

Contents

1	Introducción	2
2	Caracterización del área	3
2.1	Giro y creación de la empresa	3
2.2	Visión de la compañía	3
2.3	Objetivo General	3
2.4	Organigrama de la compañía	3
3	Objetivos: Generales y específicos	4
3.1	Objetivos generales	4
3.2	Objetivos específicos	4
4	Justificación	5
5	Problemas a resolver	6
6	Fundamento teórico	7
6.1	Desarrollo Full Stack	7
6.1.1	Frontend	7
6.1.2	Backend	8
6.1.3	Base de datos	8
6.2	APIs y comunicación entre sistemas	9
6.2.1	API REST	9
6.2.2	Importancia en la integración de frontend y backend	9
6.3	Herramientas de desarrollo	10
6.3.1	Editor de código	10
6.3.2	Control de versiones	10
7	Alcances y limitaciones	11
7.1	Alcance	11
7.2	Limitaciones	11
8	Procedimiento y descripción de las actividades realizadas	12
8.1	HDI Obra Civil	13
8.2	HDI Cristales	14
8.3	HDI Portal Verificador	21
8.4	Solcat	30
9	Conclusiones y recomendaciones	32
	Referencias Bibliográficas	33

1 Introducción

2 Caracterización del área

2.1 Giro y creación de la empresa

Blueservices SC es una compañía 100% mexicana, creada en octubre del 2016, con el propósito de ayudar a implementar software de la más alta calidad para sus socios comerciales, y desde entonces ha estado trabajando para ayudar a las empresas a aprovechar al máximo la tecnología y a enfrentar los desafíos digitales de la era moderna.

2.2 Visión de la compañía

Ser la empresa líder en desarrollo de software a la medida en Latinoamérica, reconocida por ofrecer tecnología funcional sin inversión inicial, capaz de transformar la operación de las empresas con soluciones accesibles, confiables y adaptadas a su realidad.

2.3 Objetivo General

Desarrollar soluciones tecnológicas personalizadas, accesibles y escalables que optimicen los procesos de nuestros clientes, sin barreras económicas de entrada, impulsando su crecimiento a través de software de alto impacto.

2.4 Organigrama de la compañía

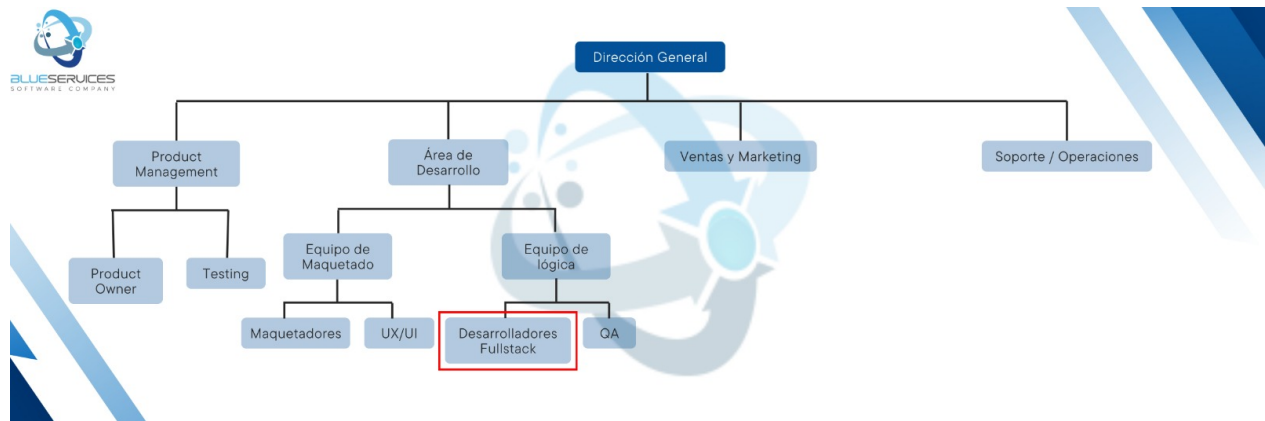


Figure 1: Organigrama de la compañía

Estuve participando en el área de "Desarrolladores Fullstack"

3 Objetivos: Generales y específicos

3.1 Objetivos generales

Realizar el ajuste funcional de los sistemas HDI mediante la corrección de errores y la implementación de nuevas funcionalidades, así como iniciar el desarrollo de un nuevo proyecto de software en la empresa Blueservices SC, durante el periodo del 14 de Julio al 08 de Septiembre de 2025, con el propósito de cubrir al menos un 80% de los problemas reportados por los clientes y optimizar los tiempos de operación.

3.2 Objetivos específicos

- Analizar los errores existentes en los sistemas HDI y proponer soluciones funcionales.
- Implementar las correcciones correspondientes para mejorar la estabilidad y funcionamiento de los sistemas.
- Desarrollar nuevas funcionalidades de acuerdo con las necesidades detectadas en HDI.
- Iniciar el desarrollo de un nuevo proyecto de software siguiendo las indicaciones del responsable del área.

4 Justificación

El ajuste funcional de los sistemas HDI y el desarrollo de nuevos proyectos de software son relevantes porque permiten mejorar la operatividad y eficiencia de las soluciones utilizadas por los clientes de la empresa. La corrección de errores y la implementación de nuevas funcionalidades contribuyen a optimizar los tiempos de operación, reducir incidencias y garantizar un servicio más confiable en los procesos internos de los clientes.

Asimismo, el desarrollo de proyectos de software encargados por los clientes brinda a la empresa la oportunidad de ampliar sus soluciones tecnológicas, ofreciendo herramientas modernas y adaptadas a requerimientos específicos.

5 Problemas a resolver

Durante la estancia profesional se identificaron diversas situaciones que representaban oportunidades de mejora en los sistemas desarrollados por la empresa, como HDI y Solcat. Entre los principales problemas se encuentran errores en la funcionalidad de los sistemas, limitaciones en ciertas operaciones, inconsistencias en la interacción entre diferentes módulos y oportunidades para optimizar la experiencia de uso de los clientes.

