

PROPOSTA DE MODELAGEM DE UM ALGORITMO PARA EXTRAÇÃO DE DADOS DA PLATAFORMA *EXPERIMENTAL DESIGN ASSISTANT*

Marcelo Maboni Hoppen, Thiago Gomes Heck, Rafael Z. Frantz e Sandro Sawicki

RESUMO

A redução do sofrimento de animais utilizados em pesquisa é um tema importante e necessário. É fundamental encontrar um equilíbrio entre o avanço científico e o respeito aos animais, buscando constantemente alternativas que minimizem o uso de animais e melhorem seu bem-estar. Felizmente, existem diretrizes e regulamentações rigorosas em muitos países que visam garantir o bem-estar dos animais utilizados em pesquisas. Essas regulamentações exigem que os pesquisadores considerem e apliquem métodos alternativos sempre que possível, reduzam o número de animais utilizados e implementem práticas que minimizem o sofrimento. No Brasil, o cuidado com experimentos envolvendo animais é regulamentado pela Lei Arouca (Lei nº 11.794/2008), que estabelece procedimentos para o uso científico de animais em pesquisa, ensino e testes de produtos. Além disso, o Conselho Nacional de Controle de Experimentação Animal (CONCEA) foi criado como um órgão responsável pela implementação e fiscalização dessas regulamentações. Essa lei estabelece que o uso de animais deve ser pautado pelo princípio dos "3Rs": Reduzir (redução do número de animais utilizados), Refinar (aperfeiçoar os procedimentos para minimizar o sofrimento) e Substituir (buscar alternativas que não envolvam o uso de animais). Além disso, cada instituição de pesquisa que utiliza animais em experimentos deve ter um Comitê de Ética no Uso de Animais (CEUA). Neste contexto, o *National Centre for the Replacement, Refinement & Reduction of Animals in Research* (NC3Rs), sediado no Reino Unido, desenvolveu uma ferramenta online para o projeto experimental com animais, chamada *Experimental Design Assistant* (EDA). Este artigo apresenta uma proposta para interpretar e transformar as informações de qualquer arquivo gerado pela ferramenta gráfica EDA com o objetivo de organizá-los e transformá-los no formato padrão CEUA/Unijuí.

Palavras-chave: JSON. *Experimental Design Assistant*. Extração de dados. Comitê de Ética no Uso de Animais.

ABSTRACT

Reducing the suffering of animals used in research is an important and necessary topic. It is essential to find a balance between scientific advancement and respect for animals, constantly seeking alternatives that minimize the use of animals and improve their well-being. Fortunately, there are strict rules and regulations in many countries to ensure the welfare of animals used in research. These regulations have encouraged researchers to consider and apply alternative methods whenever possible, reduce the number of animals used, and implement practices that minimize suffering. In Brazil, care for experiments involving animals is regulated by the Arouca Law (Law No. 11,794/2008), which establishes procedures for the scientific use of animals in research, teaching and product testing. In addition, the *Conselho Nacional de Controle e Experimentação Animal* (CONCEA) was created as an agency responsible for implementing and supervising these regulations. This law establishes that the use of animals must be guided by the principle of the "3Rs": Reduce (reduce the number of animals used), Refine (improve procedures to minimize suffering) and Replace

(search for alternatives that do not involve the use of animals). In addition, each Brazilian research institution that uses animals in experiments must have a *Comitê de Ética no Uso de Animais* (CEUA). In this context, the National Center for the Replacement, Refinement & Reduction of Animals in Research (NC3Rs), based in the United Kingdom, has developed an online tool for experimental design with animals, called Experimental Design Assistant (EDA). This article presents a proposal to interpret and transform the information of any file generated by the EDA graphic tool and organize and transform them in the CEUA/Unijuí standard format.

Keywords: JSON, Experimental Design Assistant. Data Extraction. Ethics Committee on the Animals Use.

