

# AquaMóvil

## ¡Un spa para tu auto!

*Sistema de Gestión de Servicios de Lavado de Autos a Domicilio*

### **Equipo de Proyecto AquaMóvil**

#### **Autores:**

Fernández, Ariel

Luppi, Astrid

Picone, Abigail

Pirles, Irina

Pucheta, Mauricio

Villanueva, Marcela

### **Instituto Superior Politécnico de Córdoba**

**Tecnicatura Superior en Desarrollo Web y Aplicaciones Digitales**

**Cohorte:** 2025

**Materia:** Proyecto Integrador I

**Docente:** Nievas Guardia, Verónica Elena

# Documento de Requisitos de Software

## Resumen

Este trabajo presenta el desarrollo de AquaMóvil, un sistema de software orientado a objetos que busca optimizar la gestión de servicios de lavado de autos a domicilio. La motivación principal surge de la necesidad de digitalizar y automatizar procesos en emprendimientos de servicios móviles, permitiendo mejorar la organización, asignación de pedidos y trazabilidad de pagos.

El sistema implementa un modelo de usuarios con roles diferenciados (administrador, usuario cliente y operario), gestión de pedidos con múltiples estados, integración de medios de pago y generación automática de liquidaciones a operarios.

Se diseñó a partir de un enfoque incremental, empleando diagramas de clases, modelos entidad-relación y metodologías de registro de requerimientos. Como resultado, se obtuvo un software de consola funcional que cumple con las consignas académicas y ofrece proyección hacia un modelo de software escalable.

## Introducción

En los últimos años, la digitalización de servicios tradicionales ha transformado los modelos de negocio. El lavado de autos a domicilio, antes gestionado de forma manual, presenta una oportunidad de negocio en la administración eficiente de pedidos, control de pagos y asignación de recursos humanos.

AquaMóvil surge como una propuesta innovadora dentro de la Tecnicatura Superior en Desarrollo Web y Aplicaciones Digitales, vinculando los contenidos de las materias: **Módulo Full Stack, Gestión de Contenidos y Proyecto Integrador I.**

El objetivo principal es diseñar un sistema que gestione usuarios, pedidos y pagos de manera automatizada, mejorando la eficiencia operativa y sentando bases para su escalabilidad en un modelo de plataforma digital.

## Situación Problemática

Los emprendimientos de servicios de lavado de autos a domicilio carecen de herramientas tecnológicas que les permitan:

- Registrar usuarios con distintos roles y permisos.
- Administrar pedidos con trazabilidad clara de estados.
- Gestionar pagos electrónicos y liquidaciones de operarios.
- Organizar la información de clientes de manera centralizada.

AquaMóvil cubre esta demanda mediante un sistema que digitaliza procesos clave, ofreciendo mayor transparencia, seguridad de datos y proyección hacia la economía digital.

## **Desarrollo**

### **4.1 Actividades realizadas**

- Relevamiento de requerimientos funcionales y no funcionales.
- Diseño de diagrama entidad-relación y diagrama de clases.
- Desarrollo de un sistema de consola en Python con enfoque orientado a objetos.
- Implementación de registro e inicio de sesión con validación de roles.
- Construcción de un menú dinámico según rol del usuario.

### **4.2 Tecnologías utilizadas**

- **Lenguaje:** Python
- **Base de datos:** MySQL
- **Modelado:** UML (diagrama de clases y casos de uso)
- **Control de versiones:** Git, GitHub y Git Bash
- **Entorno de desarrollo:** Visual Studio Code

### **4.3 Diseño y recolección de datos**

- Identificación de entidades: usuarios, pedidos, pagos.
- Definición de atributos clave: nombre, email, DNI, estado, rol, etc.
- Normalización de la base de datos hasta 3FN.

### **4.4 Métodos empleados**

- Metodologías ágiles Scrum (Kanban).
- Programación orientada a objetos.
- Metodología incremental y modular.
- Aplicación de estándares IEEE 830 para documentación.

## **Planificación y Ejecución del Proyecto**

El proyecto se dividió en fases:

1. **Fase de análisis:** identificación de problema y definición de requerimientos.
2. **Fase de diseño:** modelado de clases y entidades.
3. **Fase de desarrollo:** implementación de funcionalidades básicas.
4. **Fase de pruebas:** verificación de usuarios.
5. **Fase de documentación:** redacción de informe.

## **Resultados Obtenidos**

- Registro exitoso de usuarios con roles diferenciados.
- Interfaz de consola simple, validada en entorno académico.

## **Conclusiones**

AquaMóvil constituye una respuesta concreta a la necesidad de digitalización en servicios móviles. El sistema desarrollado cumple con los requerimientos planteados y se ajusta a los contenidos curriculares de la carrera.

