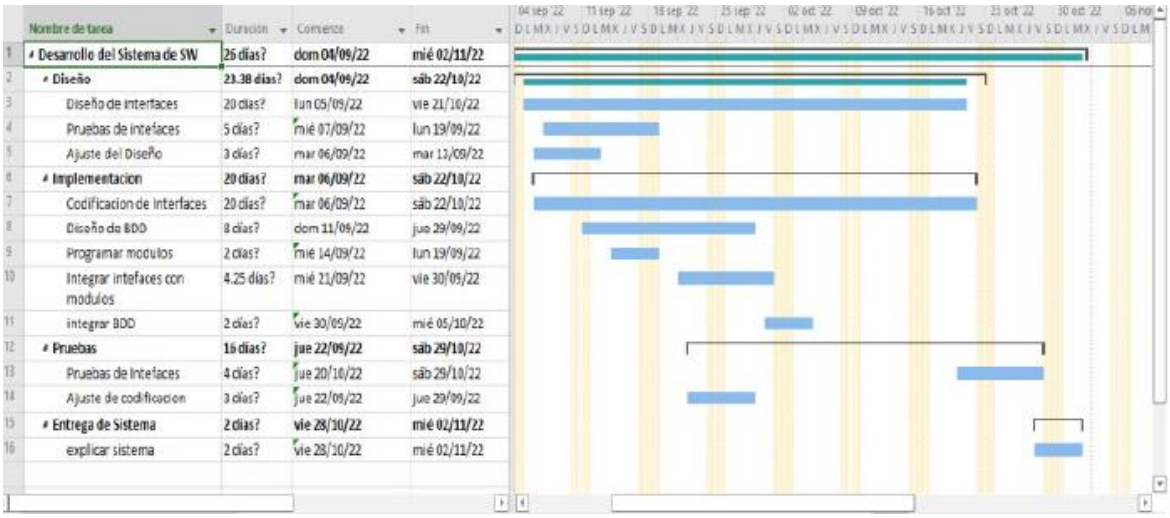
Fase de Diseño

1. planificación de trabajo



	Nombre de tarea	Duración	Comienzo	Fin
1	Desarrollo del Sistema de SW	26 días?	dom 04/09/22	mié 02/11/22
2	Diseño	23.38 días?	dom 04/09/22	sáb 22/10/22
3	Diseño de interfaces	20 días?	lun 05/09/22	vie 21/10/22
4	Pruebas de interfaces	5 días?	mié 07/09/22	lun 19/09/22
5	Ajuste del Diseño	3 días?	mar 06/09/22	mar 13/09/22
6	Implementacion	20 días?	mar 06/09/22	sáb 22/10/22
7	Codificacion de interfaces	20 días?	mar 06/09/22	sáb 22/10/22
8	Diseño de BDD	8 días?	dom 11/09/22	jue 29/09/22
9	Programar modulos	2 días?	mié 14/09/22	lun 19/09/22
10	Integrar intefaces con modulos	4.25 días?	mié 21/09/22	vie 30/09/22
11	integrar BDD	2 días?	vie 30/09/22	mié 05/10/22
12	Pruebas	16 días?	jue 22/09/22	sáb 29/10/22
13	Pruebas de Intefaces	4 días?	jue 20/10/22	sáb 29/10/22
14	Ajuste de codificacion	3 días?	jue 22/09/22	jue 29/09/22
15	Entrega de Sistema	2 días?	vie 28/10/22	mié 02/11/22
16	explicar sistema	2 días?	vie 28/10/22	mié 02/11/22

