

# Escuela Politécnica Nacional' Facultad de Ingeniería en Sistemas

ASIGNATURA: Construcción y evolución de software GRUPO: GR5

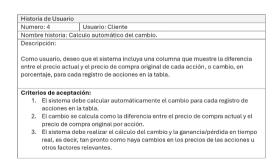
## **INTEGRANTES:**

- Daniel Mera
- George Quishpe
- Christian Agila

## Análisis de cambio:

### 1. Planificación:

 a. Historias de usuario.
 Se agrego dos historias de usuario en las cuales se representa la necesidad de las funcionalidad de "cambio" y "ganancia/perdida".



Histori	a de Usuario	
Nume	ro: 5	Usuario: Cliente
Nomb	re historia: Cal	culo automático del cambio.
Descri	pción:	
entre e		, que el sistema incluya columnas que muestren la diferencia de las acciones y el costo total original de la transacción, es decir h.
Criteri	ios de aceptac	sión:
1.		be calcular automáticamente la ganancia o pérdida para cada ciones en la tabla.
2.		e calcula como la diferencia entre el valor actual de las acciones y original de la transacción.
3.		calcula como la diferencia entre el costo total original de la el valor actual de las acciones.

#### 2. Desarrollo:

#### a. Codificación:

Se agregaron dos métodos getCambio() y setCambio(\$cambio). Estos métodos permiten obtener y establecer el valor de la propiedad cambio.

Se agregaron dos métodos getGananciaPerdida() y setGananciaPerdida(\$gananciaPerdida). Estos métodos permiten obtener y establecer el valor de la propiedad gananciaPerdida.

Se agregó un método privado calcularCambio() que calcula el cambio porcentual en función de la ganancia o pérdida y el costo total. Este método devuelve el cambio porcentual.

Se agregó un método privado getPrecioMercadoFromAPI(\$nombre) que obtiene el precio de mercado de un activo utilizando una API externa. Este método toma el nombre del activo como parámetro, llama a la API de Finnhub con el símbolo del activo y devuelve el precio actual del mercado. Si hay un error en la solicitud o la respuesta no contiene el precio ('c'), devuelve -1.

```
return $this->cambio;
public function setCamcio($cambio) {
   $this->cambio = $cambio;
public function getGananciaPerdida() {
   return $this->gananciaPerdida;
public function setGananciaPerdida($gananciaPerdida) {
   $this->gananciaPerdida = $gananciaPerdida;
private function calcularCambio() {
   if ($this->gananciaPerdida !== null) {
       $cambioPorcentaje = (($this->gananciaPerdida - $this->costoTotal) / $this->costoTotal) * 100;
       return $cambioPorcentaje;
private function getPrecioMercadoFromAPI($nombre) {
    $api_key = "cnb4dj9r01qks5iv03pgcnb4dj9r01qks5iv03q0";
   $url = "https://finnhub.io/api/v1/quote?symbol=$nombre&token=$api_key";
   $response = file_get_contents($url);
   $data = json_decode($response, true);
   if ($data && isset($data['c'])) {
       return $data['c'];
   } else {
```

Se agrego dos nuevos campos (cambio y gananciaPerdida) a la consulta SQL y ajustan la preparación de la consulta y la vinculación de parámetros para incluir estos dos nuevos valores en la inserción de datos en la tabla de acciones.

```
// Prepara una consulta SQS para insertar una neuro fila en la tabla "acciones".

// Prepara una consulta SQS para insertar una neuro fila en la tabla "acciones".

// Prepara una consulta SQS para insertar una neuro fila en la tabla "acciones".

// Prepara una consulta SQS para insertar una neuro fila en la tabla "acciones".

// Prepara una consulta SQS para insertar una neuro fila en la tabla "acciones".

// Prepara una consulta SQS para insertar una neuro fila en la tabla "acciones".

// Prepara una consulta SQS para insertar una neuro fila en la tabla "acciones".

// Prepara una consulta SQS para insertar una neuro fila en la tabla "acciones".

// Prepara una consulta SQS para insertar una neuro fila en la tabla "acciones".

// Prepara una consulta SQS para insertar una neuro fila en la tabla "acciones".

// Prepara una consulta SQS para insertar una neuro fila en la tabla "acciones".

// Prepara una consulta SQS para insertar una neuro fila en la tabla "acciones".

// Obtiene los atributos de la instancia de Accion neuro en la seccion-apeticologio ()

// Prepara una consulta SQS para insertar una neuro fila en la tabla "acciones".

// Prepara una consulta SQS para insertar una neuro fila en la tabla "acciones".

// Prepara una consulta SQS para insertar una neuro fila tabla "acciones".

// Prepara una consulta SQS para insertar una neuro fila tabla "acciones".

// Prepara una consulta SQS para insertar una neuro fila tabla "acciones".

// Prepara una consulta SQS para insertar una neuro fila tabla "acciones".

// Prepara una neuro fila tabla "acciones".

// Prepara
```

#### 3. Pruebas:

a. Pruebas de la API Finnhub

Se realizaron las pruebas unitarias de todos los métodos introucidos que hacen uso de la API Finngub

```
private function getPrecioMercadoFromAPI($nombre) {
    $api_key = "cnb4dj9r01qks5iv03pgcnb4dj9r01qks5iv03q0";
    $url = "https://finnhub.io/api/v1/quote?symbol=$nombre&token=$api_key";

$response = file_get_contents($url);

$data = json_decode($response, true);

if ($data && isset($data['c'])) {
    return $data['c'];
} else {
    return -1;
}
```

```
private function calcularCambio() {
   if ($this->gananciaPerdida !== null) {
        $cambioPorcentaje = (($this->gananciaPerdida - $this->costoTotal) / $this->costoTotal) * 100;
        return $cambioPorcentaje;
   } else {
        return null; // Maneja el caso en el que no se pueda obtener La ganancia/pérdida
   }
}
```

```
public function test_given_oneAction_when_consultPrice_then_true() {
    $accion = new Accion("AAPL", "2024-02-25", 200, 7);
    $precioMercado = $accion->getGananciaPerdida();
    $cambio = $accion->getCambio();
    $this->assertNotEquals(-1, $precioMercado);
    $this->assertNotNull($cambio);
}
```