



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

FACULTAD DE ESTUDIOS SUPERIORES  
ARAGÓN

## SISTEMA DE EMISIÓN DE LICENCIAS DE CONducIR PARA EL GOBIERNO DE GUANAJUATO

DESARROLLO DE UN CASO PRÁCTICO

QUE PARA OBTENER EL TÍTULO DE  
INGENIERO EN COMPUTACIÓN

**PRESENTA:**

CARLOS ALBERTO REYES ANGELES

**ASESOR:**

M. EN C. JESÚS HERNÁNDEZ CABRERA



Noviembre 2019



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



## Agradecimientos

La realización de este trabajo puede verse como el desenlace de un proceso de muchos años, previos y durante la carrera.

En ese sentido, quiero hacer un agradecimiento muy afectuoso a:

Mi asesor:

- M. en C. Jesús Hernández

Mis Familiares que me brindaron su apoyo siempre:

- Carmen Angeles
- Isidra Larrieta
- Isabel Angeles
- Adán López
- Cristian Reyes
- Enrique Reyes
- Filiberto Angeles
- Rubén Angeles
- y demás familia que me ayudó

Compañeros:

- Angel Martínez
- Alonso Martínez
- Enrique Falcón
- Gabriel Gutiérrez

Un agradecimiento muy especial a NEMARIS, que mediante la carrera me brindó un desarrollo profesional impagable.

# Sistema de emisión de licencias de conducir para el gobierno de Guanajuato

## Contenido

Agradecimientos .....	1
Sistema de emisión de licencias de conducir para el gobierno de Guanajuato .....	2
1. Introducción .....	4
2. Orígenes del desarrollo del sistema .....	6
2.1 Sistema anterior .....	6
2.2 Nuevo sistema .....	6
2.3 Descripción de las principales consideraciones para el desarrollo .....	7
3. Descripción del sistema y análisis de requerimientos con UML .....	11
3.1 UML (Lenguaje de Modelado Unificado) .....	11
3.2 Jerarquía de actores del sistema .....	11
3.3 Diagrama de casos de uso .....	12
3.4 Casos de uso .....	13
4. Arquitectura y diseño .....	34
4.1 Servidor .....	34
4.2 Cliente .....	38
4.3 Metodología de desarrollo .....	40
4.4 Herramientas de desarrollo .....	41
4.5 Equipo de desarrollo .....	42
5. Implementación .....	43
5.1 Comparación de huellas 1 a 1 .....	43
5.2 Cifrado y descifrado de cadenas de texto .....	44
5.3 Despliegue de un formulario usando Struts2 .....	44
5.4 Creación y eliminación de sesión usando Struts2 .....	46
5.5 Validaciones y llamadas AJAX en el procesamiento de formularios .....	47
6. Fase de pruebas .....	50
6.1 Inicio de sesión .....	50
6.2 Registro de datos del trámite Nueva Licencia .....	51
6.3 Captura de información biométrica .....	52
6.4 Impresión de licencias .....	53
7. Despliegue del sistema .....	56

7.1 Sistema central.....	56
7.2 Unidades móviles .....	58
Conclusiones.....	59
Bibliografía .....	61
Referencias.....	61

## 1. Introducción

En la actualidad la movilidad a través de vehículos es una actividad que se realiza todos los días. Suele ser a través de transporte público, transporte privado, servicios que usan vehículos de carga, entre otros usos. Dependiendo del estado del país, se usan políticas de movilidad que generalmente se supondrían son las mejores para el desarrollo y desempeño de esta actividad. Tales políticas suelen ser los reglamentos para uso de vialidad, que son el establecimiento de restricciones referentes a características de los vehículos y conductores, para el aseguramiento de la circulación efectiva a través de las vialidades públicas. En reglamentos también se establecen reglas dirigidas a las personas que hacen uso de los vehículos, dichas reglas dictan los conocimientos viales mínimos requeridos para la operación. La forma de verificar tales capacidades y conocimientos de un conductor es a través de una licencia de conducir, la cual podemos definir en palabras sencillas como un permiso que otorga el gobierno a las personas para que puedan conducir sobre las vialidades públicas. Esto significa que, si el gobierno quiere emitir una licencia, debe asegurarse de que la persona receptora cuenta con las facultades requeridas previamente, lo cual se hace a través de registros de información de la persona, exámenes teóricos, y/o pruebas de manejo. Dependerá de la entidad donde se realice la emisión.

La emisión de licencias de conducir es una operación que realiza el estado de Guanajuato. Dicha actividad se hace con un sistema basado en las tecnologías de la información, el cual tiene mucha relevancia en la región, porque a través de él, las personas pueden tramitar sus licencias para los distintos tipos usos y tipos de vehículos, lo cual a su vez les permitirá realizar diferentes actividades en un marco legal. Siendo las tecnologías de la información la piedra angular que sostiene el funcionamiento de este tipo de sistemas, se contempla usar todas las herramientas en el alcance del proyecto de la mejor forma posible. Las principales funciones de este sistema son:

- Captura de datos de conductores
- Captura de información biométrica de los conductores
- Almacenamiento de historial de conductores y sus licencias
- Impresión de licencias y permisos de conducir
- Control de materiales usados en la impresión de licencias
- Reportes

Dado el alcance del sistema los objetivos son:

- Generar un contexto, identificar los problemas de la aplicación previa y la resolución de problemas con este nuevo sistema.
- Describir las consideraciones principales al momento del desarrollo de este sistema.
- Hacer una descripción del sistema a través de los casos de uso y diagramas de flujo de la operación por parte de los usuarios.
- Describir el diseño y arquitectura de la nueva aplicación.
- Ejemplificar la implementación de tecnologías en el desarrollo del sistema.
- La documentación de pruebas que fueron realizadas al sistema.
- Exponer la distribución de los elementos del sistema a través de un diagrama de despliegue.

Previo a la realización del sistema se debieron tomar en cuenta todas las reglas que establece el cliente para la emisión, así como tener en cuenta todos los casos posibles que puedan suceder mediante el funcionamiento del sistema. Por eso deben describirse los orígenes del sistema, enlistando todas las posibles funciones del sistema. Además, se deben tener presentes las deficiencias de los sistemas previos para poder ofrecer un funcionamiento eficaz.

El diseño y elección de la arquitectura del sistema fue el primer paso en la elaboración de esta aplicación. En general la aplicación se encuentra principalmente en la plataforma web, pero al ser necesario el uso de hardware que hacen tareas muy específicas y este tipo de hardware no suele tener una integración directa con los navegadores web, se echa mano de cuatro subsistemas y un protocolo de comunicación entre navegador y procesos externos. Se elige una arquitectura llamada Cliente-Servidor, la cual es descrita en base a la aplicación de emisión de licencias, intentando ubicar cada parte y tecnología en un plano que interprete la estructura general a través de todo el sistema.

La implementación de tecnologías en este sistema es mucha. Pero se describen algunos aspectos principales que fueron usados y que podrían ser de utilidad a los demás. Hablamos de como se hace el procesamiento de un formulario, de cómo hacer una comparación de huellas dactilares, la creación y la eliminación de sesiones en los entornos web o el cifrado de datos en el lenguaje java.

Se tuvo una fase de pruebas estricta, pero por cuestiones de privacidad solo se pudo publicar una serie de pruebas sencillas, que ayudan a entender el objetivo de tal fase. Este sistema trabajará con entre 50 y 70 equipos cliente, lo cual significa que en la puesta en marcha no puede haber 70 usuarios teniendo problemas, y lo más importante, no puede haber una cantidad de solicitantes de licencia teniendo problemas con sus trámites. Las pruebas descritas en este documento buscan que las funciones que tiene el sistema trabajen de forma correcta ante el usuario.

El despliegue del sistema es donde se asientan todos los componentes que son descritos en este documento. Se hace una relación de hardware y software, así como su distribución e interacción entre todas estas piezas del sistema generando una relación del sistema y subsistemas usados con las piezas de hardware que se usan.

## 2. Orígenes del desarrollo del sistema

### 2.1 Sistema anterior

Antes del 2019 se usaba un sistema web para la emisión y control de registro de licencias.

Principales problemáticas:

- Diseño del sistema
  - El diseño de la base de datos carecía de normalización en gran parte de la estructura. Esto permitía la duplicidad de gran cantidad de información.
  - Algunas tablas en la base de datos eran muy grandes, lo cual propiciaba a un manejo muy rígido de la información, además de permitir la creación de registros con muchos campos en NULL.
- Interfaz de usuario
  - Era vieja y poco clara con el usuario.
  - No era adaptable al tamaño de pantalla de los equipos. Era una desventaja porque las máquinas donde se usa el sistema suelen tener varios tamaños de pantalla.
- Seguridad
  - Los usuarios se prestaban sus contraseñas.
  - No se tenía un registro de los movimientos de los usuarios dentro del sistema.
  - Había usuarios que usaban el sistema fuera de horarios laborales.
  - La edición de licencias era de libre acceso.

### 2.2 Nuevo sistema

Para la construcción de un nuevo sistema, se toma como base a las problemáticas anteriores y como resolución se toman las siguientes medidas:

- Diseño de base de datos
  - Se hace un nuevo diseño de la base de datos completamente normalizada. Se garantiza que los datos serán congruentes gracias al uso de las reglas de normalización.
  - Se distribuye información de las tablas en varias tablas, en forma de catálogos y otro tipo de entidades. Se restringe en la medida de lo posible el permitir el almacenamiento de registros con campos en NULL.
- Interfaz de usuario
  - Se usan componentes amigables al usuario y se optimizan los flujos del sistema anterior, para mayor velocidad en la carga del sistema y mayor velocidad en los procesos realizados por parte del usuario.
  - La interfaz del nuevo sistema es tiene un diseño responsivo, lo cual significa que es adaptable al tamaño de pantalla del dispositivo que se esté usando.
- Seguridad
  - Se añade la autenticación con huella dactilar como un segundo factor en el inicio de sesión.
  - Se maneja una bitácora de movimientos en el sistema. En esta se guardan y consultan las acciones de los usuarios.

- Se establece un horario en cada módulo de emisión, el cual es revisado al momento de iniciar sesión, si se detecta que no es horario de servicio no se podrán hacer movimientos en el sistema.
- La edición de licencias es un movimiento que solo se podrá realizar con la autorización de un supervisor (rol que será explicado más adelante).

### 2.3 Descripción de las principales consideraciones para el desarrollo

La emisión de licencias y permisos de conducir está restringida a la población del estado de Guanajuato. El sistema deberá permitir a la parte emisora el registro todos los trámites para tener una trazabilidad de las emisiones, teniendo en cuenta las siguientes consideraciones.

#### Usuarios

Los usuarios son las entidades que a través de un registro podrán utilizar el sistema. Se genera una abstracción de cada persona, así identificar a los usuarios dentro del sistema. La principal característica para identificación de usuarios es que se almacenan dos huellas dactilares.

Además de la información biométrica se almacena su nombre y apellidos, el módulo al que pertenece, un nombre de usuario que funcionará como identificador público en el sistema y una contraseña como factor de acceso.

#### Control de acceso

Para el uso del sistema es necesario usar un formulario para la identificación de usuarios. Las validaciones para el inicio de sesión son las siguientes:

- Seleccionar cual es módulo de emisión.
- Verificación de horarios de servicio.
- Ingresar usuario.
- Ingresar contraseña.
- Ingresar huella dactilar de la persona que quiere iniciar sesión a través de un escáner de huella dactilar y compararla con los datos asociados al usuario.

El protocolo de acceso usa la autenticación por dos factores. Esto se refiere a que el usuario debe proporcionar más de un indicio para la verificación de su identidad. En este caso el primer factor es la contraseña, y el segundo es su huella dactilar.

#### Perfiles y funciones

Una función es un segmento del sistema. En el sistema hay un árbol de funciones, es decir, hay funciones que clasifican a otras funciones. Las cuatro principales son las siguientes:

1. Administración
2. Catálogos
3. Reportes
4. Trámites

Hay funciones del sistema que están asociadas a cada una de las anteriores, en seguida se muestra en ejemplo de una lista de funciones que derivan de una:

1. Administración
  - a. Módulos
  - b. Usuarios

- c. Perfiles
- d. Gestión de costos
- e. Firmas del director
- f. Configuración

Un perfil es la elección de un conjunto de funciones específicas. A los usuarios de la aplicación se les asigna un perfil, de esta forma se limita las funciones que pueden realizar. Hay un perfil que tiene asociado todas las funciones, tal perfil es asignado a uno o varios usuarios que sean administradores del sistema.

#### *Módulos de emisión*

Son los sitios donde se emiten las licencias de conducir. El sistema cuenta con una parte específica para la administración de tales módulos de emisión. Se debe especificar en encargado del módulo y los horarios en lo que se puede emitir, así, los usuarios en ese módulo no puedan acceder fuera de los horarios asociados.

#### *Licencias y permisos*

La emisión de una licencia o permiso de conducir es un trámite que realizan los ciudadanos para que la secretaría de movilidad (dependiendo la localidad) autorice que pueden manejar vehículos. Dependiendo el tipo de licencia o permiso solicitado se podrá conducir una o varias clases de vehículo. A su vez, cada trámite emitirá un documento que es una tarjeta de PVC<sup>1</sup> que tiene impreso los datos del conductor, los datos de vigencia y tipos de vehículos autorizados a conducir.

La diferencia entre permisos y licencias es que las licencias se otorgan a personas que tengan 18 años o más de edad. Los permisos se otorgan a personas que tengan desde 15 años y sean menores a 18 años.

#### *Vigencia*

La vigencia de una licencia o permiso puede variar. En el caso de las licencias hay de 2, 3 y 5 años.

Para los permisos de conducir la regla es que dependiendo la edad del solicitante se calculará la vigencia del permiso. Por ejemplo, si una persona tiene 15 años, se le otorga una vigencia de hasta 3 años, siendo la fecha de vencimiento igual a la fecha en que esa persona cumpla los 18 años.

#### *Tipos de licencias*

Los tipos de licencia que se emiten son los siguientes:

- Licencias
  - Automovilista
  - Motociclista
  - Chofer de servicio público
  - Chofer de servicio de carga
- Permisos
  - Permiso de automovilista

---

<sup>1</sup> Policloruro de vinilo. Es un plástico, y es el material con el que está hecha la tarjeta.

- Permiso de motociclista

#### *Tipos de movimientos*

El sistema permite hacer los siguientes trámites en cuanto a la emisión de licencias se refiere.

- **Nueva licencia/permiso**  
Se denomina como “nuevo” cuando es la primera vez que se hace el trámite para la obtención de un tipo de licencia.
- **Duplicado de licencia/permiso**  
Este tipo de movimiento aplica para obtener una réplica idéntica a la emisión del último trámite según un tipo de licencia dado. Para este movimiento es necesario que la vigencia del trámite que se quiere duplicar no haya terminado.
- **Renovación de licencia**  
Este movimiento no aplica a los permisos de conducir. Será necesario que la licencia que se desea renovar no esté vigente. De caso contrario se deberá cancelar la última licencia e iniciar la renovación. Pero eso implicará que la licencia previa a la renovada quedará sin ningún uso.
- **Edición de licencia/permiso**  
No es un trámite, sin embargo, es un segmento de la aplicación que permite editar ciertos datos relacionados a un conductor o a la licencia.
- **Cancelar licencia/permiso**  
Significa que la licencia pasa a ser parte de los registros que quedan como historial de trámites asociados al contribuyente. Hay dos motivos para la cancelación de una licencia: el solicitante no hizo el pago para finalizar el trámite o el solicitante desea renovar la vigencia de su licencia y esta sigue vigente.
- **Otros movimientos**  
Hay movimientos que no expiden ningún documento, pero son necesarios para el funcionamiento del sistema.

#### *Consideraciones generales*

1. Un conductor sólo podrá tener asignado un número de licencia<sup>2</sup>, sin importar el tipo de licencia o el tipo de trámite que se haya hecho.
2. La persona deberá tener aprobado el examen teórico o haber aprobado el curso de manejo en las escuelas indicadas por la parte emisora.
3. Para los trámites de chofer de servicio público será necesario que la persona cuente con un tarjetón<sup>3</sup> vigente.

---

<sup>2</sup> Cadena de texto generada por el sistema que se asigna a un conductor y a todos sus permisos o licencias.

<sup>3</sup> Documento emitido por la secretaría de movilidad a los conductores que acredita puede conducir vehículos que brindan el servicio de transporte público.

*Unidades móviles*

Son vehículos como camionetas o autobuses pequeños que están equipados con el equipo para la emisión de licencias. La particularidad de estas unidades es que deben operar sin conexión a ninguna red.

*Sincronización de unidades móviles con base de datos central*

Se realizó un sistema de sincronización entre las unidades móviles y la aplicación central. El sistema de sincronización replicara los datos tomados por las unidades móviles en el servidor central para mantener un control uniforme de todas las emisiones en el servidor central.

### 3. Descripción del sistema y análisis de requerimientos con UML

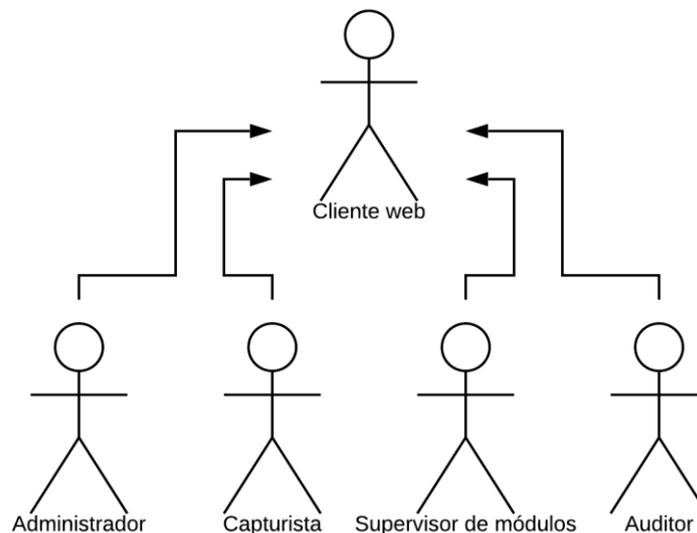
#### 3.1 UML (Lenguaje de Modelado Unificado)

Con el fin de ayudar a la explicación de los distintos segmentos del sistema se usan varios diagramas, los cuales pertenecen a UML. UML es un lenguaje que tiene usos fuera y dentro del área de sistemas informáticos. En este caso se usa para modelar, visualizar, construir y documentar el sistema, además de que es un apoyo gráfico que ayuda al lector a comprender como son las distribuciones y flujos que conforman las partes del sistema. En este documento se usan los siguientes tipos de modelado de la siguiente forma:

- Diagrama de actores: Se usa para hacer una representación de los actores y la distribución que hay entre ellos.
- Diagrama de caso de uso: Se usa para representar las relaciones entre los actores y las funciones del sistema.
- Diagramas de flujo: Se usa para describir paso a paso de manera general o específica el tránsito de los usuarios a través de los procesos y funciones del sistema.
- Diagramas de despliegue: Se usa para describir la parte de hardware o software, la relación entre ambas partes y la distribución de la implementación del sistema.

