EjercicioRepaso_Modificado

🔄 Proyecto Modificado - Sistema de Autenticación

Este es una copia modificada de `EjercicioRepaso` con un **nuevo sistema de autenticación** basado en el modelo de **ProyectoAndrea**.

📋 Cambios Realizados

✅ 1. Nueva Tabla `usuarios_api`

Se agregó una tabla para gestionar usuarios de autenticación de la API:

```
CREATE TABLE usuarios_api (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user VARCHAR(50) NOT NULL UNIQUE,
    pass VARCHAR(255) NOT NULL,
    nombre_completo VARCHAR(100),
    fecha_creacion DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

✅ 2. Modelo `GestorUsuarios`

Nuevo modelo para gestionar usuarios API con métodos:

- `obtenerPorNombre($user)` - Busca un usuario
- `validarCredenciales($user, $pass)` - Valida usuario y contraseña
- `crearUsuario($user, $pass, $nombre)` - Crea un nuevo usuario
- `actualizarPassword($user, $pass)` - Actualiza contraseña
- `listarUsuarios()` - Lista todos los usuarios

Ubicación: `app/modelos/GestorUsuarios.php`

✅ 3. Modificado `requireBasicAuth()` en Controlador

ANTES (sistema antiguo):

```
protected function requireBasicAuth() {
    $user = $_SERVER['PHP_AUTH_USER'] ?? null;
    $pass = $_SERVER['PHP_AUTH_PW'] ?? null;

    if ($user !== API_BASIC_USER || $pass !== API_BASIC_PASS) {
        header('WWW-Authenticate: Basic realm="API Restaurante"');
        $this->jsonResponse(['error' => 'Unauthorized'], 401);
    }
}
```

AHORA (sistema ProyectoAndrea):

```
protected function requireBasicAuth() {
    $user = $_SERVER['PHP_AUTH_USER'] ?? null;
    $pass = $_SERVER['PHP_AUTH_PW'] ?? null;

    if (!$user || !$pass) {
        header('WWW-Authenticate: Basic realm="API Restaurante"');
        $this->jsonResponse(['error' => 'Unauthorized'], 401);
    }

    // Buscar usuario en BD
    $modeloUsuario = $this->modelo('GestorUsuarios');
    $usuarioBD = $modeloUsuario::obtenerPorNombre($user);

    // Verificar contraseña con hash
    if ($usuarioBD && password_verify($pass, $usuarioBD['pass'])) {
        return; // Autenticación exitosa
    }

    // Credenciales incorrectas
    header('WWW-Authenticate: Basic realm="API Restaurante"');
    $this->jsonResponse([
        'error' => 'Acceso denegado: Credenciales incorrectas'
    ], 401);
}
```

✅ 4. Eliminadas Constantes de Config

Se eliminaron estas líneas de `config.php`:

```
// Ya NO se usan
define('API_BASIC_USER', 'admin');
define('API_BASIC_PASS', 'admin123');
```

🔐 Usuarios Disponibles

Usuario	Contraseña	Descripción
admin	admin123	Administrador API
profesor	1234	Profesor Test
usuario	password	Usuario Normal

🚀 Cómo Usar

1. Instalar Base de Datos

```
mysql -u root -p < bd/bd.sql
```

2. Probar la Autenticación

Con cURL:

```
# Usuario correcto
curl -u admin:admin123 http://localhost/EjercicioRepaso_Modificado/api-server/controladorproductos/productos

# Usuario incorrecto (debe fallar)
curl -u admin:wrong_pass http://localhost/EjercicioRepaso_Modificado/api-server/controladorproductos/productos
```

Con Postman:

1. En la pestaña **Authorization**
2. Selecciona **Basic Auth**
3. Ingresa:
   - Username: `admin`
   - Password: `admin123`

📊 Comparación: Sistema Antiguo vs Nuevo

Aspecto	Sistema Antiguo	Sistema Nuevo (ProyectoAndrea)
Almacenamiento	Constantes en `config.php`	Base de datos `usuarios_api`
Seguridad	Contraseña en texto plano	Hash con `password_hash()`
Flexibilidad	Solo 1 usuario hardcodeado	Múltiples usuarios en BD
Gestión	Editar código fuente	SQL o scripts PHP
Verificación	Comparación de strings	`password_verify()` contra hash

🎯 Ventajas del Nuevo Sistema

✅ **Más seguro**: Contraseñas hasheadas, nunca en texto plano
✅ **Más flexible**: Agregar usuarios sin modificar código
✅ **Más escalable**: Soporta múltiples usuarios fácilmente
✅ **Mejor práctica**: Separación de código y configuración
✅ **Auditable**: Puedes agregar logs de acceso por usuario

🛠️ Scripts Útiles

Generar Hash para Nuevo Usuario

```
php generar_hash_usuarios.php
```

Agregar Usuario Manualmente

```
INSERT INTO usuarios_api (user, pass, nombre_completo)
VALUES ('nuevo_usuario', '$2y$10$...hash...', 'Nombre Usuario');
```

Ver Usuarios Actuales

```
SELECT id, user, nombre_completo, fecha_creacion
FROM usuarios_api;
```

📝 Notas Importantes

1. **Dos Capas de Seguridad**:
   - Basic Auth (usuarios_api) → Acceso a la API
   - Login de Restaurantes → Funcionalidad específica
2. **Compatible con Sistema Original**:
   - El método `requireAuth()` sigue funcionando igual (sesiones de restaurantes)
   - Solo cambió `requireBasicAuth()` para usar BD
3. **Hashes Únicos**:
   - Cada vez que generas un hash con `password_hash()`, es diferente
   - Esto es NORMAL y seguro, `password_verify()` sigue funcionando

🎓 Aprendizaje

Este cambio enseña:

- **Hashing de contraseñas** (`password_hash`, `password_verify`)
- **Autenticación en BD** vs constantes hardcodeadas
- **Separación de concerns** (configuración vs lógica)
- **Seguridad en APIs REST**
- **Refactorización de código** manteniendo compatibilidad

Fecha de Modificación: 2 de Febrero de 2026
Basado en: ProyectoAndrea (sistema de autenticación)