Diseño e implementación de una aplicación web para el control de acceso a un autobús escolar basado en IoT

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor**: Ignacio José Serra Almenar

**Tutor**: Lenin G. Lemus Zúñiga

2016/2017

Resumen

El objetivo de este proyecto es la creación de una aplicación web que permite gestionar la entrada y la salida de los alumnos de un colegio a un autobús.

La aplicación incluye como gestión, la capacidad de creación, modificación, eliminación y listado de Autobuses, conductores, monitores, rutas, alumnos, cursos, parientes y paradas, así como la opción de enviar mails informativos y automáticos a los parientes si en una determinada ruta ha ocurrido un atasco

Además, mediante el protocolo MQTT (*Message Queue Telemetry Transport*), se ofrece al usuario la posición exacta del autobús en cualquier dispositivo con acceso a Internet y GPS. Con este servicio se espera avisar a los parientes de los alumnos de la llegada de sus hijos antes de que ésta tenga lugar.

Se destaca que con la librería Semantic*-*ui, la interfaz de la aplicación web se adapta a móviles y tabletas.

**Palabras clave:** Aplicación Web, Internet of Things, GPS, JavaScript/jQuery, NodeJS, MySQL, AJAX, Jade, MQTT, Semantic-ui

Abstract

The objective of this project is the creation of a web application that allows to manage the entry and exit of students from a school to a bus.

The application includes the ability to create, modify, delete and list buses, drivers, monitors, routes, students, courses, relatives and stops, as well as the option to send informative and automatic emails to relatives if a Route has a traffic jam.

Furthermore, through the MQTT protocol, the user is offered the exact position of the bus on any device with Internet access and GPS.

It should be noted that with the library Semantic-ui, the web application interface fits mobile and tablets.

**Keywords :** Application Web, Internet of Things, GPS, JavaScript/JQuery, NodeJS, MySQL, AJAX, Jade, MQTT, Semantic-ui.

Tabla de contenidos

[1. Introducción 8](#_Toc490409576)

[1.1 Objetivos 9](#_Toc490409577)

[1.2 Contenido de la memoria 10](#_Toc490409578)

[2. Búsqueda bibliográfica 10](#_Toc490409579)

[3. Análisis de requerimientos 13](#_Toc490409580)

[3.1 Introducción 13](#_Toc490409581)

[3.1.1 Propósito 13](#_Toc490409582)

[3.1.2 Ámbito del Sistema 13](#_Toc490409583)

[3.1.3 Definiciones, Acrónimos y Abreviaturas 13](#_Toc490409584)

[3.1.4 Referencias 13](#_Toc490409585)

[3.1.5 Visión General del Documento 13](#_Toc490409586)

[3.2 Descripción General 13](#_Toc490409587)

[3.2.1 Perspectiva del Producto 13](#_Toc490409588)

[3.2.2 Funciones del Producto 13](#_Toc490409589)

[3.2.3 Características de los Usuarios 13](#_Toc490409590)

[3.2.4 Restricciones 13](#_Toc490409591)

[3.2.5 Suposiciones y Dependencias 13](#_Toc490409592)

[3.2.6 Requisitos Futuros 13](#_Toc490409593)

[3.3 Requisitos Específicos 13](#_Toc490409594)

[3.3.1 Interfaces Externas 13](#_Toc490409595)

[3.3.2 Funciones 13](#_Toc490409596)

[3.3.3 Requisitos de Rendimiento 13](#_Toc490409597)

[3.3.4 Restricciones de Diseño 13](#_Toc490409598)

[3.3.5 Atributos del Ssitema 13](#_Toc490409599)

[3.3.6 Otros Requisitos 13](#_Toc490409600)

[4. Diseño de la aplicación 14](#_Toc490409601)

[4.1 E-R (si hay BBDD) 14](#_Toc490409602)

[4.2 Casos de uso 14](#_Toc490409603)

[5. Implementación 14](#_Toc490409604)

[6. Validación y pruebas 14](#_Toc490409605)

[7. Conclusiones 14](#_Toc490409606)

[8. Bibliografía 15](#_Toc490409607)

Índice de figuras

[Ilustración 1: App Tu Ruta Escolar - Selección de Ruta 10](#_Toc490249725)

[Ilustración 2: App Tu Ruta Escolar - GPS 10](#_Toc490249726)

[Ilustración 3: Traceus App – Gestión 11](#_Toc490249727)

[Ilustración 4: Traceus App - GPS 11](file:///C:\Users\Usuario\Desktop\TFG%20AUTOBUS\aplicacionTFG\Memoria\TFG.docx#_Toc490249728)

# Introducción

La plataforma descrita en este trabajo pretende solventar la problemática existente a la hora de efectuar la recogida de un alumno que usa transporte escolar.

