



ESCUELA POLITÉCNICA NACIONAL
INGENIERÍA DE SOFTWARE
CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

Integrantes: De La Cruz Luis, Moyano Sebastián, Zambrano Christopher

Fecha: 10/01/2024

Proyecto Primer Bimestre

1. Preparación

El proyecto consiste en desarrollar un producto software que permita llevar un registro de acciones en donde se ingresen los campos de nombre, fecha, precio de compra y cantidad de acciones. Dados todos los campos se agrega un último campo donde se calcula el costo total de la compra el cual es el producto del precio de compra por la cantidad de acciones. A continuación, se presenta una imagen de cómo se vería este registro de acciones con sus respectivos campos.

NOMBRE DE LA ACCIÓN	FECHA DE COMPRA	PRECIO DE COMPRA POR ACCIÓN	CANTIDAD DE ACCIONES	COSTO TOTAL DE COMPRA
AAPL	10/1/2023	150	10	1500
MSFT	15/2/2023	250	5	1250
TSLA	20/3/2023	700	2	1400

2. Documentación de requisitos

Epica: Como usuario, quiero llevar un registro de mis acciones para gestionar mis finanzas.

Numero	1
Título	Ver acciones
Descripción	Como usuario, Quiero ver mis acciones con los datos de nombre, fecha de compra, precio de compra por acción y cantidad de acciones Para tener un registro claro y accesible de mis transacciones financieras.
Criterios de aceptación	<ul style="list-style-type: none">- Dado que se tiene una lista vacía, cuando el usuario ingresa a la aplicación, entonces se muestra un mensaje indicando que no existen registros.- Dado que se tienen registros, cuando el usuario ingresa a la aplicación se mostrará una tabla con todas las acciones y sus campos respectivos.
Tareas	T1: Configurar la conexión a la base de datos T2: Hacer la pantalla principal donde se muestran los registros de las acciones T3: Leer los registros de las acciones en la base de datos y mostrarlos en la pantalla T4: Agregar el mensaje informativo que aparece cuando no existen registros en la base de datos

Numero	2
Título	Agregar acciones
Descripción	Como usuario, Deseo la capacidad de agregar nuevas acciones con todos sus campos correspondientes en la aplicación. Para realizar un seguimiento detallado de mis adquisiciones financieras.
Criterios de aceptación	<ul style="list-style-type: none">- Dado que se ingresa la fecha de compra, esta debe permitir la selección de una fecha válida y presentarse en un formato fácil de entender.- Al completar el formulario y clicar en el botón de agregar, la nueva acción deberá registrarse en la lista de acciones existente.
Tareas	T1: Creación de un botón con la leyenda “Agregar compra” con enlace a una nueva página para agregar una compra y estilo en css. T2: Creación de una página con un formulario con los parámetros nombre de acción, fecha, precio por acción y cantidad de compra. T3: Implementación de query de creación de registro en la base de datos utilizando a los parámetros del formulario.

Numero	3
Título	Costo total de compra
Descripción	Como usuario, Deseo la capacidad de visualizar el costo total de compra de mis acciones en función del precio de compra por acción y la cantidad de acciones adquiridas, Para conocer la cantidad total que debo pagar por las acciones que he adquirido.
Criterios de aceptación	<ul style="list-style-type: none"> - Dado que ingreso a la aplicación, cuando visualizo el inicio, entonces este deberá mostrar automáticamente el costo total de compra de mis acciones. - Dado que agrego nuevas acciones entonces el costo total deberá actualizarse instantáneamente.
Tareas	T1: Calcular el costo total de compra cada vez que se agrega una nueva acción. T2: Leer y mostrar el costo total de compra en la tabla general de acciones.

Numero	4
Título	Modificar acciones
Descripción	Como usuario, Deseo poder cambiar los detalles de una acción previamente creada Para realizar cambios o correcciones en los campos según lo necesite.
Criterios de aceptación	<ul style="list-style-type: none"> - Dado un registro existente en la tabla, al clicar el botón modificar nos mostrara un formulario donde se realizarán los campos, se guardarán y el cambio se verá reflejado en la tabla.
Tareas	T1: Crear una nueva columna a la tabla general donde se muestre un botón para actualizar. T2: Crear una nueva pantalla con un formulario para actualizar el registro. T3: Crear la función para que en la base de datos permita actualizar un registro dado un id T4: Crear la función para obtener el id del registro cuando se de click al botón y llenar el formulario con los datos del registro T5: Crear la función para que cuando se de click en enviar el formulario se actualice en la base de datos y se muestre en la lista de registros.

