



# UNIVERSIDAD NACIONAL DE TUCUMÁN

FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍA

DEPARTAMENTO DE ELECTRICIDAD, ELECTRÓNICA Y COMPUTACIÓN

## TRABAJO DE GRADUACIÓN

INFORME FINAL

---

### Web App embebida en dispositivos móviles para la gestión de registros sobre la contaminación de afluentes y ríos.

---

#### Autores

BRUNO, Ricardo Hugo (CX 1409686) - Ing. en Computación

GOMEZ VELIZ, Kevin Shionen (CX 1411828) - Ing. en Computación

#### Tutores

TUTOR - Ing. COHEN Daniel Eduardo

COTUTOR - Ing. NIETO Luis Eduardo

1 de Marzo de 2018

# Agradecimientos

Agradecemos a nuestras familias que nos apoyaron y ayudaron durante el transcurso de la carrera.

Agradecemos a la Universidad en su conjunto, pública y gratuita, que nos formó académicamente.

Gracias a nuestro tutor Cohen, Daniel Eduardo por brindarnos la posibilidad de desarrollar este proyecto. Además agradecemos a nuestro cotutor Nieto, Luis Eduardo por brindarnos ayuda y acompañamiento de igual manera que nuestro tutor.

Por último, queremos agradecer a nuestros compañeros y amigos por los momentos de estudio, logros y festejos compartidos durante la carrera, que nos motivaron para seguir adelante.

# Índice general

<b>Introducción</b>	<b>7</b>
Objetivos del sistema . . . . .	7
Selección del modelo de ciclo de vida y de la metodología de desarrollo . . . . .	8
<b>Disciplina de Requisitos</b>	<b>9</b>
Introducción . . . . .	9
Identificación de usuarios participantes . . . . .	9
Educación de requisitos . . . . .	10
Estudio de Documentación. Planificación y Realización de Entrevistas . . . . .	10
Especificación de Requisitos de Software . . . . .	10
Introducción . . . . .	10
Objetivos y alcance del sistema . . . . .	11
Definiciones, acrónimos y abreviaturas . . . . .	11
Definiciones: . . . . .	11
Descripción general . . . . .	13
Proceso actual . . . . .	13
Reingeniería de proceso . . . . .	13
Requisitos específicos . . . . .	15
Suposiciones y Dependencias: . . . . .	20
Requisitos de Usuario y Tecnológicos: . . . . .	21
Requisitos de Interfaces Externas . . . . .	22
Requisitos de Rendimiento: . . . . .	23
Requisitos de Desarrollo y Restricciones de Diseño: . . . . .	23

Ajuste a estándares: . . . . .	23
Seguridad: . . . . .	23
Estimación del proyecto . . . . .	25
Planificación de etapas . . . . .	25
Duración estimada de tareas . . . . .	26
Diagrama de Gantt . . . . .	26
<b>Disciplina de Análisis</b>	<b>27</b>
Vista de Casos de Uso . . . . .	27
Diagramas de Casos de Uso . . . . .	27
Diagrama General de Casos de Uso . . . . .	28
Gestión de Usuarios . . . . .	29
Gestión de Registros . . . . .	29
Casos de Uso Principales . . . . .	29
<b>Disciplina de Diseño</b>	<b>31</b>
Descripción Textual de Casos de Uso . . . . .	31
Caso de Uso 01: Un usuario no registrado desea registrarse en el sistema. . . . .	31
Caso de Uso 02: Un usuario registrado desea iniciar sesión en la aplicación movil. . . . .	34
Caso de Uso 03: Un usuario registrado desea crear un registro . . . . .	35
Caso de Uso 04: Un usuario registrado desea listar sus registros. . . . .	37
Caso de Uso 05: Un usuario registrado desea ver un registro en particular . . . . .	39
Caso de Uso 06: Un usuario registrado desea ver el mapa general. . . . .	40
Caso de Uso 07: Un usuario registrado desea ver su perfil. . . . .	41
Caso de Uso 08: Un usuario registrado desea modificar su perfil. . . . .	43
Caso de Uso 10: Un usuario registrado desea eliminar su cuenta. . . . .	44
Caso de Uso 11: Un administrador desea iniciar sesión. . . . .	46
Caso de Uso 12: Un administrador desea borrar o dar de baja un alumno. . . . .	48
Caso de Uso 13: Un administrador desea cambiar el rol a un usuario. . . . .	48
Caso de Uso 14: Un administrador desea listar los alumnos. . . . .	49

[illegible]

Iniciar Sesión . . . . .	70
<b>Disciplina de Implementación</b>	<b>81</b>
Arquitectura de la aplicación . . . . .	81
Diagrama de Despliegue . . . . .	82
Elección del Lenguaje . . . . .	82
Arquitectura de Android . . . . .	83
PhoneGap . . . . .	84
Ventajas de PhoneGap . . . . .	85
Desventaja de phoneGap . . . . .	86
Web services . . . . .	86
Razones para crear servicios Web . . . . .	87
REST . . . . .	87
Características de seguridad . . . . .	89
Canal de comunicación cifrado . . . . .	89
ACL para cada objeto . . . . .	90
Backbone.js . . . . .	91
Modelo Vista Presentador (MVP) . . . . .	93
jQuery . . . . .	94
jQuery Mobile . . . . .	94
Herramientas de desarrollo . . . . .	95
<b>Disciplina de Pruebas</b>	<b>97</b>
Test de Unidades . . . . .	97
Introducción . . . . .	97
Pruebas de Caja Blanca . . . . .	97
Test de Módulos . . . . .	97
Introducción . . . . .	97
Pruebas de Caja Negra . . . . .	98
Pruebas de Estrés . . . . .	98
Test de Integración . . . . .	98

<i>ÍNDICE GENERAL</i>	6
Introducción . . . . .	98
Pruebas de Integración . . . . .	98
Test de Aceptación . . . . .	99
Introducción . . . . .	99
Prueba Alfa . . . . .	99
Prueba Beta . . . . .	99
<b>Conclusiones</b>	<b>100</b>

# Introducción

La tecnología de los dispositivos móviles ha avanzado rápidamente en los últimos años, llegando a ser actualmente auténticas computadoras de bolsillo. La gran demanda por este tipo de dispositivos genera un gran interés por parte de empresas/instituciones que desean crear aplicaciones para un mercado en pleno auge, buscando aprovechar no sólo la gran cantidad de usuarios de estas plataformas, sino también la posibilidad de ofrecer funcionalidades y capacidades imposibles para sus procesos actuales.

La plataforma que actualmente posee una mayor cantidad de usuarios y mayor crecimiento es Android, debido a que se trata de un Sistema Operativo abierto que cualquier fabricante puede adaptar e instalar en sus dispositivos, que está en constante evolución, y que aporta gran cantidad de servicios y aplicaciones.

Por ello, el objetivo del presente trabajo de graduación es aplicar dichas tecnologías, los conocimientos y competencias adquiridas a lo largo de la carrera, en la construcción de un producto de software que satisface las necesidades y genera valor agregado a un cliente determinado. Dicho producto es una aplicación web embebida en dispositivos móviles, que es el tipo de producto que experimenta un vertiginoso crecimiento en la actualidad. determinado.

## Objetivos del sistema

Los principales objetivos del sistema a desarrollar son:

- Permitir que los usuarios del sistema puedan realizar el estudio de campo de una manera rápida y eficiente valiéndose de la tecnología de un dispositivo móvil.
- El sistema cuenta de dos partes, una aplicación móvil para generar registros de las



muestras del estudio de campo, y una aplicación web para gestionar y administrar dichos registros.

El sistema debe diseñarse para:

- Asegurar la escalabilidad de los requisitos.
- Mantener de forma sencilla la plataforma.
- Promover la seguridad de la información en todas sus capas.
- Ser fácil de usar.

## Selección del modelo de ciclo de vida y de la metodología de desarrollo

El desarrollo iterativo evolutivo, en contraste con el ciclo de vida de cascada o secuencial, consiste en la programación y pruebas tempranas de un sistema parcial en ciclos repetitivos. Normalmente supone que el desarrollo se inicia antes de que todos los requisitos estén definidos en detalle; el feedback se utiliza para aclarar y mejorar las características cambiantes. El prototipado como ciclo de vida se basa en la construcción de un prototipo que ayude a comprender los requisitos del sistema. Los prototipos se usan para verificar la viabilidad del diseño del software. Sirven como una herramienta iterativa del desarrollo del software donde el prototipo evoluciona hasta llegar al sistema final. La metodología Script o V-Script es una metodología de desarrollo de software que tiene un alto componente dinámico, orientado hacia la interfaz de usuario. Se adapta perfectamente al paradigma de orientación a objetos, aunque se han usado técnicas Script en metodologías estructuradas para el diseño de interfaz de usuario. Mediante el proceso Script se capturan las necesidades del usuario con la construcción de maquetas o prototipos desechables, tratando de capturar la expectativa del usuario: qué es lo que el usuario espera que haga el producto. A su vez, define las interfaces de usuario y permite integrar los aspectos del modelo estático y funcional.

# Disciplina de Requisitos

## Introducción

Esta especificación tiene como objetivo analizar y documentar las necesidades funcionales que deberán ser soportadas por el sistema a desarrollar. Para ello, se identificarán los requisitos que ha de satisfacer el nuevo sistema mediante entrevistas, el estudio de los problemas de las unidades afectadas y sus necesidades actuales. Además de identificar los requisitos se deberán establecer las prioridades, lo cual proporciona un punto de referencia para validar el sistema final que compruebe que se ajusta a las necesidades del usuario.

## Identificación de usuarios participantes

Los objetivos de esta tarea son identificar a los responsables de cada una de las unidades y a los principales usuarios implicados. Para ello se consideran los siguientes aspectos:

- Incorporación de usuarios al equipo de proyecto.
- Conocimiento de los usuarios de las funciones a automatizar.
- Repercusión del nuevo sistema sobre las actividades actuales de los usuarios.
- Implicaciones legales del nuevo sistema.

Se identificaron los siguientes usuarios:

- *Grupo de Administradores:* Formado por los solicitantes del software en cuestión.

- *Grupo de Alumnos:* Formado principalmente por alumnos de escuelas/colegios que realizan muestras, las cuales generan registros en el sistema.

Es de destacar la necesidad de una participación activa de los usuarios del futuro sistema en las actividades de desarrollo del mismo, con objeto de conseguir la máxima adecuación del sistema a sus necesidades y facilitar el conocimiento paulatino de dicho sistema, permitiendo una rápida implantación.

## Educción de requisitos

### Estudio de Documentación. Planificación y Realización de Entrevistas

Esta tarea tiene como finalidad capturar los requisitos de usuarios para el desarrollo del sistema.

Para el análisis de requisitos se usaron distintas técnicas de educación de requisitos. Entre ellas el estudio de la documentación provista por parte del administrador; entrevistas abiertas y estructuradas, análisis del proceso actual.

## Especificación de Requisitos de Software

### Introducción

Este documento es una Especificación de Requisitos Software de la Web App embebida en dispositivos móviles para la gestión de registros sobre la contaminación de afluentes y ríos. Esta documentación es fruto de las entrevistas, estudio de la documentación y del funcionamiento del proceso actual, así como del análisis llevado a cabo por el equipo de desarrollo.

El objetivo de la especificación es definir en forma clara, precisa, completa y verificable todas las funcionalidades y restricciones del sistema que se desea construir.

Esta documentación está sujeta a revisiones por el grupo de administradores que se recogerán por medio de sucesivas versiones del documento, hasta alcanzar la aprobación

por parte de los mismos. Una vez aprobado, servirá de base al equipo de desarrollo para la construcción del sistema en cuestión.

Esta especificación se ha realizado de acuerdo al estándar “IEEE Recommended Practice for software Requirements Specifications(IEEE/ANSI 830-1993)”.

## Objetivos y alcance del sistema

El presente proyecto tiene como objetivo principal ayudar al medio ambiente, utilizando un sistema de gestión que permite la creación y administración de registros, los cuales contienen datos de muestras del universo de estudio que, al ser procesadas, brinda el estado de contaminación de un río o afluente, mediante indicadores biológicos. Estos registros cuentan con contenido multimedia y coordenadas geograficas

## Definiciones, acrónimos y abreviaturas

### Definiciones:

- *Salida de campo*: La principal aportación de la salida de campo es que permite al alumnado adquirir un aprendizaje significativo en el que el principal elemento del proceso de enseñanza- aprendizaje es la construcción de significados. La persona aprende un concepto, un fenómeno, un procedimiento, un comportamiento, etc.
- *Insectos*: Nuestro universo de estudio obliga solo a tener en cuenta 4 insectos, los cuales sirven para indicar el posible grado de contaminación del agua. Estos insectos son los siguientes:
  - Elimidos.
  - Patudos.
  - Plecopteros.
  - Tricopteros.
- *Muestra*: Una muestra esta compuesta por los insectos encontrados en una salida de campo.

- *Indice de Contaminación:* Para el sistema, el indice de contaminación es el valor calculado, mediante la cantidad de diferentes insectos encontrados en el universo de estudio, de la siguiente manera:

Cantidad de insectos encontrados	Indice	Interpretación
0	0	Muy contaminado
1	1	Contaminado
2	2	Con contaminacion media
3	3	En buen estado
4	4	En excelente estado

- *Foto paisaje:* Foto obtenida del paisaje en donde se realizo la muestra de los insectos encontrados, con el fin de facilitar un punto de referencia visual para proximas salidas de campo.
- *Foto insectos:* Foto obtenida de la muestra que sirve para que los administradores de la aplicación validen, o no, el registro en cuestion.
- *Coordenadas geograficas:* Se usan para referenciar, mediante latitud y longitud, el lugar en donde se realizo el registro.
- *Mapa:* Mapa digital (Google Maps) en donde se muestran los registros realizados por los usuarios.
- *HTML:* HyperText Markup Language. Lenguaje de marcado para la elaboracion de paginas web. En nuestro caso, genera la vista final del sistema.
- *HTTPS:* Hypertext Transfer Protocol Secure.
- *BD:* Base de datos.
- *CRUD:* Es el acrónimo de “Crear, Leer, Actualizar y Borrar” (del original en ingles: Create, Read, Update and Delete), que se usa para referirse a las funciones básicas en bases de datos.

- *MCVS*: Modelo de Ciclo de Vida del Sistema.

## Descripción general

Esta sección nos presenta una descripción general del sistema con el fin de conocer las funciones que debe soportar, los datos asociados, las restricciones impuestas y cualquier otro factor que pueda influir en la construcción del mismo. El sistema nace como necesidad de un grupo de docentes de la Facultad de Ciencias Naturales e Instituto Miguel Lillo, por llevar un control mas exhaustivo de su investigación. Es prioritario el seguimiento de la contaminación de los afluentes y ríos ubicados en la provincia de Tucumán.

## Proceso actual

Este seguimiento se realizaba mediante salidas de campos con alumnos de las escuelas rurales donde realizaban las siguientes actividades.

- Los alumnos obtienen muestras del río o afluente intentando capturar algunos de los insectos del universo de estudio.
- Los docentes a cargo verifican dichas muestras, e identifican las coincidencias, obteniendo de esta forma, un indice de contaminación.

Todo esto se realiza de manera manual, anotando en papel y luego es transcripto a una planilla Excel. Todo el procedimiento antes descripto, dificulta la trazabilidad y administración de la información (introduciendo errores y demoras por manejo manual de la información), por lo que todo esto sería más efectivo y sencillo con la ayuda de la tecnología.

## Reingeniería de proceso

El objetivo de este proyecto se basa en proporcionar facilidades al proceso actual de la siguiente forma:

- Dos imágenes capturadas con la cámara de fotos del Smartphone:

- La primera imagen será una foto de los insectos encontrados en un río o afluente. El objetivo es encontrar 4 insectos diferentes para analizar la biodiversidad.
- La segunda imagen será una foto del paisaje que servirá como un futuro punto de referencia para próximas salidas de campo.
- Capturar coordenadas de manera automática con una precisión propia al GPS integrado del Smartphone, las cuales se guardaran como *latitud* y *longitud*
- El usuario deberá seleccionar, según su criterio personal, cuales de los 4 insectos fueron encontrados por él, mediante un formulario interactivo, el cual consta de imágenes reales de los mismos para una buena comparación y un campo de observaciones para realizar comentarios subjetivos sobre la muestra en cuestión.
- Al completar toda la información mencionada anteriormente, se mostrara una ruleta animada virtual, la cual mostrará un valor (índice de contaminación), indicando el posible grado de contaminación del agua en donde se realizó la muestra.
- Lo anterior se realizará sin conexión a internet (2G, 3G, Wifi, etc), generando un registro de manera local, que luego, de manera automática, se subirá a los servidores al momento de adquirir alguna conexión a internet.
- Los administradores podrán gestionar, mediante un navegador web (Google Chrome, Internet Explorer, Mozilla, etc) o desde la aplicación en el Smartphone, los registros previamente creados y guardados en el servidor.
- Analizando todos los registros, se creará un renderizado de un mapa (Google Maps) en donde los administradores y usuarios podrán visualizar, con trazos de diferentes colores, el grado de contaminación del agua en el curso de los ríos o afluentes analizados en las salidas de campo.

El sistema debe ser seguro, escalable, de fácil mantenimiento y muy simple de usar utilizando sólo interfaces táctiles para los usuarios finales. El futuro sistema llevará el nombre *Agüita*.

