

Unimar – Universidade de Marília

Análise e Desenvolvimento de Sistemas

M3.24 | Projeto Integrador Extensionista - ADS 3

Erasmão Cardoso da Silva

**Relatório do projeto**

2024

## M3.24 | Projeto Integrador Extensionista - ADS 3

Universidade de Marília  
Prof. Douglas Rodrigues  
Aluno : Erasmo Cardoso da Silva

### Introdução

Neste relatório, irei explicar e mostrar como ficou o projeto desenvolvido na atividade do Projeto Integrador. O projeto proposto na matéria consiste em uma aplicação fullstack onde é necessário realizar todas as operações do CRUD. Elaborei o banco de dados, o back-end e o front-end da aplicação. O app permite que o usuário cadastre seu nome, WhatsApp, e-mail, escolha o tipo de mensagem (crítica ou sugestão) e, por fim, digite sua mensagem.

Na tela de visualização de mensagens (READ), o usuário poderá ver uma lista de mensagens, selecionar um item para fazer modificações (UPDATE) ou, se desejar, apagar a mensagem (DELETE).

A ideia é, após realizar alguns testes, implantar este projeto no site da minha empresa (VMI Informática) para obter feedback dos clientes.

### Requisitos e ferramentas do projeto

Os requisitos descritos pelo Prof. Douglas Rodrigues para o projeto foram:

- HTML
- CSS
- PHP
- MySQL

Requisitos e ferramentas que utilizei no projeto:

- HTML
- CSS
- Bootstrap
- MySQL
- XAMPP (indicado pelo professor)
- Docker
- Docker Compose
- Composer

Optei por utilizar o Bootstrap, pois foi mencionado que, caso eu tivesse familiaridade com a ferramenta, poderia usá-la. Decidi, então, aplicar o Bootstrap em todo o projeto para melhorar a estética das telas.

O Docker e o Docker Compose foram escolhidos por serem ferramentas com as quais estou habituado, sendo até mais fácil de utilização pois utilizo Debian a anos. Isso facilitou a realização de testes em ambiente de produção bastante para mim. No entanto, como o XAMPP foi recomendado pelo professor, considere importante instalá-lo no meu sistema e projetei a aplicação para funcionar tanto no XAMPP quanto no Docker.

No arquivo README do [projeto](#), disponível no [GitHub](#), deixei todas as instruções detalhadas sobre como configurar o arquivo .env para ajustar as variáveis de ambiente, de acordo com a ferramenta escolhida para os testes (XAMPP ou Docker).

Como defini o uso do arquivo .env para o gerenciamento de variáveis de ambiente, precisei instalar uma biblioteca específica no projeto (vlucas/phpdotenv). Todas as informações relacionadas a essa configuração estão descritas no README. O uso do Composer foi necessário para a instalação dessa biblioteca, sendo assim, incluí-o como um dos requisitos do projeto.

### Estrutura do projeto “pastas e arquivos”



The screenshot shows the file structure of a GitHub repository named 'projetoIntegradorUnimar'. The interface includes a search bar, a list of files and folders, and columns for 'Nome', 'Tamanho', and 'Modifi'. The files and folders are: 'banco' (3 itens), 'www' (11 itens), 'api.http' (904 bytes), 'docker-compose.yml' (893 bytes), and 'README.md' (4,2 kB). Numbered callouts are present: '1' points to the repository name, '2' points to the 'banco' folder, and '3' points to the 'www' folder.

Nome	Tamanho	Modifi
banco	3 itens	
www	11 itens	
api.http	904 bytes	
docker-compose.yml	893 bytes	
README.md	4,2 kB	
















**1.** Pasta raiz Esta pasta deve ser utilizada apenas se for rodar o app via Docker. Nela, estão contidos:

- O arquivo README.md com a documentação do projeto;
- O arquivo api.http (plugin REST do VSCode), onde realizei os testes de funcionamento do backend (PHP) com o banco de dados (MySQL);
- O arquivo docker-compose.yml, que contém os dados necessários para que o Docker Compose busque no projeto os arquivos Dockerfile e monte os containers necessários para o funcionamento do app. São três containers: 1 – PHP, 2 – MySQL e 3 – Rede DB, para a interligação sem erros entre os dois containers mencionados. As demais pastas são explicadas nos itens 2 e 3.

## 2. Pasta banco Nesta pasta estão:

- O arquivo init.sql, que contém os dados e a configuração do banco de dados e das tabelas. As informações e o código estão descritos no README do projeto.
- O arquivo criarbanco.sql, que foi deixado como exemplo. No projeto, o init.sql é buscado para criar o banco de dados e as tabelas automaticamente, caso não existam no MySQL.
- O Dockerfile, que é responsável por montar o container do MySQL.

