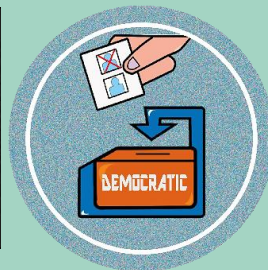


SALESIANOS

INSTITUTO TÉCNICO RICALDONE

BACHILLERATO



## “MANUAL DE REQUERIMIENTOS - SISTEMA DE GESTIÓN DE VOTOS”

**Docente:** Josué Guinea

**Grado y sección:** 1º Software 1ª

**Integrante:**

**Nº:**

**Autoevaluación:**

- Elías Sánchez Lenny Adrián
- Mejía Martínez Mónica Abigail
- Méndez Barrera Gabriela Susana
- Vargas Hernández Eduardo Ernesto

#10

5

#15

4

#16

5

#21

4

San Salvador, 22 de agosto de 2020.

# Índice

## Contenido

<b>1. Introducción.....</b>	<b>3</b>
<b>2. Objetivos.....</b>	<b>4</b>
<b>3. Manual de requerimientos.....</b>	<b>5</b>
<b>4. Requerimientos .....</b>	<b>6</b>
<b>4.1 Entorno de desarrollo .....</b>	<b>6</b>
<b>4.2 Sistema gestor de base de datos .....</b>	<b>6</b>
<b>4.3 Patrón de arquitectura .....</b>	<b>7</b>
<b>4.4 Topologías de red.....</b>	<b>9</b>
<b>4.4.1 Topología Híbrida o Mixta.....</b>	<b>9</b>
<b>4.4.2 Topología en estrella .....</b>	<b>10</b>
<b>4.4.3 Topología en Anillo .....</b>	<b>11</b>
<b>4.4.4 Topología en Malla .....</b>	<b>11</b>
<b>4.5 Framework de diseño.....</b>	<b>13</b>
<b>5. Requerimientos del sistema.....</b>	<b>14</b>

## 1. Introducción

En el presente manual técnico de requerimientos sobre **“Sistema de gestión de votos”** contiene la explicación de distintos puntos de información sobre el Sistema, como por ejemplo cuales son los programas necesarios para correr el programa, así como también los requerimientos recomendados para que el programa pueda funcionar correctamente en cualquier equipo.

El propósito principal de la elaboración de este manual es el de explicarle al usuario todo lo necesario para que el programa pueda ser instalado en su maquina sin ninguna dificultad facilitando el uso de la aplicación así como también el manual induce a que el usuario aprenda ciertos conceptos que desconocía como el de entornos de desarrollo.

Se explica apartados como, el entorno de desarrollo a utilizar, el sistema gestor de bases de datos y el patrón de arquitectura utilizado; también explica algunos tipos de topologías de red existentes, y el Framework de diseño de Visual Studio ocupado en la creación del Sistema.

El manual tiene como objetivo el dar a conocer a los usuarios los requerimientos mínimos del programa, para que estos no tengan problemas con su ejecución. Por ello también se ha elaborado un cuadro con los requerimientos mínimos y necesarios del Sistema de gestión de votos.

La importancia de la elaboración de este manual es que al no realizarlo dejaríamos al usuario con muchas dudas que tendríamos luego que responder nosotros cargándonos de más trabajo y haciendo que algunos usuarios al no comprender que hacer prefieren mejor no instalar el sistema, por cuál es recomendable la realización de este.

## **2. Objetivos**

### **Objetivo General:**

- Definir los requerimientos necesarios del sistema por medio de una investigación para que este tenga el funcionamiento óptimo y se ejecute de manera estable.

### **Objetivos específicos:**

- Describir los requerimientos operativos fundamentales mediante la creación de un cuadro comparativo para el desarrollo del sistema.
- Dar a conocer el IDE utilizado para la creación del sistema, por medio de una investigación y el uso de la aplicación que proporciona servicios integrales para comprender la factibilidad que proporciona al dar su uso.
- Reconocer las topologías de red por medio de un análisis para así poder determinar cuál es la más adecuada para el funcionamiento adecuado de nuestro sistema.

