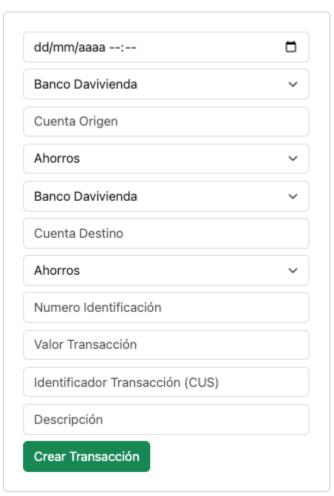
# Taller 2 - Blockchain

## 1. Crear el formulario php:



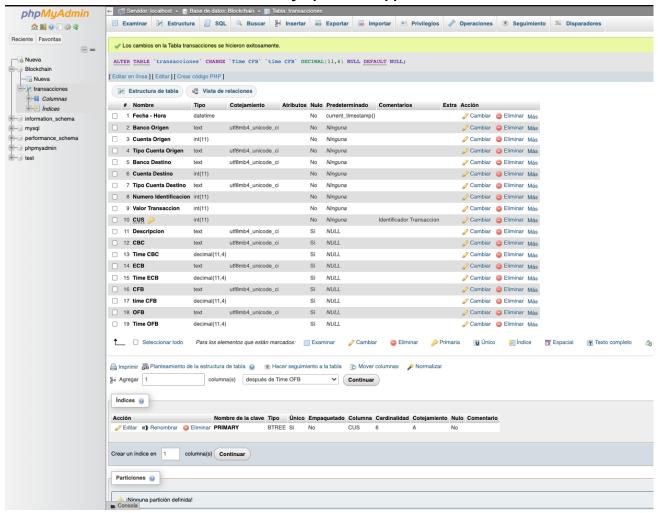
```
<div class="container p-4">
   <div class="row">
       <div class="col-md-4">
           <?php if (isset($_SESSION['message'])) { ?>
               <div class="alert alert-<?= $_SESSION['message_type']; ?> alert-dismissible fade show" role="ale
                  <?= $_SESSION['message'] ?>
                   <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
               </div>/.alert.alert-<?=.$_SESSION[message_type];.?>.alert-dismissible.fade.show
               <?php session_unset();</pre>
           <div class="card card-body">
               <form action="save_tx.php" method="POST">
                   <div class="form-group m-2">
                      <input type="datetime-local" name="fecha-hora" class="form-control" placeholder="Fecha</pre>
                          autofocus>/.form-control
                   <div class="form-group m-2">
                       <select class="form-select" aria-label="Default select example" name="banco_origen">
                           <option selected>Banco Davivienda</option>
                           <option value="1">Bancolombia</option>
                           <option value="2">Banco BBVA</option>
                          <option value="3">Banco Falabella</option>
                           <option value="4">Banco Caja Social</option>
                   </div>/.form-group.m-2
                   <div class="form-group m-2">
                       <input type="number" name="cuenta_origen" class="form-control" placeholder="Cuenta Origen"</pre>
                          autofocus>/.form-control
                   </div>/.form-group.m-2
                   <div class="form-group m-2">
                       <select class="form-select" aria-label="Default select example" name="tipo_cuenta_origen"</pre>
                           <option selected>Ahorros
                           <option value="1">Corriente</option>
                       </select>/.form-select
                   </div>/.form-group.m-2
                   <div class="form-group m-2">
                       <select class="form-select" aria-label="Default select example" name="banco_destino">
                           <option selected>Banco Davivienda</option>
```

Una vez creado el formulario con php, html y bootstrap 5 procedemos a crear el archivo donde ira toda la lógica del formulario para insertar la información en la base de datos