Estas situaciones hicieron evidente la necesidad de implementar correcciones, mejoras funcionales y nuevas características, con el objetivo de garantizar un desempeño más estable, eficiente y confiable de los sistemas, así como una experiencia más satisfactoria para los clientes que los utilizan.

6 Fundamento teórico

El desarrollo de aplicaciones web requiere contar con una base teórica que permita comprender su funcionamiento y las tecnologías involucradas en su construcción. Estos conocimientos abarcan tanto la implementación de funcionalidades en el frontend, la lógica del servidor en el backend y la gestión de la información en bases de datos, como la comunicación entre sistemas mediante APIs y el uso de herramientas que facilitan la programación y el control de versiones. Todo esto constituye el fundamento necesario para el desarrollo de aplicaciones web modernas, eficientes y funcionales.

Una aplicación web es un software que funciona directamente en el navegador sin necesidad de instalar nada. Se conecta a un servidor para procesar información y permite al usuario interactuar con funciones avanzadas, como comprar en línea, chatear, revisar el correo o hacer transacciones bancarias.

Sus principales ventajas son la accesibilidad (se puede usar desde cualquier dispositivo con internet), el desarrollo eficiente (una sola versión funciona en distintos navegadores), la simplicidad para el usuario (no requiere instalación y se actualiza sola) y la escalabilidad (puede crecer en usuarios sin necesidad de hardware adicional). (Amazon Web Services 2024)

6.1 Desarrollo Full Stack

Un desarrollador full stack es un programador capaz de trabajar tanto en el frontend (la parte visual e interactiva que ve el usuario en el navegador) como en el backend (la parte lógica y de servidor donde se procesan los datos). Su papel es conectar ambos lados para que la aplicación web funcione de manera completa, desde la interfaz hasta la base de datos.

Además de los conocimientos técnicos, se espera que un desarrollador full stack tenga competencias como proactividad, autonomía, capacidad analítica, organización, flexibilidad, resolución de problemas, rapidez para aprender, resiliencia y orientación a la calidad. (Santander 2024)

6.1.1 Frontend

El frontend constituye la capa de una aplicación web encargada de la interacción con el usuario. Su función es presentar la información de manera clara y permitir la comunicación fluida con la capa de backend, manteniendo el principio de separación de responsabilidades. En términos técnicos, el frontend se ejecuta en el navegador (cliente) y está compuesto principalmente por tres tecnologías fundamentales:

HTML	Define la estructura de los contenidos
CSS	Establece la presentación y estilos visuales
JavaScript	Aporta la lógica de interacción y el dinamismo en la interfaz

El desarrollo de frontend ha evolucionado significativamente, incorporando frameworks y librerías que permiten optimizar procesos, estandarizar diseños y aumentar la eficiencia en la construcción de interfaces. Herramientas como jQuery simplifican la manipulación del DOM y la gestión de eventos, mientras que frameworks como Bootstrap favorecen el diseño responsivo y la coherencia visual. A su vez, librerías como DataTables permiten gestionar de forma dinámica tablas de datos, ampliando la capacidad de interacción del usuario con la información.

En el contexto del proyecto desarrollado, el frontend se implementó combinando las tecnologías base (HTML, CSS y JavaScript) con el apoyo de herramientas como jQuery, Bootstrap y DataTables, logrando interfaces funcionales y adaptadas a las necesidades de la aplicación. (IONOS 2024)

6.1.2 Backend

El backend es la capa de una aplicación web responsable de la lógica de negocio, el procesamiento de datos y la comunicación con el servidor y la base de datos. A diferencia del frontend, que constituye la interfaz visible para el usuario, el backend opera de manera invisible, ejecutando las acciones necesarias para que las interacciones del usuario se traduzcan en operaciones funcionales dentro del sistema. Su correcto funcionamiento es esencial, ya que de él dependen aspectos como la seguridad, la eficiencia en el uso de recursos, la persistencia de datos y la disponibilidad de los servicios.

Entre las tareas habituales del backend se encuentran el diseño e implementación de funciones lógicas, la conexión con bases de datos, la integración de librerías del servidor y la aplicación de mecanismos de optimización como el manejo de caché o la compresión de archivos.

El desarrollo backend requiere conocimientos sólidos en programación, metodologías de desarrollo y configuración de entornos de ejecución. Lenguajes como PHP, Java, Python o .NET se encuentran entre los más utilizados, siendo elegidos en función de las necesidades y características de cada proyecto. Además, la coordinación con el frontend es indispensable para lograr sistemas plenamente funcionales, en los que la comunicación fluya a través de interfaces como las API REST.

En el contexto del proyecto desarrollado, el backend fue implementado principalmente en PHP, lenguaje que permitió gestionar la lógica del sistema, establecer la conexión con la base de datos y proporcionar los servicios necesarios para el funcionamiento del frontend. (Argonilla s.f.)

6.1.3 Base de datos

Una base de datos es un sistema organizado para almacenar, gestionar y proteger grandes volúmenes de información de manera estructurada. Permite a las organizaciones centralizar los datos, garantizar su integridad, facilitar el acceso controlado y habilitar operaciones complejas de consulta y análisis. Las bases de datos constituyen un componente esencial de las aplicaciones web, ya que mantienen la información necesaria para el correcto funcionamiento de los servicios ofrecidos a los usuarios, desde transacciones hasta registros de contenido dinámico.

Existen diferentes tipos de bases de datos, cada una adaptada a necesidades específicas. Las bases de datos relacionales organizan la información en tablas con filas y columnas, facilitando el uso de lenguaje SQL para la gestión de los datos. Son especialmente útiles para almacenar información estructurada, como registros de usuarios, inventarios o transacciones financieras. Por otro lado, las bases de datos no relacionales o NoSQL permiten manejar datos no estructurados, como documentos, gráficos o pares clave-valor, siendo más flexibles frente a la variedad de formatos de información. También existen bases de datos orientadas a objetos, vectoriales y basadas en la nube, cada una con aplicaciones particulares según el tipo de proyecto y la necesidad de escalabilidad, rendimiento y análisis avanzado.

