

Cápsulas informativas



Objetivo: publicar a los estudiantes por diferentes vías (página del Hackathon, grupos de comunicación, entre otros) la información de conocimiento base de manera progresiva de los temas a tratar en los retos

Recomendación para la secuencia de cápsulas

Para que los estudiantes construyan conocimiento progresivamente, el orden sugerido sería:

1. ¿Qué es Microsoft Azure?
2. ¿Qué es un Resource Group?
3. ¿Qué es Azure App Service?
4. ¿Qué es una API?
5. ¿Qué es una Arquitectura Frontend y Backend?
6. ¿Qué es DevOps?
7. ¿Qué es Infraestructura como Código (IaC)?
8. ¿Qué es Terraform?
9. ¿Qué es la Observabilidad?
10. ¿Qué son los Logs?
11. ¿Qué es KQL?
12. ¿Qué es Application Insights?
13. ¿Qué son los Logs, las Excepciones y los Errores 500?
14. ¿Qué es CORS y por qué existe?
15. ¿Qué es un Health Check?
16. ¿Qué son las Variables de Configuración?
17. Pensamiento de Diagnóstico Tecnológico.

Esta secuencia lleva al estudiante desde conceptos básicos de nube hasta observabilidad y resolución de problemas, que son los conocimientos que necesitará para entender un escenario moderno de aplicaciones sobre Azure.

Cloud → Aplicaciones → DevOps → IaC → Observabilidad → Logs → KQL → Diagnóstico → Corrección.

Cápsula 1

¿Qué es Microsoft Azure?

Concepto

Microsoft Azure es la plataforma de nube de Microsoft. Permite crear, desplegar y administrar aplicaciones sin necesidad de comprar ni mantener servidores físicos. Azure ofrece servicios de cómputo, almacenamiento, redes, inteligencia artificial, analítica y seguridad, accesibles a través de Internet.

¿Por qué es importante?

Actualmente miles de organizaciones utilizan Azure para ejecutar aplicaciones críticas, almacenar información, analizar datos y ofrecer servicios digitales a clientes y empleados. Las soluciones pueden crecer o reducirse según la necesidad del negocio.

Lo que debes recordar

- ✓ Azure es una plataforma cloud.
- ✓ Los recursos se consumen bajo demanda.
- ✓ Permite innovar sin necesidad de administrar infraestructura física.

Cápsula 2

¿Qué es un Resource Group?

Concepto

Un Resource Group es un contenedor lógico que organiza y administra recursos relacionados dentro de Azure.

Todos los componentes de una solución pueden agruparse dentro del mismo Resource Group para facilitar su administración.

¿Por qué es importante?

Permite organizar recursos de manera ordenada, aplicar permisos, controlar costos y simplificar operaciones de mantenimiento.

Lo que debes recordar

- ✓ Agrupa recursos relacionados.
- ✓ Facilita la administración.
- ✓ Ayuda a mantener organizados los entornos tecnológicos.

Cápsula 3

¿Qué es Azure App Service?

Concepto

Azure App Service es una plataforma administrada que permite ejecutar aplicaciones web, sitios web y APIs sin necesidad de gestionar servidores o sistemas operativos.

¿Por qué es importante?

Las empresas utilizan App Service para publicar aplicaciones de forma rápida, segura y escalable. El servicio se encarga automáticamente de la infraestructura necesaria para que la aplicación esté disponible.

Lo que debes recordar

- ✓ Permite publicar aplicaciones web.
- ✓ Soporta múltiples lenguajes de desarrollo.
- ✓ Reduce la complejidad operativa.

Cápsula 4

¿Qué es una API?

Concepto

API significa *Application Programming Interface*.

Es un conjunto de reglas que permite que distintas aplicaciones intercambien información y trabajen juntas.

¿Por qué es importante?

Las APIs son la base de gran parte de las aplicaciones modernas. Permiten conectar sistemas, compartir información y automatizar procesos.

Ejemplos cotidianos incluyen aplicaciones móviles que consultan información de vuelos, banca digital, comercio electrónico y redes sociales.

Lo que debes recordar

- ✓ Las APIs conectan sistemas.
- ✓ Permiten compartir datos de forma controlada.
- ✓ Son fundamentales en las arquitecturas modernas.

Cápsula 5

¿Qué es una Arquitectura Frontend y Backend?

Concepto

La mayoría de las aplicaciones modernas están compuestas por dos partes principales:

Frontend

Es la interfaz con la que interactúan los usuarios.

