



Creación e implementación de una aplicación web para la administración de citas para un autolavado

Creation and implementation of a web application to manage appointments for a car wash business

Ricardo Luna Santos

Universidad Tecnológica de Xicotepec de Juárez, Puebla, México
ricardo.luna@utxicotepec.edu.mx
ORCID: 0000-0001-8280-6769

Randolfo Alberto Santos Quiroz

Universidad Tecnológica de Xicotepec de Juárez, Puebla, México
randolfo.santos@utxicotepec.edu.mx
ORCID: 0000-0002-6333-4193

Edgar Carrera Carrasco

Universidad Tecnológica de Xicotepec de Juárez, Puebla, México
edgarcarrasco0710@gmail.com

Iván González Domínguez

Universidad Tecnológica de Xicotepec de Juárez, Puebla, México
rei0406@hotmail.com

María Gabriela Santaella Benavente

Universidad Tecnológica de Xicotepec de Juárez, Puebla, México
mariagabriela.santaella@utxicotepec.edu.mx

doi: <https://doi.org/10.36825/RITI.09.19.008>

Recibido: Agosto 19, 2021
Aceptado: Noviembre 10, 2021

Resumen: De acuerdo al comunicado de prensa emitido por INEGI el 2 de diciembre de 2020, 93.2% de las empresas en México reportaron afectaciones a causa de la pandemia y poco más de un millón de establecimientos cerraron sus puertas definitivamente. Dentro de este contexto surge la necesidad de crear herramientas tecnológicas que impulsen el desarrollo de las actividades de las micro, pequeñas y medianas empresas. En el presente proyecto se da a conocer el desarrollo y creación de una aplicación de autolavado que permitirá a los clientes y empleados del negocio agendar, administrar, controlar citas y pagos de servicios ofrecidos. El desarrollo de este proyecto se llevó a cabo en el marco de trabajo SCRUM, se utilizaron diferentes herramientas de desarrollo colaborativo como ASP.Net para las aplicaciones web, Visual Studio como desarrollo de entorno integrado, un sistema de base de

datos relacional y el modelo vista controlador (MVC) como patrón de arquitectura de software. Este módulo permitirá ser adaptado a cualquier tipo de negocio de igual o diferente rubro que tenga las mismas necesidades.

Palabras clave: *Aplicación Web, SCRUM, ASP.Net, MVC, E-Commerce.*

Abstract: According to a press release emitted by INEGI on December 2nd, 2020, 93.2% of the companies in Mexico reported problems caused by the current pandemic, and more than a million businesses closed their doors permanently. In this context, the need to create technological tools that promote the development of activities of micro, small and medium businesses emerged. In this project, the development and creation of an application for a car wash business that allows clients and workers to book and manage appointments and the payment of services are presented. The development of this project was carried out with the framework SCRUM, using different tools for collaborative development such as ASP.Net for web applications, Visual Studio as Integrated Development Environment (IDE), a relational database system, and Model-View-Controller (MVC) as an architectural pattern. This module can be adapted to any kind of business with the same needs in the same field or other fields.

Keywords: *Web Application, SCRUM, ASP.Net, MVC, E-Commerce.*

1. Introducción

En la actualidad, la tecnología es manejada por miles de personas sin importar su edad, género, estatus social, país o religión. Diversas soluciones tecnológicas facilitan la vida y el trabajo de miles de personas alrededor del mundo. Cientos de aplicaciones de escritorio, web y móviles nos ayudan a resolver problemas que enfrentamos en nuestra vida cotidiana. Una vida sin esta tecnología resulta ahora impensable.

Una de las ramas más beneficiadas con estos avances tecnológicos es el comercio. Aplicaciones web y móviles permiten a las empresas grandes, medianas y pequeñas llegar a cientos de posibles clientes con gran velocidad e interactuar directamente con ellos en tiempo real. Esto se traduce en una reducción significativa de costos para las empresas, gran comodidad para los clientes y, por supuesto, mayores ingresos [1].

Sin duda, uno de los mercados con más oportunidades de desarrollo tecnológico y de negocios en México, y posiblemente en el mundo, es el de las empresas o negocios dedicados a los servicios. La fiebre de la tecnología y la necesidad de generar más ingresos y mantenerse vigentes ha orillado no sólo a las grandes empresas, sino también a negocios pequeños y medianos a buscar la forma de atraer clientes a través de la tecnología.

Según la revista de emprendimiento *Entrepreneur*. Muchas personas tienen auto; sin embargo, pocas tienen tiempo suficiente para limpiarlo. En esa línea, quienes quieren tener su vehículo siempre impecable y reluciente deben llevarlo a un lugar especializado y esperar demasiado tiempo a que su coche esté listo [2].

Es por ello que algunos de los proyectos más exitosos en diferentes países es la implementación de servicios de auto lavado a domicilio sin embargo esto hace que el servicio sea más costoso además de que no es posible realizar todos los servicios que se ofrecen en un establecimiento de autolavado por lo cual en este proyecto optamos por brindar una solución más económica a través de un sistema de citas para autolavado. Partiendo del éxito de otros proyectos en el sector automotriz como el caso de sistemas de citas para servicios de la compañía Toyota [3].

El presente documento tiene la finalidad de describir el desarrollo de una aplicación para uno de estos pequeños negocios que buscan adaptarse a los nuevos tiempos, se mencionan las principales características de esta aplicación, los beneficios de su uso, las diferentes etapas del proceso de desarrollo y los resultados esperados.

2. Estado del arte

Con el fin de brindar un sustento al proyecto planteado y validar la definición del problema suscitado se han consultado contenido referente a investigaciones y proyectos relacionados, así como otros medios bibliográficos de esta forma se pretende establecer los beneficios que tendría la implementación del proyecto aquí planteado agregando valor al mismo además de permitir clarificar el concepto a través de referencias específicas.

