

UNIVERSIDAD MAYOR DE SAN SIMÓN

PROYECTO DE GRADO

Reconocimiento de Patrones en el Manejo de un Conductor

Autor:
Evelyn CUSI LÓPEZ

Tutor:
Dr. Eduardo DI SANTI

*Un proyecto de grado presentado en cumplimiento de los requisitos
para el título de Licenciatura en Ingeniería de Sistemas*

en la

Carrera de Ingeniería de Sistemas
Departamento Informática-Sistemas

8 de mayo de 2018

Agradecimientos

Le agradezco a Dios por haberme acompañado y guiado a lo largo de mi carrera, por ser mi fortaleza en los momentos de debilidad y por brindarme una vida llena de aprendizajes, experiencias y sobre todo felicidad.

A mi madre, por estar conmigo, por enseñarme a crecer y a levantarme, por apoyarme y guiarme, por ser la base que me ayudó a llegar hasta aquí.

Índice general

Agradecimientos	III
Resumen	IX
1. Introducción, justificación, objetivos e hipótesis	1
1.1. Introducción	1
1.2. Justificación	1
1.3. Objetivo general	2
1.4. Objetivos específicos	2
1.5. Hipótesis	2
2. Marco teórico	3
2.1. Reconocimiento de patrones	3
2.1.1. Sensor	4
2.1.2. Extracción de características	4
2.1.3. Clasificación o agrupamiento	4
Clasificación	4
Agrupamiento	4
2.2. Aplicaciones nativas	5
2.3. Arquitectura	5
2.3.1. Aplicación fuera de linea	5
2.3.2. Aplicación totalmente en linea	5
2.3.3. Aplicaciones de sincronización	5
2.3.4. Aplicaciones para la comunicacion entre dispositivos	6
3. Metodología	7
3.1. Selección de Herramientas	7
3.1.1. Selección de motor de Base de Datos	7
Base de Datos Relacionales	7
Base de Datos Documentales	7
3.1.2. Selección de herramientas para el servicio Web	8
Usado por?	8
Usabilidad	8
Flexibilidad	8
Fácil de aprender	8
Uso en el análisis de datos	8
Librerías populares	8
3.1.3. Selección de herramientas para la aplicación móvil	9
3.1.4. Selección de metodología	9

Metodología de la Investigación	9
3.2. Proceso de desarrollo	10
3.2.1. Especificación de requisitos	10
Requisitos funcionales	10
Requisitos no funcionales	10
3.2.2. Plan de trabajo	10
3.2.3. Prototipo	12
4. Bibliografía	17

UNIVERSIDAD MAYOR DE SAN SIMÓN

Resumen

Facultad de Ciencias y Tecnología
Departamento Informática-Sistemas

Licenciatura en Ingeniería de Sistemas

Reconocimiento de Patrones en el Manejo de un Conductor

por Evelyn CUSI LÓPEZ

El presente trabajo plantea qué, con la captura de parámetros de manejo de un conductor mediante el uso de un dispositivo móvil, es posible encontrar patrones de conducción que describan diferentes comportamientos de manejo; y de esta manera al encontrar un patrón anómalo solicitar al conductor que compruebe que es hábil para seguir conduciendo y en caso de que éste no logre hacerlo, en un determinado tiempo, notificar de esta irregularidad a los contactos de emergencia del conductor y así prevenir y/o reducir conductas de manejo riesgosas.

Capítulo 1

Introducción, justificación, objetivos e hipótesis

1.1. Introducción

Los accidentes de tránsito son resultado del avance tecnológico del mundo moderno, ya que a medida que se han incrementado las distancias entre diferentes puntos, se hizo más necesario el uso del automóvil; lo que llevo a consolidar a éste como una herramienta esencial para la vida moderna.

El presente trabajo plantea que con la captura de parámetros de manejo de un conductor, mediante el uso de un dispositivo móvil con sistema operativo Android, es posible encontrar patrones de conducción, usando algoritmos de aprendizaje automático, y determinar si éstos son correctos o en su defecto atentan contra la seguridad de los transeúntes y/o de los otros conductores que comparten las vías con él.

