



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

GÓMEZ VÉLIZ, Kevin Shionen (CX 1411828) - Ing. en Computación

Tutores

TUTOR - Ing. COHEN Daniel Eduardo

COTUTOR - Ing. NIETO Luis Eduardo

1 de Diciembre 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. Ademas 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

Introducción	6
Objetivos del sistema	8
Selección del modelo de ciclo de vida y de la metodología de desarrollo	8
Disciplina de Requisitos	10
Introducción	10
Identificación de usuarios participantes	10
Educción de requisitos	11
Estudio de Documentación. Planificación y Realización de Entrevistas . . .	11
Especificación de Requisitos de Software	11
Introducción	11
Objetivos y alcance del sistema	12
Definiciones, acrónimos y abreviaturas	12
Definiciones	12
Descripción general	14
Proceso actual	14
Reingeniería de proceso	15
Requisitos específicos	17
Suposiciones y Dependencias	22
Requisitos de Usuario y Tecnológicos	22
Requisitos de Interfaces Externas	23
Requisitos de Rendimiento	24
Requisitos de Desarrollo y Restricciones de Diseño	25

ÍNDICE GENERAL 3

Ajuste a estándares	25
Seguridad	25
Estimación del proyecto	27
Planificación de etapas	27
Duración estimada de tareas	28
Diagrama de Gantt	28
Disciplina de Análisis	29
Vista de Casos de Uso	29
Diagramas de Casos de Uso	29
Diagrama General de Casos de Uso	30
Gestión de Usuarios	31
Gestión de Registros	31
Casos de Uso Principales	32
Disciplina de Diseño	33
Descripción Textual de Casos de Uso	33
Diagrama de Clases	59
Vista general	60
Diagramas de Actividad	61
Inicio de Sesión	61
Registrar Usuario	62
Crear Registro	63
Ver Mapa Interactivo	64
Diagramas de Secuencia	65
Iniciar Sesión	65
Registrar	66
Crear Registro	67
Interfaz de usuario	68
1. Aplicación móvil	68
2. Sistema WEB	78

ÍNDICE GENERAL	4
-----------------------	---

Disciplina de Implementación	83
Diagrama conceptual estructura Cliente/Servidor/Internet	83
Arquitectura de la aplicación	84
Modelo-Vista-Controlador	84
Componentes del patrón	85
Elección del Lenguaje	85
Apache Cordova	86
Ventajas de Apache Cordova	87
Desventaja de Apache Cordova	87
Web services	88
Razones para crear servicios Web	88
REST	89
Express.js	90
Herramientas de desarrollo	90
Persistencia de clases	91
Modelo físico del sistema	91
Modelo relacional. Vista de la base de datos	92
Disciplina de Pruebas	93
Test de Unidades	93
Introducción	93
Pruebas de Caja Blanca	93
Test de Módulos	94
Introducción	94
Pruebas de Caja Negra	94
Pruebas de Estrés	94
Test de Integración	95
Introducción	95
Pruebas de Integración	95
Test de Aceptación	96
Introducción	96

ÍNDICE GENERAL

5

Prueba Alfa	97
Prueba Beta	105
Conclusiones	106

Introducción

El proyecto surge del convenio de la cátedra de Sistemas con Microprocesadores de la Facultad de Ciencias Exactas y Tecnología de la UNT, con la Facultad de Ciencias Naturales e Instituto Miguel Lillo, para el desarrollo de una solución tecnológica para el control del estado de contaminación del agua en ríos y afluentes en la provincia de Tucumán.

Este proyecto consiste en mejorar los procesos actuales que se llevan a cabo para el control de contaminación del agua. El mismo 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 (diferentes especies de insectos).

Esto se logra mediante salidas de campo de alumnos de escuelas rurales (usuarios de la aplicación) donde se busca obtener una muestra de agua del afluente en estudio; una vez obtenida la muestra y mediante el uso de la aplicación, se establece la biodiversidad que allí se encuentra y se calcula el índice de contaminación.

Por otro lado, 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 solo la gran cantidad de usuarios de estas plataformas, sino también la posibilidad de ofrecer nuevas funcionalidades y capacidades, previamente imposibles, para sus procesos actuales.

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.

Objetivos del sistema

Objetivo general:

Generar registros digitales, de muestras del agua de ríos/afluentes, realizados por los alumnos en salidas de campos para el estudio de la contaminación.

Estos registros deben contener fotos de las muestras (insectos encontrados en el agua), coordenadas geográficas (obtenida mediante GPS integrado del dispositivo móvil), foto del paisaje para futuras referencias, y el índice de contaminación en cuestión (calculado en base a las diferentes especies de insectos encontrados).

Objetivos principales

- 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 están 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, las cuales proporcionan un punto de referencia para validar el sistema final, que comprueben que se ajusten 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 Recomended 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 geográficas

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.
 - Tricópteros.
- *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 contaminación media
3	3	En buen estado
4	4	En excelente estado

- *Foto paisaje:* Foto obtenida del paisaje en donde se realizó la muestra de los insectos encontrados, con el fin de facilitar un punto de referencia visual para próximas 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 cuestión.
- *Coordenadas geográficas:* Se usan para referenciar, mediante latitud y longitud, el lugar en donde se realizó 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 elaboración de páginas web. En nuestro caso, genera la vista final del sistema.
- *BD:* Base de datos.
- *CRUD:* Es el acrónimo de “Crear, Leer, Actualizar y Borrar” (del original en inglés: 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, identificando 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 el 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 guardarán 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 visualizará una ruleta virtual animada, la cual mostrará un valor (índice de contaminación), indicando el posible grado 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 solo 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 generar registros. Los administradores pueden interactuar con los mismos, 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 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: IdUsuario + 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. En caso de no encontrar un usuario con los nombre de usuario y mail especificados, se creará y guardará el nuevo usuario, al cual se le asignará el número único IdUsuario (este número se obtiene a partir del máximo existente hasta el momento, siendo 1 para el primer usuario).

En este caso, se devolverá un mensaje de éxito y el IdUsuario. En caso que ya existiera un usuario con los nombre de usuario y mail especificados, se devolverá un mensaje de error informando tal situación

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 Actual

Proceso: El sistema comprueba la existencia previa de un usuario en base a @IdUsuario para poder modificar su estado. Hay dos tipos de estado: “Habilitado” - “Inhabilitado”. Este proceso permuta el estado actual del usuario. Si un usuario tiene estado “Habilitado”, este se cambia a “Inhabilitado” y viceversa. Los registros asociados al 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 + Cantidad Registros

- **Cambiar contraseña de usuario:**

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

Entrada: @IdUsuario + Contraseña Anterior + Contraseña Nueva

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. Hay dos tipos de roles: “Administrador” - “Usuario”. Este proceso modificará el Rol del mismo según valor de entrada. 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 mostrará un mensaje vacío, indicando que la búsqueda no arrojo resultados.

Salida: Usuario + Nombre + Apellido + Institución + Grado + Residencia + Email + Foto Perfil + Cantidad Registros

B. Gestión de Registros

- **Alta de registros:**

Introducción: El sistema permite dar de alta un nuevo registro ingresando índice, 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, lo-

calidad, provincia).

Entrada: IdRegistro + Indice + Insectos Encontrados + Fecha Creación + Latitud + Longitud + Foto Paisaje + Foto Muestra + Foto Mapa + Observaciones Usuario + IdUsuario + IdUbicación

Proceso: El sistema crea un registro al cual se le asignará el número único IdRegistro (este número se obtiene a partir del máximo existente hasta el momento, siendo 1 para el primer registro). Los estados de validación posibles son “Valido”, “Invalido” y “Pendiente”. Un registro nuevo se creará con un valor de estado de validación igual a “Pendiente”. y con observaciones del administrador sin contenido, además, las fotos se almacenarán en Base64. 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 + Observaciones Administrador

Proceso: El sistema modifica el registro agregando una observación de adminis-

trador. 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 indice, 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: Indice + Insectos Encontrados + Fecha Creación + Latitud + Longitud + Foto Paisaje + Foto Muestra + Foto Mapa + Observaciones Usuario + Observaciones Administrador + 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 indice.

Entrada: Fecha Inicio Fecha Fin + Estado Validación + Indice

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 mostrará 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: Indice + Insectos Encontrados + Fecha Creación + Latitud + Longitud + Observaciones Usuario + Observaciones Administrador + 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 indice 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 intervenientes 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 (Sistema de administración y gestión de registros) 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 última 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 menciono, 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 ejecutarán la aplicación deberán tener las siguientes características independientemente de su S.O:
 - ◊ Cámara fotográfica de 1 Mega Pixeles 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 afectado de manera 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 Design (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 viajará 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 definen 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 prevé la implementación del protocolo de aplicación HTTPS que permite encriptar el tráfico 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. Ademas, 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	-

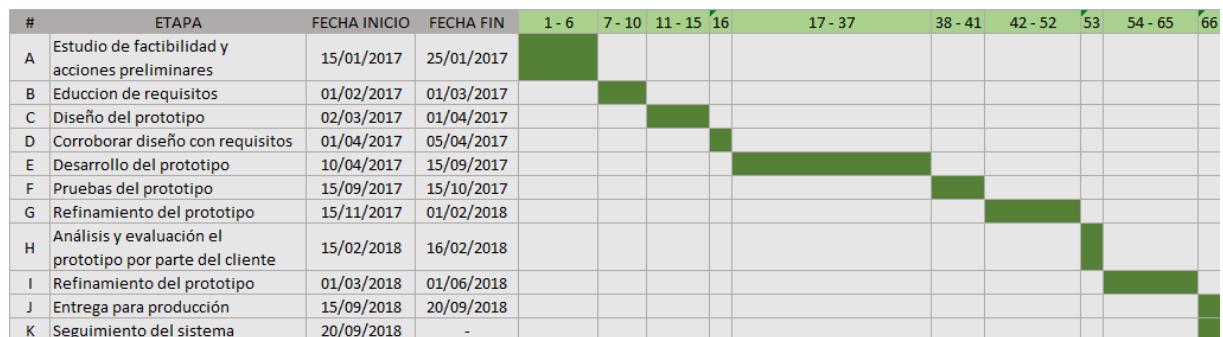
Cuadro 1: Fechas tentativas por etapa.

Duración estimada de tareas

Etapas	Semanas	Valor %
Estudio de factibilidad y acciones preliminares	6	9 %
Educción de requisitos	4	6 %
Diseño del prototipo	5	8 %
Corroborar diseño con requisitos	1	1 %
Desarrollo del prototipo	21	33 %
Pruebas del prototipo	4	6 %
Refinamiento del prototipo	11	17 %
Análisis y evaluación el prototipo por parte del cliente	1	1 %
Refinamiento del prototipo	12	18 %
Entrega para producción	1	1 %
Seguimiento del sistema	-	-
Total	66	100 %

Cuadro 2: Estimación en semanas - Valor porcentual

Diagrama de Gantt



Disciplina de Análisis

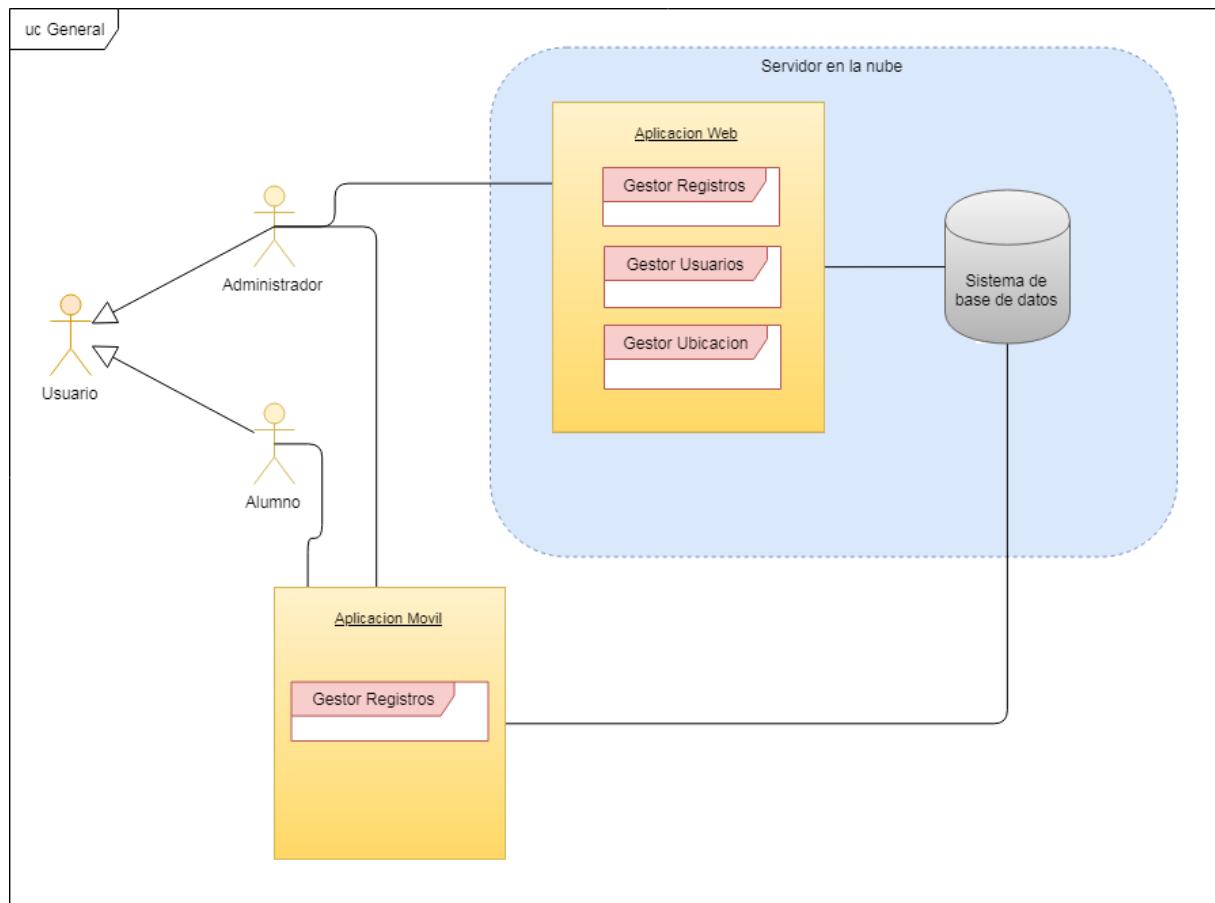
Vista de Casos de Uso

La vista de casos de uso captura el comportamiento de un sistema, subsistema, clase o componente, como lo ve un usuario externo. Partitiona 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.