El primer documento citado se titula, *lavAutos: Un Aplicativo Web para la gestión del servicio de lavado de vehículos en Bogotá para la empresa Smart CarWash*, Almesiga *et al.* [4] plantean la importancia de los autolavados en dicha ciudad y la gran demanda que estos tienen a la hora de requerir limpieza y mediante estudios

realizados utilizando herramienta como la entrevista y la encuesta a los clientes y trabajadores, expresan la necesidad que se tiene de controlar y optimizar al máximo el tiempo de los servicios; es por ello que determinan como objetivo principal la creación de una aplicación web para la administración, gestión, comunicación y acceso a los servicios de aseo y limpieza de vehículos en dicho establecimiento; cuyo enfoque de la investigación es de tipo cualitativo.

Continuando con lo mencionado previamente en cuanto a la importancia de las tecnologías en los pequeños y medianos negocios citamos el siguiente documento llamado, Campaña de Marketing Digital Autolavado Nuevo Gaitán la78, programa avanzado para la formación empresarial con énfasis en estrategias digitales, también realizado en la ciudad de Bogotá, donde Niño, Ordoñez y Silva [5] plantean que al incrementar sus ventas mediante el uso de estrategias digitales y tener más alcance en la población, todo esto lo van a ir midiendo por el alcance en redes sociales, número de búsquedas en Google, experiencia del usuario mediante los anuncios, visita en la página web y citas agendadas en dicha página. Establecen como objetivos principales la reducción del 50% en el uso de agua, mantener satisfechos a los clientes actuales y brindar una excelente calidad de servicio, además de incrementar en un 30% las ventas en un lapso no mayor a 2 meses.

Siguiendo la línea de estas ideas innovadoras en auto lavados ecológicos en la documentación de tesis maestral titulada, plan de negocios de una empresa de lavado de autos enfocada al cuidado del medio ambiente mediante la reutilización del agua y la inclusión social como factores diferenciadores, desarrollada en la ciudad de Lima, Perú. González y Miranda [6] manifiestan la necesidad de hacer algo nuevo y amigable con el medio ambiente, como el reciclaje del agua y el tratamiento especial, eso va a atraer a más personas interesadas en mejorar el planeta. Anexando a esta propuesta el registro de citas online por redes sociales y una sala de espera confort. La forma de medir su progreso va a ser mediante comentarios y publicaciones en redes sociales.

Si analizamos las propuestas anteriores se observa que la mayoría de los proyectos planteados son solo para página web no responsiva y como tal no funciona correctamente en dispositivos móviles o teléfonos inteligentes. Las mejoras que tiene nuestro proyecto es el uso de una aplicación que funcionara en todos los dispositivos móviles, mejorando la interfaz utilizando diseño responsivo e intuitivo y de esta forma agilizar el flujo de los clientes, crear un registro el cual proporcionara un usuario y contraseña, la *app* permite agendar su cita y todas las actividades quedaran almacenadas en una base de datos. Estamos convencidos que mediante el uso de las tecnologías de la información y el marco de trabajo SCRUM como metodología de desarrollo ágil se lograra una mayor organización con los desarrolladores ya que se proponen entregas parciales para brindar resultados a corto plazo, la metodología propuesta permite que el equipo de trabajo pueda generar innovación, competitividad, flexibilidad y productividad ante este mundo globalizado.

3. Materiales y métodos

Para la realización del proyecto fue necesario establecer una metodología la cual nos permitiese llevar a cabo un desarrollo óptimo del sistema en cuestión. Tomando en cuenta el alcance del proyecto y las partes que lo conforman se decidió utilizar la metodología SCRUM, ver Figura 1, la cual se puede definir como un marco de trabajo liviano que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos [7].

Esta metodología ágil permite la implementación de un trabajo colaborativo de tal forma que los roles involucrados tienen un papel definido y tareas específicas que permite establecer los procesos de manera iterativa y así se puede ver un avance constante mediante las reuniones semanales denominadas como *sprint*. Así mismo nos establecer metas específicas alcanzables dentro del proyecto e identificar las fallas mediante un proceso de retroalimentación continua [8, 9].

3.1. Metodología SCRUM

Las principales ventajas que esta metodología ofrece a nuestro proyecto son las siguientes [10]:

- Es útil para el desarrollo de proyectos complejos y que requieren de resultados de forma inmediata.
- Permite abordar requisitos cambiantes y poco definidos.
- Permite localizar problemas y solucionarlos rápidamente.
- Permite retroalimentación continua, eliminación de esperas, reducción de tiempos de entrega y la reutilización de componentes.

- Produce productos de alta calidad.
- Propicia un ambiente innovador y colectivo entre los desarrolladores.



Figura 1. Estructura Scrum.

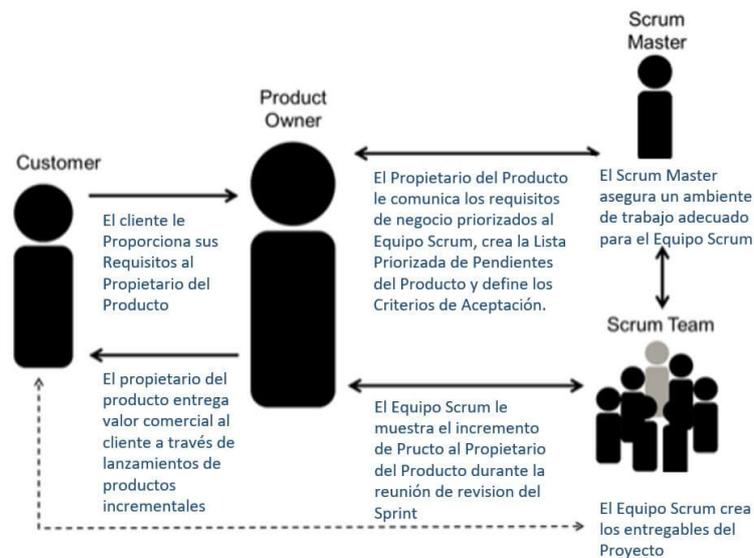


Figura 2. Equipo Scrum.

