



UNIVERSIDAD TÉCNICA DE BABAHOYO

**FACULTAD DE ADMINISTRACIÓN, FINANZAS
INFORMÁTICA**

F.A.F.I.

EXAMEN COMPLEXIVO DE GRADO O DE FIN DE CARRERA

PRUEBA PRÁCTICA INGENIERÍA EN SISTEMAS

Tema:

**ANÁLISIS COMPARATIVO DE LOS LENGUAJES DE
PROGRAMACIÓN NODE JS Y ASP.NET PARA UN SISTEMA DE
REGISTRO DE LA “FARMACIA TU AHORRO” EN LA CIUDAD DE
BABAHOYO**

Autor:

WILLIAM ROMERO GARCÍA

Tutor:

ING. OMAR MONTECE MORENO

LOS RÍOS - BABAHOYO - ECUADOR

INTRODUCCION

Desde hace varios años atrás hasta la actualidad en el mundo tecnológico se ha venido implementando el lenguaje de programación, que de forma simple, se puede conceptualizar como el conjunto de instrucciones que permiten la comunicación de usuarios a computadoras; frente a esto se han creado variedades de lenguajes programados que desarrollen este papel, entre ellos se muestra los lenguajes de programación “Node Js”, dicho programa fue generado por los desarrolladores de Java Script y “Asp.net que se creó por Microsoft. Estos sistemas fueron transformando estas programaciones a tal punto que pasaran de solo usarse en el navegador, a que se usen como una aplicación independiente para los ordenadores.

Ambas programaciones permiten desarrollar distintas aplicaciones en la web, ya sean de sitios con pequeñas implementaciones, hasta sistema a mayor escala, preparadas para soportar cantidades de trabajo y operaciones a la vez. En ese sentido, este tipo de programa se aplica en todos los lugares del mercado, incluido el mercado farmacéutico, que requieran de un sistema tecnológico con mayores avances para hacer de su trabajo más eficaz.

La farmacia Tu Ahorro, es uno de esos establecimiento, esta fue creada con el propósito de posesionarse como una de las mejores farmacias en la ciudad de Babahoyo, esto implica ofrecer a la ciudadanía del lugar y de quienes requieran sus servicios, la más optima y apropiada atención, sin embargo, se ha podido evidenciar ciertas problemáticas que están afectando el desarrollo de este centro farmacéutico, dichos problemas vienen dados por el sistema de programación con el que han venido trabajando por mucho tiempo; este al ser ya algo antiguo no permite la búsqueda rápida de medicamento que en su mayoría se encuentran en stop, generando retraso en la atención al cliente, otros de las causas que afectan a la farmacia, es que la entrega de las facturas debido a la lentitud del

sistema, demoren en hacerse físico, impacientando al cliente, haciéndolos considerar a la farmacia como un lugar ineficiente en el servicio, no se debe dejar a un lado también que en ocasiones los valores que se registran por las ventas del día no siempre son completas, entonces esto provoca un descuadre en las cuentas al cerrar caja.

El estudio de caso está enmarcado en la línea de investigación Sistemas de Información y Comunicación, Emprendimiento e Innovación, sostenida por la sublínea de Redes y Tecnologías Inteligentes de Software y Hardware. Además, el estudio de caso maneja una metodología de carácter descriptivo, que permitirá recopilar la información requerida, cuenta como técnica investigación el análisis documental, para ello utilizaremos referencias bibliográficas obtenidas de sitios web, libros o documentos que traten del tema en estudio.

Dado a lo anteriormente menciona, se debe resaltar que la farmacia Tu Ahorro requiere de una inmediata regeneración de su sistema de lenguajes de programación, y para ello se han considerado a dos lenguajes Node Js y Asp.net que pueden ser útiles para evitar problemas que afecten en su crecimiento empresarial, por tal razón se plantea la siguiente pregunta problema; ¿Cuál de los lenguajes de programación, Node Js y Asp.net, es el más apropiado para mejorar el sistema de registro en la farmacia Tu Ahorro, en la ciudad de Babahoyo.

DESARROLLO

JavaScript es un lenguaje de programación que se creó en Netscape como una herramienta de secuencias de comandos para manipular páginas web dentro de su navegador, parte del modelo comercial de Netscape era vender servidores web, que incluían un entorno llamado Netscape LiveWire que podía crear páginas dinámicas utilizando JavaScript del lado del servidor, pero no se dio el éxito esperado y JavaScript del lado del servidor no se popularizó hasta hace poco, con la introducción de Node.js.

Un factor clave que condujo al surgimiento de Node.js fue el momento. Solo unos años antes, JavaScript había comenzado a ser considerado como un lenguaje más serio, gracias a las aplicaciones "Web 2.0" (como Flickr, Gmail, etc.) que mostraron al mundo cómo podría ser una experiencia moderna en la web. (Wiki, 2020).

Los motores de JavaScript también mejoraron considerablemente, ya que muchos navegadores compitieron para ofrecer a los usuarios el mejor rendimiento. Los equipos de desarrollo detrás de los principales navegadores trabajaron arduamente para ofrecer un mejor soporte para JavaScript y encontrar formas de hacer que JavaScript se ejecute más rápido. El motor que Node.js usa bajo el capó, V8 (también conocido como Chrome V8 por ser el motor JavaScript de código abierto de The Chromium Project), mejoró significativamente debido a esta competencia.

Node.js se creó en el lugar correcto y en el momento correcto, pero la suerte no es la única razón por la que es popular hoy en día. Presenta una gran cantidad de ideas y enfoques innovadores para el desarrollo del lado del servidor de JavaScript que ya han ayudado a muchos desarrolladores.

Node.JS

Es un entorno en tiempo de ejecución multiplataforma de código abierto del lado del servidor en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como por ejemplo, servidores web. Fue creado por Ryan Dahl en 2009 y su evolución está apadrinada por la empresa Joyent, que además tiene contratado a Dahl en plantilla. (Google, 2021).

Sus aplicaciones se ejecutan en un solo proceso, sin crear un nuevo hilo para cada solicitud. Node.js proporciona un conjunto de primitivas de E/S asíncronas en su biblioteca estándar que evitan que el código JavaScript se bloquee y, en general, las bibliotecas en Node.js se escriben utilizando paradigmas que no bloquean, lo que hace que el comportamiento de bloqueo sea la excepción en lugar de la norma.

Cuando Node.js realiza una operación de E/S, como leer de la red, acceder a una base de datos o al sistema de archivos, en lugar de bloquear el subproceso y desperdiciar ciclos de CPU en espera, Node.js reanudará las operaciones cuando regrese la respuesta. (Trujillo, s.f.).