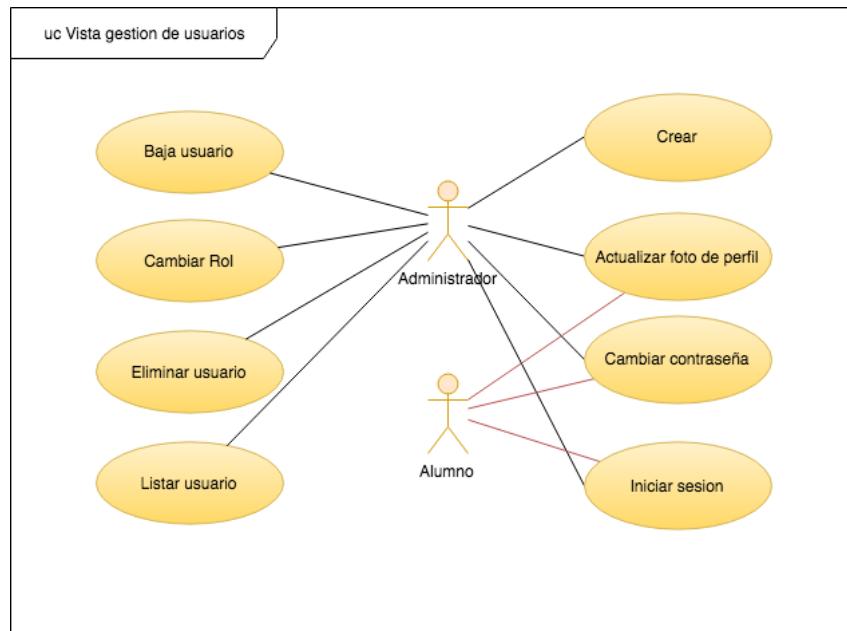
Diagramas de Casos de Uso

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)

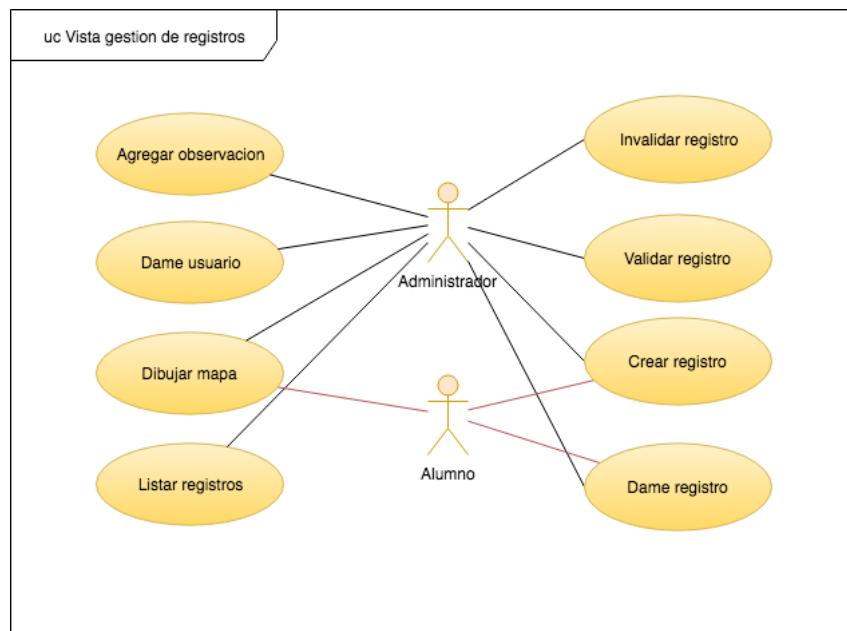
Diagrama General de Casos de Uso



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 Pareto¹, se hicieron los diagramas de casos de uso más relevantes. Basta con ellos para entender el comportamiento general de la aplicación, dado que se hace extensible su razonamiento a los faltantes.

Descripción Textual de Casos de Uso

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

CU01	Registrar				
Revisa:		Fecha		Firma:	
Resumen: Este caso de uso permite a los usuarios registrarse como usuarios de la aplicación, permitiendo introducir sus datos personales					
Actores: Usuario (primario). Servidor (en adelante S. Secundario)					
Personal Involucrado y Metas: <i>Usuario:</i> quiere transformarse en un usuario del sistema, así pueda realizar las transacciones con la aplicación de un modo seguro y personalizado. <i>Servidor:</i> quiere registrar la mayor cantidad de usuarios posibles y que el proceso sea lo más rápido y seguro posible.					

¹Cuando 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 solo un 20 % del esfuerzo”

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:

Resumen: 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.

Actores: Usuario (Primario). Servidor (en adelante S. Secundario)

Personal Involucrado y Metas:

Usuario: quiere que el sistema lo reconozca como tal, así pueda realizar las tareas con la aplicación de un modo seguro y personalizado.

App: requiere identificar, de manera local, confiablemente a sus usuarios de manera de satisfacer sus intereses en cuanto a seguridad, accesos a su cuenta personal y datos privados.

Precondiciones: 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 información 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:
Resumen: 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		
Actores: Usuario (Primario). Servidor (en adelante S. Secundario)		
Personal Involucrado y Metas: <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.		
Precondiciones: Los usuarios deben estar autenticados en APP		
Poscondiciones: 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 creación 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 conexión a internet y envía de manera segura los datos a S para que sean validados.
6. S valida los datos, realiza la creación 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 realizó.

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

Resumen: Este caso de uso permite al usuario ver el mapa general con la información de todos los registros.

Actores: 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 información 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:

Resumen: Este caso de uso permite al usuario ver su perfil con la información de sus datos.

Actores: Usuario (Primario). Servidor (Secundario)

Personal Involucrado y Metas:

Usuario: quiere visualizar de manera completa la información de su perfil.

Servidor: quiere que el usuario pueda ver la información relacionada con su perfil de forma segura.

Precondiciones: El usuario debe estar autenticado en APP.

Poscondiciones: Se muestra la información completa del perfil del usuario.

Escenario Principal:

1. El usuario selecciona la opción para ver su perfil.
2. APP envía la solicitud a S.
3. S envía los datos necesarios para generar la vista del perfil a APP.
4. APP muestra el perfil de usuario por pantalla.

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:
Resumen: Este caso de uso permite al usuario editar los datos de su perfil.		
Actores: Usuario (Primario). Servidor (Secundario)		
Personal Involucrado y Metas: <i>Usuario:</i> quiere editar los datos de su perfil. <i>Servidor:</i> 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: <ol style="list-style-type: none">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:	
Resumen: Este caso de uso permite al usuario eliminar su cuenta de usuario del servidor.					
Actores: Usuario (Primario). Servidor (Secundario)					
Personal Involucrado y Metas:					
<p><i>Usuario:</i> quiere eliminar su cuenta de usuario.</p> <p><i>Servidor:</i> quiere mantener actualizados los datos del usuario.</p>					
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:
Resumen: 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.				
Actores: Administrador (Primario). Servidor (en adelante S. Secundario)				

Personal Involucrado y Metas:

Administrador: quiere que el sistema lo reconozca como tal, así pueda realizar las tareas de gestión con la aplicación de un modo seguro y personalizado.

Servidor: requiere identificar confiablemente a sus administradores de manera de satisfacer sus intereses en cuanto a seguridad, accesos a su cuenta personal y datos privados.

Precondiciones: El administrador está registrado.

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

Escenario Principal:

1. El administrador ejecuta el sistema de gestión 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:
Resumen: Este caso de uso permite al administrador cambiar el rol a un usuario en particular.				
Actores: Administrador (Primario). Servidor (Secundario)				

Personal Involucrado y Metas:

Administrador: quiere cambiar el rol a un usuario.

Servidor: quiere mantener actualizados los datos del usuario.

Precondiciones: El administrador debe estar autenticado en SGR.

Poscondiciones: 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:
Resumen: Este caso de uso permite a un administrador listar la información de todos usuarios registrados en el sistema.				
Actores: Administrador (Primario). Servidor (Secundario).				
Personal Involucrado y Metas:				
<i>Administrador:</i> quiere visualizar de manera completa la información de los usuarios registrados.				
<i>Servidor:</i> 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:				
<ol style="list-style-type: none"> 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:
Resumen: Este caso de uso permite a un administrador listar la información de todos los registros realizados por la totalidad de los usuarios.				
Actores: Administrador (Primario). Servidor (Secundario).				
Personal Involucrado y Metas:				
<p><i>Administrador:</i> quiere visualizar de manera completa la información de los registros realizados.</p> <p><i>Servidor:</i> quiere que el administrador pueda ver la información relacionada con los registros de manera segura.</p>				
Precondiciones: El administrador debe estar autenticado en SGR.				
Poscondiciones: Se muestra un listado de registros realizados.				

Escenario Principal:

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:

Resumen: Este caso de uso permite al administrador ver un registro en particular con toda su información.

Actores: Administrador (Primario). Servidor (Secundario)

Personal Involucrado y Metas:

Administrador: quiere visualizar de manera completa la información de un registro.

Servidor: quiere que el administrador pueda ver la información relacionada con un registro de forma segura.

Precondiciones: El administrador debe estar autenticado en SGR.

Poscondiciones: Se muestra la información 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:
Resumen: Este caso de uso permite al administrador validar o invalidar un registro en particular.				

Actores: 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/Indice.

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

Resumen: Este caso de uso permite al administrador exportar un archivo formato excel con la información de los registros.

Actores: Administrador (Primario). Servidor (Secundario)

Personal Involucrado y Metas:

Administrador: quiere exportar un archivo excel con la información 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.

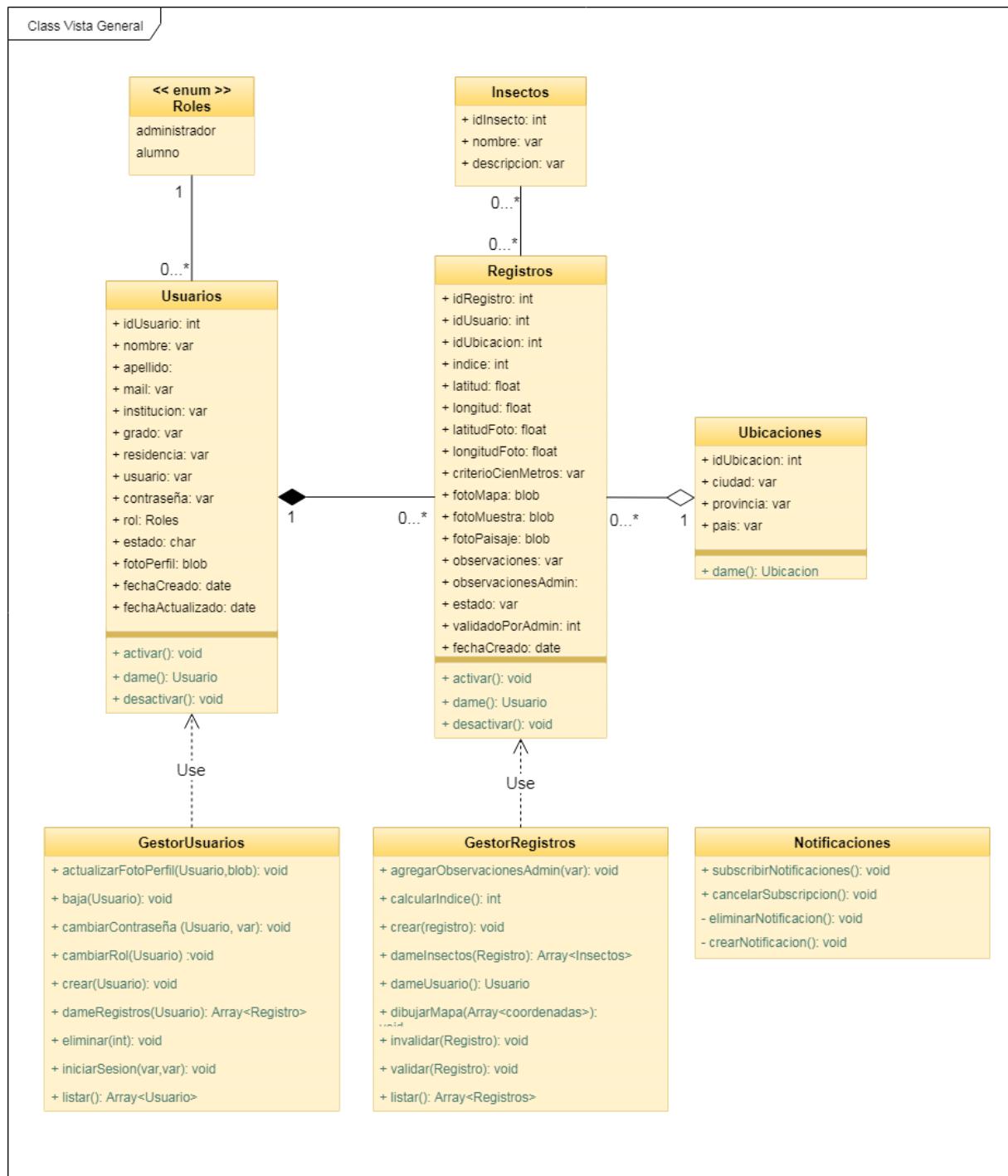
3. SGR informa al usuario el problema de conexión a través de un mensaje por la pantalla.

