



PROYECTO FINAL

ISIDRO PÉREZ PADILLA

2º DAW Tarde

- 1. Descripción del proyecto**
- 2. Stack tecnológico**
- 3. Requisitos funcionales**
- 4. Prototipo de la interfaz**
- 5. Esquema de base de datos**
- 6. Breve análisis/comparativa con las alternativas del mercado**
- 7. Conclusión**
- 8. Linkografía**

Descripción del proyecto

Como surge la idea y objetivo de la aplicación

A mi desde hace muchos años siempre me ha gustado la tecnología y recuerdo que hace muchos años vi en la televisión una noticia sobre un nuevo tipo de moneada a las que llamaron criptomonedas y estaban hablando concretamente de Bitcoin que cuando nació no valía nada y años más tarde valía 3000 euros. Y hace seis o siete años hubo un airdrop (Te regalaban criptomonedas valoradas en 50 euros) de una criptomoneda y yo por ese entonces era un chaval sin ingresos ninguno y decidí cogerlo y vi la cantidad de criptomonedas que había, y vi que el Bitcoin había llegado a valer 20000 euros, era algo que me sorprendía por que ves como pasan oportunidades de “solucionarte la vida” pero yo como no sabía de eso, lo obvie. Y en 2019 hubo otro airdrop valorado en 50 euros de una criptomoneda y en vez de cogerlos y guardármelos decidí dejarlos ahí a ver que pasaba. Meses después lo vi y no era 50 euros eran 300, entonces fue cuando dije tengo que aprender de este mundo y ahí fue cuando me puse a ello.

El comienzo fue lo más difícil, no sabía por dónde empezar, busque la criptomoneda en la que deje el dinero, Ethereum, vi dos términos que se repetían más de una vez, blockchain y smart contracts, y a partir de ahí todo fue más fácil, encontré más utilidades, más criptomoneda, más altcoins...

El mundo de las criptomonedas y su tecnología ha crecido mucho en muy poco tiempo y sigue creciendo con nuevos conceptos, ideas y tecnologías como por ejemplo son los NFT, web3, metaverso... Y cada vez hay más personas muestran interés por este mundo y se encontraran con el mismo problema que yo tuve.

El **objetivo** de mi proyecto es **ayudar al usuario en el comienzo**, para adentrarse en este mundo, **explicando algunos términos importantes** y **facilitándole algunas criptomonedas** por las que empezar a investigar. Ofreciéndole la posibilidad de **poder gestionar lo que compre y lo que venda** teniendo así control de su patrimonio invertido.

Stack tecnológico

Estos son las tecnologías que utilizo en mi proyecto:

Back-end:

- PHP 8.0

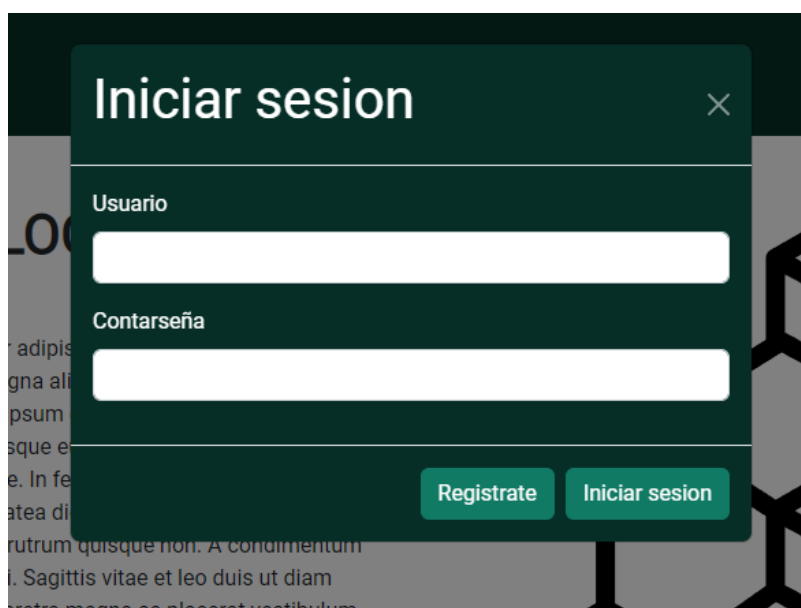
Front-end:

- APIs: jQuery 3.6.1, Charts.js 3.5, Coingecko v3
- Librerías: Bootstrap 5.2, Bootstrap Icons
- Lenguajes: JavaScript (AJAX), CSS3, HTML5

Utilizare tres APIs, jQuery para hacer las peticiones al servidor, chart.js porque mi aplicación requiere de gráficas y esta API me facilita el trabajo y Coingecko es la API de una página que proporciona información sobre precios, capitalización, capital en circulación y muchas más cosas criptomonedas.

Utilizare la librería de Bootstrap para hacer el diseño, es una librería muy utilizada y hay mucha documentación y ejemplos que me pueden ayudar en mi trabajo. Bootstrap Icons es una librería de los mismos creadores que me proporcionara iconos muchos para implementar en el diseño.

En este proyecto trabajo el back-end y el front-end por separado, lo explico con un ejemplo:



```

<div class="modal-body">
<form action="#" method="post" id="form_login" name="form_login">
  <div class="mb-3">
    <label for="usuario" class="form-label text-white">Usuario</label>
    <input type="text" class="form-control" id="usuario" name="usuario">
  </div>
  <div class="mb-3">
    <label for="password" class="form-label text-white">Contraseña</label>
    <input type="password" class="form-control" id="password" name="password">
  </div>
</div>
<div role="alert" id="error_login">

</div>
<div class="modal-footer">
  <button class="btn text-white boton" data-bs-toggle="modal" data-bs-target="#register" type="button">Registrar</button>
  <button type="submit" class="btn text-white boton">Iniciar sesión</button>
</div>

```

Vamos a usar el inicio de sesión de mi pagina, al yo rellenar el usuario y la contraseña con mis credenciales al hacer click en el botón de iniciar sesión hará lo siguiente

```

    });
  })

  //Al hacer click en el boton de form_login recogera los datos del formulario y los almacenara en
  //un array (datos) que se enviara mediante un post al controlador (loginController.php)
  login.addEventListener('submit', (e) => {
    e.preventDefault();
    var datos = $('#form_login').serializeArray();
    $.ajax({
      type: "POST",
      url: "controller/loginController.php",
      data: datos,
      dataType: "JSON",
      success: function (response) {
        let errorLogin = document.getElementById("error_login");
        sessionStorage.setItem('role', response["role"]);
        if (response) {
          location.reload()
        } else { //Si es false mostrara el mensaje siguiente en pantalla
          errorLogin.setAttribute("class", "alert alert-danger m-3");
          errorLogin.innerHTML = "Usuario o contraseña incorrectos";
        }
        window.onload
      }
    });
  });
});

```

Almacenará los datos del formulario en un array y hará una petición POST al controlador utilizando AJAX

```

oller > loginController.php > ...
<?php
require_once "../model/Usuario.php";

//Iniciamos una session y guardamos los datos del post en el objeto creado del usuario
session_start();
$post = $_POST;

$usuario = new Usuario("", $post["usuario"], "", "", "", md5($post['password']));

// Llamamos a la funcion
if (!$usuario->iniciar_session()) { // Si devuelve false enviara a la respuesta false
    echo "false";
} else {
    echo json_encode($usuario->iniciar_session());
}
}