2. Inserción de transacciones en la base de datos:

```
● taller_2_blockchain > ● save_tx.php > ...
               if (isset($_POST['save_tx'])){
                        $fecha_hora = $_POST['fecha-hora'];
                        $banco_origen = $_POST['banco_origen'];
                        $cuenta_origen = $_POST['cuenta_origen'];
                        $tipo_cuenta_origen = $_POST['tipo_cuenta_origen'];
                       $banco_destino = $_POST['banco_destino'];
$cuenta_destino = $_POST['cuenta_destino'];
                        $tipo_cuenta_destino = $_POST['tipo_cuenta_destino'];
                        $numero_identificacion = $_POST['numero_identificacion'];
                        $valor_transaccion = $_POST['valor_transaccion'];
                        $cus = $_POST['CUS'];
                        $descripcion = $_POST['descripcion'];
                        $transaccion = $fecha_hora + $cuenta_origen + $cuenta_destino + $numero_identificacion + $valor_transaccion + $cus;
                        $start_time_cbc = microtime(true);
                        $CBC = encrypt_decrypt_cbc('cifrar', $transaccion);
                        $end_time_cbc = microtime(true);
                        $elapsed_time_cbc = ($end_time_cbc - $start_time_cbc) * 1000;
                        $start time ecb = microtime(true);
                        $ECB = encrypt_decrypt_ecb('cifrar', $transaccion);
                        $end_time_ecb = microtime(true);
                        $elapsed_time_ecb = ($end_time_ecb - $start_time_ecb) * 1000;
                        $start_time_cfb = microtime(true);
                        $CFB = encrypt_decrypt_cfb('cifrar', $transaccion);
                        $end_time_cfb = microtime(true);
                        $elapsed_time_cfb = ($end_time_cfb - $start_time_cfb) * 1000;
                        $start_time_ofb = microtime(true);
                        $0FB = encrypt_decrypt_ofb('cifrar', $transaccion);
                        $end_time_ofb = microtime(true);
                        $elapsed_time_ofb = ($end_time_ofb - $start_time_ofb) * 1000;
                        $query = "INSERT INTO `transacciones`(`Fecha - Hora`, `Banco Origen`, `Cuenta Origen`, `Tipo Cuenta Origen`, `Banco Destino`
`Numero Identificacion`, `Valor Transaccion`, `CUS`, `Descripcion`, `CBC`, `Time CBC`, `ECB`, `Time ECB', `CFB`, `Time CFB'
VALUES ('$fecha_hora', '$banco_origen', '$cuenta_origen', '$tipo_cuenta_origen', '$banco_destino', '$cuenta_destino', '$tipo_cuenta_origen', '$banco_destino', '$cuenta_destino', '$tipo_cuenta_origen', '$cuenta_destino', '$cuenta_destino', '$tipo_cuenta_origen', '$cuenta_destino', '$tipo_cuenta_origen', '$cuenta_destino', '$cuenta_destino', '$tipo_cuenta_origen', '$cuenta_destino', '$tipo_cuenta_origen', '$cuenta_destino', '$cuenta_destino', '$tipo_cuenta_origen', '$cuenta_destino', '$cuenta
```

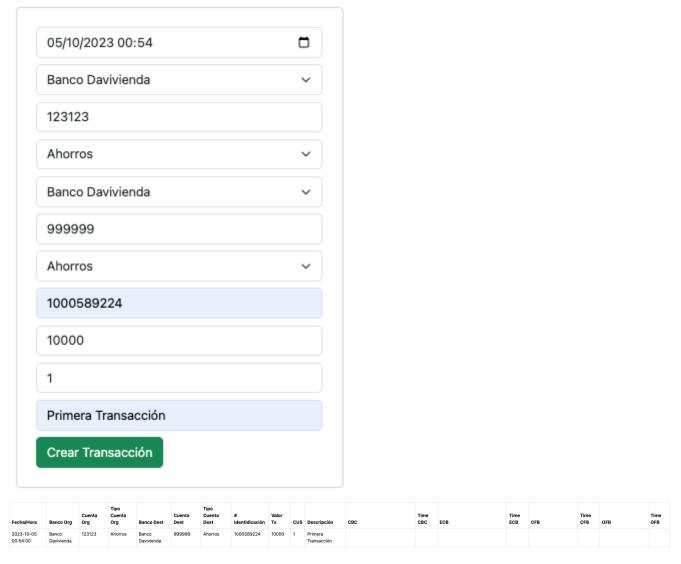
3. Creación de la base de datos en mysql con xampp:



Una vez creada la base de datos con todos los datos, procedemos a crear la conexión con php mediante el archivo bd.php