El escenario vuelve al punto 2.

Diagrama de Clases

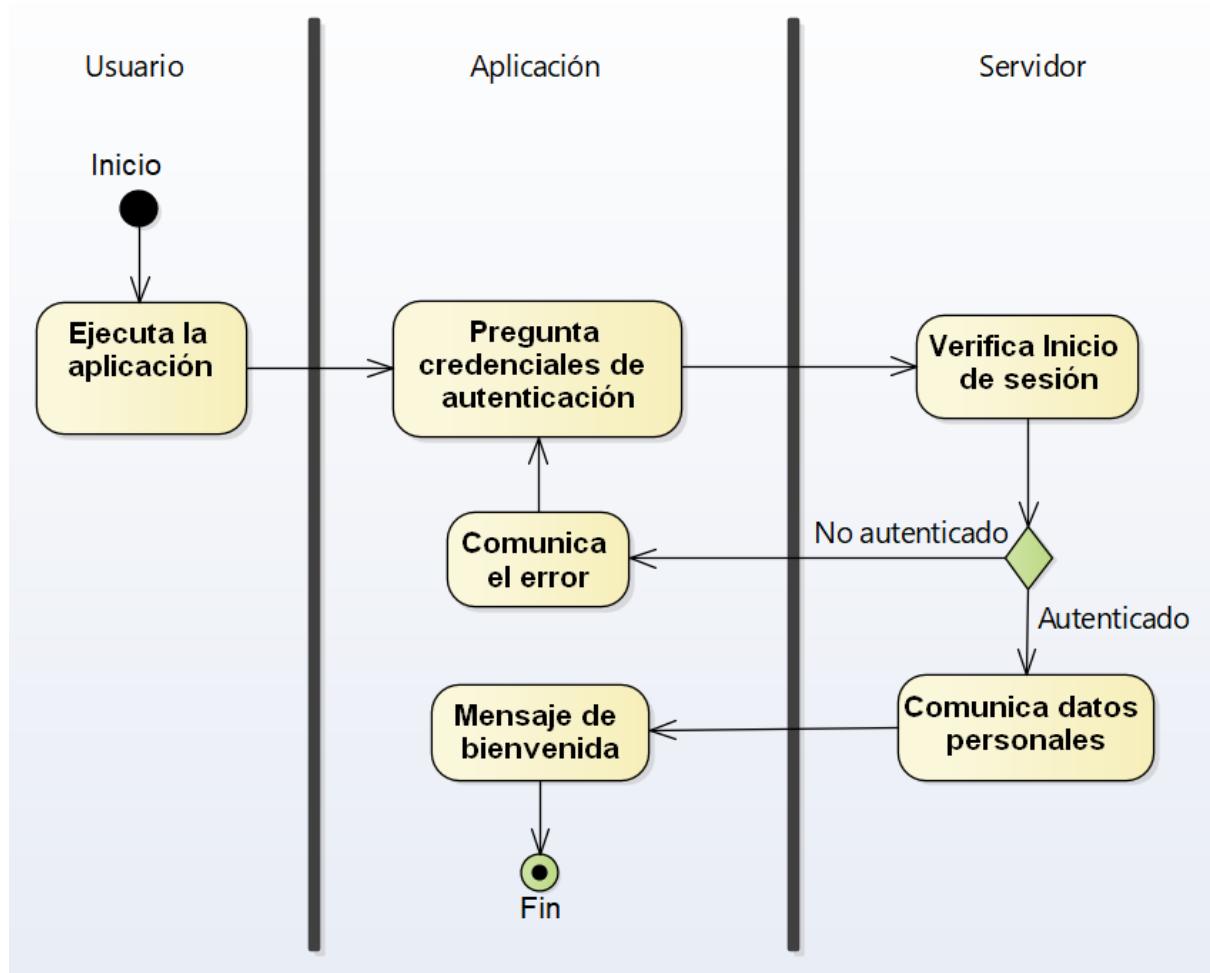
Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenido. Se puede utilizar un diagrama de clases para describir los tipos de datos y sus relaciones con independencia de su implementación. 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

