Universidade de Aveiro

# IEETA Bioinformatics - BIcenter

---


*Autores:*
89348 Diogo Silva
88931 Vasco Ramos

*Orientadores:*
José Luís Oliveira
João Almeida


30 de janeiro de 2021

# Conteúdo

# 1 Introdução

O BIcenter foi inicialmente desenvolvido como um projeto de tese do mestrado em Engenharia de Computadores e Telemática intitulado Plataforma Web para Gestão de Processos ETL em ambientes Multi Instituição desenvolvido pelo Leonardo Coelho sob a orientação do Professor José Luís Oliveira. Este projeto revolvia em volta da criação de uma web-based tool que permitisse a definição e gestão de pipelines de processos ETL (Extract-Transform-Load) em contextos multi-instituição, permitindo assim que pessoas menos versadas na programação deste tipos de processos pudessem definir pipelines de processamento que trouxessem benefícios às empresas que os utilizassem.

No seguimento da nossa bolsa, e tendo em conta que este projeto estava já em standby há alguns anos, o nosso trabalho consistiu em:

- **Revitalizar** a plataforma, atualizando quaisquer dependencias necessarias até conseguirmos que o sistema estivesse operacional.
- **Corrigr Bugs** que fôssemos encontrando durante o processo de revitalização e desenvolvimento.
- **Implementar novas features** tais como novos componentes e alteração de certos aspetos do UI.
- **Documentar** todo o trabalho feito, bem como todos os passos necessários para conseguir correr e continuar a desenvolver o BIcenter

# 2 Abordagem Inicial

Como ponto de partida para a revitalização do projeto, fizemos o trabalho de executar o projeto, bem como os vários problemas associados a esta tarefa, tais como:

- Dependências desatualizadas.
- Incompatiblidades nas versões do SBT.
- Incompatibilidades nas versões de Bases de Dados.
- Sistema com problemas funcionais nos processos de autenticação e autorização com repositório local.
- Sistema um pouco confuso de interagir.

Após conseguirmos ter o sistema funcional, procedemos ao teste de todas as suas funcionalidades para conseguirmos ter uma noção do que estava a funcionar corretamente e do que não estava. Os problemas encontrados foram listados como *issues* no repositório do projeto.

# 3 Bug Fixes

Enquanto que, no geral, a plataforma não apresentava muitos bugs complexos, houve alguns que nos chamaram à atenção e que foram, no decorrer do nosso trabalho, tratados e resolvidos.

## 3.1 Checkboxes

Havia uma pequeno erro na maneira como as *checkboxes* estavam a ser renderizadas quando tinham sido clicadas pelo utilizador (ou seja, após serem selecionadas). Se o utilizador clicasse numa checkbox, guardasse essa alteração, e voltasse a abrir, a checkbox deixaria de ser mostrada.

Este erro sucedia pela maneira como os componentes estavam a criar checkboxes (visto todo este processo estar implementado de forma dinámica). Apenas checkboxes com valor *false* tinham sido especificadas nos templates e, portanto, quando estas tinham valor *true* (após ser clicada), aquando da re-renderização dos parâmetros do componente, esta não era mostrada.

A correção, em si, foi também bastante simples e bastou adicionarmos uma especificação para a criação de checkboxes com valores tanto *false* como *true*.

## 3.2 Autenticação e Permissões, utilizando repositório local

O BIcenter está desenvolvido para funcionar com repositórios de autenticação, como o **LDAP**, mas também com repositórios locais. Contudo, quando se utilizava repositórios locais para autenticação, o sistema falhava em associar os utilizadores às suas respetivas instituições.

A alteração necessária para solucionar este problema foi alterar os processos de inicialização de utilizadores e acessos para proceder às respetivas associações.

# 4 Novas Funcionalidades

## 4.1 Execução / Infraestrutura

Tal como referido anteriormente, alguns dos problemas envolvidos na tarefa inicial de execução do sistema estavam relacionados com a infraestrutra associada (**SBT**, **bases de dados**, etc). Assim, era essencial criar meios mais simples e imediatos para executar o sistema no seu todo, o que nos levou ao desenvolvimento de **Docker** *containers* para os ambientes de desenvolvimento e produção.

