

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

GRADO EN INGENIERÍA EN ELECTRÓNICA Y  
AUTOMÁTICA INDUSTRIAL



Trabajo Fin de Grado

“Clasificación del modo de transporte mediante  
Smartphones”

Gonzalo Díaz-Romeral Marcos  
2015



# UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

## GRADO EN INGENIERÍA EN ELECTRÓNICA Y AUTOMÁTICA INDUSTRIAL

Trabajo Fin de Grado

“Clasificación del modo de transporte mediante Smartphones”

Alumno: Gonzalo Díaz-Romeral Marcos

Director: D. Luis Miguel Bergasa Pascual

**Tribunal:**

**Presidente:** D. Rafael Barea Navarro.

**Vocal 1:** D. José Manuel Rodríguez Ascariz.

**Vocal 2:** D. Luis Miguel Bergasa Pascual.

Calificación: .....

Fecha: .....



"Esto no es volar. Es caer con estilo."  
Buzz Lightyear.



# Agradecimientos

*En primer lugar, me gustaría agradecer a D. Luis Miguel Bergasa la confianza que depositó en mí al asignarme este TFG y su ayuda al guiarme hasta conseguir este objetivo. A los compañeros del equipo de investigación por compartir tantas horas en el laboratorio durante todos estos meses, en especial a Eduardo Romera por estar siempre disponible para echar una mano. A Eva, Joaquín, Eduardo, Lidia y muy especialmente a Mario por participar recogiendo datos de sus viajes, parte de este trabajo es vuestro.*

*Por otra parte, agradezco a todos mis amigos, compañeros de universidad y de prácticas, compañeros y profesores de conservatorio y conjuntos musicales, su comprensión y apoyo durante este año tan duro en el que, muchas veces, no he podido llegar a todo. A Sergio, Jesús y Mario por hacer de los miércoles la escapada de la rutina.*

*Y para finalizar, tengo que dar las gracias a los que de verdad les debo todo, los que me han aguantado, me aguantan y me aguantarán en los momentos difíciles, los que me han hecho llegar hasta aquí; a toda mi familia; a Lidia por ser mi compañera de la vida, por este camino que hemos recorrido juntos y que siempre tiende a juntarnos; a mis abuelos Isa y Emilio, porque siempre he sentido vuestro cariño y me habéis apoyado en todo, os quiero; a mi hermano Enrique, por todo lo que vales y no te lo hago saber, ahora que yo acabo te toca a ti; a mi padre, por estar siempre dispuesto a ayudarme en lo que necesite y confiar en mí, y a mi madre, por tirar del carro día tras día, por esforzarte tanto en sacarme a delante durante estos años y por darme la vida.*





# Índice general

<b>I Memoria</b>	<b>21</b>
<b>1. Introducción</b>	<b>23</b>
1.1. Estado del Arte . . . . .	24
1.2. Objetivos . . . . .	26
<b>2. Recogida de datos en desplazamientos reales</b>	<b>29</b>
2.1. Motivación . . . . .	29
2.2. Sensores del smartphone . . . . .	29
2.2.1. Acelerómetros . . . . .	30
2.2.2. Servicios de localización . . . . .	30
2.3. Aplicación prototipo . . . . .	31
<b>3. Procesamiento de datos</b>	<b>33</b>
3.1. Motivación . . . . .	33
3.2. Tratamiento inicial . . . . .	33
3.2.1. Filtro paso alto . . . . .	33
3.2.1.1. Definición . . . . .	34
3.2.1.2. Filtro paso alto en tiempo discreto . . . . .	34
3.2.2. Almacenamiento de datos . . . . .	35
3.3. Técnicas de procesamiento aplicadas . . . . .	36
3.3.1. Promedio . . . . .	36
3.3.2. Multiplicaciones . . . . .	37
3.3.3. Integral . . . . .	38
3.3.4. Diferencial . . . . .	39
3.4. Conclusiones . . . . .	44
<b>4. Clasificadores</b>	<b>45</b>
4.1. Motivación . . . . .	45
4.2. Support Vector Machines . . . . .	45
4.3. Clasificador Naïve Bayes . . . . .	48
<b>5. TransMode</b>	<b>49</b>
5.1. Motivación . . . . .	49
5.2. Algoritmo . . . . .	49
5.3. Servidor . . . . .	50
5.3.1. Diseño del servidor . . . . .	51
5.3.2. Código PHP . . . . .	51
5.3.3. Estructura de datos . . . . .	52
5.4. Análisis de datos . . . . .	52

---

<b>6. Resultados</b>	<b>55</b>
6.1. Promedio . . . . .	56
6.2. Multiplicaciones . . . . .	56
6.3. Integral . . . . .	58
6.4. Diferencial . . . . .	60
6.5. Configuración óptima . . . . .	64
<b>7. Conclusiones y Trabajos Futuros</b>	<b>65</b>
7.1. Conclusiones . . . . .	65
7.2. Trabajos Futuros . . . . .	65
<b>II Manual de Usuario</b>	<b>69</b>
7.3. Manual . . . . .	71
7.3.1. Aplicación prototipo . . . . .	71
7.3.2. TransMode . . . . .	71
<b>III Pliego de condiciones</b>	<b>73</b>
<b>IV Presupuesto</b>	<b>77</b>

# Índice de figuras

1.1. Aplicación Ecorio. . . . .	24
1.2. Jitter de la frecuencia de lectura de los acelerómetros. . . . .	25
1.3. Esquema de sensores utilizados por <i>CO<sub>2</sub>GO</i> . . . . .	26
1.4. Ejemplo de aceleraciones para distintos modos de transporte. . . . .	26
2.1. Sistema de referencia del acelerómetro . . . . .	30
2.2. Interface de la aplicación para recoger datos. . . . .	31
2.3. Datos recogidos por la aplicación . . . . .	31
3.1. Datos del acelerómetro reales vs. datos pasado por el filtro paso alto . . . . .	34
3.2. Representación de los datos de aceleración y velocidad de las distintas clases . . . . .	35
3.3. Promedio: representación de las características calculadas. . . . .	40
3.4. Promedio: distribuciones de probabilidad de densidad gaussiana de las características calculadas. . . . .	40
3.5. Multiplicaciones: representación de las características calculadas. . . . .	41
3.6. Multiplicaciones: distribuciones de probabilidad de densidad gaussiana de las características calculadas. . . . .	41
3.7. Integración: representación de las características calculadas. . . . .	42
3.8. Integración: distribuciones de probabilidad de densidad gaussiana de las características calculadas. . . . .	42
3.9. Derivación: representación de las características calculadas. . . . .	43
3.10. Derivación: distribuciones de probabilidad de densidad gaussiana de las características calculadas. . . . .	43
4.1. Ejemplo de SVM en 2D . . . . .	46
4.2. Función Kernel Polinomial . . . . .	47
4.3. Función Kernel MLP . . . . .	47
4.4. Función Kernel RBF . . . . .	47
5.1. Algoritmo de detección de modos . . . . .	50
5.2. Esquema del método LAMP . . . . .	51
5.3. Esquema de almacenaje del servidor . . . . .	52
6.1. Promedio: índice de error en función del tiempo procesado. . . . .	56
6.2. Promedio: curvas ROC SVM vs Bayes en función del tiempo muestreado. . . . .	57
6.3. Multiplicación: índices de error en función del intervalo de tiempo procesado. . . . .	58
6.4. Multiplicación: curvas ROC SVM vs Bayes en función del tiempo muestreado. . . . .	59
6.5. Integración: índices de error en función del intervalo de tiempo procesado. . . . .	61
6.6. Integración: curvas ROC SVM vs Bayes en función del tiempo muestreado. . . . .	62
6.7. Derivación: índices de error en función del intervalo de tiempo procesado. . . . .	63

7.1. Interface TransMode. . . . .	66
7.2. Impacto de TransMode en el dispositivo. . . . .	67
7.3. Interface de la aplicación para recoger datos. . . . .	71
7.4. Interface TransMode. . . . .	72

# Índice de tablas

6.1. Promedio: índices de error para clasificadores SVM. . . . .	56
6.2. Promedio: índices de error para clasificadores Bayes. . . . .	58
6.3. Multiplicación: índices de error para clasificadores SVM. . . . .	60
6.4. Multiplicación: índices de error para clasificadores Bayes. . . . .	60
6.5. Integración: índices de error para clasificadores SVM. . . . .	60
6.6. Integración: índices de error para clasificadores Bayes. . . . .	61
6.7. Derivación: índices de error para clasificadores SVM. . . . .	63
6.8. Derivación: índices de error para clasificadores Bayes. . . . .	63
6.9. Promedio: configuración óptima. . . . .	64
6.10. Índices de error para distintos valores de kUpdateFrequency. . . . .	64
6.11. Configuración seleccionada. . . . .	64
7.1. Resultados de TransMode en viaje. . . . .	66
7.2. Costes de hardware . . . . .	80
7.3. Costes de software . . . . .	80
7.4. Costes de mano de obra . . . . .	80
7.5. Costes totales de ejecución material . . . . .	80
7.6. Costes totales de ejecución material . . . . .	80
7.7. Presupuesto de ejecución por contrata . . . . .	80
7.8. Importe total del presupuesto . . . . .	81



# Resumen

En el siguiente documento se presenta un novedoso sistema de detección automática del modo de transporte en el que viaja un usuario, utilizando exclusivamente los datos aportados por los sensores de aceleración y de localización que incorpora el smartphone que debe portar con él. Este sistema se implementará en una aplicación, llamada TransMode, que realizará predicciones en tiempo real recogiendo datos continuamente y permitiendo su ejecución en segundo plano sin afectar en el correcto funcionamiento del dispositivo.

La identificación automática del modo de transporte puede ser utilizada para proveer información relevante al usuario, como el gasto de combustible y la emisión de  $CO_2$  del desplazamiento si se produce en vehículo, o el gasto calórico si se produce andando o en bicicleta. Además, una aplicación para smartphones permite recoger información de un gran número de usuarios, lo que se conoce como *big data*. Esta información está siendo demandada por entidades privadas y agencias de planificación urbana, para reconocer los patrones de movimiento, evaluar la eficiencia del transporte de una ciudad y mejorar las infraestructuras de las ciudades.

Con este propósito, se realiza un estudio que analiza la efectividad de distintas técnicas de procesamiento con los datos que recoge la aplicación, obteniendo una serie características asociadas a cada una de los modos de transporte y que se utilizan para entrenar varios clasificadores (SVM y Naïve-Bayes). Como resultado, obtendremos una configuración que nos permite realizar la clasificación con la mayor precisión, afectando lo mínimo posible a la duración de la batería y sin entorpecer el uso cotidiano del dispositivo mientras la nuestra se ejecuta en segundo plano.

**Palabras clave:** clasificación del modo de transporte, smartphone, acelerómetros, sensores de localización, big data.





# Abstract

In this paper, we present a novel transportation mode detection algorithm, using exclusively acceleration and location data provided by smartphones. This model will be implemented in an application for smartphones that perform real-time predictions.

Automatic transportation mode identification can be used to show relevant information to the user, such as fuel consumption and the  $CO_2$  emissions if it's a vehicle travel, or the number of calories burnt if it occurs by foot or by bike. In addition, smartphones can collect information from a large number of people, which is known as *big data*. This information is being used by private firms and urban planners to recognize the patterns of people movement, evaluate the transportation efficiency in a city and improve the infrastructure of the cities.

For this purpose, we made a study analyzing the effectiveness of different processing techniques with the smartphone's collected data, obtaining characteristics associated to each mode of transport, and using this characteristics to train several classifiers (SVM and Naïve-Bayes). As a result, we obtain a configuration that allows us to make the classification affecting as little as possible to the battery duration and without disturbing daily phone use, when our application is running in the background.

**Keywords:** transportation mode classification, smartphone, accelerometers, location sensors, big data.



# Resumen extendido

Los métodos convencionales de recogida de información sobre movilidad de las personas en grandes ciudades no satisfacen las necesidades de los gobiernos o grandes empresas, ya que no consiguen un análisis suficientemente individualizado de los comportamientos de los ciudadanos en cuanto al transporte. Las nuevas técnicas de recogida de datos usando dispositivos móviles están convirtiéndose en la fuente de información más importante para grandes empresas. Sin embargo, la falta de un sistema capaz de detectar cómo se efectúa un desplazamiento impide que esta información sea accesible. Por ello, el objetivo de este trabajo es implementar un algoritmo de clasificación del medio de transporte en una aplicación de smartphone haciendo uso de los sensores que este dispositivo incorpora y que funcione automáticamente, aportando en tiempo real información sobre los desplazamientos al usuario, pero también registrando sus comportamientos para poder evaluar los patrones de movilidad de grandes ciudades.

Los únicos sensores que son utilizados para realizar esta detección son los acelerómetros y los sensores de localización, evitando usar los demás para evitar un excesivo consumo de batería o que la aplicación afecte en el funcionamiento del dispositivo. Los *acelerómetros* recogen las fuerzas que actúan sobre el dispositivo, en un instante de tiempo, en los ejes cartesianos (x,y,z). Esto nos permite obtener datos de los patrones de aceleración que se experimentan en los modos de transporte estudiados. Los *sensores de localización* nos aportan los datos de latitud y longitud, y además la velocidad del desplazamiento.

El trabajo, que será desarrollado en profundidad en capítulos posteriores, consta de las diferentes partes:

1. Recogida de datos en desplazamientos reales.

En la primera fase de este trabajo es necesario acumular datos de desplazamientos reales en cada uno de los medios de transporte con los que trabajará la aplicación definitiva, para posteriormente realizar el entrenamiento de los clasificadores. Estos datos serán recogidos mediante una aplicación que leerá la información aportada por los sensores que implementa el propio smartphone, y deben reflejar, de manera real, los distintos comportamientos que se pueden experimentar en cada uno de los medios de transporte. La aplicación presenta una interfaz con interruptores que sirven para elegir el modo de transporte antes de comenzar el desplazamiento y escribir un fichero de texto donde se guardarán los datos de aceleración recogidos en los tres ejes del dispositivo, la latitud, la longitud y la velocidad para cada una de las muestras obtenidas de los sensores del dispositivo.

2. Procesamiento de datos.

Los datos de aceleración recogidos con la aplicación prototipo son datos en RAW, sobre los que se necesita aplicar un procesamiento inicial para quedarnos con la información que diferencia a cada medio de transporte. Para ello, se realiza un filtrado paso alto a las aceleraciones recogidas para eliminar la contribución de la fuerza de la gravedad, ya que

el dispositivo puede ser portado en cualquier posición, y se calcula un valor escalar que representa el valor absoluto de las aceleraciones de los tres ejes obtenida por raíz cuadrada de suma de cuadrados.

Después se emplean distintas técnicas de procesamiento con los datos obtenidos desde la aplicación prototipo, con la finalidad de diferenciar o separar en la medida de lo posible distintas características asociadas a cada medio de transporte, para posteriormente utilizarlas al entrenar los clasificadores y determinar cual es la técnica más eficaz para diferenciar los distintos tipos de desplazamientos contemplados. Las técnicas de procesamiento utilizadas son: promediado de muestras, multiplicación, integración y derivación.

### 3. Análisis de efectividad de varios clasificadores.

Se estudia la eficacia en la disgregación de los datos en las distintas clases de transporte de dos tipos de clasificadores: SVM y Naïve-Bayes, dado que son dos clasificadores ampliamente utilizados en este tipo de aplicaciones por su alto poder discriminatorio y su facilidad de entrenamiento; se escogerá el que ofrezca mayor precisión. Para ello, separamos todos los datos que obtenemos en el apartado anterior en dos grupos: uno para entrenamiento supervisado, y el otro para hacer tests. El resultado más favorable será el que permita la mejor clasificación y consuma menos recursos del dispositivo.

### 4. Desarrollo de la aplicación definitiva: TransMode.

Obtener una aplicación que funcione en segundo plano de forma continua y automática recogiendo los datos y actualizando la información en tiempo real. Esta aplicación generará un documento en el que se recogerá un informe detallado de las rutas y el medio de transporte utilizado para un desplazamiento y lo mandará al servidor.

### 5. Conclusiones y trabajos futuros.

Se expone el funcionamiento de la aplicación TransMode, analizándola en distintos medios de transporte y comprobando su precisión en la detección y el impacto que causa en el dispositivo. También se indican los trabajos futuros a realizar para conseguir mejorar la aplicación.

Parte I

**Memoria**



# Capítulo 1

## Introducción

La mayoría de la información utilizada en los estudios sobre movilidad de las ciudades proviene de encuestas, realizadas bien vía internet o puerta a puerta. Esto supone una gran limitación en la obtención de resultados, ya que la información es difícil de actualizar, costosa de obtener, y representa a un pequeño porcentaje de la población de una ciudad [1]. Este sistema para la recopilación de información no satisface las necesidades de los demandantes, empresas privadas o gobiernos, ya que las ciudades modernas son cada día más complejas y presentan conductas muy dispares; y es difícil que las personas describan exactamente los detalles de sus desplazamientos en las encuestas de movilidad. Además, estos métodos no permiten recoger información de cada uno de los viajes de manera independiente, y tampoco registran los cambios en las conductas de transporte de una misma persona; a no ser que la encuesta sea repetida en varias ocasiones, lo que acaba provocando una pérdida de interés en los encuestados, que dejan de participar en ellas.

Sin embargo, el implacable avance sufrido durante los últimos años en el campo de la telefonía móvil, que se evidencia en la aparición y expansión de los smartphones, nos permite obtener información relevante mediante innovadoras técnicas imposibles de aplicar hasta ahora. Éstas reciben el nombre de *social sensing* o *urban sensing* [2] y dan acceso a datos de un gran número de personas con mucho menos esfuerzo que los métodos tradicionales, ofreciendo la posibilidad de reconocer los comportamientos de movilidad bien a un usuario para ofrecerle una información individualizada, o analizar los datos de una ciudad entera para mejorar las planificaciones del transporte público y sus infraestructuras.

En un principio, los datos más demandados para la información sobre movilidad fueron los concernientes a la ubicación de los usuarios. Éstos son accesibles mediante los sistemas de localización [3] y la tecnología GPS (Global Positioning System) que nos aportan las coordenadas en las que se encuentra el dispositivo y la velocidad con la que se desplaza. Estos datos son utilizados en sistemas de navegación y nos dan información como el estado del tráfico o un tiempo aproximado del trayecto. Actualmente, la información demandada requiere reconocimiento individualizado de los desplazamientos efectuados. Ésta se encuentra limitada dada la dificultad en la disgregación de los distintos estilos de desplazamiento por la falta de un método para detectar automáticamente el medio de transporte utilizado en cada viajes. Mediante el uso de técnicas de automatización del proceso de obtención de datos y de clasificación del medio de transporte, se incrementa el número de usuarios que contribuyen con sus aportaciones y mejora la información para satisfacer las necesidades de usuarios, gobiernos o empresas [4] [5].

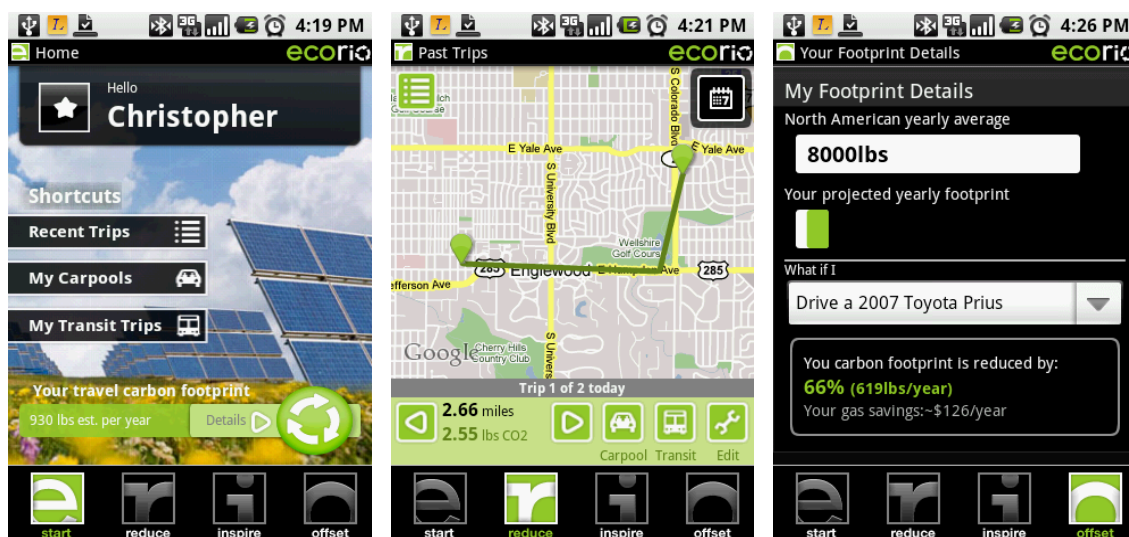
Este trabajo se encarga de la identificación automática del modo de transporte utilizado y la recogida del trayecto recorrido. Esta información puede ser utilizada para proveer de datos

interesantes al usuario, como el gasto de combustible o la emisión de  $CO_2$  del desplazamiento si se produce en vehículo, o el gasto calórico si se produce andando; pero también a entidades privadas o planificadores urbanos, que la podrán utilizar para reconocer los patrones de movimiento y evaluar la eficiencia del transporte de una ciudad, calcular la distancia media recorrida por todos los ciudadanos, estimar el medio de transporte favorito y comparar datos entre distintas ciudades.