Las funciones que debe realizar el sistema se pueden agrupar de la siguiente manera:

- *Gestión de usuarios:* Debe permitir gestionar los usuarios (CRUD). Los mismos pueden ser usuarios o administradores del sistema. Para que un usuario pueda generar un registro, deberá estar previamente registrado e iniciar sesión por una única vez en su Smartphone. Los usuarios deben poder configurar/modificar las opciones de su perfil como ser: nombre, apellido, lugar de residencia, institución a la que pertenece y grado correspondiente. Los usuarios no pueden consultar la información personal de otros usuarios.
- *Gestión de registros:* Debe permitir gestionar los registros (CRUD). Los usuarios generan registros. Los administradores pueden interactuar con los mismo cambiando su estado (pendiente - valido - invalido) según la información brindada por los mismos (no verificada por los administradores - correcta - incorrecta) respectivamente.
- *Consultas de registros realizados:* Los usuarios podrán consultar un listado de sus registros creados con la información completa. Los usuarios no podrán consultar los registros de otros usuarios. Los administradores podrán ver los registros creados por los todos los usuarios de la aplicación con toda su información correspondiente.
- *Consulta de mapa:* Los usuarios y administradores pueden ver el mapa final con toda la información recopilada de todos los registros creados por los mismos. Todos los registros que estén en un estado *rechazado* no se eliminaran de la base de datos, no obstante, los mismos no se tomaran en cuenta para el renderizado del mapa final.

## Requisitos específicos

### A. Gestión de Usuarios

- **Alta de usuarios:**

**Introducción:** El sistema permite introducir información sobre usuarios en la aplicación.

**Entrada:** Email + Usuario + Contraseña + Nombre + Apellido + Institución + Grado + Residencia + Rol + Foto Perfil + Estado

**Proceso:** El sistema comprueba la inexistencia previa de un usuario, buscando



coincidencias en el nombre de usuario y mail. Además genera un IdUsuario a partir del máximo existente hasta el momento. En caso de éxito, se devolverá un mensaje de éxito y el IdUsuario. En caso de error se devolverá un mensaje con el motivo del mismo.

**Salida:** @IdUsuario + mensaje

- **Modificación de usuarios:**

**Introducción:** El sistema permite modificar información sobre usuarios existentes en la aplicación.

**Entrada:** @IdUsuario + Nombre + Apellido + Institución + Grado + Residencia + Foto Perfil

**Proceso:** El sistema comprueba la existencia previa de un usuario en base a @IdUsuario y actualiza la información del mismo. En caso de éxito, se devolverá un mensaje de éxito y el IdUsuario. En caso de error se devolverá un mensaje con el motivo del mismo.

**Salida:** @IdUsuario + mensaje

- **Cambiar estado de usuarios:**

**Introducción:** El sistema permite habilitar/inhabilitar usuarios existentes en la aplicación.

**Entrada:** @IdUsuario + Estado

**Proceso:** El sistema comprueba la existencia previa de un usuario en base a @IdUsuario cambiando el estado del mismo. Los registros asociados a este usuario no deben eliminarse ni modificarse. En caso de error se devolverá un mensaje con el motivo del mismo.

**Salida:** @IdUsuario + Mensaje

- **Ver detalles de usuario:**

**Introducción:** El sistema permite ver detalles relacionados a los usuarios existentes en él. Se debe visualizar usuario, nombre, apellido, institución, grado,

residencia, foto de perfil y cantidad de registros generados

**Entrada:** @IdUsuario

**Proceso:** El sistema comprueba la existencia previa del usuario en base a @IdUsuario. En caso de éxito, se presenta la información del mismo. En caso de error se devolverá un mensaje con el motivo del mismo.

**Salida:** Usuario + Nombre + Apellido + Institución + Grado + Residencia + Email + Foto Perfil + CantidadRegistros

- **Cambiar contraseña de usuario:**

**Introducción:** El sistema permite cambiar la contraseña a los usuarios existentes en él.

**Entrada:** @IdUsuario + ContraseñaAnterior + ContraseñaNueva

**Proceso:** El sistema comprueba la existencia previa del usuario en base a @IdUsuario, luego se impactara la nueva contraseña, dejando en desuso la anterior. En caso de éxito, se presenta la información del mismo. En caso de error se devolverá un mensaje con el motivo del mismo.

**Salida:** @IdUsuario + Mensaje

- **Cambiar rol de usuario:**

**Introducción:** El sistema permite cambiar el Rol a los usuarios existentes en él. Esta acción solo la pueden realizar los usuarios del grupo “Administradores”

**Entrada:** @IdUsuario + Rol

**Proceso:** El sistema comprueba la existencia previa del usuario en base a @IdUsuario, luego modificara el Rol del mismo. En caso de éxito, se presenta la información del mismo. En caso de error se devolverá un mensaje con el motivo del mismo.

**Salida:** @IdUsuario + Mensaje

- **Búsqueda de usuarios:**

**Introducción:** El sistema permite introducir parámetros con los que se buscará

usuarios que coincidan con los mismos.

**Entrada:** Usuario o Nombre o Apellido o Email

**Proceso:** El sistema lista al usuario que cumpla con los parámetros de búsqueda en caso de coincidencia. En caso de no encontrar algún usuario, se mostrara un mensaje vacío, indicando que la búsqueda no arrojo resultados.

**Salida:** Usuario + Nombre + Apellido + Institución + Grado + Residencia + Email + Foto Perfil + CantidadRegistros

## B. Gestión de Registros

- **Alta de registros:**

**Introducción:** El sistema permite dar de alta un nuevo registro ingresando indice, insectos encontrados, fecha, latitud, longitud, foto del paisaje, foto de la muestra, foto del mapa (vista aérea con un PIN indicando la ubicación terrestre), observaciones del usuario, IdUsuario (creador del registro), IdUbicacion (país, localidad, provincia).

**Entrada:** Indice + InsectosEncontrados + Fecha Creación + Latitud + Longitud + FotoPaisaje + FotoMuestra + FotoMapa + ObservacionesUsuario + IdUsuario + IdUbicación

**Proceso:** El sistema crea un registro con un valor de estado de validación inicial igual a "Pendiente" con observaciones del administrador sin contenido, además, codifica todas las fotos y genera un IdRegistro a partir del máximo existente hasta el momento. En caso de éxito, se devolverá un mensaje de éxito y el IdRegistro.

**Salida:** IdRegistro + Mensaje

- **Cambiar estado de registros:**

**Introducción:** El sistema permite cambiar el estado de un registro. Solo los administradores del sistema tendrán los permisos para realizar esta acción. Se podrá cambiar el estado del registro "Valido" / "Invalido" y viceversa. Inicialmente, el registro se crea con un estado "Pendiente", el cual, una vez modificado, no se

podrá volver a asignar.

**Entrada:** @IdRegistro + Estado Validación

**Proceso:** El sistema modifica el registro con el valor de estado validación correspondiente. En caso de éxito, se devolverá un mensaje de éxito y el IdRegistro.

**Salida:** IdRegistro + Mensaje

- **Asignar observaciones de administrador a registros:**

**Introducción:** El sistema permite agregar observaciones de administrador al registro. Solo los administradores del sistema tendrán los permisos para realizar esta acción.

**Entrada:** @IdRegistro + ObservacionesAdministrador

**Proceso:** El sistema modifica el registro agregando una observación de administrador. En caso de éxito, se devolverá un mensaje de éxito y el IdRegistro.

**Salida:** IdRegistro + Mensaje

- **Ver detalles de registro:**

**Introducción:** El sistema permite ver detalles relacionados a los registros existentes en él. Se debe visualizar índice, insectos encontrados, fecha, latitud, longitud, foto del paisaje, foto de la muestra, foto del mapa, observaciones del usuario, estado de validación, usuario que lo creó, país, provincia, localidad.

**Entrada:** @IdRegistro

**Proceso:** El sistema comprueba la existencia previa del registro en base a @IdRegistro. En caso de éxito, se presenta la información del mismo. En caso de error se devolverá un mensaje con el motivo del mismo.

**Salida:** Índice + InsectosEncontrados + Fecha Creación + Latitud + Longitud + FotoPaisaje + FotoMuestra + FotoMapa + ObservacionesUsuario + ObservacionesAdministrador + Estado Validación + IdUsuario + IdUbicación

- **Búsqueda de registros**

**Introducción:** El sistema permite buscar registros filtrando por los registros

creados desde un intervalo de fechas, por estado de validación, por índice.

**Entrada:** Fecha Inicio Fecha Fin + EstadoValidación + Índice

**Proceso:** El sistema lista los registros que cumplan con los parámetros de búsqueda en caso de coincidencia. En caso de no encontrar algún registro, se mostrara un mensaje vacío, indicando que la búsqueda no arrojo resultados. Si la búsqueda no contiene parámetros, se listan todos los registros existentes en el sistema

**Salida:** Índice + InsectosEncontrados + Fecha Creación + Latitud + Longitud + ObservacionesUsuario + ObservacionesAdministrador + Estado Validación + IdUsuario + IdUbicación

### C. Gestión de Mapas

- **Ver mapa**

**Introducción:** El sistema permite ver el mapa interactivo con la información de los registros almacenados que cumplan con la condición de estado de validación igual a "Valido".

**Entrada:** Arreglo de Registros

**Proceso:** Mostrar un mapa con puntos obtenidos mediante la latitud y longitud de cada registros del arreglo. Los puntos indican, ademas de la posición geográfica del registro, el usuario que lo creó y el índice de contaminación numéricamente y ademas con un color de entre 4 diferentes para una rápida identificación visual.

**Salida:** Mapa + Puntos Geográficos

### Suposiciones y Dependencias:

- **Suposiciones:** Se asume que los requisitos en este documento son estables una vez que sean aprobados por los responsables de la aplicación. Cualquier petición de cambios en la especificación debe ser aprobada por todas las partes intervinientes y será gestionada por el equipo de desarrollo.
- **Dependencias:** El sistema trabaja en conjunto con Google Maps y el sistema de posicionamiento global mediante satélites, algún cambio que se realicen en estos, el sistema podría presentar inconsistencias, errores, y hasta dejar de funcionar.

## Requisitos de Usuario y Tecnológicos:

- **Requisitos de usuario:** Como se mencionó anteriormente, se identifican dos tipos de usuarios: Administradores y Alumnos. Los usuarios tendrán sus cuentas asociadas con Nombre de Usuario y Contraseña. Los mismos podrán iniciar sesión desde computadoras de escritorio o mediante la aplicación para dispositivos móviles Smartphones, siempre y cuando, dispongan de una cuenta valida. En caso contrario, deberán registrarse en el sistema mediante la aplicación correspondiente.
- **Requisitos tecnológicos:** Los administradores podrán iniciar sesión en el sistema mediante computadoras de escritorio, facilitando la gestión y administración del mismo. Ésta versión WEB restringe el uso a aquellos usuarios del grupo “Alumnos”, por otra parte, los alumnos podrán hacer uso del sistema solo para generar registros, y ver el mapa interactivo mediante la aplicación para Smartphones que deberán descargar la tienda. Se utilizará una plataforma de servicios en la nube o un servidor físico local provisto por el cliente de este sistema, para administrar una máquina virtual que hará de servidor web y servidor de base de datos. El sistema operativo que correrá la máquina virtual será Ubuntu Server, éste es un sistema operativo gratuito y de código abierto, por otra parte, el sistema gestor de base de datos será del tipo MySQL. Ambos sistemas (operativo y gestor de base de datos), serán instalados en su ultima versión estable al momento de la entrega del software. El sistema se ejecutara sobre un esquema de peticiones Cliente/Servidor (API Rest). La elección esta infraestructura se debe principalmente a 3 motivos:
  - Debido a que el sistema se puede usar mediante Smartphones y computadoras de escritorio de forma remota, la solución fue dividir y separar, como ya se menciona, la aplicación para los usuarios de el servidor de consultas y base de datos.
  - Experiencia del equipo de desarrollo
  - Sistemas seleccionados de licencia gratuita.

## Requisitos de Interfaces Externas

- **Interfaz de usuario:** Las interfaces de la aplicación deben ser intuitivas, fáciles de usar, amigables y de respuesta rápida. La interfaz de usuario debe ser orientada al uso táctil de los Smartphones.
- **Interfaz Hardware:**
  - Requisitos para los Smartphones:
    - Los Smartphones de los usuarios que ejecutaran la aplicación deberán tener las siguientes características independientemente de su S.O:
      - ◇ Cámara fotográfica de 1 Mega Píxeles o más.
      - ◇ GPS integrado.
      - ◇ Conexión a internet vía WiFi o Red GSM
      - ◇ Pantalla táctil de 3.5" o superior.
      - ◇ Espacio disponible de 10 MB para la instalación de la aplicación + Cantidad de MB variable ocupado por cada registro creado.
  - Requisitos mínimos para el servidor:
    - Procesador AMD Sempron 3000 o equivalente o procesador Intel Celeron o equivalente. Capacidad de virtualización
    - 1 GB de memoria RAM.
    - Conexión a internet.
  - Requisitos mínimos para las PC o notebook de los administradores
    - Procesador AMD Sempron 3000 o equivalente. Procesador Intel Celeron o equivalente.
    - 1 GB de memoria RAM.
    - Periféricos de entrada/salida.
    - Conexión a internet
- **Interfaz Software:** Sistemas operativos soportados por la aplicación:
  - Smartphones

- Android 4.0 o posterior.
  - iOS 9.0 o posterior.
- Servidor
  - El sistema operativo será Ubuntu Server en su ultima versión estable LTS.
- PC o notebook de los administradores
  - Cualquier sistema operativo con navegador web

### **Requisitos de Rendimiento:**

El Tiempo de respuesta de la aplicación de cada función solicitada por el usuario no debe ser superior a los 3 segundos en una velocidad efectiva de conexión con el servidor a través de 3G. No obstante, los registros al generarse de manera offline (sin conexión a internet), se guardarán de manera local, por lo que si se generaron varios registros, al momento de que el Smartphone detecte conexión a internet, el tiempo de respuesta se ve incrementando directamente proporcional a la cantidad de registros que se estén subiendo al servidor en la nube en ese momento.

### **Requisitos de Desarrollo y Restricciones de Diseño:**

El ciclo de vida será Prototipado Evolutivo, debiendo orientarse hacia el desarrollo de un sistema flexible que permita incorporar de manera sencilla cambios y nuevas funcionalidades.

### **Ajuste a estándares:**

Interfaz de usuario basada Material Desing (Google)

### **Seguridad:**

- **En desarrollo:** Los desarrolladores acceden a la gestión del sistema operativo y/o sus aplicaciones a través de Secure Shell o SSH, el estándar de facto para la administración remota de servidores de manera segura. Tanto para la administración propia



del servidor, como en las aplicaciones que se utilizan para administrar la base de datos (MySQL Workbench) y la gestión de archivos (SFTP), se realizan con clientes que establecen conexiones seguras con el servidor. La técnica empleada para dichas conexiones es el intercambio de claves públicas y validación con la clave privada de las aplicaciones cliente.

- **En producción:** Los usuarios del sistema acceden al mismo mediante la aplicación móvil o cualquier explorador web. Para operar con la aplicación, deben proveer un usuario y una contraseña. Esta información es procesada en el servidor en un proceso de validación de credenciales y devuelve al cliente el mensaje de inicio de sesión correcto. La comunicación entre cliente servidor viajara encriptada mediante el protocolo HTTPS.
- **Roles y permisos:** Para reforzar la seguridad de la aplicación, cada usuario posee un rol (“Administrador” o “Alumno”), el cual tiene asociados diferentes permisos. Esto permite restringir el acceso a usuarios del grupo “Alumno” a la aplicación WEB de gestión y administración de registros. El rol “Administrador” tiene todos los permisos y privilegios, pudiendo así, gestionar el sistema e incluso generar registros como lo harían los Alumnos.
- **Red:** En la capa de red se crearon reglas de acceso al servidor mediante el uso del firewall provisto por el Sistema Operativo Ubuntu. A través de estas reglas se establece que servicios podrá ofrecer el servidor a los clientes de la aplicación.
- **Sistema Operativo:** En el sistema operativo se de finen los usuarios y los permisos que poseen para realizar lecturas, escrituras y/o ejecuciones de archivos alojados en la memoria del servidor. De esta forma se previene que usuarios no autorizados puedan modificar, eliminar y ejecutar archivos en el servidor, incluso en la base de datos.
- **Servidor web:** A nivel servidor web, se preve la implementación del protocolo de aplicación HTTPS que permite encriptar el trafico de información desde el servidor

web hacia los navegadores o aplicaciones que realicen solicitudes, estableciendo un eslabón mas en la seguridad del sistema.

- **Política de respaldo:** El administrador llevará a cabo un respaldo de datos en discos externos o en la nube por el tiempo que el considere necesario. Además, se exportaran los registros en un archivo excel manteniendo la información necesaria para la recuperación de los registros a futuro. Por otro lado, el motor de Base de datos estará configurado para realizar backups cada cierto intervalo de tiempo definido por el administrador. Conservar los respectivos archivos de respaldo de los últimos 6 backups.
- **Política de Borrado:** No se ha definido

## Estimación del proyecto

### Planificación de etapas

Código	Descripción	Fecha Inicio	Fecha Fin
A	Estudio de factibilidad y acciones preliminares	15/01/2017	25/01/2017
B	Educción de requisitos	01/02/2017	01/03/2017
C	Diseño del prototipo	02/03/2017	01/04/2017
D	Corroborar diseño con requisitos	01/04/2017	05/04/2017
E	Desarrollo del prototipo	10/04/2017	15/09/2017
F	Pruebas del prototipo	15/09/2017	15/10/2017
G	Refinamiento del prototipo	15/11/2017	01/02/2018
H	Análisis y evaluación el prototipo por parte del cliente	15/02/2018	16/02/2018
I	Refinamiento del prototipo	01/03/2018	01/06/2018
J	Entrega para producción	15/09/2018	20/09/2018
K	Seguimiento del sistema	20/09/2018	-

Cuadro 1: Fechas tentativas por etapa.