Para interactuar de manera eficiente con estas bases de datos, se utilizan los Sistemas Gestores de Bases de Datos (SGBD), que permiten crear, consultar, modificar y mantener la integridad de los datos. Entre las opciones más comunes se encuentran MySQL, PostgreSQL, Microsoft SQL Server y Oracle Database. En particular, MySQL es un SGBD relacional ampliamente utilizado en entornos web por su fiabilidad, compatibilidad con múltiples plataformas y facilidad para integrarse con aplicaciones desarrolladas en diversos lenguajes de programación.

En el contexto del proyecto, la empresa utiliza MySQL como base de datos principal para gestionar la información del sistema. Para interactuar con la base de datos y facilitar su comprensión y administración, se empleó DBeaver, un gestor universal que permite visualizar esquemas, tablas y relaciones a través de diagramas gráficos, ejecutar consultas SQL y mantener un control detallado sobre los datos. (Kosinski 2024)

6.2 APIs y comunicación entre sistemas

En el desarrollo de aplicaciones web modernas, los sistemas rara vez funcionan de manera aislada. Para que distintos componentes, aplicaciones o servicios puedan intercambiar información y trabajar en conjunto, se utilizan las APIs (Application Programming Interfaces). Estas interfaces permiten que el frontend y el backend, así como otros sistemas externos, se comuniquen de forma eficiente y estructurada, asegurando que los datos y funcionalidades puedan compartirse de manera confiable y segura.

6.2.1 API REST

Una API REST (Representational State Transfer) es un tipo de interfaz de programación de aplicaciones que permite la comunicación entre sistemas de manera eficiente y estructurada, siguiendo los principios del estilo arquitectónico REST, definido por Roy Fielding en el año 2000. Este enfoque facilita la creación de APIs ligeras y escalables, ideales para el intercambio de datos entre aplicaciones, servicios web y bases de datos, así como para integrar componentes en arquitecturas modernas, como los microservicios.

Las API REST se basan en varios principios de diseño fundamentales: interfaz uniforme, desacoplamiento cliente-servidor, ausencia de estado en las solicitudes, capacidad de almacenamiento en caché, arquitectura en capas y la posibilidad de enviar código bajo demanda. Estas restricciones aseguran que las API sean flexibles, consistentes y escalables.

En la práctica, las API REST utilizan solicitudes HTTP para interactuar con los recursos de un sistema, ejecutando operaciones como crear, leer, actualizar y eliminar información. Los datos se entregan en formatos estándar como JSON, que facilita su comprensión por humanos y máquinas, y permiten una comunicación segura y controlada entre el frontend, el backend y otros servicios externos.

El uso de API REST es esencial en el desarrollo web moderno, ya que permite a los distintos componentes de un sistema intercambiar información de forma clara, confiable y compatible con múltiples lenguajes de programación y plataformas. (IBM 2025)

6.2.2 Importancia en la integración de frontend y backend

En el desarrollo moderno de aplicaciones web, la colaboración efectiva entre Frontend y Backend es esencial para crear sistemas funcionales, eficientes y fáciles de mantener. Las API REST juegan un papel fundamental en esta integración, ya que actúan como un puente estandarizado que permite al Frontend comunicarse con el Backend de manera clara y segura.

El Frontend, responsable de la interfaz y experiencia del usuario, depende de los datos y servicios proporcionados por el Backend, que gestiona la lógica de negocio, la seguridad y el almacenamiento de datos. Una API REST bien diseñada garantiza que ambos equipos puedan trabajar de manera independiente y paralela, respetando sus roles y responsabilidades, al mismo tiempo que facilita la coordinación, comunicación y consumo de datos.

Entre los beneficios principales de usar API REST para esta integración se encuentran:

- Claridad y consistencia en los datos: Los Endpoints proporcionan recursos estandarizados que permiten al Frontend acceder a la información correcta, evitando errores de interpretación.
- Independencia de tecnologías: REST permite que Frontend y Backend trabajen con lenguajes o frameworks distintos sin afectar la comunicación.
- Escalabilidad y eficiencia: Al ser sin estado, cada solicitud incluye toda la información necesaria, lo que simplifica la arquitectura y mejora la capacidad de respuesta del sistema.
- Adaptabilidad a cambios: Ante modificaciones en los requerimientos, las APIs REST permiten que ambos sistemas ajusten su trabajo sin interrumpir completamente el desarrollo del proyecto.

Las API REST son el pilar de la comunicación entre Frontend y Backend, asegurando que los sistemas puedan trabajar de forma eficiente, entregando productos coherentes, de alta calidad y adaptables a cambios futuros. (Reyes 2023)

6.3 Herramientas de desarrollo

En el desarrollo de software moderno, las herramientas juegan un papel esencial para facilitar el trabajo de los programadores y asegurar la calidad de los proyectos. Contar con un buen entorno de desarrollo no solo acelera la escritura de código, sino que también mejora la organización, la colaboración en equipo y el control de cambios. Entre las herramientas más utilizadas se encuentran los editores de código, que ofrecen funciones avanzadas para programar de forma eficiente, y los sistemas de control de versiones, que permiten gestionar la evolución de un proyecto, coordinar el trabajo de varios desarrolladores y mantener un historial confiable de modificaciones.

6.3.1 Editor de código

Los editores de código son herramientas fundamentales en el trabajo de cualquier desarrollador, ya que permiten escribir, organizar y depurar el código de manera más eficiente que un simple editor de texto. Están diseñados específicamente para el desarrollo de software e incluyen funcionalidades como el resaltado de sintaxis, auto completado, integración con sistemas de control de versiones y soporte para múltiples lenguajes de programación.