## 1.1. Estado del Arte

Las técnicas de *urban sensing* surgen en el año 2006 y permiten obtener información de una manera novedosa que años atrás no hubiera sido posible, ya que abren la puerta a la recogida de datos de un gran número de personas [3]. En 2013 empezó a crecer el interés en la recogida de datos de transporte y de localización para crear nuevos sistemas de transporte social, donde los usuarios pudieran intercambiar información y cooperar para resolver problemas [6], y para conseguir nuevas vías de obtención de información de los viajes con los teléfonos móviles [7]. Los smartphones permiten a los investigadores la posibilidad de desarrollar aplicaciones que recogen información masiva, y abre las puertas al estudio de los comportamientos de grandes poblaciones [8]. Además, los datos recogidos con smartphones, se están convirtiendo en la fuente más importante para grandes empresas [9], y, particularizando para este estudio, en la planificación de transportes [9]. Sin embargo, hasta el momento el empleo de urban sensing no se ha consolidado como un método efectivo para la recogida de datos. Ésto puede ser debido a que las aplicaciones existentes requieren entradas de información por parte del usuario [10].

Aplicaciones como *Ecorio* y *GreenMeter* calculan la estimación de las emisiones de  $CO_2$  y consumo de combustible asociadas a los desplazamientos. Para ello, el usuario tiene que introducir el medio de transporte que va a utilizar antes de comenzar el desplazamiento, y la aplicación recoge el tiempo transcurrido en el desplazamiento y los kilómetros recorridos, como podemos observar en la Figura 1.1.



(a) Interfaz para introducir información previa.

(b) Mapa del trayecto.

(c) Información detallada.

Figura 1.1: Aplicación Ecorio.

Con el objetivo de solucionar este problema y ofrecer una alternativa definitiva para con-



solidar un método de adquisición de datos multitudinario nacen las técnicas de *social sensing*. Éstas se basan en obtener los datos sin la necesidad de ninguna entrada de información por parte del usuario, sosteniendo que es posible detectar automáticamente las actividades llevadas a cabo por el usuario, mediante el uso de los sensores y la implementación de algoritmos en el smartphone. Por ejemplo, los servicios de localización nos aportan la información necesaria para conocer el recorrido del desplazamiento. Sin embargo, clasificar automáticamente el medio de transporte utilizado no es tan sencillo, pero algunos estudios ya se han centrado en el desarrollo de aplicaciones con estas características.

*CO<sub>2</sub>GO* es una reciente aplicación cuyo objetivo es detectar automáticamente el medio de transporte del usuario para estimar su nivel de emisión de *CO<sub>2</sub>*. Para ello se basa en analizar los datos de distintos sensores del smartphone: acelerómetro, localización, GPS, micrófonos, internet, consultas a mapas, etc. (<http://senseable.mit.edu/co2go/>). En la publicación de los autores de este estudio [5] se explica como los datos adquiridos con el acelerómetro son procesados mediante una transformada rápida de Fourier (FFT) para diferenciar los perfiles de los distintos medios de transporte. Este planteamiento no es del todo correcto, ya que la frecuencia de lectura del sensor de aceleración no es exacta, presentando un jitter que depende, por ejemplo, de los procesos que este ejecutando el dispositivo. Este jitter provoca que la FFT no ofrezca un resultado correcto, ya que las muestras no estén separadas por el mismo intervalo de tiempo, como podemos observar en la Figura 1.2.

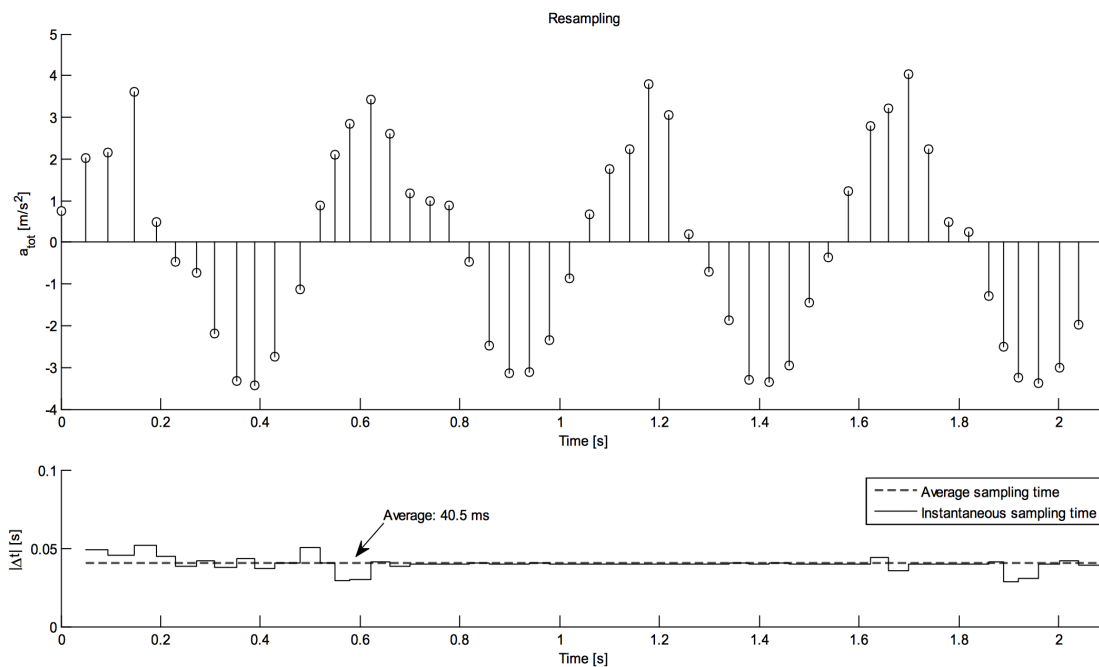


Figura 1.2: Jitter de la frecuencia de lectura de los acelerómetros.

Este hecho provoca que la precisión conseguida mediante este algoritmo deba ser mejorada aportando datos desde otros sensores: micrófono, internet, mapas y GPS (Figura 1.3), lo que se traduce en una mayor carga para el dispositivo, elevando el consumo de la batería, lo que es ya que la aplicación requiere una ejecución en segundo plano durante largos períodos de tiempo para recoger datos continuamente. Sin embargo, podemos emplear otras técnicas para reducir el impacto de la ejecución de la aplicación, como utilizar la localización aproximada basada en las torres de telefonía y las conexiones de red wifi. De esta manera conseguimos un consumo energético aceptable a costa de perder precisión en la localización.



Figura 1.3: Esquema de sensores utilizados por *CO<sub>2</sub>GO*.

Tratando de reducir el impacto de la ejecución de la aplicación en el rendimiento del dispositivo e incrementar la duración de la batería con respecto a la anterior aplicación aparece *CITYing* [4]. Esta aplicación recoge datos de los acelerómetros y de los sistemas de localización durante todo el período de ejecución. Una vez capturados, los datos son enviados a un servidor donde se efectuará el análisis de los mismos para separar las distintas actividades realizadas, y se detectará el medio de transporte mediante la media de la aceleración experimentada en los desplazamientos. De esta manera reducen la carga de trabajo del smartphone, ya que todas las operaciones necesarias para clasificar los medios de transporte no se efectúan en el teléfono móvil, sino que se realizan en el servidor. Sin embargo, la aplicación no aporta información en tiempo real, sino que lo hace a posteriori.

Estos dos estudios coinciden en que los datos aportados por los acelerómetros son claves para poder realizar la distinción entre los medios de transporte. Durante un desplazamiento cada medio de transporte (p.e., coche, autobús o tren) presenta una firma distinta (circular, vibraciones, amortiguación, o patrones de aceleración). Véase Figura 1.4. La medida de aceleración del smartphone no detecta únicamente cuándo el vehículo se pone en marcha o se detiene, sino que se puede recoger mucha más información asociada al desplazamiento que nos permita distinguir el transporte elegido por el usuario.

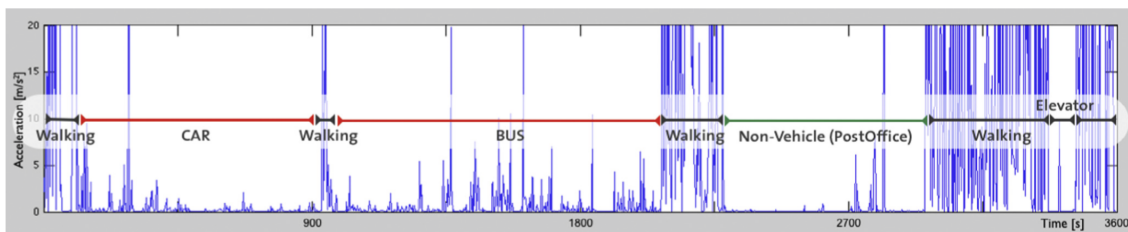


Figura 1.4: Ejemplo de aceleraciones para distintos modos de transporte.

## 1.2. Objetivos

En este Trabajo Fin de Grado se pretende definir un sistema de detección automática del medio de transporte con la información aportada por los sensores de un smartphone, que presente una clasificación óptima para el mínimo consumo de recursos del dispositivo. La importancia

de reducir el efecto de la aplicación en el rendimiento del dispositivo radica en que ésta debe ejecutarse durante largos períodos de tiempo, incluso mientras el usuario utiliza otras aplicaciones, por lo que su correcto funcionamiento no debe comprometer el uso cotidiano del dispositivo.

Se realizará una prueba de concepto que presenta los siguientes objetivos:

1. Recoger datos en desplazamientos reales. Primero, es necesario conseguir un conjunto de datos de cada medio de transporte para después trabajar con ellos y estudiar cuál es la mejor manera para diferenciarlos. Para conseguirlos, se diseñará una aplicación prototipo con la que se recogerán datos de aceleración y de localización y serán etiquetados en cada modo de transporte de manera manual.
2. Efectuar varias técnicas de procesamiento con los datos obtenidos para diferenciarlos medios de transporte. Se aplicaran distintas operaciones en los datos obtenidos para posteriormente determinar que tipo de procesamiento es el más beneficioso a la hora de diferenciar características entre distintos medios de transporte.
3. Se entrenarán varios clasificadores (SVM y Naïve-Bayes) con los datos procesados, y se efectuarán un estudio para conocer con que configuración obtenemos una mejor precisión en la clasificación.
4. Una vez conocida la configuración óptima para nuestro propósito, se implementará en una aplicación para smartphones que efectuará la clasificación del medio de transporte proporcionando al usuario y a un supervisor información útil sobre su movilidad en tiempo real.
5. Conclusiones y trabajos futuros. Se evalúa el funcionamiento de la aplicación en cuanto a la precisión de su clasificación.



## Capítulo 2

# Recogida de datos en desplazamientos reales

### 2.1. Motivación

En la primera fase de este trabajo es necesario acumular datos de desplazamientos reales en cada uno de los medios de transporte con los que trabajará la aplicación definitiva, para posteriormente realizar el entrenamiento de los clasificadores. Estos datos serán recogidos mediante una aplicación que leerá la información aportada por los sensores que implementa el propio smartphone, y deben reflejar, de manera real, los distintos comportamientos que se pueden experimentar en cada uno de los medios de transporte. Para la prueba de concepto, los datos se recogieron entre distintos usuarios, con distintos dispositivos móviles y en distintas rutas, para intentar conseguir una heterogeneidad en ellos. Los trayectos fueron realizados en las localidades de Madrid, Alcalá de Henares y Guadalajara entre marzo y junio de 2015. No obstante, a medida que la aplicación definitiva se expanda, se seguirá incrementando la base de datos utilizando la información recogida por los usuarios, y utilizándola para mejorar la clasificación de los medios de transporte.

### 2.2. Sensores del smartphone

Esta aplicación fue desarrollada para el sistema operativo iOS de Apple y programada en el entorno Xcode, en lenguaje Objective C. Fue instalada y utilizada en los siguientes modelos: iPhone 4, iPhone 4S e iPhone 5. En ella se recogen datos del acelerómetro, que ofrece tres valores de aceleración en los ejes del dispositivo según se indica en la Figura 2.1, y los servicios de localización, que nos indicarán la latitud y longitud en la que se encuentra el dispositivo, y la velocidad a la que se desplaza, de una manera aproximada. Los datos de aceleración se actualizarán con una frecuencia de 200 Hz, mientras que los de localización cada segundo. En esta aplicación no se pretende tener un consumo reducido en la batería del dispositivo, sino recoger una gran cantidad de datos de aceleración por segundo para posteriormente simular varias configuraciones con distintas frecuencias y utilizar la que permita la mejor precisión en la clasificación afectando lo mínimo posible en el comportamiento del smartphone.

### 2.2.1. Acelerómetros

Los acelerómetros son capaces de calcular la aceleración en los ejes 'x', 'y', y 'z', midiendo las fuerzas aplicadas en el dispositivo (véase Figura 2.1). Por ejemplo, cuando el teléfono móvil se encuentre en reposo encima de una mesa, la medida de aceleración será el valor de la aceleración de la gravedad ( $g = 9,81m/s^2$ ). Del mismo modo, cuando el dispositivo se encuentra en caída libre, y acelera a razón de  $9,81m/s^2$ , el acelerómetro lee una magnitud de  $g = 0m/s^2$  [4]. El acelerómetro que incorpora el iPhone nos permiten medir aceleraciones entre  $\pm 2g$ .

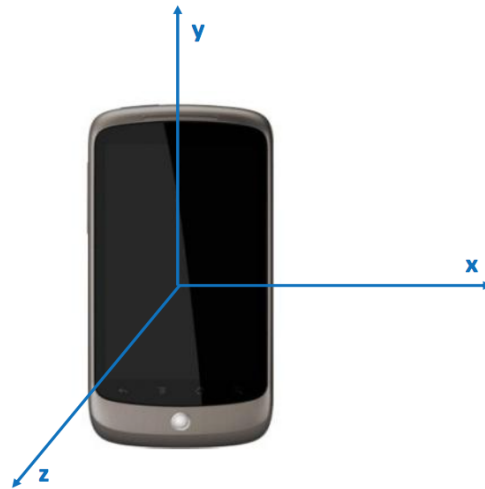


Figura 2.1: Sistema de referencia del acelerómetro

Los smartphones suelen incorporar estos sensores para determinar la orientación en la que se encuentra el dispositivo, y así poder mostrar la pantalla correctamente, bien en vertical u horizontal. Gracias a los acelerómetros se pueden registrar los patrones de aceleración asociados a los viajes en cada uno de los medios de transporte y utilizarlos para realizar la clasificación.

### 2.2.2. Servicios de localización

Dependiendo del dispositivo y de los servicios disponibles, los sistemas de localización utilizan una combinación de datos móviles, Wi-Fi, Bluetooth y GPS para determinar la ubicación del dispositivo. Si el smartphone no se encuentra dentro de la línea de visión de los satélites GPS, la ubicación se puede determinar utilizando puntos activos de conexión Wi-Fi y torres de telefonía móvil [11].

Los servicios de localización nos aportarán las coordenadas de latitud y longitud aproximadas en las que se encuentra el smartphone. De esta manera podemos determinar la ubicación en la que se encuentra, la trayectoria que sigue, y cuanta distancia se ha recorrido durante un desplazamiento. Además, calcula una aproximación de la velocidad a la que se desplaza el dispositivo, pudiendo utilizar esta información para distinguir los datos de aceleración que se recogen en parado, los desplazamientos que se realizan con un vehículo dentro de un núcleo urbano o los que se realizan en vías secundarias o autopistas.

## 2.3. Aplicación prototipo

La aplicación muestra al usuario una serie de interruptores, tal como se muestra en la Figura 2.2, que permiten elegir el medio de transporte en el que va a realizar el desplazamiento para recoger los datos y etiquetarlos de manera manual. Los tipos de desplazamiento que se han incluido en esta aplicación son: andando, coche, autobús, tren y bicicleta. Cuando se activa uno de esos interruptores se genera un fichero de texto para el modo de transporte correspondiente, donde se va escribiendo toda la información registrada durante el desplazamiento. Para cada actualización de la lectura del acelerómetro se guarda: el momento de adquisición, los datos de aceleración de cada uno de los tres ejes, latitud, longitud y velocidad, véase Figura 2.3.

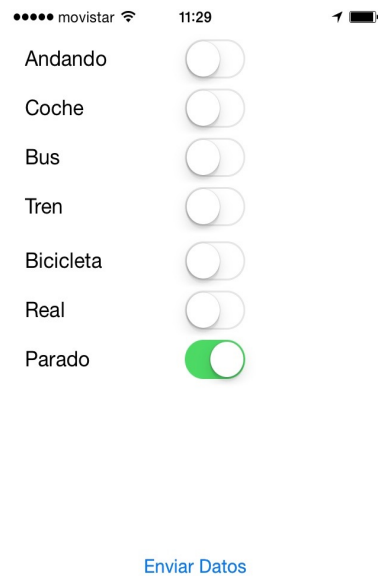


Figura 2.2: Interface de la aplicación para recoger datos.

```

20150625155755,0.043503,0.007935,0.987137,-3.35002483,40.51351377,0.8
20150625155755,0.037903,0.118942,1.103882,-3.35002483,40.51351377,0.8
20150625155755,0.028839,0.081116,1.064758,-3.35002483,40.51351377,0.8
20150625155755,0.008667,0.086243,1.112869,-3.35002483,40.51351377,0.8
20150625155755,0.126465,0.154770,1.699921,-3.35002483,40.51351377,0.8
20150625155755,0.045868,0.034622,0.967316,-3.35002483,40.51351377,0.8
20150625155755,0.088516,0.041031,1.028610,-3.35002483,40.51351377,0.8
20150625155755,0.068924,0.076889,0.946808,-3.35002483,40.51351377,0.8
20150625155755,0.059280,0.043533,1.008789,-3.35002483,40.51351377,0.8
20150625155755,0.087158,0.042023,0.998611,-3.35002483,40.51351377,0.8
20150625155755,0.076370,0.044785,0.984283,-3.35002483,40.51351377,0.8
20150625155755,0.033188,0.047043,1.003448,-3.35002483,40.51351377,0.8
20150625155756,0.000504,0.054352,1.002121,-3.35002483,40.51351377,0.8
20150625155756,0.030533,0.059692,0.997742,-3.35002483,40.51351377,0.8
20150625155756,0.026535,0.052795,1.001663,-3.35002483,40.51351377,0.8
20150625155756,0.028564,0.053802,1.005493,-3.35002483,40.51351377,0.8
20150625155756,0.019684,0.050735,0.999817,-3.35002483,40.51351377,0.8
20150625155756,0.026215,0.043060,0.997742,-3.35002483,40.51351377,0.8
20150625155756,0.027954,0.003159,0.989746,-3.35002483,40.51351377,0.8
20150625155756,0.034515,0.027588,0.997513,-3.35002483,40.51351377,0.8
20150625155756,0.033813,0.037323,0.998535,-3.35002483,40.51351377,0.8
20150625155756,0.012054,0.053558,0.999969,-3.35002483,40.51351377,0.8
20150625155756,0.009399,0.060333,1.003922,-3.35002483,40.51351377,0.8
20150625155756,0.000992,0.069916,1.004150,-3.35002483,40.51351377,0.8
20150625155756,0.009995,0.046707,1.003845,-3.35002483,40.51351377,0.8
20150625155756,0.034088,0.011017,0.997437,-3.35002483,40.51351377,0.8

```

Figura 2.3: Datos recogidos por la aplicación

Estos documentos son guardados en la memoria del smartphone, y pueden ser enviados fácilmente a una base de datos a través de un servidor. Para ello la aplicación dispone de un botón para enviar los datos recogidos, que ejecuta una función donde se comprimen todos los documentos almacenados y se comprueba la conectividad para asegurar que los datos son enviados correctamente al servidor. Una vez enviados son eliminados de la memoria del dispositivo. Todos estos datos son enviados y almacenados de manera anónima, ya que el servidor asigna una ID aleatoria a cada usuario para proteger su privacidad. Estas cuestiones serán desarrolladas en profundidad en el capítulo correspondiente al servidor.



# Capítulo 3

## Procesamiento de datos

### 3.1. Motivación

En este capítulo se describirán las distintas técnicas de procesamiento utilizadas con los datos obtenidos mediante la aplicación prototipo, con la finalidad de diferenciar o separar en la medida de lo posible distintas características asociadas a cada medio de transporte, para posteriormente utilizarlas al entrenar los clasificadores y determinar cual es la técnica más eficaz para diferenciar los distintos tipos de desplazamientos contemplados. Los datos recogidos con la aplicación prototipo son datos en RAW, necesitaremos aplicarle un procesamiento inicial a los datos de aceleración para quedarnos con la información relevante, y después utilizaremos distintas técnicas para separar las características de los diferentes modos de desplazamiento.

### 3.2. Tratamiento inicial

#### 3.2.1. Filtro paso alto

Frente a otras aplicaciones que únicamente permiten el funcionamiento colocados en una posición fija, esta aplicación permite que el dispositivo pueda ser portado en cualquier orientación, ya que está pensada para ejecutarse en situaciones cotidianas, por ejemplo cuando llevamos nuestro teléfono móvil en el bolsillo, en la mano, o en un bolso. Esto significa que la fuerza de la gravedad se repartirá de manera desconocida entre los distintos datos de aceleración de los ejes del dispositivo, y no es posible considerar su contribución como una constante. En el enfoque de este trabajo, lo importante es extraer los datos de aceleración correspondientes únicamente al comportamiento experimentado en cada medio de transporte, por lo que, la contribución de la fuerza de la gravedad debe ser eliminada de las lecturas del acelerómetro antes de realizar los procesamientos con los datos. Este problema se soluciona mediante el empleo un filtro paso alto que permite atenuar las contribuciones constantes y almacenar únicamente las aceleraciones asociadas a los movimientos o vibraciones que se recogen en los desplazamientos. En la Figura 3.1 se puede observar cómo la aportación de la gravedad en la lectura del acelerómetro (parte superior) con un valor de  $-1g$  en el eje representado en color azul es eliminada por el efecto del filtro paso alto (parte inferior) donde las aceleraciones de los tres ejes se encuentran centradas en  $0g$ , recogiendo los datos de aceleración de movimientos o vibraciones ocasionales que tienen lugar en cualquier medio de transporte.