**3. Pasta www** O projeto principal está nesta pasta. Caso você deseje utilizá-lo com o XAMPP, basta copiar o conteúdo para a pasta htdocs e executá-lo. O banco de dados e as tabelas serão criados automaticamente. Esta pasta contém:

	assecs		1 item
	crud		7 itens
	css		1 item
	vendor		6 itens
	composer.json		62 bytes
	composer.lock		16,8 kB
	create.html		2,7 kB
	Dockerfile		980 bytes
	index.html		3,1 kB
	reader.php		2,6 kB
	updatepag.php		3,8 kB

- 1- Arquivos html e php que compõem o front do app.
- 2- A pasta assecs contém logotipo usado no app.
- 3- A pasta crud contém arquivos php onde se faz conexão com banco e crud.
- 4- A pasta css, contém arquivo responsável por parte do estilo do site, contando que a maioria do estilo é definido por Bootstrap.

## Front-End (Parte visual)

Como solicitado, o front-end foi desenvolvido utilizando HTML, CSS e Bootstrap. No total, são quatro páginas distribuídas em dois arquivos HTML e dois arquivos PHP. Os arquivos PHP foram usados principalmente para servir os dados HTML ao usuário, já que foi necessário aplicar funções para exibir e buscar dados. A escolha por PHP, ao invés de JavaScript, foi feita em conformidade com a solicitação de não utilizar JavaScript no [projeto](#).

- index.html
- create.html
- reader.php
- updatepag.php

### Index.html

UnimareAD

Projeto Integrador Extensionista - ADS 3  
Prof. Douglas Rodrigues  
Erasmio Cardoso da Silva

Sugestões ou Críticas do Site

Grave , Edite ou Apague as Mensagens  
Em visualizar voce pode editar ou apagar

Criar 1 Visualizar 2

Desenvolvido por: Erasmio Cardoso da Silva

1- Botão para acesso a pagina create.html, onde serão digitados os dados que deveram ser salvos no banco de dados. (CREATE)

2- Botão Visualizar , dará acesso ao read do aplicativo onde voce vera os dados salvos no banco e lá poderá atualizar ou apagar itens salvos. (READ | UPDATE | DELETE)

## Criar | CREATE (create.html)

Deixe sua mensagem

Nome:

Email:

WhatsApp:

Tipo:

Mensagem:

Gravar Voltar

Deixe sua mensagem

Nome:

Email:

WhatsApp:

Tipo:

Mensagem:

Gravar Voltar

## Sugestão ou Crítica salva com sucesso

[Voltar](#)

- 1 -Nome do usuário obrigatório.
- 2 -Email do usuário obrigatório.
- 3 -Whatsapp não obrigatório.
- 4 -Tipo o usuário selecionara entre Sugestão ou Critica.
- 5 -Mensagem aqui entrara o texto.

Ao cliente apertar o botão gravar os dados irão para o banco de dados e recebera a mensagem de sucesso (Sugestão ou Crítica salva com sucesso)..

## Ler dados | READ (reader.php)

Conectado.

### Dados do banco de dados

1	ID	Nome	Email	WhatsApp	Tipo	Mensagem	Ações
	1	Erasmus Cardoso da Silva	erasmo@vmi-informatica.com.br	+55 11 99999-9999	Sugestao	Teste de app 01	<a href="#">Editar</a> <a href="#">Apagar</a>
	2	Erasmus Cardoso da Silva	erasmo@vmi-informatica.com.br	+55 11 98888-8888	Critica	Teste 02 --- teste	<a href="#">Editar</a> <a href="#">Apagar</a>
	3	Erasmus Cardoso da Silva	erasmo@vmi-informatica.com.br	11949224355	Sugestao	Aqui colocamos todo o texto	<a href="#">Editar</a> <a href="#">Apagar</a>
	4	Jamile	jamile@vmi-informatica.com	+55 11 99999-9999	Critica	Teste 54	<a href="#">Editar</a> <a href="#">Apagar</a>

[Voltar](#)

Nesta tela o usuário tem uma visão completa de todos os dados do banco de dados, que são separados por suas "id" gerada automaticamente ao serem salvas no banco MYSQL, Ele poderá ver diretamente e assim ter maior facilidade de ação.

