

Tarea para DWES05.

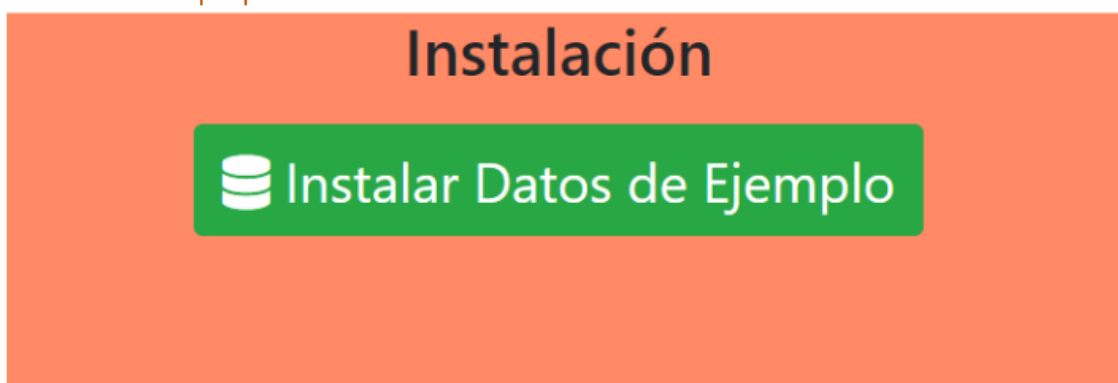
Descripción de la tarea.

Enunciado

El objetivo es desarrollar una aplicación web que permita la gestión de inversiones en fondos indexados en la bolsa. Para ello, trabajaremos con la tabla inversiones, la cual formará parte de una nueva base de datos. Podemos reutilizar el usuario gestor de ejercicios anteriores con los permisos requeridos en la base de datos.

¿Qué te pedimos que hagas?

instalacion.php



inversiones.php

Listado de Inversiones				
+ Nuevo Fondo				
Nombre del Fondo	Categoría	Valor de la Inversión	Fecha de Inversión	Rentabilidad Esperada
Nicolas, Franecki and Block Index Fund	Renta Variable	63851.24	2018-08-20	9.07%
Schamberger PLC Index Fund	Renta Fija	64600.76	2017-10-01	6.73%
Lang-Hansen Index Fund	Renta Fija	96288.81	2005-10-28	12.90%
Langosh, Bauch and Doyle Index Fund	Emergentes	40577.54	1996-09-17	10.64%
Shanahan-Oberbrunner Index Fund	Emergentes	9763.07	1995-02-04	11.13%
Graham-Kessler Index Fund	Internacional	14521.62	1991-10-16	1.94%
Jacobson Inc Index Fund	Internacional	32371.38	1986-07-19	5.66%

fcrear.php



Información de interés.

Recursos necesarios

La estructura de la tabla inversiones y los scripts necesarios para la creación de la base de datos y los permisos serán facilitados a través de un archivo `tabla_inversiones.sql`.

En la tabla inversiones guardaremos la siguiente información para cada inversión:

- Nombre del fondo indexado
- Categoría (por ejemplo, Renta Fija, Renta Variable, Global, etc.)
- Valor de la inversión
- Fecha de la inversión
- Rentabilidad esperada
- Tasa de retorno anual (opcional)

Dependencias obligatorias a instalar con Composer:

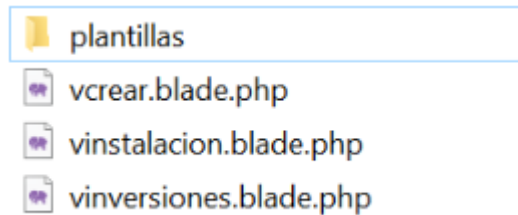
- `philo/laravel-blade` como motor de plantillas.
- `fakerphp/faker` para generar datos aleatorios de prueba.
- `autoload` con optimizador.

Consejos y recomendaciones

Estructura sugerida del proyecto:

1. `cache`: Carpeta necesaria para el funcionamiento de Blade, con permisos "777" en Linux.
2. `public`: Aquí se almacenan las páginas que se muestran en el navegador.
 - `crearDatos.php`: Página que genera datos de ejemplo para las inversiones.
 - `crearInversion.php`: Página "action" del formulario para crear una nueva inversión, con validaciones para evitar duplicados y campos vacíos.
 - `fcrear.php`: Formulario para crear la inversión. Usará la vista `vcrear.blade.php`.
 - `index.php`: Página de inicio que verifica si la tabla inversiones tiene datos y, de no ser así, redirige a `instalacion.php` para generar datos de ejemplo.
 - `instalacion.php`: Carga la vista `vinstalacion.php` que contiene un botón para generar datos de ejemplo.
 - `inversiones.php`: Muestra una lista de las inversiones realizadas en una tabla.
3. `sql`: Contendrá el archivo `.sql`.
4. `src`: Contiene las clases para gestionar la base de datos:
 - `Conexion.php`: Establece la conexión con la base de datos.

- Inversion.php: Gestiona la tabla inversiones, con los métodos para devolver las inversiones, insertar nuevas inversiones, etc.
5. vendor: Generada por Composer.
 6. views: Contiene todas las vistas y plantillas utilizadas.



- Dentro de la carpeta plantilla tenemos la plantilla de nivel superior que todas las demás incluyen:



El archivo composer.json estará presente con las dependencias del proyecto

Indicaciones de entrega

Una vez realizada la tarea elaborarás un único documento donde figuren las respuestas correspondientes. El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

apellido1_apellido2_nombre_SIGxx_Tarea

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna Begoña Sánchez Mañas para la cuarta unidad del MP de DWES, debería nombrar esta tarea como...

sanchez_manas_begona_DWES04_Tarea

Evaluación de la tarea.

Criterios de evaluación implicados

- Se han identificado las ventajas de separar la lógica de negocio de los aspectos de presentación de la aplicación.
- Se han analizado tecnologías y mecanismos que permiten realizar esta separación y sus características principales.
- Se han utilizado objetos y controles en el servidor para generar el aspecto visual de la aplicación Web en el cliente.
- Se han utilizado formularios generados de forma dinámica para responder a los eventos de la aplicación Web.
- Se han identificado y aplicado los parámetros relativos a la configuración de la aplicación Web.

- Se han escrito aplicaciones Web con mantenimiento de estado y separación de la lógica de negocio.
- Se han aplicado los principios de la programación orientada a objetos.
- Se ha probado y documentado el código.

¿Cómo valoramos y puntuamos tu tarea?

Crea la estructura correcta del proyecto y se instalan correctamente las dependencias con Composer.	2.5 puntos.
Se crean correctamente, con los métodos necesarios, las clases "Conexion" y "Inversión" y se hace uso de "namespaces" y herencia.	2 puntos.
Crea correctamente los datos de ejemplo con "phpfaker/faker".	1.25 puntos.
Las páginas de la carpeta "public" cargan las vistas apropiadamente y les pasan los parámetros necesarios.	1.25 puntos.
El formulario para crear un fondo funciona correctamente, controlando posibles errores.	1 punto.
Las vistas hacen uso de una plantilla común.	0.5 puntos.
La página de inicio comprueba si existen datos o no en la tabla "inversiones" y hace las redirecciones adecuadas.	0.5 puntos.
Se hace uso correcto de las sesiones para mostrar mensajes.	0.5 puntos.
El diseño es adecuado y el código está comentado.	0.5 puntos.