Por lo tanto se espera que mediante el uso de los sensores de un dispositivo móvil y la aplicación de algoritmos de Aprendizaje Automático sea posible reconocer patrones de conducción, que describan diferentes comportamientos de manejo, y así reconocer automáticamente el incumplimientos de algunas de las normas de circulación y conducción de vehículos en las vías de tránsito con el objetivo de reducir los accidentes viales; ya que si este trabajo se aplicara en la seguridad vial sería de gran ayuda para automatizar la detección de anomalías de conducción y así actuar lo antes posible para evitar un posible accidente.

1.2. Justificación

A continuación se detalla la justificación de la realización de este proyecto a través de diferentes puntos de vista

- **Social:** existen dos principales aspectos para la causa de los accidentes de tránsito: el primero es debido al mal diseño y falta de señalización de vías y avenidas y el segundo y aun mas complejo recae en el bajo nivel de conciencia ciudadana para usar vehículos lo que conlleva que muchas personas manejen en estado de ebriedad, manipulando sus dispositivos móviles entre otros. Esta es una situación alarmante ya que el año 2015 existieron 30.556 accidentes de tránsito en Bolivia y 1.301 fallecidos a causas de éstos, según los datos estadísticos del Instituto Nacional de Estadística (INE). Es por ello que se hace necesario contar un mecanismo

que automatice la detección de patrones anómalos en la conducción y así actuar lo antes posible para evitar posibles accidentes.

- **Científico:** los últimos años la inteligencia artificial fue tomando más y más importancia por su gran capacidad de automatizar tareas con el mínimo uso de recursos.
- **Personal:** A nivel personal el objetivo es, mediante los conocimientos adquiridos previamente en mis estudios, ser capaz de gestionar correctamente el proyecto, mejorar mis conocimientos en el desarrollo de aplicaciones nativas para plataforma Android y toda la tecnología relacionada con el proyecto.

1.3. Objetivo general

Desarrollar un mecanismo de reconocimiento de patrones de conducción mediante el uso de un dispositivo móvil y algoritmos de aprendizaje automático, para alertar de manera oportuna el hallazgo de un patrón anómalo en el manejo de un conductor tal como cansancio, ebriedad, o problemas de salud, ej, epilepsia.

1.4. Objetivos específicos

- Capturar los parámetros de manejo de un conductor mediante el uso de los sensores de un dispositivo móvil.
- Analizar los parámetros obtenidos.
- Ajustar los parámetros de conducción al algoritmo de Aprendizaje Automático elegido y generar un modelo de clasificación de patrones.
- Analizar los patrones de conducción obtenidos y determinar aquellos que son anómalos y aquellos que no lo son.
- Validar el modelo con nuevas muestras nunca antes vistas.

1.5. Hipótesis

¿Es posible encontrar patrones de manejo de un conductor mediante parámetros obtenidos por un dispositivo móvil?

Capítulo 2

Marco teórico

El análisis del comportamiento humano es un campo interdisciplinario y en la última década este campo ha atraído más la atención; un ejemplo claro es la aplicación que encuentra la correlación entre los datos del acelerómetro integrado en los teléfonos inteligentes y los niveles de estrés de sus usuarios y así detectar los comportamientos en episodios de estrés de los mismos (García-Ceja, Osmani y Mayora, 2015).

Otro trabajo importante en este ámbito es el que presentan Warren, Lipkowits y Sokolov (2017) donde implementan el procesamiento de extremo a extremo y una canalización de aprendizaje estadístico desde observaciones de los teléfonos inteligentes a la distribución de conductas de conducción, condicional de la hora del día y el día de la semana utilizando algoritmos de agrupación de datos telefónicos.

Warren, Lipkowits y Sokolov (2017), sostienen que gracias a los dispositivos móviles se tiene la oportunidad de complementar la tecnología de recolección de datos tradicional con datos extraídos de los sensores de los teléfonos móviles tales como GPS, giroscopio, acelerómetro y cámara; y esto no sólo es más práctico sino que también es menos costoso debido a que gran parte de la población ya cuenta con un dispositivo móvil de gama media con lo que es más que suficiente para los propósitos de recopilación de datos.