# MANUAL DE REQUERIMIENTOS

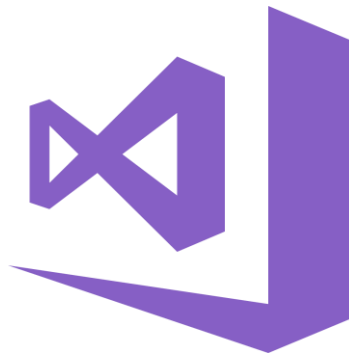


## **4. Requerimientos**

### **4.1 Entorno de desarrollo**

El entorno de desarrollo a utilizar será Visual Studio, es un conjunto de herramientas y otras tecnologías de desarrollo de software basado en componentes para crear aplicaciones eficaces y de alto rendimiento, permitiendo a los desarrolladores crear sitios y aplicaciones web, así como otros servicios web en cualquier entorno que soporte la plataforma.

En palabras más específicas, Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan todos el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML.



### **4.2 Sistema gestor de base de datos**

El sistema a utilizar para gestionar la base de datos sea MySQL Workbench es una herramienta visual unificada para arquitectos, desarrolladores y administradores de bases de datos. MySQL Workbench proporciona modelado de datos, desarrollo de SQL y herramientas de administración integrales para la configuración del

servidor, administración de usuarios, respaldo y mucho más. MySQL Workbench está disponible en Windows, Linux y Mac OS X.

MySQL Workbench proporciona una consola visual para administrar fácilmente los entornos MySQL y obtener una mejor visibilidad de las bases de datos. Los desarrolladores y administradores de bases de datos pueden utilizar las herramientas visuales para configurar servidores, administrar usuarios, realizar copias de seguridad y recuperación, inspeccionar datos de auditoría y ver el estado de la base de datos.

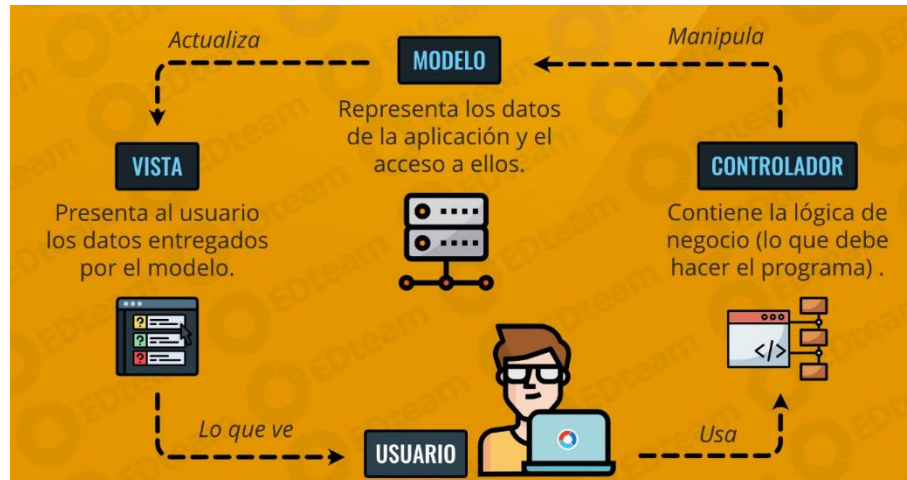


#### **4.3 Patrón de arquitectura**

El más conocido patrón de arquitectura es el Model-View-Controller (MVC: Modelo-Vista-Controlador). Es un estilo de arquitectura de software que nos permite tener una organización adecuada de nuestro código. Separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

El MVC nos permite separar los componentes de nuestra aplicación dependiendo de la responsabilidad que tienen, esto significa que cuando hacemos un cambio en alguna parte de nuestro código, esto no afecte otra parte de este. Esto respeta el principio de la responsabilidad única. Es decir, una parte de tu código no

debe de saber qué es lo que hace toda la aplicación, sólo debe de tener una responsabilidad.