- 1 - Dados salvos separados por : id, nome, email, whatsapp, tipo, Mensagem.
- 2 - Acesso a edição da ID selecionada, com a id seleciona o usuário ira para a pagina de edição(UPDATE) e assim poderá modificar as informações.
- 3 - Ao clicar em apagar fará diretamente a parte de DELETE do CRUD, como não precisa de muitas informações para apagar este item achei melhor apenas deletar.

## Editar | UPDATE (updatepag.php)

### Editar Mensagem

Nome:

Email:

WhatsApp:

Tipo:

Crítica

▼

Mensagem:

Teste 54

Salvar

Voltar

Atualizado com sucesso

[Voltar](#)

Quando o usuário clica no botão "Editar" na página de visualização (reader), o processo de edição começa com o envio do ID da sugestão para a página updatepag.php. Isso é feito por meio de um link gerado em HTML, no qual o ID da sugestão é passado como um parâmetro na URL. O código responsável por isso é o seguinte:

```
{ <a href="updatepag.php?id=<?php echo  
htmlspecialchars($suggestion['id']); ?>" class="btn btn-warning btn-  
sm">Editar</a> }
```

Aqui, a função `htmlspecialchars($suggestion['id'])` é utilizada para garantir que o ID seja devidamente escapado antes de ser inserido na URL, evitando possíveis problemas de segurança.

Ao clicar nesse link, o navegador redireciona o usuário para a página updatepag.php, e o ID da sugestão é passado como parte da URL, assim: `{ updatepag.php?id=1 }`

Na página updatepag.php, o ID é capturado usando a variável global `$_GET`, que contém os parâmetros enviados pela URL. Isso é feito da seguinte forma:

```
{ $id = isset($_GET['id']) ? intval($_GET['id']) : 0; }
```

Se o parâmetro `id` existe na URL. Se ele existir, uso a função `intval()` para convertê-lo em um número inteiro, garantindo que seja um valor seguro. Caso contrário, o valor 0 é atribuído à variável `$id`. Peço desculpa eu sei que o PROF. Pediu para não colocar código mais acho que isso é bem importante para o trabalho.



## Apagar | DELETE

### Dados do banco de dados

ID	Nome	Email	WhatsApp	Tipo	Mensagem	Ações
1	ErasmO Cardoso da Silva	erasmo@vmi-informatica.com.br	+55 11 99999-9999	Sugestao	Teste de app 01	<a href="#">Editar</a> <a href="#">Apagar</a>
2	ErasmO Cardoso da Silva	erasmo@vmi-informatica.com.br	+55 11 98888-8888	Critica	Teste 02 --- teste	<a href="#">Editar</a> <a href="#">Apagar</a>

[Voltar](#)

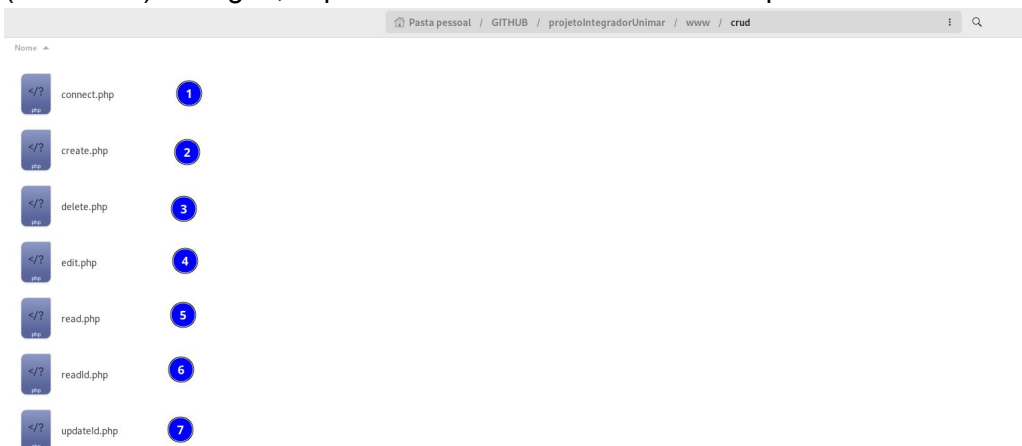
## Apagado com sucesso

[Voltar](#)

Como dito anteriormente a id é passada para o algoritmo que se encontra na pasta CRUD/delete.php , onde o algoritmo apenas apaga o item referente a id e nos retorna a mensagem “Apagado com sucesso” . Com isso fechamos todas as operações do crud.

## Backend

1. Toda a estrutura do backend da aplicação está concentrada na pasta CRUD (www/crud). A seguir, explico o funcionamento de cada arquivo:



### 1 - Connect.php

Algoritmo responsável por conectar a aplicação ao banco de dados MySQL. Se o banco de dados ou as tabelas não existirem, este script os cria automaticamente. Todas as operações de CRUD (Criar, Ler, Atualizar e Deletar) passam por essa conexão.