# Disciplina de Análisis

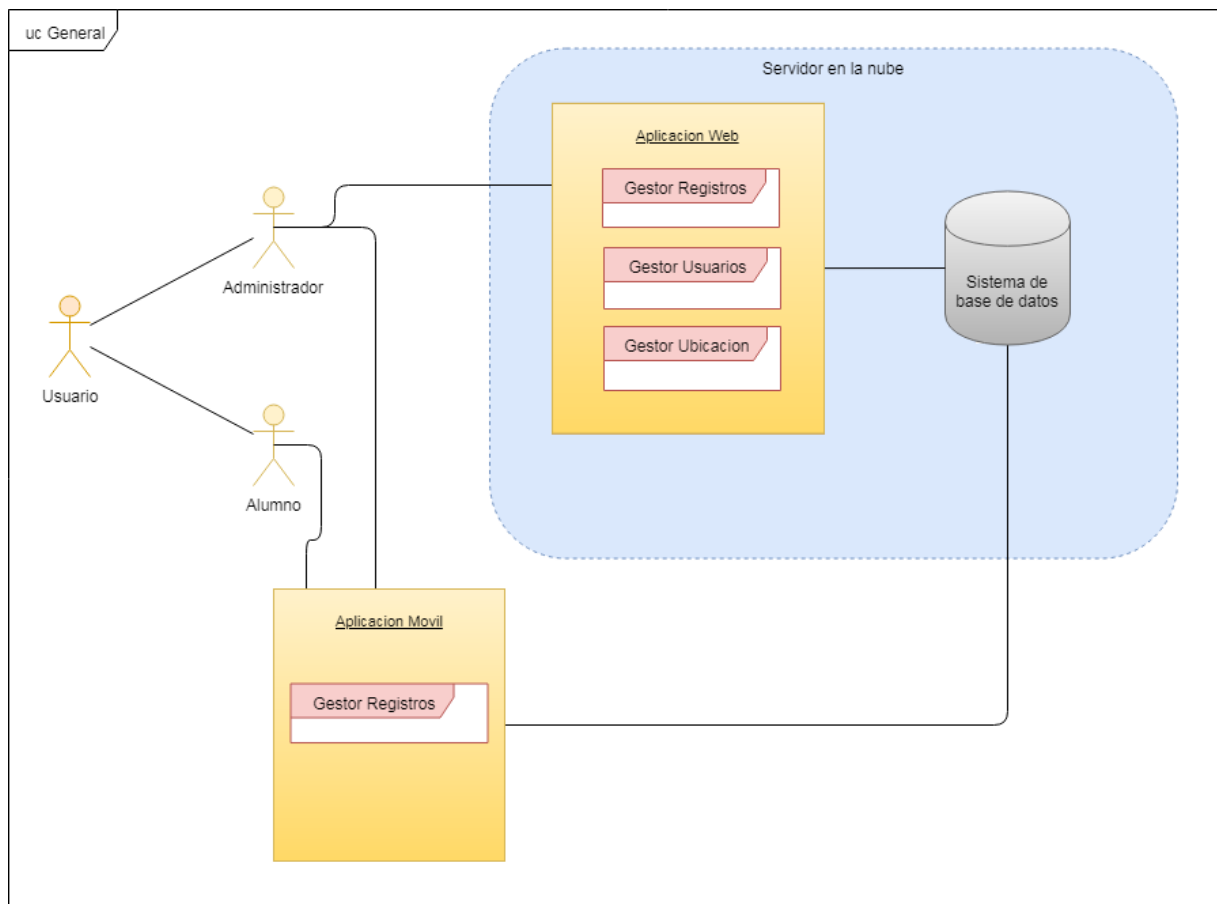
## Vista de Casos de Uso

La vista de casos de uso captura el comportamiento de un sistema, subsistema, clase o componente, como lo ve un usuario externo. Particiona la funcionalidad del sistema en transacciones significativas para los actores (usuarios idealizados) de un sistema. Las piezas de funcionalidad interactiva son llamadas “casos de uso”. Un caso de uso describe una interacción entre actores como una secuencia de mensajes entre el sistema y uno o más actores. El término *actor* incluye a personas, como también otros sistemas de computadora o procesos.

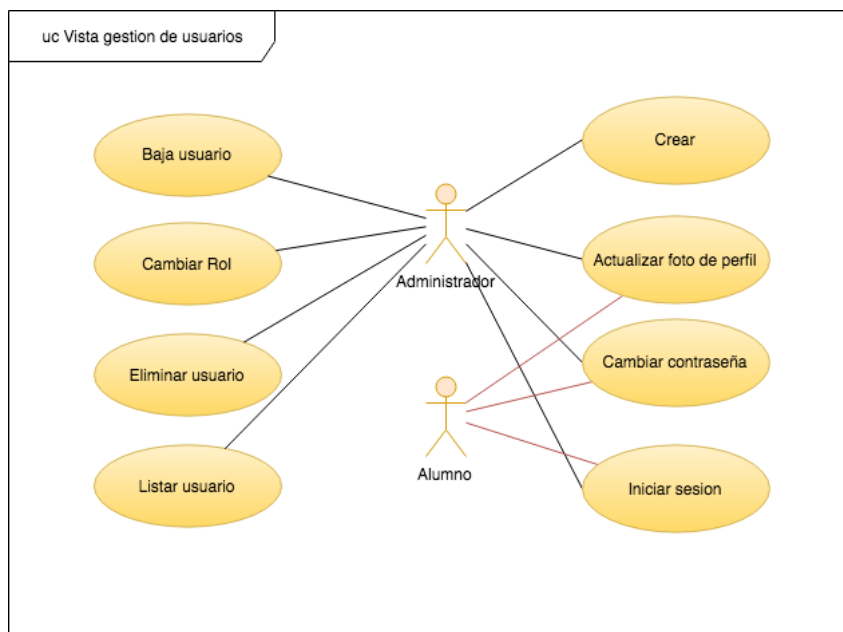
## Diagramas de Casos de Uso

Primeramente se muestra un diagrama de caso de uso general donde se agruparon los casos de uso por las acciones en común, luego se va a explorar cada caso de uso de manera mas descriptiva. Por ejemplo, en “Gestión de Registros” van a estar todos los casos de uso referidos a los mismos (alta, baja, listar, buscar, etc)

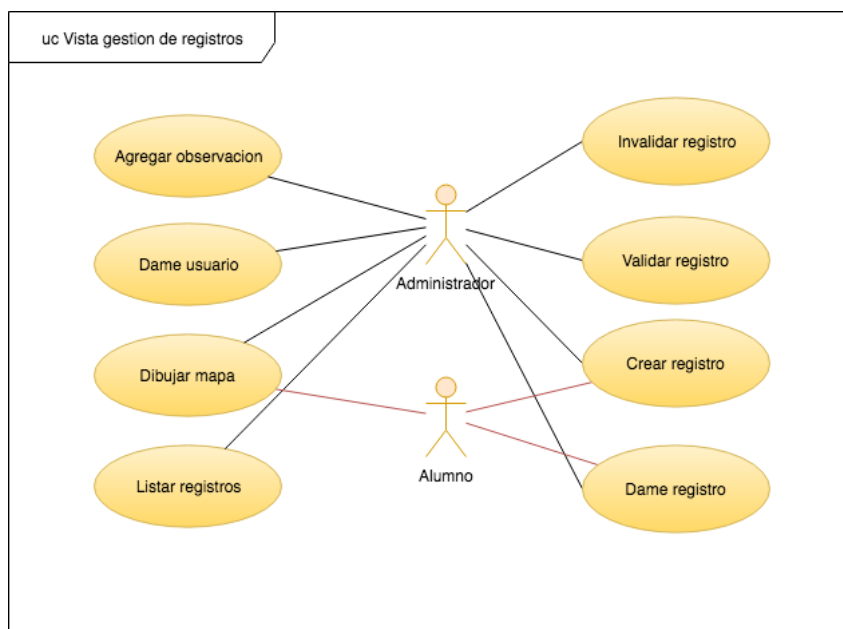
## Diagrama General de Casos de Uso



## Gestión de Usuarios



## Gestión de Registros



## Casos de Uso Principales

CU1: Un usuario no registrado desea registrarse en el sistema.

CU2: Un usuario registrado desea iniciar sesión en la aplicación móvil

- CU3: Un usuario registrado desea crear un registro.
- CU4: Un usuario registrado desea listar sus registros.
- CU5: Un usuario registrado desea ver un registro en particular.
- CU6: Un usuario registrado desea ver el mapa general.
- CU7: Un usuario registrado desea ver su perfil.
- CU8: Un usuario registrado desea modificar su perfil.
- CU9: Un usuario registrado desea cerrar sesión.
- CU10: Un usuario registrado desea eliminar su cuenta.
- CU11: Un administrador desea iniciar sesión.
- CU12: Un administrador desea borrar o dar de baja un alumno.
- CU13: Un administrador desea cambiar el rol a un usuario.
- CU14: Un administrador desea listar los alumnos.
- CU15: Un administrador desea listar los registros.
- CU16: Un administrador desea ver un registro en particular.
- CU17: Un administrador desea validar o invalidar registros.
- CU18: Un administrador desea buscar registros por fecha de creación.
- CU19: Un administrador desea buscar registros por institución.
- CU20: Un administrador desea buscar registros por índice.
- CU21: Un administrador desea ver el mapa general.
- CU22: Un administrador desea exportar a Excel la información.
- CU23: Un administrador desea cerrar sesión.

# Disciplina de Diseño

Por el Principio de Pareto<sup>1</sup>, se hicieron los diagramas relevantes. Como se procura tener homogeneidad en la implementación de las clases, basta un solo diagrama de cada tipo para una clase, para describir también el comportamiento de las otras clases.

## Descripción Textual de Casos de Uso

**Caso de Uso 01: Un usuario no registrado desea registrarse en el sistema.**

CU01	Registrar				
Revisa:		Fecha		Firma:	
<b>Resumen:</b> Este caso de uso permite a los usuarios registrarse como usuarios de la aplicación, permitiendo introducir sus datos personales					
<b>Actores:</b> Usuario (primario). Servidor (en adelante S. Secundario)					

---

<sup>1</sup>Cuando se habla de los costes de desarrollo de software enunciarse de la siguiente manera: “El 80 % del esfuerzo de desarrollo (en tiempo y recursos) produce el 20 % del código, mientras que el 80 % restante es producido con tan sólo un 20 % del esfuerzo”



**Personal Involucrado y Metas:**

*Usuario:* quiere transformarse en un usuario del sistema, así pueda realizar las transacciones con la aplicación de un modo seguro y personalizado.

*Servidor:* quiere registrar la mayor cantidad de usuarios posibles y que el proceso sea lo más rápido y seguro posible.

**Precondiciones:** El usuario no está registrado en la aplicación

**Poscondiciones:** Se registra al usuario como usuario de la aplicación. El usuario puede realizar operaciones en la aplicación.

**Escenario Principal:**

1. El usuario ejecuta la aplicación móvil (en adelante APP) en su Smartphone y decide registrarse.
2. APP muestra un formulario de carga donde ingresa sus datos personales y su nombre de usuario y contraseña.
3. APP verifica los datos ingresados.
4. APP solicita a S el registro del usuario.
5. S registra al usuario y lo informa a APP.
6. APP da la bienvenida al usuario.

**Flujos Alternativos:****A1: El sistema encuentra algún fallo para comunicarse con S**

La secuencia A1 comienza en el punto 4 del escenario principal.

5. APP informa al usuario el problema de conexión a través de un mensaje por la pantalla.

El escenario vuelve al punto 4.

**A2: Nombre de usuario existente**

La secuencia A2 comienza en el punto 4 del escenario principal.

5. S comunica que el nombre de usuario es existente.

El escenario vuelve al punto 3.

**A3: Contraseña inválida o no coincide con la confirmación**

La secuencia A3 comienza en el punto 3 del escenario principal.

4. APP informa el problema a través de un mensaje por pantalla.

El escenario vuelve al punto 2.

**A4: Dirección de correo electrónico existente**

La secuencia A4 comienza en el punto 4 del escenario principal.

5. S comunica que la dirección de correo electrónico es existente.

El escenario vuelve al punto 2.

**A5: Tipo de datos ingresados de manera incorrecta**

La secuencia A5 comienza en el punto 3 del escenario principal.

4. APP informa el problema a través de un mensaje por pantalla.

El escenario vuelve al punto 2.

**Requisitos de Interfaz de Usuario para todos los casos de uso:**

Smartphone con SO Android o iOS o Windows Mobile, con pantalla táctil, cámara y GPS integrado.

**Requisitos No-Funcionales para todos los casos de uso:**

*Tiempo de respuesta:* la interfaz debe responder dentro de un tiempo máximo de 3 segundos en una velocidad efectiva de conexión con el servidor a través de 3G.

*Disponibilidad:* debe poder accederse a toda hora, los 365 días del año.

## Caso de Uso 02: Un usuario registrado desea iniciar sesión en la aplicación móvil.

CU02	Iniciar Sesión				
Revisa:		Fecha		Firma:	
<b>Resumen:</b> Este caso de uso permite a los usuarios iniciar sesión con el nombre de usuario y contraseña de manera que el sistema le permita realizar tareas.					
<b>Actores:</b> Usuario (Primario). Servidor (en adelante S. Secundario)					
<b>Personal Involucrado y Metas:</b> <i>Usuario:</i> quiere que el sistema lo reconozca como tal, así pueda realizar las tareas con la aplicación de un modo seguro y personalizado. <i>App:</i> requiere identificar, de manera local, confiablemente a sus usuarios de manera de satisfacer sus intereses en cuando a seguridad, accesos a su cuenta personal y datos privados.					
<b>Precondiciones:</b> El usuario está registrado.					

**Poscondiciones:** Se identifica y autentica al usuario. Se conocen sus datos personales y opciones de personalización.

**Escenario Principal:**

1. El usuario ejecuta la aplicación móvil (en adelante APP) en su Smartphone.
2. APP descarga la informacion de inicio de sesión del servidor.
3. APP solicita al usuario su nombre de usuario y contraseña.
4. El usuario ingresa su nombre de usuario y contraseña.
5. APP realiza el proceso de validación del usuario
6. APP valida al usuario y comunica sus datos personales.
7. APP da la bienvenida al usuario

**Flujos Alternativos:**

**A1: Nombre de usuario inexistente**

La secuencia A1 comienza en el punto 4 del escenario principal.

5. S comunica que el nombre de usuario es inexistente.

El escenario vuelve al punto 2.

**A2: Nombre de usuario existente pero contraseña inválida**

La secuencia A2 comienza en el punto 4 del escenario principal.

4. S comunica que la contraseña es inválida.

El escenario vuelve al punto 2.

**Caso de Uso 03: Un usuario registrado desea crear un registro**

CU03	Crear Registro			
Revisa:		Fecha		Firma:
<b>Resumen:</b> Este caso de uso permite al usuario crear un registro de manera que el sistema lo almacene en una base de datos para luego mostrarlo como información del mapa general				
<b>Actores:</b> Usuario (Primario). Servidor (en adelante S. Secundario)				
<b>Personal Involucrado y Metas:</b> <i>Usuario:</i> quiere que el sistema lo reconozca como tal, así pueda realizar los registros a través de la aplicación móvil (en adelante APP) de un modo seguro y personalizado. <i>Servidor:</i> quiere identificar confiablemente a sus usuarios de manera de satisfacer sus intereses en cuanto a seguridad y datos privados.				
<b>Precondiciones:</b> Los usuarios deben estar autenticados en APP				
<b>Poscondiciones:</b> Se almacena un nuevo registro en el sistema con los datos correspondientes necesarios para luego aportar información al mapa general				

**Escenario Principal:**

1. El usuario selecciona la opción para crear registro.
2. APP solicita al usuario a través de un formulario los datos requeridos para la creacion del registro.
3. El usuario completa el formulario y presiona un botón para finalizar.
4. APP verifica que los tipos de datos ingresados en el formulario sean correctos y almacena el registro de forma local.
5. APP detecta conexion a internet y envía de manera segura los datos a S para que sean validados.
6. S valida los datos, realiza la creacion del registro y envía una confirmación.
7. APP informa al usuario que la operación se realizó exitosamente.

**Flujos Alternativos:****A1: El sistema encuentra algún fallo para comunicarse con S**

La secuencia A1 comienza en el punto 5 del escenario principal.

6. APP reintenta enviar la información de manera automática por tiempo indefinido.

El escenario vuelve al punto 5.

**Caso de Uso 04: Un usuario registrado desea listar sus registros.**

CU04	Listar Registros				
Revisa:		Fecha		Firma:	

**Resumen:** Este caso de uso permite a un usuario listar la información de todos los registros que realizo.

**Actores:** Usuario (Primario). Servidor (Secundario).

**Personal Involucrado y Metas:**

*Usuario:* quiere visualizar de manera completa la información de los registros realizados.

*Servidor:* quiere que el usuario pueda ver la información relacionada con sus registros de manera segura.