Entre los editores más utilizados en la industria destacan Sublime Text, Atom, Notepad++ y JetBrains IntelliJ, aunque uno de los más populares en la actualidad es Visual Studio Code (VS Code), desarrollado por Microsoft. Su éxito radica en que combina ligereza y rapidez con un alto grado de personalización. A través de extensiones, los desarrolladores pueden adaptar el entorno a las necesidades específicas de cada proyecto, añadiendo soporte para frameworks, herramientas de depuración o integración con servicios externos.

En mi caso, prefiero trabajar con Visual Studio Code, precisamente por estas posibilidades de personalización y por la amplia variedad de extensiones disponibles, que permiten optimizar el flujo de trabajo. Esto lo convierte en una herramienta flexible y adecuada tanto para proyectos pequeños como para desarrollos más complejos en entornos full stack. (Aurora 2024)

6.3.2 Control de versiones

El control de versiones es una herramienta esencial en el desarrollo de software, ya que permite llevar un registro de los cambios realizados en el código a lo largo del tiempo. Estos sistemas funcionan generando "instantáneas" del estado de los archivos cada vez que se realizan modificaciones, de modo que es posible recuperar versiones anteriores, comparar diferencias o incluso revertir errores sin poner en riesgo la integridad del proyecto.

Sin un sistema de control de versiones, los desarrolladores se verían obligados a mantener múltiples copias del código, lo cual resulta poco práctico y riesgoso, pues cualquier error podría provocar la pérdida de trabajo. En cambio, con un sistema centralizado de versiones, es posible administrar los cambios de manera organizada, evitando conflictos y facilitando la colaboración entre varios integrantes de un equipo de desarrollo.

Además de ofrecer la posibilidad de restaurar estados previos del software, el control de versiones permite mantener un historial completo de las modificaciones, incluyendo quién realizó cada cambio y con qué propósito. Esto resulta especialmente valioso cuando se requiere comprender la evolución del proyecto, solucionar errores heredados o documentar de forma clara el proceso de desarrollo.

En el caso de la empresa en la que realicé mis actividades, se utilizaba un sistema de control de versiones propio, lo cual aseguraba que todos los desarrolladores trabajaran bajo las mismas normas y flujos definidos internamente. (Microsoft 2023)

7 Alcances y limitaciones

7.1 Alcance

El alcance del proyecto no se limitó únicamente a la empresa donde se realizaron las actividades, ya que los cambios y ajustes implementados en el software impactaban directamente en la empresa cliente que solicitó dicho desarrollo. De esta manera, el trabajo realizado no solo beneficiaba a la organización receptora de mis prácticas, sino que también tenía repercusión en procesos externos al formar parte de un producto entregado a otra entidad.

7.2 Limitaciones

En cuanto a las limitaciones, estas estuvieron relacionadas principalmente con la confidencialidad de la información, pues ciertos detalles técnicos del software y datos del cliente no podían compartirse de manera abierta. Asimismo, el proyecto se desarrolló dentro de un tiempo determinado, lo que limitaba el número de funcionalidades o ajustes que podían implementarse durante el periodo de prácticas. Finalmente, también existían limitaciones técnicas y de recursos, ya que el trabajo se enfocaba únicamente en las tareas y módulos asignados, sin abarcar la totalidad del sistema ni la posibilidad de modificar herramientas externas a las establecidas por la empresa.

8 Procedimiento y descripción de las actividades realizadas

Para la realización de las prácticas profesionales en la empresa, se estableció un periodo comprendido entre el 14 de Julio y el 08 de Septiembre del 2025, durante el cual se desarrollaron actividades de carácter práctico enfocadas en el mantenimiento, ajuste y evolución del software solicitado por un cliente externo. Como parte de la metodología de trabajo, la empresa solicitaba la entrega de reportes semanales que permitieran dar seguimiento al avance de las tareas encomendadas y mantener un control de los resultados obtenidos.

Actividades	S E M A N A S								Horas
	14-18/07	21-25/07	28-01/08	04-08/08	11-15/08	18-22/08	25-29/08	01-05/09	
Correcciones a los estilos de HDI Obra Civil									28
Correcciones a los estilos de HDI Cristales									27
Implementación de nueva lógica backend en HDI Cristales									32
Implementación de login, autenticación y recuperación de contraseña en HDI Portal Verificador									12
Implementación de frontend en HDI Portal Verificador									39
Implementación de backend en HDI Portal Verificador									57
Desarrollo de nuevo software requisitado por los clientes									85
TOTAL									280

Figure 2: Cronograma de actividades

8.1 HDI Obra Civil

Implementación de paginado truncado en la sección Inicio y Reportería

- **Descripción:** Implementé un sistema de paginación para las secciones de Inicio y Reportería que muestra un máximo de cinco páginas cercanas a la actual, además de un enlace a la última página, con el objetivo de mejorar la navegación en tablas con grandes volúmenes de datos.
- **Proceso:** El desarrollo lo realicé en el frontend utilizando JavaScript. La lógica implementada calcula el rango de páginas a mostrar basándose en la página actual y el total de páginas. Si la página actual está en la primera mitad, se muestran las primeras n páginas; si está en la segunda mitad, se muestran páginas alrededor de la actual. Utilicé como base un código existente del proyecto HDI Cristales para mantener la coherencia.

Actividades	SEMANAS								Horas
	14-18/07	21-25/07	28-01/08	04-08/08	11-15/08	18-22/08	25-29/08	01-05/09	
Correcciones a los estilos de HDI Obra Civil									28
Correcciones a los estilos de HDI Cristales									27
Implementación de nueva lógica backend en HDI Cristales									32
Implementación de login, autenticación y recuperación de contraseña en HDI Portal Verificador									12
Implementación de frontend en HDI Portal Verificador									39
Implementación de backend en HDI Portal Verificador									57
Desarrollo de nuevo software requisitado por los clientes									85
TOTAL									280

Figure 3: cronograma

8.2 HDI Cristales

Corrección de estilo CSS para evitar superposición del sidebar en el header

- **Descripción:** Corregí un error de visualización donde el sidebar se superponía al header de la aplicación, lo cual afectaba la usabilidad y la estética de la interfaz.
- **Proceso:** Realicé la modificación en el archivo CSS. Tras revisar el código, identifiqué que una propiedad padding-top necesaria para el correcto posicionamiento se encontraba comentada. Solucioné el problema descomentando esta propiedad y ajustando su valor, lo que restableció el layout diseñado.

