

# PROGRAMACIÓN CON PATRON N CAPAS

## INTEGRANTES:

- FERNANDO JOSE PACO FEIJOO
- DIEGO FERNANDO DE LA CRUZ CACERES
- JILMER JAIR CORONEL RAMOS

## INSTRUCTOR:

- ARTURO COLLADO ALFARO

## CARRERA Y SEMESTRE

- DESARROLLO DE SOFTWARE – IV SEMESTRE

**GRUPO NEW MINDS**

## Tabla de contenido

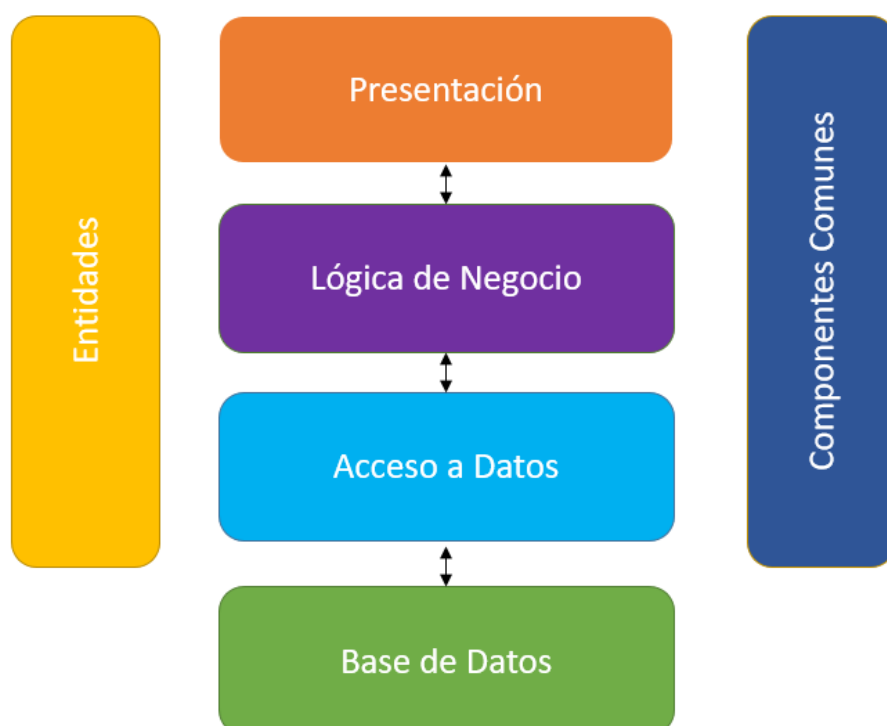
<b>INTRODUCCION .....</b>	<b>3</b>
Presentación del Tema.....	3
<b>PROGRAMACIÓN CON PATRON N CAPAS .....</b>	<b>4</b>
¿Qué es la arquitectura software? .....	4
¿Qué es la Arquitectura por capas? .....	4
<b>CAPA DE PRESENTACION .....</b>	<b>5</b>
Definición .....	5
<b>CAPA DE NEGOCIOS .....</b>	<b>6</b>
Definición .....	6
<b>CAPA DE DATOS.....</b>	<b>7</b>
Definición .....	7
<b>CAPAS Y NIVELES .....</b>	<b>8</b>
Niveles y capas .....	8
Ejemplos de esto último:.....	8
<b>BENEFICIOS .....</b>	<b>9</b>
Ejemplos:.....	9
<b>RECOMENDACIONES.....</b>	<b>10</b>
<b>CONCLUSIONES .....</b>	<b>10</b>
<b>DEDICATORIA .....</b>	<b>11</b>
<b>RESUMEN CRONOLOGICO .....</b>	<b>12</b>
<b>LISTA DE RECURSOS .....</b>	<b>13</b>
Máquinas y Equipos .....	13
Herramientas e insumos .....	13
<b>REFERENCIAS.....</b>	<b>14</b>
Páginas web.....	14
Videos Referenciales .....	14

