**DataJud Miner**

**VISÃO GERAL**

**DataJud Miner** é um projeto desenvolvido na linguagem R, baseado em open source, para resolver o Desafio 1 do ciclo de inovação aberta *Desafios CNJ Inova - Ciência de Dados e Inteligência Artificial*, proposto pelo Conselho Nacional de Justiça em conjunto com a Escola Nacional de Administração Pública - Enap.

<https://repositorio.enap.gov.br/bitstream/1/5354/1/Edital%20120%20set%202020.pdf>

O Desafio 1 consiste em apresentar soluções para, a partir da base do Datajud, identificar padrões e comparar o andamento de processos em cada unidade judiciária do Brasil, levando em consideração as peculiaridades locais e o nível de complexidade, em razão da competência e da matéria do direito.

<https://www.cnj-inova.com/home>

**OBJETIVO**

Solução que viabiliza responder as perguntas críticas sobre os processos de trabalho finalísticos do judiciário, visando identificar pontos críticos, oportunidades de melhoria e boas práticas:

1. Quais unidades judiciais atuam em matérias semelhantes
2. Quais os rituais dos processos de trabalho de cada matéria, em especial de matérias semelhantes nas diversas unidades judiciais
3. Quais as diferenças de prazo de atuação entre as unidades judiciais que atuam em matéria semelhante
4. Quais as causas das diferenças de prazo de atuação entre as unidades judiciais que atuam em matéria semelhante

**MÉTRICAS**

1. Total de processos (caso)
2. Total de movimentos
3. Tipos de movimentos
4. Tipos de fluxos
5. Perfis de responsáveis pelos movimentos
6. Tempo de Tramitação Geral
7. Tempo de Tramitação por Processo (caso)
8. Frequência de Movimentos (atividades)
9. Atividades iniciadoras
10. Cobertura de fluxos
11. Atividades do fluxo
12. Mapa de frequência do processo (absoluta)
13. Mapa de frequência do processo (relativa)
14. Mapa de frequência do processo (mediana em dias)
15. Mapa com Animação
16. Fluxos frequentes

**Fonte:** [**https://www.bupar.net/exploring.html**](https://www.bupar.net/exploring.html)

**INSTALAÇÃO**

1. Baixar e instalar o R: https://cran-r.c3sl.ufpr.br/
2. Baixar e instalar RStudio: https://rstudio.com/products/rstudio/download/
3. Baixar e descompactar o código do Github:
4. Extrair pasta Dados (com scripts). Atenção: não fazer nenhuma alteração na estrutura ou nomes
5. Entrar no RStudio e abrir o script install\_packages.R
6. Teclar CTRL+SHIFT+ENTER e aguardar consluir a instlação dos pacotes (ícone deixar de ser vermelho na janela de execução)
7. Abrir RStudio, acessar Tools - Global Options - Code - Saving e optar Default text encoding para UTF-8
8. Abrir o arquivo script app-Otimizador.R
9. Alterar o caminho dos dados para o caminho onde foi baixado o pacote de dados, até a pasta "dados" - com BARRA NORMAL (linha 38 Ex: diretorio <- 'C:/DadosCNJ/dados/')
10. Salvar script app-Otimizador.R no RStudio (CTRL+S ou File-Save)
11. Clicar em Run App

**OPERAÇÃO**

Ao concluir a instalação e clicar em “Run App”, já será aberta uma tela de navegação onde se pode selecionar a cobertura dos dados no menu lateral esquerdo e em seguida selecionar os modelos desejados no menu superior da janela.

