We create **innovative software products** that appeal to **global audiences**

AGENDA

01. Presentación
02. Evolución a Microservicios
03. Azure Service Fabric
04. Cómo crear un microservicio en Visual Studio
05. Siguientes pasos
06. Q&A

Presentación

Globant
we are ready

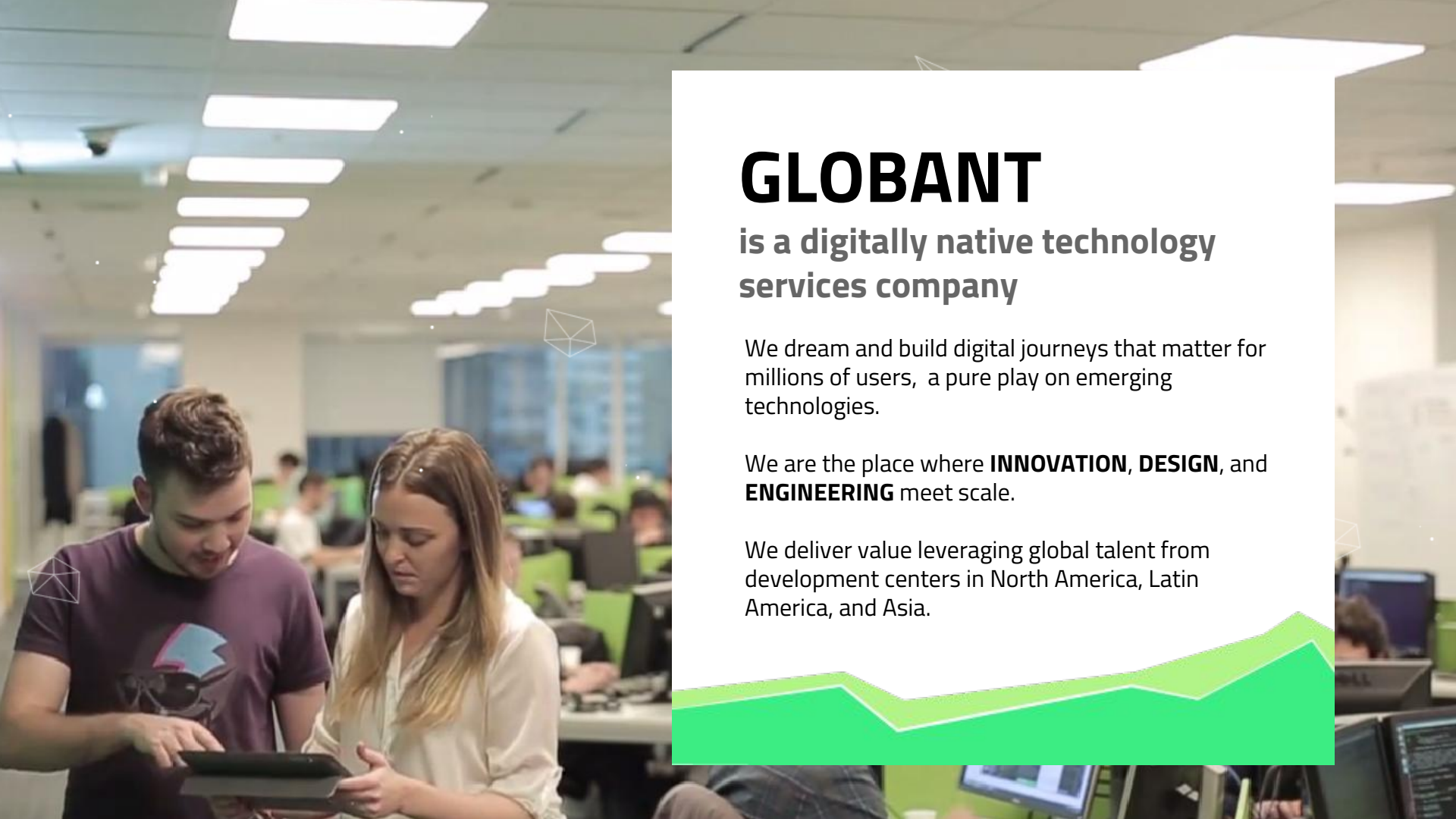
Bruno Guardia

Tech Director

Trabajando en desarrollo de software por más de 30 años, incluyendo sitios Web complejos y escalables desde 1996; enfocado en tecnologías Microsoft los últimos años pero codificando y manejando proyectos usando tecnologías Java y Unix con regularidad, for múltiples clientes, entregando más de 200 proyectos a la fecha.

Experiencia / Educación

- 30 Certificaciones Microsoft (45 exámenes):