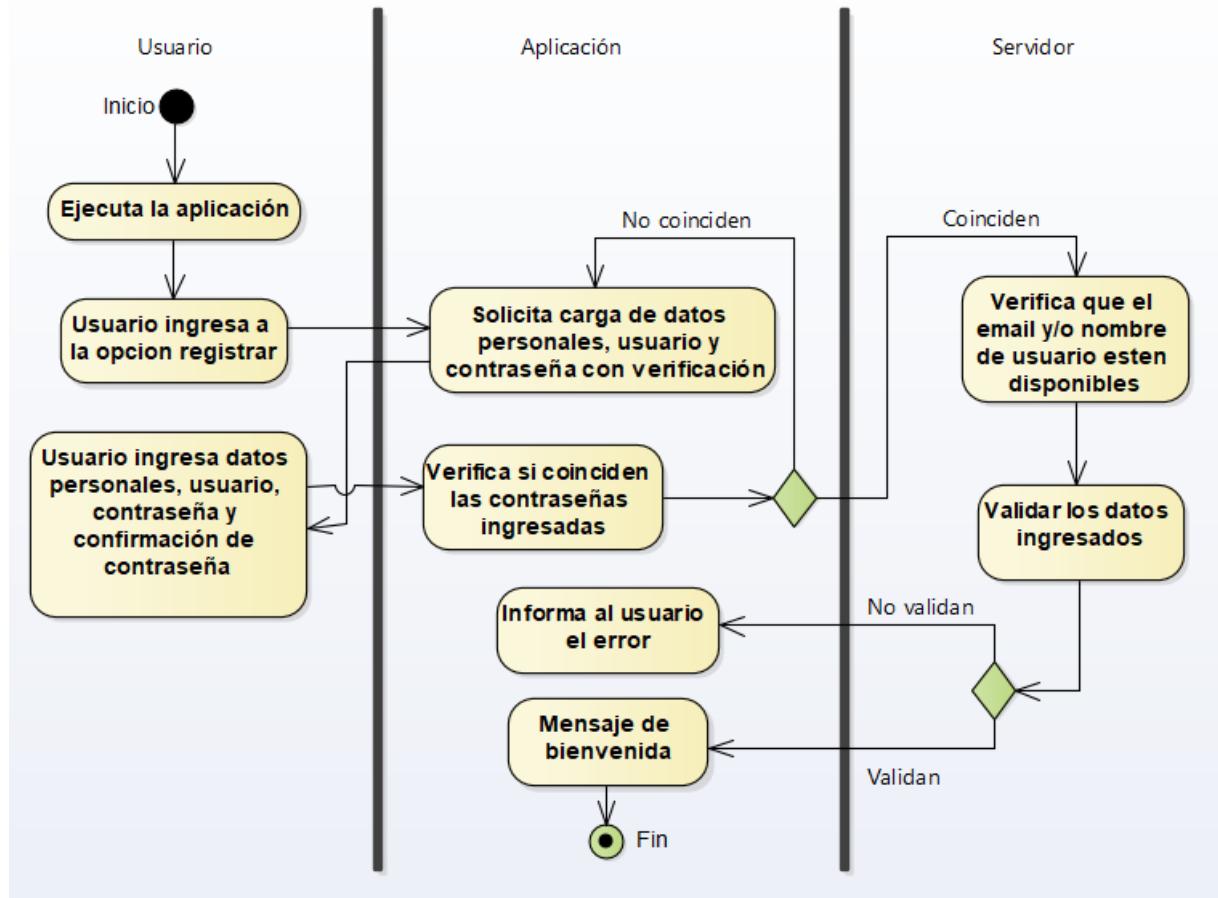


Diagramas de Actividad

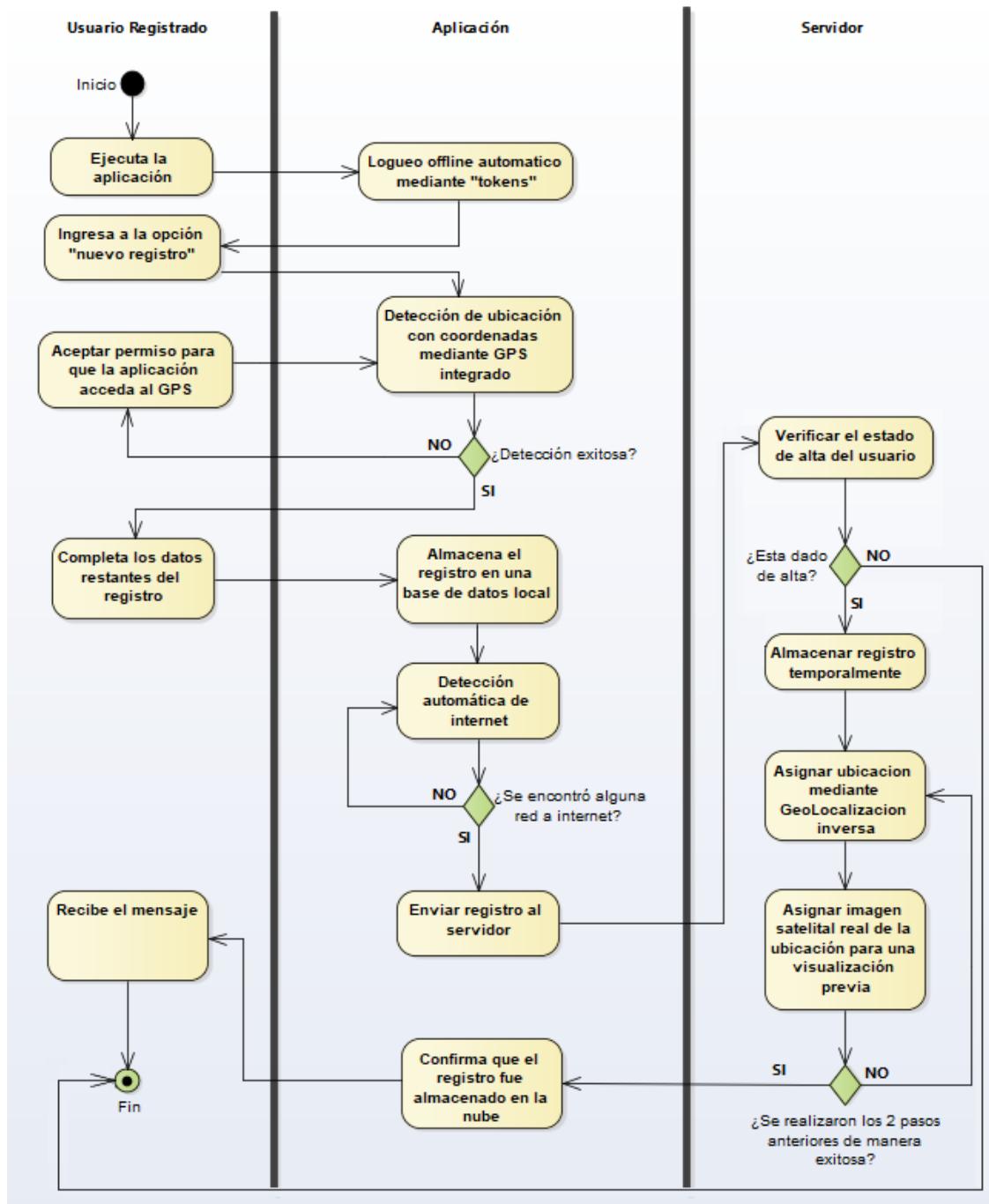
Inicio de Sesión



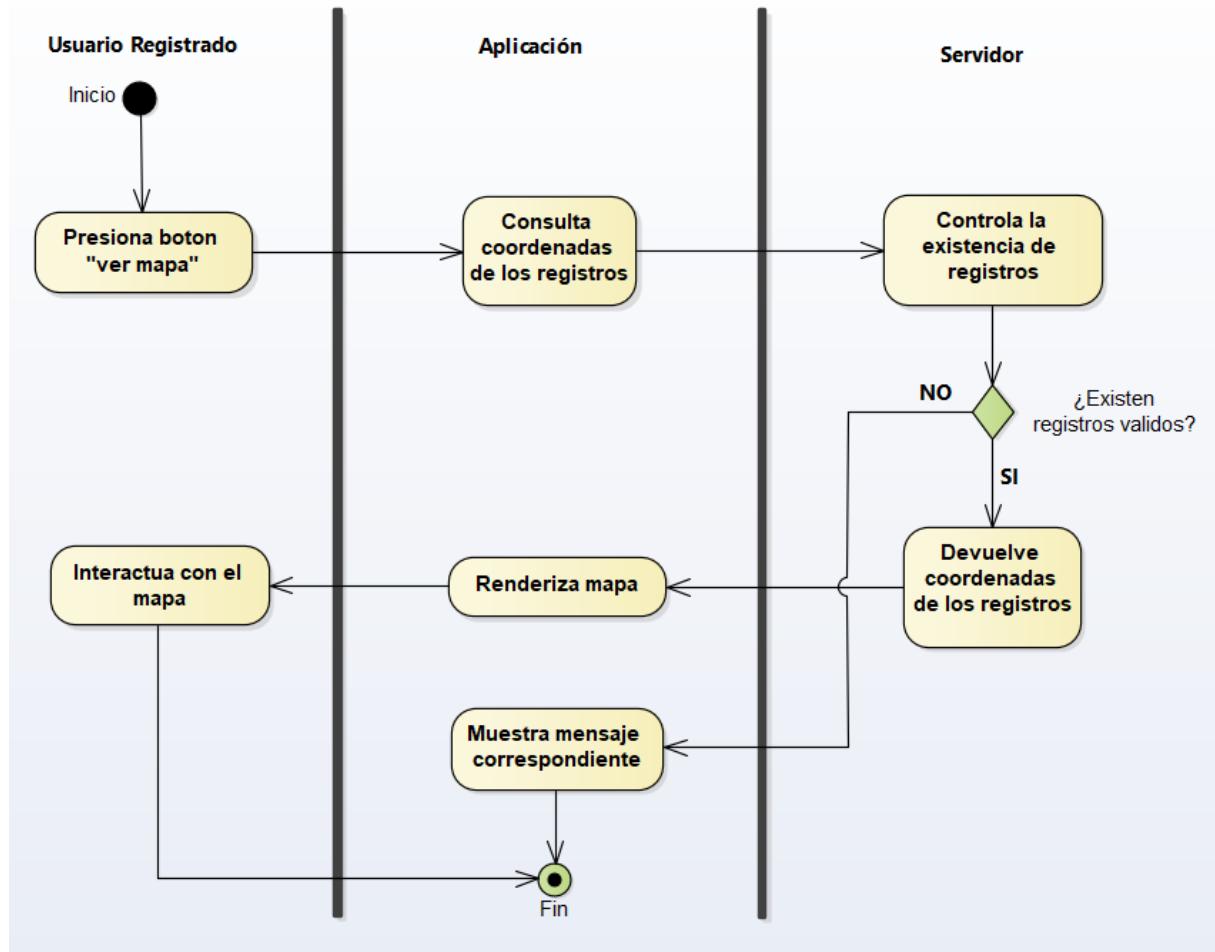
Registrar Usuario



Crear Registro

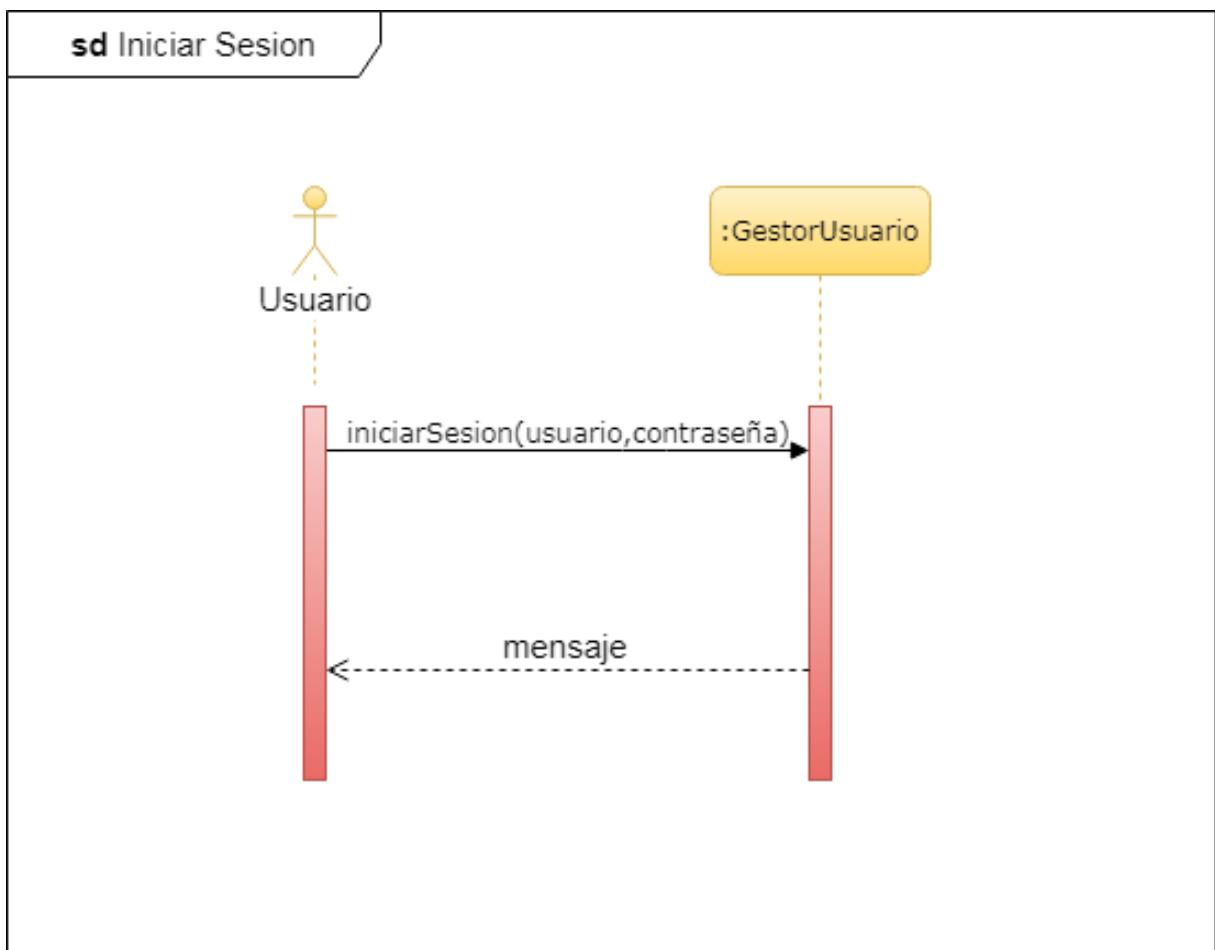


Ver Mapa Interactivo

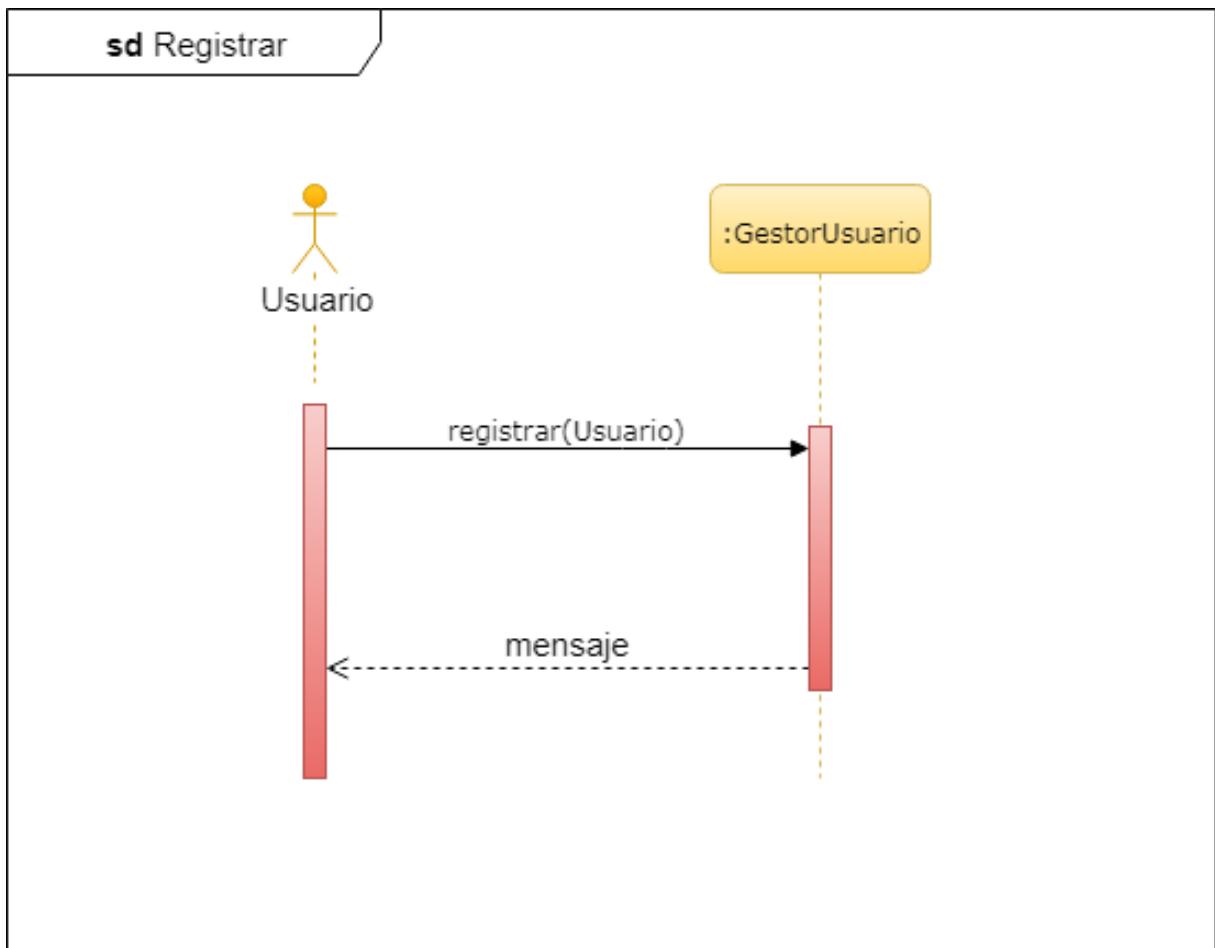


Diagramas de Secuencia

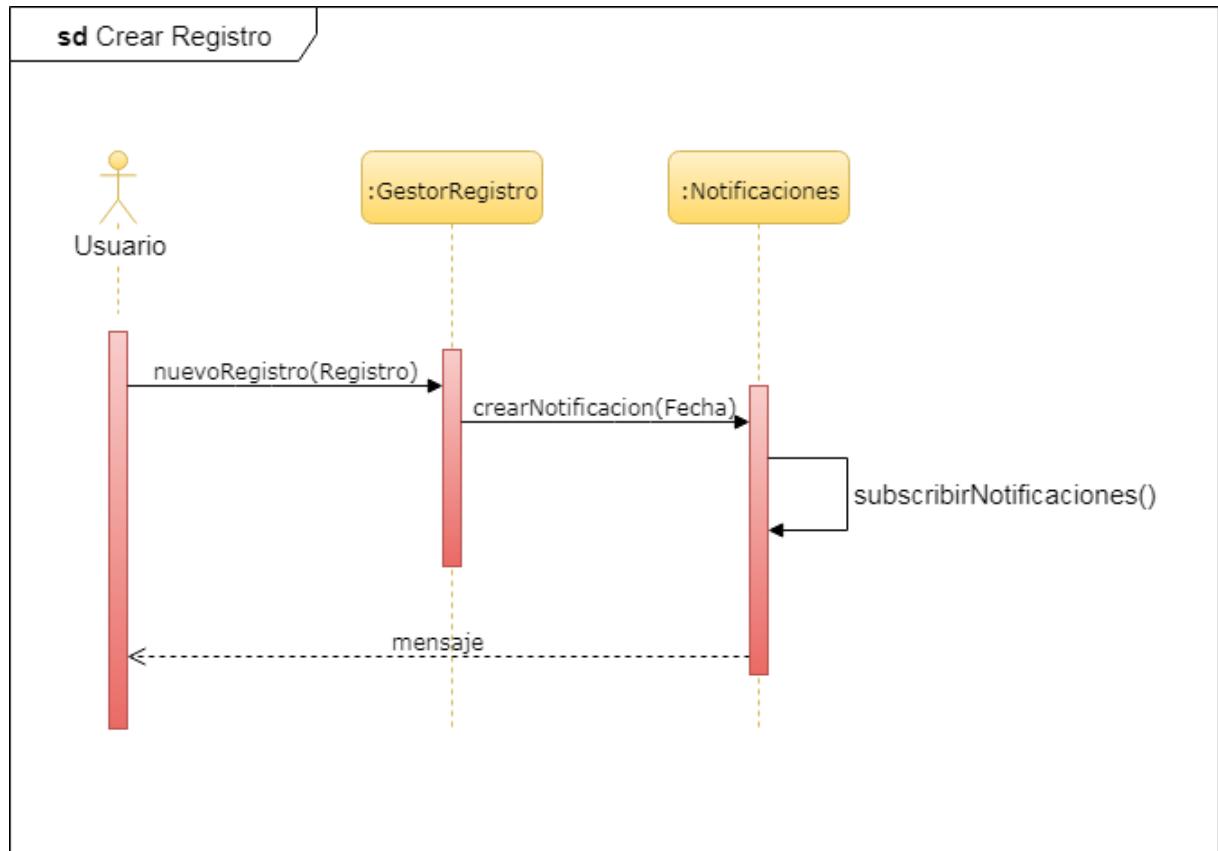
Iniciar Sesión



Registrar



