

PROCESSAMENTO DA FITA ESPELHO

DADOS ACESSADOS E CADASTRADOS PELO PROCESSAMENTO

*Esse é um documento técnico. Aqui não é explicado como o sistema é usado para esse caso.

Este documento trata de mostrar quais dados são acessados e cadastrados pelo processamento da fita espelho. Está baseado no processamento mais recente, após a reestruturação de classes e domínio.

As divisões aqui serão feitas por etapa do processamento, leia-se: carregamento do arquivo, leitura do arquivo da fita espelho, pré-processamento e processamento.

Características gerais

O novo processamento da fita espelho agora está sendo executado em cima de uma infraestrutura baseada no *spring-batch*, que auxilia na leitura do arquivo e na execução das etapas.

O layout do arquivo está definido no xml

“SIGRH\fita_espelho\br\ufrn\sigrh\fitaespelho\batch\fitaespelho-layout.xml”, que é usado pelo *spring-batch* para separar os campos do arquivo.

Uma nova estrutura de banco foi criada para dar suporte às novas funcionalidades, como o relatório de críticas mais detalhado, separado por nível de falha (alerta, erro), reprocessamento de servidores com falha. Esta nova estrutura se encontra no esquema *fita_espelho* do banco *administrativo*.

Carregamento do Arquivo

O carregamento é feito através da classe

`br.ufrn.sigrh.fitaespelho.processamento.jsf.UploadFitaEspelhoMBean`, método `carregarArquivo`. Nesse método o processador `br.ufrn.sigrh.fitaespelho.processamento.negocio.ProcessadorCarregamentoFitaEspelho` é chamado e é nele que o arquivo é inserido no banco de dados, *base_arquivos*.

Após a inserção do arquivo na base de dados, um registro é gerado na tabela *fita_espelho.fita_espelho* fazendo referência ao *id_arquivo* que foi inserido anteriormente.

Leitura do Arquivo

`br.ufrn.sigrh.fitaespelho.batch.LeituraDadosFitaEspelhoTasklet`

A leitura do arquivo é feita na tabela *public.arquivos*, do banco *base_arquivos*, através do *id_arquivo* armazenado em *fita_espelho.fita_espelho*.

Após a leitura, é feita uma busca por servidores já processados com sucesso nas tabelas *fita_espelho.processamento_fita_espelho* e *fita_espelho.processamento_servidor*.

Pré-processamento

`br.ufrn.sigrh.fitaespelho.batch.PreProcessamentoFitaEspelhoTasklet`

Nessa classe é inserido um registro em *fita_espelho.processamento_fita_espelho*, indicando que um novo processamento foi iniciado.

A tabela *fita_espelho.valores_padroes* é acessada para buscar e armazenar todos os valores padrões usados em casos que esses valores não existem no SIGRH.

Caso registros de atividades, rubricas, cargos ou ocorrências não sejam encontrados no SIGRH, o processamento usa os valores padrões para cadastrar essas entidades em:

- **rh.atividade:**
`br.ufrn.sigrh.fitaespelho.dao.DadosFuncionaisDao.inserirValorDefaultAtividade(List<Integer>)`
- **rh.cargo:**
`br.ufrn.sigrh.fitaespelho.dao.DadosFuncionaisDao.inserirValorDefaultCargo(List<Integer>, int, int)`
- **financeiro.rubrica:**
`br.ufrn.sigrh.fitaespelho.dao.DadosFinanceirosDao.inserirValorDefaultRubrica(List<Integer>, int)`
- **funcional.grupo_ocorrencia:**
`br.ufrn.sigrh.fitaespelho.dao.OcorrenciaDao.insereGrupoOcorrenciaDefault(List<Integer>)`
- **funcional.ocorrencia:**
`br.ufrn.sigrh.fitaespelho.dao.OcorrenciaDao.insereOcorrenciaDefault(String, int)`
- **funcional.associacao_grupo_ocorrencia:**
`br.ufrn.sigrh.fitaespelho.dao.OcorrenciaDao.inserirValorDefaultOcorrencia(List<Ocorrencia>, Map<Integer, Integer>)`

Ainda no pré-processamento, vários registros são buscados no banco de dados e armazenados na memória para servir como um cache. São eles:

- **rh.servidor.siape:**
`br.ufrn.sigrh.fitaespelho.dao.DadosFuncionaisDao.listAllMatriculasExistentes()`
- **funcional.estagiario.siape:**
`br.ufrn.sigrh.fitaespelho.dao.EstagiarioDao.listAllMatriculasExistentes()`
- **comum.estado.denominacao, comum.estado.id:**
`br.ufrn.sigrh.fitaespelho.batch.PreProcessamentoFitaEspelhoTasklet.carregarSiglasEstados()`
- **funcional.nivel_designacao.sigla, funcional.nivel_designacao.id_nivel_designacao:**

- ```
br.ufrn.sigrh.fitaespelho.batch.PreProcessamentoFitaEspelhoTasklet.
carregarNiveisDesignacao()
```
- **rh\_tipo.tipo\_estado\_civil.id\_tipo\_estado\_civil:**  
br.ufrn.sigrh.fitaespelho.batch.PreProcessamentoFitaEspelhoTasklet.
carregarEstadosCivis()
  - **rh.escolaridade.id\_escolaridade:**  
br.ufrn.sigrh.fitaespelho.batch.PreProcessamentoFitaEspelhoTasklet.
carregarEscolaridades()
  - **rh.formacao.id\_formacao:**  
br.ufrn.sigrh.fitaespelho.batch.PreProcessamentoFitaEspelhoTasklet.
carregarFormacoes()
  - **rh\_tipos.tipo\_nacionalidade.id\_tipo\_nacionalidade:**  
br.ufrn.sigrh.fitaespelho.batch.PreProcessamentoFitaEspelhoTasklet.
carregarNacionalidades()
  - **peessoal.pais.id\_pais:**  
br.ufrn.sigrh.fitaespelho.batch.PreProcessamentoFitaEspelhoTasklet.
carregarPaíses()
  - **rh\_tipos.tipo\_regime\_juridico.id\_tipo\_regime\_juridico,**  
**rh\_tipos.tipo\_regime\_juridico.sigla:**  
br.ufrn.sigrh.fitaespelho.batch.PreProcessamentoFitaEspelhoTasklet.
carregarRegimeJuridico()
  - **rh.situacao\_servidor.id\_situacao:**  
br.ufrn.sigrh.fitaespelho.batch.PreProcessamentoFitaEspelhoTasklet.
carregarSituacaoServidor()
  - **comum.banco.id:**  
br.ufrn.sigrh.fitaespelho.batch.PreProcessamentoFitaEspelhoTasklet.
carregarRegistrosBancos()
  - **rh\_tipos.tipo\_jornada\_trabalho.id\_tipo\_jornada\_trabalho:**  
br.ufrn.sigrh.fitaespelho.batch.PreProcessamentoFitaEspelhoTasklet.
carregarJornadaTrabalho()

Ao final do pré-processamento o total de registros de servidores é atualizado em:

```
fita_espelho.processamento_fita_espelho.total_registros.
```

## Processamento

Esta etapa pode ser realizada com a utilização de várias *threads*. O número de *threads* é definido no parâmetro: `NUMERO_THREADS_PROCESSAMENTO = "7_100100_84"`.

Além disso, o processamento é dividido por várias classes chamadas de *handlers*. Cada *handler* trata de uma parte dos dados do servidor e eles seguem o padrão “Chain of Responsibility”, ou seja, cada objeto é responsável por chamar o próximo a ser executado.

A seguir são mostrados as tabelas acessadas e cadastradas no banco por *handler*:

## **br.ufrn.sigrh.fitaespelho.chain.DadosPessoaisHandler**

Consulta apenas `comum.pessoa` e `rh.servidor`.

## **br.ufrn.sigrh.fitaespelho.chain.CadastrarDadosPessoaisHandler**

Insert em `comum.pessoa`. Sincroniza o registro entre os bancos

## **br.ufrn.sigrh.fitaespelho.chain.AtualizarDadosPessoaisHandler**

Update em `comum.pessoa` e insert em `peessoal.occurrencia_peessoal`

## **br.ufrn.sigrh.fitaespelho.chain.CadastraDadosFuncionaisServidorHandler**

Consulta em:

- **rh.servidor:**
  - `br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.p`  
`opularServidor(RegistroServidorFitaEspelho), trechos`  
`"daoFuncional.getIdServidorByMatricula" e`  
`"daoFuncional.getIdServidorByMatriculaOrigem"`
  - `br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.p`  
`opularUnidade(RegistroServidorFitaEspelho), trecho`  
`daoFuncional.getAtributoServidor`
  - `br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.p`  
`ersisteDadosFuncionais(RegistroServidorFitaEspelho), trecho`  
`"recuperarIdServidor"`
- **comum.unidade:**
  - `br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.p`  
`opularUnidade(RegistroServidorFitaEspelho), trechos`  
`"isRegistroCadastrado("Unidade", unidadeServidor)" e`  
`"daoFuncional.getIdUnidade"`
  - `br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.r`  
`ecuperarIdUnidade(RegistroServidorFitaEspelho)`
- **fita\_espelho.situacao\_servidor\_ativo:**  
`br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.getI`  
`dAtivoServidor(RegistroServidorFitaEspelho), trecho`  
`"daoFuncional.getSituacoesAtivas()"`
- **rh.cargo:**  
`br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.getC`  
`ategoriaFuncionalServidor(RegistroServidorFitaEspelho), techo`  
`"daoFuncional.getIdCategoria"`
- **rh.classe\_funcional:**  
`br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.getC`  
`lasseFuncionalServidor(RegistroServidorFitaEspelho), trechos`  
`"isRegistroCadastrado" e "daoFuncional.getIdClasseFuncional"`
- **funcional.servidor\_temporario:**  
`br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.veri`  
`ficarServidorTemporario(RegistroServidorFitaEspelho), trechos`  
`"daoFuncional.getIdServidorTemporarioAtivo" e`  
`"daoFuncional.getIdServidorServidorTemporario"`

