

**TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO**

**DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN
INGENIERÍA EN TECNOLOGÍAS DE INFORMACIÓN Y
COMUNICACIÓN.**

EMPRESA:

NovaTech.

PROYECTO:

NovaTech WebTech Manager.

ASESOR EXTERNO:

JOSÉ CRISPÍN HERNÁNDEZ HERNÁNDEZ.

INTEGRANTES:

JUAN MANUEL HERNANDEZ CORTEZ.

I .- Agradecimientos.

Quiero expresar mi más sincero agradecimiento a todas las personas y entidades que hicieron posible la realización de este proyecto.

En primer lugar, a mis padres, cuyo amor incondicional, apoyo y guía han sido fundamentales en cada paso de mi vida. Su esfuerzo y dedicación me han inspirado a seguir adelante y dar siempre lo mejor de mí.

A mis hermanas, quienes han sido una fuente constante de apoyo y motivación. Gracias por estar siempre presentes, brindándome su confianza y ánimo en los momentos en que más lo necesitaba.

A Yaz que a pesar de las adversidades que hemos pasado siempre ha sido fuente de apoyo incondicional e inspiración para poder seguir con estos proyectos.

A mis amigos, por su compañía y palabras de aliento durante este camino. Su fe en mis capacidades y su disposición para ayudarme hicieron que los retos fueran más llevaderos.

A mi negocio, PC Life, por ser el motor que me impulsó a crecer profesionalmente. Este proyecto no hubiera sido posible sin la experiencia y las lecciones que aprendí al trabajar en él. Agradezco también a nuestros clientes, quienes, con su confianza y apoyo, nos motivan día a día a innovar y mejorar.

Finalmente, a todas las personas que, de una u otra manera, contribuyeron al desarrollo de este proyecto. A cada uno de ustedes, gracias por ser parte de esta etapa tan significativa en mi vida.

II .-Resumen.

El presente proyecto tuvo como objetivo principal el desarrollo de un sistema web para la empresa Novatech, orientada a optimizar y digitalizar la gestión de recepción y seguimiento de equipos de cómputo. Este sistema se concibió con la intención de mejorar la experiencia tanto para los clientes como para el personal administrativo, facilitando procesos internos y ofreciendo una plataforma accesible para la consulta del estado de los equipos en tiempo real.

Para el desarrollo de este proyecto, se utilizaron diversas herramientas tecnológicas. El backend fue implementado con **PHP**, permitiendo una lógica robusta y adaptable para las funcionalidades requeridas. Además, se integraron APIs como **Google CAPTCHA** para garantizar la seguridad en las interacciones de los usuarios, previniendo accesos no autorizados y posibles ataques. En el diseño frontend, se emplearon tecnologías como **HTML, CSS, JavaScript** y **Bootstrap**, que permitieron desarrollar una interfaz moderna, intuitiva y completamente responsive, adaptada a diferentes dispositivos.

El proyecto se estructuró en dos módulos principales: un portal público destinado a los clientes, donde pueden conocer la empresa así como los servicios, ubicación y consultar su estado de algún dispositivo dejado en la empresa o conocer el estatus de su proyecto, y un panel de administración exclusivo para la empresa, que permite gestionar de manera eficiente las solicitudes, generar notas de recepción y realizar un seguimiento detallado de cada equipo ingresado.

Los resultados obtenidos cumplieron satisfactoriamente con los objetivos planteados. El sistema mejoró significativamente la comunicación entre la empresa y los clientes, optimizando los tiempos de respuesta y reduciendo la

dependencia de procesos manuales. Este desarrollo representa un paso importante hacia la modernización tecnológica de Novatech, sentando las bases para futuras mejoras y ampliaciones que fortalezcan la gestión empresarial y la experiencia del cliente.

III.-Índice

I .- Agradecimientos.....	2
II .-Resumen.....	3
III.-Índice.....	5
IV. Generalidades del proyecto.....	7
1.-Introducción.....	7
2 . Descripción de la empresa y área del trabajo.....	9
3.- Planteamiento del problema.....	11
4.- Objetivo General.....	13
5.- Objetivos específicos.....	13
6.-Justificación.....	13
7.- Delimitación.....	14
8.- Limitación.....	15
V. Marco Teórico.....	17
1.- Gestión de Bases de Datos.....	17
2.- Seguridad en Aplicaciones Web.....	17
3.- Modelo Vista Controlador (MVC).....	18
4.- Frameworks y Tecnologías Web.....	18
5.- Control de Versiones con GitHub.....	19
VI.- Desarrollo.....	20
1.- Uso de GitHub para el Control de Versiones.....	22
2.- Uso de Bootstrap, HTML, CSS y JavaScript para la Página Web.....	26
3.- HTML: La estructura base.....	29
4.- Estructura Modular con Includes en PHP.....	30
5.- CSS: Personalización y Estilo.....	32
6.- JavaScript: Dinamismo en la página.....	34
7.- Uso de PHP y el Modelo Vista-Controlador (MVC).....	35
8.- Creación de base de datos.....	37
9.- API de Captcha google.....	42
10.- Pruebas.....	43
VII.- Resultados.....	46
VIII.- Conclusiones.....	47
IX.- Competencias desarrolladas.....	48
X.- Fuentes de información.....	49
X I .- Anexos.....	50

IV. Generalidades del proyecto.

1.-Introducción.

La empresa NovaTech es una empresa mexicana que se destaca en el ámbito de innovar en la tecnología que está implicada en la industria, especialmente en el contexto de la industria 4.0. Ya que fue fundada desde hace 15 años, su misión es transformar y optimizar los procesos industriales mediante soluciones tecnológicas que sean avanzadas y personalizadas.

En un mundo cada vez más digitalizado, la modernización de los procesos internos de las empresas se ha vuelto una necesidad fundamental para mantenerse competitivas y ofrecer servicios de calidad. Este proyecto surge en respuesta a las necesidades específicas de **Novatech**, una empresa dedicada a la ingeniería computacional y el soporte técnico, que busca mejorar su sistema de gestión para la recepción y seguimiento de equipos de cómputo.



Ilustración 1; Logo novatech; Fuentes: Google.

Anteriormente, Novatech manejaba este proceso de forma manual, lo que implicaba mayores tiempos de espera, dificultades en la organización y un seguimiento limitado del estado de los equipos. Esto generaba inconvenientes tanto para los clientes como para el personal encargado de estas tareas. En este contexto, el desarrollo de una sistema web funcional y accesible se presentó como una solución integral para optimizar estas operaciones.

La solución propuesta tiene como objetivo central digitalizar y automatizar la recepción de equipos, así como facilitar a los clientes la consulta en tiempo real del estado de sus equipos a través de un portal en línea.

Asimismo, se desarrolló un panel administrativo que permite al personal de Novatech gestionar las solicitudes, emitir notas de recepción y llevar un control detallado de cada equipo registrado en el sistema.

Este proyecto no solo representa una mejora en términos de eficiencia operativa para Novatech, sino que también fortalece la experiencia del cliente al ofrecerle mayor transparencia y comodidad en el acceso a los servicios. Al integrar herramientas modernas como **PHP**, **Bootstrap** y APIs como **Google CAPTCHA**, se garantiza una plataforma segura, intuitiva y con la capacidad de adaptarse a las demandas actuales del mercado.

En las siguientes secciones se detalla el desarrollo de esta solución, desde su concepción hasta la implementación final, destacando los retos superados y los beneficios obtenidos para la empresa y sus clientes.

2 . Descripción de la empresa y área del trabajo.

