

# INSTITUTO TECNICO RICALDONE



## **Docente:**

Josué Guinea

## **Manual de requerimientos**

## **Integrantes:**

Josse Jared Díaz Ramírez (5/5)

Daniel Alexis Ramírez Martínez (5/5)

Lorien Odett De León Tejada (5/5)

Job Isaac Ventura Bolaños (5/5)

Ángel Fernando Chávez Viera (1/5)

## **Fecha de entrega:**

10/09/2021

## Índice

Introducción.....	3
Objetivos.....	4
General: .....	4
Específicos: .....	4
Requerimientos.....	6
IDE.....	7
SGBD.....	8
Características .....	9
Patrón de arquitectura.....	10
El modelo.....	10
Modelo del dominio.....	11
Modelo de la aplicación.....	11
La vista.....	11
El controlador.....	12
Topologías de red.....	12
Topología de bus o barra.....	12
Topología de estrella.....	13
Tipología de árbol.....	13
Topología de malla.....	14
El mas conveniente para nuestro sistema.....	14
Framework de diseño.....	15
Cuadro de requerimientos del sistema.....	16

## Introducción.

En este proyecto hablaremos sobre el proyecto llamado **DebussyStore** el cual es un programa de sucursal que servirá para poder realizar ventas, registrarlas, administración de la tienda, administración de los empleados y facturación de las compras.

Presentaremos por igual los programas que hemos utilizado para llevar a cabo todo este proyecto con mayor eficacia, así como también del **SGBD** (Sistema gestor de base de datos) que se ha utilizado en el proyecto, del entorno de desarrollo que hemos utilizado para la aplicación, el patrón de arquitectura que utilizamos en el proyecto por otro lado hablaremos también del **Framework** de diseño y de los requerimientos que va utilizar el programa los cuales van a estar divididos en dos partes, la primer parte que serán los requerimientos mínimos y la segunda parte que serán los requerimientos recomendados que por lo que su nombre lo indica son los que nosotros como creadores del proyecto recomendaremos al usuario disponer al momento de usar la aplicación .

Con este manual se busca también en informar sobre algunos aspectos de la creación del proyecto sobre como se hizo en el entorno de desarrollo que en este caso seria para nosotros el programa llamado **Visual Studio 2017** así como también poder comentar nuestros objetivos que tenemos de acuerdo con el proyecto ya sean estos objetivos generales o específicos, pero poder enseñarle al usuario cual fue/es nuestra intención con **DebussyStore**.

También como se ha venido trabajando en el modulo 5 de nuestras clases presentaremos en el manual algunas topologías de red y hablaremos en ello de la que mas corresponde para el proyecto en el cual estamos trabajando que sería este y unas generalidades sobre las topologías y lo que conllevan algunas de ellas al momento de ser utilizadas, así como de algunos problemas que cada una de estas presenta, por un lado, pero es beneficiosa por otro lado y al final dando una breve conclusión.

## Objetivos.

### General:

Poder desarrollar nuestras habilidades como estudiantes en desarrollo de software y tratar de explotar el mundo de la programación y lo que este conlleva al 100% y en el camino aprender muchísimas cosas nuevas en el transcurso del desarrollo de la aplicación y hacer de este proyecto un conocimiento extra para nosotros, así como poder agregar este proyecto mas adelante a nuestro e-portfolio de trabajos para mas adelante poder presentar nuestros proyectos esto pensado para algo a largo plazo ya que como estudiantes de esta carrera la cual en el ámbito laboral piden muestras de trabajos o alguna índole de experiencia este proyecto sería una buena idea para poder presentar junto con los demás que más adelante llegaran.

### Específicos:

1. Impulsar de cierta forma las nuevas tecnologías que se van desarrollando el mundo día con día y tratar de llevarlas al mercado, en nuestro caso el de ropa y hacer de la venta de esta un proceso más fácil y rápido.
2. Elevar las ventas en el segundo sector de la economía de nuestro país que serian los productos textiles (Ropa) ya que al hacer mejoraríamos la economía de El Salvador ya que por ende los productos generados se venderían a un ritmo más rápido lo que conllevaría a la mayor producción de estos a gran escala ya sean camisa, pantalones, zapatos, sombreros etc.
3. Hablando de casi lo mismo del segundo punto, generaríamos mas empleos en el sector de producción textil ya que como lo decíamos en el segundo lo que ocurre ahí conllevaría a más producción y más producción se refiere a más trabajo que hacer y más trabajo que hacer se refiere a más empleos.

