



CICLO 4a

[FORMACIÓN POR CICLOS]

Desarrollo de **APLICACIONES WEB**

Microservicios

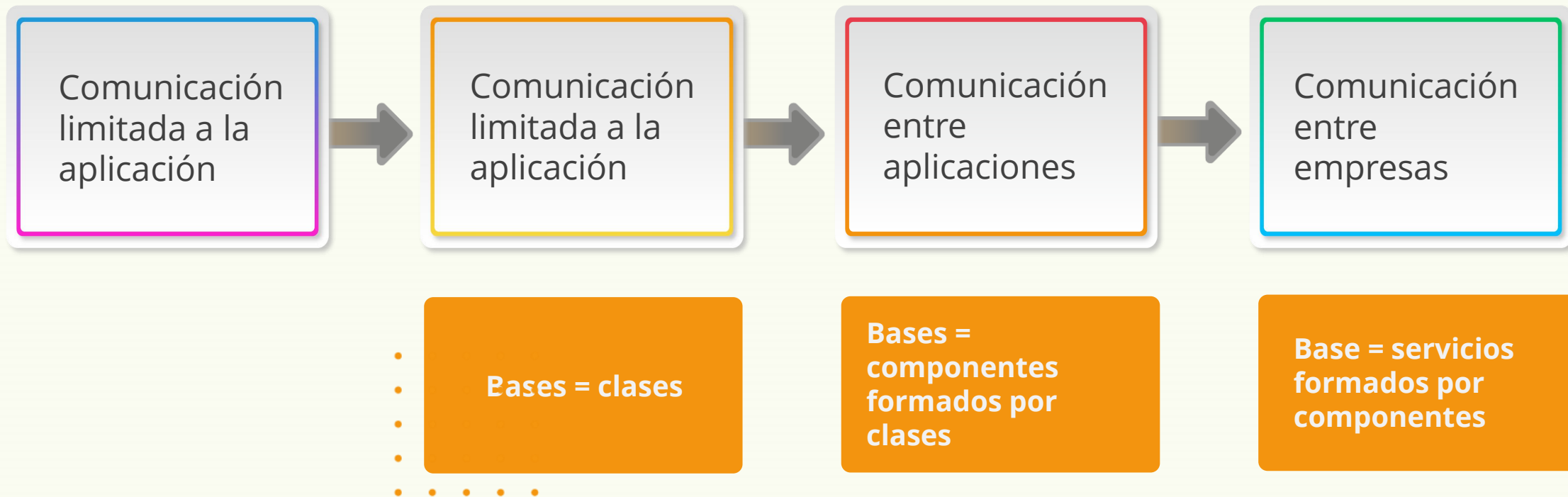


**UNIVERSIDAD
DE ANTIOQUIA**

Facultad de Ingeniería

Microservicios

Es una arquitectura orientada a servicios compuesta por pequeños componentes, cada uno corriendo en sus propios procesos y comunicándose mediante mecanismos livianos, por lo general un API de HTTP



