1. **Considerando a complexidade das operações logísticas,financeira e gerenciamento de produtos para o mundo marketplace, qual seria a arquitetura mais adequada para um sistema de microsserviços voltado para a gestão das operações mencionadas , levando em conta requisitos como escalabilidade, resiliência e integração de dados em tempo real?**

Resposta:

1. **Arquitetura de Microsserviços:**

*Serviços Independentes e Coesos*, pois isso facilita a escalabilidade e a manutenção, além de permitir o desenvolvimento ágil.

1. **Escalabilidade:**

***Containerização (Docker),*** pois orquestrados pelo **Kubernetes**, permite escalar os serviços de forma dinâmica, adicionando instâncias conforme a carga aumenta.Alem disso podemos o **Microsserviços** podem ser replicados horizontalmente para aumentar a capacidade de processamento sem necessidade de mudanças na infraestrutura

1. **Resiliência e Alta Disponibilidade:**

***Redundância***: A infraestrutura deve ter failover configurado, com backups automáticos e distribuição de carga entre múltiplas instâncias e regiões para garantir alta disponibilidade.

1. **Integração de Dados:**

**Apache:** comuniquem de forma assíncrona por meio de tópicos, facilitando a troca de dados em tempo real, como atualizações de inventário ou transações financeiras.Exemplo do **Apache Kafka ou Apache Storm** para processar grande volumes de dados

1. **Segurança:**

***Autenticação e Autorização***: Utilizar **OAuth 2.0** ou **JWT** (JSON Web Tokens) para garantir a segurança de APIs e garantir o controle de acesso entre microsserviços.

1. **Suponha que você precise integrar uma API REST externa em um projeto PHP. A API fornece informações sobre o clima de uma determinada cidade. Como você abordaria essa integração? Descreva os passos necessários para fazer uma solicitação à API, receber os dados de resposta e armazená-los em um banco de dados MySQL local para uso posterior.Considere questões como autenticação, tratamento de respostas e cronograma de atualização dos dados.**

**Resposta: Citarei alguns passos que normalmente sigo para inicia esse projetos, levando o modelo com projeto em php com Laravel**

1. **Analise e Obter os dados:**

**Documentação da Api:** Geralmente os microserviços desmobilizar uma documentação dos dados e os valores

**Autenticação:** verificação de uso de chave de acesso ou tokens

**Exemplos:** Urls usadas e autorizadas como modelos assim como a linguagem padrão permitidos

1. **Inicio do Projeto em Laravel**

**Analisa qual versão o Laravel permitido e compatível**

**Instalar Dependências:** se na documentação tive a biblioteca em laravel compatível

**Criar Arquivos de Configuração:** se projeto necessita uma configurançao automática devemos cria arquivo de configuração que ja vem com autenticação e modelo de dados configurado

**Armazenar os Dados:** como é necessário fica no banco Mysql, criaria modelos migration que configurações de no arquivo “ .env” para formata as tabelas

1. **Configurar Atualizações Automáticas**

**Script de Atualização:** Criar um script que atualiza os dados regularmente

**Agendamento com Cron:** Configurar uma tarefa cron para executar o script em intervalos regulares:

1. **Tratamentos de Erros:**

**Verificar códigos de erro HTTP** (exemplo: 401 Unauthorized, 429 Too Many Requests).

**Implementar logs de erros .**

**Respeitar o limite de requisições da API.** Se necessário, implementar um sistema de cache

**Adaptar a lógica para suportar alterações no formato da API ou parâmetros adicionais**.

1. **Segurança:**

**Acesso Restrito**: Configure as permissões no servidor para que apenas os usuários necessários possam acessar os arquivos de configuração.

**HTTPS** ou **Certificados SSL/TLS**: protege os dados em trânsito contra ataques de interceptação

***Autenticação e Autorização***: Utilizar **OAuth 2.0** ou **JWT** (JSON Web Tokens) para garantir a segurança de APIs e garantir o controle de acesso entre microsserviços.

1. **Quais são as melhores práticas para garantir um desenvolvimento de software seguro em PHP?**

**Resposta: Com relação a segurança do código PHP acredito em alguns hábitos e portos que irei cita:**

1. **Validação e Saneamento de Entrada** : Sempre limpe e valide os dados recebidos de usuários, mesmo sendo por formulários ou arquivos externos
2. **Proteção contra SQL Injection:** principalmente para evitar dados maliciosos no banco de dados e evitando interpolação direta de variáveis em consultas SQL
3. **Autenticação e Senhas:** Nunca armazene senhas em texto puro. Use password\_hash para armazenar e password\_verify para verificar.
4. **Autenticação Multi-Fator (MFA)**: Muito usado aplicação de rede social conhecido autenticação de duas etapas
5. **Gerenciamento de Erros:** Nunca exiba mensagens de erro detalhadas em produção. Configure o PHP para logar erros e mostrar mensagens genéricas.
6. **Atualização e Versionamento do código:** Sempre fica atento com novas versões do php e criar mecanismo de versionamento do código como git