

# Extrator Genérico de Dados

Fernando Henrique Canto, José Luis Machado, Ricardo Vieira

Centro de Processamento de Dados  
Universidade Federal do Rio Grande do Sul  
{fernando, zeluis, vieira}@cpd.ufrgs.br

**Resumo.** *Uma grande quantidade de sistemas desenvolvidos pelo CPD gera a necessidade de relatórios gerenciais, que serão acessados apenas pelos gerentes do sistemas. Devido à criticidade dessas demandas e pela semelhança entre os diferentes relatórios, desenvolvemos uma ferramenta de apoio ao desenvolvedor, que permite a geração automática de relatórios a partir de Extrações de Dados cadastradas no banco de dados. As tabelas, juntamente com uma biblioteca de classes PHP, fornece um mecanismo flexível e poderoso de criação de relatórios com uma interface para o usuário, utilizando HTML e CSS, e consultas SQL.*

## Introdução

Neste trabalho, apresentamos uma biblioteca desenvolvida na linguagem PHP para realizar, de forma simples e genérica, extrações em uma base de dados, a partir de uma consulta qualquer, devolvendo os dados em um determinado formato de arquivo.

A ideia surgiu quando se percebeu que esse tipo de operação ocorre em diversos sistemas, com mínimas variações. Para evitar retrabalhos desnecessários, pensou-se em uma forma de transformar a criação de uma nova extração em uma tarefa simples e rápida.

## O que é uma Extração?

Entendemos como “Extração” uma consulta ao banco de dados que retorna dados formatados em um arquivo. É importante ressaltar que isso não é semelhante a gerar *relatórios*: uma extração não se preocupa com formatação visual e layout, disposição dos dados ou coisas do gênero. Geralmente, extrações são usadas para se fazer importações. Por exemplo, um usuário pode desejar criar uma base de dados Access em sua máquina, e populá-la com dados retirados da base institucional. O melhor caminho seria exportar os dados para um arquivo do tipo CSV, e importar esse arquivo para dentro do Access.

Uma extração serve para fazer exatamente isso: exportar dados. A única preocupação com formatação é referente ao tipo de arquivo para o qual os dados serão exportados; alguns exemplos são CSV (Comma Separated Values, que na prática é um arquivo texto com valores separados por vírgulas – ou outro caractere qualquer – e, opcionalmente, delimitados por aspas, uma linha para cada registro), planilha Excel (que pode ser gerada através de uma tabela HTML) e XML. O núcleo de uma extração, porém, são os dados em si: quais registros e quais colunas de quais tabelas serão exportadas.

## Como Definir uma Extração

Uma Extração de dados, basicamente, é representada através de uma consulta SQL. A consulta pode ser a mais complexa que se desejar, e é ela que vai definir quais os registros e quais as colunas serão exportadas.

Para se criar uma extração, deve-se criar um registro em uma tabela de Extrações (tab. 1). A extração terá um número identificador único, uma consulta SQL associada, bem como diversas outras opções de segurança e controle de acesso. É importante deixar claro que esse registro define praticamente todo o funcionamento daquela extração, o que significa que não é necessário alterar o código-fonte das aplicações quando a extração precisar ser alterada (e.g. adicionar ou remover colunas, modificar os filtros, etc.).

O identificador numérico da extração é então usado para instanciar essa extração dentro das aplicações. A biblioteca inclui uma classe, naturalmente chamada *Extracao*, que faz as duas principais operações de maneira automática: exibe a interface com o usuário, e efetua a busca e exportação dos dados a partir das escolhas do usuário.