En primer lugar, tenemos un colegio con alumnos divididos en diferentes grupos. Los alumnos están inscritos por curso escolar. De forma que vamos a tener alumnos de infantil1, infantil2, primero, segundo, tercero, cuarto, quinto y sexto.

Cada grupo irá destinado a un autobús, el cual tendrá su propia línea y su propio horario.

Inicialmente se van a utilizar cinco líneas en este problema con opción a poder ser ampliadas.

Cada línea va a tener asignada un conjunto de paradas, en las cuales se indica posición, tiempo estimado de llegada y tiempo estimado de partida.

El autobús dispondrá de un monitor, el cual va a llevar una lista electrónica de alumnos pertenecientes a su grupo.

Cuando los alumnos se dirijan al autobús, el monitor estará presente para controlar la cantidad de alumnos que hay.

Cada vez que éstos vayan subiendo al autobús, ficharán la entrada en la lista electrónica del monitor para aparecer como registrados en la aplicación.

De esta manera, en sus datos registrados aparecerá la entrada del fichaje, por lo que utilizando la aplicación web los parientes del alumno sabrán que está dentro del autobús. También podrán comprobar la ruta que seguirá y las paradas establecidas.

En cada parada, los alumnos que bajen ficharán la salida. Al realizar este último fichaje, en la pantalla aparecerán las fotos del tutor o tutores, los cuales serán parientes registrados para recoger al alumno. El monitor se encargará de controlar que el pariente que recoge al alumno en la parada se corresponde con alguno de los tutores que aparecen en la pantalla.

Si se diera el caso de que no ha llegado el pariente que debe de recoger al alumno, el autobús seguirá su ruta con el alumno dentro. Si el pariente no llega a tiempo, mediante la aplicación web en su móvil puede comprobar la ruta del autobús del alumno y adelantarse al autobús en cualquiera de las siguientes paradas.

La persona que recoge al estudiante, debe de pasar una tarjeta que muestre la ficha del alumno y en caso de que sea el niño que se baja el monitor pulsará un botón para almacenar que el estudiante se dejó a cargo de la persona tutora, así como la fecha.

La última parada de todos los autobuses será el colegio.

## Objetivos

Los objetivos del proyecto son los siguientes:

* Ofrecer un servicio que permita gestionar la creación, modificación y eliminación de autobuses, conductores, monitores, alumnos, parientes de alumnos, paradas, rutas, y cursos escolares.
* Permitir a los parientes conocer el estado en el que se encuentra su hijo mediante la aplicación web o a través de notificaciones *push* en grupos de Telegram.
* Capacidad de avisar por correo electrónico a los padres de los alumnos si una ruta ha sufrido un atasco y va a retrasar la llegada del alumno.

Con estos objetivos se pretende que el colegio encargado gestione su servicio de autobuses escolares ofreciendo a los parientes información constante y automatizada del estado de los alumnos.

## Contenido de la memoria

La memoria del proyecto va a contener las diferentes fases del desarrollo que se han seguido para la creación de la aplicación web:

* **Búsqueda bibliográfica**: localización de aplicaciones parecidas para estudiar y analizar su estructura.
* **Análisis de requerimientos**: requisitos necesarios de la aplicación web, distinguiendo entre requisitos funcionales y no funcionales.
* **Diseño de la aplicación**: casos de uso de la aplicación, diagrama de clases, diagrama de la base de datos, prototipo de las vistas y versión final.
* **Implementación**: lenguaje de programación y código escrito para la realización de la aplicación web.
* **Validación y pruebas**: pruebas realizadas y comprobaciones de la aplicación web.
* **Conclusiones**: funcionalidades que he hecho, que no he podido y que podría hacer en un futuro.
* **Bibliografía**: libros y páginas utilizadas para el estudio y realización del proyecto.

# Búsqueda bibliográfica

El primer paso para el desarrollo del proyecto es la búsqueda de aplicaciones ya existentes y parecidas, con el fin de poder conocer su estructura y lo que ofrecen.

