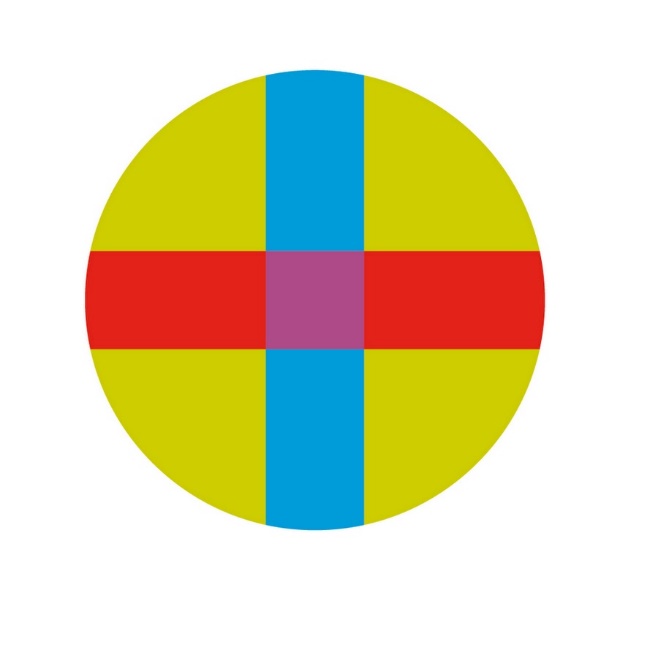
UNIVERSIDAD SAN PABLO - CEU

ESCUELA POLITÉCNICA SUPERIOR

GRADO EN INGENIERÍA DE SISTEMAS DE TELECOMUNICACIÓN



TRABAJO FIN DE GRADO

**TÍTULO DEL TFG**

**Título del TFG en inglés**

Autor: Nombre y apellidos del autor

Director: Nombre y apellidos del director

mes y año de la convocatoria

UNIVERSIDAD SAN PABLO-CEU



ESCUELA POLITÉCNICA SUPERIOR

División de Ingeniería

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Datos del alumno   |  | | --- | | NOMBRE: Jaime Arradi |   Datos del Trabajo   |  | | --- | | TÍTULO DEL PROYECTO: “Diseño e implementación de un servicio web RESTful para facilitar  las quedadas entre moteros |   Tribunal calificador   |  |  | | --- | --- | | PRESIDENTE: | FDO.: |  |  |  | | --- | --- | | SECRETARIO: | FDO.: |      |  |  | | --- | --- | | VOCAL: | FDO.: |  |  | | --- | | Reunido este tribunal el \_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_, acuerda otorgar al Trabajo Fin de Grado presentado por Don\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_la calificación de \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. | |

**Abstract**

resumen

*Actualmente, el mercado de la moto está en alza debido a cambios legislativos y de nuestro estilo de vida, lo que está provocando que surjan muchas oportunidades de negocio que deben ser exploradas. Por ejemplo, hoy en día, muchas personas optan por tener una moto como medio de transporte debido a la comodidad de aparcar en el centro de las ciudades (debido a las nuevas restricciones de emisiones), la posibilidad de evadirse de la densidad del tráfico o porque es un medio asociado a un mejor perfil medioambiental (puesto que emite poca contaminación y tiene un bajo consumo). No obstante, los usuarios de este tipo de transporte (conocidos como moteros) generalmente están asociados a un movimiento (el verdadero espíritu motero) que se basa en las reuniones junto con otros conductores para recorrer puertos de montaña o carreteras sinuosas y, así, probar los límites de la moto. Precisamente, éste que es el punto agradable de encuentro con otros moteros (denominado en la jerga “quedada”) y de disfrute del vehículo, presenta como inconveniente principal, el encontrar rutas que sean amenas, ajustadas a las habilidades del conductor, ajustadas a las características de la moto y, sobre todo, encontrar compañeros de actividad (que tengan disponibilidad temporal, que compartan los mismos gustos en cuanto a las rutas a seguir, etcétera). En este TFG, vamos a intentar paliar estos problemas ofreciendo una aplicación que se encargará de gestionar todos los aspectos relacionados con una quedada motera.*

INDex

[Capítulo 1. INTRODUCCIÓN 1](#_Toc72579462)

[1.1 Análisis 1](#_Toc72579463)

[1.1.1 Subsubsection 1](#_Toc72579464)

[1.2 References 2](#_Toc72579465)

[1.3 Figures 2](#_Toc72579466)

[1.4 Tables 3](#_Toc72579467)

[Capítulo 2. Estado del arte 4](#_Toc72579468)

[2.1 ¿Qué son los Servicios Web y cómo funcionan? 4](#_Toc72579469)

[2.2 Servicios web tradicionales SOAP 6](#_Toc72579470)

[2.3 Servicios web RESTful 6](#_Toc72579471)

[2.4 PLAY Framework 8](#_Toc72579472)

[Capítulo 3. DESCRIPCIÓN INFORMÁTICA 12](#_Toc72579473)

[3.1 Análisis 12](#_Toc72579474)

[3.1.1 Usuarios 16](#_Toc72579475)

[3.1.2 Rutas 18](#_Toc72579476)