```

Este recibe la petición llama al modelo del usuario, donde esta la función:

```

// Comprueba que exista el usuario y la password si existe lo almacena en un array
// si el array esta vacio devolvera al controlador false y si esta lleno devolvera la session rellena
public function iniciar_session()
{
    $sql = "SELECT * FROM usuarios
            WHERE usuario LIKE '{$this->usuario}' and password LIKE '{$this->password}'";

    $result = $this->conexion->query($sql);
    $array = array();
    while ($row = $result->fetch_assoc()) {
        $array = $row;
    }
    if (empty($array)) {
        return false;
    } else {
        $_SESSION["role"] = $array["role"];
        $_SESSION["nombre"] = $array["nombre"];
        $_SESSION["email"] = $array["email"];
        $_SESSION["apellido"] = $array["apellido"];
        $_SESSION["usuario"] = $array["usuario"];
        $_SESSION["id"] = $array["id"];
        return $_SESSION;
    }
}
}

```

La función devuelve lo que tenga que devolver al controlador y mediante un echo devuelve el resultado

```

success: function (response) {
  let errorLogin = document.getElementById("error_login");
  sessionStorage.setItem('role', response["role"]);
  if (response) {
    location.reload()
  } else { //Si es false mostrara el mensaje siguiente en pantalla
    errorLogin.setAttribute("class", "alert alert-danger m-3");
    errorLogin.innerText = "Usuario o contraseña incorrectos";
  }
  window.onload
}

```

Recibe la respuesta y en función a lo que reciba inicia sesión o muestra un mensaje de error

The image shows a dark-themed login modal window titled "Iniciar sesion" with a close button (X) in the top right corner. Inside the modal, there are two input fields: "Usuario" with the text "admin" and "Contraseña" with masked characters ".....". Below these fields is a pink error message box that reads "Usuario o contraseña incorrectos". At the bottom of the modal, there are two buttons: "Registrate" and "Iniciar sesion". The background of the page is a dark gray with a subtle geometric pattern.

Así funciona prácticamente toda la pagina

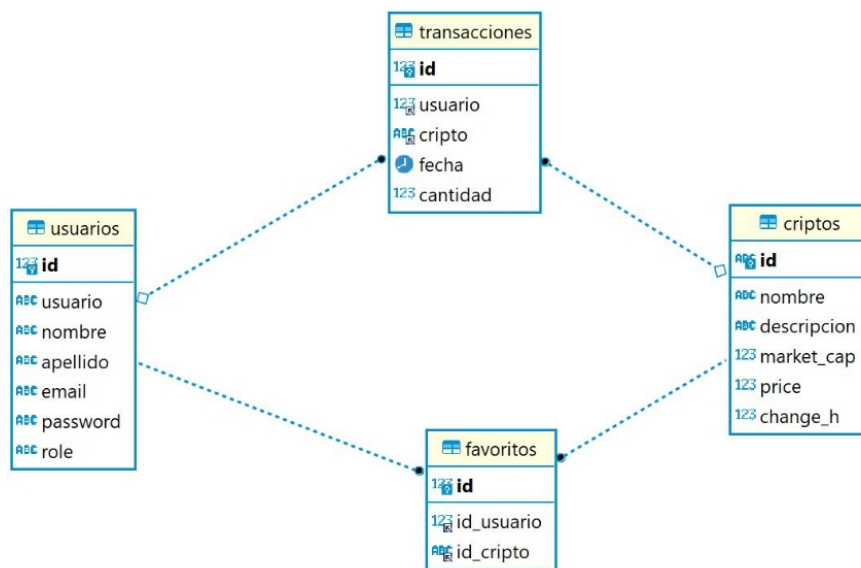
Requisitos funcionales

- REQ01: El sistema podrá registrar usuarios
- REQ02: El sistema podrá permitir iniciar sesión al usuario
- REQ03: El sistema podrá permitir al usuario añadir criptos a favoritos
- REQ03: El sistema mostrará una vista con los datos de cada criptomoneda
- REQ04: El sistema podrá permitir al usuario eliminar criptos de favoritos
- REQ05: El sistema podrá permitir al usuario editar sus datos
- REQ06: El sistema podrá permitir al usuario añadir nuevas transacciones
- REQ07: El sistema mostrará al usuario un resumen de las criptomonedas que tiene
- REQ08: El sistema podrá permitir al administrador agregar nuevas criptomonedas
- REQ09: El sistema podrá permitir al administrador eliminar las criptomonedas
- REQ10: El sistema podrá permitir al administrador editar la información de las criptomonedas
- REQ11: El sistema podrá permitir al administrador editar los datos y el rol de los usuarios
- REQ12: El sistema podrá permitir al administrador eliminar los usuarios