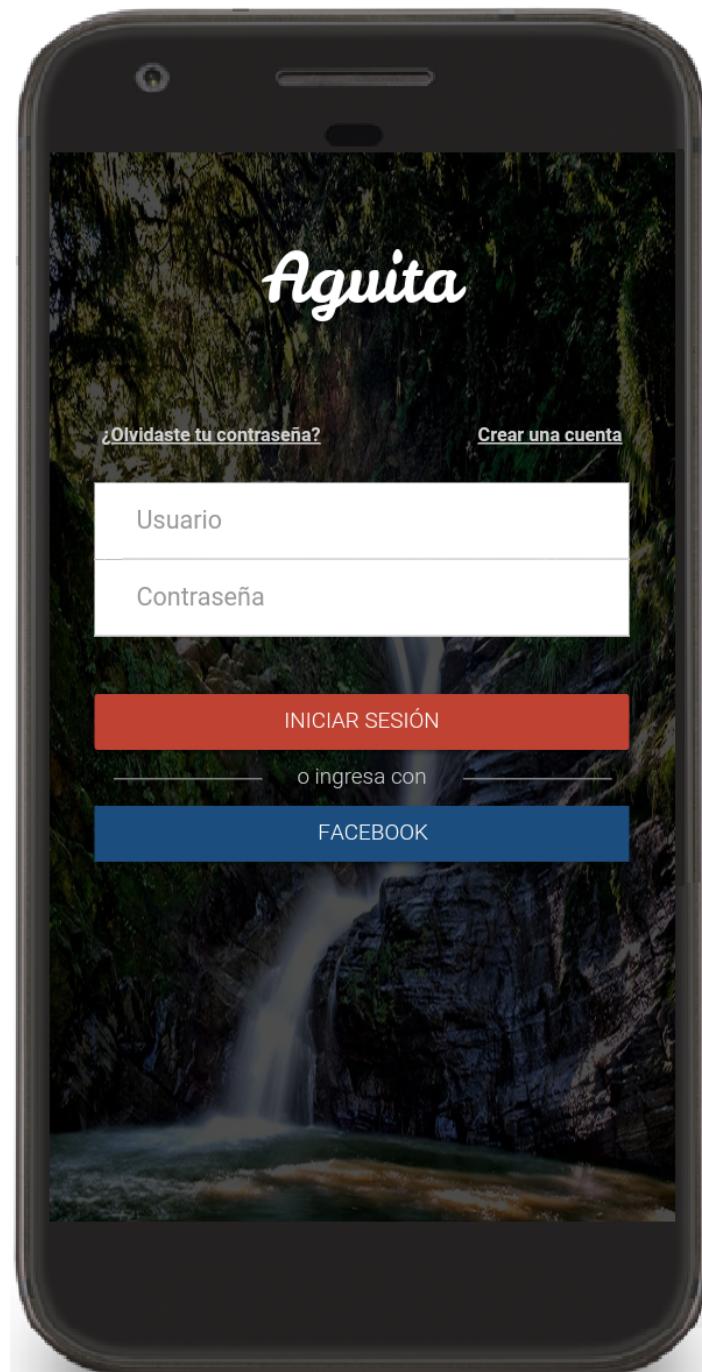
Crear Registro



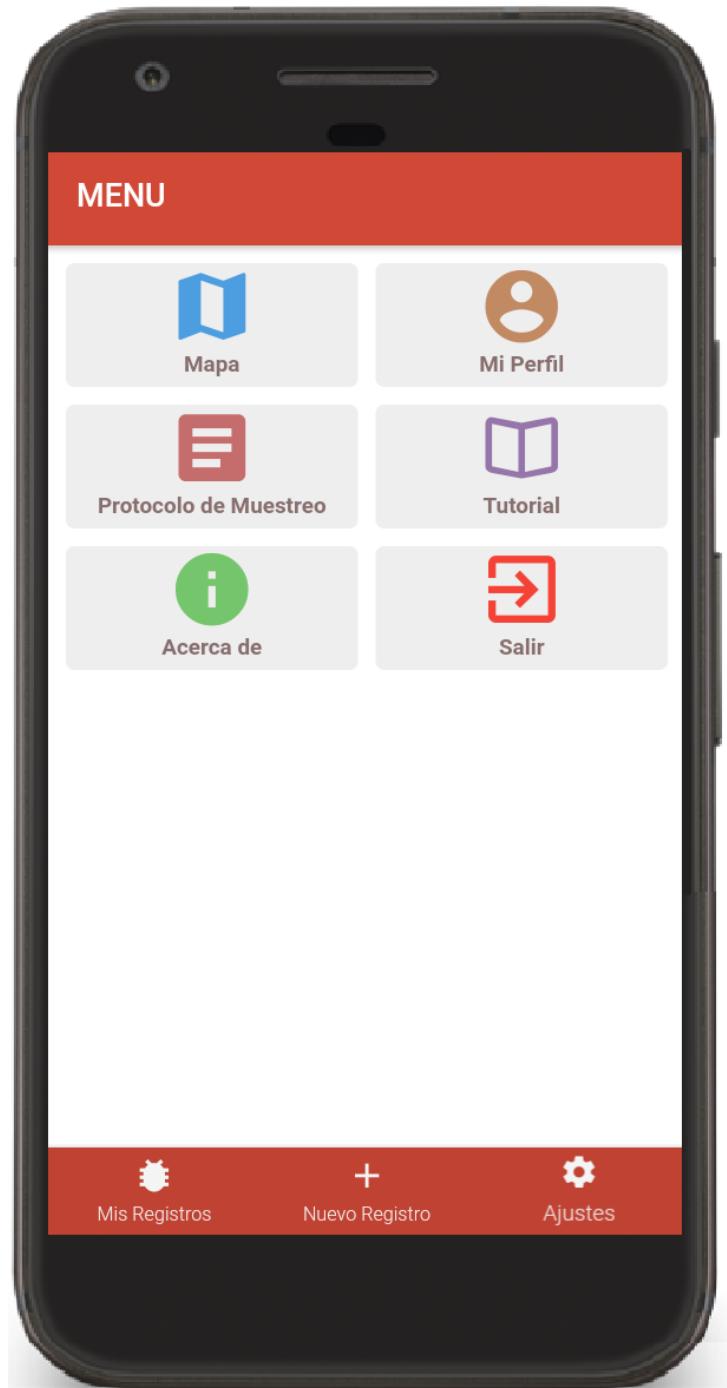
Interfaz de usuario

1. Aplicación móvil

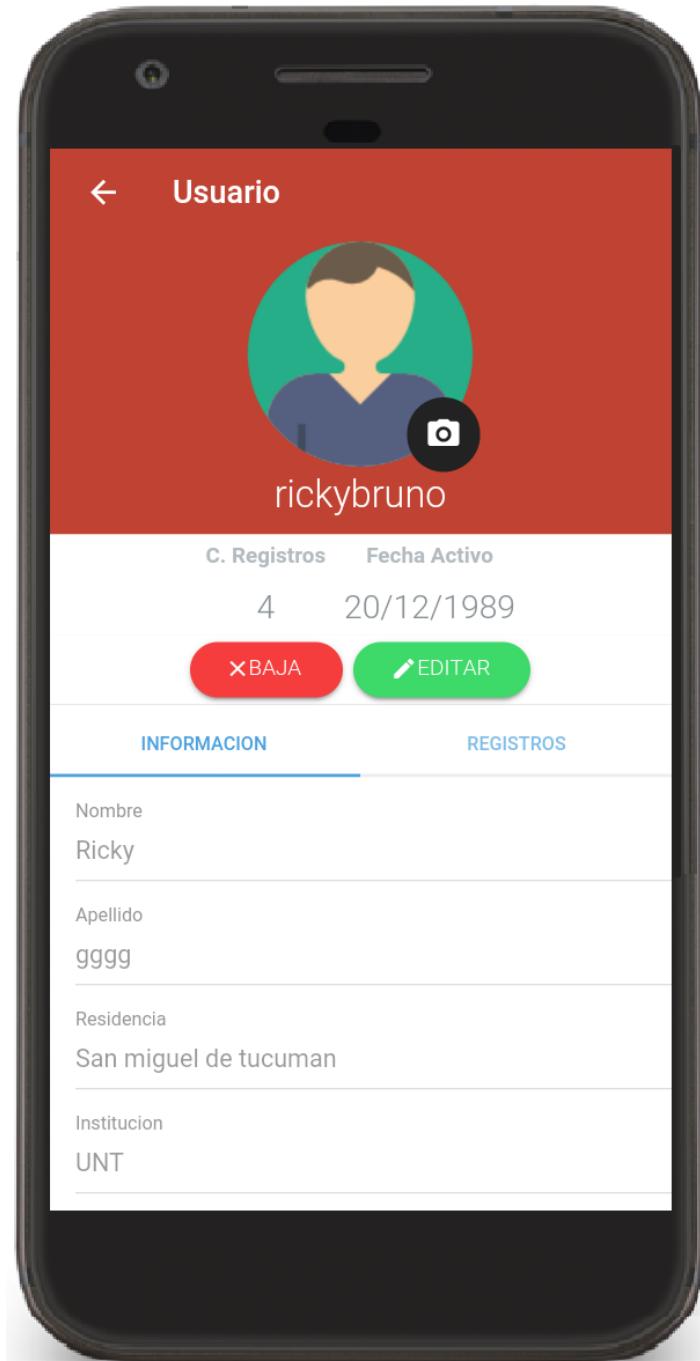
Inicio de sesión



Configuración



Perfil



Nuevo registro

Paso 1

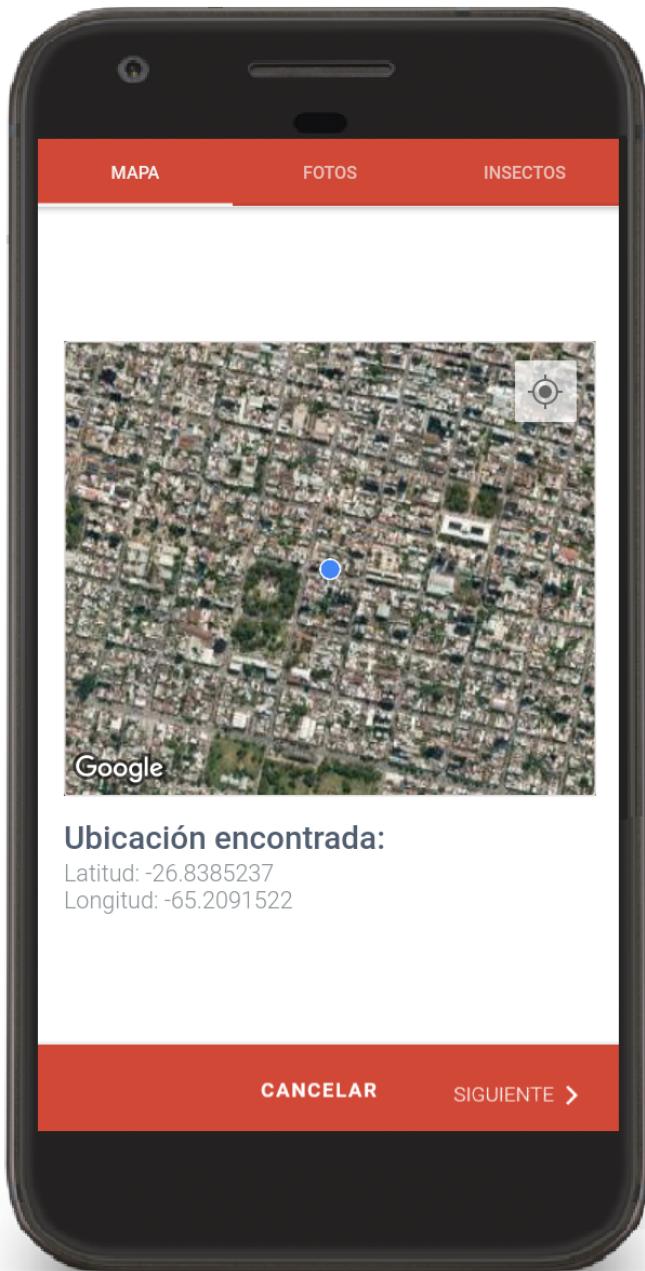


Figura 1: Obtención automática de coordenadas

Paso 2. Sin fotos



Figura 2: Obtención manual de fotografías (vista limpia)