Novatech es una empresa ubicada en Apizaco, Tlaxcala, que se especializa en ofrecer servicios de ingeniería computacional. Su cartera de servicios incluye soporte y mantenimiento de equipos de cómputo, desarrollo de software, instalación de sistemas de videovigilancia y accesos biométricos, venta de consumibles y soluciones tecnológicas personalizadas. La empresa se destaca por su compromiso con la calidad y la innovación, buscando constantemente mejorar sus procesos internos y la experiencia de sus clientes.

Dentro de este contexto, el área en la que el estudiante desarrolló su proyecto estuvo directamente relacionada con la **digitalización de procesos internos**. Específicamente, el objetivo fue optimizar el sistema de recepción de equipos de cómputo a través de un sistema web funcional, que no solo facilitará la gestión interna, sino que también ofreciera a los clientes una manera eficiente de consultar el estado de sus equipos.

Durante el desarrollo del proyecto, el estudiante asumió el rol de **desarrollador web**, colaborando estrechamente con los responsables del área operativa de Novatech. Las responsabilidades incluyeron:

- **Análisis de necesidades:** Identificar los principales retos del sistema manual y las expectativas tanto del personal interno como de los clientes.
- **Diseño e implementación de la plataforma:** crear una página web utilizando herramientas como PHP, Bootstrap y APIs, como Google CAPTCHA, para garantizar seguridad y funcionalidad.

- **Integración de un panel administrativo:** diseñar un módulo interno para la recepción, registro y seguimiento de equipos, facilitando el trabajo del personal.
- **Pruebas y ajustes:** realizar pruebas en tiempo real, recibiendo retroalimentación del equipo de Novatech y ajustando el sistema según sus necesidades específicas.

La contribución no solo se limitó al desarrollo tecnológico, sino que también ayudó a mejorar la organización de los flujos de trabajo en la empresa. Este proyecto representa un paso significativo hacia la transformación digital de Novatech, fortaleciendo su competitividad en el mercado y consolidando su relación con los clientes mediante un servicio más ágil y transparente.

3.- Planteamiento del problema.

Durante el diagnóstico inicial del proyecto, se identificaron diversos problemas en los procesos internos de Novatech relacionados con la gestión de recepción y seguimiento de equipos de cómputo. Estos problemas fueron priorizados en función de su impacto en la operación diaria y su urgencia para ser resueltos.

1. Falta de un sistema centralizado para gestionar la recepción de equipos

Impacto: alto.

Descripción: el proceso de recepción de equipos se realizaba de manera manual, utilizando registros en papel o herramientas básicas como hojas de cálculo. Esto generaba inconsistencias en el manejo de datos, riesgo de extravío de información y dificultades para realizar un seguimiento eficiente.

2. Ausencia de transparencia en la comunicación con los clientes

Impacto: alto.

Descripción: los clientes no contaban con un medio práctico para conocer el estado de sus equipos durante el proceso de reparación o mantenimiento. Esto resultaba en múltiples consultas vía telefónica o presencial, lo que incrementa la carga administrativa.

3. Dificultad para organizar y consultar el historial de servicios

Impacto: medio.

Descripción: la falta de un registro digitalizado dificulta la revisión del historial de equipos atendidos, afectando la capacidad de la empresa para generar reportes de servicios prestados o identificar tendencias en las solicitudes.

4. Riesgo de errores humanos en la asignación de tareas y seguimiento

Impacto: medio.

Descripción: al depender de procesos manuales, existía un mayor riesgo de errores en la asignación de tareas al personal técnico o en el registro de detalles importantes, como números de serie, problemas reportados y costos estimados.

5. Baja eficiencia en la generación de reportes y notas de recibido

Impacto: bajo.

Descripción: la creación de notas de recibido y reportes operativos se realizaba manualmente, lo que consumía tiempo y recursos que podrían ser destinados a otras actividades clave.

Priorización de problemas

- Implementar un sistema centralizado para la gestión de recepción de equipos.
- Mejorar la comunicación y transparencia con los clientes mediante una herramienta de consulta en línea.
- Digitalizar y organizar el historial de servicios para facilitar consultas y análisis.
- Automatizar tareas repetitivas para reducir errores humanos.
- Optimizar la creación de notas de recibido y reportes mediante el uso de plantillas digitales.

La resolución de estos problemas fue clave para mejorar la eficiencia operativa de Novatech, optimizar recursos y ofrecer una experiencia más satisfactoria a los clientes.

4.- Objetivo General.

Desarrollar una plataforma web integral para NovaTech, que permita la gestión eficiente de la recepción, seguimiento y actualización del estado de equipos de cómputo de los clientes, optimizando los procesos internos y mejorando la experiencia del cliente.

5.- Objetivos específicos.

- Diseñar y desarrollar un sistema web de presentación que refleje la identidad y los servicios de NovaTech, asegurando una interfaz amigable y profesional.
- Implementar un sistema de autenticación para el acceso a un panel de administración destinado al personal de NovaTech.
- Desarrollar un módulo de recepción de equipos que permita registrar los detalles de los equipos recibidos y generar notas de recibo automáticas.
- Crear un sistema de gestión de clientes que almacene información relevante de los clientes y sus equipos, permitiendo el seguimiento de métricas de servicio.
- Desarrollar un apartado para la consulta del estado del equipo por parte del cliente, accesible desde la página web, que ofrezca actualizaciones en tiempo real.
- Asegurar la seguridad y confidencialidad de la información mediante la implementación de buenas prácticas de seguridad web.

6.-Justificación.

El desarrollo de este proyecto surge como una respuesta estratégica para mejorar los procesos internos de Novatech, especialmente en la gestión de recepción y seguimiento de equipos de cómputo. En un mercado competitivo, la eficiencia operativa y la satisfacción del cliente son factores determinantes para el éxito de cualquier empresa.

El diseño e implementación de un sistema automatizado permitirá a Novatech modernizar sus operaciones, reduciendo tiempos de respuesta y minimizando errores humanos. Esto no solo optimiza los recursos internos, sino que también fortalece la confianza de los clientes al ofrecerles transparencia y acceso a información actualizada sobre el estado de sus equipos.

Adicionalmente, la solución planteada tiene un impacto significativo en la capacidad de la empresa para manejar un mayor volumen de solicitudes, posicionándose como un negocio competitivo y preparado para el crecimiento. El uso de herramientas tecnológicas como PHP y APIs específicas, como la de CAPTCHA para seguridad, no solo asegura la funcionalidad del sistema, sino también su escalabilidad y protección frente a accesos no autorizados.

Este proyecto no solo aborda las necesidades inmediatas de la empresa, sino que sienta las bases para futuros desarrollos tecnológicos, consolidando a Novatech como un referente en innovación y servicio al cliente en el ámbito de la ingeniería computacional. Su implementación, por lo tanto, es esencial para garantizar la sostenibilidad y el éxito de la empresa a largo plazo.

7.- Delimitación.

El proyecto se centra exclusivamente en el diseño e implementación de un sistema web para la gestión de la recepción de equipos de cómputo en Novatech. Este sistema incluye una página pública para los clientes, donde podrán iniciar sesión y consultar el estado de sus equipos, y un panel de administración privado para los empleados, que permite registrar y monitorear los equipos recibidos.

Las herramientas empleadas para el desarrollo son PHP para el backend, APIs específicas como CAPTCHA de Google para la seguridad, y tecnologías de frontend como HTML, CSS, JavaScript y Bootstrap.

