Evaluación sumativa Unidad N°3

Resultado e Indicadores de evaluación por evaluar

Resultados de aprendizajes	Indicadores de logro
3.1 Utiliza el framework definido para construir operaciones CRUD (Crear, Leer, Actualizar y Eliminar), manipulando datos mediante la aplicación.	3.1.1 Diseña operaciones CRUD, siguiendo el framework definido, para manipular datos en la aplicación. 3.1.2 Implementa operaciones CRUD (Crear, Leer, Actualizar y Eliminar) de manera eficiente y precisa, para acceder y modificar datos en la aplicación. 3.1.3 Adapta el framework CRUD a diferentes tipos de datos y estructuras dentro de la aplicación. 3.1.4 Depura errores en la implementación de las operaciones CRUD, para asegurar el correcto funcionamiento de la aplicación. 3.1.5 Optimiza el uso del framework CRUD para mejorar el rendimiento y la eficiencia de la aplicación.

Tiempo: indicado por el docente o plataforma, según sea el caso, en caso de incompatibilidad se aplicará las instrucciones del docente.

Puntaje: 100 puntos.

Característica de la Evaluación

1. Esta es una actividad calificada de la unidad y es de carácter individual.
2. Para conocer cómo serás evaluado/a, revisa la rúbrica disponible en plataforma.
3. Esta es una actividad evaluativa de desarrollo con respuesta extendida, donde se debe analizar el problema, ejecutando los conocimientos y competencias adquiridas en la unidad.
4. Esta actividad evaluativa implicará la entrega de un producto que deberá cargar a la plataforma.

Instrucciones de la evaluación

Evaluación VcM: Sistema de Distintivos Visuales

(ProREP)

Contexto de la situación problema

El Product Owner del proyecto "Buscador de Gestores ProREP" ha solicitado una mejora en la experiencia de usuario. Actualmente, las empresas se listan sin una distinción visual clara de su rubro. Se requiere que las empresas gestoras sean fácilmente identificables mediante símbolos visuales (iconos o etiquetas de color) en los resultados de búsqueda.

El objetivo es construir un prototipo funcional que consuma datos desde un servicio backend y los renderice dinámicamente en el navegador.

Intrucciones

1. **Gestores**. Se puede trabajar con las empresas que son gestoras de residuos.

[GET] api/ v1/ gestores/


```
[
    {
        "id": 1,
        "nombre": "Reciclajes Santiago Ltda.", "empresa_tipo": {
            "id": 1,
            "nombre": "Reciclador de Base",
            "codigo": "R",
            "icono": {
                "FontAwesome": "fa-recycle",
            },
            "color": {
                "Tailwind": "bg-green-100",
                "css": "rgb(220 252 231)",
            },
        },
        "activo": true
    }
]
```


[GET] api/ v1/ gestores/?id=1


```
{
    "id": 1,
    "nombre": "Reciclajes Santiago Ltda.",
    "empresa_tipo": {
        "id": 1,
        "nombre": "Reciclador de Base",
        "codigo": "R",
        "icono": {
```

```
            "FontAwesome": "fa-recycle",
        },
        "color": {
            "Tailwind": "bg-green-100",
            "css": "rgb(220 252 231)",
        },
    },
    "activo": true
}
```


[POST] api/ v1/ gestores/

body json


```
{
    "nombre": "Reciclajes Santiago Ltda.",
    "empresa_tipo": {
        "id": 1
    }
}
```


[PUT] api/ v1/ gestores/

body json


```
{
    "nombre": "Reciclajes Santiago Ltda.",
    "empresa_tipo": {
        "id": 1
    }
}
```


[DELETE] api/ v1/ gestores/ ?id=1

Para desactivar (disable) el registro del id.

[PATCH] api/ v1/ gestores/ ?id=1

Para activar (enable) el registro del id.

2. **Empresa Tipo**. Se trabaja con los tipos de empresas gestoras disponibles.

[GET] api/ v1/ empresaTipo/


```
[
    {
        "id": 1,
        "nombre": "Reciclador de Base",
        "codigo": "R",
        "icono": {
            "FontAwesome": "fa-recycle",
        },
        "color": {
            "Tailwind": "bg-green-100",
            "css": "rgb(220 252 231)",
        },
        "activo": true
    },
    {
        "id": 2,
        "nombre": "Valorizador", "codigo": "V",
        "icono": {
            "FontAwesome": "fa-industry",
        },
        "color": {
            "Tailwind": "bg-blue-100",
            "css": "#eff6ff",
        },
        "activo": true
    },
    {
        "id": 3,
        "nombre": "Consultor",
        "codigo": "C", "icono": {
            "FontAwesome": "fa-user-tie",
        },
        "color": {
```

```
            "Tailwind": "bg-gray-100",
            "css": "#f3f4f6",
        },
        "activo": true
    },
    {
        "id": 4,
        "nombre": "Transportista", "codigo":
        "T",
        "icono": {
            "FontAwesome": "fa-truck",
        },
        "color": {
            "Tailwind": "bg-yellow-100", "css":
            "rgb(254 249 195)",
        },
        "activo": true
    },
    {
        "id": 5,
        "nombre": "Gestor Integral",
        "codigo": "G",
        "icono": {
            "FontAwesome": "fa-globe-americas",
        },
        "color": {
            "Tailwind": "bg-purple-100", "css":
            "rgb(243, 232, 255)",
        },
        "activo": true
    }
]
```


[GET] api/ v1/ empresaTipo/ ?id=1


```
{
    "id": 1,
    "nombre": "Reciclador de Base",
    "codigo": "R",
    "icono": {
```

```
            "FontAwesome": "fa-recycle",
        },
        "color": {
            "Tailwind": "bg-green-100",
            "css": "rgb(220 252 231)",
        },
        "activo": true
    }
```