Actividades	S E M A N A S								Horas
	14-18/07	21-25/07	28-01/08	04-08/08	11-15/08	18-22/08	25-29/08	01-05/09	
Correcciones a los estilos de HDI Obra Civil									28
Correcciones a los estilos de HDI Cristales									27
Implementación de nueva lógica backend en HDI Cristales									32
Implementación de login, autenticación y recuperación de contraseña en HDI Portal Verificador									12
Implementación de frontend en HDI Portal Verificador									39
Implementación de backend en HDI Portal Verificador									57
Desarrollo de nuevo software requisitado por los clientes									85
TOTAL									280

Figure 4: cronograma

Implementación de paginado truncado en la sección Siniestros Pausados y Administrador

- **Descripción:** Realicé una revisión del código para verificar la implementación del paginado truncado en las secciones de Siniestros Pausados y Administrador.
- **Proceso:** Confirmé que la funcionalidad de paginado ya estaba implementada correctamente en el JavaScript de estas secciones, utilizando la misma lógica de la tarea de HDI Obra Civil. La falta de datos suficientes en el entorno de prueba era la razón por la que el paginado no era visible, pero el código estaba operativo.

Actividades	S E M A N A S								Horas
	14-18/07	21-25/07	28-01/08	04-08/08	11-15/08	18-22/08	25-29/08	01-05/09	
Correcciones a los estilos de HDI Obra Civil									28
Correcciones a los estilos de HDI Cristales									27
Implementación de nueva lógica backend en HDI Cristales									32
Implementación de login, autenticación y recuperación de contraseña en HDI Portal Verificador									12
Implementación de frontend en HDI Portal Verificador									39
Implementación de backend en HDI Portal Verificador									57
Desarrollo de nuevo software requisitado por los clientes									85
TOTAL									280

Figure 5: cronograma

Permitir ingresar un . (punto) como valor en el Input de “Recibimos de HDI Seguros S.A. de C.V.”

- **Descripción:** Modifiqué la validación de un campo input de tipo numérico para que aceptara el carácter punto (.) como una representación válida de un monto cero (\$0), una necesidad requerida por el cliente en el área de Soporte Documental.
- **Proceso:** Implementé una solución algo arriesgada, ya que es técnicamente imposible hacer que un input de tipo numérico acepte un valor que no sea numérico. Agregué un event listener (keypress) para detectar la pulsación de la tecla del punto y utilicé una bandera (puntoPresionado) para rastrear este evento. Luego, en la función de validación del formulario (validarDocumentos), si el campo estaba vacío y la bandera era verdadera, asigné automáticamente el valor 0 al campo.

Actividades	SEMANAS								Horas
	14-18/07	21-25/07	28-01/08	04-08/08	11-15/08	18-22/08	25-29/08	01-05/09	
Correcciones a los estilos de HDI Obra Civil									28
Correcciones a los estilos de HDI Cristales									27
Implementación de nueva lógica backend en HDI Cristales									32
Implementación de login, autenticación y recuperación de contraseña en HDI Portal Verificador									12
Implementación de frontend en HDI Portal Verificador									39
Implementación de backend en HDI Portal Verificador									57
Desarrollo de nuevo software requisitado por los clientes									85
TOTAL									280

Figure 6: cronograma

Habilitación del modal de notificaciones y animación del icono

- **Descripción:** Habilité la funcionalidad para desplegar un modal al hacer clic en el ícono de notificaciones y añadí una animación a dicho ícono cuando existían notificaciones sin leer.
- **Proceso:** En el frontend, vinculé la función preexistente abrirCerrarModal() al evento click del ícono de notificaciones. Para la animación, creé una nueva clase CSS que definía un efecto de parpadeo y desarrollé la lógica para agregar o quitar esta clase al ícono basándome en la presencia de notificaciones.

Actividades	S E M A N A S								Horas
	14-18/07	21-25/07	28-01/08	04-08/08	11-15/08	18-22/08	25-29/08	01-05/09	
Correcciones a los estilos de HDI Obra Civil									28
Correcciones a los estilos de HDI Cristales									27
Implementación de nueva lógica backend en HDI Cristales									32
Implementación de login, autenticación y recuperación de contraseña en HDI Portal Verificador									12
Implementación de frontend en HDI Portal Verificador									39
Implementación de backend en HDI Portal Verificador									57
Desarrollo de nuevo software requisitado por los clientes									85
TOTAL									280

Figure 7: cronograma

Desarrollo del backend para obtener notificaciones y mostrarlas en un modal

- **Descripción:** Desarrollé la funcionalidad completa del sistema de notificaciones, incluyendo la creación de un endpoint API y su consumo desde el frontend para mostrar las notificaciones en un modal.
- **Proceso:**
 1. **Backend:** Creé un archivo backNotificaciones.php e implementé una consulta SQL para seleccionar las notificaciones pendientes de la tabla data_generates asociadas al ID del usuario.
 2. **Frontend:** Realicé una consulta con fetch al endpoint correspondiente, enviando el ID de usuario y el tokenAPI para consultar las notificaciones y mostrarlas en la tabla del modal. En caso de no existir notificaciones, se despliega un mensaje informativo; si ocurre un error, se muestra un modal genérico de error. Una vez recibidos los datos, se cuentan las filas resultantes mediante la función implementada en la tarea previa, lo que permite determinar si se debe animar el ícono. Finalmente, esta consulta se ejecuta automáticamente cada minuto.