### 2 – Create.php

Este script é utilizado para inserir novos registros no banco de dados. Ele recebe os dados do front-end e os insere em uma nova posição no banco.

### 3 – Delete.php

Recebe o ID específico de um registro e o deleta do banco de dados. A exclusão é feita de acordo com a ID fornecida, garantindo que o registro correto seja removido.

### 4 – Edit.php

Atualizar os dados de um registro existente no banco. Ele recebe as novas informações e as aplica à posição identificada pela ID selecionada.

### 5 – Read.php

Faz a leitura de todos os registros do banco de dados e envia os resultados para o front-end, permitindo que os dados sejam exibidos ao usuário.

### 6 – ReadId.php

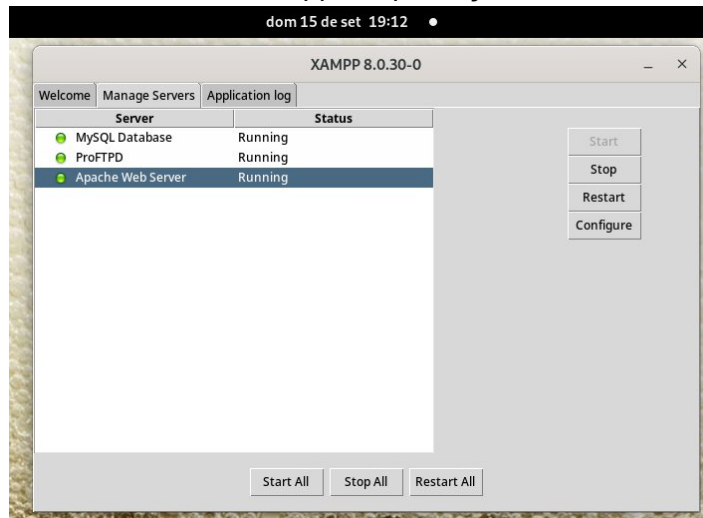
Realiza a busca de um registro específico com base no ID fornecido. É utilizado principalmente para carregar os dados de um item quando o usuário deseja editar um registro..

### 7 – UpdateId.php

Este script faz a atualização de um registro no banco de dados com base no ID passado. Ele executa um comando UPDATE para alterar os dados do item específico.

## Testes Xampp e Docker

### xampp em produção



### Docker em produção

```
erasto@erasto-poseih61ce: ~/GITHUB/projetoIntegradorUnimar
erasto@erasto-poseih61ce:~/GITHUB/projetoIntegradorUnimar$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
e39494a2f7f2   projetointegradorunimar-web         "docker-php-entrypoi..." 3 minutes ago  Up 3 minutes  0.0.0.0:8080-
78992391a790   projetointegradorunimar-db         "docker-entrypoint.s..." 3 minutes ago  Up 3 minutes  0.0.0.0:3306-
erasto@erasto-poseih61ce:~/GITHUB/projetoIntegradorUnimar$ docker network ls
NETWORK ID     NAME                                  DRIVER  SCOPE
e611e0ee9fe2   bridge                               bridge  local
2d73e41f0490   host                                 host    local
851c74b4e787   none                                 null    local
f1270f1d6d2c   projetointegradorunimar_default     bridge  local
cb991ef4ac89   projetointegradorunimar_backend     bridge  local
erasto@erasto-poseih61ce:~/GITHUB/projetoIntegradorUnimar$
```

# Link do projeto no GITHUB

LINK  
[projetoIntegradorUnimar](https://github.com/erascardsilva/projetoIntegradorUnimar)

erascardsilva / projetoIntegradorUnimar

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights

Settings

projetoIntegradorUnimar

Private

Unwatch 1

Fork 0

main 1 Branch 0 Tags

Go to file

Add file

<> Code

erascardsilva Update README.md 514f1f3 · 3 hours ago 36 Commits

banco	update readme.md	3 days ago
www	modificado .env para uso no xampp	3 days ago
README.md	Update README.md	3 hours ago
api.http	update back	5 days ago
docker-compose.yml	update docker	5 days ago

README

## Projeto Integrador UNIMAR

### Descrição do Projeto

Este é um projeto desenvolvido para o material de Projeto Integrador da UNIMAR, sob a orientação da Prof.

About

No description, website provided.

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

[github.com/erascardsilva/projetoIntegradorUnimar](https://github.com/erascardsilva/projetoIntegradorUnimar)