**"Lou Paginar Mysql Class" package**

This PHP package, called "Lou Paginar Mysql", provides a reusable and organized way to implement pagination for data retrieved from a MySQL database. It consists of two main components:

The main advantages are:

1. Encapsulation:

Database connection and paging logic are encapsulated within classes, making the code more organized and reusable.

1. Centralized logic:

All paging-related code is in one place, making it easy to maintain and modify.

1. Simplified use:

DatabasePaginationWrapper simplifies the pagination deployment process by requiring minimal code in the main application file.

1. Clean code:

The code is well structured and commented, making it easy to understand.

1. Reusable:

Classes can be easily reused in other projects.

1. Clear separation of concerns:

Database connection and paging logic are separated into different classes.

In essence, this package is a well-designed, object-oriented solution for adding pagination to PHP applications that interact with MySQL databases.

DatabasePaginationWrapper class:

Objetive:

This class acts as a wrapper, encapsulating the database connection and paging logic. It simplifies the process of connecting to the database, retrieving paginated results, and displaying pagination links.

Features:

1. Database Connection: Handles connection to MySQL database using mysqli.
2. Error handling: checks for connection errors and terminates execution if an error occurs.
3. Pagination Integration: Creates an instance of the Pagination class to manage the pagination logic.

Méthods:

* ExibirResultados():
  + Takes an SQL query as input.
  + Retrieves the paginated results from the database using the Paginacao
  + Displays pagination links (First, Previous, page numbers, Next, Last).
  + Returns the result set for the current page.
* FecharConexao(): Closes the connection to the database.
* Constructor:
  + Takes as arguments the database connection parameters (host, user, password, database name) and the number of results per page.
  + Establishes connection to the database.
  + Initializes the Pagination object.

Paginação class:

Objective:

This class handles the main paging logic. Calculates the total number of pages, determines the current page and generates the pagination links.

Méthods

* obterResultados():
  + Takes an SQL query as input.
  + Counts the total number of results that match the query.
  + Calculates the total number of pages
  + Determines the OFFSET of the current
  + Add ORDER BY id DESC and LIMIT clauses to the query to retrieve only results from the current page.
  + Executes the modified query and returns the result set.
* getLinks():
  + Generates HTML links for pagination (First, Previous, page numbers, Next, Last).
  + Constructs the URLs for each link by including the page parameter in the query string (e.g. index.php?page=2
  + Returns links as a string, separated by " | ".
* Constructor:
  + Takes as arguments the database connection object and the number of results per page.
  + Sets the number of results per page
  + Determines the current page from the $\_GET['page'] parameter (defaults to 1 if not defined).

Operation (em index.php)

* Includes the paginate.php file, making the classes available.
* Database connection details (host, user, password, database name) and number of results per page are defined.
* A DatabasePaginationWrapper object is created, passing configuration details to its constructor.
* An SQL query is defined (for example, SELECT \* FROM noticias
* The wrapper's exibirResultados() method is called, passing the query.
* Retrieves paged results.
* Displays pagination links.
* Returns the result set for the current page.
* Iterates through the result set and displays the data for each row.
* Close connection.