INTRODUÇÃO

A utilização de animais em estudos pré-clínicos tem se consolidado a partir do século XIX, onde fisiologistas franceses como François Magendie (1783-1853) e Claude Bernard (1813-1878) dedicavam boa parte de seus estudos à viviseção de animais, visando melhorar o entendimento de órgãos e sistemas do corpo humano, desenvolvimento e aprimoramento de medicamentos e técnicas cirúrgicas. Mas em sua sede pelo conhecimento, não avaliavam o bem estar dos animais como uma questão relevante, viviseccionando animais sem se preocupar com a dor e o sofrimento dos mesmos.

Com o passar dos anos, o sofrimento dos animais passa a ter relevância, tanto que Russel et al. (1959) publicaram o livro *“The Principles of Humane Experimental Technique”*, o qual entre outros pontos importantes, visava melhorar as condições dos animais envolvidos em experimentos inserindo o conceito dos 3Rs, sendo eles:

(a) *Replacement*: métodos que evitam ou substituem o uso de animais em determinadas pesquisas, através da utilização de cultura de células e tecidos de vertebrados, formas iniciais de vida ou ainda por invertebrados;

(b) *Reduction*: métodos que diminuem o número de animais utilizados por experimento para um mínimo necessário para se obter resultados estatisticamente válidos, planejando cuidadosamente o experimento e escolhendo métodos estatísticos adequados, bem como extrair o máximo de informações possíveis de cada experimento;

(c) *Refinement*: métodos que melhoram o bem estar dos animais, assim como a redução da dor, sofrimento ou dano duradouro. Não só preocupa-se com o momento do experimento, mas sim com o período completo da vida dos animais envolvidos nos experimentos.

Para Graham (2015), o conceito dos 3Rs desde que foi proposto passou a ser amplamente aceito como uma estrutura ética robusta para reduzir o uso e o sofrimento de animais, refletindo as crescentes preocupações da sociedade em relação às pesquisas com animais, as quais têm recebido uma grande atenção nos últimos anos. Aplicando o conceito dos 3Rs é possível afirmar que haverá melhora nas condições de tratamento dos animais na pesquisa, mas além disso é necessário conduzir pesquisas com qualidade, segundo Freedman et al (2015) mais da metade das pesquisas pré-clínicas são irreprodutíveis. Já du Sert et al. (2017) destaca que falhas no projeto dos experimentos, análises estatísticas inapropriadas e relatórios inadequados como sendo os principais complicadores para a reprodutibilidade dos experimentos.

Visando melhorar as pesquisas com animais, em 2004 foi instituído no Reino Unido o UK National Centre for the Replacement, Refinement and Reduction of Animals in Research (NC3Rs), o qual entre outras atividades, desenvolveu a ferramenta *Experimental Design Assistant* (EDA) cujo objetivo é orientar pesquisadores em seus experimentos envolvendo animais, onde os experimentos são desenhados de forma gráfica com a utilização de figuras geométricas que são arrastadas e conectadas umas nas outras por meio de setas. Durante o desenho do experimento a ferramenta aponta possíveis falhas e permite calcular o número exato de animais necessários para o experimento, sugere melhorias e incrementa a confiabilidade e reprodutibilidade dos experimentos, pois possui uma base de conhecimento muito extensa e utiliza-se resultados de experimentos anteriores para ajudar na elaboração de novos projetos.

No Brasil, para que o experimento possa ser realizado é necessário uma autorização por parte do Comitê de Ética no Uso de Animais (CEUA), a qual ocorre por meio de análise de um formulário bastante extenso e detalhado que precisa ser preenchido pelo pesquisador. No entanto, muitos dos dados necessários para o preenchimento do formulário são também necessários durante o desenho do experimento na ferramenta EDA, mas não há até o momento uma forma automatizada de importar as informações da ferramenta EDA para o formulário. O objetivo deste trabalho é desenvolver um modelo computacional capaz de ler um arquivo no formato EDA e mapear as informações para um formulário padrão CEUA.

METODOLOGIA

Este estudo consiste em uma pesquisa exploratória com foco na extração de dados da ferramenta EDA. Nesse contexto, foram efetuadas pesquisas em diversos artigos relacionados à extração de informações de arquivos especialmente nos formatos XML (*Extensible Markup Language*) e JSON (*JavaScript Object Notation*), onde pôde ser notada uma grande variedade de técnicas de leitura e busca de padrões para classificação das estruturas internas dos arquivos.