  - Microsoft Certified Master, SQL Server
  - MCPD .Net
- Ingeniero en Sistemas Computacionales
- Maestro en Ciencias Computacionales con especialidad en Inteligencia Artificial
- Historia laboral:
  - 2016 - Globant
  - 2015-2016 Software Next Door
  - 2014-2015 EDX Solutions / 360 TXP
  - 2013-2014 Starbucks Corp, Seattle (CW)
  - 2007-2010 Microsoft Corp, Redmond
  - 1998-2007 Sinergia-Web
  - 1995-1997 Grupo Siglo
  - 1994-2005 ITESM CCM
  - 1992-1995 Programas, Ingeniería y Computadoras



GLOBANT

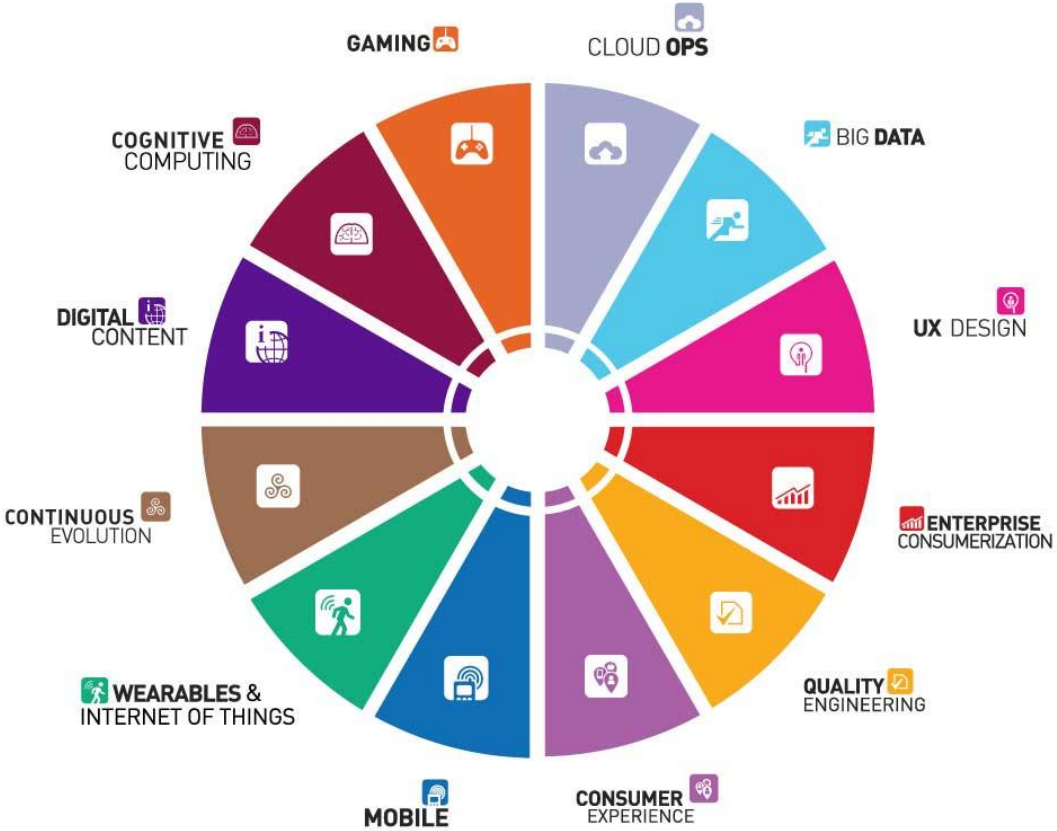
is a digitally native technology services company

We dream and build digital journeys that matter for millions of users, a pure play on emerging technologies.