Nome da coluna	Tipo	Descrição
CodExtracao	Número (PK)	Identificador único, utilizado para instanciar a extração na aplicação;
DenominacaoExtracao	Texto	Denominação textual, exibida para o usuário;
SQLExtracao	Texto	Consulta SQL que retornará os dados extraídos;
TipoArquivo	Texto	Tipos de arquivo que podem ser gerados pela extração;
DiretorioOrigemExtracao	Texto	Diretório a partir do qual a extração pode ser utilizada pelos scripts PHP;
CodPessoaAnalista	Número (FK)	Código do analista responsável pela extração;
UsuarioUltimaAtu	Número (FK)	Código da pessoa que fez a última alteração ao registro;
DataUltimaAtu	Data	Data da última alteração;
CodSemanticoExtracao	Texto	Identificador textual, usado na aplicação;

Tabela 1: Estrutura da tabela EXTRACAO

### Opções da Extração

Um dos grandes objetivos que tínhamos ao criar essa biblioteca era de que as extrações pudessem ser alteradas e corrigidas sem que fosse necessário alterar as aplicações. Portanto, era necessário que a tabela de extrações contivesse o máximo de atributos e opções possíveis, e que a classe PHP apenas executasse as operações de acordo com essas opções. Decidiu-se, então, que todo o funcionamento da extração seria retirado da consulta SQL, e uma das partes mais sofisticadas dessa tarefa era reconhecer as colunas da extração.

O usuário teria a possibilidade de selecionar apenas as colunas que ele quisesse recuperar. Para tanto, ele seria apresentado com uma lista das colunas, cada uma com um *checkbox* associado. Com isso, bastaria marcar e desmarcar as colunas correspondentes.

A classe de Extração faz isso lendo a consulta SQL, utilizando uma expressão regular para reconhecer as colunas a partir da cláusula SELECT. A única imposição é que cada coluna deve ter um *alias* associado (por exemplo, *select DataNascimento as 'Data de Nascimento'*); isso faz com que as colunas tenham um nome *apresentável* para o usuário, pois o *alias* é usado como nome de exibição.

Uma outra opção possível de ser colocada na tabela se refere aos tipos de arquivo que podem ser exportados. Por exemplo, uma determinada extração pode prever arquivos CSV, XML e planilha Excel, enquanto outra permite apenas CSV. Esse tipo de opção *também* pode ser definida no código fonte, tornando-se flexível para cada sistema.

Após os testes iniciais, os analistas sentiram a necessidade de definir atributos para cada coluna. Um desses atributos é a obrigatoriedade de seleção; isso significa que uma dada coluna deve *obrigatoriamente* fazer parte da extração, aparecendo na lista de colunas sem poder ser desmarcada. Outro atributo é a possibilidade de filtragem, isto é, permitir que o usuário faça um refinamento da extração a partir de um determinado critério. Para resolver esse problema, decidiu-se usar o próprio *alias* das colunas, determinando *prefixos*. Os prefixos implementados no momento são:

- OB: coluna obrigatória. Ela não poderá ser desmarcada na lista de colunas;
- TX: filtro *textual*. O usuário poderá preencher um valor textual para refinar a consulta;

- NU: filtro *numérico*. O usuário poderá preencher um valor numérico para refinar a consulta;
- DT: filtro *temporal*: O usuário poderá informar um período para refinar um campo do tipo *data*;
- GR: filtro por *grupo*: O usuário poderá escolher o valor do critério de refinamento a partir de todos os valores existentes para aquela coluna; por exemplo, em um campo *Sexo*, os valores possíveis seriam “F” e “M”.

Esses prefixos podem ser combinados, embora a única combinação válida entre os prefixos acima seja do “OB” com um prefixo de filtro. Para tanto, pode-se fazer uma consulta deste tipo:

```
select Cod_Usuario as 'OB_Código do Usuário',
       Nome_Pessoa as 'Nome da Pessoa',
       Data_Nascimento as 'DT_Data de Nascimento',
       Sexo as 'OB_GR_Sexo'
from usuarios
where datepart(year, DataIngresso) = 2009
```

Uma extração definida com a consulta acima retornaria todos os usuários de um sistema que entraram no ano de 2009. As colunas a serem exibidas seriam “Código do Usuário”, obrigatória; “Nome da Pessoa”; “Data de Nascimento”, com a opção de refinamento por período; e “Sexo”, obrigatória, e com a possibilidade de refinamento a partir dos valores “F” e “M”.

É importante ressaltar que toda a consulta após a primeira cláusula *from* é livre para ser tão complexa quanto se deseje; inclusive, o *from* pode especificar uma *outra* consulta, ainda mais complexa, com *unions* e operações semelhantes. Isso não interfere no funcionamento da biblioteca.

