# **Ejercicios de aplicación- conceptuales**

1. En una evaluación de arquitectura uno de los interesados preguntó con curiosidad sobre la disponibilidad que el arquitecto había especificado (99.9%). La respuesta que recibió fue: ”… es la disponibilidad que nos asegura el proveedor de servicios de la nube”

1.1 ¿Cuánto tiempo fuera de servicio admite la especificación de 99.9%?

* + 1. 3 días y 15,6 horas en un año
    2. 52 minutos y 34 segundos en un año
    3. 8 horas, 0 minutos y 46 segundos en un año
    4. 32 segundos en un año

1.2 ¿Cómo considera la respuesta del arquitecto? **Justifique** su respuesta

1. Correcta
2. Incorrecta --<ok
3. Dadas las siguientes decisiones que tomó un arquitecto de software durante el diseño de un sistema en el cual se le solicitó **una buena disponibilidad**. Identifique el nombre de las tácticas que seleccionó.

|  |  |
| --- | --- |
| Táctica | Descripción de la decisión |
| Heartbeat | Incorporó un mecanismo en los componentes mediante el cual cada 20 segundos los componentes envían una señal indicando que están disponibles |
| Monitor | Diseño un componente que, mediante el uso de pulso, ping, e información de *timestamp* detecta y corrige problemas con componentes que fallan o que están respondiendo en forma degradada |
| Manejo de excepciones | Solicitó a todos los desarrolladores que utilicen bloques try/ catch en el código e incorporen dentro de los mismos mecanismos que intenten recuperar al sistema |
| Removal from service | Incorporó un mecanismo que permite marcar a un componente como fuera de servicio cuando el mismo no responde correctamente y que no afecta funciones críticas del sistema |

3. Identifique y explique las Tácticas concretas para los siguientes casos:

3.1) Una aerolínea ha instalado 50 kioskos para que los usuarios hagan su check-in con antelación. El servidor central cuenta con un componente de software que envía una comunicación a cada Kiosko y espera su respuesta. Con esta mecánica el servidor central sabe cuándo uno de los kioskos necesita atención. 🡪 AC: DISPONIBILIDAD. TACTICA: PING/ECHO , CATEGORIA: DETECTAR FALLAS.

3.2) Los programadores de una aplicación notaron que el manejo de logeo de los distintos tipos de excepciones en cada capa de la aplicación podría generalizarse en una capa lógica transversal a ellas, de modo que la lógica común al manejo de errores y su registro estuviera centralizada. 🡪 AC: MODIFICABILIDAD. TACTICA:ABSTRAER SERVICIOS COMUNES