Esta metodología permite a los diferentes participantes de nuestro proyecto tener roles definidos y conocer sus responsabilidades dentro de dicho proyecto, ver Figura 2. Los diferentes roles descritos dentro de esta metodología son los siguientes:

- **Product owner.** Este rol le pertenece a nuestro cliente, se asegura que el equipo trabaje de forma adecuada desde la perspectiva del negocio. Representa a los accionistas y clientes que usan el *software*. Se focaliza en la parte de negocio y es responsable del ROI (Retorno de Inversión). Traslada la visión del proyecto al equipo.

- **ScrumMaster.** Dirige al equipo para que cumpla las reglas y procesos de la metodología, al mismo tiempo la persona encargada de llevar a cabo este rol gestiona la reducción de impedimentos del proyecto y trabaja con el *Product Owner* para maximizar el ROI.
- **Team.** Se refiere principalmente al grupo de profesionales con los conocimientos técnicos necesarios y que desarrollan el proyecto de manera conjunta. En este caso, los desarrolladores son los encargados de conformar el *Team* de este proyecto.

3.2. Roles en SCRUM

Es necesario entender las diferentes etapas que permiten el desarrollo y flujo de trabajo en este *framework*. Normalmente el desarrollo se conforma de los siguientes componentes [11]:

- **Product Backlog.** Se compone de los requerimientos del proyecto, el cliente o dueño del producto (*Product owner*) define las prioridades de cada requisito, las descripciones son generales y pueden estar sujetas a cambios en el transcurso del desarrollo del proyecto.
- **Sprint Backlog.** Cada Sprint es un conjunto de tareas a realizarse, las cuales son extraídas del *Product backlog*. Dichas actividades deberán ser ejecutadas entre una semana como mínimo y cuatro semanas como máximo. Las tareas realizadas cumplirán con los requerimientos solicitados por el *Product owner*.
- **Ejecución del Sprint.** Periodo en el que el equipo de desarrollo (*Scrum team*) realizará las tareas establecidas en el *Sprint Backlog*. Cabe mencionar que una vez que se inicia la ejecución del Sprint, no podrá ser modificado hasta que haya concluido.
- **Entrega o incremento.** Es una entrega funcional del proyecto, la cual se realiza una vez concluida la ejecución del sprint. Cada entrega o incremento es revisada por el *Product owner*, el cual evalúa las funcionalidades entregadas y retroalimenta al equipo de trabajo (*Scrum team*). El equipo hará las correcciones necesarias en la siguiente ejecución del *Sprint*.

Este proceso se repite hasta alcanzar la finalización del proyecto.

3.3. Definición de la arquitectura

En este caso en particular con el fin de brindar una mayor calidad en el desarrollo y siguiendo con las prácticas para el desarrollo híbrido se ha decidido separar el proyecto en dos partes las cuales denominaremos como *backend* y *frontend*. De esta forma podremos implementar una migración de este proyecto en diferentes *frameworks* y tecnologías partiendo todos desde una lógica de negocios funcional a través del uso de micro servicios mediante un api *Restful* [12].

Con la interacción de estos servicios se buscan dos cosas la primer es separar la lógica de negocio de tal forma que se facilite la posibilidad de un crecimiento futuro con la implementación en multiplataformas la segunda es establecer un flujo claro en la información como se muestra en la Figura 3 de esta forma podemos identificar de una manera más fácil los errores que surjan y principalmente en la etapa de producción del proyecto se pueden realizar los cambios en los diferentes componentes sin necesidad de detener que detener el funcionamiento del resto de los módulos.

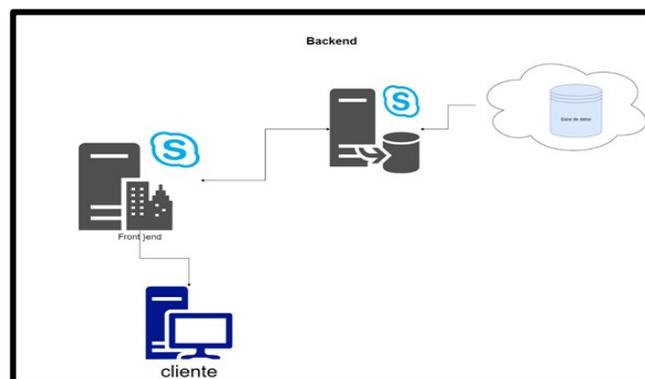


Figura 3. Arquitectura del proyecto. Fuente: Elaboración propia.

3.3.1. Backend

Conforme a ello establecimos 3 aspectos en el modelo de interacción como se puede apreciar en la Figura 4 en los cuales se definen las capas para la representación del *backend* conforme a la lógica de negocios.

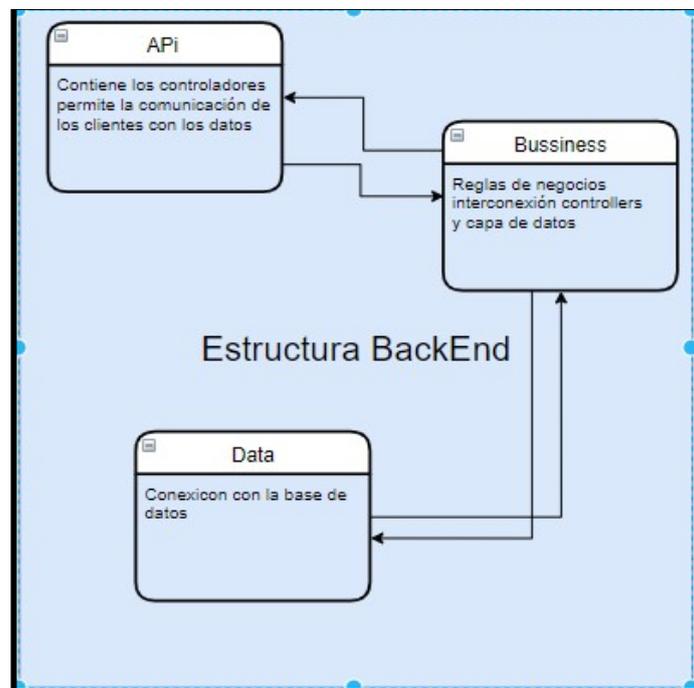


Figura 4. Capas *backend*. Fuente: Elaboración propia.