#### 3.2 Jerarquía de actores del sistema

Todos los actores pertenecen al actor Cliente web como se muestra en la *Ilustración 1*. Los cuatro principales actores en el sistema que son Administrador, Capturista, Supervisor de módulos y el Auditor, que serán los encargados de usar en conjunto todas las funciones del sistema.



**Ilustración 1 – Diagrama de actores.**

- Administrador. Este actor puede manejar la administración de usuarios, perfiles, módulos de emisión y costos. También controla los valores de algunos catálogos.
- Supervisor de módulo. Puede hacer las actividades del capturista y además puede hacer autorizar las ediciones de licencia.
- Auditor. Solo tiene acceso a la parte de reportes y bitácora.

- Capturista. Tiene acceso a todos los movimientos que requieren captura de datos de los solicitantes o impresión de plásticos.

### 3.3 Diagrama de casos de uso

Los casos de uso se dividen en diferentes conjuntos según el tipo de operación al que está ligado, por ejemplo, la creación de un usuario está asociado a la Administración, así como el registro de una nueva licencia está ligado a los Trámites. En este caso Administración y Trámites se denominaron como “casos de uso padre”. Los principales conjuntos de casos de uso son:

- Inicio de sesión
  - Es el primer contacto entre los actores y el sistema. Los actores deben pasar por un control de acceso para poder hacer uso de las funciones según su perfil.
- Administración
  - Módulos. Creación de módulos y edición de módulos.
  - Perfiles. Creación y edición de nuevos perfiles.
  - Usuarios. Creación y edición de usuarios
  - Costos. Edición de los costos de los diferentes trámites.
- Tramites
  - Se permiten hacer los siguientes trámites relacionados a la emisión de licencias:
    - Nueva licencia
    - Duplicado de licencia
    - Renovación de licencia
    - Edición de licencia
    - Cancelar licencia
- Reportes
  - Bitácora. Despliegue y filtrado de las acciones registradas en el sistema.
  - Reporte de licencias emitidas. Listado de las licencias emitidas por día, modulo y por tipo de trámite.
  - Reporte de recaudación. Número y suma de los costos de las licencias emitidas en un intervalo de tiempo.
- Catálogos
  - Alergias. Creación y edición de alergias.
  - Colonias. Creación y edición de colonias.

Los casos de uso van ligados a los distintos tipos de actores. Sin embargo, los actores no pueden usar todos los casos de uso pertenecientes a los casos de uso padre.

En la *Ilustración 2* se visualizan los elementos y funciones del sistema, sus principales actores, así como la relación que hay entre ellos.

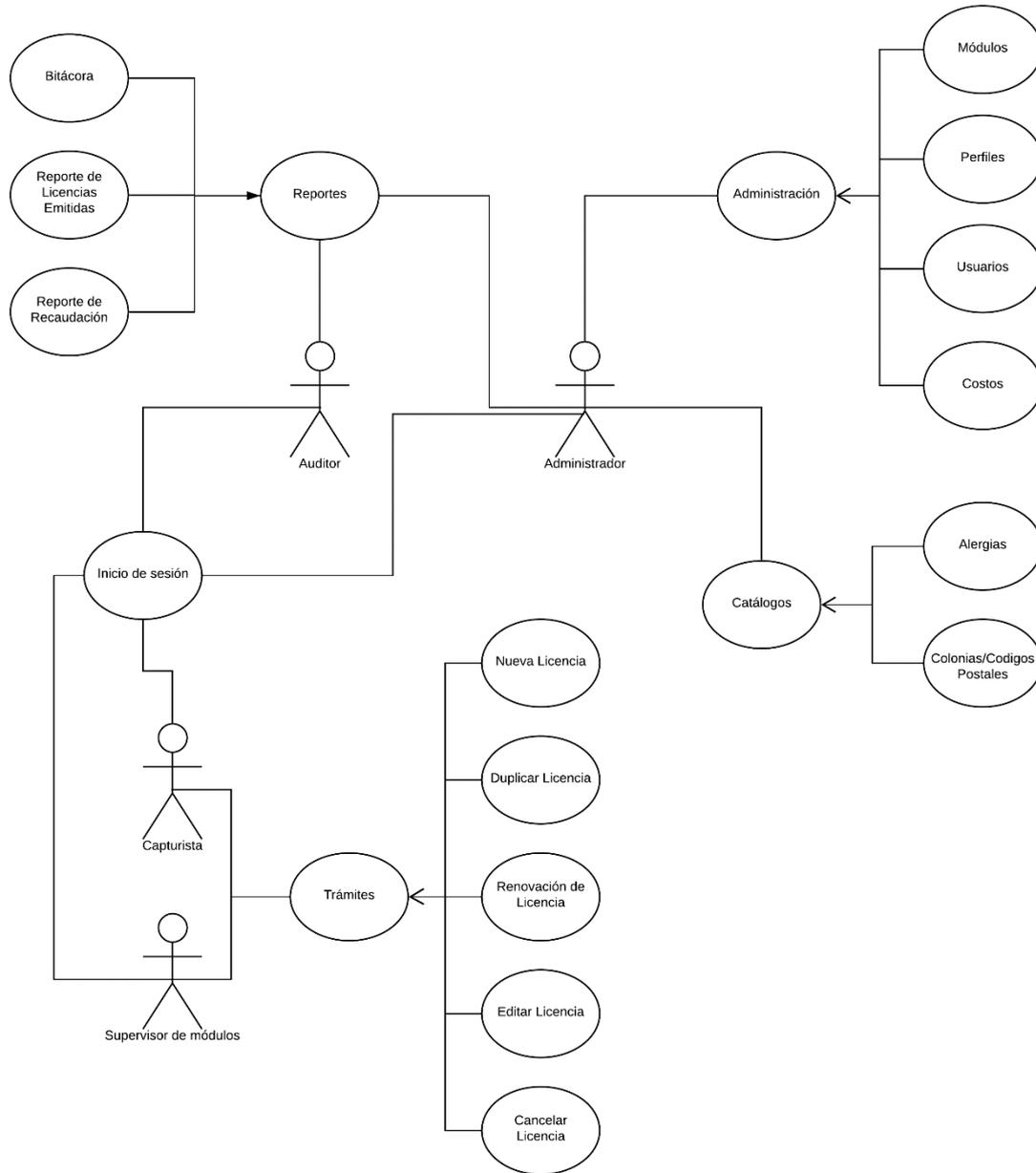


Ilustración 2 - Diagrama general del sistema.

### 3.4 Casos de uso

Un caso de uso es la descripción de una funcionalidad. Estos sirven para guiar el proceso de desarrollo, ya que se pueden generar modelos, diseños y distintas soluciones de acuerdo con la descripción de la funcionalidad. También ayudan a la parte de pruebas para generar un contexto del uso del sistema, y tener una idea del comportamiento y resultados de acuerdo con la funcionalidad descrita.

Se describen los principales casos de uso del sistema. Se omiten varios casos de uso por cuestiones de privacidad y seguridad.

Caso de uso: Inicio de sesión

Narrativa

Los actores necesitan generar una sesión en el sistema para hacer las acciones asignadas de acuerdo con su perfil. Se valida el horario de servicio del módulo emisor, la contraseña asignada al usuario y también la huella dactilar que quede almacenada por usuario. El flujo se describe en la Ilustración 4.

Actores

- Administrador
- Capturista
- Supervisor de módulo
- Auditor

Diagrama UML

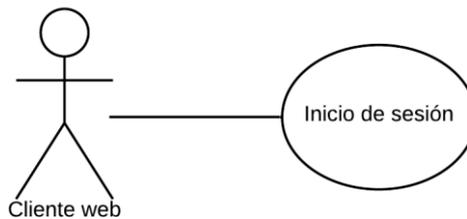


Ilustración 3 - Diagrama UML de Inicio de sesión.

Flujo básico

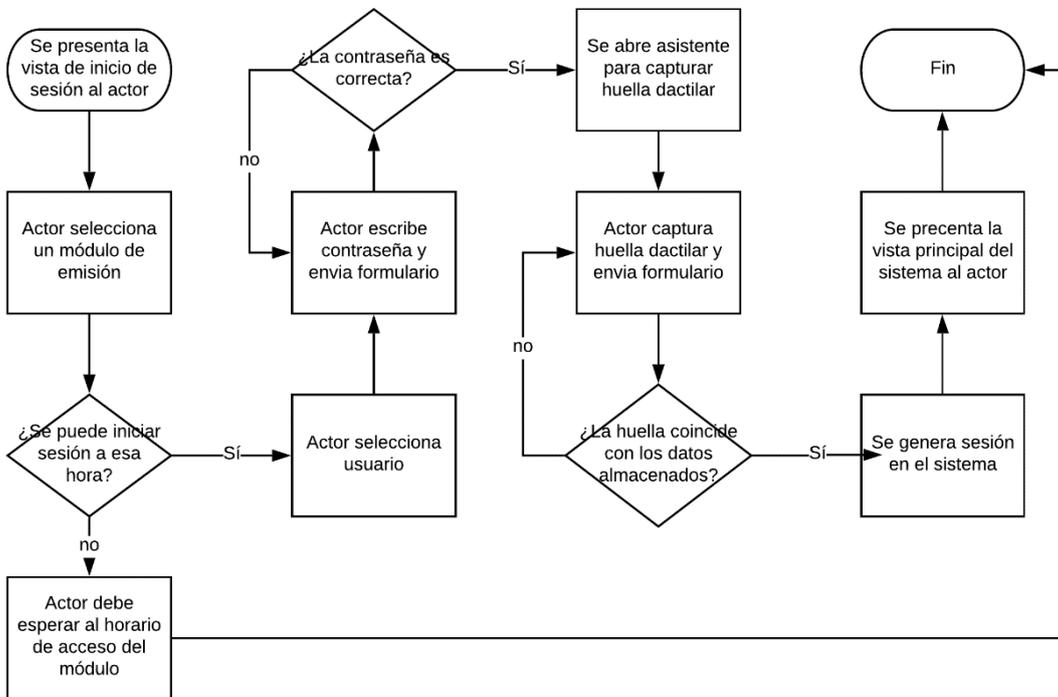


Ilustración 4 - Diagrama de flujo de Inicio de sesión.

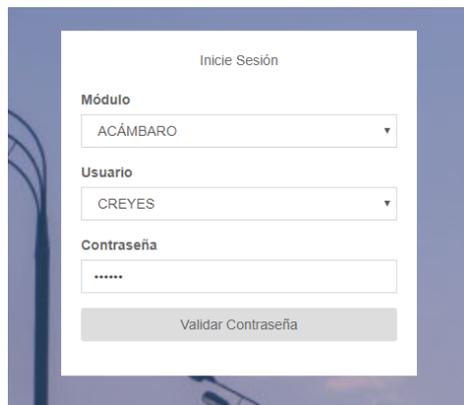
### Errores

- La contraseña ingresada por el actor puede ser incorrecta, se notificará con una alerta indicando el error.
- Si el actor envía el formulario de huella dactilar sin información, se notificará con una alerta indicando el error.
- La huella dactilar ingresada no coincide con la del usuario seleccionado, se enviará una alerta indicando el error.

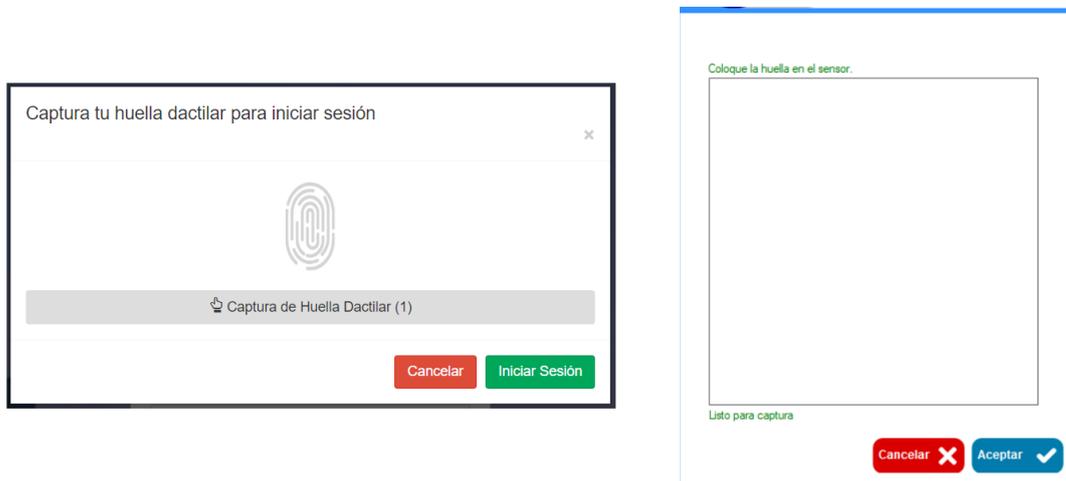
### Excepciones

- Si el proceso de la base de datos se encuentra fuera de servicio la vista se cargará sin ningún módulo de emisión, sin permitir la creación de sesiones por el sistema.

### Ilustraciones



**Ilustración 5 - Formulario de inicio de sesión. El usuario selecciona el módulo, posteriormente su usuario y escribe la contraseña.**



**Ilustración 6 - Formulario de captura de huella dactilar. Si la contraseña del usuario es correcta, podrá capturar su huella e iniciar sesión.**

### Caso de uso: Administración de usuarios

#### Narrativa

Esta parte del sistema se encarga de la creación, edición, habilitación e inhabilitación de los usuarios. Para los usuarios se almacena su nombre, alias, una contraseña, dos huellas dactilares.

También se le asocia un perfil y un módulo de emisión. Los movimientos en esta parte sólo pueden ser realizados por un administrador. El flujo se describe en la *Ilustración 8*.

Actores

- Administrador

Diagrama UML

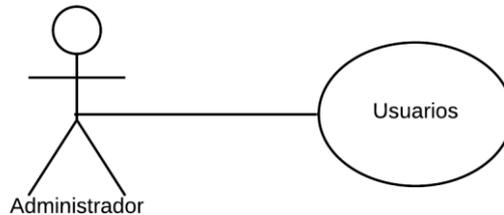


Ilustración 7 - Diagrama UML de Administración de usuarios.

Flujo básico

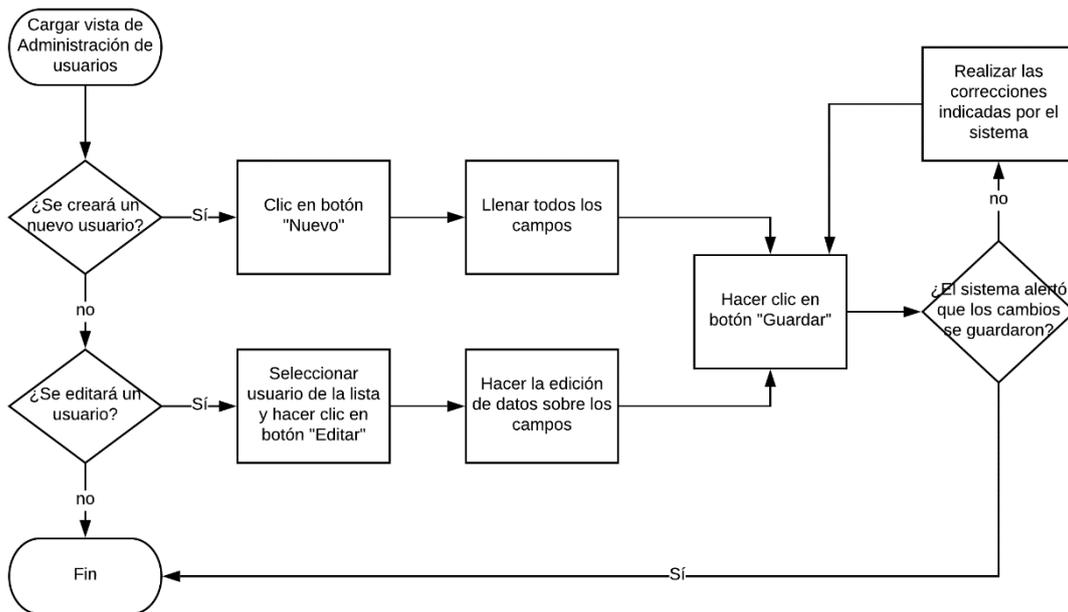
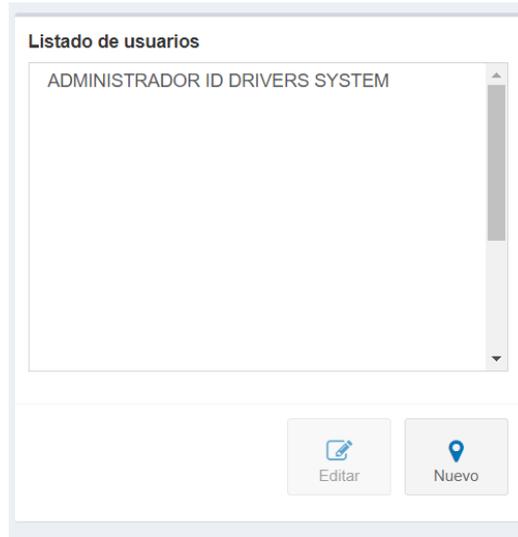


Ilustración 8 - Diagrama de flujo de Administración de usuarios.