El alcance del proyecto está delimitado a las funciones específicas de recepción y seguimiento de equipos, excluyendo otras áreas como inventarios, facturación o manejo de historial de clientes, o bien el consumo de APIs de IA que puedan dar un diagnóstico previo del equipo o bien alguna serie de pasos para su solución, que podrán desarrollarse en fases futuras.

8.- Limitación.

1. **Tiempo de Desarrollo:** el proyecto tiene un marco temporal limitado de 5 meses, lo cual restringe la posibilidad de incluir características adicionales o iterar múltiples versiones del sistema.
2. **Recursos Humanos:** el equipo de desarrollo está compuesto por un único desarrollador, lo cual limita la velocidad de implementación y pruebas.
3. **Infraestructura Técnica:** al depender del entorno actual de la empresa, las capacidades del servidor y la infraestructura tecnológica podrían influir en el desempeño del sistema y limitar la implementación de funcionalidades más avanzadas.
4. **Restricciones Presupuestarias:** el proyecto debe desarrollarse bajo un presupuesto establecido, lo que restringe el uso de herramientas o tecnologías más costosas que podrían mejorar el rendimiento o la experiencia de usuario.

5. **Alcance de la Seguridad:** aunque se implementará CAPTCHA para evitar accesos no autorizados, el sistema no incluye características avanzadas de ciberseguridad como la encriptación de extremo a extremo o monitoreo en tiempo real, que podrían considerarse en fases futuras.

Estas limitaciones fueron consideradas durante la planeación y ejecución del proyecto, priorizando las funcionalidades esenciales para garantizar su entrega dentro del tiempo establecido y con los recursos disponibles.

V. Marco Teórico.

El desarrollo del proyecto se fundamentó en diversos conceptos y tecnologías que fueron determinantes para su diseño e implementación. Estos elementos no solo brindaron las bases técnicas necesarias, sino que también aportaron metodologías modernas para garantizar un sistema funcional, seguro y escalable.

1.- Gestión de Bases de Datos

La gestión de datos es una parte esencial de cualquier sistema que maneje información sensible, como el caso del presente proyecto. A través de una base de datos relacional, se organizó y estructuró la información de clientes y equipos, estableciendo relaciones claras y eficientes entre las entidades. Esto permitió no solo un almacenamiento adecuado de los datos, sino también una recuperación y manipulación rápida y segura de la información cuando fuese necesario. Se prestó especial atención a la integridad referencial y a la optimización de consultas, asegurando un sistema robusto y capaz de manejar grandes volúmenes de datos.

2.- Seguridad en Aplicaciones Web

La seguridad en el ámbito de aplicaciones web es un factor crítico, especialmente en sistemas que manejan información confidencial. Este proyecto implementó diversas medidas para proteger los datos y garantizar que solo los usuarios autorizados accedieron a la información. Entre estas medidas destacan:

- **Cifrado de contraseñas:** Se utilizó un algoritmo de cifrado robusto, asegurando que las contraseñas almacenadas en la base de datos no fueran legibles en caso de un ataque.
- **Controles de acceso:** Se implementó un sistema de autenticación basado en roles, permitiendo un acceso diferenciado según los privilegios de cada usuario.

- **Integración de Google CAPTCHA:** Esta API fue incorporada para prevenir ataques automatizados, como el relleno de formularios o intentos de inicio de sesión masivos, fortaleciendo aún más la seguridad del sistema.

3.- Modelo Vista Controlador (MVC)

El modelo MVC fue adoptado como la arquitectura principal para el desarrollo del sistema utilizando PHP. Este patrón separa claramente los tres componentes esenciales:

- **Modelo:** Se encarga de la lógica de negocio y de interactuar con la base de datos.
- **Vista:** Administra la presentación de datos al usuario final, asegurando una experiencia clara y agradable.
- **Controlador:** Actúa como intermediario entre el modelo y la vista, gestionando las solicitudes del usuario y los flujos de datos.

Esta separación no solo mejora la organización del código, sino que facilita su mantenimiento y escalabilidad al permitir realizar cambios en una parte del sistema sin afectar las demás.

4.- Frameworks y Tecnologías Web

En el desarrollo de la interfaz de usuario se utilizaron tecnologías como HTML, CSS y JavaScript, que permitieron la creación de una plataforma visualmente atractiva y funcional. Estas herramientas se complementaron con **Bootstrap**, un framework de diseño que optimiza el tiempo de desarrollo y garantiza la responsividad del sistema en dispositivos móviles y de escritorio.

Además, se utilizó JavaScript para dotar de interactividad a la plataforma, haciendo que ciertas funcionalidades fueran más dinámicas y amigables para los usuarios.

5.- Control de Versiones con GitHub

Durante el desarrollo del proyecto, Git se utilizó como herramienta principal de control de versiones. Su implementación permitió:

- Llevar un registro detallado de los cambios realizados en el código.
- Proteger el trabajo mediante copias de seguridad distribuidas, asegurando que el progreso del proyecto no se perdiera ante fallas técnicas.

Git también posibilitó la documentación de cada etapa del desarrollo, manteniendo un historial claro y organizado del avance del proyecto.

El proyecto integra conceptos clave en la gestión de datos, seguridad, diseño de interfaces y metodologías de desarrollo, con herramientas modernas que aseguran tanto la funcionalidad técnica como la experiencia del usuario. La combinación de estas tecnologías y enfoques metodológicos fue esencial para construir un sistema que no solo satisface las necesidades actuales de la empresa, sino que también está preparado para futuras ampliaciones y mejoras.

VI.- Desarrollo.

El desarrollo del sitio web Tech Manager se fundamenta en un diseño estructurado y estratégico que garantice la funcionalidad, accesibilidad y eficiencia del sistema. Para lograr este objetivo, se utiliza un diagrama de construcción que detalla las conexiones y relaciones entre los diversos componentes del sitio, desde las páginas principales hasta los subsistemas internos.

El mapa de conexión anexo presenta la arquitectura general del sitio web. Este incluye:

Sección de usuario: Diseño orientado a facilitar la experiencia de navegación para clientes, incluyendo accesos a módulos como la recepción de equipos y consulta de estatus.

Panel de administración: Herramienta principal para la gestión interna de la empresa, diseñada para manejar las operaciones administrativas y técnicas relacionadas con el negocio.

Integración de APIs externas: Herramientas clave como la API de CAPTCHA de Google, que garantizan la seguridad del sitio y previenen el acceso no autorizado.

El diagrama es esencial para identificar los flujos de datos, las dependencias y las tecnologías clave que interactúan en la construcción del proyecto. Sirve como guía para el equipo de desarrollo y como referencia para documentar la progresión del proyecto, asegurando que todos los elementos funcionen de manera óptima y estén alineados con los objetivos del cliente.



1.- Uso de GitHub para el Control de Versiones

Objetivo:

El principal objetivo de utilizar Git en este proyecto fue facilitar la realización de pruebas, prevenir errores y garantizar un respaldo constante del código. Esto permitió trabajar de manera organizada, mantener la integridad del proyecto y asegurar que cada cambio realizado fuera documentado y controlado.

Herramientas utilizadas:

- **GitHub:** servicios de alojamiento de repositorios Git remotos, utilizados para compartir el código y colaborar de manera eficiente.

Flujo de trabajo con Git:

1. **Creación del repositorio:** desde el inicio del proyecto, se creó un repositorio local y remoto (en GitHub) para centralizar el código y trabajar de forma ordenada.
2. **Estructura de ramas (branches):** se implementó una estrategia de ramas para organizar el flujo de trabajo:
 - **Rama principal (main o master):** contiene siempre el código que está listo para producción.
 - **Rama de desarrollo (Juan):** rama donde se integran los cambios de características y correcciones antes de ser fusionados a la rama principal.

