

Guia de Implantação - Auditor de Código

1. Contextualização

O sistema Eligere possibilita que parte do seu código seja auditado. Para essa auditoria, disponibilizamos uma implementação padrão e open-source que pode ser usada para auditar o código do sistema.

2. Auditoria Externa do Código Fonte

Acesse a tela de auditoria de código fonte e baixe o código fonte disponível.

Página de Auditoria de Código Fonte

O sistema ELIGERE possui um processo para auditoria do seu código fonte.

A auditoria possibilita qualquer pessoa baixar e verificar que o código que está executando no momento da eleição.

São auditados:

1. Todo o Código de Registro de Votos
2. Todo o Código de Homologação dos Resultados
3. Todo o Código da Parte de Auditoria

Partes restritas do código, como a parte de criptografia das informações do voto e geração de chaves de segurança não é auditável por questões óbvias de segurança

Hash Atual do Código do Sistema: **fa949be4f37212a99dae42c745ac7c8bdb2bb0ceabe313fa089682634e123322** gerado em 02/09/2020 09:52:34

[Baixar Código Auditável](#)

[Ver Código Fonte Auditável](#)

Obtenha o auditor de código externo de duas maneiras: 1) Ou baixando o seu executável ou ainda 2) abaixando o seu código fonte

2.1 Utilizando o executável

Acesso o diretório do arquivo AuditorCodigoFonte-x.y.z.jar e digite o comando

```
java -jar AuditorCodigoFonte-x.y.z.jar
```

```
jadson@localhost:~/git/AuditorCodigoFonte/libs$ ls
AuditorCodigoFonte-1.0.0.jar
jadson@localhost:~/git/AuditorCodigoFonte/libs$ java -jar AuditorCodigoFonte-1.0.0.jar
```

Isso abrirá a tela do auditor:

Auditor de Código
Selecione os Arquivos com o Código Auditável da Aplicação e informe a data em que o hash foi gerado, para validar o HASH.

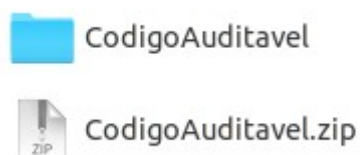
Hash:

Data de Geração do Hash:

Selecione os arquivos de Auditoria

Aplicação Iniciada ...

Descompacte o arquivo **CodigoAuditavel.zip** baixado em um diretório de sua preferência.



Retorne a tela do sistema Eligere e copie o hash e data de geração do hash informados pela aplicação.

Cole o hash e data nos respectivos campos do auditor e selecione todos os arquivos dentro do diretório **CodigoAuditavel**

Selecione os Arquivos com o Código Auditável da Aplicação e informe a data em que o hash foi gerado, para validar o HASH.

Hash:

fa949be4f37212a99dae42c745ac7c8bdb2bb0ceabe313fa089682634e123322

Data de Geração do Hash:

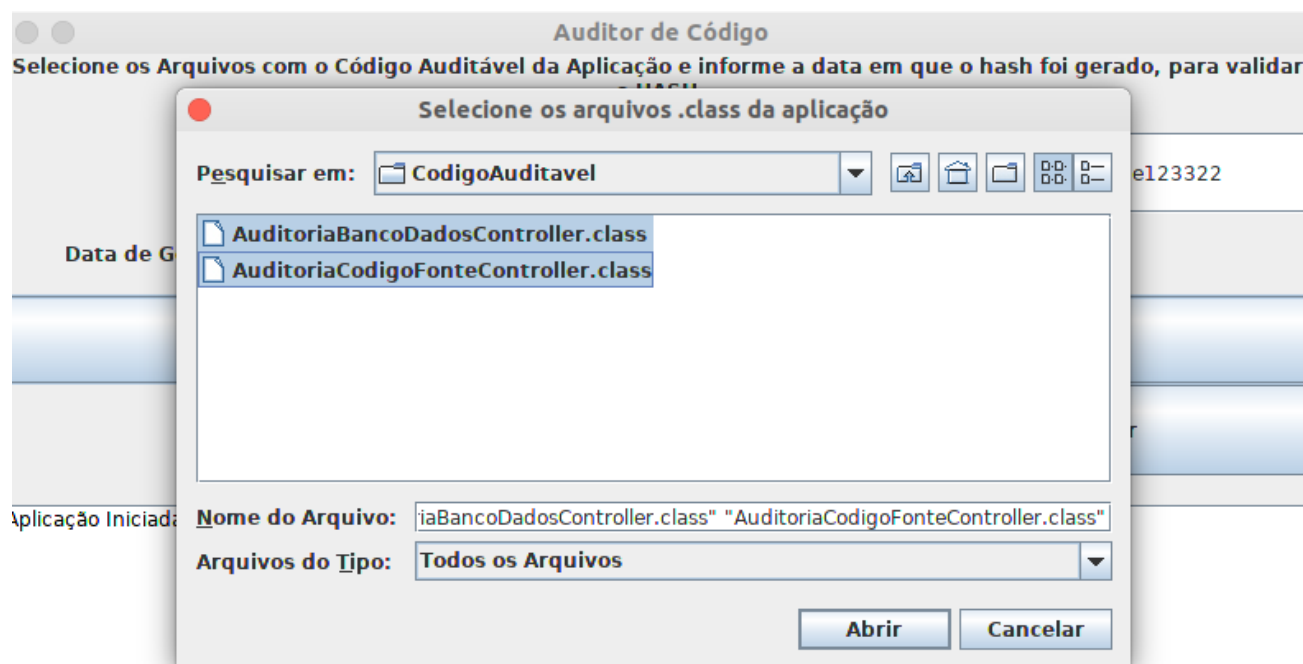
02/09/2020 09:52:34

Selecione os arquivos de Auditoria

Verificar

Limpar

Aplicação Iniciada ...



Agora clique no botão verificar que se o hash e a data forem informadas corretamente, uma mensagem “Hash válido” será mostrada.

Auditor de Código

Selecione os Arquivos com o Código Auditável da Aplicação e informe a data em que o hash foi gerado, para validar o HASH.

Hash:

fa949be4f37212a99dae42c745ac7c8bdb2bb0ceabe313fa089682634e123322

Data de Geração do Hash:

02/09/2020 09:52:34

Selecione os arquivos de Auditoria

Verificar

Limpar

O Hash calculado foi: fa949be4f37212a99dae42c745ac7c8bdb2bb0ceabe313fa089682634e123322

!!!! Hash Válido !!!!

Se quiser visualizar o código fonte auditável por uma forma externa ao sistema , acessar o site <http://www.javadecompilers.com/>.

Selecione um dos arquivos da auditoria e o **Procyon** como decompilador.

Escolher arquivo

Nenhum arquivo selecionado

Upload and Decompile

[Twitter](#)
[Facebook](#)
[Stumbleupon](#)
[LinkedIn](#)

Select a decompiler

☒ Procyon - fast and and well-supported decompiler for modern Java

☐ CFR - very good and well-supported decompiler for modern Java

Clique em “Upload e Decompile”, será mostrado o código fonte decompilado do sistema e ele poderá ser analisado.

```

</> AuditoriaCodigoFonteController.class

//
// Decompiled by Procyon v0.5.36
//

package br.ufrn.eligere.mvc.web.controller.auditoria;

import com.strobel.decompiler.ITextOutput;
import com.strobel.decompiler.Decompiler;
import java.io.Writer;
import com.strobel.decompiler.PlainTextOutput;
import java.io.OutputStreamWriter;
import com.strobel.decompiler.DecompilerSettings;
import com.google.common.collect.UnmodifiableIterator;
import com.google.common.reflect.ClassPath;
import java.util.ArrayList;
import java.util.zip.ZipEntry;
import java.io.OutputStream;
import java.util.zip.ZipOutputStream;
import java.io.ByteArrayOutputStream;

```

2.1 Utilizando o Código Fonte

dd.....