

Ej1

Se crea el fichero module.xml en la carpeta Sillah/etc y se introduce el nombre del módulo.
También se crea el fichero registration.php al que también se le incluye el nombre del módulo.
Se ejecuta el comando **dockergento magento setup:upgrade**.
Se accede al fichero config del vendor para comprobar que el módulo está activo y creado.

Ej2

Creo un xml llamado schema, en el xml introduzco los campos que voy a necesitar

```
<?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 prueba 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" scale="2" precision="4" unsigned="true" nullable="false"/>
    <constraint xsi:type="primary" referenceId="PRIMARY">
      <column name="id_exam"/>
    </constraint>
  </table>
</schema>
```

Realizo una actualización del repositorio
dockergento magento setup:upgrade

Compruebo que existe en la base de datos

Query 1 X hiberus_curso X magento.hiberus_exam X								
Info Columns Indexes Triggers Foreign keys Partitions Grants DDL								
Column	Type	Default Value	Nulla	Characte	Collation	Privileges	Extra	Comments
id_exam	int(10) unsigned		NO			select,insert,update,refere	auto_increment	
firstname	varchar(100)		NO	utf8	utf8_general	select,insert,update,refere		
lastname	varchar(250)		NO	utf8	utf8_general	select,insert,update,refere		
mark	decimal(4,2) un		NO			select,insert,update,refere		

Ej3

Se crean las interfaces tanto para el modelo como para el repositorio

Modelo:

```
namespace Hiberus\Sillah\Api\Data;

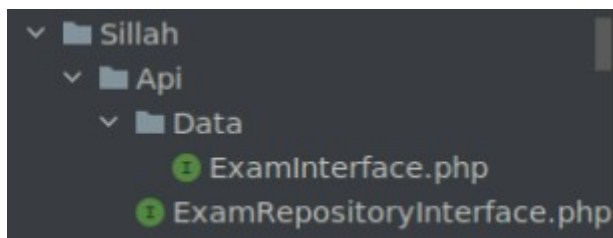
interface ExamInterface extends \Magento\Framework\Api\ExtensibleDataInterface
{
    const TABLE_NAME='hiberus_exam';
    const COLUMN_ID='id_exam';
}
```

Repositorio:

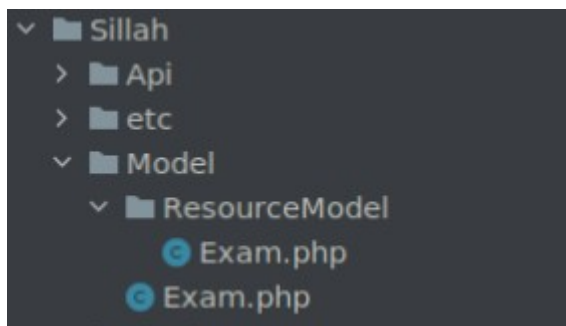
```
interface ExamRepositoryInterface
{
    /**
     * @param \Hiberus\Sillah\Api\Data\ExamInterface $examInterface
     * @return \Hiberus\Sillah\Api\Data\ExamInterface
     */
    public function save(\Hiberus\Sillah\Api\Data\ExamInterface $examInterface);
}
```

Ej4

Se crean las interfaces en la ruta Sillah/Api y Sillah/Api/Data



A continuación el modelo y el repositorio en las carpetas model y resource model

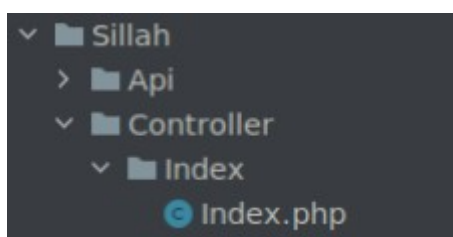


Se adapta la lógica de estos archivos según la base de datos y las interfaces creadas.

Apunte: No adjunto capturas debido al tamaño de los archivos

Ej 5

Se crea la carpeta controller en el módulo, en su interior otra carpeta llamada index con el nombre del controlador, en este caso index



