Prueba profesional de competencia para evaluar las UC0491_3, UC0492_3, UC0493_3

Objetivo

Preparar una pequeña aplicación para gestionar clientes (alta, baja, modificación y listado) y entregarla de forma que cualquier persona pueda instalarla, ejecutarla y verificar su funcionamiento básico sin ayuda.

Tecnologías permitidas

Puedes usar cualquier stack para implementar la prueba.

UI mínima (para cubrir UC0491_3)

Añade una interfaz simple en carpeta public/ compuesta al menos por:

- Archivo 'public/index.html' con formularios (alta/edición) y un listado de clientes.
- Archivo 'public/app.js' con llamadas a la API y manejo de respuestas/errores.
- Archivo 'public/styles.css' (opcional) para estilos básicos.

Cómo entregar:

1. Crea una carpeta llamada codigo_fuente que incluya:
    a. src/cliente: HTML/CSS/JS (o lo que uses)
    b. src/servidor: código de controladores, rutas, consultas.
    c. src/db: db.sql con datos de ejemplo
    d. No incluyas ninguna otra dependencia pesada
2. Crea una carpeta con el nombre tuapellido1_tuapellido2_tunombre_prueba_UC0226_3
3. Incluye en esa carpeta:
    a. Este documento en formato pdf.
    b. La carpeta codigo_fuente
    c. db.sql (o similar)
    d. README.md
    e. Un tag de Git v0.1.0 con release notes (5–8 líneas).
4. Comprime la carpeta en formato zip (no rar).
5. Envía por correo electrónico a jguija2@educarex.es

Demostración en vivo por videoconferencia

El sábado día 15 de Noviembre, en session de evaluación por videoconferencia, el candidato demostrá que la aplicación funciona en los siguientes 5 casos:

- Alta: crear un cliente válido → aparece en el listado.
- Validación: crear con email inválido → responde 400 y la UI muestra el error.
- Modificación: cambiar un dato → se ve actualizado en la tabla.
- Baja: borrar un cliente → ya no aparece en el listado (con confirmación).
- Búsqueda (o paginación): filtrar por nombre/apellidos/email → lista correcta.

Estructura de repositorio (sugerida)

```
/ (raíz)
├─ src/              # código de la app (PHP o Node)
├─ public/           # UI mínima (index.html, app.js, styles.css)
├─ README.md         # obligatorio
├─ api.md            # obligatorio (puede ir dentro del README si prefieres)
├─ db.sql            # obligatorio
```

Requisitos funcionales mínimos (CRUD)

- Campos del cliente: id, nombre, apellido1, apellido2, email (único), telefono, fecha_alta, notas.
- Operaciones: crear, listar (con búsqueda por email), modificar y borrar.
- Validación básica en servidor: email válido; evitar duplicado por email.

Contenido de cada archivo

1) README.md (Ejemplo)

Incluye estas secciones (puedes usar el texto siguiente como plantilla).

Título y descripción

```
Gestor de clientes (CRUD) – Entrega v0.1.0
Aplicación mínima para alta, baja, modificación y listado de clientes en
MySQL.
```

Requisitos

- Opción A (PHP): PHP 8.x, Apache/Nginx, MySQL 8.x, extensión PDO.

Instalación (4–6 pasos numerados)

```
1) Crear BD:
   mysql -u root -p < db.sql
2) PHP: copiar src/ a tu servidor y apuntar el DocumentRoot a public/
3) Abrir http://localhost:8080 (o el puerto que indiques) y probar.
```

Estructura de carpetas

2–5 líneas explicando qué hay en src/, public/, etc.

2) db.sql (modelo)

Debe crear la BD o al menos la(s) tabla(s) necesarias y añadir 2–3 filas de ejemplo.

```
CREATE DATABASE IF NOT EXISTS clientesdb DEFAULT CHARACTER SET utf8;
USE clientesdb;

CREATE TABLE IF NOT EXISTS clientes (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(80) NOT NULL,
  apellidos VARCHAR(120) NOT NULL,
  email VARCHAR(120) NOT NULL UNIQUE,
  telefono VARCHAR(30),
  fecha_alta DATE NOT NULL DEFAULT (CURRENT_DATE),
  notas TEXT
);

INSERT INTO clientes (nombre, apellidos, email, telefono, notas) VALUES
('Angélica', 'García', 'López', 'angelica@dominio.com', '600000001', 'Cliente
inicial'),
('Aniceto', 'Cáceres', 'Díaz', 'aniceto@dominio.com', '600000002', 'VIP');
```

3) Tag (etiquetas de version) y release notes (notas de versión)

3.1. Comandos de git:

```
git tag -a v0.1.0 -m "Primera versión"
git push origin v0.1.0
```

1.2. Modelo de release notes (Un **resumen breve (4 ó 5 líneas)** de qué incluye esa versión y cómo afecta al despliegue**):**

```
v0.1.0
- CRUD mínimo de clientes (crear, listar con búsqueda, modificar, borrar).
- Validación de email y control de duplicados.
- Script db.sql con datos de ejemplo.
- README de instalación y api.md con ejemplos.
```

1.3. Historial con 3–5 commits con mensajes claros (captura).