Arquitectura monolítica vs. microservicios





Microservicios: características

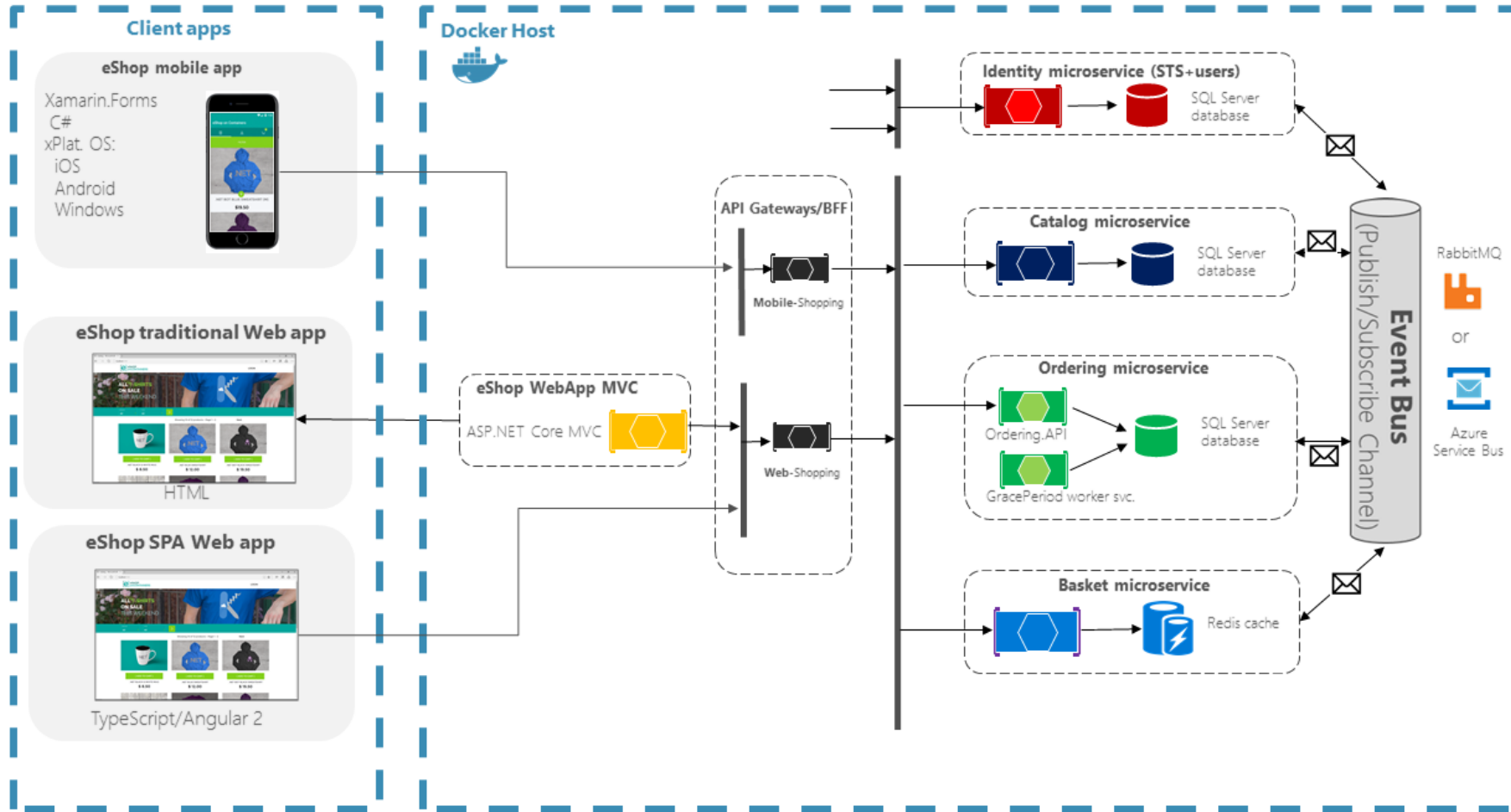
- Se puede implementar cualquier lenguaje en microservicios de manera independiente.
- Los despliegues son rápidos y seguros.
- Ahorro de *hardware* (tamaños diferentes).
- Mantenimiento efectivo (versiones y cambios).
- Escalable de manera independiente.



Estructura de microservicios

eShopOnContainers reference application

(Development environment architecture)