Dado que este trabajo se centrara en recolección de datos y aprendizaje automático es fundamental presentar definiciones necesarias para la comprensión cabal de lo que se abordara en el presente proyecto.

2.1. Reconocimiento de patrones

El reconocimiento de patrones se encarga de la descripción y clasificación (reconocimiento) de objetos, personas, señales, representaciones, etc; trabaja en base a un conjunto previamente establecido de todos los posibles patrones individuales a reconocer. Se puede definir el reconocimiento de patrones como: "la ciencia que se ocupa de los procesos sobre ingeniería, computación y matemáticas relacionados con objetos físicos y/o abstractos, con el propósito de extraer información que permita establecer propiedades entre conjuntos de dichos objetos, los cuales nos permitan interpretar el mundo que nos rodea" (Ruiz-Shulcloper, Guzmán y Martínez-Trinidad, 1999).

Un sistema de reconocimiento de patrones consta de tres partes (Carrasco, J., y Martínez, J., (2011)):

- Sensor
- Extracción de características

- Clasificación o agrupamiento

2.1.1. Sensor

Es la primera etapa del sistema su propósito principal es proporcionar una representación factible de los elementos del universo a ser clasificados, su función es crucial para el sistema ya que determina los límites del rendimiento en todo el sistema.

En la presente investigación se utilizarán los valores de los sensores: GPS, acelerómetro y giroscopio de un dispositivo móvil como atributos (parámetros) del manejo de un conductor.

2.1.2. Extracción de características

Es la selección de atributos relevantes a partir del conjunto total de atributos que se les puede medir a los objetos de estudio; es decir que su función principal es eliminar la redundancia en los atributos con lo que en consecuencia se lograría optimizar la siguiente etapa de este sistema que es la clasificación o agrupamiento. Esta etapa requiere un alto grado de análisis debido a que si las tareas en esta etapa se realizan exitosamente la clasificación y/o agrupamiento obtendrán resultados mucho más precisos por la eliminación de redundancia adecuada.

2.1.3. Clasificación o agrupamiento

El objetivo principal de este subsistema es dado un ejemplo nunca antes visto clasificarlo con el patrón correcto al que pertenece.

Clasificación

Se denomina también como un tipo de aprendizaje supervisado; en este efecto existen una gran cantidad de algoritmos que realizan esta tarea, el problema recae en que algoritmo escoger para un problema determinado lo cual no tiene una solución como tal, ya que la solución más utilizada.^{es} probar con diferentes clasificadores y seleccionar aquel que obtenga los mejores resultados.

Agrupamiento

Se denomina también como un tipo de aprendizaje no supervisado ya que no cuenta con un conocimiento previo de cuantas clases (patrones) existen dado un conjunto de datos de entrenamiento, por lo que su objetivo es encontrar clases que se adecuen a nuestros propósitos dado un conjunto de datos de entrenamiento. El agrupamiento también tiene el problema de cómo seleccionar el algoritmo correcto para agrupar sus datos, para ello se propusieron distintas técnicas, las cuales son:

- Agrupamientos jerárquicos
- Técnica de reagrupamiento
- Agrupamiento basado de grafos

2.2. Aplicaciones nativas

Las aplicaciones nativas son las aplicaciones propias de cada plataforma. Deben ser desarrolladas pensando en la plataforma concreta. No existe ningún tipo de estandarización, ni en las capacidades ni en los entornos de desarrollo, por lo que los desarrollos que pretenden soportar plataformas diferentes suelen necesitar un esfuerzo extra (Ramirez, PID 00176755).

2.3. Arquitectura

Existen muchas arquitecturas de aplicación posibles en lo que respecta a las aplicaciones móviles. Las arquitecturas más habituales en el desarrollo de aplicaciones para dispositivos móviles son:

2.3.1. Aplicación fuera de línea

Las aplicaciones "fuera de línea" son aplicaciones que, una vez descargadas, no requieren en absoluto de conexión (a excepción de las actualizaciones) para poder funcionar. Estas aplicaciones solo necesitan desarrollar la aplicación del dispositivo móvil (no son necesarios más componentes).

