

UNIVERSIDAD NACIONAL DE ASUNCIÓN

FACULTAD POLITÉCNICA

INGENIERÍA EN INFORMÁTICA



Trabajo Final de Grado como requisito para obtener el título de Ingeniero en
Informática

“Evaluación de la calidad de la experiencia para servicios de datos móviles: un enfoque basado en Machine Learning”

Autores:

Brenda R. Quiñónez
Luis F. Fernández

Supervisores:

D.Sc. Diego Pinto Roa
D. Sc. Carlos Núñez Castillo
Ms. María García Díaz

29 de julio de 2015

Este trabajo está dedicado a nuestras familias, profesores y compañeros por el apoyo incondicional durante este arduo proceso.

Agradecimientos

A nuestros supervisores: MsC. María Elena García, PhD. Diego Pinto y PhD. Carlos Núñez por sus consejos y sus esfuerzos en el empeño de salir adelante con este proyecto.

A la Facultad Politécnica por darnos la oportunidad de llevar a cabo este proyecto de grado y por sobre todo a los alumnos, funcionarios y docentes que nos ayudaron durante el proceso de realización de las pruebas.

Al Ing. Joaquín Lima por abrirnos las puertas de Konecta S.A. para realizar las pruebas dentro de la entidad.

Principalmente, a nuestros familiares, amigos y compañeros, por brindarnos su comprensión, fuerza y apoyo en todo momento.

Resumen

En la actualidad, se experimenta un importante aumento en el tráfico de datos a través de las redes móviles. Cuando existe gran cantidad de datos intercambiados, uno de los principales desafíos a afrontar es la degradación de la calidad percibida por los usuarios de los servicios móviles.

Desde el punto de vista de las empresas de telecomunicaciones, la calidad de la experiencia (QoE) percibida por los usuarios móviles juega un papel importante en el posicionamiento de las empresas y en la fidelidad de los usuarios. Este hecho ha llevado a las empresas de telecomunicaciones y a la comunidad científica a desarrollar métodos y estrategias que permitan medir el QoE. Muchos estudios se han centrado en identificar la relación entre los parámetros de la calidad de servicio (QoS) y la percepción del usuario. Algunos se han enfocado en identificar la relación entre QoE y QoS para un tipo de tráfico determinado, mientras que otros han tratado de definir un modelo genérico que establezca una relación entre ambos conceptos.

Este trabajo se enfoca en el establecimiento de un modelo estimación del QoE considerando parámetros a nivel de red, dispositivo y aplicación para dos contextos de uso habituales: navegación web y video streaming.

Para este fin, se implementan una plataforma de transmisión de datos móviles y una aplicación Android que son utilizadas durante el proceso de recolección de datos. Los datos obtenidos se emplean como entrada de distintos algoritmos de aprendizaje automático, cuyo desempeño es comparado con el fin de identificar los más adecuados para la creación de un modelo que permita estimar el QoE en los contextos de uso mencionados.

Abstract

Nowadays, a significant increase in mobile data traffic is experienced. When large data is exchanged, one of the main challenges is the degradation of the quality perceived by the users.

From the point of view of operators, quality of experience (QoE) plays an important role in the business model and is a key success factor for the positioning of telecommunications companies and loyalty of users. This has led to mobile network operators and the scientific community to develop methods and strategies to measure the QoE. Many studies have focused on identifying the relationship between the parameters of the quality of service (QoS) and user perception. Some have focused on identifying the relationship between QoE and QoS for a given type of traffic, while others have attempted to define a generic model that establishes a relationship between the two concepts.

This work focuses on the establishment of a monitoring and QoE model evaluation within mobile video streaming context considering parameters from three layers: network, device and application.

To this end, a platform for mobile data transmission and an Android application were created, in order to collect QoE data to be used as input for various machine learning algorithms. These algorithms were compared in order to identify those most suitable for the creation of a correlation model for estimating the QoE in the context of mobile video streaming.

Contenido

Contenido	5
Lista de Figuras	9
Lista de Tablas	12
Lista de Algoritmos	14
Símbolos	16
1. Introducción	1
1.1. Motivación	3
1.2. Objetivos	3
1.2.1. Objetivos específicos	4
1.3. Organización del libro	4
2. Monitoreo de los servicios y la experiencia	5
2.1. QoS y QoE	6
2.2. Factores influyentes en el QoE	7
2.3. Monitoreo de QoE y QoS	8
2.3.1. Monitoreo de QoS	9
2.3.2. Monitoreo de QoE	11
2.4. Resumen	12
3. Evaluación de la calidad de la experiencia	13
3.1. Clasificación de aplicaciones según el tipo de tráfico.	14
3.2. Clasificación de los métodos de evaluación de QoE	15
3.2.1. Pruebas subjetivas	15
3.2.2. Métodos objetivos	16

3.3. Pseudo-Subjective Quality Assessment - PSQA	18
3.4. Revisión literaria acerca de la evaluación del QoE	20
3.5. Discusión	27
4. Aprendizaje automático	28
4.1. Introducción	28
4.1.1. Aplicaciones y problemas	29
4.1.2. Definiciones	30
4.1.3. Tipos de aprendizaje	30
4.2. El problema de la clasificación	31
4.3. Métodos de aprendizaje automático	31
4.3.1. Redes neuronales artificiales	32
4.3.2. Máquinas de soporte vectorial (Support Vector Machine - SVM)	34
4.3.3. Árboles de decisión	35
4.3.4. Tabla de decisiones	36
4.3.5. Clasificadores bayesianos	37
4.3.6. Clasificadores basados en instancias	38
4.3.7. Sistemas ensamblados (Ensemble Systems)	40
4.4. Estimación de Error	41
4.4.1. Validación cruzada - Cross Validation (CV)	42
4.4.2. Stratified k-fold Cross Validation	44
4.5. Resumen	45
5. Modelado del problema	46
5.1. Metodología de evaluación	46
5.2. Plataforma de transmisión de datos móviles	49
5.2.1. Servidor transmisor	50
5.2.2. Servidor de configuración	50
5.2.3. Punto de acceso WiFi	51
5.2.4. Dispositivos móviles	51
5.2.5. Servidor de persistencia	51
5.3. Aplicación móvil Android	52
5.3.1. Prueba subjetiva	52
5.3.2. Agente móvil	54
5.4. Recolección de datos	55
5.5. Análisis y selección de Datos	57

5.6. Resumen	58
6. Entrenamiento y comparación de algoritmos de clasificación	59
6.1. Resultados de las pruebas subjetivas	59
6.2. Definición de los conjuntos de datos	60
6.3. Distribución de clases	61
6.4. Experimentación	65
6.4.1. Configuración del experimento	67
6.5. Métricas de evaluación	67
6.5.1. Porcentaje de aciertos	68
6.5.2. Kappa	68
6.5.3. Medida-F	69
6.6. Análisis de resultados	70
6.6.1. Video streaming	71
6.6.2. Navegación web	81
6.7. Discusión	95
6.8. Obtención del Modelo de Correlación	97
7. Conclusiones	99
7.1. Conclusiones Generales	99
7.2. Aporte de este trabajo	100
7.3. Trabajos Futuros	100
Referencias	102
Apéndices	107
A. Detalles de los contenidos del servidor transmisor	109
A.1. Contenidos multimedia	109
A.1.1. Video 1	109
A.1.2. Video 2	110
A.1.3. Video 3	110
A.1.4. Página Web	111
B. Servidor de Configuración	112
B.1. Conceptos	112
B.2. Variación de parámetros de red	112

C. Detalles de la Aplicación Android	116
C.1. Prueba Subjetiva	116
C.2. Herramientas de Medición del Agente Móvil	121
C.2.1. Uso de CPU	121
C.2.2. Uso de RAM	121
C.2.3. Red	121
C.2.4. Parámetros a nivel de aplicación	123
D. Interpretación de gráficos	125
D.1. Diagrama de Cajas - Boxplot	125
E. Influencia de parámetros sobre el QoE	127

Lista de Figuras

1.1.	Diagrama del proyecto MOSAIC	2
2.1.	Concepto de QoE y QoS	7
2.2.	Factores que influyen en la calidad de la experiencia	8
2.3.	Elementos funcionales de un agente móvil QoS	11
4.1.	Ejemplos de clasificación	31
4.2.	Modelo de una neurona artificial	33
4.3.	Ejemplo de una red neuronal	34
4.4.	Clasificación binaria por medio de una máquina de soporte vectorial . .	35
4.5.	Ejemplo de un árbol de decisión	36
4.6.	Ejemplo de un sistema ensamblado	41
4.7.	Ejemplo de validación cruzada	43
5.1.	Vista general del proceso seguido para la obtención del modelo de co- rrelación QoE.	48
5.2.	Plataforma de transmisión de datos móviles	49
5.3.	Proceso de cada prueba subjetiva	56
6.1.	Distribución de clases para el conjuntos de datos de streaming de video. .	61
6.2.	Distribución de clases para el conjunto de datos de navegación web. . .	63
6.3.	Distribución de clases del conjunto de datos de streaming de video ba- lanceado	64
6.4.	Distribución de clases del conjunto de datos de navegación web balanceado	65
6.5.	Diagrama de cajas con respecto al porcentaje de aciertos considerando las datos de video streaming desbalanceado.	73
6.6.	Diagrama de caja con respecto al porcentaje de aciertos considerando las datos de video streaming balanceado.	74

Lista de Figuras

6.7. Diagrama de cajas con respecto a la métrica kappa considerando las datos de video streaming desbalanceado.	77
6.8. Diagrama de cajas con respecto a la métrica kappa considerando las datos de video streaming balanceado.	78
6.9. Diagrama de cajas con respecto a la métrica medida-F.	80
6.10. Diagrama de cajas con respecto a la métrica medida-F.	81
6.11. Diagrama de cajas con respecto al porcentaje de aciertos considerando las datos de navegación web desbalanceado.	84
6.12. Diagrama de cajas con respecto al porcentaje de aciertos considerando las datos de navegación web balanceado.	85
6.13. Diagrama de cajas con respecto a la métrica Kappa considerando los datos de navegación web desbalanceado.	87
6.14. Diagrama de cajas con respecto a la métrica Kappa considerando los datos de navegación web balanceado.	88
6.15. Diagrama de cajas con respecto a la métrica Medida F considerando las datos de navegación web desbalanceado.	90
6.16. Diagrama de cajas con respecto a la métrica Medida F considerando las datos de navegación web balanceado.	91
6.17. Esquema del sistema ensamblado por mayoría de votos	93
6.18. Resultados de la ejecución del algoritmo ensamblado	94
6.19. Resultados de la ejecución del algoritmo ensamblado	95
 C.1. Inicio Prueba Subjetiva	116
C.2. Datos (1)	116
C.3. Datos (2)	116
C.4. Test PING	117
C.5. Test IPERF	117
C.6. Características del dispositivo	117
C.7. Test de Navegación Web	118
C.8. Página Web	118
C.9. Evaluación de la Navegación Web	118
C.10. Inicio del Test de Video	119
C.11. Evaluación del Video	119
C.12. Video 1	119
C.13. Video 2	119

Lista de Figuras

C.14.Video 3	119
C.15.Envío de Datos	120
C.16.Fin de la Prueba Subjetiva	120
D.1. Ejemplo de diagrama de caja	126
E.1. Variación Ancho de Banda - MOS en video streaming	127
E.2. Variación Ancho de Banda - MOS en navegación web	128
E.3. Variación Delay - MOS en video streaming	129
E.4. Variación Delay - MOS en navegación web	130
E.5. Variación Paquetes perdidos - MOS en video streaming	131
E.6. Variación Paquetes perdidos - MOS en navegación web	132
E.7. Variación Jitter - MOS en video streaming	133
E.8. Variación Jitter - MOS en navegación web	134
E.9. Variación Uso CPU - MOS en video streaming	135
E.10.Variación Uso CPU - MOS en navegación web	136
E.11.Variación Uso RAM - MOS en video streaming	137
E.12.Variación Uso RAM - MOS en navegación web	138
E.13.Variación tiempo de carga de página - MOS en navegación web	139
E.14.Variación carga inicial video - MOS en video streaming	140
E.15.Variación cantidad de Buffering - MOS en video streaming	141
E.16.Variación tiempo promedio de Buffering - MOS en video streaming	142

Lista de Tablas

3.1.	Clasificación de ejemplos de aplicaciones según el tipo de tráfico [SCJ11].	15
3.2.	Escala Absoluta MOS [ITU08]	16
3.3.	Resumen de la revisión literaria acerca de la evaluación de QoE.	26
5.1.	Parámetros de red establecidos en el servidor de configuración	55
5.2.	Datos recolectados durante el proceso de recolección de datos.	57
6.1.	Atributos de las instancias de los conjuntos de datos	62
6.2.	Algoritmos de clasificación utilizados en el proceso de experimentación.	66
6.3.	Matriz de confusión abstracta.	68
6.4.	Resultados de la ejecución de los algoritmos de aprendizaje con respecto a la métrica de porcentaje de aciertos. Se muestra en negrita los algoritmos que en promedio obtuvieron el mejor desempeño.	72
6.5.	Resultados de la ejecución de los algoritmos de aprendizaje con respecto a la métrica Kappa.	76
6.6.	Resultados de la ejecución de los algoritmos de aprendizaje con respecto a la métrica medida-F.	79
6.7.	Resultados de la ejecución de los algoritmos de aprendizaje con respecto a la métrica de porcentaje de aciertos.	83
6.8.	Resultados de la ejecución de los algoritmos de aprendizaje con respecto a la métrica Kappa.	86
6.9.	Resultados de la ejecución de los algoritmos de aprendizaje con respecto a la métrica Medida F.	89
6.10.	Resultados de la ejecución del sistema ensamblado con los conjuntos de datos de navegación web y video streaming.	92
A.1.	Metadatos del video 1.	109
A.2.	Metadatos del video 2.	110

Lista de Tablas

A.3. Metadatos del video 3.	110
A.4. Componentes de la página web de prueba.	111
B.1. Parámetros de red establecidos en el servidor de configuración	115

Lista de Algoritmos

1.	Algoritmo k-Nearest Neighbor	40
2.	Algoritmo k-fold CV [JS11a]	44
3.	Algoritmo Stratified k-fold CV [JS11a]	45

Símbolos

Acrónimos / Abreviaciones

AP Access Point - Punto de Acceso

CN Core Network

CQM Customer QoS management - Gestión de QoS del Cliente

CV Cross-Validation - Validación Cruzada

EML Element Management Layer - Capa de manejo de elemento

GSM Global System for Mobile communications

ICMP Internet Control Message Protocol - Protocolo de Mensajes de Control de Internet

ISP Internet service provider - Proveedor de servicios de Internet

k-fold CV K-Fold Cross-Validation

kNN K-Nearest Neighbor

KPI Key Performance Indicators - Indicadores clave de rendimiento

LTE Long Term Evolution

MBO Mobile Data Offloading

MNO Mobile Network Operator - Operador de redes móviles

MOS Mean Opinion Score

MPS Mobile Phone Sensing

Símbolos

NE Network Element - Elemento de red

NML Network Management Layer - Capa de manejo de red

NMS Network Management System - Sistema de administración de red

PSQA Pseudo-Subjective Quality Assessment - Evaluación de la calidad pseudo-subjetiva

QoE Quality of experience - Calidad de la experiencia

RNN Random Neural Network - Red Neuronal Aleatoria

SMO Sequential Minimal Optimization

SQM Service quality management - Gestión de calidad del Servicio

SVM Support Vector Machine - Máquina de Soporte Vectorial

UE User Equipment - Equipo de usuario

Capítulo 1

Introducción

En los últimos años ha existido un incremento significativo en el desarrollo y la investigación en torno a las redes móviles. Esto ha ido acompañado al avance de los terminales móviles (como los *smartphones* y las *tablets*).

Cada vez surgen más dispositivos con potentes capacidades multimedia y tecnologías de acceso a Internet como: 3G, LTE, WiFi, ofreciendo al usuario la posibilidad de estar conectado desde cualquier lugar y en el momento que lo desee. Por tal motivo, han cambiado radicalmente los modelos de negocio que se han utilizado en el mundo de las prestadoras de servicios de telefonía móvil. La naturaleza de los datos intercambiados ha pasado de voz y texto plano (llamadas y SMS) a un conjunto de datos más amplio, tales como: emails, imágenes, archivos de sonido y videos, entre otros [SLC07].

La gran proliferación de dispositivos móviles, la facilidad de acceso a Internet proporcionada a los usuarios y los avances tecnológicos en la infraestructura de las redes móviles han generado una gran demanda por los servicios de datos móviles. Como resultado de ello, actualmente se experimenta un importante aumento del tráfico de datos a través de las redes móviles, lo que ocasiona la saturación de las infraestructuras de las redes móviles actuales y en consecuencia, la degradación de la calidad percibida por los usuarios [MOS13].

Con el fin de satisfacer dicha demanda, los enfoques de descarga de datos que utilizan tecnologías complementarias (Femtocells, WiFi) son aplicados para los diferentes usuarios finales. Sin embargo, aunque los mecanismos parecen ser eficientes y escalables, los modelos básicos de descarga utilizan algoritmos simplistas sin garantizar la calidad de los servicios [MOS13]. Por otra parte, con los mecanismos utilizados actualmente, los operadores no tienen visibilidad de las condiciones que afectan a la calidad de la experiencia. De hecho, la migración de los datos de diferentes redes colaborati-

1. Introducción

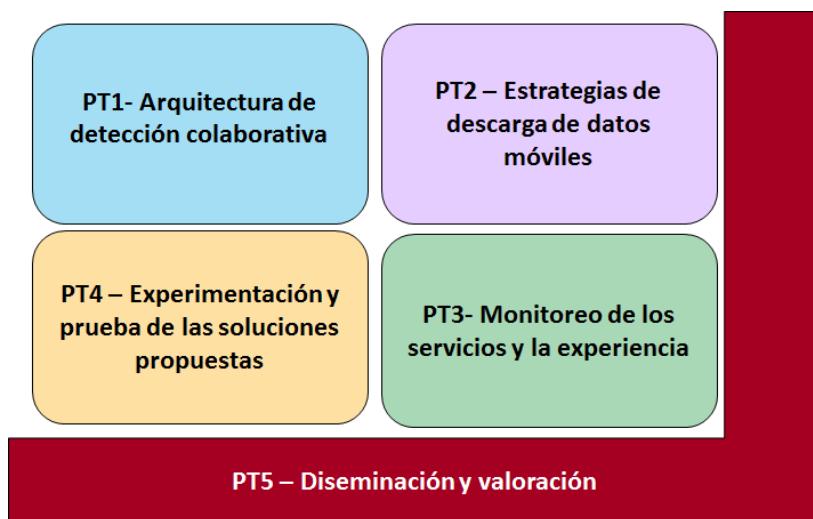


Figura. 1.1 Diagrama del proyecto MOSAIC

vas conduce a puntuales pérdidas de la calidad de la experiencia y finalmente de los clientes.

A partir de estas problemáticas en el área de redes móviles se está llevando a cabo un proyecto de investigación colaborativo denominado *MOSAIC (Mobile Crowd Sensing and data Offloading in Collaborative Networks)*, cuyo objetivo principal es hacer frente a la desafiante cuestión de la calidad del servicio y calidad de la experiencia de los usuarios finales (operadores, proveedores y clientes) cuando existe una gran cantidad de datos intercambiados a través de las redes móviles [MOS13]. Un nuevo e innovador aspecto que pretende este proyecto es el de definir técnicas inteligentes de monitoreo y arquitecturas para los problemas de descarga existentes dentro de la corriente de descarga de datos móviles. MOSAIC planea evaluar los enfoques propuestos sobre un emulador de red y sobre una plataforma francesa real. En la figura 1.1 se muestra como se encuentran distribuidos los cinco principales paquetes de trabajo del proyecto MOSAIC.

En base a los paquetes de trabajo establecidos por el proyecto MOSAIC, este proyecto de final de grado abordará parte del paquete de trabajo número tres acerca del *Monitoreo de los servicios y la experiencia*, el cual será detallado en el siguiente capítulo. A continuación describiremos brevemente el objetivo de cada uno de los paquetes restantes:

- La meta del primer paquete es el diseño de una arquitectura de percepción colaborativa que utilice las capacidades de percepción de los dispositivos móviles por medio de los sensores instalados.

- El segundo paquete del proyecto pretende definir nuevas estrategias de descarga de datos en redes móviles que utilicen redes complementarias para distribuir los datos originalmente destinados a la red celular, lo que permitirá reducir el congestionamiento en la red celular y como consecuencia mejorar la calidad de servicio ofrecida a los usuarios.
- El tercer paquete pretende definir métodos para el monitoreo de servicios y la calidad de la experiencia.
- El cuarto paquete se encargará de evaluar todos los enfoques propuestos en los paquetes anteriores sobre un emulador de red y una plataforma francesa real, de esta manera se ilustrará el rendimiento, la escalabilidad y la eficiencia de las soluciones propuestas.
- El quinto y último paquete pretende garantizar una amplia difusión de las contribuciones científicas del proyecto MOSAIC.

1.1. Motivación

En Paraguay, el servicio de telefonía móvil se ha convertido en una de las áreas de comercio con mayor penetración en la población en general. Por tal motivo, se ha incrementado el interés por conocer la opinión de las personas acerca del desempeño y la calidad del servicio que brindan las compañías prestadoras de telefonía móvil. En base a esta afirmación y a los objetivos planteados por el proyecto MOSAIC surge esta tesis de final de grado, la cual se centrará en la definición de técnicas de estimación de la calidad de la experiencia de los usuarios en el uso de servicios de datos móviles.

Conocer cómo los usuarios de los servicios móviles perciben la calidad de los mismos permitirá a las compañías prestadoras de servicios mejorar de manera eficaz y rentable los servicios brindados y de esta forma mantener la fidelidad de los clientes y la competitividad en el mercado.

1.2. Objetivos

En base a la problemática de la estimación de la calidad percibida por el usuario final y considerando el estado del arte analizado, este trabajo tendrá como objetivo general y aporte principal:

1.3 Organización del libro

El diseño de un modelo de estimación de la calidad de la experiencia para servicios de datos móviles, más específicamente en los contextos de uso de navegación web y video streaming, considerando parámetros a nivel de red, dispositivo y aplicación.

1.2.1. Objetivos específicos

1. Desarrollo de una plataforma de transmisión de datos móviles que emule una red de transmisión celular.
2. Desarrollo de una aplicación móvil (para la plataforma Android) que permita:
 - La recolección de parámetros a nivel de red, dispositivo y aplicación.
 - La realización de pruebas subjetivas para la evaluación del QoE para servicios de datos móviles en los contextos de navegación web y vídeo streaming.
3. Recolección de datos que servirán de base para el objetivo de esta propuesta utilizando la plataforma y la aplicación móvil desarrollada.
4. Comparación e identificación de los algoritmos de aprendizaje automático más adecuados para la estimación de la calidad de la experiencia en base a los parámetros obtenidos con la aplicación móvil.
5. Creación de un modelo de estimación de la calidad de la experiencia para servicios de datos móviles, más específicamente para navegación web y video streaming, considerando parámetros a nivel de red, dispositivo y aplicación.

1.3. Organización del libro

Este libro está estructurado de la siguiente manera: el capítulo 2 presenta los conceptos fundamentales para la comprensión del tema desarrollado. En los capítulos 3 y 4 se presenta un análisis del estado del arte acerca de los métodos utilizados en la evaluación de la calidad de la experiencia y las técnicas de aprendizaje automático aplicadas al problema de clasificación, respectivamente. El capítulo 5 describe el método propuesto y el proceso general a seguir, estableciendo además el alcance del trabajo. El capítulo 6 presenta los experimentos llevados a cabo, así como los resultados obtenidos y la forma en que se evaluaron y compararon. Finalmente, en el capítulo 7, se presentan las conclusiones en relación a los objetivos trazados como así también las propuestas de los posibles trabajos futuros relacionados a este trabajo.

Capítulo 2

Monitoreo de los servicios y la experiencia

Los servicios de datos móviles se han convertido rápidamente en un componente esencial de las estrategias de negocio de las empresas de telecomunicaciones. Actualmente, existen diversas aplicaciones móviles que van desde la navegación web hasta el streaming multimedia. El crecimiento en los requisitos de las diferentes aplicaciones móviles ha planteado un nuevo reto en cuanto a la gestión de los recursos de red. Esto es un factor esencial para otorgar mejores servicios a los usuarios finales sin sobredimensionar los recursos de red.

La *Calidad de la Experiencia* (*Quality of Experience* - QoE) es el término utilizado para describir las percepciones del usuario acerca del rendimiento de un servicio [SLC07]. La *calidad del servicio* (*Quality of Service* - QoS), por otra parte, es la habilidad de la red de proporcionar un servicio a un nivel garantizado [ITUa]. Con el fin de proporcionar un mejor QoE de una forma rentable, competitiva y eficiente, la red y los proveedores de servicios deben manejar el QoS de la red y la provisión de servicios de forma eficiente y efectiva [SLC07].

En resumen, el objetivo de los proveedores de servicios debe ser lograr la máxima calificación del usuario (QoE), mientras que la calidad de la red (QoS) es la base para llegar a ese objetivo de manera eficaz. El QoE, sin embargo, no se limita solo a las prestaciones técnicas de la red sino también a aspectos no técnicos como la facilidad de uso, el costo del servicio, entre otros, los cuales influyen significativamente en la percepción de los usuarios.

2.1. QoS y QoE

En esta sección se define y se explica las diferencias entre QoE y QoS. Esto nos ayudará a comprender los requerimientos del operador y del usuario final:

- *QoS* es definido formalmente por la ITU (*International Telecommunication Union*) como la habilidad de una red o porción de una red de proporcionar un servicio a un nivel garantizado [ITUa]. El QoS abarca todas las funciones, mecanismos y procedimientos en la red móvil y el terminal que garanticen la prestación de la calidad de servicio negociada entre el equipo de usuario (*user equipment - UE*) y la red central (*core network - CN*) [SLC07].
- *QoE* es cómo el usuario percibe la facilidad de uso de un servicio cuando lo está utilizando - qué tan satisfecho está con un servicio en términos de, por ejemplo, facilidad de uso, accesibilidad, rentabilidad e integridad del servicio [SLC07].

El término QoE se refiere a la percepción del usuario acerca de la calidad de un servicio en particular o de la red. Está expresado en sensaciones humanas tales como: “bueno”, “excelente”, “pobre”, etc. Por otra parte, el QoS es intrínsecamente un concepto técnico. Puede ser medido, expresado y comprendido en términos de red y elementos de red, que por lo general tienen poco significado para el usuario final. Los conceptos de QoE y QoS se ilustran en la figura 2.1 [SLC07].

Aunque un mejor QoS de la red puede, en muchos casos, resultar en un mejor QoE, el cumplimiento de todos los parámetros de QoS del tráfico no garantizará que el usuario se encuentre satisfecho. Por ejemplo: un excelente rendimiento (*throughput*) en una parte de la red podría no ayudar si no existe una buena cobertura en los alrededores del usuario final.

En lo que se refiere a las medición de parámetros de red (porcentaje de paquetes perdidos, ancho de banda efectivo, entre otros), estas estadísticas dicen al operador muy poco sobre el nivel de satisfacción del cliente. Por ejemplo, una transmisión perfecta de paquetes ilegibles no hace a los usuarios felices. Por tanto, la inferencia de que el QoE mejora debido a los mecanismos de QoS que se utilizan para mejorar los parámetros de red puede no ser precisa en todas las circunstancias.

2.2 Factores influyentes en el QoE

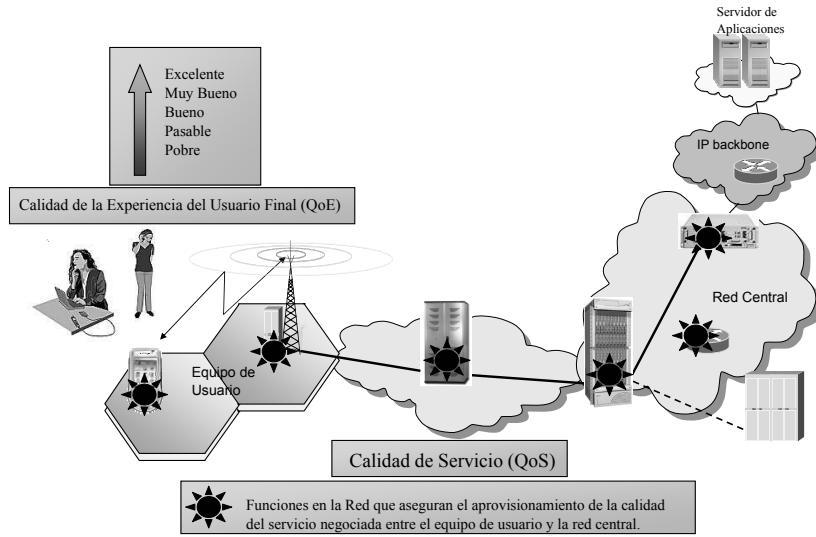


Figura. 2.1 El QoE es expresado en sensaciones en lugar de métricas. QoS implica todos los mecanismos, funciones y procedimientos en la red y el terminal que implementan los atributos de calidad negociables entre el UE y el CN. [SLC07]

2.2. Factores influyentes en el QoE

En [SCJ11], se presentan los diversos factores que influyen en el QoE. En esta sección hablaremos brevemente sobre ellos.

1. Calidad del servicio (QoS): como se mencionó anteriormente, el QoS se refiere a la capacidad de una red o de una porción de red de proporcionar servicio a un nivel garantizado.
2. Grado del Servicio (*Grade of Service - GoS*) ¹: describe todos los fenómenos que ocurren durante la configuración de la conexión, la liberación y el mantenimiento. Ejemplo de parámetros GoS son: retraso en el establecimiento de la conexión, probabilidad de bloqueo, retraso en la autenticación, probabilidad de corte de una conexión activa.
3. Calidad de la Resiliencia (*Quality of Resilience - QoR*): es la capacidad de “sobrevivencia” de una red ante fallos. Se refiere principalmente al aspecto de la disponibilidad, esto es, la probabilidad que un servicio se encuentre operativo.

Además de los factores mencionados anteriormente, la evaluación del QoE también se encuentra afectada por factores como el entorno, factores psicológicos, factores

¹GoS: este término es utilizado usualmente en el contexto de redes de telefonía

2.3 Monitoreo de QoE y QoS

sociológicos, que incluyen la expectativas del usuario y la experiencia con servicios similares, opiniones de otros usuarios, políticas de precios, el contexto (localización, actividad, etc.), el perfil del usuario, entre otros.

En la figura 2.2, se muestran los diversos factores que influyen en la calidad de la experiencia.