Prototipo de la interfaz

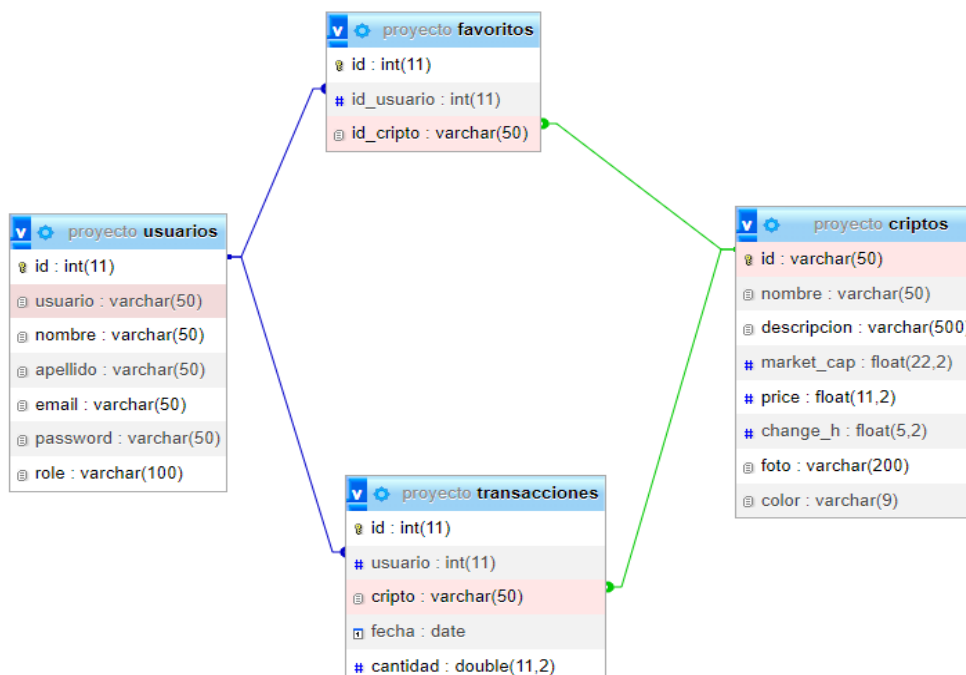
Puedes ver el prototipo haciendo clic [aquí](#).

Esquema de la base de datos



Versión 1.0 consta de **cuatro tablas**:

- **Usuarios:** donde estarán los datos del usuario.
- **Criptos:** donde estarán los datos de cada criptos almaceno también los valores dinámicos para poder mostrar los últimos datos actualizados por si la API dejase de funcionar.
- **Favoritos:** donde almacenara las criptos que están en la lista de seguimiento de cada usuario
- **Transacciones:** Almacena todos los registros de transacciones hecho por los usuarios



Versión 1.1: Añade **dos campos nuevos** a la tabla de **criptos**

Breve análisis/comparativa con las alternativas del mercado

Mi proyecto está inspirado en algunos aspectos en coinmarketcap o coingecko que te ofrecen muchísima información sobre el mercado de las criptos, pero no comparten un mismo objetivo. El objetivo de mi aplicación es ofrecer una información mínima para que los usuarios aprendan a moverse en el mercado.

Linkografía

[Bootstrap](#)

[Bootstrap Icons](#)

[MDN web docs javascript](#)

[PHP](#)

[jQuery](#)

[Chart.js](#)

[StackOverflow](#)

[w3Schools](#)

[Github](#)