Figura 3.1: Datos del acelerómetro reales vs. datos pasado por el filtro paso alto

### 3.2.1.1. Definición

Es un filtro electrónico que permite el paso de señales con una frecuencia superior a una frecuencia de corte establecida mediante el diseño del filtro, y atenúa las señales que presenten una frecuencia inferior a esa frecuencia de corte. Se puede usar, por ejemplo, para bloquear el paso de la corriente continua a un circuito. En este estudio, una aportación a la aceleración recogida que permanece constante, como la de la gravedad, se considera irrelevante para el estudio y debe ser filtrada.

### 3.2.1.2. Filtro paso alto en tiempo discreto

A pesa de que la lectura de los sensores puede presentar un pequeño jitter, en el caso de un filtro paso alto con una frecuencia de corte muy baja, el error asumido en el filtrado es insignificante. Consecuentemente, por simplicidad se asume que las muestras obtenidas son equiespaciadas mediante un intervalo ( $dt$ ) que será la inversa de la frecuencia de actualización de la lectura de los acelerómetros ( $kUpdateFrequency$ ), como se puede observar en la Ecuación 3.1.

$$dt = \frac{1}{kUpdateFrequency} \quad (3.1)$$

La constante de tiempo del filtro  $RC$  se calcula como la inversa de la frecuencia de corte del filtro ( $cutOffFrequency$ ). Véase Ecuación 3.2.

$$RC = \frac{1}{cutOffFrequency} \quad (3.2)$$

Si se representan las entradas al filtro con la secuencia  $(x_1, x_2, \dots, x_n)$  y las salidas con la secuencia  $(y_1, y_2, \dots, y_n)$  el funcionamiento del filtro se puede expresar mediante la siguiente ecuación [12]:

$$y_i = RC \left( \frac{x_i - x_{i-1}}{dt} - \frac{y_i - y_{i-1}}{dt} \right)$$

Reordenando:

$$y_i = \frac{RC}{RC + dt} y_{i-1} + \frac{RC}{RC + dt} (x_i - x_{i-1})$$

Por tanto, podemos decir que la implementación de un filtro paso alto en tiempo discreto se puede expresar mediante la Ecuación 3.3.

$$y_i = \alpha y_{i-1} + \alpha (x_i - x_{i-1}) \quad (3.3)$$

Donde:

$$\alpha = \frac{RC}{RC + dt}$$

### 3.2.2. Almacenamiento de datos

Una vez filtrados los datos de aceleración procedemos, el siguiente paso es crear una matriz con todos los datos de aceleración y de localización recogidos y preprocesados. Se calcula un valor de aceleración total del dispositivo para cada lectura del acelerómetro, mediante la raíz cuadrada de la suma de cuadrados de la aceleración de cada uno de los ejes [5] [4]. Véase la Ecuación 3.4. De esta forma logramos independizar el resultado de la posición del smartphone, ya que se se trabaja con la aportación de todas las componentes de aceleración al cuadrado, evitando de esta manera también el signo de la aceleración acumulada.

$$a_t(t) = \sqrt{a_x(t)^2 + a_y(t)^2 + a_z(t)^2} \quad (3.4)$$

La velocidad se utiliza para diferenciar los desplazamientos en coche y autobús dentro de ciudad o en el resto de carreteras (autovías, carreteras secundarias). De esta manera, si la velocidad es inferior a 60 km/h el los datos se etiquetan como un desplazamiento en coche o autobús urbano, mientras que si supera los 70 km/h será etiquetado como un desplazamiento interurbano. Una vez finalizado el almacenamiento, obtenemos una matriz compuesta por los datos de aceleración total y la velocidad para los distintos tipos de transporte: andando, coche en ciudad, coche en autovía, autobús en ciudad, autobús en autovía y tren. Como estos datos posteriormente serán utilizados para entrenar los clasificadores, nos quedamos con el mismo número de muestras para cada una de las clases, obteniendo un conjunto de aproximadamente diez minutos para cada clase de transporte a detectar. En la figura 3.2 podemos observar una representación de los datos de aceleración y velocidad de las distintas clases después del preprocesamiento.

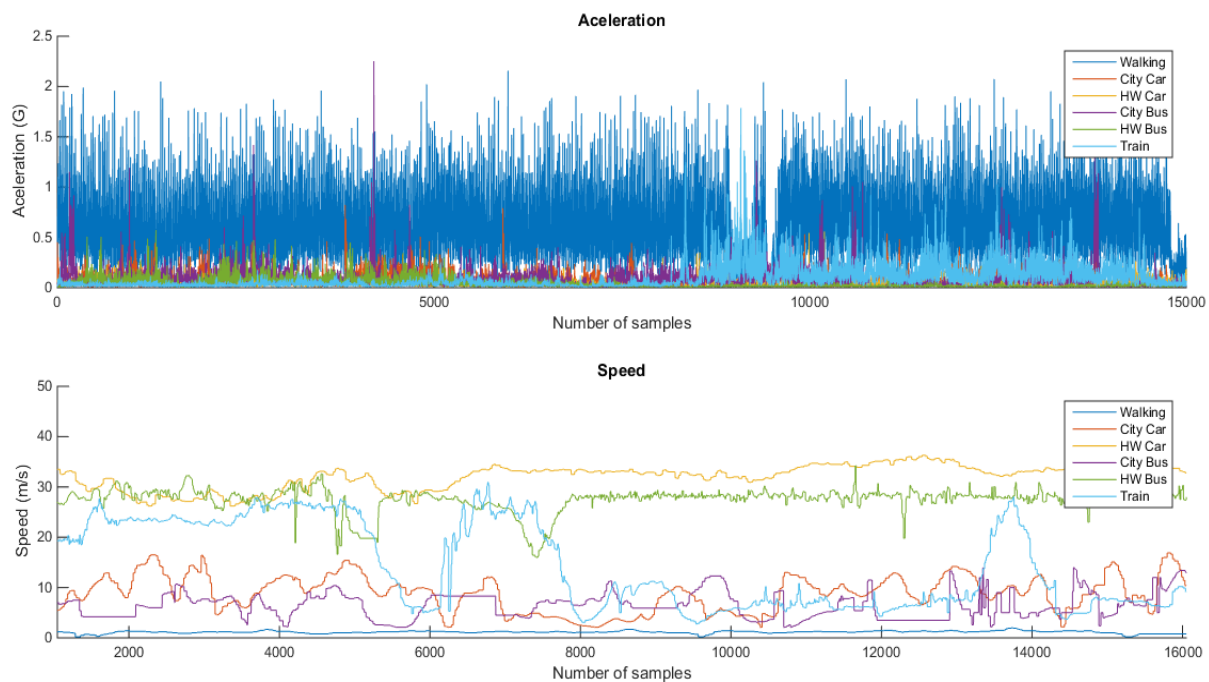


Figura 3.2: Representación de los datos de aceleración y velocidad de las distintas clases

### 3.3. Técnicas de procesamiento aplicadas

Concluido el preprocesamiento y almacenados los datos relevantes, el siguiente paso es aplicar diferentes operaciones matemáticas para diferenciar varias características y poder entrenar con ellas los clasificadores para comprobar de que manera se obtienen las predicciones más precisas. Se calcularán cuatro características en cada procesamiento, dos para los datos de aceleración y dos para los de velocidad. Las características obtenidas tomarán como base los valores de media y desviación estándar de una determinada serie temporal.

- Media [13].

Hace referencia al valor medio de un conjunto de datos discretos y se obtiene calculando la suma de todos los valores y dividiendo entre el número de sumandos. Si se dispone de un conjunto de valores  $x_1, x_2, \dots, x_n$ , la media de estos se expresa en la Ecuación 3.5.

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i \quad (3.5)$$

- Desviación Estándar [14].

Es una medida utilizada para cuantificar la variación o dispersión de un conjunto de datos. Si la desviación estándar es próxima a cero significa que el conjunto de datos es muy cercano a la media, mientras que un elevado valor de la desviación estándar significa que los datos están más separados. La desviación estándar de un conjunto de datos se calcula mediante la raíz cuadrada de su varianza. Si se dispone del conjunto de datos  $x_1, x_2, \dots, x_n$ , la desviación estándar se define como se muestra en la Ecuación 3.6.

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n |x_i - \mu|^2} \quad (3.6)$$

Donde  $\mu$  es la media del conjunto de valores.

Se define la longitud de una ventana temporal que contenga un número de muestras (*nsamples*) con las que realizamos la operación de cada técnica, y el número de muestras que avanza esta ventana en cada iteración (*ngap*). Estas dos variables pueden ser modificadas para obtener la configuración con mayor porcentaje de éxito en la clasificación. Para cada una de las técnicas de procesamiento empleadas, representaremos los valores de entrada como el conjunto  $x_1, x_2, \dots, x_n$  y las salidas como el conjunto  $y_1, y_2, \dots, y_m$ .

#### 3.3.1. Promedio

Se calcula la media de los datos de aceleración y velocidad del conjunto de muestras (*nsamples*) incluidos en la ventana y repetimos este proceso a lo largo de todo el conjunto de datos. En cada una de las iteraciones la ventana se desplaza el número de muestras definido con *ngap*. Véase Ecuación 3.7.

$$y_1 = \frac{1}{n_{samples}} \sum_{i=1}^{n_{samples}} x_i$$

$$\begin{aligned}
y_2 &= \frac{1}{n_{samples}} \sum_{i=n_{gap}+1}^{n_{gap}+n_{samples}} x_i \\
&\vdots \\
y_m &= \frac{1}{n_{samples}} \sum_{i=(m-1)\cdot n_{gap}+1}^{(m-1)\cdot n_{gap}+n_{samples}} x_i
\end{aligned} \tag{3.7}$$

Realizamos el mismo procedimiento para el cálculo de las desviaciones estándar. Véase Ecuación 3.8.

$$\begin{aligned}
y_1 &= \sqrt{\frac{1}{n_{samples} - 1} \sum_{i=1}^{n_{samples}} |x_i - \mu|^2} \\
y_2 &= \sqrt{\frac{1}{n_{samples} - 1} \sum_{i=n_{gap}+1}^{n_{gap}+n_{samples}} |x_i - \mu|^2} \\
&\vdots \\
y_m &= \sqrt{\frac{1}{n_{samples} - 1} \sum_{i=(m-1)\cdot n_{gap}+1}^{(m-1)\cdot n_{gap}+n_{samples}} |x_i - \mu|^2}
\end{aligned} \tag{3.8}$$

Los resultados que obtenemos con este procesamiento se representan en la Figura 3.3, donde se muestran los valores de aceleración media, velocidad media, y las desviaciones estándar de los valores de aceleración y velocidad. Esta simulación se configuró con una frecuencia de adquisición de  $kUpdateFrequency = 50Hz$ , un conjunto de muestras recogidos durante 60 segundos  $n_{samples} = kUpdateFrequency \cdot 60$  y en cada iteración desplazamos la ventana 1 segundo  $n_{gap} = kUpdateFrequency$ . Las distribuciones de probabilidad gaussianas asociadas a estas cuatro características se muestran en la Figura 3.4.

### 3.3.2. Multiplicaciones

Se multiplica la media de dos ventanas de longitud  $n_{samples}$  desplazadas una con respecto a la otra  $n_{gap}$ , véase Ecuación 3.9

$$\begin{aligned}
y_1 &= \frac{1}{n_{samples}} \sum_{i=1}^{n_{samples}} x_i \cdot \frac{1}{n_{samples}} \sum_{i=n_{gap}+1}^{n_{gap}+n_{samples}} x_i \\
y_2 &= \frac{1}{n_{samples}} \sum_{i=n_{gap}+1}^{n_{gap}+n_{samples}} x_i \cdot \frac{1}{n_{samples}} \sum_{i=2\cdot n_{gap}+1}^{2\cdot n_{gap}+n_{samples}} x_i \\
&\vdots \\
y_m &= \frac{1}{n_{samples}} \sum_{i=(m-1)\cdot n_{gap}+1}^{(m-1)\cdot n_{gap}+n_{samples}} x_i \cdot \frac{1}{n_{samples}} \sum_{i=m\cdot n_{gap}+1}^{m\cdot n_{gap}+n_{samples}} x_i
\end{aligned} \tag{3.9}$$

Realizamos el mismo procedimiento con las desviaciones estándar de las ventanas, véase Ecuación 3.10.

$$\begin{aligned}
y_1 &= \sqrt{\frac{1}{n_{samples} - 1} \sum_{i=1}^{n_{samples}} |x_i - \mu|^2} \cdot \sqrt{\frac{1}{n_{samples} - 1} \sum_{i=n_{gap}+1}^{n_{gap}+n_{samples}} |x_i - \mu|^2} \\
y_2 &= \sqrt{\frac{1}{n_{samples} - 1} \sum_{i=n_{gap}+1}^{n_{gap}+n_{samples}} |x_i - \mu|^2} \cdot \sqrt{\frac{1}{n_{samples} - 1} \sum_{i=n_{gap}+1}^{n_{gap}+n_{samples}} |x_i - \mu|^2} \\
&\quad \vdots \\
y_m &= \sqrt{\frac{1}{n_{samples}} \sum_{i=(m-1) \cdot n_{gap}+1}^{(m-1) \cdot n_{gap}+n_{samples}} x_i} \cdot \sqrt{\frac{1}{n_{samples}} \sum_{i=2 \cdot n_{gap}+1}^{2 \cdot n_{gap}+n_{samples}} x_i} \quad (3.10)
\end{aligned}$$

Podemos observar los resultados de este proceso en la Figura 3.5. En este caso la frecuencia de adquisición de valores de aceleración sigue siendo  $kUpdateFrequency = 50Hz$ , mientras que la ventana contiene datos de cinco segundos  $n_{mean} = kUpdateFrequency \cdot 5$  y se va desplazando un segundo en cada iteración  $n_{gap} = kUpdateFrequency$ . Las distribuciones de densidad gaussiana se muestran en la Figura 3.6.

### 3.3.3. Integral

Se calcula la media de dos ventanas de longitud  $n_{samples}$  con un desfase de  $n_{gap}$  y se suman. Véase Ecuación 3.11.

$$\begin{aligned}
y_1 &= \frac{1}{n_{samples}} \sum_{i=1}^{n_{samples}} x_i + \frac{1}{n_{samples}} \sum_{i=n_{gap}+1}^{n_{gap}+n_{samples}} x_i \\
y_2 &= \frac{1}{n_{samples}} \sum_{i=n_{gap}+1}^{n_{gap}+n_{samples}} x_i + \frac{1}{n_{samples}} \sum_{i=2 \cdot n_{gap}+1}^{2 \cdot n_{gap}+n_{samples}} x_i \\
&\quad \vdots \\
y_m &= \frac{1}{n_{samples}} \sum_{i=(m-1) \cdot n_{gap}+1}^{(m-1) \cdot n_{gap}+n_{samples}} x_i + \frac{1}{n_{samples}} \sum_{i=m \cdot n_{gap}+1}^{m \cdot n_{gap}+n_{samples}} x_i \quad (3.11)
\end{aligned}$$

Realizamos el mismo proceso con las desviaciones estándar de los datos de las ventanas, véase la Ecuación 3.12.

$$\begin{aligned}
y_1 &= \sqrt{\frac{1}{n_{samples} - 1} \sum_{i=1}^{n_{samples}} |x_i - \mu|^2} + \sqrt{\frac{1}{n_{samples} - 1} \sum_{i=n_{gap}+1}^{n_{gap}+n_{samples}} |x_i - \mu|^2} \\
y_2 &= \sqrt{\frac{1}{n_{samples} - 1} \sum_{i=n_{gap}+1}^{n_{gap}+n_{samples}} |x_i - \mu|^2} + \sqrt{\frac{1}{n_{samples} - 1} \sum_{i=n_{gap}+1}^{n_{gap}+n_{samples}} |x_i - \mu|^2}
\end{aligned}$$

$$\begin{aligned}
& \vdots \\
y_m = & \sqrt{\frac{1}{n_{samples}} \sum_{i=(m-1) \cdot n_{gap}+1}^{(m-1) \cdot n_{gap}+n_{samples}} x_i} + \sqrt{\frac{1}{n_{samples}} \sum_{i=2 \cdot n_{gap}+1}^{2 \cdot n_{gap}+n_{samples}} x_i} \quad (3.12)
\end{aligned}$$

En la simulación se configuran los parámetros  $kUpdateFrquency = 50Hz$ ,  $n_{samples} = kUpdateFrquency \cdot 30$  y  $n_{gap} = kUpdateFrquency \cdot 1$  y obtenemos las características representadas en la Figura 3.7. Las distribuciones normales de estas características se muestran en la Figura 3.8.

### 3.3.4. Diferencial

En este caso, se resta la media de dos ventanas de longitud  $n_{samples}$  con un desfase de  $n_{gap}$ . Véase Ecuación 3.9.

$$\begin{aligned}
y_1 = & \frac{1}{n_{samples}} \sum_{i=1}^{n_{samples}} x_i - \frac{1}{n_{samples}} \sum_{i=n_{gap}+1}^{n_{gap}+n_{samples}} x_i \\
y_2 = & \frac{1}{n_{samples}} \sum_{i=n_{gap}+1}^{n_{gap}+n_{samples}} x_i - \frac{1}{n_{samples}} \sum_{i=2 \cdot n_{gap}+1}^{2 \cdot n_{gap}+n_{samples}} x_i \\
& \vdots \\
y_m = & \frac{1}{n_{samples}} \sum_{i=(m-1) \cdot n_{gap}+1}^{(m-1) \cdot n_{gap}+n_{samples}} x_i - \frac{1}{n_{samples}} \sum_{i=m \cdot n_{gap}+1}^{m \cdot n_{gap}+n_{samples}} x_i \quad (3.13)
\end{aligned}$$

Realizamos el mismo procedimiento con las desviaciones estándar de las ventanas, véase Ecuación 3.14.

$$\begin{aligned}
y_1 = & \sqrt{\frac{1}{n_{samples} - 1} \sum_{i=1}^{n_{samples}} |x_i - \mu|^2} - \sqrt{\frac{1}{n_{samples} - 1} \sum_{i=n_{gap}+1}^{n_{gap}+n_{samples}} |x_i - \mu|^2} \\
y_2 = & \sqrt{\frac{1}{n_{samples} - 1} \sum_{i=n_{gap}+1}^{n_{gap}+n_{samples}} |x_i - \mu|^2} - \sqrt{\frac{1}{n_{samples} - 1} \sum_{i=n_{gap}+1}^{n_{gap}+n_{samples}} |x_i - \mu|^2} \\
& \vdots \\
y_m = & \sqrt{\frac{1}{n_{samples}} \sum_{i=(m-1) \cdot n_{gap}+1}^{(m-1) \cdot n_{gap}+n_{samples}} x_i} - \sqrt{\frac{1}{n_{samples}} \sum_{i=2 \cdot n_{gap}+1}^{2 \cdot n_{gap}+n_{samples}} x_i} \quad (3.14)
\end{aligned}$$

En este caso en la Figura 3.9 se representa una simulación con una  $kUpdateFrquency = 50Hz$ , longitud de ventana  $n_{samples} = kUpdateFrquency$  y  $n_{gap} = kUpdateFrquency$ . Las distribuciones normales se representan en la Figura 3.10

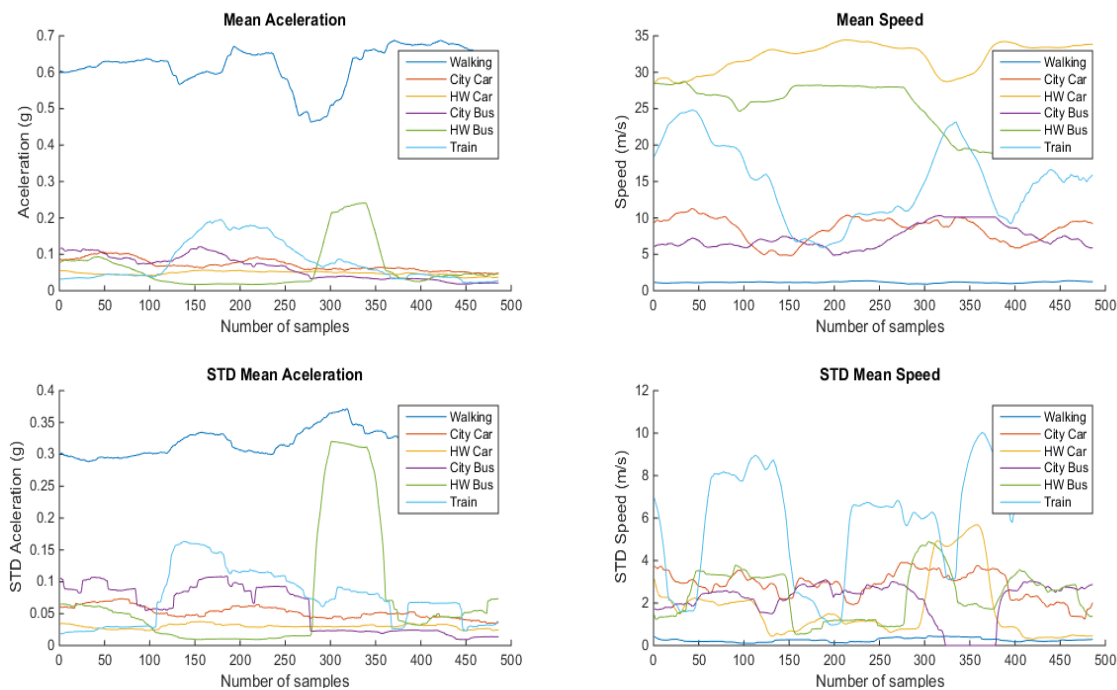


Figura 3.3: Promedio: representación de las características calculadas.