Numero	5
Título	Eliminar acciones
Descripción	Como usuario, Deseo eliminar los registros de acciones previamente creadas Para deshacerme de registros que considere inútiles o incorrectos de la tabla de acciones.
Criterios de aceptación	<ul style="list-style-type: none"> - Dado un registro mostrado en la tabla, al clicar el botón “eliminar” el registro ya no aparecerá en los registros de la tabla.
Tareas	T1: Creación del botón eliminar dentro de la tabla. T2: Dar estilo al botón eliminar mediante CSS.

3. Documentación de pruebas

Pruebas nomenclatura unitarias

Para las pruebas unitarias se utiliza PHPUnit, se seguirá la siguiente notación:

Dado<acción>**Entonces**<AccionEsperada>

Definición de pruebas

En el caso de esta iteración se desarrollaron las siguientes pruebas:

- DadoAgregarUnaAccionEntoncesEstaAccionSeRegistra
- DadoUnIDCuandoSeEliminaEntoncesEsteRegistroSeElimina
- DadoAccionConUnIDyCamposcambiadosEntoncesEstaSeActualiza
- DadoCalcularTotalEntoncesDevuelveElvalor
- DadoBuscarUnaAccionPorIDEntoncesDevuelveLaAccion
- DadoListarAccionesEntoncesDevuelveUnaListaNoNula

Pruebas

En esta primera prueba se crea un nuevo objeto y se verifica que se allá guardado dentro del registro.

```
public function DadoAgregarUnaAccionEntoncesEstaAccionSeRegistra()
{
    $accion = new Accion(['nombre' => 'Prueba', 'fecha' => '2024-01-11', 'precio' => 10, 'cantidad' => 5]);
    $accion->crear();
    $this->assertNotNull(Accion::buscar("44"));
}
```

En la segunda prueba verificamos si al recibir un ID existente este encuentra algún valor.

```
public function DadoBuscarUnaAccionPorIDEntoncesDevuelveLaAccion()
{
    $accion = Accion::buscar("44");
    $this->assertNotNull($accion);
}
```

En esta prueba se crea un nuevo objeto tipo Acción para poder cambiar los datos mediante el ID, una vez actualizados los campos, se procede a verificar el registro.

```
public function DadoAccionConUnIDyCamposcambiadosEntoncesEstaSeActualiza()
{
    $accion = new Accion(['nombre' => 'Prueba_act', 'fecha' => '2024-02-11', 'precio' => 30, 'cantidad' => 5]);
    $accion->setID("44");
    $accion->actualizar();
    $accionActualizada = Accion::buscar($accion->getId());
    $this->assertEquals('Prueba_act', $accionActualizada->getNombre());
    $this->assertEquals(30, $accionActualizada->getPrecio());
}
```

En la cuarta prueba se asigna un id, para eliminar el objeto y se verifica que, una vez eliminado este objeto, no se guarde un registro, es decir que verificamos que al buscar sea nulo.

```
public function DadoUnIDCuandoSeEliminaEntoncesEsteRegistroSeElimina()
{
    $accion = new Accion(['nombre' => 'Prueba', 'fecha' => '2024-01-11', 'precio' => 10, 'cantidad' => 5]);
    $accion->setID("44");
    $accion->eliminar();
    $accionEliminada = Accion::buscar("44");
    $this->assertNull($accionEliminada);
}
```

En la quinta prueba se verifica que el método obtenga una lista de objetos de acciones para ser mostrados.

```
public function DadoListarAccionesEntoncesDevuelveUnaListaNoNula()
{
    $acciones = Accion::listar();
    $this->assertNotEmpty($acciones);
}
```

En la última prueba verificamos que el costo total sea correcto.

```
public function DadoCalcularTotalEntoncesDevuelveElvalor()
{
    $accion = new Accion(['precio' => 10, 'cantidad' => 5]);
    $this->assertEquals(50, $accion->getCostoTotal());
}
```