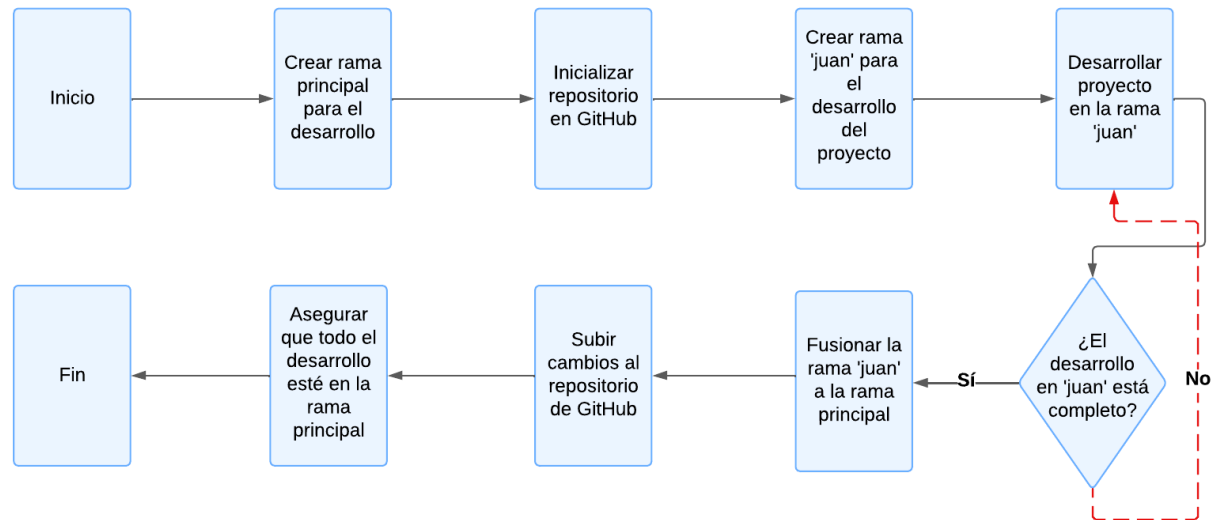


Ilustración 2; Flujo de trabajo de Github con el proyecto; Fuentes: Propias.

Para la preparación del desarrollo se utilizó el diagrama de flujo anexo el cual muestra los pasos que se siguieron para el control de versiones.

3. **Commits:** Se realizaron commits periódicos para registrar los cambios en el código. Cada commit se documentó con un mensaje claro y conciso que explica los cambios realizados.

Commits

main
All users
All time

Commits on Dec 16, 2024

Actualización del proyecto: Cambios en estilos, eliminación de index.html y adición de nuevos componentes.

Juanhnc74 committed 3 weeks ago

4e9f0f1

Commits on Nov 13, 2024

Merge branch 'juan' of <https://github.com/Juanhnc74/WEBTECH> into juan

Juanhnc74 committed on Nov 13, 2024

4da2194

cambios contenido index

Juanhnc74 committed on Nov 13, 2024

a0889c9

modificacion de index

Juanhnc74 committed on Nov 13, 2024

cc7fe26

Ilustración 3; Commits de Github con el proyecto; Fuentes: Propias.

4. **Pull requests (PRs):** Antes de integrar una nueva funcionalidad o corrección a la rama principal, se crearon pull requests, lo cual permitió revisar el código, realizar pruebas y asegurar la calidad del mismo.
5. **Resolución de conflictos:** En caso de que se presentaran conflictos al fusionar ramas, estos fueron resueltos de manera colaborativa, asegurando que no se perdieran cambios importantes y manteniendo la integridad del código.
6. **Implementación continua:** En algunos casos, se utilizó integración continua (CI) mediante GitHub Actions, Travis CI o Jenkins, lo cual permitía ejecutar pruebas automáticas sobre el código cada vez que se realizan cambios importantes.

Beneficios destacados:

Realización de cambios y commits: Después de hacer cambios en el código , se guardaron y registraron mediante commits

- **Seguimiento de cambios:** Git permite mantener un historial completo de todos los cambios realizados, lo que facilita la identificación de errores y la restauración de versiones anteriores del código.
- **Pruebas controladas:** Las ramas permitieron implementar y verificar cambios de forma aislada antes de fusionarlos con el código estable.
- **Prevención de errores:** El control de versiones ayudó a evitar sobreescrituras accidentales o pérdida de datos, especialmente durante pruebas y ajustes.
- **Respaldo constante:** La sincronización con GitHub aseguró que el código estuviera siempre respaldado y accesible en caso de fallos en el entorno local.
- **Documentación del desarrollo:** Los commits detallados y el historial de cambios sirvieron como registro completo del progreso del proyecto.

En este proyecto individual, Git se utilizó como una herramienta esencial para organizar, probar y respaldar el desarrollo, asegurando estabilidad y calidad en cada etapa del proceso.

2.- Uso de Bootstrap, HTML, CSS y JavaScript para la Página Web

El diseño de la página web fue desarrollado utilizando una combinación de tecnologías front-end modernas, como **Bootstrap**, **HTML**, **CSS** y **JavaScript**. Estas herramientas permitieron crear una interfaz de usuario atractiva, funcional y accesible tanto para clientes como para administradores. A continuación, se detallan los aspectos principales de su implementación:

Bootstrap: La base del diseño responsivo

1. Versión Utilizada:

- Para este proyecto, se utilizó **Bootstrap 5**, que ofrece nuevas funcionalidades como mejor soporte para Flexbox y Grid, además de componentes mejorados.
- Esta elección permitió construir rápidamente un diseño **responsive** y adaptativo que se ajusta automáticamente a cualquier dispositivo (escritorio, tablet o móvil).

2. Componentes Implementados:

- **Navbar**: Se usó la barra de navegación de Bootstrap para facilitar el acceso a las secciones principales de la página, como el inicio de sesión y el seguimiento de equipos.

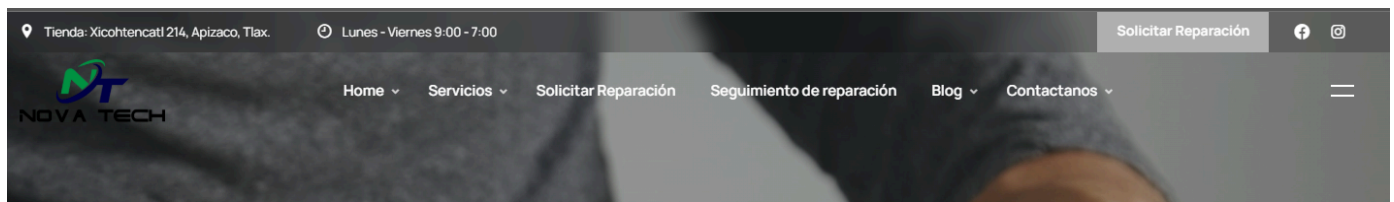


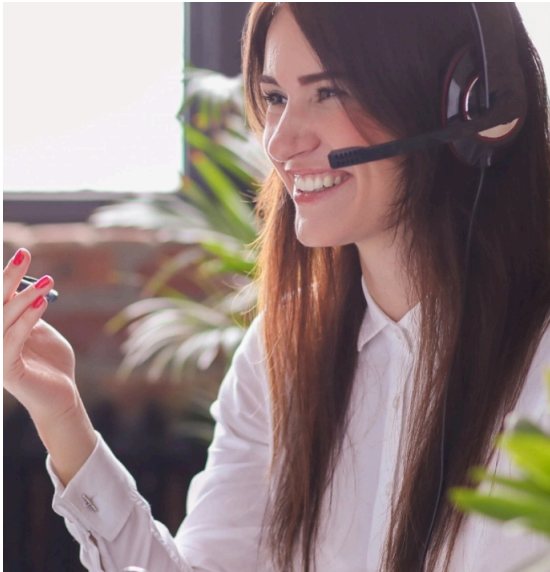
Ilustración 4; Barra de navegación principal; Fuentes: Página Web novatech.