2.3.2. Aplicación totalmente en línea

Las aplicaciones totalmente "en línea" son aplicaciones que no pueden funcionar sin conexión a Internet. Estas arquitecturas requieren, sin lugar a dudas, de una parte de servidor, y están pensadas para mantener una comunicación constante con dicha parte servidora.

Tienen como desventaja que el usuario no puede utilizar la aplicación cuando no tiene conexión, pero disponen de información constante de las interacciones del usuario. Es necesario desarrollar, al menos, la parte servidora, tal vez una parte de desarrollo en el cliente y, en ocasiones, la comunicación entre ambos. Al necesitar estar siempre conectados, tienen un consumo extra de batería. En ocasiones no es necesario realizar la parte servidora, ya que se trata de aplicaciones llamadas mash-ups, que son aplicaciones que aprovechan API existentes en la red para interactuar con datos, (como, por ejemplo, la API de Twitter o datos públicos del estado).

2.3.3. Aplicaciones de sincronización

Las aplicaciones de sincronización son aplicaciones que pueden funcionar en ambos modos, "en línea" y "fuera de línea", y permiten realizar las mismas acciones o acciones muy parecidas en ambos casos. La aplicación debe sincronizar los datos de la situación "fuera de línea" cuando se encuentre "en línea" y gestionar los posibles conflictos. Esto supone un beneficio para el usuario, ya que le permite trabajar en cualquier lugar y tener la información lo más actualizada posible.

2.3.4. Aplicaciones para la comunicacion entre dispositivos

Las aplicaciones para la comunicación entre dispositivos son aplicaciones que interconectan dos (unicast) o más (multicast) dispositivos e intercambian información.

Capítulo 3

Metodología

3.1. Selección de Herramientas

3.1.1. Selección de motor de Base de Datos

Actualmente existe una gran cantidad de oferta de sistemas de bases de datos, aunque en modelo de base de datos mas habitual es el relacional, que se vincula estrechamente con el **lenguaje SQL**.

Base de Datos Relacionales

Las bases de datos relacionales se basan en almacenar, relacionar, consultar y transferir datos dentro de tablas y registros de tuplas. Las tablas se relacionan entre si por medio de campos compartidos y se pueden establecer muchas relaciones entre las mismas formando un entramado de datos super-organizado y estructurado.

Se puede decir que al principal problema de las bases de datos relacionales es precisamente que al crecer el volumen de datos suelen crecer las estructuras (Scale Up) y se convierten en verdaderos laberintos de conexiones que hacen que la organización y gestión de los datos pueda llegar a ser pesada y ralentice las famosas Queries así como la transmisión de datos.

Otro problema importante de estas bases de datos es que al cambiar la estructura de una sola tabla puede provocar impacto en otras muchas, cualquier variación supone un coste en compatibilidad, tiempo y recursos.

Base de Datos Documentales

Las bases de datos documentales tales como MongoDB o PostgreS centran su valor en almacenar los datos en documentos en lugar de tablas, estos **documentos** son recipientes de datos en un formato semi-estructurado, que permite que en un mismo tipo de recipiente se almacene distintos tipos de datos en un mismo tipo de elementos. Por lo tanto se podría decir que las bases de datos documentales tienen la ventaja de ser super-flexibles, se adaptan a los cambios y tienen optimización de consultas en base de datos para grandes cantidades de datos.

Este objetivo de este proyecto es monitorear el manejo de un conductor, lo que esto conlleva que se hará uso de una gran cantidad de datos, haciendo las Bases de Datos Documentales ideales para este propósito, para la etapa del desarrollo del prototipo se

propone el uso de MongoDB para el almacenamiento de datos del manejo (almacenaje de datos del lado del servidor) y SQLite como almacenamiento interno de los datos propios de la aplicación.

3.1.2. Selección de herramientas para el servicio Web

Básicamente se usara el servicio Web para el entrenamiento y predicción de patrones de manejo del conductor, por lo que es necesario encontrar una herramienta que se adecue a este propósito. A continuación se detallaran algunos puntos principales por lo que se decidió usar el lenguaje de programación Python en esta etapa.

Usado por?