Interacción entre microservicios

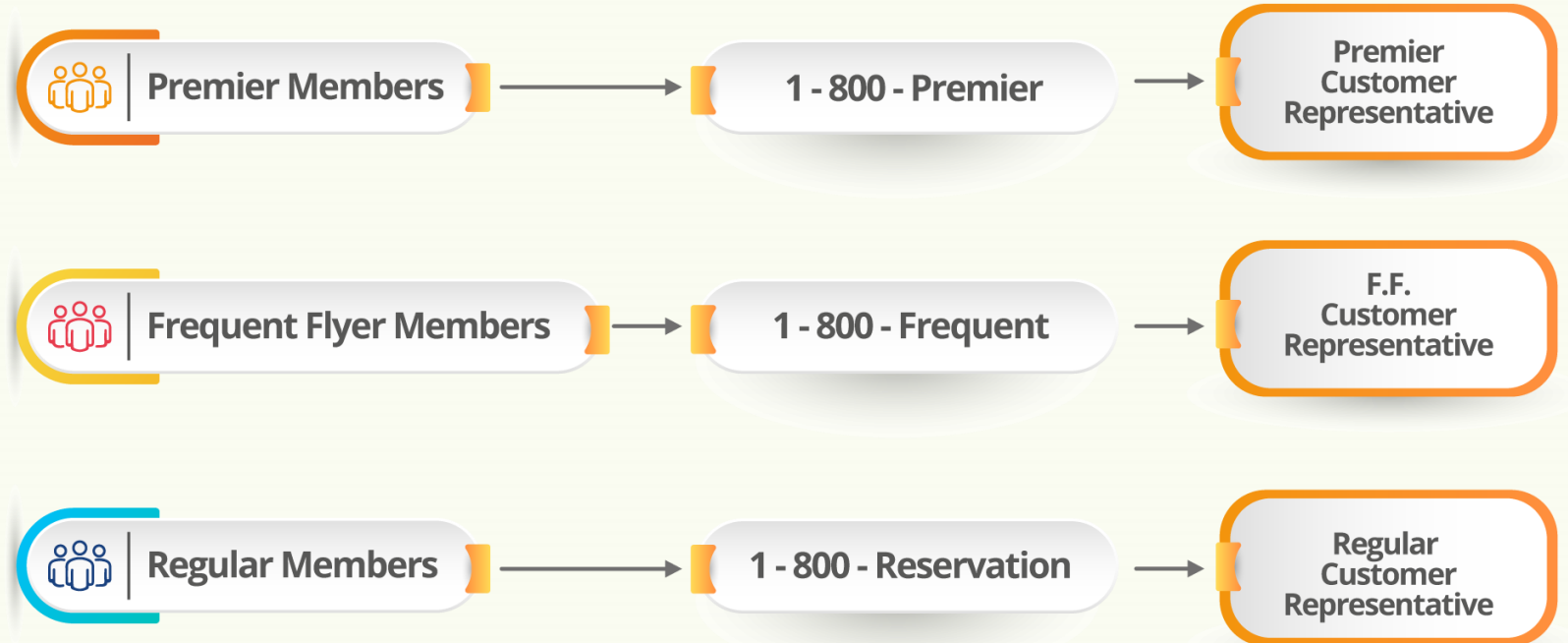
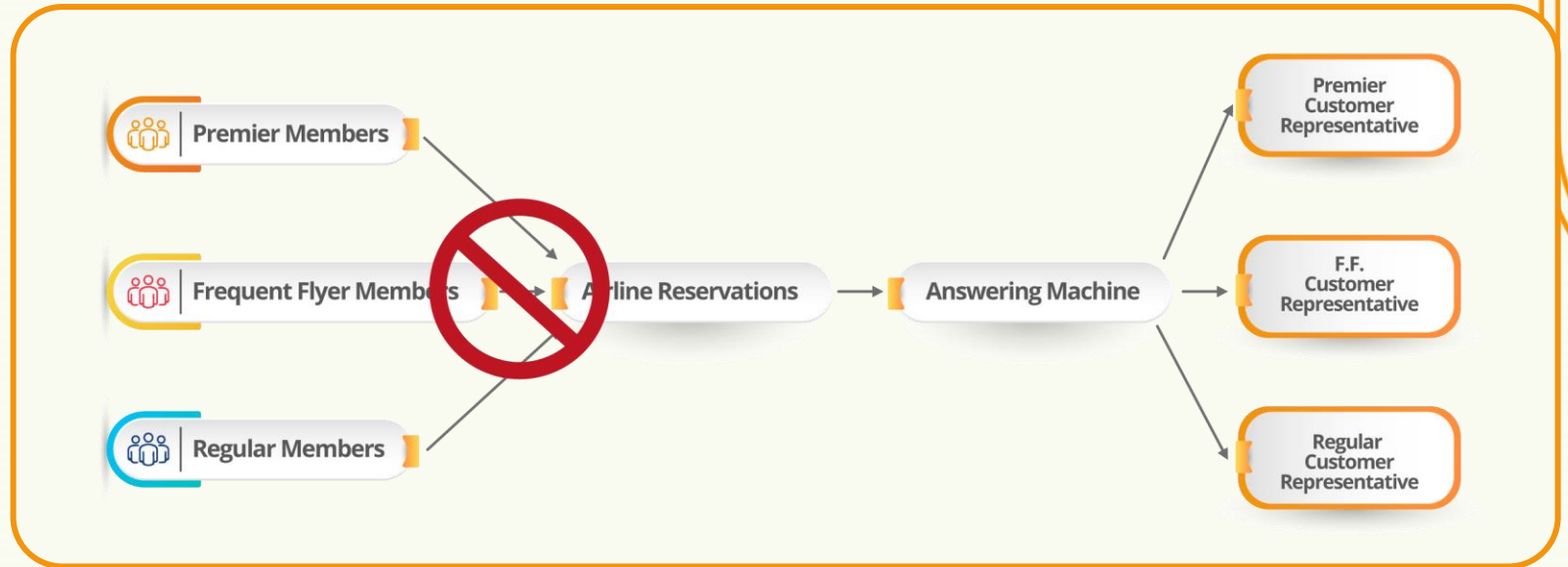
- La intención de los microservicios es el bajo acoplamiento, por lo que se sugiere que la comunicación entre ellos sea de manera asíncrona.
- Lo habitual es que su comunicación se realice por un protocolo de comunicación como HTTP, AMQP o TCP. Esto, en función de la naturaleza del servicio.
- Normalmente los microservicios que componen una aplicación de un externo a otro se establecen mediante comunicaciones REST en lugar de protocolos complejos.



