

necesito para un sistema de pago que funciona sin api y que el carrito esta en la landing y que desde ahi despliegue el pago sin api y que los registros los haga para el panel de administrador y que desde ahi pueda autorizar el pago para realizar la debida inscriocion si se entiende?

Para tu requerimiento de un sistema de pago **sin usar APIs externas**, con carrito en la landing page, que despliegue el pago desde ahí mismo, y que registre las solicitudes de pago para que el administrador pueda autorizar manualmente la inscripción, te propongo una solución básica con PHP, MySQL y MVC orientado a objetos.

Concepto general

- El carrito está en la landing page (frontend).
- Al confirmar el pago, se registra un **pedido pendiente** en la base de datos.
- No se realiza pago automático ni con API, sino que el administrador revisa el pedido y autoriza el pago manualmente (por ejemplo, tras recibir transferencia o pago offline).
- Cuando el administrador autoriza el pago, el sistema cambia el estado a "pagado" y se realiza la inscripción al curso.
- El usuario puede ver el estado de su pedido (pendiente, aprobado, rechazado).

Ejemplo básico de implementación

1. Tablas MySQL necesarias

```
CREATE TABLE usuarios (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(100),  
    email VARCHAR(100) UNIQUE,  
    password VARCHAR(255),  
    rol ENUM('admin','estudiante') DEFAULT 'estudiante'  
);  
  
CREATE TABLE cursos (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    titulo VARCHAR(255),  
    descripcion TEXT,  
    precio DECIMAL(10,2)
```

```
);

CREATE TABLE pedidos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    usuario_id INT,
    total DECIMAL(10,2),
    estado ENUM('pendiente', 'aprobado', 'rechazado') DEFAULT 'pendiente',
    fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (usuario_id) REFERENCES usuarios(id)
);

CREATE TABLE pedido_detalle (
    id INT AUTO_INCREMENT PRIMARY KEY,
    pedido_id INT,
    curso_id INT,
    cantidad INT DEFAULT 1,
    precio_unitario DECIMAL(10,2),
    FOREIGN KEY (pedido_id) REFERENCES pedidos(id),
    FOREIGN KEY (curso_id) REFERENCES cursos(id)
);
```

2. Flujo del carrito y pedido

- En la landing page el usuario agrega cursos al carrito (puedes usar JavaScript para manejar el carrito en frontend).
- Al enviar el pedido, se envía el resumen al backend (PHP) para crear un registro en pedidos y sus detalles en pedido_detalle con estado pendiente.
- Se muestra mensaje al usuario: "Pedido recibido, pendiente de aprobación".
- El administrador en su panel ve los pedidos pendientes y puede aprobar o rechazar.
- Al aprobar, el sistema cambia el estado y registra la inscripción del usuario al curso.

3. Código PHP simplificado para crear pedido (Controlador)

```
class PedidoController {
    private $db;

    public function __construct($pdo) {
        $this->db = $pdo;
    }

    // Recibe el carrito desde POST: array de cursos con id y cantidad
    public function crearPedido($usuario_id, $carrito) {
        try {
            $this->db->beginTransaction();

            // Calcular total
            $total = 0;
            foreach ($carrito as $item) {
                $stmt = $this->db->prepare("SELECT precio FROM cursos WHERE id = ?");
```

```

        $stmt->execute([$item['curso_id']]);
        $precio = $stmt->fetchColumn();
        $total += $precio * $item['cantidad'];
    }

    // Insertar pedido
    $stmt = $this->db->prepare("INSERT INTO pedidos (usuario_id, total) VALUES (?, ?)");
    $stmt->execute([$usuario_id, $total]);
    $pedido_id = $this->db->lastInsertId();

    // Insertar detalles
    foreach ($carrito as $item) {
        $stmt = $this->db->prepare("SELECT precio FROM cursos WHERE id = ?");
        $stmt->execute([$item['curso_id']]);
        $precio = $stmt->fetchColumn();

        $stmt = $this->db->prepare("INSERT INTO pedido_detalle (pedido_id, curso_id, cantidad, precio) VALUES (?, ?, ?, ?)");
        $stmt->execute([$pedido_id, $item['curso_id'], $item['cantidad'], $precio]);
    }

    $this->db->commit();

    return ['success' => true, 'pedido_id' => $pedido_id];
} catch (Exception $e) {
    $this->db->rollBack();
    return ['success' => false, 'error' => $e->getMessage()];
}
}

// Listar pedidos para admin
public function listarPedidosPendientes() {
    $stmt = $this->db->query("SELECT p.*, u.nombre, u.email FROM pedidos p JOIN usuarios u ON p.usuario_id = u.id");
    return $stmt->fetchAll(PDO::FETCH_ASSOC);
}

// Aprobar pedido
public function aprobarPedido($pedido_id) {
    // Cambiar estado a aprobado
    $stmt = $this->db->prepare("UPDATE pedidos SET estado = 'aprobado' WHERE id = ?");
    $stmt->execute([$pedido_id]);

    // Aquí puedes agregar lógica para inscribir al usuario en los cursos
    // Por ejemplo, insertar en tabla inscripciones (no incluida en este ejemplo)
}
}

```

4. Ejemplo de formulario en landing para enviar carrito (simplificado)

```

<form action="/pedido/crear" method="post" id="form-pedido">
    <!-- Aquí puedes listar cursos con checkbox o cantidad -->
    <input type="hidden" name="carrito" id="carrito-json">

    <button type="submit">Realizar Pedido</button>
</form>

```

```

<script>
  // Ejemplo simple: carrito como array de objetos {curso_id, cantidad}
  const carrito = [
    {curso_id: 1, cantidad: 1},
    {curso_id: 2, cantidad: 2}
  ];

  document.getElementById('carrito-json').value = JSON.stringify(carrito);

  document.getElementById('form-pedido').addEventListener('submit', function(e){
    e.preventDefault();

    fetch('/pedido/crear', {
      method: 'POST',
      headers: {'Content-Type': 'application/json'},
      body: JSON.stringify({carrito: carrito})
    }).then(res => res.json())
      .then(data => {
        if(data.success) alert('Pedido creado, pendiente de aprobación');
        else alert('Error: ' + data.error);
      });
  });
</script>

```

5. Panel de administrador para aprobar pedidos

- Listar pedidos con estado pendiente.
- Botones para aprobar o rechazar (que llamen a métodos PHP que cambian el estado).
- Al aprobar, se registra la inscripción.

Resumen

- No hay integración con API de pagos.
- El pago es manual y el sistema solo registra pedidos pendientes.
- El admin autoriza el pago y activa la inscripción.
- El carrito y la creación de pedido se manejan en la landing con JavaScript y PHP.
- El backend registra todo en MySQL para seguimiento y control.

Si quieres, puedo ayudarte a generar el código completo para cada parte (carrito, controlador, vistas, panel admin) o darte un ejemplo más detallado para alguna funcionalidad específica. Solo dime qué necesitas.