Na ferramenta EDA estão disponíveis cinco exemplos distintos de experimentos, os quais servem como inspiração para que novos experimentos sejam elaborados, esses modelos de experimentos podem ser salvos em arquivos, os quais são baixados localmente para o computador do pesquisador. Os arquivos precisam ser abertos e seu conteúdo mapeado para realizar a leitura dos dados presentes no desenho do plano experimental, visando estudar sua estrutura interna e realizar um mapeamento dos construtores do formato EDA a fim de se obterem os campos de conteúdo informados na plataforma e abastecer um banco de dados local com as informações disponíveis.

Após compreender a forma de estruturação das informações internas em um arquivo EDA, será possível desenvolver um modelo computacional capaz de ler os dados de qualquer arquivo no padrão EDA para que seja possível estruturar um banco de dados para servir como depósito intermediário e também desenvolver uma ferramenta capaz de se conectar a este banco de dados. Com isso, busca-se automatizar o preenchimento de parte do “Formulário Unificado para Solicitação de Autorização para Uso de Animais em Pesquisa e Experimentação”, aproveitando assim todos os dados preenchidos na ferramenta EDA.

FERRAMENTA EXPERIMENTAL DESIGN ASSISTANT (EDA)

O NC3Rs em busca de prestar auxílio aos pesquisadores no bom planejamento de seus experimentos, bem como na aplicação dos conceitos dos 3Rs, desenvolveu uma ferramenta chamada *Experimental Design Assistant* (EDA), disponível em <https://eda.nc3rs.org.uk>, a qual pode ser utilizada por qualquer pessoa por se tratar de um produto para uso livre, é muito bem documentada, de fácil utilização e acessada de forma online.

A ferramenta EDA auxilia o trabalho do pesquisador durante o processo de planejamento e desenho do experimento, o qual é feito utilizando elementos gráficos em forma de figuras geométricas que são chamados de “nodes”, eles são arrastados para uma área de trabalho e aqueles *nodes* que possuem correlação podem ser ligados entre si por meio de setas. A Figura 1 demonstra o ambiente da ferramenta EDA, onde na esquerda está a relação de diferentes nodes e à direita encontra-se a área de trabalho onde esses nodes serão inseridos ao serem clicados e arrastados. Os nodes estão classificados em três grupos: “Experiments”, “Practical Steps” e “Analysis”