## **“Manual de requerimientos”**



**Instituto Técnico Ricaldone.**

## **“Buenos cristianos y honrados ciudadanos”**

### **Integrantes del equipo:**

Daniel Alexis Ramírez Martínez

Job Isaac Ventura Bolaños

Lorien Odett De León Tejada

Josse Jared Díaz Ramírez

Ángel Fernando Chávez Viera

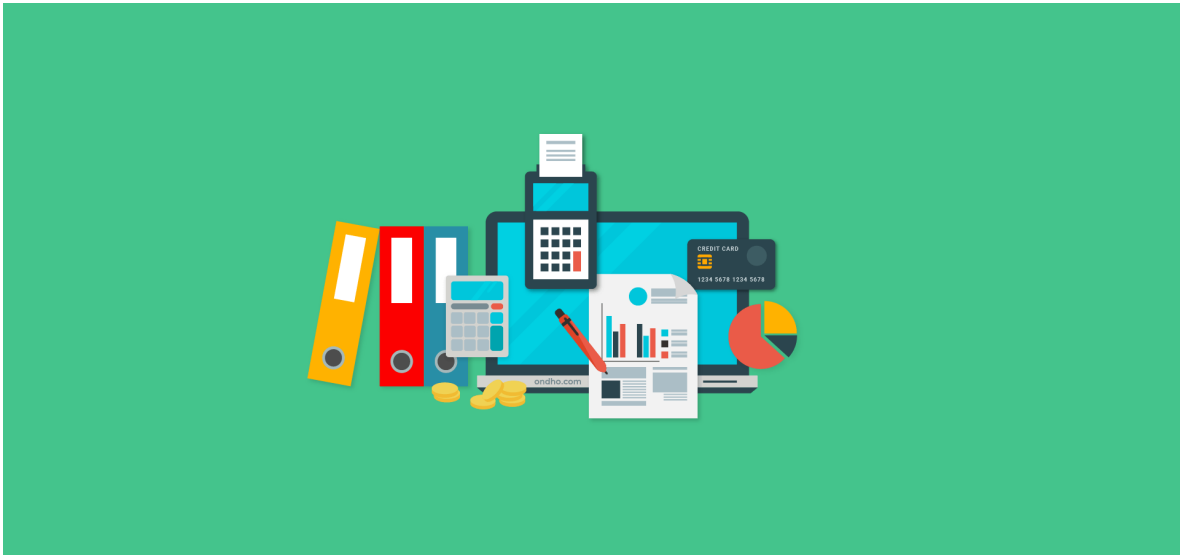
**Primer año de bachillerato – Desarrollo de software “B2”**

**Tarea 2 – Fase 2 Expo técnica.**

**El Salvador 2021 – ITR.**

## Requerimientos.

A continuación, mostraremos los requerimientos de todo nuestro proyecto donde mostraremos las cosas y programas variado que utiliza nuestro proyecto, así como dándole un breve resumen a cada uno junto con su pequeña investigación.





El entorno de desarrollo utilizado en el proyecto es **Visual Studio**, este entorno te permite editar, depurar y compilar código para luego poder publicar una aplicación. Este programa es rico en funciones las cuales se pueden utilizar en muchos aspectos del desarrollo de software. Muchos mas **IDE** solo ofrecen un editor y depurador estándar, **Visual Studio** incluye compiladores, herramientas de finalización de código, diseñadores gráficos y muchas demás funciones que en el proceso facilitan el desarrollo de software.

Al momento de abrir de un proyecto de **Visual Studio** aparecen varias ventanas o zonas de herramientas que serán de ayuda para nosotros las cuales son:

**Explorador de soluciones:** Este se encuentra situado arriba a la derecha del programa esta herramienta permite navegar, ver y administrar los archivos de nuestro código. El explorador de soluciones puede ayudar a organizar su código agrupando los archivos en soluciones y proyectos o poder llevar un patrón de arquitectura.

**Ventana del editor:** Se ubica en el centro del programa y es donde probablemente donde pasaremos la mayor parte de nuestro tiempo ya que esto muestra el contenido de los archivos. Esta zona es donde podemos editar código o diseñar una interfaz de usuario como por ejemplo una ventana con botones y cuadros de texto.