Puede ser una aplicación web, una aplicación móvil o un portal.

Su función es mostrar información y capturar acciones del usuario.

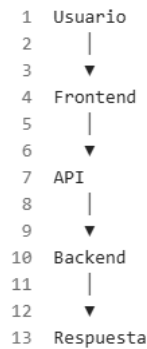
Backend

Es la parte que procesa la información y ejecuta la lógica del negocio.

El backend recibe solicitudes, consulta datos, aplica reglas y devuelve respuestas.

¿Cómo trabajan juntos?

Cuando un usuario realiza una acción, el frontend envía una solicitud al backend a través de una API.



¿Por qué es importante?

Esta separación permite desarrollar, mantener y escalar cada componente de forma independiente.

Es una de las arquitecturas más utilizadas en aplicaciones empresariales, banca digital, comercio electrónico y plataformas de servicios.

Lo que debes recordar

- ✓ El frontend es lo que ve el usuario.
- ✓ El backend procesa la información.
- ✓ Ambos componentes se comunican mediante APIs.
- ✓ Si uno falla, la experiencia completa puede verse afectada.

Cápsula 6

¿Qué es DevOps?

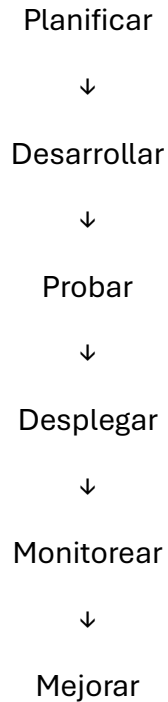
Concepto

DevOps es una cultura de trabajo que integra desarrollo de software y operaciones tecnológicas para acelerar la entrega de soluciones de calidad.

¿Por qué es importante?

Las organizaciones necesitan actualizar aplicaciones constantemente. DevOps ayuda a reducir errores, mejorar la colaboración y automatizar procesos de despliegue y monitoreo.

Ciclo típico



Lo que debes recordar

- ✓ DevOps es una metodología.
- ✓ Promueve automatización y colaboración.
- ✓ Busca entregas rápidas y confiables.

Cápsula 7

¿Qué es Infraestructura como Código (IaC)?

Concepto

Infraestructura como Código es una práctica que permite definir infraestructura tecnológica mediante archivos de configuración escritos en código.

¿Por qué es importante?

Facilita despliegues repetibles, reduce errores humanos y permite construir entornos completos en pocos minutos.

Lo que debes recordar

- ✓ La infraestructura puede crearse automáticamente.
 - ✓ Los cambios quedan documentados.
 - ✓ Mejora la consistencia entre entornos.
-

Cápsula 8

¿Qué es Terraform?

Concepto

Terraform es una herramienta utilizada para implementar Infraestructura como Código.

Permite crear y administrar recursos tecnológicos mediante archivos declarativos.

¿Por qué es importante?

Es ampliamente utilizado para automatizar despliegues en entornos cloud, garantizando consistencia y repetibilidad.

Lo que debes recordar

- ✓ Automatiza la creación de infraestructura.
 - ✓ Reduce tareas manuales.
 - ✓ Facilita la gestión de entornos complejos.
-

Cápsula 9

¿Qué es la Observabilidad?

Concepto

La observabilidad es la capacidad de comprender el comportamiento interno de un sistema analizando la información que genera.

Los tres pilares fundamentales

Logs

Registros de eventos y actividades.

Métricas

Indicadores numéricos de desempeño.

Trazas

Seguimiento completo del recorrido de una solicitud dentro de un sistema.

¿Por qué es importante?

Permite detectar problemas, entender comportamientos inesperados y tomar decisiones basadas en evidencia.

Lo que debes recordar

- ✓ No se puede mejorar lo que no se puede medir.
- ✓ La observabilidad ayuda a encontrar causas raíz.

Cápsula 10

¿Qué son los Logs?

Concepto

Un log es un registro automático generado por una aplicación, servicio o sistema mientras está funcionando.

Los logs permiten conocer qué ocurrió, cuándo ocurrió y qué resultado tuvo una operación.

¿Qué información pueden contener?

- Solicitudes recibidas.
- Usuarios autenticados.
- Procesos ejecutados.
- Errores detectados.
- Eventos de seguridad.
- Integraciones con otros sistemas.

Ejemplo

```
1 10:15:01 - Solicitud recibida
2 10:15:02 - Consulta procesada
3 10:15:02 - Respuesta enviada
```

¿Por qué son importantes?