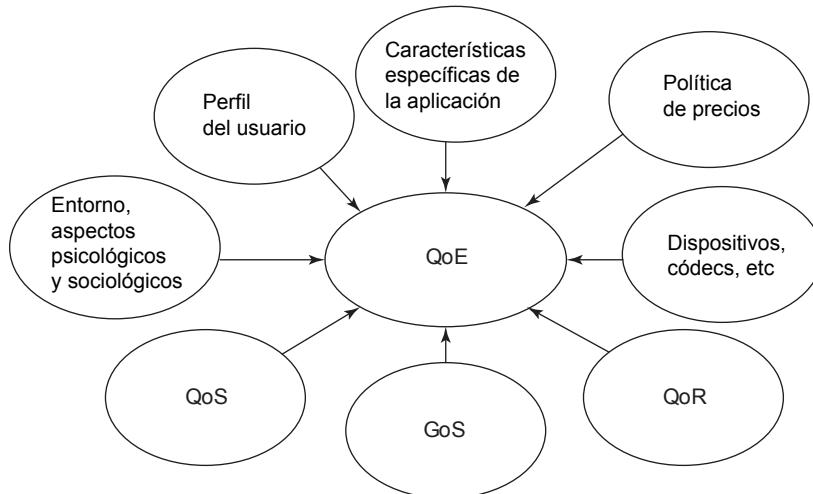


Figura. 2.2 Factores que influyen en la calidad de la experiencia

2.3. Monitoreo de QoE y QoS

Con el crecimiento de los servicios de datos móviles, se ha vuelto muy importante para el operador medir el QoE y el QoS de la red de forma precisa y mejorarlos de una manera eficaz y rentable para lograr la fidelidad del cliente y mantener la competitividad. Una mala calidad de la experiencia se traducirá en clientes insatisfechos, lo que lleva a una mala percepción en el mercado y a la pérdida de los clientes [SLC07].

El monitoreo de QoE y QoS en redes móviles consiste en la recolección/procesamiento de las estadísticas de rendimiento de servicio y de la red, uso de datos y fallas relacionadas a la calidad de servicio.

El proceso de monitoreo de QoE y QoS consiste de las siguientes funciones [SLC07]:

- Recuperar los datos de rendimiento de QoE y QoS de los elementos de red, incluidos los terminales móviles.
- Recolectar y procesar el uso de datos.
- Generar reportes de QoE y QoS.

2.3 Monitoreo de QoE y QoS

- Llevar a cabo un análisis de tendencias de los parámetros clave de QoE y QoS.
- Auditarse/Analizar los parámetros de QoE y QoS recolectados contra los valores esperados.

2.3.1. Monitoreo de QoS

Para monitorear la calidad de servicio (QoS) recibida en los dispositivos móviles se utiliza una herramienta denominada *agente móvil QoS*. Un agente móvil es una pieza de software que se ejecuta en un terminal móvil estándar con el propósito de medir la calidad de servicio [Sol06]. Se lleva a cabo el monitoreo pasivo y/o activo de acuerdo con un perfil de monitoreo configurado en el teléfono. Debido a su proximidad a los usuarios finales, los agentes móviles son capaces de capturar con precisión las experiencias de servicio de los mismos. Los agentes móviles pueden instalarse en un grupo seleccionado de dispositivos móviles de los usuarios finales. Esto convierte a miles de teléfonos estaciones de sondeo de calidad de servicio [Sol06].

Las responsabilidades de un agente móvil QoS incluyen:

- La medición de la calidad de servicio multimedia móvil.
- La producción y comunicación de las estadísticas de rendimiento a los servidores centrales de gestión.

Un agente móvil puede realizar mediciones tanto activas como pasivas:

- *Mediciones activas*: el agente inicia activamente los servicios y registra el desempeño de cada instancia de servicio.
- *Mediciones pasivas*: el agente observa los servicios iniciados por un usuario final y registra el rendimiento del servicio.

Un agente presenta periódicamente informes a un servidor central. Concede tiempo, ubicación, posición, información de cada servicio supervisado y envía alarmas al servidor cuando se superan umbrales. Un servidor de administración central deriva los indicadores claves de servicio de los informes proporcionados por los agentes QoS y también gestiona a los agentes QoS, es decir, los distribuye, los instala, los activa y los desactiva. Esto permite que se calculen indicadores claves de rendimiento para diferentes zonas, lo que posibilita la creación de un mapa con diferentes colores que indiquen el rendimiento en cada área [Sol06].

Los datos recolectados por los agentes móviles pueden ser utilizados para:

2.3 Monitoreo de QoE y QoS

- Comparar el rendimiento con otros operadores mediante el equipamiento con agentes móviles QoS de los teléfonos habilitados con tarjetas SIM de los competidores.
- La recolección de la calidad de redes extranjeras para realizar una comparación basada en hechos.
- Mejorar las ventas asegurando lo que el operador puede entregar.
- Entender mejor a los clientes mediante la recolección de datos de diferentes categorías de clientes.
- El lanzamiento de servicios exitosos.
- Cuidar segmentos particulares de clientes, tales como clientes corporativos y mantener el nivel de servicio acordado.

Elementos claves de un agente móvil de QoS.

En la figura 2.3 se observan los elementos claves de un agente móvil de QoS. Dichos componentes son descriptos a continuación [Sol06]:

El *motor del agente* es el elemento central que coordina las actividades, analiza los comandos recibidos y toma las acciones correspondientes. Para mediciones activas, el motor del agente activa sub-agentes de acuerdo con el perfil de monitoreo y se asegura que no existan dos sub-agentes que estén llevando a cabo la vigilancia al mismo tiempo. El sub-agente notifica al motor del agente después que una instancia de medición se ha completado. Luego el motor del agente espera por lo menos un intervalo de tiempo predefinido antes de activar otro agente. Esto permite que cualquier tarea asociada con el primer agente se ejecute por debajo.

Un *sub-agente* se encarga de los detalles de monitoreo de un servicio específico, almacenamiento, recuperación y eliminación de las mediciones del repositorio y la agregación de las mediciones para derivar indicadores de rendimiento de un servicio específico.

La *tabla de despacho* se actualiza cada vez que se añade un nuevo sub-agente para el control de un nuevo servicio. El uso de una tabla de despacho y sub-agentes hace que la arquitectura de agentes sea escalable. Para el seguimiento de un nuevo servicio solo tenemos que añadir un nuevo sub-agente y una nueva fila a la tabla de despacho, dejando el resto del código sin cambios.

2.3 Monitoreo de QoE y QoS

El *repositorio* almacena las mediciones y los datos persistentes como la identificación del agente, nombre del perfil activo, lista de servicios que se están supervisando, entre otros. Tienen acceso al mismo el motor del agente y los sub-agentes.

El *perfil* contiene todas las instrucciones sobre la forma de controlar cada tipo de servicio.

El agente utiliza el protocolo *SyncML DM*² como interfaz para comunicarse con servidores externos.

El motor del agente también instruye a la interfaz gráfica de usuario (*GUI- Graphical User Interface*) permitiendo visualizar el estado o progreso de las mediciones. Todos los comandos que se envían desde el servidor también se pueden introducir desde la GUI.

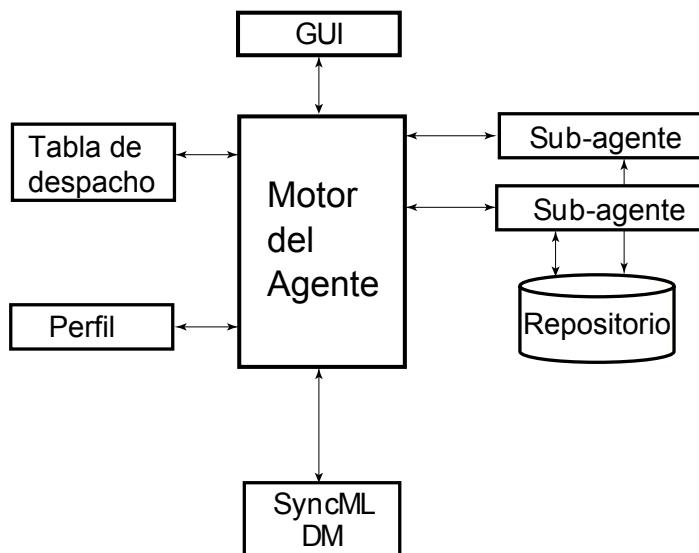


Figura. 2.3 Elementos funcionales de un agente móvil QoS de acuerdo a [Sol06]

2.3.2. Monitoreo de QoE

Es muy importante el diseño de una estrategia para medir y monitorear la calidad de la experiencia de la forma más realista posible, ya que esto le dará al operador un cierto conocimiento del rendimiento de la red a nivel general de satisfacción del cliente en términos de fiabilidad, disponibilidad, escalabilidad, velocidad, precisión y eficiencia. Con este objetivo han surgido varios modelos de monitoreo y evaluación del QoE que serán profundizados en el siguiente capítulo.

²SyncML (*Synchronization Markup Language*) es un protocolo de la familia XML, el cual proporciona sincronización remota para dispositivos móviles

2.4. Resumen

Este capítulo presentó los conceptos básicos acerca de la calidad del servicio (QoS) y la calidad de la experiencia (QoE). Si bien el QoE se encuentra influencia por el QoS, existen otros factores que influyen en la calidad percibida por los usuarios: el entorno, las políticas de precios, factores sociológicos, entre otros. Es muy importante para el operador medir el QoE y el QoS de la red de forma precisa y mejorarlos de una manera eficaz y rentable para lograr la fidelidad del cliente y mantener la competitividad. En particular, para el monitoreo del QoS en los dispositivos móviles se utilizan los agentes móviles, los cuales son capaces de obtener diversos tipos de datos acerca de la red directamente desde los dispositivos móviles.

Capítulo 3

Evaluación de la calidad de la experiencia

Como se explicó en el capítulo anterior, el concepto de QoE está relacionado con la experiencia subjetiva del usuario que utiliza el servicio. Desde el punto de vista de los operadores, el QoE juega un rol importante en el modelo de negocio, ya que es un factor clave de éxito para el posicionamiento de la empresa frente a sus competidores y la fidelidad de los usuarios. Este hecho ha llevado a los operadores de redes móviles y a la comunidad científica a desarrollar métodos y estrategias que permitan medir la calidad de la experiencia [SLC07].

Los conceptos de QoE y QoS son diferentes, sin embargo, muchos estudios se han centrado en identificar la relación entre los parámetros de QoS y la percepción del usuario. El objetivo de estos estudios consiste en establecer una relación o un modelo relacional que nos permita expresar la calidad de la experiencia (QoE) en términos de la calidad del servicio (QoS). Algunos trabajos han centrado su investigación en identificar la relación entre QoE y QoS para un tipo de tráfico determinado, mientras que otros han tratado de definir un modelo genérico que establezca una relación entre ambos conceptos [LGAPRD⁺12].

En este capítulo realizaremos una breve revisión de los trabajos existentes sobre los métodos de evaluación de QoE. Primeramente, clasificamos varias aplicaciones de acuerdo al tipo de tráfico. Luego realizamos una descripción acerca de los modelos de evaluación de QoE subjetivos, objetivos e híbridos, resaltando la metodología híbrida PSQA. Para finalizar, se realiza una revisión literaria de los métodos utilizados en la evaluación y medición de QoE para los tipos de tráfico principales.

3.1 Clasificación de aplicaciones según el tipo de tráfico.

3.1.1 Clasificación de aplicaciones según el tipo de tráfico.

La clasificación más general de las aplicaciones en cuanto al tipo de tráfico se hace con respecto a las siguientes características: tráfico elástico/inelástico o tráfico interactivo/no interactivo [SCJ11]. Esto da lugar a una distinción de cuatro tipos de tráficos:

- Elástico interactivo: incluye a aquellas aplicaciones que esperan por los datos y no los descartan por más que tarden en llegar. Por el hecho de ser un tráfico interactivo requiere plazos razonables debido a la interacción del usuario que está involucrado en el proceso.
- Elástico no interactivo: incluye a aquellas aplicaciones que esperan por los datos y no requieren de plazos razonables para la entrega de los mismos debido a que no implican la interacción directa del usuario. En tales casos, el retraso no está relacionado con el rendimiento promedio de la aplicación.
- Inelástico interactivo: es aquel tráfico que está vinculado a un retardo. Si los paquetes llegan después de un retraso máximo aceptable, los paquetes no tienen valor para el receptor y son descartados por la aplicación. Para este tipo de tráfico la interacción con el usuario es directa.
- Inelástico no interactivo: este tipo de tráfico también se encuentra vinculado a un retraso máximo pero a diferencia del anterior el usuario no está involucrado directamente en el proceso.

En la tabla 3.1 se muestran la clasificación de algunos ejemplos de aplicaciones en los diferentes tipos de tráficos.

3.2 Clasificación de los métodos de evaluación de QoE

Elástico		Inelástico	
No interactivo	Interactivo	No Interactivo	Interactivo
<ul style="list-style-type: none">• Descarga de archivos.• Copia de seguridad remota.• Intercambio de archivos P2P.• E-mail.• Almacenamiento de datos médicos.• Distribución digital.	<ul style="list-style-type: none">• Navegación web.• Grid computing.• Control de red de mensajería instantánea.	<ul style="list-style-type: none">• Emisión de radio.• Video bajo demanda.• TV en vivo.	<ul style="list-style-type: none">• VoIP.• Videoconferencia.• Video-Chat.• Juegos en línea.• Cirugía a distancia.• Comunicaciones de emergencia.

Cuadro. 3.1 Clasificación de ejemplos de aplicaciones según el tipo de tráfico [SCJ11].

3.2. Clasificación de los métodos de evaluación de QoE

Los métodos de evaluación de QoE pueden ser clasificados en [SCJ11]:

- Pruebas subjetivas.
- Métodos objetivos.

3.2.1. Pruebas subjetivas

Las *pruebas subjetivas* se llevan a cabo con la participación de usuarios del servicio. La idea es que las personas, que utilizan el servicio en un entorno real, evalúen el servicio completando encuestas. Las preguntas y respuestas de los cuestionarios reflejan diversas características del servicio. La principal ventaja de las pruebas subjetivas es que son el único medio para evaluar los impactos psicológicos y sociológicos del usuario sobre el QoE. Además es considerado el método más fiable para la evaluación de la calidad de la experiencia. Sin embargo, es costoso y no es fácil lograr un alto nivel de credibilidad. La selección apropiada de los usuarios, el tamaño de la muestra, el contenido, el entorno, entre otros, tienen una importancia crítica para la credibilidad

3.2 Clasificación de los métodos de evaluación de QoE

del método. Afortunadamente, algunas pautas para las pruebas subjetivas están disponibles en la ITU (Unión Internacional de Telecomunicaciones) en donde se establecen metodologías para la evaluación subjetiva de: la calidad de las imágenes por televisión, la calidad de vídeo para aplicaciones multimedia, la calidad audiovisual para aplicaciones multimedia, la calidad de transmisión, la calidad de la conversación, la calidad de los servicios telefónicos basados en sistemas conversacionales, entre otros.

En general, existen dos clases de evaluaciones subjetivas. En primer lugar, hay evaluaciones que determinan la calidad de funcionamiento de sistemas bajo condiciones óptimas, las que típicamente se denominan *evaluaciones de calidad*. En segundo lugar, hay evaluaciones que determinan la capacidad de los sistemas de mantener la calidad en condiciones no óptimas, estas se denominan *evaluaciones de degradación o deterioro* [SCJ11].

MOS (Mean Opinion Score-MOS) es la medida subjetiva más popular de QoE en la que los sujetos asignan su opinión sobre la calidad de la experiencia percibida en base a una escala predefinida de valores [SCJ11]. En las recomendaciones ITU P.800/P.910 definen una escala absoluta MOS de cinco puntos [ITU96, ITU08]. La mayoría de las métricas utilizan escalas absolutas y una descripción textual de la calidad que se asigna a cada puntaje. Las descripciones pueden ser orientadas a la calidad u orientadas al deterioro. Un ejemplo de una escala absoluta MOS se muestra en el cuadro 3.2.

Escala Absoluta MOS		
Valor	Calidad	Deterioro
5	Excelente	Imperceptible
4	Buena	Perceptible
3	Razonable	Poco Molesto
2	Pobre	Molesto
1	Mala	Muy Molesto

Cuadro. 3.2 Escala Absoluta MOS [ITU08]

3.2.2. Métodos objetivos

Los *métodos objetivos* proporcionan una evaluación QoE basada en la medición de varios parámetros relacionados con la prestación de servicios particulares o indicadores de calidad de servicio en la señal de salida del canal de transmisión, es decir, la señal que sería percibida por un usuario real. Para cada servicio, se realiza un análisis de varias características de la señal. Por ejemplo, en el caso de la transmisión de vídeo, la

3.2 Clasificación de los métodos de evaluación de QoE

calidad puede ser analizada con respecto a la presencia de varias degradaciones de la imagen de vídeo (desenfoque, sacudidas, pixelado, contraste y brillo, congelamiento, entre otros) [SCJ11].

Tres métodos de evaluación objetivos de QoE pueden ser distinguidos: *Full Reference (FR)*, *No-Reference (NR)* y *Reduced Reference (RR)*. En FR, la señal de la fuente (referencia) y la señal distorsionada (recibida) son comparadas y de esta manera se estiman las métricas QoE. Algunos ejemplos son: PEVQ (*Perceptual Evaluation of Video Quality*) [ITUc] y SSIM (*Structural Similarity Index*) para video, y PESQ (*Perceptual Evaluation of Speech Quality*) para voz. En muchos de los sistemas, la señal de la fuente no está disponible, entonces se utilizan los métodos NR donde se analiza únicamente la señal de salida. Ejemplos de métricas NR para video se dan en [CO03], mientras que para la calidad de voz ITU-T Rec. P.563 es aplicada. Por su parte, los métodos RR utilizan información parcial de la señal de referencia, disponible a través de un canal de datos auxiliares [SCJ11].

Una subcategoría de los métodos de evaluación objetivos son los *modelos de planificación de red*, ya que dan una estimación de la calidad de la experiencia sin implicar a los usuarios. Una característica distintiva es que no requieren el examen de la señal real, pero estiman el QoE esperado utilizando una función que mapea varios parámetros medibles de calidad de servicio (QoS) a una métrica QoE. Un ejemplo de una métrica que pertenece a esta categoría es E-model para la evaluación de la calidad de voz (ITU-T Rec. G.107 [ITUb]), en donde se tiene en cuenta varios factores que influyen en la calidad de la voz percibida [SCJ11].

La desventaja de los métodos objetivos es que pueden ser poco informativos y dar resultados inexactos. Es comúnmente sabido que las métricas como PSNR (*Peak Signal-To-Noise Ratio*) y MSE (*Mean Squared Error*) pueden dar resultados que son totalmente incompatibles con la evaluación QoE humana. Por lo tanto, los métodos objetivos se desarrollan cuidadosamente y son verificados por pruebas subjetivas [SCJ11].

El *Institut National de Recherche en Informatique et en Automatique* de Francia propuso un método híbrido entre las evaluaciones subjetivas y objetivas de la calidad de la experiencia llamado *Pseudo-Subjective Quality Assessment (PSQA)* [PSQ07].

PSQA toma lo mejor de los modelos subjetivos y objetivos, es un método no invasivo que obtiene datos en tiempo real y los resultados son en términos de Mean Opinion Score (MOS) [LGAPRD⁺12]. Este método de evaluación de QoE será descripto a continuación.

3.3. Pseudo-Subjective Quality Assessment - PS-QA

En esta sección daremos una pequeña descripción acerca de la metodología PSQA (*Pseudo-Subjective Quality Assessment*). Esta metodología, básicamente, nos permite ser capaces de medir la calidad *como es percibida por el usuario*, de manera automática y precisa. Esto significa que PSQA nos permite obtener un valor de la calidad experimentada cercano al valor promedio que nos daría un observador humano [Rub05].

Para describir como funciona PSQA, considere el siguiente ejemplo [Rub05]. Se desea analizar una aplicación de video streaming, la cual envía un flujo desde el origen al destino. Se procede como sigue:

1. En primer lugar se identifican un conjunto de parámetros, los cuales tendrán un impacto importante en la calidad percibida. En el caso de aplicación de video streaming, éstos parámetros se dividen en dos clases: los relacionados al codec (bit rate, frame rate) y los asociados a la red de transporte de datos (retraso, jitter, tasa de pérdidas).
2. Para cada parámetro seleccionado, se escoge un rango, que corresponde al sistema a ser analizado. Por ejemplo, en el caso del *loss rate* se considera un rango 0 %, 1 %, 2 %, 3 %, 5 %, 10 %. A cada combinación de valores para estos parámetros se los denomina una *configuración* del sistema.
3. Se seleccionan M configuraciones entre el conjunto de todas las configuraciones posibles. Este proceso de selección se realiza por medio de la combinación de dos procedimientos: *a)* diferentes configuraciones son construidas escogiendo aleatoriamente los valores de los parámetros en el espacio de valores dado y *b)* varias configuraciones son escogidas cubriendo de alguna manera los extremos de los rangos de los parámetros.

Se separa aleatoriamente el conjunto de las M configuraciones seleccionadas en dos subconjuntos $C = \{\gamma_1, \dots, \gamma_K\}$ y $C' = \{\gamma'_1, \dots, \gamma'_{K'}\}$. Así se tiene $K + K' = M$. El conjunto C será utilizado para “aprender” y el conjunto C' será utilizado para “validar”.

4. Se construye una plataforma que permita: *a)* Enviar una secuencia de video a través de una conexión IP y *b)* Controlar el conjunto de parámetros escogidos en el primer paso.

3.3 Pseudo-Subjective Quality Assessment - PSQA

5. Se escoge una secuencia representativa σ de la población de secuencias a ser considerada. Se envía σ exactamente M veces desde el origen al destino, utilizando la plataforma, y en cada vez se utilizan los valores de las configuraciones seleccionadas (conjuntos C y C'). Se obtienen dos conjuntos de *copias distorsionadas* de σ : la copia correspondiente a la configuración $y_i, 1 \leq i \leq K$ ($y'_j, 1 \leq j \leq K'$) es denotada por σ_i (σ'_j).
6. Se realiza un experimento de pruebas subjetivas utilizando las M secuencias $\sigma_1, \dots, \sigma_K, \sigma'_1, \dots, \sigma'_{K'}$. Esto consiste en seleccionar un panel de observadores humanos y pedirles que evalúen numéricamente la calidad (*como ellos la perciben*) de las M secuencias. Así, cada secuencia recibirá un valor de MOS (*Mean Score Opinion*). Se denota al valor de MOS de la secuencia σ_i (o σ'_j) por Q_i (o por Q'_j). En más detalle, se asume que existen R observadores, y un observador r , al final del experimento, da un valor q_{ri} a la secuencia σ_i (y q'_{rj} a la secuencia σ'_j). Luego, se realiza un test estadístico para detectar a los “observadores malos” (aquellos que no están de acuerdo con la mayoría). Asuma que (luego de reordenar los índices de los observadores) los observadores $R' + 1, \dots, R$ son los “observadores malos”. Entonces, sus puntajes son removidos del conjunto y se obtiene el promedio de los valores MOS dado por los “observadores buenos”, esto es:

$$Q_i = \sum_{r=1}^{R'} \frac{q_{ri}}{R'} \quad (y \quad Q'_j = \sum_{r=1}^{R'} \frac{q'_{rj}}{R'})$$

7. Se identifica la configuración γ_i con el valor de calidad Q_i (y γ'_j con valor Q'_j), y buscamos una función real $v()$ de las n variables asociado con los n parámetros seleccionados, tal que para cualquier conjunto de valores en el conjunto C , la función $v()$ retorne un número cercano al valor de MOS asociado. Esto es, para cualquier $\gamma_i \in C$, $v(\gamma_i) \approx Q_i$. Esta es la fase de *aprendizaje*. Esta fase puede ser realizada por diferentes herramientas, tales como Redes Neuronales Artificiales (*Artificial Neural Networks*), Redes Bayesianas (*Bayesian Networks*), Redes Neuronales Aleatorias (*Random Neural Networks*), entre otras.
8. Luego de haber encontrado la función $v()$, se procede a la fase de *validación*, que consiste en probar el valor de la función $v()$ sobre el conjunto de configuraciones en C' . Si para $\gamma'_i \in C'$, su valor $v(\gamma'_i) \approx Q'_i$, entonces la función $v()$ es válida y el proceso finaliza.
9. El uso del evaluador ahora consiste en la medición de los parámetros considera-

3.4 Revisión literaria acerca de la evaluación del QoE

dos, por ejemplo en el lado receptor de la comunicación, y llamar a la función $v()$ con los valores medidos como entrada. Esto puede ser realizado en tiempo real ya que las entradas pueden ser recolectadas en tiempo real y la evaluación de la función $v()$ no es computacionalmente costosa.

3.4. Revisión literaria acerca de la evaluación del QoE

Un problema importante en el campo de la evaluación de la calidad de la experiencia es la cuantificación automática de la calidad experimentada por el usuario final. En particular, la capacidad de realizar esta tarea en tiempo real. Este problema es interesante ya que existen diversos parámetros, en diversos niveles (red, aplicación, usuario) que afectan a la calidad y la identificación del efecto combinado de los mismos requiere de un análisis con cierto grado de complejidad.

Por medio de las evaluaciones subjetivas se obtienen buenos resultados, pero carecen de la capacidad de realizarse automáticamente. Por tanto, en esta sección haremos una revisión de la literatura existente con respecto a la evaluación de la calidad de la experiencia en servicios de distribución de diversos tipos de tráfico: video, voz y datos.

En el trabajo [PVBK09], los autores se enfocan en la medición de la calidad de la experiencia para video streaming sobre redes inalámbricas. Los autores hacen uso de enfoques de medición subjetivos, objetivos y un enfoque híbrido, y miden el rendimiento de estos enfoques en un entorno inalámbrico. Específicamente, se centran en un enfoque híbrido mencionado anteriormente, PSQA y demuestran que éste enfoque genera una buena estimación en comparación a las métricas objetivas, tales como: PSNR (*Peak Signal to Noise Ratio*). También demuestran que por medio de dicho enfoque se obtienen resultados similares en comparación a los test subjetivos realizados por observadores humanos.

En [MCC11], los autores investigan cómo afectan los tres niveles de la calidad del servicio (QoS), sobre streaming de video en HTTP, a la calidad percibida. Los niveles considerados fueron: QoS a nivel de red, QoS a nivel de aplicación, QoS a nivel de usuario (QoE). Abordaron el problema dividiéndolo en dos subproblemas: medición de la correlación entre el QoS a nivel de red y el QoS a nivel de aplicación y la medición de la correlación entre el QoS y el QoE. A partir del análisis, se revela que la frecuencia

3.4 Revisión literaria acerca de la evaluación del QoE

de rebuffering¹ es el factor principal en las variaciones de QoE para el servicio de streaming de video.

En [MR02], se propone una metodología capaz de realizar una cuantificación automática de la calidad de un flujo de video transportado por medio de una red de paquetes. Esta metología posee las siguientes propiedades: *a)* evaluación en tiempo real y *b)* el valor asignado a la calidad del flujo está cercanamente relacionado al valor que podría ser obtenido por medio de test subjetivos. Para ello, los autores hacen uso de una red neuronal (*Random Neural Network - RNN*), la cual recibe constantemente los parámetros seleccionados y realiza la evaluación. Entre los parámetros considerados, se tienen: *bit rate*, *frame rate*, *tasa de perdidas de la red*, *burst loss size*, entre otros. Además de ello, analizan el impacto de los parámetros citados anteriormente en la calidad del video con el fin de ayudar a comprender el comportamiento del flujo de video en tiempo real transmitido sobre redes IP.

En [TSS14], los autores presentan un modelo objetivo en donde una aplicación móvil es desarrollada para obtener las métricas de QoS y evaluar el QoE en un entorno móvil. La aplicación móvil, desarrollada para la plataforma Android, actúa como un agente móvil que realiza las mediciones de los métricas específicas de un servicio.

Para clasificar la calidad de la experiencia del usuario, los autores han considerado una escala de MOS de cinco puntos (1 = malo, 2 = pobre, 3 = aceptable, 4 = bueno, 5 = excelente). Así, por medio de un modelo de cálculo del MOS y los parámetros recabados por el agente móvil, se obtiene un valor de MOS, el cual representa una estimación de la calidad de la experiencia de un usuario en un entorno móvil.

Los parámetros son clasificados como incrementales o decrementales, dependiendo si el incremento de los mismos aumenta o disminuye el valor de MOS para un servicio dado. Tomando en cuenta dicha clasificación, se deben identificar los parámetros que afectan la calidad de la experiencia para un servicio en particular, clasificarlos y definir tres valores para cada uno de los parámetros: un valor mínimo (para el cual el MOS es de 1), un valor máximo (para el cual el MOS es 5) y un valor x_0 (para el cual el MOS tiene un valor de 4).

En [RTV06], los autores describen una metodología de evaluación de la experiencia del usuario basada en PSQA utilizando una RNN (*Random Neural Network*). Demuestran el uso de PSQA en el análisis de la calidad de voz percibida, más específicamente en el campo de las aplicaciones VoIP. Además realizan una comparación del uso de RNN con respecto a otros métodos de aprendizaje automático (Redes neuronales ar-

¹Interrupción de la reproducción de video debido a las condiciones de las red.

3.4 Revisión literaria acerca de la evaluación del QoE

tificiales y Clasificadores bayesianos). Entre los parámetros que afectan a la calidad de voz, se consideraron: el códec utilizado, la tasa de perdida (*loss rate*), el tipo de corrección de errores utilizado, tamaño promedio de perdidas en ráfaga (*mean loss burst size*), el intervalo de paquetización (esto es, la duración de voz contenida en cada paquete), el delay y el jitter.

En cuanto a la relación entre el QoE y el QoS, específicamente en la navegación web, en [KH02], investigan en detalle cómo la satisfacción del usuario es afectada por dos parámetros de red principales: el ancho de banda y la latencia. El objetivo del trabajo fue medir el nivel de insatisfacción del usuario con respecto a la calidad de distribución del contenido web. Los autores realizan un análisis del impacto que tienen el tiempo de respuesta y el ancho de banda efectivo sobre la calidad de la experiencia tomando como base la tasa de eventos de cancelación de carga de una página. Así, el número de eventos de cancelación de carga se utiliza como una forma de medir el nivel de insatisfacción de la calidad de distribución de contenido web. A partir de sus análisis concluyen que el ancho de banda juega un rol importante en la satisfacción del usuario con respecto a la navegación. Si bien, la latencia también cumple un papel importante, éste es menor al del ancho de banda, e incluso para valores pequeños (entre 50-500ms), su influencia sobre la calidad experimentada no es apreciable.

En el trabajo [RRAP⁺13], presentan un modelo de correlación QoS-QoE para servicios de Internet móvil utilizando el enfoque de redes neuronales. En la metodología propuesta, se hace uso de pruebas subjetivas para tres tipos de servicios de Internet móvil (navegación web, video streaming, descarga de archivos) y además utiliza agentes móviles como medio para obtener parámetros de red. A partir de los datos recabados, la red neuronal (*Perceptrón multicapa*) es entrenada con el fin de obtener una correlación entre los parámetros de QoS y QoE.

Otro trabajo con respecto a la relación entre el QoE y el QoS se da en [FHTG10], en el que los autores proponen una fórmula genérica en la que los parámetros de QoS y el QoE se relacionan por medio de una relación exponencial, llamada *hipótesis IQX*. Esta fórmula relaciona los cambios en el QoE con respecto al QoS y al nivel actual de QoE. Los autores argumentan que la sensibilidad del QoE es más pronunciada a medida que la calidad experimentada es mayor. Si el QoE es alto, una pequeña perturbación en el QoS, hará decrementar fuertemente el QoE. En cambio, si el QoE es bajo, una perturbación adicional del QoS, no degenera el QoE de manera significativa. Por tanto, los autores asumen que los cambios del QoE dependen del nivel de QoE actual. Los autores evalúan la hipótesis IQX tanto para servicios de voz como para navegación

3.4 Revisión literaria acerca de la evaluación del QoE

web.