- **comum.conta\_bancaria:**  
br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.atualizaDadosBancarios(RegistroServidorFitaEspelho), trecho  
"daoPessoa.atualizaDadosBancarios"
- **comum.conta\_pessoal:**  
br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.atualizaDadosBancarios(RegistroServidorFitaEspelho), trecho  
"daoPessoa.atualizaDadosBancarios"
- **concurso.vaga:**  
br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.geraHistoricoServidor(RegistroServidorFitaEspelho), trecho  
"isRegistroCadastrado("Vaga",  
reg.getRegFuncional().getCodigoVaga())"
- **funcional.historico\_servidor:**  
br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.geraHistoricoServidor(RegistroServidorFitaEspelho), trechos  
"daoFuncional.isHistoricoVagaServidor" e  
"daoFuncional.getIdHistoricoServidor"

Inserção em:

- **rh.servidor:**  
br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.persisteDadosFuncionais(RegistroServidorFitaEspelho), trechos  
"daoFuncional.inseraNovoServidor" e  
"daoFuncional.sincronizaDadosServidor"
- **funcional.ocorrencia\_funcional:**
  - br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.persisteDadosFuncionais(RegistroServidorFitaEspelho), trecho  
"daoOcorrencia.insereOcorrenciasFuncionais"
- **pessoal.ocorrencia\_pessoal:**
  - br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.persisteDadosPessoais(RegistroServidorFitaEspelho), trecho  
"ocorrenciaDao.insereOcorrenciasPessoais"
- **comum.conta\_bancaria:**  
br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.atualizaDadosBancarios(RegistroServidorFitaEspelho), trecho  
"daoPessoa.atualizaDadosBancarios"
- **comum.conta\_pessoal:**  
br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.atualizaDadosBancarios(RegistroServidorFitaEspelho), trecho  
"daoPessoa.atualizaDadosBancarios"
- **concurso.vaga:**  
br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.geraHistoricoServidor(RegistroServidorFitaEspelho), trecho  
"daoVagaConcurso.adicionarVagaConcurso"
- **funcional.historico\_servidor:**  
br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.geraHistoricoServidor(RegistroServidorFitaEspelho), trechos

```
"daoFuncional.insererHistoricoServidor"
```

Atualização em:

- **rh.servidor:**
  - `br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.persisteDadosFuncionais(RegistroServidorFitaEspelho), trechos`  
`"daoFuncional.atualizaCamposServidor"`
- **funcional.servidor\_temporario:**
  - `br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.persisteDadosFuncionais(RegistroServidorFitaEspelho), trechos`  
`"daoFuncional.atualizaServidorTemporario"`
- **comum.pessoa:**
  - `br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.persisteDadosPessoais(RegistroServidorFitaEspelho), trechos`  
`"dao.atualizarDadosPessoais"`
- **funcional.historico\_servidor:**  
`br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisServidorHandler.geraHistoricoServidor(RegistroServidorFitaEspelho), trechos`  
`"daoFuncional.atualizaHistoricoServidor"`

### **br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisServidorHandler**

Difere do cadastro (classe anterior) na chamada do método "validaAtualizacaoServidor", que no cadastrado é chamado o "inserirServidor". Para todos os outros tem o mesmo comportamento.

Consulta em:

- **rh.servidor:**
  - `br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisServidorHandler.validaAtualizacaoServidor(RegistroServidorFitaEspelho), trechos` `"daoFuncional.getAtributoStringServidor",`  
`"getAtributoServidor", "getAtributoDateServidor",`  
`"getAtributoBooleanServidor"`
  - `br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisServidorHandler.processarRegistroVagaConcurso(RegistroServidorFitaEspelho), trecho` `"getAtributoServidor"`
- **funcional.occurencia:**
  - `br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisServidorHandler.validaAtualizacaoServidor(RegistroServidorFitaEspelho), trecho` `"daoOccurencia.getProvimento"`
  - `br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisServidorHandler.processarSuspensaoPagamento(RegistroServidorFitaEspelho), trecho` `"daoOccurencia.getProvimento"`
  - `br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisServidorH`

```
andler.verificarMudancaCargo(RegistroServidorFitaEspelho),
trecho "daoOcorrencia.getProvimento"
```