Se comprobó que el modelo orientado a objetos permite escalabilidad, y la integración de una base de datos relacional asegura consistencia en la gestión de información. El trabajo demuestra la viabilidad de implementar un software académico con proyección hacia distintos segmentos de clientes del mercado.

## **Proyecciones posibles**

- Evolución hacia una aplicación móvil y de escritorio.
- Integración con geolocalización y agendas inteligentes.
- Modelo SaaS para permitir que terceros utilicen la plataforma como franquicia digital.

## **Viabilidad / Presupuesto**

- **Viabilidad técnica:** Alta, dado el uso de tecnologías accesibles y de código abierto.
- **Viabilidad económica:** Baja inversión inicial, con potencial de escalabilidad.
- **Presupuesto estimado (dado que se desarrolla en un ámbito académico):**
  - Desarrollo: horas de trabajo de los estudiantes.
  - Infraestructura: PC y software libre.
  - Base de datos: MySQL (open source).

#### **Campos de Acción, potenciales clientes**

- Emprendimientos de servicios de lavado de autos a domicilio.
- Pequeñas empresas de servicios móviles.
- Lavaderos de autos con un local físico.
- Emprendedores independientes.

## Ficha del documento

Fecha	Revisión	Autor	Referencia de Cambio
17/08/2025	1	Luppi, Astrid	Se comienza a redactar la sección "Descripción general"
17/08/2025	2	Villanueva, Marcela	Se agrega RF
18/08/2025	3	Fernández, Ariel	Se agrega RNF
18/08/2025	4	Pirles, Irina	Se agrega Introducción. Revisión apartado 3.1, anexos de diagramas
18/08/2025	5	Pucheta, Mauricio	Se agrega sección "Restricciones y supuestos"
19/08/2025	6	Picone, Abigail	Se completa cuadro correspondiente al Sprint 0
20/08/2025	7	Luppi, Astrid	Se mejora "Descripción general"
21/08/2025	8	Picone, Abigail	Se pasa en limpio el documento. Corrección de tipografía y estilo

# AquaMóvil

## ¡Un spa para tu auto!

### *Sistema de Gestión de Servicios de Lavado de Autos a Domicilio*

**Autores:**

Abigail Picone

Ariel Fernández

Astrid Luppi

Irina Pirles

Marcela Villanueva

Mauricio Pucheta

**Instituto Superior Politécnico de Córdoba**

**Tecnicatura Superior en Desarrollo Web y Aplicaciones Digitales**

**Cohorte:** 2025

**Área:** Ciencias Aplicadas a la Tecnología

**Subárea:** Desarrollo de Software / Desarrollo de Sistemas

**Materia:** Proyecto Integrador I

**Docente:** Verónica Elena NIEVAS GUARDIA

## Contenido

<b>Ficha del documento</b>	<b>3</b>
<b>Contenido</b>	<b>4</b>
<b>1. Introducción</b>	<b>6</b>
1.1 Propósito	7
1.2 Alcance	7
1.3 Personal involucrado	7
1.4 Definiciones, acrónimos y abreviaturas	9
1.5 Referencias	10
1.6 Resumen	10
<b>2. Descripción general</b>	<b>11</b>
2.1 Perspectiva del producto	11
2.2 Características de los usuarios	11
<b>2.3 Restricciones</b>	<b>12</b>
<b>2.4 Supuestos</b>	<b>14</b>
<b>3. Requisitos específicos</b>	<b>16</b>
3.1 Requerimientos funcionales	16
3.1.1 Gestión de usuarios	16
3.1.2 Gestión de pedidos	16
3.1.3 Gestión de pagos	16
3.1.4 Gestión de estados y notificaciones	16
3.1.5 Consultas y reportes	17
3.2 Requerimientos no funcionales	17
3.2.1 Rendimiento	17
3.2.2 Usabilidad	18
3.2.3 Seguridad	18
3.2.4 Fiabilidad	19
3.2.5 Mantenibilidad	19
3.2.6 Portabilidad	19
<b>Product Backlog</b>	<b>20</b>
Cliente	20
Empleado lavacoches	21
Administrador	22
<b>Proyecciones</b>	<b>22</b>
<b>Viabilidad / Presupuesto</b>	<b>23</b>
<b>Presupuesto estimado (dado que se desarrolla en un ámbito académico):</b>	<b>23</b>
<b>Campos de Acción</b>	<b>23</b>
<b>Conclusión</b>	<b>24</b>



<b>Sprints.</b>	<b>24</b>
Sprint Backlog	24
<b>Anexo I</b>	<b>27</b>
Diagrama Entidad-Relación	27
Diagrama de Clases	27
Repositorio de GitHub	27
Wiki	27

---

## 1. Introducción

En los últimos años, la digitalización de servicios tradicionales ha transformado los modelos de negocio. El lavado de autos a domicilio, antes gestionado de forma manual, presenta vacancias en la administración eficiente de pedidos, control de pagos y asignación de recursos humanos.