- **Data.**

La capa de Data albergará la conexión de datos y la separación en las diferentes agrupaciones para el consumo de los procedimientos almacenados en la base de datos. Para lo cual se deberá crear una clase denominada *DataRepository* que permitirá una conexión única a la base de datos y será inyectada en las diferentes clases de la capa de negocio.

- **Business.**

En la capa de *Business* se establecerán las reglas para la consulta e implementación de datos por ejemplo si tenemos una regla de negocio que nos indica que al ser creado un usuario se enviará un correo de confirmación, al revisar la bandeja del cliente registrado se observara el correo donde se informa que su registro fue exitoso.

- **Presentación.**

En esta capa mostraremos la información al cliente, recuperada de la capa de negocios, y la capa de datos.

3.3.2. Frontend

Para el desarrollo del *frontend* utiliza la arquitectura de software conocida como Modelo Vista Controlador (MVC). La arquitectura MVC consiste en un patrón de diseño que separa los datos, la metodología y la interfaz gráfica de una aplicación. La ventaja principal de esta técnica de programación es que permite modificar cada una de estas partes sin tener que modificar las demás. Esto permite desarrollar aplicaciones modulares y escalables.

Los tres componentes de esta arquitectura son como se presenta en la Figura 5:

- **Modelo:** Se encarga de manipular, gestionar y actualizar los datos.
- **Vista:** Se encarga de mostrarle al usuario final las pantallas o formularios como resultado de las solicitudes enviadas a través del controlador.
- **Controlador:** Es el componente principal de la aplicación. Aquí se especifican los métodos y funcionalidades de la aplicación. Se encarga de gestionar las instrucciones que se reciben, atenderlas y procesarlas para finalmente mostrar los datos obtenidos en la vista [12].

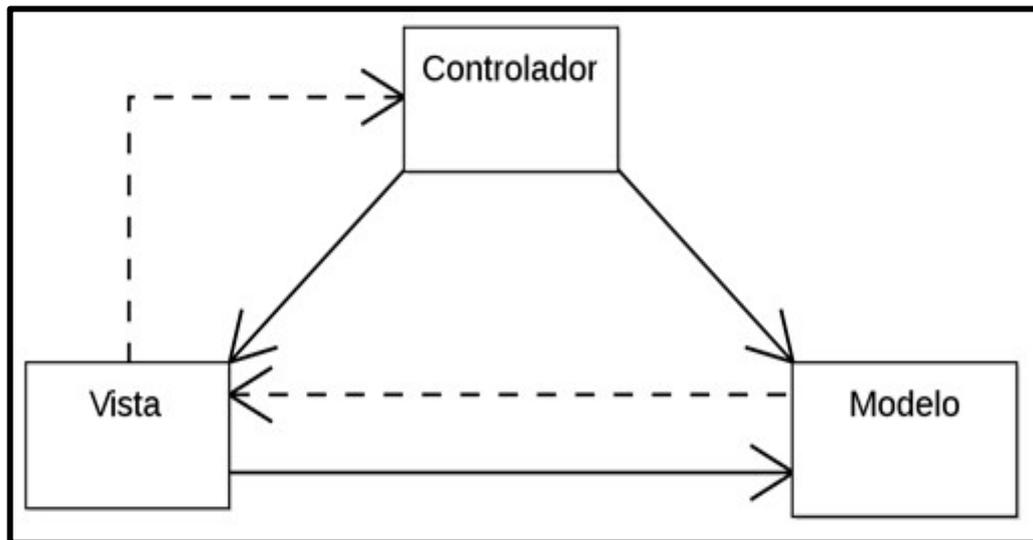


Figura 5. Modelo Vista Controlador.

3.4. Selección de Tecnologías

Con la finalidad de obtener un desarrollo óptimo se optó por implementar dos tecnologías bastantes robustas y capaces de adaptarse a la convivencia con otras tecnologías sin necesidad de implementación extra a las configuraciones iniciales del proyecto.

La primera de ellas es ASP.NET Core utilizando el lenguaje de programación C# dicha tecnología fue elegida por ser bastante robusta además de contar con implementación de seguridad bastante sencillas y adaptables a un entorno corporativo.

Además de ser una tecnología ligera y con una extensa comunidad de desarrollo que la respalda además de una extensa cantidad de librerías capaces de adaptarse a situaciones específicas de una forma bastante simple. En cuanto a el entorno de desarrollo para la parte web del proyecto se implementará ASP.Net *framework* MVC 4 en conjunto con Javascript y la librería JQuery para realizar módulos dinámicos que respondan a la interacción de los usuarios ya que como menciona Tamayo [13].

ASP.NET MVC implica un cambio radical para los desarrolladores web que utilizan tecnología .NET debido a que su diseño se basa en una arquitectura limpia, fundamentada en la inclusión de patrones de diseño para agilizar el desarrollo y en la aplicación de pruebas unitarias para garantizar la calidad del software. jQuery y ASP.NET MVC pueden combinarse de manera natural para desarrollar aplicaciones de alta calidad. Al tener un código más limpio y con mayor calidad reducimos los riesgos en la implementación del proyecto.

3.5. Entorno de desarrollo

Como en todo desarrollo de *software* es necesario establecer las herramientas adecuadas tanto a la metodología como a las tecnologías que componen al proyecto. Comenzando por aquellas herramientas que nos ayudarán a tener una mejor gestión de las etapas tales como:

Notion la cual es una herramienta óptima para el desarrollo de proyectos en los cuales podemos establecer las fases correspondientes a las etapas del desarrollo, así como tener un control de las actividades.