Paso 2. Con fotos

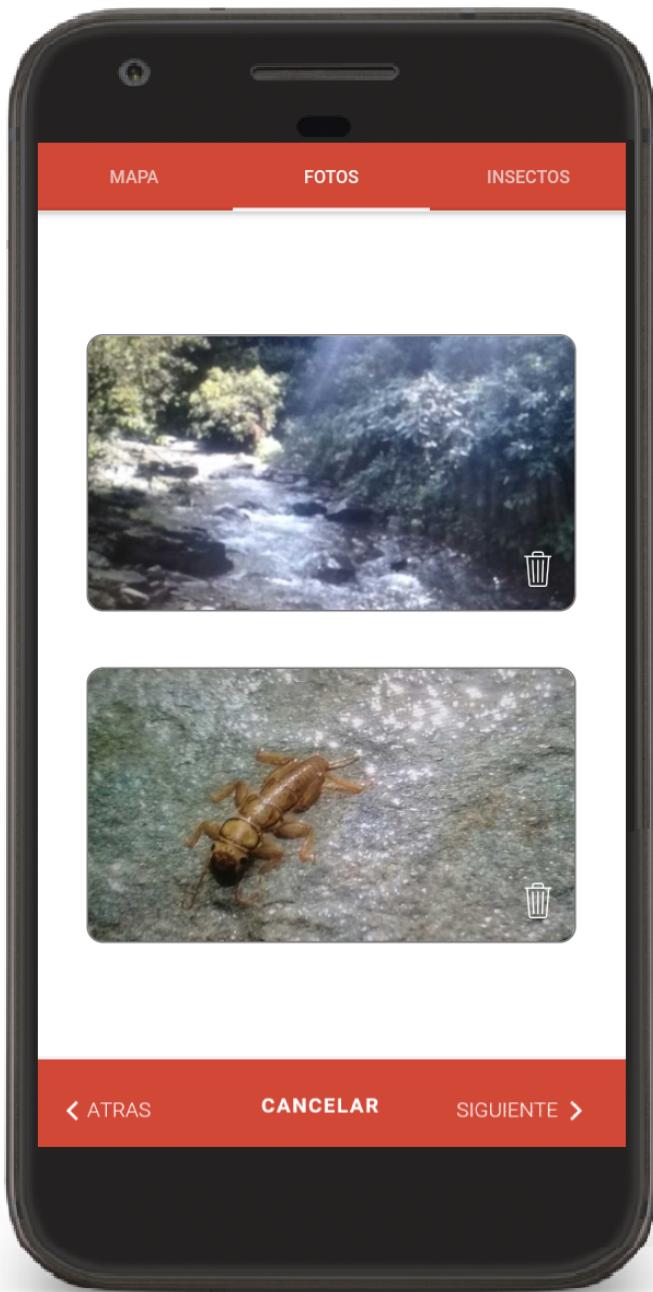


Figura 3: Vista con fotos capturadas

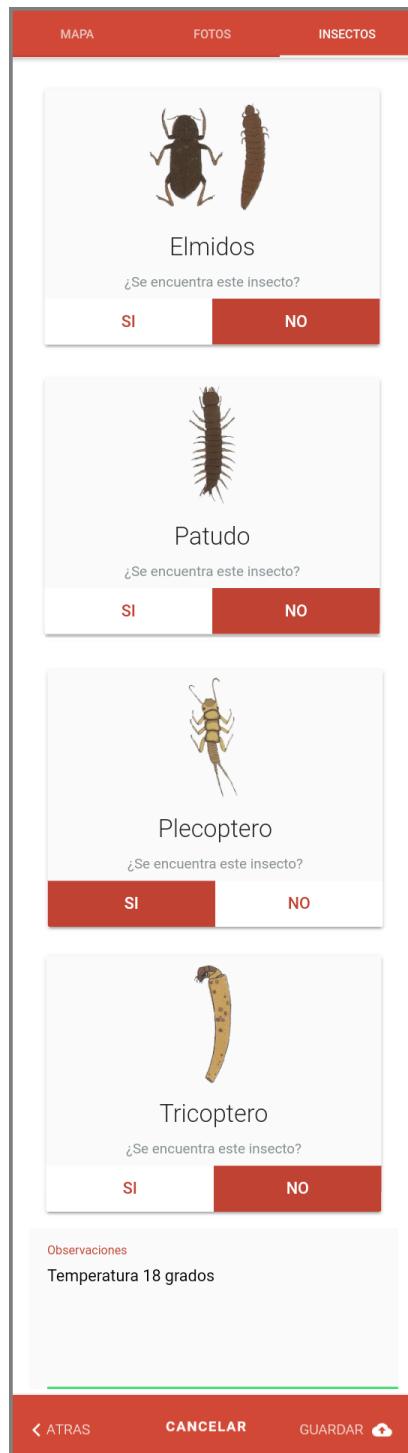
Paso 3. Imagen completa

Figura 5: Paso 3: Formulario para selección de insectos encontrados vista general a modo ilustrativo.

Paso 4



Figura 6: Paso 4: Ruleta animada indicadora de indice según coincidencias de insectos

Listar registros creados

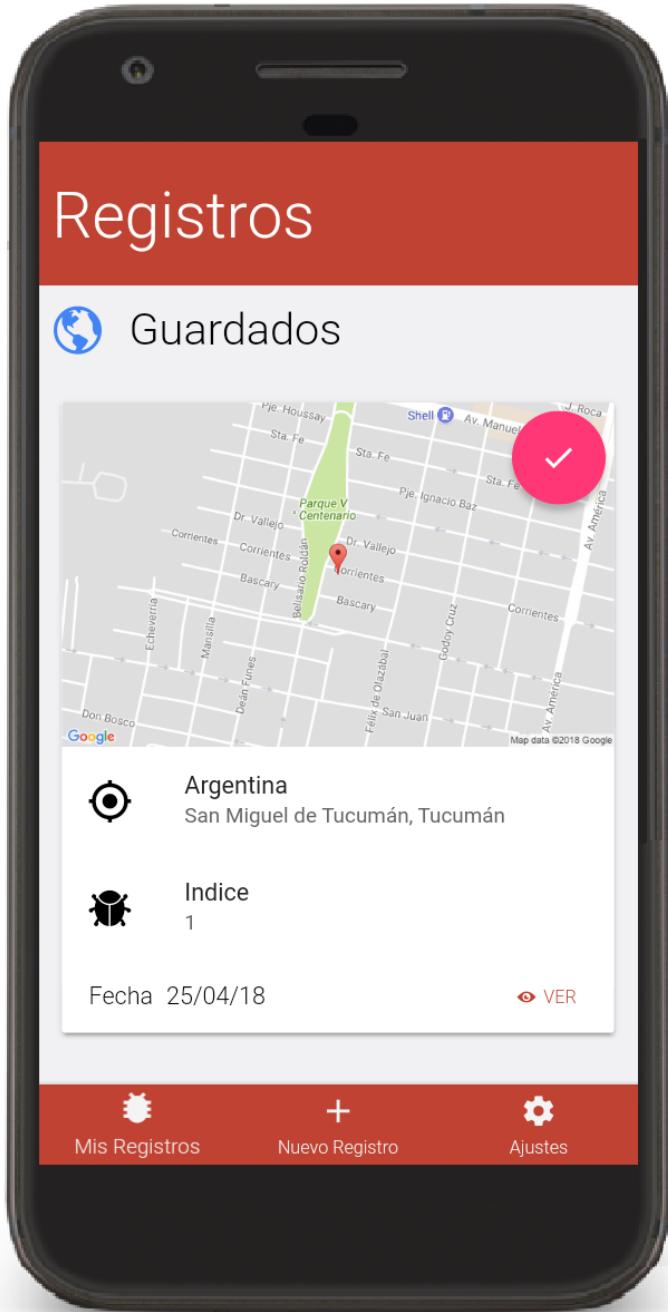


Figura 7: Lista con desplazamiento vertical para ver registros correspondientes al usuario logueado

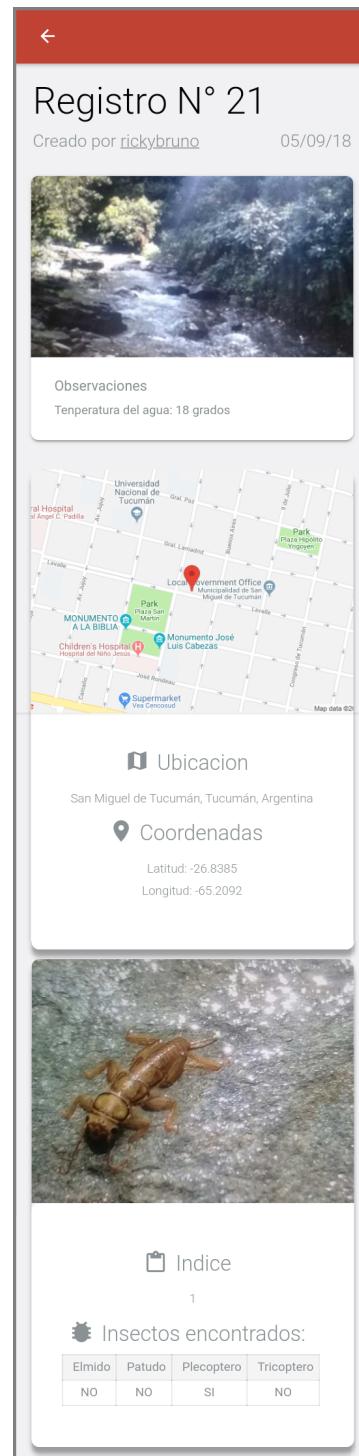
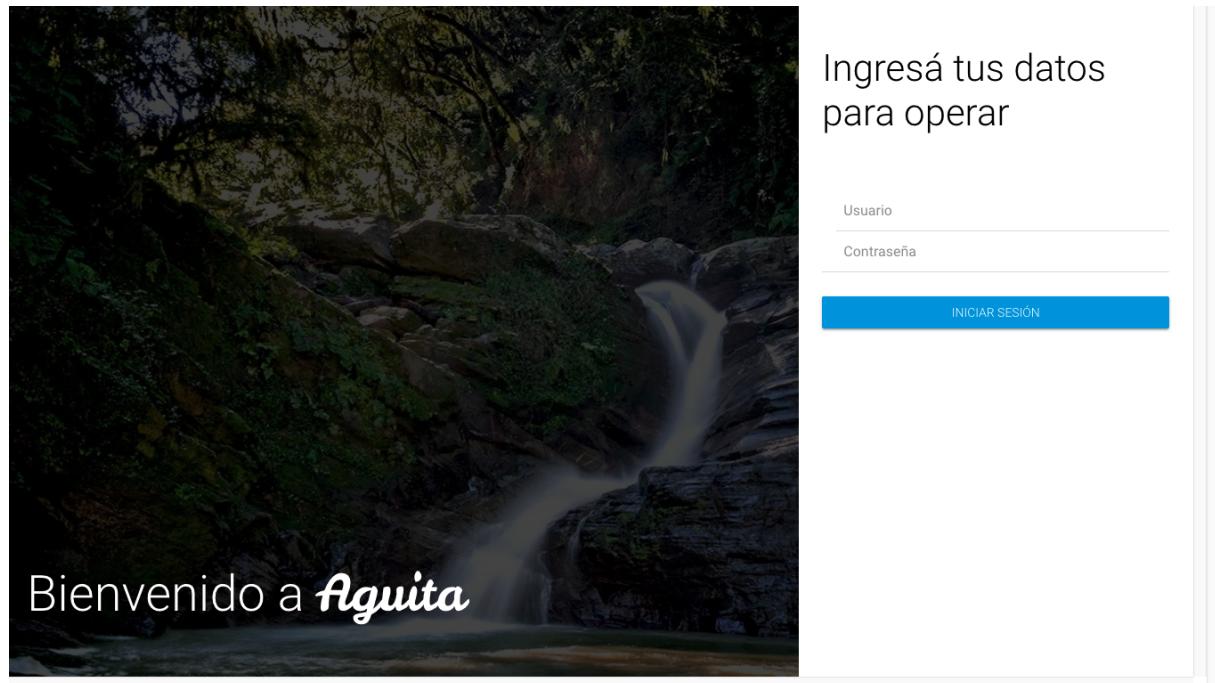
Vista completa

Figura 9: Vista completa de un registro con desplazamiento vertical. Modo ilustrativo

2. Sistema WEB

Inicio de sesión



Lista global de registros

This screenshot shows the main dashboard of the system. On the left is a sidebar with icons for Usuarios, Registros (selected), and Salir. The main area has a header "Registros" and a user profile "Bruno, Ricardo". Below the header is a search bar with filters for activity (Ultima Semana, Ultimo Mes, Desde, Hasta, Filtrar) and status (Pending, Valid, Invalid, All). A "Validar Selección" and "Invalidar Selección" button are also present. The main content area displays a table titled "Registros de los últimos 7 días (30 ago. 2018 - 06 sep. 2018)". The table has columns: Checkbox, Fecha, Alumno, Ubicacion, Estado, Indice, and Acciones. It shows 0 Registro. At the bottom, there's a pagination section with "Elementos por página: 3" and navigation arrows.

Fecha	Alumno	Ubicacion	Estado	Indice	Acciones

Ver registro

Figura 10: Visualización rápida de registro en el sector derecho de la lista del punto anterior

Lista global de usuarios

Figura 11: Pantalla de resultados de la búsqueda de usuarios con filtros

Ver perfil de usuario

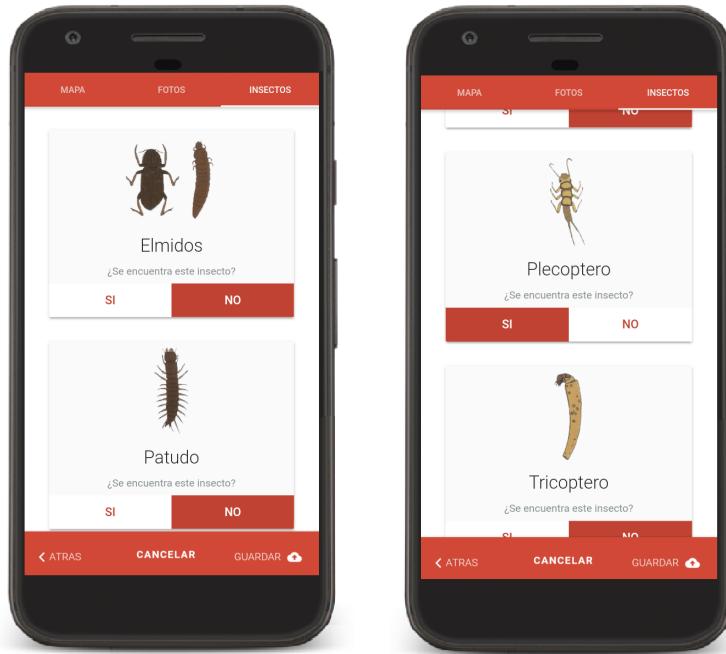
The screenshot shows a user interface for managing users. On the left, there's a sidebar with icons for Usuarios (selected), Registros, and Salir. The main area has a header "Usuarios" and a search bar "Apellido o Nombre o Mail" with a magnifying glass icon. Below that are buttons for "Filtros" (Filters), "Activos" (Active), and "Inactivos" (Inactive). A table titled "Usuarios" lists three users:

Alumno	Mail	Institución	Estado	Registros	Opciones
Bruno, Ricardo	ricardobruno_89@hotmail.com	UNT, 4	Valido	1	<input type="button" value="Ver"/>
Gomez, Kevin	masterk63@gmail.com	,	Valido	0	<input type="button" value="Ver"/>
Gomez Veliz, Kevin Shionen	masterk631@gmail.com	UNT, 8	Valido	17	<input type="button" value="Ver"/>

On the right, a detailed user profile for "Bruno, Ricardo" is displayed. It includes a profile picture, the username "rickybruno", a registration count of "4", and a date of birth of "20/12/1989". There's also a toggle switch for "Permisos de Administrador" (Administrator permissions) which is turned on. Below this are tabs for "INFORMACION" (Information) and "REGISTROS". Under "INFORMACION", fields show "Nombre" as "Ricardo", "Apellido" as "Bruno", "Residencia" as "San miguel de tucuman", "Institución" as "UNT", and "Grado" as "4". Buttons for "Baja" (Delete) and "Editar" (Edit) are also present.