- **Cards:** Se emplearon tarjetas para mostrar información de manera organizada, como servicios disponibles o detalles del estado de los equipos.



Ilustración 5; Tarjetas de información; Fuentes: Página Web novatech.

- **Formularios:** Bootstrap fue clave en la creación de formularios estilizados con validaciones visuales que hacen uso de sus clases integradas.



CONTACT US

Get In Touch

El host local no está en la lista de dominios admitidos para esta clave del sitio web.



Ilustración 6; Formulario de contacto; Fuentes: Página Web novatech.

- **Botones y Modalidades:** Los botones de interacción y ventanas emergentes se diseñaron usando componentes predefinidos de Bootstrap.

NUESTRO EXPERTOS

Confía en nuestros.

Nuestro equipo de técnicos expertos ofrece reparaciones de alta calidad para tus dispositivos, con amplia experiencia y certificaciones avanzadas para resolver desde problemas sencillos hasta complejos.

Ilustración 7; Modal de texto con botón; Fuentes: Página Web novatech.

3. Buenas Prácticas:

- Se organizó el código utilizando un sistema de **clases personalizadas**, evitando la sobrescritura directa de las clases predeterminadas de Bootstrap.
- Se siguió una jerarquía en las secciones para mantener la coherencia y facilidad de mantenimiento del código.

4. Notas sobre el Proyecto:

- **Archivo Minificado:** Se utilizó la versión minificada de Bootstrap para reducir el peso de la página y mejorar los tiempos de carga.

3.- HTML: La estructura base

Estructura Semántica:

- Se utilizó una estructura semántica en el HTML para garantizar la accesibilidad y cumplir con estándares web. Secciones como `<header>`, `<main>` y `<footer>` facilitaron la comprensión del diseño.

Separación por Módulos:

- Para mantener un diseño escalable, se dividió el proyecto en múltiples páginas HTML con propósitos específicos:
 - i. `index.php`: Página principal para visitantes y clientes.
 - ii. `login.php`: Página de inicio de sesión.
 - iii. `dashboard.php`: Panel administrativo para la gestión de equipos.
 - iv. `status.php`: Página de consulta del estado de los equipos para clientes.

4.- Estructura Modular con Includes en PHP

Carpeta **includes**:

- Para garantizar flexibilidad, mantenimiento y la **reutilización de componentes**, se creó una carpeta llamada **includes** que contiene los siguientes archivos:
 - i. **header.php**: Contiene el encabezado y la barra de navegación que son comunes en todas las páginas.
 - ii. **footer.php**: Incluye el pie de página con información de contacto y enlaces relevantes.
- Estas partes fueron **heredadas dinámicamente** en cada página utilizando la función **include()** de PHP, lo que evitó la duplicación de código y permite realizar cambios globales de forma rápida.

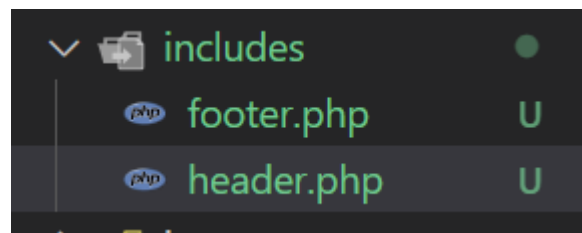


Ilustración 8;Estructuración de archivos; Fuentes:VS code proyecto.

Beneficios del Enfoque Modular:

- **Mantenimiento Eficiente**: Cualquier actualización en el encabezado o pie de página sólo requiere modificar los archivos en la carpeta **includes**.

- **Consistencia:** Se garantiza una apariencia uniforme en todas las páginas del sitio

```
web > index.php
1 <?php include 'includes/header.php'; ?>
2 <!-- Resto del contenido de la página -->
3
4
5
6 <!-- content begin -->
7 <div class="no-bottom no-top" id="content">
8
9     <div id="top"></div>
10
11     <section class="section-dark text-light no-top no-bottom position-relative z-1000">
12         <div class="v-center">
13             <div class="swiper">
14                 <!-- Additional required classes -->
```

Ilustración 9;Estructuración de código de header heredado; Fuentes:VS code proyecto.

Modularización del Código:

- Cada sección de la página fue dividida en archivos individuales, como `index.html` para la página principal, `login.html` para el inicio de sesión, y `status.html` para el seguimiento de equipos además de todas las páginas que se utilizan desde la carpeta de vistas.
- Esto no solo facilitó el mantenimiento, sino que también permitió trabajar de forma paralela en distintas secciones.

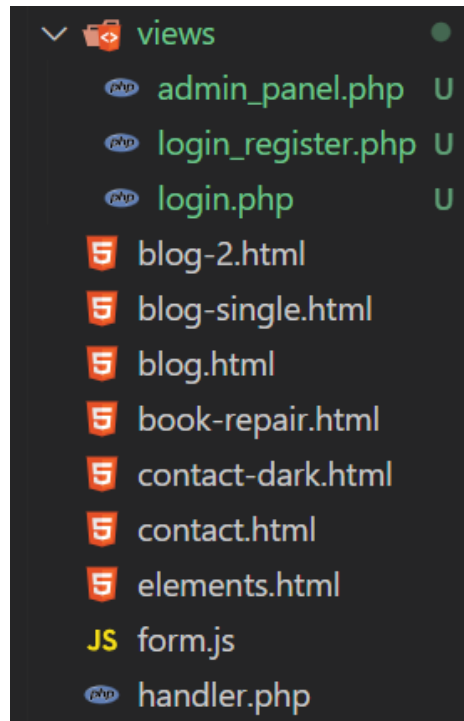


Ilustración 10;Estructuración de Módulos; Fuentes:VS code proyecto.

5.- CSS: Personalización y Estilo

1. Uso de un Archivo Externo:

- Se creó un archivo CSS personalizado (**style.css**) para agregar estilos adicionales y sobrescribir ciertos aspectos de Bootstrap, logrando una identidad visual única para el negocio.

2. Estilos Definidos:

- Colores: Se definió una paleta de colores corporativos basada en los tonos de la empresa.

```
web > css > colors > css scheme-01.css >  :root
1  :root {
2      --primary-color:  #B0B0B0;
3      --primary-color-rgb: 77, 108, 119;
4      --secondary-color:  #C6E76C;
5      --secondary-color-rgb: 198, 231, 108;
6      --tertiary-color:  #95c4d4;
7      --tertiary-color-rgb: 149, 196, 212;
8  }
```

Ilustración 11;Colores Principales usados; Fuentes:VS code proyecto.

- Tipografía: se integraron fuentes personalizadas utilizando **Google Fonts** para mantener una imagen profesional y moderna.
- Espaciado: se optimizaron márgenes y paddings para mejorar la legibilidad y usabilidad.

3. Buenas Prácticas:

- Los estilos personalizados se escribieron siguiendo la convención **BEM (Block, Element, Modifier)** para mantener la consistencia y escalabilidad.
- **Compresión:** el archivo CSS fue modificado en la etapa final para optimizar el rendimiento.