**Cambios en Git:** Esta herramienta se ubica abajo a la derecha, le permite realizar un seguimiento de los elementos de trabajo y compartir código con otros mediante tecnologías de control de versiones como Git y GitHub esto más que todo sirve para poder trabajar en equipo en proyectos, así como el caso de este proyecto denominado **DebussyStore**.

## SGBD.



El sistema gestor de base de datos que hemos utilizado para este proyecto es phpMyAdmin el cual es una herramienta de software la cual es gratuita y está escrita en php y esta destinado a manejar la administración de MySQL a través de la web. phpMyAdmin admite una amplia gama de operaciones en MySQL y MariaDB. Las operaciones de uso frecuente son: Administración de base de datos, tablas, columnas, relaciones, índices, usuarios, permisos y muchas cosas más.

Estos procesos se pueden realizar a través de la interfaz de usuario, mientras aun tiene la capacidad de ejecutar directamente cualquier declaración SQL.

Hay que destacar que phpMyAdmin cuenta con una amplia gama de documentación la cual podemos consultar y así también los usuarios pueden actualizar las paginas wiki donde pueden compartir idea, procedimientos y diversas operaciones.

Hay que recordar que phpMyAdmin es un proyecto maduro con una base de código estable y flexible. El proyecto phpMyAdmin es miembro de Software Freedom Conservancy que es una organización sin fines de lucro que ayuda a



promover, mejorar, desarrollar y defender proyectos de software libre y de código abierto.

## Características

1- Interfaz web intuitiva

2- Soporte para la mayoría de las funciones de MySQL

- Examinar y soltar bases de datos, tablas, vistas, campos e índices
- Crear, copiar, eliminar, renombrar y modificar bases de datos, tablas, campos e índices
- Servidor de mantenimiento, bases de datos y tablas, con propuestas de configuración del servidor
- Ejecutar, editar y marcar cualquier declaración SQL, incluso consultas por lotes
- Administrar cuentas de usuario y privilegios de MySQL
- Administrar procedimientos almacenados y disparadores

3- Importar datos de CSV y SQL

4- Exporte datos a varios formatos:

- CSV
- SQL
- XML
- PDF
- ISO
- IEC 26300
- Texto y hoja de cálculo OpenDocument, Word y otros

5- Administrar varios servidores

6- Creación de gráficos del diseño de su base de datos en varios formatos

7- Creando consultas complejas usando Query-by-example (QBE)

- 8- Búsqueda global en una base de datos o en un subconjunto de ella
- 9- Transformar los datos almacenados en cualquier formato utilizando un conjunto de funciones predefinidas, como mostrar datos BLOB como imagen o enlace de descarga.

## Patrón de arquitectura.



El patrón de arquitectura utilizado para nuestro proyecto es el patrón MVC, durante toda la década, algunos lenguajes fueron construyendo gradualmente el paradigma de programación orientada a objetos y estableciendo conceptos como tales como objetos, clases, encapsulación, herencia y polimorfismo, si es cierto que los lenguajes que empezaron esto ya no siguen vigentes pero los conceptos que dejaron en el mundo del desarrollo de software aun siguen en auge en la actualidad y son la base de lenguajes modernos como C++, Java o C#.

SmallTalk también fue el primer lenguaje de programación que permitió diseñar interfaces de usuario con múltiples "ventanas" desplegadas en una misma pantalla, concepto que después fue aplicado por GEMS, Macintosh, X11, Windows y otras interfaces gráficas de usuario modernas. El concepto central detrás de las librerías de interfaz de usuario provistas por SmallTalk está basado en el patrón de diseño MVC, creado por el profesor Trygve Reenskaug.

## El modelo.

El modelo es un conjunto de clases que representan la información del mundo real que el sistema debe procesar, así por ejemplo un sistema de administración de datos climatológicos tendrá un modelo que representará la

temperatura, la humedad ambiental, el estado del tiempo esperado, etc. sin tomar en cuenta ni la forma en la que esa información va a ser mostrada ni los mecanismos que hacen que esos datos estén dentro del modelo, es decir, sin tener relación con ninguna otra entidad dentro de la aplicación.

El modelo desconoce la existencia de las vistas y del controlador. Ese enfoque suena interesante, pero en la práctica no es aplicable pues deben existir interfaces que permitan a los módulos comunicarse entre sí, por lo que SmallTalk sugiere que el modelo en realidad esté formado por dos submódulos: El modelo del dominio y el modelo de la aplicación. El presente artículo utiliza la propuesta de SmallTalk por ser la base de la implementación de la librería Swing que es vista más adelante.