[POST] api/ v1/ empresaTipo/

body json


```
    {
        "nombre": "Reciclador de Base",
        "codigo": "R",
        "icono": {
            "FontAwesome": "fa-recycle",
        },
        "color": {
            "Tailwind": "bg-green-100",
            "css": "rgb(220 252 231)",
        }
    }
```


[PUT] api/ v1/ empresaTipo/

body json


```
    {
        "id": 1,
        "nombre": "Reciclador de Base",
        "codigo": "R",
        "icono": {
            "FontAwesome": "fa-recycle",
        },
        "color": {
            "Tailwind": "bg-green-100",
            "css": "rgb(220 252 231)",
        }
    }
```


[DELETE] api/ v1/ empresaTipo/ ?id=1

Para desactivar (disable) el registro del id.

[PATCH] api/ v1/ empresaTipo/ ?id=1

Para activar (enable) el registro del id

Tareas a Realizar

3.1.1 – Diseña operaciones CRUD siguiendo el framework definido (REST) para manipular datos

Tarea 1 – Diseñar la estructura completa de la API REST en PHP puro

Debes definir una estructura mínima limpia:

```
/api/
        /v1/
                gestores.php
                empresaTipo.php
/common/
        database.php
        response.php
        validators.php
```

La estructura debe reflejar un diseño RESTful:

- Uso de rutas correctas (/api/v1/gestores)
- Métodos HTTP correctos por requerimiento (GET, POST, PUT, DELETE, PATCH)
- Organización clara entre recursos (gestores vs empresaTipo)
- Separación lógica entre lectura de datos, validación y respuesta

Tarea 2 – Definir la estructura de cada endpoint según el diseño entregado en la evaluación

Para cada recurso debes diseñar:

- Qué parámetros recibe
- Qué JSON retorna
- Qué códigos HTTP corresponden
- Estructura de respuesta común como:

```
{
  "mensaje": "OK",
  "data": {...}
}
```

Esto demuestra la capacidad de **diseñar CRUD siguiendo el marco REST definido**, sin cambiar los requisitos.

3.1.2 – Implementa operaciones CRUD eficientes y precisas

Tarea 3 – Implementar todos los métodos CRUD requeridos por la evaluación

Debes implementar en PHP puro:

Método	Recurso	Acción
GET	gestores	listar y buscar por id
POST	gestores	crear
PUT	gestores	actualizar

Método	Recurso	Acción
DELETE	gestores	desactivar
PATCH	gestores	activar

(mismas operaciones para empresaTipo)

Cada operación debe:

- Leer correctamente del archivo/JSON/base simulada
- Modificar los datos con precisión
- Validar entradas antes de escribir
- Retornar JSON consistente
- Mantener la integridad entre gestores y empresaTipo

Tarea 4 – Aplicar un manejador centralizado de respuestas y errores

Crear una función:

```
function jsonResponse($statusCode, $message, $data = null) {
    http_response_code($statusCode);
    echo json_encode([
        "mensaje" => $message,
        "data" => $data
    ]);
    exit;
}
```

3.1.3 – Adaptar el framework CRUD a diferentes tipos de datos

Tarea 5 – Manejar correctamente estructuras anidadas en 'empresa_tipo'

Los gestores incluyen:

```
"empresa_tipo": {
    "id": 1,
    "nombre": "Reciclador de Base",
    "codigo": "R",
    "icono": {...},
    "color": {...}
}
```

Debes:

- Permitir recibir empresa_tipo como objeto o como id
- Validar que el id exista
- Al devolver datos, **rellenar automáticamente el objeto completo**, no solo el id
- Reutilizar lógica para ambos recursos

Tarea 6 – Implementar conversión automática entre estructuras

Crear funciones como:

```
function expandEmpresaTipo($id) {
  // devuelve el objeto empresaTipo completo
}
```

3.1.4 – Depurar errores para asegurar correcto funcionamiento

Tarea 7 – Implementar manejo de errores detallado

Debes validar:

- id inexistentes
- JSON mal formado
- Campos obligatorios ausentes
- Métodos HTTP incorrectos
- Intentos de activar un registro ya activo
- Intentos de desactivar un registro ya desactivado

Debes responder con mensajes claros:

- "Registro no encontrado"
- "Datos incompletos"
- "Error procesando la solicitud"

Tarea 8 – Implementar control básico de excepciones en PHP puro

Ejemplo:

```
try {
  // CRUD operations
} catch (Exception $e) {
  jsonResponse(500, "Error interno del servidor", ["detalle" => $e->getMessage()]);
}
```

3.1.5 – Optimiza el framework CRUD para rendimiento y eficiencia

Tarea 9 – Reutilizar funciones comunes

Debes crear funciones reutilizables para:

- Leer datos
- Guardar datos
- Validar id
- Manejar estados (activo / inactivo)
- Construir respuestas

Tarea 10 – Minimizar operaciones de lectura/escritura

Ejemplo:

- Cargar los datos solo una vez por petición
- Reutilizar la carga en todas las operaciones
- Escribir solo cuando sea necesario

Tarea 11 – Generar un archivo JSON bien organizado y liviano

Para optimización:

- Usar nombres de campos claros
- Evitar datos redundantes
- Mantener el archivo ordenado para lecturas más rápidas

.

Formato y Condiciones de Entrega General

1. **El código php (no se aceptará otro lenguaje)**
2. **Proyecto comprimido en formato zip (no se aceptará otro formato)**
3. **Deberá diferenciar entrega de frontend como de backend.**
4. **Formato nombre Individual: eva3_apellido_nombre.zip**