Esto permite que Node.js maneje miles de conexiones simultáneas con un solo servidor sin presentar la carga de administrar la concurrencia de subprocesos, lo que podría ser una fuente importante de errores, incluye un entorno REPL para depuración interactiva

CARACTERISTICAS

Concurrencia: Funciona con un modelo de evaluación de un solo subproceso, utilizando entradas y salidas asíncronas que pueden ejecutarse hasta cientos de miles simultáneamente sin la sobrecarga asociada con el cambio de contexto. El diseño de compartir un solo hilo para ejecutar todas las solicitudes satisface las necesidades de las aplicaciones altamente sincrónicas, donde cada proceso que ingresa y sale debe tener una función de devolución de llamada. Una desventaja de este enfoque de subproceso único es que Node.js requiere módulos adicionales, como clústeres, para escalar la aplicación con la cantidad de núcleos de procesador en la máquina en la que se ejecuta.

V8: es un entorno de tiempo de ejecución de JavaScript diseñado para Google Chrome, es gratuito desde el año 2008, fue escrito en C que compila el código fuente de JavaScript en código de máquina en lugar de interpretarlo en tiempo real.

Libuv: Es una capa de abstracción para funciones de red y sistema de archivos en sistemas Windows y POSIX como Linux, Mac OS X y Unix. Las operaciones básicas en Node.js están escritas en JavaScript con métodos de soporte escritos en C++.

Módulos: Combina varios "módulos centrales" que se agrupan en binario, como el módulo de red, que proporciona una capa para la programación de red asíncrona, y otros módulos básicos como ruta, sistema de archivos, caché, temporizador y un subproceso más general. Se pueden utilizar módulos desarrollados por terceros, ya sea como archivos ".node" precompilados o como archivos JavaScript normales, se implementan de acuerdo con la especificación CommonJS para módulos, utilizando una variable de exportación

para dar acceso a estos scripts a las funciones y variables que implementan los módulos. (Jd, 2021).

Los módulos de terceros pueden extender node.js o agregar un nivel de abstracción e implementar varias utilidades de middleware para usar en aplicaciones web, como marcos de conectores y expresiones. Si bien los módulos se pueden instalar como archivos simples, por lo general se instalan con Node Package Manager (npm), lo que facilita la compilación, instalación y actualización de módulos, así como la administración de dependencias; los estereotipos que no están instalados en la tarjeta de administración virtual de los botones deberán usar un camino relativo para encontrar.

Desarrollo homogéneo entre cliente y servidor: Se puede combinar con bases de datos de documentos (como MongoDB o CouchDB) y bases de datos relacionales como MySQL, PostgreSQL y otras, lo que le permite desarrollar en un único entorno de desarrollo. Con la adaptación de patrones de desarrollo del lado del servidor como MVC y sus variantes MVP, MVVM, etc. Node.js facilita la reutilización de código del mismo estilo de interfaz entre el lado del cliente y el lado del servidor.

Bucle de eventos: Se registra con el sistema operativo y cada vez que el cliente establece una conexión, se ejecuta la devolución de llamada, durante el tiempo de ejecución de Node.js, cada conexión recibe una pequeña asignación dinámica de espacio de memoria sin necesidad de crear un subproceso de ejecución pero a diferente de otros eventos dirigidos por los eventos de Node.js, los eventos no se llaman claramente, sino que se activan al final de cada función de recuperación, su gestión de eventos termina cuando no hay evento.

Ventajas de Node.js

Node.JS es respetado como la opción líder para el diseño de microservicios, además, Node.JS puede ser una innovación frágil. Es muy sencillo para los ingenieros escalar las aplicaciones tanto en nivel como en vertical. Es realmente sencillo incluir más microservicios sobre el mejor del existente.

Debido a que sirve tanto a clientes como a aplicaciones del lado del servidor, se lo conoce como JS de pila completa. Después de esto, no es necesario contratar ingenieros aislados para mejorar el back-end y el front-end. Esto ahorra mucho tiempo, dinero y esfuerzos. (Lima, 2021).

La comunidad de diseñadores de Node.JS está enérgica continuamente con la mejora y mejora de Node.JS, en un futuro cercano, se anticipa que será testigo de un ecosistema más positivo y rico en arreglos.

Tabla 1. Ventajas de Node js

Elaborado por: William Romero García

ASP.NET

Es un entorno para aplicaciones web desarrollado y comercializado por Microsoft, y los desarrolladores pueden usar este marco para crear sitios web, aplicaciones web y servicios web dinámicos. Enero de 2002 con la versión 1.0 de .NET Framework, que es el sucesor de la tecnología Active Server Pages (ASP). ASP.NET se basa en Common Language Runtime, lo que permite a los programadores escribir código ASP.NET en cualquier idioma compatible con .NET Framework. (Sites, 2021).

La Empresa de Microsoft introdujo una tecnología llamada Active Server Pages en diciembre de 1996 siendo este un entorno de aplicación gratuito y abierto donde puede integrar componentes HTML, secuencias de comandos y ActiveX del lado del servidor para crear soluciones eficientes y dinámicas para la web.

Forma parte de Internet Information Services (IIS) desde la versión 3.0 y es una tecnología de página activa que permite utilizar archivos. Varios comandos y elementos se combinan con HTML tradicional para representar páginas generadas dinámicamente. Active Server Pages (Tareas, 2020).

Después del lanzamiento del Internet Information Services 4.0 en 1997, Microsoft comenzó a investigar las posibilidades para un nuevo modelo de aplicaciones web que pudiera resolver las quejas comunes sobre ASP, especialmente aquellas con respecto a la separación de la presentación y el contenido y ser capaz de escribir código "limpio".



CARACTERISTICAS

Páginas: Las páginas de ASP.NET, conocidas oficialmente como "web forms" (formularios web), son el principal medio de construcción para el desarrollo de aplicaciones web. Los formularios web están contenidos en archivos con una extensión ASPX; en jerga de programación, estos archivos típicamente contienen etiquetas HTML o XHTML estático, y también etiquetas definiendo Controles Web que se procesan del lado del servidor y Controles de Usuario donde los desarrolladores colocan todo el código estático y dinámico requerido por la página web. Adicionalmente, el código dinámico que se ejecuta en el servidor puede ser colocado en una página dentro de un bloque `<% -- código dinámico -- %>` que es muy similar a otras tecnologías de desarrollo como PHP, JSP y ASP, pero esta práctica es, generalmente, desaconsejada excepto para propósitos de enlace de datos pues requiere más llamadas cuando se genera la página; no sólo funciona sobre el servidor de Microsoft IIS, también lo hace sobre Apache. (Vela, 2021).

El modelo Code-behind: Microsoft recomienda que para realizar programación dinámica se use el modelo code-behind, o de respaldo, que coloca el código en un archivo separado o en una etiqueta de script especialmente diseñada. Los nombres de los archivos code-behind están basados en el nombre del archivo ASPX tales como MiPagina.aspx.cs o MiPagina.aspx.vb (esta práctica se realiza automáticamente en Microsoft Visual Studio y otros entornos de desarrollo). Cuando se usa este estilo de programación, el desarrollador escribe el código correspondiente a diferentes eventos, como la carga de la página, o el clic en un control, en vez de un recorrido lineal a través del documento.