Como resumen, en la tabla 3.3 se expone un cuadro comparativo acerca de los métodos utilizados para la evaluación y medición de QoE presentados en este apartado:

3.4 Revisión literaria acerca de la evaluación del QoE

Resumen						
Año	Ref	Modelo	Tipo de Tráfico	Servicio	Parámetros QoS	Otros Aspectos
2009	[PVBK09]	Modelo Híbrido	Inelástico No Interactivo	Video Streaming	Tasa de pérdida de paquetes (LS), duración media de una secuencia de paq. consecutivos perdidos en un periodo de tiempo (MLBS).	Comparación de modelos subjetivo, objetivo e híbrido. Utiliza PSQA.
2011	[MCC11]	Metodo Subjetivo	Inelástico No Interactivo	Video Streaming	Tiempo de buffering inicial, duración media de un evento rebuffering y la frecuencia de rebuffering.	
2002	[MR02]	Modelo Objetivo	Inelástico No Interactivo	Video	Bit rate, frame rate, tasa de pérdidas de la red, burst loss.	Utiliza redes neuronales.

3.4 Revisión literaria acerca de la evaluación del QoE

2014	[TSS14]	Modelo Objetivo	Varios tipos de tráfico.	Navegación Web, VoIP Video Streaming	Rendimiento, latencia, tiempo de carga de páginas web, resolución de video.	Clasificación de parámetros incrementales y decrementales Ecuación MOS.
2006	[RTV06]	Modelo Híbrido	Inelástico Interactivo	VoIP	Códec, la tasa de pérdida, el tipo de corrección de errores, mean loss burst size, el intervalo de paquetización , el delay y el jitter.	Comparación entre redes neuronales y clásificadores Bayesianos.
2002	[KH02]	Modelo Objetivo	Elástico Interactivo.	Navegación Web	Ancho de banda y latencia.	
2013	[RRAP ⁺ 13]	Modelo Híbrido	Varios tipos de tráfico.	Navegación Web, Video Streaming, Descarga de Archivos	ancho de banda, latencia y fuerza de la señal.	Utiliza red neuronal perceptrón multicapa.

3.4 Revisión literaria acerca de la evaluación del QoE

2010	[FHTG10]	Modelo Objetivo	Varios tipos de tráfico.	VoIP y Navegación Web	Para voz: jitter, pérdida y reordenamiento en la red. Para navegación: tiempo de respuesta, tasa de cancelación.	Relación exponencial entre QoE y QoS, hipótesis IQX.
------	----------	--------------------	--------------------------------	-----------------------------	--	---

Cuadro. 3.3 Resumen de la revisión literaria acerca de la evaluación de QoE.

3.5. Discusión

En vista al estado del arte acerca de los métodos de evaluación de QoE expuesto podemos concluir en las siguientes líneas futuras de trabajo:

- Los reportes presentados no consideran todos los factores que inciden en el QoE. Por lo tanto, es necesario, diseñar un sistema de monitoreo que, además de monitorear los parámetros de calidad de servicio a nivel de red, realice el monitoreo a nivel de aplicación y que también considere el monitoreo de los parámetros de estado del dispositivo móvil (memoria disponible, uso de CPU, entre otros).
- Recolección de datos de pruebas subjetivas en el uso de servicios de datos móviles. Esto es debido a la limitada o nula publicación de datos acerca de estudios en donde se evalúa el QoE y, en particular, en el uso de redes móviles.
- Comparar distintas técnicas de aprendizaje automático para el proceso de estimación del QoE y seleccionar la que arroja resultados más cercanos a lo que sería la percepción del usuario.
- Presentar nuevos modelos de correlación de los diversos parámetros influyentes en la calidad percibida por el usuario final y la calidad de la experiencia global del usuario de servicios móviles.

Capítulo 4

Aprendizaje automático

4.1. Introducción

Desde que las computadoras fueron inventadas, nos hemos preguntado si serían capaces de *aprender* y mejorarse así mismas en base a la experiencia. Por ejemplo: una computadora que aprende a diagnosticar enfermedades a partir de registros médicos, otra que aprenda a reconocer patrones, son solo algunas de las posibilidades. De manera sencilla, decimos que el principal objetivo del aprendizaje automático es el de construir un modelo computacional a partir de las *experiencias* pasadas que el computador haya observado. Aquí, la *experiencia* se refiere a la información pasada al aprendiz (*learner*), la cual toma la forma de datos electrónicos disponibles para su análisis [MRT12]. Así, el aprendizaje automático (*o machine learning*) se refiere a sistemas que son capaces de adquirir e integrar *conocimiento* de manera autónoma. En [Mit97], el autor lo define como un campo multidisciplinario, con bases en la Inteligencia Artificial, Probabilidad y Estadística, Teoría de la Información, entre otros, y abarca cuestiones acerca de cómo construir programas que mejoren su rendimiento en el desarrollo de una tarea a través de la experiencia.

Debido a su amplio interés, diversas técnicas de aprendizaje automático han sido exitosamente aplicadas a diferentes áreas: procesamiento de imágenes, procesamiento del lenguaje natural, finanzas, telecomunicaciones, entre otros [Mit97]. En particular, juega un rol fundamental dentro de la *Minería de Datos*, donde el aprendizaje automático ha sido extensamente utilizado para extracción, agrupación, clasificación y regresión de modelos.

Entre los problemas más abordados por el aprendizaje automático se encuentra la clasificación de ítems, el cual es aplicado en el proceso de aprendizaje de la metodología

PSQA previamente mencionada. Por ello, en las siguientes secciones se presentarán los conceptos básicos acerca del aprendizaje automático y algunos de los métodos de aprendizaje automático más utilizados en los problemas de clasificación.

4.1.1. Aplicaciones y problemas

Los algoritmos de aprendizaje han sido exitosamente utilizados en una gran variedad de aplicaciones:

- Clasificación de textos.
- Procesamiento del lenguaje natural.
- Reconocimiento de voz.
- Biología computacional.
- Visión computarizada.
- Diagnóstico médico.

Entre los diversos tipos de problemas abordados por el aprendizaje automático, se encuentran:

- *Clasificación*: asignar una categoría a cada ítem. Por ejemplo, la clasificación de QoE a partir de los parámetros de QoS.
- *Regresión*: predecir un valor real para cada ítem. Los ejemplos de regresión incluyen la predicción de variaciones de variables económicas.
- *Ranking*: ordenar los ítems de acuerdo a algún criterio. Por ejemplo, en las búsquedas web, retornando las páginas web más relevantes con respecto a una consulta.
- *Clustering*: particionar ítems en regiones homogéneas. El clustering es utilizado a menudo para analizar un gran conjunto de datos. Por ejemplo, en el contexto del análisis de las redes sociales, los algoritmos de clustering intentan identificar “comunidades” dentro de grandes grupos de personas.
- *Reducción de la dimensionalidad o manifold learning*: transformar una representación inicial de ítems en una representación de menor dimensión de dichos ítems mientras se preserva algunas de sus propiedades.

4.1.2. Definiciones

En este apartado se introducirá la terminología básica utilizada en el campo del aprendizaje automático [MRT12].

- *Ejemplos (Examples)*: son ítems o instancias de datos utilizados para el aprendizaje o la evaluación.
- *Características (Features)*: es el conjunto de atributos, representado a menudo como un vector, asociados a un ejemplo. En el caso de la calidad del servicio, algunas características pueden ser: ancho de banda, delay, jitter, entre otras.
- *Etiquetas (Labels)*: son los valores o las categorías asignadas a los ejemplos.
- *Conjunto de entrenamiento (Training set)*: son los ejemplos utilizados en el proceso de entrenamiento del algoritmo de aprendizaje.
- *Conjunto de prueba (Test set)*: son los ejemplos utilizados para evaluar el algoritmo de aprendizaje. El conjunto de prueba es separado del conjunto de entrenamiento y no se encuentra disponible durante la etapa de aprendizaje.
- *Función de error*: es una función que mide la diferencia entre la etiqueta predicha y la etiqueta real.
- *Conjunto de hipótesis (Hypothesis set)*: es un conjunto de funciones que mapean características (vectores de características) al conjunto de etiquetas.

4.1.3. Tipos de aprendizaje

Dependiendo de los datos o de la retroalimentación disponible para el sistema de aprendizaje, los tipos de aprendizaje se pueden clasificar en [DHS99]:

- *Aprendizaje supervisado (Supervised learning)*: en este tipo de aprendizaje se proveen las etiquetas de los ítems dentro del conjunto de entrenamiento, así la base de conocimientos del sistema está formada por los ejemplos etiquetados anteriormente.
- *Aprendizaje no supervisado (Unsupervised learning)*: en este tipo de aprendizaje no se encuentran disponibles las etiquetas de los ítems, por tanto, el proceso de aprendizaje se lleva a cabo sobre un conjunto de ejemplos tomados como entrada. Este tipo de aprendizaje es utilizado para el reconocimiento de patrones.

- *Aprendizaje por refuerzo (Reinforcement learning)*: en este tipo no se provee una etiqueta deseada, en lugar de ello, el proceso de aprendizaje se realiza en base a la retroalimentación obtenida del entorno exterior.

4.2. El problema de la clasificación

El rango de problemas que se pueden abordar con el aprendizaje automático es extensamente amplio. Entre ellos, el problema de la clasificación de ítems es uno de los más estudiados en esta rama [SV08]. En su forma más sencilla, el problema de la clasificación, ocurre cuando un objeto necesita ser asignado a un grupo o clase predefinido basándose en un número de atributos observados relacionados al objeto. En la figura 4.1 se observa de manera gráfica la solución a un problema de clasificación, en donde la línea curva denota el límite de separación entre las diferentes etiquetas que representan las diferentes clases existentes. Así, en la figura de la izquierda, la curva denota el límite de separación entre los triángulos azules y las estrellas rojas. De la misma manera, en la figura de la derecha, la curva separa los triángulos azules de estrellas rojas y los rombos verdes.

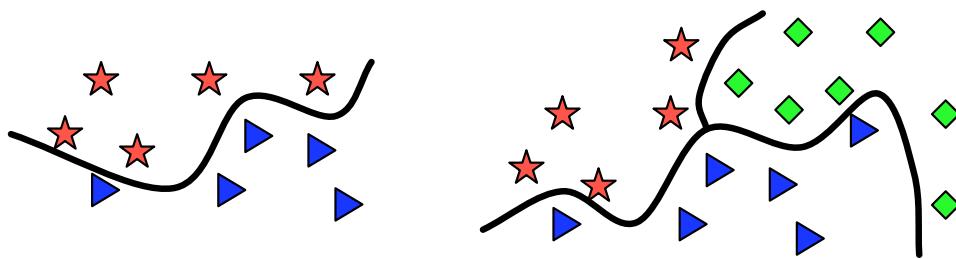


Figura. 4.1 Izquierda: Clasificación binaria. Derecha: Clasificación de tres clases.

4.3. Métodos de aprendizaje automático

En esta sección se dará una breve introducción a algunos de los métodos de aprendizaje más utilizados en el problema de la clasificación de ítems. Esta sección no pretende ser bibliografía detallada de cada método, por tanto implementaciones y detalles acerca de los métodos de aprendizaje citados quedan fuera del alcance de este trabajo, sin embargo en cada sección se citarán las bibliografías correspondientes en caso de ser necesaria una mayor profundización.

4.3.1. Redes neuronales artificiales

Este método de aprendizaje automático está inspirado en el funcionamiento del sistema biológico de aprendizaje de los seres humanos, construido a partir de la interconexión de neuronas. De manera análoga, las redes neuronales artificiales se construyen a partir de un conjunto densamente interconectado de unidades simples (neuronas), donde cada unidad toma cierta cantidad de números reales como entrada (que posiblemente representen las salidas de otras unidades) y produce un único valor de salida (el cual puede convertirse en la entrada de muchas otras unidades) [Mit97]. Así, cada unidad (neurona) puede ser vista como una unidad de procesamiento de información que es fundamental para el funcionamiento de la red en su totalidad. Dentro del modelo artificial de una neurona (Ver figura 4.2), pueden identificarse tres elementos básicos [Hay98]:

- Un conjunto de enlaces, cada uno de los cuales está caracterizado por un peso. Específicamente, una señal x_j en la entrada del enlace j conectado a la neurona k es multiplicado por el peso w_{kj} .
- Un *sumador (adder)* para sumarizar las señales de entrada ponderadas.
- Una función de activación para limitar la amplitud de la salida de la neurona. Con ella se limita la señal de salida de una neurona a un valor finito.

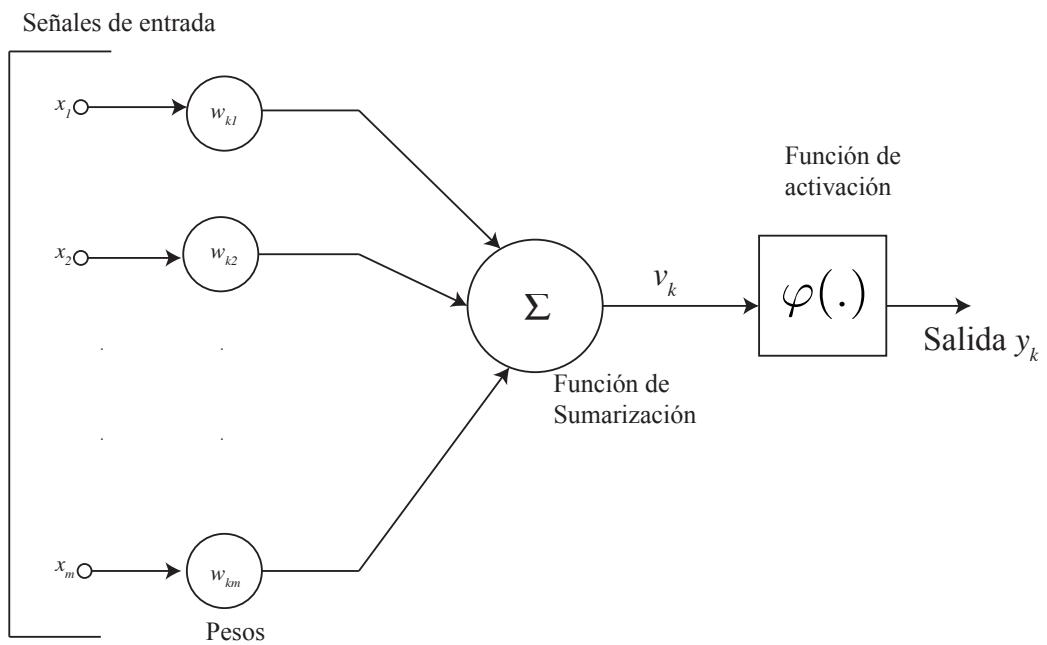


Figura. 4.2 Modelo de una neurona artificial.

Un ejemplo de un modelo de red neuronal genérica se puede observar en la figura 4.3.

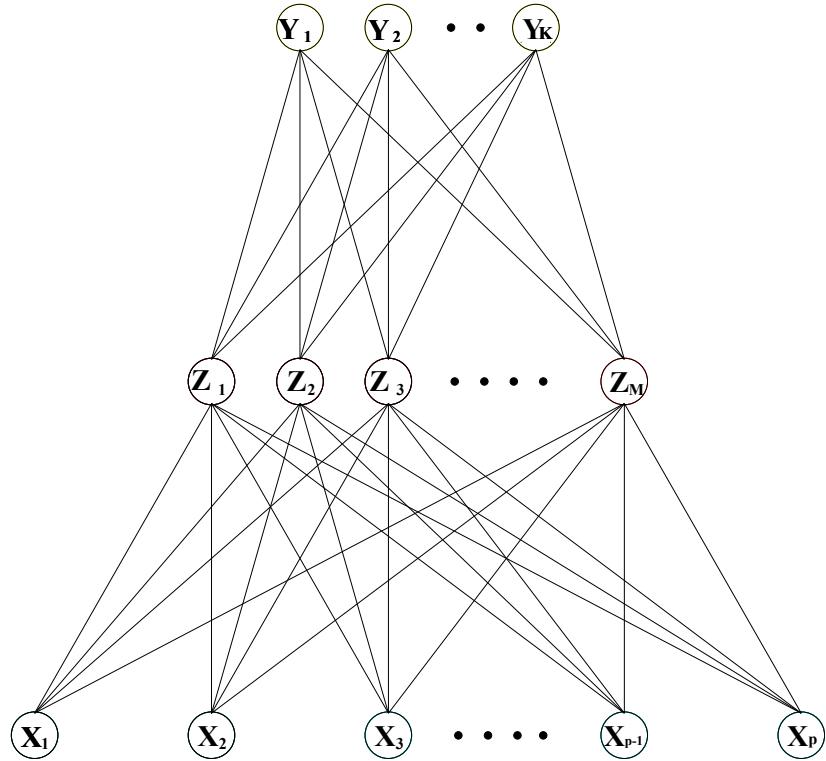


Figura. 4.3 Ejemplo de una red neuronal interconectada.

Mayor detalle acerca del proceso de aprendizaje de las redes neuronales se puede ver en [Hay98, Mit97].

4.3.2. Máquinas de soporte vectorial (Support Vector Machine - SVM)

Para describir las ideas básicas tras las máquinas de soporte vectorial [BC00], consideremos el problema de la clasificación binaria con datos x_i ($i = 1, \dots, m$) con sus correspondientes etiquetas $y_i = \pm 1$. Sea la función de clasificación $f(x) = \text{sign}(w \cdot x - b)$. El vector w determina la orientación de un plano discriminante. El escalar b determina el desplazamiento del plano a partir del origen. Asumimos que los dos conjuntos son linealmente separables, esto es, existe un plano que correctamente clasifica todos los puntos en dos conjuntos. Así, SVM busca encontrar el hiperplano óptimo que separa dos clases, mientras maximiza el margen entre los puntos de las clases más cercanos. Esto se observa en la figura 4.4.

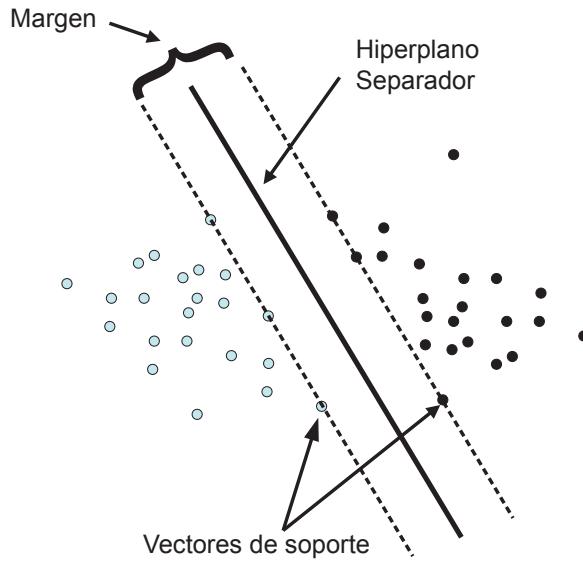


Figura. 4.4 Clasificación binaria por medio de SVM.

Mayor detalle acerca de los fundamentos de este método de aprendizaje se puede ver en [Hay98, SV08].

4.3.3. Árboles de decisión

Es natural e intuitivo clasificar un patrón a través de una secuencia de preguntas, en el que la respuesta a la siguiente pregunta dependa de la pregunta actual. Esta secuencia de preguntas es particularmente útil para datos que no son métricas, ya que todas las preguntas pueden ser respondidas de la forma “sí/no” o “falso/verdadero”. Así, tal secuencia de preguntas puede ser vista como un árbol de decisión o simplemente árbol, donde el *primer nodo* o *nodo raíz* está conectado por medio de *ramas* o *enlaces* a los demás nodos. Estos enlaces siguen hasta que alcanzan los *nodos hoja*, los cuales ya no se conectan a otros nodos, tal y como sucede con las estructuras en forma de árbol [DHS99].

Para el problema de la clasificación de ítems, el proceso comienza en el nodo raíz, en el cual se pregunta por un atributo en particular acerca del ítem. Los diferentes enlaces desde el nodo raíz corresponden a los diferentes valores posibles de dicho atributo. En base a la respuesta, se sigue el enlace adecuado al *nodo subsecuente* o *descendiente*. El siguiente paso es tomar la decisión en el nodo subsecuente apropiado, el cual es considerado como el nodo raíz del subárbol. Esto continúa así hasta alcanzar un nodo hoja, en el cual ya no existan preguntas por responder. tanto, cada nodo hoja conduce

a una etiqueta de una categoría en particular. Un ejemplo de una clasificación basada en árboles de decisión se puede observar en la figura 4.5.

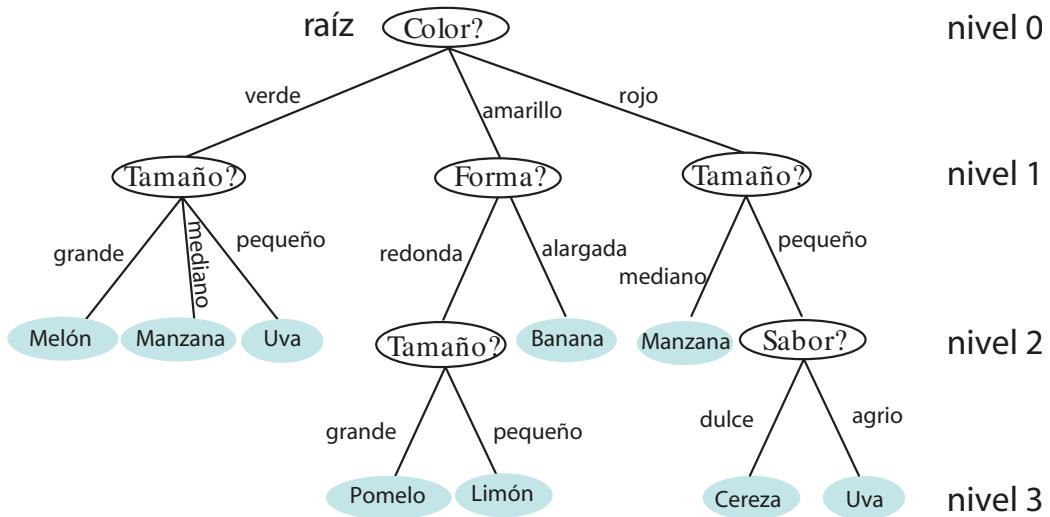


Figura. 4.5 Ejemplo de un árbol de decisión

Mayor detalle acerca de los fundamentos de este método de aprendizaje se puede ver en [WF05, SV08].

4.3.4. Tabla de decisiones

Dado un conjunto de datos (*dataset*) de instancias etiquetadas, los algoritmos de aprendizaje supervisado buscan una hipótesis que prediga de manera correcta la etiqueta (o clase) de futuras instancias no etiquetadas. En los anteriores apartados vimos algunas de las técnicas de aprendizaje automático tales como: árboles de decisión, redes neuronales. Una técnica alternativa a la citadas anteriormente son las *tablas de decisiones*. Una *tabla de decisión* es una forma de modelar un conjunto de reglas complejas y sus correspondientes acciones. En el caso de la clasificación, las tablas modelan un conjunto de reglas con sus correspondientes clases. De acuerdo a [K⁺95], un tipo de representación es la *Decision Table Majority - DTM*. Esta representación posee dos componentes: un *esquema*, que es el conjunto de características (*features*) que son incluidas en la tabla y un *cuerpo* que lo conforman instancias etiquetadas a partir del espacio definido por las características del esquema. Dada una instancia no etiquetada, un clasificador basado en tabla de decisiones busca por coincidencias exactas en la tabla de decisión utilizando solo las características en el esquema. Si no encuentran instancias coincidentes, la clase con mayor ocurrencia (la clase mayoría)

en la tabla de decisión es retornada; caso contrario, la clase con mayor ocurrencia de todas las instancias coincidentes es retornada. Para construir la tabla de decisiones, el algoritmo de inducción debe decidir que características se incluirán en el esquema y que instancias se almacenarán en el cuerpo. Dado una función f y una hipótesis de clase H , se definen las *características óptimas - optimal features* como las características utilizadas en una hipótesis h en H que tiene la mayor precisión de predicción futura con respecto a f [K⁺⁹⁵]. Así, dada una instancia sin etiquetar I , la etiqueta asignada a la instancias por el clasificador basado en DTM sigue el siguiente proceso. Sea L el conjunto de instancias etiquetadas por el DTM que coinciden exactamente con la instancia dada I , donde solo las características en el esquema son requeridas para realizar la correspondencia y las otras son ignoradas. Si L es un conjunto vacío, se retorna la clase con mayor ocurrencia en la DTM; caso contrario, se retorna la clase con mayor ocurrencia en el conjunto L . Valores desconocidos son tratados como valores distintos en el proceso de correspondencia (*matching process*).

Como en los casos anteriores, mayor detalle acerca de este método de aprendizaje automático puede encontrarse en [K⁺⁹⁵].

4.3.5. Clasificadores bayesianos

Este enfoque está basado en cuantificar las compensaciones (*tradeoffs*) entre varias decisiones de clasificación utilizando las probabilidades y los costos que acompañan a tales decisiones [DHS99]. Para su cometido, este tipo de enfoque hace uso del *Teorema de Bayes*, el cual lo describiremos brevemente junto a su uso dentro del aprendizaje automático. En [Mit97], el autor afirma que dentro del aprendizaje automático a menudo se está interesado en determinar la mejor hipótesis para algún espacio H (donde H es el espacio de funciones/hipótesis candidatas), dado los datos de entrenamiento D . La *mejor hipótesis* se refiere a aquella que se cumpla con mayor probabilidad, dado los datos D más cualquier conocimiento inicial acerca de las probabilidades *a priori* de varias hipótesis en H . El teorema de Bayes proporciona una forma de calcular la probabilidad de una hipótesis basándose en su probabilidad *a priori* y el dato observado. Para definirlo más precisamente, introduciremos algunas notaciones. Sea $P(h)$ la probabilidad inicial de que la hipótesis h se cumpla, antes de observar los datos de entrenamiento. $P(h)$ es la *probabilidad a priori* de h y refleja cualquier conocimiento inicial que se tenga acerca de que h sea una hipótesis correcta. Si no se cuenta con dicha probabilidad inicial, simplemente se asigna la misma probabilidad a cada hipótesis candidata. Del mismo modo, utilizamos $P(D)$ para denotar a la *probabilidad a*

priori que los datos de entrenamiento D sean observados (esto es, la probabilidad de D dado que no se cuenta con el conocimiento de cual hipótesis se cumple). Luego, denotamos con $P(D|h)$ a la probabilidad de observar los datos D dado algún universo en el que la hipótesis h se cumpla. En el aprendizaje automático estamos interesados en la probabilidad $P(h|D)$, esto es, que h se cumpla dado que se ha observado los datos de entrenamiento D . $P(h|D)$ es denominada la *probabilidad a posteriori* de h , ya que refleja nuestra confianza de que h se mantenga (o se cumpla) luego de haber observado los datos de entrenamiento D . Así, el Teorema de Bayes proporciona una forma de calcular la probabilidad *a posteriori* $P(h|D)$, a partir de la probabilidad *a priori* $P(h)$, junto con $P(D)$ y $P(D|h)$.

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

En muchas situaciones, se cuenta con un conjunto de hipótesis H y se está interesado en encontrar la hipótesis más probable $h \in H$ dado los datos de entrenamiento observados (o al menos una de las mayormente probables, si es que existen varias). Tal hipótesis, denominada, *hipótesis máxima a posteriori* (MAP), se puede determinar utilizando el teorema de Bayes para calcular la probabilidad a posteriori de cada hipótesis candidata, esto es:

$$\begin{aligned} h_{MAP} &= \operatorname{argmax}_{h \in H} P(h|D) \\ &= \operatorname{argmax}_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\ &= \operatorname{argmax}_{h \in H} P(D|h)P(h) \end{aligned}$$

Mayor detalle acerca de los fundamentos de este método de aprendizaje se puede ver en [Mit97, WF05].

4.3.6. Clasificadores basados en instancias

El aprendizaje en estos algoritmos consiste simplemente en almacenar los datos de entrenamiento presentados. Cuando una nueva instancia de consulta es encontrada, se procede recuperar un conjunto de instancias relacionadas similares de la memoria y se las utiliza para clasificar la nueva instancia [Mit97]. De acuerdo a [WF05], los algoritmos basados en instancias consisten en el almacenamiento de ejemplos de entrenamiento y el uso de una *función de distancia* con el propósito de determinar cuál

miembro del conjunto de entrenamiento es el más “cercano” a una instancia desconocida. Una vez que el ejemplo (o instancia) de entrenamiento más cercano ha sido localizado, se predice la clase para la instancia desconocida. Uno de los algoritmos de aprendizaje basado en instancias más utilizado es el *k-Nearest Neighbor*.

Función de distancia

La mayoría de los algoritmos basados en instancias utilizan la distancia Euclídea [WF05]. La distancia entre una instancia con los valores de atributos $a_1^{(1)}, a_2^{(1)}, \dots, a_k^{(1)}$ (donde k es el número de atributos) y otra instancia con valores $a_1^{(2)}, a_2^{(2)}, \dots, a_k^{(2)}$ está definida como:

$$\sqrt{\left(a_1^{(1)} - a_1^{(2)}\right)^2 + \left(a_2^{(1)} - a_2^{(2)}\right)^2 + \dots + \left(a_k^{(1)} - a_k^{(2)}\right)^2}$$

Diferentes atributos se encuentran a menudo en diferentes escalas, por tanto cuando se utiliza la distancia Euclídea directamente, el efecto de algunos atributos pueden estar afectados por otros que tenga una mayor escala. Por consiguiente, es usual la normalización de todos los atributos, calculando:

$$a_i = \frac{v_i - \min v_i}{\max v_i - \min v_i}$$

donde v_i es el valor real del atributo i , y el máximo y el mínimo son tomados sobre todas las instancias del conjunto de entrenamiento. Estas fórmulas implícitamente asumen atributos numéricos. Para atributos nominales, aquellos que toman valores simbólicos en lugar de numéricos, la diferencia entre dos valores que no son iguales es representada con 1, mientras que si ambos poseen el mismo valor la diferencia se representa con un 0.

k-Nearest Neighbor

El algoritmo basado en instancias más básico es el *k-Nearest Neighbor - KNN*. Este algoritmo asume que todas las instancias corresponden a puntos en el espacio n -dimensional \mathbb{R}^n . Los vecinos más cercanos (nearest neighbors) de una instancia son definidos en términos de la distancia Euclídea, calculada con la fórmula:

$$d(a^{(i)}, a^{(j)}) \sqrt{\left(a_1^{(i)} - a_1^{(j)}\right)^2 + \left(a_2^{(i)} - a_2^{(j)}\right)^2 + \dots + \left(a_k^{(i)} - a_k^{(j)}\right)^2}$$

para dos instancias i, j . Luego de ello, se obtienen las k menores distancias a los ejemplos de entrenamiento. Por último, la etiqueta de la instancia de consulta consiste en aquella con mayor ocurrencia dentro del conjunto de las k instancias más cercanas. En el algoritmo 1, se describe el algoritmo para el entrenamiento y clasificación del algoritmo KNN.

Algoritmo 1: Algoritmo k-Nearest Neighbor

```

1  $x$ : instancia
2  $y$ : etiqueta o clase de la instancia
3 Algoritmo de entrenamiento:
4 foreach ejemplo de entrenamiento ( $x, y$ ) do
5     agregar el ejemplo a la lista ejemplos_entrenamiento
6 end
7 Algoritmo de Clasificación:
8 Sea  $x_q$  una instancia de consulta a ser clasificada
9 for  $i \leftarrow 1$  to  $m$  do
10    Calcular la distancia  $d(x_i, x_q)$ 
11 end
12 Calcular el conjunto  $I$  que contiene los índices para las  $k$  distancias menores
       $d(x_i, x_q)$ 
13 return La etiqueta o clase con mayor ocurrencia  $y_i$  (donde  $i \in I$ )

```

Mayor detalle acerca de los fundamentos de este método de aprendizaje se puede ver en [Mit97, WF05].