- **funcional.dados\_complementares\_servidor:**  
br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisServidorHandler.processarRegistroVagaConcurso(RegistroServidorFitaEspelho),  
trecho "daoVagaConcurso.hasIdVagaDadosComplementares"
- **funcional.occorrencia\_funcional:**
  - br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisServidorHandler.processarSuspensaoPagamento(RegistroServidorFitaEspelho),  
trecho "daoOcorrencia.getIdOcorrenciaFuncional"
  - br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisServidorHandler.verificarMudancaCargo(RegistroServidorFitaEspelho), trechos  
"isRegistroCadastrado("OcorrenciaFuncional", ...)" e  
"daoOcorrencia.getIdOcorrenciaFuncional"

Inserção em:

- **funcional.occorrencia\_funcional:**
  - br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisServidorHandler.validaAtualizacaoServidor(RegistroServidorFitaEspelho),  
trecho "insereOcorrenciaFuncional"
  - br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisServidorHandler.processarSuspensaoPagamento(RegistroServidorFitaEspelho),  
trecho "insereOcorrenciaFuncional"
  - br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisServidorHandler.verificarMudancaCargo(RegistroServidorFitaEspelho), trecho  
"insereOcorrenciaFuncional"
- **peessoal.occorrencia\_peessoal:**  
br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisServidorHandler.validaAtualizacaoServidor(RegistroServidorFitaEspelho), trecho  
"insereOcorrenciaPessoal"
- **funcional.dados\_complementares\_servidor:**  
br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisServidorHandler.processarRegistroVagaConcurso(RegistroServidorFitaEspelho),  
trecho "daoVagaConcurso.adicionarDadosComplementares"

Atualização em:

- **funcional.dados\_complementares\_servidor:**  
br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisServidorHandler.processarRegistroVagaConcurso(RegistroServidorFitaEspelho),  
trecho "daoVagaConcurso.atualizarVagaDadosComplementares"

**br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisServidorHandler**

Insert em funcional.historico\_dados\_pessoais.

## **br.ufrn.sigrh.fitaespelho.chain.CadastraDadosFuncionaisEstagiarioHandler**

Consulta em:

- **funcional.estagiario:**  
`br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisEstagiarioHandler.processaDadosComum(RegistroServidorFitaEspelho), trecho "daoEstagiario.getIdEstagiario"`
- **comum.unidade:**  
`br.ufrn.sigrh.fitaespelho.chain.DadosFuncionaisEstagiarioHandler.recuperarUnidade(RegistroServidorFitaEspelho), trecho "daoFuncional.getIdUnidade"`

Inserção em:

- **funcional.estagiario:**  
`br.ufrn.sigrh.fitaespelho.chain.CadastraDadosFuncionaisEstagiarioHandler.insereEstagiario(RegistroServidorFitaEspelho), trecho "daoEstagiario.insereEstagiario"`
- **comum.conta\_bancaria:**  
`br.ufrn.sigrh.fitaespelho.chain.CadastraDadosFuncionaisEstagiarioHandler.atualizaDadosBancarios(RegistroServidorFitaEspelho), trecho "daoPessoa.atualizaDadosBancarios"`
- **comum.conta\_pessoa:**  
`br.ufrn.sigrh.fitaespelho.chain.CadastraDadosFuncionaisEstagiarioHandler.atualizaDadosBancarios(RegistroServidorFitaEspelho), trecho "daoPessoa.atualizaDadosBancarios"`

## **br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisEstagiarioHandler**

Consulta em:

- **funcional.estagiario:**
  - `br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisEstagiarioHandler.validaDadosFuncionaisEstagiario(RegistroServidorFitaEspelho), trechos "daoEstagiario.getAtributoStringEstagiario", "daoEstagiario.getAtributoDateEstagiario", "daoEstagiario.getAtributoEstagiario", "daoEstagiario.getAtributoBooleanEstagiario"`
  - `br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisEstagiarioHandler.processaOcorrenciaExclusao(RegistroServidorFitaEspelho), trecho "daoEstagiario.getAtributoDateEstagiario"`

Atualiza em:

- **comum.pessoa:**  
`br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisEstagiarioHandler.persisteDadosAtualizacao(RegistroServidorFitaEspelho), trecho "daoPessoa.atualizarDadosPessoais"`
- **funcional.estagiario:**  
`br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFuncionaisEstagiarioHa`



```
andler.persisteDadosAtualizacao(RegistroServidorFitaEspelho), trecho
"daoEstagiario.atualizaDadosEstagiario"
```

## **br.ufrn.sigrh.fitaespelho.chain.RegistrarUnidadeServidorHandler**

Consulta em:

- **comum.unidade:**
  - `br.ufrn.sigrh.fitaespelho.chain.RegistrarUnidadeServidorHandler.recuperarLocalizacao(RegistroServidorFitaEspelho), trecho`  
`"daoFuncional.getIdUnidade"`
  - `br.ufrn.sigrh.fitaespelho.chain.RegistrarUnidadeServidorHandler.registrarUnidadeServidor(RegistroServidorFitaEspelho), trecho`  
`"isRegistroCadastrado("Unidade", ...)"`
- **rh.servidor:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarUnidadeServidorHandler.registrarUnidadeServidor(RegistroServidorFitaEspelho), trecho`  
`"daoFuncional.getAtributoServidor"`
- **funcional.ocorrencia:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarUnidadeServidorHandler.registrarUnidadeServidor(RegistroServidorFitaEspelho), trecho`  
`"daoOcorrencia.getProvimento"`
- **funcional.ocorrencia\_funcional:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarUnidadeServidorHandler.registrarUnidadeServidor(RegistroServidorFitaEspelho), trechos`  
`"isRegistroCadastrado("OcorrenciaFuncional", ...)" e`  
`"daoOcorrencia.getIdOcorrenciaFuncional"`

Inserção em:

- **funcional.ocorrencia\_funcional:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarUnidadeServidorHandler.registrarUnidadeServidor(RegistroServidorFitaEspelho), trecho`  
`"daoOcorrencia.insereOcorrenciaFuncional"`
- **funcional.historico\_lotacao:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarUnidadeServidorHandler.registrarUnidadeServidor(RegistroServidorFitaEspelho), trecho`  
`"daoFuncional.insereHistoricoLotacao"`

Atualização em:

- **rh.servidor:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarUnidadeServidorHandler.registrarUnidadeServidor(RegistroServidorFitaEspelho), trecho`  
`"daoFuncional.atualizaCamposServidor"`

## **br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaInatividadeHandler**

Consulta em:

- **funcional.ocorrencia, funcional.associacao\_grupo\_ocorrencia, funcional.grupo\_ocorrencia:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaInatividadeHandler.registrarOcorrenciaInatividade(RegistroServidorFitaEspelho),  
trechos "daoOcorrencia.getIdOcorrencia",  
"daoOcorrencia.getIdGrupoOcorrencia",  
"daoOcorrencia.existeChaveOcorrencia"`
- **rh.servidor:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaInatividadeHandler.registrarOcorrenciaInatividade(RegistroServidorFitaEspelho),  
trecho "daoFuncional.getAtributoDateServidor"`
- **funcional.inatividade:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaInatividadeHandler.registrarOcorrenciaInatividade(RegistroServidorFitaEspelho),  
trechos "isRegistroCadastrado("Inatividade", ...)" e  
"daoOcorrencia.getIdInatividade(regFunc.getIdServidor())"`
- **rh\_tipos\_tipo\_documento:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaInatividadeHandler.registrarOcorrenciaInatividade(RegistroServidorFitaEspelho),  
trecho "isRegistroCadastrado("TipoDocumento", ...)"`
- **comum.unidade:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaInatividadeHandler.registrarOcorrenciaInatividade(RegistroServidorFitaEspelho),  
trecho "isRegistroCadastrado("Unidade", ...)"`

Inserção em:

- **funcional.inatividade:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaInatividadeHandler.registrarOcorrenciaInatividade(RegistroServidorFitaEspelho),  
trecho "daoOcorrencia.insereInatividade"`

Atualização em:

- **rh.servidor:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaInatividadeHandler.registrarOcorrenciaInatividade(RegistroServidorFitaEspelho),  
trecho "daoFuncional.atualizaCamposServidor"`

## **br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaExclusaoHandler**

Consulta em:

- **rh.servidor:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaExclusaoHandler.processar(RegistroServidorFitaEspelho), trecho`

```
"daoFuncional.getAtributoDateServidor"
```

Atualização em:

- **rh.servidor:**  
bbr.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaExclusaoHandler.  
.processar(RegistroServidorFitaEspelho), trecho  
"daoFuncional.atualizaCamposServidor"

## **br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaAfastamentoHandler**

Consulta em:

- **funcional.ocorrencia, funcional.associacao\_grupo\_ocorrencia, funcional\_grupo\_ocorrencia:**  
br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaAfastamentoHandler.  
registrarOcorrenciaAfastamento(RegistroServidorFitaEspelho),  
trechos "daoOcorrencia.getIdOcorrencia",  
"daoOcorrencia.getIdGrupoOcorrencia",  
"daoOcorrencia.existeChaveOcorrencia"
- **funcional.ausencia:**
  - br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaAfastamentoHandler.  
isAusenciaServidorCadastrada(RegistroServidorFitaEspelho,  
Integer, Integer, Date, int), trecho "daoOcorrencia.getIdAusencia"
  - br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaAfastamentoHandler.  
registrarNovoAfastamento(RegistroServidorFitaEspelho,  
Integer, Integer, Date, int), trecho "daoOcorrencia.hasAusencias"
- **rh\_tipos\_tipo\_documento:**  
br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaAfastamentoHandler.  
registrarNovoAfastamento(RegistroServidorFitaEspelho, Integer,  
Integer, Date, int), trecho  
"isRegistroCadastrado("TipoDocumento", ...)"