AquaMóvil surge como una propuesta innovadora dentro de la Tecnicatura Superior en Programación y Base de Datos, vinculando contenidos de Programación I, Base de Datos, Análisis de Sistemas y Metodologías de Desarrollo.

Los emprendimientos de servicios de lavado de autos a domicilio carecen de herramientas tecnológicas que les permitan:

- Registrar usuarios con distintos roles y permisos.
- Administrar pedidos con trazabilidad clara de estados.
- Gestionar pagos electrónicos y liquidaciones de operarios.
- Organizar la información de clientes de manera centralizada.

AquaMóvil cubre esta demanda mediante un sistema que digitaliza procesos clave, ofreciendo mayor transparencia, seguridad de datos y proyección hacia la economía digital.

El objetivo principal es diseñar un sistema que gestione usuarios, pedidos y pagos de manera automatizada, mejorando la eficiencia operativa y sentando bases para su escalabilidad en un modelo de plataforma digital (SaaS).

Este documento es una Especificación de Requisitos Software (ERS) para el desarrollo de un sistema de lavado de vehículos a domicilio. Su intención es dejar en claro las bases del funcionamiento del sistema incluyendo requerimientos necesarios para su comprensión.

La estructura de documentación, se basa en las directrices dadas por el estándar IEEE, recomendada para Especificaciones de Requisitos Software, utilizada para garantizar la coherencia y calidad de la documentación.

## 1.1 Propósito

El presente documento tiene como objetivo definir los requisitos funcionales y no funcionales, para el desarrollo de un sistema que resolverá una gran necesidad detectada, de una forma que agilizará la vida de los clientes, brindándoles opciones varias como lo son la carga y gestión de vehículos, registro de clientes y administradores, visualización y cancelación de turnos, procesamiento de pagos, entre otras.

## 1.2 Alcance

El sistema estará orientado de forma que al usuario le resulte sencillo navegar para poder conocer los múltiples servicios ofrecidos, pudiendo así mismo contratar el que le parezca necesario.

Dentro de su alcance, algunas de sus funcionalidades son:

- Gestión de empleados, que pueden aceptar turnos y marcar servicios como finalizados. Todo el sistema funciona por consola y está pensado para ofrecer una experiencia clara, ordenada y útil para clientes y trabajadores.
- Solicitud de servicios de lavado eligiendo tipo de servicio, ubicación, fecha y hora.
- Visualización y cancelación de turnos pendientes.
- Inicio de sesión para acceder al perfil y las funciones del sistema.
- El sistema estará disponible tanto para los clientes como para los administradores, dependiendo sea uno u otro tendrá diferentes tipos de opciones dentro del mismo espacio.

## 1.3 Personal involucrado

Proyecto Integrador  
**Especificación de requisitos de software**

<b>Nombre</b>	Abigail Yazmin Picone Ambrosioni
<b>Rol</b>	Scrum Master
<b>Categoría Profesional</b>	Coordinadora
<b>Responsabilidad</b>	Organización del equipo, tareas y seguimiento
<b>Información de contacto</b>	abigailpicone.ispc@gmail.com

<b>Nombre</b>	Irina Pirles
<b>Rol</b>	Desarrolladora
<b>Categoría Profesional</b>	Desarrolladora
<b>Responsabilidad</b>	Programar el sistema, cumplir las tareas impuestas
<b>Información de contacto</b>	fridaandrade1313@gmail.com

<b>Nombre</b>	Astrid Ana Luppi
<b>Rol</b>	Desarrolladora
<b>Categoría Profesional</b>	Desarrolladora
<b>Responsabilidad</b>	Programar el sistema, cumplir las tareas impuestas
<b>Información de contacto</b>	astrita1970@gmail.com

<b>Nombre</b>	Vanessa Marcela Villanueva Casini
<b>Rol</b>	Desarrolladora
<b>Categoría Profesional</b>	Desarrolladora
<b>Responsabilidad</b>	Programar el sistema, cumplir las tareas impuestas
<b>Información de contacto</b>	profe.marcelavillanueva@gmail.com

<b>Nombre</b>	Mauricio Nahuel Pucheta
<b>Rol</b>	Desarrollador
<b>Categoría Profesional</b>	Desarrollador
<b>Responsabilidad</b>	Programar el sistema, cumplir las tareas impuestas