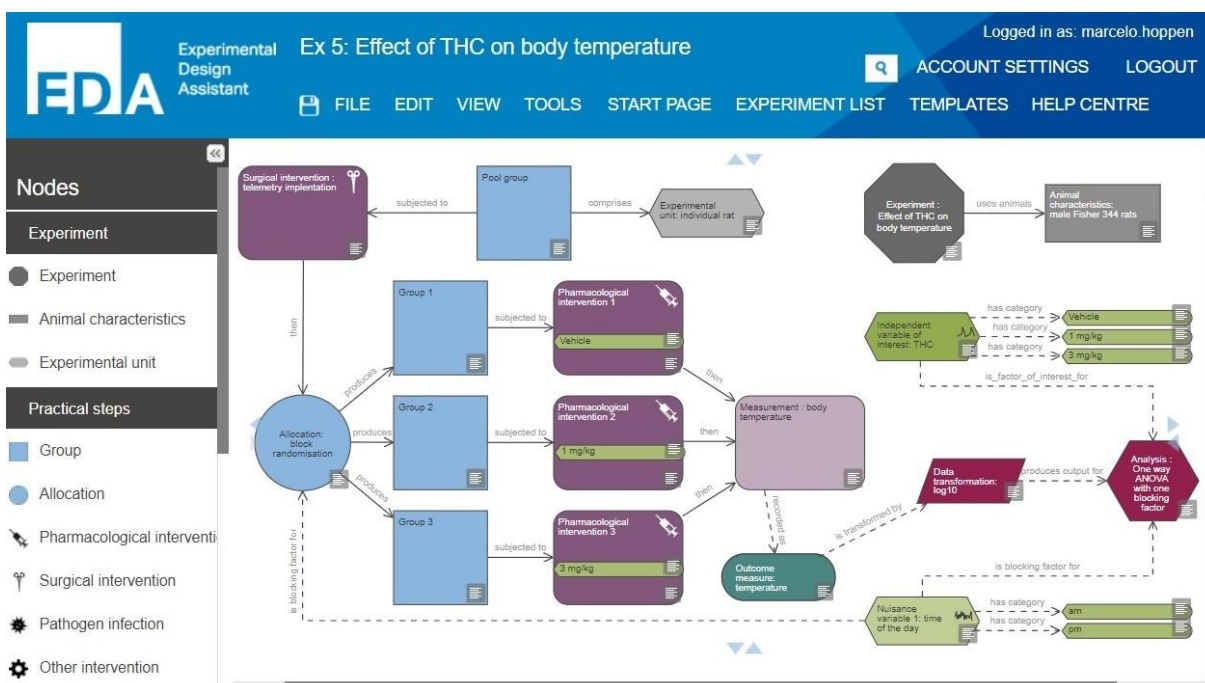


Figura 1: Ambiente da ferramenta EDA.

No grupo de nodes Experiment encontram-se os seguintes elementos:

Experiment: todos os diagramas precisam de um *node Experiment*, pois nele são armazenadas informações cruciais para que a ferramenta consiga fornecer um *feedback* do diagrama. Além de informações como nome e descrição do diagrama, nele são armazenadas informações referentes a: hipótese nula, hipótese alternativa, efeito de interesse, tamanho do efeito de interesse, entre outras.

Animal characteristics: é onde serão armazenadas informações sobre as características dos animais utilizados, então é necessário haver pelo menos um node desse

tipo conectado ao *node Experiment*, mas serão necessários múltiplos *nodes Animal characteristics* caso os animais possuam características diferentes, por exemplo: macho e fêmea e/ou diferentes genótipos.

Experimental unit: recebe as informações referentes à unidade experimental, que pode ser um animal individual, ou em certos casos podendo ser: uma gaiola de animais, uma ninhada, uma parte de um animal, um animal por um período de tempo, entre outras possibilidades. O *node Experimental unit* é conectado a um *node Group*.

No grupo de *nodes Practical steps* encontram-se os seguintes elementos:

Group: representa um grupo de animais e contém informações sobre o tamanho da amostra e se é grupo de tratamento ou grupo de controle. Ele precisa estar conectado a um *allocation*, *measurement* ou a algum *node intervention*.

Allocation: utilizado caso se tenha mais de um *group* de animais, pois descreve como os grupos são formados e contém informações da estratégia de randomização. Na prática é ele quem faz a divisão do grupo total de animais em outros grupos menores.

Pharmacological intervention: um *node intervention* representa um processo ao qual um grupo de animais será submetido, no caso o *Pharmacological intervention* representa a dose e modo de administração de drogas. Os *nodes intervention* de modo geral podem vir depois de um *Group*, *Measurement* ou outro *node Intervention*, a depender da sequência de eventos proposta.

Surgical intervention: destinado especificamente para processos cirúrgicos, onde podem ser informados: regime de analgesia e anestesia, duração da anestesia, possíveis efeitos adversos, entre outros.

Pathogen infection: é também um *node intervention* projetado especificamente para representar infecções, armazenando informações sobre o agente infeccioso.

Other intervention: destinado para qualquer outro tipo de intervenção que não se encaixe nas três anteriores, como por exemplo exposição a luz, condicionamento, estímulo ambiental em um ensaio comportamental, entre outros.

Euthanasia: também do tipo *intervention*, mas destinado a representar o abate dos animais, onde serão informados dados sobre o momento da intervenção, método utilizado e tecidos coletados na autópsia.

Measurement: é um dos tipos de *nodes* utilizados para representar a coleta de dados e precisa ser utilizado em conjunto com pelo menos um *node Outcome measure*. O *Measurement* deve ser utilizado após um *Group*, um *Intervention* ou outro *Measurement*.

