Autor: Francisco Jesús García López

1 - Nuevo módulo

Para crear el módulo creo el directorio y los archivos registration.php y etc/module.xml.

2, 3, 4 - Creación de la tabla en la base de datos y rellenarla con datos.

Se ha creado el archivo etc/db_schema.xml para definir la tabla de la base de datos, las interfaces en el directorio Api y los models y resource models en Model. Para rellenar la base de datos se usa el archivo Setup/Patch/Data/PopulateDataModel.php, se ha hecho con csv. Y se ha creado archivo etc/di.xml.

Setup:upgrade para crear la tabla en la bd y use el data patch para meterle los datos.

5 - Creación del controlador

Creado el controlador en Controller/Garciacontroller/Index.php. Creado archivo etc/frontend/routes.xml para establecer la ruta.

6 - Asociar layout, template y bloque.

Creados en el directorio view/frontend los archivos layout/garcia_garciacontroller_index.xml para establecer el layout y señalar al bloque creado.

templates/examlist.phtml que es la plantilla del bloque donde ponemos la estructura del contenido (titulo, el listado de exámenes, el total de alumnos, etc).

Block/GarciaBlock.php definimos la clase de nuestro bloque y los métodos necesarios para sacar los datos de la bd.

Ponemos __(texto: %1,[1]) para luego traducir en i18n/es_ES.

7 - Js para ocultar y mostrar las notas

Creo el js en el directorio view/frontend/web/js y lo uso con notación declarativa en el bloque. En el js uso is(':visible') para ver si esta visible, hide() para esconderlo y show() para mostrar.

8 - Usando Less

En el directorio view/frontend/web/css/source creo el _module.less. Uso media-widht con screen_m para 768 pixeles para cambiar de color al titulo. Para margin-left intermitente uso nth-child(odd) y aplico algunos estilos más para dejarlo con la mejor apariencia que puedo sin perder mucho tiempo.

9 - Botón para mostrar alerta con máxima nota

Nuevo archivo js llamado showmaxmark-garcia.js que uso también con notación declarativa. Uso la función alert de js. Uso un parámetro en la declaración del script para pasarle la máxima nota (que es una función creada en el bloque).

10 - Nueva fila para nota media

Nueva función en la clase del bloque y pongo un div en el que llamo a la función para obtener la nota media.

Autor: Francisco Jesús García López

11 - Plugin para modificar la lista

Creo el plugin en el directorio Plugin y lo declaro en el di.xml.

En el plugin uso un afterGetList, para modificar el resultado de la lista.

12 - Alumnos aprobados y suspensos con color diferente

No he sabido hacerlo usando LESS. Se ha hecho una función en la clase del bloque para obtener el color a mostrar y usarlo en un style directamente en el phtml, llamando a dicha función para saber si está suspenso o no. Pongo "Mark: nota" en rojo si es suspenso o en verde si es aprobado.

13 - Tres mejores de manera destacada

Para destacar a los 3 mejores, uso un sortOrderBuilder para obtener los exámenes en orden de nota descendente y además les he puesto un borde negro desde css a los 3 primeros elementos de la lista con nth-child(-n+3).

14 - Comando para mostrar la lista de exámenes

Creo los archivos necesarios para el comando en el directorio Console/Command. En el archivo ShowExamsCommands.php se configura todo el comando, para determinar lo que tiene que mostrar, en la función process. Y creo una función nueva getExamList para recoger los datos de la base de datos con el repository.

Se hace también en Input/ShowExams/ListInputValidator.php y Options/ShowExams/ListOptions.php para que, si hubiera opciones, mostrarlas.

15 - Api Rest con Swagger

En etc/webapi.xml defino las 3 rutas desde las que se verán los exámenes, borrará y crearán exámenes. Estos dos últimos con POST, para pasarles un json con los datos necesarios.

16 - Sección de configuración en el admin

Creo los archivos menú.xml y system.xml en el directorio etc/adminhtml. En Menú determino el tab del menú izquierdo del admin de magento y la parte de la configuración. En system creo el grupo y los fields que se necesitan (2 fields de tipo texto). En etc/config.xml establezco los valores por defecto.

Creo un Helper para obtener los valores de la configuración, que inyecto como dependencia en la clase del bloque para determinar el número de exámenes que se muestran (si es 0 se muestran todos) y la nota de corte de los aprobados.

17 - Log personalizado sin usar una clase

En el di.xml creo dos virtual types para definir el archivo de log y la forma de loggear, y le inyecto como dependencia ese virtual type a la clase que lo va a usar, en este caso, la clase del bloque (GarciaBlock.php).

Desde la clase del bloque, inyecto como dependencia la clase Psr\Log\LoggerInterface, y uso su función debug para crear logs (uso debug porque le he puesto ese tipo en el di.xml).