Inserção em:

- **funcional.ausencia:**  
br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaAfastamentoHandler.  
registrarNovoAfastamento(RegistroServidorFitaEspelho, Integer,  
Integer, Date, int), trecho "daoOcorrencia.insereAusencia"

Atualização em:

- **funcional.ausencia:**  
br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaAfastamentoHandler.  
isAusenciaServidorCadastrada(RegistroServidorFitaEspelho,  
Integer, Integer, Date, int), trecho "daoOcorrencia.atualizaAusencia"

## **br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaObitoHandler**

Consulta em:

- **rh.servidor:**  
br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaObitoHandler.reg  
istrarOcorrenciaObito(RegistroServidorFitaEspelho), trecho  
"daoFuncional.getAtributoDateServidor"
- **funcional.obito:**  
br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaObitoHandler.reg  
istrarOcorrenciaObito(RegistroServidorFitaEspelho), trechos  
"isRegistroCadastrado("Obito", ...)" e "daoOcorrencia.getIdObito"
- **funcional.ocorrencia, funcional.associacao\_grupo\_ocorrencia,  
funcional.grupo\_ocorrencia:**  
br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaObitoHandler.reg  
istrarOcorrenciaObito(RegistroServidorFitaEspelho), trecho  
"daoOcorrencia.getIdOcorrencia",  
"daoOcorrencia.existeChaveOcorrencia"

Inserção em:

- **funcional.obito:**  
br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaObitoHandler.reg  
istrarOcorrenciaObito(RegistroServidorFitaEspelho), trecho  
"daoOcorrencia.insereObito"

Atualização em:

- **rh.servidor:**  
br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaObitoHandler.reg  
istrarOcorrenciaObito(RegistroServidorFitaEspelho), trecho  
"daoFuncional.atualizaCamposServidor"

**br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaModificacaoFuncionalHandler,  
br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaPosseHandler,  
br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaReversaoHandler**

Consulta em:

- **funcional.ocorrencia, funcional.associacao\_grupo\_ocorrencia,  
funcional.grupo\_ocorrencia:**  
br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaFuncionalHandler  
.processar(RegistroServidorFitaEspelho), trecho  
"daoOcorrencia.getIdOcorrencia",  
"daoOcorrencia.existeChaveOcorrencia"
- **funcional.ocorrencia\_funcional:**  
br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaFuncionalHandler  
.processar(RegistroServidorFitaEspelho), trechos  
"isRegistroCadastrado("OcorrenciaFuncional", ...)" e  
"daoOcorrencia.getIdOcorrenciaFuncional"
- **rh.tipos\_tipo\_documento:**  
br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaFuncionalHandler  
.processar(RegistroServidorFitaEspelho), trecho  
"isRegistroCadastrado("TipoDocumento", ...)"

- **funcional.ocorrencia:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaFuncionalHandler  
.processar(RegistroServidorFitaEspelho), trechos  
"daoOcorrencia.getProvimento" e "daoOcorrencia.findByPrimaryKey"`

Inserção em:

- **funcional.ocorrencia\_funcional:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarOcorrenciaFuncionalHandler  
.processar(RegistroServidorFitaEspelho), trecho  
"daoOcorrencia.insereOcorrenciaFuncional"`

## **br.ufrn.sigrh.fitaespelho.chain.RegistrarVinculoServidorHandler**

Consulta em:

- **funcional.ocorrencia, funcional.associacao\_grupo\_ocorrencia, funcional.grupo\_ocorrencia:**
  - `br.ufrn.sigrh.fitaespelho.chain.RegistrarVinculoServidorHandler.  
registrarVinculoFederal(RegistroServidorFitaEspelho), trecho  
"recuperarOcorrencia"`
  - `br.ufrn.sigrh.fitaespelho.chain.RegistrarVinculoServidorHandler.  
encerrarVinculoFederal(int, int, String, Date, Integer), trecho  
"daoOcorrencia.getIdOcorrencia"`
- **funcional.vinculo\_trabalho:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarVinculoServidorHandler.reg  
istrarVinculoFederal(RegistroServidorFitaEspelho), trecho  
"daoVinculo.getIdVinculoFederal"`
- **rh.servidor:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarVinculoServidorHandler.reg  
istrarVinculoFederal(RegistroServidorFitaEspelho), trecho  
"daoFuncional.getAtributoDateServidor"`
- **funcional.desligamento\_vinculo:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarVinculoServidorHandler.enc  
errarVinculoFederal(int, int, String, Date, Integer), trecho  
"daoVinculo.getIdDesligamentoVinculo"`