Errores

- El nombre de usuario se repite.
- La contraseña ingresada no cumple con los requisitos mínimos de seguridad.
- Al guardar los datos no se llenaron de forma correcta todos los formularios.

Ilustraciones



**Ilustración 9 – Panel de listado de usuarios. Para hacer un nuevo usuario solo debe hacer clic en nuevo. Si desea editar debe seleccionar el usuario previamente.**

**Ilustración 10 - Formulario de datos de usuarios. Este formulario contiene toda la información que se necesita de los usuarios.**

*Caso de uso: Administración de módulos de emisión*

Narrativa

Los módulos de emisión son las locaciones donde se captura e imprime las licencias y permisos. Se tiene un control de estos módulos dado que pueden abrir nuevos o cerrar viejos. Las licencias y usuarios, así como sus movimientos, son asociadas a un módulo. Así se tiene una relación de quien hizo el registro de la licencia y el lugar donde se expidió. El flujo para crear o editar un módulo se describe en la *Ilustración 12*.

Actores

- Administrador

Diagrama UML

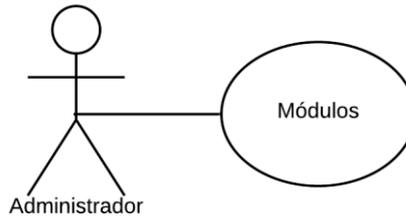


Ilustración 11 - Diagrama UML de Administración de módulos.

Flujo básico

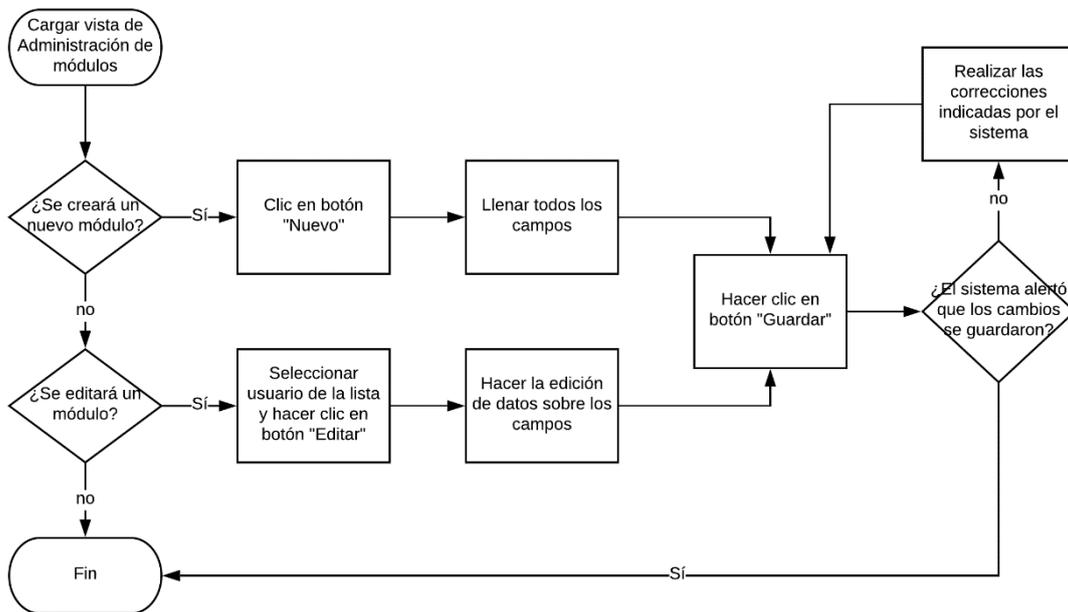


Ilustración 12 - Diagrama de flujo de Administración de módulos.

Errores

- El nombre del módulo se repite con un registro previo.
- El formato de los campos de hora entrada u hora salida no es el correcto.

Ilustraciones

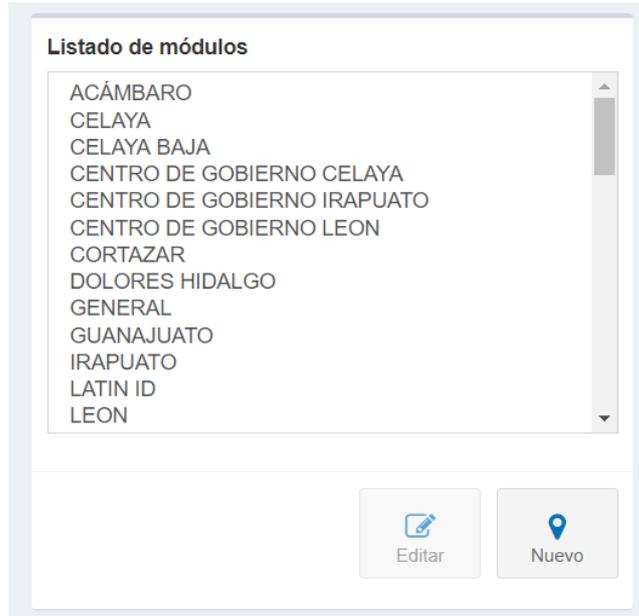


Ilustración 13 – Listado de módulos en el formulario de administración de módulos de emisión.

Información del módulo

<b>Nombre del módulo</b>	<b>Encargado del módulo</b>
T ACÁMBARO	T INACTIVO
<b>Prefijo</b>	<b>Consecutivo</b>
T ME	T 200
<b>Hora de Entrada (24 horas)</b>	<b>Hora de Salida (24 horas)</b>
🕒 8:30:00	🕒 20:00:00
<input type="checkbox"/> Activo	
<input type="button" value="Cancelar"/> <input type="button" value="Guardar"/>	

Ilustración 14 - Formulario de registro y edición de módulos. Esta es la información que se guarda acerca de un módulo de emisión.

Caso de uso: Administración de alergias

Narrativa

Para los conductores que sufren de alergias deben de decir cuales son en el registro de la licencia. Para esto se tiene un catálogo de dichas alergias. Pero como cabe la posibilidad de que el listado de alergias que se tiene no tenga todas las posibles alergias, por eso hay un segmento del sistema que se encarga de generar nuevos registros de alergias o editar los registros previos. Los pasos para crear o editar una alergia se describen en la *Ilustración 16*.

Actores

- Administrador

Diagrama UML

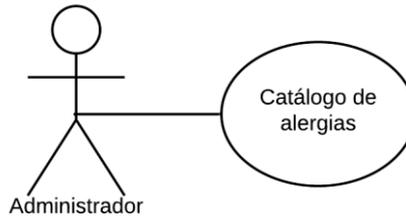


Ilustración 15 - Diagrama UML de Administración del catálogo de alergias.

Flujo básico

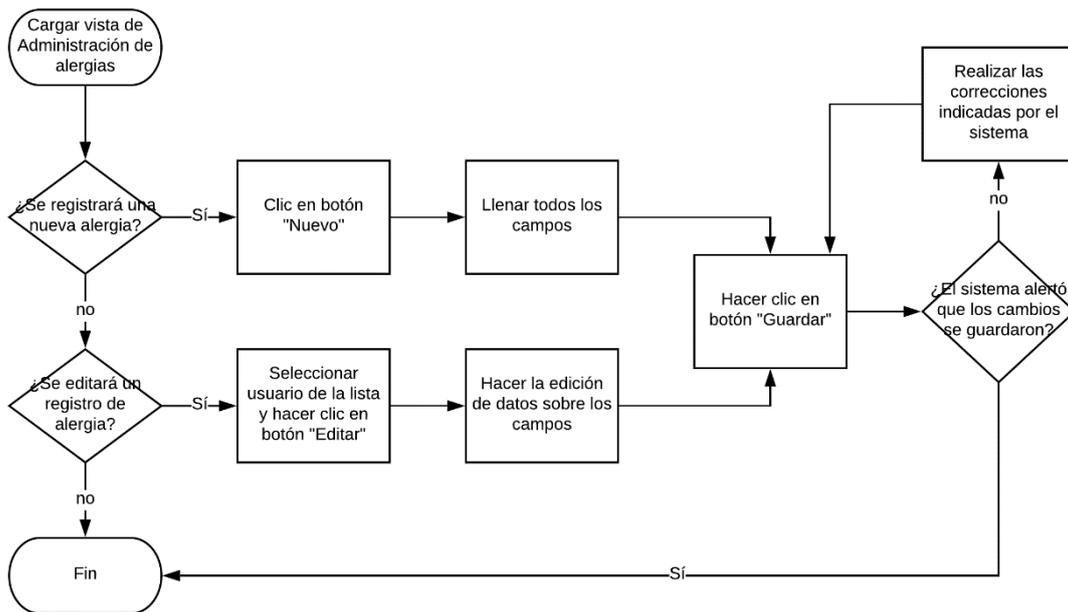


Ilustración 16 - Diagrama de flujo del Catálogo de alergias.

Errores

- Se registre una alergia ya existente.

Ilustraciones

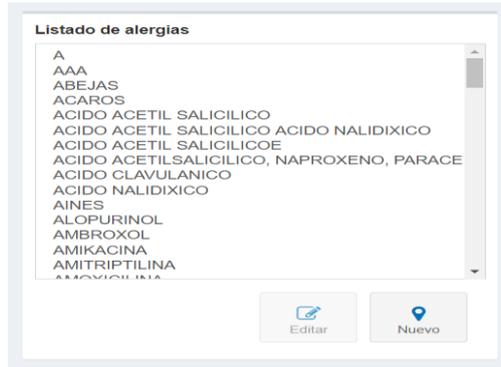


Ilustración 17 - Panel de listado de alergias.

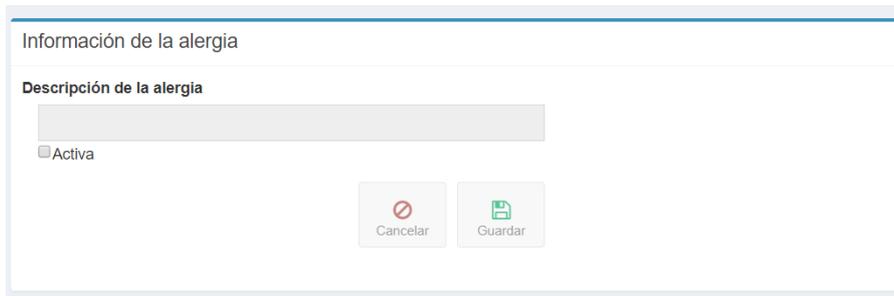


Ilustración 18 - Formulario del catálogo de alergias. Respecto a las alergias, solo es necesario almacenar el nombre y si es un registro activo.

Caso de uso: Nueva licencia

Narrativa

Una nueva licencia es la primera emisión de un tipo de licencia para una persona. Después de que la persona termina este trámite, tendrá acceso a un duplicado y una renovación del mismo tipo de licencia.

El proceso de búsqueda de conductor se describe en la *Ilustración 20* y la captura de datos se describe en la *Ilustración 21*.

Este caso se divide en tres partes:

1. Búsqueda de conductor: Se realiza una búsqueda del conductor en la base de datos del sistema para comprobar que el solicitante no cuenta con un registro previo de ese tipo. Si la persona no está registrada en la base de datos, se hace una petición a un servicio web<sup>4</sup> que provee RENAPO<sup>5</sup> para obtener los datos del conductor. De no tener como respuesta los datos del conductor, el capturista deberá ingresar los datos que el solicitante verificando la identidad a través de una identificación oficial. A través de otro servicio web se verifica que el solicitante haya aprobado un examen teórico de conocimientos viales, de no aprobarlo deberá aplicarlo nuevamente. Todo eso pasa en la búsqueda inicial.

<sup>4</sup> Servicio informático que recibe peticiones a través de un protocolo de internet. Recibe y responde datos con formato estandarizado.

<sup>5</sup> Registro nacional de Población.

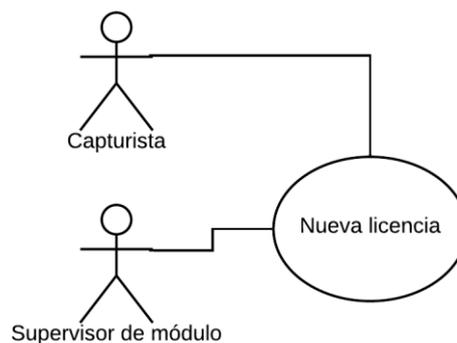
2. Captura de datos: El capturista llena una serie de campos en un formulario con información del conductor (información personal, domicilio e información de media filiación<sup>6</sup>). Como se muestra en la *Ilustración 25*, la mayor parte de campos en la sección de “Datos de la solicitud” están deshabilitados, porque son datos que se obtuvieron en la búsqueda del conductor o información de una emisión de licencia previa de la persona. Los campos “lugar de la prótesis” y el “detalle de otra restricción” se activan según el valor elegido en el campo “restricciones”. En la *Ilustración 27* está la sección de “Donador”, donde los datos se almacenan para posibles accidentes. Se guarda el tipo de sangre y a que es alérgico la persona. Así como un contacto de emergencia.
3. Captura de biometría: En este caso consiste en guardar ciertos datos biológicos propios de la persona en imágenes. Si bien el almacenamiento de biometría no solo es en imágenes, en este sistema solo fue requerido de esta forma.
  - a. Captura facial, se muestra en la *Ilustración 28*. Cuando se haga la toma se recortará la imagen a la parte que este dentro del rectángulo rojo. El software reconoce si hay rostros.
  - b. Captura de dos huellas dactilares
  - c. Captura de firma. Se muestra en la *Ilustración 29*. A través del tablero de firmas el conductor realizará su firma y se almacenará en digital. Esta aparecerá en la licencia impresa.

El proceso de impresión se maneja como parte aislada del registro de datos. Sin embargo, una vez concluido el paso 3, se tiene acceso a la visualización digital de la licencia como se muestra en la *Ilustración 31*.

#### Actores

- Capturista
- Supervisor de módulo

#### Diagrama UML



**Ilustración 19 - Diagrama UML de Nueva licencia.**

<sup>6</sup> Descripción física de la persona.

Flujo básico

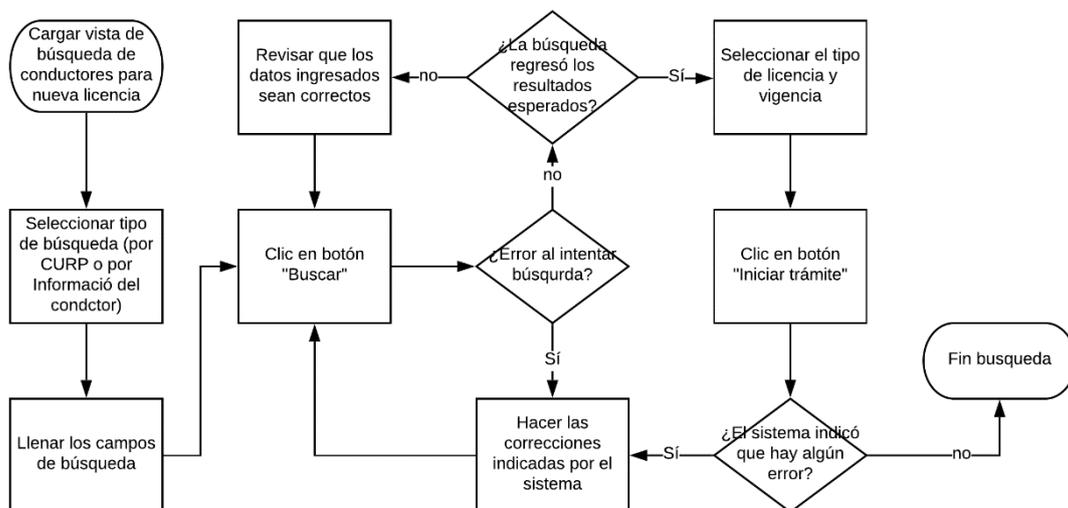


Ilustración 20 - Diagrama de flujo de Búsqueda de nueva licencia.

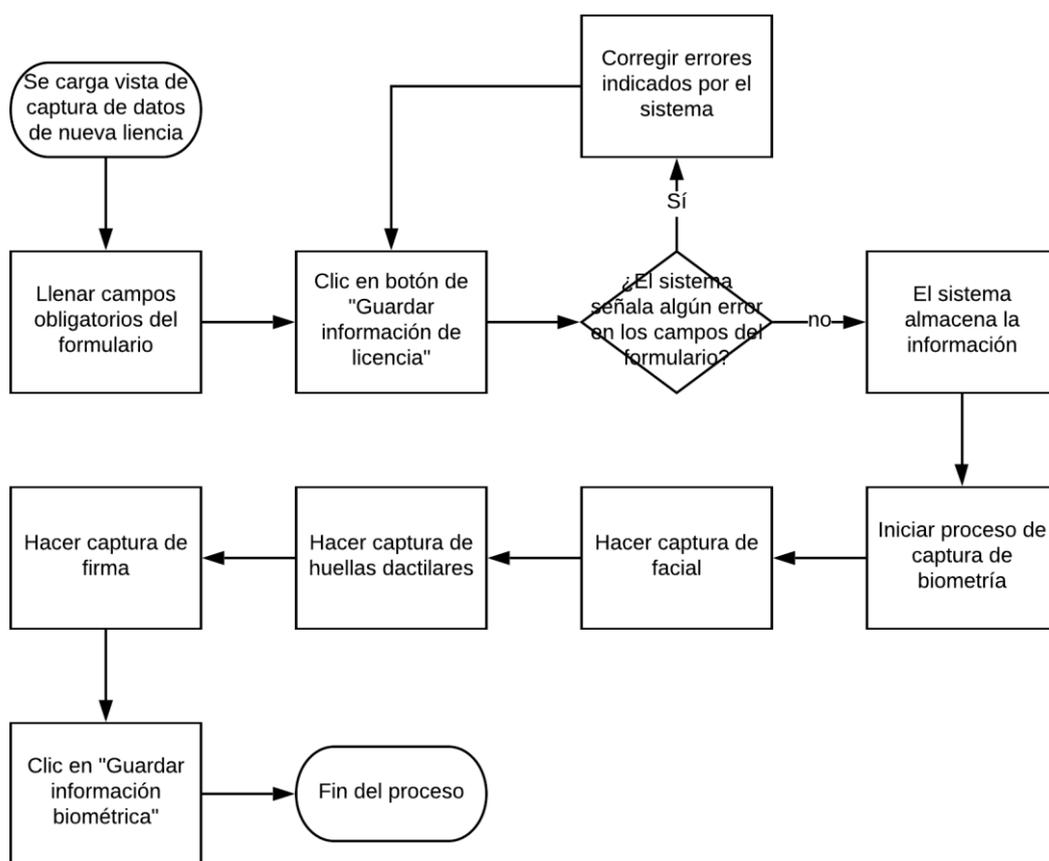


Ilustración 21 - Diagrama de flujo de Captura de datos de licencia.