**DESENVOLVIMENTO**

**Processo de desenvolvimento**

KDD (*knowledge discovery in databases*):

1. **Seleção**: esta etapa consiste em selecionar um conjunto ou subconjunto de dados que farão parte da análise. As fontes de dados podem ser variadas (planilhas, sistemas gerenciais, data warehouses) e possuir dados com formatos diferentes (estruturados, semiestruturados e não-estruturados).
2. **Pré-Processamento**: esta etapa consiste em fazer a verificação da qualidade dos dados armazenados. A base passa por um processo de limpar, corrigir ou remover dados inconsistentes, verificar dados ausentes ou incompletos, identificar anomalias (outliers).
3. **Transformação**: esta etapa consiste em aplicar técnicas de transformação como: normalização, agregação, criação de novos atributos, redução e sintetização dos dados. Aqui os dados ficam disponíveis agrupados em um mesmo local para a aplicação dos modelos de análise.
4. **Mineração de Dados**: esta etapa consiste em construir modelos ou aplicar técnicas de mineração de dados. Essas técnicas têm por objetivo (1) verificar uma hipótese, (2) descobrir novos padrões de forma autônoma. Além disso, a descoberta pode ser dividida em: preditiva e descritiva. Esses modelos geralmente são aplicados e refeitos inúmeras vezes dependendo do objetivo do projeto.
5. **Interpretação e Avaliação**: esta etapa consiste em avaliar o desempenho do modelo, aplicando em cima de dados que não foram utilizados na fase de treinamento ou mineração. A validação pode ser feita de diversas formas, algumas delas são: utilizar medidas estatísticas, passar pela avaliação dos profissionais de negócio.

**Recursos utilizados**

1. **Seleção:**

Foram analisados os arquivos disponibilizados pelo CNJ em <https://www.cnj.jus.br/owncloud/index.php/s/yIby5NidzxB1sQ8> *(senha: @cI4AGgsXwVa)*. De acordo com o glossário tratam-se de dados oriundos de diversos sistemas do judiciário.

1. **Pré-processamento e Transformação**
   * Conversão dos arquivos em formato não-relacional (.json) para relacional (.csv)
   * Identificação de problemas nas estruturas dos arquivos .json
   * Identificação de problemas na falta de utilização de tabelas do CNJ
   * Inclusão de novo atributo identificador único de processo (número mais classe do processo)
   * Inclusão de novo atributo identificador dos processos com baixa definitiva (arquivados) e filtro de seleção
   * Normalização dos formatos de data e hora dos atributos de data/hora Ajuizamento e data/hora Movimento
   * Inclusão de novo atributo tipo\_responsável com a identificação de Magistrado ou do perfil de Serventuário (Arquivista, Auxiliar da Justiça, Contador, Distribuidor, Escrivão/Diretor de Secretaria/Secretário Jurídico e Oficial de Justiça)
   * arquivos e scripts disponibilizados no GitHub :
     + R-Project (GNU Affero General Public License)
     + RStudio (GNU Affero General Public License)
     + Hmisc ([GPL-2](https://cran.r-project.org/web/licenses/GPL-2) | [GPL-3](https://cran.r-project.org/web/licenses/GPL-3) [expanded from: GPL (≥ 2)]) - concise Statistical Description of a Vector, Matrix, Data…;
     + tidyr([MIT](https://cran.r-project.org/web/licenses/MIT) + file [LICENSE](https://cran.r-project.org/web/packages/tidyr/LICENSE)) - help you create tidy data;
     + RJSONIO([BSD\_3\_clause](https://cran.r-project.org/web/licenses/BSD_3_clause) + file [LICENSE](https://cran.r-project.org/web/packages/RJSONIO/LICENSE)) - allows conversion to and from data in Javascript object notation (JSON) format;
     + xlsx([GPL-3](https://cran.r-project.org/web/licenses/GPL-3)) - to read, write and format Excel files;
     + dplyr([MIT](https://cran.r-project.org/web/licenses/MIT) + file [LICENSE](https://cran.r-project.org/web/packages/dplyr/LICENSE)) - grammar of data manipulation;
     + purrr([GPL-3](https://cran.r-project.org/web/licenses/GPL-3) | file [LICENSE](https://cran.r-project.org/web/packages/purrr/LICENSE)) - set of tools for working with functions and vectors;
     + klaR([GPL-2](https://cran.r-project.org/web/licenses/GPL-2) | [GPL-3](https://cran.r-project.org/web/licenses/GPL-3)) - functions for classification and visualization;
     + factoextra([GPL-2](https://cran.r-project.org/web/licenses/GPL-2)) - extract and visualize the output of multivariate data analyses;
     + reshape2([MIT](https://cran.r-project.org/web/licenses/MIT) + file [LICENSE](https://cran.r-project.org/web/packages/reshape2/LICENSE)) - transform data between wide and long formats;
     + heuristicsmineR([MIT](https://cran.r-project.org/web/licenses/MIT) + file [LICENSE](https://cran.r-project.org/web/packages/heuristicsmineR/LICENSE)) - for process discovery as proposed by Weijters et al. (2011);
     + lubridate([MIT](https://cran.r-project.org/web/licenses/MIT) + file [LICENSE](https://cran.r-project.org/web/packages/tidyr/LICENSE)) - makes it easier to work with dates and times;
     + pm4py([GPL-3](https://cran.r-project.org/web/licenses/GPL-3)) - a process mining library in 'Python'
     + shiny([GPL-3](https://cran.r-project.org/web/licenses/GPL-3) | file [LICENSE](https://cran.r-project.org/web/packages/shiny/LICENSE)) - to build interactive web applications with R;
     + shinycssloaders([MIT](https://cran.r-project.org/web/licenses/MIT) + file [LICENSE](https://cran.r-project.org/web/packages/shinycssloaders/LICENSE)) - add a loading animation ("spinner") to outputs
2. **Mineração dos dados**
   * Associação dos atributos de mineração de processos aos atributos dos dados transformados (*event log object* com lifecycle\_id, activity\_instance, case\_id, activity\_id, activity\_instance\_id, lifecycle\_id, timestamp e resource\_id). Ver: <https://www.bupar.net/creating_eventlogs.html>.
   * Clusterização para agrupamento de órgãos julgadores com base no volume de processos por assunto com a utilização do algoritmo *k-means* com o uso do R e pacotes: klaR, xlsx, dplyr, factoextra, tidyr e reshape2. Foram identificados os seguintes grupos:
     + Varas Federais Comuns com JEF sem especialização
     + Varas Federais Comuns sem especialização
     + Varas Federais Criminais
     + Varas Federais Previdenciárias
     + Varas Federais Comuns Diferenciadas
     + Varas Federais de Execução Fiscal
     + Vara Federal Comum tipo A
     + Vara Federal Comum tipo B
     + Vara Federal Comum tipo C
   * Elaboração de filtros que permitem a redução da visão de modelos do geral para mais específicos: Grupos de órgãos julgadores, Assuntos, Classes e Órgãos julgadores.
   * Criação de modelo de análise descritiva de processos composto pelas seguintes informações:
     + XX Completar a partir da tabela de informações
3. **Avaliação**