## INTRODUCCION

### Presentación del Tema

La realización de Sistemas de Información se ha venido desarrollando en base a técnicas de programación, principalmente; la programación estructurada, luego en combinación utilizando la programación por eventos, actualmente se pudiera decir que se ha llegado a una madurez con la potencialidad de la programación orientada a objetos por la ventaja en la reutilización de código. En adición a ellas, se cuenta actualmente con la programación en n capas que hace uso de la programación orientada a objetos; la cual consiste en separar el código fuente según el rol, responsabilidad y funcionalidad; por ende, el desarrollo es más rápido, y resulta más fácil el darle mantenimiento al Sistema

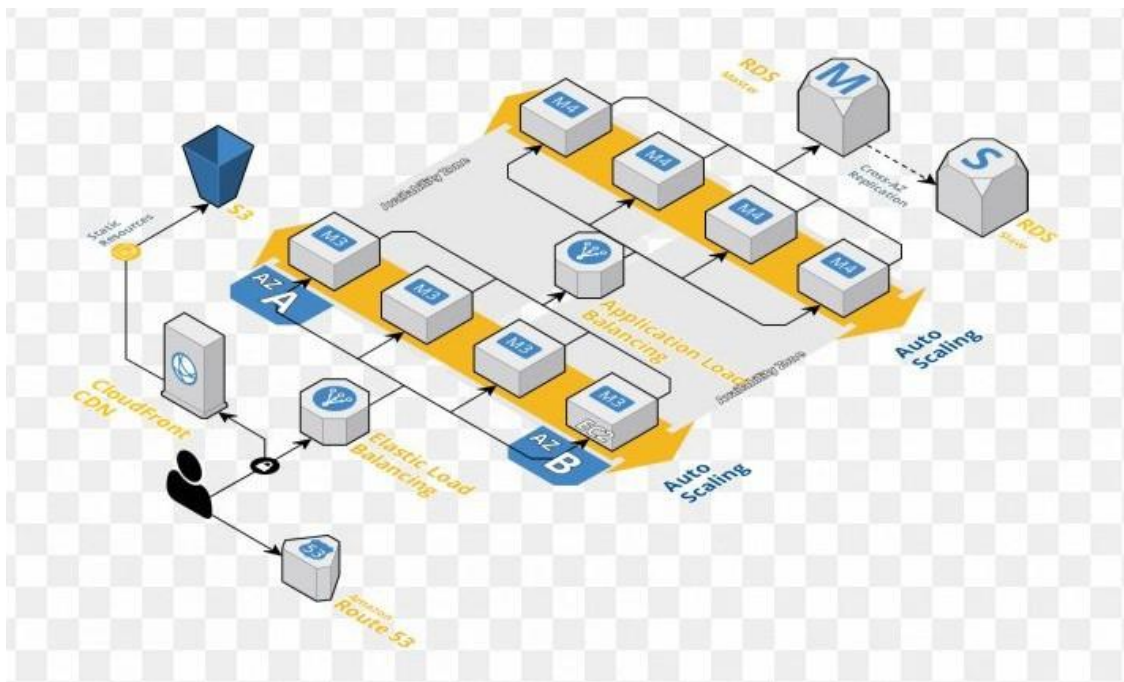
La programación por capas es una arquitectura cliente-servidor en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario. La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo 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).



## PROGRAMACIÓN CON PATRON N CAPAS

### ¿Qué es la arquitectura software?

La arquitectura del software es el diseño de más alto nivel de la estructura de un sistema. Consiste en un conjunto de abstracciones que forman el “marco” del software. La arquitectura se diseña en la fase posterior a la de requisitos, la llamada fase de diseño.