### Modelo del dominio.

Se podría decir que el modelo del dominio es el conjunto de clases que un ingeniero de software modela al analizar el problema que desea resolver; así, pertenecerían al modelo del dominio: El cliente, la factura, la temperatura, la hora, etc. El modelo del dominio no debería tener relación con nada externo a la información que contiene.

### Modelo de la aplicación.

El modelo de la aplicación es un conjunto de clases que se relacionan con el modelo del dominio, que tienen conocimiento de las vistas y que implementan los mecanismos necesarios para notificar a éstas últimas sobre los cambios que se pudieren dar en el modelo del dominio. El modelo de la aplicación es llamado también coordinador de la aplicación.

### La vista.

Las vistas son el conjunto de clases que se encargan de mostrar al usuario la información contenida en el modelo. Una vista está asociada a un modelo, pudiendo existir varias vistas asociadas al mismo modelo; así, por ejemplo, se

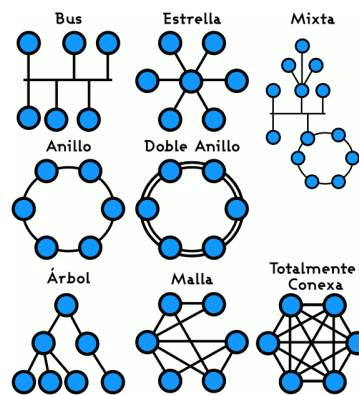
puede tener una vista mostrando la hora del sistema como un reloj analógico y otra vista mostrando la misma información como un reloj digital.

Una vista obtiene del modelo solamente la información que necesita para desplegar y se actualiza cada vez que el modelo del dominio cambia por medio de notificaciones generadas por el modelo de la aplicación.

## El controlador.

El controlador es un objeto que se encarga de dirigir el flujo del control de la aplicación debido a mensajes externos, como datos introducidos por el usuario u opciones del menú seleccionadas por él. A partir de estos mensajes, el controlador se encarga de modificar el modelo o de abrir y cerrar vistas. El controlador tiene acceso al modelo y a las vistas, pero las vistas y el modelo no conocen de la existencia del controlador.

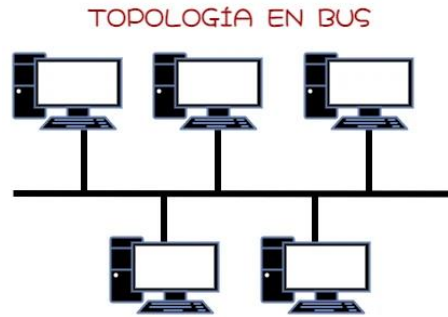
## Topologías de red.



La topología de una red representa la disposición de los enlaces que conectan los nodos de una red. Las redes pueden tomar muchas formas diferentes dependiendo de cómo están interconectados los nodos. Hay dos formas de describir la topología de una red: física o lógica.

## Topología de bus o barra.

Todos los nodos están conectados a un cable común o compartido. Las redes Ethernet normalmente usan esta topología.



### Topología de estrella.

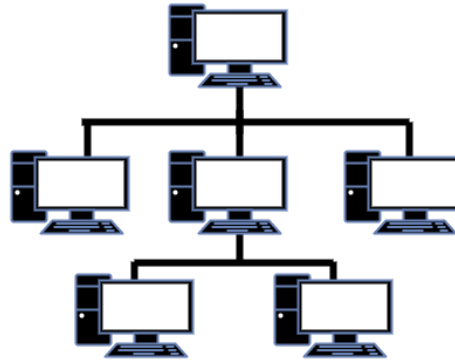
Cada nodo se conecta directamente a un concentrador central. En una topología de estrella todos los datos pasan a través del concentrador antes de alcanzar su destino. Esta es una topología común tanto en redes Ethernet como inalámbricas.



### Tipología de árbol.

Una combinación de las topologías de bus y estrella. Un conjunto de nodos configurados como estrella se conectan a una dorsal (Backbone).