Los logs representan la principal fuente de evidencia cuando se investiga un problema.

Cuando una aplicación presenta un comportamiento inesperado, los equipos técnicos suelen comenzar revisando los registros generados.

Lo que debes recordar

- ✓ Los logs cuentan la historia de lo que ocurrió.
- ✓ Permiten investigar incidentes.
- ✓ Son una de las herramientas más utilizadas en operaciones y soporte tecnológico.
- ✓ Una aplicación sin logs es difícil de diagnosticar.

Cápsula 11

¿Qué es KQL (Kusto Query Language)?

Concepto

KQL (Kusto Query Language) es el lenguaje de consultas utilizado por Microsoft para analizar información almacenada en Azure Monitor, Log Analytics y Application Insights.

Su propósito es ayudarte a encontrar información específica dentro de grandes volúmenes de datos de monitoreo.

¿Por qué es importante?

Las aplicaciones modernas generan miles o millones de registros diariamente.

KQL permite:

- Buscar errores.
- Filtrar información.

- Identificar tendencias.
- Analizar excepciones.
- Revisar solicitudes.
- Investigar incidentes.

¿Cómo funciona?

KQL consulta tablas que almacenan datos recopilados por Azure.

Por ejemplo:

- Requests
- Exceptions
- Traces
- Dependencies
- Metrics

Ejemplo sencillo

Consultar las últimas solicitudes registradas:

```
1 requests
2 | take 10
```

Consultar las excepciones registradas:

```
1 exceptions
2 | take 10
```

Conceptos básicos de KQL

Tabla

Es el lugar donde se almacenan los datos.

Ejemplos:

```
1 requests
2 exceptions
3 traces
4 dependencies
```

Filtrar

Permite mostrar únicamente la información que nos interesa.

Ejemplo:

```
1 exceptions
2 | where severityLevel >= 3
```

Ordenar

Permite visualizar primero la información más reciente.

Ejemplo:

```
1 exceptions
2 | order by timestamp desc
```

Contar

Permite conocer cuántos eventos ocurrieron.

Ejemplo:

```
1 requests
2 | count
```

Lo que debes recordar

- ✓ KQL es el lenguaje para consultar datos de monitoreo en Azure.
- ✓ Permite analizar logs, excepciones y métricas.
- ✓ Es una herramienta fundamental para observabilidad y diagnóstico.
- ✓ No modifica datos; únicamente los consulta.

Cápsula 12

¿Qué es Application Insights?

Concepto

Application Insights es una funcionalidad de Azure Monitor diseñada para supervisar aplicaciones y analizar su comportamiento en tiempo real.

¿Qué información proporciona?

- Rendimiento.
- Excepciones.
- Disponibilidad.
- Solicitudes.
- Dependencias.
- Trazas.

¿Por qué es importante?

Ayuda a los equipos técnicos a identificar problemas, diagnosticar incidentes y mejorar continuamente el desempeño de las aplicaciones.

Lo que debes recordar

- ✓ Convierte datos técnicos en información útil.
 - ✓ Facilita el diagnóstico de problemas.
 - ✓ Proporciona visibilidad sobre el estado de las aplicaciones.
-

Cápsula 13

¿Qué son los Logs, las Excepciones y los Errores 500?

Concepto

Cuando una aplicación se está ejecutando, genera información que ayuda a comprender lo que está ocurriendo internamente.

Tres de los elementos más importantes son:

- Logs
- Excepciones
- Códigos de error

¿Qué es un Log?

Un log es un registro de eventos generado por una aplicación.

Puede contener información sobre:

- Inicio de procesos.
- Acciones realizadas.
- Resultados obtenidos.
- Advertencias.
- Errores.

Los logs ayudan a reconstruir lo que ocurrió dentro de un sistema.

Ejemplo

```
1 Usuario autenticado correctamente.
```

```
1 Solicitud recibida.
```

```
1 Proceso completado.
```

¿Qué es una Excepción?

Una excepción ocurre cuando una aplicación encuentra una situación inesperada que le impide continuar normalmente.

Por ejemplo:

- Un dato requerido no existe.
- Una configuración es incorrecta.
- Un servicio externo no responde.

Las excepciones suelen registrarse automáticamente para facilitar el diagnóstico.

Ejemplo

```
1 Archivo no encontrado.  
2 ``
```

```
1 Variable de configuración inexistente.
```

```
1 Conexión no disponible.
```

¿Qué es un Error 500?