6.- JavaScript: Dinamismo en la página

1. Funcionalidades Implementadas:

- Validación de Formularios:
 - Se usaron scripts personalizados para verificar campos obligatorios antes de enviar datos, en complemento con las validaciones nativas de HTML5.
- Interacciones:
 - Botones que muestran ventanas modales, interacciones dinámicas en el seguimiento de equipos, y notificaciones de éxito o error.
- Animaciones:
 - Se implementaron transiciones suaves en los elementos para mejorar la experiencia de usuario.

2. Uso de Librerías:

- **SweetAlert2:** para notificaciones visuales de errores o confirmaciones, como al enviar un formulario.
- **Fetch API:** para realizar peticiones HTTP al servidor, especialmente para consultar el estado de los equipos en tiempo real.

3. Buenas Prácticas:

- Todo el código JavaScript fue modularizado en un archivo separado (**scripts.js**), manteniendo el HTML limpio.
- Se evitó el uso de bibliotecas innecesarias para garantizar un rendimiento óptimo.

7.- Uso de PHP y el Modelo Vista-Controlador (MVC).

El desarrollo de la aplicación web para Novatech se basó en el modelo arquitectónico Vista-Controlador (MVC), lo que permitió una separación clara entre la lógica de la aplicación, la interfaz de usuario y el manejo de datos.

Componentes del Modelo Vista-Controlador:

1. **Modelo:** encargado de interactuar con la base de datos. En este caso, los modelos definieron las operaciones de CRUD (Crear, Leer, Actualizar, Eliminar) para los datos de los equipos, usuarios y estados.
2. **Vista:** encargada de mostrar la información al usuario. Todas las vistas se desarrollaron utilizando una combinación de HTML, CSS y JavaScript, con el apoyo de Bootstrap para un diseño responsivo.
3. **Controlador:** gestiona la comunicación entre el modelo y la vista. Los controladores procesaron las solicitudes de los usuarios, llamaron a los modelos para obtener datos, y finalmente enviaron la información a las vistas correspondientes.

Este enfoque modular facilitó el mantenimiento y la escalabilidad del sistema, ya que cada componente podría modificarse independientemente.

Uso de Includes para Header y Footer

Con el objetivo de maximizar la reutilización de código y garantizar la consistencia en todas las páginas, se creó una carpeta llamada **includes**, donde se alojaron los archivos **header.php** y **footer.php**.

- **header.php**: contiene elementos comunes como el menú de navegación, estilos globales y scripts esenciales. Este archivo fue incluido al inicio de cada página utilizando la función `include()`.

```
web > includes > header.php
1  <!DOCTYPE html>
2  <html lang="es">
3
4  <head>
5      <title>Nova Tech</title>
6      <link rel="icon" href="images/icon.png" type="image/gif" sizes="16x16">
7      <meta content="text/html; charset=utf-8" http-equiv="Content-Type">
8      <meta content="width=device-width, initial-scale=1.0" name="viewport" >
9      <meta content="Nova Tech" name="description" >
10     <meta content="" name="keywords" >
11     <meta content="" name="author" >
12     <!-- CSS Files
13     ===== -->
14     <link href="css/bootstrap.min.css" rel="stylesheet" type="text/css" id="bootstrap">
15     <link href="css/plugins.css" rel="stylesheet" type="text/css" >
16     <link href="css/swiper.css" rel="stylesheet" type="text/css" >
17     <link href="css/style.css" rel="stylesheet" type="text/css" >
18     <link href="css/coloring.css" rel="stylesheet" type="text/css" >
19     <!-- color scheme -->
20     <link id="colors" href="css/colors/scheme-01.css" rel="stylesheet" type="text/css" >
21
22 </head>
```

Ilustración 12; Contenido de header; Fuentes: VS code proyecto.

- **footer.php**: contiene elementos como enlaces finales de scripts, pie de página y créditos. Este archivo fue incluido al final de cada página. Esto no solo redujo el código duplicado, sino que también facilitó futuras actualizaciones, ya que cualquier cambio en el **header.php** o **footer.php** se reflejan automáticamente en todas las páginas del sitio.

```
web > includes > footer.php
1  <footer class="section-dark">
2      <div class="container">
3          <div class="row gx-5">
4              <div class="col-lg-4 col-sm-6">
5                  
6                  <div class="spacer-20"></div>
7                  <p>Nos encanta mantener tus dispositivos en perfecto estado. Nuestro equipo de téc
8
9                  <div class="social-icons mb-sm-30">
10                     <a href="https://www.facebook.com/Nova.Tech.Computadoras"><i class="fa-brands fa-f
11                     <a href="https://www.instagram.com/nova_tech_apizaco/?fbclid=IwY2xjawFtt2VleHRuA2F
12                     </div>
13                 </div>
14                 <div class="col-lg-4 col-sm-12 order-lg-1 order-sm-2">
15                     <div class="row">
16                         <div class="col-lg-6 col-sm-6">
17                             <div class="widget">
18                                 <h5>Company</h5>
19                                 <ul>
20                                     <li><a href="index.html">Home</a></li>
21                                     <li><a href="services">Servicios</a></li>
22                                     <li><a href="track.html">Seguimiento de mi reparación</a></li>
23                                     <li><a href="book-repair.html">Solicitar Reparación</a></li>
24                                     <li><a href="contact.html">Contacto</a></li>
25                                 </ul>
26                             </div>
27                     </div>
28                 </div>
29             </div>
30         </div>
31     </div>
32 </div>
```

Ilustración 13; Contenido de Footer; Fuentes: VS code proyecto.

8.- Creación de base de datos.

El diseño de la base de datos para Novatech WebTech Manager se centra en gestionar información relacionada con clientes, equipos de cómputo, notas de recepción y el estado de los equipos. Este diseño asegura un manejo eficiente de los datos y facilita la escalabilidad del sistema.

Tablas Principales:

Usuarios

Atributos:

- id_usuario (PK): identificador único del usuario.
- Nombre: completo del usuario.
- correo: correo electrónico asociado al usuario.
- contraseña: contraseña cifrada para garantizar la seguridad.
- rol: determina si el usuario es administrador o cliente.

Funcionalidad:

- Gestionar credenciales de acceso.
- Diferenciar permisos según el rol.

Cientes

Atributos:

- id_cliente (PK): Identificador único del cliente.
- nombre: nombre completo del cliente.
- telefono: número de contacto del cliente.
- direccion: dirección física del cliente.
- id_usuario (FK): relación con la tabla de usuarios.

Funcionalidad:

- Almacenar información de los clientes registrados.
- Facilitar la identificación rápida de clientes.

Equipos

Atributos:

- id_equipo (PK): identificador único del equipo.
- descripcion: descripción general del equipo.
- marca: marca del equipo.
- modelo: modelo específico del equipo.
- numero_serie: número de serie del equipo.
- id_cliente (FK): relación con la tabla de clientes.

Funcionalidad:

- Registrar equipos asociados a un cliente.
- Proveer detalles necesarios para reparaciones o seguimientos.

Notas de Recepción

Atributos:

- id_nota (PK): identificador único de la nota.
- fecha_recepcion: fecha en que se recibió el equipo.
- detalle_problema: descripción del problema reportado.
- estado: estado actual del equipo (recibido, en reparación, entregado).
- id_equipo (FK): relación con la tabla de equipos.

Funcionalidad:

- Documentar la recepción de equipos para reparación.
- Controlar el estado del equipo durante el proceso de reparación.