Python es utilizado por programadores que quieren profundizar en el análisis de datos o aplicar técnicas estadísticas, y por desarrolladores que recurren a datos científicos.

Usabilidad

La codificación y la depuración es más fácil de hacer en Python, principalmente debido a la sintaxis "agradable", además cualquier pieza de funcionalidad siempre se escribe de la misma manera en Python lo que no es igual en otras herramientas como por ejemplo R.

Flexibilidad

Python es flexible para hacer algo nuevo que nunca se haya hecho antes. Los desarrolladores también pueden usarlo para crear scripts en un sitio web u otras aplicaciones.

Fácil de aprender

El enfoque de Python en legibilidad y simplicidad hace que su curva de aprendizaje sea relativamente baja y gradual. Python se considera un buen lenguaje para iniciar programadores.

Uso en el análisis de datos

Python se usa generalmente cuando las tareas de análisis de datos deben integrarse con aplicaciones web o si el código de estadísticas debe incorporarse en una base de datos de producción.

Librerías populares

- **Pandas** para manipular datos fácilmente.
- **SciPy / NumPy** para la informática científica.
- **sckikit** aprende a utilizar los métodos de aprendizaje automatico.
- **matplotlib** para hacer gráficos.

- **statsmodels** para explorar datos, estimar modelos estadísticos y realizar pruebas estadísticas y pruebas unitarias.

3.1.3. Selección de herramientas para la aplicación móvil

La aplicación móvil para el monitoreo del manejo de conducción requiere el uso de uno o mas sensores propios del dispositivo móvil, por lo que se vio optimo el uso de desarrollar la misma de manera nativa, mas propiamente dicha en **Android nativo** debido a que este tipo de aplicaciones tienen menos capas para poder llegar las funcionalidades propias del dispositivo móvil para realizar una acción, por lo tanto si se desea acceder a la cámara, GPS o sensores del dispositivo, el código esta optimizado para que la funcionalidad se lleve a cabo rápidamente.

Entre otras ventajas de desarrollo de aplicaciones nativas se pueden nombrar las siguientes:

- Mejor rendimiento
- Menor consumo de memoria
- Mayor velocidad
- Aprovechamiento total del hardware del dispositivo(Camara, GPS, Sensores, entre otros).

3.1.4. Selección de metodología

Metodología de la Investigación

La **metodología de la investigación** es una disciplina de conocimiento encargada de elaborar, definir y sistematizar el conjunto de técnicas, métodos y procedimientos que se deben seguir durante el desarrollo de un proceso de investigación para la producción de conocimiento.

Orienta la manera en que vamos a enfocar una investigación y la forma en que vamos a recolectar, analizar y clasificar los datos, con el objetivo de que nuestros resultados tengan validez y pertinencia, y cumplan con los estándares de exigencia científica.

La **metodología de la investigación**, en este sentido, es también la parte de un proyecto de investigación donde se exponen y describen razonadamente los criterios adoptados en la elección de la metodología, sea esta **cuantitativa** o **cualitativa**.

- **Metodología cuantitativa** La metodología cuantitativa es aquella empleada por las ciencias naturales o fácticas, que se vale de datos cuantificables a los cuales accede por observación y medición.

Para su análisis, procede mediante la utilización de las estadísticas, la identificación de variables y patrones constantes. Su método de razonamiento es deductivo, para lo cual trabaja con base en una muestra representativa del universo estudiado.

Por lo tanto para el cumplimiento del objetivo se hará el uso de una metodología de investigación cuantitativa debido al análisis de datos que se requiere.

3.2. Proceso de desarrollo

3.2.1. Especificación de requisitos

El presente proyecto conlleva una serie de requisitos funcionales explícitos y otra serie de requisitos no funcionales implícitos los cuales son presentados a continuación:

Requisitos funcionales

Los requisitos del presente proyecto constan de módulos principales con un serie de funcionalidades requeridas.

El la tabla CUADRO 3.1 se puede observar una columna con el nombre principal del modulo, otra columna con las funcionalidades requeridas para dicho modulo y una columna de descripción de cada una de las funcionalidades.

Requisitos no funcionales

Estos requisitos son aquellos que aunque no estan especificados en el trabajo, son indispensables en el desarrollo del mismo.