Errores

- Errores en los campos en el formulario de búsqueda o de registro de licencia.

Ilustraciones

Por Datos del Conductor:
  Por CURP:

**Fecha de Nacimiento \***

**Nombre \***

**Apellido Paterno \***

**Apellido Materno \***

**Sexo \***

**CURP \***

El solicitante cuenta con licencia de otro estado

**Ilustración 22 - Formulario de Búsqueda de nueva licencia. Es lo primero que debe hacer el capturista para iniciar el trámite.**

Conductor



**Trámites anteriores**

AUTOMOVILISTA

---

Trámites disponibles

**Elige un trámite**

**Elige una vigencia**

**Ilustración 23 - Panel de descripción de conductor en el formulario de búsqueda de nueva licencia.**

Datos de la solicitud				
No. Solicitud	No. de Licencia	Tipo Licencia	Vigencia (años)	Tipo de movimiento
<input type="text"/>	CS1881930033	PERMISO AUTOMOVILIS' ▾	3 AÑOS ▾	NUEVA ▾
Fecha de Expedición	Fecha de Vencimiento	Fecha de Antigüedad	Costo	
31/05/2019	29/05/2022	29/05/2019	200.0000	

**Ilustración 24 – Datos de solicitud en el formulario de registro de licencia. Estos campos no son editables por el usuario, son llenados de forma automática.**

Información personal			
<b>Nombre Completo</b> CARLOS	<b>Apellido Paterno</b> REYES	<b>Apellido Materno</b> ANGELES	<b>Sexo</b> MASCULINO
<b>Fecha de Nacimiento</b> 29/05/2004	<b>Originario de</b> JALISCO	<b>Teléfono*</b>	<b>Email</b> correo@ejemplo.com
<b>RFC</b> REAC940828	<b>CURP</b> REAC940828HDFYNR02	<b>Migrante</b> No	<b>Fecha de Permiso RFM</b> dd/mm/yyyy
<b>Restricciones</b>	<b>Lugar de Prótesis</b>	<b>Detalle de Otra Restricción</b>	<b>Autorización UAIP</b> No

Ilustración 25 – Información personal del conductor en el formulario de registro de licencia.

Señas particulares					
<b>Complexion*</b> DELGADO	<b>Color de piel*</b> MORENO	<b>Color de cabello*</b> NEGRO	<b>Frente*</b> ANCHO	<b>Cejas*</b> POBLADAS	<b>Ojos*</b> CAFE
<b>Nariz*</b> ANCHA	<b>Boca*</b> GRANDE	<b>Mentón*</b> OVALADO	<b>Estatura (en centímetros)*</b> 10		
<b>Peso (en kilogramos)*</b> 10	<b>Seña Particular</b>		<b>Observaciones</b>		

Ilustración 26 – Señas particulares del conductor en el formulario de registro de licencia. A través de los campos de esta sección el capturista debe describir al conductor.

Donador			
<b>Donador</b> No	<b>Tipo Sanguíneo*</b> O+	<b>Contacto emergencia*</b> Nombre del Contacto	<b>Teléfono emergencia*</b>
<b>Alergias</b>	<b>Medicamento especial</b>		
<b>Examen Visual</b> Aprobado	<b>Examen Práctico</b> Aprobado	<b>Examen Teórico</b> Aprobado	

Ilustración 27 – Información para casos de emergencia del conductor en el formulario de registro de licencia.



Ilustración 28 - Interfaz de captura facial.



Ilustración 29 - Interfaz de captura de firma.

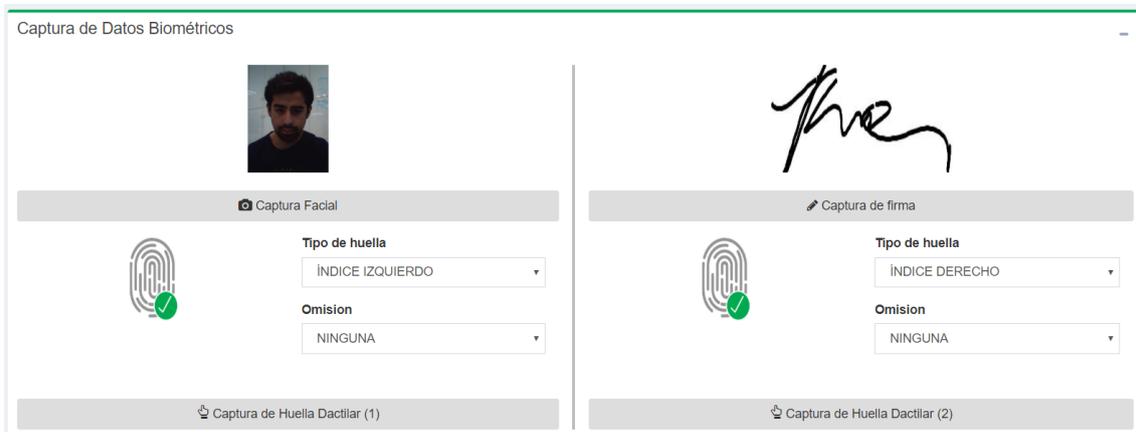


Ilustración 30 - Formulario de captura de biometría. Para iniciar la captura biométrica se usa esta sección. Donde a modo de formulario y resumen de capturas, se muestran las visualizaciones de las tomas.

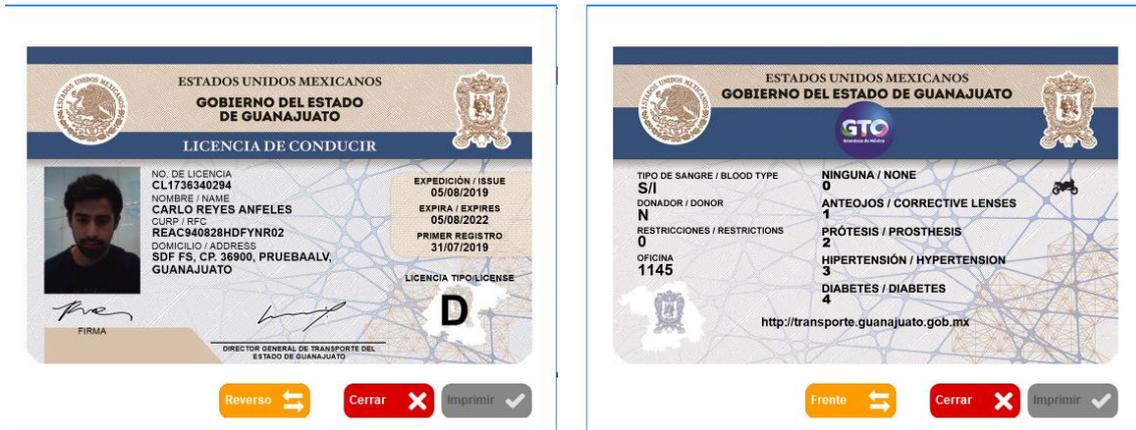


Ilustración 31 - Visualización de la licencia previo a la impresión.

Caso de uso: Duplicado licencia

Narrativa

Cuando un conductor ha hecho un trámite de licencia y tal licencia se encuentra vigente, el conductor puede realizar un duplicado de esta. Esto por los casos de extravío o robo del documento.

El proceso para la emisión es el siguiente:

1. Búsqueda del conductor en el sistema
2. Confirmación de datos (esta acción guarda el nuevo registro copiando los datos del registro anterior)
3. Impresión de licencia

Se describe el proceso de duplicado en la *ilustración 33*.

En el duplicado no existe la toma de biometría porque se copian los datos de la licencia anterior.

Actores

- Capturista
- Supervisor de módulo

Diagrama UML

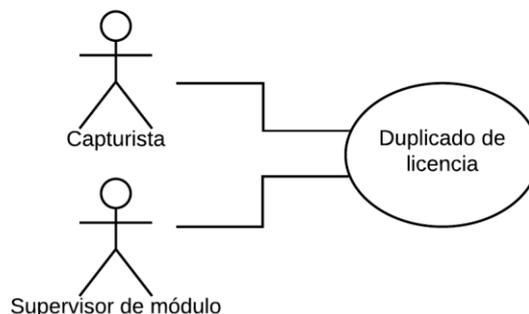


Ilustración 32 - Diagrama UML de Duplicado de licencia.

Flujo básico

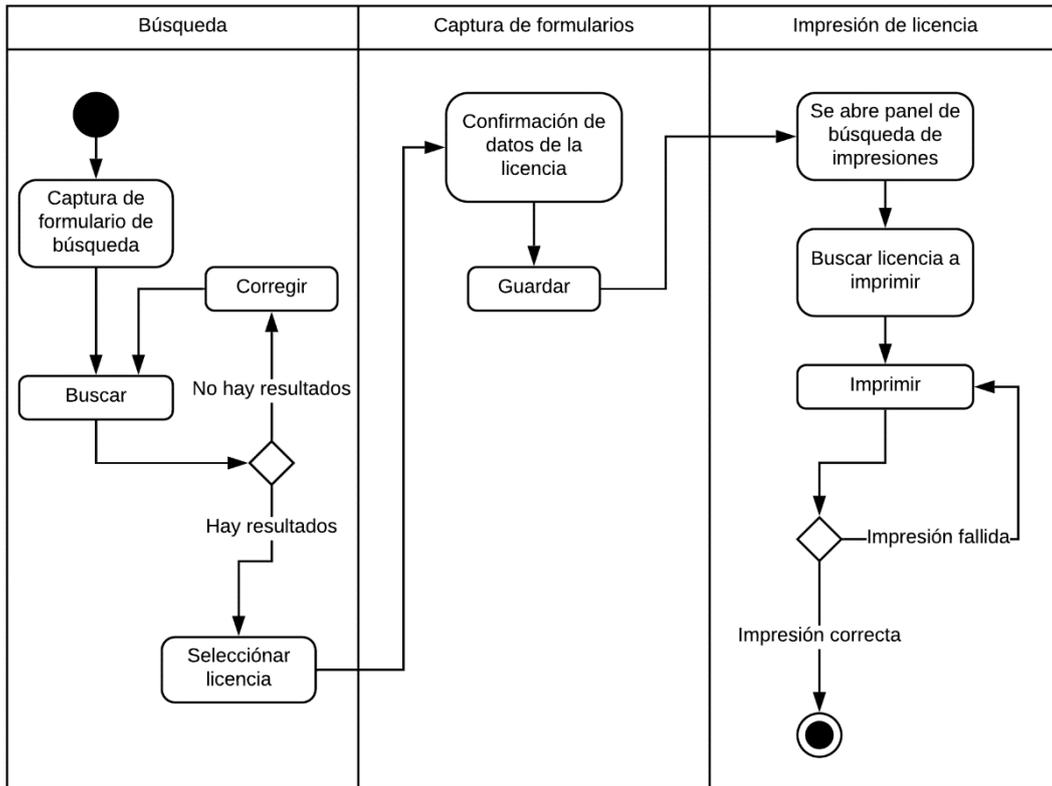


Ilustración 33 - Diagrama de flujo de un duplicado de licencia.

Errores

- Al guardar el duplicado de la licencia es posible que surja un error en los archivos de biometría. De ser el caso el sistema forzará al usuario a recapturar los datos biométricos de la persona.

Ilustraciones

### Duplicado de licencia

Búsqueda en el Sistema

Por Datos del Conductor:

**Fecha de Nacimiento \***

**Nombre \***

**Apellido Paterno \***

**Apellido Materno \***

**Sexo \***

Por CURP:

**CURP \***

Por No.de Licencia:

**No. de Licencia \***

Por No.de Solicitud:

**No. de Solicitud \***

**Ilustración 34 - Formulario de búsqueda para emisión de duplicado de licencia.**

Resultados

Conductor	Licencia	Tipo de Licencia
CARLO REYES ANFELES	CL1736340294	CHOFER DE SERVICIO DE CARGA

Datos de Trámite



Licencia Tipo	C
Vigencia	2 AÑOS
Último Trámite	NUEVA
Fecha Expedición	02/08/2019
Fecha Vencimiento	02/08/2021
Estatus	VIGENTE
Estatus de Captura	LICENCIA IMPRESA

Duplicado por cambio de domicilio



**Ilustración 35 – Panel de resultados el formulario de búsqueda de licencias.**

*Caso de uso: Renovación de licencia*

Narrativa

La renovación aplica cuando un conductor tiene una licencia no vigente o desea aplazar la fecha de vencimiento de su licencia actual, cancelando la vieja y haciendo el nuevo registro.

El proceso para la emisión es el siguiente:

1. Búsqueda del conductor en el sistema

2. Confirmación de datos (esta acción guarda el nuevo registro)
3. Captura de biometría
4. Impresión de licencia

En la *Ilustración 37* se describe un poco más detallado el proceso de renovación de licencia.

Los permisos de conducir no pueden ser renovados, la vigencia de estos es la fecha de la mayoría de edad del conductor. Por eso no pueden ser renovados.

Actores

- Capturista
- Supervisor de módulo

Diagrama UML

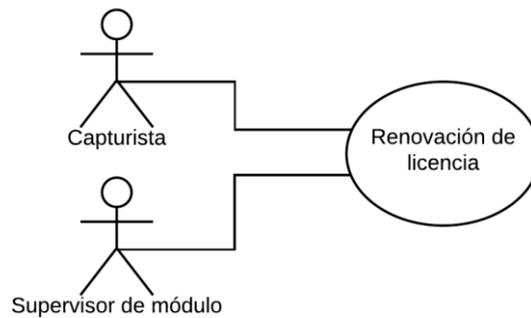


Ilustración 36 - Diagrama UML de Renovación de licencia.

Flujo básico

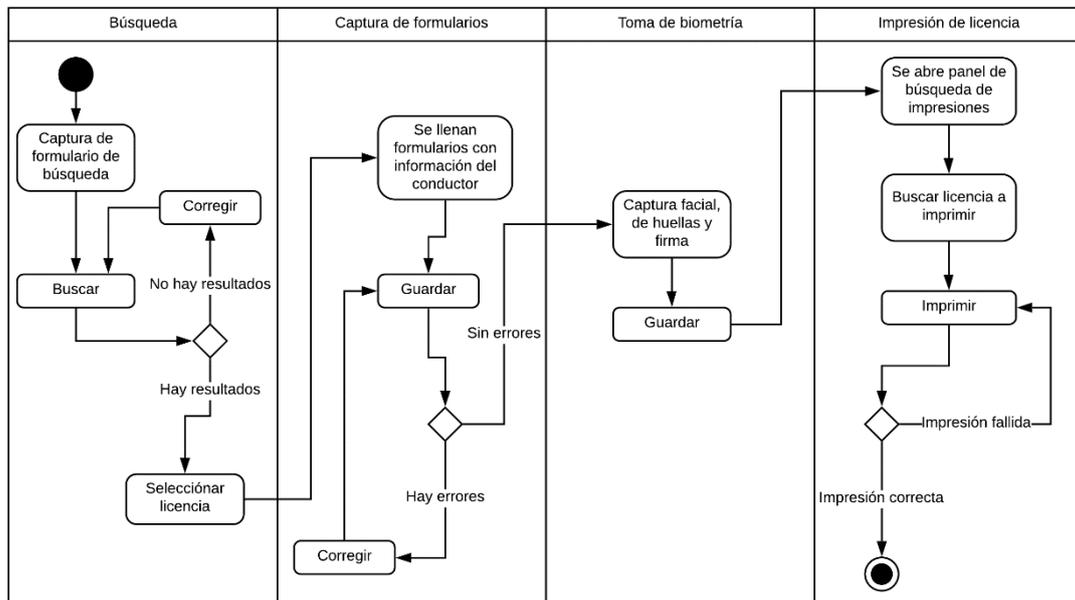


Ilustración 37 - Diagrama de flujo de una renovación de licencia.

## Errores

- Errores en los campos en el formulario de búsqueda o de registro de licencia.

## Ilustraciones

Revisar las *Ilustraciones 24, 25, 26, 27, 28, 29 y 30* que pertenecen también al flujo de renovación de licencia.

**Ilustración 38 - Formulario de búsqueda de conductores para Renovación de licencia.**

## Caso de uso: Edición de licencia

### Narrativa

La edición de licencia es un proceso que sirve para editar algunas características de una licencia que esté en proceso de emisión, es decir que no han sido impresas. Los registros de las licencias que están impresas no pueden ser editados.

El proceso de edición es el siguiente:

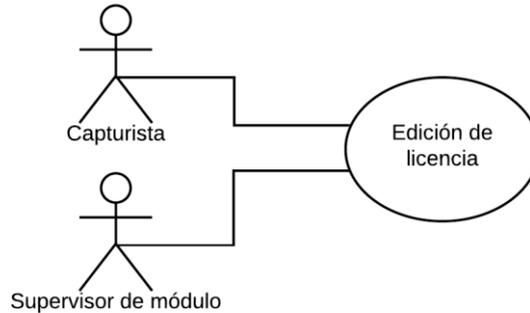
1. Se realiza una búsqueda de la licencia o del conductor. Ver *Ilustración 34* e *Ilustración 35*.
2. El supervisor del módulo debe autorizar la edición con un proceso de identificación vía huella dactilar. Un capturista no puede editar los datos de una licencia sin previa autorización. Esto es para controlar y proteger los datos de los contribuyentes. Ver *Ilustración 41*. La autorización se hace a través de la huella dactilar. Con la toma de la huella que se hace a través del dispositivo lector, se obtienen los detalles de las huellas y se compara con las huellas de los supervisores. De haber coincidencia se permite la edición de la mayoría de los campos. Ver *Ilustración 42*.
3. Se hacen los cambios en los formularios o se toma nuevamente la información biométrica.