We are the place where **INNOVATION**, **DESIGN**, and **ENGINEERING** meet scale.

We deliver value leveraging global talent from development centers in North America, Latin America, and Asia.

OUR STUDIOS - Deep expertise



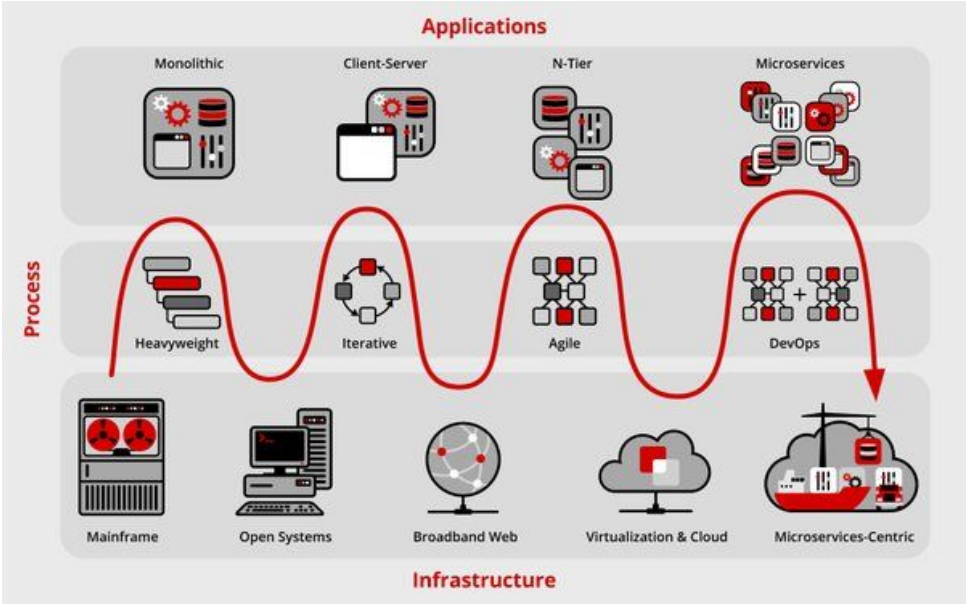
Globant
we are ready

Evolución a Microservicios

Globant
we are ready

EVOLUCIÓN A MICROSERVICIOS

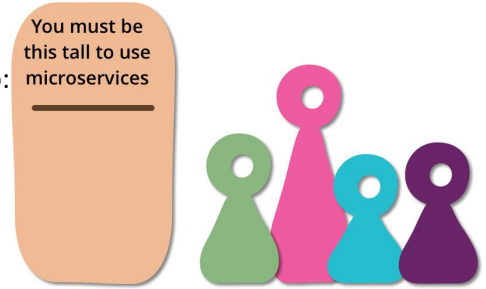
- 1970s Mainframes - Cobol
- PCs - 1980s
- Cliente Servidor - 1990s
- 3 capas - 1995s
- N capas - 2000s
- SOA - 2005s
- Microservicios, reactive 2010s