Ejecución de pruebas

Para la ejecución de pruebas se requiere la instalación de PHPUnit con el servidor XAMPP.

```
C:\xampp\htdocs\ProyectoCESw_1B\Test>phpunit --bootstrap Accion.php AccionTest.php
PHP Warning: The use statement with non-compound name 'Accion' has no effect in C:\xampp\htdocs\ProyectoCESw_1B\Test\AccionTest.php on line 3

Warning: The use statement with non-compound name 'Accion' has no effect in C:\xampp\htdocs\ProyectoCESw_1B\Test\AccionTest.php on line 3
PHPUnit 10.5.5 by Sebastian Bergmann and contributors.

Runtime:       PHP 8.2.4

WW..                                         6 / 6 (100%)

Time: 00:01.623, Memory: 24.00 MB

OK, but there were issues!
Tests: 6, Assertions: 7, Warnings: 3.
C:\xampp\htdocs\ProyectoCESw_1B\Test>
```

4. Desarrollo del software

Para el desarrollo del software se utilizó un enfoque ágil debido a la complicación del tiempo.

En este caso se utilizó programación extrema en conjunto de TDD.

El desarrollo se dividió en 3 iteraciones:

- Definición de pruebas
- Primera iteración correspondientes a las historias 1, 2 y 3.
- Segunda iteración implementación de las ultimas historias de usuario.

Las tecnologías utilizadas en este proyecto fueron las siguientes:

- MySQL base de datos
- PHP lenguaje programación centrado en diseño de web y servidor
- CSS para diseño llamativo
- PHPUnit para desarrollo y ejecución de pruebas.

5. Configuraciones

Configuración de la base de datos

Para la configuración de la base, se utiliza variables de entornos reservadas, ya que la base de datos se encuentra en la nube, y utiliza el gestor de bases de datos MySQL.

Se tiene únicamente una tabla para almacenar las acciones con sus campos respectivos.

- ❖ Campos de la tabla acción
 - Nombre
 - Fecha
 - Precio
 - Cantidad
 - Costo Total

Para conexión se requiere las siguientes variables de entorno:

\$servername

\$username

\$password

\$dbname

Estructura del código

En el proyecto actual se utiliza la siguiente estructura de código, este se divide en tres carpetas principales y raíz.

Raíz: archivos .php los cuales funcionaran como páginas de visualización.

Clases: archivos relacionados a la lógica del programa.

Estilos: archivos .css permiten dar estética a la página web.

Js: Códigos javascript para mensajes emergentes o dinamismo.

Test: archivos .php que permiten probar las pruebas unitarias relacionadas a las clases que aportan funcionalidad.

Implementación local

Para esto es necesario, instalar XAMPP, para tener el servidor local, luego clonar el repositorio del proyecto dentro de la ruta de **XAMAPP/Htdocs**, una vez clonado el repositorio, abrir la aplicación de XAMPP para habilitar el puerto, una vez habilitado el puerto se puede acceder al aplicativo web de forma local.

Manual de usuario

El aplicativo comienza con la página principal es decir con el index, donde podemos observar los datos guardados.

NOMBRE DE ACCION	FECHA DE COMPRA	PRECIO DE COMPRA POR ACCION	CANTIDAD DE ACCIONES	COSTO TOTAL DE COMPRA	ACCIONES
Ejemplo1	2024-01-11	10.00	5	50.00	<button>Eliminar</button> <button>Actualizar</button>
Ejemplo1	2024-01-11	10.00	5	50.00	<button>Eliminar</button> <button>Actualizar</button>
Ejemplo1	2024-01-11	10.00	5	50.00	<button>Eliminar</button> <button>Actualizar</button>
Ejemplo3	2024-01-12	20.00	5	100.00	<button>Eliminar</button> <button>Actualizar</button>

Como se puede observar se tiene un botón con el signo “+”, al presionarlo nos llevara a la pagina de registro de una nueva acción.

Proyecto 1

Nueva compra

Nombre de Accion:
Ingrese nombre...