<b>Información de contacto</b>	mauriciopucheta21@gmail.com
--------------------------------	-----------------------------

<b>Nombre</b>	Ariel Orlando Fernández
<b>Rol</b>	Desarrollador
<b>Categoría Profesional</b>	Desarrollador
<b>Responsabilidad</b>	Programar el sistema, cumplir las tareas impuestas
<b>Información de contacto</b>	lic.arielfernandez.hys@gmail.com

#### 1.4 Definiciones, acrónimos y abreviaturas

<b>Nombre</b>	<b>Descripción</b>
<b>Usuario</b>	Persona que usará el sistema para gestionar procesos
<b>SIS-I</b>	Sistema de Información Web para la Gestión de Procesos Administrativos y Académicos
<b>ERS</b>	Especificación de Requisitos Software
<b>RF</b>	Requerimiento Funcional
<b>RNF</b>	Requerimiento No Funcional
<b>FTP</b>	Protocolo de Transferencia de Archivos
<b>Moodle</b>	Aula Virtual

#### 1.5 Referencias

<b>Título del Documento</b>	<b>Referencia</b>
Standard IEEE 830 - 1998	IEEE

## 1.6 Resumen

El documento está dividido en las siguientes secciones:

- **Introducción:** Define el propósito del documento, su estructura, términos relevantes, su propósito y alcance.
- **Desarrollo:** Nos brinda una visión general del producto, incluyendo datos sobre posibles usuarios y restricciones del sistema.
- **Requisitos específicos:** Habla sobre múltiples tipos de gestiones, consultas, reportes y requerimientos no funcionales que abarca tanto seguridad, disponibilidad, compatibilidad y otros aspectos clave.

# 2. Descripción general

## 2.1 Perspectiva del producto

El sistema **AquaMóvil** será un producto diseñado para trabajar en entornos Web y Móvil, lo que permitirá su utilización de forma rápida y eficaz. Estará orientado a la gestión integral de servicios de lavado de autos a domicilio, facilitando la interacción entre clientes, administradores y empleados lavacoches.

El sistema se apoyará en una arquitectura cliente-servidor, con interfaz accesible mediante navegadores y dispositivos móviles, y con persistencia de datos en una base de datos relacional. Esto permitirá:

- Registrar usuarios con distintos roles.
- Gestionar pedidos de servicio.
- Administrar pagos y generar liquidaciones para los operarios.
- Monitorear el estado de cada solicitud en tiempo real.

## 2.2 Características de los usuarios

<b>Tipo de usuario</b>	Administrador. Usuario administrador.
<b>Formación</b>	Manejo básico de herramientas informáticas y de gestión.
<b>Actividades</b>	<ul style="list-style-type: none"> <li>• Control y manejo del sistema en general.</li> <li>• Alta, baja y modificación de usuarios.</li> <li>• Supervisión de pedidos y asignación de empleados.</li> <li>• Gestión de reportes y estadísticas de uso.</li> </ul>

<b>Tipo de usuario</b>	Cliente. Usuario estándar
<b>Formación</b>	Manejo básico de entornos web o aplicaciones móviles.
<b>Actividades</b>	<ul style="list-style-type: none"> <li>• Registrarse e iniciar sesión en la plataforma.</li> <li>• Solicitar un servicio de lavado de autos a domicilio.</li> <li>• Seleccionar tipo de vehículo y servicio (básico, premium, etc.).</li> <li>• Especificar detalles adicionales y método de pago.</li> <li>• Consultar el estado de sus pedidos.</li> </ul>

<b>Tipo de usuario</b>	Empleado (lavacoches). Usuario estándar
<b>Formación</b>	Manejo básico de entornos web o aplicaciones móviles.
<b>Actividades</b>	<ul style="list-style-type: none"> <li>• Registrarse e iniciar sesión en la plataforma.</li> <li>• Consultar pedidos asignados.</li> <li>• Actualizar el estado del servicio (Pendiente, En proceso, Listo, Entregado).</li> <li>• Confirmar la finalización del trabajo.</li> <li>• Visualizar información de liquidaciones o pagos pendientes.</li> </ul>

<b>Tipo de usuario</b>	Visitante. Usuario ocasional, no requiere login
------------------------	---

<b>Formación</b>	Manejo básico de entornos web o aplicaciones móviles.
<b>Actividades</b>	<ul style="list-style-type: none"> <li>• Observar e indagar información general de los servicios brindados.</li> <li>• Acceder a descripciones, planes y tarifas.</li> <li>• No posee acceso a funcionalidades críticas hasta registrarse.</li> </ul>

## 2.3 Restricciones

**Tecnologías:** El desarrollo deberá basarse en tecnologías **open-source** ampliamente utilizadas y con soporte activo de la comunidad, tales como:

- **Motor de base de datos:** MySQL o equivalente (MariaDB, PostgreSQL en caso de migración futura).
- **Backend:** Python con frameworks como Django, Flask o FastAPI.
- **Frontend:** HTML5, CSS3, JavaScript (ES6+) y frameworks/librerías opcionales como React, Vue o similares.
- Se excluye en esta primera etapa el uso de tecnologías propietarias o dependientes de licencias costosas, con el fin de mantener la escalabilidad y sostenibilidad del sistema.
- **Internacionalización e idioma:** La primera versión (MVP) estará disponible únicamente en **idioma español**, aunque se contemplará la integración de servicios externos como **Google Translate** para habilitar la traducción automática a futuro.
- **Compatibilidad y navegadores:** La aplicación debe ser accesible desde **navegadores modernos** (Google Chrome, Mozilla Firefox, Microsoft Edge, Opera y Brave). El sistema deberá garantizar soporte oficial para las **dos últimas versiones estables** de cada navegador, asegurando así compatibilidad y un alcance amplio a usuarios finales.
- **Rendimiento y UX:** El tiempo de respuesta en operaciones estándar (peticiones GET, POST, PUT, DELETE bajo condiciones normales de carga) **no**



**debe superar los 3 segundos.** Para alcanzar dicho objetivo, se priorizará el uso de consultas optimizadas, caching en memoria y un diseño eficiente de la base de datos.

**Seguridad y normativa:** El sistema deberá cumplir en todas sus etapas con la **Ley 25.326 de Protección de Datos Personales (Argentina)** y normativas equivalentes, incorporando:

- **Autenticación en dos factores (2FA).**
- **Cifrado de contraseñas** mediante algoritmos robustos (ej: bcrypt).
- **Encriptación de datos sensibles en tránsito y en reposo (HTTPS).**
- **Gestión de sesiones seguras** con expiración automática y JWT. Estas medidas buscan resguardar la confidencialidad, integridad y disponibilidad de la información.

**Alcance funcional limitado (MVP):** La autenticación se realizará únicamente mediante usuario y contraseña internos al sistema. **No se contemplará integración con terceros (ej: redes sociales, SSO corporativos)** en esta primera entrega.

**Integraciones externas:** Se habilitarán únicamente aquellas solicitadas explícitamente por el cliente, siendo la principal la **integración con pasarelas de pago** (ejemplo: MercadoPago). No se prevén integraciones adicionales (ej: sistemas contables o de RRHH) durante la primera versión.

**Reportes:** Los reportes de lavados y facturación deberán generarse en **PDF y/o Excel**, según la preferencia del cliente, garantizando exportación clara, con encabezados, filtros y totales.

**Plazos de entrega:** El software deberá entregarse dentro del **plazo previamente acordado** con el cliente. Cualquier extensión deberá ser validada formalmente.

## 2.4 Supuestos

- **Conectividad de los usuarios:** Se asume que los usuarios finales contarán con una **conexión a internet estable y de velocidad media/alta**, condición necesaria para garantizar tiempos de respuesta acordes a los objetivos de UX.
- **Volumen inicial de usuarios concurrentes:** Se estima que el sistema deberá soportar hasta **100 usuarios concurrentes** en tiempo real durante la primera etapa. Este límite se toma como referencia para dimensionar la arquitectura y no implica que el sistema no pueda escalar en versiones futuras.
- **Validación de datos ingresados:** El sistema asumirá que los **datos provistos por los usuarios serán válidos y completos**, aplicando únicamente validaciones básicas (ejemplo: formato de patente). Casos excepcionales (datos incompletos, tickets perdidos) serán gestionados manualmente por el personal administrativo.
- **Despliegue y hosting:** La **infraestructura de producción (servidor, dominio y hosting)** será provista por el cliente, aunque el equipo de desarrollo podrá brindar asesoramiento técnico para la correcta implementación.
- **Volumen inicial de datos:** Se prevé que en la primera etapa no se superarán los **5.000 registros de lavados por mes**, siendo una carga fácilmente soportada por la base de datos relacional seleccionada.
- **Ámbito de uso:** El sistema estará orientado a una **única sucursal** en su primera entrega. Procesos inter-sucursal o integraciones de múltiples sedes serán responsabilidad del cliente y/o contemplados en futuras versiones.
- **Gestión manual de excepciones:** Situaciones fuera de la operatividad estándar, como **pérdida de tickets, lavados no registrados o conciliaciones entre sucursales**, serán resueltas manualmente por el cliente en esta primera etapa.
- **Capacitación del personal:** Se asume que el personal administrativo y de atención al cliente recibirá **capacitación adecuada** en el uso del sistema, lo que permitirá reducir incidencias y asegurar un correcto aprovechamiento de las funcionalidades.