El flujo se describe en la *Ilustración 40*.

Actores

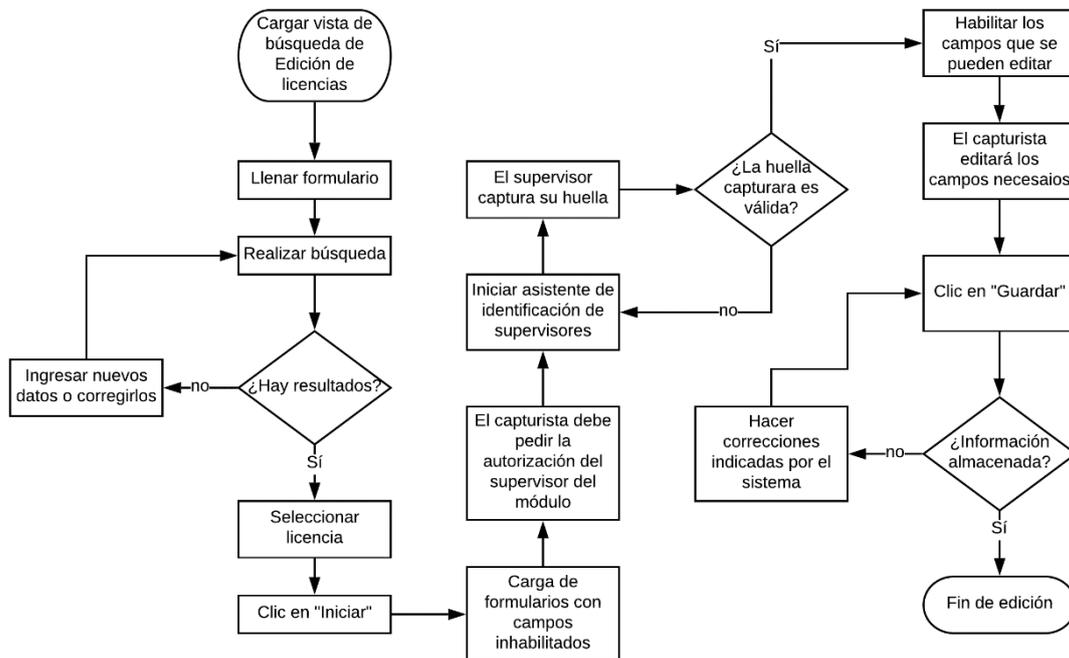
- Capturista
- Supervisor de módulo

Diagrama UML



**Ilustración 39 - Diagrama UML de Edición de licencia.**

Flujo básico



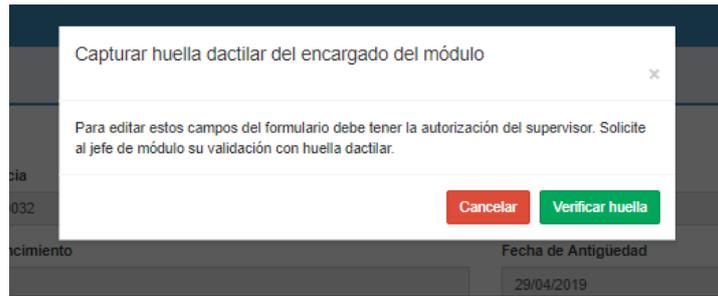
**Ilustración 40 - Diagrama de flujo de una edición de licencia.**

Errores

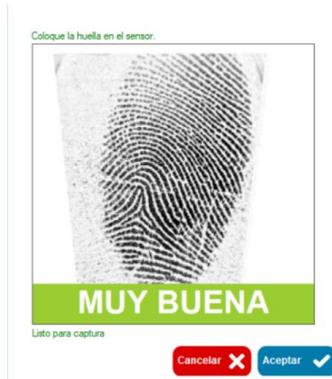
- Errores en la captura de campos en los formularios

Ilustraciones

Revisar las *Ilustraciones 24, 25, 26, 27, 28, 29 y 30* que pertenecen también al flujo de edición de licencia.



**Ilustración 41 - Aviso al usuario para que pueda editar los campos registrados.**



**Ilustración 42 - Asistente para captura de huella.**

## 4. Arquitectura y diseño

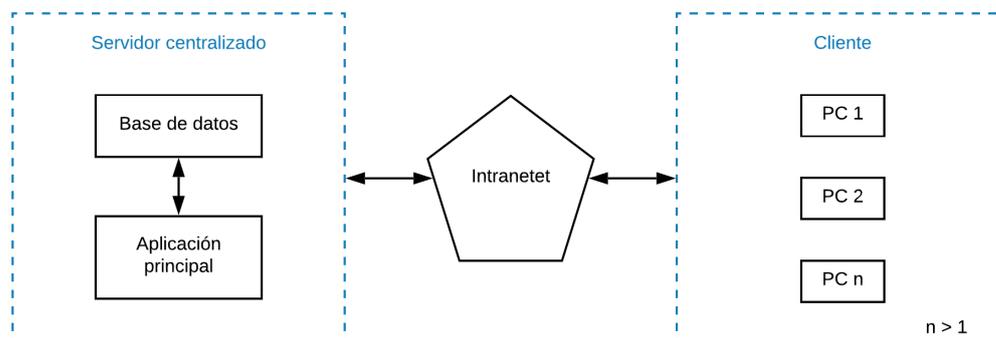
La arquitectura elegida fue Cliente-Servidor. Describiendo la arquitectura:

- Servidor. Es una o varias aplicaciones que pueden o no trabajar en conjunto y que están en ejecución en alguna computadora. Por lo tanto, afirmamos que un servidor es un proceso en ejecución. Este proceso tiene entradas y salidas, donde las entradas son las peticiones que hacen los clientes vía un puerto de servicio de red (siendo el de esta aplicación el 8080), y las salidas es la respuesta que se genera a dicha petición por el mismo puerto.
- Cliente. El cliente vive generalmente en una máquina distinta a la del servidor. En el caso de las aplicaciones web, el cliente son los navegadores web donde se interpretarán los documentos que el servidor envía para comenzar la interacción con el sistema.

Las principales ventajas son:

- El usuario del sistema puede hacer uso de este desde un navegador y no tiene que instalar más herramientas.
- Las actualizaciones del sistema no implican reinstalar componentes en las máquinas cliente.
- Se garantiza que todos los usuarios están usando la misma versión del sistema web.

Si lo vemos en un diagrama:



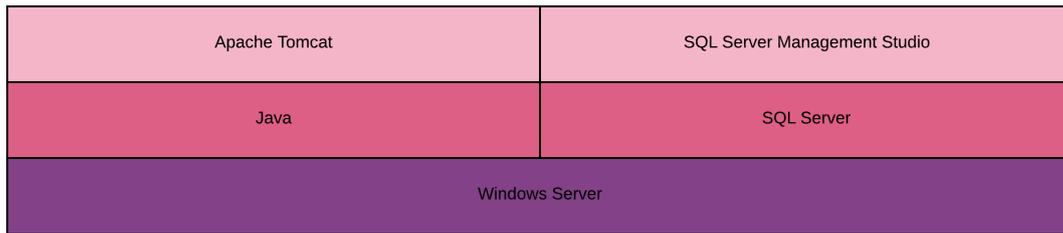
**Ilustración 43 - Representación de la arquitectura del sistema.**

La intranet es una red privada que funciona como el internet, pero solo hay comunicación con las máquinas que estén dentro de la red, por lo tanto, este sistema no cuenta con acceso desde internet.

### 4.1 Servidor

En la máquina servidor hay dos servicios que necesitan estar en ejecución para que el cliente pueda funcionar.

1. Manejador de base de datos.
2. Servidor de aplicaciones Java Apache Tomcat.



**Ilustración 44 - Software base en el servidor centralizado.**

#### *Base de datos*

Se sigue un modelo relacional para el almacenamiento de datos de la aplicación. Por otro lado, para el almacenamiento de archivos como imágenes y capturas biométricas se tiene un sistema de archivos que quedan referenciados a registros en base de datos.

Descripción de las principales tablas:

- Conductor: Almacena los datos del conductor que no cambian con el tiempo. Por ejemplo, el nombre, apellidos y fecha de nacimiento.
- Licencia: Se guardan los datos del trámite, como fecha de vencimiento, número de licencia, tipo de trámite, entre otros.
- Foto: Se guarda la información referente a la foto del solicitante y su licencia.
- Bitácora: Se guardan los movimientos que realizan todos los usuarios mientras este tenga una sesión abierta.

Algunas tablas que funcionan como catálogos son:

- estado
- municipio
- colonia
- estatus\_licencia
- modulo
- perfil

El siguiente diagrama muestra algunas tablas del sistema, sin embargo, solamente tienen el atributo id y no se muestran demás campos que las conforman.

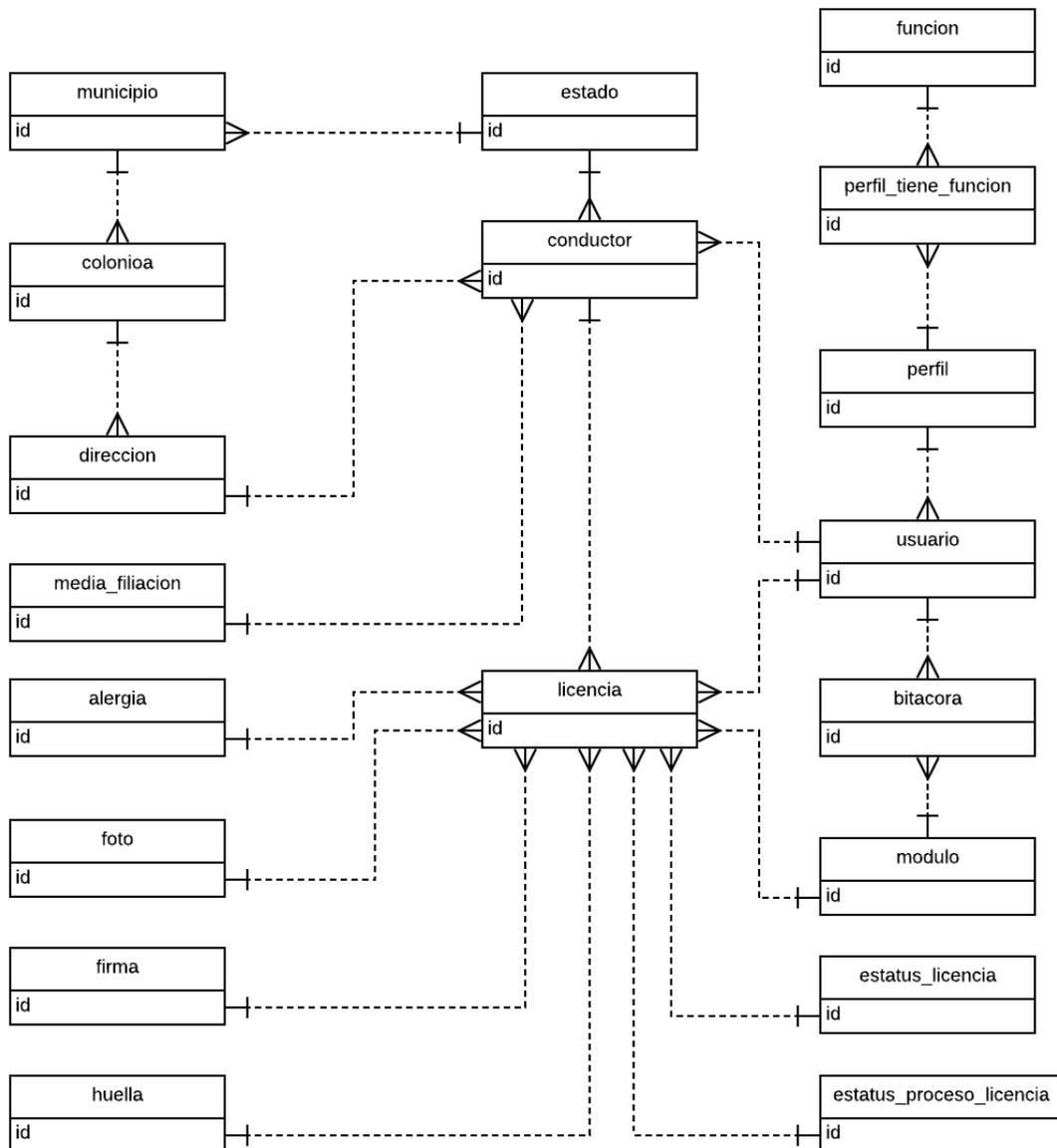


Ilustración 45 - Diagrama entidad relación de la base de datos.

### Aplicación principal (Back-end)

Back-end quiere decir que, esta parte de la aplicación es ejecutada por el servidor principal. Todo el código back-end en el caso de esta aplicación está escrito en el lenguaje de programación Java en su versión 8 y en la plataforma Java EE<sup>7</sup>. Esta edición de Java tiene soporte para JavaServer Pages que es la tecnología que más se usa en el caso de proyectos web. A continuación, tenemos una ilustración de los niveles de la aplicación en base a las tecnologías con las que está construida.

<sup>7</sup> Es la plataforma que proporciona Java para el desarrollo de aplicaciones enfocadas a la arquitectura Cliente-Servidor. Las aplicaciones suelen ejecutarse en servidores de aplicaciones que proveen una interfaz de acceso con uno o más clientes.

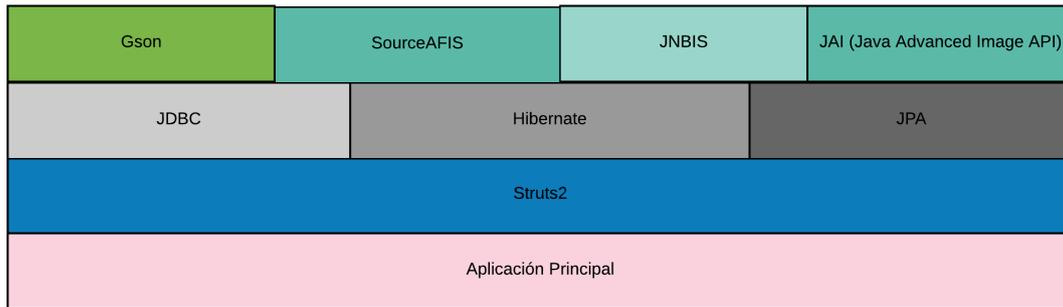


Ilustración 46 - Estructura back-end.

### Java

Se eligió este lenguaje de programación porque tiene mucha documentación, y provee herramientas para usar esquemas de trabajo basados en el patrón de diseño MVC, lo cual facilitó el diseño y escritura de código del programador. Además, permite hacer aplicaciones web muy robustas.

### JDBC, Hibernate y JPA

El conjunto de estas tres API<sup>8</sup> de Java se usa para el almacenamiento y acceso a datos que se encuentran en la base de datos. Hibernate es la parte del sistema que mapea una serie de tablas pertenecientes a una base de datos a una serie de objetos Java, dichos objetos son los denominados POJO's (Plain Older Java Object), que son clases que contienen solo los atributos, constructores y métodos get y set de los atributos. Si bien el paradigma orientado a objetos tiene diferencias en cuanto al paradigma de las bases de datos relacionales, es posible generar una representación del esquema o modelo de la base de datos (o varias) que se usen en nuestra aplicación con objetos de Java. JDBC hace el manejo de conexiones y manejo de las transacciones con la base de datos. JPA se encarga del manejo de entidades apoyándose principalmente en archivos XML<sup>9</sup> y anotaciones en las clases de Java.

### Struts2

Es un framework<sup>10</sup> para sistemas web, hecho para el lenguaje Java y provee las herramientas para generar sistemas basados en el patrón de diseño Modelo Vista Controlador. Este framework ofrece dos beneficios muy grandes, la primera es que cuenta con una forma muy sencilla para el procesamiento de formularios, y la segunda ventaja es la facilidad que tiene para generar llamadas AJAX<sup>11</sup>. A si mismo ofrece un acoplamiento muy bueno con el framework Hibernate, con el cual compartirá gran parte de código.

<sup>8</sup> API (Application Programming Interface). Las API son programas que el desarrollador suele implementar en los desarrollos, usando como herramientas las funciones que provee la API. Por ejemplo, la API JDBC ayuda a proveer una conexión desde un programa Java y una base de datos.

<sup>9</sup> XML es un formato de texto para la escritura de datos

<sup>10</sup> Es un esquema de trabajo que ayuda al desarrollador dándole herramientas de organización y estructura siguiendo un patrón marcado por el esquema.

<sup>11</sup> Asynchronous JavaScript and XML. Técnica de desarrollo web, que permite trabajar con distintos elementos de forma asíncrona sin recargar la página.

### Otras API

Las siguientes herramientas tienen una Licencia Apache, la cual permite uso comercial, modificación, distribución, uso privado entre otras condiciones y limitaciones. La licencia Apache indica que son opensource<sup>12</sup>.

- SourceAFIS. Es un algoritmo de comparación de huellas dactilares a través de imágenes. Tiene una implementación como biblioteca Java y es necesario el uso de tal biblioteca especialmente para el inicio de sesión.
- Java Implementation of NIST Biometric Image Software (NBIS). Es una biblioteca escrita en Java para poder decodificar las imágenes con formato WSQ. En este formato se guardan las imágenes que se toman de las huellas dactilares.
- GSON. Es una biblioteca de Google hecha en Java. La principal función de esta API es convertir objetos Java en una cadena de texto con formato JSON<sup>13</sup>, y de forma inversa, convertir cadenas de texto con formato JSON a instanciar objetos Java. Esta API se usa ampliamente en toda la aplicación principal, dado que el contenido de las peticiones y resultados del servidor se hacen usando el formato JSON. Entre las mayores ventajas que ofrece esta API es el gran acoplamiento con la API Collections de Java, además de que permite generar nuevas formas de implementación para los diferentes tipos de datos o estructura de datos que se puedan usar en las distintas implementaciones.

