

Consolidando o seu conhecimento

Chegou a hora de você seguir todos os passos realizados por mim durante esta aula. Caso já tenha feito, excelente. Se ainda não, é importante que você execute o que foi visto nos vídeos para poder continuar com a próxima aula.

1) Abra o arquivo `view/cursos/listar-cursos.php` e adicione dois novos links para alterar e excluir um curso. Além disso, use as classes do bootstrap para estilizar os elementos. Segue o element `<li>` completo:

```
<li class="list-group-item d-flex justify-content-between">
    <?= $curso->getDescricao(); ?>

    <span>
        <a href="/alterar-curso?id=<?= $curso->getId(); ?>" class="btn btn-info btn-sm">
            Alterar
        </a>
        <a href="/excluir-curso?id=<?= $curso->getId(); ?>" class="btn btn-danger btn-sm">
            Excluir
        </a>
    </span>
</li>
```

2) Defina duas novas rotas no arquivo `config/routes.php` , dentro do array:

```
'/excluir-curso' => Exclusao::class,
'/alterar-curso' => FormularioEdicao::class,
```

Opcionalmente pode reorganizar e agrupar os imports.

3) Crie um novo controlador `Exclusao` para implementar a exclusão:

```
<?php

namespace Alura\Cursos\Controller;

use Alura\Cursos\Entity\Curso;
use Alura\Cursos\Infra\EntityManagerCreator;

class Exclusao implements InterfaceControladorRequisicao
{
    /**
     * @var \Doctrine\ORM\EntityManagerInterface
     */
    private $entityManager;

    public function __construct()
    {
        $this->entityManager = (new EntityManagerCreator())
            ->getEntityManager();
    }
}
```

```

public function processaRequisicao(): void
{
    $id = filter_input(
        INPUT_GET,
        'id',
        FILTER_VALIDATE_INT
    );

    if (is_null($id) || $id === false) {
        header('Location: /listar-cursos');
        return;
    }

    $curso = $this->entityManager->getReference(
        Curso::class,
        $id
    );
    $this->entityManager->remove($curso);
    $this->entityManager->flush();
    header('Location: /listar-cursos');
}
}

```

4) Na pasta `view/cursos/`, abra o arquivo `formulario.php` para funcionar tanto para inserção quanto alteração:

```

<?php include __DIR__ . '/../inicio-html.php'; ?>

<form action="/salvar-curso<?= isset($curso) ? '?id=' . $curso->getId() : ''; ?>" method="post">
    <div class="form-group">
        <label for="descricao">Descrição</label>
        <input type="text"
            id="descricao"
            name="descricao"
            class="form-control"
            value="<?= isset($curso) ? $curso->getDescricao() : ''; ?>"
        </div>
        <button class="btn btn-primary">Salvar</button>
    </form>

<?php include __DIR__ . '/../fim-html.php'; ?>

```

5) Na pasta `src/Controller`, crie uma classe abstrata `ControllerComHtml` que possui a funcionalidade de renderizar o HTML:

```

<?php

namespace Alura\Cursos\Controller;

abstract class ControllerComHtml
{
    public function renderizaHtml(string $caminhoTemplate, array $dados): string
    {
        extract($dados);
        ob_start();
        require __DIR__ . '/../../view/' . $caminhoTemplate;
    }
}

```

```

$html = ob_get_clean();

return $html;
}
}

```

6) Ainda na pasta `src/Controller`, crie uma nova classe `FormularioEdicao` para chamar o formulário de alteração. A classe deve estender a classe `ControllerComHtml`:

```

<?php

namespace Alura\Cursos\Controller;

use Alura\Cursos\Entity\Curso;
use Alura\Cursos\Infra\EntityManagerCreator;

class FormularioEdicao extends ControllerComHtml implements InterfaceControladorRequisicao
{
    /**
     * @var \Doctrine\Common\Persistence\ObjectRepository
     */
    private $repositorioCursos;

    public function __construct()
    {
        $entityManager = (new EntityManagerCreator())
            ->getEntityManager();
        $this->repositorioCursos = $entityManager
            ->getRepository(Curso::class);
    }

    public function processaRequisicao(): void
    {
        $id = filter_input(
            INPUT_GET,
            'id',
            FILTER_VALIDATE_INT
        );

        if (is_null($id) || $id === false) {
            header('Location: /listar-cursos');
            return;
        }

        $curso = $this->repositorioCursos->find($id);

        echo $this->renderizaHtml('cursos/formulario.php', [
            'curso' => $curso,
            'titulo' => 'Alterar curso ' . $curso->getDescricao(),
        ]);
    }
}

```

7) Abra a classe `ListarCursos` e estenda a classe `ControllerComHtml`:

```
class ListarCursos extends ControllerComHtml implements InterfaceControladorRequisicao
```

Além disso, altere o método `processaRequisicao` que fica:

```
public function processaRequisicao(): void
{
    echo $this->renderizaHtml('cursos/listar-cursos.php', [
        'cursos' => $this->repositorioDeCursos->findAll(),
        'titulo' => 'Lista de cursos',
    ]);
}
```

8) Faça as mesmas alterações na classe `FormularioInsercao`. Isto é, estender a classe `ControllerComHtml` e alterar o método `processaRequisicao`

```
class FormularioInsercao extends ControllerComHtml implements InterfaceControladorRequisicao
```

E o método fica como:

```
public function processaRequisicao(): void
{
    echo $this->renderizaHtml('cursos/formulario.php', [
        'titulo' => 'Novo curso'
    ]);
}
```

9) Abra a classe `src/Controller/Persistencia` e altere de maneira que ela funcione para inserção e alteração. Segue apenas o método `processaRequisicao` já com as alterações:

```
public function processaRequisicao(): void
{
    $descricao = filter_input(
        INPUT_POST,
        'descricao',
        FILTER_SANITIZE_STRING
    );

    $curso = new Curso();
    $curso->setDescricao($descricao);

    $id = filter_input(
        INPUT_GET,
        'id',
        FILTER_VALIDATE_INT
    );

    if (!is_null($id) && $id !== false) {
        $curso->setId($id);
        $this->entityManager->merge($curso);
    } else {
        $this->entityManager->persist($curso);
    }
}
```

```
}

$this->entityManager->flush();

header('Location: /listar-cursos', true, 302);
}
```

10) Na linha de comando, navegue até a raiz do seu projeto e execute o comando Doctrine para processar as classes entidades:

```
vendor\bin\doctrine orm:generate-proxies
```

11) Garanta que o servidor PHP está rodando (`php -S localhost:8080 -t public`) e verifique as novas funcionalidades. Tente alterar e excluir um curso.