Actividades	SEMANAS								Horas
	14-18/07	21-25/07	28-01/08	04-08/08	11-15/08	18-22/08	25-29/08	01-05/09	
Correcciones a los estilos de HDI Obra Civil									28
Correcciones a los estilos de HDI Cristales									27
Implementación de nueva lógica backend en HDI Cristales									32
Implementación de login, autenticación y recuperación de contraseña en HDI Portal Verificador									12
Implementación de frontend en HDI Portal Verificador									39
Implementación de backend en HDI Portal Verificador									57
Desarrollo de nuevo software requisitado por los clientes									85
TOTAL									280

Figure 8: cronograma

Modificación del funcionamiento para pausar y reanudar siniestros

- **Descripción:** Optimicé la experiencia de usuario en las pantallas de gestión de siniestros para que las acciones de pausar y reanudar actualizaran la interfaz de manera dinámica e inmediata, eliminando la necesidad de recargar manualmente la página para visualizar los cambios de estado.
- **Proceso:** Modifiqué el código en las pantallas de "Mis Siniestros" y "Siniestros Pausados" para que, tras consultar a la API que cambia el estado, la fila del siniestro afectado se removiera automáticamente de la tabla de listado. Para la pantalla de "Detalle del Siniestro", implementé una lógica que, tras cambiar el estado, actualizaba dinámicamente el icono y la funcionalidad del botón principal: si el siniestro estaba en curso, el botón cambiaba a un icono de pausa y su acción correspondiente; si estaba pausado, el botón se actualizaba a un icono de reanudar, ajustando también su funcionalidad para ejecutar la acción inversa.

Actividades	S E M A N A S								Horas
	14-18/07	21-25/07	28-01/08	04-08/08	11-15/08	18-22/08	25-29/08	01-05/09	
Correcciones a los estilos de HDI Obra Civil									28
Correcciones a los estilos de HDI Cristales									27
Implementación de nueva lógica backend en HDI Cristales									32
Implementación de login, autenticación y recuperación de contraseña en HDI Portal Verificador									12
Implementación de frontend en HDI Portal Verificador									39
Implementación de backend en HDI Portal Verificador									57
Desarrollo de nuevo software requisitado por los clientes									85
TOTAL									280

Figure 9: cronograma

Implementación de la pantalla para verificar la firma de documentos de siniestros

- **Descripción:** Implementé una nueva pantalla para consultar y verificar la firma digital de documentos por parte de los afectados de un siniestro.
- **Proceso:** Desarrollé la interfaz con HTML, CSS y JavaScript, implementando la búsqueda de siniestros, la visualización de datos asociados y la descarga de documentos. El componente final fue la implementación de la lógica para enviar notificaciones de recordatorio al cliente. Además, realicé correcciones de estilos CSS para asegurar la correcta visualización.

Actividades	S E M A N A S								Horas
	14-18/07	21-25/07	28-01/08	04-08/08	11-15/08	18-22/08	25-29/08	01-05/09	
Correcciones a los estilos de HDI Obra Civil									28
Correcciones a los estilos de HDI Cristales									27
Implementación de nueva lógica backend en HDI Cristales									32
Implementación de login, autenticación y recuperación de contraseña en HDI Portal Verificador									12
Implementación de frontend en HDI Portal Verificador									39
Implementación de backend en HDI Portal Verificador									57
Desarrollo de nuevo software requisitado por los clientes									85
TOTAL									280

Figure 10: cronograma

8.3 HDI Portal Verificador

Desarrollo e implementación del sistema de Login y recuperación de contraseña

- **Descripción:** Implementé el sistema de autenticación para el nuevo proyecto HDI Portal Verificador, permitiendo a los usuarios iniciar sesión y recuperar su contraseña.
- **Proceso:** Basándome en la estructura de HDI Cristales, añadí los archivos esenciales de backend: conexión a la base de datos (mysql.php), funciones auxiliares (funciones.php) y definición de rutas (archivos_rutas.php). Programé la lógica de sesiones para validar las credenciales. Para la recuperación de contraseña, implementé un flujo completo que incluye la solicitud por correo, validación por token y la creación de una nueva contraseña.

Actividades	S E M A N A S								Horas
	14-18/07	21-25/07	28-01/08	04-08/08	11-15/08	18-22/08	25-29/08	01-05/09	
Correcciones a los estilos de HDI Obra Civil									28
Correcciones a los estilos de HDI Cristales									27
Implementación de nueva lógica backend en HDI Cristales									32
Implementación de login, autenticación y recuperación de contraseña en HDI Portal Verificador									12
Implementación de frontend en HDI Portal Verificador									39
Implementación de backend en HDI Portal Verificador									57
Desarrollo de nuevo software requisitado por los clientes									85
TOTAL									280

Figure 11: cronograma

Creación de APIs GET para consulta de datos (usuarios, puestos, líneas de negocio, subdirecciones, oficinas)

- **Descripción:** Desarrollé un conjunto de APIs de tipo GET para proporcionar los datos necesarios para los formularios del sistema.
- **Proceso:** Programé endpoints para consultar usuarios filtrados por tipo de puesto, permitiendo obtener listas de administradores, investigadores, etc. Además, creé APIs independientes para obtener las listas completas de puestos, líneas de negocio, subdirecciones y oficinas disponibles en el sistema.

Actividades	SEMANAS								Horas
	14-18/07	21-25/07	28-01/08	04-08/08	11-15/08	18-22/08	25-29/08	01-05/09	
Correcciones a los estilos de HDI Obra Civil									28
Correcciones a los estilos de HDI Cristales									27
Implementación de nueva lógica backend en HDI Cristales									32
Implementación de login, autenticación y recuperación de contraseña en HDI Portal Verificador									12
Implementación de frontend en HDI Portal Verificador									39
Implementación de backend en HDI Portal Verificador									57
Desarrollo de nuevo software requisitado por los clientes									85
TOTAL									280

Figure 12: cronograma

Creación de API POST para el alta de usuarios en el sistema

- **Descripción:** Desarrollé una API para el registro de nuevos usuarios en el sistema.
- **Proceso:** Implementé un endpoint que recibe por método POST los datos del usuario (nombre, correo, teléfono, id de puesto, oficina y subdirección). Incluí validaciones para verificar que ningún campo obligatorio estuviera vacío y que los IDs de puesto, oficina y subdirección existieran en la base de datos. Utilicé Postman para realizar pruebas exhaustivas de todos los escenarios posibles.