**Precondiciones:** El usuario debe estar autenticado en APP.

**Poscondiciones:** Se muestra un listado de registros realizados.

**Escenario Principal:**

1. El usuario selecciona la opción para listar los registros realizados.
2. APP envía la solicitud a S.
3. S envía los datos necesarios para generar el listado de registros a APP.
4. APP muestra el listado al usuario.

**Flujos Alternativos:****A1: El sistema encuentra algún fallo para comunicarse con S**

La secuencia A1 comienza en el punto 2 del escenario principal.

3. APP informa al usuario el problema de conexión a través de un mensaje por la pantalla.

El escenario vuelve al punto 2.

### Caso de Uso 05: Un usuario registrado desea ver un registro en particular

CU05	Ver Registro			
Revisa:		Fecha		Firma:
<b>Resumen:</b> Este caso de uso permite al usuario ver un registro específico que él realice.				
<b>Actores:</b> Usuario (Primario). Servidor (Secundario)				
<b>Personal Involucrado y Metas:</b> <i>Usuario:</i> quiere visualizar de manera completa la información de un registro, realizado por él, en particular. <i>Servidor:</i> quiere que el usuario pueda ver la información relacionada con su registro de manera segura.				
<b>Precondiciones:</b> El usuario debe estar autenticado en APP.				
<b>Poscondiciones:</b> Se muestra la información completa de un registro.				



**Escenario Principal:**

1. El usuario selecciona la opción para ver un registro realizado.
2. APP envía la solicitud a S.
3. S envía los datos necesarios para generar la vista de un registro a APP.
4. APP muestra el registro al usuario.

**Flujos Alternativos:****A1: El sistema encuentra algún fallo para comunicarse con S**

La secuencia A1 comienza en el punto 2 del escenario principal.

3. APP informa al usuario el problema de conexión a través de un mensaje por la pantalla.

El escenario vuelve al punto 2.

**Caso de Uso 06: Un usuario registrado desea ver el mapa general.**

CU06	Ver Mapa				
Revisa:		Fecha		Firma:	
<b>Resumen:</b> Este caso de uso permite al usuario ver el mapa general con la información de todos los registros.					
<b>Actores:</b> Usuario (Primario). Servidor (Secundario)					

**Personal Involucrado y Metas:**

*Usuario:* quiere visualizar de manera completa la información del mapa general.

*Servidor:* quiere que el usuario pueda ver la información relacionada con el mapa general de forma segura.

**Precondiciones:**

El usuario debe estar autenticado en APP.

Debe existir al menos 1 registro en el servidor.

**Poscondiciones:** Se muestra la informacion completa del mapa general.

**Escenario Principal:**

1. El usuario selecciona la opción para ver el mapa general.
2. APP envía la solicitud a S.
3. S envía los datos necesarios para generar la vista del mapa general a APP.
4. APP muestra el mapa general al usuario.

**Flujos Alternativos:****A1: El sistema encuentra algún fallo para comunicarse con S**

La secuencia A1 comienza en el punto 2 del escenario principal.

3. APP informa al usuario el problema de conexión a través de un mensaje por la pantalla.

El escenario vuelve al punto 2.

**Caso de Uso 07: Un usuario registrado desea ver su perfil.**

CU07	Ver Perfil de Usuario			
Revisa:		Fecha		Firma:
<b>Resumen:</b> Este caso de uso permite al usuario ver su perfil con la informacion de sus datos.				
<b>Actores:</b> Usuario (Primario). Servidor (Secundario)				
<b>Personal Involucrado y Metas:</b> <i>Usuario:</i> quiere visualizar de manera completa la información de su perfil. <i>Servidor:</i> quiere que el usuario pueda ver la información relacionada con su perfil de forma segura.				
<b>Precondiciones:</b> El usuario debe estar autenticado en APP.				
<b>Poscondiciones:</b> Se muestra la informacion completa del perfil del usuario.				
<b>Escenario Principal:</b> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción para ver su perfil.</li> <li>2. APP envía la solicitud a S.</li> <li>3. S envía los datos necesarios para generar la vista del perfil a APP.</li> <li>4. APP muestra el perfil de usuario por pantalla.</li> </ol>				

**Flujos Alternativos:****A1: El sistema encuentra algún fallo para comunicarse con S**

La secuencia A1 comienza en el punto 2 del escenario principal.

3. APP informa al usuario el problema de conexión a través de un mensaje por la pantalla.

El escenario vuelve al punto 1.

**Caso de Uso 08: Un usuario registrado desea modificar su perfil.**

CU08	Modificar Perfil de Usuario			
Revisa:		Fecha		Firma:
<b>Resumen:</b> Este caso de uso permite al usuario editar los datos de su perfil.				
<b>Actores:</b> Usuario (Primario). Servidor (Secundario)				
<b>Personal Involucrado y Metas:</b> <i>Usuario:</i> quiere editar los datos de su perfil. <i>Servidor:</i> quiere mantener actualizados los datos del usuario.				
<b>Precondiciones:</b> El usuario debe estar autenticado en APP.				
<b>Poscondiciones:</b> Se registran los cambios de los datos del perfil de usuario en el servidor.				

**Escenario Principal:**

1. El usuario selecciona la opción para editar su perfil.
2. APP solicita al usuario a través de un formulario los datos de su perfil que pueden ser modificados o mantenidos.
3. El usuario completa el formulario y presiona un botón para finalizar.
4. APP envía de manera segura los datos a S para que sean validados.
5. S valida los datos, realiza la actualización y envía una confirmación.
6. APP informa al usuario que la operación se realizó exitosamente.

**Flujos Alternativos:****A1: El sistema encuentra algún fallo para comunicarse con S**

La secuencia A1 comienza en el punto 4 del escenario principal.

5. APP informa al usuario el problema de conexión a través de un mensaje por la pantalla.

El escenario vuelve al punto 2.

**Caso de Uso 10: Un usuario registrado desea eliminar su cuenta.**

CU10	Eliminar Cuenta De Usuario			
Revisa:		Fecha		Firma:
<b>Resumen:</b> Este caso de uso permite al usuario eliminar su cuenta de usuario del servidor.				
<b>Actores:</b> Usuario (Primario). Servidor (Secundario)				

**Personal Involucrado y Metas:**

*Usuario:* quiere eliminar su cuenta de usuario.

*Servidor:* quiere mantener actualizados los datos del usuario.

**Precondiciones:** El usuario debe estar autenticado en APP.

**Poscondiciones:** Se registran los cambios de los datos del perfil de usuario en el servidor.

**Escenario Principal:**

1. El usuario selecciona la opción para eliminar su perfil.
2. APP solicita al usuario a través de una alerta la confirmación para eliminar el perfil.
3. El usuario presiona el botón aceptar.
4. APP envía la solicitud a S.
5. S envía confirmación.
6. APP informa al usuario que la operación se realizó exitosamente.

**Flujos Alternativos:****A1: El sistema encuentra algún fallo para comunicarse con S**

La secuencia A1 comienza en el punto 4 del escenario principal.

5. APP informa al usuario el problema de conexión a través de un mensaje por la pantalla.

El escenario vuelve al punto 1.

**Caso de Uso 11: Un administrador desea iniciar sesión.**

CU11	Iniciar Sesión				
Revisa:		Fecha		Firma:	
<b>Resumen:</b> Este caso de uso permite a los administradores iniciar sesión con el nombre de usuario y contraseña de manera que el sistema le permita realizar tareas.					
<b>Actores:</b> Administrador (Primario). Servidor (en adelante S. Secundario)					
<b>Personal Involucrado y Metas:</b> <i>Administrador:</i> quiere que el sistema lo reconozca como tal, así pueda realizar las tareas de gestión con la aplicación de un modo seguro y personalizado. <i>Servidor:</i> requiere identificar confiablemente a sus administradores de manera de satisfacer sus intereses en cuando a seguridad, accesos a su cuenta personal y datos privados.					
<b>Precondiciones:</b> El administrador está registrado.					
<b>Poscondiciones:</b> Se identifica y autentica al administrador. Se conocen sus datos personales y opciones de personalización.					

**Escenario Principal:**

1. El administrador ejecuta el sistema de gestion de registros mediante (en adelante SGR) una URL en un navegador web.
2. SGR solicita al administrador su nombre de usuario y contraseña.
3. El administrador ingresa su nombre de usuario y contraseña.
4. SGR solicita a S la validación del administrador
5. S valida al administrador y comunica sus datos personales.
6. SGR da la bienvenida al administrador

**Flujos Alternativos:****A1: El sistema encuentra algún fallo para comunicarse con S**

La secuencia A1 comienza en el punto 4 del escenario principal.

5. SGR informa al administrador el problema de conexión a través de un mensaje por la pantalla.

El escenario vuelve al punto 2.

**A2: Nombre de usuario inexistente**

La secuencia A2 comienza en el punto 4 del escenario principal.

5. S comunica que el nombre de usuario es inexistente.

El escenario vuelve al punto 2.

**A3: Nombre de usuario existente pero contraseña inválida**

La secuencia A3 comienza en el punto 4 del escenario principal.

4. S comunica que la contraseña es inválida.

El escenario vuelve al punto 2.



**Caso de Uso 12: Un administrador desea borrar o dar de baja un alumno.**

**Caso de Uso 13: Un administrador desea cambiar el rol a un usuario.**

CU13	Cambiar el rol a un usuario				
Revisa:		Fecha		Firma:	
<b>Resumen:</b> Este caso de uso permite al administrador cambiar el rol a un usuario en particular.					
<b>Actores:</b> Administrador (Primario). Servidor (Secundario)					
<b>Personal Involucrado y Metas:</b> <i>Administrador:</i> quiere cambiar el rol a un usuario. <i>Servidor:</i> quiere mantener actualizados los datos del usuario.					
<b>Precondiciones:</b> El administrador debe estar autenticado en SGR.					
<b>Poscondiciones:</b> Se registran los cambios de los datos de la cuenta de usuario en el servidor.					

**Escenario Principal:**

1. El administrador selecciona la opción para cambiar el rol a un usuario.
2. SGR solicita al administrador a través de una alerta la confirmación para dicha acción.
3. El administrador presiona el botón aceptar.
4. SGR envía la solicitud a S.
5. S envía confirmación.
6. SGR informa al administrador que la operación se realizó exitosamente.

**Flujos Alternativos:****A1: El sistema encuentra algún fallo para comunicarse con S**

La secuencia A1 comienza en el punto 4 del escenario principal.

5. SGR informa al administrador el problema de conexión a través de un mensaje por la pantalla.

El escenario vuelve al punto 1.

**Caso de Uso 14: Un administrador desea listar los alumnos.**

CU14	Listar Usuarios				
Revisa:		Fecha		Firma:	
<b>Resumen:</b> Este caso de uso permite a un administrador listar la información de todos usuarios registrados en el sistema.					
<b>Actores:</b> Administrador (Primario). Servidor (Secundario).					

**Personal Involucrado y Metas:**

*Administrador:* quiere visualizar de manera completa la información de los usuarios registrados.

*Servidor:* quiere que el administrador pueda ver la información relacionada con los perfiles de usuarios de manera segura.

**Precondiciones:** El administrador debe estar autenticado en SGR.

**Poscondiciones:** Se muestra un listado de usuarios registrados.

**Escenario Principal:**

1. El administrador selecciona la opción para listar los usuarios registrados.
2. SGR envía la solicitud a S.
3. S envía los datos necesarios para generar el listado de usuarios a SGR.
4. SGR muestra el listado de usuarios al administrador.

**Flujos Alternativos:****A1: El sistema encuentra algún fallo para comunicarse con S**

La secuencia A1 comienza en el punto 2 del escenario principal.

3. SGR informa al usuario el problema de conexión a través de un mensaje por la pantalla.

El escenario vuelve al punto 2.

**Caso de Uso 15: Un administrador desea listar los registros.**

CU15	Listar Registros	
------	------------------	--

Revisa:	Fecha	Firma:
<b>Resumen:</b> Este caso de uso permite a un administrador listar la información de todos los registros realizados por la totalidad de los usuarios.		
<b>Actores:</b> Administrador (Primario). Servidor (Secundario).		
<b>Personal Involucrado y Metas:</b> <i>Administrador:</i> quiere visualizar de manera completa la información de los registros realizados. <i>Servidor:</i> quiere que el administrador pueda ver la información relacionada con los registros de manera segura.		
<b>Precondiciones:</b> El administrador debe estar autenticado en SGR.		
<b>Poscondiciones:</b> Se muestra un listado de registros realizados.		
<b>Escenario Principal:</b>  1. El administrador selecciona la opción para listar los registros realizados.  2. SGR envía la solicitud a S.  3. S envía los datos necesarios para generar el listado de registros a SGR.  4. SGR muestra en pantalla el listado de registros al administrador.		

**Flujos Alternativos:****A1: El sistema encuentra algún fallo para comunicarse con S**

La secuencia A1 comienza en el punto 2 del escenario principal.

3. SGR informa al administrador el problema de conexión a través de un mensaje por la pantalla.

El escenario vuelve al punto 2.

### Caso de Uso 16: Un administrador desea ver un registro en particular.

CU16	Ver Registro				
Revisa:		Fecha		Firma:	
<b>Resumen:</b> Este caso de uso permite al administrador ver un registro en particular con toda su informacion.					
<b>Actores:</b> Administrador (Primario). Servidor (Secundario)					
<b>Personal Involucrado y Metas:</b> <i>Administrador:</i> quiere visualizar de manera completa la información de un registro. <i>Servidor:</i> quiere que el administrador pueda ver la información relacionada con un registro de forma segura.					
<b>Precondiciones:</b> El administrador debe estar autenticado en SGR.					
<b>Poscondiciones:</b> Se muestra la informacion completa del registro de un usuario.					

**Escenario Principal:**

1. El administrador selecciona la opción de listar los registros.
2. El administrador, sobre un registro de la lista, selecciona la opción de ver registro.
3. SGR envía la solicitud a S.
4. S envía los datos necesarios para generar la vista del registro a SGR.
5. SGR muestra el registro completo por pantalla.

**Flujos Alternativos:****A1: El sistema encuentra algún fallo para comunicarse con S**

La secuencia A1 comienza en el punto 3 del escenario principal.

4. SGR informa al usuario el problema de conexión a través de un mensaje por la pantalla.

El escenario vuelve al punto 2.

**Caso de Uso UC17: Un administrador desea validar o invalidar registros.**

CU17	Validar o Invalidar Registro				
Revisa:		Fecha		Firma:	
<b>Resumen:</b> Este caso de uso permite al administrador validar o invalidar un registro en particular.					
<b>Actores:</b> Administrador (Primario). Servidor (Secundario)					

**Personal Involucrado y Metas:**

*Administrador:* quiere validar o invalidar un registro creado por un usuario.

*Servidor:* quiere mantener actualizados los datos de los registros.

**Precondiciones:** El administrador debe estar autenticado en SGR.

**Poscondiciones:** Se registran los cambios de los datos del registro en cuestión en el servidor.

**Escenario Principal:**

1. El administrador ingresa a la opción para ver un registro en particular.
2. El administrador selecciona la opción de validar o invalidar registro.
3. SGR solicita al administrador a través de una alerta la confirmación para validar o invalidar el registro.
4. El administrador presiona el botón aceptar.
5. SGR envía la solicitud a S.
6. S envía confirmación.
7. SGR informa al administrador que la operación se realizó exitosamente.

**Flujos Alternativos:****A1: El sistema encuentra algún fallo para comunicarse con S**

La secuencia A1 comienza en el punto 5 del escenario principal.

6. SGR informa al administrador el problema de conexión a través de un mensaje por la pantalla.

El escenario vuelve al punto 2.

**Caso de Uso UC18/19/20: Un administrador desea buscar registros por fecha de creación/Institución/Índice.**

CU18/19/20	Buscar Registro				
Revisa:		Fecha		Firma:	
<b>Resumen:</b> Este caso de uso permite al administrador realizar una búsqueda de registros filtrados por un rango de fechas.					
<b>Actores:</b> Administrador (Primario). Servidor (Secundario)					
<b>Personal Involucrado y Metas:</b> <i>Administrador:</i> quiere obtener los registros que coincidan con un determinado criterio de búsqueda. <i>Servidor:</i> quiere ofrecer al administrador los registros que mejor se ajustan a su búsqueda					
<b>Precondiciones:</b> El administrador debe estar autenticado en SGR.					
<b>Poscondiciones:</b> Se obtiene un listado de registros que cumplen con la condición de búsqueda.					



**Escenario Principal:**

1. El administrador ingresa el texto que buscará (rango de fechas o institución o índice)
2. El administrador presiona un botón para comenzar la búsqueda.
3. SGR envía la solicitud a S.
4. S verifica los registros que cumplen con el criterio buscado.
5. S retorna el listado de registros
6. SGR muestra el listado al administrador.

**Flujos Alternativos:****A1: El sistema encuentra algún fallo para comunicarse con S**

La secuencia A1 comienza en el punto 3 del escenario principal.

4. SGR informa al administrador el problema de conexión a través de un mensaje por la pantalla.

El escenario vuelve al punto 2.

## **Caso de Uso UC22: Un administrador desea exportar a Excel la información.**

CU22	Exportar Excel				
Revisa:		Fecha		Firma:	
<b>Resumen:</b> Este caso de uso permite al administrador exportar un archivo formato excel con la informacion de los registros.					

**Actores:** Administrador (Primario). Servidor (Secundario)

**Personal Involucrado y Metas:**

*Administrador:* quiere exportar un archivo excel con la informacion de los registros.