### 4.2 Cliente

El sistema del lado del cliente se usa con el navegador Google Chrome por la capacidad de usar extensiones web y usar el protocolo Native Messaging para la comunicación con aplicaciones fuera del navegador. Por eso también es necesario que el sistema operativo donde trabaje el cliente se a Windows, dado que hay 4 subsistemas que se comunican con el navegador para el funcionamiento del sistema.

En el siguiente esquema se muestra la composición del sistema en la plataforma web.

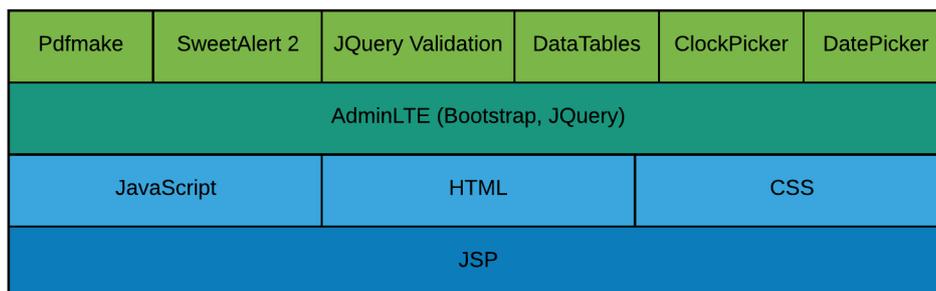


Ilustración 47 - Estructura del sistema en el front-ent.

<sup>12</sup> Opensource indica que el desarrollo de una aplicación o API está abierta a colaboración de todos así como el uso libre de la misma, salvo algunas restricciones que indique el tipo de licencia.

<sup>13</sup> JavaScript Object Notation. Es una cadena de texto que contiene datos con un formato basado en los objetos del lenguaje de programación JavaScript.

### Application principal (front-end)

#### JSP (JavaServer Pages)

Es la tecnología que desarrollada para Java usada generar contenido web HTML. Permite usar de forma embebida código Java, esto dará una capacidad de generación de código HTML de forma dinámica.

#### AdminLTE

Es una plantilla hecha con HTML que implementa el diseño de Bootstrap 3 y JQuery. Se usó para hacer la interfaz gráfica de la aplicación principal. Gran cantidad de componentes de esta interfaz fueron usados, además de que al ser una implementación de Bootstrap 3 se tiene un diseño responsivo a los distintos tamaños de pantalla que el usuario pueda usar.

#### Pdfmake

Es una biblioteca escrita en JavaScript y se usa para generar archivos tipo PDF de forma sencilla y estructurada. Los documentos se generan desde el navegador y permite la impresión. Esta es una biblioteca de mucha utilidad, ya que la generación de documentos de forma dinámica se hace muy sencilla gracias a esta herramienta.

#### SweetAlert2

Para los mensajes y notificaciones al usuario se usó SweetAlert2. Es una biblioteca JavaScript que se usa para generar mensajes con un estilo muy agradable y con una potencia muy grande. Tiene las clásicas alertas de éxito, error, alerta y pregunta.

#### JQuery Validation

Para el procesamiento de formularios en los navegadores web fue necesario hacer una validación de datos para verificar la consistencia de la información que llega al servidor y no se generen errores en el procesamiento de formularios o haciendo peticiones innecesarias al servidor. JQuery Validation ofrece una forma muy sencilla y ordenada para añadir varias validaciones a los campos en los formularios que tenga la aplicación.

#### ClockPicker y DatePicker

Son dos bibliotecas JavaScript, usadas para los campos en los formularios donde sea necesario ingresar datos que sean referentes al tiempo. En el caso de DatePicker se usa para que el usuario pueda ingresar una fecha con el formato "dd/MM/yyyy". ClockPicker se usa para que el usuario pueda ingresar datos que expresen horas con el formato "hh:mm:ss".

#### DataTables

El sistema cuenta con un apartado de reportes y una bitácora. En ambos se usa la librería JavaScript DataTables, para la organización de datos en formas de tablas y además sea fácil de entender la información para el usuario.

#### Subsistemas

En las computadoras cliente se debe instalar tres subsistemas que usan la plataforma de escritorio para el uso de hardware externo. El hardware con el que se trabaja es un lector de huella dactilar, una cámara web y un tablero de firmas. Para hacer la comunicación entre los subsistemas y el sistema principal que funciona en plataforma web es necesario usar el navegador Google Chrome.

#### Extensión web

Una extensión del navegador es un pequeño programa que el navegador ejecuta, permitiendo crear funciones extra al navegador. Generalmente están escritas en JavaScript y son ejecutadas

como un hilo hijo del navegador. Las principales ventajas de las extensiones es que puedes interactuar con los elementos que aparecen en los documentos HTML, por ejemplo, extrayendo un texto para usarlo de cualquier forma o llenando formularios usando datos de las cookies, u casi cualquier cosa que nos permita el navegador, incluso la comunicación con programas externos. Esto es para usar un protocolo de comunicación llamado Native Messaging.

Los navegadores web no tienen la capacidad de interactuar con hardware directamente. Este sistema usa 4 dispositivos periféricos en distintos procesos, por lo tanto, es necesario utilizar aplicaciones externas que trabajen con los dispositivos externos y el resultado sea entregado al navegador a través de Native Messaging, que es un protocolo de comunicación entre una aplicación que esté en procesamiento en la máquina cliente y una extensión en el navegador.

#### Captura de huella dactilar

Las características del software son:

- Las huellas deberán cumplir con el estándar ISO/IEC19794-4
- Las huellas deberán guardarse con el número de licencia del conductor de forma individual con el formato WSQ
- El software deberá permitir la omisión de cualquiera de las huellas, permitiendo seleccionar la causa y el dedo faltante.
  - Amputado
  - Imposibilitado

ISO/IEC 19794-4 especifica el cómo guardar, leer y la transmisión de las imágenes de uno o más dedos o incluso una palma de una mano humana.

#### Captura Facial

Las características del software son:

- El software realizara la verificación de calidad de imagen, cumpliendo con el estándar ISO/IEC 19794-5.
- Los archivos de fotografía deberán ser guardadas en formato JPEG.

ISO/IEC 19794-5 especifica el formato en que se guarda, como se lee y como se transmite la información de una o más imágenes faciales, las mejores prácticas para la fotografía de rostros, así como los atributos de la imagen digital en las tomas faciales.

#### Captura de firma

- El software deberá permitir que el capturista seleccione el grosor de la firma para una mejor calidad de esta.
- El software permitirá limpiar la firma para una nueva toma en caso de ser necesario.
- La firma deberá ser almacenada en formato TIFF con una dimensión de 485 x 105 pixeles.

### 4.3 Metodología de desarrollo

Se usó el Unified Process (Proceso Unificado) como metodología de desarrollo de software. Es un esquema de trabajo que ayuda a segmentar el proceso de ingeniería y desarrollo en fases definidas. Este esquema de trabajo indica que el proyecto es dirigido por los casos de uso, que en teoría son los modelos de los requerimientos funcionales del cliente. Para todos los casos de uso es aplicado un diseño, desarrollo, las pruebas y despliegue.

Unified Process se denomina como iterativo e incremental, basándonos en eso, el desarrollo de cada una de las partes del sistema es una iteración. Cuando se concluyen se concluye una iteración, se considerará como un incremento. De este modo se tiene una medición o referencia cuantitativa del avance de trabajo y el crecimiento general del proyecto.

El Unified Process se tiene que ver como un ciclo. Cada ciclo está confirmado por cuatro fases, las cuales se describen usando el sistema de emisión de licencias:

- Inicio. Se definió un alcance del sistema, pensando cuáles serán sus principales funciones, un plan genérico del proyecto y una estimación del costo en tiempo del proyecto. También se propone una arquitectura sobre la que funcionará el sistema.
- Elaboración. Se presentaron con detalle los casos de uso y se elaboran diseños para la comprensión del problema y la resolución que se le dará. Se segmentó el proyecto en subproyectos basándonos en los casos de uso para el modelado y la planeación de desarrollo del sistema. Por ejemplo, aquí listamos algunos de los subproyectos:
  - Control de acceso
  - Módulo de impresión de plásticos
  - Módulo de captura de huella dactilar
  - Administración de usuarios
  - Captura de datos de licencia
- Construcción. En esta fase se programaron y probaron todos los elementos del sistema. Las pruebas retroalimentaron la parte de programación para reducir al mínimo los defectos. Se desarrollaba subsistema por subsistema, cuidando la integración de todas las iteraciones con todo el proyecto. Por ejemplo, el caso de la lectura de la huella dactilar. El mismo lector se usa para iniciar sesión en el sistema, hacer la captura de huellas de los conductores y además la autorización de edición de datos de licencia por parte de los supervisores de los módulos, por lo tanto, se debe hacer una solución que funcione bien para los tres segmentos del sistema.
- Transición. Los usuarios iniciaron a usar el sistema. Se da mantenimiento al sistema reparando bugs reportados. Las funciones faltantes que sean requeridas se pueden agregar mientras la última versión está en marcha, pero el despliegue de las nuevas funciones implica que el ciclo de Unified Process inicie nuevamente.

#### 4.4 Herramientas de desarrollo

Se listan las herramientas y los principales usos que se les daban:

- Netbeans IDE versión 8.2
  - Escritura de código de lenguaje Java, Javascript y documentos JSP
  - Ejecución de aplicaciones Java EE
  - Integración con Strus3 e Hibernate a través de complementos al IDE
  - Traza de errores
  - Integración del servidor de aplicaciones Apache Tomcat
  - Ejecución paso a paso para Java y Javascript
- Atom versión 1.38
  - Visualización de archivos en formato JSON
  - Edición de archivos JavaScript

- Notepad ++ versión 7.5
  - Visualizador de texto en general
- Microsoft SQL Server Management Studio version 17.3
  - Administración de base de datos
  - Escritura de consultas a la base de datos
- LucidChart Diagram Software & Visual Solution
  - Realización de todo tipo de diagramas visuales para la aplicación.

#### 4.5 Equipo de desarrollo

Fueron cuatro personas las involucradas en el desarrollo y despliegue del sistema. Si segmentamos el proyecto con sus respectivos participantes tenemos:

- Desarrollo para plataforma web
  - Alberto Reyes → 85%
  - Persona 3 → 15%
- Desarrollo con dispositivos periféricos
  - Persona 1 → 100%
- Migración de base de datos
  - Persona 2 → 100%
- Implementación
  - Alberto reyes → 85%
  - Persona 2 → 15%

## 5. Implementación

En este capítulo se describe la implementación de algunos casos específicos en el desarrollo del sistema.

### 5.1 Comparación de huellas 1 a 1

Supongamos que tenemos dos imágenes que tienen una huella dactilar llamándolas “muestra” y “candidato” respectivamente. De cada imagen podemos extraer minucias<sup>14</sup>. La API de SourceAFIS ayudará a obtener un template<sup>15</sup> conformado de minucias. SourceAFIS también nos ayudará a realizar la comparación en base de los template obtenidos. La comparación entre la muestra y el candidato regresa una calificación, el usuario debe fijar cual es la calificación mínima para considerar que las huellas son de la misma persona o no. En código tenemos:

```
import java.util.Base64;
import java.io.ByteArrayInputStream;
import java.io.IOException;
import java.io.InputStream;

public String huellasCoinciden() {
    byte[] muestra = Base64.getDecoder().decode(huella1);
    byte[] candidato = Base64.getDecoder().decode(huella2);
    boolean isHuellaCorrecta = comparaHuellas(muestra, candidato);

    if (huellaCorrecta) {
        inputStream = new ByteArrayInputStream("1".getBytes());
        break;
    } else {
        inputStream = new ByteArrayInputStream("0".getBytes());
    }

    return "success";
}

private boolean comparaHuellas(byte[] muestra, byte[] candidato) {
    try {
        FingerprintTemplate tPrueba = new FingerprintTemplate()
            .dpi(500)
            .create(muestra);
        FingerprintTemplate tCandidato = new FingerprintTemplate()
            .dpi(500)
            .create(candidato);
        double calificacion = new FingerprintMatcher()
            .index(tPrueba)
            .match(tCandidato);
        boolean coincidencias = calificacion >= 40;
        return coincidencias;
    } catch (Exception e) {
        return false;
    }
}
```

Visto de otra forma el flujo es el siguiente:

<sup>14</sup> Las minucias son las líneas que forman las figuras que las huellas dactilares tienen, como el núcleo de la huella, un fin de camino o una bifurcación.

<sup>15</sup> Es una plantilla que se usa para dar uniformidad a los elementos que se generen.

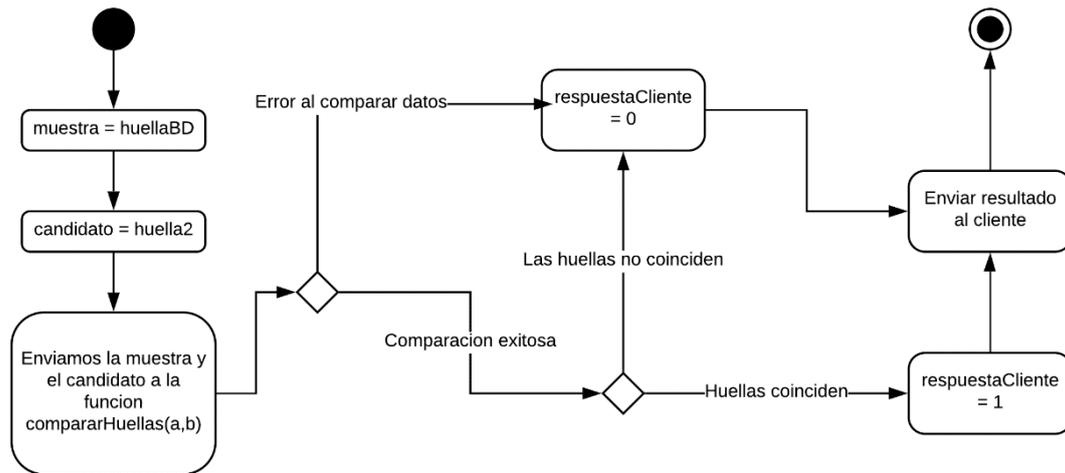


Ilustración 48 - Diagrama de flujo de comparación de huellas 1 a 1

## 5.2 Cifrado y descifrado de cadenas de texto

El cifrado es hacer la transformación de un conjunto de datos (puede ser un texto, archivo, etc.) a través de un algoritmo de cifrado. El descifrado es usar el mismo algoritmo de cifrado para obtener los datos originales. Se debe tomar en cuenta lo que se cifra es un conjunto de bytes. Dependiendo el algoritmo se necesitará una llave para poder cifrar o descifrar y tal vez más elementos.

En código tenemos los siguientes dos métodos de cifrado y descifrado:

```

public String encrypt(byte[] text){
    byte[] encryptedData;
    try{
        _cipher.init(Cipher.ENCRYPT_MODE, _password, _IVParamSpec);
        encryptedData = _cipher.doFinal(text);
    } catch (Exception e) {
        return null;
    }
    return Base64.getEncoder().encodeToString(encryptedData);
}

public byte[] decrypt(String text) {
    try {
        _cipher.init(Cipher.DECRYPT_MODE, _password, _IVParamSpec);
        byte[] decodedValue = Base64
            .getDecoder()
            .decode(text.getBytes());
        byte[] decryptedVal = _cipher.doFinal(decodedValue);
        return decryptedVal;
    } catch (Exception e) {
        return null;
    }
}
  
```

## 5.3 Despliegue de un formulario usando Struts2

A lo largo del sistema se suelen usar muchos formularios. Por lo tanto, la implementación de los formularios es parte fundamental del sistema. Es necesario dividir esto en tres partes:

1. JSP

2. Controlador (clase java)
3. struts.xml (documento donde se listan los métodos java parte del controlador o controladores con los que el cliente puede interactuar)

El catálogo de alergias tiene el formulario más pequeño, por lo tanto podemos usarlo para una representación básica de los formularios de la aplicación.

Un ejemplo del código JSP sería el siguiente (archivo alergias.jsp):

```
<%@taglib prefix="s" uri="/struts-tags" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<s:form id="form" >
  <div class="form-group col-md-12">
    <s:hidden name="alergia.alergiaId" id="alergiaId"/>
    <s:textfield type="text" name="alergia.alergia" id="alergia"
    maxLength="75" disabled="true" />
    <input name="alergia.activo" type="checkbox" id="activa"
    value="true" />Activa
  </div>
  <div class="col-md-12">
    <button id="btnGuardar" type="submit" disabled>
      Guardar
    </button>
    <button id="btnCancelar" type="button" disabled>
      Cancelar
    </button>
  </div>
</s:form>
```

El código en el archivo struts.xml para el despliegue de este formulario sería el siguiente:

```
<action name="alergias"
class="com.latinid.controladores.AlergiasController"
method="execute">
  <result name="success">jsp/catalogos/alergias.jsp</result>
</action>
```

El formulario como resultado sería:

The screenshot shows a web form titled "Descripción de la alergia". It contains a single-line text input field. Below the input field is a checkbox labeled "Activa". At the bottom of the form, there are two buttons: "Cancelar" (with a red circle and slash icon) and "Guardar" (with a green floppy disk icon).