- **Requisitos técnicos del cliente:** Los usuarios finales deberán disponer de **navegadores actualizados (versiones estables)** para prevenir inconsistencias o errores de compatibilidad.
- **Operatividad continua de integraciones:** Se asume que las integraciones externas críticas, como la **pasarela de pagos**, estarán disponibles de forma continua. Caídas prolongadas o incidencias en estos servicios quedan fuera del alcance del presente desarrollo.

### 3. Requisitos específicos

#### 3.1 Requerimientos funcionales

##### 3.1.1 Gestión de usuarios

- ◆ **RF1.** El sistema permitirá el registro de nuevos usuarios (cliente, empleado, administrador).
- ◆ **RF2.** El sistema permitirá a los administradores gestionar (alta, baja, modificación) usuarios.
- ◆ **RF3.** El sistema validará credenciales de inicio de sesión con seguridad (usuario y contraseña).

##### 3.1.2 Gestión de pedidos

- ◆ **RF4.** El cliente podrá registrar un pedido de servicio de lavado especificando: tipo de vehículo, tipo de servicio y observaciones.
- ◆ **RF5.** El empleado podrá consultar los pedidos asignados y actualizar su estado.
- ◆ **RF6.** El administrador podrá supervisar todos los pedidos en curso.

### 3.1.3 Gestión de pagos

- ◆ **RF7.** El cliente podrá abonar el servicio mediante tarjeta de crédito o transferencia bancaria.
- ◆ **RF8.** El sistema generará automáticamente la liquidación correspondiente para el empleado una vez finalizado el servicio.

### 3.1.4 Gestión de estados y notificaciones

- ◆ **RF9.** El sistema permitirá actualizar los estados del pedido: *pendiente, en proceso, listo, entregado*.
- ◆ **RF10.** El sistema enviará notificaciones al cliente sobre el progreso de su pedido.

### 3.1.5 Consultas y reportes

- ◆ **RF11.** El administrador podrá generar reportes sobre usuarios, pedidos y pagos.
- ◆ **RF12.** El cliente podrá consultar su historial de pedidos.
- ◆ **RF13.** El empleado podrá visualizar su historial de servicios realizados.

## 3.2 Requerimientos no funcionales

### 3.2.1 Rendimiento

El sistema Aquamóvil está desarrollado como una aplicación de línea de comandos, lo que permite tiempos de respuesta inmediatos en entornos locales. La autenticación se realiza mediante comparación directa de credenciales en estructuras internas, sin acceso a bases de datos externas ni procesos asincrónicos.

- Tiempo de respuesta: El sistema responde a entradas del usuario en menos de 2 segundos en condiciones normales, validado mediante pruebas manuales en CLI.
- Autenticación: El proceso de login se completa en menos de 3 segundos, gracias a la validación directa en estructuras de datos.
- Concurrencia: Aunque no se ha implementado concurrencia real, el diseño modular permite escalar hacia múltiples sesiones simultáneas en futuras versiones.

### 3.2.2 Usabilidad

La interfaz de Aquamóvil está diseñada para ser clara y funcional en entornos de línea de comandos. Cada rol tiene un menú específico, y las opciones están numeradas y acompañadas de mensajes descriptivos.

- Interfaz por rol: Los usuarios acceden a menús diferenciados según su tipo (Cliente, Empleado, Administrador), con opciones específicas para cada perfil.
- Validaciones: Todos los campos requeridos son validados. Se implementan mensajes de error específicos como “campo vacío” o “formato incorrecto”.
- Registro: El sistema permite hasta 9 intentos de registro antes de bloquear temporalmente el proceso, con retroalimentación en cada paso.
- Navegación: El flujo de interacción es secuencial e intuitivo, con confirmaciones antes de operaciones críticas.

### 3.2.3 Seguridad

La seguridad en Aquamóvil se implementa a nivel lógico, con restricciones de acceso y control de autenticación. Aunque no se utiliza cifrado aún, el diseño contempla su incorporación.

- Contraseñas: Se exige una longitud mínima de 8 caracteres. La validación incluye la presencia de letras y números.

- Control de acceso: Los usuarios no registrados no pueden acceder al sistema. Se limita el login a 3 intentos consecutivos por sesión.
- Protección de administrador: El sistema impide que un administrador elimine su propia cuenta, preservando el control del sistema.
- Cifrado pendiente: Actualmente las contraseñas se almacenan en texto plano, pero se ha planificado la implementación de hash con bcrypt o hashlib.