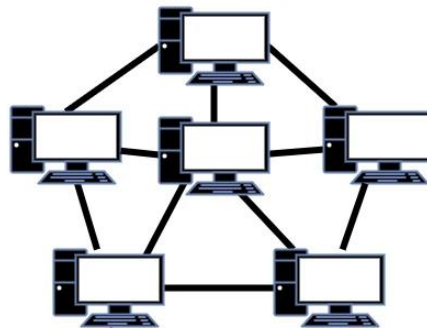
TOPOLOGÍA DE ÁRBOL



### Topología de malla.

Es una topología de red en la que cada nodo está conectado a todos los nodos. De esta manera es posible llevar los mensajes de un nodo a otro por distintos caminos. Si la red de malla está completamente conectada, no puede existir absolutamente ninguna interrupción en las comunicaciones. Cada servidor tiene sus propias conexiones con todos los demás servidores.

TOPOLOGÍA EN MALLA



### El mas conveniente para nuestro sistema.

La topología de árbol, luego de un breve análisis, vemos que es el que más cumple con los requerimientos de nuestro sistema, aparte que sus riesgos son mínimos ya que solo si el equipo principal llegara a caer solo aso se perdería la conexión a todos los dispositivos, a no ser que se respalde un switch o la información se maneje por un router, además no genera colisiones ni saturación de datos ya que hay un canal para cada equipo, y eso es una gran ventaja para nuestro sistema ya que esta contemplado que vayan a ser para varias personas ya que es una tienda.

## Framework de diseño.



En el proyecto de **DebussyStore** se esta utilizando el framework que ya viene con **Visual Studio** el cual es el .NET Framework 4.6.1.

.NET Framework 4.6.1 Developer Pack es un paquete que permite a los desarrolladores crear aplicaciones destinadas a .NET Framework 4.6.1 mediante **Visual Studio 2015, Visual Studio 2013, Visual Studio 2012** o entornos de desarrollo integrados **IDE** de terceros. Es un paquete único que incluye .NET Framework 4.6.1, .NET 4.6.1 Targeting Pack y el kit de desarrollo de software **SDK** .NET 4.6.1. Los paquetes de idioma asociados para Developer Pack contienen los recursos específicos del idioma para .NET Framework 4.6.1, .NET 4.6.1 **SDK** y el idioma IntelliSense.

Este Framework incluye junto con los siguientes componentes:

- .NET Framework 4.6.1 Developer Pack (NDP461-DevPack-KB3105179-**ENU**.exe): contiene .NET Framework 4.6.1, .NET 4.6.1 Targeting Pack, .NET 4.6.1 **SDK** y **ENU** IntelliSense.
- Paquete de idioma de .NET Framework 4.6.1 Developer Pack (NDP461-DevPack-KB3105179- (Lang) .exe): contiene el paquete de idioma de .NET Framework 4.6.1 correspondiente, el paquete de idioma de .NET 4.6.1 SDK y el idioma IntelliSense.

Cosa importante a tener en cuenta es que este paquete no debe utilizarse si tenemos la intención de distribuir .Net Framework. En ese caso, debe descargar un instalador web el cual se encuentra en la página de Microsoft.

### Cuadro de requerimientos del sistema.

Requerimientos de computador que funcionara como cliente			
Estas características u requerimientos sirven para poder llevar un buen funcionamiento en el computador de escritorio o laptop, entre los requerimientos que se necesitan mejor ira el programa en el ordenador, viceversa si son peores			
Características		Requerimientos mínimos	Requerimientos óptimos
	<b>Sistema operativo (32 bits– 64 bits)</b>	Windows vista – Windows XP - Windows 7, sistema operativo de MacOS	Windows 10 – Windows 11 – Sistema operativo MacOS más reciente
	<b>Navegador web</b>	Google Chrome, Microsoft Edge en sus versiones más recientes	Google Chrome, Microsoft Edge en sus versiones más recientes
	<b>Procesador</b>	Intel Core 2 Dúo a 1,0 GHz o superior	Intel Core i3 a 2,5 GHz o superior
	<b>Memoria RAM</b>	2 GB de RAM	4 GB de RAM
	<b>Espacio de almacenamiento</b>	500 MB de espacio disponible	1 GB de espacio disponible
	<b>Tarjeta de red</b>	10 Mbit/s Mbps – 100 Mbit/s Mbps de velocidad	1000 Mbit/s Mbps de velocidad



**Conexión a  
internet**

Se necesita conexión a internet  
para su descarga.  
Requiere como 1 Mbps de  
descarga y 0.50 Mbps de subida

Se necesita conexión a internet  
para su descarga.  
Requiere 5 Mbps de descarga y  
2 Mbps de subida