### 4. Conexión Mysql con PHP

## 5. Realizamos la primera inserción sin realizar la encriptación:



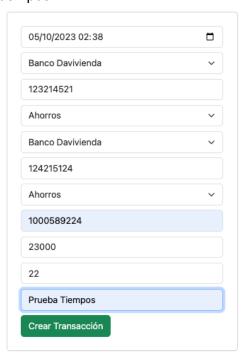
Ya que validamos que funciona la conexión y podemos realizar la inserción de datos, procedemos a implementar el programa visto en clase utilizando los 4 tipos de encriptación solicitados en el taller

6. Implementación función de encriptación con los 4 tipos solicitados y su respectiva medición de tiempos con la función **microtime** 

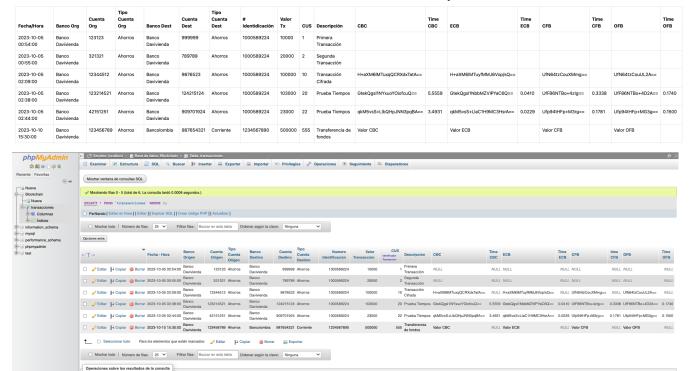
```
$start_time_cbc = microtime(true);
$CBC = encrypt_decrypt_cbc('cifrar', $transaccion);
$end_time_cbc = microtime(true);
$elapsed_time_cbc = ($end_time_cbc - $start_time_cbc) * 1000;
$start_time_ecb = microtime(true);
$ECB = encrypt_decrypt_ecb('cifrar', $transaccion);
$end_time_ecb = microtime(true);
$elapsed_time_ecb = ($end_time_ecb - $start_time_ecb) * 1000;
$start_time_cfb = microtime(true);
$CFB = encrypt_decrypt_cfb('cifrar', $transaccion);
$end_time_cfb = microtime(true);
$elapsed_time_cfb = ($end_time_cfb - $start_time_cfb) * 1000;
$start_time_ofb = microtime(true);
$0FB = encrypt_decrypt_ofb('cifrar', $transaccion);
$end_time_ofb = microtime(true);
$elapsed_time_ofb = ($end_time_ofb - $start_time_ofb) * 1000;
```

Una vez encriptado de las 4 formas almacenamos el tiempo de cada encriptación y su respectivo texto encriptado en la base de datos junto con los datos de la transacción:

7. Creación de la transacción con sus respectivos encriptaciones (4) y toma de tiempos:



Evidenciamos la inserción de los datos en la base de datos y en la tabla:



### **CUADRO COMPARATIVO:**

СВС		Tamaño Bloque	ECB	Time ECB	Tamaño Bloque	CFB	Time CFB	Tamaño Bloque	OFB	Time OFB	Tamaño Bloque
GtekQgsl1NYx	5,555		GtekQgsl 1NbbMZ VIPYaC6	0,04		UfF86N TBo+4z					
uoYOiofcuQ==	9	24	Q==	10	24	Ig==	0,3338	16	UfF86NTBo+4D2A==	0,1740	16
	3,493		qkM5vsS +LlaC1H 9MC3Hsr	0,02		Ufp94tH Fp+M3r					
A==	1	24	A==	29	24	g==	0,1781	16	Ufp94tHFp+MG3g==	0,1500	16

#### **Conclusiones:**

- El algoritmo más eficiente en tiempo es el ECB
- El algoritmo menos eficiente en tiempo es el CBC con gran diferencia de los demás
- El algoritmo más completo es el ECB dado que es más eficiente en tiempo y además genera un bloque cifrado más largo por lo tanto más seguro.

## Características Máquina Utilizada:

RAM: 8gbCPU: Chip M1Memoria: 256 Gb

Link codigo y script: https://github.com/DanielHurtado-040801/Blockchain-Taller\_2.git