[3.1.3 Quedada 19](#_Toc72579477)

[3.1.4 Posibles incorporaciones futuras 20](#_Toc72579478)

[3.2 Aálisis 22](#_Toc72579479)

[Capítulo 4. CONCLUSIONES Y TRABAJO FUTURO 24](#_Toc72579480)

[Capítulo 5. CONCLUSIONS 25](#_Toc72579481)

[Capítulo 6. REFERENCES 26](#_Toc72579482)

índice de ilustraciones

[Figure 1 All figures must have a number, which will increase consecutively throughout the document, and a caption. 2](#_Toc476493435)

TablE INDEX

[Table 1 All tables must have a number, which will increase consecutively through the document, and a caption. 3](#_Toc476494203)

# INTRODUCCIÓN

## Análisis

The main sections structure of this template is recommended, but not mandatory. The structure could be changed if not adequate. As a general rule, it usually helps when writing in a non-native language using short sentences. Use an automatic grammar checker to help you. Here there are some paragraphs with sample text.

Biomedical engineering is the application of engineering principles and design concepts to medicine and biology for healthcare purposes (e.g. diagnostic or therapeutic). This field seeks to close the gap between engineering and medicine, combining the design and problem solving skills of engineering with medical and biological sciences to advance health care treatment, including diagnosis, monitoring, and therapy.

Biomedical engineering has only recently emerged as its own study, compared to many other engineering fields. Such an evolution is common as a new field transitions from being an interdisciplinary specialization among already-established fields, to being considered a field in itself. Much of the work in biomedical engineering consists of research and development, spanning a broad array of subfields (see below). Prominent biomedical engineering applications include the development of biocompatible prostheses, various diagnostic and therapeutic medical devices ranging from clinical equipment to micro-implants, common imaging equipment such as MRIs and EEGs, regenerative tissue growth, pharmaceutical drugs and therapeutic biologicals.

### Subsubsection

Biomedical engineering achievements range from early devices, such as crutches, platform shoes, and wooden teeth to more modern equipment, including pacemakers, dialysis machines, diagnostic equipment, imaging technologies of every kind, and artificial organs, medical implants and advanced prosthetics.

* 1791: Luigi Galvani invented the frog galvanoscope.
* 1851: Hermann von Helmholtz invented the ophthalmoscope.
* 1881: Samuel von Basch invented the blood pressure meter (also known as sphygmomanometer).
* 1895: Conrad Roentgen (Germany) discovered the X-ray using gas discharged tubes.

## References

List all bibliographic references, in order of appearance in the document, at the end of the communication. When referring to them in the text, write the corresponding number in square brackets, following [1] if it is a book, if it is an article in magazine use [2], if it is a communication in a conference [3] and if it is a reference to a resource on the Internet use [4]. In case of multiple references, indicate references separated by a comma [1,3] if they are not consecutive or a dash in the middle [1-3] if they are consecutive references.

## Figures

All figures should be referenced in the text (see Figure 1). All figures must have a number, which will increase consecutively throughout the document, and a caption.



Figure 1 All figures must have a number, which will increase consecutively throughout the document, and a caption.

## Tables

All tables should be referenced in the text (see Table 1). All figures must have a number, which will increase consecutively throughout the document, and a caption. Try to use a consistent number of significant figures in the tables and throughout the text.

Table 1 All tables must have a number, which will increase consecutively through the document, and a caption.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Column 1** | **Column 2** | **Column 2** |
| **Row 1** | 6.00 | 6.00 | 6.00 |
| **Row 2** | 15.00 | 6.20 | 4.20 |
| **Row 3** | 1.00 | 1.00 | 1.00 |
| **Row 4** | 6.00 | 6.00 | 6.00 |

# Estado del arte

Internet ha dado un vuelco a nuestra existencia. Ha revolucionado las comunicaciones, hasta el punto de que ahora es nuestro medio preferido de comunicación cotidiana. En casi todo lo que hacemos, utilizamos Internet. Pedir una pizza, comprar una televisión, organizar una quedada amigos, enviar una foto por mensajería instantánea.

La propia Internet se ha transformado. En sus inicios (hace relativamente muy poco tiempo) era una red estática diseñada para transportar una pequeña carga de bytes o un mensaje corto entre dos terminales; era un depósito de información donde el contenido era publicado y mantenido sólo por codificadores expertos. Hoy, sin embargo, se cargan y descargan inmensas cantidades de y el contenido es propia, pues ahora todos pueden ser comentaristas, editores y creadores.

La aparición de la web 2.0 en la primera década del siglo XXI supuso en sí misma una revolución en la corta historia de Internet, al propiciar el auge de los medios sociales y otras herramientas de comunicación interactivas.

Con los nuevos desarrollos de software se han creado nuevas arquitecturas de programación y desarrollo web. El software ahora está más articulado y se utilizan FrameWorks de trabajo, se reutiliza código y se intercalan diferentes lenguajes.

Gracias a los servicios web, cualquier usuario puede acceder a través de un servidor gracias al internet y eso es una ventaja sin igual.

## ¿Qué son los Servicios Web y cómo funcionan?

Los servicios web son un medio estandarizado para permitir la comunicación entre las aplicaciones del cliente y el servidor. Un servicio web es un módulo de software diseñado para realizar un determinado conjunto de tareas.