Repeated measurement: utilizado para representar uma coleta de dados repetida muitas vezes consecutivamente. Assim como no caso do *Measurement*, precisa ser utilizado em conjunto com pelo menos um *node Outcome measure*.

No grupo de *nodes* "**Analysis**" encontram-se os seguintes elementos:

Independent variable of interest: representa um parâmetro manipulado especificamente para testar a hipótese. É um fator que um pesquisador manipula dentro de um ambiente controlado para testar o impacto que a mudança dos níveis da variável independente tem no resultado medido.

Nuisance variable: representa uma variável independente que apesar de não haver um interesse particular nela mesma, precisa ser controlada na análise estatística para não ocultar o efeito de uma variável de interesse.

Variable category: um *node category* pode ser ligado a um *node* de variável *Independent* ou *Nuisance* caso ela seja categórica, pode ser conectado a um *node Intervention* quando se deseja nomear as diferentes intervenções com cada categoria possível, também pode ser conectado a um *node Group* quando por exemplo existem dois fatores de interesse combinados e se deseja deixar marcado em cada grupo identificando a sua composição, finalmente também pode ser conectado a um *node Measurement* quando a variável independente está relacionada a uma medição em diferentes períodos de tempo, sendo que cada ponto de tempo relevante é representado como um *node Category*.

Outcome measure: conhecido também como variável dependente ou variável resposta, pode ser conectado dentro ou após um *node Measurement*.

Data transformation: usado para indicar uma transformação nos dados antes de serem analisados, são conectados após um *node Outcome measure* de forma individual (dois *nodes Outcome measure* não podem estar ligados no mesmo *Data transformation*, mesmo que a transformação siga o mesmo padrão).

Analysis: contém detalhes de como os dados são analisados, precisa estar conectado recebendo dados de pelo menos um *Outcome measure* e pelo menos um *Independent variable of interest*.



LEITURA DE ARQUIVO JSON

O plano experimental exportado pela ferramenta EDA possui sua organização interna no formato JSON, o qual pode ser legível por humanos com uma certa facilidade, pois é basicamente composto por elementos formados por pares de atributo e valor. Sbai et al. (2019) classifica JSON como um formato de intercâmbio de dados leve que foi primeiramente especificado e popularizado por Douglas Crockford. É baseado em um subconjunto da linguagem de programação JavaScript e por sua simplicidade bem como a habilidade de ser facilmente processado tanto por humanos quanto por máquinas tornou-se amplamente utilizado.

JSON define um pequeno conjunto de regras de formatação para representar dados, é composto por quatro tipos primitivos de dados (*number*, *string*, *boolean* e *null*) e também por dois tipos estruturados (*object* e *array*). O tipo estruturado *object* é uma coleção não ordenada formada de zero ou mais pares de atributo/valor, onde atributo é uma *string* e o valor pode ser de qualquer um dos tipos primitivos ou estruturados citados anteriormente. Já o tipo estruturado *array* é uma sequência ordenada de zero ou mais valores. Por questões práticas para leitura de um arquivo no formato JSON, iremos classificar os valores dos atributos conforme três possibilidades que nomeamos da seguinte forma:

SimpleJSONValue: corresponde a valores de um atributo que sejam de um dos tipos primitivos;

JSONObject: corresponde a valores em formato *object*, ou seja, quando o valor de um atributo é formado por uma lista não ordenada de diversos outros pares de atributo/valor, descreve muito bem uma situação onde há elementos dentro de outro elemento de forma aninhada.

JSONArray: corresponde a valores em formato *array*, ou seja, quando o valor de um atributo é formado por uma lista ordenada de diversos outros pares de atributo/valor, descreve muito bem o relacionamento de “um para muitos”.

O processo de leitura de um arquivo no formato JSON pode ser realizado conforme visto na Figura 2 onde é demonstrado um fluxograma com as etapas a serem percorridas pela rotina de leitura, a qual inicia com a abertura do arquivo e a leitura do elemento raiz, seguida de uma função que implementa um laço de repetição construído para percorrer todos os

atributos que estejam aninhados dentro do elemento raiz, no momento que chegar ao fim do arquivo (quando terminar de percorrer todos os atributos presentes no elemento raiz) a leitura estará finalizada.