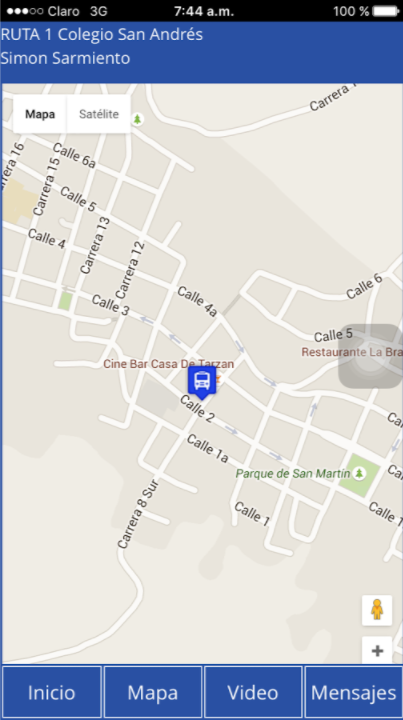
**Tu Ruta Escolar**



Se observa que utiliza un diseño minimalista y *responsive*. Una vez registrados y con la sesión iniciada, la primera vista disponible nos indica que tenemos que vincular la ruta donde se encuentra el alumno el cual queremos comprobar el estado.

Una mejora podría ser la vinculación por alumno, ya que de este modo el pariente no necesitaría aprenderse a qué ruta está destinado su hijo.

Ilustración 1: App Tu Ruta Escolar - Selección de Ruta



Una vez vinculada la ruta, nos muestra la posición del autobús por GPS en vista de mapa o de satélite y la posibilidad de ver vídeo por *streaming* del autobús y mandar mensajes de texto a los encargados del éste.

En definitiva, es una aplicación de constante monitorización para los parientes de los alumnos.

Ilustración 2: App Tu Ruta Escolar – GPS

**Traceus**

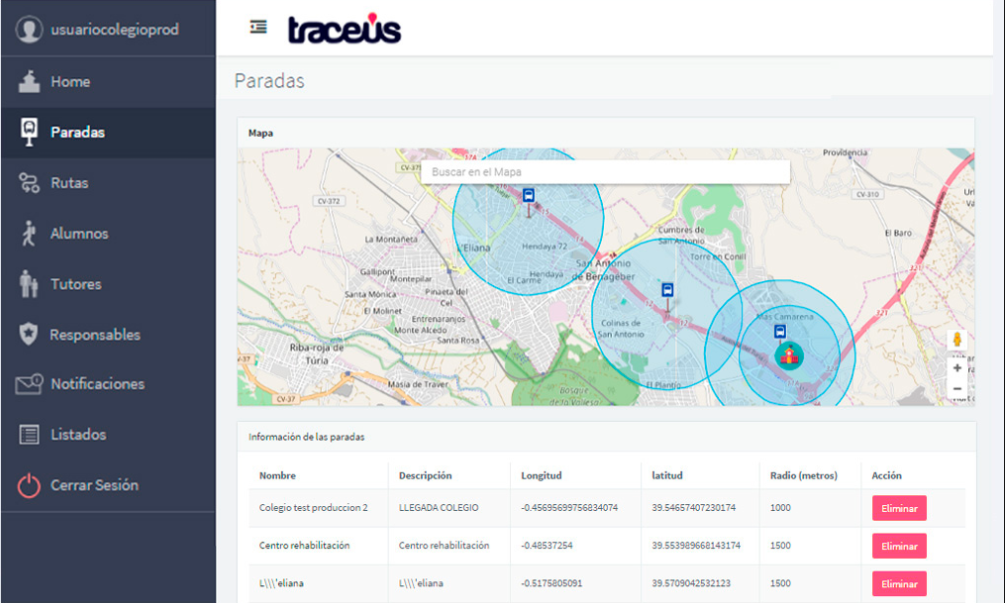


Ilustración 3: Traceus App – Gestión

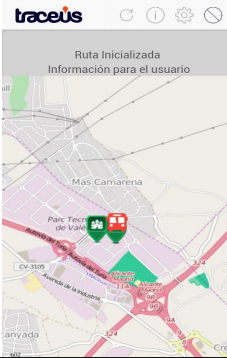


Ilustración : Traceus App - GPS

Traceus es una aplicación que permite gestionar un servicio escolar mediante funciones CRUD[[1]](#footnote-1) y además proporcionar a los parientes notificaciones de la posición de sus hijos. Tiene un diseño *responsive* y específico para móvil y web.

De las dos aplicaciones, la que más se va a asemejar al proyecto es la de Traceus, ya que va a tener un servicio de administración parecido, con la inclusión de un sistema de guardado de imágenes, que permitirá reconocer al pariente que recoja a su hijo, y un envío de notificaciones de estado a los padres mediante la aplicación de Telegram.