2. Diseño Arquitectónico

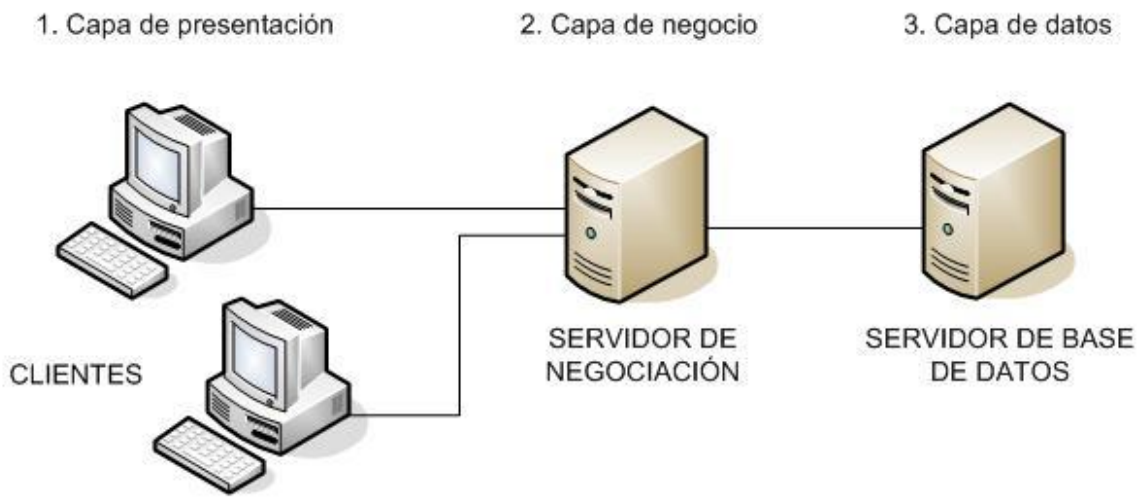
Arquitectura De Desarrollo De Software a Utilizar (Proyecto)

Los sistemas o arquitecturas en capas constituyen uno de los estilos que aparecen con mayor frecuencia mencionados como categorías mayores del catálogo, o, por el contrario, como una de las posibles encarnaciones de algún estilo más envolvente.

En [GS94] Garlan y Shaw definen el estilo en capas como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. (más de 100 patrones o variantes de este estilo).

EC – Arquitectura Cliente / Servidor

Un componente servidor, que ofrece ciertos servicios, escucha que algún otro componente requiera uno; un componente cliente solicita ese servicio al servidor a través de un conector. El servidor ejecuta el requerimiento (o lo rechaza) y devuelve una respuesta.

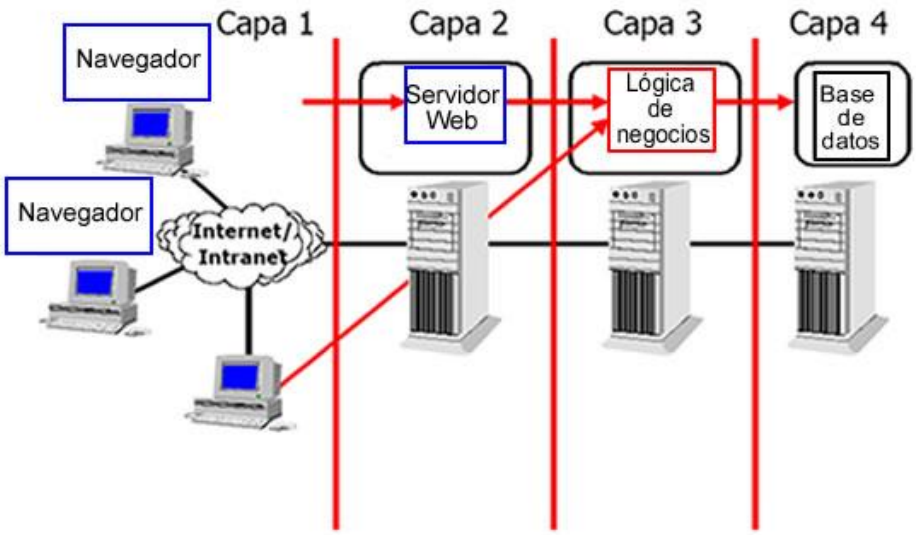


Ventajas de los Modelos en Capas

• El estilo soporta un diseño basado en niveles de abstracción crecientes, lo cual a su vez permite los implementadores la partición de un problema complejo en una secuencia de pasos incrementales.