Globant
we are ready

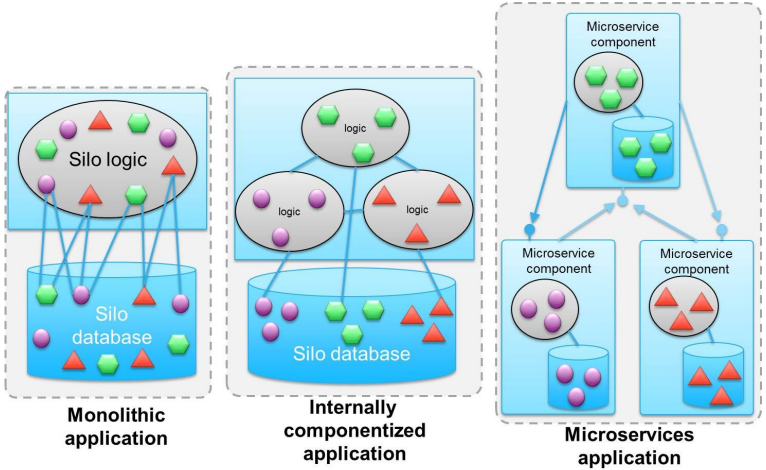
PRERREQUISITOS

Según Martin Fowler, para hablar de microservicios necesitaremos tener de antemano:



1) Provisionamiento rápido
   a) Infraestructura de auto servicio
   b) Bajo demanda
2) Monitoreo básico - Dev & Ops
3) Despliegue rápido de aplicaciones

https://martinfowler.com/bliki/MicroservicePrerequisites.html



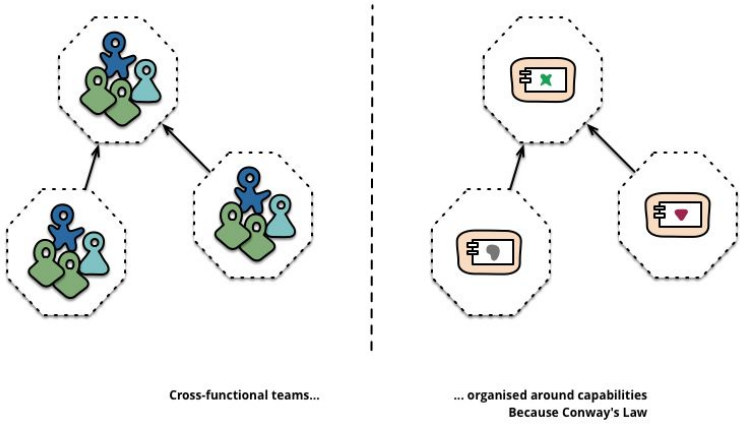
https://www.ibm.com/developerworks/websphere/library/techarticles/1601_clark-trs/1601_clark.html

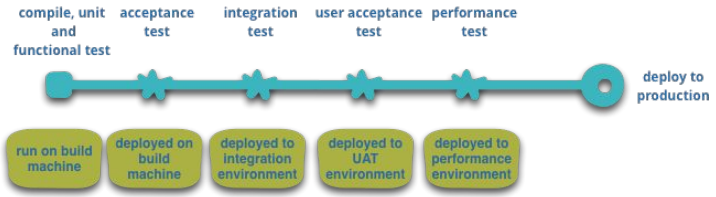
Globant
we are ready

CARACTERÍSTICAS

También de acuerdo a Martin Fowler, los microservicios se caracterizan por:

1. Componentes construidos por medio de servicios
   a. Preferidos sobre librerías
   b. Despliegue independiente
2. Organizados alrededor de las capacidades de negocio
3. Productos, no proyectos
4. Puntos de entrada inteligentes; comunicación tonta
5. Permiten descentralización
   a. Gobierno / Arquitectura empresarial
   b. Administración de datos
6. Automatización de infraestructura
7. Diseñados para tolerancia a fallas
   a. Fallas intencionales en producción para probar resiliencia
8. Diseño evolutivo
   a. Agrupados por interdependencia de cambios





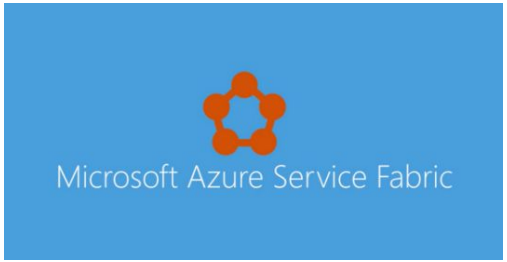
https://martinfowler.com/articles/microservices.html#CharacteristicsOfAMicroserviceArchitecture