Una vez invocado, el servicio web a través de la red podrá proporcionar la funcionalidad al cliente que lo invoque.

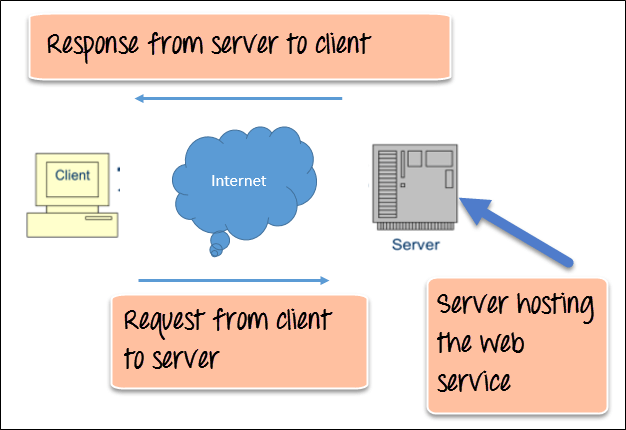


Figure 1 Visión servicio web.

El diagrama anterior muestra una visión muy simplista de cómo funcionaría realmente un servicio web. El cliente invocaría una serie de llamadas al servicio web mediante peticiones a un servidor que albergaría el servicio web real.

Estas peticiones se realizan a través de lo que se conoce como llamadas a procedimientos remotos. Las llamadas a procedimientos remotos son llamadas a métodos alojados en el servicio web correspondiente.

Por ejemplo, Amazon ofrece un servicio web que proporciona los precios de los productos vendidos en línea a través de amazon.com. El front-end o capa de presentación puede estar en .Net o en Java, pero cualquiera de los dos lenguajes de programación tendría la capacidad de comunicarse con el servicio web. [2]

El componente principal del diseño de un servicio web son los datos que se transfieren entre el cliente y el servidor, y eso es XML. XML (Extensible markup language) es una contraparte de HTML y fácil de entender el lenguaje intermedio que es entendido por muchos lenguajes de programación.

Así, cuando las aplicaciones hablan entre sí, lo hacen en XML. Esto proporciona una plataforma común para que las aplicaciones desarrolladas en varios lenguajes de programación se comuniquen entre sí.

## Servicios web tradicionales SOAP

Los servicios web utilizan algo conocido como SOAP (Simple Object Access Protocol) para enviar los datos XML entre aplicaciones. Los datos se envían a través de HTTP normal. Los datos que se envían desde el servicio web a la aplicación se denominan mensajes SOAP. El mensaje SOAP no es más que un documento XML. Como el documento está escrito en XML, la aplicación cliente que llama al servicio web puede estar escrita en cualquier lenguaje de programación.

## Servicios web RESTful

Los servicios web RESTful están construidos para funcionar mejor en la web. La Transferencia de Estado Representacional (REST) es un estilo que especifica restricciones, como la interfaz uniforme, que si se aplican a un servicio web inducen propiedades deseables, como el rendimiento, la escalabilidad y la modificabilidad, que permiten que los servicios funcionen mejor en la Web. En el estilo arquitectónico REST, los datos y la funcionalidad se consideran recursos y se accede a ellos mediante identificadores de recursos uniformes (URI), que suelen ser enlaces en la web. Se actúa sobre los recursos (arquitectura orientada a recursos) utilizando un conjunto de operaciones simples y bien definidas. El estilo arquitectónico REST limita la arquitectura a una arquitectura cliente/servidor y está diseñado para utilizar un protocolo de comunicación sin estado, normalmente HTTP. En el estilo de arquitectura REST, los clientes y los servidores intercambian representaciones de recursos utilizando una interfaz y un protocolo estandarizados.

Los siguientes principios fomentan que las aplicaciones RESTful sean simples, ligeras y rápidas:

* Identificación de recursos mediante URI: Un servicio web RESTful expone un conjunto de recursos que identifican los objetivos de la interacción con sus clientes. Los recursos se identifican mediante URIs, que proporcionan un espacio de direccionamiento global para el descubrimiento de recursos y servicios.
* Interfaz uniforme: Los recursos se manipulan mediante un conjunto fijo de cuatro operaciones de creación, lectura, actualización y eliminación: PUT, GET, POST y DELETE. PUT crea un nuevo recurso, que puede ser eliminado mediante DELETE. GET recupera el estado actual de un recurso en alguna representación. POST transfiere un nuevo estado a un recurso.
* Mensajes autodescriptivos: Los recursos se desvinculan de su representación para que se pueda acceder a su contenido en una variedad de formatos, como HTML, XML, texto plano, PDF, JPEG, JSON y otros. Los metadatos sobre el recurso están disponibles y se utilizan, por ejemplo, para controlar el almacenamiento en caché, detectar errores de transmisión, negociar el formato de representación adecuado y realizar la autenticación o el control de acceso. Para más información, consulte Respuesta a métodos y solicitudes HTTP y uso de proveedores de entidades para asignar cuerpos de entidades de respuesta y solicitud HTTP.
* Interacciones con estado a través de hipervínculos: Cada interacción con un recurso es sin estado; es decir, los mensajes de solicitud son autocontenidos. Las interacciones con estado se basan en el concepto de transferencia de estado explícita. Existen varias técnicas de intercambio de estado, como la reescritura de URI, las cookies y los campos de formulario ocultos. El estado puede ser incrustado en los mensajes de respuesta para apuntar a futuros estados válidos de la interacción. Para más información, consulte Uso de proveedores de entidades para asignar cuerpos de entidades de respuesta y solicitud HTTP y "Construcción de URI" en el documento Descripción general de JAX-RS.