#### **3.2.4 Fiabilidad**

El sistema maneja errores mediante estructuras try/except, evitando caídas inesperadas. Las operaciones críticas requieren validación previa.

- Manejo de errores: Se implementan bloques try/except para capturar excepciones en ingreso de datos y navegación.
- Reintentos: En caso de error, el sistema permite reintentar sin perder el estado del flujo.
- Validación previa: Antes de eliminar usuarios o cambiar roles, se verifica la existencia del usuario en el sistema.

#### **3.2.5 Mantenibilidad**

El código está organizado en módulos y clases por tipo de usuario, lo que facilita su extensión y mantenimiento.

- Modularización: Clases como Cliente, Empleado, Administrador y Direcciones\_empleado encapsulan funcionalidades específicas.
- Funciones descriptivas: Los métodos tienen nombres claros como validar\_datos(), mostrar\_menu(), registrar\_usuario(), lo que facilita su comprensión.
- Documentación: Se han incluido comentarios explicativos en el código. Se recomienda formalizar la documentación con docstrings en formato Google o NumPy.

### 3.2.6 Portabilidad

Aquamóvil está desarrollado en Python 3.10, sin dependencias externas, lo que permite su ejecución en múltiples plataformas.

- Entorno: Compatible con cualquier sistema que soporte Python 3.10 o superior.
- Sistemas operativos: Probado en Windows y Linux. Se proyecta compatibilidad con macOS.
- Dependencias: No se utilizan librerías externas, lo que simplifica la instalación y despliegue.

## Product Backlog

### Cliente

1. **HU-CL1:** Como cliente, quiero registrarme en la plataforma para poder solicitar servicios de lavado.
  - Prioridad: Alta
  - Criterio de aceptación: El sistema valida datos y confirma creación de cuenta.
2. **HU-CL2:** Como cliente, quiero iniciar sesión con mi cuenta para acceder a mis pedidos y datos.
  - Prioridad: Alta
  - Criterio de aceptación: El sistema autentica y redirige al panel del cliente.
3. **HU-CL3:** Como cliente, quiero solicitar un servicio de lavado a domicilio para mantener limpio mi vehículo.
  - Prioridad: Alta

- Criterio de aceptación: La solicitud queda registrada y asignada a un empleado.
- 4. **HU-CL4:** Como cliente, quiero seleccionar el tipo de vehículo y tipo de servicio (básico, premium, etc.) para personalizar el pedido.
  - Prioridad: Alta
- 5. **HU-CL5:** Como cliente, quiero especificar detalles adicionales y método de pago para asegurar que se cumpla mi pedido.
  - Prioridad: Media
- 6. **HU-CL6:** Como cliente, quiero consultar el estado de mis pedidos en tiempo real para saber cuándo llegará el servicio.
  - Prioridad: Alta
- 7. **HU-CL7:** Como cliente, quiero recibir notificaciones (mail/app) para enterarme de cambios en mis pedidos.
  - Prioridad: Media

#### **Empleado lavacoches**

- 8. **HU-EMP1:** Como empleado, quiero registrarme e iniciar sesión en la plataforma para acceder a mis pedidos.
  - Prioridad: Alta
- 9. **HU-EMP2:** Como empleado, quiero consultar los pedidos asignados para saber qué servicios debo realizar.
  - Prioridad: Alta
- 10. **HU-EMP3:** Como empleado, quiero actualizar el estado de un servicio (Pendiente, En proceso, Listo, Entregado) para informar avances al cliente.



- Prioridad: Alta

11. **HU-EMP4:** Como empleado, quiero confirmar la finalización del trabajo para que quede registrado y se libere el pago.

- Prioridad: Alta

12. **HU-EMP5:** Como empleado, quiero visualizar liquidaciones o pagos pendientes para tener claridad de mis ingresos.

- Prioridad: Media

### Administrador

13. **HU-ADM1:** Como administrador, quiero dar de alta, baja o modificar usuarios para gestionar el sistema.

- Prioridad: Alta

14. **HU-ADM2:** Como administrador, quiero supervisar pedidos y asignar empleados para garantizar que todos los pedidos se atiendan.

- Prioridad: Alta

15. **HU-ADM3:** Como administrador, quiero generar reportes de uso y estadísticas para analizar el rendimiento del servicio.

- Prioridad: Media

16. **HU-ADM4:** Como administrador, quiero gestionar pagos y liquidaciones para cumplir con la administración financiera.

- Prioridad: Alta

### Visitante

17. **HU-VIS1:** Como visitante, quiero consultar información general del servicio (planes, tarifas, descripción) para decidir si registrarme.

- Prioridad: Media

18. **HU-VIS2**: Como visitante, quiero poder registrarme en la plataforma para convertirme en cliente.