Otra de las herramientas necesarias en todo proyecto es el control de versiones por lo que se decidió el uso de GitHub una herramienta que permite llevar a cabo proyectos colaborativos de diferentes tecnologías trabajando directamente con Git de esta forma se puede tener un control total tanto de las versiones y ramificaciones necesarias para el proyecto además que puede ser fácilmente adaptado a diferentes entornos de desarrollo e IDE's tales como Atom, Visual Studio, Visual Studio Code, entre otras.

En cuanto al IDE de desarrollo se estableció que la mejor herramienta en este caso tomando en cuenta las características propias de la tecnología y las recomendaciones oficiales de las mismas era el uso de Visual Studio 2019 en su versión *community*.

3.6. Modelo de datos

Para el realizar el manejo de datos se implementó una estructura relacional de datos de tal forma que establezcamos las relaciones entre los diferentes elementos que conforman las reglas de negocio además este tipo de bases de datos ayudan a mantener una mayor fiabilidad de la información permitiendo la generación de recursos estadísticos para el análisis del comportamiento del sistema en un futuro. Partiendo de ello se seleccionó como motor de datos MySQL en la versión 8.0 esto principalmente por tener una arquitectura más ligera y una mayor compatibilidad en un entorno de desarrollo lo cual nos permite obtener un mayor desempeño en cuanto a recursos de hardware se refiere.

El modelo de datos relacional establece las siguientes entidades:

- Auto
- Cliente
- Usuario
- Tipo de usuario
- Servicios (paquetes)

Además de ello se cuentan con las siguientes relaciones:

- Un cliente puede tener muchos autos
- Un auto puede tener múltiples servicios y a su vez un servicio puede tener relaciones con muchos autos
 - Un usuario tiene un rol
- Un usuario puede registrar múltiples servicios
- Un usuario puede atender múltiples citas

La estructura de la base de datos establece bajo un conjunto de normas en las que las tablas principales se pretenden por “c_” mientras que las tablas relacionales que satisfacen una relación de muchos a muchos se precede por “r_” de igual forma las columnas se preceden por las iniciales de las tablas tal como se muestra en la Figura 6.

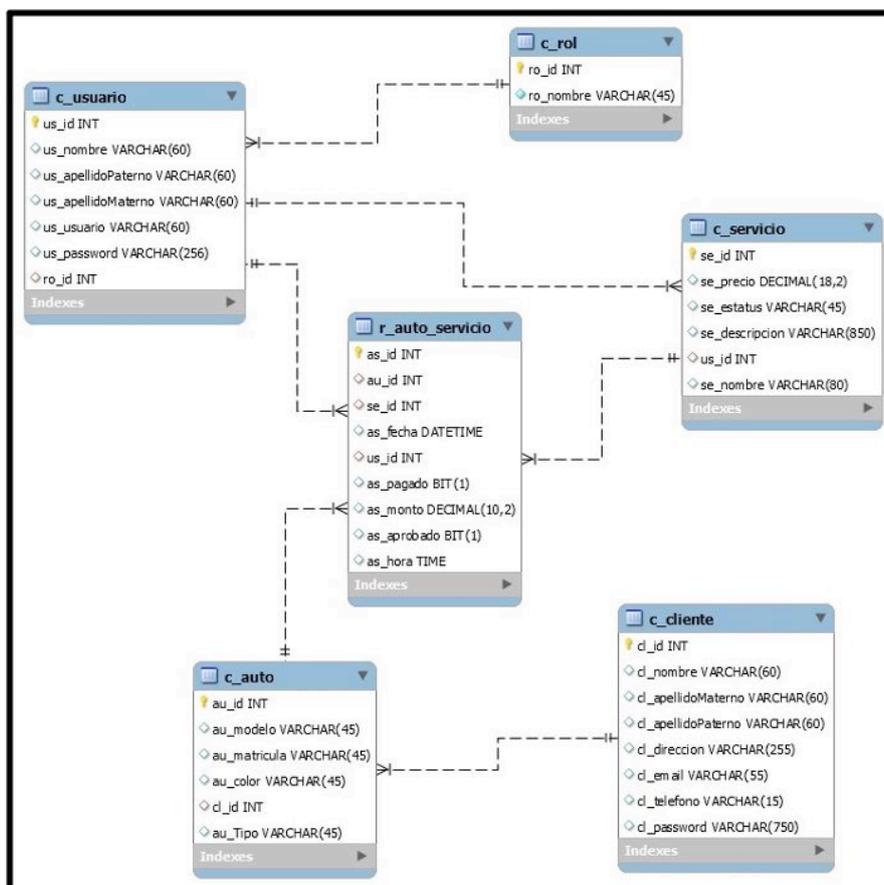


Figura 6. Diagrama ER. Fuente: Elaboración propia.

4. Resultados

El desarrollo de este proyecto has sido sumamente gratificante debido a que todo surgió como una idea para cubrir una necesidad y poco a poco fue creciendo hasta convertirse en un producto tangible. Muchas cosas han cambiado a lo largo del desarrollo. El plan de negocios se ha modificado en repetidas ocasiones, mejorando cada vez más. Las funcionalidades del sitio web, el diseño e incluso el lenguaje de programación cambiaron en repetidas ocasiones. Sin embargo, poco a poco, la idea inicial fue tomando forma, convirtiéndose en sistema altamente funcional basado en el modelo de negocios que se muestra a continuación en la Figura 7.