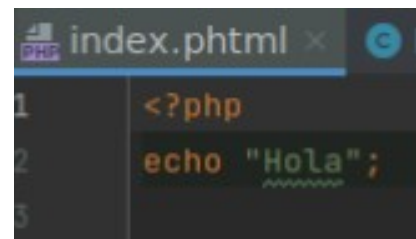
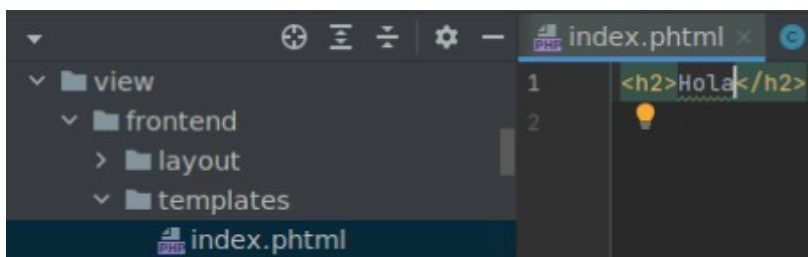
A continuación se crea el constructor y el método execute asociando este método con la clase que creamos anteriormente

```
protected \Magento\Framework\View\Result\PageFactory $pageFactory;
private \Hiberus\Sillah\Model\ExamFactory $examFactory;

public function __construct(\Magento\Framework\App\Action\Context $context,
    $this->pageFactory=$pageFactory;
    $this->examFactory=$examFactory;
}

public function execute()
{
    //return $this->pageFactory->create();
    $exam= $this->examFactory->create();
    $collection=$exam->getCollection();
    foreach ($collection as $item){
        echo $item->getData();
    }
    return $this->pageFactory->create();
}
```

Se crea el template que contendrá el mensaje hola en la carpeta templates dentro de view/frontend/templates



Se crea la clase collection para poder mostrar

Ej6

Se asocian layout, bloque y plantilla

```
<?xml version="1.0"?>

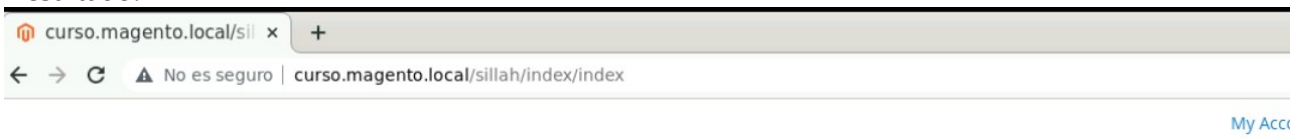
<page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" layout="1column" xsi:noNamespaceSchemaLocation="urn:magento:framework:View/Layout/etc/page_configuration">
    <referenceContainer name="content">
        <block class="Hiberus\Sillah\Block\Index" name="sillah_index_index" template="Hiberus_Sillah::index.phtml"/>
    </referenceContainer>
</page>
```

Se edita la plantilla

```
use Hiberus\Curso\Block\Index;
$exam=$block->getExams();
//var_dump($exam->getData());die;
$exams = $block->getExams();
?>
<h2 class="title">Exam</h2>
<ul>
    <?php foreach ($exam as $item){
        ?>

        <li><?=$item->getFirstname()?></li>
        <li><?=$item->getLastName()?></li>
        <li><?=$item->getMark()?></li>
        <?php } ?>
    </ul>
```

Resultado:



hiberus

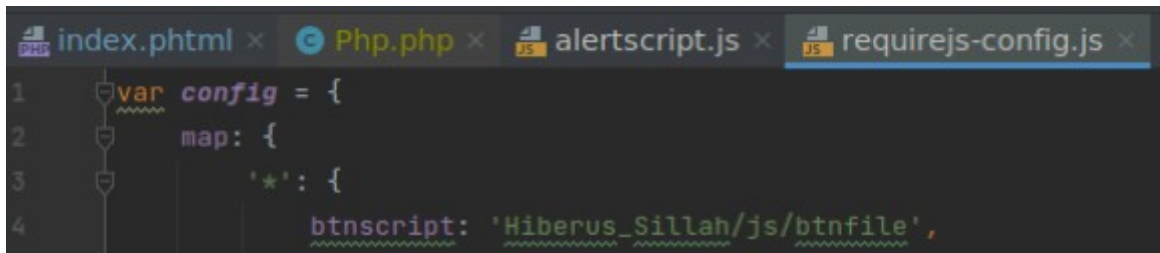
Exam

- carlos
- jimenez
- 7.30
- ivan
- costas
- 4.12

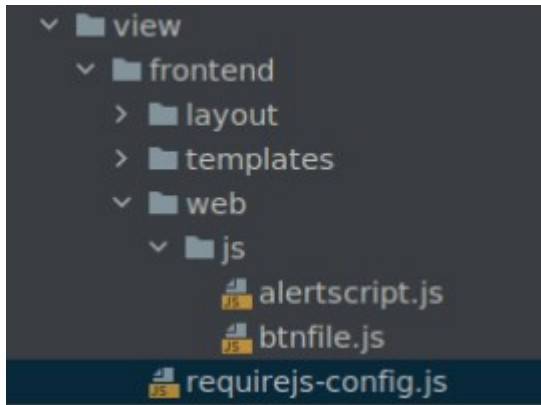
Total number of 2 students

Ej7

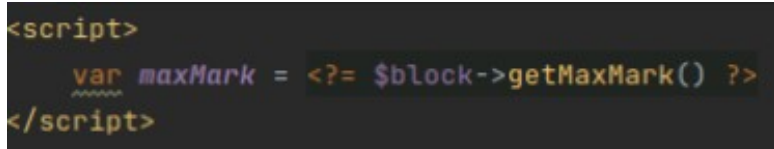
Se crea el fichero js que va a contener la configuración/funciones que se van a utilizar



```
1 var config = {
2   map: {
3     '*': {
4       btnscript: 'Hiberus_Sillan/js/btnfile',
```