El modelo code-behind de ASP.NET marca la separación del ASP clásico y alienta a los desarrolladores a construir aplicaciones con la idea de presentación y contenido separados en mente. En teoría, esto permite a un diseñador web, por ejemplo, enfocarse en la creación del diseño con menos posibilidades de alterar el código de programación mientras lo hace. Esto es similar a la separación en el Modelo Vista Controlador

Controles de usuario: ASP.NET permite la creación de componentes reutilizables a través de la creación de Controles de Usuario, sigue la misma estructura que un formulario web, excepto que los controles derivan de la clase `System.Web.UI.UserControl`, y son almacenados en archivos ASCX. Como los archivos ASPX, un ASCX contiene etiquetas HTML o XHTML, además de etiquetas para definir controles web y otros controles de usuario. También pueden usar el modelo code-behind. (Salazar, 2020).

Los programadores pueden agregar sus propias propiedades y métodos, y manejadores de eventos. Un mecanismo de eventos en burbuja proporciona la capacidad de pasar un evento disparado por un control de usuario a la página que lo contiene .

Administración del estado: Las aplicaciones ASP.NET son alojadas en un servidor web y se tiene acceso a ellas mediante el protocolo sin estado HTTP, que no guarda ninguna información sobre conexiones anteriores. Por lo tanto, si la aplicación requiere interacción entre conexiones, tiene que implementar su propia administración del estado. ASP.NET proporciona varias maneras de administrar el estado de las aplicaciones ASP.NET.

Estado de la aplicación: El estado de la aplicación (Application state) es una colección de variables definidas por el usuario que son compartidas por todas las invocaciones de una aplicación ASP.NET. Estas son establecidas e inicializadas cuando el evento `Application_OnStart` se dispara en la carga de la primera instancia de las

aplicaciones y están disponible hasta que la última instancia termina. Las variables de estado o variables de sesión de la aplicación son identificadas por nombres.

✚ **Estado de la sesión:**El estado de la sesión (Session state) es una colección de variables definidas por el usuario, las cuales persisten durante la sesión de un usuario. Estas variables son únicas para diferentes instancias de una sesión de usuario, y son accedidas usando la colección `Session`. Las variables de sesión pueden ser preparadas para ser automáticamente destruidas después de un determinado tiempo de inactividad, incluso si la sesión no ha terminado. Del lado del cliente, una sesión de usuario es identificada por una cookie o codificando el ID de la sesión en la misma URL.

ASP.NET proporciona tres modos de persistencia para variables de sesión:

InProc: Las variables de sesión son mantenidas dentro del proceso. Sin embargo, en este modo, las variables son destruidas cuando el proceso ASP.NET es reciclado o terminado.

StateServer: En este modo, ASP.NET ejecuta un servicio de Windows separado que mantiene las variables de estado. Como esta administración de estado ocurre fuera del proceso ASP.NET, tiene un impacto negativo en el rendimiento, pero permite a múltiples instancias de ASP.NET compartir el mismo estado del servidor, permitiendo que una aplicación ASP.NET pueda tener su carga balanceada y escalada en múltiples servidores. También, como el servicio de administración del estado se ejecuta independiente de ASP.NET, las variables pueden persistir a través de las finalizaciones del proceso ASP.NET.

SqlServer: En este modo, las variables de estado son almacenadas en un servidor de base de datos, accesible usando SQL. Las variables de sesión pueden persistir a través de finalizaciones de procesos también en este modo.

Estado de la vista: se refiere al mecanismo de administración de estado a nivel de página que utilizan las páginas HTML generadas por las aplicaciones ASP.NET para mantener el estado de los controles y elementos de la interfaz de usuario en los formularios web. El estado de los controles se cifra y se envía al servidor en cada envío de formulario en un campo oculto llamado `__VIEWSTATE`. El servidor devuelve variables para que cuando se vuelva a ver la página, los campos vuelvan a su estado final. En el lado del servidor, la aplicación puede cambiar el estado de la vista, si los resultados de la vista actualizan el estado de cualquier control. El estado de los controles individuales en el servidor se decodifica y está disponible para su uso en ASP.NET mediante la colección `ViewState`.

Motor de plantillas: Al ser liberado, ASP.NET carecía de un motor de plantillas. Debido a que .NET framework es orientado a objetos y permite herencia, muchos desarrolladores podrían definir una nueva clase que herede desde "System.Web.UI.Page", escribir métodos en ella que generen HTML y hacer que las páginas de su aplicación hereden de esta clase. Aunque esto permite que haya elementos comunes dentro de un sitio web, añade complejidad y mezcla código fuente con lenguaje de etiquetas. Además, este método puede visualizarse únicamente al ejecutar la aplicación, no mientras se está diseñando. Otros desarrolladores usan la inclusión de archivos y otros trucos para evitar la implementación de enlaces de navegación y otros elementos en cada página.

ASP.NET 2.0 presentó el concepto de página maestra (Master Page), que permite el desarrollo de páginas basado en plantillas web. Una aplicación web puede tener una o más páginas maestras, las cuales pueden ser anidadas. Las plantillas maestras contienen controles contenedores, llamados `ContentPlaceHolders` para indicar donde irá el contenido dinámico, además de HTML y JavaScript que será compartido a través de las páginas hijas.

Las páginas hijas también usan esos controles ContentPlaceholder, que deben ser relacionados con el ContentPlaceholder de la página maestra que contiene a esta página hija. El resto de la página está definido por las partes compartidas de la página maestra. Todo el lenguaje de marcado y controles de servidor en la página de contenido deben ser colocadas dentro del control ContentPlaceholder.

Cuando una solicitud es hecha por una página de contenido, ASP.NET mezcla la salida de la página de contenido con la salida de la página maestra, y envía el resultado al usuario. La página maestra permanece completamente accesible a la página del contenido. Esto significa que la página de contenidos puede manipular los encabezados, cambiar el título, configurar la cache, etc. Si la página maestra expone propiedades públicas o métodos, el contenido de la página puede utilizar estos también.

Hoy en día con el nuevo producto de Microsoft WebMatrix, se dio a conocer el nuevo motor de plantillas conocido como Razor. Ya antes el IDE de Visual Studio incluía otros motores. Facilitando a las páginas ASP.NET Modelo Vista Controlador.

La estructura de los directorios de ASP.NET generalmente puede ser determinada por las preferencias del desarrollador. Aparte de unos pocos nombres de directorios reservados, el sitio puede expandirse a cualquier número de directorios. La estructura es típicamente reflejada directamente en las urls. Los nombres de directorios de ASP.NET tenemos:

- ✚ **App_Browsers:** Contiene archivos de definición específicos para navegadores.
- ✚ **App_Code:** El servidor ASP.NET traduce automáticamente los archivos (y subdirectorios) en ese directorio en una colección a la que se puede acceder desde cualquier página del sitio. App_Code se usa comúnmente para códigos de acceso a datos, códigos de formulario o códigos comerciales. Además, en esta guía se incluyen todos los identificadores http específicos del sitio, los módulos de publicación y los servicios web. En lugar de usar App_Code, el desarrollador

puede optar por proporcionar un ensamblado independiente con código precompilado.