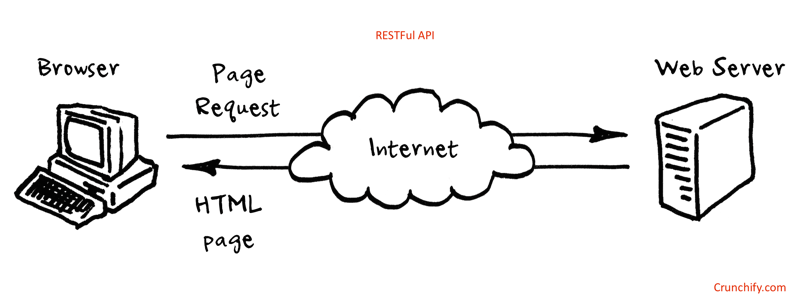


Figure 3 Diagrama de la arquitectura Rest [9].

## Modelo de Madurez de Richardson

## PLAY Framework

Play es un marco de trabajo de alta productividad para aplicaciones web en Java y Scala que integra componentes y APIs para el desarrollo de aplicaciones web modernas. Play fue desarrollado por desarrolladores web para el desarrollo de aplicaciones web.

Java es un lenguaje de programación basado en clases y orientado a objetos, diseñado para tener menos dependencias. Es uno de los lenguajes más utilizados y preferido por los programadores. Java también es una plataforma para el desarrollo de aplicaciones. Java es rápido, seguro y fiable y se utiliza ampliamente para el desarrollo de aplicaciones en ordenadores portátiles, centros de datos, consolas de juegos, superordenadores científicos, teléfonos móviles, etc.

Scala combina la programación funcional y orientada a objetos en un lenguaje conciso y de alto nivel. Los “Scala’s static types” ayudan a evitar errores en aplicaciones complejas, y sus tiempos de ejecución JVM y JavaScript le permiten construir sistemas de alto rendimiento con fácil acceso a un enorme ecosistema de bibliotecas.

Marco de trabajo o Framework en inglés se refiere a una estructura conceptual y tecnológica de soporte definido que sirve como base para el código). Los frameworks existen para simplificar la programación ayudándonos a: no escribir código repetitivo, mantener una buena estructura en el código y realizar código más complejo del que somos capaces de hacer sin ayuda [4]. Gracias a todas estas ventajas programar se convierte en algo más sencillo y rápido.

API de las siglas “Applicaction Programming Interface” es un software intermediario que permite la comunicación entre dos aplicaciones, un ejemplo concreto sería cuando usas una aplicación en tu teléfono móvil, la aplicación se conecta a Internet y envía datos a un servidor. El servidor recupera esos datos, los interpreta, realiza las acciones necesarias y los devuelve al teléfono [7]. A continuación, la aplicación interpreta esos datos y te presenta la información que querías de forma legible.

Play proporciona patrones de programación concisos y funcionales. Como marco de trabajo completo, Play incluye todos los componentes necesarios para construir aplicaciones web y servicios REST.

La arquitectura de Play utiliza Akka y Akka Streams de forma encubierta para proporcionar un consumo de recursos predecible y mínimo (CPU, memoria, hilos). Gracias a su modelo reactivo, las aplicaciones escalan de forma natural, tanto horizontal como verticalmente.

Una aplicación Play solo necesita el Play JAR file para funcionar correctamente, se puede utilizar tanto Scala como Java para programar le proyecto. Por tanto para funcionar solo necesita Java SE 1.8 o superior.

Las aplicaciones play también cuentan con soporte para sbt, entre otras cosas permite que el proyecto se ejecute en cualquier “maquina” y que automáticamente al compilar el proyecto, se descarguen todas las librerías y archivos necesarios para su correcto funcionamiento.[6]

Sbt funciona de forma diferente a otras tareas de construcción tradicionales. Sbt es fundamentalmente un motor de tareas. Su construcción se representa como un árbol de dependencias de tareas que necesitan ser ejecutadas, por ejemplo, la tarea de compilación depende de la tarea de fuentes, que depende de la tarea sourceDirectories y la tarea sourceGenerators, y así sucesivamente [5].

La estructura de una aplicación Play se compone de:

* App: El directorio de la aplicación contiene todos los artefactos ejecutables: Código fuente de Java y Scala, plantillas y fuentes de activos compilados.

Hay tres paquetes en el directorio app, uno para cada componente del patrón arquitectónico MVC: app/controllers, app/modelos y app/vistas.

* Public: Los recursos almacenados en el directorio público son activos estáticos que están servidos directamente por el servidor web.