4.3.7. Sistemas ensamblados (Ensemble Systems)

En situaciones en donde existe una gran importancia en las decisiones a tomar, a menudo siempre buscamos una segunda, tercera o muchas más opiniones. En este proceso, le damos un valor de importancia a cada opinión individual y las combinamos con el fin de obtener una decisión final. Así, este proceso se convierte en una “consulta a varios expertos” antes de tomar una decisión final. De manera análoga, los *sistemas ensamblados (ensemble systems)*, combinan los resultados de clasificadores individuales con la finalidad de producir un resultado mejor al que daría un único clasificador [Pol06]. En la figura 4.6 se puede observar el proceso de combinación de tres algoritmos de clasificación para conformar el sistema ensamblado.

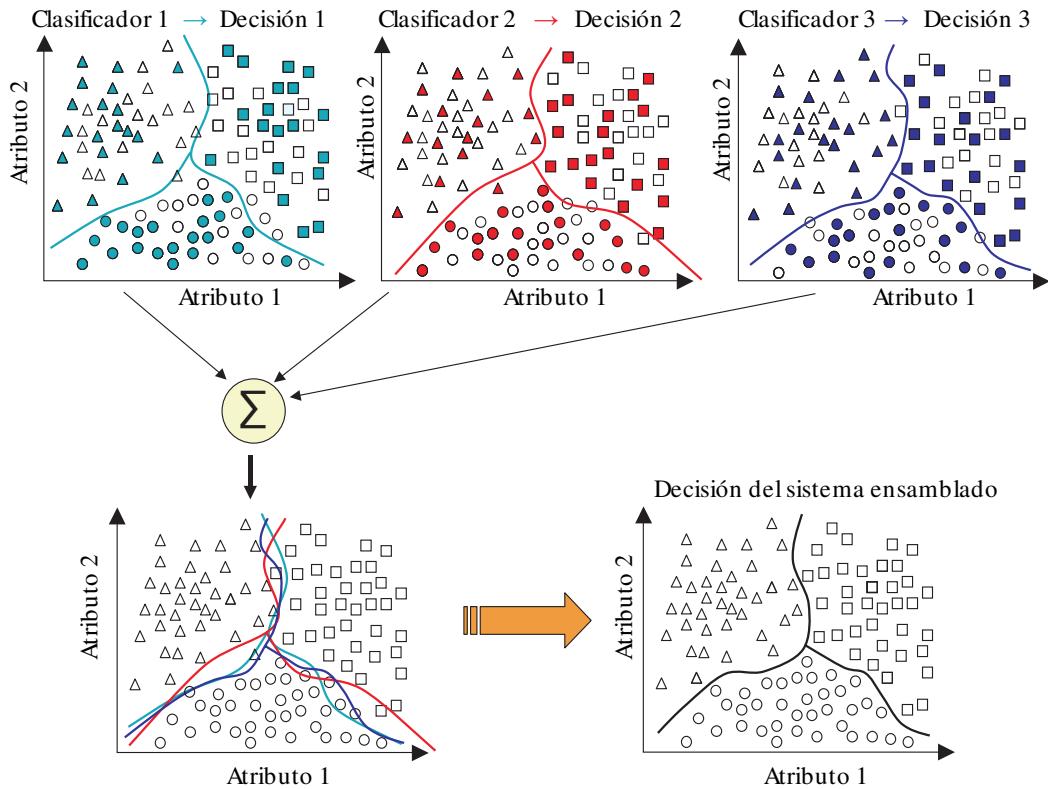


Figura. 4.6 Combinación de diferentes clasificadores [Pol06].

Existe diversas formas de combinar los clasificadores, entre los cuales se puede citar: votación por mayoría, bagging, boosting, stacking, entre otros. Mayor detalle acerca de los fundamentos de este método de aprendizaje se puede ver en [Pol06, WF05].

4.4. Estimación de Error

En los problemas de clasificación, es natural medir el rendimiento de un clasificador en términos de *tasa de error* [WF05]. El clasificador predice la clase de cada instancia: si es correcta, se cuenta como un *acerto*, caso contrario es un *error*. Así, la tasa de error es nada más que la proporción de errores hechos sobre la totalidad de las instancias. Por tanto, para predecir el rendimiento de un clasificador sobre nuevos datos, necesitamos evaluar su tasa de error sobre un conjunto de datos que no forma parte de la formación (proceso de aprendizaje) del clasificador. Este conjunto de datos independiente es lo que anteriormente se denominó *conjunto de prueba (testing set)*. En el caso de tenerse una gran cantidad de datos, se puede tomar una gran muestra para formar el conjunto de entrenamiento e independientemente, tomar la parte restante para el

conjunto de prueba, así, la tasa de error sobre el conjunto de prueba dará una buena indicación sobre el rendimiento del clasificador. Desafortunadamente, en la realidad, existe una escasa cantidad de datos. Esto limita el número de datos que pueden ser utilizados para el proceso de validación (testing) del clasificador. Una alternativa a ello es la reutilización ingeniosa de tal cantidad limitada de datos dentro del proceso de validación con el fin de obtener un número suficientemente grande de muestras. A este tipo de “reuso” se lo denomina *remuestreo (resampling)* [JS11a]. Las técnicas de remuestreo se pueden clasificar nuevamente en: *remuestreo simple (simple resampling)* y *remuestreo múltiple (multiple resampling)*. Las técnicas de remuestreo simple tienden a usar un dato una sola vez, en tanto que las técnicas de remuestreo múltiple tienden a reutilizar el dato más de una vez mediante aplicaciones de técnicas de remuestreo. Una de las técnicas de remuestreo más ampliamente utilizadas dentro de la estimación de errores y medidas de rendimiento de algoritmos de aprendizaje automático es la *validación cruzada (cross validation)* que se describirá a continuación.

4.4.1. Validación cruzada - Cross Validation (CV)

La técnica de remuestreo *Cross Validation*, y en particular la técnica *k-fold Cross Validation* consiste en la división de un conjunto de datos S de m muestras en k subconjuntos de datos con aproximadamente igual tamaño. Cada subconjunto se denomina *fold*. El algoritmo de aprendizaje es entrenado con los $k - 1$ subconjuntos y es validado con el subconjunto k restante. Este procedimiento es repetido k veces con un subconjunto de validación diferente en cada iteración. Así, cada uno de los k *folds* se convierte en el conjunto de prueba una vez. El proceso retorna k estimaciones de error correspondiente a las tasas de error generadas por el clasificador en cada iteración. Se obtiene el promedio de dichas estimaciones, el cual representa el error estimado global para el algoritmo. En el algoritmo 2 se muestra el procedimiento realizado durante el *k-fold CV*. Un ejemplo de validación cruzada se puede ver en la figura 4.7 para $k = 5$ iteraciones de prueba donde la quinta parte de los datos se utilizará para validación. De acuerdo a [JS11a, K⁺95], el valor de k más utilizado en el caso de los algoritmos de clasificación es de $k = 10$, con lo cual el conjunto de datos es dividido en 10 partes iguales, realizándose 10 iteraciones del proceso de entrenamiento.

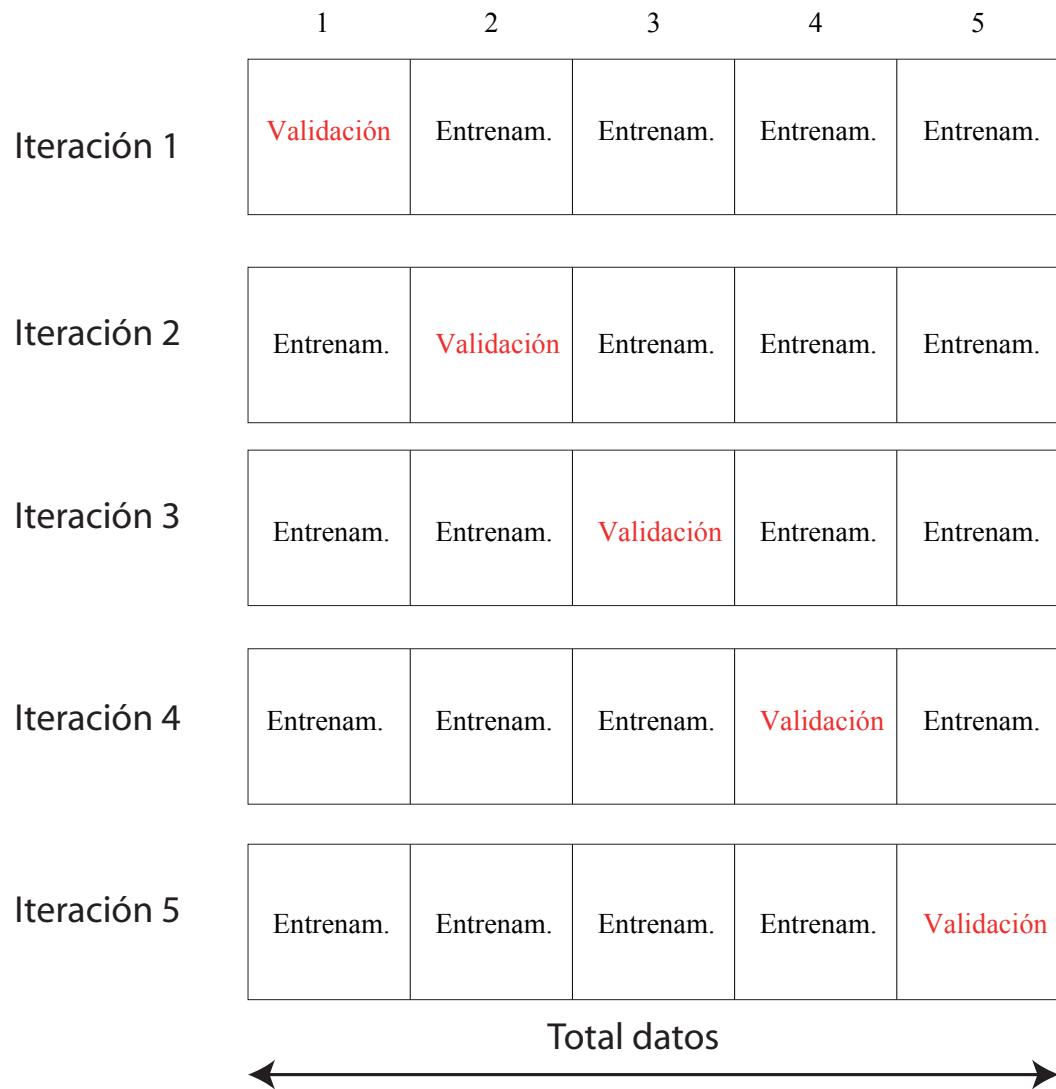


Figura. 4.7 Método de validación cruzada con 5 iteraciones.

Algoritmo 2: Algoritmo k-fold CV [JS11a]

```

1  $S$ : conjunto de entrenamiento.
2  $m$ : tamaño del conjunto  $S$ .
3  $k$ : número de divisiones.
4 Dividir el conjunto de entrenamiento  $S$  disponible en  $k$  subconjuntos disjuntos
    $S_i$ ,  $i = 1, \dots, k$  de tamaño  $\frac{m}{k}$ (aproximadamente).
5 Inicializar  $i = 1$ .
6 while  $i <= k$  do
7   Marcar el  $i$ -ésimo subconjunto  $S_i$  como conjunto de validación (test set).
8   Para el conjunto de validación  $S_i$ , generar el conjunto de entrenamiento  $S_T$ 
     que contenga todos los ejemplos (instancias) de  $S$  excepto aquellos incluidos
     en  $S_i$ . Esto es:  $S_T = \{x / x \in S \wedge x \notin S_i\}$ .
9   Entrenar y validar el algoritmo de aprendizaje sobre  $S_T$  y  $S_i$ ,
     respectivamente.
10  Obtener la(s) métrica(s) de interés del algoritmo de aprendizaje obtenido
     como resultado del entrenamiento sobre  $S_T$ .
11   $i \leftarrow i + 1$ 
12 end
13 Obtener el promedio de la(s) métrica(s) sobre las  $k$  iteraciones.
14 return el(los) promedio de la(s) métrica(s) analizada(s).

```

4.4.2. Stratified k-fold Cross Validation

Incluso con el uso de métodos de remuestreo como el *k-fold cross validation*, la división de los datos en entrenamiento y prueba puede ser “desigual”. Esto es, la división no podría tener en cuenta la distribución de las etiquetas (clases) de los ejemplos mientras se generan los subconjuntos de entrenamiento y de prueba. Ejemplo: considere un problema de clasificación con tres clases. Las clases son: y_1 , y_2 , y_3 . Asuma que el conjunto de datos tiene la siguiente distribución: 30 % de los ejemplos son de la clase y_1 , 60 % de los ejemplos son de la clase y_2 y 10 % corresponden a la clase y_3 . Una división aleatoria podría bien omitir los ejemplos de la clases y_3 en cualquiera de los subconjuntos (entrenamiento o prueba). Una solución a ello consisten en la *estratificación (stratification)* de los datos. La *estratificación (stratification)* consiste en tener en cuenta la representación de cada clase en el dataset global y asegurarse de que dicha representación es respetada tanto en el subconjunto de entrenamiento como en el de prueba [JS11a]. En el algoritmo 3 se muestra el procedimiento realizado para el *Stratified k-fold Cross Validation*.

Algoritmo 3: Algoritmo Stratified k-fold CV [JS11a]

-
- 1 D : conjunto de entrenamiento
 - 2 l : número de clases
 - 3 k : número de divisiones
 - 4 Dividir el conjunto de entrenamiento S disponible en l conjuntos (uno para cada clase).
 - 5 Generar k subconjuntos para cada uno de los l conjuntos.
 - 6 Combinar un subconjunto a partir de cada uno de los l conjuntos para obtener k subconjuntos que mantengan la distribución de clases original.
 - 7 Realizar el procedimiento *k-fold CV* sobre estos k subconjuntos.
-

4.5. Resumen

Los algoritmos de aprendizaje automático han demostrado tener una gran aplicabilidad en una amplia variedad de dominios, principalmente en los problemas de *Minería de Datos*.

Entre los tipos de aprendizaje más comunes se encuentran: el aprendizaje supervisado, no supervisado y el aprendizaje por refuerzo.

Uno de los problemas ampliamente abarcados en el aprendizaje automático es la clasificación de ítems. Dicho problema se refiere a la asignación de una categoría o clase a un objeto en base a un conjunto de atributos observados relacionados al objeto.

A partir del problema de clasificación de ítems, surgen diversos algoritmos de aprendizaje automático que tratan este problema, cada uno con una perspectiva diferente. Entre las principales categorías de tales algoritmos se encuentran: los basados en árboles, las máquinas de soporte vectorial, los basados en redes neuronales, los clasificadores bayesianos, entre otros.

Una manera válida para medir el rendimiento de un algoritmo de clasificación es la tasa de errores, esto es la proporción de desaciertos a la hora de clasificar un grupo de instancias. Un método común para calcular la tasa de errores es por medio de la validación cruzada, en la cual se separan, en n iteraciones, los datos que serán utilizados como entrenamiento de los datos de prueba. Se obtiene en cada iteración la tasa de errores correspondiente, y al finalizar el proceso, se toma como resultado global el promedio de la tasa de errores de todas las iteraciones.

Capítulo 5

Modelado del problema

Como lo mencionábamos en el capítulo 1, el servicio de telefonía móvil se ha convertido en una de las áreas de mayor penetración a nivel país. Esto es debido a que en la actualidad se experimenta un gran incremento en el uso de dispositivos móviles inteligentes (smartphones, tablets), los cuales ofrecen a los usuarios la posibilidad de conectarse a Internet en el momento y lugar que lo deseen a través de las redes móviles e inalámbricas. A causa de estos factores se ha generado una gran demanda por los servicios de datos móviles, con lo cual la gestión y el aseguramiento de la “calidad” de estos servicios se ha convertido en todo un reto para las operadoras. Por sobre todo, la monitorización automática de parámetros técnicos y su relación con la calidad experimentada por el usuario se han convertido en factores claves para promover el mejoramiento en la prestación de servicios. Debido a su importancia dentro del aseguramiento de la calidad del servicio y de la experiencia del usuario, abordaremos el problema del modelado de la correlación entre la calidad de experiencia y los parámetros que la afectan, como así también su monitorización en el campo de los servicios de datos móviles.

5.1. Metodología de evaluación

Rememorando, en el capítulo 3 se presentó una metodología de evaluación de calidad de la experiencia que combina los métodos de evaluación objetivos y subjetivos. Esta metodología, denominada *PSQA - Pseudo Subjective Quality Assessment*, nos permite obtener un valor de la calidad experimentada cercano al valor promedio que nos daría un observador humano. Así, para este trabajo utilizaremos una adaptación de dicha metodología de evaluación con el fin de generar un modelo de correlación entre

5.1 Metodología de evaluación

la calidad experimentada por el usuario y los parámetros de calidad de servicio a nivel de red, dispositivo y aplicación. En la figura 5.1 se puede observar el procedimiento general llevado a cabo para la obtención del modelo de correlación mencionado.

5.1 Metodología de evaluación

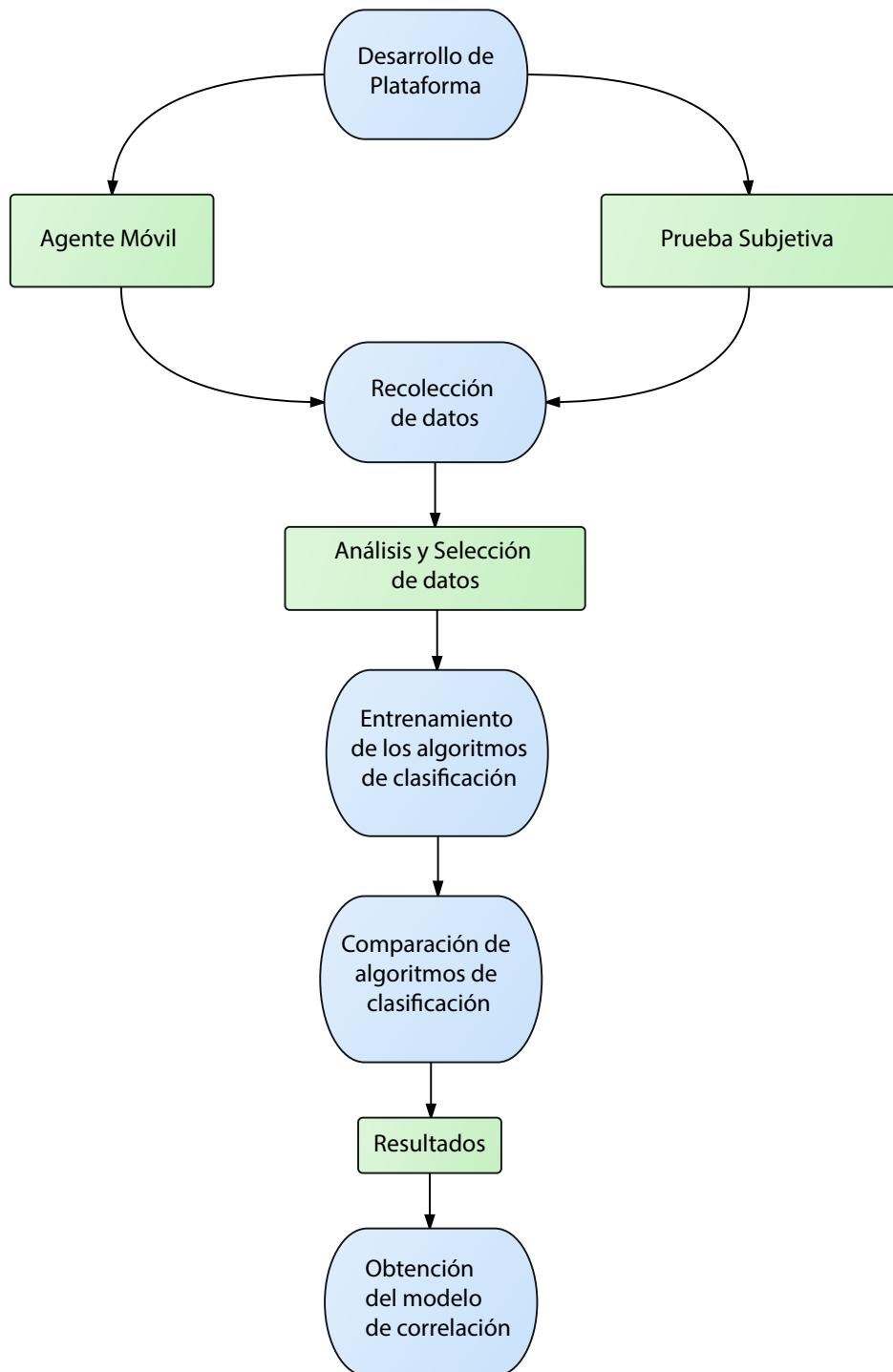


Figura. 5.1 Vista general del proceso seguido para la obtención del modelo de correlación QoE.

5.2. Plataforma de transmisión de datos móviles

El proceso de recolección de datos consiste en obtener la información necesaria acerca de cómo los usuarios evalúan la calidad de distintos servicios móviles ofrecidos bajo diversas condiciones de red y utilizando diferentes dispositivos móviles. Debido a la importancia de la diversidad de condiciones de red para este estudio y sumando a ello el hecho de no contar con un entorno propicio para la realización del proceso de recolección, se optó por construir un entorno de laboratorio conformado por una plataforma de transmisión de datos móviles que nos permite emular los escenarios de una red de transmisión celular teniendo un mayor control de las condiciones de la red y que a la vez nos ofrece facilidad a la hora de realizar de mediciones de parámetros de red en los dispositivos. Así, en esta sección se detallará la arquitectura, los componentes y las funciones de los mismos dentro de dicho entorno. En la figura 5.2 se puede observar la composición general de la plataforma de transmisión.

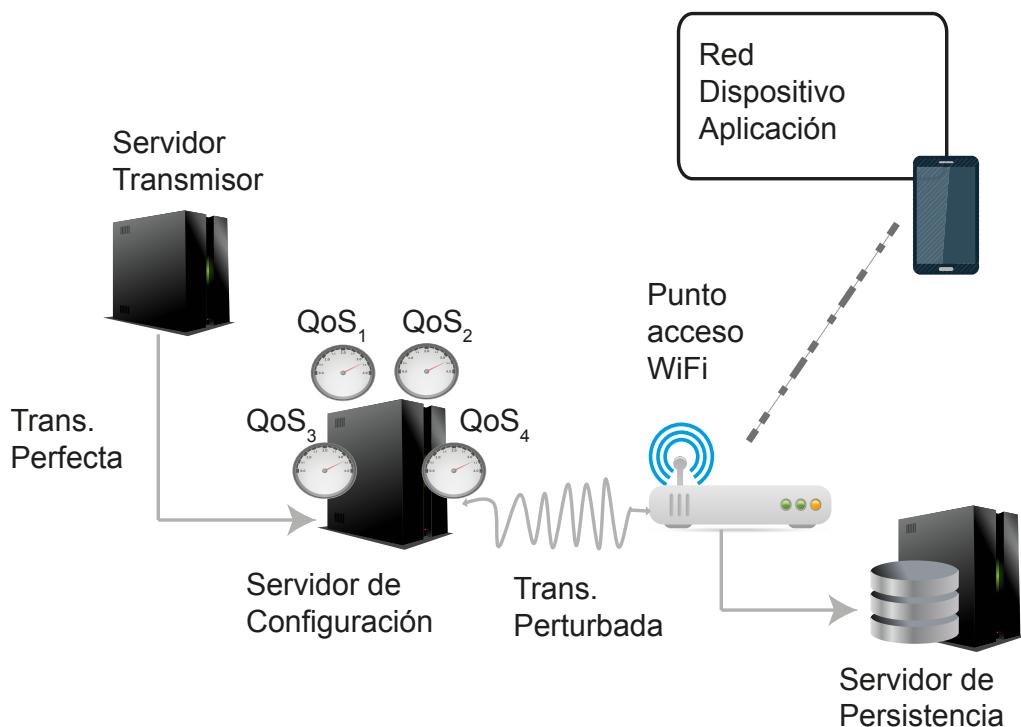


Figura. 5.2 Plataforma de transmisión de datos móviles

5.2.1. Servidor transmisor

Este servidor es el encargado de almacenar el contenido de los servicios (videos, página web, ver Apéndice A) y luego transmitirlo a los diversos dispositivos conectados a la plataforma. A parte de ello, este servidor forma parte del proceso de medición de parámetros de red. Para lograr la transmisión de datos hacia los dispositivos móviles hace uso de diversas tecnologías web:

NGINX

*NGINX*¹ es un servidor web/proxy *open source* ligero de alto rendimiento y bajo consumo de memoria. Su uso en conjunto con el módulo *ngx_http_mp4* permite su utilización como servidor streaming de archivos *MP4*.

Progressive Download

El servidor web *NGINX* permite hacer uso de una técnica de transmisión denominada *Progressive Download*. Esta técnica permite la transferencia de archivos multimedia desde el servidor a un cliente a través del protocolo *HTTP*. Esto permite el uso del buffer en el cliente como almacenamiento temporal hasta que el contenido sea completamente descargado.

5.2.2. Servidor de configuración

El componente central de la arquitectura de transmisión es el *servidor de configuración*. Este servidor recibe una transmisión perfecta por parte del servidor transmisor y es el encargado de realizar las variaciones de las condiciones de red, de forma a generar tráfico imperfecto. Para lograr lo anterior, el servidor modifica cuatro parámetros básicos de calidad del servicio: *ancho de banda, delay, jitter y paquetes perdidos* (más detalles, ver Apéndice B). Para cumplir su cometido, este servidor hace uso de comandos de control de tráfico incluidos en el sistema operativo Linux:

- **tc [Hub]**: es un comando de Linux (parte del paquete *iproute2*) para configurar y controlar el planificador de red del kernel de Linux. Permite mostrar y configurar el control de tráfico de red.

¹<http://nginx.org/>

5.2 Plataforma de transmisión de datos móviles

- **netem** [Fou]: usado en conjunto con **tc**, ofrece la funcionalidad de emular diversas propiedades de una red, entre ellas: el porcentaje de paquetes perdidos, el delay y el jitter.

En base a estos comandos se ha generado un script bash (sh) utilizado para configurar los parámetros de red en el servidor de configuración (ver en Apéndice B).

Como resultado de la aplicación de estas herramientas de control de tráfico, todo el tráfico dirigido desde el servidor transmisor hacia los dispositivos móviles es alterado de acuerdo a la variación de los parámetros de red ya mencionados. Así, como se puede observar en la figura 5.2, este servidor actúa como *puerta de enlace* entre el servidor transmisor y el punto de acceso WiFi por el cual se conectan los dispositivos móviles.

5.2.3. Punto de acceso WiFi

Como su nombre lo indica, es el punto de acceso de los dispositivos móviles a la red conformada por el servidor de configuración y de transmisión. Dentro del entorno de laboratorio, el punto de acceso WiFi juega una rol similar a las torres de transmisión de datos. Así, todo el tráfico generado por el transmisor y variado previamente por el servidor de configuración, pasa a través del punto de acceso WiFi y fluye hasta los dispositivos móviles conectados, los cuales consumen los contenidos (páginas web o videos).

5.2.4. Dispositivos móviles

Se escogieron dispositivos móviles con sistema operativo Android debido a su facilidad y flexibilidad a la hora de desarrollar aplicaciones para el mismo. Estos dispositivos son los encargados de conectarse al punto de acceso WiFi y ejecutar una aplicación Android de encuestas (pruebas subjetivas) junto con un agente móvil. Los datos obtenidos por la aplicación Android y el agente móvil serán almacenados en el *servidor de persistencia*.

5.2.5. Servidor de persistencia

El servidor de persistencia es el encargado de servir como repositorio de información de los datos enviados por los dispositivos móviles y que posteriormente serán utilizados como datos de entrada para los algoritmos de clasificación utilizados en el

proceso de aprendizaje automático. Entre los datos enviados por los dispositivos móviles se encuentran: las evaluaciones realizadas por los usuarios, información acerca del dispositivo, información acerca del estado del dispositivo, entre otros.

5.3. Aplicación móvil Android

De acuerdo al procedimiento expuesto en la sección 5.1, para la obtención de los datos se debe realizar un conjunto de pruebas subjetivas con la finalidad de obtener la apreciación del usuario acerca de la calidad experimentada con los servicios móviles. Para realizar este conjunto de pruebas subjetivas se utilizó la plataforma de transmisión de datos móviles descripta anteriormente y se desarrolló una aplicación Android cuyo propósito es realizar una encuesta a los usuarios acerca de la calidad experimentada con servicios determinados, recolectar los parámetros de red y obtener las características y el estado del dispositivo móvil que la está ejecutando. Esta aplicación se divide en dos partes principales: *prueba subjetiva* y *agente móvil*.

5.3.1. Prueba subjetiva

Una prueba subjetiva consiste en la evaluación de la calidad percibida por parte de los usuarios cuando se experimentan diferentes degradaciones en ciertos parámetros de red (parámetros QoS). Para ello, se les presenta a los usuarios una aplicación en forma de encuesta. En dicha encuesta se les expone a dos contextos de uso cotidiano de datos móviles: *navegación web* y *visualización de videos*. Al finalizar la visualización (web, video), se le pide a cada participante que evalúe de acuerdo a diversos criterios la calidad experimentada. La evaluación es realizada mediante la selección de un valor dentro de una escala absoluta MOS de cinco puntos (ver cuadro 3.2).

Navegación Web

Uno de los contextos de uso cotidiano de las redes de datos móviles es la navegación web. El objetivo de esta prueba es obtener un estimativo acerca de la calidad experimentada por el usuario con respecto a la rapidez con la que se cargan los contenidos de la página web. La página web de prueba consiste en un conjunto de animaciones en formato GIF (*Graphics Interchange Format*) junto con un pequeño texto descriptivo que acompaña a cada una. En [KH02], los autores analizan la calidad de la experiencia en el contexto de la navegación web, realizando un análisis no solo de parámetros

técnicos (ancho de banda, delay), sino también de parámetros “más cercanos a los usuarios” como el tiempo de respuesta y la tasa cancelación de carga de la página. En base a esta referencia, en este tipo de prueba se realizarán mediciones acerca del tiempo aproximado de carga de la página (en segundos) y se registrará si la página en cuestión fue cargada completamente o fue cancelada. Estos datos también forman parte del conjunto de parámetros enviados al servidor de persistencia. Por otra parte, esta prueba subjetiva en el contexto de navegación web adopta las recomendaciones establecidas en ITU Rec. P.1501 [ITU14].

Streaming de video sobre HTTP

Otro contexto de uso importante de las redes de datos móviles es la visualización de videos mediante streaming. En la actualidad servicios como Youtube generan una gran cantidad de tráfico de datos móviles. Por este motivo, este contexto también forma parte del conjunto de las pruebas subjetivas. Debido a que el streaming de contenidos se realiza por medio de HTTP (sobre TCP, orientado a la conexión), no existen degradaciones de calidad experimentada durante la transmisión, en cambio surgen otros eventos que podrían afectar al QoE. Entre ellos, uno de los principales es el evento de *rebuffering*. El evento de *rebuffering* consiste en la interrupción de reproducción de un video debido a las condiciones de la red. En [DPDMJ⁺13], los autores realizan experimentos con el objetivo de cuantificar la influencia de estos eventos en la calidad experimentada por el usuario. En [MCC11], los autores definen tres métricas influyentes en la calidad de la experiencia sobre video streaming:

- Tiempo de buffering inicial: métrica que mide el tiempo entre la carga del video y el comienzo de su reproducción.
- Promedio de duración de buffering: mide la duración promedio de los eventos de rebuffering.
- Frecuencia de rebuffering: mide la frecuencia de ocurrencias de eventos de rebuffering.