- ✚ **App_Data:** Directorio por defecto para las bases de datos, tales como archivos mdb de Microsoft Access y archivos mdf de Microsoft SQL Server. Este directorio es usualmente el único con permisos de escritura en la aplicación.
- ✚ **App_LocalResources:** Contiene archivos de recursos localizados para páginas individuales del sitio.
- ✚ **App_GlobalResources:** Contiene archivos **resx** con recursos localizados disponibles para cada página del sitio. Este es donde el desarrollador ASP.NET típicamente almacenara mensajes que serán usados en más de una página.
- ✚ **App_Themes:** usado para temas alternativos del sitio.
- ✚ **App_WebReferences:** Usado para archivos de descubrimiento y archivos WSDL para referencias a servicios web para ser consumidos en el sitio.
- ✚ **Bin:** Contiene código compilado (archivos .dll) para controles, componentes, y otro código que pueda ser referenciado por la aplicación. Cualquier clase representada por código en la carpeta Bin es automáticamente referenciada en la aplicación. Son archivos o librerías que tienen como principal acción ejecutar una función cuando estas son llamadas o se invocan.

ASP.NET actualmente admite tres modelos de programación:

- ✚ ASP.NET Web Forms
- ✚ ASP.NET MVC
- ✚ ASP.NET Web Pages.

Cada modelo construye la aplicación de formas completamente diferentes, aprovecha diferentes métodos de desarrollo y satisface las necesidades de diferentes perfiles de

usuario. Algunas características que son fortalezas de algunos paradigmas de programación pueden considerarse debilidades de otros. ¿Qué es más importante, el desarrollo a un alto nivel de abstracción o el control total sobre todos los aspectos de la aplicación? Simplicidad en comparación con el control. Flexibilidad vs. Eficiencia. Son estas compensaciones las que deben tenerse en cuenta al elegir. En esta serie, veremos las diferencias entre los tres paradigmas de programación y los escenarios favorables para cada uno.

ASP.NET Web Forms fue el primero de los tres modelos de programación en existir, y proporciona un gran nivel de abstracción con un modelo de programación familiar basado en eventos y controles que favorece la productividad mediante la programación declarativa reduciendo la cantidad de código necesaria para implementar una determinada funcionalidad.

ASP.NET MVC se concibió como alternativa a Web Forms y proporciona un modelo de programación basado en el popular patrón de arquitectura MVC. Entre sus principales características destacan su completa integración con pruebas unitarias y su separación más clara entre la lógica de presentación, la lógica de negocio y la lógica de acceso a datos.

ASP.NET Web Pages es el último de los tres modelos de programación y ha sido creado en respuesta a la creciente demanda de desarrolladores web que no tienen experiencia previa con ASP.NET, para quienes introducir ASP .NET Web Forms o MVC es una inversión. El tiempo inicial es demasiado grande. La web proporciona un modelo de programación más fácil y rápido sin renunciar a todas las funciones y la flexibilidad de ASP.NET. En la actualidad una aplicación .NET puede ejecutarse de dos formas distintas que las explicaremos a continuación:

✚ **Aplicaciones cliente/servidor:** Estas aplicaciones generalmente están en coordinación traducida. Pueden combinar toda la abundancia de la interfaz de usuario, y este es el caso con las aplicaciones de rendimiento y productividad, pero la lógica del trabajo no se encuentra como un recurso que se puede utilizar nuevamente. Además, es poco probable que se ejecuten y puedan expandir otras aplicaciones.

✚ **Aplicaciones que utilizan el navegador:** Estas aplicaciones cuentan con una interfaz web muy rica y útil. La GUI integra muchas tecnologías como HTML, XHTML, scripts, etc. Siempre que el navegador utilizado soporte estas tecnologías.

Ventaja de Asp.net
<ul style="list-style-type: none"> ✚ Disminuye radicalmente la suma de código necesario para construir aplicaciones enormes.
<ul style="list-style-type: none"> ✚ Con la verificación de Windows integrada y la disposición por aplicación, sus aplicaciones están seguras y protegidas.
<ul style="list-style-type: none"> ✚ Ofrece una ejecución superior al aprovechar las primeras administraciones autorizadas, la compilación justo a tiempo, la optimización local y el almacenamiento en caché desde el primer momento.
<ul style="list-style-type: none"> ✚ El sistema se complementa con un rico conjunto de herramientas y un creador dentro del entorno de avance de coordenadas de Visual Studio. La alteración WYSIWYG, los controles del servidor de arrastrar y soltar y la implementación automática son algunos de los aspectos más destacados que ofrece este eficaz instrumento.

Tabla 2. Ventajas de Asp.net

Elaborado por: William Romero García

ASP .NET	NODE.JS
Es un marco de aplicación web de código abierto creado por Microsoft.	Es un entorno de tiempo de ejecución de JavaScript multiplataforma de código abierto que ejecuta código JavaScript en el lado del servidor.
Suele usarse en un sitio de la era moderna utilizando formas web Html 5, JavaScript, CSS.	Lenguaje de secuencias de comandos del lado del servidor.
Fácil de memorizar en comparación con otros dialectos de programación, la mejora del sitio utilizando Asp.net es excepcionalmente simple.	Emplea un espectáculo poco convencional impulsado por eventos y está planeado para componer aplicaciones web adaptables, especialmente servidores web; obtiene una gran ejecución basada en los modelos de numerosas aplicaciones web.
Existen numerosos sitios web en el mundo actual que se han creado en Asp.net.	Soporte y manejo de volúmenes expansivos de información de clientes.
Está escrito en lenguajes .NET como c #.	Está escrito en código abierto C, C++, Javascript.
Codificación algo fácil.	Codificación algo complicada.

Tabla 3. Comparaciones de Asp.net y Node js

Elaborado por: William Romero García

CONCLUSIONES

Finalmente podemos concluir con este estudio de caso acotando que tenemos en comparativa a dos Lenguajes de Programación dentro del Top 10 de los más usados Node.js es una apuesta segura, especialmente para micro servicios, aplicaciones web, aplicaciones en tiempo real como serian mensajería instantánea, chat en vivo y software de transacciones de comercio electrónico.

Dado es el caso que ciertos desarrolladores deseen incluir ciertas herramientas de C # pueden ir con ASP.NET, ya que está añade un glosario de elementos para crear aplicaciones de calidad de manera eficiente. Al ser OpenSource tiende a poseer una comunidad enorme y herramientas que brindan apoyo a sus desarrolladores para a futuro brindar buenas soluciones informáticas, ASP.NET también será una buena opción para aplicaciones de pequeña escala.