Se generan los scripts que ejecutarán las acciones



```
<script>
  var maxMark = <?= $block->getMaxMark() ?>
</script>
```



```
define(['jquery'], function($) {
  "use strict";
  return function btnscript(btnId, ulId)
  {
    $(btnId).click(function() {
      $(ulId).slideToggle();
    });
  }
});
```

Resultado:

Exam

Hide/Show

- carlos
- jimenez
- 7.40
- ivan
- costas
- 4.9

Exam

Hide/Show

Show maximum mark

Total number of 5 students

Average Exam marks 7.552

Ej8

Se crea el método en el fichero mixin.less

```
.exam-mixin(@margin-left-primary){  
    margin-left: @margin-left-primary;  
}
```

Se comprueba que se importa el archivo mixin.less

```
@import "_mixin.less";
```

Se llama al método en el extends.less de magento-theme creando antes la variable

```
@margin-left-primary:20px;  
& when (@media-common = true) {  
    .page-footer{  
        display: none;  
    }  
  
    .title{  
        color: darkblue;  
    }  
  
    ul:nth-child(even) {  
        .exam-mixin(@margin-left-primary);  
    }  
}
```

Resultado:

hiberus

Exam

- carlos
- jimenez
- 7.30
- ivan
- costas
- 4.12

Total number of 2 students

Ej9

Se declara el fichero y el método en el fichero de configuración de js

```
alertscript: 'Hiberus_Sillah/js/alertscript',
```

También se llama al script en la plantilla

```
<script>
  require(['jquery', 'alertscript'], function($, alertscript) {
    alertscript('#maxmark');
  });
</script>
```

```
<script>
  var maxMark = <?= $block->getMaxMark() ?>
</script>
```

Se genera la función que muestra la nota

```
define(['jquery'], function($) {
  "use strict";
  return function alertscript(btnId)
  {
    $(btnId).click(function() {
      alert('Maximum mark: '+maxMark);
    });
  }
});
```


Resultado:

hiberus

curso.magento.local dice
Maximum mark: 9.15

Aceptar

Exam

Hide/Show

Show maximum mark

Total number of 5 students

Average Exam marks 7.552

Ej10

Se crea un método capaz de obtener todas las notas y realizar la media de notas en Block/Index

```
public function getAverageMarks(){  
    $total=$this->getExams();  
    $marks=[];  
    foreach ($total as $item){  
        $marks[]=$item->getMark();  
    }  
    $averageMark=array_sum($marks)/count($marks);  
    return $averageMark;  
}
```

Se obtiene el resultado en la plantilla y se muestra en el navegador

```
<p><?= __('Average Exam marks %1', [$block->getAverageMarks()]) ?></p>
```

Exam

- carlos
- jimenez
- 7.30

- ivan
- costas
- 4.12

Total number of 2 students

Average Exam marks 5.71

Ej 11

Se agrega en el di.xml el plugin

```
<!-- Plugin de notas suspensas -->

<type name="Hiberus\Sillah\Model\Exam">
  <plugin name="exam_plugin" type="Hiberus\Sillah\Plugin\Mark\MarksPlugin" sortOrder="10" />
</type>
```

Se crean las carpetas plugin y mark, también la clase MarksPlugin

```
use Hiberus\Sillah\Model\Exam;

class MarksPlugin
{
    public function afterGetMark(Exam $subject, $result)
    {
        if ($subject->getData( key: 'mark') < 5.0) {
            $subject->setMark( mark: 4.9);
            $result = $subject->getData( key: 'mark');
            //var_dump($subject->getData());die;
        }
        return $result;
    }
}
```

Resultado:

Exam

- carlos
- jimenez
- 7.30
- ivan
- costas
- 4.9

Total number of 2 students

Average Exam marks 6.1

Ej 12

Se crea una condición que nos permita distinguir entre aprobados y suspensos

```
if($item->getMark() > 4.9){  
    $evaluation="aprobado";  
}else{  
    $evaluation="suspense";  
}  
?>  
  
<li class="<?=$evaluation?>"><?=$item->getMark()?></li>
```

A continuación se crea un método que configure el color de las notas en el mixin

```
.aprobado{  
    .exam-mixin2(green);  
}  
  
.suspense{  
    .exam-mixin2(red);  
}
```