# Análisis de requerimientos

## Introducción

### Propósito

La especificación de requisitos tiene como finalidad conocer que se espera obtener de la aplicación, definiendo de manera clara y exacta las funcionalidades y restricciones que presentará la aplicación que se quiere desarrollar.

### Ámbito del Sistema

La aplicación web a desarrollar, va a ser una plataforma que permite la gestión de autobuses escolares a encargados de un colegio, y la consulta por parte de parientes, del estado en el que se encuentran sus hijos.

### Definiciones, Acrónimos y Abreviaturas

**Alumno:** Todas las personas matriculadas en el colegio.

**Monitor:** Persona encargada de la entrada y salida de alumnos matriculados al autobús asignado.

**Conductor:** Persona que conduce el autobús designado y se encarga de efectuar las paradas pertinentes para recoger o dejar a los alumnos.

**Parada:** Lugar donde los alumnos matriculados esperan la llegada de su autobús designado o puntos de destino donde se reúnen parientes para recoger a sus hijos.

**Centro:** Lugar donde se imparten los diferentes niveles de educación.

**Rutas:** Conjunto de paradas que sigue el autobús con el fin de llevar a los alumnos a su destino.

**Navegador:** Visualizador de páginas web a través de internet que permiten al usuario realizar consultas o gestiones.

**Servidor web:** Programa informático que procesa una aplicación desde el lado del servidor, realizando conexiones con el cliente y generando respuestas.

**Servidor Apache:** Servidor HTTP de código abierto para plataformas Unix, Linux, Windows y Macintosh.

**MySQL:** Sistema de gestión de bases de datos relacional.

**JavaScript:** Lenguaje de programación interpretado, débilmente tipado y dinámico.

**NodeJS:** Entorno en tiempo de ejecución multiplataforma para JavaScript, de código abierto

**CSS:** Hojas de estilo en cascada, lenguaje formal de ordenador usado para definir la presentación de un documento estructurado en HTML o XML.

***Framework:***Estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado. Facilitan el desarrollo y evitan los detalles de bajo nivel.

**Semantic-UI:** Framework utilizado en aplicaciones y páginas web que ayuda a crear entornos *responsive* y estilizados.

**Jade:** Lenguaje de plantillas centrado en la escritura de código rápida de HTML

**HTML:** Lenguaje de marcas de hipertexto para la elaboración de páginas web.

**AJAX:** *Asynchronous JavaScript And XML*, es una técnica de desarrollo web para crear aplicaciones interactivas.

### Referencias

* IEEE Std. 830-1998 Guía del IEEE para la Especificación de Requisitos Software.
* Ejemplos IEEE

### Visión General del Documento

El documento se desglosa en dos secciones. La Descripción Global, que va a tratar de aquellos factores que afectan al producto a construir, dando a conocer sus funciones principales, datos requeridos y restricciones que afecten a su desarrollo. Los Requisitos Específicos, cuyo contenido va a ofrecer un nivel de detalle suficiente como para permitir crear un sistema que satisfaga los requisitos.

## Descripción General

### Perspectiva del Producto

La aplicación web es totalmente independiente.

Las funciones varían dependiendo del tipo de usuario.

Los parientes van a poder realizar consultas, y los administradores gestionar y consultar

La plataforma podrá ser visitada por cualquier usuario independientemente del navegador que utilice, sistema operativo o dispositivo mientras esté disponible la conexión a Internet.

### Funciones del Producto

Las funciones que va a satisfacer la aplicación web son las siguientes.

**Usuario**

* Ingreso de DNI y correo electrónico.
* Consulta de estado de alumnos a su cargo.
* Consulta de mapa con coordenadas de las paradas.

**Administrador**

* Creación, modificación y eliminación de autobuses.
* Creación, modificación y eliminación de conductores.
* Creación, modificación y eliminación de monitores.
* Creación, modificación y eliminación de rutas.
* Creación, modificación y eliminación de rutas de ida.
* Creación, modificación y eliminación de rutas de vuelta.
* Creación, modificación y eliminación de paradas.
* Creación, modificación y eliminación de cursos escolares.
* Creación, modificación y eliminación de matriculados.
* Filtros en cada tabla por campos relevantes.
* Envío de correos electrónicos automáticos a los parientes que tienen un hijo cuya ruta ha sufrido un atasco.
* Envío de notificaciones por Telegram del estado de los alumnos a los parientes.

### Características de los Usuarios

Nuestra aplicación va a tener dos tipos de usuarios:

**Usuario:** aquellas personas que desean conocer el estado de sus hijos e ingresan su DNI y correo electrónico.

**Administradores:** son las personas encargadas de la gestión del autobús escolar, trabajadores del centro, como monitores del autobús, o personal de secretaría.