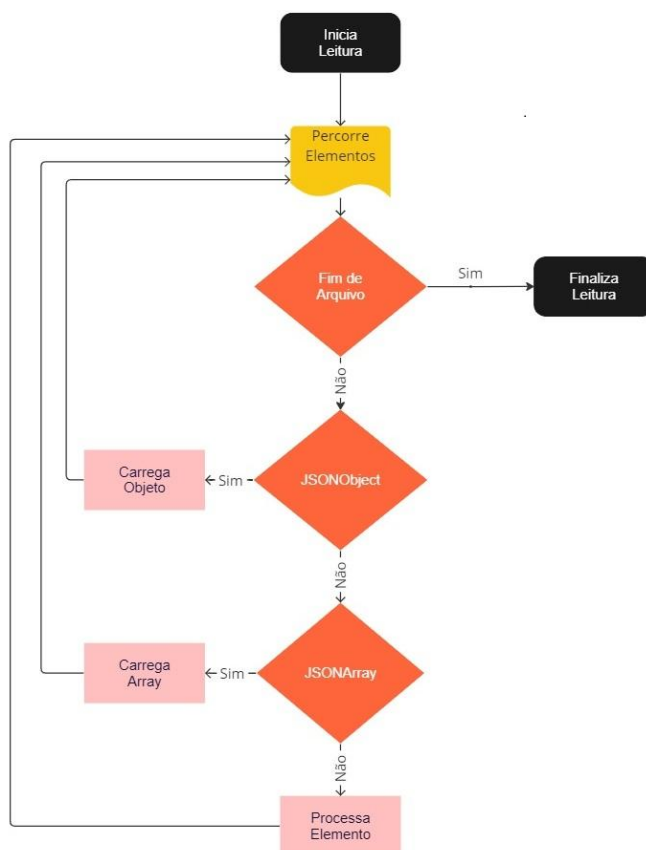


Figura 2: Fluxograma da lógica de leitura proposta para um arquivo JSON.

Durante a leitura, enquanto não for atingido o final do arquivo, é necessário realizar testes para definir o tipo de valor de cada atributo, o primeiro teste verifica se o valor do atributo é do tipo JSONObject, o qual na prática é composto por uma nova estrutura no formato JSON e precisa de um novo processo de leitura pois é necessário que todos os seus atributos também sejam percorridos, o segundo teste visa identificar se o valor do atributo é do tipo JSONArray. Nesse caso também é necessário realizar a leitura de todos os atributos da *array*, sendo que para ambos os casos o ato de retornar à função “Percorre Elementos” é uma chamada recursiva. Caso o valor do atributo não seja de nenhum dos dois tipos anteriores, significa que o valor dele é de um tipo primitivo, então é efetuada a leitura desse valor na etapa “Processa Elemento”.

ORGANIZAÇÃO DO ARQUIVO COM FORMATO EDA

Os arquivos oriundos da plataforma EDA apresentam uma estrutura de organização das informações que permite construir um processo específico de leitura dos dados. É necessário identificar como estão dispostas as informações gerais do plano experimental e também as informações de cada objeto que compõem o desenho do experimento. Percebe-se que cada um desses objetos é chamado de *node*, sabendo que também há a possibilidade de certos objetos conterem em seu interior outros objetos, o que na prática acontece quando há um *node* dentro de outro *node*.

A área de trabalho da plataforma EDA, bem como cada *node* utilizado para desenhar o experimento e também cada um dos objetos de ligação (setas) que são utilizados para interligar os *nodes*, dão origem a um conjunto de elementos (pares de atributo/valor) dentro do arquivo JSON.

A área de trabalho da plataforma EDA, onde o pesquisador que está projetando o experimento arrasta e solta os *nodes* correspondentes aos objetos que ele pretende adicionar ao experimento, é responsável por definir alguns dados gerais referentes ao desenho do experimento e principalmente servir de objeto raiz (servir de base) para todos os objetos adicionados diretamente à área de trabalho da ferramenta EDA.