En vista de la importante influencia de los eventos de rebuffering en la calidad experimentada dentro del contexto del streaming de video sobre HTTP, se realizaron recolecciones de las métricas mencionadas, así como del evento de cancelación de la reproducción del video. Por lo tanto, dichas métricas también forman parte del conjunto de datos recolectados durante las pruebas subjetivas. Por tanto, el objetivo

principal de esta prueba subjetiva es el de evaluar la fluidez en la reproducción de video cuando éste se transmite sobre HTTP.

5.3.2. Agente móvil

Junto a la aplicación Android para la realización de las encuestas, se ejecutó en forma de servicios (en *background*) un agente móvil capaz de recabar información acerca del dispositivo, tráfico y estado de la red, uso de la CPU, uso de la memoria RAM, nivel de la batería del dispositivo, entre otros. Para medir de manera precisa los parámetros de red (ancho de banda, paquetes perdidos, delay y jitter) se hicieron uso de dos herramientas básicas en el monitoreo de redes en Linux: PING e IPERF².

PING

Es una utilidad de diagnóstico que permite comprobar la accesibilidad a un host remoto mediante el envío de mensajes *ICMP*. Por medio de la misma es posible obtener la latencia (RTT) entre el servidor transmisor y el dispositivo móvil.

IPERF

Es una herramienta de medición de diversos parámetros de red que inyecta flujos de datos TCP y UDP a la red. Para ello, utiliza un esquema cliente-servidor, el cual permite medir los parámetros en ambas direcciones. En el entorno creado, el servidor transmisor representa al lado del servidor mientras que el dispositivo móvil sirve como cliente (ver figura 5.2). Debido a que en la ejecución de IPERF el cliente es quien inyecta el flujo de datos, se debió ejecutar el programa en modo reverso de manera a obtener el rendimiento de la red desde el servidor hasta el dispositivo móvil. La versión de IPERF utilizada tanto en el servidor como en el dispositivo móvil es la 2.0.5. Cabe destacar que la herramienta IPERF no se encuentra disponible de forma nativa para dispositivos Android. Para ejecutarlo se realizó un proceso de compilación (*cross-compiling*) del código fuente por medio del *Android Native Development Kit - NDK*, con lo cual se obtiene un ejecutable binario compatible con dispositivos con procesadores ARM.

Una descripción más detallada del juego completo de herramientas utilizadas se puede ver en el Apéndice C.

²<https://iperf.fr/>

5.4. Recolección de datos

El proceso de recolección de datos consiste en la realización de pruebas subjetivas con usuarios reales utilizando la plataforma de transmisión de datos móviles expuesta en la figura 5.2 y la aplicación Android desarrollada. Cada prueba subjetiva se realiza bajo condiciones de red distintas con el fin de emular las posibles condiciones que podrían darse en una red celular real. Para generar las distintas condiciones de red utilizadas en este experimento escogimos diversos valores para los parámetros de red seleccionados que pueden ser observados en el cuadro 5.1.

Cuadro. 5.1 Parámetros de red establecidos en el servidor de configuración

Parámetros	Valores	Unidades
Ancho de banda	350, 750, 1200, 1600	kbps.
Delay	100, 300, 500	ms.
Paquetes perdidos	0, 2,5, 5, 10	%
Jitter	0, 25, 50	ms.

De esta manera, cada condición de red está dada por las diversas combinaciones de los valores presentados en el cuadro 5.1. Todas estas configuraciones son guardadas en un archivo de configuración, el cual es utilizado por el servidor de configuración para establecer la condición de red durante el proceso de pruebas subjetivas.

Así, el proceso de cada prueba subjetiva realizada por un usuario consta de los siguientes pasos:

1. El usuario instala la aplicación Android desarrollada en su dispositivo móvil y se conecta al punto de acceso WiFi indicado.
2. Una vez iniciada la aplicación, se solicita al usuario completar su información personal básica (sexo, edad, entre otros).
3. La aplicación móvil solicita una nueva configuración de parámetros de red al servidor de configuración, la cual es generada en base a los valores de red presentados en la tabla 5.1. Esta configuración se mantendrá durante toda la prueba subjetiva. Cuando se realiza una nueva prueba, la configuración de las condiciones de red estarán condicionados por valores diferentes al anterior.

5.4 Recolección de datos

4. A partir de este punto la recolección consta de dos fases:

- Medición de parámetros: en este punto se ejecutan las instrucciones de medición de los parámetros de red (ping e iperf), en tanto que se inician los servicios de monitoreo de los parámetros a nivel de dispositivo y aplicación.
- Evaluación de la calidad de la experiencia: se realizan los test de calidad de navegación web y visualización de videos. Al finalizar cada test se le solicita al usuario información acerca de la calidad percibida, teniendo en cuenta una serie de aspectos específicos para cada servicio y la calidad global experimentada. La información obtenida a partir de las respuestas del usuario a estas cuestiones es un valor MOS (Malo, Pobre, Bueno, Muy Bueno o Excelente) para cada contexto de uso.

5. Las datos recolectados por la aplicación móvil en las fases anteriores son enviados al servidor de persistencia para su almacenamiento.

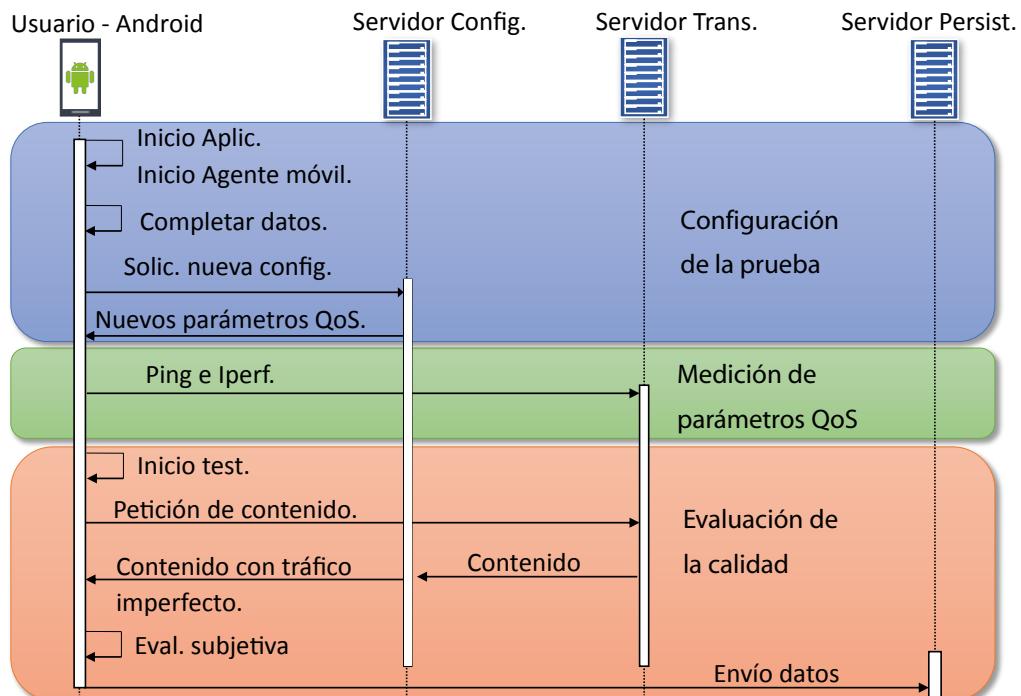


Figura. 5.3 Proceso seguido por cada prueba subjetiva realizada por un usuario.

Las capturas de pantalla correspondientes al procedimiento de las pruebas subjetivas puede ser visualizada en el Apéndice C. En el cuadro 5.2 se muestran los datos recolectados durante las pruebas subjetivas.

5.5 Análisis y selección de Datos

Cuadro. 5.2 Datos recolectados durante el proceso de recolección de datos.

Componente	Datos
Dispositivo	Vendedor, marca, modelo, cantidad de RAM, resolución de pantalla, número de núcleos de CPU, frecuencia de CPU, gama, versión del sistema operativo
Datos del Usuario	Edad, sexo, profesión, frecuencia en el uso de datos móviles, aplicaciones de uso frecuente.
Estado del dispositivo	Porcentaje de uso de la CPU, porcentaje de uso de la memoria RAM, porcentaje restante de batería, tipo de conexión a datos, intensidad de la señal de la conexión activa.
Mediciones de Red	Latencia (configurado y efectivo), ancho de banda (configurado y efectivo), paquetes perdidos (configurado y efectivo), jitter (configurado y efectivo).
Prueba	Fecha y hora de realización, el valor <i>MOS</i> y su correspondiente test (navegación web o visualización de video).
Navegación Web	Tiempo de carga de la página web. Un número que indica el evento de cancelación de carga (0= no cancelado, 1= cancelado).
Video streaming	Tiempo de carga inicial, cantidad de eventos de rebuffering, tiempo promedio de los eventos de rebuffering. Un número que indica el evento de cancelación de reproducción (0= no cancelado, 1= cancelado).

5.5. Análisis y selección de Datos

Los datos obtenidos en el proceso anterior luego pasan por un análisis que nos permite detectar y eliminar los datos inconsistentes o fuera de rango, debido a que estos presentan comportamientos atípicos que podrían influir en el entrenamiento de los algoritmos de clasificación.

Durante este proceso se analizó cada prueba subjetiva recolectada teniendo en cuenta los siguientes criterios para la eliminación de las mismas:

- Medición correcta de los parámetros recolectados por la aplicación (parámetros de red, estado del dispositivo, entre otros.)
- Realización de las pruebas subjetivas correctamente. Por ejemplo, mediante la observación de los datos podemos identificar si el usuario canceló el test antes de realizar la visualización del video o de la página web.
- Concordancia entre la evaluación de calidad dada por el usuario en cada test y los parámetros de red efectivos (medidos desde el dispositivo móvil).

Las pruebas subjetivas que cumplían con los criterios mencionados se utilizan como base para el proceso de entrenamiento de los algoritmos de clasificación que detallaremos en el siguiente capítulo.

5.6. Resumen

Este capítulo presentó la metodología utilizada (la cual es una adaptación de PS-QA) en el proceso de creación de un modelo de correlación del QoE y los parámetros a nivel de dispositivo, red y aplicación. También se describió en detalle el proceso de recolección de datos y los componentes principales dentro de este proceso: la plataforma de transmisión de datos móviles, la cual otorga la capacidad de emular diversas condiciones de red de forma controlada, como así también la aplicación móvil Android, la cual posibilita la realización de pruebas subjetivas y la recolección de diversos parámetros a nivel de red, dispositivo y aplicación, los cuales serán utilizados en el proceso de entrenamiento de los algoritmos de clasificación.

Capítulo 6

Entrenamiento y comparación de algoritmos de clasificación

En este capítulo se describirá en detalle el proceso de entrenamiento, evaluación y comparación de los distintos algoritmos de aprendizaje ofrecidos por el software WEKA [HFH⁺09]. Para ello, en primera instancia, se presentarán los datos obtenidos a partir de las pruebas subjetivas realizadas (las cuales servirán como entrada en el proceso de entrenamiento de los algoritmos de aprendizaje). Posteriormente, describiremos las distintas métricas de rendimiento aplicables al problema tratado. Por último se analizarán los resultados de aplicar los algoritmos de aprendizaje en base a las métricas citadas.

6.1. Resultados de las pruebas subjetivas

Como parte del proceso de pruebas subjetivas, fueron encuestadas un total de 70 personas de entre 22 a 40 años, con diversos perfiles (estudiantes, funcionarios, entre otros), cada una de ellas realizó 5 pruebas con distintas configuraciones de red. En cuanto a los dispositivos móviles, se contabilizó un total de 30 dispositivos diferentes catalogados en tres gamas (baja, media y alta) de acuerdo a las prestaciones del procesador y el procesador gráfico, la memoria RAM y los detalles de la pantalla (resolución y densidad de píxeles por pulgada).

Posterior al proceso de recolección, de acuerdo al capítulo anterior, se realizó un proceso de análisis y limpieza de datos (*data cleaning*). Como resultado del procesamiento de datos anterior se obtuvieron los siguientes resultados:

6.2 Definición de los conjuntos de datos

- 295 muestras correspondientes a las pruebas subjetivas basadas en streaming de video.
- 320 muestras correspondientes a las pruebas subjetivas basadas en navegación web.

6.2. Definición de los conjuntos de datos

El conjunto de datos de entrada al proceso de entrenamiento, de acuerdo a lo mencionado en el capítulo 4, constituye el conjunto de ejemplos utilizados en el proceso de entrenamiento y evaluación del algoritmo de aprendizaje. En el presente caso, cada una de las pruebas subjetivas corresponde a un ejemplo o instancia del conjunto, mientras que cada uno de los datos recolectados por la misma corresponde a un atributo de la instancia.

De acuerdo al contexto analizado, se dividieron los datos de las pruebas subjetivas recolectadas en diferentes conjuntos: uno para navegación web y otro para el streaming de video.

Como parte de la evaluación de la calidad de la experiencia en dichos contexto de uso, hemos decidido analizar los atributos de las instancias de los conjuntos en tres categorías diferentes: correspondientes al dispositivo, correspondientes al estado de la red y aquellos correspondientes a métricas a nivel de aplicación. En el cuadro 6.1 se detallarán los atributos que forman parte de las instancias de los conjuntos de datos (navegación web y streaming de video), separados en las categorías mencionadas e indicando el contexto de uso.

Como se puede observar en el cuadro, las diferencias entre ambos conjuntos se encuentran en los atributos a nivel de aplicación, cada uno correspondiente al contexto de uso en cuestión. A cada una de las instancias de los conjuntos de datos, tanto para el conjunto de navegación web como el de streaming de video, le acompaña una clase o etiqueta de la instancia. Este atributo corresponde al valor MOS asignado por los usuarios para el respectivo contexto de uso, éste puede tomar los siguientes valores: Malo, Pobre, Bueno, Muy Bueno y Excelente.

6.3. Distribución de clases

Como se había mencionado anteriormente, se obtuvo como resultado del proceso de recolección y análisis de datos lo siguiente:

- 295 muestras correspondientes a las pruebas subjetivas basadas en streaming de video .
- 320 muestras correspondientes a las pruebas subjetivas basadas en navegación web.

En las figuras 6.1 y 6.2 se presenta la distribución de clases para cada uno de los contextos de uso.

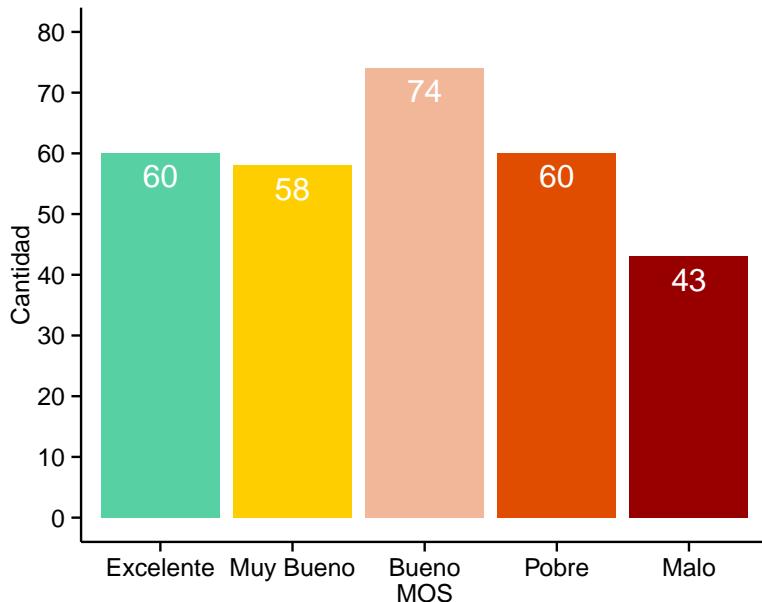


Figura. 6.1 Distribución de clases para el conjunto de datos de streaming de video.

6.3 Distribución de clases

Cuadro. 6.1 Atributos de las instancias de los conjuntos de datos

Categoría	Atributo	Descripción	Contexto
Dispositivo	Uso de CPU	Porcentaje de uso de la CPU del dispositivo.	Web/Video
	Uso de RAM	Porcentaje de uso de la memoria RAM del dispositivo.	Web/Video
	RAM	Cantidad total de memoria disponible medida en MB.	Web/Video
	Nivel de batería	Porcentaje de batería disponible.	Web/Video
	Gama	Categorización del dispositivo de acuerdo a sus prestaciones.	Web/Video
Red	Ancho de Banda	Ancho de banda efectivo medido en kbps.	Web/Video
	Paquetes Perdidos	Porcentaje de perdida de paquetes.	Web/Video
	Delay (RTT)	Tiempo de viaje de ida y vuelta de un paquetes medido en ms.	Web/Video
	Jitter	Variación del delay en ms.	Web/Video
Aplicación	Tiempo de carga web	Cantidad transcurrida de tiempo durante el proceso de carga de la página web medida en segundos.	Web
	Cancelado	Atributo binario que indica la cancelación del proceso de carga de la pág. web.	Web
	Tiempo de carga inicial del video	Cantidad transcurrida de tiempo durante el proceso de carga inicial del video.	Video
	Tiempo promedio de buffering	Duración promedio de los eventos de rebuffing.	Video
	Frecuencia de buffering	Cantidad de ocurrencias de eventos de rebuffing.	Video
	Cancelado	Atributo binario que indica la cancelación de la reproducción del video.	Video

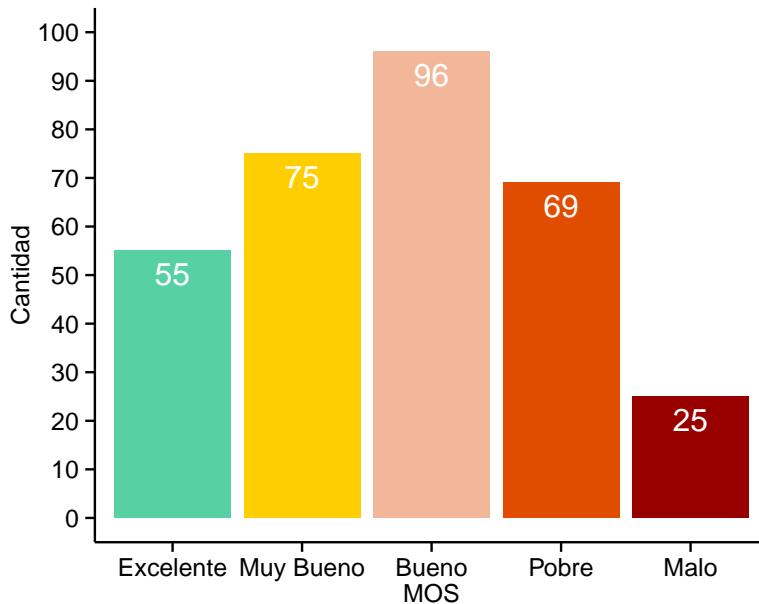


Figura. 6.2 Distribución de clases para el conjunto de datos de navegación web.

Debido a la inequidad en la cantidad de ocurrencias de cada uno de los valores de las clases se realizó un proceso de *balanceo* mediante el cual se generan nuevos conjuntos de datos (uno para cada contexto de uso) que presentan una distribución equitativa de los valores de las clases. Para realizar el proceso de balanceo se utilizaron las capacidades de preprocessamiento del software WEKA. La herramienta *SpreadSubSample* es un tipo de filtro que permite producir una submuestra aleatoria en base al conjunto de datos original con una distribución uniforme [WF05].

Esto se realizó con la finalidad de paliar cualquier influencia de la distribución de las clases a la hora de aplicar los algoritmos de aprendizaje. Como resultado de dicho proceso se obtuvo lo siguiente:

6.3 Distribución de clases

- 215 instancias para el contexto de streaming de video.
- 125 instancias para el contexto de navegación web.

Los gráficos de las distribuciones de valores de clase se pueden observar en las figuras 6.3 y 6.4, para los contextos de streaming de video y navegación web, respectivamente.

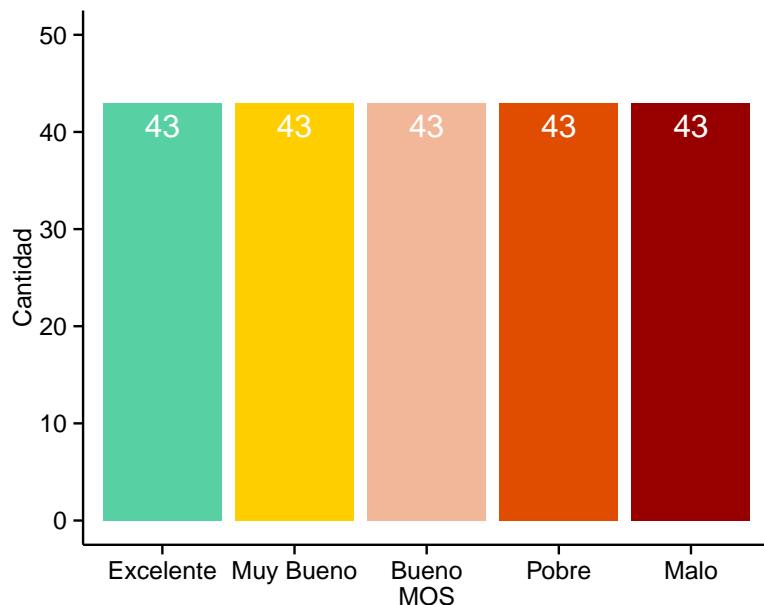


Figura. 6.3 Distribución clases para el conjunto de datos de streaming de video luego del proceso de balanceo.

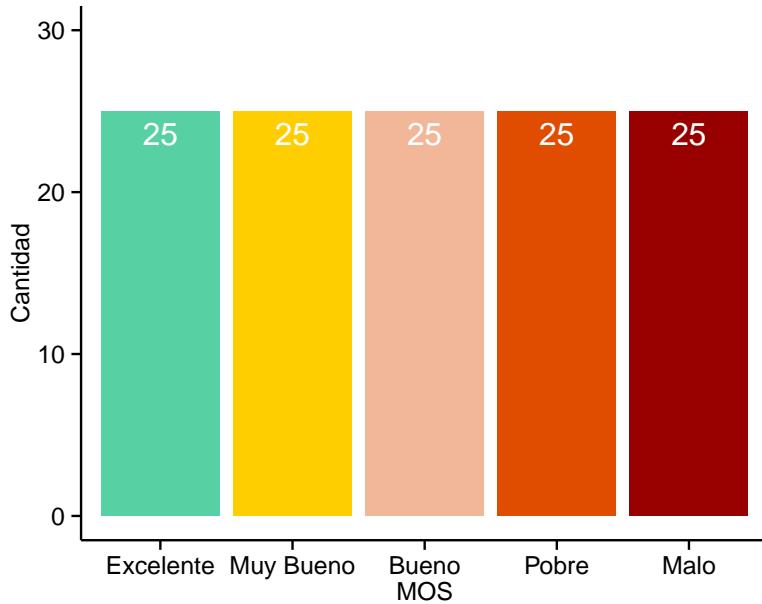


Figura. 6.4 Distribución clases para el conjunto de datos de navegación web luego del proceso de balanceo.

6.4. Experimentación

En este punto, luego de haber definido los conjunto de datos, procedemos al proceso de experimentación y comparación de los algoritmos de clasificación. Con este fin utilizamos una herramienta de aprendizaje automático denominada WEKA [HFH⁺09]. Esta herramienta, hecha en el lenguaje JAVA, fue diseñada para realizar diversas tareas de *Minería de Datos*. WEKA proporciona una colección de algoritmos de aprendizaje automático y además contiene herramientas para el preprocesamiento de datos, clasificación, regresión y visualización de datos [atUoW15]. De este modo, los conjuntos de datos definidos y analizados en la fase anterior fueron convertidos al formato ARFF, utilizado por WEKA. A partir de allí, fueron aplicados los diferentes algoritmos de clasificación para cada conjunto de datos. En el siguiente cuadro se citan los algoritmos utilizados, clasificándolos de acuerdo a las técnicas de aprendizaje automático descriptas en capítulo 4.

6.4 Experimentación

Cuadro. 6.2 Algoritmos de clasificación utilizados en el proceso de experimentación.

Método	Algoritmos	Referencia
Redes neuronales	MultilayerPerceptron RBFNetwork RBFClassifier	[WF05] [WF05] [WF05]
Basados en modelos de regresión	SimpleLogistic Logistic	[LHF05, SFH05] [ICvH92]
Máquinas de soporte vectorial	Sequential minimal optimization (SMO)	[Pla98, KSBM01, HT98]
Clas. Bayesianos	Bayes Ingenuo (Naive Bayes) A1DE A2DE BayesNet RandomForest	[JL95] [WBW05] [WBW05] [WF05] [Bre01]
Árboles de decisión	REPTree J48 RandomTree	[WF05] [Qui93] [WF05]
Clas. basados en instancias	IBk ($k = 1, k = 5, k = 10$)	[AK91]
Tabla de decisiones	Decision Table DTNB	[K ⁺ 95] [HF08]

6.4.1. Configuración del experimento

Con el fin de analizar y comparar los algoritmos de aprendizaje proporcionados por WEKA, hemos hecho uso de las capacidades proporcionadas por este software en cuanto a la experimentación con los algoritmos de aprendizaje. Para ello utilizamos la siguiente configuración:

- Cada uno de los algoritmos fue ejecutado con los parámetros configurados por defecto por WEKA en su versión 3.7, a excepción del algoritmo *IBk*, para el que se utilizaron valores de $k = 1, 5, 10$.
- Debido a la limitada cantidad de datos disponibles, se utilizó la técnica de remuestreo *stratified k-cross validation*. Esto también ayuda durante el proceso de estimación de error.
- Para una mayor precisión, se hizo un total de 10 ejecuciones de *10-cross validation* para cada uno de los algoritmos en los conjuntos de datos citados, de acuerdo a lo concluido en [Bou03].

6.5. Métricas de evaluación

A continuación presentamos las métricas de evaluación aplicables al problema de clasificación de ítems multiclase tratado, de acuerdo a [JS11b], las cuales posteriormente se utilizan para realizar la evaluación del rendimiento de los distintos clasificadores y compararlos entre sí. Primeramente se introduce la definición de *matriz de confusión* debido a que las métricas están basadas en este concepto.

Matriz de confusión

En el campo de la Inteligencia Artificial, una matriz de confusión es una herramienta de visualización que se emplea en el aprendizaje supervisado. La matriz de confusión nos permite observar como el modelo realiza las predicciones [Ste97]. Cada fila de la matriz representa a las instancias que realmente pertenecen a una clase, mientras que cada columna representa el número de predicciones de cada clase. Por lo tanto, los elementos de la diagonal principal muestran el número de clasificaciones correctas hechas para cada clase y los elementos fuera de la diagonal muestran los errores cometidos [Ste97]. En el cuadro 6.3 se presenta una matriz de confusión abstracta que se utilizará posteriormente para notación.

Cuadro. 6.3 Matriz de confusión abstracta.

		Clase Predicha			
		Clase 1	Clase 2	Clase ...	Clase N
Clase Real	Clase 1	f_{11}	f_{12}	...	f_{1N}
	Clase 2	f_{21}	f_{22}	...	f_{2N}
	Clase
	Clase N	f_{N1}	f_{N2}	...	f_{NN}

6.5.1. Porcentaje de aciertos

El *porcentaje de aciertos* es una métrica de evaluación basada en la matriz de confusión. Mide la precisión global del modelo y se emplea generalmente para comparar el desempeño entre diferentes clasificadores. El porcentaje de aciertos de un clasificador está dado por la cantidad de clasificaciones correctas sobre la cantidad de instancias evaluadas [Ste97]. Si f_{ii} de la matriz 6.3 representa el número de clasificaciones correctas para cada clase i , n es el número total de instancias evaluadas y N el número total de clases, entonces la fórmula de porcentaje de aciertos está dada por:

$$\text{porcentaje aciertos} = \frac{\sum_{i=1}^N f_{ii}}{n} * 100 \quad (6.1)$$

6.5.2. Kappa

El *coeficiente kappa* es una medida estadística que ajusta el efecto del azar en la proporción de concordancia esperada. El cálculo de *kappa* se basa en la diferencia entre la cantidad de acuerdos que están realmente presentes (precisión observada PO) en comparación con la cantidad de acuerdos que se espera estén presentes por pura casualidad (precisión esperada PE) [Fle71]. En general se cree que es una medida más robusta que el cálculo del porcentaje de aciertos, ya que tiene en cuenta la cantidad de acuerdos que ocurren por azar. La fórmula de *kappa* esta dada por:

$$\kappa = \frac{PO - PE}{1 - PE} \quad (6.2)$$

donde:

$$PO = \frac{1}{n} \sum_{i=1}^N f_{ii} \quad (6.3)$$

y

$$PE = \sum_{k=1}^N \left[\frac{1}{n^2} \left(\sum_{i=1}^N f_{ik} \times \sum_{j=1}^N f_{kj} \right) \right] \quad (6.4)$$

Así la *precisión observada* (*PO*) es el número de instancias que fueron clasificadas correctamente de toda la matriz de confusión sobre el total de instancias, mientras que la *precisión esperada* (*PE*) está directamente relacionada con el número de instancias de cada clase y el número de instancias que el clasificador concuerda con el valor verdadero de la clase.

Kappa es siempre menor o igual a 1. Un valor de 1 indica un acuerdo perfecto entre la precisión esperada y la precisión observada. En raras ocasiones, *kappa* puede ser negativo lo que indica que no existe un acuerdo efectivo entre la precisión esperada y la observada [Fle71].

6.5.3. Medida-F

La medida-F es el promedio ponderado de las medias armónicas que combinan los valores de la precisión y la exhaustividad de cada clase. Para comprender mejor introduciremos, primeramente, los conceptos de precisión y exhaustividad involucrados en esta métrica [JS11a].

Precisión: la precisión respecto a una clase indica la probabilidad de que una instancia pertenezca realmente a dicha clase dado que fue identificada como tal por el clasificador [JS11a]. La precisión para una clase *i* determinada está dada por:

$$precision_i = \frac{f_{ii}}{\sum_{j=1}^N f_{ji}} \quad (6.5)$$

Exhaustividad: también denominada *Recall* o *True Positive Rate*, se refiere a la capacidad para identificar una clase correctamente. La exhaustividad representa el número de predicciones acertadas de una clase sobre el total de elementos que realmente pertenecen a dicha clase [JS11a]. La exhaustividad para una clase *i* determinada está dada por:

$$exhaustividad_i = \frac{f_{ii}}{\sum_{j=1}^N f_{ij}} \quad (6.6)$$

Una vez definidos los conceptos subyacentes, presentamos la fórmula para calcular

el valor de la medida F para una clase determinada i [Sas07]:

$$medidaF_{\beta}(i) = \frac{(1 + \beta^2)(precision_i * exhaustividad_i)}{(\beta^2 \times precision_i) + exhaustividad_i} \quad (6.7)$$

donde β indica la ponderación otorgada a la exhaustividad y precisión. Un valor de $\beta > 1$ indica una mayor ponderación de la exhaustividad, un valor de $\beta < 1$ corresponde a una mayor ponderación de la precisión, en tanto que, un valor de $\beta = 1$ indica una ponderación equitativa de la precisión y la exhaustividad.