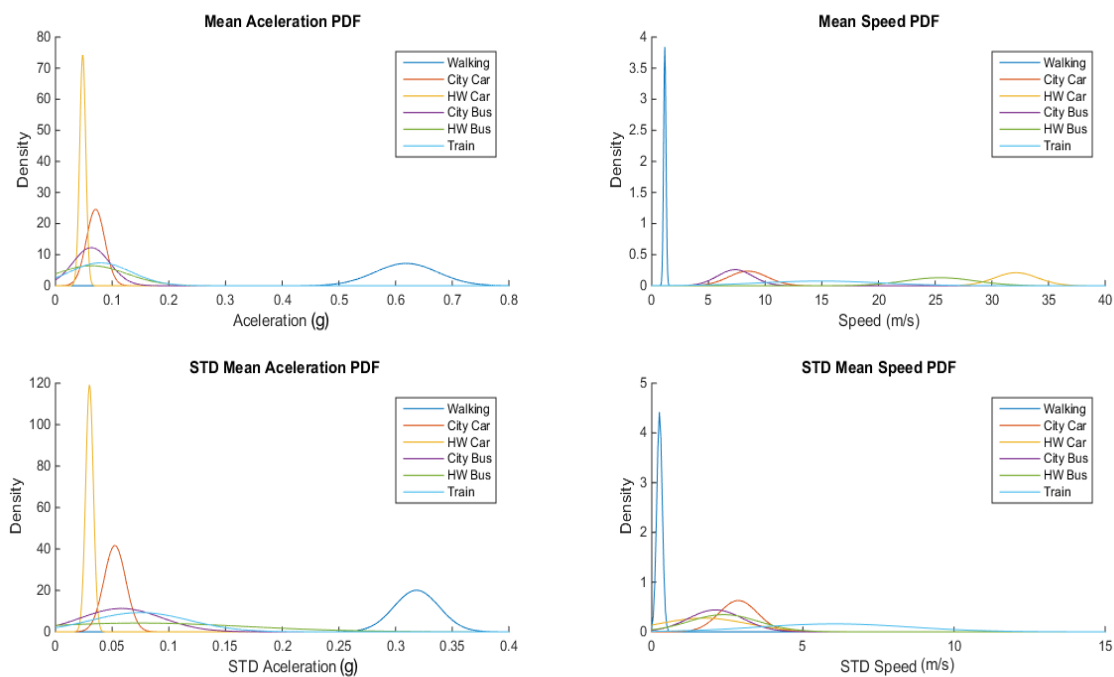


Figura 3.4: Promedio: distribuciones de probabilidad de densidad gaussiana de las características calculadas.



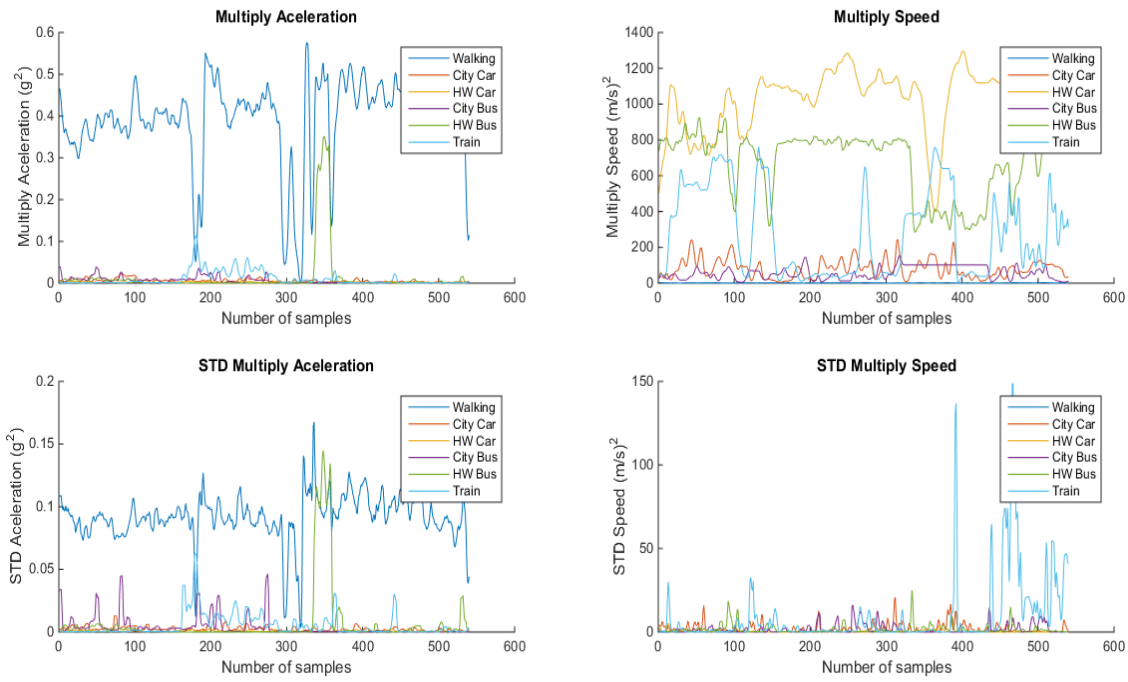


Figura 3.5: Multiplicaciones: representación de las características calculadas.

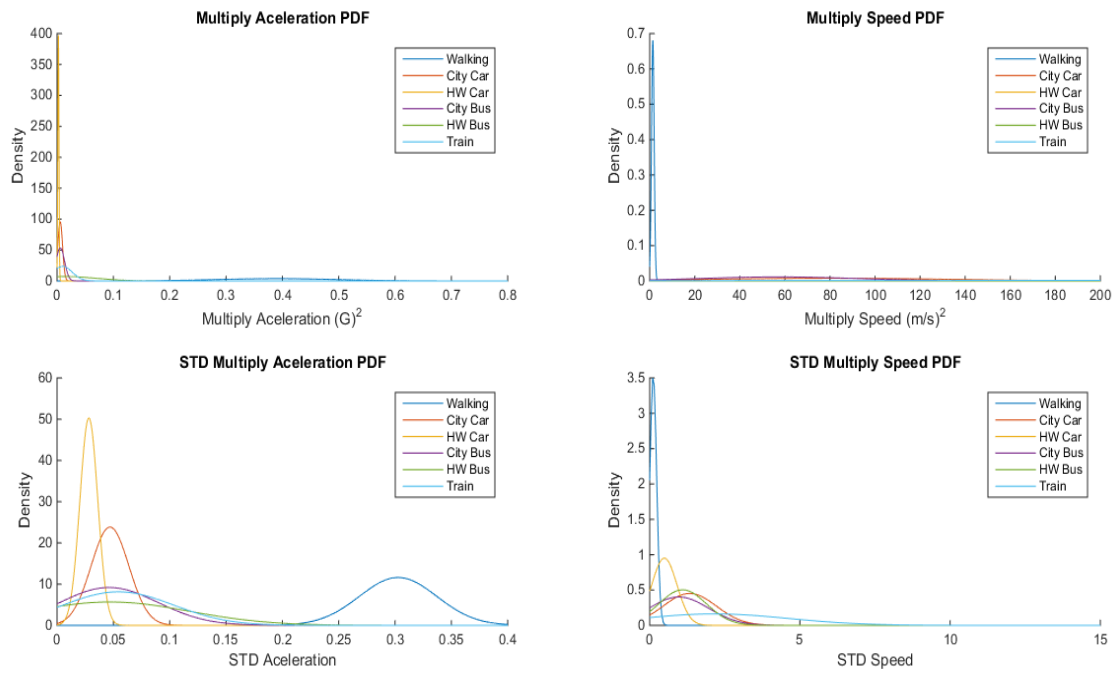


Figura 3.6: Multiplicaciones: distribuciones de probabilidad de densidad gaussiana de las características calculadas.

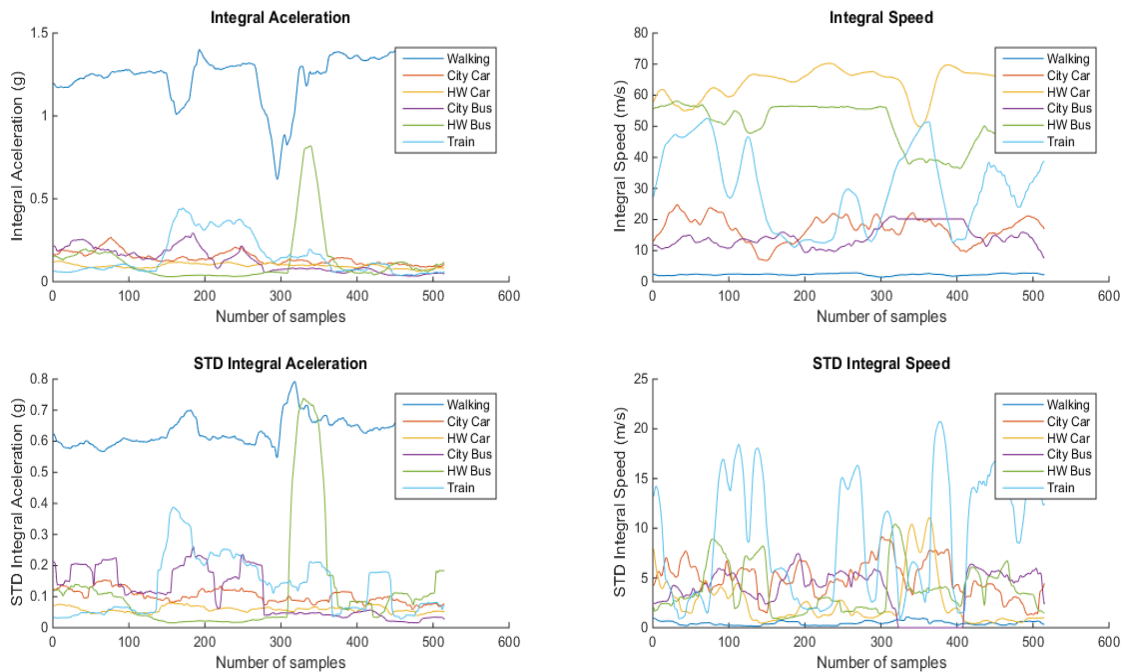


Figura 3.7: Integración: representación de las características calculadas.

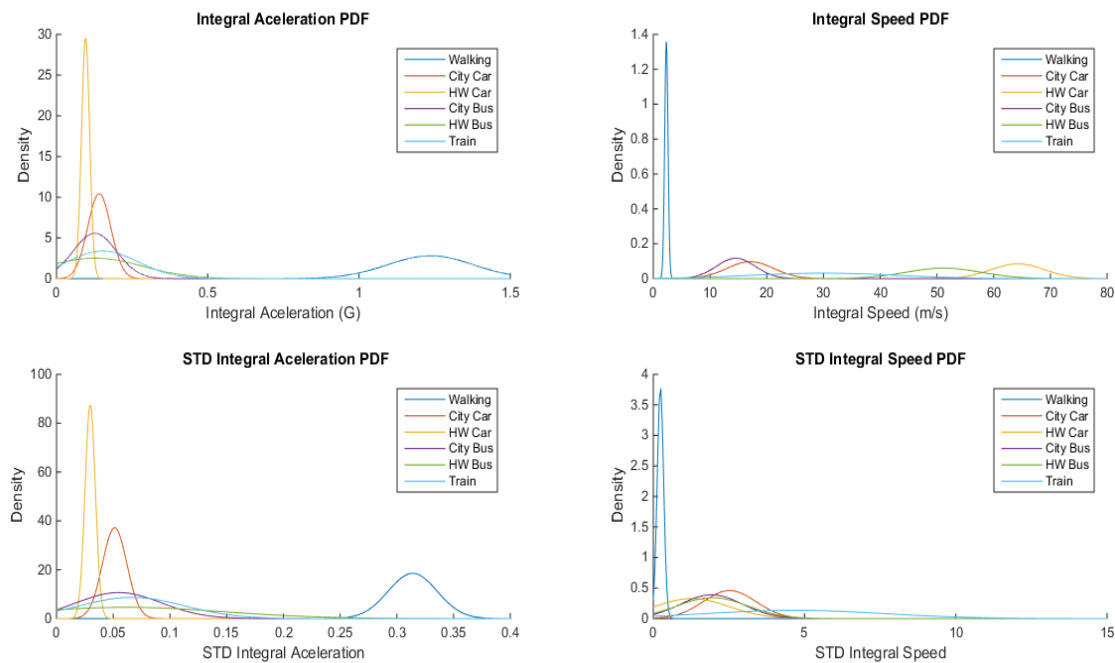


Figura 3.8: Integración: distribuciones de probabilidad de densidad gaussiana de las características calculadas.

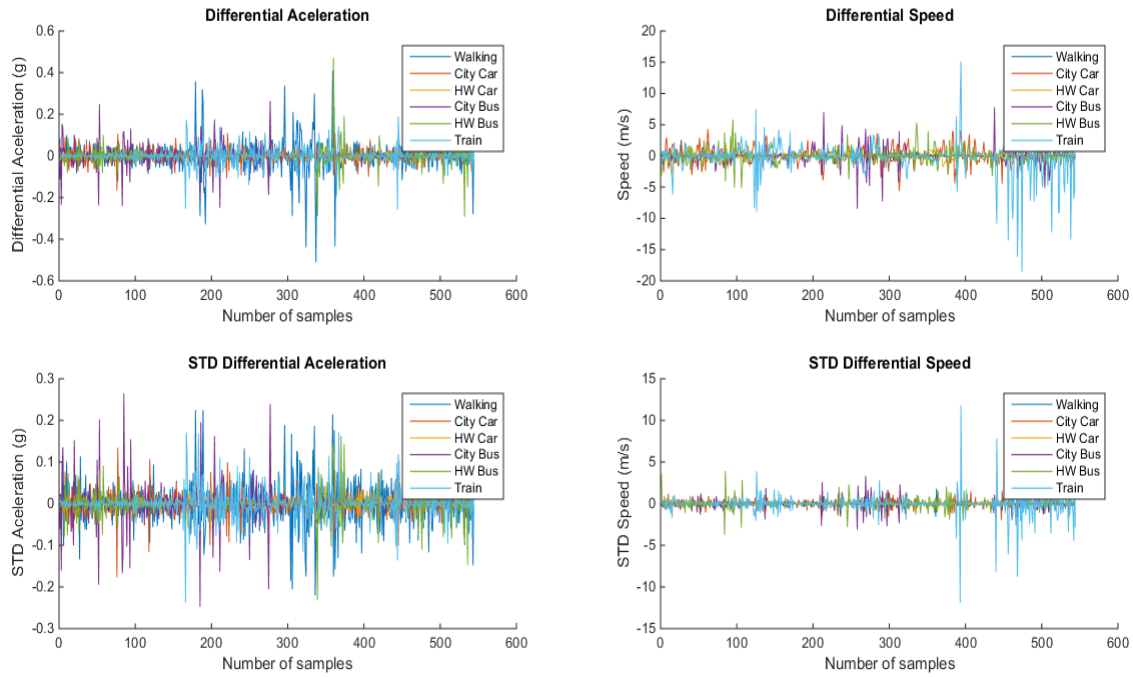


Figura 3.9: Derivación: representación de las características calculadas.

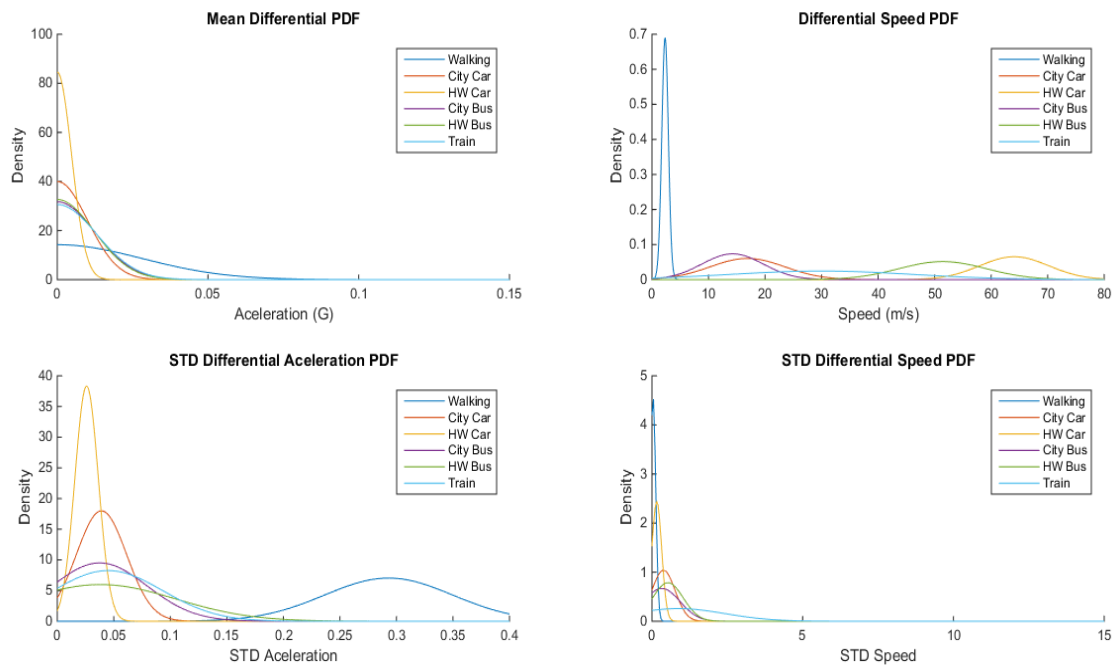


Figura 3.10: Derivación: distribuciones de probabilidad de densidad gaussiana de las características calculadas.

### 3.4. Conclusiones

Estas gráficas muestran que las características de aceleración recogidas en los desplazamientos efectuados en vehículo se encuentran bastante próximas, pero se observan distintas medias y desviaciones en cada una de las clases. Los datos de velocidad distinguen de manera evidente cuando los desplazamientos se producen dentro de la ciudad o en carreteras interurbanas. A priori, los desplazamientos que se llevaron a cabo andando son los que mejor se diferencian del resto para cualquiera de las características, ya que su media de aceleración es notablemente superior al del resto de modos de transporte, mientras que la velocidad es claramente inferior, mientras que las clases de datos con mayor dificultad para distinguir son los correspondientes a coche dentro de ciudad y autobús dentro de ciudad, ya que presentan las características con mayor superposición en las distintas gráficas. No obstante, al utilizar estas características juntas para entrenar a los clasificadores se puede conseguir representaciones en más de 3 dimensiones que pueden separar las diferencias entre estas clases de datos.

# Capítulo 4

## Clasificadores

### 4.1. Motivación

Las técnicas de *Machine Learning* están orientadas a permitir la toma de decisiones por parte de los computadores. Se trata de crear algoritmos de los que se puede extraer información y hacer predicciones. Utiliza el razonamiento estadístico para encontrar soluciones aproximadas para las predicciones.

En este trabajo estudiaremos la eficacia en la disgregación de los datos en las distintas clases de transporte de dos tipos de clasificadores: SVM y Naïve-Bayes, dado que son dos clasificadores ampliamente utilizados en este tipo de aplicaciones por su alto poder discriminatorio y su facilidad de entrenamiento; se escogerá el que ofrezca mayor precisión. Para ello, separamos todos los datos que obtenemos en el apartado anterior en dos grupos: uno para entrenamiento supervisado, y el otro para hacer tests. Con este procedimiento se puede medir la precisión en la clasificación de los distintos clasificadores entrenados.

### 4.2. Support Vector Machines

Usamos *Support Vector Machine (SVM)* cuando queremos diferenciar dos clases con una gran cantidad de datos y gran ruido de medida. Mediante esta técnica de clasificación se representan los datos en el hiperespacio en función de un conjunto de características. Cada una de estas características se traduce en una dimensión más, esto posibilita que las distintas clases se separen y sea más precisa la clasificación de los dos tipos de datos. Para la clasificación, SVM busca el hiperplano consiga una mejor separación entre las dos clases, es decir, el que presente una mayor distancia o margen, para ello utiliza únicamente un conjunto limitado de datos que se encuentran en la frontera de separación de las clases conocidos como *support vectors*. (Figura 4.1)

Los datos para entrenamiento son un conjunto de vectores  $x_i$  pertenecientes a su clase  $y_i$ . Para  $d$  dimensiones, en función del número de características que se incluyen en cada vector, los vectores  $x_i \in R^d$  y las clases  $y_i = \pm 1$ . La ecuación del hiperplano es

$$\langle w, x \rangle + b = 0,$$

donde  $w \in R^d$  y es el margen de separación entre las clases,  $\langle w, x \rangle$  es el producto escalar de  $w$  y  $x$ , y  $b$  es un número real [15] [16].