### Restricciones

* Todas las personas que quieran acceder a la aplicación web van a necesitar cualquier dispositivo con acceso a Internet.
* Solo el usuario administrador va a poder crear modificar y eliminar registros de la aplicación.
* El lenguaje de programación que se utilizará será NodeJS
* Se utilizarán plantillas Jade para la estructura de las vistas
* El *framework* para los estilos de las vistas será Semantic-UI.
* Se trabajará con bases de datos MySQL.

### Suposiciones y Dependencias

Correcto funcionamiento de la base de datos y del sistema donde se ejecute, así como la realización de copias de seguridad.

### Requisitos Futuros

Como futuras mejoras al sistema, se pueden tener en cuenta:

* Chat directo entre parientes, alumnos y monitores a través de la aplicación.
* Registro único por persona en la aplicación web.
* Avisos automáticos a los parientes cuando el alumno entra y sale del autobús.

## Requisitos Específicos

### Interfaces Externas

#### Interfaces de usuarios

La interfaz de usuario constará de una barra de navegación superior, la cual incluirá un menú lateral con las acciones disponibles y un contenedor principal donde estará la información relacionada con el menú seleccionado.

#### Interfaces hardware

Los usuarios tienen que disponer de un ordenador o un dispositivo que permita conexión a internet para poder acceder a la aplicación web.

#### Interfaces software

La aplicación está desarrollada en NodeJS, JQuery, Semantic-UI y con una base de datos basada en MySQL, por lo que de este modo puede funcionar sobre cualquier navegador y sistema operativo.

#### Interfaces de comunicaciones

La comunicación entre el cliente y el servidor consiste en una comunicación de petición y respuesta, mediante el protocolo HTTP, y enviadas entre cliente/servidor con el protocolo TCP/IP.

### Funciones

#### Usuario

* Búsqueda de alumno
  + Introducción: Búsqueda de alumnos que hacen uso del autobús
  + Entrada: DNI y correo electrónico del usuario
  + Proceso: Búsqueda de alumno matriculado que coincida con los datos del pariente introducido.
  + Salida: Tabla de estado del alumno.
* Listado de hijos:
  + Introducción: Listado de hijos del pariente que ha realizado la consulta
  + Entrada: Datos del pariente
  + Proceso: Se realiza una búsqueda con los registros coincidentes.
  + Salida: Se muestra el resultado en una tabla con los datos del alumno.