Así, WEKA utiliza un valor $\beta = 1$, por tanto se tiene:

$$medidaF_1(i) = \frac{2(precision_i \times exhaustividad_i)}{precision_i + exhaustividad_i} \quad (6.8)$$

A esta medida se la denomina comúnmente F1 (ya que $\beta = 1$). Esta medida es muy utilizada en los problemas de clasificación debido a que la precisión y la exhaustividad describen propiedades distintas del clasificador. Entonces, la medida F proporciona una cuantificación general que permite evaluar la calidad del clasificador. La medida F se puede globalizar para todas las clases por medio de la siguiente fórmula:

$$medidaF_{global} = \frac{1}{n} \left[\sum_{k=1}^N \left(medidaF(k) \times \sum_{j=1}^N f_{kj} \right) \right] \quad (6.9)$$

donde la $medidaF(k)$ representa la medida F para la clase k . Esta formulación global representa el promedio ponderado (macro) de las medidas F de todas las clases [SL09].

6.6. Análisis de resultados

En esta sección se describen los resultados obtenidos luego de la ejecución de los algoritmos de aprendizaje sobre los conjuntos de datos definidos anteriormente. Para la comparación de los algoritmos haremos uso de las métricas de evaluación citadas en la sección anterior. Cada contexto de uso será analizado por separado en un apartado diferente. En cada uno de los apartados, se consideran como datos de salida los resultados del proceso de 10 iteraciones completas de *10-fold cross validation*. En cada cuadro se puede visualizar tanto el promedio como la desviación estándar asociada a las salidas de la ejecución de cada algoritmo, resaltando aquellos con el mejor y el peor rendimiento. Para facilitar la comprensión de los resultados se hará uso de diagramas de cajas, las cajas se encuentran ordenadas de mayor a menor en cuanto al rendimiento

promedio del algoritmo de aprendizaje para la métrica analizada. Información acerca de los diagramas de cajas y su interpretación se puede ver en el Apéndice D.

6.6.1. Video streaming

En este apartado se presentan los resultado de la ejecución de los algoritmos de aprendizaje sobre los conjuntos de datos de video streaming.

Porcentaje de aciertos

En el cuadro 6.4 se muestran los resultados promedio (junto con las desviaciones estándar) de la ejecución de los algoritmos de aprendizaje con respecto al porcentaje de clases correctamente etiquetadas, tanto para el conjunto de datos desbalanceado como el balanceado. Acorde al cuadro, para el conjunto de datos desbalanceado, los clasificadores bayesianos A2DE y A1DE en promedio obtienen los mejores resultados, seguido por el algoritmo de árboles ensamblados RandomForest y el algoritmo SimpleLogistic. Por otra parte, en cuanto a la variabilidad, se observa en el diagrama de la figura 6.5 que de entre estos cuatro algoritmos que lograron un mayor desempeño en esta métrica, el algoritmo A2DE fue el que presentó una menor variación durante las 100 ejecuciones.

También se observa que los clasificadores basados en instancias IBk (k-Nearest Neighbor), con diferentes valores de k , junto con el algoritmo RBFClassifier fueron los que tuvieron, en promedio, menor desempeño.

Por otra parte, de acuerdo al cuadro 6.4 se observa que, para el conjunto de datos balanceado, el algoritmo RandomForest obtuvo el mejor promedio de porcentaje de aciertos, seguido por los clasificadores NaiveBayes, SimpleLogistic y A2DE. En cambio, como se observa en el diagrama de cajas de la figura 6.6, el algoritmo SimpleLogistic presenta una menor variabilidad. Luego le siguen los algoritmos A2DE, A1DE, RandomForest y Naive Bayes, en ese orden. Al igual que en el conjunto de datos desbalanceado, los clasificadores basados en instancias IBk y el algoritmo RBFClassifier fueron, en promedio, los que tuvieron menor desempeño.

6.6 Análisis de resultados

Cuadro. 6.4 Resultados de la ejecución de los algoritmos de aprendizaje con respecto a la métrica de porcentaje de aciertos. Se muestra en **negrita** los algoritmos que en promedio obtuvieron el mejor desempeño.

Algoritmos	Porcentaje promedio de aciertos	
	Desbalanceado	Balanceado
A2DE	79,24368 ± 6,668772	71,35281 ± 9,10097
A1DE	78,56552 ± 7,035416	71,17316 ± 9,238917
Random Forest	78,15747 ± 6,255714	75,00866 ± 9,152358
SimpleLogistic	76,85402 ± 6,535044	72,09524 ± 9,221911
BayesNet	76,60575 ± 7,301375	70,32468 ± 9,23754
DTNB	75,51839 ± 7,593249	70,20563 ± 9,723088
Naive Bayes	74,63103 ± 7,789159	72,91558 ± 8,583166
RBF Network	73,56552 ± 6,934105	71,1645 ± 8,880594
J48	73,45747 ± 7,313177	69,08442 ± 9,299147
REPTree	72,91954 ± 6,723416	67,79437 ± 9,727463
MultiLayer Perceptron	70,32069 ± 7,843912	66,93939 ± 8,569811
SMO	69,76782 ± 6,602116	62,46753 ± 10,33105
Decision Table	66,06437 ± 8,419521	65,48052 ± 10,23489
RandomTree	65,27126 ± 7,927795	62,75758 ± 10,75262
RBF Classifier	60,64828 ± 8,802131	56,01732 ± 8,64811
IBk ($k = 1$)	60,02529 ± 8,282496	60,53896 ± 8,461615
IBk ($k = 5$)	58,66782 ± 8,529	51,06277 ± 10,78967
IBk ($k = 10$)	56,61954 ± 8,223143	52,37229 ± 9,343614

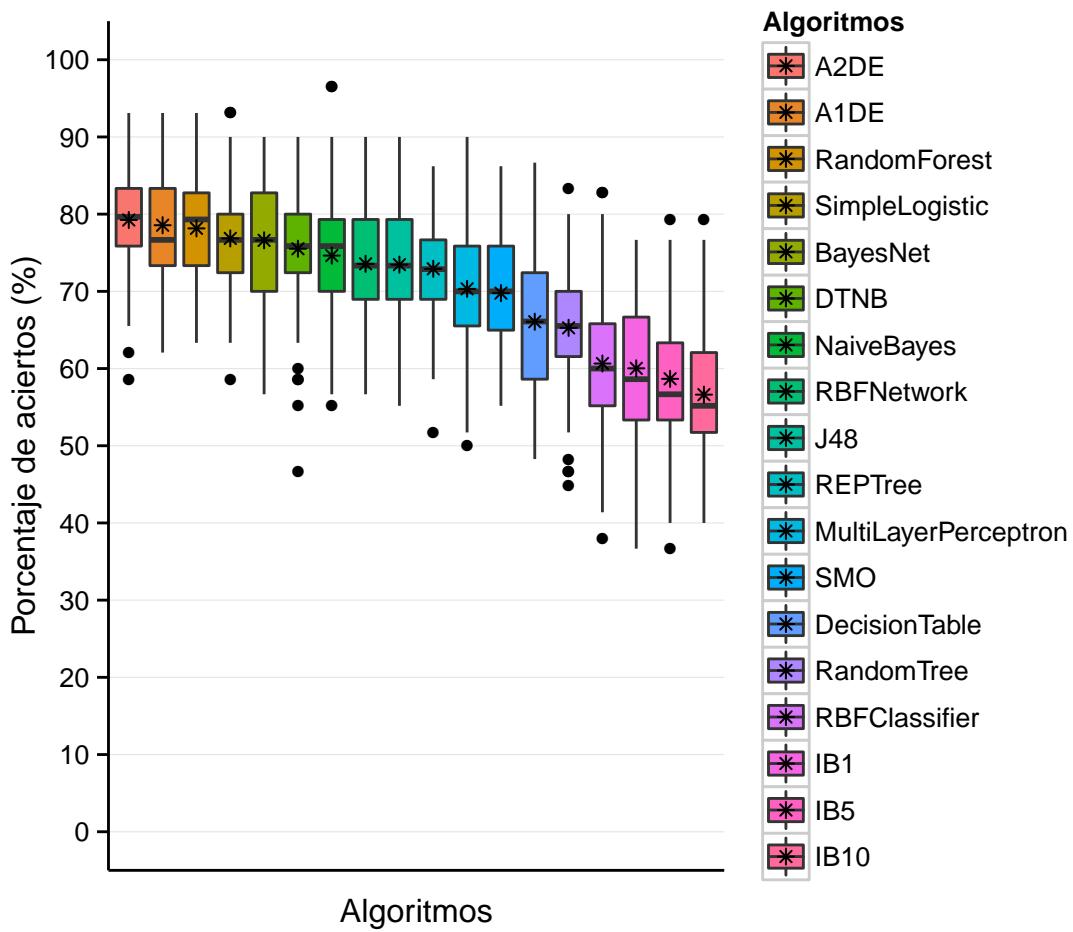


Figura. 6.5 Diagrama de caja con respecto al porcentaje de aciertos considerando las datos de video streaming desbalanceado.

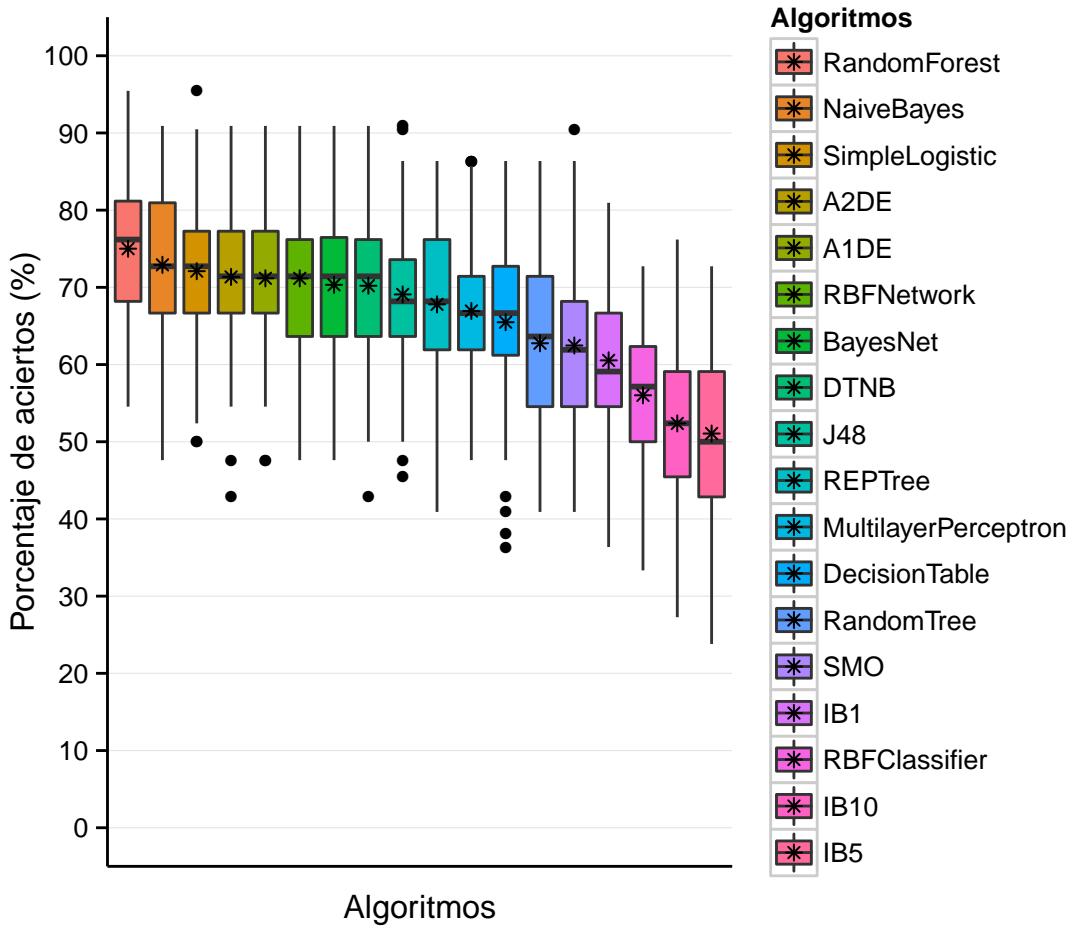


Figura. 6.6 Diagrama de caja con respecto al porcentaje de aciertos considerando las datos de video streaming balanceado.

Kappa

En el cuadro 6.5 se muestran los resultados de la ejecución de los algoritmos de aprendizaje con respecto a la métrica kappa. Los resultados se muestran gráficamente, tanto para el conjunto de datos desbalanceado como el balanceado, en las figuras 6.7 y 6.8.

Para esta métrica, nuevamente se observa que los clasificadores A2DE, A1DE, RandomForest y SimpleLogistic, en promedio, obtienen los mejores desempeños. En cuanto a la variabilidad, el algoritmo A2DE presenta una menor variabilidad, la cual es muy similar a la que presenta el algoritmo RandomForest (ver figura 6.7). Al igual que con la métrica porcentaje de aciertos, los clasificadores basados en instancias IBk y el clasificador RBFClassifier obtuvieron, en promedio, los peores desempeños.

6.6 Análisis de resultados

Para el conjunto de datos balanceado, de acuerdo al cuadro 6.5, el algoritmo RandomForest, en promedio, posee un desempeño ligeramente superior a otros algoritmos como NaiveBayes, SimpleLogistic, A2DE, o el A1DE. En cuanto a la variabilidad, como se observa en el diagrama de la figura 6.8, los algoritmos Random Forest y NaiveBayes se comportan de manera muy similar. Por otra parte, los clasificadores basados en instancias IBk, RBFCClassifier, RamdomTree y SMO son los que presentan, en promedio, los peores desempeños para el conjunto de datos balanceado con respecto a esta métrica.

6.6 Análisis de resultados

Cuadro. 6.5 Resultados de la ejecución de los algoritmos de aprendizaje con respecto a la métrica Kappa.

Algoritmos	Kappa	
	Desbalanceado	Balanceado
A2DE	0,7386051 ± 0,08386978	0,6414048 ± 0,1137775
A1DE	0,7301379 ± 0,08845891	0,6391935 ± 0,1154213
Random Forest	0,7242175 ± 0,07906886	0,6871424 ± 0,1144409
SimpleLogistic	0,7081632 ± 0,0823815	0,650652 ± 0,1153682
BayesNet	0,7054384 ± 0,09179641	0,6286006 ± 0,1154282
DTNB	0,6916571 ± 0,09550224	0,6271934 ± 0,1213563
NaiveBayes	0,6793643 ± 0,09818825	0,660966 ± 0,106999
RBFNetwork	0,6659207 ± 0,08794005	0,6388524 ± 0,1114165
J48	0,6652991 ± 0,09212218	0,6126583 ± 0,1165797
REPTree	0,6588382 ± 0,08420378	0,5968874 ± 0,1216925
MultiLayer Perceptron	0,6258377 ± 0,09874556	0,5861803 ± 0,1070324
SMO	0,6173024 ± 0,08403359	0,5305662 ± 0,1287563
Decision Table	0,5707757 ± 0,1067196	0,5679664 ± 0,1281385
RandomTree	0,5630241 ± 0,09961472	0,5334655 ± 0,134454
RBFClassifier	0,502433 ± 0,1119337	0,4502876 ± 0,1077432
IBk ($k = 1$)	0,4957066 ± 0,1049757	0,5059577 ± 0,1057246
IBk ($k = 5$)	0,4775146 ± 0,1084813	0,3875394 ± 0,1342291
IBk ($k = 10$)	0,4499093 ± 0,1040601	0,4042986 ± 0,1165948

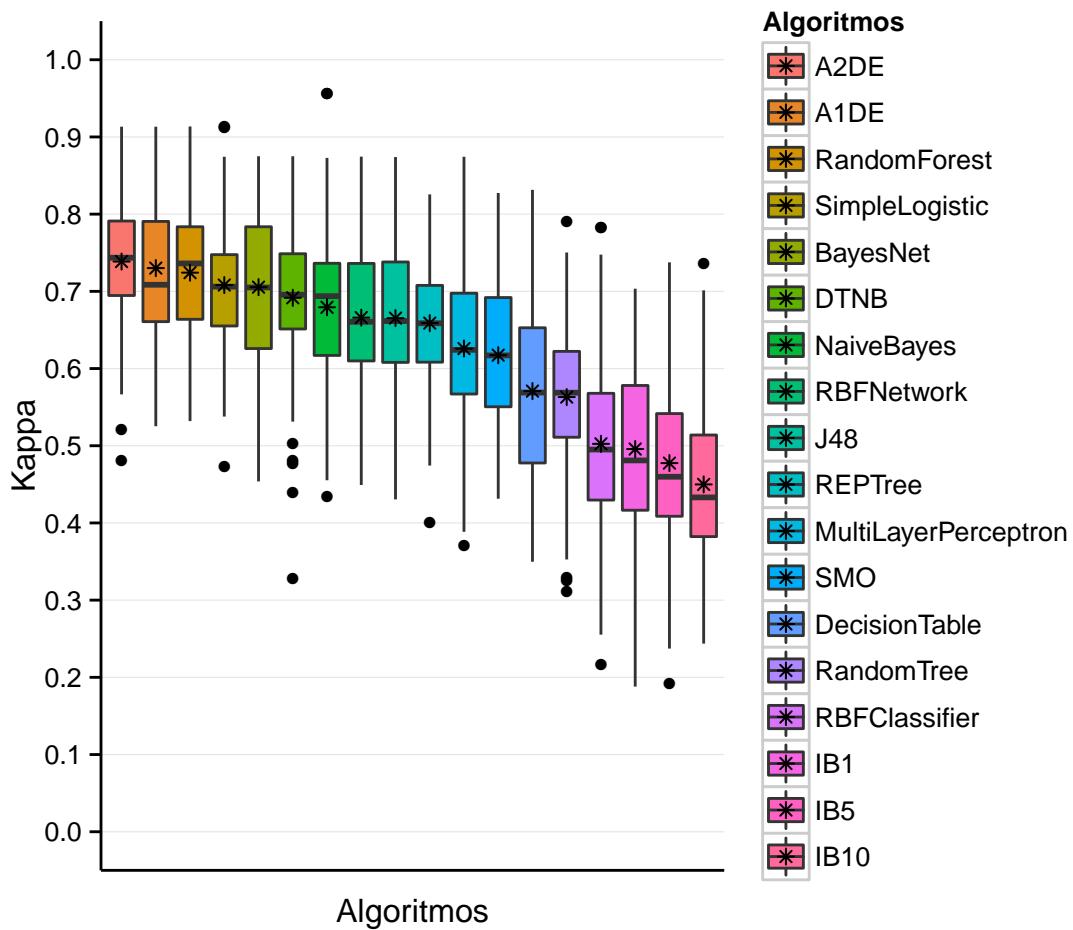


Figura. 6.7 Diagrama de caja con respecto a la métrica kappa considerando los datos de video streaming desbalanceado.

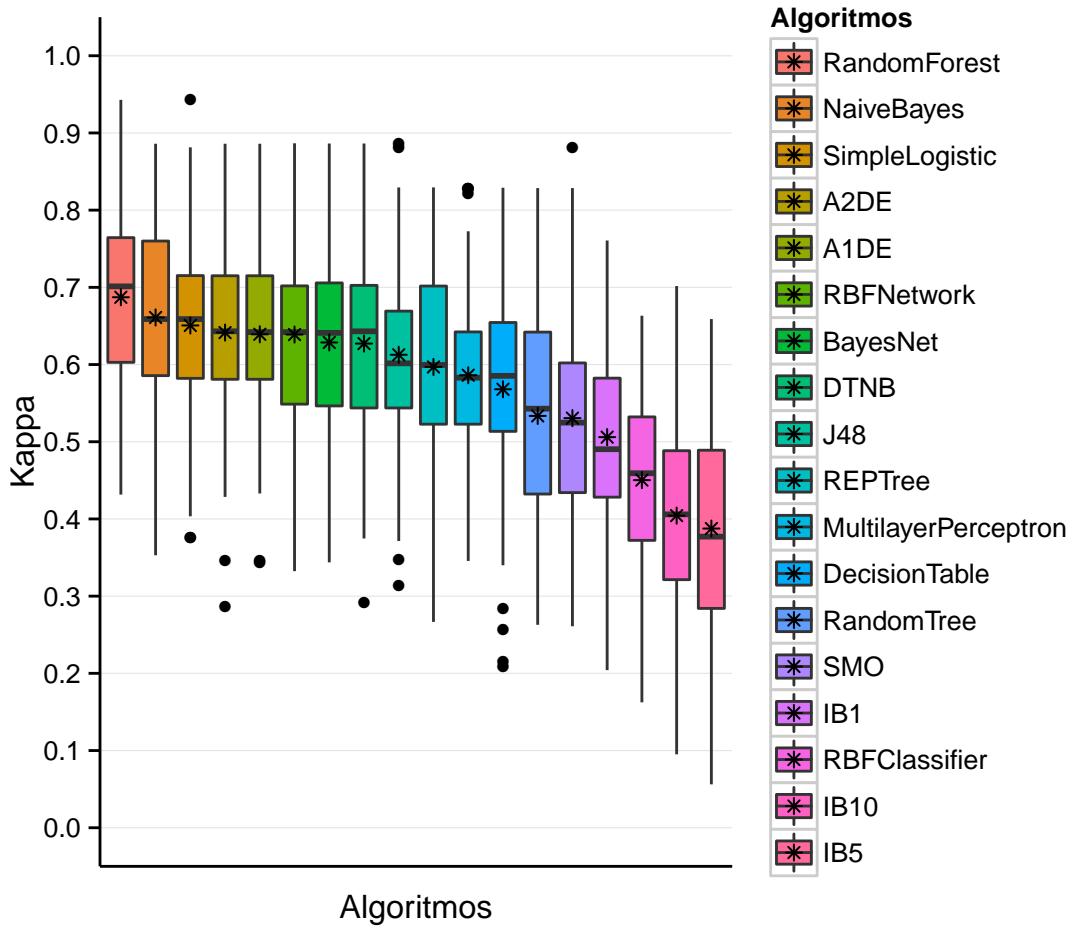


Figura. 6.8 Diagrama de cajas con respecto a la métrica kappa considerando los datos de video streaming balanceado.

Medida-F

En el cuadro 6.6 se muestran los resultados de la ejecución de los algoritmos de aprendizaje con respecto a la métrica medida F. Nuevamente, los resultados se muestran gráficamente, tanto para el conjunto de datos desbalanceado como el balanceado, en las figuras 6.9 y 6.10.

Para esta métrica, al igual que con las métricas anteriores, el algoritmo A2DE obtiene el mejor desempeño, para el conjunto de datos desbalanceado. A este algoritmo le siguen los algoritmos A1DE, RandomForest y SimpleLogistic. De acuerdo al diagrama de cajas de la figura 6.9, se puede observar que el algoritmo A2DE presenta resultados más acotados en la media, lo que da como resultado una menor variabilidad. Un comportamiento bastante similar se puede observar para el algoritmo RandomForest,

6.6 Análisis de resultados

seguido por el algoritmo A1DE con una mayor variabilidad.

En cuanto al conjunto de datos balanceado, en promedio, el algoritmo RandomForest obtiene el mejor resultado. A éste le siguen los algoritmos NaiveBayes, SimpleLogistic y A2DE. De acuerdo a la figura 6.10 se puede observar que los citados algoritmos presentan comportamientos de variabilidad similares, sin embargo, el algoritmo RandomForest se desempeña mejor de acuerdo a los valores mínimos y máximos que presenta. Nuevamente se constata que los algoritmos que tuvieron un menor desempeño tanto para el conjunto de datos desbalanceado como el balanceado son: IB10, IB5, RBFClassifier, IB1 y RandomTree.

Cuadro. 6.6 Resultados de la ejecución de los algoritmos de aprendizaje con respecto a la métrica medida-F.

Algoritmos	Medida F	
	Desbalanceado	Balanceado
A2DE	$0,7892146 \pm 0,06978092$	$0,7090881 \pm 0,09261402$
A1DE	$0,7834329 \pm 0,07216741$	$0,7064382 \pm 0,09397792$
Random Forest	$0,776391 \pm 0,06447893$	$0,7470776 \pm 0,09411592$
SimpleLogistic	$0,7647982 \pm 0,06714861$	$0,716422 \pm 0,09323135$
BayesNet	$0,7637954 \pm 0,07462658$	$0,6978782 \pm 0,09468497$
DTNB	$0,7519969 \pm 0,07750481$	$0,6960121 \pm 0,1001418$
Naive Bayes	$0,7398493 \pm 0,07997514$	$0,7216307 \pm 0,09121794$
RBF Network	$0,7296717 \pm 0,07185618$	$0,7056985 \pm 0,091643$
J48	$0,7270175 \pm 0,077268$	$0,6849132 \pm 0,09539091$
REPTree	$0,7204991 \pm 0,0721369$	$0,6647741 \pm 0,1055245$
MultiLayer Perceptron	$0,6956533 \pm 0,08004716$	$0,6591519 \pm 0,09059375$
SMO	$0,6883247 \pm 0,06930298$	$0,613908 \pm 0,1055583$
Decision Table	$0,6475877 \pm 0,08786936$	$0,6338589 \pm 0,1125715$
RandomTree	$0,6464814 \pm 0,07888933$	$0,6167155 \pm 0,113549$
IBk ($k = 1$)	$0,5920726 \pm 0,08764257$	$0,5988458 \pm 0,08776478$
RBF Classifier	$0,5821305 \pm 0,09562511$	$0,5170022 \pm 0,09337735$
IBk ($k = 5$)	$0,5772204 \pm 0,08885218$	$0,4970216 \pm 0,110408$
IBk ($k = 10$)	$0,5504346 \pm 0,0855672$	$0,5097333 \pm 0,09681821$

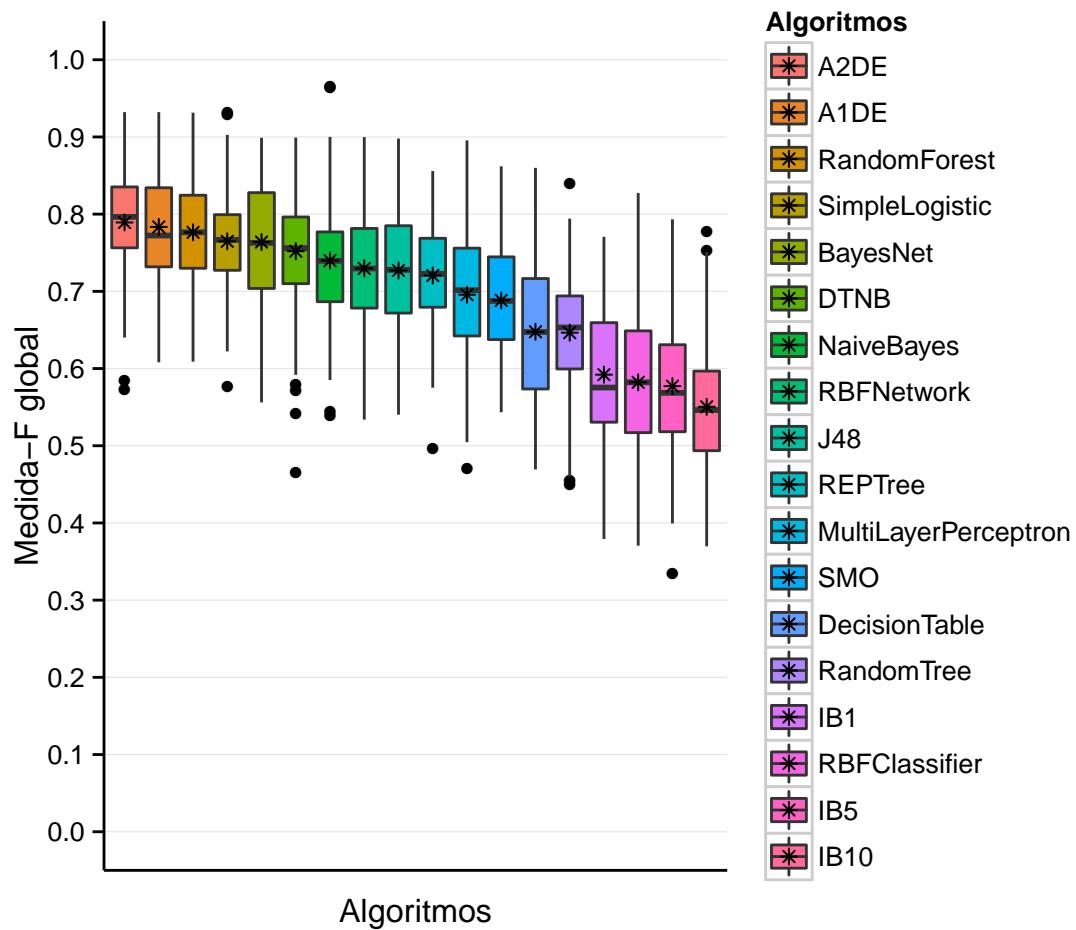


Figura. 6.9 Diagrama de cajas con respecto a la métrica medida-F considerando considerando los datos de video streaming desbalanceado.

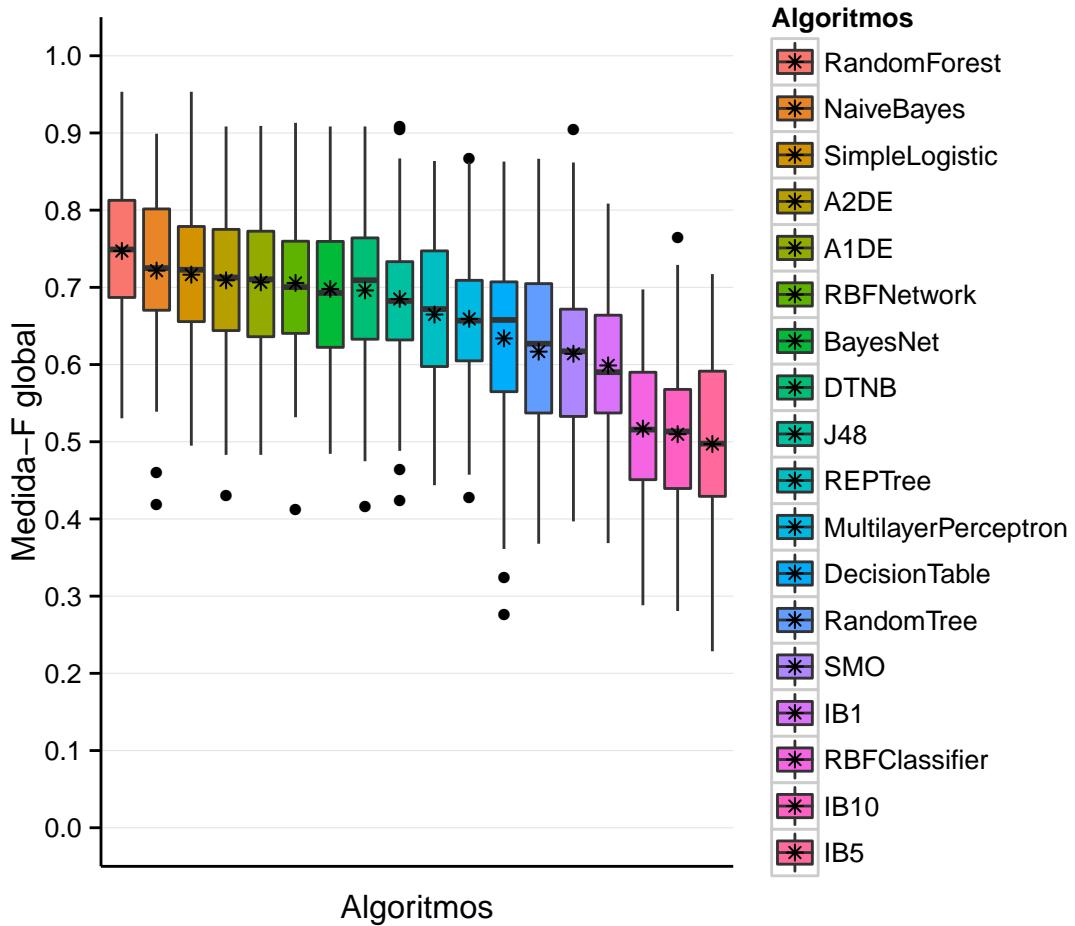


Figura. 6.10 Diagrama de cajas con respecto a la métrica medida-F considerando los datos de video streaming balanceado.