*Servidor:* quiere ofrecer al administrador el archivo excel con los registros.

**Precondiciones:**

El administrador debe estar autenticado en SGR.

**Poscondiciones:** Se facilita un archivo excel para la descarga.

**Escenario Principal:**

1. El administrador selecciona la opción para ver exportar un archivo excel.
2. SGR envía la solicitud a S.
3. S genera el archivo excel con los registros.
4. S envía el archivo a SGR.
5. SGR proporciona una ventana para la descarga del archivo excel.

**Flujos Alternativos:**

**A1: El sistema encuentra algún fallo para comunicarse con S**

La secuencia A1 comienza en el punto 2 del escenario principal.

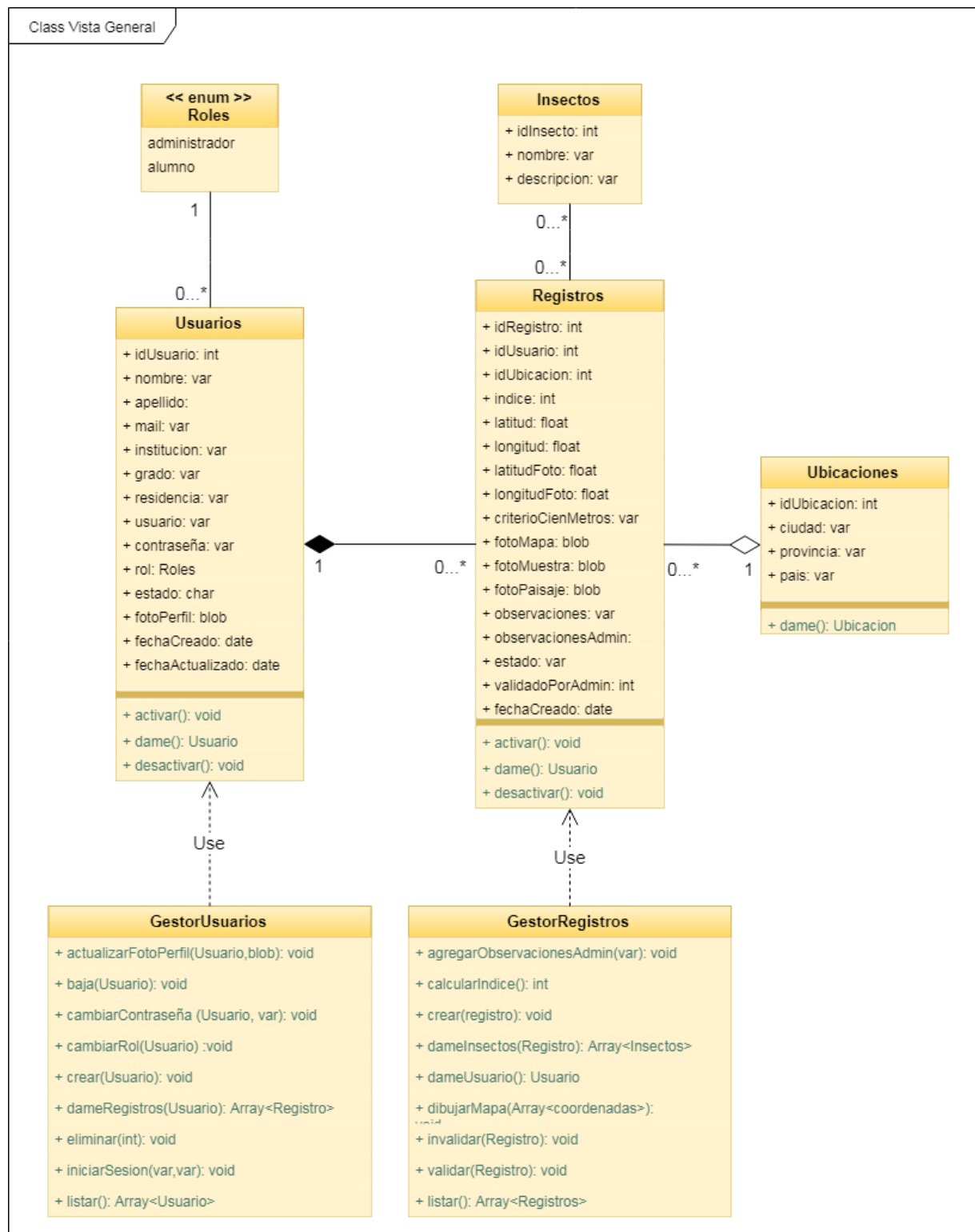
3. SGR informa al usuario el problema de conexión a través de un mensaje por la pantalla.

El escenario vuelve al punto 2.

## **Diagrama de Clases**

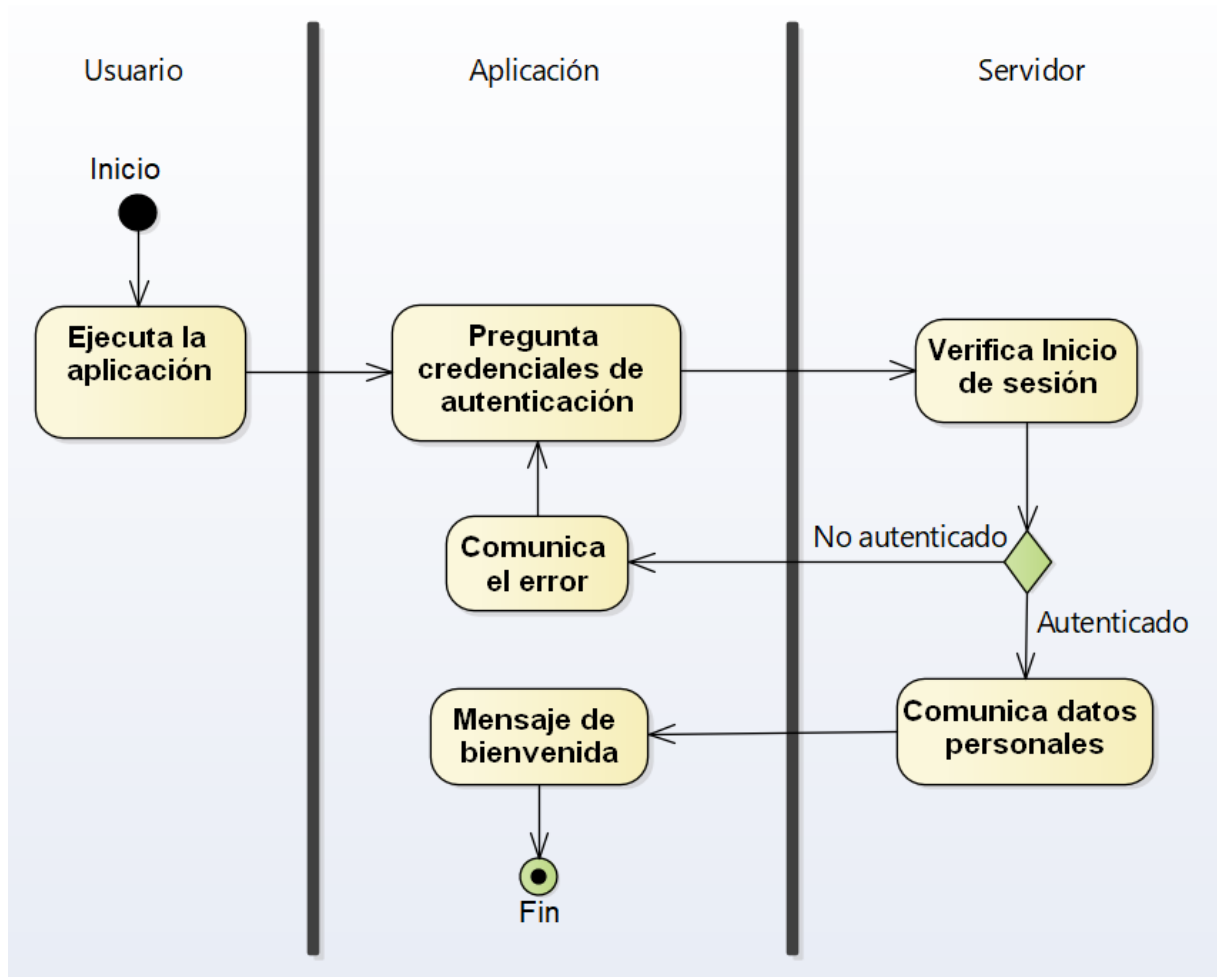
Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenimiento. Se puede utilizar un diagrama de clases para describir los tipos de datos y sus relaciones con independencia de su implementación. El diagrama se utiliza para que la atención se centre en los aspectos lógicos de las clases en lugar de en su implementación.