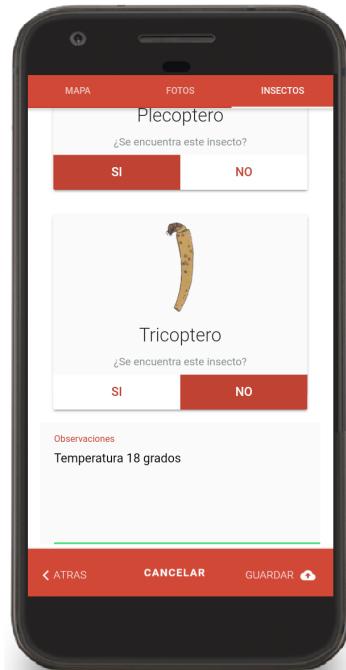
Figura 12: Visualización rápida del perfil de usuario en el sector derecho de la pantalla

Paso 3



(a) Captura 1/3

(b) Captura 2/3



(c) Captura 3/3

Figura 4: Formulario para selección de insectos encontrados

Ver registro



Registro N° 21

Creado por [rickybruno](#) 05/09/18

Observaciones
Temperatura del agua: 18 grados



Ubicacion
San Miguel de Tucumán, Tucumán, Argentina

Coordenadas
Latitud: -26.8385
Longitud: -65.2092

(a) Captura 1/3



Indice

1

Insectos encontrados:

Elmido	Patudo	Plecoptero	Tricóptero
NO	NO	SI	NO

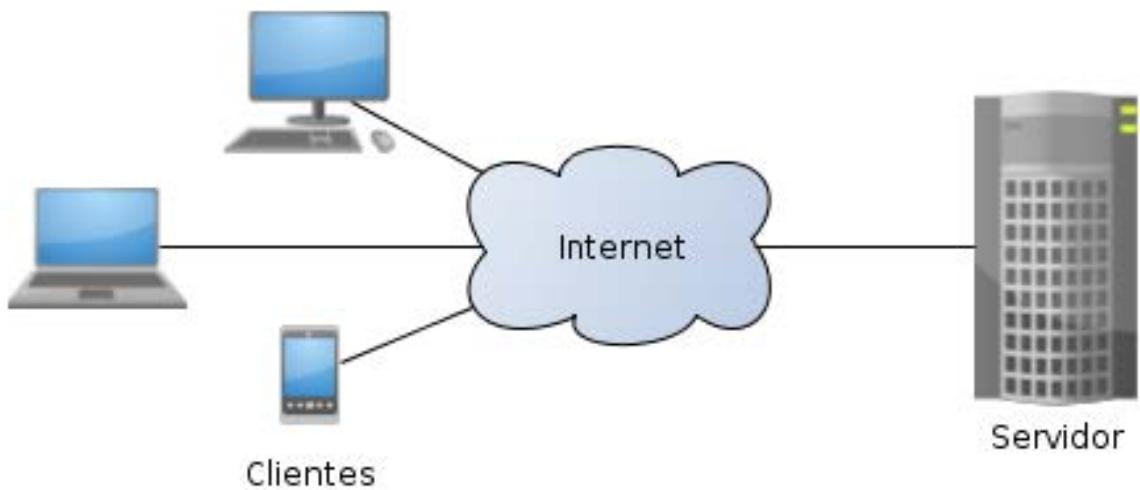
(b) Captura 2/3

(c) Captura 3/3

Figura 8: Formulario para selección de insectos encontrados

Disciplina de Implementación

Diagrama conceptual estructura Cliente/Servidor/Internet



Arquitectura de la aplicación

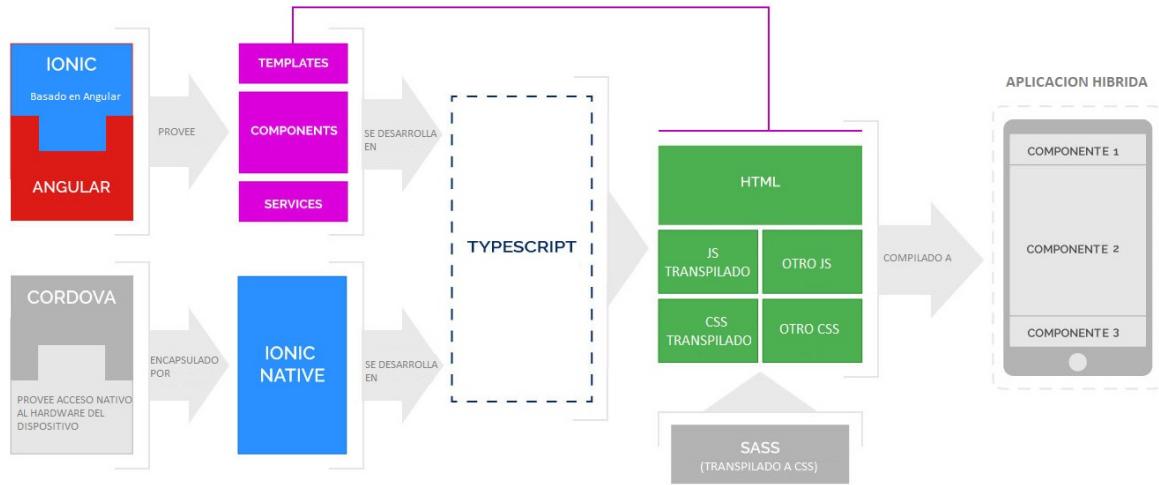


Figura 13: Interfaz de Usuario

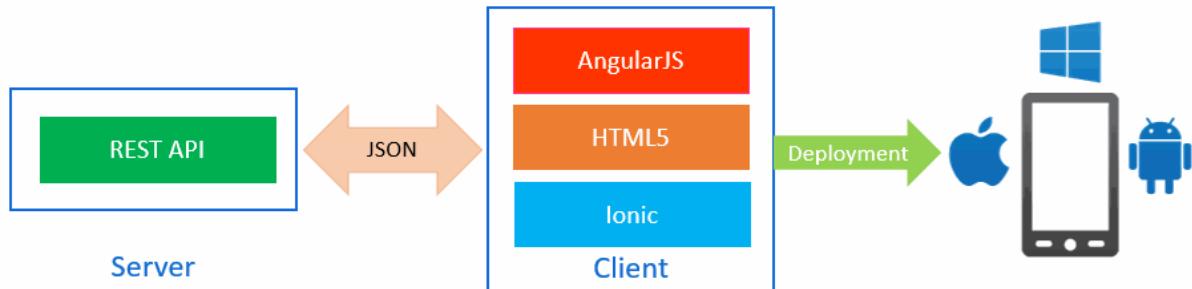


Figura 14: Diagrama de Despliegue

Modelo-Vista-Controlador

Modelo-Vista-Controlador (en adelante MVC) es un patrón de diseño que permite separar la capa de datos y lógica de negocios (modelos y controladores), de la interfaz de usuario (vista).

Componentes del patrón

- **Modelo:** Es la representación de la información con la cual el sistema opera. Se puede pensar en un modelo como una tabla en el sistema gestor de bases de datos. Si bien este enfoque resulta sumamente útil para trabajar con el patrón MVC, el modelo también requiere de funcionalidades que faciliten la interacción con los demás controladores, vistas y modelos del sistema.
- **Vista:** Es la representación gráfica de los datos. Provee la interfaz mediante la cual los usuarios realizan la entrada de los datos y observan el resultado del procesamiento de dichos datos.
- **Controlador:** Sirve como nexo entre las vistas y los modelos. El controlador responde usualmente a eventos que son generados por el usuario en su interacción con la vista. La vista brinda al usuario un mecanismo para comunicarse con un controlador, y es este ultimo el encargado de obtener los datos ingresados por el usuario, pasarlos al modelo adecuado para su procesamiento, recibir la salida del modelo y enviar dicha salida a la vista para su visualización.

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, CSS y JavaScript es posible aplicar el conocimiento en el desarrollo de aplicaciones web desarrolladas para navegadores en equipos de escritorio

Apache Cordova

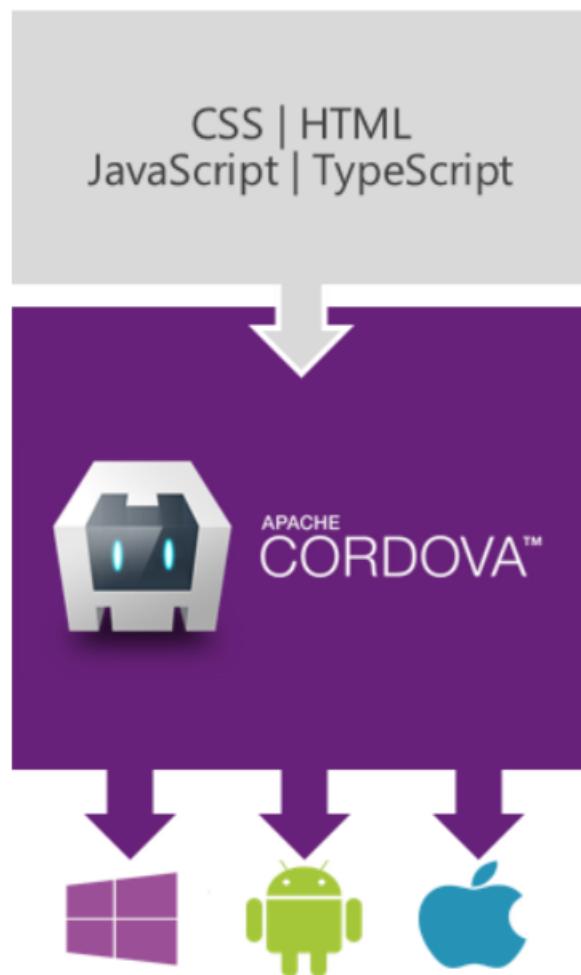


Figura 15: Arquitectura de Apache Cordova

Apache Cordova es un Framework (entorno de trabajo) de código abierto que actúa como un intermediario entre las aplicaciones web y los dispositivos móviles. Permite crear aplicaciones móviles listas para instalar utilizando tecnología web: JavaScript, HTML5 y CSS3.

Las aplicaciones resultantes no son totalmente nativas, ni puramente basado en la web. La desventaja de que una aplicación sea totalmente nativa es que solo se podrá utilizar para la plataforma para la que fue realizada, es decir si se hace una aplicación para Android luego no se podrá reutilizar el código para hacer la misma aplicación para iOS.

Con Apache Cordova se puede reutilizar el código de una aplicación para crear el paquete instalador de cualquiera de las 7 plataformas móviles soportadas: iOS, Android, Blackberry, Windows Phone, WebOS de Palm, Symbian, Bada, entre otras.

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

Ventajas de Apache Cordova

- Soporte multiplataforma
- 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 Apache Cordova

- Normalmente las aplicaciones realizadas con Apache Cordova 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. Diferentes 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 HTTPS sobre TCP/IP (Internet Protocol Suite) en el puerto 443. 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/IP salvo el 443, 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. Los sistemas que siguen los principios REST se llaman con frecuencia RESTful.

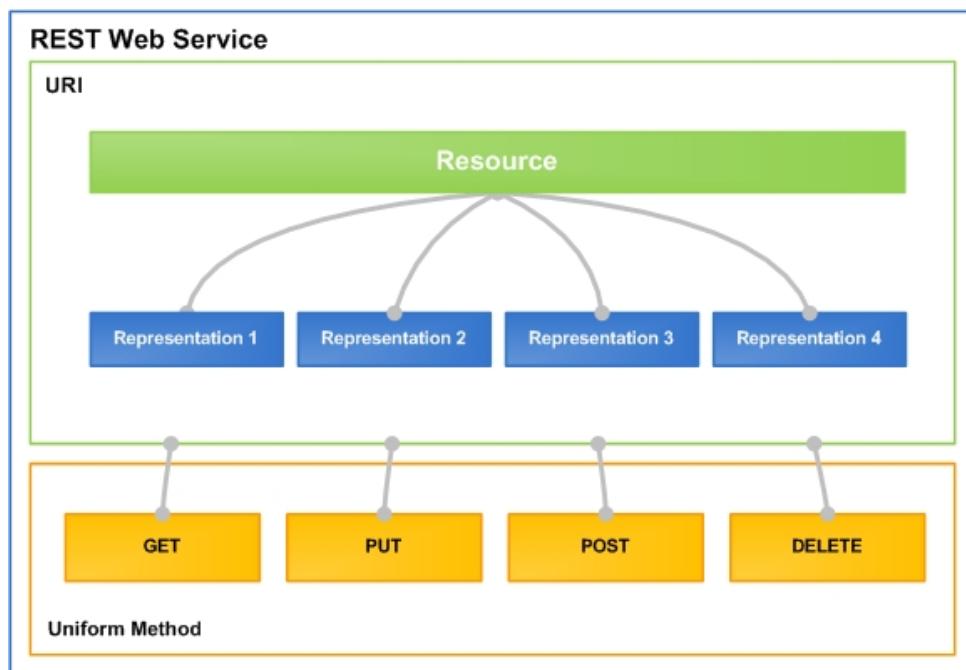


Figura 16: Web Services REST

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

- *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 re escritura 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 URI (Identificador de recursos uniforme).