**Ilustración 49 - Formulario de alergias.**

El código en la clase java controlador sería el siguiente (archivo AlergiasController.java):

```

@Override
public String execute() throws Exception {
    if (sesion.isEmpty()) {
        return LOGIN;
    }
    HCatAlergias hCatAlergias = new HCatAlergias();
    alergias = hCatAlergias.obtenerAlergias();
    usuarioSesion = (CatUsuarios) sesion.get("session");
    menu = Menu.getMenu(usuarioSesion.getCatPerfiles());

    return SUCCESS;
}

```

Para representar como es el funcionamiento de esto se presenta el siguiente diagrama de flujo:

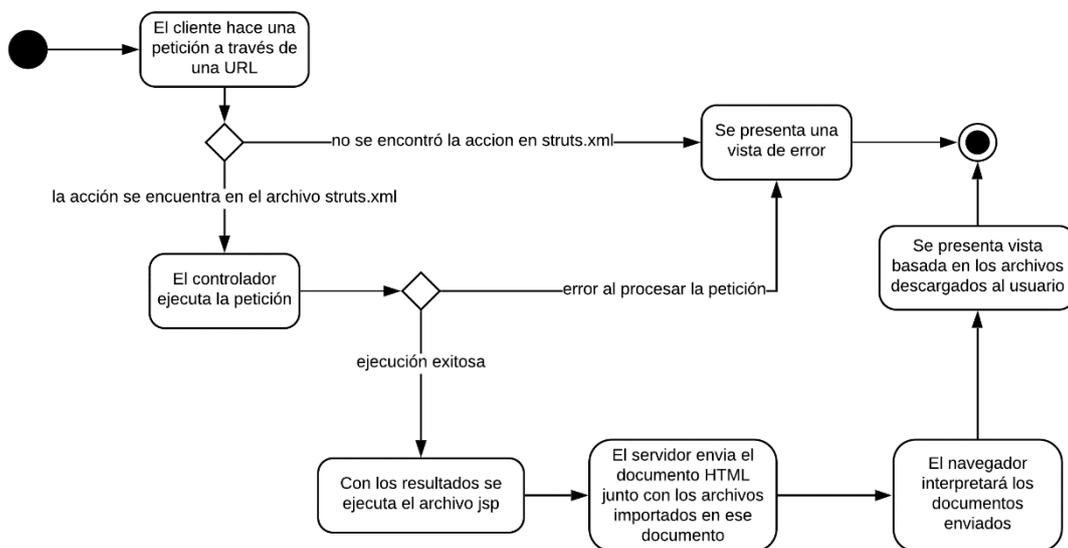


Ilustración 50 - Diagrama de flujo de despliegue de formulario usando Struts2

#### 5.4 Creación y eliminación de sesión usando Struts2

Struts2 cuenta con un administrador de sesiones llamado `SessionAware`. Este administrador permite almacenar (en un `Map`<sup>16</sup>) ciertos datos de forma temporal según el desarrollo lo requieran. En este caso son datos del usuario y el módulo los que se almacenan. Se dice que se guardan los datos de forma temporal dado que el sistema tiene un parámetro que indica cual es el límite de tiempo de inactividad para que la sesión sea eliminada. Por ejemplo, si los usuarios no tienen interacción con el servidor durante 30 minutos (asumiendo que se configuró a 30 minutos) la aplicación terminará tal sesión. En este sistema la única forma de iniciar una sesión es terminando la autenticación de usuario que se describió en el control de acceso.

El código para añadir una nueva sesión y para terminarla se presenta a continuación:

<sup>16</sup> Map es una interfaz que provee el lenguaje Java para la representación de la estructura de datos tipo lista clave-valor.

```

public class LoginController extends ActionSupport implements SessionAware {

    private Map sesion;

    @Override
    public void setSession(Map<String, Object> map) {
        this.sesion = map;
    }

    public String inicioSesion() {
        /**
         * código de verificación de huella de usuario
         */
        if (huellaCorrecta) {
            // añadimos al usuario a la sesión
            sesion.put("sesion", usuarioMuestra);
            inputStream = new ByteArrayInputStream("1".getBytes());
        } else {
            // respondemos al cliente que no se inició la sesión
            inputStream = new ByteArrayInputStream("0".getBytes());
        }
        return SUCCESS;
    }

    public String logout() {
        if (!sesion.isEmpty()) {
            sesion.clear();
        }
        return SUCCESS;
    }
}

```

### 5.5 Validaciones y llamadas AJAX en el procesamiento de formularios

Una validación de formularios es una acción que simplemente revisará los campos en el formulario, si los campos cumplen con las reglas especificadas en el diseño de la aplicación, el formulario se podrá enviar, de forma contraria se debe indicar el error en el campo donde este suceda para que el usuario pueda rectificar los datos.

AJAX es una tecnología que permite que un cliente web haga peticiones a un servidor y que la respuesta no cause la carga de la página completa nuevamente. Es ejecutada con JavaScript y en esta aplicación se utiliza jQuery que provee una forma muy sencilla de generar dichas peticiones.

Lo que se necesita contemplar para esta implementación:

- Un documento HTML con jQuery y jQuery Validation
- Un elemento en el documento HTML que dispare la llamada AJAX (suelen ser botones)
- La instrucción AJAX

Para ejemplificar esta implementación se usa el formulario de búsqueda de Nueva licencia.

En el JSP tenemos:

```

<s:form id="formBuscaConductor">
  <div class="col-lg-7">
    <h5>
      <s:checkbox name="busquedaDatos" id="busquedaDatos"/>
      Por Datos del Conductor:
    </h5>
    <div class="form-group">
      <label>Fecha de Nacimiento *</label>
      <s:textfield name="conductor.fechaNacimiento" maxLength="10"
type="text" id="fechaNacimiento" disabled="true"/>
    </div>
    <div class="form-group">
      <label>Nombre *</label>
      <s:textfield name="conductor.nombre" id="nombre" type="text"
disabled="true"/>
    </div>
    <div class="form-group">
      <label>Apellido Paterno *</label>
      <s:textfield name="conductor.paterno" id="paterno" type="text"
disabled="true"/>
    </div>
    <div class="form-group">
      <label>Apellido Materno *</label>
      <s:textfield name="conductor.materno" id="materno" type="text"
disabled="true"/>
    </div>
    <div class="form-group">
      <label>Sexo *</label>
      <s:select name="conductor.catSexo.sexoId" id="sexo" list="sexos"
listKey="sexoId" listValue="sexo" headerValue="Selecciona una opción"
disabled="true"/>
    </div>
  </div>
  <div class="col-lg-5">
    <h5>
      <s:checkbox name="busquedaCurp" id="busquedaCurp"/>
      Por CURP:
    </h5>
    <div class="form-group">
      <label>CURP *</label>
      <s:textfield name="conductor.curp" id="curp" type="text"
maxlength="18" disabled="true" />
    </div>
    <div style="padding-top: 70%">
      <button type="submit" id="btnBuscar"> Buscar</button>
      <button type="button" id="btnLimpiar"> Limpiar</button>
    </div>
  </div>
</s:form>

```

En el archivo JavaScript tenemos la validación y el procesamiento del formulario a través de AJAX. Se puede ver que en la parte de "rules" se especifica el nombre del input y las restricciones que tendrá. Después está la parte de "messages" donde se especifica el mensaje que se muestra al usuario cuando tiene un error en algún campo. Al final está la parte "submitHandler" (manejador de envío). Cuando todos los campos cumplen con las reglas se ejecuta el código dentro del submitHandler, en el que precisamente tenemos la implementación de AJAX.

```

$formBuscaConductor.validate({
  rules: {
    "conductor.curp":
      {
        minlength: 18,
        required: true,
        maxlength: 18
      },
    "conductor.catSexo.sexoId": {required: true},
    "conductor.paterno": {required: true},
    "conductor.nombre": {required: true},
    "conductor.fechaNacimiento": {required: true}
  },
  messages: {
    "conductor.curp": "CURP es requerido",
    "conductor.catSexo.sexoId": "Elige un sexo",
    "conductor.paterno": "Escribe el apellido paterno",
    "conductor.nombre": "Escribe el nombre",
    "conductor.fechaNacimiento": "Escribe la fecha de nacimiento"
  },
  submitHandler: function (form) {
    // enviamos el formulario
    $.ajax({
      type: 'POST',
      url: "busquedaConductorNL.action",
      scriptCharset: 'utf-8',
      data: $formBuscaConductor.serialize()
    }).done(function (msg) {
      /**
       * código para presentar los resultados de la búsqueda
       */
    }).fail(ajaxFail);
  }
});

```

El formulario y las validaciones que se hacen, el usuario las puede ver de esta manera:

The screenshot shows a search form with two columns of input fields. On the left, there are five fields: 'Fecha de Nacimiento \*', 'Nombre \*', 'Apellido Paterno \*', 'Apellido Materno \*', and 'Sexo \*'. Each field has a green validation message below it: 'Escribe la fecha de nacimiento', 'Escribe el nombre', 'Escribe el apellido paterno', and 'Elige un sexo'. On the right, there are two radio buttons: 'Por Datos del Conductor:' (checked) and 'Por CURP:'. Below them is a 'CURP \*' field. At the bottom right, there are two buttons: a blue 'Buscar' button and an orange 'Limpiar' button. A checkbox 'El solicitante cuenta con licencia de otro estado' is also present.

Ilustración 51 - Formulario de búsqueda. Los mensajes en color verde en la parte de abajo de los campos del formulario indican que no se hace la búsqueda hasta que los datos cumplan con los criterios.

## 6. Fase de pruebas

El desarrollo de sistemas lleva un diseño previo a la escritura de código fuente. Al momento de diseñar esta aplicación se solía hacer una división en distintos segmentos para obtener las funciones que iba cumplir el sistema.

Para la mayoría de los diseños se usa un contexto (que son las consideraciones previas antes de hacer operación en el sistema), una entrada y una salida. Para comprobar que cada segmento del sistema hace lo esperado se deben hacer una serie de pruebas a dichos segmentos de forma aislada, así como en conjunto, formando las posibles interacciones entre cada uno de los segmentos que pueda haber.

Los métodos de pruebas de software son de forma manual o de forma automatizada. La forma manual es cuando una persona revisa el software por su cuenta, teniendo la perspectiva de un usuario final. La forma automatizada es cuando se usan herramientas hechas por software, siendo su función principal encontrar defectos. Las pruebas documentadas aquí pertenecen al conjunto de pruebas manuales.

Dentro de la fase de pruebas existen las denominadas pruebas de caja negra. Estas pruebas se centran en las entradas que reciben el segmento del sistema y las salidas resultantes de la ejecución de tal segmento del sistema. Se dice que son de caja negra porque no se toma en cuenta el funcionamiento a nivel de código, es decir, lo primero que se busca con estas pruebas es ver el que se hace más no el cómo se hace. Estas pruebas se pueden realizar sabiendo los pasos de ejecución sobre la interfaz que se despliega al usuario.

Las pruebas descritas aquí son del tipo caja negra, dado que son de este tipo, no necesitan el análisis del código, sin embargo, dan una idea muy clara de las funciones del sistema.

### 6.1 Inicio de sesión

Diseñado por:	Alberto Reyes
Probado por:	Alberto Reyes
Tipo de prueba:	Caja negra
Precondiciones:	La PC cliente debe contar un lector de huella dactilar, la extensión web del sistema y el asistente de captura de llueva instalado.
Descripción de la prueba:	Para que el usuario pueda iniciar sesión deberá tener en cuenta todas las validaciones. La primera que solo podrá iniciar sesión en el horario de servicio asignado a su módulo. Deberá tener un usuario y una contraseña. El usuario deberá tener asociado dos huellas dactilares para la comparación con la captura que se realiza en la toma del inicio de sesión.
Elementos por probar	
1	Horario de servicio del módulo
2	Comprobación de contraseña del usuario
3	Comprobación de huella dactilar ingresada por el usuario
Configuración de la prueba (hardware, software)	
Hardware	<ul style="list-style-type: none"> <li>• PC cliente</li> <li>• Lector de huella dactilar</li> </ul>
Software	<ul style="list-style-type: none"> <li>• Navegador web</li> </ul>

<ul style="list-style-type: none"> <li>• Extensión web</li> <li>• Asistente de captura de huella dactilar (huella.exe)</li> </ul>
---

Especificaciones		
Entrada	Resultado esperado	Resultado obtenido
<ul style="list-style-type: none"> <li>• El usuario debe seleccionar un módulo</li> <li>• El usuario debe ingresar su usuario y contraseña</li> <li>• El usuario debe hacer la captura de su huella dactilar</li> </ul>	<ul style="list-style-type: none"> <li>• El sistema debe permitir o no permitir al usuario la generación de sesiones de acuerdo con los datos que ingrese para la comprobación de su identidad</li> </ul>	<ul style="list-style-type: none"> <li>• El sistema cumplió con los requerimientos esperados</li> </ul>

## 6.2 Registro de datos del trámite Nueva Licencia

Diseñado por:	Alberto Reyes
Probado por:	Alberto Reyes
Tipo de prueba:	Caja negra
Precondiciones:	El perfil del usuario debe ser capturista.
Descripción de la prueba:	El usuario hace una búsqueda en el sistema para obtener información del conductor o para saber si puede obtener una nueva licencia. Si el conductor es apto para una nueva licencia se procede al registro de datos. Cuando se registra una nueva licencia se guardan los datos de la licencia, los datos del conductor, los datos de la dirección del conductor y los datos de media filiación del conductor.
Elementos por probar	
1	Búsqueda del conductor
2	Procesamiento del formulario
3	Almacenamiento de datos
Configuración de la prueba (hardware, software)	
Hardware	<ul style="list-style-type: none"> <li>• PC cliente</li> </ul>
Software	<ul style="list-style-type: none"> <li>• Navegador web</li> </ul>

Especificaciones		
Entrada	Resultado esperado	Resultado obtenido

<ul style="list-style-type: none"> <li>El solicitante debe proveer sus nombres, fecha de nacimiento o su CURP</li> </ul>	<ul style="list-style-type: none"> <li>La base de datos debe tener los registros del nuevo trámite</li> </ul>	<ul style="list-style-type: none"> <li>La base de datos tiene los registros del nuevo trámite</li> </ul>
--	---	--

Pasos que seguir		Probado
1	Abrir vista de Búsqueda de nueva licencia	Ok
2	Seleccionar "Búsqueda por CURP" o "Búsqueda por datos de licencia"	Ok
3	Capturar los datos según el tipo de búsqueda	Ok
4	Si el sistema arroja resultados para continuar con la búsqueda, seleccionar el tipo de licencia	Ok
5	Seleccionar la vigencia de la licencia	Ok
6	Hacer clic en botón "Iniciar"	Ok
7	Completar los formularios con la información del conductor, dirección y descripción media filiación	Ok
8	Hacer clic en el botón "Guardar información de licencia"	Ok

### 6.3 Captura de información biométrica

La captura de biometría se hace cuando se genera una renovación o una nueva licencia.

Diseñado por:	Alberto Reyes
Probado por:	Alberto Reyes
Tipo de prueba:	Caja negra
Precondiciones:	La máquina cliente debe contar con la extensión instalada, así como los controladores de los dispositivos periféricos usados en la prueba.
Descripción de la prueba:	El proceso de captura de biometría inicia cuando se guarda la información del conductor. El primer paso es hacer una captura facial, el segundo paso es hacer una captura de la firma y como tercer paso es capturar dos huellas. Todas las tomas son al solicitante.
Elementos por probar	
1	Toma facial
2	Toma de firma
3	Toma de huellas dactilares
4	Almacenamiento de las tomas en sistema de archivos y base de datos
Configuración de la prueba (hardware, software)	
Hardware	<ul style="list-style-type: none"> <li>Webcam</li> <li>Tablero de firmas</li> <li>Lector de huella dactilar</li> </ul>

Software	<p>La PC del cliente debe tener los tres subsistemas:</p> <ul style="list-style-type: none"> <li>• camara.exe</li> <li>• firma.exe</li> <li>• huella.exe</li> </ul>
----------	---

Especificaciones		
Entrada	Resultado esperado	Resultado obtenido
<ul style="list-style-type: none"> <li>• El solicitante debe estar presente para ingresar su información</li> </ul>	<ul style="list-style-type: none"> <li>• En base de datos deben estar registrados los metadatos de las imágenes</li> <li>• Las imágenes capturadas deben estar en el sistema de archivos con la estructura indicada</li> </ul>	<ul style="list-style-type: none"> <li>• Los metadatos de las capturas llegaron a la base de datos correctamente</li> <li>• Las capturas llegaron al sistema de archivos de forma correcta</li> </ul>

Pasos que seguir		Probado
1	Una vez terminada la captura de datos, hacer clic en el botón "Captura facial"	Ok
2	Se abrirá el asistente de fotografía. Cuando el solicitante esté en posición hacer clic en el botón "Capturar"	Ok
3	Hacer clic en el botón "Capturar firma"	Ok
4	Se abrirá el asistente de captura de firmas. Se debe realizar el trazo sobre el tablero de firmas. Posteriormente clic en el botón "Capturar"	Ok
5	Hacer clic en el botón "Capturar huella 1"	Ok
6	Se abrirá el asistente de captura de huellas. Se deberá apoyar un dedo sobre el lector de huellas. Cuando el asistente muestre la captura hacer clic en el botón "Aceptar"	Ok
7	Hacer clic en el botón "Capturar huella 2"	Ok
8	Cuando se haya tomado toda la información se hará clic en el botón "Guardar". Se mostrará un mensaje con el estatus de la acción realizada.	Ok

#### 6.4 Impresión de licencias

Diseñado por:	Alberto Reyes
Probado por:	Alberto Reyes
Tipo de prueba:	Caja negra
Precondiciones:	El capturista debió realizar la toma de los datos personales y biométricos de un conductor para nueva/duplicado/renovación de licencia.