Fecha:
dd/mm/aaaa

Precio por Accion:
Ingrese el precio por accion

Cantidad de Acciones:
Ingrese la cantidad de acciones

Agregar Compra
Cancelar

En esta pantalla se nos permite rellenar un formulario para agregar un nuevo registro, datos solicitados son los siguientes:

- Nombre de Acción: tipo texto
- Fecha: Tipo date “fecha”, esta se puede colocar con ayuda del calendario que aparece al lado derecho.
- Precio por acción: tipo numérico.
- Cantidad: tipo entero.

En cualquiera de los dos casos sea agregar compra o cancelar, esto nos llevara de nuevo al index.

Nueva compra

Nombre de Acción:
Manual

Fecha:
11/01/2024

Precio por Acción:
15.6

Cantidad de Acciones:
4

[Agregar Compra](#)

[Cancelar](#)

Si agregamos una compra, al llevarnos al index la tabla se actualizará.

Acciones

+
Agregar Compra

NOMBRE DE ACCION	FECHA DE COMPRA	PRECIO DE COMPRA POR ACCION	CANTIDAD DE ACCIONES	COSTO TOTAL DE COMPRA	ACCIONES
Ejemplo1	2024-01-11	10.00	5	50.00	Eliminar Actualizar
Ejemplo1	2024-01-11	10.00	5	50.00	Eliminar Actualizar
Ejemplo1	2024-01-11	10.00	5	50.00	Eliminar Actualizar
Ejemplo3	2024-01-12	20.00	5	100.00	Eliminar Actualizar
Manual	2024-01-11	15.60	4	62.40	Eliminar Actualizar

« 1 2 3 4 »

Si existe algún problema al registrar la acción, esta puede ser eliminada o actualizada, esto con los botones de cada fila.

Al hacer clic en actualizar vemos que se nos abre una pestaña similar al agregar, pero con los datos casi completados, aquí podemos cambiar los datos de este registro.

Actualizar

Nombre de Accion:

Fecha:

Precio por Accion:

Cantidad de Acciones:

[Agregar Compra](#)
[Cancelar](#)

Actualizar

Nombre de Accion:

Fecha:

Precio por Accion:

Cantidad de Acciones:

[Agregar Compra](#)
[Cancelar](#)

Como se observa en la imagen anterior se cambiamos el precio y la cantidad, una vez puesto agregar compra o cancelar, esto nos llevara al index, si en el caso que se haga clic en cancelar la modificación no se realizara.

Acciones					
<div>+</div> <div>Agregar Compra</div>					
NOMBRE DE ACCION	FECHA DE COMPRA	PRECIO DE COMPRA POR ACCION	CANTIDAD DE ACCIONES	COSTO TOTAL DE COMPRA	ACCIONES
Ejemplo1	2024-01-11	10.00	5	50.00	<div>Eliminar</div> <div>Actualizar</div>
Ejemplo1	2024-01-11	10.00	5	50.00	<div>Eliminar</div> <div>Actualizar</div>
Ejemplo1	2024-01-11	10.00	5	50.00	<div>Eliminar</div> <div>Actualizar</div>
Ejemplo3	2024-01-12	20.00	5	100.00	<div>Eliminar</div> <div>Actualizar</div>
Manual	2024-01-11	20.00	5	100.00	<div>Eliminar</div> <div>Actualizar</div>
				« 1 2 3 4 »	

Para eliminar una acción, presionamos el botón eliminar del elemento que deseamos, una vez hecho clic este se eliminara.

Acciones					
<div>+</div> <div>Agregar Compra</div>					
NOMBRE DE ACCION	FECHA DE COMPRA	PRECIO DE COMPRA POR ACCION	CANTIDAD DE ACCIONES	COSTO TOTAL DE COMPRA	ACCIONES
Ejemplo1	2024-01-11	10.00	5	50.00	<div>Eliminar</div> <div>Actualizar</div>
Ejemplo1	2024-01-11	10.00	5	50.00	<div>Eliminar</div> <div>Actualizar</div>
Ejemplo3	2024-01-12	20.00	5	100.00	<div>Eliminar</div> <div>Actualizar</div>
				« 1 2 3 4 »	

Código Fuente

El código fuente de la aplicación web se puede acceder mediante el siguiente link:

https://github.com/WSebastianML/ProyectoCESw_1B