

MAGENTO 2020

DICIEMBRE 3

Creado por: María Cortés Palacios



Creación módulo

Crear un módulo que administre los exámenes guardando el nombre y el apellido de los alumnos y la nota que han obtenido

Para empezar creamos un directorio dentro de app/code con el nombre de nuestro vendor y nuestro modulo. Quedará tal que: app/code/Hiberus/Cortes.

Los módulos necesitan dos archivos necesarios para funcionar: registration.php y etc/module.xml.

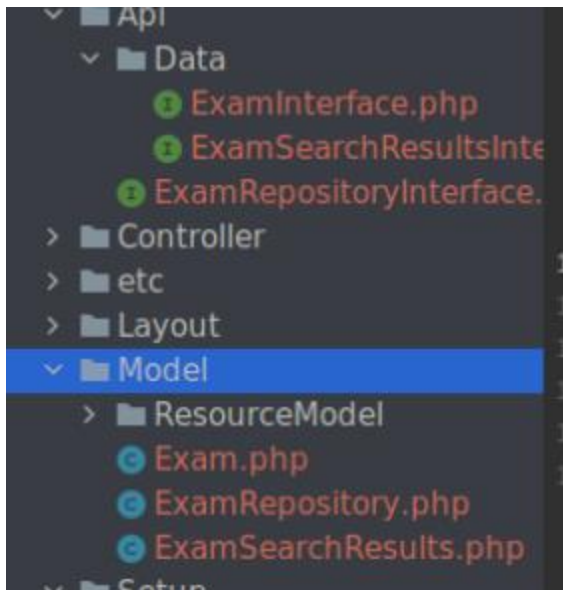
```
<?php
\Magento\Framework\Component\ComponentRegistrar::register(
    type: \Magento\Framework\Component\ComponentRegistrar::MODULE,
    componentName: 'Hiberus_Cortes',
    path: __DIR__
);
```

```
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:Module/etc/module.xsd">
    <module name="Hiberus_Cortes" />
</config>
```

A continuación creamos la estructura que tendrá nuestros datos, lo que vamos a ver en la base de datos. Esto lo realizamos mediante el archivo etc/db_schema.xml

```
<?xml version="1.0"?>
<schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:Setup/Declaration/Schema">
    <table name="hiberus_exam" resource="default" engine="innodb" comment="Hiberus Exam Table">
        <column xsi:type="int" name="id_exam" padding="10" unsigned="true" nullable="false" identity="true" comment="Primary key"/>
        <column xsi:type="varchar" name="firstname" length="100" nullable="false" comment="First Name"/>
        <column xsi:type="varchar" name="lastname" length="250" nullable="false" comment="Last Name"/>
        <column xsi:type="float" name="mark" scale="2" precision="4" nullable="false" comment="Mark"/>
        <constraint xsi:type="primary" referenceId="PRIMARY">
            <column name="id_exam"/>
        </constraint>
    </table>
```

Y posteriormente, para rellenar nuestra base de datos creamos primero la Api y el Model.



A partir de ahí ya podremos crear nuestro data patch. Creamos los archivos Setup/Patch/Data/LinkDataModel.php Setup/Patch/Data/PopulateDataModel.php.

Para que se rellene tenemos que entrar en el contenedor y ejecutar bin/magento setup:upgrade. Borramos cache y podemos comprobar en workbench (o en el programa de administración que usemos) que nos ha rellenado la tabla (hiberus_exam).

Creación controlador frontend

Crear un controlador de frontend y poner un echo para comprobar que funciona

Creamos un archivo Controller/MiControlador/index.php. Y en el directorio etc creamos frontend/routes.xml. Dentro de index.php añadimos un echo. Al borrar cache y hacer setup:upgrade si entramos en la ruta indicada en routes.xml nos saldrá el echo comprobando que el controlador funciona.

```
<?php
namespace Hiberus\Cortes\Controller\MiControlador;

class Index extends \Magento\Framework\App\Action\Action {
    protected $pageFactory;

    public function __construct(
        \Magento\Framework\App\Action\Context $context,
        \Magento\Framework\View\Result\PageFactory $pageFactory
    ) {
        $this->pageFactory = $pageFactory;
        parent::__construct($context);
    }

    public function execute() {
        echo ("hola");
        return $this->pageFactory->create();
    }
}
```

```
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="urn:magento:framework:App/etc/routes.xsd">
    <router id="standard">
        <route id="cortes" frontName="listado">
            <module name="Hiberus_Cortes" />
        </route>
    </router>
</config>
```