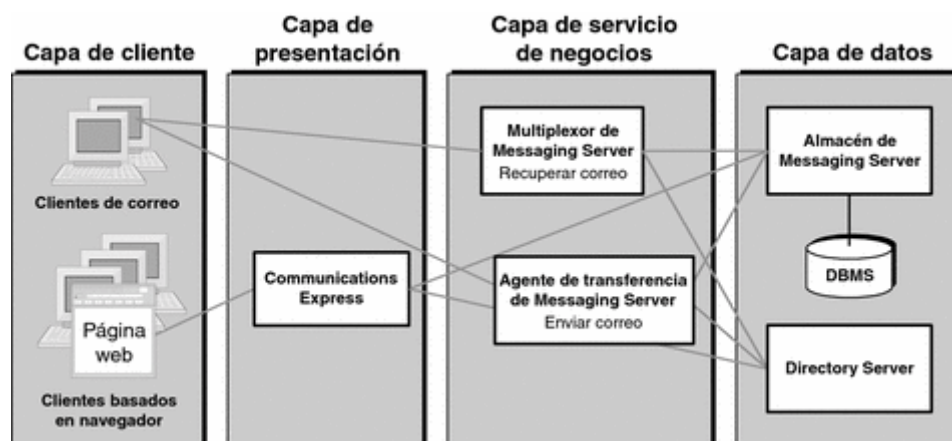


### ¿Qué es la Arquitectura por capas?

La arquitectura en capas es un patrón de arquitectura software usada en la gran mayoría de sistemas.

Se centra en una distribución jerárquica de las roles y responsabilidades proporcionando una separación efectiva de las preocupaciones (cada cual se encarga de lo que le corresponde).

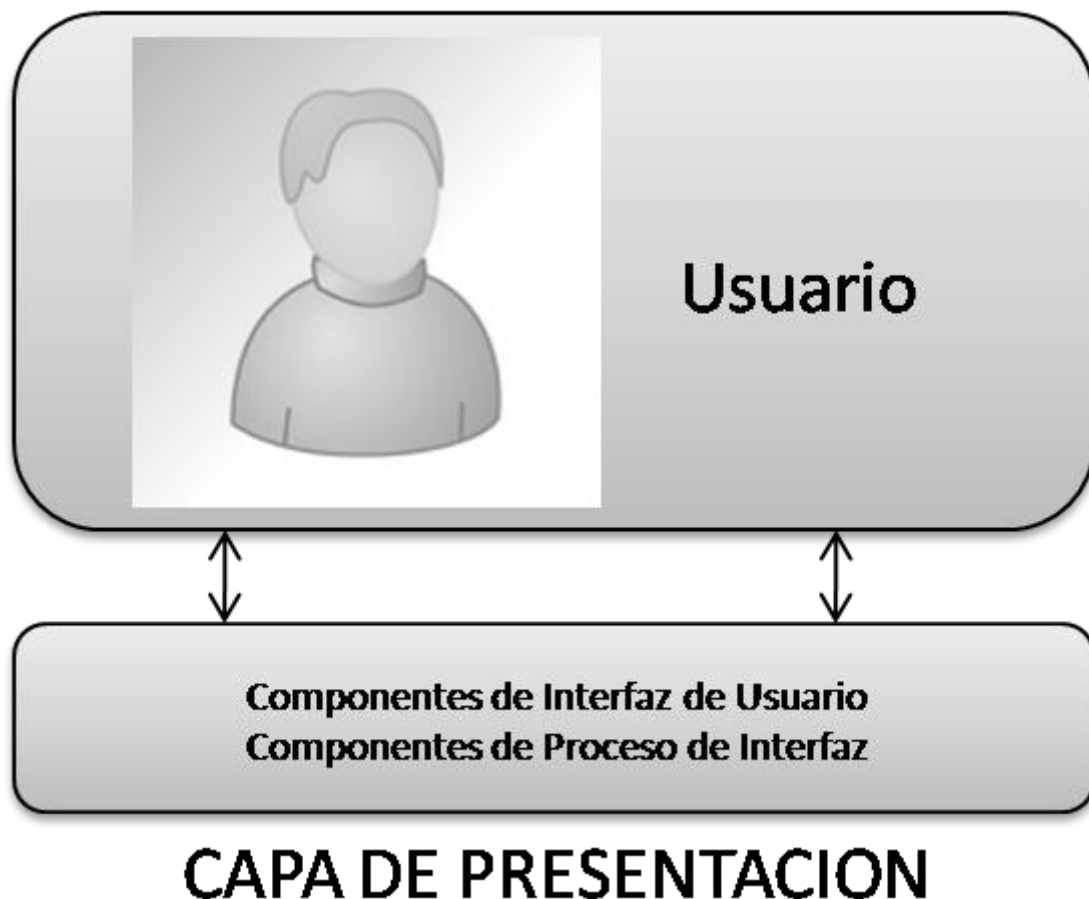
Al pensar en un sistema en términos de capas, se imaginan los principales subsistemas de software ubicados de la misma forma que las capas de un pastel, donde cada capa descansa sobre la inferior.