Um dos atributos presentes na raiz de um arquivo EDA é o "*childShapes*", que pode ser visualizado na Figura 3, o qual contém um *array* de elementos que armazena todos os objetos adicionados diretamente na área de trabalho da ferramenta EDA, sendo que para cada um dos itens desse *array* temos atributos importantes, os quais podemos destacar:

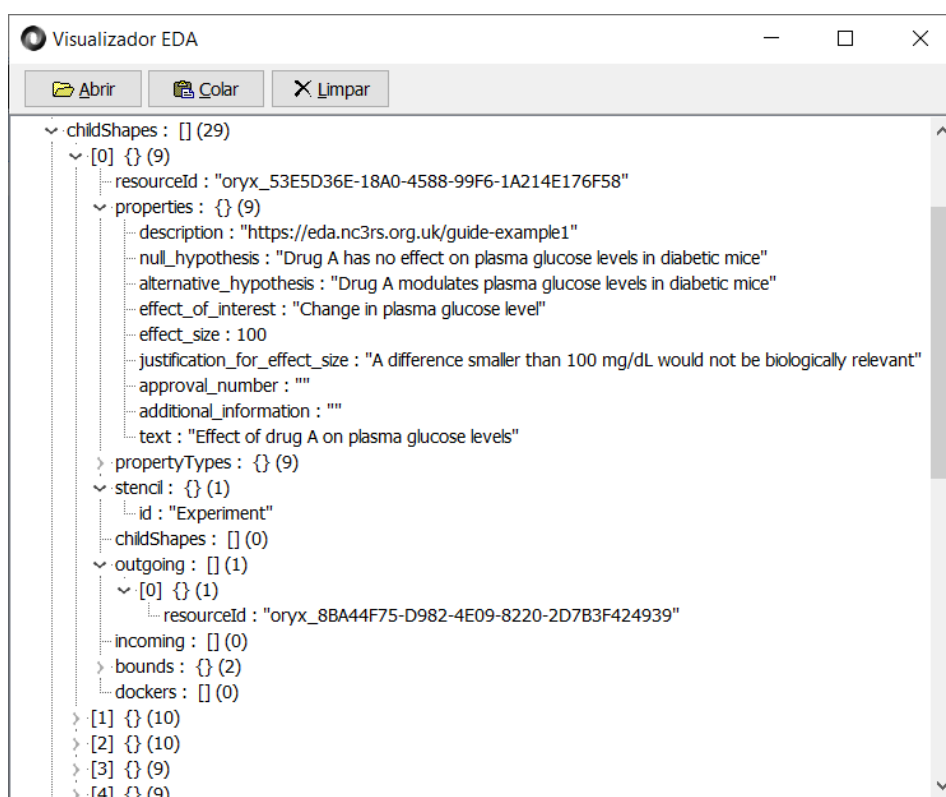


Figura 3: Conteúdo parcial de um arquivo no formato EDA.

resourceId: possui como valor uma chave única para identificação do objeto;

properties: do tipo `JSONObject`, possuindo uma série de elementos atributo/valor, que são responsáveis por guardar informações dos atributos do objeto em questão;

stencil: um importante elemento que descreve o nome do tipo de objeto que está armazenado neste índice da `array`;

childShapes: elemento do tipo `array`, armazena informações de objetos que estejam aninhados dentro do objeto em questão;

outgoing: armazena o id de um objeto para o qual se está conectando;

incoming: armazena o id de um objeto de onde está vindo uma conexão;

Qualquer um dos 18 tipos diferentes de `nodes` irá apresentar essa mesma estrutura de atributos, com exceção do conteúdo de `properties`, o qual é composto por atributos condizentes ao tipo de `node` em questão. Além dos 18 tipos de `nodes` conhecidos, há também uma série de objetos que representam as conexões entre os `nodes`, sendo que o elemento gráfico utilizado na ferramenta EDA para esse objeto de ligação é uma seta. Há diversos tipos

de objetos de ligação e no momento do desenho do plano experimental, ao serem conectados 2 *nodes* diferentes, serão apresentados os tipos de setas possíveis para tal combinação. Internamente no arquivo JSON, o que diferencia um tipo de seta de outro tipo de seta é apenas o valor de *stencil.id*, sendo esse um dado muito importante pois contextualiza a conexão entre os *nodes*, os valores de *outgoing* e *incoming* também são muito importantes pois serão informados respectivamente para qual *node* a seta aponta e de qual *node* ela recebe ligação.