### Parâmetros da Consulta

Outra necessidade da extração é de se definir valores dinâmicos em uma extração; tomando o exemplo da consulta anterior, é possível fazer com que a coluna *DataIngresso* não seja filtrada *apenas* pelo ano de 2009. Logicamente seria possível definir várias Extrações, cada uma com um ano, mas o recurso de *Parâmetros* torna isso desnecessário.

A biblioteca *Extrator* é internamente integrada com a biblioteca de consultas SQL *DBPHP*, apresentada no III Workshop de TI das IFES em 2009, e utiliza seus recursos de criação de consultas para facilitar essa tarefa. Assim, a consulta acima seria adaptada com essa alteração:

```
(...)
where datepart(year, DataIngresso) = :Ano[numero]
```

Essa alteração deveria ser registrada na tabela de extrações no banco de dados. Logicamente, a sintaxe no final da linha não faz parte da linguagem SQL, sendo um recurso da biblioteca PHP. Isso serve para que a aplicação defina, dinamicamente, através da classe *Extracao*, o valor daquele parâmetro. Este valor poderia vir de um formulário, por exemplo, onde o próprio usuário informa o ano desejado; ou então a aplicação realizaria um controle onde o *Ano* é sempre o ano anterior.

Outro detalhe é que, no caso de *Ano* ser um valor informado pelo usuário, a cláusula *where* da consulta acima pode ser abandonada, dando lugar a um uso mais inteligente dos filtros de coluna. Considere a consulta abaixo:

```
select Cod_Usuario as 'OB_Código do Usuário',
       Nome_Pessoa as 'Nome da Pessoa',
       Data_Nascimento as 'DT_Data de Nascimento',
       Sexo as 'OB_GR_Sexo',
       datepart(year, DataIngresso) as 'NU_Ano'
from usuarios
```

Essa alteração faz com que o *Ano* agora faça parte da consulta, e o preenchimento do ano seja feito com a própria interface da extração. O uso de uma função do lado direito da cláusula *as* não interfere de maneira nenhuma no funcionamento da classe; de fato, aquela posição pode ser ocupada por cláusulas *case*, outros *selects* e qualquer tipo de função que seja necessária, respeitando-se a obrigatoriedade da cláusula *as*.

### A classe *Extracao*

Como foi dito, a biblioteca também é composta de uma classe, denominada *Extracao*. É importante ressaltar que essa classe é um dos aspectos mais importantes da biblioteca, pois ela almeja ser extremamente simples e mínima. A classe se responsabiliza pelas duas tarefas essenciais do uso de extrações: a montagem do formulário, e a exportação dos dados propriamente ditos.

Um objeto *Extracao* é sempre instanciado a partir do número identificador da Extração que se deseja realizar. Neste momento, a classe realiza uma das operações essenciais: a leitura e interpretação do SQL para reconhecimento das colunas. Além de se reconhecer os nomes da coluna, reconhece-se, também, os prefixos de cada uma. Essa operação é necessária para as duas tarefas.

Após a instanciação, devem ser atribuídos os valores dos parâmetros da consulta. Esse é o único momento em que o programador deve ter um conhecimento um pouco mais profundo da consulta, pois *todos* os parâmetros devem ser devidamente atribuídos – mesmo que com o valor *NULL*. Para tanto, é recomendável uma mínima documentação para cada extração. Outra parametrização possível é a restrição dos tipos de arquivo a serem gerados. Embora essa opção pode estar na tabela de extrações, pode-se desejar alterar esse filtro em alguns casos específicos.

A montagem do formulário é simples, e consiste, primariamente, da geração de uma lista de *checkboxes*, um para cada coluna (fig. 1). O usuário deve marcar as colunas que deseja receber no arquivo exportado. Além disso, essa lista incorpora também os filtros, conforme foram determinados nos prefixos das colunas. Além disso, o formulário apresenta as opções de tipo de arquivo para o usuário, caso haja mais de uma opção. Tudo isso deve ser colocado dentro de um *form* HTML, cujo *action* deve apontar para a página que vai gerar os dados. Todo o formulário é desenhado a partir de uma única chamada de método da classe.