Seguimiento

Atributos:

- id_seguimiento (PK): identificador único del seguimiento.
- id_nota (FK): relación con la tabla de notas de recepción.
- fecha_actualizacion: fecha de actualización del seguimiento.
- comentarios: detalles adicionales sobre el estado del equipo.
- estado_actual: estado actualizado del equipo.

Funcionalidad:

- Registrar actualizaciones sobre el estado del equipo.
- Informar al cliente sobre el progreso de la reparación.

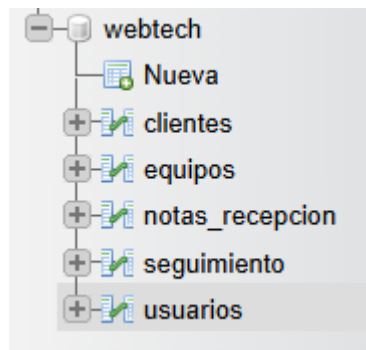


Ilustración 14; Tablas creadas; Fuentes: Panel de phpmyadmin.

Relaciones:

- Usuarios y Clientes: Relación 1:N (un usuario puede estar vinculado a varios clientes).
- Clientes y Equipos: Relación 1:N (un cliente puede tener varios equipos registrados).
- Equipos y Notas de Recepción: Relación 1:N (un equipo puede tener varias notas de recepción asociadas).
- Notas de Recepción y Seguimientos: Relación 1:N (una nota de recepción puede tener varias entradas de seguimiento).

Diseño UML

Se elaboró un diagrama de clases UML para representar de manera visual las entidades, sus atributos y métodos principales, así como las relaciones entre ellas.

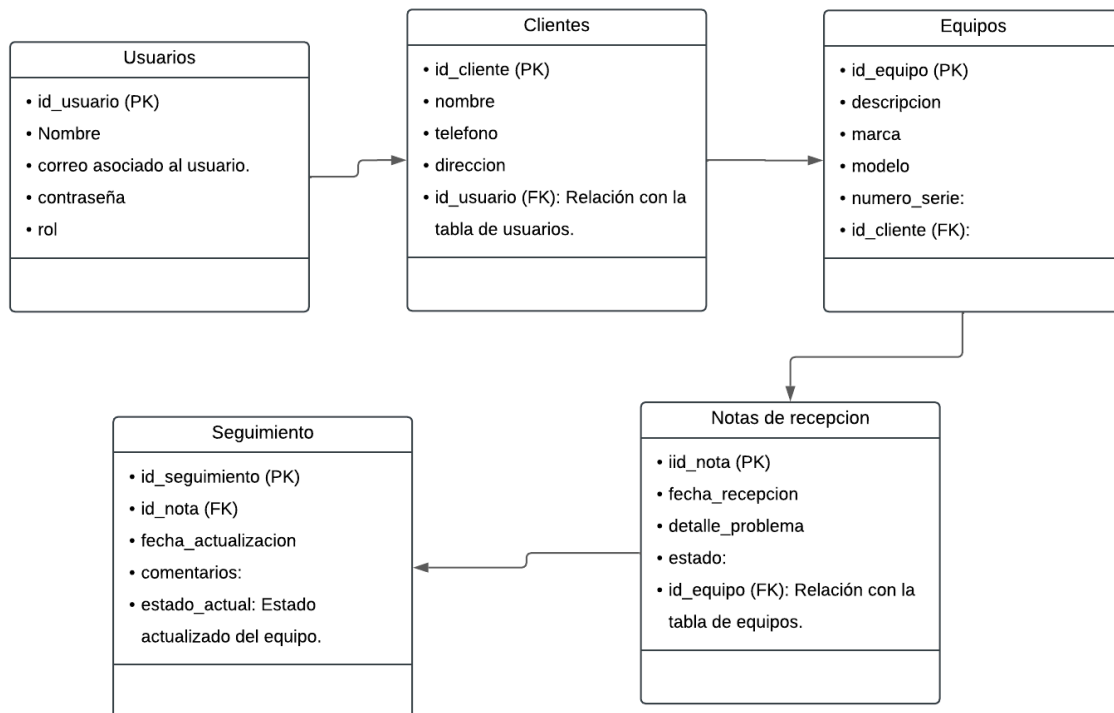


Ilustración 15;Diseño UML de las entidades; Fuentes:Propias.

9.- API de Captcha google.

Para proteger el sistema contra envíos de formularios automatizados, se integró la API de Google reCAPTCHA. Este componente asegura que sólo usuarios humanos puedan interactuar con los formularios.

Pasos Detallados:

1. Registro en Google reCAPTCHA:

- Se accedió al sitio de Google reCAPTCHA y se registró el dominio del proyecto.
- Se obtuvieron la clave de sitio y la clave secreta necesarias para la configuración.

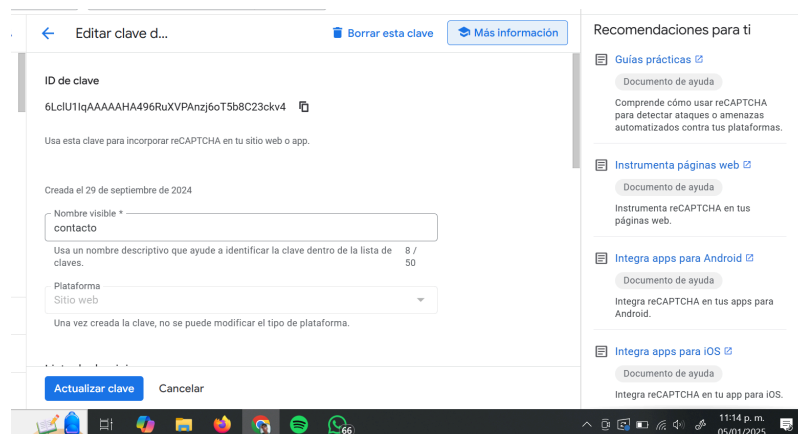


Ilustración 16; Claves de Captcha; Fuentes: Panel de administración captcha Google.

2. Configuración del Frontend:

En el formulario HTML, se incluyó el widget de reCAPTCHA mediante el siguiente código:

```
<script src="https://www.google.com/recaptcha/api.js" async defer></script>
```

- `<div class="g-recaptcha" data-sitekey="[CLAVE_DE_SITIO]"></div>`

```
$pp->requireReCaptcha();  
$pp->getReCaptcha()->initSecretKey('6LclU1IqAAAAAMY5Rdtk2H9c6DJMnbLqch91Ra0J');  
  
$pp->sendEmailTo('juanhnc74@gmail.com'); // ← email al que se enviaran los formularios  
  
echo $pp->process($_POST);
```

Ilustración 17; Inserción de clave Captcha; Fuentes: Entorno de VS code.

10.- Pruebas.

Para garantizar el correcto funcionamiento del sistema, se llevaron a cabo pruebas exhaustivas en diferentes entornos y escenarios:

1. Pruebas en Ambiente Local:

- Se utilizó XAMPP para simular el servidor local.
- Se probaron todas las funcionalidades del panel administrativo, como la generación y actualización de notas de recepción.

Iniciar sesión

Nombre de usuario:

Contraseña:

Ilustración 18; Inserción de claves para loguearse; Fuentes: Login página web.

2. Pruebas en Hosting Estático:

- La página de presentación se desplegó en Netlify, verificando:
 - Tiempos de carga.
 - Compatibilidad con navegadores (Chrome, Firefox, Edge).
 - Responsividad en dispositivos móviles.