## CAPA DE PRESENTACION

### Definición

(Funcionalidad relacionada con la User Interface) es la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio.



## CAPA DE NEGOCIOS

### Definición

(Procesamiento de reglas de negocio). También denominada Lógica de Dominio, esta capa contiene la funcionalidad que implementa la aplicación. Involucra cálculos basados en la información dada por el usuario y datos almacenados y validaciones. Controla la ejecución de la capa de acceso a datos y servicios externos. Se puede diseñar la lógica de la capa de negocios para uso directo por parte de componentes de presentación o su encapsulamiento como servicio y llamada a través de una interfaz de servicios que coordina la conversación con los clientes del servicio o invoca cualquier flujo o componente de negocio.

Es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es, aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.



## CAPA DE DATOS

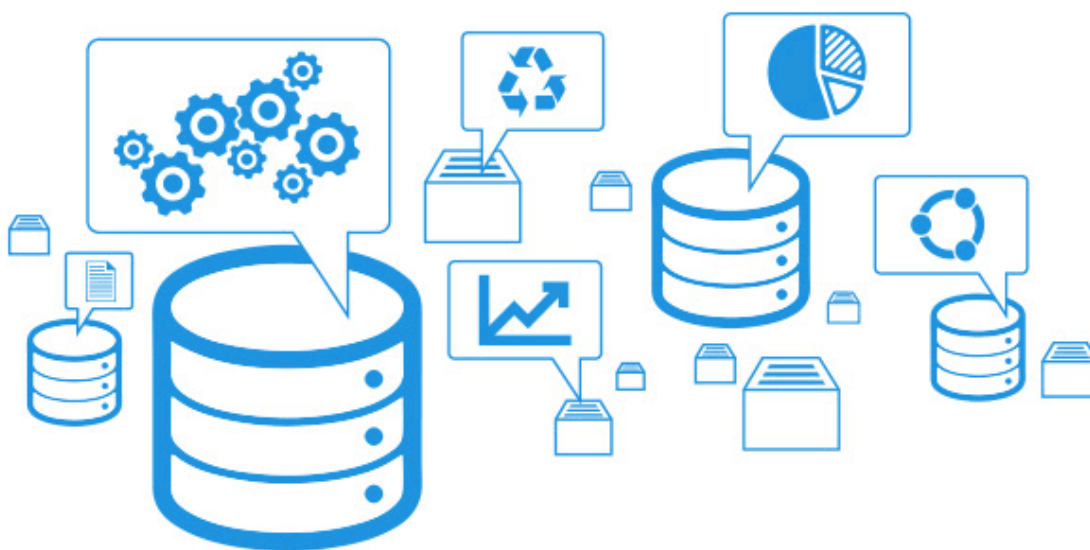
### Definición

(Funcionalidad relacionada con el acceso a datos). Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Todas estas capas pueden residir en un único ordenador, si bien lo más usual es que haya una multitud de ordenadores en donde reside la capa de presentación (son los clientes de la arquitectura cliente/servidor). Las capas de negocio y de datos pueden residir en el mismo ordenador, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más ordenadores. Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varios ordenadores los cuales recibirán las peticiones del ordenador en que resida la capa de negocio.

Si, por el contrario, fuese la complejidad en la capa de negocio lo que obligase a la separación, esta capa de negocio podría residir en uno o más ordenadores que realizarían solicitudes a una única base de datos. En sistemas muy complejos se llega a tener una serie de ordenadores sobre los cuales corre la capa de negocio, y otra serie de ordenadores sobre los cuales corre la base de datos.





