**\*INSTALANDO COMPOSER (CMD)**

composer global require fxp/composer-asset-plugin:^1.2.0 (global)

composer create-project --prefer-dist yii2-app-basic yii2-basico (criando uma aplicaçao)

composer create-project --prefer-dist yiisoft/yii2-app-basic basic (criando um aplicação-correto)

composer create-project --prefer-dist --stability=dev yiisoft/yii2-app-basic basic (Nova instalação)

**\*CASO DE ERROR SWIFTMAILER**

Apaga o swiftmailer do "composer.json";

No terminal dá um "composer install" no diretório da aplicação.

-----------------------------------------------------------------------------------------------------------------------------

**\*MIGRATION**

php yii migrate/create create\_usuario\_table (criar a migração nas pastas)

cria a migração na pasta da aplicação para você fazer as possíveis preenchimento de dados e inserts.

php yii migrate (criar a tabela)

cria a migração que você fez acima na pasta da aplicação, migrando para o banco de dados.

Na pasta config, é preciso acrescentar o código abaixo dentro de componentes:

'authManager' => [

'class' => 'yii\rbac\DbManager'

]

*php yii migrate --migrationPath=@yii/rbac/migrations*

cria a migration, informando o caminho que você quer que rode esse migrate. rodando dentro da pasta vendor yii ...

Observação: quando se cria um authManager no arquivo "web" também se cria no arquivo "console", assim ele cria no banco as tabelas (auth\_assignment, auth\_item, auth\_item\_child, auth\_rule).

**\*DAVIDSON AUTENTICAÇÃO**

**\*MIGRATION**

É a definição que se dá ao gerenciamento de mudanças incrementais e reversíveis em esquemas (estrutura) de banco de dados. Isso permite que seja possível ter um controle "das versões" do banco de dados.

**FILTRO DE CONTROLE DE ACESSO (ACF)**

É o método simples de autorização implementado como yii\filters\AcessControl, que no caso é utilizado na função behaviors ().

only => determina que o ACF só deva ser aplicado nas ações colocadas nele.

rules => lista as regras de acesso se (?)não for convidado pode acessar, e se (@) estiver logado pode acessar a ação.

allow => regra que irá dizer se o usuário está autorizado ou não. Se nenhuma das regras forem correspondidas, significa que o usuário não está autorizado, e a ACF vai parar a próxima execução.

matchCallback => Permite criar uma chamada, onde esta página pode ser acessada somente a data colocada na função.

**CONTROLE DE ACESSO BASEADO EM ROLE (RBAC)**

role => Uma role representa uma coleção de permissões (ex. criar posts, atualizar posts). Uma role pode ser atribuído a um ou vários usuários.

AUTH\_ASSIGNMENT

assignmentTable: tabela para armazenar tarefas de itens de autorização. O padrão é "auth\_assignment".

AUTH\_ITEM

itemTable: tabela para armazenar itens de autorização. O padrão é "auth\_item".

AUTH\_ITEM\_CHILD

itemChildTable: tabela para armazenar hierarquia de itens de autorização. O padrão é "auth\_item\_child".

AUTH\_RULE

ruleTable: tabela para armazenar as regras. O padrão é "auth\_rule".

**CRIANDO AS TABELAS (RBAC)**

É preciso criar essas tabelas no banco de dados. Para fazer isto, você pode usar o migration armazenado em @yii/rbac/migrations:

*yii migrate --migrationPath=@yii/rbac/migrations*

O authManager já pode ser acessado via \Yii::$app->authManager.

Se a sua hierarquia de permissões não se altera e você tem um número fixo de usuários pode-se criar um console command que irá iniciar os dados de autorização uma vez através das APIs oferecidas pelo authManager:

Depois de executar o comando com yii rbac/init nós vamos chegar a seguinte hierarquia:

EX: <https://www.yiiframework.com/doc/guide/2.0/pt-br/security-authorization>

Autor pode criar post, admin pode atualizar post e fazer tudo que autor pode.

Para aplicações que requerem controle de acesso complexo com dados de autorização atualizados dinamicamente, interfaces de usuário especiais (Isto é: painel de administração) pode ser necessário desenvolver usando APIs oferecidas pelo authManager.

OBSERVAÇÃO: quando se cria o usuário que faz a ligação com o perfil de acesso.