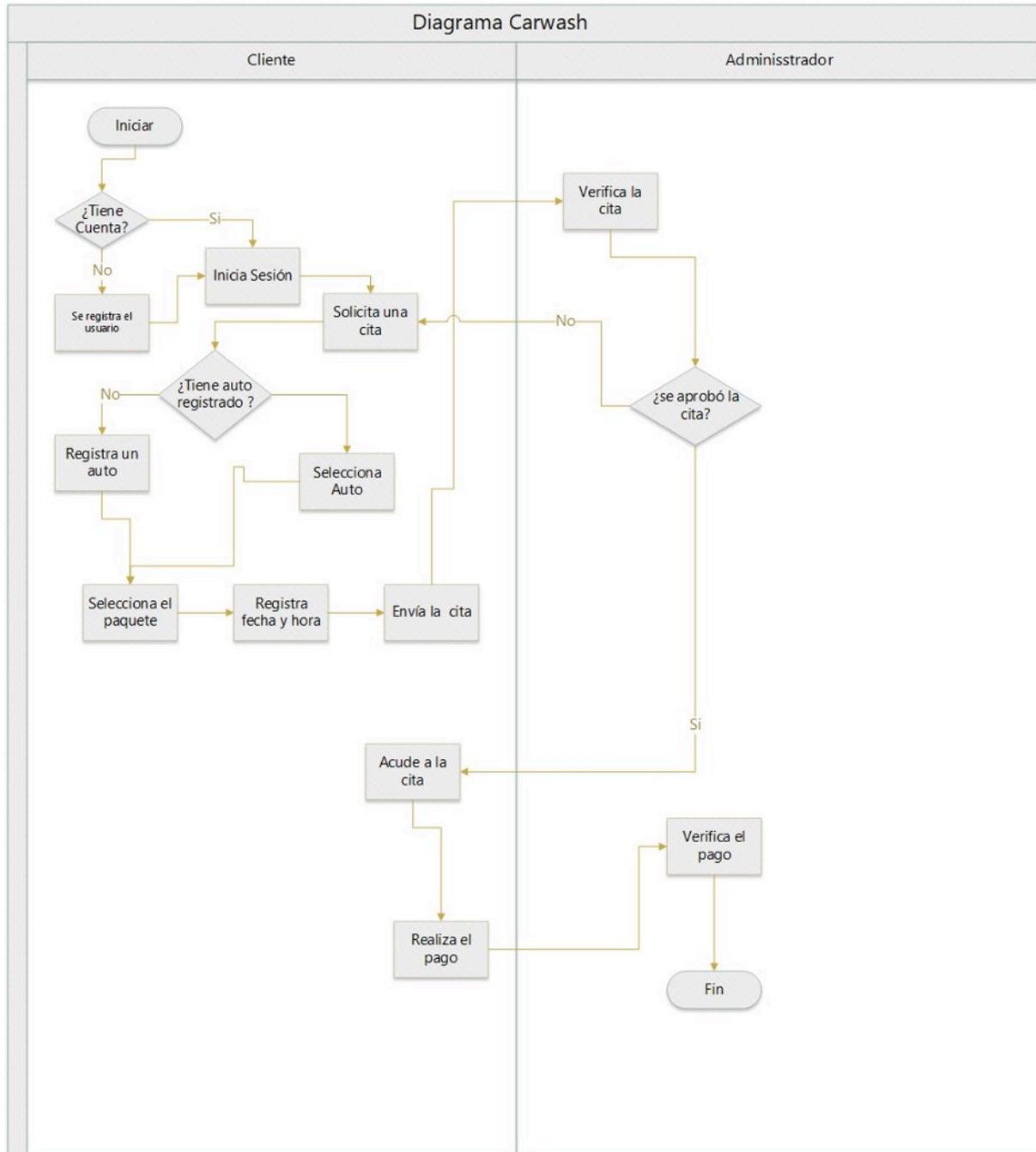


Figura 7. Flujo de Negocios. Fuente: Elaboración propia.

Como resultado, tenemos una aplicación web que cumple con las metas y objetivos del proyecto. Algunas características importantes obtenidas al concluir esta aplicación web son las siguientes:

1. Diseño atractivo. Varios aspectos del diseño se cuidaron durante el maquetado del sitio web. Se cuidó el tamaño de textos, botones, colores, imágenes, íconos, etc. Sin embargo, el enfoque siempre estuvo en la funcionalidad del sitio. En una segunda etapa se realizará un trabajo más profundo con el diseño del sitio en donde se buscará de colaboración de un diseñador gráfico para desarrollar la imagen final del sitio. A continuación, se muestra la interfaz de inicio de la aplicación en la Figura 8.

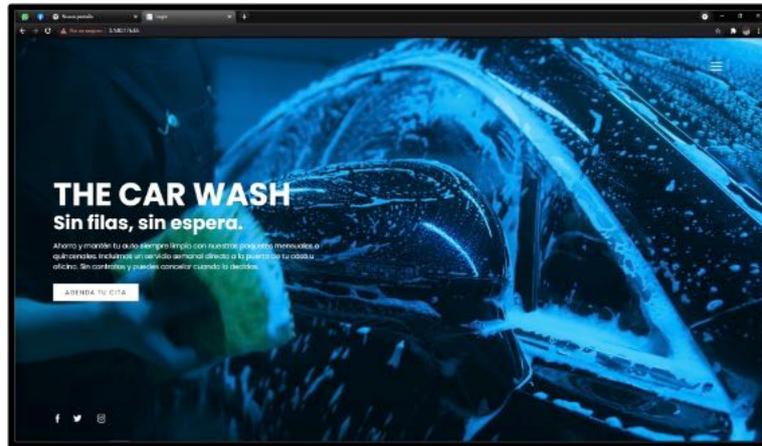


Figura 8. Interfaz de inicio. Fuente: Elaboración propia.

2. Interfaz limpia. Se busco crear un sitio minimalista, limpio, pero altamente funcional. Se eliminó texto, íconos, imágenes o cualquier otro elemento innecesario. Se diseñaron formularios simples, pero visualmente agradables que cumplen con las funciones CRUD de forma eficiente. En la Figura 9, se muestra el *Login* para ingresar a la administración de las citas. En la Figura 10, se muestra la interfaz de administración de citas.
3. Navegación intuitiva. Se diseñó una navegación lineal, secuenciada, es decir, una página lleva a otra, y esta a su vez a otra y así sucesivamente, lo cual facilita la navegación a través de las distintas vistas y formularios.