- **Entorno de desarrollo y compilación** El desarrollo de la aplicación se llevará a cabo usando Android Studio como entorno de desarrollo con las herramientas necesarias para la creación de aplicaciones en toda clase de dispositivos Android.
- **Control de versiones de código fuente** Se usará un sistema de control de versiones para el código fuente de la aplicación de tipo GIT, el elegido es Gitlab, ya que no obliga a hacer público el repositorio.
- **Persistencia de datos** Los datos de la aplicación deben almacenarse en un sistema, para ello se hará uso de un backend en el que se almacenarán los objetos necesarios.
- **Lenguaje de programación** Para desarrollar de forma nativa para Android se utilizará el lenguaje de programación Java, requisito imprescindible para tal fin.
- **Entorno de ejecución** Se proporcionará un ejecutable en formato apk para poder instalar en dispositivos Android.
- **Pruebas** Se realizarán las pruebas unitarias con JUnit y los test de usuario que resulten necesarios.
- **Documentación** Se proporcionara un Documento Final y diapositivas explicativas del modo de uso de la aplicación móvil.

3.2.2. Plan de trabajo

A continuacion se detalla de forma resumida los objetivos y entregables de cada fase.

1. **Plan de trabajo** *Objetivos:* Definir el proyecto y realizar su planificación inicial. *Entregables:* Plan de trabajo. El plan de trabajo servirá como guía para el resto del desarrollo.

MODULO	FUNCIONALIDAD REQUERIDA	DESCRIPCIÓN
Gestión de usuarios	Creación de cuentas de usuario	La creacion de cuentas como pre-requisito de acceso a la aplicación
	Ingreso a la aplicación	Para mantener la privacidad de los datos del usuario, el acceso a la aplicación se realiza mediante un ingreso con nombre de usuario y contraseña.
Captura de datos	Lectura de sensores	Captura de datos de sensores necesarios para encontrar patrones de manejo en la conducción de un específico usuario.
	Envío y almacenaje de datos al servidor	Captura y almacenaje de datos de los sensores en el servidor.
Entrenamiento	Captura de datos de entrenamiento	Almacenaje de datos en el servidor para su posterior entrenamiento.
	Generación de un modelo de predicción	Generar un modelo de predicción con los datos de entrenamiento proporcionados y un algoritmo de aprendizaje no supervisado.
Predicción	Predecir si un patrón es anómalo o no	Con el modelo generado en la fase de entrenamiento determinar si el conductor conduce de manera correcta o anómala según el entrenamiento proporcionado previamente.
Alertas	Consultar capacidad de conducción	Consultar estado del conductor en caso de que el sistema encuentre un patrón anómalo en la conducción
	Enviar notificaciones contactos de emergencia	En caso de que el conductor no pueda confirmar que se encuentra capaz de seguir conduciendo el sistema enviara notificaciones de alerta a los contactos de emergencia del conductor con la ubicación GPS del mismo.
Incrementar el aprendizaje	Agregar nuevos datos a la rutina de conducción	En caso de que el conductor confirme que se encuentra conduciendo bien, previa consulta realizada se procederá a incrementar este patrón al modelo de predicción.

CUADRO 3.1: Requerimientos funcionales

2. **Análisis, diseño y prototipo** *Objetivos:* Realizar el análisis de tareas, diseño y prototipo. *Entregables:* Documentación que incluye escenarios de uso, diagramas de flujo de interacción y un prototipo de alto nivel.
3. **Implementación** *Objetivos:* Implementar la solución del proyecto y documentación complementaria. *Entregables:* Código fuente, instalables y documentación complementaria. Es la fase final del proyecto en la que se obtiene el producto final. Puede requerir iterar con las fases anteriores en caso de detectar problemas en el producto.
4. **Entrega final** *Objetivos:* Finalizar el proyecto y documentarlo. *Entregables:* Documento y diapositiva de presentación del proyecto. En esta fase se finalizará la fase anterior en caso de no haberlo hecho y se presentará al público objetivo nuestro producto.