Este directorio está dividido en tres subdirectorios: imágenes, hojas de estilo CSS y archivos JavaScript. Se deberíanorganizar así los activos estáticos para mantener la coherencia en todas las aplicaciones de Play.

* Conf: El directorio conf contiene los archivos de configuración de la aplicación. Hay dos archivos de configuración principales:
  + application.conf es el archivo de configuración principal de la aplicación.
  + routes es el archivo de definición de los routers.
* Lib: El directorio lib es opcional y contiene las dependencias de las bibliotecas no gestionadas, es decir, todos los archivos JAR que se queiran gestionar manualmente fuera del sistema de construcción.
* Build.sbt: Las principales declaraciones de construcción del proyecto se encuentran generalmente en build.sbt en la raíz del proyecto.
* Project: El directorio del proyecto contiene las definiciones de construcción de sbt:
  + plugins.sbt define los plugins sbt utilizados proyecto.
  + build.properties contiene la versión de sbt que se utilizará para construir la aplicación.
* Target: El directorio de destino contiene todo lo generado por el sistema de construcción. Puede ser útil saber qué se genera aquí:
  + classes/ contiene todas las clases compiladas (tanto de fuentes Java como Scala).
  + classes\_managed/ contiene sólo las clases que son gestionadas por el framework (como las clases generadas por el router o el sistema de plantillas).
  + resource\_managed/ contiene recursos generados, típicamente activos compilados como LESS CSS y resultados de compilación de CoffeeScript.
  + src\_managed/ contiene fuentes generadas, como las fuentes Scala generadas por el sistema de plantillas.
  + web/ contiene activos procesados por sbt-web como los de las carpetas app/assets y public.

# DESCRIPCIÓN INFORMÁTICA

Gjfhfg hgjghj hgjgh jghjhg jghj ghj hgj ghj gh jgh j hgj hg j ghj gh jhg j hgj hg j ghj hgj ghj ghj hg j hgj gh jhg j ghj ghj hgj hg j hgj

## Análisis

Los moteros son personas aficionadas a las motos que les gusta invertir su tiempo en cuidar su vehículo y salir “a pasear” lo máximo posible. En este sentido, como cualquier persona que posea una afición o un hobby, se encuentran la necesidad de compartir esta afición con otros iguales, compartiendo la misma pasión y comentar todo lo que sucede en relación a la moto.

Una de las principales aficiones de estos usuarios, es disfrutar del tiempo libre que proporciona el fin de semana para pasear en su moto. Generalmente, esta actividad se realiza con otros aficionados, permitiendo disfrutar de momentos de conducción al aire libre, disfrutar del buen tiempo en gran parte del año, disfrutar (durante las paradas) de la gastronomía de parajes o localidades remotas además de, evidentemente, disfrutar de una buena y desafiante ruta, poblada de muchas y buenas curvas, que permita obtener esa sensación de velocidad y hermanamiento con la moto.

Sin embargo, gestionar una reunión motera no es una cuestión sencilla. Por ejemplo, cada motero tiene su propio estilo al conducir: desde los que les gusta disfrutar del paseo e ir a velocidades contenidas hasta aquellos que les gusta llevar un ritmo más animado. Además, no todos los trazados son del gusto de todos los conductores: algunos disfrutan con rutas cuyos trazados contengan múltiples curvas y dificultades, mientras que otros prefieren rutas que incluyan pequeños puntos de interés turístico o gastronómico; haciendo que la ruta (y la quedada) sea algo más que innumerables kilómetros de curvas y velocidad. Sin embargo, esta diversidad es la que fomenta que existan muchos aficionados a las motos, por sus diferentes alternativas de ocio.

Generalmente, para gestionar una reunión motera hay que buscar a compañeros de viaje (personas conocidas que tengan moto y que tengan disponibilidad horaria en los días que se planea). Además, los compañeros de la quedada suelen ser personas de un nivel de conducción similar (a nadie le gusta sentirse retrasado o forzado en su conducción). Y, por último, deben ponerse de acuerdo en la ruta que se seguirá durante la quedada. Todas estas restricciones hacen que, muchas veces, la organización de una reunión sea inviable (ya que el número de personas conocidas es reducido).

En cuanto a la elección de la ruta, muchas veces se aprovecha el conocimiento de alguno de los participantes, otras veces se puede extraer de internet o se puede haber descubierto mientras se realizaba algún otro recorrido. Sea cual sea la forma en la que se elige, es fundamental tener conocimiento de ciertos aspectos inherentes a la misma: qué cantidad de curva tiene (y su dificultad), cómo está el estado del asfalto, qué nivel de tráfico tiene habitualmente en los días que se realizará la quedada, si es una ruta con mucho ambiente motero (lo que permitirá conocer a nuevos moteros y abrir más el círculo de futuros participantes en otras reuniones), qué tipo de paisaje tiene (verde frondoso o árido… sobre todo muy útil cuando se plantea una ruta turística), si tiene puntos de restauración conocidos o interesantes (si la ruta tiene un interés gastronómico), etcétera. Y, una vez seleccionada la ruta, se suele poner en común la cantidad de kilómetros que se realizarán, la duración aproximada, las paradas para descanso y comida, las gasolineras en la ruta.

Generalmente las rutas son de 200km y se suele quedar por la mañana para aprovechar las horas de luz y que la carretera esté más caliente, agarrando así más el neumático.

