

Francisco Javier Sanchez Peña

## 1. Crear Modulo en Magento:

Existen dos modos de crear un modulo en magento, basadas en la localizacion de los archivos, yo la creo de la siguiente forma:

Archivo “registration.php” en la ruta: App/Code/Hiberus/sanchez/registration.php

```
<?php
use Magento\Framework\Component\ComponentRegistrar;

ComponentRegistrar::register(ComponentRegistrar::MODULE, 'Hiberus_sanchez', __DIR__);
```

Y el Archivo “module.xml” en la ruta: App/Code/Hiberus/sanchez/etc/module.xml

```
<?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_sanchez" setup_version="1.0.0">
    </module>
</config>
```

Despues haremos uso del comando “dockergento magento setup:upgrade” y del comando “dockergento magento module\_enable Hiberus\_sanchez” para activarlo

## 2. Se crea el archivo “db\_schema” en la ruta: App/Code/Hiberus/sanchez/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/etc/schema.xsd">
    <table name="hiberus_exam" resource="default" engine="innodb" comment="Tabla Examen Hiberus">
        <column xsi:type="int" name="id_exam" padding="11" unsigned="true" nullable="false" identity="true" />
        <column xsi:type="varchar" name="firstname" length="100" unsigned="true" nullable="false" />
        <column xsi:type="varchar" name="lastname" length="250" unsigned="true" nullable="false" />
        <column xsi:type="decimal" name="mark" unsigned="true" nullable="false" precision="4" scale="2"/>
        <constraint xsi:type="primary" referencId="PRIMARY">
            <column name="id_exam"/>
        </constraint>
    </table>
</schema>
```

Después, se lanza el comando “dockergento magento setup:upgrade”

- 3.
4. No se puede realizar, no lo hemos visto en el curso

5. Se crea el archivo index.php en la ruta:

“App/Code/Hiberus/sanchez/Controller/Index/index.php”, que contendra la funcion execute();

```
class Index implements HttpGetActionInterface
{
    protected PageFactory $pageFactory;

    public function __construct(
        Context $context,
        PageFactory $pageFactory
    ){
        $this->pageFactory=$pageFactory;
    }

    public function execute()
    {
        echo "hola";
        exit;
    }
}
```

Y en “App/Code/Hiberus/sanchez/etc/frontend/routes.xml” indicamos el controlador creado antes.

```
<?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 frontName="examen" id="examen">
            <module name="Hiberus_sanchez"/>
        </route>
    </router>
</config>
```

6. Se crea la clase “index.php” dentro en “sanchez/Block/index.php”

```
class Index extends Template
{
    public $collection;

    public function __construct(
        Context $context,
        CollectionFactory $collectionFactory,
        array $data = []
    ) {
        $this->collection=$collectionFactory;
        parent::__construct($context, $data);
    }

    public function getCollection(){
        return $this->collection->create();
    }
}
```

Se crea el archivo “index.phtml” en la ruta “/sanchez/view/froned/templates/index.phtml”

Se crea el archivo “examen\_index\_index.html” en la ruta  
“/sanchez/view/frontend/layout/examen\_index\_index.html”

```
<?xml version="1.0"?>
<page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" layout="1column"
xsi:noNamespaceSchemaLocation="urn:magento:framework:View/Layout/etc/
page_configuration.xsd">
    <referenceContainer name="content">
        <block class="Hiberus\sanchez\Block\Index" name="examen_index_index"
template="Hiberus_Pardo::index.phtml" />
    </referenceContainer>
    <head>
        <css src="Hiberus_Sanchez::custom.css"/>
    </head>
</page>
```

6.1

```
<legend><h2 class="title">Alumnos</h2></legend>
<ul class="listaAlumnos">
```

6.2 <li id="alumnoSuspenso">Nombre: <?php echo \$alumno->getFirstname()." ".\$alumno->getLastName()." -> ".\$nota; ?></li>

6.3 Total number of students: <?php echo count(\$alumnos) ?>

6.4 No se explicó.

7. No se puede realizar, no lo hemos visto en el curso

8. Se crea el archivo “custom.less” en la ruta “/sanchez/view/froned/web/custom.less” y creamos la relacion en con el archivo “/sanchez/view/frontend/layout/examen\_index\_index.xml”

```
<head>
    <css src="Hiberus_Sanchez::custom.css"/>
</head>
```

8.1 En el archivo de css custom agregado anteriormente, agregamos los estilos:

```
@color: blue;
@colorWidth: darkorange;

.title{
    color:@color;
}

@media (max-width: 768px){
    .title{
        color:@colorWidth;
    }
}
```

8.2

8.3

```
@margin-left-primary:20px;
```

```
.listaAlumnos li:nth-child(odd){  
    margin-left: @margin-left-primary;  
}
```

9. No se puede realizar, no lo hemos visto en el curso

10. Para calcular la media de las notas se hace uso del siguiente código:

```
foreach($alumnos as $key=>$alumno){  
    $aux[$key]=$alumno;  
    $auxNotas[$key]=$alumno->getMark();  
}
```

Y posteriormente se saca por pantalla con:

```
<?php echo round($sumaNotas/count($alumnos), 2); ?>
```

11.

12. Para esto se crea un condicional que asigne el id alumnoSuspenso si la nota es menor a 5 y el id alumnoAprobado si el alumno está aprobado

```
if($nota<5){  
    ?>  
    <li id="alumnoSuspenso">Nombre: <?php echo $alumno->getFirstname()." ".$alumno->getLastName()." -> ".$nota; ?></li>  
    <?php
```

Y se edita el archivo css para colorear por suspenso y aprobado:

```
#alumnoAprobado{  
    .aprobado;  
}  
  
#alumnoSuspenso{  
    .suspenso;  
}  
  
.aprobado{  
    color:green;  
}  
  
.suspenso{  
    color:red;  
}
```

13. Para destacar los 3 mejores alumnos, se crea un array auxiliar que almacene los alumnos y sus notas, posteriormente con el metodo array\_multisort ordenamos de mayor a menor nota, y a la hora de pintarlo, añadimos una condicion para que los 3 primeros utilice otro color

```
foreach($alumnos as $key=>$alumno){
    $aux[$key]=$alumno;
    $auxNotas[$key]=$alumno->getMark();
}
$cont=0;
array_multisort($auxNotas, SORT_DESC, $aux);
```

```
if($cont<3){
    ?>
```

```
.top3{
    font-weight: bold;
}
```

14. Para ello crearemos el archivo “/sanchez/Console/ExamenCommand.php” de la siguiente forma:

En la función configure añadimos los datos para la ejecución del comando por consola:

```
public function configure(){
    $this->setName('hiberus:sanchez');
    $this->setDescription('Mostrar datos de la tabla');
    parent::configure();
}
```

Y en la execute los dato mediante un bucle que lea los datos:

```
protected function execute(
    InputInterface $input,
    OutputInterface $output
){
    $exámenes=$this->collection->create();

    if(!empty($exámenes)){
        $output->writeln('<info>Alumnos:</info>');
        foreach($exámenes as $examen){
            $id=$examen->getIdExam();
            $firstname=$examen->getFirstname();
            $lastname=$examen->getLastname();
            $mark=$examen->getMark();
            $output->writeln('<info>'.$id.'.'.$firstname.'.'.$lastname.'.'.$mark.'</info>');
        }
    }else{
        $output->writeln('<info>Vacio</info>');
    }
}
```

15. No se puede realizar, no lo hemos visto en el curso

16.

17. No se puede realizar, no lo hemos visto en el curso