## 4.2 Componentes

Toda a plataforma do BICenter serve como um *pseudo-wrapper* do Pentaho Data Integration (PDI). Como tal, para além dos componentes e processos de ETL já presentes na plataforma, é nos possível implementar novos a partir da vasta lista de componentos do PDI. Com isto em conta, foram portanto, durante o decurso da bolsa, adicionados dois novos elementos - **CSVFileInput** e **UsagiMapper**.

### 4.2.1 CSV File Input

O BICenter tem a capacidade de receber dados de uma variedade de sistemas de bases de dados através do componente **Table Input**. São depois estes dados, que após serem carregado, podem ser processados e visualizados. Dada a importancia desta taréfa, foi declarado que seria furtuito oferecer aos utilizadores do BICenter a capacidade de conseguirem carregar dados vindos de outras fontes, tais como os comunamente utilizados em meios empresariais, ficheiros **CSV**.

Felizmente, o PDI fornece já um componente capaz de abrir ficheiros CSV e retirar de la dados e informações. Infelizmente, porém, o BICenter não possuia nenhuma capacidade relacionada com o carregamento, upload e processamento de ficheiros vindos do utilizador. A maior dificuldade encontrada com a implementação deste componente foi portanto extender o sistema por parte do frontend para que este fosse capaz de receber um ficheiro e empacota-lo no **JSON** que é utilizado para enviar os parametros e especificações dos componentes do frontend para o backend. Fomos encontrando alguns pontos de paragem nesta implementação, maioritariamente graças a medidades de segurança implementadas pelo AJAX no que toca ao envio e storaging de ficheiros vindos dos utilizadores.

Mesmo tendo em conta estas dificuldades, porém, acabamos por conseguir realizar o envio dos ficheiros. De seguida extendemos a AbstractStep.java class, que realiza o preprocessamento dos componentes antes de serem chamados os métodos da biblioteca do PDI, de forma a que pudessemos ler o ficheiro, carregar o *header*

do CSV (caso este exista) e efetivamente ler os dados.

No final ficamos com o componente **CSV Input**, presente na subsecção de **Input**, que permite ao utilizador carregar um ficheiro de CSV e ter os seus conteudos lidos e interpretados nas restantes fazes da pipeline de ETL.

### 4.2.2 USAGI Mapper

O segundo componente que foi adicionado foi o do **Usagi Mapper**. O **USAGI** é uma ferramenta da **OHDSI** que permite realizar o mapeamento de valores e conceitos de forma a conseguir "standardizar"um conjunto de bases de dados que possam referir os mesmos conceitos mas com numenclaturas diferentes. Este programa é capaz de gerar um ficheiro CSV, chamado *UsagiExport*, que especifica, entre outros campos, o valor original do conceito, e o código ou nome para o qual ele deve ser mapeado.

Foi nos então pedido que adicionassemos um componente ao BICenter capaz de ler ficheiros UsagiExport e realizar o mapeamento dos dados como especificado. Esta adição teve dois problemas. Um, era o facto de que tinhamos que de novo conseguir ler ficheiros CSV especificados pelo utilizador, e o outro é o facto que o PDI não possui nenhum componente que interprete ficheiros UsagiExport. O primeiro problema foi resolvido com a criação do componente **CSVInput**, visto que este também necessitava duma framework de leitura de ficheiros, e visto ter este componente sido criado primeiro, não tivemos que voltar a lidar com ele. O segundo foi relativamente mais exoterico.

Já foi mencionado que o BICenter é, no seu core, um wrapper de componentes do PDI. Isto limita-nos a que toda a nossa logica de baixo nivel, e adição de componentes esteja dependente da existencia destes na biblioteca do PDI. Enquanto que o PDI não possui diretamente nenhum componente capaz de interpretar ficheiros UsagiExport, após alguma pesquisa, descobrimos que possui, porém, um componente chamado **ValueMapper**.