El mejor hiperplano de separación se obtendrá al encontrar  $w$  y  $b$  que minimicen  $\|w\|$  tal

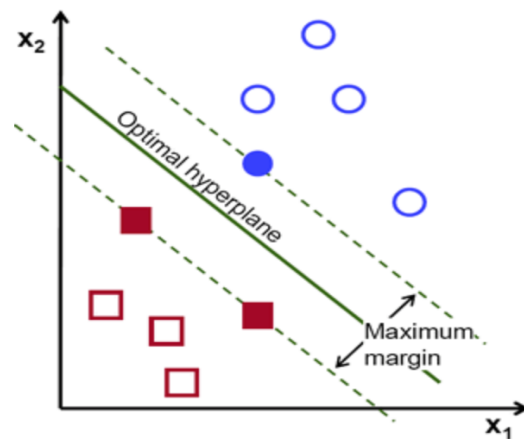


Figura 4.1: Ejemplo de SVM en 2D

que para todos los puntos de datos  $(x_i, y_i)$

$$y_i(\langle w, x_i \rangle + b) \geq 1.$$

Los support vectors son los  $x_i$  que se encuentran en el límite del hiperplano de su clase, aquellos para los que  $y_i(\langle w, x_i \rangle + b) = 1$ .

Esta es la forma más sencilla para separar dos clases con SVM, sin embargo, no siempre es posible este tipo de delimitación, ya que algunos conjuntos de datos no pueden ser completamente separados mediante hiperplanos o se usan clasificadores de más de dos categorías (multiclase). Para solucionar estos problemas utilizamos las funciones *Kernel*, que proyectan la información en un espacio de características de mayor dimensión, lo que aumenta la capacidad computacional de las máquinas de aprendizaje lineal.

$$x = x_1, x_2, \dots, x_n \rightarrow \varphi(x) = \varphi(x_1), \varphi(x_2), \dots, \varphi(x_n)$$

A continuación se muestran los kernels más utilizados:

- Polinomial-homogénea: Utiliza combinaciones de las diferentes características introducidas al vector para la clasificación. (Figura 4.2).

$$K(x_i, x_j) = (x_i \cdot x_j)^n$$

- Multilayer Perceptron (MLP) (Figura 4.3):

$$K(x_i, x_j) = \|x_i - x_j\|$$

- Función de base radial Gaussiana (RBF) (Figura 4.4):

$$K(x_i, x_j) = e^{-\frac{(x_i - x_j)^2}{2\sigma^2}}$$

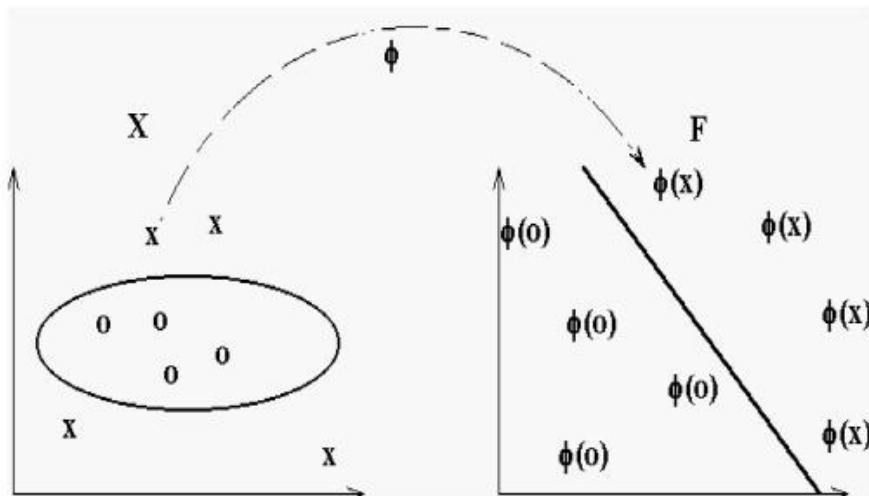


Figura 4.2: Función Kernel Polinomial

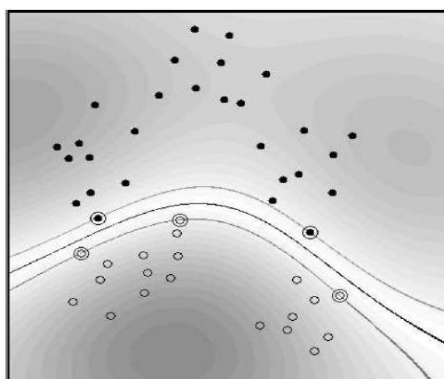


Figura 4.3: Función Kernel MLP

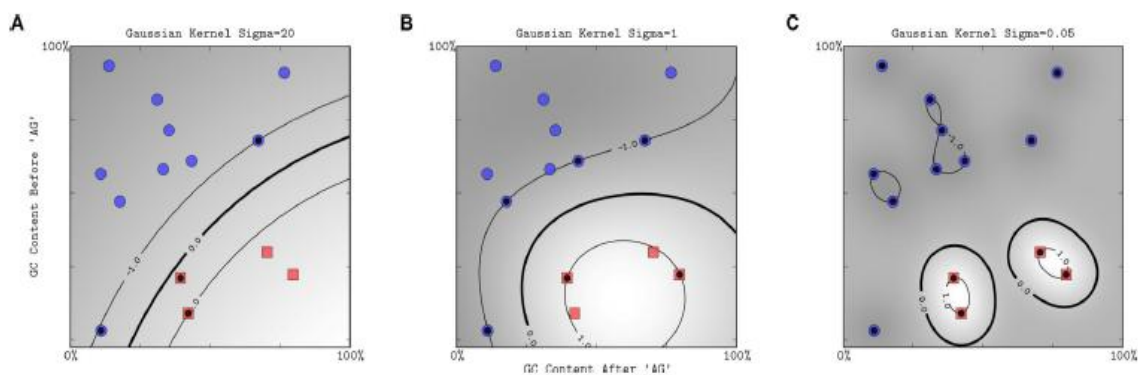


Figura 4.4: Función Kernel RBF

Como nuestra aplicación va a trabajar con varios medios de transporte, necesitamos utilizar un SVM multiclase. En nuestro caso, utilizamos una clasificación denominada *uno contra resto*, esto significa que utilizamos tantos SVM biclase como número de medios de transporte queremos detectar. De esta manera, cada clasificador separa la región de datos de un medio de transporte, que será la clase positiva, frente al conjunto de regiones de datos del resto de medios de transporte, la clase negativa. Para cada vector incluimos las características calculadas en el apartado de procesamientos para velocidad y aceleración, por tanto trabajamos con clasificadores SVM con un vector de medida de cuatro dimensiones.

### 4.3. Clasificador Naïve Bayes

El clasificador Naïve Bayes se emplea en casos en los que las clases tienen una gran dispersión respecto de la media y además hay solape entre clases. Su principal virtud radica en que se asume que las variables son aleatorias con lo que tienen una mayor capacidad para clasificar clases solapadas. El *teorema de bayes* es el soporte matemático sobre el que se apoya este clasificador:

$$p(\omega_i/X) = \frac{p(X/\omega_i)p(\omega_i)}{p(X)} \quad (4.1)$$

- Probabilidad a posteriori ( $p(\omega_i/X)$ ) de la hipótesis  $\omega_i$  habiéndose medido  $X$  (evidencia).
- Probabilidad condicionada ( $p(X/\omega_i)$ ): probabilidad de que dada  $\omega_i$  el valor de la variable aleatoria sea  $X$ , es decir es la función densidad de probabilidad (pdf) de la clase  $\omega_i$ .
- Probabilidad a priori ( $p(\omega_i)$ ) de que se presente un elemento de la clase  $\omega_i$ .
- $p(X)$  Probabilidad de que se presente un elemento a clasificar con un vector de características  $X$ . Opera como un factor de escala ya que aparece en todas las clases.

Todos los parámetros del modelo se pueden aproximar con frecuencias relativas del conjunto de entrenamiento. Estas son las estimaciones de mayor verosimilitud de las probabilidades.

Como en el caso del clasificador anterior separamos los datos obtenidos anteriormente en dos conjuntos, unos para entrenar el clasificador, y otros para hacer tests. Clasifica los datos en dos pasos:

- Proceso de entrenamiento: Utiliza los datos de entrenamiento y estima una distribución probabilística de las características introducidas, suponiendo que son variables independientes en cada clase.
- Proceso de predicción: para los datos de prueba, el clasificador calcula la probabilidad de que una muestra pertenezca a cada una de las clases. Después de calcular cada una de las probabilidades de cada clase, asigna la muestra a la clase que mayor probabilidad ha presentado en este proceso.



# Capítulo 5

## TransMode

### 5.1. Motivación

TransMode es una aplicación que recoge datos de aceleración, localización y velocidad continuamente mientras se está ejecutando. Esta aplicación se puede ejecutar en segundo plano, es decir, sigue funcionando mientras se utilizan otras aplicaciones o el dispositivo se encuentra en reposo. Este hecho justifica la importancia de que tenga un reducido consumo de la batería y que no suponga un impedimento en el rendimiento del dispositivo mientras se usan otras aplicaciones, ya que de no cumplir estas condiciones la aplicación no será utilizada por los usuarios al ver reducido el tiempo de funcionamiento de su dispositivo entre cargas.

La aplicación lee la información del acelerómetro con una cierta frecuencia y calcula un promedio de todas las muestras recogidas por segundo. Los datos de localización y velocidad son actualizados una vez por segundo. Este paquete de datos se escribe en un fichero de texto, para poder ser enviados, y se almacenan en un array para su posterior procesamiento junto con el resto de datos recogidos durante el intervalo de tiempo que será definido tras el estudio realizado en el capítulo de resultados. Dado que la aplicación está diseñada para un uso masivo, es importante recoger datos de muchos usuarios para poder analizarlos y mejorar el funcionamiento de cara al futuro. Una de las ventajas que presentan los smartphones es la facilidad de comunicación que ofrecen, ya que podemos conectarnos a internet usando la conexión de datos móviles o mediante conexiones de wifi y enviar los datos recogidos. Por ello, se ha desarrollado un servidor para recoger todos estos datos de los usuarios de TransMode.

### 5.2. Algoritmo

La aplicación recoge los datos de los sensores automáticamente. Los datos de aceleración son preprocesados por el filtro paso alto y cada segundo se obtiene una media de aceleración y un valor de velocidad. Como se observó durante los análisis de las técnicas de procesamiento, los desplazamientos que se realizan andando son los más sencillo de separar del resto de actividades, ya que presentan valores de aceleración superiores y velocidades más bajas que los desplazamientos realizados en un vehículo. Para realizar dos desplazamientos en vehículos distintos siempre debe aparecer un tramo intermedio realizado andando mientras el usuario se desplaza de uno a otro. El algoritmo se basa en la facilidad de distinción de los trayectos realizados andando para separar las actividades.

Se define un valor de velocidad máxima andado de 2 m/s y otro con la aceleración media

mínima andando de 0.4 g. Si se experimenta una aceleración media superior a la mínima aceleración andando definida durante un intervalo de diez segundos, la aplicación detectará el inicio de un desplazamiento andando ("modo andando"). Para finalizar el "modo andando" se debe detectar aceleraciones inferiores a la mínima aceleración media andando durante 50 segundos, así se evita detectar cambios de actividad, por ejemplo al esperar en un paso de cebra. Mientras que si las aceleraciones son inferiores a la mínima aceleración andando y la aplicación no se encuentra en "modo andando", TransMode distinguirá si se está efectuando un desplazamiento o si el usuario se encuentra parado mediante la velocidad experimentada. En el caso de que la velocidad registrada sea superior a la máxima velocidad andando (2 m/s), la aplicación detectará que se está efectuando un desplazamiento en vehículo y se realizará el procesamiento que será seleccionado tras el estudio de los resultados y la predicción del clasificador implementado para detectar el medio de transporte utilizado. Si la velocidad recogida es inferior, TransMode asume que el usuario se encuentra parado. La Figura 5.1 muestra este algoritmo.

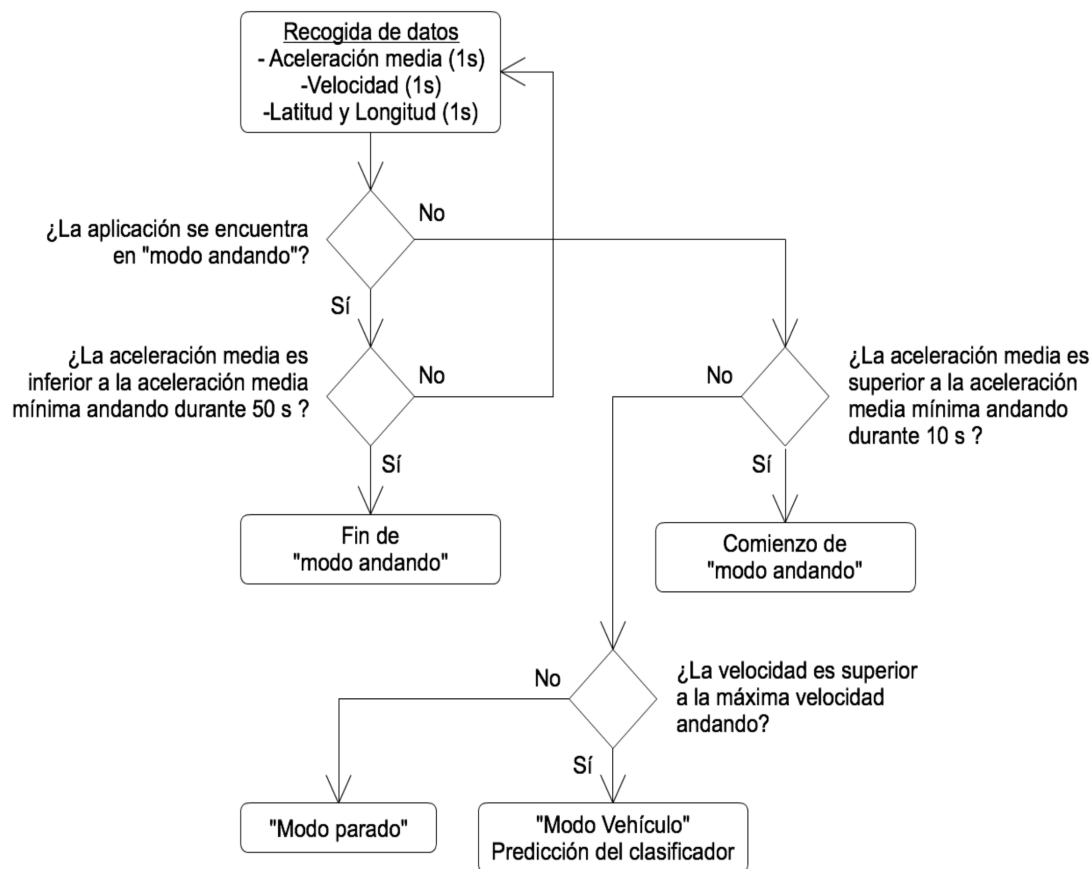


Figura 5.1: Algoritmo de detección de modos

### 5.3. Servidor

Dentro de la aplicación TransMode, se ha desarrollado una función que permite enviar los datos recogidos durante el período de ejecución al servidor. Cada vez que se cambia de medio de transporte detectado, los datos se guardan como un fichero de texto en una carpeta donde permanecerán hasta ser enviados. TransMode genera una identificación única y aleatoria para cada dispositivo, de esta manera los datos recogidos son almacenados de manera anónima, y

garantizamos la privacidad de nuestros usuarios.

La aplicación comprueba la conectividad con el servidor de manera frecuente durante su ejecución. Cuando la conexión está disponible, todos los fichero que están en la cola para ser subidos son enviados. Actualmente el servidor se encuentra alojado en la dirección `http://drivesafe.uah.es`.

### 5.3.1. Diseño del servidor

[17] El servidor ha sido desarrollado siguiendo el método LAMP (Linux, Apache, MySQL, PHP), véase Figura 5.2. [17]

- Linux: sistema operativo que permite un cómodo manejo de los recursos de red y el almacenamiento de datos.
- Apache: un servidor web que implementa un protocolo HTTP que permite la recepción de los ficheros por parte de los usuarios.
- MySQL: permite la implementación de una base de datos. Actualmente no está en uso, pero en el futuro será utilizada cuando el manejo de los ficheros por parte de muchos usuarios sea complicado.
- PHP es un lenguaje de programación que permite programar las envíos de datos.

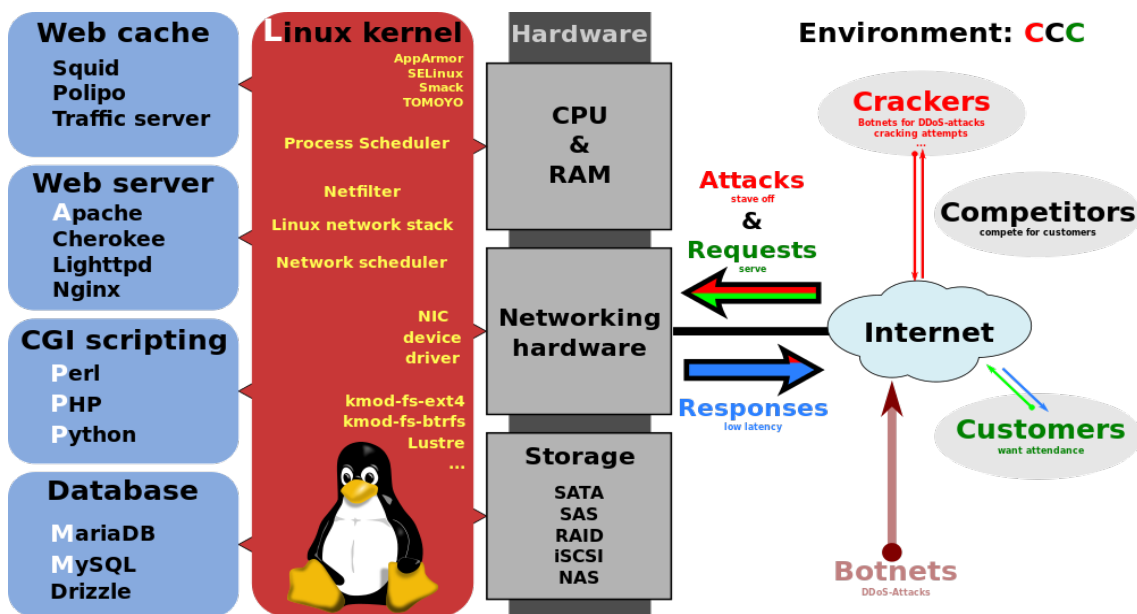


Figura 5.2: Esquema del método LAMP

### 5.3.2. Código PHP

El código PHP controla las peticiones de envío de los datos por parte de los usuarios de Trans-Mode. Cada envío tiene que pasar una serie de filtros para evitar la corrupción de los ficheros. Primero, los envíos están protegidos por una ID y una contraseña que solo están disponibles en el código de la aplicación, de esta manera aseguramos la aplicación oficial de TransMode es

la única que puede mandar datos. Segundo, el tamaño y el tipo de fichero son revisados para controlar que cumplen con los requisitos estándar. Por último, se comprueba la integridad del fichero y se almacena en el servidor. El código también controla la creación de carpetas para cada usuario o dispositivo que utiliza la aplicación TransMode cuando es necesario (realiza el primer envío). Si todo esto se cumple, el código manda de vuelta una respuesta HTTP para que el usuario pueda enviar sus ficheros con éxito.

### 5.3.3. Estructura de datos

En la Figura 5.3 se describe cómo se almacena los ficheros en el servidor por parte del código PHP. El nombre del fichero mandado contiene los identificadores que permite ordenar el envío en la carpeta correspondiente. La identificación de la app, permite el uso de varias aplicaciones en el mismo servidor. La ID única para cada usuario permite clasificar los datos por usuario. La identificación del smartphone permite ordenar los envíos según el tipo de dispositivo (iPhone 4S, iPhone 6,...) y así poder conocer las características de estos, como los acelerómetros con los que están equipados. Por último, los datos son guardados con el nombre del día y hora en el que el fichero fue generado. Actualmente, los datos se guardan en formato zip. Como trabajo futuro, el código PHP también se configurará para extraer los ficheros y ejecutar el análisis de los datos.