#### Administrador

* Crear autobuses
  + Introducción: Crear autobús nuevo.
  + Entrada: Datos del autobús, matrícula, ruta, capacidad, conductor y monitor.
  + Proceso: Añade un nuevo registro a la base de datos de la tabla autobús.
  + Salida: Confirmación de creación correcta del autobús.
* Listar autobuses
  + Introducción: Listado de autobuses disponibles en el centro.
  + Entrada: -
  + Proceso: Recupera de la base de datos todos los autobuses registrados.
  + Salida: Muestra una tabla con todos los autobuses y atributos respectivos.
* Modificar autobuses
  + Introducción: Modifica los datos del autobús.
  + Entrada: Matrícula, ruta, capacidad, conductor y monitor.
  + Proceso: Modifica el registro en la base de datos del autobús con los datos introducidos.
  + Salida: Retorno al listado de autobuses con el autobús modificado.
* Eliminar autobuses
  + Introducción: Elimina el autobús seleccionado.
  + Entrada: Se pasa la id por contexto al pulsar el botón “Eliminar” del autobús correspondiente.
  + Proceso: Borra el registro del autobús de la base de datos.
  + Salida: -
* Crear conductores
  + Introducción: Creación de conductor nuevo.
  + Entrada: Nombre, teléfono.
  + Proceso: Añade un nuevo registro a la base de datos de la tabla conductor.
  + Salida: Confirmación de conductor creado correctamente.
* Listar conductores
  + Introducción: Listado de conductores del centro.
  + Entrada: -
  + Proceso: Recupera de la base de datos todos los conductores registrados.
  + Salida: Muestra una tabla con todos los conductores y atributos respectivos.
* Modificar conductores
  + Introducción: Modifica los datos del conductor.
  + Entrada: Nombre, teléfono.
  + Proceso: Modifica el registro en la base de datos del conductor con los datos introducidos.
  + Salida: Retorno al listado de conductores con el conductor modificado.
* Eliminar conductores
  + Introducción: Elimina al conductor seleccionado.
  + Entrada: Se pasa la id por contexto al pulsar el botón “Eliminar” del conductor correspondiente.
  + Proceso: Borra el registro del conductor de la base de datos.
  + Salida: -
* Crear monitores
  + Introducción: Creación de monitor nuevo.
  + Entrada: Nombre, email, teléfono.
  + Proceso: Añade un nuevo registro a la base de datos de la tabla conductor.
  + Salida: Confirmación de monitor creado correctamente.
* Listar monitores
  + Introducción: Listado de monitores del centro.
  + Entrada: -
  + Proceso: Recupera de la base de datos todos los monitores registrados.
  + Salida: Muestra una tabla con todos los monitores y atributos respectivos.
* Modificar monitores
  + Introducción: Modifica los datos del monitor seleccionado.
  + Entrada: Nombre, email, teléfono.
  + Proceso: Modifica el registro en la base de datos del monitor con los datos introducidos.
  + Salida: Retorno al listado de monitores con el monitor modificado.
* Eliminar monitores
  + Introducción: Elimina al monitor seleccionado
  + Entrada: Se pasa la id por contexto al pulsar el botón “Eliminar” del monitor correspondiente.
  + Proceso: Borra el registro del monitor de la base de datos.
  + Salida: -
* Crear rutas
  + Introducción: Creación de ruta nueva.
  + Entrada: Nombre, paradas, ruta de ida, ruta de vuelta.
  + Proceso: Añade un nuevo registro a la base de datos de la tabla ruta.
  + Salida: Confirmación de ruta creada correctamente.
* Listar rutas
  + Introducción: Listado de rutas existentes realizadas por los autobuses.
  + Entrada: -
  + Proceso: Recupera de la base de datos todas las rutas registradas.
  + Salida: Muestra una tabla con todas las rutas y atributos respectivos.
* Modificar rutas
  + Introducción: Modifica los datos de la ruta seleccionada.
  + Entrada: Nombre, paradas, ruta de ida, ruta de vuelta.
  + Proceso: Modifica el registro en la base de datos de la ruta con los datos introducidos.
  + Salida: Retorno al listado de rutas con la ruta modificada.
* Eliminar rutas
  + Introducción: Elimina la ruta seleccionada
  + Entrada: Se pasa la id de la ruta por contexto al pulsar el botón “Eliminar” de la ruta correspondiente.
  + Proceso: Borra el registro de la ruta de la base de datos.
  + Salida: -
* Enviar correo
  + Introducción: Se envía un correo desde la ruta seleccionada para informar de la existencia de un atasco a los parientes que tengan hijos viajando en esa ruta.
  + Entrada: Se pasa la id de la ruta por contexto al enviar el correo.
  + Proceso: A partir de la id de la ruta, se buscan a los alumnos que estén viajando en ella y se envía un correo electrónico automatizado a los parientes.
  + Salida: Correo de salida avisando de atasco en la ruta en la que se encuentra el alumno.
* Crear rutas de ida
  + Introducción: Creación de rutas de ida.
  + Entrada: Nombre, paradas de ida.
  + Proceso: Añade un nuevo registro a la base de datos de la tabla rutaIda.
  + Salida: Confirmación de ruta de ida creada correctamente.
* Listar rutas de ida
  + Introducción: Listado de rutas de ida existentes.
  + Entrada: -
  + Proceso: Recupera de la base de datos todas las rutas de ida registradas.
  + Salida: Muestra una tabla con todas las rutas de ida y atributos respectivos.
* Modificar rutas de ida
  + Introducción: Modifica los datos de la ruta de ida seleccionada.
  + Entrada: Nombre, paradas ida.
  + Proceso: Modifica el registro en la base de datos de la ruta de ida con los datos introducidos.
  + Salida: Retorno al listado de rutas de ida con la ruta de ida modificada
* Eliminar rutas de ida
  + Introducción: Elimina la ruta de ida seleccionada
  + Entrada: Se pasa la id de la ruta de ida por contexto al pulsar el botón “Eliminar”.
  + Proceso: Borra el registro de la ruta de ida de la base de datos.
  + Salida: -
* Crear rutas de vuelta
  + Introducción: Creación de rutas de vuelta.
  + Entrada: Nombre, paradas de vuelta.
  + Proceso: Añade un nuevo registro a la base de datos de la tabla rutaVuelta.
  + Salida: Confirmación de ruta de vuelta creada correctamente.
* Listar rutas de vuelta
  + Introducción: Listado de rutas de vuelta existentes.
  + Entrada: -