Node.js es un lenguaje de secuencias de comandos del lado del servidor que se ejecuta en el lado del servidor; creado por Ryan Dahl, detalla un entorno de tiempo de ejecución JavaScript multiplataforma de código abierto capaz de ejecutar código JavaScript del lado del servidor dicho esto solo se usa para la validación del lado del cliente en node.js, y cumple su función tanto en el front-end como en el back-end. Mientras que ASP.Net es un marco de aplicación web de código abierto. Es un lenguaje de secuencias de comandos del lado del servidor que crea páginas web dinámicas. Node.js ofrece un rendimiento y una velocidad excepcionales y es una solución ideal para desarrollar aplicaciones de mensajería o chat.

BIBLIOGRAFÍA

- Google, I. (10 de Septiembre de 2021). *Google Arte y Cultura*. Obtenido de <https://artsandculture.google.com/entity/m0bbxf89?hl=es>
- Jd, A. (15 de Marzo de 2021). *Copia ficheros asíncrona JavaScript, cómo compilar JavaScript a ejecutable*. Obtenido de <https://proyectoa.com/copia-ficheros-asincrona-javascript-como-compilar-javascript-a-ejecutable/>
- Lima, A. (19 de Enero de 2021). *Diferencia entre Node.js y Asp.Net*. Obtenido de <https://es.acervolima.com/diferencia-entre-node-js-y-asp-net>
- Salazar, M. (1 de Mayo de 2020). *Issuu*. Obtenido de https://issuu.com/melanisalazar3137/docs/presentaci_n1
- Sites, G. (11 de Mayo de 2021). *Programación VI Umca*. Obtenido de <https://sites.google.com/site/programacionviumca/tema-02>
- Tareas, B. (15 de Marzo de 2020). *Historia de Asp*. Obtenido de <https://www.buenastareas.com/ensayos/Historia-De-Asp/69381445.html>
- Trujillo, S. (s.f.). *PeakU*. Obtenido de ¿Qué significa "primitivas de E/S asíncronas" en nodejs?: <https://peaku.co/es/preguntas/17702-%C2%BFque-significa-%26quot;primitivas-de-e-s-asincronas%26quot;-en-nodejs>
- Vela, M. (11 de Junio de 2021). *Asp.Net*. Obtenido de Winxgo: <https://winxgo.com/2021/06/11/asp-net/>
- Wiki, T. (1 de Junio de 2020). *Una breve historia de Node.js*. Obtenido de <https://tech-wiki.online/es/node-history.html>

ANEXOS

ENTREVISTA

Tema: Análisis comparativo de los lenguajes de programación Node.js y Asp.Net para un sistema de registro de la farmacia tu ahorro de la ciudad de Babahoyo

Dirigido: Trabajadores de la empresa

Objetivo: Obtener información concreta

Fecha: 19/07/2022

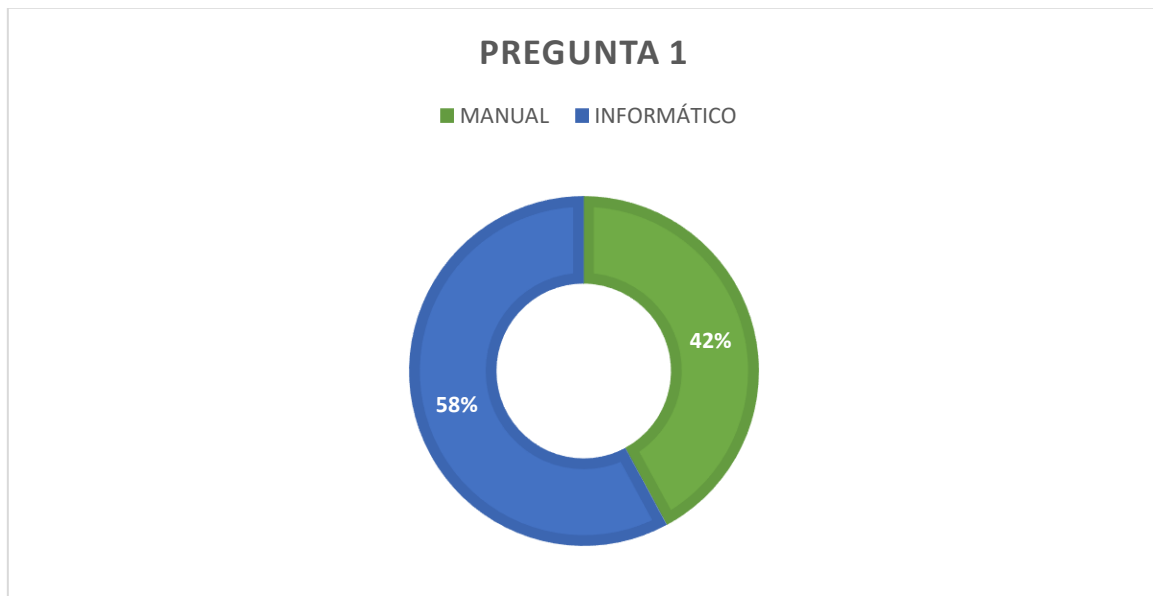
PREGUNTAS

- 1. ¿Qué tipo de proceso utilizan en la farmacia tu ahorro?**
- 2. ¿Qué tan eficiente son los procesos en la farmacia tu ahorro?**
- 3. ¿Está a gusto con el sistema de registro de la farmacia?**
- 4. ¿Si tuviera un mejor sistema de registro lo usaría?**
- 5. ¿Está dispuesto a cambiar el sistema de registro por uno más eficiente y rápido?**
- 6. ¿Usted cree que traerá beneficios cambiar el sistema de registro?**
- 7. ¿Conoce algún lenguaje de programación que realice sistemas de registro?**
- 8. ¿Qué tal le parece hacer una inversión para mejorar el sistema de registro?**

ANEXOS

1. ¿Qué tipo de proceso utilizan en la farmacia tu ahorro?