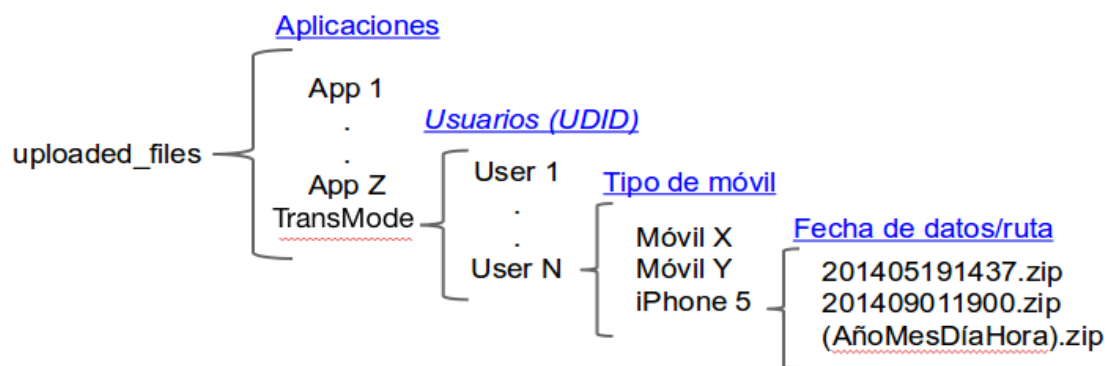


Figura 5.3: Esquema de almacenaje del servidor

## 5.4. Análisis de datos

Un servidor que recoge los datos de un uso masivo de la aplicación es esencial ya que:

- Permite un feedback para mejorar la aplicación de cara al futuro. Conocer los fallos o aciertos de la aplicación nos ayudará a corregir y mejorar el funcionamiento.
- Recibir datos de transporte de un gran número de personas, nos permitirá entrenar los clasificadores con datos de distintas ciudades, distintos conductores, distintas situaciones climatológicas, etc. Esto permitirá mejorar la precisión de la clasificación.
- Análisis de datos. La globalización pone en manifiesto la importancia de trabajar con datos multitudinarios, en términos estadísticos, como una fuente de información que a día de hoy es esencial para cualquier institución. El campo del análisis de datos está en evolución constante desde el año pasado. En el campo de los medios de transporte, la disponibilidad de informes sobre los comportamientos de conducción, los medios de

---

transporte preferidos, las rutas más transitadas, las horas puntas, etc... son de gran interés para muchas compañías y gobiernos. Estas técnicas son conocidas como *big data* y deberán ser estudiadas en trabajos futuros.



# Capítulo 6

## Resultados

En el siguiente capítulo se presentan los resultados obtenidos en el estudio de las distintas técnicas de procesamiento empleadas, actuando sobre las variables con el fin de obtener la clasificación más precisa. La configuración que ofrezca los mejores resultados utilizando los recursos mínimos del smartphone será implementada en la aplicación definitiva, TransMode.

Los parámetros que se pueden modificar para realizar el estudio son:

- *kUpdateFrequency*: número de lecturas de datos del acelerómetro por segundo. En la aplicación prototipo se recogieron datos a 200 Hz. En este estudio se reducirá la frecuencia hasta llegar a obtener una frecuencia más baja que permita la correcta clasificación.
- *cutOffFrequency*: frecuencia de corte del filtro paso alto. No debe ser muy elevada, ya que queremos filtrar las aportaciones constantes de la gravedad. Tras la realización de varias pruebas con distintos valores de frecuencia de corte se llegó a la conclusión de que 5 Hz es la mejor frecuencia de corte para abstraer los comportamientos de aceleración de los viajes realizados. Se debe considerar que la implementación de este filtro está directamente relacionada con la frecuencia de muestreo de la aceleración, de este modo *kUpdateFrequency* nunca debe ser menor que *cutOffFrequency*.
- *n\_samples*: número de muestras que se incluyen en la ventana para el procesamiento de datos. Haremos un estudio de los índices de error para muestras recogidas en intervalos temporales que van desde un segundo ( $n_{samples} = kUpdateFrequency$ ) hasta dos minutos ( $n_{samples} = kUpdateFrequency \cdot 120$ ), ya que se ha considerado que la información aportada por la aplicación debe actualizarse por debajo de dos minutos.
- *n\_gap*: número de muestras que se desplaza la ventana en cada iteración del procesamiento. En este estudio se fija en un valor de  $n_{gap} = kUpdateFrequency$ , es decir, para que en cada iteración la ventana temporal avance un segundo en los datos recogidos.

Para obtener los resultados en este estudio se realiza, en primer lugar, el procesamiento con los datos recogidos a máxima frecuencia, es decir a 200Hz, y los usamos para entrenar los clasificadores y realizar los test que nos aportarán el índice de error. Este índice de error es un valor que expresa el número de muestras que no han sido identificadas correctamente durante el test, aportando una referencia para identificar en que zona de operación presenta las mejores resultados. Seguidamente, se comparan los mejores resultados obtenidos para los clasificadores SVM y los Naïve-Bayes utilizados. Por último, se repiten los estudios a distintas frecuencias de adquisición de aceleración con la mejor configuración obtenida.

## 6.1. Promedio

Las características extraídas del procesado con medias son utilizados para entrenar los clasificadores y realizar los correspondientes tests. Los índices de error que conseguimos, en función del tiempo incluido en la ventana temporal, son representados en la Figura 6.1, algunos son expuestos en la Tabla 6.1 para los clasificadores SVM y en la Tabla 6.2 para los clasificadores bayesianos. En la Figura 6.2 se representan las curvas ROC (Receiver Operating Characteristics) de los clasificadores estudiados para distintos intervalos de tiempo procesados.

Se observa que las predicciones de los clasificadores mejoran conforme se incrementa el intervalo de tiempo de datos procesados en la venta temporal, y que los clasificadores SVM son notablemente mejores que los bayesianos.

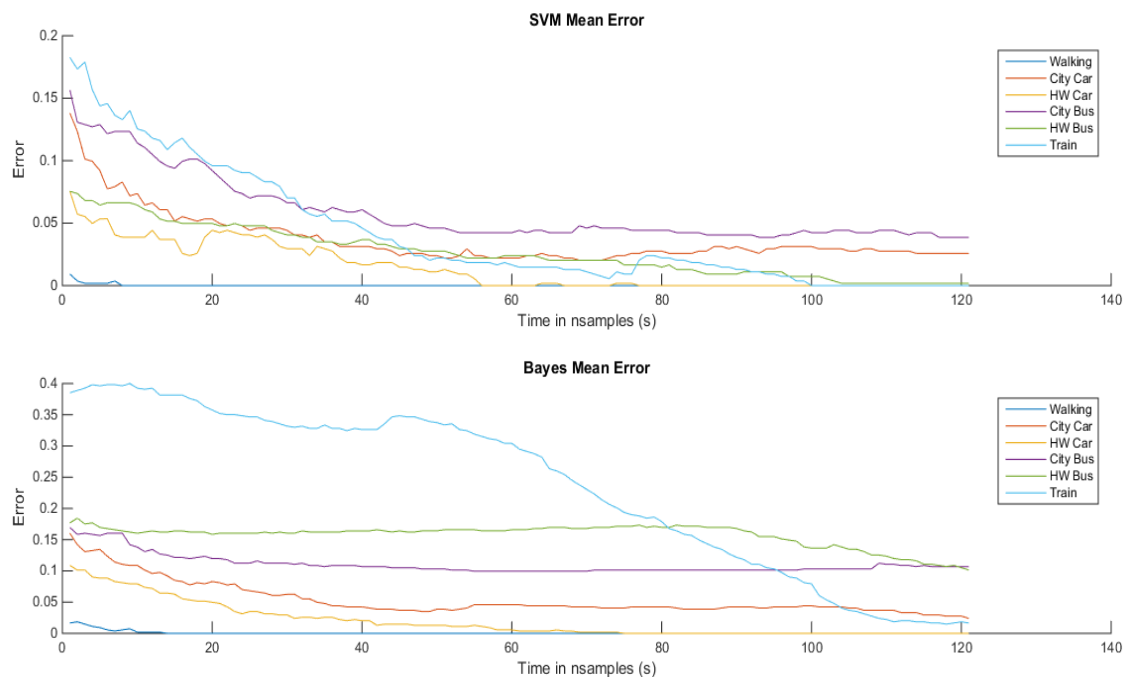


Figura 6.1: Promedio: índice de error en función del tiempo procesado.

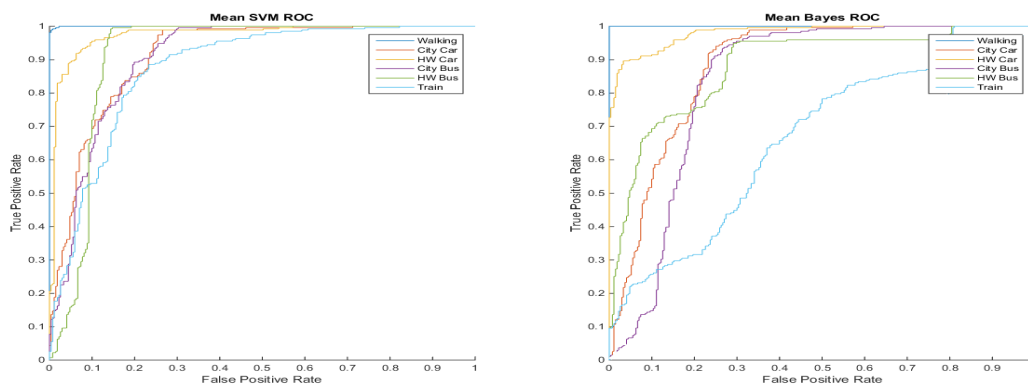
$n_{samples}$	Walking	CityCar	HWCAR	CityBus	HWBUS	Train
1 s	0.92 %	13.81 %	7.56 %	15.66 %	7.55 %	18.26 %
20 s	0.00 %	5.16 %	4.25 %	8.66 %	4.79 %	9.78 %
40 s	0.00 %	3.13 %	1.66 %	5.71 %	3.68 %	4.61 %
60 s	0.00 %	2.58 %	0.00 %	4.60 %	2.39 %	1.84 %
80 s	0.00 %	2.95 %	0.00 %	4.60 %	1.66 %	2.59 %
100 s	0.00 %	3.32 %	0.00 %	4.42 %	0.74 %	0.37 %
120 s	0.00 %	3.13 %	0.00 %	4.42 %	0.18 %	0.37 %

Tabla 6.1: Promedio: índices de error para clasificadores SVM.

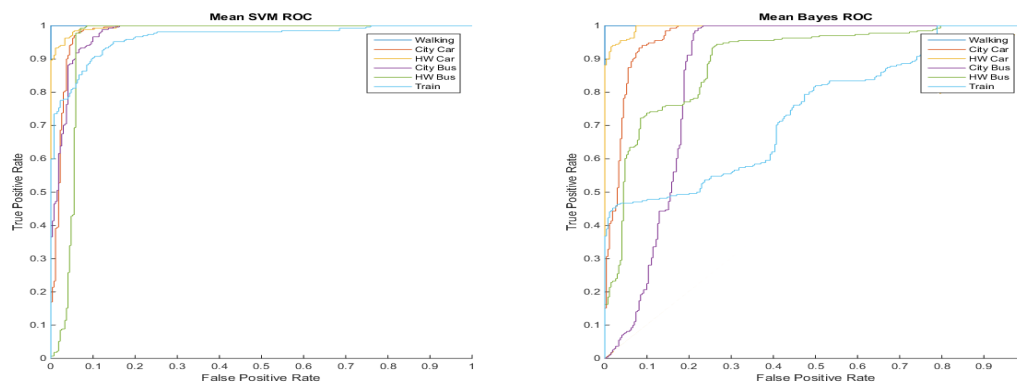
## 6.2. Multiplicaciones

En este caso utilizamos las características obtenidas tras el procesamiento de multiplicaciones para entrenar y realizar test con ambos clasificadores. Los índices de error obtenidos en este caso

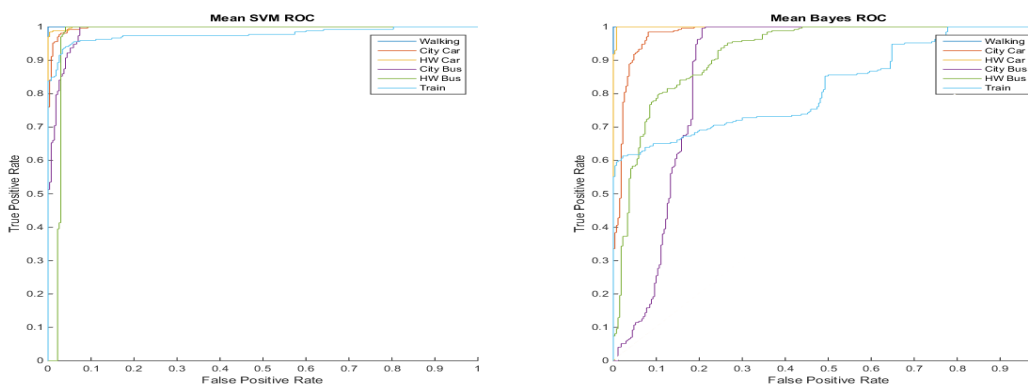




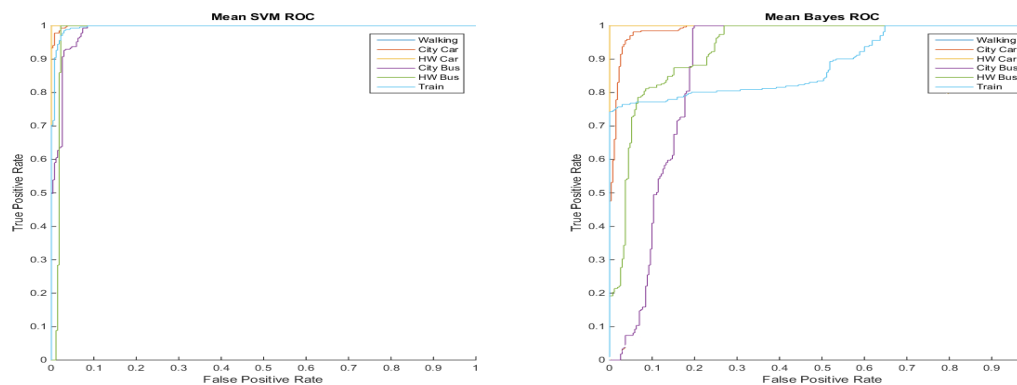
(a) Curvas ROC: ventana de 1s.



(b) Curvas ROC: ventana de 20s.



(c) Curvas ROC: ventana de 40s.



(d) Curvas ROC: ventana de 60s.

Figura 6.2: Promedio: curvas ROC SVM vs Bayes en función del tiempo muestreado.

<i>time</i>	Walking	CityCar	HWCAR	CityBus	HWBus	Train
1 s	1.66 %	16.03 %	10.87 %	16.95 %	17.69 %	38.52 %
20 s	0.18 %	8.29 %	4.79 %	11.97 %	16.04 %	35.39 %
40 s	0.18 %	5.34 %	1.11 %	10.68 %	16.41 %	32.26 %
60 s	0.37 %	5.16 %	0.18 %	9.94 %	16.59 %	30.42 %
80 s	0.37 %	4.79 %	0.18 %	10.13 %	17.15 %	18.62 %
100 s	0.37 %	5.16 %	0.00 %	10.31 %	13.46 %	9.03 %
120 s	0.37 %	4.98 %	0.37 %	10.68 %	9.77 %	1.85 %

Tabla 6.2: Promedio: índices de error para clasificadores Bayes.

se representan en la Figura 6.3. Algunos de estos valores se exponen en la Tabla 6.3 para los clasificadores SVM y en la Tabla 6.4 para los clasificadores bayesianos. En la Figura 6.4 se muestran las curvas ROC de los clasificadores utilizados para clasificar muestras con distintos intervalos de tiempo procesado.

Se observa que un clasificador SVM funciona mejor para distinguir los desplazamientos realizados en coche en autovía, autobús en autovía y tren. Mientras que los trayectos de coche en ciudad y autobús en ciudad son algo más precisos utilizando clasificadores Naïve-Bayes. En general, los resultados obtenidos con este procesamiento no alcanzan la calidad de los conseguidos con el procesamiento con promedios.

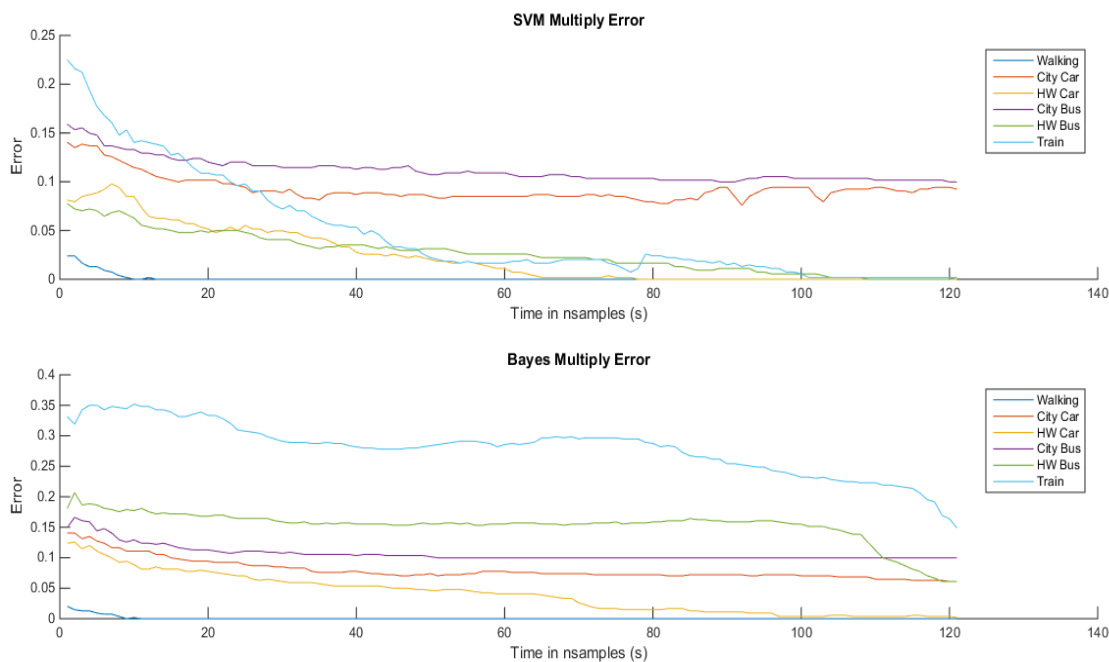
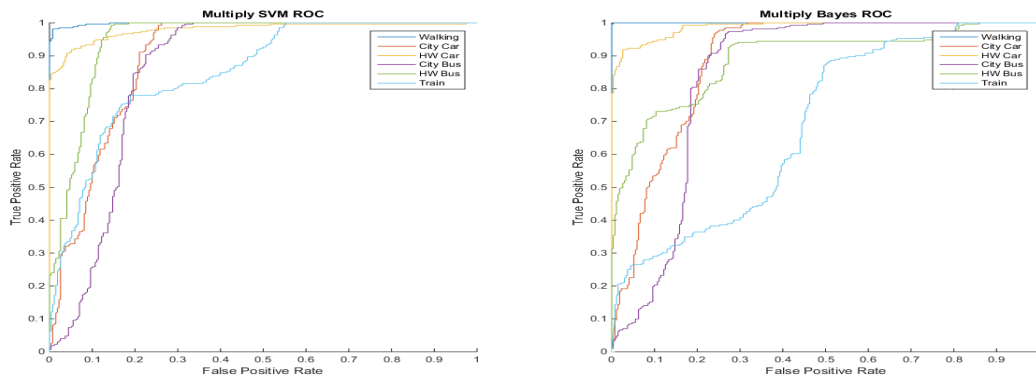


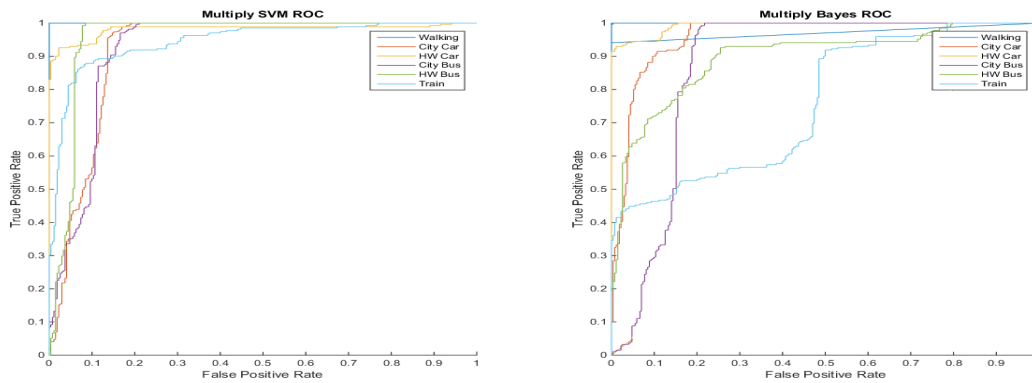
Figura 6.3: Multiplicación: índices de error en función del intervalo de tiempo procesado.