Inserção em:

- **funcional.vinculo\_trabalho:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarVinculoServidorHandler.reg  
istrarVinculoFederal(RegistroServidorFitaEspelho), trecho  
"daoVinculo.insereVinculoFederal"`
- **funcional.provimento\_vinculo:**  
`br.ufrn.sigrh.fitaespelho.chain.RegistrarVinculoServidorHandler.reg  
istrarVinculoFederal(RegistroServidorFitaEspelho), trecho  
"daoVinculo.insereProvimentoVinculo"`
- **funcional.desligamento\_vinculo:**

```
br.ufrn.sigrh.fitaespelho.chain.RegistrarVinculoServidorHandler.encerrarVinculoFederal(int, int, String, Date, Integer), trecho
"daoVinculo.insereDesligamentoVinculo"
```

Atualização em:

- **funcional.vinculo\_trabalho:**  
br.ufrn.sigrh.fitaespelho.chain.RegistrarVinculoServidorHandler.registrarVinculoFederal(RegistroServidorFitaEspelho), trecho  
"daoVinculo.atualizarVinculoFederal"
- **funcional.provimento\_vinculo:**  
br.ufrn.sigrh.fitaespelho.chain.RegistrarVinculoServidorHandler.registrarVinculoFederal(RegistroServidorFitaEspelho), trecho  
"daoVinculo.atualizarProvimentoVinculo"
- **funcional.desligamento\_vinculo:**  
br.ufrn.sigrh.fitaespelho.chain.RegistrarVinculoServidorHandler.encerrarVinculoFederal(int, int, String, Date, Integer), trecho  
"daoVinculo.atualizarDesligamentoVinculo"

**br.ufrn.sigrh.fitaespelho.chain.AtualizaDesignacaoCadastradaHandler,**  
**br.ufrn.sigrh.fitaespelho.chain.AtualizaDesignacaoIniciadaHandler**

Consulta em:

- **rh.servidor:**
  - br.ufrn.sigrh.fitaespelho.chain.DesignacaoGeralHandler.getIdServidor(Integer, Integer)
  - br.ufrn.sigrh.fitaespelho.chain.DesignacaoGeralHandler.recuperarAtributoServidor(String, int)
- **comum.unidade:**
  - br.ufrn.sigrh.fitaespelho.chain.DesignacaoGeralHandler.recuperarUnidadeByCodigo(int)
  - br.ufrn.sigrh.fitaespelho.chain.AtualizaDesignacaoCadastradaHandler.processarDesignacaoCadastrada(RegistroServidorFitaEspelho, DesignacaoFuncional), trecho  
"isRegistroCadastrado("Unidade", funcao.getIdUnidade())"
  - br.ufrn.sigrh.fitaespelho.chain.DesignacaoGeralHandler.getDescricaoUnidade(int)
- **rh.designacao:**
  - br.ufrn.sigrh.fitaespelho.chain.DesignacaoGeralHandler.recuperarDesignacao(Integer, Date, Integer), chamado por  
"getDesignacaoCadastrada" em "recuperarDesignacao"
  - br.ufrn.sigrh.fitaespelho.chain.DesignacaoGeralHandler.getDesignacaoById(int)
  - br.ufrn.sigrh.fitaespelho.dao.DesignacaoFuncionalDao.getIdResponsabilidadeUnidadeByDesignacao(int), trecho

`"getResponsabilidadeByDesignacao"`

- **rh.atividade:**
  - `br.ufrn.sigrh.fitaespelho.chain.DesignacaoGeralHandler.recuperarAtividade(RegistroServidorFitaEspelho)`
  - `br.ufrn.sigrh.fitaespelho.chain.DesignacaoGeralHandler.recuperarNivelByAtividade(int)`
  - `br.ufrn.sigrh.fitaespelho.chain.DesignacaoGeralHandler.getDescricaoAtividade(int)`
- **comum.responsavel\_unidade:**
  - `br.ufrn.sigrh.fitaespelho.dao.DesignacaoFuncionalDao.getIdResponsabilidadeUnidadeByDesignacao(int), trecho "getResponsabilidadeComumByDesignacao"`
  - `br.ufrn.sigrh.fitaespelho.chain.DesignacaoGeralHandler.isResponsabilidadeAtiva(Integer, Integer, String, Date)`

Inserção em:

- **comum.unidade:**  
`br.ufrn.sigrh.fitaespelho.chain.DesignacaoGeralHandler.atualizaUnidade(int, int), chamado por "atualizaUnidade"`
- **rh.designacao:**  
`br.ufrn.sigrh.fitaespelho.chain.AtualizaDesignacaoCadastradaHandler.popularBaseDeDados(RegistroServidorFitaEspelho, Integer, String), trecho "dao.inserirDesignacao"`
- **comum.responsavel\_unidade:**  
`br.ufrn.sigrh.fitaespelho.chain.DesignacaoGeralHandler.sincronizarResponsavelUnidade(RegistroFuncional, Funcao, String)`