## Vista general

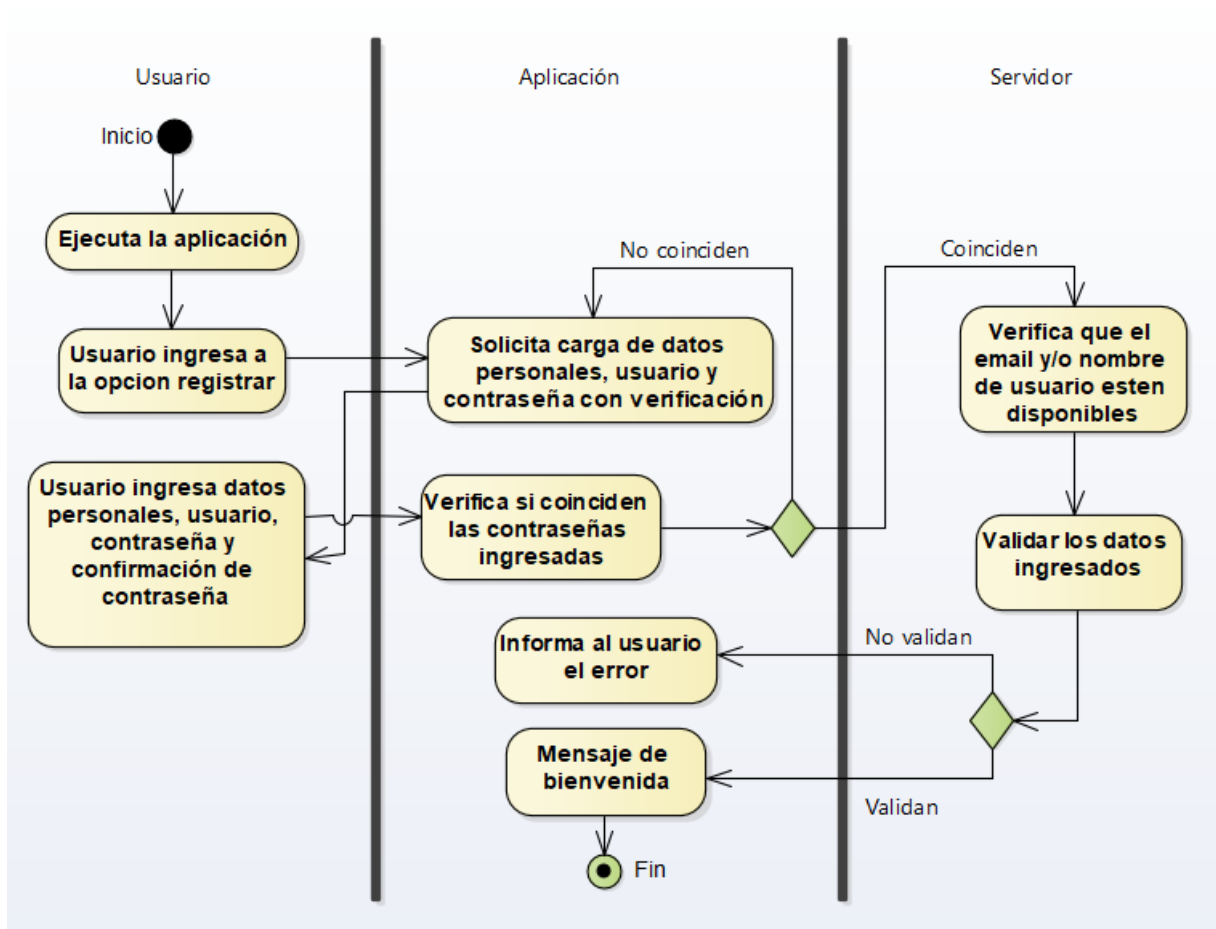


## Diagramas de Actividad

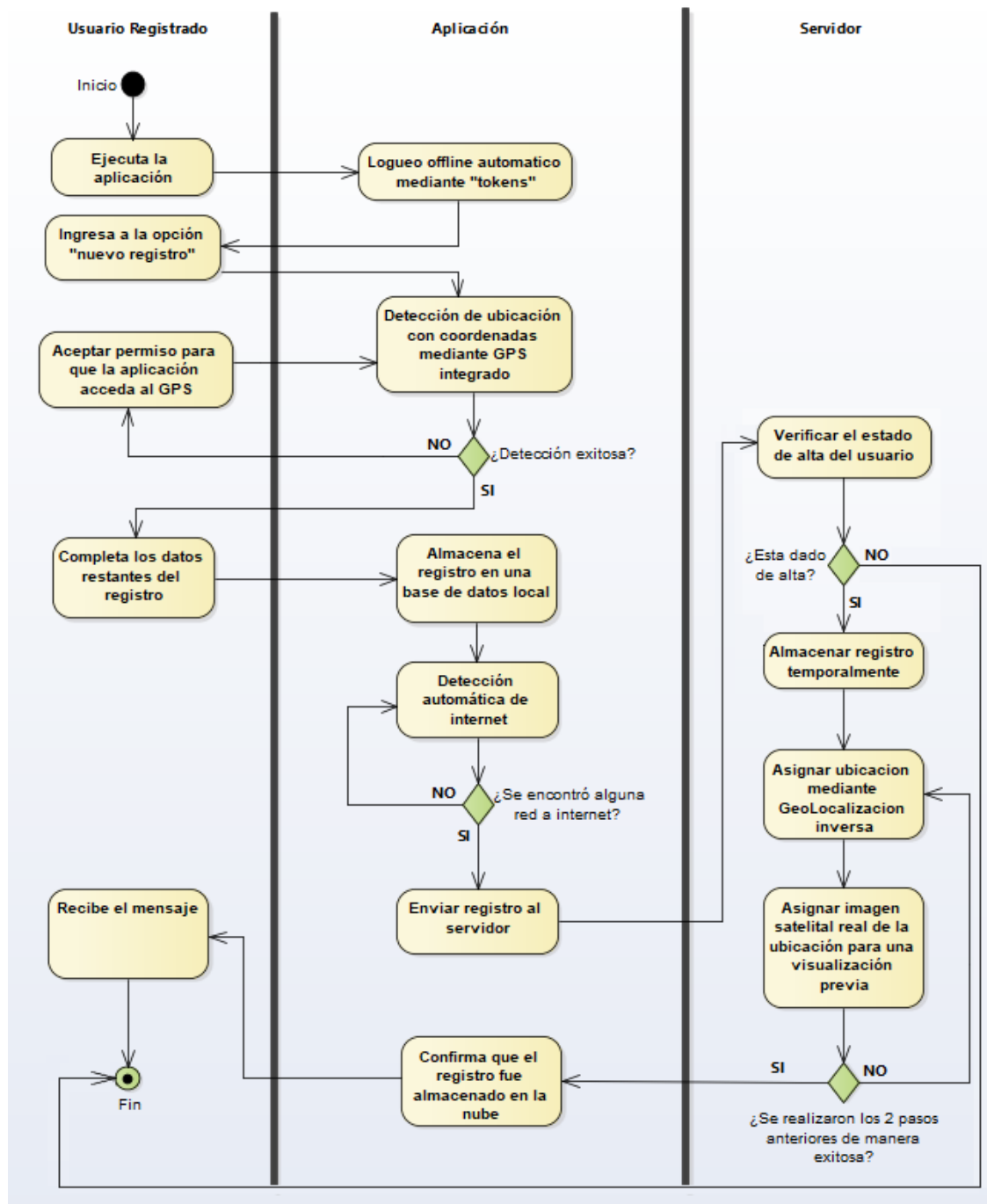
### Inicio de Sesión



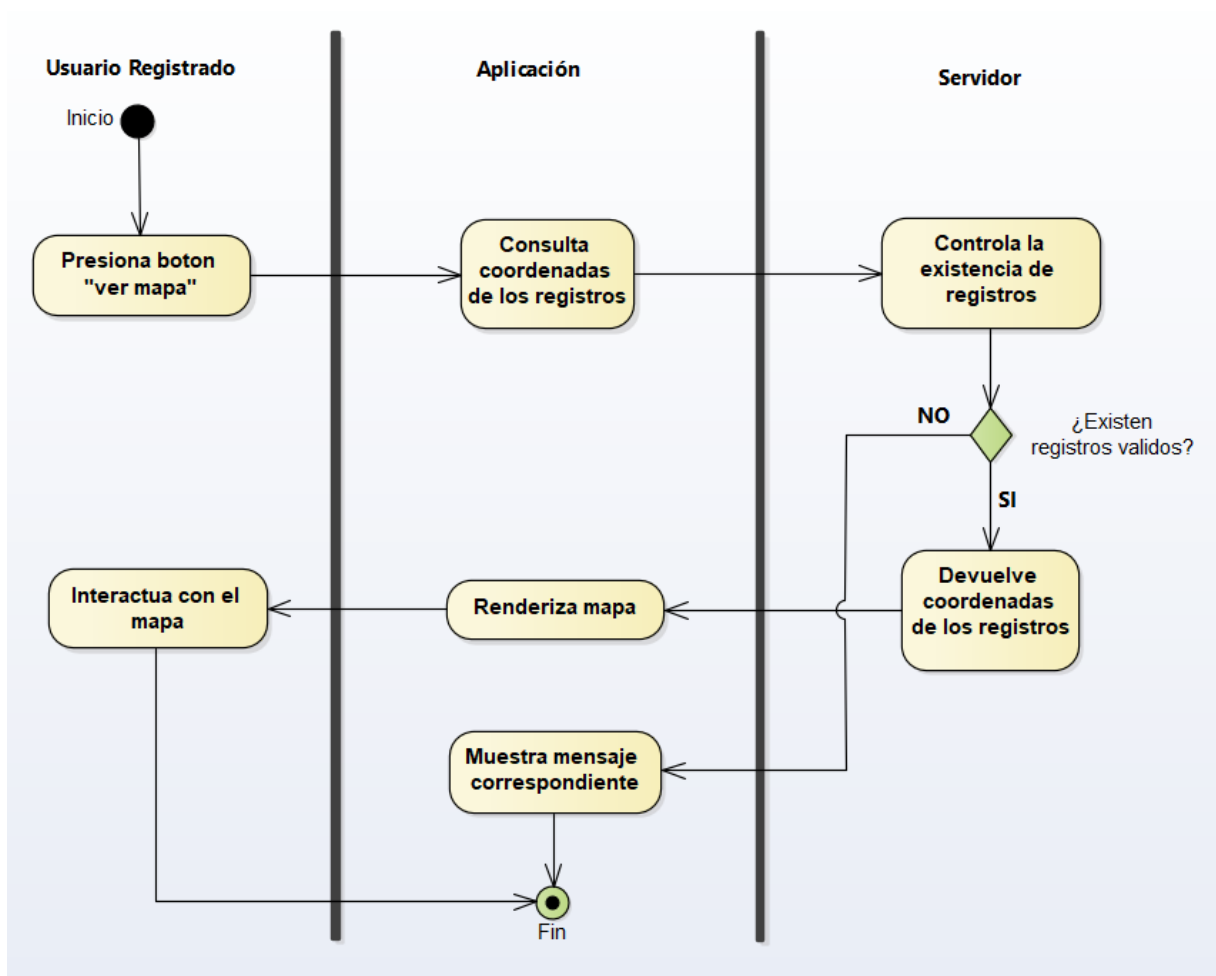
## Registrar Usuario



## Crear Registro



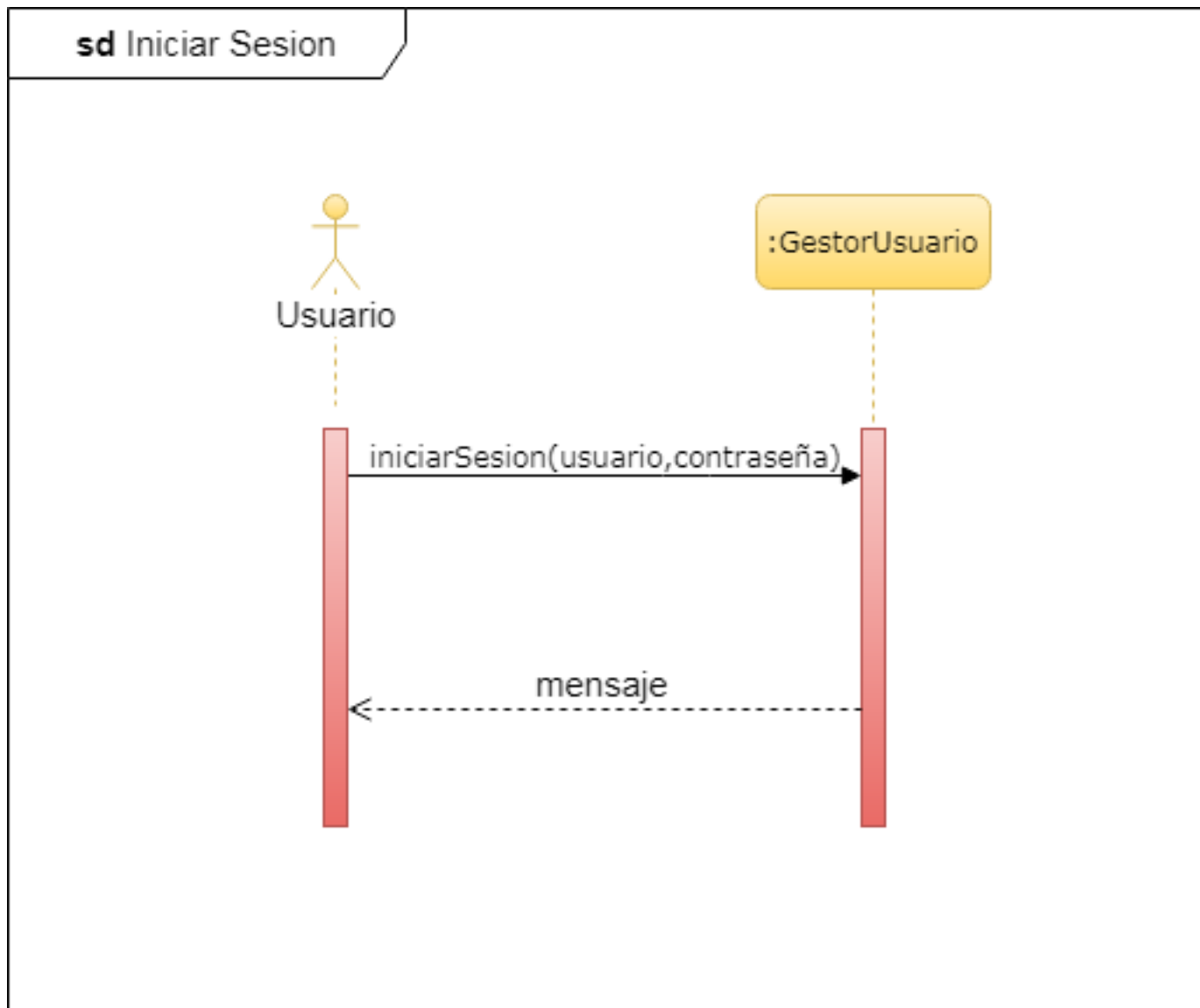
## Ver Mapa Interactivo



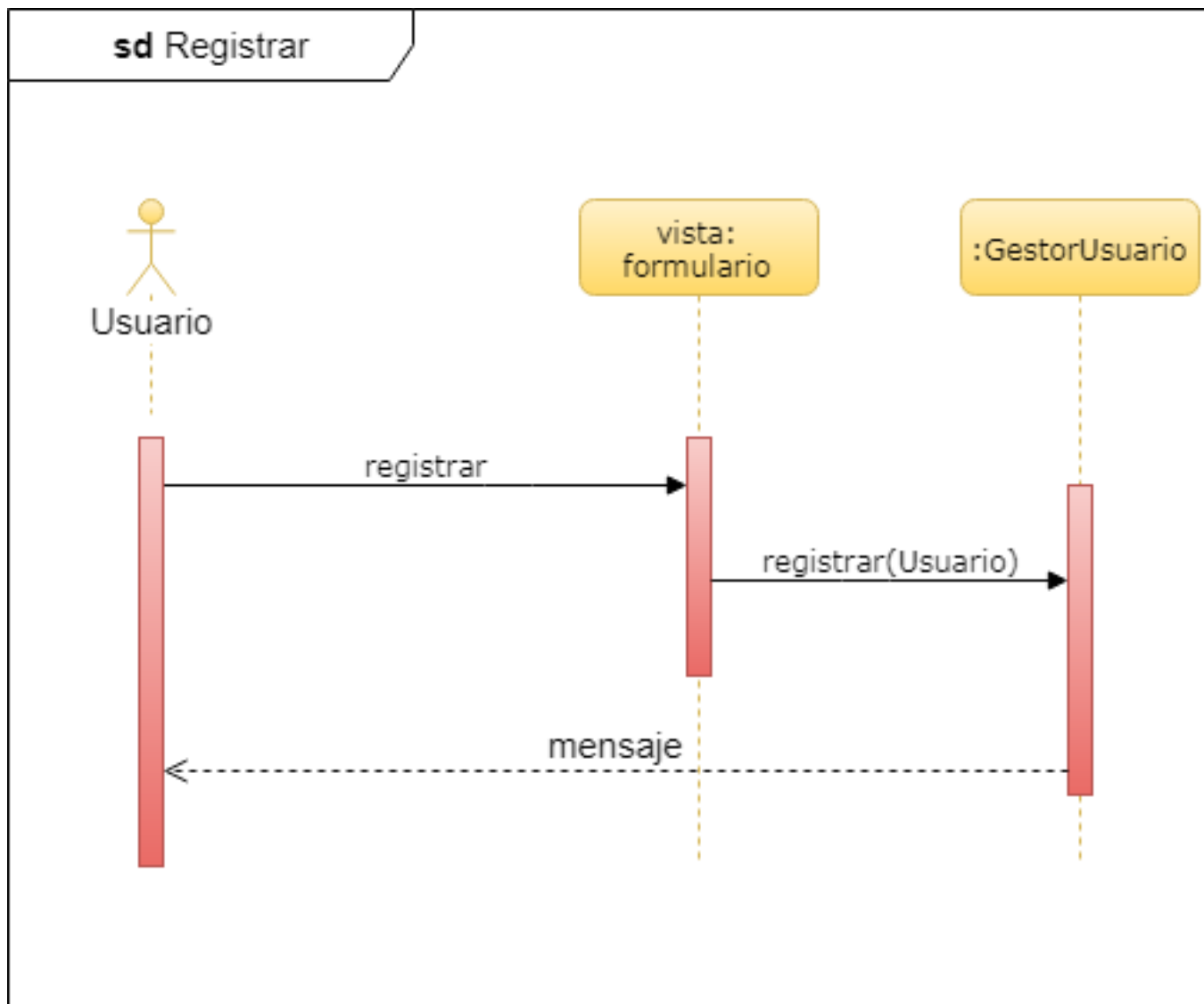


## Diagramas de Secuencia

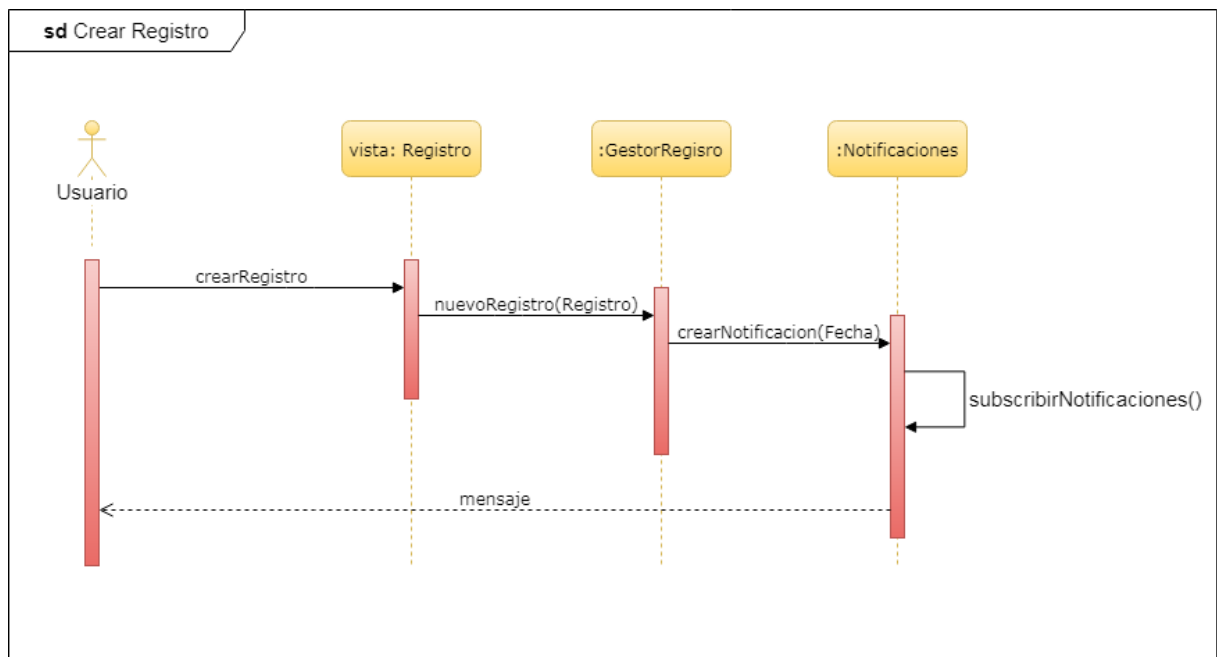
### Iniciar Sesion



## Registrar



## Crear Registro



## Interfaz de Usuario

### Iniciar Sesión



Figura 1: Pantalla inicial

3G 9:11

**DroidCommerce**

Usuario

Contraseña

**Ingresar**

También podés registrarte:

**+ Registrar**

**Home Tienda Opciones**

Figura 2: Pantalla de login

3G 9:11

**DroidCommerce**

\* Nombre de Usuario

\* Contraseña

\* Confirmar contraseña

\* Email

**Registrar**

Home Tienda Opciones

Figura 3: Pantalla de registro

**Iniciar Sesion**

**Iniciar Sesion**

**Iniciar Sesion**

**Iniciar Sesion**

**Iniciar Sesion**

**Iniciar Sesion**

**Iniciar Sesion**

**Iniciar Sesion**

**Iniciar Sesion**

**Iniciar Sesion**

**Iniciar Sesion**

**Iniciar Sesion**

3G 9:13

**DroidCommerce**

\* Nombre

Moneda:

US\$ AR\$ €

Descripción

Logo

+ Agregar ✕ Borrar L...

Home Tienda Opciones

Figura 4: Pantalla de creación de tienda, parte superior






Figura 5: Pantalla de creación de tienda, parte inferior



Figura 6: Pantalla de administración de la tienda



3G 9:19

**DroidCommerce**

\* Nombre

Categoría

**Libros, revistas y cómics**

Descripción

\* Precio

\* Precio de Envío

Home Tienda Opciones

Figura 7: Pantalla de creación de producto, parte superior



Figura 8: Pantalla de creación de producto, parte inferior



Figura 9: Pantalla de administración de productos



Figura 10: Pantalla de tienda



Figura 11: Pantalla de resultados de la búsqueda de productos



Figura 12: Pantalla de producto





Figura 13: Pantalla de listado de ventas

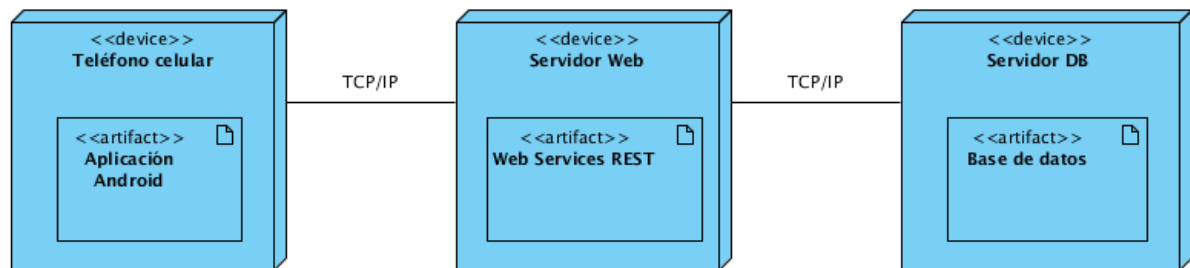
# Disciplina de Implementación

## Arquitectura de la aplicación



Figura 14: Arquitectura de la aplicación, mostrando los frameworks utilizados

## Diagrama de Despliegue



## Elección del Lenguaje

Independientemente del paradigma de ingeniería de software, el lenguaje de programación tendrá impacto en la planificación, el análisis, el diseño, la codificación, la prueba y el mantenimiento de un proyecto. Para la construcción de la aplicación se eligió la utilización de los lenguajes web HTML5, CSS3 y JavaScript.

La elección de estos lenguajes para la construcción de la aplicación se debe a las siguientes ventajas que ofrecen:

- *Mayor portabilidad:* Al ser tecnologías estándares y soportadas por la mayoría de los teléfonos celulares modernos, es posible que una misma aplicación sea muy fácilmente adaptable a varias plataformas móviles.
- *Soporte futuro:* Todas las plataformas móviles están trabajando para mejorar el soporte que ofrecen a las tecnologías web, ofreciendo una mejor experiencia al usuario.
- *Aprovechamiento de conocimiento de desarrollo de aplicaciones web:* Desarrollando aplicaciones móviles en **HTML5**, **CSS3** y **Javascript** es posible aplicar el conocimiento en el desarrollo de aplicaciones web desarrolladas para navegadores en equipos de escritorio.

## Arquitectura de Android

Los componentes principales del sistema operativo Android, que pueden verse en la figura 15, son:

- *Aplicaciones:* PRUEBA GLOSARIO PARA SO Sistema Operativo las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. La mayoría de las aplicaciones están escritas en lenguaje de programación Java, aunque existen TCP/IP (Internet Protocol Suite)s para desarrollar aplicaciones en otros lenguajes como HTML, CSS y Javascript o Python.
- *Marco de trabajo de aplicaciones:* los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.
- *Bibliotecas:* Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.
- *Runtime de Android:* Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecuta archivos en el formato Dalvik Executable (.dex), el cual está optimizado para minimizar el consumo de memoria. La Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato.dex por la herramienta incluida "dx".

