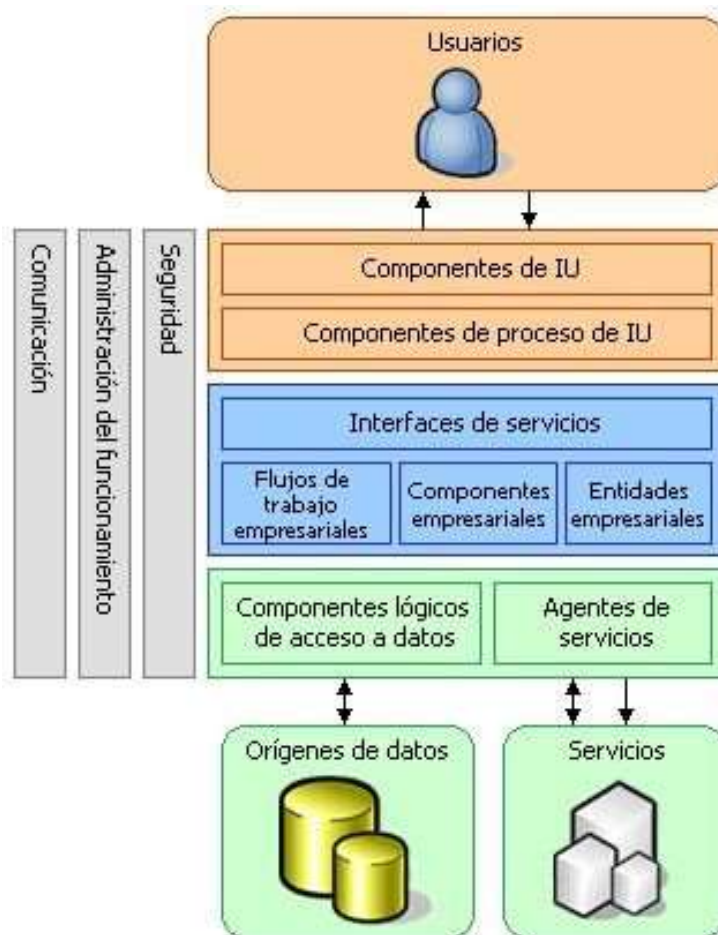


Aplicación en 3 capas y pasaje de información entre capas

El diseño de aplicaciones distribuidas no es una tarea sencilla. Es necesario tomar un gran número de decisiones a nivel de arquitectura, diseño e implementación. Estas decisiones tendrán un impacto en las "capacidades" de la aplicación (seguridad, escalabilidad, disponibilidad y mantenimiento, entre otras), así como en la arquitectura, el diseño y la implementación de la infraestructura de destino. La guía le ayudará a comprender las distintas opciones que se presentan a la hora de diseñar las capas de una aplicación distribuida; estas opciones se presentan como un conjunto de capas de componentes que se podrán utilizar para modelar la aplicación. En la figura 1 se muestran las capas de los componentes lógicos que este documento utiliza para estructurar sus instrucciones.



Capas de componentes de servicios y aplicaciones distribuidas creadas con .NET

La programación en capas es una arquitectura cliente-servidor en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario.

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, solo se ataca al nivel requerido sin tener que revisar entre código mezclado. Un buen ejemplo de este método de programación sería el modelo de interconexión de sistemas abiertos.

Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API que existe entre niveles.

En el diseño de sistemas informáticos actual se suelen usar las arquitecturas multinivel o Programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten).

La capa de negocio contiene la lógica principal de procesamiento de datos dentro de nuestra aplicación. Se comunica con la capa de presentación para obtener las entradas del usuario y presentar la información resultante, así como la capa de acceso a datos o directamente con servicios para realizar sus operaciones.

GUI - Graphical User Interface - Capa de presentación o interfaz gráfica de Usuario

Se refiere a la presentación del programa frente al usuario, esta presentación debe cumplir su propósito con el usuario final, una presentación fácil de usar y amigable. También las interfaces deben ser consistentes con la información dentro del software (Por ejemplo; en los formularios no debe haber más que lo necesario), tomar en cuenta los requerimientos del usuario, la capa de presentación va de la mano con capa de la lógica de negocio.

BLL - Business Logic Layer - Capa Lógica de Negocios

En esta capa es donde se encuentran los programas que son ejecutados, recibe las peticiones del usuario y posteriormente envía las respuestas tras el proceso. Esta capa es muy importantes pues es donde se establecen todas aquellas reglas que se tendrán que cumplir, decía anteriormente que la capa de presentación tiene comunicación con la capa de lógica de negocio ya que se tienen que comunicar para recibir las solicitudes y presentar los resultados.

DAL - Data Access Layer - Capa de Acceso a datos

Esta capa es la que se encarga de hacer las transacciones con la base de datos y con otros sistemas para descargar o insertar información al sistema. La consistencia en los datos es sumamente importante, es decir, los datos que se ingresan o insertan deben ser precisos y conscientes. Aquí definimos las consultas que vamos a realizar en la base de datos, o consultas para reporte. La comunicación de esta capa con la capa de lógica de negocio se refiere a que la capa de datos es la que le enviara información a la capa de negocio para que sea procesada e ingresada en objetos según sea necesario (encapsulamiento).

Recomendaciones:

Para profundizar y complementar los contenidos expuestos en esta clase, se sugiere abordar los siguientes cursos:

- Introducción a Bases de Datos y SQL
- Fundamentos de Programación con SQL Server 2017
- Entity Framework y LINQ