## CAPAS Y NIVELES

### Niveles y capas

¿Niveles y capas con lo mismo?, los niveles corresponden a la forma en que las capas lógicas se encuentran distribuidas de forma física.

### Ejemplos de esto último:

Una solución de 3 capas (presentación, lógica del negocio, datos) que residen en un solo ordenador (Presentación+lógica+datos). Se dice que la arquitectura de la solución es de tres capas y un nivel.

Una solución de 3 capas (presentación, lógica del negocio, datos) que residen en dos ordenadores (Presentación+ lógica, por un lado; lógica +datos por el otro lado). Se dice que la arquitectura de la solución es de tres capas y dos niveles.



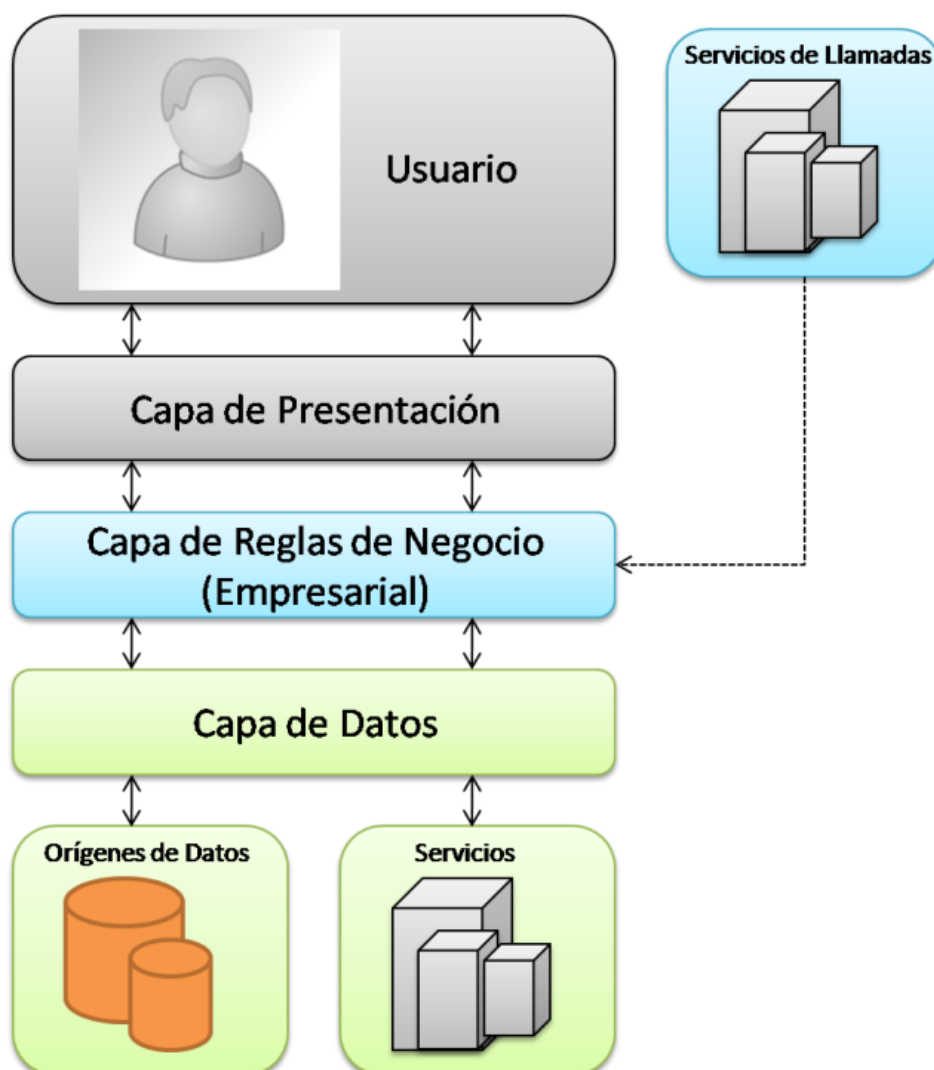


## BENEFICIOS

### Ejemplos:

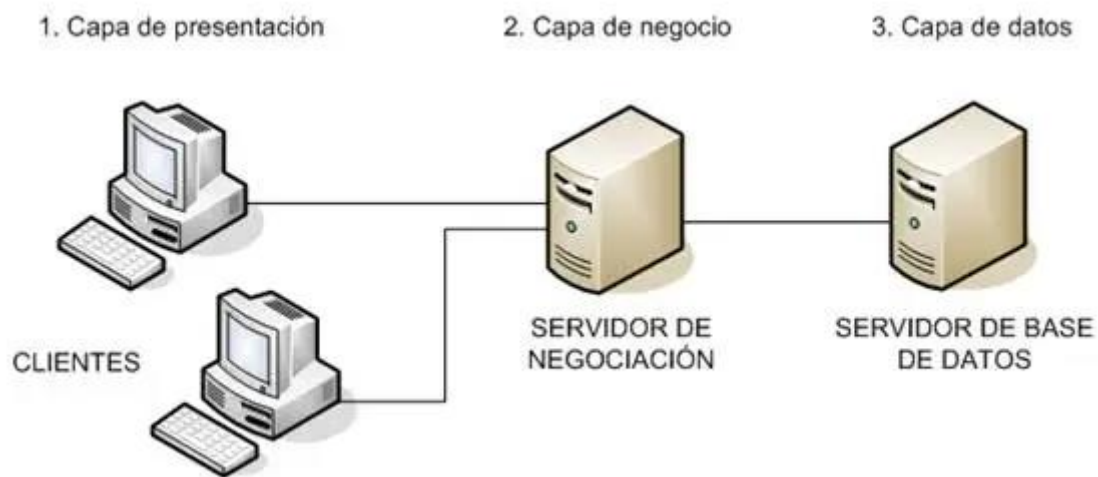
- Abstracción
- Encapsulamiento
- Capas de funcionalidad muy bien definidas,
- Alta cohesión
- Reusabilidad (las capas inferiores de la pirámide no tienen dependencias de las superiores por lo que es fácil reutilizarlas)
- Débil acoplamiento (comunicación basada en abstracciones)

La separación, reduce el riesgo y el impacto de los cambios tecnológicos. Por ejemplo, al cambiar el mecanismo de persistencia de la aplicación las capas que lo consumen no tienen por qué realizar cambios. Además, aumenta el rendimiento (a partir de cierta carga de trabajo) al distribuir las capas sobre múltiples capas físicas. También aumenta la escalabilidad y la tolerancia a fallos.



## RECOMENDACIONES

- No exceder en el número de creación de capas, es posible que exista pérdida de eficiencia.
- Se recomienda usar la programación con 3 capas, porque es la más convencional y la programación es más ordenada y segura.
- Es importante separar bien estas capas, ya que por ejemplo el negocio no podrá tener acceso directamente sin pasar primero por la capa acceso a datos.
- La capa de presentación no puede hacer ningún tipo de cálculo importante de la aplicación o acceder directamente a base de datos.



## CONCLUSIONES

El estilo de programación en N capas se basa en segmentar un proyecto en varias partes para realizar una programación independiente en cada una de ellas.

- Facilita la reutilización en capas.
- Permite una mejor estandarización.

El trabajo por parte de los analistas es complejo, pero al final se crea una arquitectura más fácil de comprender y de implementar.

El desarrollo de una aplicación de N-Capas es básicamente un estilo de programación en el cual se separan lógicamente cada uno de los elementos de la aplicación, esto es, cada una de las capas en la que estará formada nuestra aplicación.

La programación por capas es un estilo de programación en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño.