**El modelo:** contiene una representación de los datos que maneja el sistema, es el encargado de la comunicación con la base de datos.

**La Vista:** o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos de interacción con éste.

**El Controlador:** actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

### Ventajas:

- La implementación se realiza de forma modular (subdividir una aplicación en partes más pequeñas).
- Las modificaciones a las vistas no afectan al modelo de dominio, simplemente se modifica la representación de la información, no su tratamiento.



- MVC está demostrado ser un patrón de diseño bien elaborado pues las aplicaciones que lo implementan presentan una extensibilidad y una mantenibilidad únicas comparadas con otras aplicaciones basadas en otros patrones.

#### **Desventajas:**

- Para desarrollar una aplicación bajo el patrón de diseño MVC es necesario una mayor dedicación en los tiempos iniciales del desarrollo. Normalmente el patrón exige al programador desarrollar un mayor número de clases que, en otros entornos de desarrollo, no son necesarias.
- MVC es un patrón de diseño orientado a objetos por lo que su implementación es sumamente costosa y difícil en lenguajes que no siguen este paradigma.

### **4.4 Topologías de red**

#### **¿Qué es una Topología de Red?**

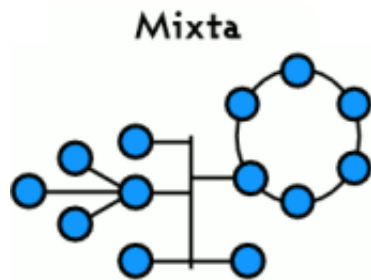
La topología de red se define como una familia de comunicación usada por los computadores que conforman una red para intercambiar datos. En otras palabras, la forma en que está diseñada la red, sea en el plano físico o lógico. El concepto de red puede definirse como "conjunto de nodos interconectados". Un nodo es el punto en el que una curva se intercepta a sí misma. Lo que un nodo es concretamente, depende del tipo de redes a que nos refiramos.

#### **4.4.1 Topología Híbrida o Mixta**

Topología híbrida, las redes pueden utilizar diversas tipologías para conectarse, como por ejemplo en estrella. La topología híbrida es una de las más

frecuentes y se deriva de la unión de varios tipos de topologías de red, de aquí el nombre de híbridas. Ejemplos de topologías híbridas serían: en árbol, estrella-estrella, bus-estrella, etc.

Su implementación se debe a la complejidad de la solución de red, o bien al

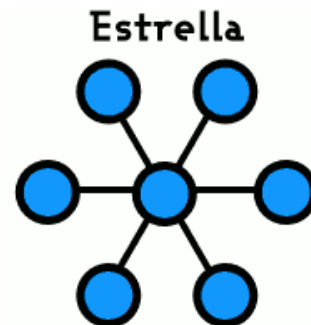


aumento en el número de dispositivos, lo que hace necesario establecer una topología de este tipo. Las topologías híbridas tienen un costo muy elevado debido a su administración y mantenimiento, ya que cuentan con segmentos de diferentes tipos, lo que obliga a invertir en equipo adicional para lograr la

conectividad deseada.

#### 4.4.2 Topología en estrella

Reduce la posibilidad de fallo de red conectando todos los nodos a un nodo central. Cuando se aplica a una red basada en la topología estrella este concentrador central reenvía todas las transmisiones recibidas de cualquier nodo periférico a todos los nodos periféricos de la red, algunas veces incluso al nodo que



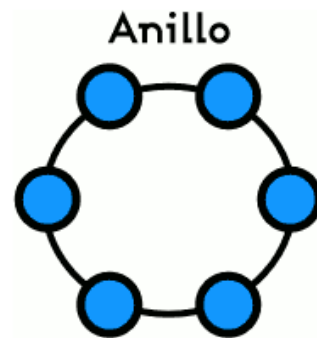
lo envió. Todos los nodos periféricos se pueden comunicar con los demás transmitiendo o recibiendo del nodo central solamente. Un fallo en la línea de conexión de cualquier nodo con el nodo central provocaría el aislamiento de ese nodo respecto a los demás, pero el resto de los sistemas permanecería intacto. El tipo de concentrador hub se utiliza en esta topología, aunque ya es muy obsoleto; se suele usar comúnmente un switch.

