PHPActiveRecord/ORM:

https://web.archive.org/web/20170516222345/http://www.phpactiverecord.org/

O que levar feito (pelo menos) para a próxima aula:

1. Um controlador por cada tema (modelo)
2. Um modelo para cada tabela (à exceção da tabela resultante do M-M)
    - Aplicar validações para cada modelo (pelo menos de presence_of) – ver a documentação do tópico Validations do site PHPActiveRecord.org.
3. O login a funcionar:
    - Requer uma vista de login (com o formulário de login);
    - Requer uma rota GET para abrir o formulário;
    - Requer uma rota POST para submeter o formulário;
    - Requer um método no controlador (possivelmente UserController) para ler o formulário e verificar se existe algum utilizador com aquelas credenciais lidas do formulário. Dica: utilizar os dynamic finders da documentação do PHPActiveRecord.org.
        - Se sim: guarda a informação do user numa variável de sessão;
        - Se não: redireciono para a vista de login de novo;
4. Efetuar o registo do utilizador;

   IMPORTANTE: o campo da data de nascimento na tabela da base de dados não deve ter espaços, underscores (_), hífens ou qualquer letra maiúscula.

   Exemplo:

Como deve ser?	Como **não** deve ser?
birthdate	birthDate / birth_date / birth-date

5. Uma outra funcionalidade do Passageiro (à vossa escolha)

Pequenos apontamentos:

Relação entre a base de dados e a framework:

- Configuração da base de dados no ficheiro db.php (webapp/start-up/config)
- Nome das tabelas em Inglês e no plural;
- Criar um modelo para cada tabela;
- Nome do modelo deve ser igual ao da tabela mas no singular.

  Exemplo:

Base de dados (tabela)	Modelo
Cars	Car

- Criar as relações no modelo, se necessário (ver o menu Associations da documentação do site PHPActiveRecord.org)

Rotas:

GET:

Devo utilizar uma rota GET quando apenas quero devolver uma vista (com ou sem informação para a vista interpretar)

- Método no controlador que me devolve a vista com ou sem dados

POST:

Devo utilizar uma rota POST quando submeto um formulário.

- Criar uma rota POST
- Devo colocar a chamada à rota no atributo action do form.
- Devo ter um método no controlador que vai:
  - ler o formulário através do Post::get('informacao') ou Post::getAll();
  - guardar alguma informação na base de dados.

Resource

- Devo utilizar uma rota RESOURCE quando o meu controlador será CRUD e irá implementar a interface.

CRUD:

Create **R**ead **U**pdate e **D**elete

Se vou aplicar todos estes métodos então devo implementar a interface: ResourceControllerInterface.

Recordo que ao aplicar esta interface, as funções são automaticamente geradas, mas para isso devem clicar em cima da lâmpada vermelha que vai aparecer no início da declaração da classe.

(ver o exemplo do BookController, que se encontra na pasta wlstudio)

Autorização:

Este tópico serve para autorizar ou não os utilizadores em entrar nas vossas páginas.

Devem sempre que necessário, validar se o utilizador está autenticado e se tem ou não o perfil adequado para aceder a tal página.

Dado que colocam numa variável de sessão o utilizador quando este se autentica então devem utilizar essa variável de sessão para fazer a validação.

Como?

A classe Session, possui o método has() que vos indica se a variável que estão a procura existe ou não, está ou não definida. Exemplo:

```
if(Session::has('user')) {

    // posso ver o botão x ou y

} else {

    //escondo o botão de logout (por exemplo)

}
```