### 6.3. Integral

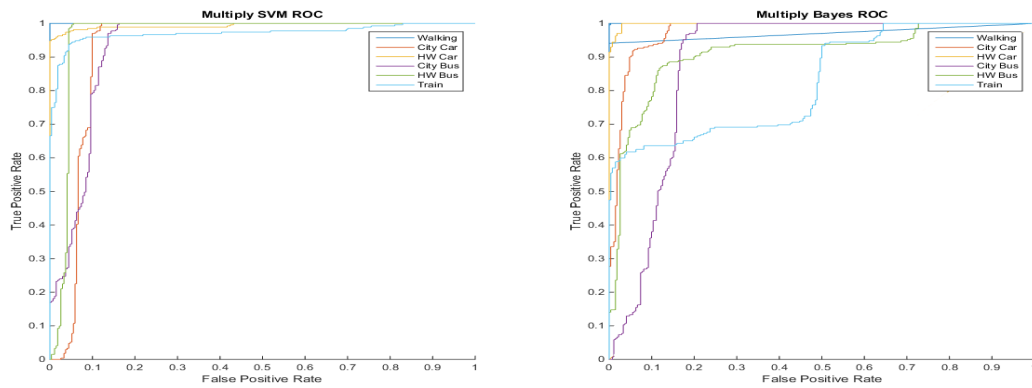
En este caso se evalúan las posibles configuraciones con las características obtenidas en el procesamiento mediante integrales. Los índices de error que se obtienen tras realizar los test para distintos intervalos de datos incluidos en la ventana temporal son representados en la Figura 6.5



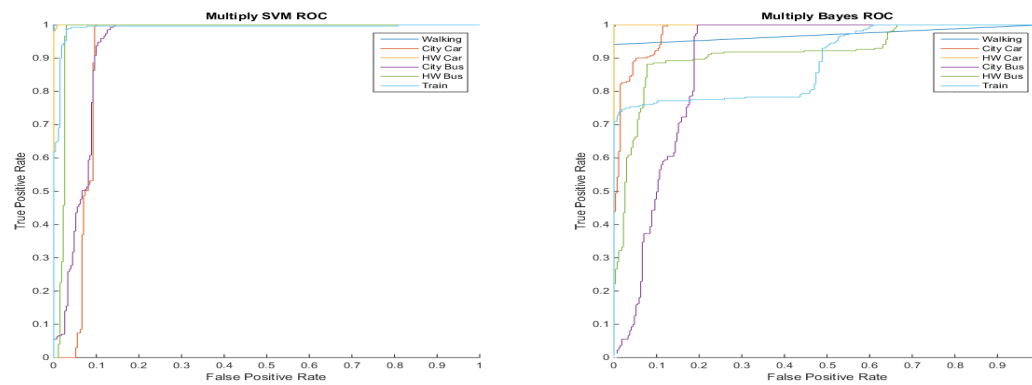
(a) Curvas ROC: ventana de 1s.



(b) Curvas ROC: ventana de 20s.



(c) Curvas ROC: ventana de 40s.



(d) Curvas ROC: ventana de 60s.

Figura 6.4: Multiplicación: curvas ROC SVM vs Bayes en función del tiempo muestreado.

<i>time</i>	Walking	CityCar	HWCAR	CityBus	HWBus	Train
1 s	2.40 %	14.02 %	7.93 %	15.87 %	7.75 %	22.51 %
20 s	0.37 %	10.33 %	4.80 %	11.81 %	4.98 %	10.52 %
40 s	0.37 %	9.41 %	2.58 %	11.44 %	3.51 %	5.35 %
60 s	0.37 %	8.67 %	0.74 %	10.52 %	2.58 %	2.40 %
80 s	0.37 %	8.86 %	0.00 %	10.15 %	1.66 %	1.48 %
100 s	0.37 %	9.78 %	0.00 %	10.33 %	0.55 %	0.74 %
120 s	0.00 %	9.41 %	0.00 %	9.96 %	0.00 %	0.74 %

Tabla 6.3: Multiplicación: índices de error para clasificadores SVM.

<i>n<sub>samples</sub></i>	Walking	CityCar	HWCAR	CityBus	HWBus	Train
1 s	2.03 %	13.84 %	12.36 %	14.94 %	17.71 %	33.21 %
20 s	0.37 %	9.41 %	7.56 %	11.07 %	16.97 %	34.13 %
40 s	0.00 %	7.56 %	5.35 %	10.52 %	15.50 %	28.23 %
60 s	0.00 %	7.75 %	3.32 %	9.96 %	15.68 %	28.97 %
80 s	0.00 %	7.20 %	1.11 %	9.96 %	15.85 %	28.60 %
100 s	0.00 %	7.01 %	0.37 %	9.96 %	14.76 %	23.80 %
120 s	0.00 %	6.46 %	0.18 %	9.96 %	5.90 %	21.22 %

Tabla 6.4: Multiplicación: índices de error para clasificadores Bayes.

y se muestran en la Tabla 6.5 para los clasificadores SVM y en la Tabla 6.6 para los clasificadores Naïve-Bayes. El análisis ROC de los clasificadores entrenados con estas características se representan en la Figura 6.6.

Los resultados conseguidos mediante este procesamiento muestran un mejor comportamiento en los clasificadores SVM con respecto a los clasificadores Naïve-Bayes. La precisión en la clasificación mejora a medida que aumenta el intervalo de tiempo de la ventana que contiene las muestras a procesar.

<i>time</i>	Walking	CityCar	HWCAR	CityBus	HWBus	Train
1 s	0.92 %	13.16 %	5.72 %	14.21 %	7.20 %	17.34 %
20 s	0.00 %	5.17 %	4.24 %	8.49 %	4.98 %	10.89 %
40 s	0.00 %	3.14 %	1.85 %	5.35 %	3.51 %	4.80 %
60 s	0.00 %	3.32 %	0.00 %	4.61 %	2.40 %	1.85 %
80 s	0.00 %	3.51 %	0.00 %	4.61 %	1.48 %	0.92 %
100 s	0.00 %	3.51 %	0.00 %	4.43 %	0.92 %	0.37 %
120 s	0.00 %	2.95 %	0.00 %	4.61 %	0.37 %	0.37 %

Tabla 6.5: Integración: índices de error para clasificadores SVM.

## 6.4. Diferencial

En este último caso se utilizan las características que se obtienen tras el procesamiento de los datos mediante el cálculo diferencial. Los índices de error obtenidos para distintos intervalos de tiempo de procesamiento se representan en la Figura 6.7. Los resultados se muestran en la Tabla 6.7 para los clasificadores SVM y en la Tabla 6.8 para los bayesianos.

Observamos que los índices de error obtenidos en este caso son bastante peores que los

$n_{samples}$	Walking	CityCar	HWCAR	CityBus	HWBus	Train
1 s	2.21 %	14.94 %	9.04 %	15.50 %	17.53 %	40.22 %
20 s	0.37 %	8.49 %	5.35 %	11.62 %	16.24 %	37.08 %
40 s	0.37 %	6.09 %	1.29 %	10.70 %	16.42 %	32.47 %
60 s	0.37 %	5.72 %	0.55 %	10.15 %	16.79 %	31.18 %
80 s	0.37 %	5.72 %	0.18 %	10.15 %	17.34 %	20.30 %
100 s	0.37 %	5.90 %	0.18 %	10.52 %	13.84 %	12.36 %
120 s	0.37 %	5.54 %	0.18 %	10.89 %	9.23 %	4.06 %

Tabla 6.6: Integración: índices de error para clasificadores Bayes.

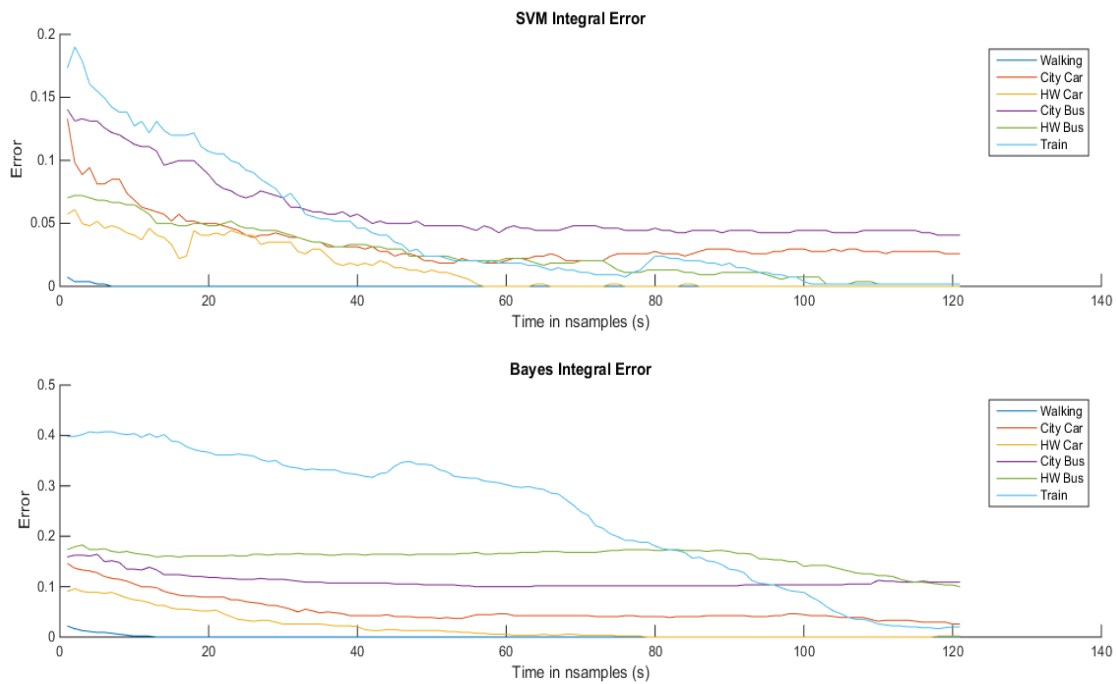
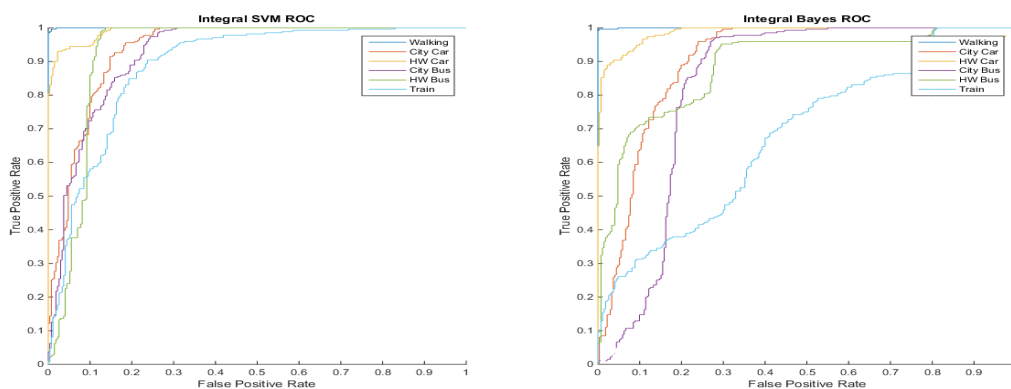
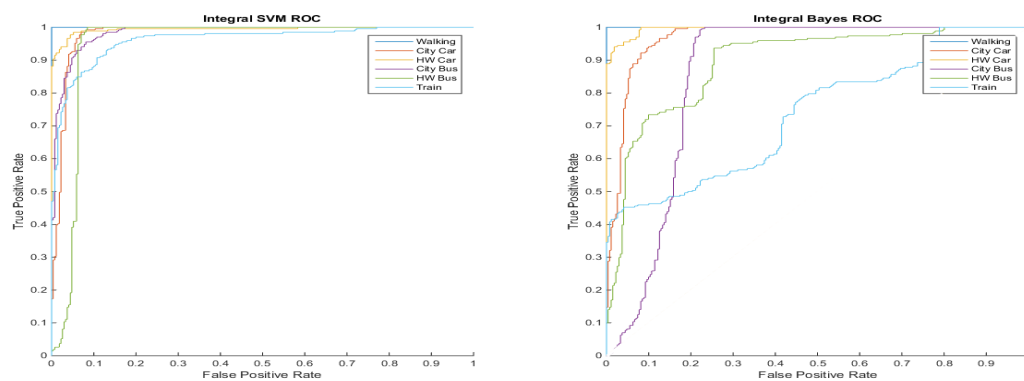


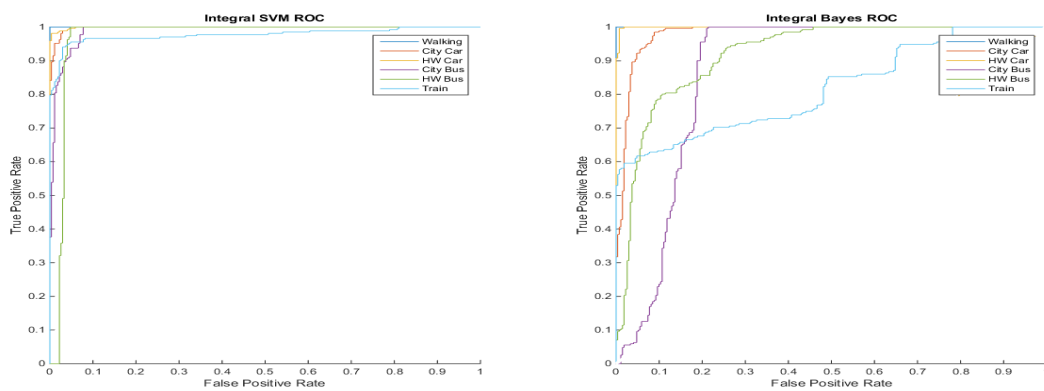
Figura 6.5: Integración: índices de error en función del intervalo de tiempo procesado.



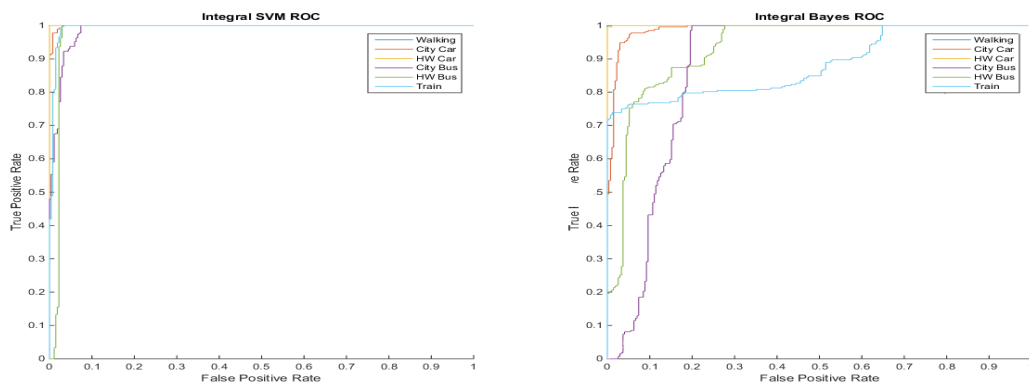
(a) de 1s.



(b) Curvas ROC: ventana de 20s.



(c) Curvas ROC: ventana de 40s.



(d) Curvas ROC: ventana de 60s.

Figura 6.6: Integración: curvas ROC SVM vs Bayes en función del tiempo muestreado.

conseguidos con el resto de procesamiento, por lo que no se profundiza en su análisis.

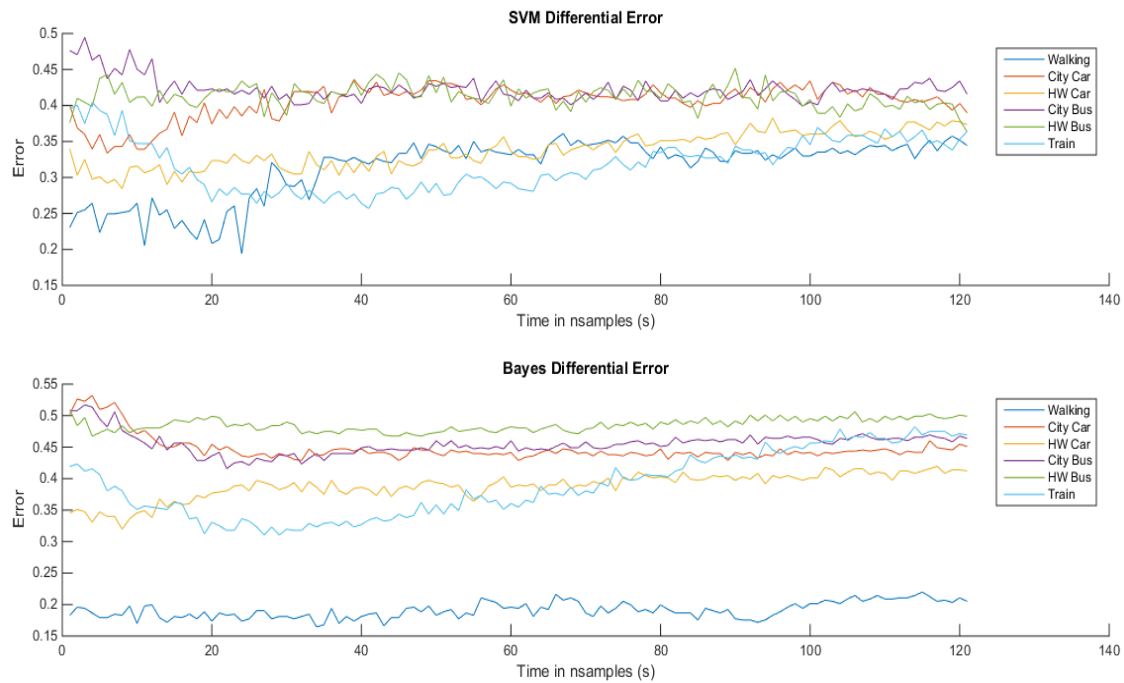


Figura 6.7: Derivación: índices de error en función del intervalo de tiempo procesado.

<i>time</i>	Walking	CityCar	HWCAR	CityBus	HWBus	Train
1 s	20.11 %	40.41 %	33.21 %	46.49 %	41.88 %	39.85 %
20 s	22.51 %	36.35 %	34.50 %	43.36 %	42.99 %	28.78 %
40 s	26.01 %	41.70 %	34.87 %	43.54 %	42.99 %	26.94 %
60 s	25.83 %	42.80 %	35.42 %	41.14 %	42.44 %	27.12 %
80 s	26.20 %	42.07 %	34.50 %	41.14 %	44.28 %	29.70 %
100 s	28.97 %	43.73 %	36.72 %	40.41 %	43.73 %	35.06 %
120 s	29.52 %	43.54 %	35.61 %	40.77 %	42.07 %	34.87 %

Tabla 6.7: Derivación: índices de error para clasificadores SVM.

<i>n<sub>samples</sub></i>	Walking	CityCar	HWCAR	CityBus	HWBus	Train
1 s	17.53 %	50.55 %	33.76 %	49.63 %	52.95 %	41.70 %
20 s	16.97 %	44.65 %	42.25 %	45.02 %	48.52 %	32.84 %
40 s	16.42 %	46.31 %	43.73 %	45.39 %	48.15 %	31.00 %
60 s	19.74 %	49.08 %	44.46 %	47.05 %	48.34 %	35.06 %
80 s	18.63 %	49.45 %	45.57 %	47.79 %	48.15 %	40.41 %
100 s	20.11 %	49.45 %	46.68 %	47.97 %	47.79 %	43.91 %
120 s	20.48 %	49.45 %	47.60 %	48.15 %	47.60 %	45.57 %

Tabla 6.8: Derivación: índices de error para clasificadores Bayes.

## 6.5. Configuración óptima

Tras el estudio de los clasificadores para las distintas técnicas de procesamiento se concluye en que los mejores resultados son los obtenidos con medias y con integraciones. Se debe encontrar un intervalo temporal que ofrezca una alta precisión en la clasificación, sin ser excesivamente elevado para no perder información en caso de un cambio de actividad, ni excesivamente pequeño ya que incrementaría el consumo de la aplicación. La Tabla 6.5 muestra los resultados que cumplen estas condiciones. Se observa que los clasificadores entrenados con características del procesamiento con medias y los entrenados con integraciones presentan unos comportamientos prácticamente idénticos. Se opta por utilizar el procesamiento más sencillo en la aplicación con la finalidad de reducir el consumo de batería y el uso de la CPU del dispositivo, en este caso el procesamiento con medias.

Classifier	Time	Walking	CityCar	HWCAR	CityBus	HWBus	Train
Mean SVM	95 s	0.00 %	3.67 %	0.22 %	3.02 %	1.30 %	1.08 %
Int SVM	95 s	0.00 %	3.44 %	0.22 %	3.44 %	1.29 %	1.30 %

Tabla 6.9: Promedio: configuración óptima.

La Tabla 6.5 muestra como afecta la reducción de la frecuencia de muestro de los valores de aceleración en la precisión de la clasificación para la configuración seleccionada, con características obtenidas mediante el procesamiento de promediado y una ventana temporal de 95s. Se observa que reduciendo la frecuencia de muestreo del acelerómetro a 10 Hz la precisión se mantiene prácticamente constante, esto se debe a larga duración de la ventana en esta configuración. Esto permite reducir en un 95 % la frecuencia de adquisición de datos de aceleración.

<i>kUpdateFrequency</i>	Walking	CityCar	HWCAR	CityBus	HWBus	Train
200 Hz	0.00 %	3.67 %	0.22 %	3.02 %	1.30 %	1.08 %
100 Hz	0.00 %	4.08 %	0.23 %	2.72 %	1.36 %	1.11 %
50 Hz	0.00 %	3.85 %	0.23 %	2.27 %	1.36 %	1.11 %
20 Hz	0.00 %	4.31 %	0.23 %	2.72 %	1.36 %	1.11 %
10 Hz	0.00 %	3.63 %	0.23 %	2.49 %	1.36 %	1.11 %

Tabla 6.10: Índices de error para distintos valores de *kUpdateFrequency*.

La Tabla 6.5 muestra las características de la configuración seleccionada para implementar en TransMode.

Clasificador	Procesamiento	<i>kUpdateFrequency</i>	<i>cutOffFrequency</i>	Tiempo vent
SVM	Promediado	10 Hz	5 Hz	95 s

Tabla 6.11: Configuración seleccionada.



