🍽️ Aprendizaje sobre el Proyecto Voltereta

Programación

1. Elementos fundamentales del código

- Variables (`let nombre`), constantes (`const PI`)
- Operadores: `+`, `-`, `*`, `/`
- Tipos: number, string, boolean, objeto

```
const PI = 3.14;
let nombre = "Juan";
let edad = 20;
let activo = true;
let usuario = { id: 1, nombre: "Ana" };
```

2. Estructuras de control

- Selección: `if`, `else if`, `else`
- Repetición: `for`, `while`
- Saltos: `break`, `continue`

```
if (edad > 18) {
  console.log("Mayor de edad");
} else {
  console.log("Menor de edad");
}
```

3. Control de excepciones y gestión de errores

- Uso de `try-catch` para capturar errores
- Middleware `errorHandler.js` en Express para gestión centralizada

```
try {
  // Código que puede fallar
} catch (error) {
  console.error(error);
}
```

4. Documentación del código

- Comentarios `//` y docstrings en funciones
- Archivos markdown para documentación

5. Paradigma aplicado

- Programación estructurada y orientada a objetos (OOP) en backend
- Facilita organización y reutilización

6. Clases y objetos principales

- Objetos: `usuario`, `empleado`, `pedido`, etc.
- Clases posibles en backend: `Empleado`, `Pedido`

7. Conceptos avanzados

- No herencia/polimorfismo explícito
- Interfaces implícitas y funciones reutilizables

8. Gestión de información y archivos

- Base de datos MySQL
- Archivos: logs, imágenes de perfil
- Interfaz web (HTML, CSS, JS)

9. Estructuras de datos utilizadas

- Arrays y objetos

```
let empleados = [
  { id: 1, nombre: "Ana" },
  { id: 2, nombre: "Luis" },
];
```

10. Técnicas avanzadas

- Expresiones regulares para validación
- Flujos de E/S para logs y archivos

Sistemas Informáticos

1. Características del hardware

- Desarrollo: PC/Mac, Intel/AMD, 8GB+ RAM
- Producción: VPS o máquina dedicada

2. Sistema operativo

- macOS y Windows en desarrollo
- Linux (Ubuntu) en producción

3. Configuración de redes

- Red local en desarrollo, pública en producción
- HTTP/HTTPS, firewall y autenticación

4. Copias de seguridad

- Backups periódicos de BD y archivos
- Scripts automáticos o herramientas del hosting

5. Integridad y seguridad de datos

- Contraseñas cifradas, validación de entradas, backups
- Accesos limitados por roles

6. Usuarios, permisos y accesos

- Usuarios y permisos en SO
- Solo admin modifica archivos críticos

7. Documentación técnica

- Archivos markdown y README
- Pasos de instalación y despliegue

Entornos de Desarrollo

1. Entorno de desarrollo (IDE)

- Visual Studio Code
- Extensiones: JavaScript, Node.js, MySQL

2. Automatización de tareas

- Scripts en `package.json` para servidor, tests, etc.

3. Control de versiones

- Git y GitHub
- Ramas para features y revisiones

4. Refactorización

- Mejoras periódicas, nombres claros, sin duplicidades

5. Documentación técnica

- Markdown (`README.md`, `aprendizaje.md`), comentarios

6. Diagramas

- Diagramas de clases/flujo (draw.io, plantUML)

Bases de Datos

1. Sistema gestor

- MySQL (robusto, fácil integración Node.js)

2. Modelo entidad-relación

- Tablas: usuarios, empleados, pedidos, etc.
- Relaciones: un pedido pertenece a un usuario

3. Vistas, procedimientos, disparadores

- Vistas y procedimientos para automatizar tareas (uso limitado)

4. Protección y recuperación de datos

- Backups y validaciones

Lenguajes de Marcas y Gestión de Información

1. Estructura de HTML

- Etiquetas semánticas (`<header>`, `<main>`, `<footer>`)
- Buenas prácticas para accesibilidad y SEO

2. Tecnologías frontend

- HTML, CSS, JavaScript (estándar web)

3. Interacción con el DOM

- JavaScript para modificar el DOM dinámicamente

4. Validación de HTML y CSS

- Validadores online y extensiones del IDE

5. Conversión de datos (XML, JSON)

- JSON para intercambio frontend-backend

6. Integración con sistemas de gestión

- Aplicación de gestión empresarial (restaurante)
- Gestión de empleados, pedidos, reservas, etc.

Proyecto Intermodular

1. Objetivo del software

- Gestionar restaurante: empleados, pedidos, reservas, menús

2. Necesidad o problema que soluciona

- Evita gestión manual, reduce errores, mejora eficiencia

3. Stack de tecnologías

- **Frontend:** HTML, CSS, JavaScript
- **Backend:** Node.js, Express
- **Base de datos:** MySQL
- **Control de versiones:** Git, GitHub

4. Desarrollo por versiones

- v1: Login y gestión básica de empleados
- v2: Gestión de pedidos y reservas
- v3: Reportes y estadísticas