El código HTTP 500 significa:

```
1 Internal Server Error
```

Indica que ocurrió un problema dentro del servidor mientras intentaba procesar una solicitud.

No significa necesariamente que el servidor esté apagado.

Frecuentemente indica que existe una excepción o una configuración incorrecta que impide completar la operación.

¿Por qué es importante?

Los equipos de tecnología utilizan logs y excepciones para:

- Investigar incidentes.
- Identificar causas raíz.
- Validar configuraciones.
- Mejorar la estabilidad de las aplicaciones.

La mayoría de las herramientas modernas de monitoreo recopilan y analizan automáticamente esta información.

Lo que debes recordar

- ✓ Los logs cuentan la historia de lo que sucedió.

- ✓ Las excepciones indican problemas específicos.
- ✓ Un error 500 señala un fallo interno en la aplicación.
- ✓ Analizar evidencias es una de las principales habilidades de un profesional de tecnología.

Cápsula 14

¿Qué es CORS y por qué existe?

Concepto

CORS significa **Cross-Origin Resource Sharing**.

Es un mecanismo de seguridad utilizado por los navegadores para controlar qué aplicaciones pueden comunicarse entre sí.

¿Qué es un origen (Origin)?

Un origen está formado por:

- Protocolo
- Dominio
- Puerto

Por ejemplo:

`https://miapp.com`

es diferente de:

`https://api.miapp.com`

Aunque pertenezcan a la misma organización.

¿Por qué existe CORS?

Sin controles de seguridad, cualquier sitio web podría intentar acceder a información de otras aplicaciones sin autorización.

CORS protege a los usuarios limitando qué sistemas pueden realizar solicitudes a una aplicación.

Ejemplo sencillo

Imagina que una aplicación web intenta consultar información de otra plataforma.

Antes de responder, el navegador verifica si esa comunicación está permitida.

Si no lo está, la solicitud es bloqueada.

¿Por qué es importante?

Muchas aplicaciones modernas están formadas por múltiples componentes distribuidos.

Configurar correctamente los orígenes permitidos es fundamental para garantizar seguridad y funcionamiento adecuado.

Lo que debes recordar

- ✓ CORS es una medida de seguridad.
- ✓ Controla quién puede acceder a una aplicación desde el navegador.
- ✓ Un sistema puede funcionar correctamente y aun así ser bloqueado por CORS.
- ✓ Es común encontrar configuraciones CORS en APIs web.

Cápsula 15

¿Qué es un Health Check?

Concepto

Un Health Check es un mecanismo utilizado para verificar automáticamente si una aplicación está funcionando correctamente.

¿Por qué es importante?

Las organizaciones necesitan asegurar que sus aplicaciones estén disponibles para los usuarios. Los Health Checks permiten validar el estado operativo de los componentes de manera continua.

Lo que debes recordar

- ✓ Una aplicación puede estar encendida y no funcionar correctamente.
- ✓ Los Health Checks ayudan a detectar problemas rápidamente.

- ✓ Son una práctica común en sistemas modernos.
-

Cápsula 16

¿Qué son las Variables de Configuración?

Concepto

Las aplicaciones utilizan parámetros de configuración para definir su comportamiento sin necesidad de modificar el código fuente.

Algunos ejemplos son:

- URLs de servicios.
- Credenciales.
- Parámetros de integración.
- Configuraciones de seguridad.

¿Por qué es importante?

Permiten adaptar una aplicación a distintos entornos como desarrollo, pruebas y producción.

Lo que debes recordar

- ✓ Separan la configuración del código.
 - ✓ Facilitan los despliegues.
 - ✓ Son esenciales para la operación de aplicaciones modernas.
-

Cápsula 17

Pensamiento de Diagnóstico Tecnológico

Concepto

Los profesionales de tecnología resuelven problemas basándose en evidencia y datos.

Ante un incidente, suelen seguir un proceso estructurado:

Observar



Investigar



Analizar



Corregir



Validar

```
1 Observar
2 ↓
3 Investigar
4 ↓
5 Analizar
6 ↓
7 Corregir
8 ↓
9 Validar
```

¿Por qué es importante?

Un análisis basado en evidencia permite encontrar la causa real de los problemas y evita realizar cambios innecesarios o incorrectos.

Lo que debes recordar

- ✓ Los datos ayudan a tomar mejores decisiones.
 - ✓ La observación es tan importante como la corrección.
 - ✓ Comprender el problema es el primer paso para resolverlo.
-