Descripción de la prueba:	La impresión de la licencia se conforma de una búsqueda y dar la instrucción a la impresora de plasmar las imágenes sobre la tarjeta de PVC.
Elementos por probar	
1	Búsqueda de conductor/licencia
2	Proceso de impresión sobre el plástico
3	Datos registrados en base de datos posterior a la impresión de la licencia
Configuración de la prueba (hardware, software)	
Hardware	Impresora de tarjetas de PVC
Software	Impresion.exe

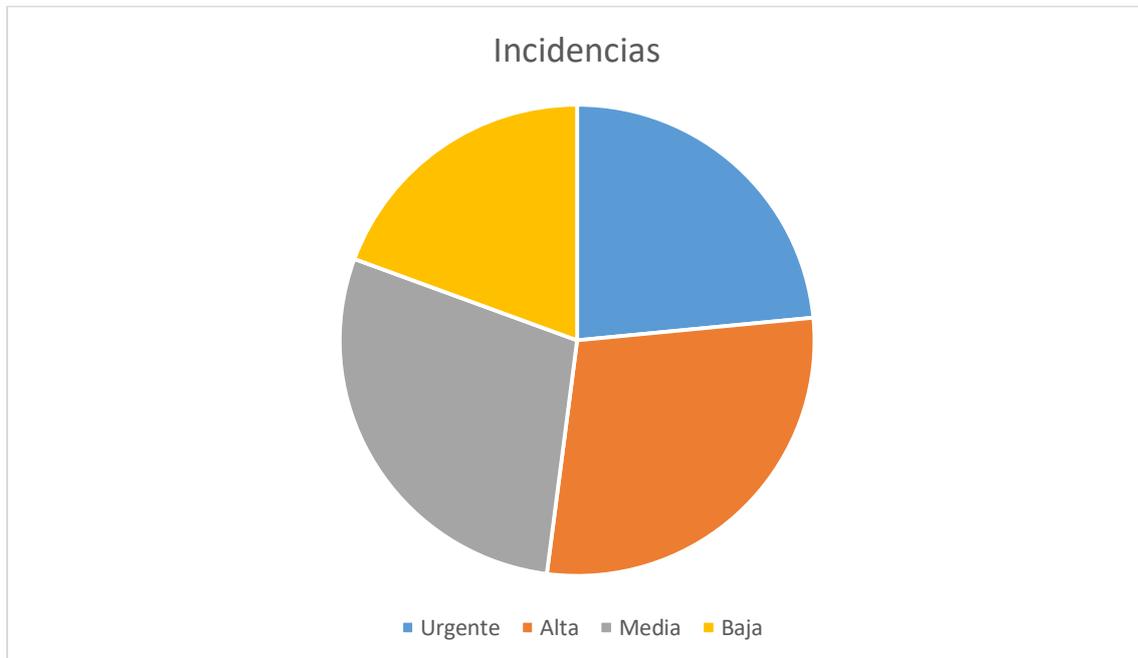
Especificaciones		
Entrada	Resultado esperado	Resultado obtenido
<ul style="list-style-type: none"> <li>CURP del solicitante</li> <li>Tarjeta de PVC</li> </ul>	<ul style="list-style-type: none"> <li>Registro en la base de datos referente a la impresión</li> <li>Plástico impreso con la información de la licencia</li> </ul>	<ul style="list-style-type: none"> <li>Registro en la base de datos referente a la impresión</li> <li>Plástico impreso con la información de la licencia</li> </ul>

Pasos que seguir		Probado
1	Iniciar sesión.	Ok
2	Ir al submenú "Impresión de licencias" y realizar la búsqueda del conductor.	Ok
3	Seleccionar la licencia que se imprimirá (el sistema enviará solo las licencias pendientes por imprimir).	Ok
4	Hacer clic en el botón iniciar impresión. Se iniciará el asistente de impresión (impresion.exe) mostrando una vista previa de la impresión.	Ok
5	Se debe revisar ambos lados de la impresión, para posterior a eso hacer clic en imprimir.	Ok
6	La impresora tomará una tarjeta del depósito y la imprimirá.	Ok

La fase de pruebas fue terminada para el primer ciclo del desarrollo del sistema. Se le daba una prioridad a los errores o incidencias y esa se clasificaba por la rapidez o urgencia con la que era necesaria la resolución del problema.

La tabla de resumen al final de la fase de pruebas es la siguiente:

Tabla de resumen			
Prioridad	Abiertos	Resueltos	% Resueltos
Urgente	0	23	100%
Alta	0	28	100%
Media	0	28	100%
Baja	0	19	100%



Si bien es complicado anticiparse a todas las posibilidades en el uso de un sistema web, se logró prevenir posibles fallas. De igual forma, un sistema puede mejorarse y afinarse muchas veces, pero por cuestiones de tiempo de entrega, se debe cuidar no perder los objetivos del desarrollo y tener en mente que si algo funciona bien, no es necesario invertir más tiempo del necesario. Una vez concluida esta fase podemos continuar con el despliegue del sistema.

## 7. Despliegue del sistema

### 7.1 Sistema central

El despliegue del sistema es la implementación completa del sistema. Equivale a la distribución del hardware que se usa, así como las distintas piezas de software que conforma al sistema. Lo siguiente es usar un diagrama de despliegue para la representación de todas las partes del sistema en el entorno real. Nos apoyamos en este para especificar los requerimientos de hardware su función dentro del ecosistema del sistema.

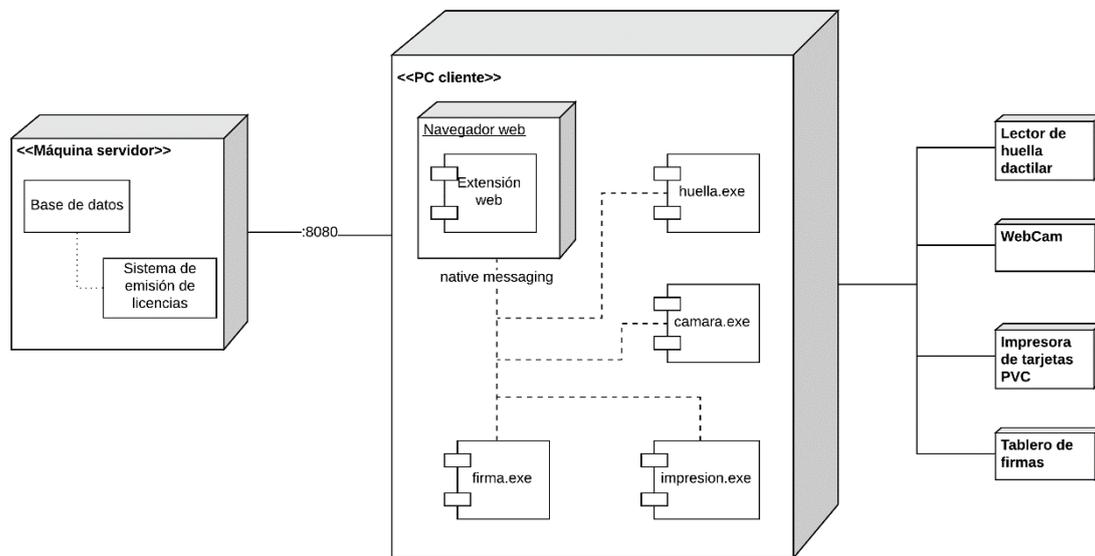


Ilustración 52 - Diagrama de despliegue.

La máquina servidor funcionara como repositorio de datos, así como unidad de procesamiento de los servicios de base de datos y el sistema emisor de licencias. El sistema emisor de licencias depende de la base de datos completamente para que pueda funcionar de forma correcta. Las especificaciones mínimas de la máquina servidor son:

- Sistema operativo Windows
- 32 GB de memoria principal
- Arquitectura de procesador y sistema operativo de 64 bits
- Procesador Intel Xeon(r) a 3.50 GHz.

La comunicación entre la máquina servidor y las máquinas cliente se hace a través del puerto 8080 usando el protocolo http (protocolo de transferencia de hipertexto).

La comunicación entre el navegador web y las aplicaciones de escritorio se hace a través de Native Messaging.

El PC del cliente es donde el navegador interpretará la aplicación principal. En la PC del cliente podemos decir que la aplicación principal será la parte interpretada por el navegador, así como los cuatro subsistemas que no forman parte del navegador de internet. Los cuatro subsistemas (huella.exe, camara.exe, firma.exe e impresiones.exe) tienen comunicación con dispositivos externos a la PC. Las especificaciones mínimas de la PC<sup>17</sup> cliente:

<sup>17</sup> Personal Computer (computadora personal)

- Sistema operativo Windows 10
- Navegador web Google Chrome
- 8 GB de memoria principal
- Procesador Intel Core i5
- Controlador de la impresora de tarjetas
- Controlador de lector de huella dactilar
- Controlador de tablero de firmas

El lector de huellas se usa para la captura de huellas de los usuarios y de los solicitantes de licencias. Las especificaciones técnicas deben ser las siguientes:

- Debe capturar una sola huella a la vez
- Captura de huella en 8 bits de escala de grises
- Compatible con USB 2.0
- Resolución de 512 ppp<sup>18</sup>

La webcam se usa específicamente para realizar las capturas faciales de los solicitantes de las licencias. Debe cumplir con estas especificaciones:

- Resolución 720p/30 fps<sup>19</sup>
- Enfoque automático
- Campo visual 69°

La impresora de tarjetas es la encargada de generar el producto final del sistema. Esta imprime sobre las tarjeas de PVC que serán el principal y único comprobante del solicitante de que cuenta con los derechos que sean especificados sobre el plástico.

El tablero de firmas se usa para que los solicitantes hagan su firma autógrafa sobre una almohadilla digital. Así se obtiene una imagen en el que se almacena la firma de la persona. El tablero debe cumplir con las siguientes especificaciones:

- Compatible con USB 2.0
- Pantalla LCD
- Pantalla sensible al tacto de una pluma hecha para la pantalla
- Tamaño de la pantalla de 112mm x 33mm

---

<sup>18</sup> Puntos por pulgada. Es una medida para resoluciones de impresión.

<sup>19</sup> Fotogramas por segundo. Es una medida de velocidad. Un fps es igual a las tomas que hace y presenta la cámara en un segundo.

## 7.2 Unidades móviles

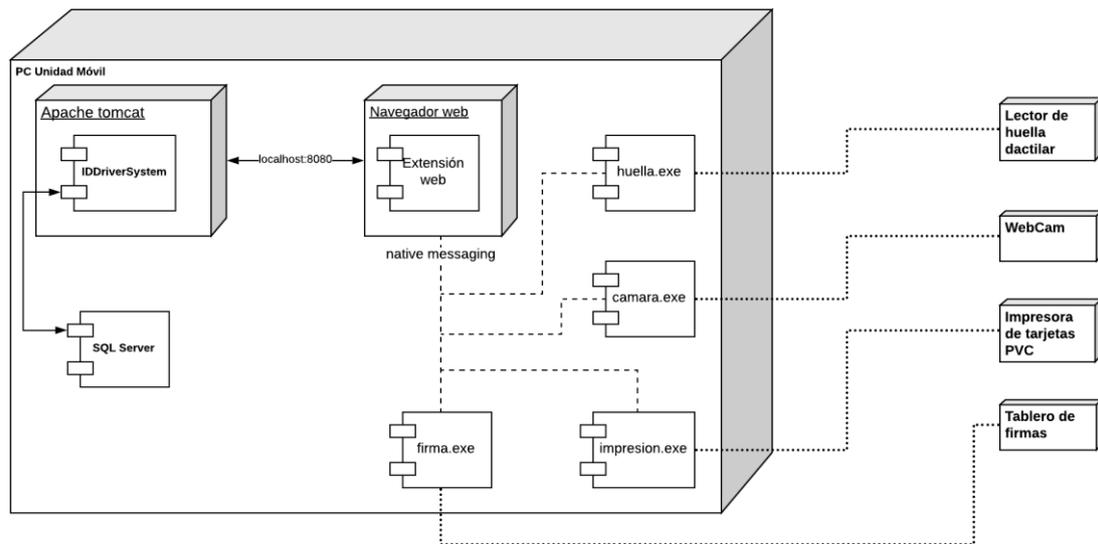


Ilustración 53 - Diagrama de despliegue de Unidad móvil

Una unidad móvil dará servicio dentro de un automóvil. Este está equipado con los elementos de emisión que se tiene en un módulo emisor. En el caso de las unidades móviles, el despliegue es similar pero no del todo igual. Están conformadas por una sola máquina, lo cual significa que esa PC será servidor y cliente. Hay cambios a nivel sistema y en la distribución de los elementos que lo conforman.

## Conclusiones

El desarrollo de este y cualquier otro sistema es algo laborioso, más allá de escribir código o diseñar interfaces, se requiere de mucho tiempo de análisis, lo cual es un proceso largo y que de no hacerse con una planeación correcta o sin tener los alcances fijados de una forma muy específica, el tiempo de trabajo se puede alargar sin control y generar trabajo doble. De ahí la importancia del uso de metodología, seguir una o varias guías que mantengan a los involucrados en el desarrollo centrados en los objetivos. Se debe de hacer el reconocimiento de los actores y funciones que se involucrarán con la aplicación.

Escoger las tecnologías que se adecuan al sistema para simplificar el desarrollo ayudará bastante, ahorrara tiempo y hará más fácil la implementación de las mismas.

En el desarrollo de software surgen errores que pueden ser desde una falta de ortografía en un título, hasta problemas de concurrencia en el almacenamiento de datos. Es por eso por lo que la fase de pruebas y retroalimentación de errores es de las partes más importantes en desarrollo. Por eso lo mejor es llevar un control de todos los problemas que un sistema tenga, especificando un contexto previo y los pasos para replicar cualquier incidente.

Este tipo de sistemas suelen tener cambios de un día para otro, por ejemplo, un nuevo director de transporte o una nueva vigencia en las licencias. Por lo tanto, se debe tener una buena organización del código, y tener en cuenta todas las implementaciones de tecnologías que tiene el sistema. Tener una base de datos bien diseñada y que siga reglas de normalización ayuda a agilizar mucho estos cambios.

No perder de vista al usuario mediante todo el desarrollo es primordial. La interfaz del software debe ser muy clara y no permitir hacer acciones más allá de las permitidas. Como desarrollador se debe manejar en el código e interfaz un buen esquema de restricciones para no tener problemas más adelante.

La puesta en marcha de un sistema puede llegar a ser sencilla si todos los requerimientos del cliente fueron cumplidos o al menos los comprometidos a la fecha de salida del sistema, los inconvenientes pueden ser las variables que no están siempre controladas, en este caso era la gran variedad de computadoras de distintos modelos que se usaron en el ambiente de producción. Como contemplación extra se debe hacer pruebas en el mayor número de modelos de máquinas cliente posible para poder asistir los problemas de compatibilidad.

En desarrollo de sistemas posteriores plantearía el uso de alguna metodología ágil para agregar más flexibilidad en el transcurso de vida del proyecto, además incluiría al cliente a lo largo del desarrollo de forma más constante. Otra cosa que plantearía para futuros desarrollos sería buscar la forma de dividir este tipo de aplicaciones en dos. Una aplicación que sea completamente dedicada al front-end usando una tecnología ligera como React.js, y el back-end optaría por generar una arquitectura enfocada a microservicios, y tal vez segmentar el back-end a su vez en una o varias aplicaciones independientes.

En ingeniería en computación de la UNAM - FES Aragón y me parece que en el caso general de la profesión, el desarrollo va apegado a la capacidad y práctica de autodidactismo, lo cual, es una de las principales cosas a las que te alienta esta carrera. La influencia de ICO se ve en la capacidad de abstraer problemas en distintas formas, ya sea en flujos, ilustraciones, en modelos, código o en otro tipo de representaciones; el adaptarte a usar diversas tecnologías; usar y crear dispositivos periféricos e integrarlos con software; tener un panorama solido de sistemas

operativos y seguridad en computación. En general esos fundamentos me ayudaron a terminar este proyecto, retroalimentarme y continuar mi desarrollo, siempre buscando mejorar las cosas que hago.

## Bibliografía

- Bloch, J. (2018). *Effective Java*. Addison Wesley.
- Castro, C. (2016). *Arquitectura de software*. Cengage Learning.
- Cecil, R. (202). *UML for Java Programmers*. Prentice Hall.
- Flower, M. (1999). *UML gota a gota*. Addison Wesley Logman de México, S.A. de C.V.
- ISO. (2011). *International Organization for Standarization*. Récupéré sur ISO/IEC 19794-5:2011: <https://www.iso.org/standard/50867.html>
- ISO/IEC. (2011). *ISO/IEC 19794-4 Information technology - Biometric data interchange formats*. ISO/IEC.
- James Rumbaugh, I. J. (1999). *The Unified Modeling Language Reference Manual*. Reading Massachusetts: Addison Wesley Longman, Inc.
- Pfleeger, C. P. (2015). *Security Computing*. Prentice Hall.
- Toledo, F. (2014). *Introduccion a las pruebas de sistemas de información*. Montevideo, Uruguay: Abstracta.
- Torossi, G. (s.d.). *El Proceso Unificado de Desarrollo de Software*. Addison Wesley Longman, Inc.

## Referencias

### AdminLTE

- ColorlibHQ. (2019, August 26). ColorlibHQ/AdminLTE. Retrieved from <https://github.com/ColorlibHQ/AdminLTE>

### Apache Tomcat

- Project, A. T. Apache Tomcat®. Retrieved from <http://tomcat.apache.org/>

### DataTables

- DataTablesManual. Retrieved from <https://datatables.net/manual/index>

### DatePicker

- Datepicker. Retrieved from <https://bootstrap-datepicker.readthedocs.io/en/latest/>

### Gson

- Google. (2019, June 4). google/gson. Retrieved from <https://github.com/google/gson>

### Hibernate

- Hibernate. (2019, July 31). Retrieved from <https://es.wikipedia.org/wiki/Hibernate>

## **JAI**

- Java Advanced Imaging API. Retrieved from <https://www.oracle.com/technetwork/java/iio-141084.html>

## **JDBC**

- (2019, July 29). Java Database Connectivity. Retrieved from [https://es.wikipedia.org/wiki/Java\\_Database\\_Connectivity#targetText=Java%20Database%20Connectivity%20\(en%20espa%C3%B1ol,de%20datos%20que%20se%20utilice](https://es.wikipedia.org/wiki/Java_Database_Connectivity#targetText=Java%20Database%20Connectivity%20(en%20espa%C3%B1ol,de%20datos%20que%20se%20utilice).

## **jQuery Validation**

- jQuery Validation Plugin. Retrieved from <https://jqueryvalidation.org/>

## **JNBIS**

- Mhshams. (2018, October 29). mhshams/jnbis. Retrieved from <https://github.com/mhshams/jnbis>

## **PDFmake**

- PDFMAKE. Retrieved from <https://pdfmake.github.io/docs/>

## **SourceAFIS**

- Va, R. SourceAFIS. Retrieved from <https://sourceafis.machinezoo.com/>

## **Struts**

- Getting Started. Retrieved from <https://struts.apache.org/getting-started/>

## **SweetAlert2**

- SweetAlert2. Retrieved from <https://sweetalert2.github.io/>