Figura 9. Login para la interfaz de administración. Fuente: Elaboración propia.

Tipo de Auto	Email Propietario	Fecha	Servicio	Aprobado	Pagado	Total pago	Opciones
4x4	edgarcarreras0710@gmail.com	23-AGOSTO-2021 13:08	Lacado Premium	Pendiente	Por Pagar	90	☰
4x4	edgarcarreras0710@gmail.com	11-AGOSTO-2021 00:08	Lacado Premium	Pendiente	Por Pagar	90	☰
4x4	edgarcarreras0710@gmail.com	15-JULIO-2021 14:07	Lacado Premium	Aprobado	Pagado	70	☰
4x4	edgarcarreras0710@gmail.com	30-AGOSTO-2021 10:08	Express	Pendiente	Por Pagar	80	☰
Deportivo	edgarcarreras0710@gmail.com	05-JULIO-2021 16:07	Express	Aprobado	Pagado	145	☰
Sedan	re0406@hotmail.com	01-AGOSTO-2021 10:08	Lacado Premium	Pendiente	Por Pagar	80	☰
SUV	edgar1@gmail.com	INVALID DATE	Lacado Premium	Aprobado	Por Pagar	145	☰
SUV	edgar1@gmail.com	07-JULIO-2021 13:07	Lacado Premium	Aprobado	Pagado	145	☰
Suv	edker071@gmail.com	22-JULIO-2021 14:07	Lacado Premium	Aprobado	Pagado	70	☰
Suv	edker071@gmail.com	22-JULIO-2021 14:07	Lacado Premium	Aprobado	Pagado	0	☰

Figura 10. Interfaz de administración. Autoría propia. Fuente: Elaboración propia.

4. Funciones CRUD y conexiones a base de datos rápidas. La función principal de este sitio es el almacenamiento de información del cliente, información de su automóvil, las citas registradas. La consulta, administración posterior de esta información en la base de datos forma parte de la funcionalidad. Por este motivo, se trabajó mucho en las conexiones adecuadas a la base de datos a través de modelos y controladores. Además, se hizo la selección de un servidor basado en Windows server 2019, ejecutamos el servidor web IIS, servidor de base de datos SQL Server para mejorar los tiempos de respuesta. El uso del modelo MVC también ayuda a controlar los recursos del servidor, debido a que en ocasiones es posible olvidar cerrar conexiones a la base de datos, sobrecargando el servidor. Esto no sucede con el modelo MVC.
5. Seguridad. Debido a que la información del cliente requiere de mucha seguridad. La utilización de un modelo MVC nos permite ocultar la información de conexión a la base de datos. Además, con la utilización de encriptación podemos proteger información sensible del negocio.
6. Escalabilidad. La metodología de programación SCRUM, el modelo vista controlador (MVC) y la utilización de APIs nos permite escalar este proyecto fácilmente. Se pueden crear nuevos módulos que se acoplen perfectamente con los módulos ya existentes. Además, gracias a los servicios web implementados en el proyecto, es posible crear aplicaciones del lado del *frontend* para que funciones en dispositivos móviles, computadoras de escritorio y en múltiples sistemas operativos utilizando el trabajo previamente hecho. A continuación, se muestra la interfaz de pruebas de microservicios en la Figura 11.

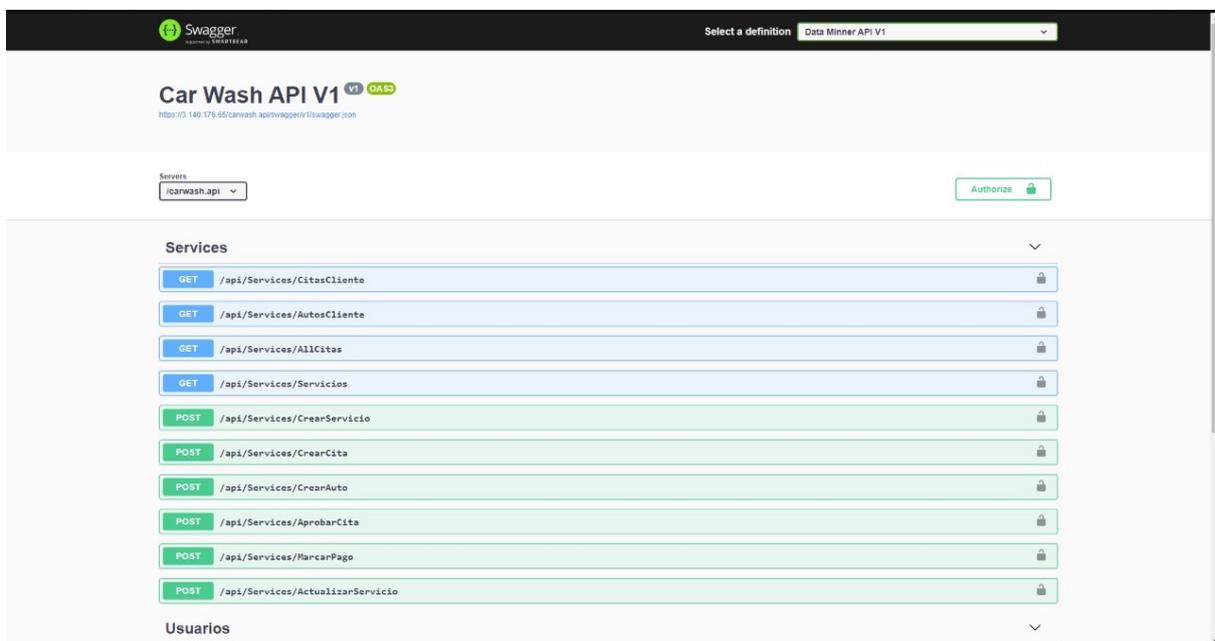


Figura 11. Interfaz de microservicios. Autoría propia. Fuente: Elaboración propia.