REST

Es una arquitectura estándar para construir y comunicarse con servicios web

Learning REST by example. Roger L. Costello, Timothy D. Kehoe



REST

Exige que los recursos de la web se representen en formato de texto, como JSON, HTML o XML.

Dado que fue construido para aprovechar el protocolo HTTP, trabaja con los métodos:

- GET
- POST
- PUT



Tipos de solicitudes HTTP

POST

Se utiliza para enviar datos, y normalmente para *crear* nuevas entidades o editar entidades ya existentes

GET

Se usa para solicitar datos del servidor, y generalmente para *leer* datos

PUT

Se usa para reemplazar completamente el recurso con el recurso enviado, y generalmente para *actualizar* datos

DELETE

Se utiliza para *eliminar* una entidad del servidor





Node Js

Entorno de tiempo de ejecución multiplataforma de código abierto que permite crear aplicaciones en JavaScript sin necesidad de un navegador web.

- **Node** optimiza el rendimiento y la escalabilidad.
- Se puede **instalar en cualquier SO**.
- Tiene herramientas que permiten comunicarse directamente con el SO (archivos, red, etc.).
- Ejecuta JS en modo servidor.
- Backend.





Express

Es una de las librerías de HTTP más populares para Node.js, ya que node.js no es tan amigable para el desarrollo con API. Express lo simplifica, pues abstrae el modelo necesario para configurar un servidor



JSON

- Notación de objetos en JavaScript. Es un formato ligero de intercambio de datos.
- Se basa en una colección de pares de nombre/valor.
- Es una estructura de datos que prácticamente manejan todos los lenguajes.

```
{
  "books": [
    {
      "isbn": "9781593279509",
      "title": "Eloquent JavaScript, Third Edition",
      "subtitle": "A Modern Introduction to
Programming",
      "author": "Marijn Haverbeke",
      "published": "2018-12-04T00:00:00.000Z",
      "publisher": "No Starch Press",
      "pages": 472,
      "description": "JavaScript lies at the heart of
almost every modern web application, from social apps
like Twitter to browser-based game frameworks like Phaser
and Babylon. Though simple for beginners to pick up and
play with, JavaScript is a flexible, complex language
that you can use to build full-scale applications.",
      "website": "http://eloquentjavascript.net/"
    }
  ]
}
```

