todos os sistemas, softwares e bibliotecas atualizados com as últimas correções. Faça verificações regulares em busca de vulnerabilidades nos sistemas.

Princípio do Menor Privilégio: Cada módulo (como um processo, um usuário ou um programa, dependendo do caso) deve ser capaz de acessar apenas as informações e recursos necessários para seu propósito legítimo.

```
var
  httpServer: TIdHTTPServer;
begin
  try
    httpServer := TIdHTTPServer.Create(nil);
    try
      // Configuração SSL para comunicação segura.
      httpServer.IOHandler := TIdServerIOHandlerSSLOpenSSL.Create(httpServer);
      TIdServerIOHandlerSSLOpenSSL(httpServer.IOHandler).SSLOptions.Method := sslvTLSv1_2;

      // Configuração do servidor HTTP.
      httpServer.DefaultPort := 443; // Usar a porta padrão HTTPS (443).
      httpServer.Active := True;

      Writeln('Servidor HTTP seguro ativo na porta 443.');

      // Mantenha o servidor em execução.
      ReadLn;

    finally
      httpServer.Free;
    end;

  except
    on E: Exception do
      Writeln('Erro: ', E.ClassName, ' - ', E.Message);
  end;
end.
```

Padrões Seguros por Padrão: Uma postura segura por padrão significa que as configurações padrão são as mais seguras possíveis, projetadas para garantir que o sistema seja seguro por padrão.

Onde fazer/Como fazer ?

A prova *Information Security Foundation based on ISO IEC 27001 (ISFS)* custa R$1.440,00.

Fonte (09/2023):
https://opiceblumacademy.com.br/curso/information-security-management-iso-27001/#:~:text=Inscreva-se%20agora%20no%20EXIN,em%20um%20mundo%20altamente%20conectado.&text=A%20prova%20Information%20Security%20Foundation,)%20custa%20R%241.440%2C00

Gean Nascimento

{ Analista de Sistemas em Aquasoft

gean.nascimento@aquasoft.com.br

(51) 993-596-779

Embarcadero Conference 2023

Acesse e cadastre para retirar seu brinde

O que você achou da palestra?

Acesse o link do QR Code ao lado e responda a pesquisa.