# Propuesta de arquitectura del proyecto ABFB

*Descripción del caso:*
*El almacén bazar FreBor es una empresa instituida el año 2009, el cual buscaba ayudar a los estudiantes de las escuelas cercanas, los dueños de esta empresa al ser profesores con conscientes de lo que necesitan los estudiantes y en base a eso crearon esta empresa para ser de ayuda, a lo largo de 15 años esta empresa a crecido de forma rápida y a llegado a a la frontera digital, al no estar en ese ámbito y luego de una propuesta se nos a asignado el trabajo de realizarles un sistema que supere la barrera digital incertando la empresa a este mundo digital*

**Lenguaje de programación:**

- backend y frontend: visual studio code con lenguaje php, mysql, javascript y sql
- frontend: HTML y java

**Base de datos:**

- mysql

**Arquitectura MVT (Modelo-Vista-Template):**

**Descripción y beneficios**

- Modelo: lógica de negocios y gestión de datos.
- Vista: interfaz del usuario.
- Template: estructura visual del frontend.

**Beneficios**:

- Separación clara de las responsabilidades de los miembros del proyecto.
- Facilidad al momento de la escalabilidad y su mantenimiento.

**Monolito**:
**Beneficios**:

- Desarrollo del proyecto menos complejo y su despliegue es más sencillo.

**Microservicios**:
**Beneficios empresa**:

- Aumento de ventas y consumo de productos
- Mejor control de inventario productos y proveedores

**Beneficios usuario**:

- Acceso rápido y simple a los materiales
- Opción de programar compras para no olvidar materiales

**Microfronted:**
**Beneficios:**

- Desarrollo y despliegue independiente de los módulos frontend.

- Mejora la escalabilidad y el mantenimiento.

**bff backend for frontend:**
**Beneficios:**

- Optimización de las solicitudes frontend
- Mayor flexibilidad y rendimiento
- Fácil de integrar
- Mejor accesibilidad
- Facilidad de aprendizaje
- Servidor web incorporado
- Buena documentación

**Python:**
**Beneficios:**

- Interactividad en tiempo real
- Asincronía
- Facilidad de integración
- Ecosistema de desarrollo
- Soporte de navegadores
- Soporte para programación orientada a objetos
- Vertibilidad

**Patrones de diseño a utilizar:**
Se implementarán los siguientes patrones de diseño:
**Singleton:**
Para gestionar conexiones a la base de datos o servicios externos de manera eficiente.

**Repository:**
Para encapsular la lógica de acceso a la base de datos.

**Decorator:**
Para agregar funcionalidades adicionales a clases existentes por ejemplo para manejar el almacenamiento en cache.

**Strategy:**
Para definir algoritmos como los de recomendación de manera que sea intercambiable.

**Beneficios de la propuesta:**

**escalabilidad:**

Arquitectura modular y opción de migrar hacia micro servicios para futuras escalabilidad.

**separacion de responsabilidades:**

Modelo MVT y patrones de diseño para una clara separación y mantenimiento simplificado.

**optimizacion de rendimiendo:**

Uso de Javascript para una experiencia de usuario rápida y eficiente.

**diversidad de productos:**

Integración con marcas reconocidas para ofrecer una amplia variedad y calidad de productos.

**presencia digital:**

Inclusión de una plataforma en línea para compras rápidas y cómodas.