Actividades	S E M A N A S								Horas
	14-18/07	21-25/07	28-01/08	04-08/08	11-15/08	18-22/08	25-29/08	01-05/09	
Correcciones a los estilos de HDI Obra Civil									28
Correcciones a los estilos de HDI Cristales									27
Implementación de nueva lógica backend en HDI Cristales									32
Implementación de login, autenticación y recuperación de contraseña en HDI Portal Verificador									12
Implementación de frontend en HDI Portal Verificador									39
Implementación de backend en HDI Portal Verificador									57
Desarrollo de nuevo software requisitado por los clientes									85
TOTAL									280

Figure 13: cronograma

Creación de API POST para el alta de siniestros

- **Descripción:** Desarrollé una API para dar de alta siniestros en el sistema.
- **Proceso:** Implementé un endpoint que recibe por método POST los datos del siniestro. Incluí validaciones para verificar que ningún campo obligatorio estuviera vacío, que los datos tuvieran el formato correcto y que los IDs de investigador, agente, oficina y línea de negocio existieran en sus tablas correspondientes de la base de datos. Utilicé Postman para realizar pruebas exhaustivas con diferentes JSON que contenían todos los campos requeridos.

Actividades	S E M A N A S								Horas
	14-18/07	21-25/07	28-01/08	04-08/08	11-15/08	18-22/08	25-29/08	01-05/09	
Correcciones a los estilos de HDI Obra Civil									28
Correcciones a los estilos de HDI Cristales									27
Implementación de nueva lógica backend en HDI Cristales									32
Implementación de login, autenticación y recuperación de contraseña en HDI Portal Verificador									12
Implementación de frontend en HDI Portal Verificador									39
Implementación de backend en HDI Portal Verificador									57
Desarrollo de nuevo software requisitado por los clientes									85
TOTAL									280

Figure 14: cronograma

Réplica de la pantalla de administración basada en HDI Cristales

- **Descripción:** Repliqué la interfaz completa de administración de HDI Cristales para el Portal Verificador, creando las secciones para gestionar usuarios y puestos.
- **Proceso:** Adapté el código HTML, CSS y JavaScript de la pantalla de administración. Implementé las tablas para listar usuarios y puestos, integrando las APIs previamente creadas para cargar los datos. También añadí funcionalidades como búsqueda y paginación para manejar grandes volúmenes de información de manera eficiente.

Actividades	S E M A N A S								Horas
	14-18/07	21-25/07	28-01/08	04-08/08	11-15/08	18-22/08	25-29/08	01-05/09	
Correcciones a los estilos de HDI Obra Civil									28
Correcciones a los estilos de HDI Cristales									27
Implementación de nueva lógica backend en HDI Cristales									32
Implementación de login, autenticación y recuperación de contraseña en HDI Portal Verificador									12
Implementación de frontend en HDI Portal Verificador									39
Implementación de backend en HDI Portal Verificador									57
Desarrollo de nuevo software requisitado por los clientes									85
TOTAL									280

Figure 15: cronograma

Implementación de la sección “Lista Artículos 69” con funcionalidad CRUD

- **Descripción:** Implementé una nueva sección dentro de la administración que permite visualizar, crear, habilitar e inhabilitar artículos de un reglamento.
- **Proceso:** En el backend, programé el código para consultar, crear, editar y eliminar artículos. En el frontend, desarrollé la interfaz con una tabla, un modal para agregar nuevos artículos y botones para cambiar su estado (habilitado/inhabilitado).

Actividades	S E M A N A S								Horas
	14-18/07	21-25/07	28-01/08	04-08/08	11-15/08	18-22/08	25-29/08	01-05/09	
Correcciones a los estilos de HDI Obra Civil									28
Correcciones a los estilos de HDI Cristales									27
Implementación de nueva lógica backend en HDI Cristales									32
Implementación de login, autenticación y recuperación de contraseña en HDI Portal Verificador									12
Implementación de frontend en HDI Portal Verificador									39
Implementación de backend en HDI Portal Verificador									57
Desarrollo de nuevo software requisitado por los clientes									85
TOTAL									280

Figure 16: cronograma

Implementación de paginado y select de cantidad de artículos a mostrar

- **Descripción:** Mejoré la usabilidad de la tabla de artículos añadiendo paginación y la posibilidad de elegir cuántos elementos visualizar por página.
- **Proceso:** Adapté la lógica de paginación truncada utilizada en otros proyectos para la tabla de artículos. Modifiqué tanto el backend para que aceptara los parámetros de página y límite, como el frontend para que mostrara los controles de paginación y un desplegable para seleccionar el número de registros (10, 25 o 50).

Actividades	S E M A N A S								Horas
	14-18/07	21-25/07	28-01/08	04-08/08	11-15/08	18-22/08	25-29/08	01-05/09	
Correcciones a los estilos de HDI Obra Civil									28
Correcciones a los estilos de HDI Cristales									27
Implementación de nueva lógica backend en HDI Cristales									32
Implementación de login, autenticación y recuperación de contraseña en HDI Portal Verificador									12
Implementación de frontend en HDI Portal Verificador									39
Implementación de backend en HDI Portal Verificador									57
Desarrollo de nuevo software requisitado por los clientes									85
TOTAL									280

Figure 17: cronograma

Adición de modales de confirmación para acciones críticas

- **Descripción:** Añadí modales de confirmación para prevenir la eliminación accidental de usuarios o el cambio inadvertido de estado de un artículo.
- **Proceso:** Programé la lógica para que, al hacer clic en los botones de "Eliminar" usuario o "Habilitar/Inhabilitar" artículo, se mostrara un modal con un mensaje de advertencia. La acción sólo se ejecutaba tras la confirmación del usuario.