ALTERNATIVAS	FRECUENCIA	PORCENTAJE
MANUAL	40	42%
INFORMÁTICO	55	58%
TOTAL	95	100%



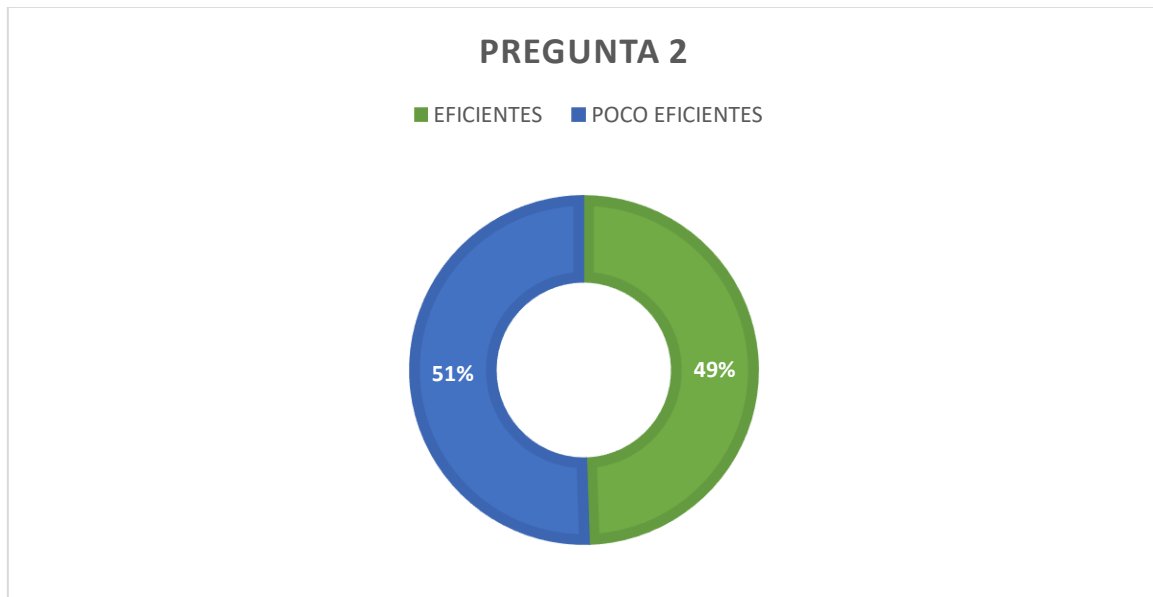
Análisis e Interpretación

Al consultar a los Trabajadores de la Farmacia nos arrojó como resultado que el 58% del personal usa procesos informáticos como medio de información y estadística, mientras tanto el 42% usa de forma manual con estos resultados.

La mayor cantidad de los trabajadores ha usado el Sistema informático con finalidades estadísticas ya que en este medio solo se usa para almacenar información más no como material de estudio en base a sus registros.

2. ¿Qué tan eficiente son los procesos en la farmacia tu ahorro?

ALTERNATIVAS	FRECUENCIA	PORCENTAJE
EFICIENTES	47	49%
POCO EFICIENTES	48	51%
TOTAL	95	100%

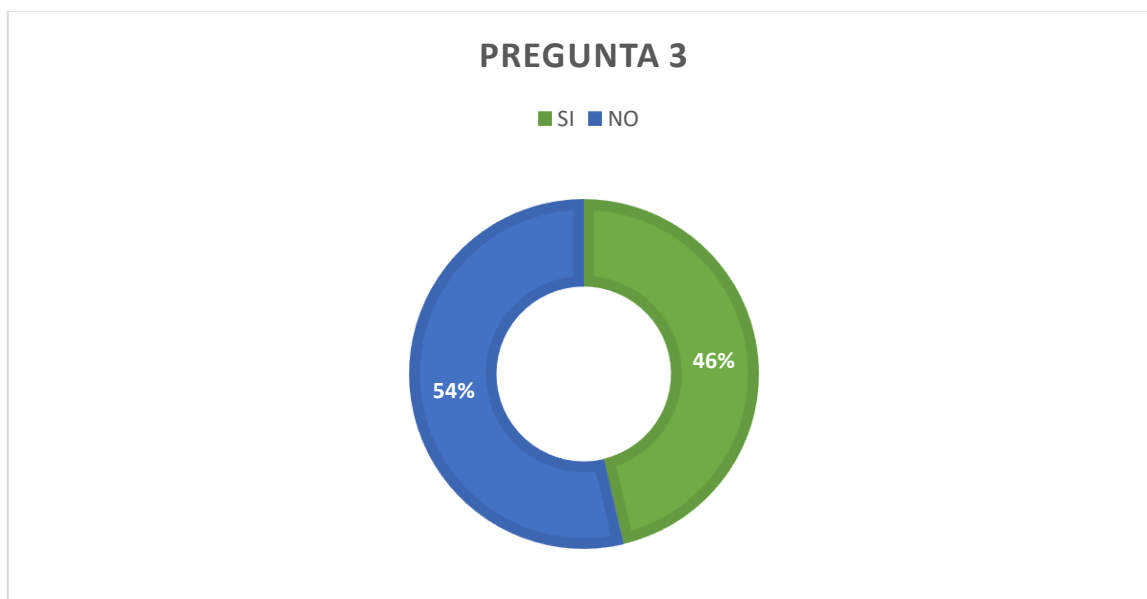


Análisis e Interpretación

Según los resultados obtenidos se determina que el 49% en la población usa una la aplicación para asegurar la autonomía y eficacia, mientras que el 51% afirma que tras usar herramientas los procesos son poco eficientes; sabiendo que los registros almacenados en una base de datos son información valiosa la mayor cantidad de la población nos indica que se puede usar estos registros con fines muy prácticos en diferencia a solo el almacenamiento.

3. ¿Está a gusto con el sistema de registro de la farmacia?

ALTERNATIVAS	FRECUENCIA	PORCENTAJE
SI	44	46%
NO	51	54%
TOTAL	95	100%

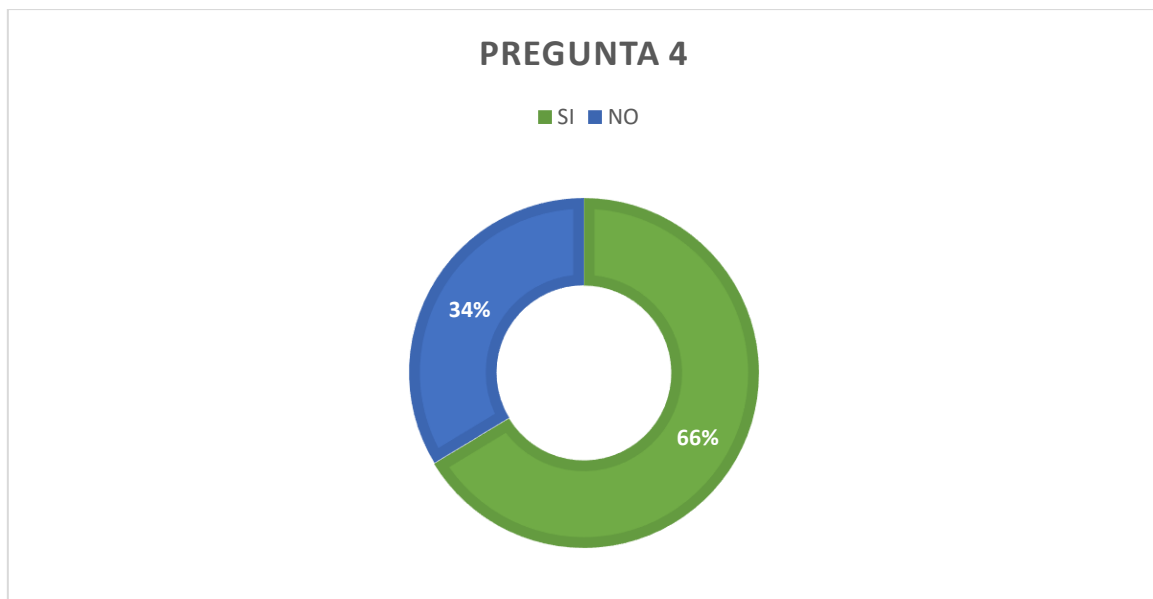


Análisis e Interpretación

Según los resultados obtenidos se determina que el 46% de los entrevistados se sienten a gusto con el actual Sistema informático de la Farmacia, mientras que el 54% parte no se siente apto con el uso del Sistema de registro. La mayor población tiene en cuenta que no está a gusto con el actual sistema,, se necesita tener un programa que permita realizar estas instrucciones por nosotros brindando los resultados esperados.