3.2.3. Prototipo

Para instalar la aplicación se debe tener la apk en el dispositivo móvil, y también tener habilitado la opción orígenes desconocidos como se muestra en la Figura 3.1. A

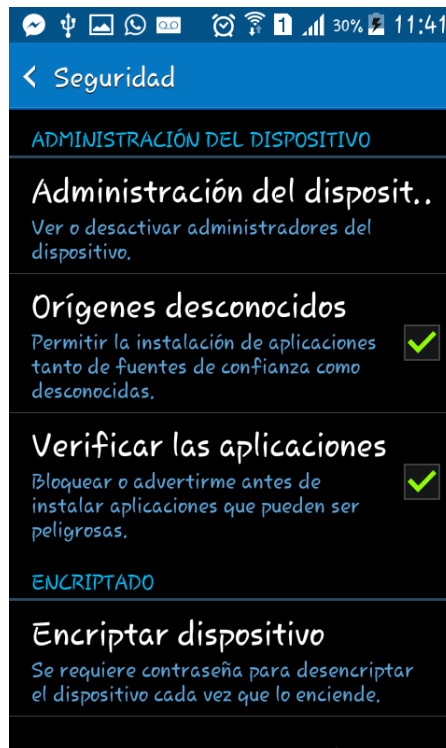


FIGURA 3.1: Habilitar orígenes desconocidos

continuación se muestran pantallas del prototipo inicial.

- **Icono de la aplicación**

Una vez instalada la aplicación se puede observar en el dispositivo una nueva aplicación denominada "Driving App" como se puede ver en la Figura 3.2.

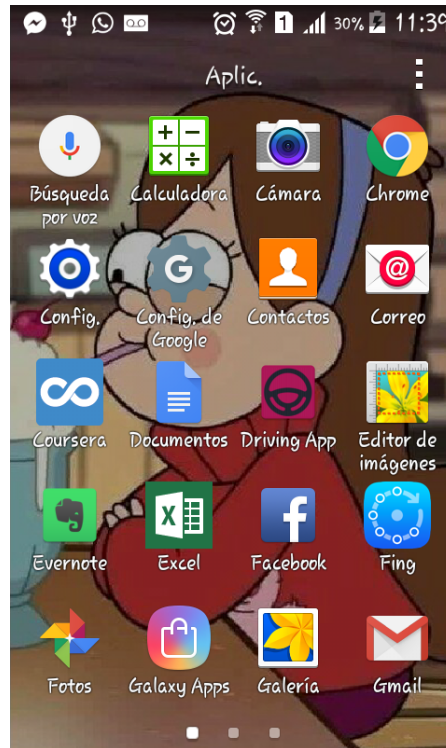


FIGURA 3.2: Icono de la aplicación

- **Ingreso a la aplicación** Al abrir la aplicación se puede observar una pantalla de logueo como se muestra en la Figura 3.3.; donde para ingresar a la misma se pone root como nombre de usuario y "123" como contraseña.
- **Pantalla principal** La pantalla principal consta de un botón (Ver Figura 3.4) el cual sirve para empezar a guardar los datos (de manera local en el prototipo) de conducción (datos capturados del acelerómetro en la etapa de prototipo).
- **Menú** El menú consta de dos opciones que se detallarán más adelante (Ver Figura 3.5).
- **Funcionalidad agregar contactos** Esta funcionalidad se habilita al presionar en el menú la opción "Agregar contactos"; consta de agregar contactos de emergencia para el usuario, su apariencia es como se muestra en la Figura 3.6.
- **Listar contactos** Esta funcionalidad se habilita al presionar en el menú la opción "Mis contactos"; consta en listar los contactos de emergencia para el usuario, su apariencia es como se muestra en la Figura 3.7.