Actividades	S E M A N A S								Horas
	14-18/07	21-25/07	28-01/08	04-08/08	11-15/08	18-22/08	25-29/08	01-05/09	
Correcciones a los estilos de HDI Obra Civil									28
Correcciones a los estilos de HDI Cristales									27
Implementación de nueva lógica backend en HDI Cristales									32
Implementación de login, autenticación y recuperación de contraseña en HDI Portal Verificador									12
Implementación de frontend en HDI Portal Verificador									39
Implementación de backend en HDI Portal Verificador									57
Desarrollo de nuevo software requisitado por los clientes									85
TOTAL									280

Figure 18: cronograma

Investigación y consumo de la API “INVESTIGADORES QA”

- **Descripción:** Realicé un análisis de la API externa “INVESTIGADORES QA” para entender su funcionamiento e intentar consumirla.
- **Proceso:** Identifiqué y probé los endpoints de la API utilizando Postman. Logré obtener un token de acceso autenticándome con las credenciales dadas y probé el endpoint para subir comentarios de un siniestro. Sin embargo, los endpoints de consulta (GET) siempre devolvían un error 404, por lo que documenté este impedimento para su posterior revisión, aunque no logré resolverlo.

Actividades	SEMANAS								Horas
	14-18/07	21-25/07	28-01/08	04-08/08	11-15/08	18-22/08	25-29/08	01-05/09	
Correcciones a los estilos de HDI Obra Civil									28
Correcciones a los estilos de HDI Cristales									27
Implementación de nueva lógica backend en HDI Cristales									32
Implementación de login, autenticación y recuperación de contraseña en HDI Portal Verificador									12
Implementación de frontend en HDI Portal Verificador									39
Implementación de backend en HDI Portal Verificador									57
Desarrollo de nuevo software requisitado por los clientes									85
TOTAL									280

Figure 19: cronograma

8.4 Solcat

Implementación de selects con búsqueda para Estado y Municipio

- **Descripción:** Reemplacé un input de texto simple por dos desplegables con búsqueda en vivo para la selección de Estado y Municipio en el modal de "Destino local", mejorando significativamente la usabilidad y precisión de los datos.
- **Proceso:** Reemplacé el input "Origen - Destino" por dos elementos select con búsqueda: uno para Estado y otro para Municipio. Implemente la dependencia entre ambos campos: la selección de un Estado activaba y cargaba el desplegable de Municipios, mostrando únicamente los datos correspondientes. Además, implementé un botón que tomaba los valores seleccionados y los añadía a una tabla, permitiendo registrar múltiples combinaciones de Estado y Municipio.

Actividades	S E M A N A S								Horas
	14-18/07	21-25/07	28-01/08	04-08/08	11-15/08	18-22/08	25-29/08	01-05/09	
Correcciones a los estilos de HDI Obra Civil									28
Correcciones a los estilos de HDI Cristales									27
Implementación de nueva lógica backend en HDI Cristales									32
Implementación de login, autenticación y recuperación de contraseña en HDI Portal Verificador									12
Implementación de frontend en HDI Portal Verificador									39
Implementación de backend en HDI Portal Verificador									57
Desarrollo de nuevo software requisitado por los clientes									85
TOTAL									280

Figure 20: cronograma

Modificación de tablas de base de datos e inserción de datos de Origen-Destino

- **Descripción:** Adapté la estructura de la base de datos y la lógica del backend para soportar el nuevo formato de datos de Origen-Destino, que ahora incluía Estado y Municipio de manera separada.
- **Proceso:** Añadí las columnas estado y municipio a las tablas origenServicios y destinoServicios en la base de datos. Modifiqué el código PHP del backend para que, en lugar de recibir un solo valor de "Origen-Destino", procesara los arrays de estados y municipios enviados desde la tabla en el frontend, realizando inserciones múltiples en la base de datos por cada par seleccionado por el usuario.

Actividades	S E M A N A S								Horas
	14-18/07	21-25/07	28-01/08	04-08/08	11-15/08	18-22/08	25-29/08	01-05/09	
Correcciones a los estilos de HDI Obra Civil									28
Correcciones a los estilos de HDI Cristales									27
Implementación de nueva lógica backend en HDI Cristales									32
Implementación de login, autenticación y recuperación de contraseña en HDI Portal Verificador									12
Implementación de frontend en HDI Portal Verificador									39
Implementación de backend en HDI Portal Verificador									57
Desarrollo de nuevo software requisitado por los clientes									85
TOTAL									280

Figure 21: cronograma

9 Conclusiones y recomendaciones

Referencias Bibliográficas

- Amazon Web Services, Inc. (2024). *¿Qué es una aplicación web?* URL: <https://aws.amazon.com/es/what-is/web-application>.
- Argonilla, Rafa (s.f.). *BackEnd*. URL: <https://rafarjonilla.com/que-es/backend>.
- Aurora (2024). *¿Qué es un editor de código?* URL: <https://iddigitalschool.com/bootcamps/que-es-un-editor-de-codigo>.
- IBM (2025). *¿Qué es la API REST?* URL: <https://www.ibm.com/mx-es/think/topics/rest-apis>.
- IONOS (2024). *¿Qué es el Frontend?* URL: <https://www.ionos.com/es-us/digitalguide/paginas-web/creacion-de-paginas-web/que-es-el-frontend>.
- Kosinski, Matthew (2024). *¿Qué es una base de datos?* URL: <https://www.ibm.com/mx-es/think/topics/database>.
- Microsoft (2023). *¿Qué es el control de versiones?* URL: <https://learn.microsoft.com/es-es/devops/develop/git/what-is-version-control>.
- Reyes, Dennys José Márquez (2023). *Colaboración Efectiva FrontEnd/BackEnd: Flujo de Trabajo Adecuado y Responsabilidades en el Desarrollo de Aplicaciones Web*. URL: <https://dev.to/dennysjmarquez/colaboracion-efectiva-frontendbackend-flujo-de-trabajo-adecuado-y-responsabilidades-en-el-desarrollo-de-aplicaciones-web-3ae0>.
- Santander (2024). *Desarrollador full stack*. URL: <https://santandersmartbank.es/talent-scanner/tecnologia-y-data/full-stack-developer>.