Resultado:

Exam

- carlos
- jimenez
- 7.30
- ivan
- costas
- 4.9

Total number of 2 students

Average Exam marks 6.1

Ej13

Se genera un método capaz de obtener las tres mejores notas

```
public function getMaxMarks(){
    $total=$this->getExams();
    $marks=[];
    $maxMarks=[];
    foreach ($total as $item){
        $marks[]=$item->getMark();
    }
    rsort( &array: $marks);
    foreach ($marks as $mark){
        if(count($maxMarks)<3){
            $maxMarks[]=$mark;
        }
    }
    return $maxMarks;
}
```

En la plantilla se obtienen dichos valores para poder representar dichas notas

```
if(in_array($item->getMark(),$maxMarks)){
    ?>
    <li class="<?=$evaluation?>"><strong><?=$item->getMark()?></strong></li>
    <?php
}else{
    ?>
    <li class="<?=$evaluation?>"><?=$item->getMark()?></li>
    <?php
}
```

Resultado:

Exam

Hide/Show

- carlos
- Jimenez
- 7.40
- Ivan
- costas
- 4.9
- ismael
- fernandes
- 8.87
- claudio
- pellegrini
- 7.44
- leon
- kingston
- 9.15

Ej 14

Se crea la carpeta Console y la clase que contendrá al comando

```
<?php

namespace Hiberus\Sillah\Console;

use Hiberus\Sillah\Block\Index;
use Hiberus\Sillah\Model\Exam;
use Symfony\Component\Console\Command\Command;
use Symfony\Component\Console\Input\InputInterface;
use Symfony\Component\Console\Output\OutputInterface;
use Symfony\Component\Console\Input\InputOption;

class MarksCommand extends Command
{
    /**
     * @var Exam
     */
    protected Exam $exam;
}
```

Se listan los objetos de la base de datos

```
$exams = $this->block->getExams();  
foreach ($exams as $item){  
    $this->exam->setIdExam($item->getIdExam());  
    $this->exam->setFirstname($item->getFirstname());  
    $this->exam->setLastname($item->getLastname());  
    $this->exam->setMark($item->getMark());  
    $output->writeln( messages: '<info> ID: ' . $this->exam-
```

Se ejecuta el comando:

```
pue@pue-KVM:~/proyectos/curso-magento$ dockergento magento hiberus:sillah  
ID: 1 | Nombre: carlos | Apellidos: jimenez | Nota: 7.3  
ID: 2 | Nombre: ivan | Apellidos: costas | Nota: 4.9
```

Ej 16

Se crean los campos que contendrá el admin en el fichero system.xml

```
</tab>  
<section id="hiberus_sillah" showInDefault="1" sortOrder="10">  
    <label>Sillah Attributes</label>  
    <tab>sillah</tab>  
    <resource>Hiberus_Sillah::config</resource>  
    <group id="general" showInDefault="1" sortOrder="10">  
        <label>General</label>  
        <field id="element_general" type="text" showInDefault="1" sortOrder="10">  
            <label>Elementos</label>  
            <validate>integer</validate>  
        </field>  
        <field id="number_general" type="text" showInDefault="1" sortOrder="10">  
            <validate>validate-number</validate>  
            <label>Calificacion minima General</label>  
        </field>  
    </group>  
</section>
```

Se guardan los datos a configurar

Elementos
[global]

Calificacion minima General
[global]

Se obtienen los datos configurados

```
public function getDefaultMark() {
    $defaultMark= $this->scopeConfig->getValue( 'hiberus_sillah/general/number_general', ScopeInterface::SCOPE_STORE);
    if(is_null($defaultMark)||$defaultMark<=0){
        $defaultMark=5;
    }
    return $defaultMark;
}

public function getElements() {
    $elements= $this->scopeConfig->getValue( 'hiberus_sillah/general/element_general', ScopeInterface::SCOPE_STORE);
    return $elements;
}
```

Adaptamos la vista conforme a los datos obtenidos

```
$defaultMark=$block->getDefaultMark();
$elements=$block->getElements();
$iterable=0;
?>
<h2 class="title">Exam</h2>
    <?php
        foreach ($exam as $item){
            if ($iterable++ == $elements && $elements>0){
                break;
            }
        }
    ?>
<ul>
    <li><?=$item->getFirstname()?></li>
    <li><?=$item->getLastname()?></li>
    <?php
        if($item->getMark() >= $defaultMark){
            $evaluation="aprobado";
        }else{
            $evaluation="suspense";
        }
    ?>
```

Resultado:

Exam

- carlos
- jimenez
- 7.30

Total number of 2 students

Average Exam marks 6.1

Ejercicios 15 y 17:

Apunte: No se han visto