6.6.2. Navegación web

En este apartado se presentan los resultados de la ejecución de los algoritmos de aprendizaje sobre los conjuntos de datos de navegación web con respecto a las métricas analizadas. Del mismo modo que el análisis sobre los conjuntos de datos de video streaming, se toman las 100 salidas de 10 iteraciones de *10-fold cross validation*.

Porcentaje de aciertos

En el cuadro 6.7 se muestran los resultados de la ejecución de los algoritmos de aprendizaje con respecto al porcentaje de clases correctamente etiquetadas tanto para el conjunto de datos desbalanceado como balanceado. Los diagramas de cajas para

6.6 Análisis de resultados

los conjuntos desbalanceado y balanceado se presentan en las figuras 6.11 y 6.12, respectivamente.

Como se observa en el cuadro 6.7, el algoritmo RandomForest obtiene en promedio un resultado de 82 % de clases correctamente etiquetadas para el conjunto de datos desbalanceado, un resultado ligeramente superior al obtenido por sus competidores: DTNB, A2DE y A1DE. En el diagrama de cajas 6.11 se puede observar que, si bien el algoritmo RandomForest presenta resultados ligeramente mejores, sus resultados se encuentran más dispersos de la media que los algoritmos DTNB, A2DE y A1DE.

Para el caso del conjunto de datos balanceado, el ranking de los cuatro algoritmos con mejor desempeño es: RandomForest, DTNB, A2DE, A1DE. Sin embargo, al igual que en el caso anterior, el algoritmo RandomForest presenta una mayor fluctuación que los demás algoritmos. Así, los algoritmos DTNB, A1DE y A2DE son ligeramente más estables que el RandomForest (presentan una mayor concentración de resultados cercanos a la media). Al igual que en los conjuntos de datos de video streaming, los algoritmos IBk y el algoritmo RBFClassifier presentan los peores resultados, tanto para el conjunto balanceado como el desbalanceado. En ambos conjuntos, el clasificador basado en instancias IBk ($k = 10$) obtuvo el peor desempeño, con un porcentaje de aciertos del 58,66 % para el caso desbalanceado y 45,9 % para el caso balanceado.

6.6 Análisis de resultados

Cuadro. 6.7 Resultados de la ejecución de los algoritmos de aprendizaje con respecto a la métrica de porcentaje de aciertos.

Algoritmos	Porcentaje promedio de aciertos	
	Desbalanceado	Balanceado
Random Forest	$82 \pm 6,996459$	$82,30128 \pm 11,5449$
DTNB	$81,15625 \pm 7,154486$	$81,29487 \pm 10,35964$
A2DE	$80,78125 \pm 6,563702$	$80,32051 \pm 9,778891$
A1DE	$80,46875 \pm 6,653262$	$80,32051 \pm 9,950143$
BayesNet	$80,25 \pm 6,556259$	$80,23718 \pm 10,26291$
SimpleLogistic	$79,6875 \pm 7,438988$	$68,44231 \pm 12,55086$
J48	$79,375 \pm 7,472065$	$80,3141 \pm 10,92361$
Decision Table	$78,875 \pm 7,578503$	$76,69231 \pm 11,43458$
REPTree	$77,03125 \pm 7,761831$	$75,26923 \pm 12,36069$
Naive Bayes	$74,78125 \pm 6,970115$	$68,73077 \pm 12,05001$
MultiLayer Perceptron	$74,5625 \pm 7,217152$	$65,47436 \pm 12,85156$
RBF Network	$73,875 \pm 7,002097$	$71,0641 \pm 12,60865$
SMO	$68,09375 \pm 7,528552$	$65,3141 \pm 12,06836$
RandomTree	$67,40625 \pm 8,035087$	$66,91667 \pm 13,70368$
RBF Classifier	$62,53125 \pm 7,622306$	$51,14103 \pm 12,29382$
IBk ($k = 1$)	$62,25 \pm 8,186201$	$55,60897 \pm 13,91453$
IBk ($k = 5$)	$59,65625 \pm 7,024249$	$52,58974 \pm 12,60698$
IBk ($k = 10$)	$58,65625 \pm 7,980887$	$45,89744 \pm 13,64057$

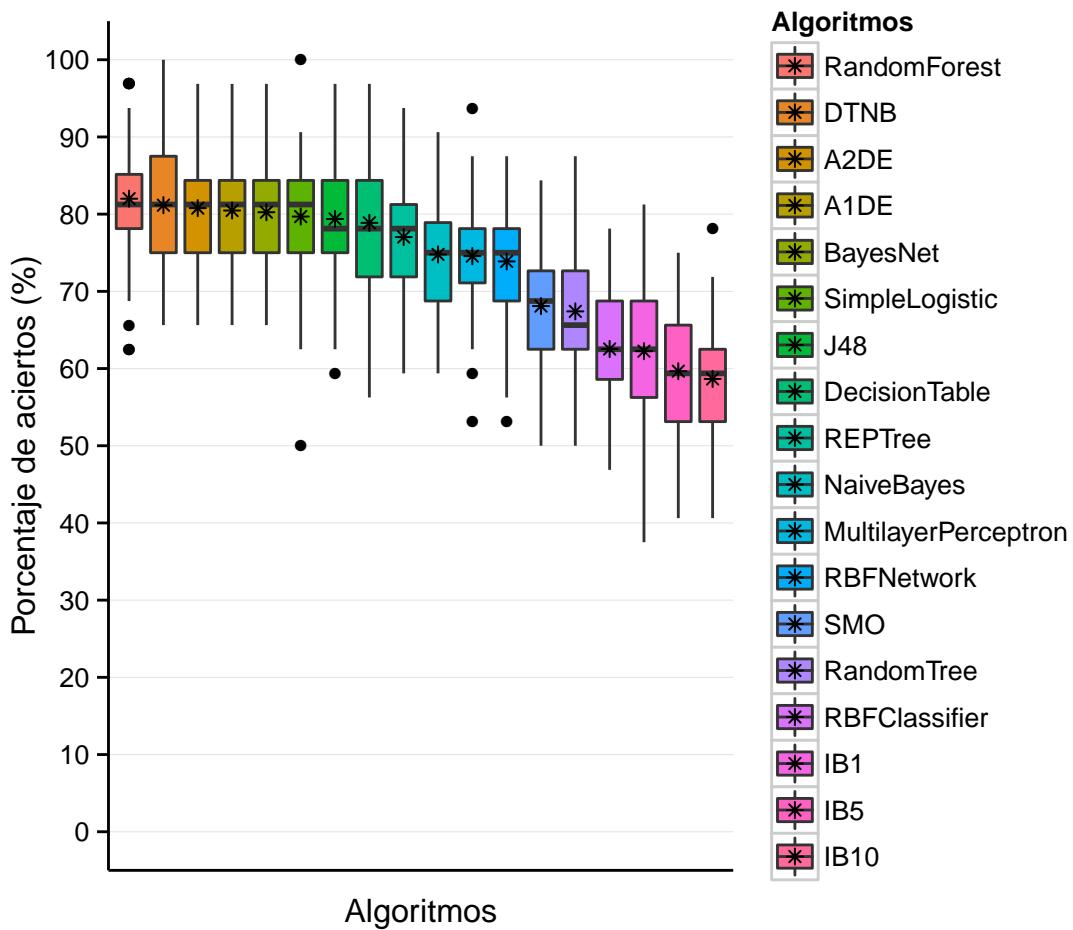


Figura. 6.11 Diagrama de cajas con respecto al porcentaje de aciertos considerando las datos de navegación web desbalanceado.

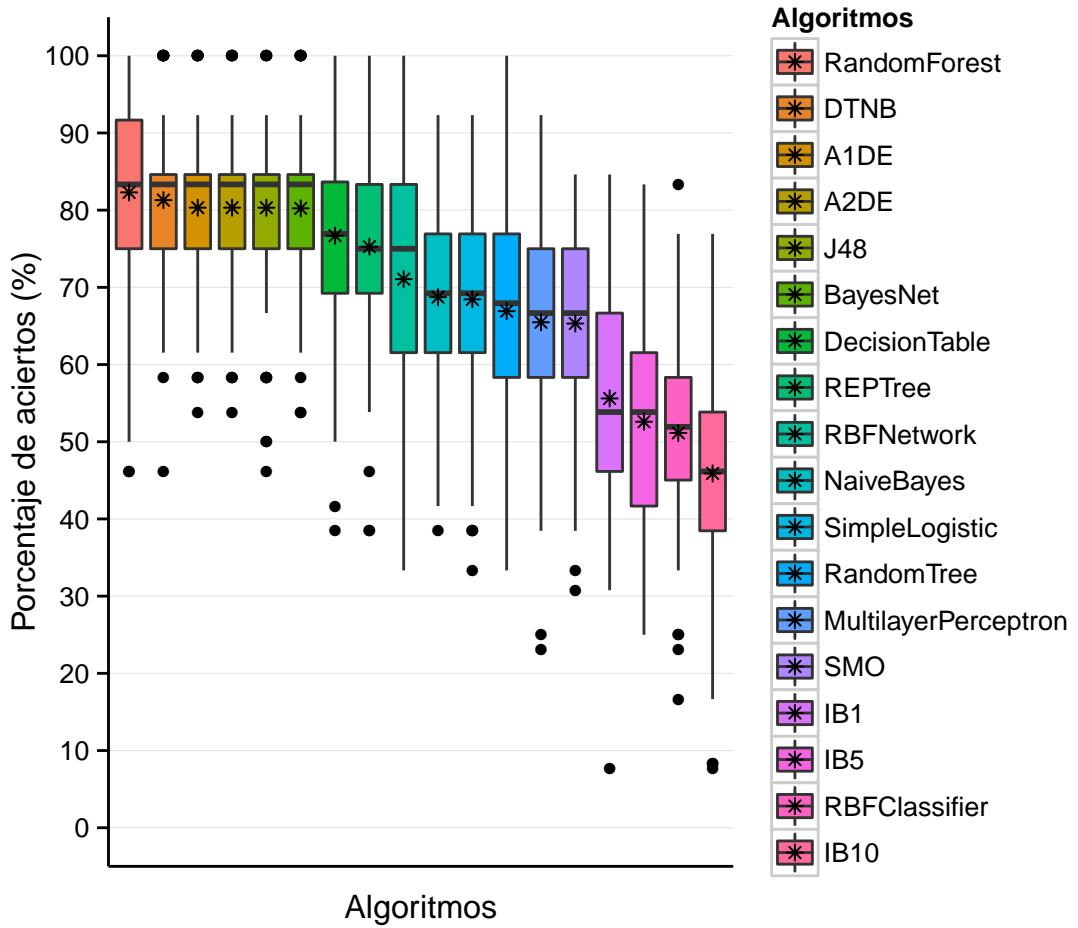


Figura. 6.12 Diagrama de cajas con respecto al porcentaje de aciertos considerando las datos de navegación web balanceado.

Kappa

En el cuadro 6.8 se muestran los resultados de la ejecución de los algoritmos de aprendizaje con respecto a la métrica Kappa para los conjuntos de datos desbalanceado y balanceado. En las figuras 6.13 y 6.14 se muestran los diagramas de cajas correspondientes a los conjuntos desbalanceado y balanceado, respectivamente.

De acuerdo al cuadro, para el conjunto desbalanceado, los algoritmos que presentan los mejores resultados en promedio son: RandomForest, DTNB, A2DE y A1DE. Contrario a lo observado en la métrica anterior, de acuerdo al diagrama de la figura 6.13, el algoritmo RandomForest presenta una mayor concentración de resultados en la media, lo que se traduce en una menor variabilidad y mayor estabilidad con respecto a sus competidores.

6.6 Análisis de resultados

Para el conjunto balanceado, la lista de los cuatro mejores algoritmos se mantiene invariable. Sin embargo, a diferencia del caso desbalanceado, aquí el algoritmo RandomForest presenta una mayor variabilidad que los otros algoritmos (DTNB, A1DE y A2DE). Nuevamente se cumple que, tanto para el conjunto balanceado como el desbalanceado, los algoritmos IBk, RBFClassifier y SMO presentan los peores resultados y una alta variabilidad en sus resultados.

Cuadro. 6.8 Resultados de la ejecución de los algoritmos de aprendizaje con respecto a la métrica Kappa.

Algoritmos	Kappa	
	Desbalanceado	Balanceado
Random Forest	0,7652323 ± 0,0920652	0,7771394 ± 0,1448767
DTNB	0,755059 ± 0,09316022	0,7645941 ± 0,1304743
A2DE	0,7503178 ± 0,08553355	0,7523364 ± 0,1228252
A1DE	0,7463893 ± 0,08646823	0,7523159 ± 0,1251248
BayesNet	0,7436741 ± 0,08502292	0,751096 ± 0,1293962
SimpleLogistic	0,7375967 ± 0,09589194	0,6037474 ± 0,1578558
J48	0,7316595 ± 0,09754544	0,7520462 ± 0,1375
Decision Table	0,7245142 ± 0,0991568	0,706651 ± 0,1439699
REPTree	0,7002179 ± 0,1016827	0,6897992 ± 0,1541097
Naive Bayes	0,6748784 ± 0,08971387	0,607625 ± 0,1498298
MultiLayer Perceptron	0,6704692 ± 0,09320141	0,5668827 ± 0,1603247
RBF Network	0,6622343 ± 0,09008978	0,6364396 ± 0,1583493
SMO	0,5856046 ± 0,09745924	0,5650792 ± 0,1499384
RandomTree	0,5781974 ± 0,1043391	0,5842519 ± 0,1719054
IBk ($k = 1$)	0,5094427 ± 0,1061421	0,4446043 ± 0,1727507
RBF Classifier	0,5084932 ± 0,1002811	0,3903624 ± 0,1499987
IBk ($k = 5$)	0,4708681 ± 0,09069501	0,4070752 ± 0,1550943
IBk ($k = 10$)	0,4544149 ± 0,1041418	0,3244367 ± 0,16888

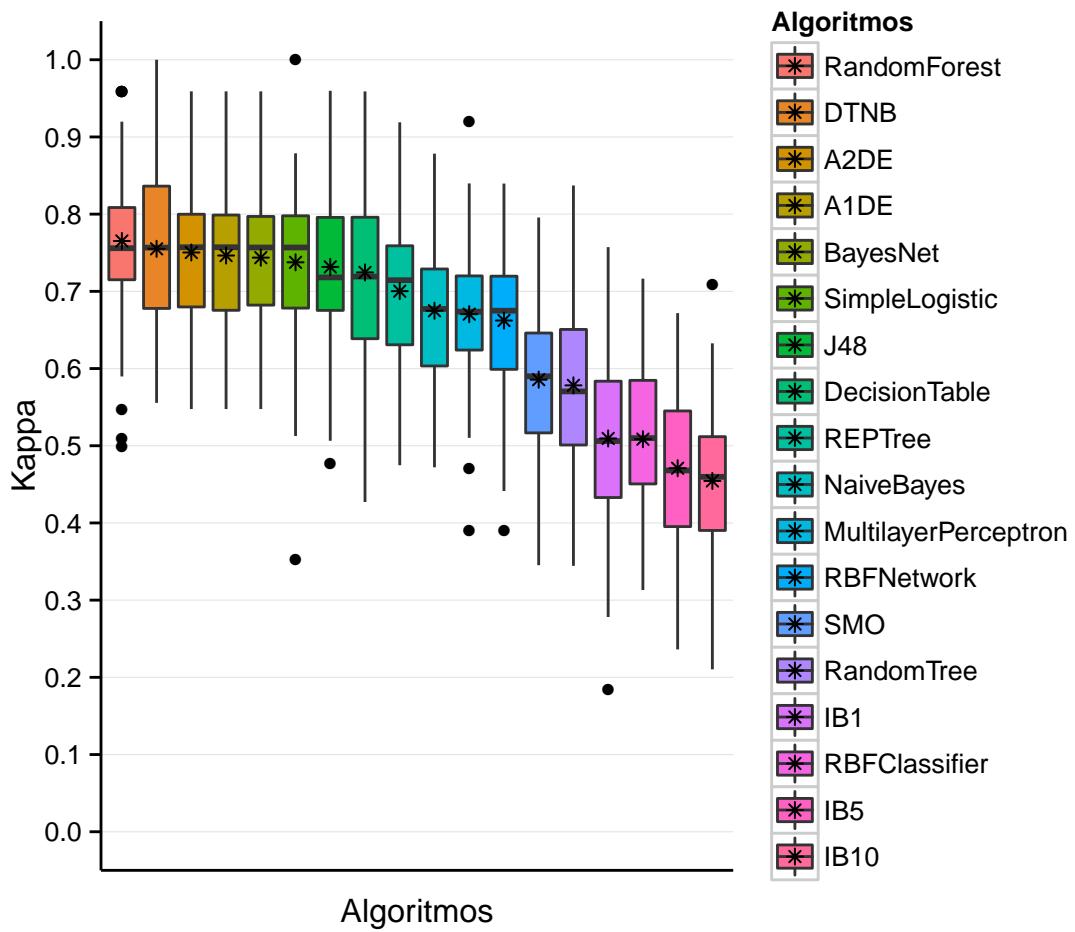


Figura. 6.13 Diagrama de cajas con respecto a la métrica Kappa considerando los datos de navegación web desbalanceado.

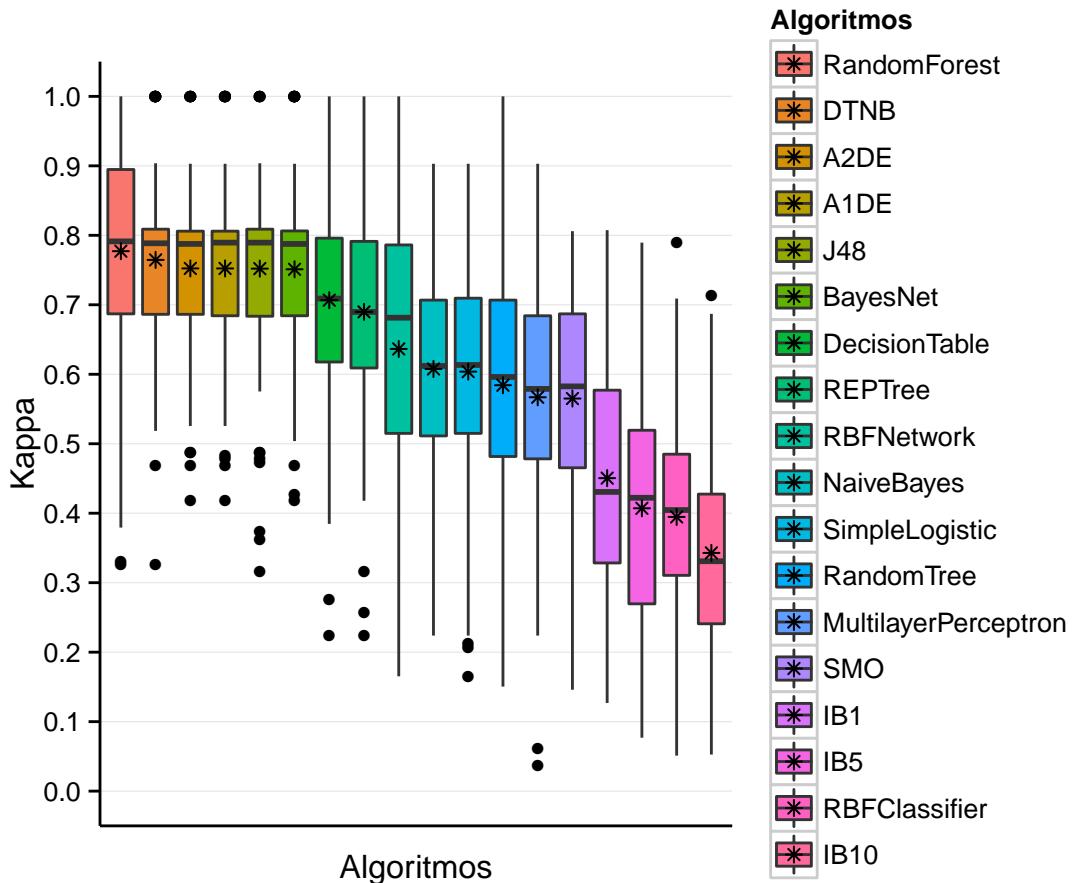


Figura. 6.14 Diagrama de cajas con respecto a la métrica Kappa considerando los datos de navegación web balanceado.

Medida-F

Por último, para la métrica medida F, en el cuadro 6.9 se muestran los resultados de la ejecución de los algoritmos de aprendizaje sobre los conjuntos de datos desbalanceado y balanceado. Las figuras 6.15 y 6.16 muestran los diagramas de caja correspondientes a los conjuntos de datos desbalanceado y balanceado, respectivamente.

Como se observa en el cuadro 6.9, los algoritmos que presentan los mejores resultados, en promedio, para el conjunto desbalanceado son: RandomForest, DTNB, A2DE y A1DE. De acuerdo al diagrama de cajas de la figura 6.15 se puede notar que los algoritmos citados anteriormente, a excepción del DTNB, presentan un comportamiento similar en cuanto a la fluctuación en los resultados, mostrando solo ligeros cambios en

6.6 Análisis de resultados

los promedios de los mismos.

En cuanto al conjunto balanceado, el orden los algoritmos con mejores resultados varía ligeramente: nuevamente RandomForest y DTNB se mantienen igual que en el caso desbalanceado, a ellos le siguen los algoritmos J48, A1DE y A2DE, en ese orden. Por otra parte, de acuerdo a la figura 6.16, se puede observar que si bien el RandomForest en promedio obtiene el mejor resultado, los algoritmos DTNB, J48, A1DE y A2DE tienen una menor fluctuación que el algoritmo RandomForest. Para ambos conjuntos de datos, la lista de algoritmos con resultados inferiores queda invariable (IB10, RFBClassifier, IB5, IB1 y SMO).

Cuadro. 6.9 Resultados de la ejecución de los algoritmos de aprendizaje con respecto a la métrica Medida F.

Algoritmos	Medida F	
	Desbalanceado	Balanceado
Random Forest	$0,8145624 \pm 0,07400057$	$0,8119257 \pm 0,1246064$
DTNB	$0,8079323 \pm 0,07357631$	$0,8028024 \pm 0,110017$
A2DE	$0,8037905 \pm 0,06777179$	$0,7916021 \pm 0,1060364$
A1DE	$0,8005344 \pm 0,06857475$	$0,7916465 \pm 0,1076405$
BayesNet	$0,7981346 \pm 0,06766835$	$0,7898452 \pm 0,1101281$
SimpleLogistic	$0,7947646 \pm 0,07570453$	$0,67012 \pm 0,1324923$
J48	$0,7887189 \pm 0,07881335$	$0,7929579 \pm 0,1167423$
Decision Table	$0,7822092 \pm 0,07954342$	$0,7501464 \pm 0,1268684$
REPTree	$0,7637399 \pm 0,08059436$	$0,732563 \pm 0,1381539$
Naive Bayes	$0,7439162 \pm 0,0720421$	$0,6640649 \pm 0,1308847$
MultiLayer Perceptron	$0,7402861 \pm 0,07474512$	$0,6334158 \pm 0,1349935$
RBF Network	$0,7336891 \pm 0,07156549$	$0,6974332 \pm 0,1310725$
RandomTree	$0,6673282 \pm 0,0841034$	$0,6461314 \pm 0,1463268$
SMO	$0,6531225 \pm 0,07987196$	$0,6233623 \pm 0,1335375$
IBk ($k = 1$)	$0,6151805 \pm 0,08562548$	$0,5382912 \pm 0,1407996$
RBF Classifier	$0,5849163 \pm 0,08216077$	$0,4402508 \pm 0,1305872$
IBk ($k = 5$)	$0,571762 \pm 0,07153015$	$0,4999609 \pm 0,136962$
IBk ($k = 10$)	$0,5613417 \pm 0,08047951$	$0,4189646 \pm 0,1331685$

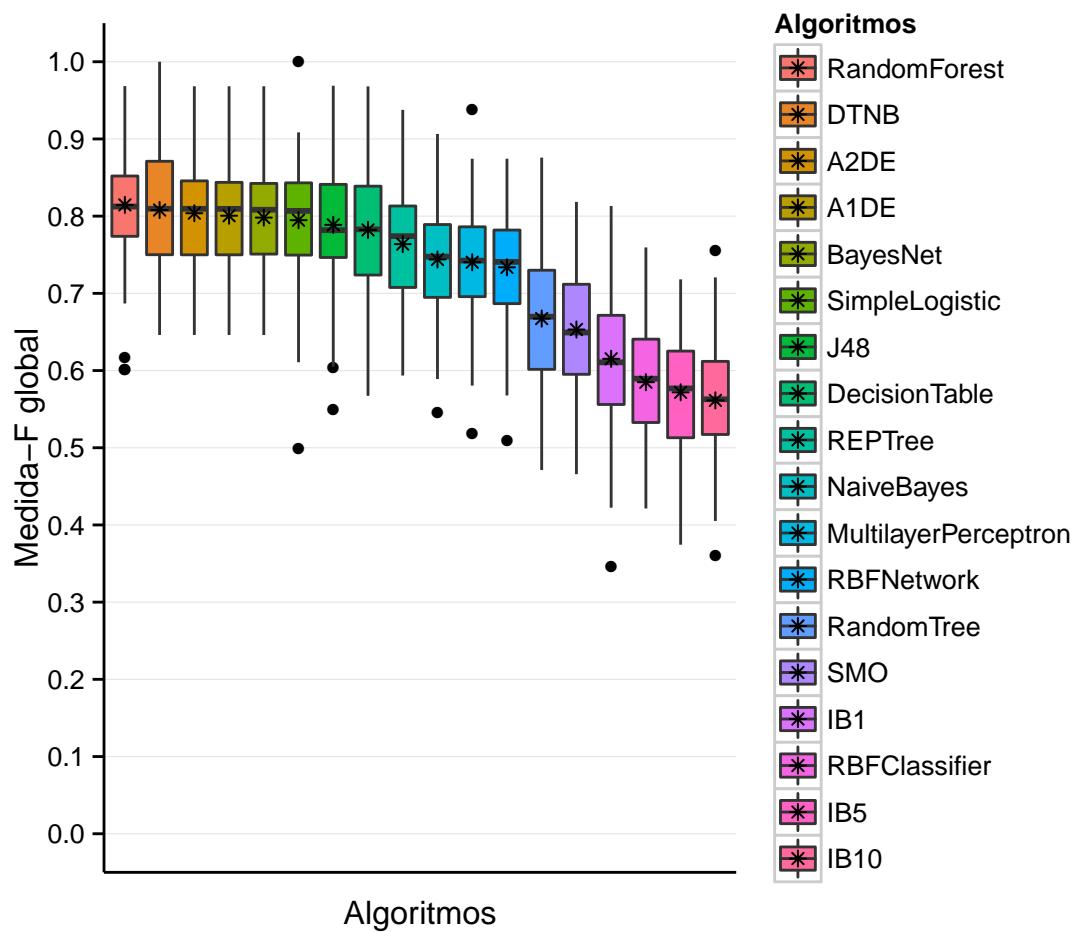


Figura. 6.15 Digrama de cajas con respecto a la métrica Medida F considerando los datos de navegación web desbalanceado.

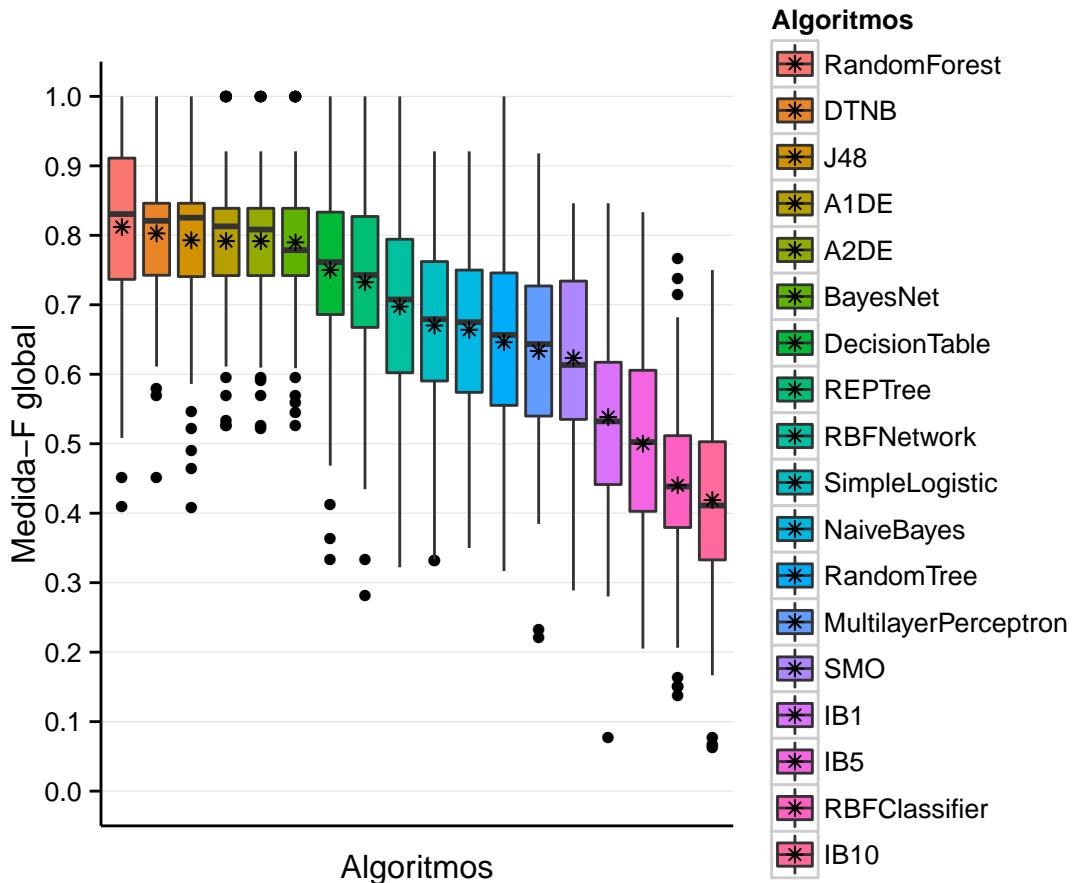


Figura. 6.16 Diagrama de cajas con respecto a la métrica Medida F considerando los datos de navegación web balanceado.