- *Núcleo Linux*: Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.



Figura 15: Arquitectura de Android

## PhoneGap

PhoneGap es un **TCP/IP (Internet Protocol Suite)** de código abierto que actúa como un intermediario entre las aplicaciones web y los dispositivos móviles. Permite crear aplicaciones móviles instalables utilizando tecnología web: **Javascript**, **HTML5** y **CSS3**.

Las aplicaciones resultantes no son totalmente nativas, ni puramente basadas en la web. La desventaja de que una aplicación sea totalmente nativa es que sólo se podrá utilizar para la plataforma para la que fue realizada, es decir si se hace una aplicación

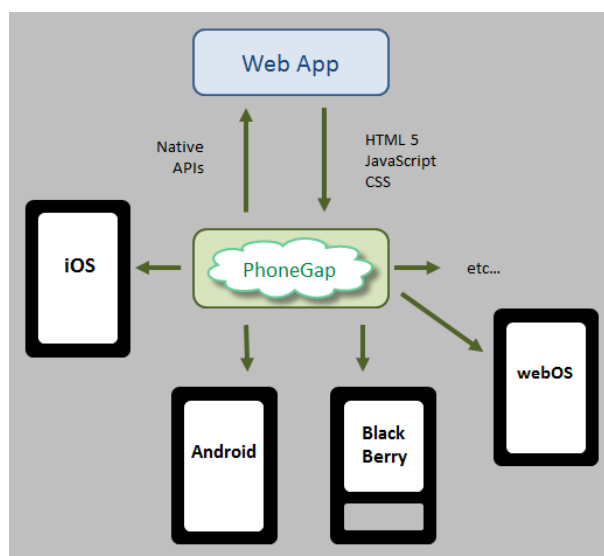


Figura 16: Arquitectura de PhoneGap

para Android luego no se podrá reutilizar el código para hacer la misma aplicación para iOS.

Con PhoneGap se puede reutilizar el código de una aplicación para crear el paquete instalable de cualquiera de las 7 plataformas móviles soportadas: iOS, Android, Blackberry, Windows Phone, WebOS de Palm, Symbian y Bada.

PhoneGap permite acceder a funciones nativas como el acelerómetro, cámara, brújula, contactos, archivos, ubicación geográfica, almacenamiento y notificaciones.

PhoneGap para Android esta dividido en dos partes:

- *Librerías nativas (phonegap.jar)*: Agrega acceso JavaScript para APIs nativas.
- *Archivos javascript (phonegap.js)*: Contenedores JavaScript para llamados de APIs nativas.

## Ventajas de PhoneGap

- Soporta 7 plataformas móviles: iOS, Android, Blackberry, Windows Phone, WebOS de Palm, Symbian y Bada.
- Acceso a características nativas de cada plataforma a través de su API, a las que una aplicación web visitada desde el navegador no podría acceder, como acceso a la

cámara de fotos, acelerómetro, notificaciones, etc.

- Permite ejecutar a través de JavaScript plugins escritos en código nativo.
- Permite distribuir aplicaciones realizadas utilizando HTML5 y JavaScript a través de las tiendas de aplicaciones oficiales de cada plataforma.

## Desventaja de phoneGap

- Normalmente las aplicaciones realizadas con PhoneGap tienen un menor rendimiento en tareas que requieren alta capacidad de procesamiento, sobre todo en versiones antiguas de las plataformas sobre las que se usa.
- Se pierde la posibilidad de acceder a algunas características nativas, como los diferentes elementos de interfaz de usuario propios de cada plataforma, aunque estos pueden imitarse mediante el uso de CSS.

## Web services

Un servicio web (en inglés, Web service) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos.

Las ventajas de los servicios web son:

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.

- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.

## Razones para crear servicios Web

La principal razón para usar servicios Web es que se pueden utilizar con HTTP sobre **TCP/IP (Internet Protocol Suite)** en el puerto 80. Dado que las organizaciones protegen sus redes mediante firewalls -que filtran y bloquean gran parte del tráfico de Internet, cierran casi todos los puertos TCP salvo el 80, que es, precisamente, el que usan los navegadores. Los servicios Web utilizan este puerto, por la simple razón de que no resultan bloqueados. Es importante señalar que los servicios web se pueden utilizar sobre cualquier protocolo, sin embargo, TCP es el más común.

Otra razón por la que los servicios Web son muy prácticos es que pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro. Esta flexibilidad será cada vez más importante, dado que la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños es cada día más utilizada. Se pueden desarrollar servicios web como parte de una aplicación web, permitiendo acceder a los mismos datos que esta.

## REST

REST es una técnica de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web. REST describe cualquier interfaz web simple que utiliza **TCP/IP (Internet Protocol Suite)** (o **TCP/IP (Internet Protocol Suite)**) y HTTP, sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes como el protocolo de servicios web **TCP/IP (Internet Protocol Suite)**.

Los sistemas que siguen los principios REST se llaman con frecuencia RESTful.

REST afirma que la web ha disfrutado de escalabilidad como resultado de una serie de diseños fundamentales clave:



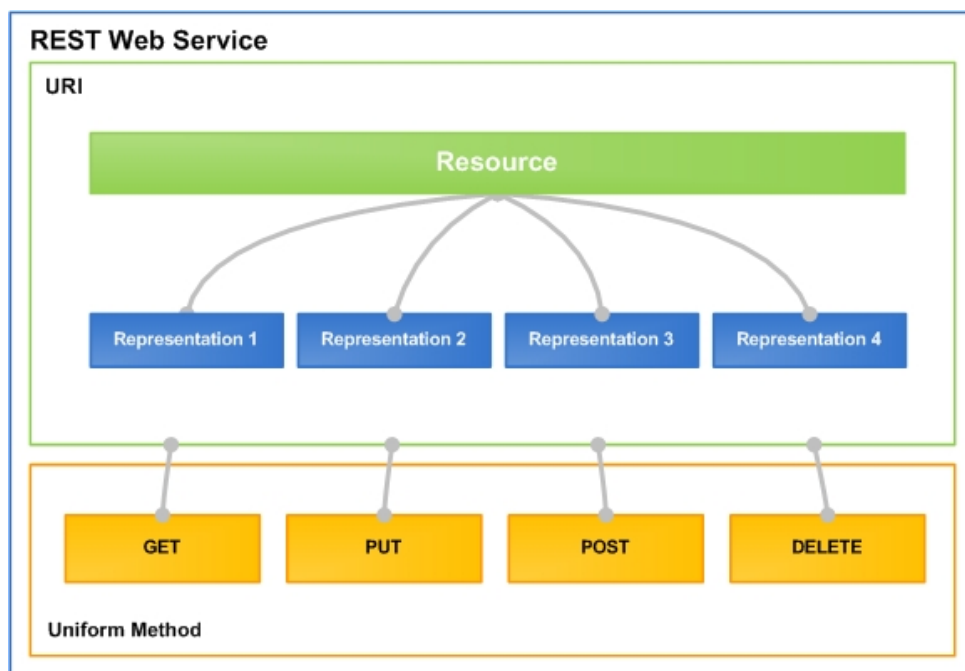


Figura 17: Web Services REST

- *Un protocolo cliente/servidor sin estado:* cada mensaje HTTP contiene toda la información necesaria para comprender la petición. Como resultado, ni el cliente ni el servidor necesitan recordar ningún estado de las comunicaciones entre mensajes. Sin embargo, en la práctica, muchas aplicaciones basadas en HTTP utilizan cookies y otros mecanismos para mantener el estado de la sesión (algunas de estas prácticas, como la reescritura de URLs, no son permitidas por REST)
- *Un conjunto de operaciones bien definidas que se aplican a todos los recursos de información:* HTTP en sí define un conjunto pequeño de operaciones, las más importantes son POST, GET, PUT y DELETE.
- Una sintaxis universal para identificar los recursos. En un sistema REST, cada recurso es direccionable únicamente a través de su **TCP/IP (Internet Protocol Suite)**.
- El uso de hipermedios, tanto para la información de la aplicación como para las transiciones de estado de la aplicación: la representación de este estado en un sistema REST son típicamente **HTML**, XML o JSON. Como resultado de esto, es posible navegar de un recurso REST a muchos otros, simplemente siguiendo enlaces sin

requerir el uso de registros u otra infraestructura adicional.

## Recursos

Un concepto importante en REST es la existencia de recursos (elementos de información), que pueden ser accedidos utilizando un identificador global (un Identificador Uniforme de Recurso).

Para manipular estos recursos, los componentes de la red (clientes y servidores) se comunican a través de una interfaz estándar (HTTP) e intercambian representaciones de estos recursos (los ficheros que se descargan y se envían).

La petición puede ser transmitida por cualquier número de conectores (por ejemplo clientes, servidores, cachés, túneles, etc.) pero cada uno lo hace sin "ver más allá" de su propia petición. Así, una aplicación puede interactuar con un recurso conociendo el identificador del recurso y la acción requerida, no necesitando conocer si existen cachés, proxys, cortafuegos, túneles o cualquier otra cosa entre ella y el servidor que guarda la información. La aplicación, sin embargo, debe comprender el formato de la información devuelta (la representación), que es por lo general un documento **HTML**, **TCP/IP (Internet Protocol Suite)** o **TCP/IP (Internet Protocol Suite)**, aunque también puede ser una imagen o cualquier otro contenido.

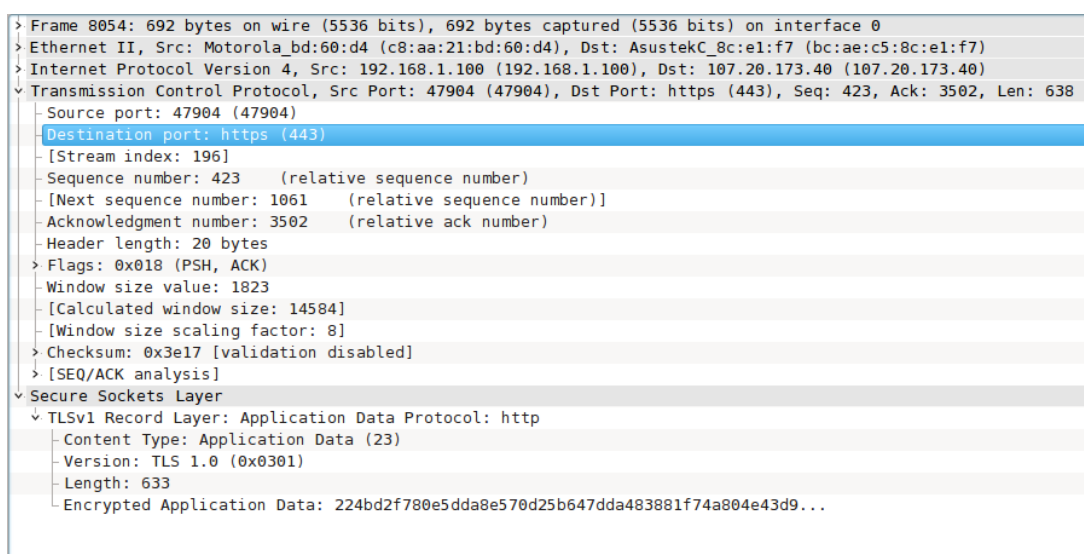
## Características de seguridad

### Canal de comunicación cifrado

En todos los casos en los que se realiza transferencia de datos confidenciales del usuario, como sus credenciales de acceso, datos personales u operaciones realizadas, es necesario asegurar que tanto las solicitudes y las respuestas se envían a través de un canal de comunicación cifrado.

La utilización de protocolos de comunicación inseguros, como HTTP, pueden hacer que la comunicación pueda ser interceptada utilizando ataques **TCP/IP (Internet Protocol Suite)**, permitiendo que el atacante pueda ver todos los datos intercambiados e incluso manipularlos o generar solicitudes falsas.

Para mitigar este riesgo, las comunicaciones de la aplicación se realizan utilizando el protocolo de comunicación segura **HTTPS**. Este está basado en HTTP, pero utiliza un cifrado basado en **TCP/IP (Internet Protocol Suite)** para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto) más apropiado para el tráfico de información sensible que el protocolo HTTP. De este modo se consigue que la información sensible no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que lo único que obtendrá será un flujo de datos cifrados que le resultará imposible de descifrar, como se muestra en la figura 18.



```

> Frame 8054: 692 bytes on wire (5536 bits), 692 bytes captured (5536 bits) on interface 0
> Ethernet II, Src: Motorola_bd:60:d4 (c8:aa:21:bd:60:d4), Dst: AsustekC_8c:e1:f7 (bc:ae:c5:8c:e1:f7)
> Internet Protocol Version 4, Src: 192.168.1.100 (192.168.1.100), Dst: 107.20.173.40 (107.20.173.40)
> Transmission Control Protocol, Src Port: 47904 (47904), Dst Port: https (443), Seq: 423, Ack: 3502, Len: 638
  - Source port: 47904 (47904)
  - Destination port: https (443)
  - [Stream index: 196]
  - Sequence number: 423 (relative sequence number)
  - [Next sequence number: 1061 (relative sequence number)]
  - Acknowledgment number: 3502 (relative ack number)
  - Header length: 20 bytes
  - Flags: 0x018 (PSH, ACK)
  - Window size value: 1823
  - [Calculated window size: 14584]
  - [Window size scaling factor: 8]
  - Checksum: 0x3e17 [validation disabled]
  - [SEQ/ACK analysis]
> Secure Sockets Layer
  - TLSv1 Record Layer: Application Data Protocol: http
    - Content Type: Application Data (23)
    - Version: TLS 1.0 (0x0301)
    - Length: 633
    - Encrypted Application Data: 224bd2f780e5dda8e570d25b647dda483881f74a804e43d9...

```

Figura 18: Captura de la solicitud cifrada como lo vería un atacante utilizando Wireshark

## ACL para cada objeto

Como ya se habló anteriormente, los Web services pueden ser públicamente accedidos a través de Internet, por lo que es imprescindible contar con un mecanismo que impida la realización de consultas o modificaciones no autorizadas. En la aplicación es necesario que los datos estén definidos correctamente: sólo un vendedor puede realizar la modificación de los datos de una tienda y de sus productos, sólo puede modificar los datos personales el usuario al que pertenece la cuenta activa y las órdenes no pueden modificarse una vez creadas y sólo pueden ser vistas por el usuario comprador o el vendedor.

Para asegurar que los datos sólo son accesibles por usuarios que están autorizados a

leerlos o modificarlos se implementó un **TCP/IP (Internet Protocol Suite)** para cada objeto. Este dato se almacena cuando el objeto es creado, y está representado con un listado en formato **TCP/IP (Internet Protocol Suite)** incluyendo los permisos de los usuarios sobre ese objeto. En el siguiente ejemplo se muestra el caso de un objeto que es visible públicamente pero sólo el usuario con id `idUsuario` puede realizar modificaciones:

```
{
  "idUsuario":
    {
      "read":true,
      "write":true
    },
  "*":
    {
      "read":true
    }
}
```

Esta medida de seguridad debe ser controlada en el servidor web que recibe las peticiones, debido a que controlarlo en el cliente no soluciona el problema de las modificaciones no autorizadas. Una solicitud a un objeto sin usar las credenciales de un usuario autorizado debe ser denegada por el servidor, como se muestra en la figura 19. Es importante también que el mensaje retornado por el servidor, en caso de que el objeto no tenga permisos de visualización por el usuario actual, no revele evidencia de la existencia del objeto.

## Backbone.js

Backbone.js es un **TCP/IP (Internet Protocol Suite)** para Javascript con un interfaz RESTful por **TCP/IP (Internet Protocol Suite)**, basada en el paradigma de diseño de aplicaciones Modelo Vista Presentador (MVP). Está diseñado para desarrollar aplicaciones de una única página y para mantener las diferentes partes de las aplicaciones web

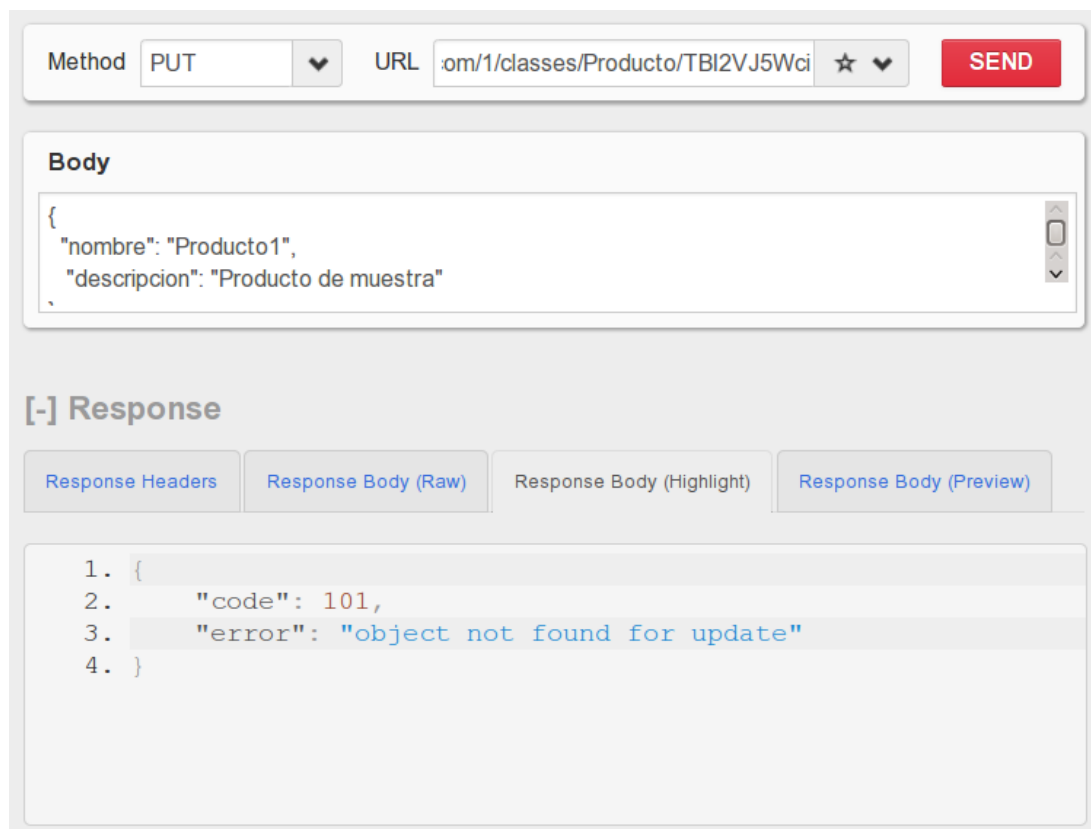


Figura 19: Solicitud de modificación rechazada a un objeto existente utilizando REST Client

(p.e. múltiples clientes y un servidor) sincronizadas.