## DEDICATORIA

**Este presente trabajo está dedicado a mi instructor, compañeros y a mi equipo que fue capaz de realizar este proyecto y a la gran dedicación de estos mismos.**

**Gracias.**

**Atte New Minds.**



## RESUMEN CRONOLÓGICO

### Sábado 08 de octubre del 2022

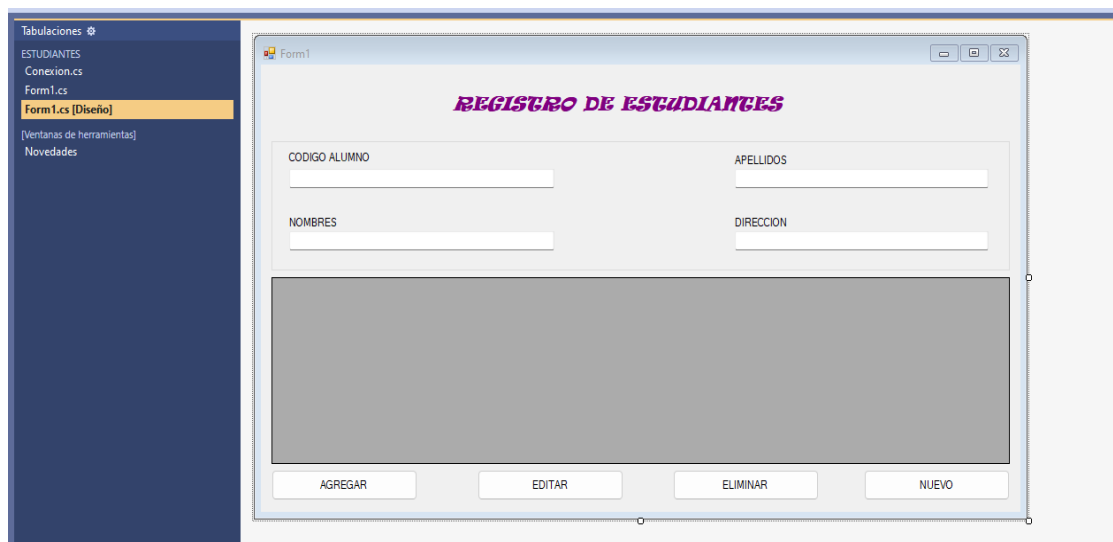
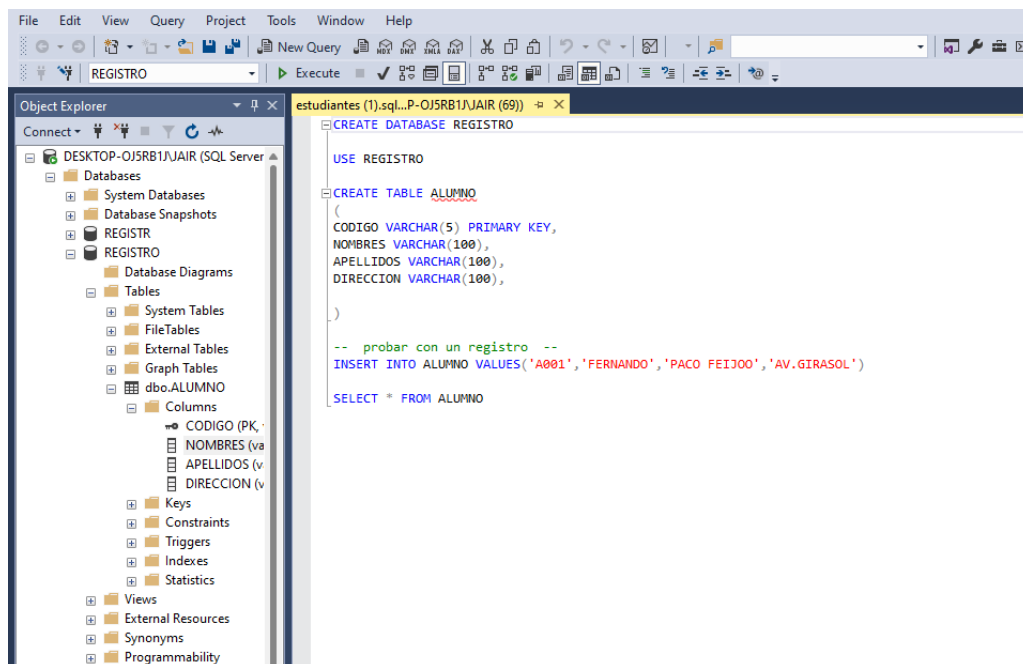
- Realizar las aplicaciones CRUD con visual Basic y conectarlo con SQL server.
- Establecer detalles en la página web y distribuir los temas de exposición.