Comparar resultados dos modelos, fazer comparações dos modelos e elaborar conclusões de boas práticas e oportunidades de melhoria.

**Implementação**

1. **Pré-processador e transformador de dados**
   1. Descompactar o arquivo disponibilizado pelo CNJ em <https://www.cnj.jus.br/owncloud/index.php/s/yIby5NidzxB1sQ8> *(senha: @cI4AGgsXwVa)*.
   2. Abrir e setar no script **Leitura\_Arquivos.R** o diretório base dos arquivos descompactados. O diretório deve estar setado na forma: *setwd('D:/Inova/Dados-CNJ\_disponibilizados/base')*. Importante ter atenção para a barra invertida.
   3. Selecionar o arquivo desejado XX
   4. Rodar o script com o atalho Ctrl + Shift + Enter.
   5. Será gerado o arquivo com os dados planificados na pasta 0.
2. **Clusterizador de órgãos julgadores**
   1. Abrir e setar no script **Cluster\_KMeans.R**  o diretório base dos arquivos descompactados. O diretório deve estar setado na forma: *setwd('D:/Inova/Dados-CNJ\_disponibilizados/base')*. Importante ter atenção para a barra invertida.
   2. Selecionar o arquivo desejado: no exemplo do script utilizamos o arquivo do TRF2.
   3. Rodar o script com o atalho Ctrl + Shift + Enter.
   4. Será gerado arquivo em formato .csv com os dados planificados na pasta 1. No exemplo o script gera o arquivo TRF2.csv.