# Capítulo 7

## Conclusiones y Trabajos Futuros

### 7.1. Conclusiones

Una vez finalizado el estudio de los resultados, implementamos en TransMode la configuración seleccionada para la clasificación del medio de transporte. Para ello utilizamos las librerías de OpenCV que nos permiten utilizar los clasificadores en la aplicación de iOS, y entrenamos un clasificador SVM con los datos que nos aportaron los mejores resultados en el apartado anterior. Cuando el algoritmo de TransMode detecte un desplazamiento en vehículo durante el tiempo suficiente para conseguir completar la ventana temporal (95 segundos), se pasará un vector con las características de ese intervalo de tiempo y se realizará una predicción por parte del clasificador. Cuanto mayor duración tenga el desplazamiento en vehículo, se consigue un mayor número de predicciones, esto permite evaluar un desplazamiento con un conjunto de predicciones. Por ejemplo, si durante un desplazamiento de media hora las predicciones han tenido como resultado más frecuente desplazamientos en coche en ciudad y coche en autovía, la aplicación clasificará ese desplazamiento como una actividad en coche. El resultado de cada predicción es mostrado por pantalla, como se muestra en la Figura 7.1. TransMode ha sido probada en varios desplazamientos con los distintos medios de transporte obteniendo los resultados de la Tabla 7.1.

En cuanto al impacto de TransMode en el dispositivo, en la Figura 7.2 se muestra el porcentaje de uso de la memoria y de la CPU. Se puede observar que el uso de la memoria RAM del dispositivo se encuentra en el 1.1% con unos 10.9 MB, y el de la CPU se encuentra en el 2%. Esto demuestra que la aplicación puede ejecutarse en segundo plano sin afectar en el funcionamiento del smartphone, ya que la carga en el dispositivo es prácticamente nula. La duración de la batería no ha experimentado cambios significativos mientras se ejecutaba TransMode. Se realizó una prueba de uso intensivo durante dos horas y en ambos casos el consumo de la batería fue cercano al 30%. En otra prueba la aplicación ejecutaba en segundo plano mientras el dispositivo se encontraba bloqueado, y en dos horas el porcentaje de batería bajo en un 1%.

### 7.2. Trabajos Futuros

- Ampliar la base de datos. La aplicación está preparada para recoger datos y enviarlos al servidor, pero para poder mejorar la clasificación debemos estar seguros de que los datos que se usan para entrenar al clasificador están correctamente etiquetados. Para ello se utilizará una pantalla que pregunte si la clasificación se ha realizado correctamente, o en caso negativo permitir que el usuario indique en que medio de transporte ha efectuado su

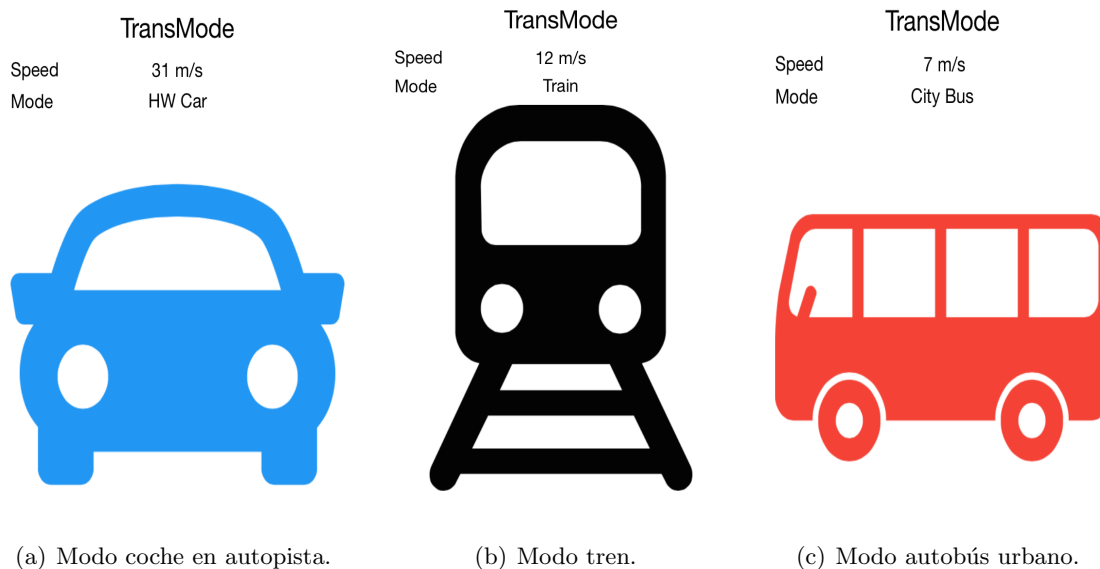
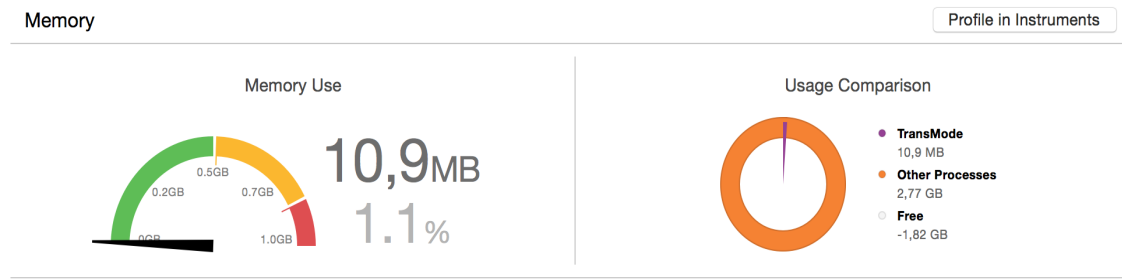


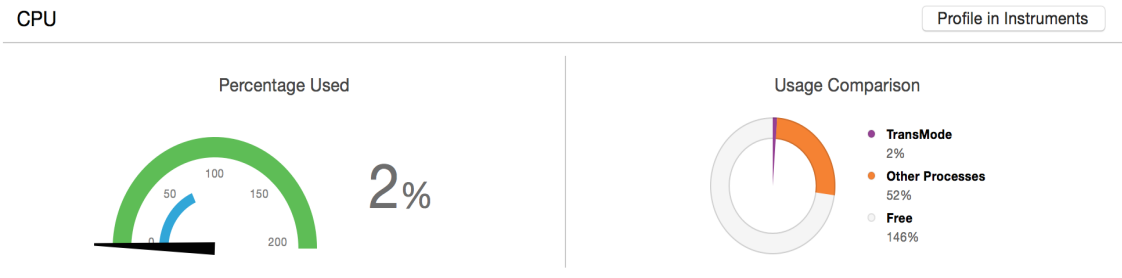
Figura 7.1: Interface TransMode.

Vehículo	Trayecto	Duración	Predicciones	Resultado
Coche	Guadalajara Campus UAH	25 min	11 HWCAR	Coche (81.3%) Bus (18.7%)
			2 CityCar 2 HWBus 1 CityBus	
Coche	Campus UAH Gaudalajara	25 min	11 HWCAR	Coche (93.3%) Bus (6.7%)
			3 CityCar 1 CityBus	
Autobús	Guadalajara Urbano	15 min	5 CityBus	Bus (55.5%) Coche (33.3%) Tren(11.1%)
			3 CityCar	
			1 Train	
Tren	Guadalajara Campus UAH	20 min	7 Train	Tren (70.0%) Bus (20.0%) Coche (10.0%)
			3 CityBus	
			1 CityCar	
Tren	Campus UAH Guadalajara	20 min	8 Train	Tren (72.7%) Bus (27.3%)
			3 CityBus	

Tabla 7.1: Resultados de TransMode en viaje.



(a) Uso de memoria.



(b) Uso de la CPU.

Figura 7.2: Impacto de TransMode en el dispositivo.

desplazamiento.

- Configurar el código PHP del servidor. Esto permitirá la automatización en el proceso de análisis de datos y entrenamiento de los clasificadores. El propio servidor, al recibir un nuevo paquete de datos será capaz automáticamente de procesar los datos con la configuración deseada y entrenar al clasificador SVM que implementa TransMode para mejorar el funcionamiento.
- Ampliar el número de medios de transporte que pueden ser identificados. En esta primera aplicación únicamente se ha trabajado para detectar coches, autobuses y trenes. Para hacer llegar la aplicación al mayor número de usuarios posible se debe añadir al algoritmo de clasificación más medios de transporte, por ejemplo: moto, camión, bicicleta, corriendo...
- Incorporar a la aplicación la información relacionada a los desplazamientos detectados para atraer al usuario. Por ejemplo, las emisiones de  $CO_2$  asociadas al usuario, el combustible gastado en el desplazamiento, el nivel de contaminación de ciudades, el número de calorías quemadas en una actividad de tracción humana.



**Parte II**

**Manual de Usuario**



## 7.3. Manual

En esta sección se resumirá el manual de usuario de las aplicaciones desarrolladas en este trabajo.

### 7.3.1. Aplicación prototipo

La aplicación que se utilizó para recoger los primeros datos presenta un funcionamiento muy sencillo. Al ejecutar esta aplicación se mostrará una pantalla en la que aparecerá una serie de interruptores, uno para cada medio de transporte del que se quieran recoger datos. El usuario debe de activar el botón correspondiente justo antes de comenzar su desplazamiento, la aplicación comenzará a escribir los datos recogidos en un fichero de texto. Esta aplicación puede ejecutarse en segundo plano, así que es posible bloquear el dispositivo o utilizar otras aplicaciones mientras funciona. Una vez finalizado el trayecto, el usuario debe apagar el interruptor para dejar de recoger datos. En la parte de abajo de la pantalla aparece un botón (Enviar datos) que permite comprimir todos los ficheros recogidos con la aplicación en un zip y enviarlos al servidor. Véase Figura 7.3.

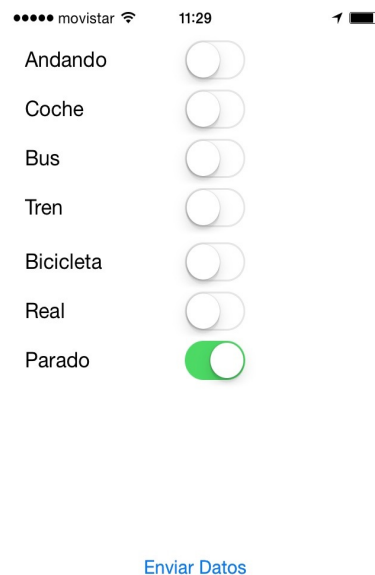


Figura 7.3: Interface de la aplicación para recoger datos.

### 7.3.2. TransMode

Esta aplicación es totalmente automática por el momento. El usuario solo tiene que ejecutarla y aparecerá una pantalla donde se mostrará la velocidad con la que se esta desplazando y la predicción del medio de transporte que realiza el clasificador. Véase 7.4.

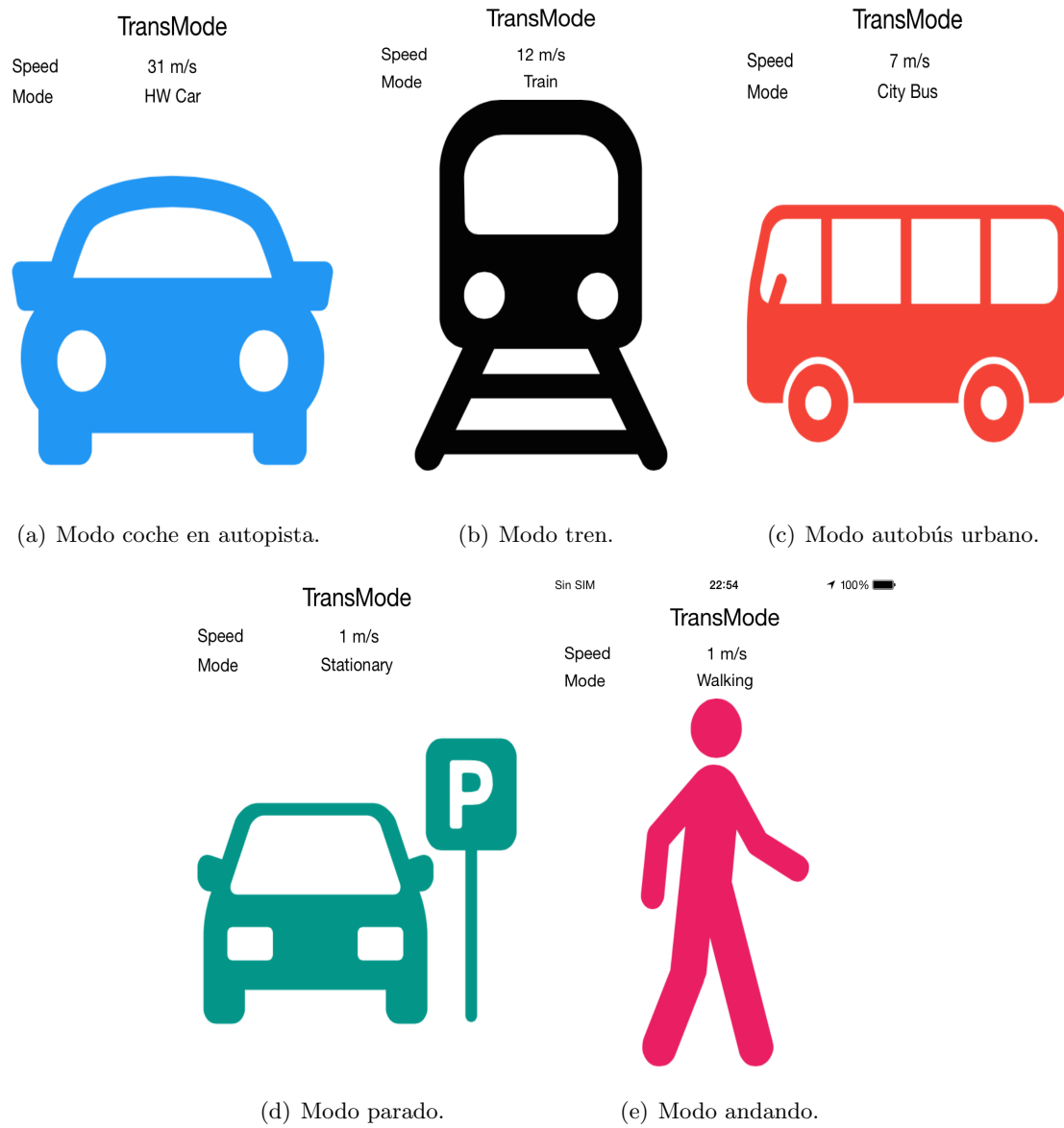


Figura 7.4: Interface TransMode.



## Parte III

# Pliego de condiciones



# Pliego de condiciones

## Requisitos de Hardware

Los requisitos de hardware de este Trabajo Fin de Grado han sido:

- Ordenador MacBook Pro con las siguientes características:
  - CPU de 2.20 GHz.
  - Memoria RAM de 16 GB 1600 MHz DDR3.
  - Tarjeta gráfica Intel HD Graphics 5300.
  - Disco duro de 250 GB.
- Ordenador Samsung Intel Core i5 con las siguientes características:
  - CPU de 2.20 GHz.
  - Memoria RAM de 6 GB.
  - Tarjeta gráfica Intel HD Graphics 4600.
  - Disco duro de 500 GB.
- iPhone 4
- iPhone 4S
- iPhone 5

## Requisitos de Software

Los requisitos de software se presentan a continuación:

- Sistemas operativos OS X Yosemite 10, Linux distribución Ubuntu 14.04.2 LTS y Microsoft Windows 7.
- Herramienta de software matemático Matlab R2014b.
- Entorno de programación Xcode 6.2.
- Librerías de OpenCV para iOS.
- Software de acceso a la información del terminal iExplorer 3.
- Editor de texto para LaTeX Texmaker 4.4.1.



Parte IV

Presupuesto



# Presupuesto

## Costes de ejecución material

En esta sección se adjunta un desglose de los costes estimados para este proyecto..

## Gastos generales y beneficio industrial

Los gastos generales y el beneficio industrial son desembolsos obligatorios que proceden de la utilización de las instalaciones de trabajo y los ingresos generados a nivel de manufactura. Para estas cuentas se estima un porcentaje del 16 % sobre el coste total de ejecución material. Lo resultados son los que se presentan en la Tabla 7.6.

## Presupuesto de ejecución por contrata

En este presupuesto se realiza el sumatorio de los resultados de los costes de ejecución material, los gastos generales y el beneficio industrial.

## Importe total del presupuesto

En último término, se calcula el presupuesto final del proyecto a partir de los costes totales de ejecución por contrata más el porcentaje de IVA del 21 % aplicado.

El importe total del presupuesto asciende a:

**34324€(Treinta y cuatro mil trescientos veinticuatro Euros).**

Concepto	Precio unitario	Cantidad	Subtotal
Ordenador MacBook Pro	2000€	1	2000€
Ordenador Samsung Intel Core i5	700€	1	700€
iPhone 4	250€	1	250€
iPhone 4S	345€	1	345€
iPhone 5	400€	1	400€
<b>Subtotal final</b>			<b>3695€</b>

Tabla 7.2: Costes de hardware

Concepto	Precio unitario	Cantidad	Subtotal
S.O. OS X Yosemite	0€	1	0€
S.O. Ubuntu 14.04.2 LTS	0€	1	0€
S.O. Windows 8	135€	1	135€
XCode 6.2	0€	1	0€
OpenCV 3	0€	1	0€
Matlab R2014b	2000€	1	2000€
iExplorer 3	0€	1	0€
Texmaker 2.1.3	0€	1	0€
<b>Subtotal final</b>			<b>2135€</b>

Tabla 7.3: Costes de software

Concepto	Precio unitario	Cantidad	Subtotal
Ingeniería	25€	640	16000€
Mecanografía	15€	160	2400€
<b>Subtotal final</b>			<b>18400€</b>

Tabla 7.4: Costes de mano de obra

Concepto	Subtotal
Costes de hardware	3695€
Costes de software	2135€
Costes de mano de obra	18400€
<b>Subtotal final</b>	<b>24230€</b>

Tabla 7.5: Costes totales de ejecución material

Concepto	Subtotal
Costes de ejecución material	24230€
Porcentaje estimado	16%
<b>Subtotal final</b>	<b>3287€</b>

Tabla 7.6: Costes totales de ejecución material

Concepto	Subtotal
Costes de ejecución material	24230€
Gastos generales y beneficio industrial	3287€
<b>Subtotal final</b>	<b>23831€</b>

Tabla 7.7: Presupuesto de ejecución por contrata



---

<b>Concepto</b>	<b>Subtotal</b>
Presupuesto de ejecución por contrata	29319€
21 % de IVA aplicado	5005€
<b>Total final</b>	<b>34324€</b>

Tabla 7.8: Importe total del presupuesto



# Bibliografía

- [1] F. Calabrese, M. Diao, G. D. Lorenzo, J. F. Jr., and C. Ratti, “Understanding individual mobility patterns from urban sensing data: A mobile phone trace example,” *Transportation Research Part C: Emerging Technologies*, vol. 26, pp. 301 – 313, 2013.
- [2] D. Cuff, M. Hansen, and J. Kang, “Urban sensing: Out of the woods,” *Commun. ACM*, vol. 51, no. 3, pp. 24–33, mar 2008.
- [3] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson, “People-centric urban sensing,” in *Proceedings of the 2Nd Annual International Workshop on Wireless Internet*, ser. WICON '06. New York, NY, USA: ACM, 2006. [Online]. Available: <http://doi.acm.org/10.1145/1234161.1234179>
- [4] D. Shin, D. Aliaga, B. TunÅşer, S. M. Arisona, S. Kim, D. ZÃ¼nd, and G. Schmitt, “Urban sensing: Using smartphones for transportation mode classification,” *Computers, Environment and Urban Systems*, no. 0, 2014.
- [5] K. K. V. Manzoni, D. Maniloff and C. Ratti, “Transportation mode identification and real-time co2 emission estimation using smartphones,” SENSEable City Lab, Tech. Rep., 2013.
- [6] H. J. Miller, “Beyond sharing: cultivating cooperative transportation systems through geographic information science,” *Journal of Transport Geography*, vol. 31, no. 0, pp. 296 – 308, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0966692313000719>
- [7] J. Gould, “Cell phone enabled travel surveys: The medium moves the message.” *Transport Survey Methods: Best Practice for Decision Making*, vol. 0, pp. 51 – 70, 2013.
- [8] *Campaignr: A Framework for Participatory Data Collection on Mobile Phones.*, 2007.
- [9] F. Alt, A. S. Shirazi, A. Schmidt, U. Kramer, and Z. Nawaz, “Location-based crowdsourcing: Extending crowdsourcing to the real world,” in *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, ser. NordiCHI '10. New York, NY, USA: ACM, 2010, pp. 13–22. [Online]. Available: <http://doi.acm.org/10.1145/1868914.1868921>
- [10] A. C. Santos, J. M. Cardoso, D. R. Ferreira, P. C. Diniz, and P. ChaÃ±ho, “Providing user context for mobile and social networking applications,” *Pervasive and Mobile Computing*, vol. 6, no. 3, pp. 324 – 341, 2010.
- [11] [Online]. Available: <https://support.apple.com/es-es/HT201357>
- [12] [Online]. Available: [https://en.wikipedia.org/wiki/High-pass\\_filter](https://en.wikipedia.org/wiki/High-pass_filter)
- [13] [Online]. Available: <http://es.mathworks.com/help/matlab/ref/mean.html>

- [14] [Online]. Available: <http://es.mathworks.com/help/matlab/ref/std.html>
- [15] R. T. Hastie, T. and J. Friedman., “The elements of statistical learning,” *Springer*, 2008.
- [16] N. Christianini and J. Shawe-Taylor, “An introduction to support vector machines and other kernel-based learning methods,” *Cambridge University Press*, 2000.
- [17] E. Romera and B. L. M., “Driver behavior evaluation by using smartphones,” Master’s thesis, Universidad de Alcala, 2015.