Una vez establecida la ruta, la gente y la hora, se hace la quedada o salida, se suele quedar en una gasolinera para ultimar los detalles, revisar la presión de las ruedas ya que se someterá a la moto a unas condiciones más extremas que de costumbre y para poner gasolina, las motos tienen un depósito mucho más pequeño que los coches y su autonomía suele oscilar entorno a los 250km. Finalmente se decide un poco el orden en el que se quiere ir y sobre todo quien irá primero guiando el camino, esto a veces es más sencillo si las personas tiene un intercomunicador en el casco ya que al igual que los manos libres de los coches, permite la comunicación entre los conductores en marchas introduciendo un pequeño micrófono y unos altavoces dentro del casco.

Toda esta gestión de la organización de una reunión se suele realizar, hoy en día, de manera informal, mediante grupos de WhatsApp, limitados generalmente a personas conocidas al requerir el número de teléfono, además es algo desorganizado ya que la gente acaba hablando de temas diferentes que se entremezclan y hacen difícil su lectura si solo se buscan los mensajes sobre futuras quedadas. Estas quedadas incluyen desde 2 miembros hasta concentraciones de más de mil motos, sobre todo si están organizadas por alguna persona famosa, que aprovecha sus múltiples seguidores en redes sociales para promocionar la quedada.

Durante la ruta se alternan las posiciones de las motos para “coger rueda”, es decir, ir detrás, de una persona diferente, es algo necesario al haber distintos niveles entre los conductores, así se consigue un mayor dinamismo. Además, los distintos conductores suelen intentar coger las curvas cada vez más rápido, tumbar más la moto, medir la velocidad media o incluso ver en qué tipos de curvas van más rápido para saber dónde deben mejorar su conducción.

Los moteros son un grupo amplio de personas, pero con pocas relaciones amistosas entre ellos. Generalmente las primeras quedadas se realizan con algún amigo anterior a la adquisición de la moto. Probablemente el nuevo motero conozca a un par de amigos con moto con los que empezará sus primeras rutas y con bastante probabilidad, alguno de ellos traerá a consigo a otros amigos, ampliando así poco a poco el círculo de moteros conocidos.

Pero el problema no se queda ahí, ya que la gente que acaba de empezar en el mundo motero no solo tiene dificultades para encontrar a otros moteros dispuestos a salir de ruta, sino que esta persona no se conoce las zonas más frecuentadas por los moteros o las mejores zonas y puertos de montaña para salir de ruta. De esta manera cogen tramos largos de autopista (algo a evitar) ya que lo que se busca son curvas interminables, algo difícil de encontrar si no se conoce bien la zona.

Tras varias quedadas cada vez con más personas nuevas, el repertorio de contactos se va ampliando, pero una vez encontrado un grupo de gente, hay que encontrar un día e intervalo horario que venga bien a todos, algo muy difícil de cuadrar y más teniendo en cuenta que no puede llover ya que la ruta pasaría de ser algo divertido a ser algo peligroso.

Dentro de este círculo, cada motero intentará buscar a personas afines a el mismo y que puedan hacer match para futuras rutas. Por ejemplo, un motorista se puede fijar en el nivel de conducción del compañero, su moto (comprendiendo la marca, el modelo, el estilo y la potencia), la edad, su personalidad o incluso su permiso de conducción, ya que eso limitará la potencia máxima de la moto que puede conducir.

Como decía anteriormente, la moto concreta de cada individuo es algo influyente en la quedada, lo más determinante es el estilo de moto y su potencia, esto suele ir directamente relacionado con la personalidad del conductor, sus gustos y su carné actual.

Existen moteros con motos “Trail”, motos más grandes de lo normal, permitiendo viajar con mayor comodidad. Estas motos también son hibridas, para poder ir tanto por el campo como por asfalto. A este tipo de moteros les suele gustar hacer viajes largos con las motos (incluso de días o semanas) y hacer rutas largas por su zona a poder ser circulando por zonas de tierra.

Existen también las motos “R”, son motos muy deportivas y sus dueños suelen preferir hacer rutas por zonas con curvas, con pocas paradas o incluso hacer rodadas en circuito.

Están los moteros con motos “Custom”, motos como las “Harley Davidson” o similares, estos moteros prefieren ir por zonas con mucho paisaje y con pocas curvas o curvas no muy pronunciadas ya que son motos más grandes y pesadas que no pueden llevar una conducción tan deportiva. Estos moteros suelen pertenecer a grupos, llevar vestimentas acordes al estilo “Harley” y quedar en bares moteros específicos.

Finalmente existen los usuarios más comunes, suelen tener motos “Naked”, motos muy polivalentes para todo tipo de uso sin llegar a ser las mejores en ningún aspecto. Estos moteros suelen buscar rutas con curvas y parar a comer o tomar algo.

### Usuarios

Los usuarios serán las personas que tienen una moto y buscan en la aplicación quedadas para conocer a más moteros en las distintas rutas disponibles.