- Prioridad: Alta

## Sprints.

N° de sprint	00
<b>Sprint Backlog</b>	<p>US01: Generar README.md</p> <p>US02: Redactar sección "Descripción general".</p> <p>US03: Entregar diagrama de clases.</p> <p>US04: Redactar Requisitos Funcionales.</p> <p>US05: Entregar DER.</p> <p>US06: Redactar Requisitos No Funcionales.</p> <p>US07: Publicar diagramas en la Wiki.</p> <p>US08: Documentar Restricciones y Supuestos.</p> <p>US09: Validar coherencia Diagrama–DER.</p> <p>US10: Unificar y publicar versión final IEEE 830.</p> <p>US11: Configurar flujo de trabajo (rama develop, Kanban, milestone).</p> <p>US12: Mantener flujo de issues y PRs.</p> <p>TK01: Subir README.md al repo.</p> <p>TK02: Subir IEEE 830 a documentacion/IEEE 830/</p> <p>TK03: Revisar consistencia IEEE 830</p> <p>TK04: Identificar clases, atributos y métodos</p> <p>TK05: Definir relaciones</p> <p>TK06: Diagramar en Draw.io</p> <p>TK07: Exportar PDF del diagrama</p> <p>TK08: Identificar entidades y atributos</p> <p>TK09: Definir relaciones y cardinalidades</p> <p>TK10: Diagramar DER en Draw.io</p> <p>TK11: Exportar DER (PDF)</p> <p>TK12: Enlazar DER en la Wiki</p> <p>TK13: Agregar RNF al documento IEEE 830</p> <p>TK14: Subir diagramas a la Wiki.</p> <p>TK15: Organizar y verificar accesibilidad</p>

	<p>TK16: Listar restricciones y supuestos</p> <p>TK17: Redactar sección “restricciones y supuestos” en IEEE 830</p> <p>TK18: Comparar Diagrama–DER</p> <p>TK19: Reportar ajustes en comentario de issue</p> <p>TK20: Revisar y unificar formato IEEE 830</p> <p>TK21: Subir versión final a documentacion/IEEE830/</p> <p>TK22: Crear rama develop</p> <p>TK23: Configurar tablero Kanban (Backlog / In Progress / In Review / Ready)</p> <p>TK24: Crear milestone “Sprint 0” y asociar issues</p> <p>TK25: Definir rutina de mantenimiento del flujo</p> <p>TK26: Revisar y aprobar pull requests hacia develop</p> <p>TK27: Agregar RF al documento IEEE 830</p>
<b>Responsabilidades</b>	<p>Picone Abigail: US05, US10 - US12, TK08-TK11, TK20 - TK26</p> <p>Pirles Irina: US01, TK01 - TK03</p> <p>Luppi Astrid: US02 - US03, TK04 - TK07</p> <p>Villanueva Marcela: US04, TK08 - TK27</p> <p>Fernández Ariel: US06 - US07, TK12 - TK15</p> <p>Pucheta Mauricio: US08 - US09, TK16 - TK19</p>
<b>Calendario</b>	<p>Inicio: 14 de Agosto 2025</p> <p>Cierre: 24 de Agosto 2025</p>
<p><b>Inconvenientes:</b></p> <ul style="list-style-type: none"> <li>• <u>Dependencias entre tareas</u>: retraso en la redacción de la sección “Descripción general” bloquea el diagrama de clases (las clases surgen de los RF generales); sin el DER definitivo, no se puede validar completamente el diagrama de clases y viceversa; la sección de RF/RNF incompleta impide cerrar la versión final del IEEE.</li> <li>• <u>Inconsistencias de formato y estilo</u>: diferentes formatos APA en cada sección generan retrabajo al unificar el PDF; márgenes, tipografías o numeración distintas en los documentos subidos provocan revisiones adicionales.</li> </ul>	

- Disponibilidad del equipo: Miembros con carga académica o profesional alta pueden demorar la entrega de sus secciones; conflictos de horario o falta de sincronización en daily meetings.
- Gestión de pull requests: PRs que quedan sin revisores asignados, alargando el merge a develop; conflictos de merge frecuentes si dos o más personas editan el mismo apartado del IEEE al mismo tiempo.
- Comunicación y seguimiento: Kanban desactualizado, por ejemplo, issues en “In Progress” que en realidad están en proceso de revisión; milestone “Sprint 0” sin issues asociados correctamente, perdiendo visibilidad del avance.
- Calidad de la revisión: Revisión superficial del IEEE puede omitir errores de contenido o referencias APA.

## Anexo I

[Diagrama Entidad-Relación](#)

[Diagrama de Clases](#)

[Repositorio de GitHub](#)

[Wiki](#)