Proyecto máquina expendedora virtual.

Componentes

**1. Hardware**

* **Unidad de procesamiento / control**
  + **Raspberry Pi 4** (flexibilidad y facilidad de integración con periféricos, buena para desarrollo rápido).
  + **Arduino Mega / Uno / ESP32** (control directo de motores y sensores; el ESP32 además maneja WiFi nativo).
  + Opción híbrida: Raspberry Pi para lógica de alto nivel + Arduino/ESP32 para control físico.
* **Mecanismo de dispensado**
  + Motores paso a paso con driver (por ejemplo, NEMA 17 + A4988 o DRV8825).
  + Servomotores (para compuertas o mecanismos más simples).
  + Motores DC con reductora y sensores de posición.
* **Sensores**
  + Sensor óptico o infrarrojo para detectar la entrega del producto.
  + Microinterruptores para final de carrera.
  + Sensores de peso (celdas de carga + HX711) para verificar si se retiró un producto.
* **Pantalla y control de usuario**
  + Pantalla táctil LCD/LED (por ejemplo, 7" HDMI para Raspberry Pi).
  + Botonera física y/o display LCD 16x2 o 20x4.
  + Lector de códigos QR (para leer el QR de pago de Mercado Pago).
* **Conectividad**
  + Módulo WiFi (ESP32 o integrado en Raspberry Pi).
  + Ethernet si se busca mayor estabilidad.
  + Opcional: módulo 4G/LTE para ubicación remota.
* **Alimentación eléctrica**
  + Fuente de alimentación con regulación (por ejemplo, 12V y 5V).
  + Batería UPS o respaldo para cortes de energía.
* **Estructura física**
  + Chasis de madera, acrílico o metal.
  + Bandejas o tolvas para almacenar el producto.
  + Sistema de seguridad física (cerraduras, refuerzos).

**2. Software**

* **Aplicación de control**
  + Python, Node.js o C++ según plataforma.
  + Interfaz gráfica (Tkinter, PyQt, o web app en HTML/CSS/JS).
* **Integración con Mercado Pago**
  + Uso de **Mercado Pago API** (Checkout Pro o Checkout QR).
  + Generación de QR dinámico vinculado a la transacción.
  + Webhooks o polling para verificar acreditación del pago.
* **Lógica de operación**
  + Secuencia: mostrar producto → generar QR → esperar confirmación de pago → activar dispensador → registrar transacción.
  + Registro de ventas en base de datos local o remota.
* **Base de datos**
  + SQLite (local, simple).
  + MySQL/PostgreSQL (si es en red).
  + Cloud Firestore o equivalente si se requiere conexión a la nube.

**3. Servicios y métodos**

* **Mercado Pago Developers**
  + Configuración de credenciales (client\_id, client\_secret, access\_token).
  + Implementación de webhook para recibir confirmaciones de pago.
* **Seguridad**
  + Encriptación de datos de transacciones.
  + Validación de tokens y control de accesos.
* **Replicación de métodos comerciales**
  + Uso de QR dinámicos (no estáticos) para cada compra.
  + Confirmación en tiempo real antes de habilitar la entrega.
  + Registro automático de inventario.
  + Interfaz amigable al usuario final.

Enlaces

<https://youtu.be/DO3AciBz_-A?si=AGaABoVpVOVBhcKK>

<https://youtu.be/x2otDesU6_k?si=dUH6devuLa087ZD6>

<https://gist.github.com/Sevenmojoe/e456be5fb79ad2630e88424d6e7c40d7>

<https://www.youtube.com/watch?v=l1qTtP01Pno>

NUEVOS COMPONENTES

**Microcontrolador**

* **ESP32** (recomendado) o ESP8266
  + Tiene WiFi integrado para conectarse a internet.
  + Puede manejar librerías para mostrar QR en pantallas.
* Alternativa: Arduino UNO + Ethernet/WiFi shield, pero es más complicado por HTTPS.

**2️⃣ Pantalla para mostrar el QR**

* **Pantalla TFT o LCD a color** (320x240 o 240x240 mínimo)  
  Ejemplos:
  + **TFT 2.8" SPI ILI9341**
  + **TFT 3.5" ILI9488**
  + **OLED 1.3" a 2.8"** (si quieres mini QR)
* Librerías: **TFT\_eSPI**, **Adafruit\_GFX** y **qrcode** para generar QR.

**3️⃣ Botón o sensor de inicio**

* **Pulsador** (opcional) para generar QR al presionar.
* **Sensor de stock o proximidad** (opcional) para generar QR automáticamente según la lógica del negocio.

**4️⃣ Fuente de alimentación**

* **5V USB** o **Batería LiPo 3.7V con regulador 5V** (según la pantalla y ESP32).

**5️⃣ Servidor o intermediario**

* Puede ser un **PC, Raspberry Pi o hosting web**.
* Necesita tener **PHP, Node.js o Python** para manejar la **API de Mercado Pago**.
* También requiere **Access Token** de Mercado Pago.

**6️⃣ Cables y protoboard**

* Conexiones **SPI o I2C** para la pantalla.
* **Cables dupont** y **protoboard** para probar el prototipo.

**7️⃣ Librerías de software**

Para Arduino/ESP32:

* **WiFi.h** o **ESP8266WiFi.h**
* **HTTPClient.h** (para hacer peticiones al servidor)
* **TFT\_eSPI.h** o **Adafruit\_GFX.h** (para pantalla)
* **qrcode.h** o **qrcodegen.h** (para generar QR a partir de URL)

En el servidor:

* **Mercado Pago SDK** (PHP/Node.js/Python) para generar QR dinámico.