Sus atributos principales serán:

* Nombre
* Edad
* Sexo
* Bio: donde el usuario se describirá a libre elección de una manera personal para que otros usuarios le puedan conocer de una manera más profunda.
* Terreno: donde el usuario indicará si prefiere ir por carretera asfaltada, por lugares de tierra o por ambos.
* Carné: aquí el usuario indicará su carné actual, lo que limitará las motos que puede conducir.

En la actualidad existen 3 tipos de carné:

* + A1: autoriza la conducción de motos automáticas o de marchas siempre que no excedan los 15kW de potencia, los 125cm³ de cilindrada o una relación peso a potencia de 0.1 kW/kg. Este carné está disponible a partir de los 16 años o tras 3 años con el carné de coche B siempre y cuando solo sea conducción por territorio nacional.
  + A2: autoriza la conducción de motos automáticas o de marchas siempre que no excedan los 35kW de potencia o una relación peso a potencia de 0.2 kW/kg. Este carné está disponible a partir de los 18 años. También se pueden conducir motos de mayor potencia limitadas a los 48kW de potencia reglamentarios siempre y cuando estás no excedan lo 70kW de potencia.
  + A: autoriza la conducción de motos automáticas o de marchas sin límite de de potencia o relación peso/potencia. Este carné está disponible a tras dos años de posesión del carné A2.
* Nivel: aquí el usuario indicara su nivel de conducción en un rango del 1 (siendo el nivel más bajo) al 4 (siendo un conductor experto).
* Valoración: este campo estará rellenado por otros usuarios tras una quedada conjunta donde darán una valoración del 1 al 5, siendo el 5 la valoración más alta.
* Moto: el usuario indicará su moto actual y ciertas características.

Atributos de las motos:

* Estilo
  + Trail
  + R
  + Custom
  + Naked
* Marca
* Modelo
* Potencia
* Intercomunicador: indicará si el usuario posee o no intercomunicador lo que permitirá la comunicación entre varios usuarios durante la ruta.

### Rutas

Las rutas indicarán principalmente el camino físico que las motos seguirán. Las rutas no contemplan una quedada concreta y temporal sino algo atemporal.

Sus atributos principales serán:

* Nombre
* Tipo: donde se indicará el tipo de la ruta, según esté la preferirán más un tipo de usuarios u otro. Una ruta puede estar sujeta simultáneamente a más de un tipo.

Los distintos tipos de ruta son:

* Gastronómica: rutas con varias paradas en bares de tapas típicos de montaña.
* Curvas: rutas en las que se busca sacar el máximo partido a la moto y al nivel de conducción debido a la alta cantidad de curvas.
* Paisajes: rutas con mucho paisaje, vistas, paradas en miradores, naturaleza etc.
* Km: se indicará la distancia total de la ruta.
* Recorrido: se indicará punto de partida, punto de llegada y las carreteras que hay que seguir (durante cuantos kilómetros y la salida más inmediata).
* Estado del asfalto: indicará si la carretera está en estado bueno (recién asfaltada), intermedio o malo.
* Terreno: indicará si la ruta es por carretera asfaltada o por lugares de tierra.
* Dificultad: indicara la dificultad de la ruta del 1 al 4 siendo el 4 la dificultad más alta.
* Valoración: esta la indicarán distintos usuarios y facilitará la elección de la ruta a distintos usuarios. Contará con una valoración del 1 al 5, siendo 5 la puntuación más alta y un pequeño comentario a libre elección.
* Duración: duración total en minutos de la ruta.

### Quedada

Las quedadas organizan el tiempo y lugar de partida junto con los usuarios concretos que recorrerán una ruta concreta.

Sus atributos principales serán:

* Hora inicial
* Hora final
* Lugar de partida
* Lugar final
* Usuarios iniciales
* Usuarios invitados
* Ruta concreta elegida para esta quedada
* Paradas: aquí se indicarán las paradas concretas que se harán en esta quedada, ya sean checkpoints, miradores, gasolineras, restaurantes, bares etc.
* Valoración de la quedada

### Posibles incorporaciones futuras

Algo muy conocido y usado entre los moteros son los saludos en orden de marcha, durante las rutas estos se suelen cruzar con otros grupos y hay una sensación en el aire de comunidad, por esta razón existe un saludo. Este saludo consiste en hacer una “V” con el índice y el corazón y guardar el resto de los dedos. El saludo se remonta a la guerra de los 100 años y durante el paso de los años lo han ido adoptando diferentes personas, pero Barry Sheene, un reconocido piloto del motociclismo de los años 70, fue el primero en hacer este saludo a sus seguidores y poco a poco todos los moteros se empezaron a saludar de esta manera cuando se cruzaban, sobre todo en concentraciones moteras o en rutas. En futuras versiones de la aplicación, está podría mostrar los saludos y ciertas recomendaciones para la ruta antes de iniciar esta. Este es un ejemplo de ellos:

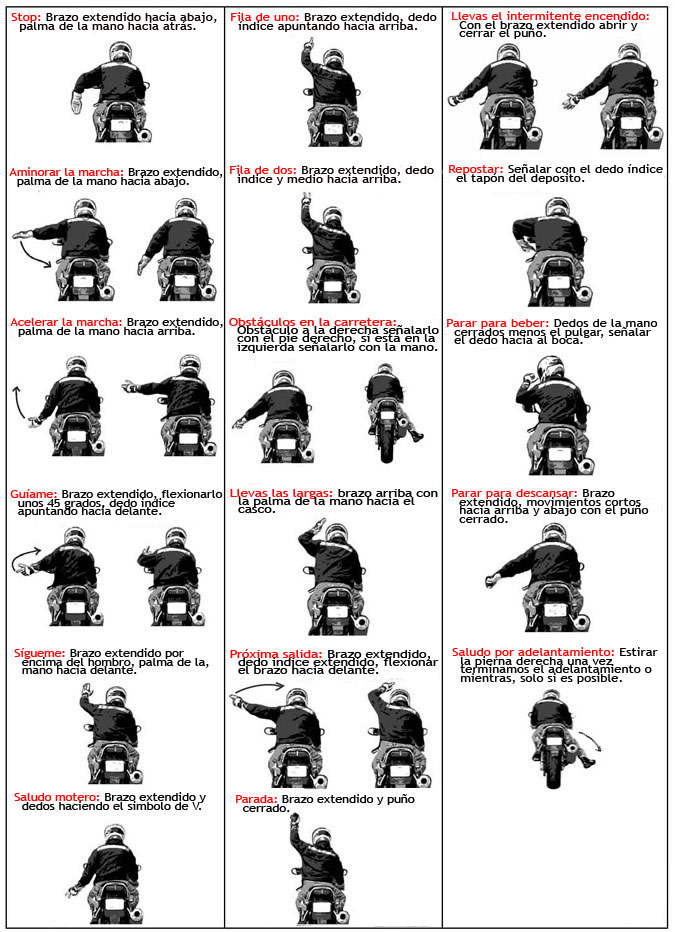


Figure 2 Saludos y señas moteras [1].

La comunidad motera no solo necesita gente con la que salir de ruta sino también gente con la que hablar y compartir sus experiencias y conocimientos sobre motos. Por ello la incorporación de un foro con diferentes temáticas como: Motos (general), compraventa, Moto GP, rutas, equipación, mecánica-averías y modificaciones.

* Otra función muy interesante sería añadir telemetría a la aplicación, de manera que está esté activa en el teléfono móvil mientras la persona realiza la ruta y al finalizarla indique valores útiles para un conductor como: velocidad media, velocidad punta, grado de inclinación máxima a derechas e izquierdas, velocidad máxima y media en curva y cantidad de curvas a derechas e izquierdas. Por último, ver el trazado completo con colores asociado a velocidad o inclinación y otros datos citados anteriormente sería algo muy útil para mejorar la conducción y conocer puntos fuerte y débiles.

## Aálisis

# CONCLUSIONES Y TRABAJO FUTURO

# CONCLUSIONS

# REFERENCES

1. Página web de Motopoliza. <https://www.motopoliza.com/consejos-seguros-y-motos/ventajas-de-las-motos/dudas-de-motos/lenguaje-de-gestos-moteros/> (Ultimo acceso: marzo 2021).
2. Página web de guru99. <https://www.guru99.com/web-service-architecture.html> (Último acceso: marzo 2021).
3. Página web de Oracle. <https://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html> (Ultimo acceso: marzo 2021).
4. Página web de Orix. https://www.orix.es/que-es-un-framework-y-para-que-se-utiliza (Ultimo acceso: marzo 2021).
5. Página web de playframework. <https://www.playframework.com/documentation/2.8.x/Requirements> (Ultimo acceso: marzo 2021).
6. Página web de playframework. [https://www.playframework.com/documentation/2.8.x/BuildOverview](https://www.playframework.com/documentation/2.8.x/Requirements) (Ultimo acceso: marzo 2021).
7. Página web de Mulesoft. <https://www.mulesoft.com/resources/api/what-is-an-api> (Último acceso: marzo 2021).
8. Página web de Guru99. <https://www.guru99.com/java-platform.html> (Último acceso: marzo 2021).
9. Página web de SoloCodigoWeb. <http://solocodigoweb.com/blog/2016/09/15/cuando-usar-soap-cuando-usar-rest/> (Último acceso: marzo 2021).
10. Página web de Java Platform. <https://javaee.github.io/tutorial/jaxrs001.html#:~:text=Resource%20identification%20through%20URI%3A%20A,for%20resource%20and%20service%20discovery>. (ultimo acceso marzo 2021).
11. Bronzino JD. The Biomedical Engineering Handbook. Vol 1. CRC Press, 1999 (ISBN: 084930461X).
12. Doarn CR, Adilova F, Lam D. Review of telemedicine in Uzbekistan. *Journal of Telemedicine and Telecare*, vol 11, sup 3, 2005, pp 135-9 (ISSN: 1357-633X).
13. Gómez JA, Galeano G, Fernández F. Circuito para la aceleración de la etapa de comparación de un algoritmo paralelo bioinspirado. *Actas del I Congreso Español de Algoritmos Evolutivos y Bioinspirados (AEB'02)*, Cáceres, 2002, pp 410-5 (ISBN: 84-607-3913-9).
14. Web page of the XXIII Congreso Anual de la Sociedad Española de Ingeniería Biomédica (CASEIB 2005). http://caseib2005.gbt.tfo.upm.es (Accessed: May 2015).