• El estilo admite muy naturalmente optimizaciones y refinamientos.

• Proporciona amplia reutilización. Al igual que los tipos de datos abstractos, se pueden utilizar diferentes implementaciones o versiones de una misma capa en la medida que soporten las mismas interfaces de cara a las capas adyacentes. Esto conduce a la posibilidad de definir interfaces de capa estándar, a partir de las cuales se pueden construir extensiones o prestaciones específicas.



RESUMEN MODELO EN CAPAS

La arquitectura basada en capas como bien sabemos se enfoca en la distribución de roles y responsabilidades de forma jerárquica proveyendo una forma muy efectiva de separación de responsabilidades. El rol indica el modo y tipo de interacción con otras capas, y la responsabilidad indica la funcionalidad que está siendo desarrollada.

Estas son las características porque voy a usar esta arquitectura:

- Porque describe la descomposición de servicio de forma que la mayoría de la interacción ocurre solamente entre capas vecinas

- Porque las capas que se utilizaran en una aplicación pueden residir en la misma maquina física (misma capa) o puede estar distribuido sobre diferentes computadoras (n-capas).

- Porque los componentes de cada capa se comunican con otros componentes en otras capas a través de interfaces muy bien definidas.

Este modelo ha sido descrito como una “pirámide invertida de re-uso” donde cada capa agrega responsabilidad y abstracción a la capa directamente sobre ella.

Tipos de Métricas

Experiencia de Usuario

Usabilidad

En este aspecto se busca poner a prueba la capacidad que tiene el producto de software para ser entendido, aprendido y resultar atractivo para el usuario final, que es la persona o público objetivo para que utilice la herramienta que debe dar solución a muchas de sus necesidades

Accesibilidad

Se realiza la revisión del grado en el que las propiedades del software, permitan que sea utilizado por persona con capacidades diferentes también para quienes no tienen algún tipo de limitación física

Diseño

Formato

Al momento de empezar la codificación y desarrollar se mantiene una estética del software, ya que el no mantener un formato adecuado puede llegar a producir problemas en tiempo de ejecución por que no sabe como está ordenado el código

Diseño Arquitectónico

El respetar como se debe manejar el diseño elegido es sumamente importante ya que el diseño le permitirá dar estructura a su software, de está forma se puede revisar si se está respetando la forma en la que está conformada la arquitectura del software a desarrollar

Distribución

Es importante verificar la distribución de cada uno de los componentes claves y así mantener una excelente distribución que va a permitir al desarrollador no solo trabajar de forma más ordenada sino también de una manera más limpia evitando así cometer errores en el momento de codificar

Seguridad

Verificación Campos de texto

Se necesita buscar vulnerabilidades y los campos de texto, ya que son una de las principales fugas de seguridad, puesto que es el área en donde se tiene interacción con información ingresada por el usuario de esta forma se puede vulnerar la integridad del software y poner en peligro información confidencial

Tratamiento de datos

Así mismo el tratamiento del dato es importante y aquí es donde se verifica que se esté desarrollando con las medidas de seguridad tales como, utilizar procedimientos almacenados y no indicar que se va a realizar con la información en la aplicación, esto es para poder mejorar la calidad de nuestro software.

Métricas de Código

Revisión de Funciones

A nivel de código una de las métricas adecuadas es la revisión de la eficacia de las funciones puesto que se tiende a cargar de tareas innecesarias a una sola función, así mismo se trata de evitar este tipo de mala práctica al momento de desarrollar sin embargo se busca también revisar que tan eficaces son las funciones desarrolladas y que cumplan con su cometido