Cada um dos elementos HTML gerados trazem classes CSS, para que o desenvolvedor altere a apresentação e o layout do formulário, de acordo com as opções que o CSS fornece. Com isso é possível, por exemplo, dispor a lista na vertical ou na horizontal (com as opções da tag `<ul>`), utilizar cores de fundo para as linhas (inclusive utilizando um padrão de “zebra”, para facilitar a visualização) e alterar a face e o tamanho da fonte (fig. 2).

Quando o usuário mandar submeter esse formulário, ele será encaminhado para uma página PHP que deverá exportar os dados. Essa exportação, assim como a montagem do formulário, é realizada com uma única chamada de método, que deverá receber como argumento o vetor com os dados recebidos, geralmente o próprio vetor `$_POST`. Essa chamada deve ser precedida da instanciação do objeto e da atribuição dos valores dos parâmetros, assim como foi feito na exibição do formulário.

Neste momento, a classe executa a consulta daquela extração, aplicando os filtros que o usuário informou, e imprime os dados das colunas selecionadas de acordo com o formato de arquivo escolhido. O usuário receberá uma janela de diálogo para download do arquivo gerado. Com o uso de *headers* HTML, o navegador nem mesmo sai da tela do formulário, e o arquivo pode ser salvo ou aberto pelo usuário conforme ele desejar.

### Campos para extração:

<input checked="" type="checkbox"/> TipoOrgao	
<input type="checkbox"/> Extraído em	
<input checked="" type="checkbox"/> CartãoChefeDpto	
<input type="checkbox"/> Código	Valores: <input type="text"/>
<input type="checkbox"/> Unidade	Critério: <input type="text"/> <input checked="" type="radio"/> Termo inicial <input type="radio"/> Qualquer termo
<input type="checkbox"/> Depto	
<input type="checkbox"/> Solicitante	
<input type="checkbox"/> AreaConhecimento	
<input type="checkbox"/> Ramal	
<input type="checkbox"/> EMail	Critério: <input type="text"/> <input checked="" type="radio"/> Termo inicial <input type="radio"/> Qualquer termo
<input type="checkbox"/> Justificativa	
<input type="checkbox"/> Pré-requisitos	
<input type="checkbox"/> Didática	
<input type="checkbox"/> Escrita	
<input type="checkbox"/> Entrevista	
<input type="checkbox"/> Currículo	
<input type="checkbox"/> Disciplinas	
<input type="checkbox"/> Encaminhado em	Período: De <input type="text"/> até <input type="text"/>
<input type="checkbox"/> Certificação Direção	
<input type="checkbox"/> Obs Certificação Direção	
<input type="checkbox"/> Substituto Anterior	
<input type="checkbox"/> Análise PROGRAD em	
<input type="checkbox"/> Parecer PROGRAD	
<input type="checkbox"/> Deferimento	

[\[marcar todos\]](#)

Tipo de arquivo:

- Arquivo texto
- Planilha
- XML
- CSV

Figura 1: Formulário de extração sem estilos aplicados

### Segurança e Permissão

A biblioteca *Extração* é integrada ao sistema de segurança, que atribui permissões de acesso a sistemas e aplicações para determinados grupos, ou para usuários individualmente. Através desse sistema, pode-se saber se uma pessoa possui acesso a uma determinada aplicação. Cada extração é relacionada a uma aplicação, o que significa que o acesso a ela é restrito aos usuários que possuem essa relação. Essa verificação de acesso é automática, e realizada pela própria classe; ela não

permite que uma extração seja instanciada caso a pessoa que esteja logada no momento não tiver esse acesso.