Este componente lê dados inseridos num passo anterior da pipeline, e pede ao utilizador para especificar, usando uma tabela, os conceitos base e os nomes para os quais esses conceitos são mapeados. Através deste componente, conseguimos criar o **USAGI Mapper**, simplesmente pedindo ao utilizador para carregar um ficheiro UsagiExport, e, com processamento a priori, ler esse ficheiro, carregar os conceitos e mapeamentos para um dicionario, e de seguida fornecer esse dicionario ao ValueMapper. Do ponto de vista do utilizador a unica coisa que ele efetivamente tem que fazer é carregar o ficheiro.

## 4.3 Interface

Um dos outros problemas tem essencialmente a ver com a primeira página à qual se tem acesso, após proceder ao *login* no **BIcenter**. O sistema manda-nos diretamente para o editor de pipelines de ETL, mesmo quando ainda não está selecionada nenhuma *task*.

Tendo em conta isto, e para facilitar a utilização do sistema, decidimos:

- Criar uma nova homepage/dashboard, onde atualmente são apresentadas todas as instituições a que um utilizador tem acesso, bem como as funcionalidades de adicionar, editar e remover entidades (sejam *Instituições*, *Tasks*, *Data Sources*, etc), tendo em conta os seus previlégios no sistema.
- Alterámos a *sidebar* existente para apenas permitar manipular os componentes de *tasks*, deixando a administração de entidades apenas na *homepage*.

### 4.3.1 Homepage / Dashboard

A nova homepage serve para listar as instituições existentes a que um dado utilizador tem acesso, bem como proceder à adminstração dessas instituições e dos recursos a estas associado (*tasks*. *data sources* e *execution servers*).

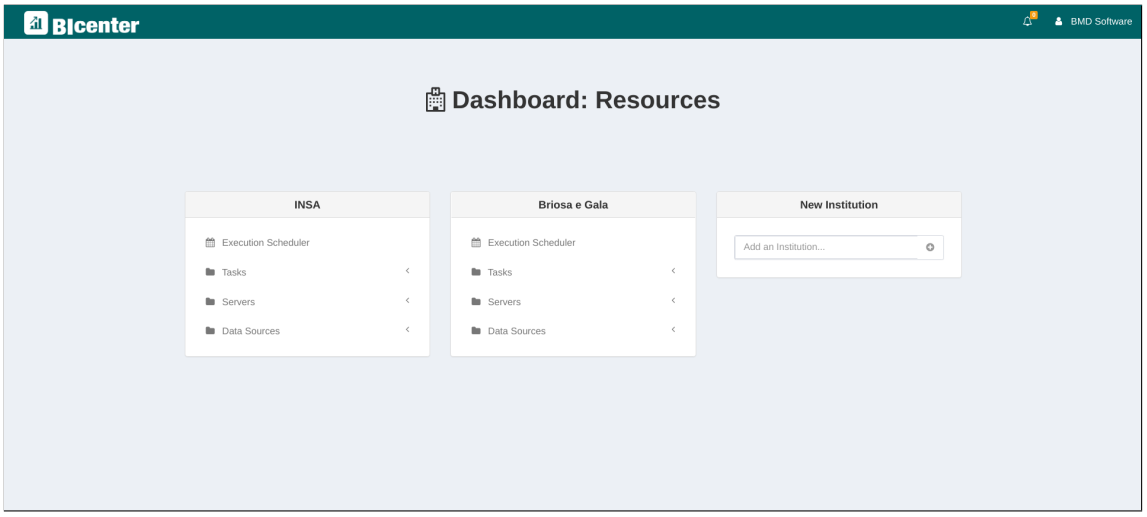


Figura 1: Homepage atual

### 4.3.2 Sidebar

Para além da criação da homepage, a sidebar do BIcenter foi também atualizada. Este antes apresentava dois submenus: um com as funcionalidades agora

presentes na *homepage* e outra para lidar com os componentes de ETL para usar nas *tasks*.

A nova *sidebar* contém apenas um menu (o último referido), para os utilizadores poderem criar, editar e manipular as suas ETL tasks, como lhes convém. Foi também adicionado um shortcut visual para voltar para a *homepage*.