Azure Service Fabric

Globant
we are ready

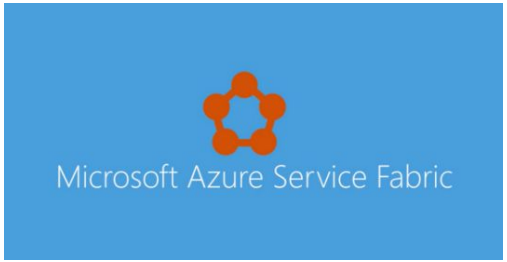
AZURE SERVICE FABRIC

- Plataforma para construir aplicaciones distribuidas
- Enfocada en
  - Escalabilidad
  - Confiabilidad
  - Facilidad de administración
- Servicios del sistema:
  - Failover Manager
  - Cluster manager
  - Naming service: Resolución
  - File store service
- Diversos tipos de servicios “fiables”
  - Stateless
  - Stateful
  - Actor
  - Guest executable
  - Stateless Web API
  - Contenedores



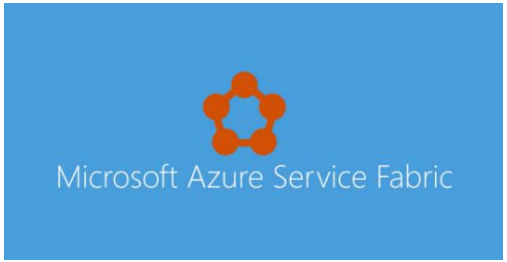
SERVICIOS FIABLES

- Stateless
  - Un punto de entrada
  - Múltiples réplicas / instancias escuchando
  - OnOpenAsync
  - OnCloseAsync
  - OnAbort
- Stateful
  - Roles: Primario/Secundario
  - OnRoleChange
  - Usan colecciones “fiables”
- Actor
  - Sin concurrencia
    - Sin bloqueos
    - Ejecución por turnos
  - Estado aislado



SERVICIOS FIABLES

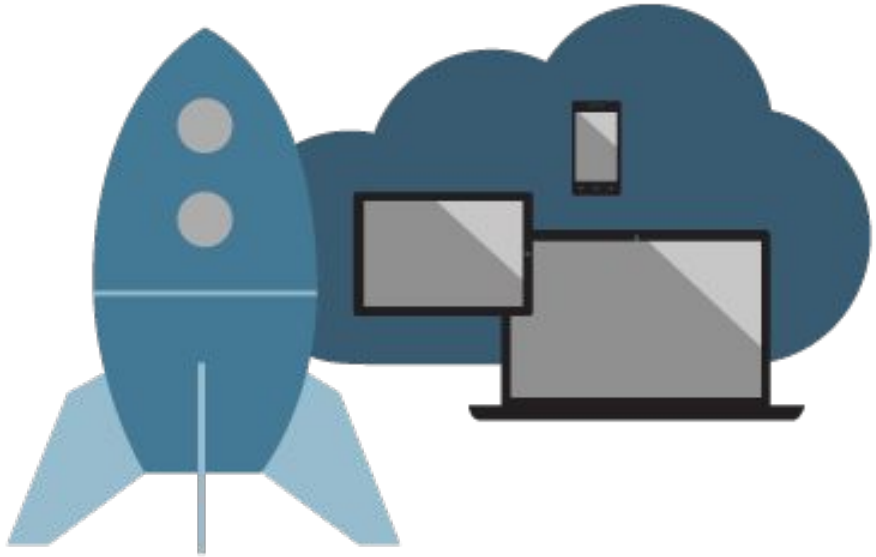
- Guest executable
  - Se tratan como servicios sin estado
  - Alta disponibilidad
  - Aislamiento / densidad
  - Genera copias - autocontenido código + configuración
- Web API
  - ASP.Net WebAPI 2, OWin self-host
- Contenedores
  - Docker
  - Windows 2016
    - IIS, legacy ASP.NET MVC