**Campos para extração:**

<input checked="" type="checkbox"/> TipoOrgao			
<input type="checkbox"/> Extraído em			
<input checked="" type="checkbox"/> CartãoChefeDpto			
<input type="checkbox"/> Código	Valores:	<input type="text"/>	
<input type="checkbox"/> Unidade	Critério:	<input type="text"/>	<input checked="" type="radio"/> Termo inicial <input type="radio"/> Qualquer termo
<input type="checkbox"/> Depto			
<input type="checkbox"/> Solicitante			
<input type="checkbox"/> AreaConhecimento			
<input type="checkbox"/> Ramal			
<input type="checkbox"/> EMail	Critério:	<input type="text"/>	<input checked="" type="radio"/> Termo inicial <input type="radio"/> Qualquer termo
<input type="checkbox"/> Justificativa			
<input type="checkbox"/> Pré-requisitos			
<input type="checkbox"/> Didática			
<input type="checkbox"/> Escrita			
<input type="checkbox"/> Entrevista			

Figura 2: Trecho do formulário de extração com estilos CSS aplicados

É extremamente simples restringir o acesso a uma extração; o sistema e a aplicação estão na tabela de extrações. Portanto, para vincular uma extração a uma aplicação, basta preencher os respectivos campos do registro. É possível ainda outro tipo de restrição, determinando o diretório a partir do qual a extração pode ser acessada. Isso significa que qualquer arquivo PHP que estiver fora daquele caminho também não poderá instanciar aquela extração. Para isso, basta preencher o campo correspondente.

O objetivo disso é proteger dados sensíveis. Como a extração permite a exportação de grandes volumes de dados (e dependendo do propósito da extração, pode-se obter dados sensíveis e sigilosos), não é aceitável permitir o acesso irrestrito a essas páginas. A integração da extração com os sistemas de segurança é essencial, e não há dificuldade em integrar a biblioteca a outros diversos tipos de mecanismo de segurança.

### Grupos de Extrações

Outro recurso oferecido pela biblioteca é a criação de grupos de extração. A ideia é reunir extrações relacionadas a uma determinada aplicação em um sistema. Esse agrupamento pode ser feito a partir do *sistema* ou a partir do *grupo de usuários*.

Para este propósito, há uma tabela chamada GRUPOEXTRACAO. Cada linha associa uma extração a um grupo e a um serviço, sendo possível que uma mesma extração seja associada a vários grupos e serviços. A classe *GrupoExtracao* serve para trazer esse agrupamento para dentro dos scripts PHP.

Um *GrupoExtracao* pode ser instanciado a partir de um grupo *ou* de um serviço. Isso depende muito da aplicação, portanto as duas possibilidades foram implementadas. A classe se encarrega de procurar todas as extrações associadas à aplicação ou ao grupo passado como argumento. A classe então tenta instanciar cada uma dessas extrações, e devido aos mecanismos de segurança da própria classe *Extracao*, somente serão instanciadas as extrações às quais o usuário tem direito de acesso.

De forma semelhante à classe *Extração*, a classe *GrupoExtracao* possui duas operações básicas: imprimir o formulário e realizar a extração propriamente dita. A impressão do formulário consiste na impressão de todas as extrações, porém cada uma é encapsulada em um elemento HTML que

pode ser escondido e colapsado. A realização da extração é feita de forma semelhante, bastando apenas ao programador invocar um único método da classe, passando o vetor \$\_POST como argumento; a verificação de qual extração foi requisitada é feita pela própria classe.

Um problema ainda não resolvido nessa classe é a atribuição de parâmetros. Seguindo o modelo atual, o programador não tem acesso às extrações individualmente, pois a ideia é de que a criação e alteração desses grupos é feita exclusivamente no banco de dados, sem envolver alterações nos *scripts*. Porém, isso se torna um problema quando as extrações possuem parâmetros. Uma solução simples é possibilitar ao programador atribuir valores aos atributos a todo o grupo de extrações, o que faria com que o mesmo valor fosse atribuído a todas as extrações.

## **Conclusões**

A biblioteca *Extracao* pode ser considerada um sucesso, pois demonstrou que é possível criar soluções simples, úteis e genéricas, a partir da análise de problemas comuns em diversos contextos. O modelo, embora robusto e flexível, requer um número extremamente reduzido de tabelas, classes e arquivos PHP a serem incluídos, tornando-se simples de usar.

Até o presente momento, a biblioteca vem sendo integrada nos mais diversos sistemas, sendo um dos mais recentes o sistema de solicitação de selos de estacionamento, apresentando bons resultados. Cremos que o modelo é possível de ser utilizado, com facilidade, em diversos ambientes, com reduzida necessidade de adaptação e uma excelente relação custo/benefício.