La desventaja radica en la carga que recae sobre el nodo central. La cantidad de tráfico que deberá soportar es grande y aumentará conforme vayamos agregando más nodos periféricos, lo que la hace poco recomendable para redes de gran tamaño. Además, un fallo en el nodo central puede dejar inoperante a toda la red. Esto último conlleva también una mayor vulnerabilidad de la red, en su conjunto, ante ataques.

#### **4.4.3 Topología en Anillo**

Si el nodo central es pasivo, el nodo origen debe ser capaz de tolerar un eco de su transmisión. Una red, en estrella activa, tiene un nodo central activo que normalmente tiene los medios para prevenir problemas relacionados con el eco.

Una red en anillo es una topología de red en la que cada estación tiene una única conexión de entrada y otra de salida. Cada estación tiene un receptor y un transmisor que hace la función de traductor, pasando la señal a la siguiente estación. En este tipo de red la comunicación se da por el paso de un token o testigo, que se puede conceptualizar como un cartero que pasa recogiendo y entregando paquetes de información, de esta manera se evitan eventuales pérdidas de información debidas a colisiones. En un anillo doble (Token Ring), dos anillos permiten que los datos se envíen en ambas direcciones (Token passing). Esta configuración crea redundancia (tolerancia a fallos). Evita las colisiones.



#### **4.4.4 Topología en Malla**

Una red en malla es una topología de red en la que cada nodo (Punto de conexión o de unión de varios elementos) está conectado a todos los nodos. De esta manera es posible llevar los mensajes de un nodo a otro nodo por distintos caminos. Si la red de malla está completamente conectada, no puede existir ninguna

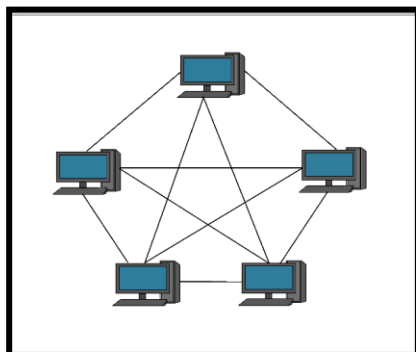
interrupción en las comunicaciones. Cada servidor tiene sus propias conexiones con todos los demás servidores.

A diferencia de los nodos en una topología en estrella, que requieren un router para entregar servicios de internet, los nodos de red pueden "hablar" directamente entre sí sin necesidad de la ayuda de una conexión a internet. Una gran ventaja de esta topología descentralizada es que no puede haber un único punto de fallo (SPOF por sus siglas en inglés de Single point of failure). Si un nodo ya no puede funcionar, todo el resto todavía puede comunicarse entre sí, directamente o a través de uno o más nodos intermedios.

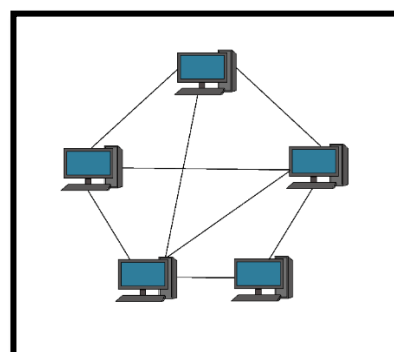


Las redes de mallas pueden utilizar una topología de malla completa o una topología de malla parcial. En una topología de malla completa, cada nodo de red está conectado a todos los otros nodos de red. En una Topología de malla parcial, al menos un nodo se conecta directamente a todos los demás nodos, mientras que otros pueden solo conectarse a esos nodos con los que intercambian datos de manera frecuente.