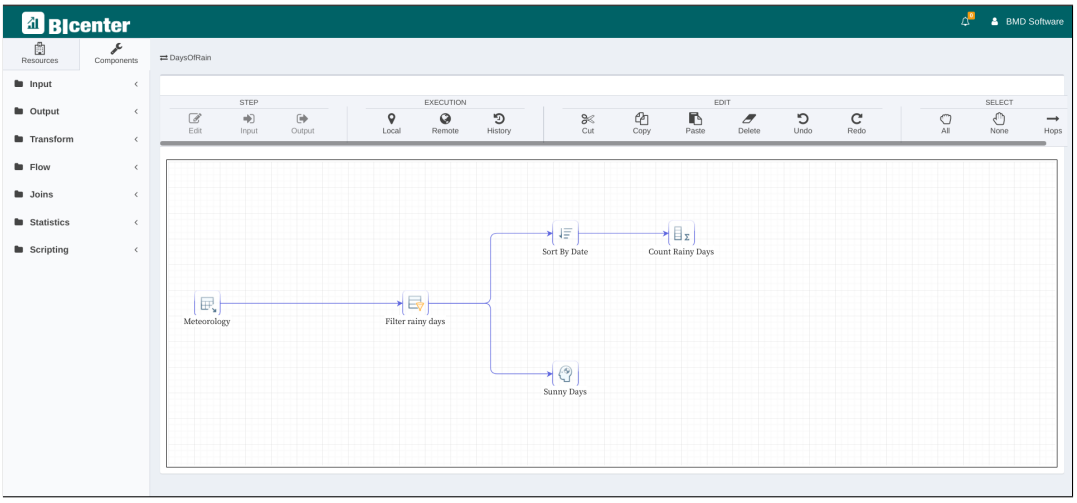


Figura 2: Sidebar - antes

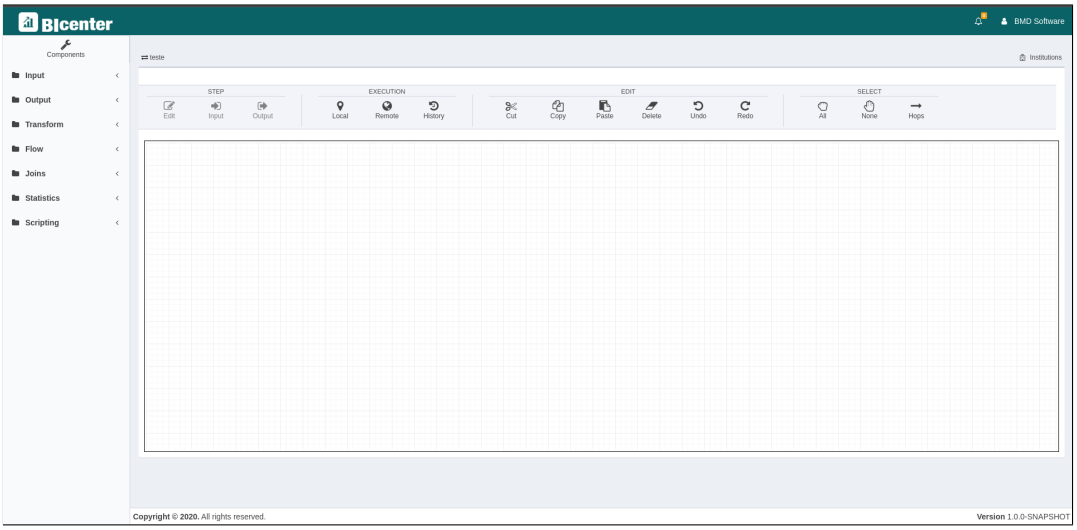


Figura 3: Sidebar - depois

# 5 Conclusão

Concluimos então este relatorio dizendo que, através deste projeto de bolsa, foi-nos possível aprofundar os nossos conhecimentos sobre ETL e pipelines de processamento de dados, tendo ganho uma nova perspetiva do valor que este tipo de ferramentas podem trazer para uma empresa.

Fomos capazes de revitalizar a ferramenta do BIcenter, bem como adicionar novas funcionalidades, que trazem ainda mais valor à ferramenta.

Estamos, também, contentes com as atualizações que realizámos à interface da plataforma, sem esquecer toda a documentação que foi produzida de forma a ajudar qualquer pessoa que venha, no futuro, a trabalhar no BIcenter, e que nos teria sido bastante util aquando inicializamos o nosso proprio trabalho.