  + Proceso: Recupera de la base de datos todas las rutas de vuelta registradas.
  + Salida: Muestra una tabla con todas las rutas de vuelta y atributos respectivos.
* Modificar rutas de vuelta
  + Introducción: Modifica los datos de la ruta de vuelta seleccionada.
  + Entrada: Nombre, paradas vuelta.
  + Proceso: Modifica el registro en la base de datos de la ruta de vuelta con los datos introducidos.
  + Salida: Retorno al listado de rutas de vuelta con la ruta de vuelta modificada
* Eliminar rutas de ida
  + Introducción: Elimina la ruta de vuelta seleccionada
  + Entrada: Se pasa la id de la ruta de vuelta por contexto al pulsar el botón “Eliminar”.
  + Proceso: Borra el registro de la ruta de vuelta de la base de datos.
  + Salida: -
* Crear matriculados
  + Introducción: Creación de alumnos matriculados.
  + Entrada: Nombre, nombre 2º, apellido paterno, apellido materno, DNI, ruta, curso escolar, estado, ruta ida, ruta vuelta.
  + Proceso: Añade un nuevo registro a la base de datos de la tabla matriculados.
  + Salida: Confirmación de alumno creado correctamente.
* Listar matriculados
  + Introducción: Listado de los alumnos matriculados en el centro.
  + Entrada: -
  + Proceso: Recupera de la base de datos todos los alumnos registrados.
  + Salida: Muestra una tabla con todos los alumnos matriculados y atributos respectivos.
* Modificar matriculados
  + Introducción: Modifica los datos de los alumnos.
  + Entrada: Nombre, nombre 2º, apellido paterno, apellido materno, DNI, ruta, curso escolar, estado, ruta ida, ruta vuelta.
  + Proceso: Modifica el registro en la base de datos del alumno con los datos introducidos.
  + Salida: Retorno al listado de matriculados con el alumno modificado.
* Eliminar matriculados
  + Introducción: Elimina al alumno seleccionado.
  + Entrada: Se pasa la id del alumno por contexto al pulsar el botón “Eliminar”.
  + Proceso: Borra el registro del alumno de la base de datos.
  + Salida: -
* Notificación de estado del matriculado.
  + Introducción: Envío de mensaje informativo a los parientes por Telegram del estado del matriculado.
  + Entrada: -
  + Proceso: Se envía una notificación *push* al móvil de los parientes teniendo en cuenta el estado guardado en la base de datos del matriculado.
  + Salida: Mensaje enviado por Telegram del estado del alumno.
* Crear cursos escolares
  + Introducción: Crea un curso escolar nuevo.
  + Entrada: Curso, clase.
  + Proceso: Añade un nuevo registro a la tabla curso escolar de la base de datos.
  + Salida: Confirmación de curso escolar creado correctamente.
* Listar cursos escolares
  + Introducción: Listado de cursos escolares del centro.
  + Entrada: -
  + Proceso: Recupera de la base de datos todos los cursos escolares registrados.
  + Salida: Muestra una tabla con todos los cursos escolares y atributos respectivos.
* Modificar cursos escolares
  + Introducción: Modifica el curso escolar seleccionado.
  + Entrada: Curso, clase.
  + Proceso: Modifica el registro en la tabla curso escolar de la base de datos con los datos introducidos.
  + Salida: Retorno a la lista de cursos escolares con el curso escolar modificado.
* Eliminar cursos escolares
  + Introducción: Elimina el curso escolar seleccionado.
  + Entrada: Se pasa la id del curso escolar al pulsar el botón “Eliminar”.
  + Proceso: Borra el registro de la base de datos.
  + Salida: -
* Crear parientes
  + Introducción: Crea un nuevo pariente.
  + Entrada: Nombre, foto, hijo/a, email, teléfono.
  + Proceso: Añade un nuevo registro a la tabla pariente de la base de datos.
  + Salida: Confirmación de pariente creado correctamente.
* Listar parientes
  + Introducción: Listado de parientes.
  + Entrada: -
  + Proceso: Recupera de la base de datos todos los parientes registrados.
  + Salida: Muestra una tabla con todos los parientes y atributos respectivos.
* Modificar parientes
  + Introducción: Modifica el pariente seleccionado.
  + Entrada: Nombre, foto, hijo/a, email, teléfono.
  + Proceso: Modifica el registro en la tabla pariente de la base de datos con los datos introducidos.
  + Salida: Retorno a la lista de parientes con el pariente modificado.
* Eliminar parientes
  + Introducción: Elimina al pariente seleccionado.
  + Entrada: Se pasa la id del pariente por contexto al pulsar el botón “Eliminar”.
  + Proceso: Borra el registro de la tabla pariente en la base de datos.
  + Salida: -
* Subir foto
  + Introducción: Al crear o modificar un pariente se subirá una foto, la cual se quedará guardada en una carpeta estática.
  + Entrada: Foto en formato PNG o JPG.
  + Proceso: Una vez elegida la foto, se va a guardar en la base de datos el nombre del fichero y en una carpeta estática el contenido de la imagen, para a continuación ser mostrada en el listado de parientes
  + Salida: Imagen seleccionada.
* Crear paradas
  + Introducción: Crea una parada nueva.
  + Entrada: Nombre, ruta, ruta de ida, ruta de vuelta, tiempo, longitud, latitud, hora ida de llegada, hora ida de salida, hora vuelta de llegada, hora vuelta de salida.
  + Proceso: Recupera de la base de datos todos los conductores registrados.
  + Salida: Confirmación de parada creada correctamente.
* Listar paradas
  + Introducción: Listado de paradas.
  + Entrada: -
  + Proceso: Recupera de la base de datos todas las paradas registradas.
  + Salida: Muestra una tabla con todas las paradas y atributos respectivos.
* Modificar paradas
  + Introducción: Modifica la parada seleccionada con nuevos datos.
  + Entrada: Nombre, ruta, ruta de ida, ruta de vuelta, tiempo, longitud, latitud, hora ida de llegada, hora ida de salida, hora vuelta de llegada, hora vuelta de salida.
  + Proceso: Modifica el registro en la tabla paradas de la base de datos con los datos introducidos.
  + Salida: Retorno a la lista de paradas con la parada modificada.
* Eliminar paradas
  + Introducción: Elimina la parada seleccionada.
  + Entrada: Se pasa la id de la parada por contexto al pulsar el botón “Eliminar”.
  + Proceso: Borra el registro de la tabla parada en la base de datos
  + Salida: -

