

Instrucciones

1. El alumnado de manera individual leerá y analizará el siguiente artículo [The REST architectural style](#) .
2. Una vez realizada la lectura, llenará el recuadro vacío que se encuentra a la derecha del estilo con sus propias palabras.

Restricción	Descripción
Estilo Nulo	<p>El Estilo Nulo es un concepto en arquitecturas de software que se refiere a la ausencia de restricciones arquitectónicas específicas. Es el punto de partida teórico antes de aplicar cualquier estilo o patrón arquitectónico.</p> <p>En el Estilo Nulo, no hay reglas predefinidas sobre cómo los componentes del sistema deben interactuar o cómo se debe estructurar la aplicación. Esto proporciona máxima flexibilidad, pero también la mínima guía para el diseño del sistema.</p>
Cliente-Servidor	<p>El estilo arquitectónico Cliente-Servidor es un enfoque fundamental en el diseño de sistemas distribuidos. Se caracteriza por la separación de responsabilidades entre dos tipos principales de componentes: el cliente, que maneja la interfaz de usuario, y el servidor, que se encarga del almacenamiento de datos y procesamiento de solicitudes.</p>
Sin estado	<p>En arquitecturas REST, el concepto "sin estado" se refiere a que cada solicitud del cliente al servidor debe contener toda la información necesaria para entender y procesar la petición. El servidor no almacena ningún contexto de sesión entre solicitudes.</p>
Caché	<p>En arquitecturas REST, el caché es un componente que almacena respuestas a solicitudes previas para su reutilización futura. Su objetivo principal es mejorar la eficiencia y el rendimiento del sistema.</p> <p>Cuando un cliente realiza una solicitud, el caché puede interceptarla y devolver una respuesta almacenada si es aplicable, evitando así la necesidad de que el servidor procese la solicitud nuevamente. Esto reduce la latencia, disminuye la carga en el servidor y mejora la experiencia del usuario.</p>
Interface	<p>La Interfaz Uniforme es un principio clave en las arquitecturas REST.</p>

<p>Uniforme</p>	<p>Establece un conjunto estándar de operaciones para interactuar con los recursos en un sistema. Este enfoque hace que la arquitectura sea más simple y flexible, permitiendo que las diferentes partes del sistema se desarrollen de forma independiente.</p>
<p>Sistema de capas</p>	<p>El Sistema de Capas es un principio importante en las arquitecturas REST, diseñado para mejorar el comportamiento de los sistemas a escala de Internet. Este enfoque organiza la arquitectura en capas jerárquicas, donde cada componente solo puede interactuar con la capa inmediatamente adyacente.</p> <p>Esta restricción limita el conocimiento que cada componente tiene del sistema en general. Como resultado, se reduce la complejidad global del sistema y se promueve la independencia entre las diferentes partes. Esto hace que el sistema sea más fácil de entender, mantener y escalar.</p>
<p>Código bajo demanda</p>	<p>El Código bajo Demanda es una restricción opcional en las arquitecturas REST. Este principio permite a los sistemas ampliar la funcionalidad del cliente mediante la descarga y ejecución de código en forma de applets o scripts.</p> <p>La principal ventaja de este enfoque es la simplificación de los clientes. Al permitir la descarga de funciones después de la implementación, se reduce el número de características que deben estar pre-implementadas en el cliente. Esto mejora significativamente la extensibilidad del sistema.</p> <p>Sin embargo, el Código bajo Demanda tiene una desventaja: reduce la visibilidad del sistema. Esto significa que es más difícil entender y predecir el comportamiento del cliente, ya que parte de su funcionalidad puede cambiar dinámicamente.</p>