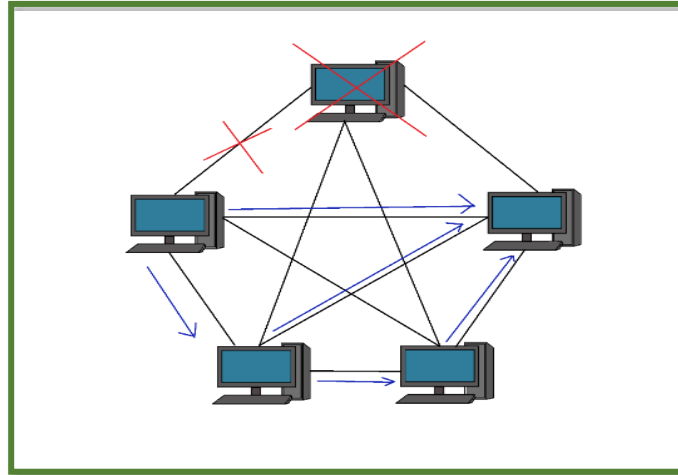
**Topología malla completa**



**Topología malla parcial**



Ejemplo de una Topología malla en caso de una interrupción:



### **Desventajas:**

El costo de la red puede aumentar en los casos que se implemente de forma alámbrica, la topología de red y las características de la misma implican el uso de una mayor cantidad de recursos.

### **4.5 Framework de diseño**

La Framework a implementar será la NET Framework 4.5.2 es una actualización local altamente compatible con Microsoft .NET Framework 4, Microsoft .NET Framework 4.5 y Microsoft .NET Framework 4.5.1. El paquete sin conexión se puede usar en los casos en que el instalador web no esté disponible por falta de conexión a Internet.



## Bunifu Framework



Los controles de la interfaz de usuario de Bunifu son herramientas basadas en DLL para ayudar a crear interfaces de aplicaciones de escritorio increíbles. Garantiza una gran experiencia de usuario en sus aplicaciones y reduce el tiempo de desarrollo para los desarrolladores de software Microsoft Visual Studio .NET.

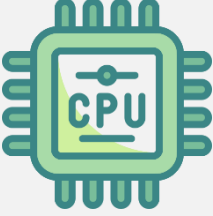
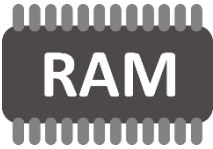
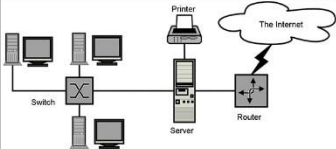



Bunifu UI y Bunifu Dataviz trabajan actualmente con lenguajes C# y VB.NET, diseñados específica y especialmente para aplicaciones de Windows Forms. Y el producto se ofrece como un archivo DLL importado a nuestro entorno de estudio visual.

Las DLL se instalan en su entorno .NET y se construyen sobre WinForms. Permite a los desarrolladores arrastrar y soltar, por lo tanto, una codificación más rápida.

### 5. Requerimientos del sistema

N°	Requerimiento	Valores	Descripción
1	 Resolución Mínima	1366 x 768 req. Mínimo – 1600 x 900 req. Máximo	Para un funcionamiento óptimo del programa se recomienda el uso de monitores con un requerimiento mínimo de 1366 x 768 y un requerimiento máximo de 1600 x 900.
2	 .NET Framework	Framework 4.5.2	Es necesario tener instalada la versión indicada para el funcionamiento del sistema.

3	 <p>Procesador</p>	Procesador de 1GHz o más rápido de 32 bits (x86) req. Mínimo	Es necesario un procesador para la velocidad con la que el programa será ejecutado.
4	 <p>RAM</p>	1GB de RAM (32 bits)	Está orientado a la arquitectura del sistema operativo.
5	 <p>Medios de conexión</p>	Medios físicos y de red (Preferible los medios físicos)	Conectado con cables de red RJ45 para una estabilidad garantizada y entendible comunicación con el servidor o encontrarse conectado por red al host.
6	 <p>Topología de Red</p>	Topología de Malla	Permite que todos los usuarios tengan acceso en cualquier momento que deseen ingresar.
7	 <p>Servidor</p>	XAMPP v3.2.4	Licencia gratuita y no requiere pago futuro, con bases de datos mariadb.