### Requisitos de Rendimiento

La respuesta dada por la página web va a ser en tiempo real.

La respuesta de la base de datos va a ser de milisegundos, por lo que vía web no va a haber ningún retardo en la muestra y procesamiento de datos. Para que hubiera retardo, la base de datos debería de tener más de cien mil registros.

### Restricciones de Diseño

El diseño va a variar en función del dispositivo que se esté usando.

Se ha realizado un diseño general para cualquier tipo de situación.

### Atributos del Sistema

#### Integridad

Para poder guardar los datos, se ha utilizado una base de datos MySQL, permitiendo la persistencia de datos y con la posibilidad de realizar *backups[[2]](#footnote-2)*

#### Mantenimiento

El mantenimiento debe de ser llevado por el administrador de sistema, al cual se le facilitaría acceso a la máquina donde se encuentra instalada la aplicación.

#### Seguridad

Se dispone de un inicio de sesión para distinguir administradores de usuarios normales. De este modo solo el administrador podrá realizar la parte de gestión, y el usuario solo podrá consultar sus datos.

### Otros Requisitos

#### Base de datos

La aplicación hace uso de una base de datos MySQL, donde se va a almacenar toda la información de los autobuses, matriculados, rutas, paradas, conductores, monitores y parientes.

# Diseño de la aplicación

## E-R (si hay BBDD)

## Casos de uso

….

# Implementación

codigo

# Validación y pruebas

# Conclusiones

¿Qué iba a hacer?

¿Qué he hecho?

¿Qué no he hecho y por qué?

¿Qué más se podría hacer?

Conclusiones personales

# Bibliografía

Formato APA 6.

UTILIZAR MENDELEY

TIENES QUE LEER LIBROS y ARTICULOS

Apache - <https://es.wikipedia.org/wiki/Servidor_HTTP_Apache>

Javascript- <https://es.wikipedia.org/wiki/JavaScript>

NodeJS - <https://es.wikipedia.org/wiki/Node.js>

MySQL - <https://es.wikipedia.org/wiki/MySQL>

Servidor Web - <https://es.wikipedia.org/wiki/Servidor_web>

CSS - <https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada>

Framework - <http://www.alegsa.com.ar/Dic/framework.php>

Semantic UI - <https://semantic-ui.com/>

Jade - <http://learnjade.com/>

HTML - <https://es.wikipedia.org/wiki/HTML>

AJAX - <https://es.wikipedia.org/wiki/AJAX>

1. CRUD: (Create, Read, Update and Delete) Operaciones básicas que se realizan en una base de datos. [↑](#footnote-ref-1)
2. Backup: Proceso realizado cuando se quiere guardar una copia de la base de datos. [↑](#footnote-ref-2)