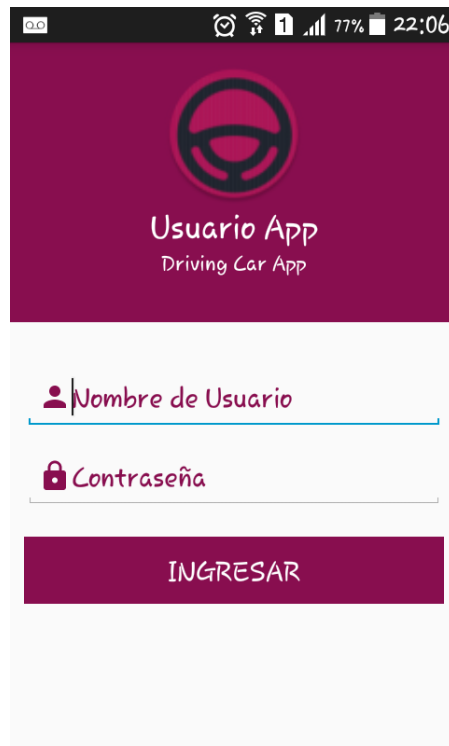


FIGURA 3.3: Ingreso a la aplicación



FIGURA 3.4: Pantalla principal de la aplicación



FIGURA 3.5: Menú de la aplicación

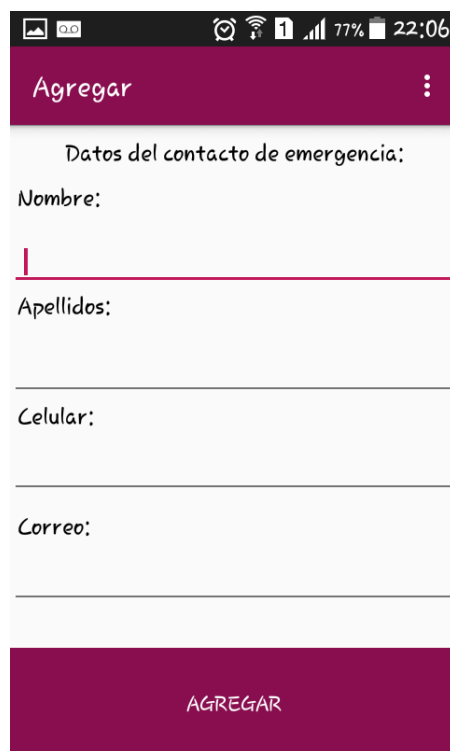


FIGURA 3.6: Opción agregar contactos

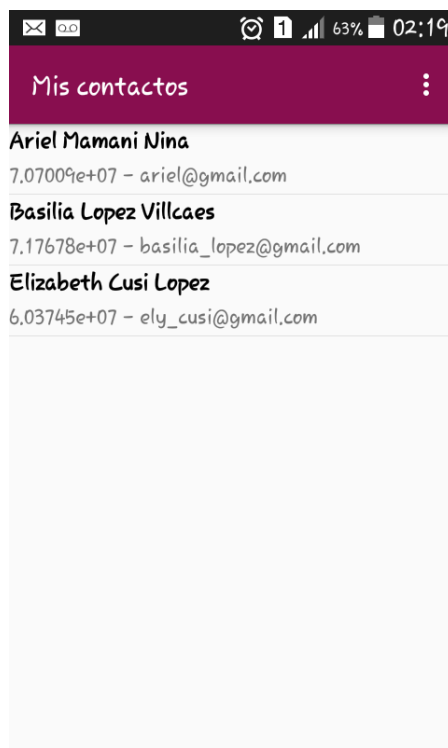


FIGURA 3.7: Opción mis contactos

Capítulo 4

Bibliografía

1. Carrasco, J., y Martinez, J., (2011). Reconocimiento de patrones. *Rev. Komputer Sapiens, III*(Vol. II.), 5-9.
2. Ruiz-Shulcloper J., Guzmán Arenas A., Martinez-Trinidad J.F. (1999) “Enfoque Lógico Combinatorio al Reconocimiento de Patrones I”, *Selección de Variables y Clasificación Supervisada*, IPN, México
3. Warren, J., y Sokolov, V., (2017). Clusters of Driving Behavior from Observational Smartphone Data. *arXiv:1710.04502v2*(cs.AI)
4. Garcia-Ceja, E., Osmani, V., y Mayora, O., (2015). Automatic Stress Detection in Working Environments from Smartphones Accelerometer Data: A First Step. *arXiv:1510.04221v1*(cs.HC)
5. Ramirez, R., (PID 00176755) Metodos para el desarrollo de aplicaciones moviles.