4. ¿Si tuviera un mejor sistema de registro lo usaría?

ALTERNATIVAS	FRECUENCIA	PORCENTAJE
SI	63	66%
NO	32	34%
TOTAL	95	100%

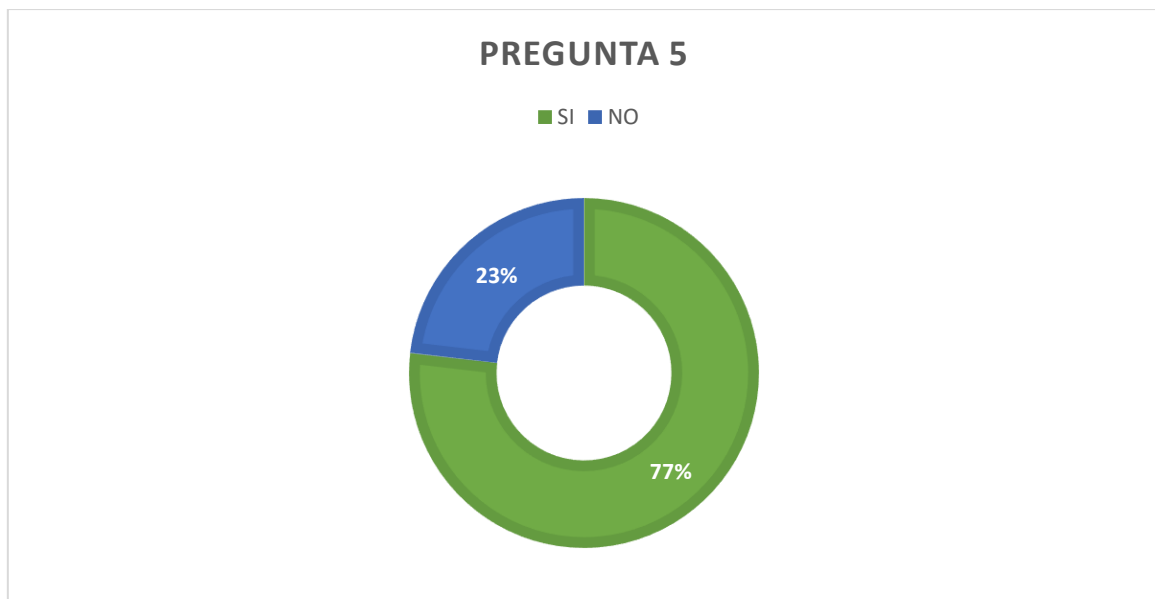


Análisis e Interpretación

Al consultar nos dio como resultado que el 66% de población quieren usar nuevas herramientas con interfaz mas amigables, mientras que el 34% de la población esta a gusto con el Sistema de registro que se ha trabajando. La mayor parte no se encuentra familiarizado con dicho aplicativo, misma razon que requieren un cambio de manera rapida y eficaz.

5. ¿Está dispuesto a cambiar el sistema de registro por uno más eficiente y rápido?

ALTERNATIVAS	FRECUENCIA	PORCENTAJE
SI	73	77%
NO	22	23%
TOTAL	95	100%

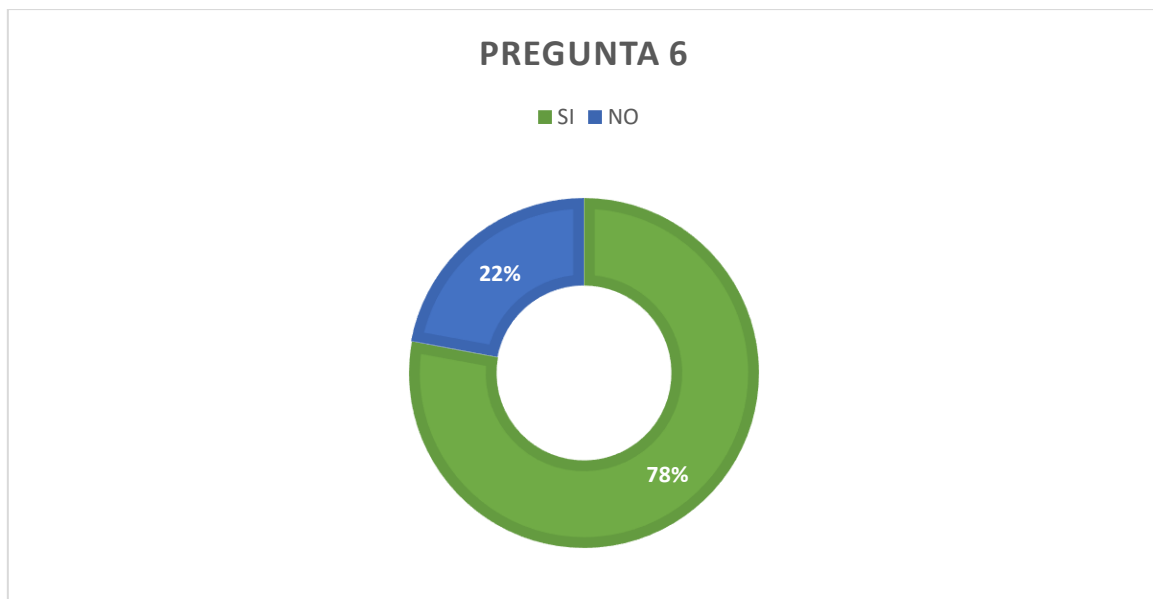


Análisis e Interpretación

Teniendo la muestra de la encuesta, gran parte de la población específicamente el 77% está consciente de que el cambio de Sistema es eminente, mientras que el 23% de la población tiene conocimientos acerca de manipular su actual medio de registros. Como no se tienen conocimiento actualizado, tampoco estarán asociados con la plataforma que esta usa para brindar resultados positivos.

6. ¿Usted cree que traerá beneficios cambiar el sistema de registro?