LEITURA DE ARQUIVO NO FORMATO EDA

Para proceder a leitura e extração de dados dos arquivos no formato EDA, foram implementadas diversas rotinas de programação, iniciando pela rotina responsável pela descompactação do arquivo, carregamento do objeto raiz referente à área de trabalho do plano experimental. Na sequência ela carrega o conteúdo do elemento “*childShapes*” o qual contém todos os objetos adicionados no plano experimental exportado pela ferramenta EDA. Então utiliza um laço de repetição para percorrer todos os itens da *array* “*childShapes*” e para cada um deles, crie uma instância , passando como parâmetro o item do *array* atual.

Outra rotina de processamento identifica o valor do atributo *stencil.id* do elemento em questão, o qual classifica como sendo de um dos 18 tipos de *nodes*, ou se o item é um objeto de ligação entre *nodes*. Caso seja um *node*, então pelo seu nome (*stencil.id*) é possível identificar quais atributos ele contém e realiza a leitura específica de cada um deles, haja visto que há uma grande diversidade de atributos para cada um dos diferentes tipos de *node*. Caso seja um objeto de ligação, apenas três atributos são importantes: o nome que classifica o tipo de ligação, *incoming* e *outgoing* que representam respectivamente de onde a ligação chega e para onde a ligação vai.

Caso o elemento analisado seja classificado como um dos 18 diferentes tipos de *nodes*, ele pode em seu conteúdo ter um elemento “*childShapes*”, o qual está presente para contemplar a necessidade de haver objetos aninhados dentro de objetos. Para esse caso, a rotina de processamento deve implementar um laço de repetição que irá percorrer todos os itens da *array* “*childShapes*” e de forma recursiva, instanciar uma nova “*ProcessaElemento*” passando como parâmetro o item a ser analisado até que todo o conteúdo do arquivo JSON seja concluído.

CONSIDERAÇÕES FINAIS

A grande preocupação da sociedade perante o uso de animais em ensino e pesquisa, especialmente expressa na concepção e aplicação dos conceitos dos 3Rs, torna necessário que cada vez mais se tenham projetos de pesquisa bem elaborados, que sejam reprodutíveis e alcancem um bom nível de confiabilidade, tornando assim a utilização da ferramenta EDA indispensável.

Este estudo permite a automatização do processo de preenchimento dos campos do formulário CEUA/Unijuí. Com isso, foi possível identificar a estrutura de organização dos elementos presentes nos arquivos EDA e como eles se conectam uns aos outros.

Através da implementação das rotinas apresentadas, foi possível efetuar a leitura de qualquer arquivo EDA, podendo extrair todos os dados informados no plano experimental projetado na ferramenta e compondo um banco de dados com informações organizadas e prontas para automatização do preenchimento do formulário CEUA/Unijuí.

REFERÊNCIAS BIBLIOGRÁFICAS

du Sert, N. P. et al. **The experimental design assistant**. PLoS biology, v. 15, p. 1-9, 2017.

Freedman, L. P. et al. **The economics of reproducibility in preclinical research**. PLoS Biology, v. 13, p. 1-9, 2015.

Graham, M. L.; Prescott, M. J. **The multifactorial rol of the 3rs in shifting the harm-benefit analysis in animal models of disease**. European Journal of Pharmacology, v. 759, p. 19-29, 2015.

Russel, M. S. W.; Burch L. R. **Principles of human experimental technique**. Potters Bar, England, 1959.

Sbai, S. et al. **Jsonoonto: Building owl2 ontologies from json documents**. International Journal of Advanced Computer Science and Applications, v. 10, p. 2013-2018, 2019.