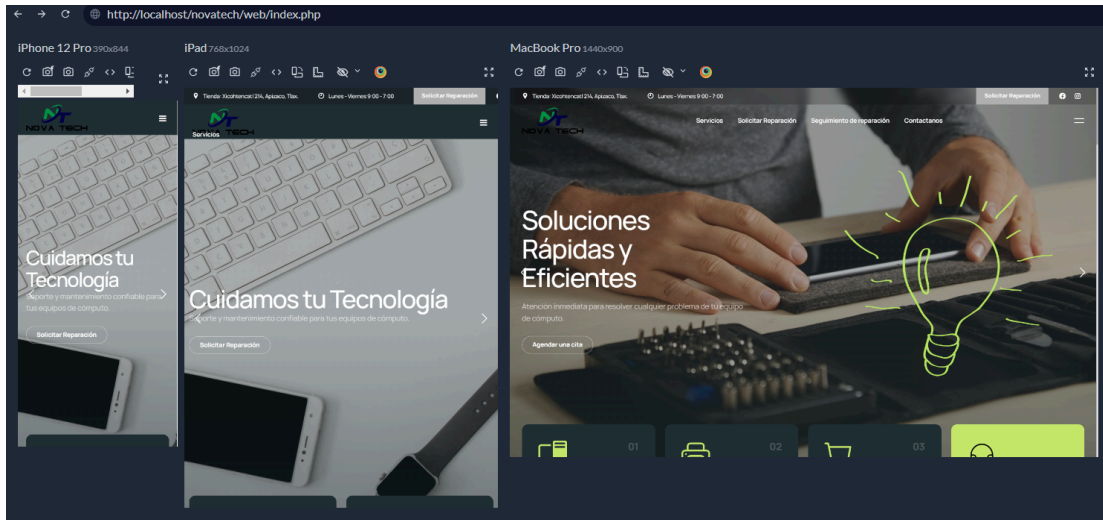


Ilustración 19; Vista desde distintos dispositivos; Fuentes: Interfaz de Responsively app.

3. Pruebas de Seguridad:

- Se realizaron intentos de inyección SQL en los formularios para comprobar la sanitización de datos.
- Se validó la protección contra bots mediante el CAPTCHA.

4. Pruebas de Usabilidad:

- Usuarios simulados interactuaron con el sistema para registrar equipos y consultar estados.
- Se recopiló retroalimentación para mejorar la experiencia de usuario.

5. Pruebas de Escalabilidad:

- Se insertaron datos masivos en la base de datos para evaluar su rendimiento bajo carga.
- Se analizaron los tiempos de respuesta de las consultas.

6. Pruebas de Integración:

- Se verificó la comunicación entre el frontend y backend, asegurando que los datos se transmitieran correctamente.
- Se simuló el flujo completo: desde el registro de un cliente hasta la entrega del equipo reparado.

7. Pruebas de Recuperación:

- Se evaluó la capacidad del sistema para manejar fallos inesperados, como interrupciones de conexión o errores en el servidor.
- Se implementaron mensajes de error amigables para informar al usuario

VII.- Resultados.

El proyecto Novatech WebTech Manager cumplió con los objetivos planteados al inicio de su desarrollo. Entre los principales resultados obtenidos destacan:

- Implementación Exitosa de Funcionalidades:
 - Registro y seguimiento eficiente de clientes y equipos.
 - Generación automática de notas de recepción y actualizaciones de estado.
- Mejora en los Procesos de Gestión:
 - Se optimizó la gestión interna de la empresa mediante un sistema digital que reemplaza procesos manuales.
 - Reducción significativa de errores en el registro de información.
- Interfaz Intuitiva y Accesible:
 - La página web y el panel administrativo presentan un diseño amigable y fácil de usar para los empleados y clientes.
 - Pruebas de usuario confirmaron la satisfacción con la usabilidad del sistema.
- Protección y Seguridad:
 - La integración del reCAPTCHA y la sanitización de datos protegieron el sistema contra ataques y accesos no autorizados.

VIII.- Conclusiones.

El desarrollo de **Novatech WebTech Manager** demostró ser una solución efectiva para atender las necesidades de gestión y seguimiento de equipos dentro de la empresa.

- **Cumplimiento de Objetivos:** Todos los requisitos del proyecto fueron cumplidos dentro del plazo establecido.
- **Impacto Positivo:** El sistema redujo los tiempos de atención a clientes y mejoró la organización interna de datos.
- **Escalabilidad:** El diseño modular de la base de datos permite agregar nuevas funcionalidades en el futuro, como reportes detallados o notificaciones automáticas para clientes.

IX.- Competencias desarrolladas.

Durante el desarrollo del proyecto, se fortalecieron diversas competencias técnicas y profesionales, entre las que destacan:

1. Habilidades Técnicas:

- Diseño y gestión de bases de datos relacionales.
- Implementación de APIs externas como Google reCAPTCHA.
- Desarrollo de aplicaciones web utilizando tecnologías modernas.

2. Resolución de Problemas:

- Análisis y solución de desafíos técnicos durante las pruebas y la integración de componentes.
- Identificación de mejoras en la experiencia del usuario.

3. Trabajo en Equipo:

- Colaboración efectiva con miembros de la empresa para recopilar requisitos y validar funcionalidades.
- Comunicación constante para asegurar el alineamiento con los objetivos del proyecto.

4. Gestión del Tiempo:

- Cumplimiento de hitos y entregables dentro de los plazos establecidos.
- Priorización de tareas críticas para garantizar el éxito del proyecto.

El proyecto no sólo representó un avance para la empresa, sino también una experiencia valiosa para el equipo de desarrollo al enfrentar y superar retos reales en el ámbito profesional.

X.- Fuentes de información.

1. Google Developers. (n.d.). *reCAPTCHA*. Recuperado de <https://www.google.com/recaptcha>
 - Fuente oficial para la integración de la API de CAPTCHA.
2. Connolly, T., & Begg, C. (2019). *Database systems: A practical approach to design, implementation, and management* (6th ed.). Pearson.
 - Recurso utilizado para estructurar y diseñar la base de datos.
3. Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill.
 - Proporciona conceptos fundamentales sobre ingeniería de software.
4. Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson Education.
 - Guía completa sobre el ciclo de vida del software.
5. W3C. (n.d.). *Introduction to Web Accessibility*. Recuperado de <https://www.w3.org/WAI/fundamentals/accessibility-intro/>
 - Referencia para principios de accesibilidad en diseño web.
6. Welling, L., & Thomson, L. (2017). *PHP and MySQL web development* (5th ed.). Addison-Wesley.
 - Manual detallado para el desarrollo backend con PHP y MySQL.
7. Mozilla Developer Network (MDN). (n.d.). *HTML: HyperText Markup Language*. Recuperado de <https://developer.mozilla.org/>
 - Documentación técnica utilizada para el desarrollo del frontend.
8. Lucidchart. (n.d.). *Herramienta de diagramación UML*. Recuperado de <https://www.lucidchart.com>
 - Plataforma empleada para la creación de diagramas UML.
9. Netlify. (n.d.). *Hosting y despliegue de aplicaciones*. Recuperado de <https://www.netlify.com>
 - Recurso para pruebas de hosting en entornos de despliegue.



Instituto Tecnológico de Apizaco



10. ISO/IEC. (2011). *ISO/IEC 25010:2011* -

Systems and software quality requirements and evaluation (SQuaRE) -

System and software quality models. Recuperado de

<https://www.iso.org/standard/35733.html>

- Referencia estándar para evaluar la calidad del software.

X I .- Anexos.

- Repositorio de Github del proyecto:
<https://github.com/Juanhnc74/WEBTECH.git>