-----------------------------------------------------------------------------------------------------------------------------

**\*SERVIDOR YII2 (CMD)**

No CMD você roda o código abaixo dentro da pasta da aplicação ex: c:\xampp\htdocs\yii2-basico

php yii serve

**\*CRIAR SERVIDOR E FAZER COM QUE TODAS AS REQUIZIÇÕES VÃO PARA UM LOCAL ESPECIFICO**

=> php -S localhost:8000 -t public public/index.php

=> php -S localhost:8080 -t public/

-----------------------------------------------------------------------------------------------------------------------------

**\*BASE DE DADOS E COMPONENTES DE FUNDAÇÃO PARA TODAS AS EXTENSÕES DO YII2 KRAJEE**

composer require kartik-v/yii2-krajee-base "@dev-master" (base)

composer require kartik-v/yii2-password "@dev-master" (password)

composer require kartik-v/yii2-number (número - máscara de dinheiro)

composer require kartik-v/yii2-datecontrol (data e tempo)

composer require kartik-v/yii2-widgets "@dev" (widgets)

composer require kartik-v/yii2-builder "@dev" (FormBuilder)

composer require kartik-v/yii2-tabs-x (Tabs - abas em uma página)

composer require kartik-v/yii2-grid "@dev" (GridView)

composer require kartik-v/yii2-detail-view "@dev" (Detail-view)

composer require dmstr/yii2-adminlte-asset "\*" (Tema Admin LTE no Yii2)

composer require "2amigos/yii2-tinymce-widget" "\*" (Editor de texto)

composer require kartik-v/yii2-mpdf "\*" (PDF)

composer require kartik-v/yii2-widget-select2 "dev-master" (DropDownList)

composer require miloschuman/yii2-highcharts-widget (GRÁFICOS JS)

composer require 2amigos/yii2-chartjs-widget:~2.0 (GRÁFICOS JS - 2º OPÇÃO)

-----------------------------------------------------------------------------------------------------------------------------

**\*FLUXO YII2 MVC**

View -> Usuário -> ação -> requisição -> controllers -> models

controllers -> response -> view -> Usuário.

**\*CONTROLLER**

render => renderiza o html que está na view, ou seja, mostra o templete da página (index) como se fosse uma rota.

Yii::$app->request; => quando você chama o Yii com y maiúsculo você vai pode chamar e ter acesso a quase tudo na sua aplicação.

Request => requisição

--------------------------------------------------------------------------------------------------------------------------

**\*RASCUNHO**

TIC -> Tecnologia de Informação e Comunicação

TAP -> Termo de Abertura do Projeto

1. Priorização - quantos níveis vai ter e quais?
2. Quando o responsável está preenchendo o formulário, no final do formulário que será incluso o botão "Concluir Planejamento Anual de TIC" ou irá salvar primeiro e na página view que vai incluir o botão?
3. Depois do planejamento enviado pelo responsável ele poderá ver a lista ou editar? ou clicou em enviar já era, e somente a STI poderá ver?

---------------------------------------------------------------------------------------------------------------------------

**\*CÓDIGOS**

$model->setAttribute ('senha', Yii::$app->security->generatePasswordHash($model->getAttribute('senha')));

é o mesmo que você fazer assim:

$senha = Yii::$app->security->generatePasswordHash($model->getAttribute('senha'));

$model->setAttribute ('senha', $senha);

---------------------------------------------------------------------------------------------------------------------------

**\*LINKS IMPORTANTES**

https://www.linhadecomando.com/php/php-porcentagem-entre-data-inicial-e-final

---------------------------------------------------------------------------------------------------------------------------

**\*SISTEMA - ALTERAÇÃO SISEMANET**

Nome no sistema - Models - conexão banco oracle

Usuário Sisemanet = tbpessoa - (MBPO.tb\_usuario)

Empreendedor = pessoajuridica - (MBPU.tb\_pessoa)

Empreendimento = tbempreendimento - (MBPU.tb\_empreendimento)

Pessoa Jurídica Carregada = tbpessoajuridicacarregada - (MBPU.tb\_pessoa\_juridica\_carregada)

-----------------------------------------------------------------------------------------------------------------------------

**\*CURSOS**

https://rocketseat.com.br/starter

Ordem para ver os cursos:

Javascript

Javascript ES6+

NodeJS

ReactJS

React Native

---------------------------------------------------------------------------------------------------------------------------

Davidson Almeida da Silva

Diretoria de Gestão de Tecnologia da Informação - DGTI

(31) 3915-1595 - davidson.silva@meioambiente.mg.gov.br

Secretaria de Estado de Meio Ambiente e Desenvolvimento Sustentável - SEMAD

---------------------------------------------------------------------------------------------------------------------------