### Domingo 09 de octubre del 2022

- Realizar las pruebas a nuestra página web de presentación
- Acordar el orden de la exposición y la revisión detallada del documento escrito.

### Lunes 10 de octubre del 2022

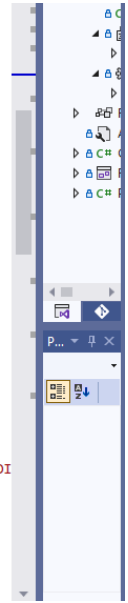
- Realizar las pruebas a nuestra página web de presentación y nuestra vista en visual.NET
- Acordar el orden de la exposición y la revisión detallada del documento escrito.



```

0      return dt;
1
2    }
3
4  1 referencia | 0 cambios | 0 autores, 0 cambios
5  private void button1_Click(object sender, EventArgs e)
6  {
7      Conexion.Conectar();
8      string insertar = "INSERT INTO ALUMNO (CODIGO,NOMBRES,APELLIDOS,DIRECCION)VALUES(@CODIGO,@NOMBRES,@APELLIDOS,@DIRECCION)";
9      SqlCommand cmd1 = new SqlCommand(insertar,Conexion.Conectar());
10     cmd1.Parameters.AddWithValue("@CODIGO", txtcodigo.Text);
11     cmd1.Parameters.AddWithValue("@NOMBRES", txtnombres.Text);
12     cmd1.Parameters.AddWithValue("@APELLIDOS", txtapellidos.Text);
13     cmd1.Parameters.AddWithValue("@DIRECCION", txtdireccion.Text);
14
15     cmd1.ExecuteNonQuery();
16
17     MessageBox.Show("Datos agregados correctamente");
18
19     dataGridView1.DataSource = llenar_grid();
20 }
21
22 1 referencia | 0 cambios | 0 autores, 0 cambios
23 private void button2_Click(object sender, EventArgs e)
24 {
25     Conexion.Conectar();
26     string actualizar = "UPDATE ALUMNO SET CODIGO=@CODIGO,NOMBRES=@NOMBRES,APELLIDOS=@APELLIDOS,DIRECCION=@DIRECCION WHERE CODIGO=@CODI
27     SqlCommand cmd2 = new SqlCommand(actualizar, Conexion.Conectar());
28     cmd2.Parameters.AddWithValue("@CODIGO", txtcodigo.Text);
29     cmd2.Parameters.AddWithValue("@NOMBRES", txtnombres.Text);
30     cmd2.Parameters.AddWithValue("@APELLIDOS", txtapellidos.Text);
31     cmd2.Parameters.AddWithValue("@DIRECCION", txtdireccion.Text);
32
33     cmd2.ExecuteNonQuery();

```



## LISTA DE RECURSOS

### Máquinas y Equipos

- Pc de Escritorio con todos sus componentes.



### Herramientas e insumos

- SQL Server
- Visual Studio
- CSS/HTML
- YouTube
- Navegador Web
- Google Meet
- Microsoft Word
- Java Script



## REFERENCIAS

### Páginas web

- Wikipedia: [https://es.wikipedia.org/wiki/Programación\\_por\\_capas](https://es.wikipedia.org/wiki/Programación_por_capas)
- Juan Peláez en  
Geeks: <https://geeks.ms/jkpelaez/2009/05/30/arquitectura-basada-en-capas/>
- MariaEugeniaArevaloLizardo <https://arevalomaria.wordpress.com/2010/12/02/introduccion-al-patron-de-arquitectura-por-capas/>  
<https://instintobinario.com/arquitectura-en-tres-capas/>

### Videos Referenciales

- Introducción a la programación N Capas:  
[https://youtu.be/M\\_M8T8G4Og8](https://youtu.be/M_M8T8G4Og8)
- Patrón de Arquitecturas por Capas  
<https://youtu.be/uFDT-NHPe-s>