Cómo crear un servicio

Globant
we are ready

CREACIÓN DE MICROSERVICIO



MICROSERVICIO - STATEFUL

```
var myDictionary = await this.StateManager.GetOrAddAsync<IReliableDictionary<string, long>>("myDictionar

while (true)
{
    cancellationToken.ThrowIfCancellationRequested();

    using (var tx = this.StateManager.CreateTransaction())
    {
        var result = await myDictionary.TryGetValueAsync(tx, "Counter");

        ServiceEventSource.Current.ServiceMessage(this.Context, "Current Counter Value: {0}",
            result.HasValue ? result.Value.ToString() : "Value does not exist.");

        await myDictionary.AddOrUpdateAsync(tx, "Counter", 0, (key, value) => ++value);

        // If an exception is thrown before calling CommitAsync, the transaction aborts, all changes are
        // discarded, and nothing is saved to the secondary replicas.
        await tx.CommitAsync();
    }

    await Task.Delay(TimeSpan.FromSeconds(1), cancellationToken);
```

MICROSERVICIO - API (HOST)

```
/// <summary>
/// Optional override to create listeners (like tcp, http) for this service instance.
/// </summary>
/// <returns>The collection of listeners.</returns>
protected override IEnumerable<ServiceInstanceListener> CreateServiceInstanceListeners()
{
    return new ServiceInstanceListener[]
    {
        new ServiceInstanceListener(serviceContext => new OwinCommunicationListener(
            Startup.ConfigureApp, serviceContext,
            ServiceEventSource.Current, "ServiceEndpoint"))
    };
}
```

MICROSERVICIO - API (CONTROLLER)

```
namespace WebApi1.Controllers
{
    [ServiceRequestActionFilter]
    public class ValuesController : ApiController
    {
        // GET api/values
        public IEnumerable<string> Get()
        {
            return new string[] { "value1", "value2" };
        }

        // GET api/values/5
        public string Get(int id)
        {
            return "value";
        }
```

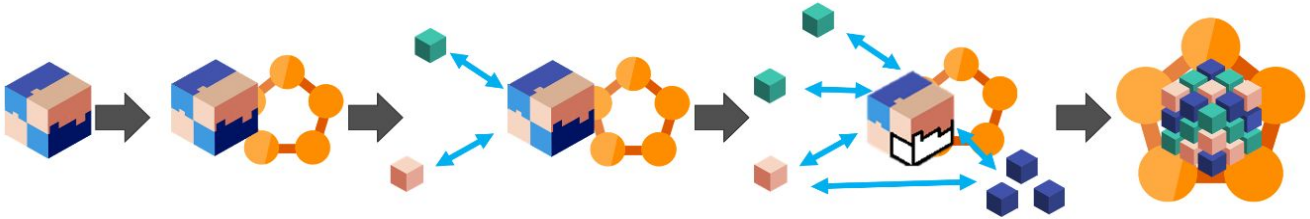
Siguientes Pasos

Globant
we are ready

SIGUIENTES PASOS

- Integración y publicación con Azure
- Definir una línea de despliegue
- Pruebas unitarias y de integración
- Pruebas de carga y estrés
- Definición de capacidad para cada servicio
- Migrar vs. reescribir
- Comunicación avanzada entre servicios
- Seguridad
- Auto documentación
- API Management

Migrating a traditional application



1) Traditional app
2) Hosted as guest executable or container in Service Fabric
3) With new microservices added alongisde
4) Breaking into microservices
5) Transformed into microservices

https://mva.microsoft.com/en-US/training-courses/Azure-Service-Fabric-Patterns-and-Practices-16925?l=mudwqISGD_6005167344


Thank
You!
Globant
we are ready