ALTERNATIVAS	FRECUENCIA	PORCENTAJE
SI	74	78%
NO	21	22%
TOTAL	95	100%

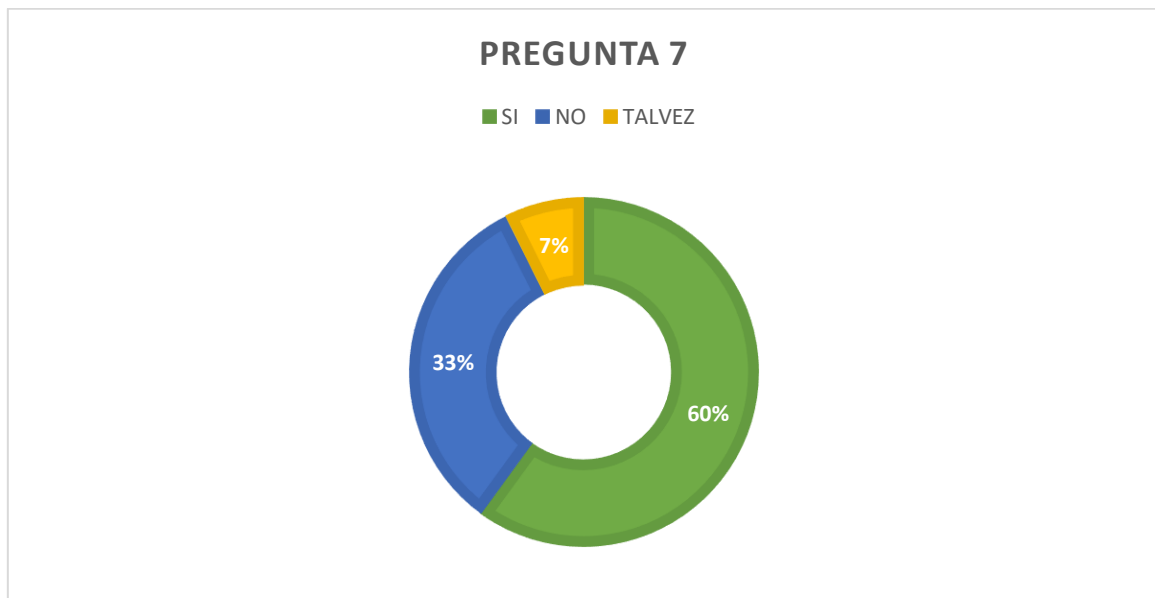


Análisis e Interpretación

Según el estudio realizado el 78% indicó que es factible actualizar dicho aplicativo para así poder tener una innovación en la misma, mientras que el 22% nos indica que no es necesario que se incorpore nada en el ambiente de trabajo. Aunque no se tenga mucha información es factible la utilización de nuevas tecnologías, esto demuestra la evolución en cuanto a la utilización de herramientas que no se conocen a profundidad pero que son muy indispensables para la vida profesional de cada individuo.

7. ¿Conoce algún lenguaje de programación que realice sistemas de registro?

ALTERNATIVAS	FRECUENCIA	PORCENTAJE
SI	57	60%
NO	31	33%
TALVEZ	7	7%
TOTAL	95	100%

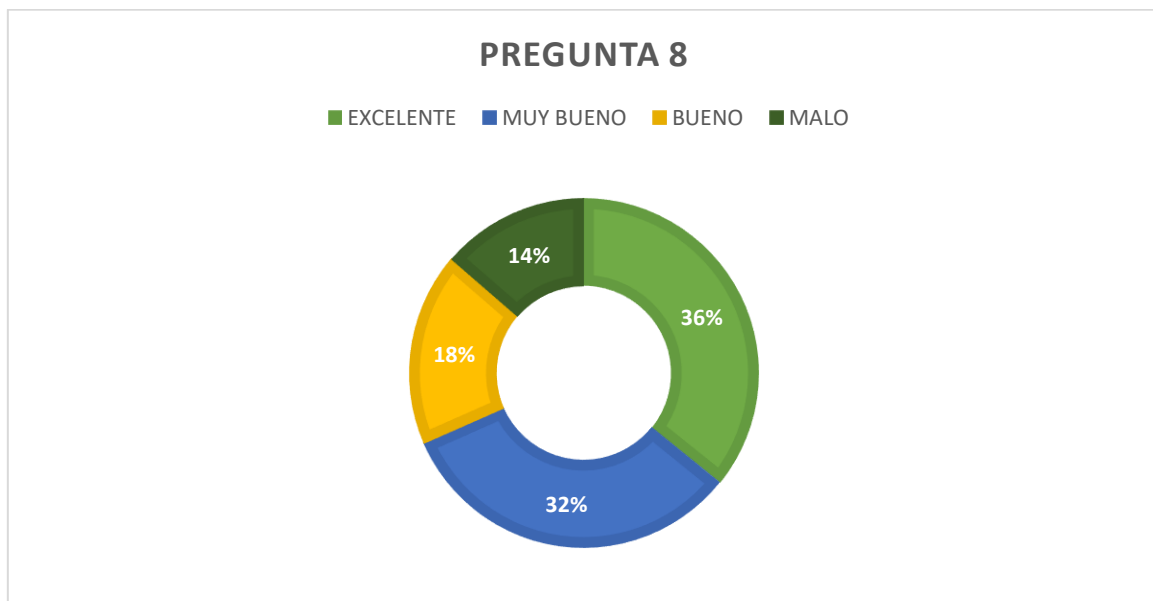


Análisis e Interpretación

De acuerdo a los resultados obtenidos se pudo determinar que el 60% de la población le encantan las cosas con velocidad y rapidez eficiente teniendo en consideración que al implementar una nueva aplicación será más intensa, mientras que el 33% nos dio a entender que podría ser de manera moderada y el otro 7% lenta. Gran parte de la población indicó que usando una nueva herramienta podría cumplir los procesos más rápidos.

8. ¿Qué tal le parece hacer una inversión para mejorar el sistema de registro?

ALTERNATIVAS	FRECUENCIA	PORCENTAJE
EXCELENTE	34	36%
MUY BUENO	31	32%
BUENO	17	18%
MALO	13	14%
TOTAL	95	100%



Análisis e Interpretación

De acuerdo a los resultados obtenidos se determina que el 36% población redacto que debe ser excelentes, el 33% muy buenos, mientras que 18% respondió que bueno y el 13% malo al hablar de velocidad de transferencia. Gran parte de la población indica que estará satisfecho con la velocidad que tiene la aplicación ya que al momento de realizar alguna operación no otorgará problema alguno.

Magister

Eduardo Galeas Guijarro

**DECANO DE LA FACULTAD DE ADMINISTRACIÓN FINANZAS E
INFORMÁTICA**

En su despacho.

De mis consideraciones:

Me dirijo a usted para dar contestación a la solicitud recibida por la Universidad Técnica de Babahoyo y la Facultad de Administración y Finanzas e Informática para que el Sr.: **WILLIAM ROBERTO ROMERO GARCIA** con cédula de identidad **1207487529**, estudiante de la carrera de Ingeniería en Sistemas matriculada en el proceso de titulación **PERIODO ABRIL 2022 – SEPTIEMBRE 2022**, trabajo de titulación modalidad Estudio de Caso para la obtención del grado académico profesional universitario de tercer nivel como INGENIERA EN SISTEMAS. El Estudio de caso es **ANÁLISIS COMPARATIVO DE LOS LENGUAJES DE PROGRAMACIÓN NODE JS Y ASP.NET PARA UN SISTEMA DE REGISTRO DE LA "FARMACIA TU AHORRO" EN LA CIUDAD DE BABAHOYO.**

Documento que remito a usted para los fines pertinentes.


Atentamente.

Vanessa Romero Montalvan

CI:1204020257