Atualização em:

- **rh.servidor:**  
`br.ufrn.sigrh.fitaespelho.chain.AtualizaDesignacaoCadastradaHandler.popularBaseDeDados(RegistroServidorFitaEspelho, Integer, String)` e `br.ufrn.sigrh.fitaespelho.chain.AtualizaDesignacaoIniciadaHandler.popularBaseDeDados(RegistroServidorFitaEspelho), trecho "dao.funcional.atualizaCamposServidorJDBC"`
- **comum.responsavel\_unidade:**  
`br.ufrn.sigrh.fitaespelho.chain.DesignacaoGeralHandler.sincronizarResponsavelUnidade(RegistroFuncional, Funcao, String)`
- **rh.designacao:**  
`br.ufrn.sigrh.fitaespelho.chain.AtualizaDesignacaoCadastradaHandler.popularBaseDeDados(RegistroServidorFitaEspelho, Integer, String)` e `br.ufrn.sigrh.fitaespelho.chain.AtualizaDesignacaoIniciadaHandler.popularBaseDeDados(RegistroServidorFitaEspelho), trecho "dao.atualizaDesignacoes"`

## **br.ufrn.sigrh.fitaespelho.chain.EncerrarRegistrosFuncionaisHandler**

Consulta em:

- **rh.servidor:**  
`br.ufrn.sigrh.fitaespelho.chain.EncerrarRegistrosFuncionaisHandler.recuperarDataInativacao(RegistroServidorFitaEspelho)`
- **financeiro.adicional, rh\_tipos.tipo\_adicional\_percentual, rh\_tipos.tipo\_adicional:**  
`br.ufrn.sigrh.fitaespelho.chain.EncerrarRegistrosFuncionaisHandler.encerrarAdicionais(RegistroServidorFitaEspelho), trecho`  
`"daoFuncional.getAdicionaisByServidor"`
- **funcional. ausencia:**  
`br.ufrn.sigrh.fitaespelho.chain.EncerrarRegistrosFuncionaisHandler.encerrarAusencias(RegistroServidorFitaEspelho), trecho`  
`"daoFuncional.getAusenciasByServidor"`

Atualização em:

- **financeiro.adicional:**  
`br.ufrn.sigrh.fitaespelho.chain.EncerrarRegistrosFuncionaisHandler.encerrarAdicionais(RegistroServidorFitaEspelho), trecho`  
`"daoFuncional.finalizarAdicional"`
- **funcional. ausencia:**  
`br.ufrn.sigrh.fitaespelho.chain.EncerrarRegistrosFuncionaisHandler.encerrarAusencias(RegistroServidorFitaEspelho), trecho`  
`"daoFuncional.finalizarAusencias"`

## **br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFinanceirosHandler**

Consulta em:

- **rh.servidor:**  
`br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFinanceirosHandler.getIdServidor(RegistroFuncional)`
- **financeiro.rubrica:**  
`br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFinanceirosHandler.processar(RegistroServidorFitaEspelho), trecho`  
`"isRegistroCadastrado("Rubrica", ...)"`
- **financeiro.ficha\_financeira:**  
`br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFinanceirosHandler.recuperarFichaFinanceira(RegistroServidorFitaEspelho, RegistroFinanceiro)`

Inserção em:

- **financeiro.ficha\_financeira:**  
`br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFinanceirosHandler.popularBaseDeDados(RegistroServidorFitaEspelho), trecho`  
`"daoFinanceiro.persisteDadosFichaFinanceira"`

Atualização em:



- **rh.servidor:**  
`br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFinanceirosHandler.atualizarRegistroServidor(RegistroServidorFitaEspelho)`
- **capacitacao.item\_requisicao\_curso\_concurso:**  
`br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFinanceirosHandler.popularBaseDeDados(RegistroServidorFitaEspelho), trecho  
"daoFinanceiro.atualizaStatusPagamentoCursoConcurso"`
- **financeiro.ficha\_financeira:**  
`br.ufrn.sigrh.fitaespelho.chain.AtualizaDadosFinanceirosHandler.popularBaseDeDados(RegistroServidorFitaEspelho), trecho  
"daoFinanceiro.persisteDadosFichaFinanceira"`

## Tópico especial: atalhos no eclipse

Ctrl+Shift+R: Procurar por algum recurso, leia-se classe, arquivos xml, arquivos *properties*, etc.

Ctrl+O: Procurar por um método dentro de uma classe.

Ctrl+F: Procurar por um texto dentro de uma classe.

Ctrl+H: Procurar por textos em todos o workspace (aba File Search)