Con respecto a áreas de oportunidad presentes en este proyecto, existe el plan de desarrollar una aplicación móvil para los clientes. Esta aplicación servirá para realizar sus citas de servicio y recibir notificaciones en tiempo real. El desarrollo podría llevarse a cabo con el *framework* Flutter, se podría usar nuevamente el modelo MVC y se reutilizaría las APIs desarrolladas anteriormente.

5. Conclusiones

Afortunadamente la mayor parte de los objetivos fueron alcanzados, a excepción de un módulo de pago que no pudo ser desarrollado en esta etapa del proyecto pero que se pretende desarrollar en el futuro. Sin embargo, el resto de los objetivos parciales y objetivos generales fueron cumplidos en su totalidad.

El desarrollo de esta aplicación siguió el curso planeado, la utilización de la metodología SCRUM fue muy útil para mantener al equipo enfocado en las diferentes tareas necesarias durante el desarrollo. Los módulos desarrollados son funcionales, pasaron exitosamente las diferentes pruebas a las que fueron sometidos y se encuentran integrados adecuadamente entre ellos.

Sin duda, la aplicación web desarrollada representará una herramienta muy importante para el incremento de clientes del negocio familiar de auto lavado. La implementación de esta aplicación también nos permitirá evaluar su aceptación y uso por parte de los clientes e identificar nuevos requerimientos para realizar mejoras en el futuro.

De igual manera, es posible implementar esta aplicación para otros tipos de negocios realizando cambios o adaptaciones menores. Esto representa una oportunidad importante para ayudar a muchos negocios locales que atraviesan una difícil situación durante de la pandemia.

Como área de oportunidad frente a las plataformas y sistemas ya elaborados se menciona que nuestro software mantiene una gran escalabilidad gracias a la tecnología de desarrollo basado en la web además incorpora la creación de servicios e implementación de Apis, se utiliza la tecnología de cómputo en la nube haciendo uso de los Servicios Web de Amazon (AWS) que garantizan el funcionamiento, estabilidad y disponibilidad de la información.

6. Referencias

- [1] Decaro Santiago, L. G., Soriano Hernández, M. G., Ocaña Delgado, R. (2017). Los modelos de negocios y las tecnologías de la información y comunicación. *Tlatemoani*, 8, (26), 170-188. Recuperado de: <https://www.eumed.net/rev/tlatemoani/26/modelos-negocios.pdf>
- [2] Sánchez, M. (2017). *Lavado de autos a domicilio*. *Entrepreneur*. Recuperado de: <https://www.entrepreneur.com/article/291454>
- [3] Navarro Quiroz, R., Tacle Delgado, V., Villon, G., Freire, L. (2009). *Proyecto de Sistemas de Citas de la Empresa Toyota S.A., Toyota*. Recuperado de: https://www.lareferencia.info/vufind/Record/EC_0e08e2bd26ed843fec8e086ce16eb61e
- [4] Almesiga Riaño, D. A., Bocachica Peñuela, J. A., Rodríguez Castro, J. S., Sánchez Arteta, J. S. (2020). *LavAutos: Un Aplicativo Web para la gestión del servicio de lavado de vehículos en Bogotá para la empresa Smart CarWash* (Tesis de Licenciatura). Facultad de Ingeniería, Bogotá, Colombia. Recuperado de: https://repositoriocirai.ucompensar.edu.co/bitstream/handle/compensar/800/Documento%20LavAutos%20Un%20Aplicativo%20Web%20para%20la_Grupo%20Investigacin%20I.pdf?sequence=1&isAllowed=y
- [5] Niño Leguizamón, Y. A., Ordoñez Garzon, I. D., Silva Granados, D. L. (2020). *Campaña de Marketing Digital Auto lavado Nuevo Gaitán la78* (Tesis de Licenciatura). Universidad del Rosario, Bogotá. Recuperado de: <https://repository.urosario.edu.co/handle/10336/30665>
- [6] González Mestaza, M. C., Miranda Huerta, O. G. (2018). *Plan de negocios de una empresa de lavado de autos enfocada en el medio ambiente mediante la reutilización del agua y la inclusión social como factores diferenciadores* (Tesis de Maestría). Universidad San Ignacio de Loyola, Lima, Perú. Recuperado de: http://repositorio.usil.edu.pe/bitstream/USIL/2995/1/2018_Gonzales_Empresa-de-lavado-de-autos.pdf
- [7] Francia Huambachano, J. (2017). *¿Qué es Scrum?*. Recuperado de: <https://www.scrum.org/resources/blog/que-es-scrum>
- [8] Proyectosagiles.org. (2021). *¿Qué es SCRUM?*. Recuperado de: <https://proyectosagiles.org/que-es-scrum/>
- [9] Canive, T. (2020). *Metodología SCRUM: ¿qué es y cómo aplicarlo en tu trabajo?*. Recuperado de: <https://www.sinnaps.com/blog-gestion-proyectos/metodologia-scrum>
- [10] Schwaber, K., Sutherland, J. (2016). *La Guía de Scrum. La Guía Definitiva de Scrum: Las Reglas del Juego*. Recuperado de: <https://scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Spanish.pdf#zoom=100>
- [11] Master, G. D. P. (2021). *Roles y Responsabilidades en un Proyecto Scrum. Gestión de Proyectos Master*. Recuperado de: <https://www.gestiondeproyectos-master.com/roles-y-responsabilidades-en-un-proyecto-scrum/>
- [12] Marcano, A. (2020). *ASP.NET CORE 3.1 MVC Y WEB API: Ejemplos y Ejercicios en C# (Spanish Edition)*. Independently published.
- [13] Tamayo, A., Katrib, M. (2021). ASP.NET MVC y jQuery. Más poder, más control, ¿más esfuerzo? *DotNetManía*, 12-19.