

Stack sugerido para todos: XAMPP/WAMP, PHP ≥8.1, MySQL/MariaDB, mysqli o PDO, sesiones nativas de PHP, HTML+CSS vanilla (o Bootstrap si lo desean), estructura /public + /includes + /db.

1) TaskMate – Gestor de tareas

- **Idea:** lista de tareas personales con estado.
- **Mínimo:** login, crear/editar/borrar tarea, marcar “hecha”, filtro por estado.
- **Tablas:**
 - users(id, nombre, email, pass_hash, created_at)
 - tasks(id, user_id, titulo, descripcion, estado ENUM('pendiente','hecha'), fecha_limite, created_at)
- **Endpoints:** /login.php, /register.php, /tasks.php (list/create), /task_edit.php, /task_delete.php, /logout.php.

2) Notelt – Bloc de notas con categorías

- **Mínimo:** CRUD de notas, categorías, búsqueda por título.
- **Tablas:**
 - users(...)
 - categories(id, user_id, nombre)
 - notes(id, user_id, category_id, titulo, contenido, created_at, updated_at)
- **Endpoints:** /notes.php, /note_edit.php, /note_delete.php, /categories.php.

3) BudgetEasy – Ingresos/Egresos

- **Mínimo:** registrar movimientos (ingreso/egreso), ver saldo, filtrar por rango de fechas.
- **Tablas:**
 - users(...)
 - movements(id, user_id, tipo ENUM('ingreso','egreso'), concepto, monto DECIMAL(10,2), fecha, created_at)

- **Endpoints:** /movements.php, /movement_edit.php, /reports.php?from&to.

4) QuizTime – Cuestionarios simples

- **Mínimo:** crear cuestionario, preguntas Opción Múltiple, presentar y calificar.
- **Tablas:**
 - users(...)
 - quizzes(id, user_id, titulo, created_at)
 - questions(id, quiz_id, texto)
 - options(id, question_id, texto, es_correcta TINYINT)
 - attempts(id, user_id, quiz_id, puntaje, created_at)
- **Endpoints:** /quiz_new.php, /quiz_edit.php, /quiz_take.php?quiz_id, /quiz_result.php.

5) MyBooks – Préstamos de libros

- **Mínimo:** catálogo, registro de préstamos y devoluciones.
- **Tablas:**
 - users(...)
 - books(id, titulo, autor, isbn, stock)
 - loans(id, user_id, book_id, fecha_prestamo, fecha_devolucion, devuelto TINYINT)
- **Endpoints:** /books.php, /loan_new.php, /loan_return.php, /loans_history.php.

6) FitLog – Diario de ejercicio

- **Mínimo:** registrar sesiones (tipo, duración, distancia), resumen semanal.
- **Tablas:**
 - users(...)
 - workouts(id, user_id, tipo, duracion_min INT, distancia_km DECIMAL(5,2), fecha, notas)
- **Endpoints:** /workouts.php, /workout_edit.php, /stats.php?week.

7) RecipeBook – Recetario

- **Mínimo:** CRUD recetas, búsqueda por ingrediente.
- **Tablas:**
 - users(...)
 - recipes(id, user_id, titulo, instrucciones, created_at)
 - ingredients(id, nombre UNIQUE)
 - recipe_ingredient(recipe_id, ingredient_id) (*tabla puente*)
- **Endpoints:** /recipes.php, /recipe_edit.php, /search.php?ingredient=....

8) DailyMood – Registro de ánimo

- **Mínimo:** una entrada al día (feliz/neutral/triste), notas, gráfico simple (tabla HTML basta).
- **Tablas:**
 - users(...)
 - moods(id, user_id, fecha UNIQUE(user_id, fecha), estado ENUM('feliz','neutral','triste'), nota)
- **Endpoints:** /mood_today.php, /mood_history.php?month.

9) SimpleShop – Inventario mini

- **Mínimo:** CRUD productos, entradas/salidas, existencia actual.
- **Tablas:**
 - users(...)
 - products(id, nombre, sku UNIQUE, existencias INT DEFAULT 0)
 - kardex(id, product_id, tipo ENUM('entrada','salida'), cantidad INT, fecha)
- **Endpoints:** /products.php, /stock_move.php, /kardex.php?product_id.

10) ClassAssist – Asistencia con códigos

- **Mínimo:** grupos, alumnos, pases de lista por fecha; reporte de % asistencia.
- **Tablas:**
 - users(...) (*docentes*)

- groups(id, user_id, nombre)
- students(id, group_id, nombre, matricula)
- attendances(id, student_id, fecha, estado
ENUM('asistio','falto','retardo'))
- **Endpoints:** /groups.php, /students.php?group_id,
/attendance.php?group_id&date, /report.php?group_id.

Reglas comunes (para todos)

- **Autenticación:** register.php, login.php, logout.php con password_hash() / password_verify(), sesiones en \$_SESSION.
- **Conexión DB:** db.php con PDO y ATTR_ERRMODE => EXCEPTION.
- **Validación:** filter_input() / htmlspecialchars() para XSS, consultas preparadas para SQLi.
- **Estructura mínima de vistas:** lista → formulario crear/editar → borrar con token CSRF simple.
- **Reporte:** 1 página de **resumen** con al menos 2 consultas (agregaciones o filtros por fecha).

“Plus” opcionales (suben puntos sin complicarse)

- Exportar a **CSV** (descarga) desde el listado.
- **Búsqueda** por palabra clave en cada módulo.
- **Paginación** simple (LIMIT/OFFSET) en listados grandes.
- **Roles** básicos (admin/usuario) en 1–2 proyectos (ej. SimpleShop, ClassAssist).

Rúbrica rápida para estos proyectos “acreditación”

- **Funcionalidad CRUD (40%):** alta si todo crea/lee/actualiza/borra sin errores.
- **Uso correcto de PHP+SQL (30%):** consultas preparadas, estructura de tablas coherente.
- **Autenticación y sesiones (15%):** login/registro funcionando, restricciones por usuario.

- **Cuidado básico de UI/UX (10%):** formularios claros, mensajes de validación.
- **Reporte/estadística simple (5%):** conteos, totales o filtro por periodo.