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).

Express.js

Express es una infraestructura de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles. Con miles de métodos de programa de utilidad HTTP y middleware a su disposición, la creación de una API sólida es rápida y sencilla. Express proporciona una delgada capa de características de aplicación web básicas, que no ocultan las características de Node.js.

Herramientas de desarrollo

- *Visual Code:* Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows , Linux y Mac OS. Incluye soporte para la depuración, control integrado de Git. Es gratuito y de código abierto.
- *Draw.io:* Es un stack de tecnología de código abierto para crear aplicaciones de diagramación, entre ellas UML Permite dibujar todos los tipos de diagramas de

clases, código inverso, generar código desde diagramas y generar documentación.

- *LATEX*: Es un sistema de composición de textos, orientado especialmente a la creación de libros, documentos científicos y técnicos que contengan fórmulas matemáticas.
LATEX 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 *LATEX* es comparable a la de una editorial científica de primera línea.
- *GitHub*: Es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de computadora.

Persistencia de clases

Modelo físico del sistema

En esta actividad se define la estructura física de los datos utilizada en el sistema. Este esquema contempla las maneras de acceder físicamente a los datos, por medio de claves, índices, etc., para cumplir todos los requisitos. En esta fase se tienen que tener en cuenta aspectos de funcionalidad y rendimiento en el funcionamiento de la aplicación. Es decir, se modificará el diseño lógico de datos, para convertirlo en un modelo físico, que contemple todas estos aspectos de funcionalidad y rendimiento. Se comentarán de manera individualizada el porque de cada una de las modificaciones realizadas en el modelo físico, justificando su ejecución.

Modelo relacional. Vista de la base de datos

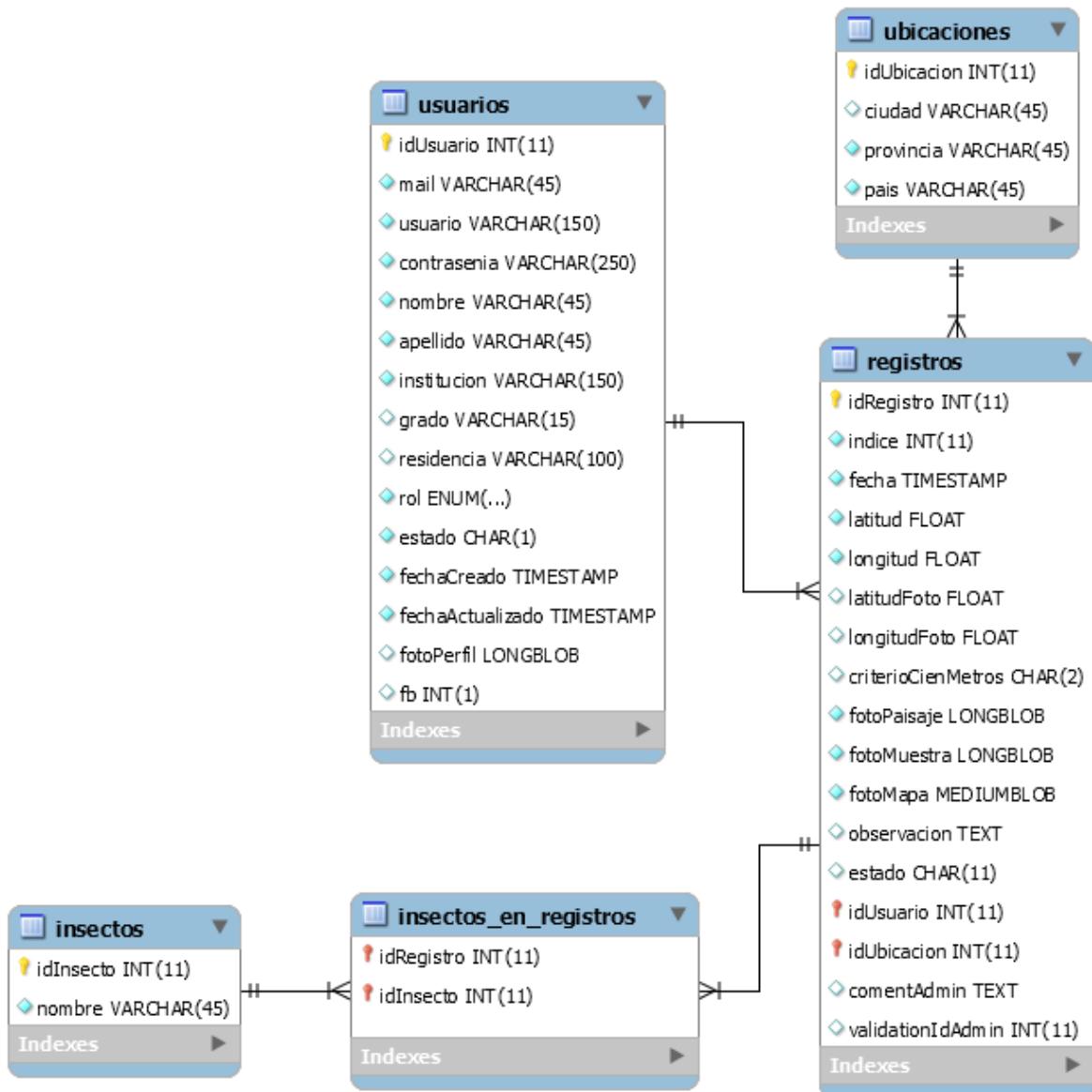


Figura 17: Modelo completo

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.

Pruebas realizadas: Para el mantenimiento del código, se implementaron test unitarios por módulos, para asegurarse que el código este siempre funcionando correctamente. Algunos de los módulos con test unitarios son: formulario para alta usuario, formulario para nuevos registros, entre otros. Todos estos test son usados para controlar la integridad de los datos ingresados.

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 realizadas: Al finalizar los módulos, se realizaron los test de caja negra correspondientes, validando así que la entradas de datos genere la salida esperada. Por ejemplo, al crear un registro, si no se captura la foto de la muestra, el usuario recibe el correspondiente mensaje de error. Esta acción se realiza para los requisitos obligatorios de un registro, hasta comprobar su validez, para luego ser guardado en la base de datos.

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.

Pruebas realizadas: Para las pruebas de estrés se sometió a la base de datos no relacional (couchDB) a la creación de 100 registros prototipo, donde obtuvieron los siguientes resultados no satisfactorios:

- Demora en la sincronización entre las BD.
- Falla en la sincronización.
- La BD crecía exponencialmente su tamaño debido al versionado nativo de los documentos al actualizarlos.

Por estos motivos, se rediseño el sistema para trabajar con una BD Relacional (MySql) realizando los siguientes cambios:

- Se modifco el protocolo de comunicación entre el sistema y el servidor
- Se creo un protocolo de sincronización automática que se ajusta a los requisitos del sistema
- Se implemento web socket para ver los cambios de la sincronización en tiempo real.

Pruebas realizadas con el rediseño:

Se realizaron las mismas pruebas, pero esta vez, con resultados satisfactorios para todos los puntos enunciados anteriormente.

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 subfunción, 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. Probando a mano, todos los casos escenarios posibles, como por ejemplo:

- Creación de registros.
- Iniciar sesión
- Comprobar el estado del mapa general, con los registros validados.
- Acceder al perfil

Todos estos casos fueron probados de manera online y offline. En funciones como ver el mapa general, que requieren conexión a internet y nos encontramos offline, se muestra un mensaje de alerta y no se permite el acceso a dicha función. Por otro lado, funciones que no requieren conexión a internet, de manera offline, pueden ser accedidas sin problemas

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 acabo en un entorno controlado.

Conclusiones de estas pruebas: Tras esta prueba, nos dimos cuenta de algunos errores y posibles mejoras, no solo de funcionamiento, sino también de la interfaz de usuario, ya que no era muy amigable. Para esta prueba se realizó una salida de campo real con el cliente (docente Daniel Dos Santos), en la cual pudimos apreciar que nuestro sistema cumplió correctamente con los requisitos planteados. Esta prueba se realizó con un smartphone de gama baja, y aun así, obtuvimos fotos legibles, coordenadas del GPS correctas, fluidez en la interfaz de usuario y un buen *performance* del dispositivo móvil en lo se refiere a procesamiento, creación y guardado de los registros.

Fotos de la salida de campo:



Figura 18: Foto paisaje de la muestra que sirve para futuro reconocimiento



Figura 19: Red de pie utilizada para la obtención de la muestra



Figura 20: Proceso de validación de la muestra



Figura 21: Proceso de identificación y extracción de los insectos objetivos



Figura 22: Foto real del insecto *Plecoptero*



Figura 23: Foto real del insecto *Patudo*

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.

Pruebas realizadas:

No se realizaron las pruebas betas, sin embargo se programó una para finales de octubre del 2018, con la participación de los alumnos de escuelas rurales (usuario final del sistema).

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 las cuales no estábamos familiarizados.

En cuanto a la dificultad, consideramos que fue media/alta debido a varias razones, una de ellas fue acceder al hardware del dispositivo mediante herramientas “web” requirió mucha investigación, pruebas y corrección de mínimos detalles para facilitar la interacción del usuario final con la aplicación. Otra de las razones fue implementar CouchDB (Base de datos no relacional) por ser robusta en lo que se refiere a replicaciones, dado que se amoldaba a nuestro requisito de generar registros de manera offline y al detectar conexión a internet, subirlos automáticamente. Pero esto nos generó un gran problema, dicha base de datos realizaba versionados, por lo que para cada actualización y replicación, la misma crecía exponencialmente su tamaño en disco, lo que iba a ser un problema en un futuro por la cantidad de registros que el sistema gestionaría. Consideramos que tomamos una decisión errónea al seleccionar e implementar esa base de datos en el sistema, solo habiendo investigado y leído la documentación oficial, de manera parcial y no en su totalidad. Este error nos costó bastante en lo que a tiempo se refiere, ya que tuvimos que rediseñar la lógica de la aplicación para poder implementar una base de datos relacional (MySQL) y esto implicó crear un protocolo de auto-sincronización, lo que generó más trabajo y más tiempo de desarrollo (aproximadamente 18 meses).

En las pruebas alpha realizadas, pudimos observar una buena respuesta al uso de la

aplicación de los docentes de la Facultad de Ciencias Naturales e Instituto Miguel Lillo, sin embargo, se detectaron algunas fallas y falta de funcionalidades. Esto genero una lista con ítems a corregir e implementar, lo que genero aun mas tiempo de desarrollo para la puesta en producción del sistema en su versión final.

Con la futura puesta en funcionamiento de este proyecto podremos generar un producto que da valor agregado a la metodología de trabajo existente de los docentes de la Facultad de Ciencias Naturales e Instituto Miguel Lillo en conjunto con alumnos de escuelas rurales, que no poseen automatismo en dicha metodología, cumpliendo así, con los objetivos planteados inicialmente, obteniendo un producto seguro, escalable, de fácil mantenimiento y simple de usar.

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.

Algunas de nuestras ideas para una actualización y agregación de funciones al sistema seria incorporar inteligencia artificial para el reconocimiento automático de imágenes al sacar fotos de los insectos. Ademas se podría desarrollar un termómetro con una conexión mediante Bluetooth al dispositivo, para medir la temperatura del agua al momento de realizar un registro, ya que este parámetro es importante para el análisis futuro de la biodiversidad.

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.. 85

Framework (entorno de trabajo)

Es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar. En el desarrollo de software, un entorno de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.. 86

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.. 85

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.. 25

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.. 85

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.. 90

Mac OS

Mac OS (del inglés Macintosh Operating System, en español Sistema Operativo de Macintosh) es el nombre del sistema operativo creado por Apple para su línea de computadoras Macintosh, también aplicado retroactivamente a las versiones anteriores a System 7.6, y que apareció por primera vez en System 7.5.1. Es conocido por haber sido uno de los primeros sistemas dirigidos al gran público en contar con una interfaz gráfica compuesta por la interacción del mouse con ventanas, iconos y menús.. 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.. 88

UML

El lenguaje unificado de modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Object Management Group (OMG). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un “plano” del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.. 90

URI (Identificador de recursos uniforme)

Es una cadena de caracteres que identifica los recursos de una red de forma única. La diferencia respecto a un localizador de recursos uniforme (URL) es que estos últimos hacen referencia a recursos que, de forma general, pueden variar en el tiempo. Normalmente estos recursos son accesibles en una red o sistema. Los URI pueden ser localizador de recursos uniforme (URL), uniform resource name (URN), o ambos.. 90

Bibliografía

- [1] Craig Larman. Applying UML and Patterns second edition.
- [2] Documentación oficial de Ionic.(<https://ionicframework.com/docs/>)
- [3] Documentación oficial de Apache Cordova.(<https://cordova.apache.org/docs/en/latest/>)
- [4] Documentación oficial de Express.js.(<http://expressjs.com/>)
- [5] Documentación oficial de Angular.(<https://angular.io/docs>)
- [6] Roger S. Pressman. Software Engineering sixth edition.
- [7] Wikipedia en inglés (<http://en.wikipedia.org/>).
- [8] Wikipedia en castellano (<http://es.wikipedia.org/>).
- [9] Wikilibros: Manual de L^AT_EX.
- [10] Gabriel Valiente Feruglio. Composición de textos científicos con L^AT_EX. 1999.