Backbone.js posee cuatro clases principales:

- Model
- View
- Router
- Collection

## Modelo Vista Presentador (MVP)

El patrón Modelo-Vista-Presentador (MVP) surge como una variación del patrón Modelo-Vista-Controlador (MVC).

Los componentes básicos de este patrón son:

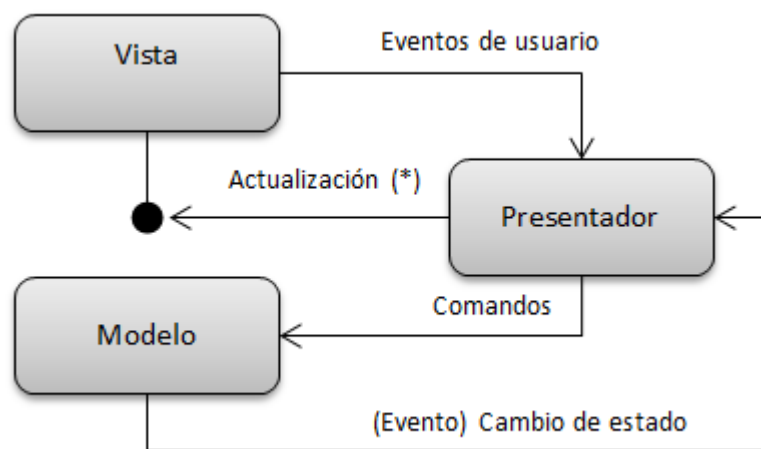


Figura 20: Componentes MVP

- *Modelo*: El modelo es normalmente los datos de la aplicación y la lógica para recuperar y conservar los datos. A menudo, se trata de un modelo de dominio que puede basarse en una base de datos o los resultados de los servicios web. En algunos casos, que el modelo de dominio corresponde perfectamente a lo que se ve en la pantalla, pero en otros casos ha de ser adaptada, agregados o extendido para ser utilizable.

- *Vista*: La vista es típicamente un control de usuario o formulario que combina varios en una interfaz de usuario. El usuario puede interactuar con los controles en la vista
- *Presentador*: El presentador tiene toda la lógica de la vista y es responsable de sincronizar el modelo y la vista. Cuando la vista notifica el presentador que el usuario ha hecho algo (por ejemplo, hacer clic en un botón), el presentador a continuación, actualizar el modelo y sincronizar los cambios entre el modelo y la vista.

## jQuery

jQuery es una biblioteca de **Javascript** que permite simplificar la manera de interactuar con los documentos **HTML**, manipular el árbol **TCP/IP (Internet Protocol Suite)**, manejar eventos, desarrollar animaciones y agregar interacción con la técnica **TCP/IP (Internet Protocol Suite)** a páginas web.

Es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y cerrados.

Al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y líneas de código.

## jQuery Mobile

jQuery Mobile es un **TCP/IP (Internet Protocol Suite)** para Javascript utilizado en el desarrollo de la interfaz de usuario para aplicaciones web adaptadas a dispositivos móviles. Incluye elementos de UI y soporte a eventos relacionados con el uso de pantallas táctiles.

Las principales características de jQuery Mobile son:

- Es compatible con otros frameworks que utilizamos para el desarrollo de la aplicación móvil, tales como Phonegap o Backbone.js.
- Es compatible con las principales plataformas móviles, así como todos los navegadores de escritorio principales, incluyendo Android.

- Construido sobre jQuery.
- Permite la creación de temas personalizados mediante el agregado de estilos CSS. Proporciona además una base de temas que permite a los desarrolladores personalizar las combinaciones de colores y determinados aspectos de las características de interfaz de usuario.
- Tiene mínimas dependencias, aumentando la velocidad y reduciendo el consumo de memoria.
- Adaptación automática del diseño al tamaño de la pantalla.

## Herramientas de desarrollo

- *Eclipse*: Es un entorno de desarrollo integrado (IDE) de código abierto multiplataforma.
- *Visual Paradigm para UML*: Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

- *L<sup>A</sup>T<sub>E</sub>X*: Es un sistema de composición de textos, orientado especialmente a la creación de libros, documentos científicos y técnicos que contengan fórmulas matemáticas.  
*L<sup>A</sup>T<sub>E</sub>X* facilita el uso del lenguaje de composición tipográfica. Es muy utilizado para la composición de artículos académicos, tesis y libros técnicos, dado que la calidad tipográfica de los documentos realizados con *L<sup>A</sup>T<sub>E</sub>X* es comparable a la de una editorial científica de primera línea.
- *ShareLaTeX*: Es un editor online (en tiempo real) de *L<sup>A</sup>T<sub>E</sub>X*, que puede ser utilizado por varios usuarios a la vez.



- *Dropbox y Google Drive:* Son servicios de alojamiento de archivos multiplataforma en la nube, operado por las compañías Dropbox y Google.

Permiten a los usuarios almacenar y sincronizar archivos en línea y entre computadoras y compartir archivos y carpetas con otros.

# Disciplina de Pruebas

## Test de Unidades

### Introducción

El Test de Unidades consiste en realizar pruebas de las unidades individuales de código. En esta fase se realizan las pruebas de caja blanca.

### Pruebas de Caja Blanca

Es un tipo de método de prueba que permite detectar errores internos del código de cada módulo.

Con estas pruebas se pueden garantizar que se ejercitan por lo menos una vez todos los caminos independientes de cada módulo, que las decisiones lógicas se evalúan en sus dos variantes (verdadera y falsa), que se ejecutan todos los bucles en sus límites operacionales y que se ejercitan las estructuras internas de datos para asegurar su validez.

## Test de Módulos

### Introducción

El Test de Módulos consiste en realizar pruebas de los módulos funcionales del sistema. En esta fase se realizan las pruebas de caja negra y las pruebas de estrés.

## Pruebas de Caja Negra

En este método de prueba se ve a cada módulo como una caja negra y se generan conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales del programa, observando las salidas.

Con estas pruebas se pueden detectar funciones incorrectas o ausentes, errores de interfaz, errores de rendimiento, etc.

## Pruebas de Estrés

Esta prueba se centra en realizar el análisis de valores límites, y en condiciones límites, ya que se ha demostrado que los errores tienden a darse más en los límites del campo de entrada y sometidos a condiciones límites.

## Test de Integración

### Introducción

El Test de Integración consiste en realizar pruebas de la estructura modular del programa y su interacción a través de la prueba de integración.

### Pruebas de Integración

En este tipo de prueba los errores surgen al integrar los módulos. En esta fase se pueden detectar errores como por ejemplo que las subfuncion, es cuando se combinan pueden no producir la función principal, un módulo puede tener un efecto adverso e inadvertido sobre otro, etc.

El objetivo es tomar los módulos probados y construir una estructura de programa que esté de acuerdo con lo que dicta la especificación C.

Existen dos tipos de integración:

Integración descendente: En este tipo se integran los módulos moviéndose hacia abajo por la jerarquía de control, comenzando con el módulo de control inicial.

Integración ascendente: En este tipo se integran los módulos atómicos primero y luego se continúa con el nivel inmediato superior.

En el desarrollo de este sistema se utilizó la integración descendente.

## **Test de Aceptación**

### **Introducción**

El Test de Aceptación consiste en realizar la prueba del software para validar si funciona de acuerdo con las expectativas razonables del cliente. En esta fase se llevan a cabo las pruebas Alfa y Beta.

### **Prueba Alfa**

Esta prueba es conducida por el cliente en el lugar de desarrollo. Se usa el software de forma natural (previa capacitación), con el encargado de desarrollo mirando “por encima del hombro” del usuario y registrando errores y problemas de uso. Se lleva a cabo en un entorno controlado.

### **Prueba Beta**

Esta prueba se lleva a cabo en uno o más lugares de clientes, por los usuarios finales de software. El encargado de desarrollo no está presente. El cliente registra todos los problemas (reales o imaginarios) que encuentra durante la prueba o informa a intervalos regulares al equipo de desarrollo. Se lleva a cabo en un entorno no controlado.

Los procedimientos de prueba se diseñaron para asegurar que se satisfacen todos los requisitos funcionales y que se alcanzan todos los requisitos de rendimiento.

# Conclusiones

El desarrollo de nuestro proyecto final nos permitió poner en práctica temas que aprendimos en distintas asignaturas durante el transcurso de nuestra carrera, el aprendizaje y la experiencia de implementar nuevas tecnologías, enfrentándonos a problemas reales de diseño e integración que nos forzaron a tomar decisiones a fin de encontrar soluciones eficientes a los mismos. Además, nos dio la posibilidad de aprender a trabajar con herramientas con los cuales no estábamos familiarizados.

Con la puesta en funcionamiento de este proyecto damos valor agregado a la aplicación web existente cumpliendo con los objetivos planteados inicialmente, obteniendo un producto seguro, escalable, de fácil mantenimiento y simple de usar.

Se logró implementar a través de Web services una comunicación eficiente con la base de datos, con mecanismos de seguridad que impiden la visualización o alteración indebida de los datos sensibles de los usuarios.

Vimos la importancia del uso de un desarrollo modular para la realización de proyectos flexibles, escalables, más legibles y manejables en aplicaciones actuales.

También podemos decir que este sistema queda abierto a la incorporación de futuras actualizaciones para satisfacer las necesidades de los usuarios cuando los procesos así lo requieran.

Con respecto a la estimación del tiempo necesario para el desarrollo realizada al inicio del proyecto, cuyo resultado fue de alrededor de 1500 horas hombre, al finalizar el desarrollo del mismo estimamos haber dedicado alrededor de 1300 horas hombre quedando dentro del margen de error del método utilizado.

# Glosario

## CSS

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML.. [82](#), [83](#), [84](#)

## HTML

HTML, siglas de HyperText Markup Language («lenguaje de marcado de hipertexto»), hace referencia al lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (`<i>`,`<i>`). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo, JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.. [82](#), [83](#), [84](#), [88](#), [89](#), [94](#)

## HTTPS

Hypertext Transfer Protocol Secure (o Protocolo seguro de transferencia de hipertexto), más conocido por sus siglas HTTPS, es un protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de Hiper Texto, es decir, es la versión segura de HTTP.. [89](#)

## Java

Java es un lenguaje de programación publicado en el 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. El lenguaje deriva mucho de su sintaxis de C y C++, pero tiene menos facilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede correr en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora. Java es un lenguaje de programación de propósito general, concurrente, basado en clases, y orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir del 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados.. [83](#)

## Javascript

JavaScript es un lenguaje de programación interpretado, orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web aunque existe una forma de JavaScript del lado del servidor. Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.. [82](#), [83](#), [84](#), [94](#)

## Linux

Linux es un núcleo libre de sistema operativo basado en Unix. Es uno de los principales ejemplos de software libre. Linux está licenciado bajo la GPL v2 y está desarrollado por colaboradores de todo el mundo.. [83](#)

## Python

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.. [83](#)

## Sistema Operativo

Un sistemaasdasdad operativo (SO, frecuentemente OS, del inglés Operating System) es un programa o conjunto de programas que en un sistema informático gestiona los recursos de hardware y provee servicios a los programas de aplicación, ejecutándose en modo privilegiado respecto de los restantes.. [83](#)

## TCP/IP (Internet Protocol Suite)

Es un conjunto de protocolos de comunicación que se utiliza en Internet y en otras redes similares. Sus componentes más importantes son TCP e IP, y se encuentran en representados en un modelo de capas, que van desde la capa de enlace hasta la capa de aplicación, pasando por la capa de Internet y la de transporte.. [94](#)

## TCP/IP (Internet Protocol Suite)

Es un conjunto de protocolos de comunicación que se utiliza en Internet y en otras redes similares. Sus componentes más importantes son TCP e IP, y se encuentran en representados en un modelo de capas, que van desde la capa de enlace hasta la capa de aplicación, pasando por la capa de Internet y la de transporte.. [94](#)

## TCP/IP (Internet Protocol Suite)

Es un conjunto de protocolos de comunicación que se utiliza en Internet y en otras redes similares. Sus componentes más importantes son TCP e IP, y se encuentran en representados en un modelo de capas, que van desde la capa de enlace hasta la capa de aplicación, pasando por la capa de Internet y la de transporte.. [90](#)



**TCP/IP (Internet Protocol Suite)**

Es un conjunto de protocolos de comunicación que se utiliza en Internet y en otras redes similares. Sus componentes más importantes son TCP e IP, y se encuentran en representados en un modelo de capas, que van desde la capa de enlace hasta la capa de aplicación, pasando por la capa de Internet y la de transporte.. 89

**TCP/IP (Internet Protocol Suite)**

Es un conjunto de protocolos de comunicación que se utiliza en Internet y en otras redes similares. Sus componentes más importantes son TCP e IP, y se encuentran en representados en un modelo de capas, que van desde la capa de enlace hasta la capa de aplicación, pasando por la capa de Internet y la de transporte.. 89

**TCP/IP (Internet Protocol Suite)**

Es un conjunto de protocolos de comunicación que se utiliza en Internet y en otras redes similares. Sus componentes más importantes son TCP e IP, y se encuentran en representados en un modelo de capas, que van desde la capa de enlace hasta la capa de aplicación, pasando por la capa de Internet y la de transporte.. 88

**TCP/IP (Internet Protocol Suite)**

Es un conjunto de protocolos de comunicación que se utiliza en Internet y en otras redes similares. Sus componentes más importantes son TCP e IP, y se encuentran en representados en un modelo de capas, que van desde la capa de enlace hasta la capa de aplicación, pasando por la capa de Internet y la de transporte.. 87

**TCP/IP (Internet Protocol Suite)**

Es un conjunto de protocolos de comunicación que se utiliza en Internet y en otras redes similares. Sus componentes más importantes son TCP e IP, y se encuentran en representados en un modelo de capas, que van desde la capa de enlace hasta la capa de aplicación, pasando por la capa de Internet y la de transporte.. 87, 89, 90, 91

**TCP/IP (Internet Protocol Suite)**

Es un conjunto de protocolos de comunicación que se utiliza en Internet y en otras redes similares. Sus componentes más importantes son TCP e IP, y se encuentran en representados en un modelo de capas, que van desde la capa de enlace hasta la capa de aplicación, pasando por la capa de Internet y la de transporte.. 87, 89

**TCP/IP (Internet Protocol Suite)**

Es un conjunto de protocolos de comunicación que se utiliza en Internet y en otras redes similares. Sus componentes más importantes son TCP e IP, y se encuentran en representados en un modelo de capas, que van desde la capa de enlace hasta la capa de aplicación, pasando por la capa de Internet y la de transporte.. 87

**TCP/IP (Internet Protocol Suite)**

Es un conjunto de protocolos de comunicación que se utiliza en Internet y en otras redes similares. Sus componentes más importantes son TCP e IP, y se encuentran en representados en un modelo de capas, que van desde la capa de enlace hasta la capa de aplicación, pasando por la capa de Internet y la de transporte.. 83, 84, 91, 94

# Bibliografía

- [1] Craig Larman. Applying UML and Patterns second edition.
- [2] Maximiliano Odstreil. Apuntes de Clase Ingeniería de Software I. 2009 .
- [3] Maximiliano Odstreil. Apuntes de Clase Ingeniería de Software II. 2010.
- [4] Documentación oficial de PhoneGap.(<http://phonegap.com/>)
- [5] Thomas Myer. Beginning PhoneGap
- [6] Documentación oficial de Backbone.js.(<http://backbonejs.org/>)
- [7] Documentación oficial de jQuery Mobile.(<http://jquerymobile.com/>)
- [8] Maximiliano Firtman. jQuery Mobile: Up and Running.
- [9] Roger S. Pressman. Software Engineering sixth edition.
- [10] Wikipedia en inglés (<http://en.wikipedia.org/>).
- [11] Wikipedia en castellano (<http://en.wikipedia.org/>).
- [12] Wikilibros: Manual de L<sup>A</sup>T<sub>E</sub>X.
- [13] Gabriel Valiente Feruglio. Composición de textos científicos con L<sup>A</sup>T<sub>E</sub>X. 1999.
- [14] átopos. L<sup>A</sup>T<sub>E</sub>Xpara Humanidades. 28 de Noviembre de 2005.