A partir de los resultados analizados, se obtuvo el siguiente ranking de algoritmos de aprendizaje automático de acuerdo al conjunto de datos desbalanceados para ambos contextos (video streaming y navegación web). Debido a la mínima variación con respecto a los conjuntos desbalanceados, los conjuntos de datos balanceados no se han tenido en cuenta para dicho ranking.

- | Video streaming | Navegación web |
|---|---|
| 1. A2DE.
2. A1DE.
3. RandomForest.
4. SimpleLogistic. | 1. RandomForest.
2. DTNB.
3. A2DE.
4. A1DE. |

6.6 Análisis de resultados

Se puede observar que los algoritmos A2DE, A1DE y RandomForest son comunes para ambos conjuntos de datos, en tanto que la lista difiere en los algoritmos SimpleLogistic y DTNB.

Con la finalidad de seleccionar un algoritmo capaz de desempeñarse adecuadamente en ambos conjuntos de datos se ha creado un sistema ensamblado formado por los cinco algoritmos con mejores desempeños: A1DE, A2DE, SimpleLogistic, RandomForest y DTNB. La operación de combinación de los resultados de los algoritmos base es sencilla: se toma como etiqueta (clase) aquella votada por la mayoría de los algoritmos base. Para este fin, se ha utilizado el metaclasificador *Vote* [Kun04, KHDM98], proporcionado por WEKA, con la opción de combinación de resultados por mayoría (ver figura 6.17). En casos en los cuales no se llega a la mayoría de votos para una etiqueta, de acuerdo a la implementación del sistema ensamblado en WEKA, se toma de manera aleatoria una etiqueta dentro de la lista de etiquetas votadas por los clasificadores componentes. Otros metaclasificadores proporcionados por WEKA para la formación de algoritmos ensamblados se pueden observar en [WF05]. En las figuras 6.18 se muestran los resultados obtenidos por el sistema ensamblado para el conjunto de video streaming desbalanceado. De acuerdo al cuadro 6.10, en promedio, el sistema ensamblado presenta mejores resultados en todas las métricas analizadas. Por otra parte en las figuras 6.19 se observan los resultados de la ejecución del algoritmo ensamblado para el conjunto de datos de navegación web desbalanceado. Los resultados del sistema ensamblado son bastante similares a los obtenidos por los algoritmos individuales (ver cuadro 6.10).

Cuadro. 6.10 Resultados de la ejecución del sistema ensamblado con los conjuntos de datos de navegación web y video streaming.

Métrica	Resultados	
	Video streaming	Navegación web
Porcentaje de aciertos	$80,2 \pm 6,323$	$81,8 \pm 6,6$
Kappa	$0,75 \pm 0,08$	$0,76 \pm 0,086$
Medida-F global	$0,80 \pm 0,065$	$0,81 \pm 0,07$

6.6 Análisis de resultados

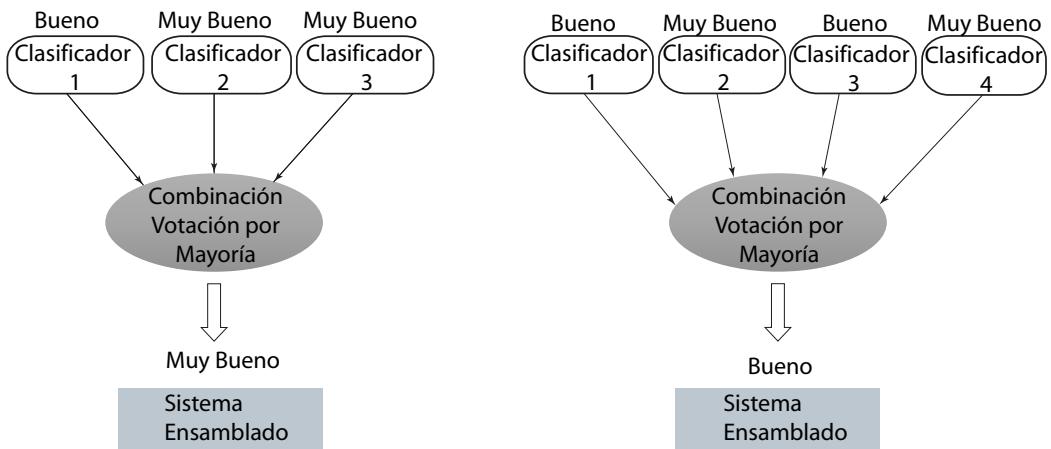


Figura. 6.17 Esquema del sistema ensamblado por mayoría de votación. En caso de no existir un consenso de mayoría, se escoge de manera aleatoria una etiqueta dentro de la lista de etiquetas tomadas en cuenta por los clasificadores componentes.

Como conclusión, observando los diagramas de cajas de las figuras 6.18 y 6.19, se demuestra una menor variabilidad en los resultados del sistema ensamblado para ambos contextos de uso, notándose que la mayor parte de los resultados están centrados en la media.

6.6 Análisis de resultados

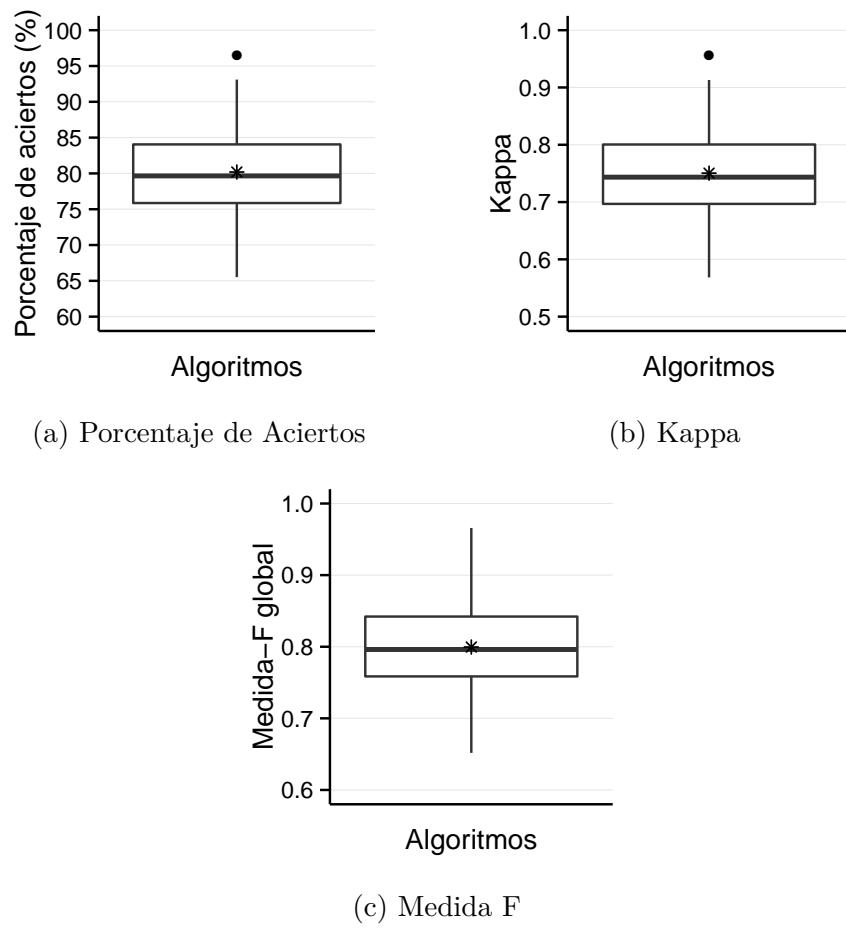


Figura. 6.18 Diagramas de cajas con respecto a las métricas analizadas para el algoritmo ensamblado sobre el conjunto de datos de video streaming.

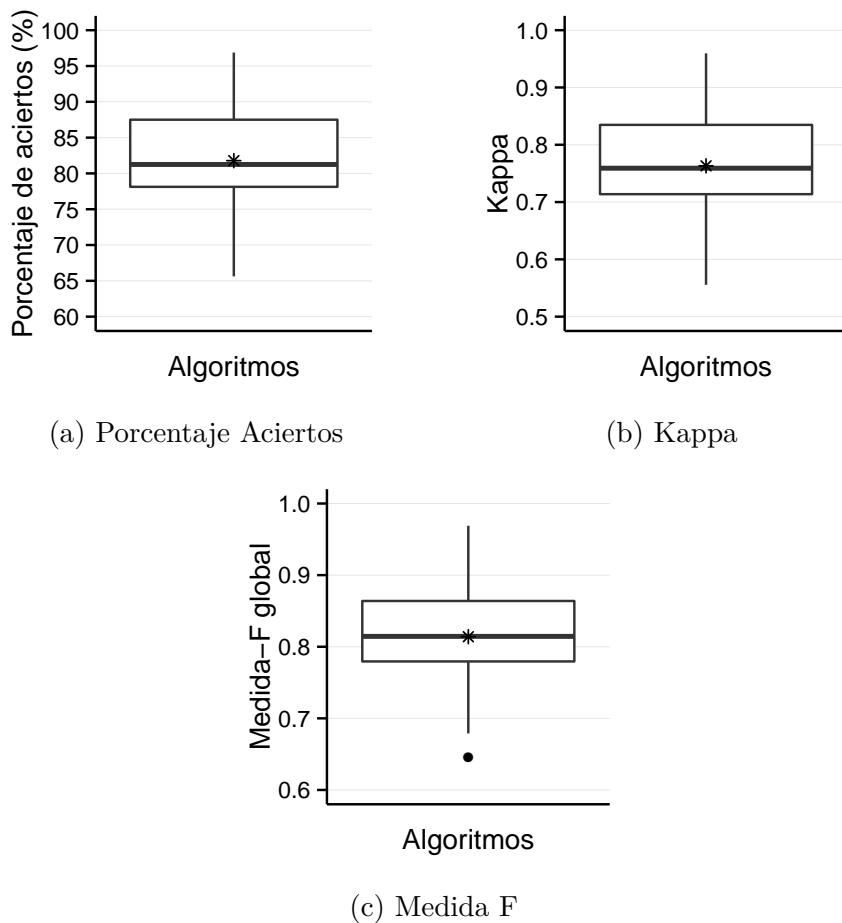


Figura. 6.19 Diagramas de cajas con respecto a las métricas analizadas para el algoritmo ensamblado sobre el conjunto de datos de navegación web.

6.7. Discusión

Este capítulo abarcó el análisis y las comparaciones de los diferentes algoritmos de aprendizaje automático aplicados al problema de la estimación de la calidad de experiencia del usuario en el uso de datos móviles para los contexto de video streaming y navegación web, con respecto a tres niveles de influencia: *dispositivo, red y aplicación*. Las comparaciones se realizaron en base a un experimento de 10 iteraciones de 10-*cross validation* utilizando gran parte de los algoritmos proporcionados por el software WEKA. Formaron parte del experimento tanto los conjuntos de datos referentes al contexto de video streaming como los de navegación web. Así también, cada uno de estos conjuntos pasó por un proceso de “balanceo” con la finalidad de obtener una distribución uniforme con respecto a las clases (Malo, Pobre, Bueno, Muy Bueno,

Excelente).

Como resultado de los experimentos, se evaluaron los algoritmos con respecto a tres métricas generales: porcentaje de aciertos, kappa y la medida F global. Para el conjunto de datos de video streaming, a partir de los resultados y los diagramas de cajas, se puede observar que los clasificadores A2DE y A1DE, en la mayoría de los escenarios obtienen, en promedio, los mejores resultados en la mayoría de las métricas analizadas. En tanto que para el conjunto de datos de navegación web, el algoritmo RandomForest obtiene mejores resultados. Sin embargo, las diferencias con otros clasificadores, como los bayesianos A2DE y A1DE, es mínima.

Si bien, en problemas con una distribución de clases desbalanceada, el proceso de balanceo puede mejorar el desempeño de los algoritmos de clasificación, en el caso del conjunto balanceado de video streaming ocurrió exactamente lo contrario, se observó un desempeño ligeramente inferior en todas las métricas analizadas sobre el conjunto de datos balanceado. Sin embargo, esto no ocurrió con el conjunto de datos balanceado de navegación web, en el que los algoritmos sí presentaron un desempeño mayor (en pequeñas proporciones). Esto quizás se deba a que el conjunto de navegación web cuenta con una menor cantidad de atributos que el conjunto de video streaming.

Por otra parte, algoritmos tales como: IBk, RandomTree, SMO y RFBClassifier obtuvieron valores inferiores en la mayoría de las métricas analizadas para ambos contextos de uso. Por tanto, estos algoritmos pueden ser descartados para su uso en este tipo de problemas.

Para finalizar el análisis se realizó un experimento adicional, el cual consiste en la ejecución de un sistema ensamblado de votación por mayoría conformado por los algoritmos de aprendizaje que obtuvieron los mejores desempeños en las métricas analizadas. A partir de los gráficos se puede visualizar una mejora mínima de este sistema en comparación a sus algoritmos base, ya sea en el rendimiento de las métricas analizadas como en la variabilidad en los resultados obtenidos. En cuanto a la relación de los parámetros analizados y el QoE se pudo concluir lo siguiente:

- Existe una alta influencia de los parámetros de red como el ancho de banda y paquetes perdidos. Estos tienden a aumentar el QoE a medida que mejoran.
- Los parámetros a nivel de aplicación juegan un rol importante en el QoE del usuario, donde un incremento en sus valores disminuye el QoE percibido.
- Caso especial se presenta en los parámetros a nivel de dispositivo, en donde, si bien existe influencia sobre el QoE, ésta es menor. Sin embargo, no se pue-

6.8 Obtención del Modelo de Correlación

de obtener una conclusión debido a que durante las pruebas realizadas no se proporcionó un elevado estrés a los dispositivos como para influir de manera significativa en el QoE final. Otros tipos de pruebas, como por ejemplo el cambio entre aplicaciones, son necesarias para obtener mayor detalle acerca de la influencia de los parámetros del dispositivo sobre el QoE.

Parte de los parámetros analizados en función al QoE (valor MOS) puede ser vistos gráficamente en el Apéndice E.

6.8. Obtención del Modelo de Correlación

Como se observa en el esquema 5.1 del capítulo 1, el objetivo final del trabajo, por el cual se llevaron a cabo los pasos anteriores, es la obtención de un modelo de correlación de los parámetros correspondientes al estado de la red, al estado del dispositivo y a métricas de nivel de aplicación con la calidad percibida por el usuario final para cada contexto de uso.

En la sección 6.6 se realizó una comparación de rendimiento de los distintos algoritmos de clasificación ofrecidos por la herramienta WEKA, utilizando como conjunto entrenamiento los datos recolectados por la aplicación móvil: parámetros a nivel de red, dispositivo y aplicación, más las calificaciones asignadas por los usuarios a la calidad percibida con distintas configuraciones de dichos parámetros. De acuerdo a los resultados del análisis, se seleccionó al clasificador ensamblado como modelo de correlación para ambos contextos de uso.

WEKA nos permite guardar los modelos obtenidos por los clasificadores entrenados como objetos JAVA serializados en un archivo binario [atUoW]. Esto permite que los modelos creados puedan ser utilizados nuevamente, ya sea con la interfaz de WEKA o dentro de un programa JAVA que controla la ejecución del mismo independientemente de la intefaz de WEKA [WJC].

Utilizando WEKA se crearon dos archivos binarios que representan a los modelos, uno para cada contexto de uso, los cuales podrán ser nuevamente utilizados por medio de la librería JAR del software WEKA.

Los modelos de correlación obtenidos al final del proceso de entrenamiento permiten realizar predicciones acerca de la calidad de la experiencia (valor MOS) para cada contexto de uso en base a un conjunto de atributos de entrada. Como resultado, se obtiene una herramienta que permite estimar la calidad percibida por los usuarios

6.8 Obtención del Modelo de Correlación

para cada contexto (navegación web y streaming de video) en base al monitoreo de un conjunto de parámetros a nivel de red, dispositivo y aplicación.

Capítulo 7

Conclusiones

7.1. Conclusiones Generales

Este trabajo abarcó en gran medida el estudio de la calidad de experiencia percibida por los usuarios en los contextos de navegación web y video streaming sobre transmisiones de datos móviles. Como ya se había mencionado en el capítulo 1, el monitoreo y la evaluación de la calidad de la experiencia juegan un rol importante dentro del modelo de negocio de las prestadoras de servicios móviles. Así, partiendo de la base de la problemática de correlacionar una medida subjetiva de evaluación con diversas variables medibles (parámetros de red, estado del dispositivo, parámetros de aplicación), se realizó un análisis del estado del arte acerca de las metodologías de evaluación de la calidad de la experiencia, entre las que se destaca la metodología *PSQA*, la cual fue adaptada para este trabajo.

Como resultado, se presenta un enfoque híbrido de evaluación de la calidad de la experiencia con un análisis en tres capas: dispositivo móvil, red y aplicación mediante el uso de métodos de aprendizaje automático.

Por otra parte, el trabajo presenta una comparación de rendimiento de diversos algoritmos de aprendizaje automático (clásificadores) aplicados a dicho problema, en base a diversas métricas de evaluación.

Para finalizar, se escoge un modelo de correlación de la calidad de la experiencia y los parámetros analizados. Este modelo toma la forma de un sistema ensamblado conformado por los algoritmos con mayor desempeño, en las métricas analizadas, como resultado del estudio y la comparación anteriormente citada.

7.2. Aporte de este trabajo

Este trabajo de grado otorga como resultados los siguientes ítems:

- Una plataforma de transmisión de datos móviles que permite emular los escenarios de una red de transmisión celular.
- Una agente móvil que permite monitorear los parámetros de calidad de servicio a nivel de red y dispositivo. Así también, este agente móvil es capaz de obtener otro tipo de información como: localización por gps, ID de la celda conectada, tipo de tecnología de acceso, entre otros; otorgándole al agente móvil la capacidad de ser extendido para su uso en entornos móviles operativos.
- Una aplicación móvil en forma de encuestas que permite obtener la apreciación del usuario acerca de la calidad experimentada con los servicios móviles en dos contextos de uso cotidiano: navegación web y video streaming.
- Un modelo de estimación de la calidad de la experiencia basado en diversos parámetros influyentes a nivel de red, dispositivo y aplicación.

7.3. Trabajos Futuros

En función de las conclusiones obtenidas y con el objetivo de futuras mejoras, presentamos a continuación una serie de propuestas que podrían dar continuidad a este trabajo:

- Estudio y análisis del impacto de otros factores influyentes en la calidad de la experiencia tales como: políticas de precios, el contexto (localización, actividad, horario, etc.), el perfil del usuario, entre otros.
- Aplicar el agente móvil en un entorno de red móvil real con la finalidad de recolectar datos en diversas situaciones.
- Validación del modelo de correlación con datos de un entorno operativo real.
- Realizar una optimización de los algoritmos de aprendizaje proporcionados por el software WEKA por medio de la selección de los parámetros óptimos.
- Expandir este trabajo a otros contextos de uso sobre transmisiones de datos móviles (live streaming, videoconferencias, entre otros).

7.3 Trabajos Futuros

- Utilizar el modelo de correlación obtenido para proporcionar a las operadoras visibilidad en las condiciones que afectan a la calidad de la experiencia, con el fin de diseñar estrategias innovadoras de descarga de datos móviles que garanticen la calidad de la experiencia de los usuarios.

Referencias

- [AK91] D. Aha and D. Kibler. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.
- [atUoW] Machine Learning Group at the University of Waikato. WEKA - Serialization. URL: <https://weka.wikispaces.com/Serialization>.
- [atUoW15] Machine Learning Group at the University of Waikato. Weka, 2015. URL: <http://www.cs.waikato.ac.nz/~ml/weka>.
- [BC00] Kristin P Bennett and Colin Campbell. Support vector machines: hype or hallelujah? *ACM SIGKDD Explorations Newsletter*, 2(2):1–13, 2000.
- [Bou03] Remco R Bouckaert. Choosing between two learning algorithms based on calibrated tests. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 51–58, 2003.
- [Bre01] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [Cha12] Winston Chang. *R graphics cookbook*. O'Reilly Media, Inc., 2012.
- [CO03] Jorge E Caviedes and Franco Oberti. No-reference quality metric for degraded and enhanced video. In *Visual Communications and Image Processing 2003*, pages 621–632. International Society for Optics and Photonics, 2003.
- [DC02] Carlo Demichelis and Philip Chimento. IP packet delay variation metric for IP performance metrics (IPPM). 2002.
- [DHS99] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons,, 1999.
- [DIC15] The american heritage® science dictionary. Mar 2015. URL: <http://dictionary.reference.com/browse/bandwidth>.
- [DPDMJ⁺13] Toon De Pessemier, Katrien De Moor, Wout Joseph, Lieven De Marez, and Luc Martens. Quantifying the influence of rebuffering interruptions on the user's quality of experience during mobile video watching. *Broadcasting, IEEE Transactions on*, 59(1):47–61, 2013.
- [ETS15] ETSI. Edge, 2015. URL: <http://www.etsi.org/technologies-clusters/technologies/mobile/edge>.

- [FHTG10] Markus Fiedler, Tobias Hossfeld, and Phuoc Tran-Gia. A generic quantitative relationship between quality of experience and quality of service. *Network, IEEE*, 24(2):36–41, 2010.
- [Fle71] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.
- [Fou] Linux Foundation. Netem. URL: <http://www.linuxfoundation.org/collaborate/workgroups/networking/netem>.
- [Goo14a] Google. Android developers - ActivityManager, 2014. URL: <http://developer.android.com/reference/android/app/ActivityManager.MemoryInfo.html>.
- [Goo14b] Google. Android Developers - ConnectivityManager, 2014. URL: <http://developer.android.com/reference/android/net/ConnectivityManager.html>.
- [Goo14c] Google. Android Developers - TelephonyManager, 2014. URL: <http://developer.android.com/reference/android/telephony/TelephonyManager.html>.
- [Goo14d] Google. Android Developers - TrafficStats, 2014. URL: <http://developer.android.com/reference/android/net/TrafficStats.html>.
- [Goo14e] Google. Android developers - VideoView, 2014. URL: <http://developer.android.com/reference/android/widget/VideoView.html>.
- [Goo14f] Google. Android Developers - WebView, 2014. URL: <http://developer.android.com/reference/android/webkit/WebView.html>.
- [Goo14g] Google. Android Developers - WiFiManager, 2014. URL: <http://developer.android.com/reference/android/net/wifi/WifiManager.html>.
- [Hay98] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 1998.
- [HF08] Mark Hall and Eibe Frank. Combining naive bayes and decision tables. In *Proceedings of the 21st Florida Artificial Intelligence Society Conference (FLAIRS)*, pages 318–319. AAAI press, 2008.
- [HFH⁺09] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.
- [HT98] Trevor Hastie and Robert Tibshirani. Classification by pairwise coupling. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. MIT Press, 1998.

- [Hub] Bert Hubert. *tc command manual*. URL: <http://lartc.org/manpages/tc.txt>.
- [ITUa] ITU-T Recommendation E.800. Quality of Telecommunication Services: Concepts, Models, Objectives and Dependability Planning. Terms and Definitions related to the Quality of Telecommunication Services. Technical report, International Telecommunication Union. URL: <http://www.itu.int/rec/T-REC-E.800-200809-I>.
- [ITUb] ITU-T Recommendation G.107. The E-model: a computational model for use in transmission planning, series=ITU-T Series G: Transmission Systems and media, digital systems and networks, url=<https://www.itu.int/rec/T-REC-G.107-201402-I/es>, institution = "international telecommunication union", type=recommendation", year = 2014. Technical report.
- [ITUc] ITU-T Recommendation J.247. Objective perceptual multimedia video quality measurement in the presence of a full reference, series=ITU-T Series J: Cable networks and transmission of television, sound programme and other multimedia signals, url=<http://www.itu.int/rec/T-REC-J.247-200808-I/en>, institution = "international telecommunication union", type=recommendation", year = 2008. Technical report.
- [ITU96] ITU-T Recommendation P.800. Methods for Subjective Determination of Transmission Quality. Recommendation, International Telecommunication Union, 1996. URL: <https://www.itu.int/rec/T-REC-P.800-199608-I/es>.
- [ITU08] ITU-T Recommendation P.910. Subjective video quality assessment methods for multimedia applications. Recommendation, International Telecommunication Union, 2008. URL: <http://www.itu.int/rec/T-REC-P.910-199909-S/en>.
- [ITU14] ITU-T Recommendation P.1501. Subjective testing methodology for web browsing. Technical report, International Telecommunication Union, 2014. URL: <https://www.itu.int/rec/T-REC-P.1501-201402-I>.
- [JL95] George H. John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345, San Mateo, 1995. Morgan Kaufmann.
- [JS11a] Nathalie Japkowicz and Mohak Shah. *Evaluating learning algorithms: a classification perspective*. Cambridge University Press, 2011.
- [JS11b] Nathalie Japkowicz and Mohak Shah. *Evaluating learning algorithms: a classification perspective*. Cambridge University Press, 2011.
- [K⁺⁹⁵] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, 1995.

- [Ker14] Michael Kerrisk. proc - process information pseudo-filesystem, 2014. URL: <http://man7.org/linux/man-pages/man5/proc.5.html>.
- [KH02] Stas Khirman and Peter Henriksen. Relationship between quality-of-service and quality-of-experience for public internet service. In *In Proc. of the 3rd Workshop on Passive and Active Measurement*, 2002.
- [KHDM98] J. Kittler, M. Hatef, Robert P.W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.
- [KSBM01] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, and K.R.K. Murthy. Improvements to platt’s smo algorithm for svm classifier design. *Neural Computation*, 13(3):637–649, 2001.
- [Kun04] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley and Sons, Inc., 2004.
- [lCvH92] S. le Cessie and J.C. van Houwelingen. Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201, 1992.
- [LGAPRD⁺¹²] Carlos Lozano-Garzon, Christian Ariza-Porras, Sebastian Rivera-Diaz, Horacio Riveros-Ardila, and Yezid Donoso. Mobile network QoE-QoS decision making tool for performance optimization in critical web service. *International Journal of Computers Communications & Control*, 7(5):892–899, 2012.
- [LHF05] Niels Landwehr, Mark Hall, and Eibe Frank. Logistic model trees. 95(1-2):161–205, 2005.
- [MCC11] Ricky KP Mok, Edmond WW Chan, and Rocky KC Chang. Measuring the quality of experience of http video streaming. In *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*, pages 485–492. IEEE, 2011.
- [Mit97] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.
- [MOS13] Mobile crowd sensing and data offloading in collaborative networks, 2013.
- [MR02] Samir Mohamed and Gerardo Rubino. A study of real-time packet video quality using random neural networks. *Circuits and Systems for Video Technology, IEEE Transactions on*, 12(12):1071–1083, 2002.
- [MRT12] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.
- [Pla98] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schoelkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998. URL: <http://research.microsoft.com/~jplatt/smo.html>.

- [Pol06] Robi Polikar. Ensemble based systems in decision making. *Circuits and Systems Magazine, IEEE*, 6(3):21–45, 2006.
- [PSQ07] The PSQA project, 2007. URL: <http://www.irisa.fr/armor/lesmembres/Rubino/myPages/psqa.html>.
- [PVBK09] Kandaraj Piamrat, Cesar Viho, J Bonnin, and Adlen Ksentini. Quality of experience measurements for video streaming over wireless networks. In *Information Technology: New Generations, 2009. ITNG'09. Sixth International Conference on*, pages 1184–1189. IEEE, 2009.
- [Qui93] Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.
- [R C15] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015. URL: <http://www.R-project.org/>.
- [RRAP⁺13] Sebastian Rivera, Horacio Riveros, Christian Ariza-Porras, Carlos Lozano-Garzon, and Yezid Donoso. Qos-qoe correlation neural network modeling for mobile internet services. In *Computing, Management and Telecommunications (ComManTel), 2013 International Conference on*, pages 75–80. IEEE, 2013.
- [RTV06] Gerardo Rubino, Pierre Tirilly, and Martin Varela. Evaluating users’ satisfaction in packet networks using random neural networks. In *Artificial Neural Networks–ICANN 2006*, pages 303–312. Springer, 2006.
- [Rub05] Gerardo Rubino. Quantifying the quality of audio and video transmissions over the internet: the psqa approach. *Design and operations of communication networks: a review of wired and wireless modeling and management challenges*. Imperial College Press, London, 2005.
- [Sas07] Yutaka Sasaki. The truth of the f-measure. *Teach Tutor mater*, pages 1–5, 2007.
- [SCJ11] Rafal Stankiewicz, Piotr Cholda, and Andrzej Jajszczyk. QoX: what is it really? *Communications Magazine, IEEE*, 49(4):148–158, 2011.
- [SFH05] Marc Sumner, Eibe Frank, and Mark Hall. Speeding up logistic model tree induction. In *9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 675–683. Springer, 2005.
- [SL09] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.
- [SLC07] David Soldani, Man Li, and Renaud Cuny. *QoS and QoE management in UMTS cellular systems*. John Wiley & Sons, 2007.

- [Sol06] David Soldani. Means and methods for collecting and analyzing qoe measurements in wireless networks. In *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*, pages 531–535. IEEE Computer Society, 2006.
- [Ste97] Stephen V Stehman. Selecting and interpreting measures of thematic classification accuracy. *Remote sensing of Environment*, 62(1):77–89, 1997.
- [SV08] Alex Smola and SVN Vishwanathan. Introduction to machine learning, 2008.
- [TSS14] Christos Tsiaras, Anuj Sehgal, and Sebastian Seeber. Towards evaluating type of service related quality-of-experience on mobile networks, 2014.
- [WBW05] G. Webb, J. Boughton, and Z. Wang. Not so naive bayes: Aggregating one-dependence estimators. *Machine Learning*, 58(1):5–24, 2005.
- [WF05] Ian H Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.
- [Wic09] Hadley Wickham. *ggplot2: elegant graphics for data analysis*. Springer New York, 2009. URL: <http://had.co.nz/ggplot2/book>.
- [Wik14] Wikipedia. Network delay — wikipedia, the free encyclopedia, 2014. [Online; accessed 3-March-2015]. URL: http://en.wikipedia.org/w/index.php?title=Network_delay&oldid=631555777.
- [Wik15] Wikipedia. Packet loss — wikipedia, the free encyclopedia, 2015. [Online; accessed 3-March-2015]. URL: http://en.wikipedia.org/w/index.php?title=Packet_loss&oldid=647555402.
- [WJC] Use WEKA in your java code. URL: <https://weka.wikispaces.com/Use+Weka+in+your+Java+code#Classifyinginstances>.

Apéndices

Apéndice A

Detalles de los contenidos del servidor transmisor

A.1. Contenidos multimedia

En esta sección se detallan las características técnicas acerca de los contenidos almacenados en el servidor transmisor.

A.1.1. Video 1

Cuadro. A.1 Metadatos del video 1.

Característica	Descripción
Formato	MPEG-4
Códec Video	AVC - <i>Advanced Video Coding</i>
Bitrate (video)	Máx: 1024 <i>kbytes per second</i> (variable)
Framerate	Máx: 24.390 <i>fps</i> (variable)
Códec de audio	AAC - <i>Advanced Audio Codec</i>
Bitrate (audio)	128 <i>kbytes per second</i> (constante)
Resolución	854x358 píxeles
Bitrate total	1186 <i>kbytes per second</i>
Tamaño del archivo	15.8 MB
Duración	1 minuto y 40 segundos.

A.1.2. Video 2

Cuadro. A.2 Metadatos del video 2.

Característica	Descripción
Formato	MPEG-4
Códec Video	AVC - <i>Advanced Video Coding</i>
Bitrate (video)	Máx: 606 <i>kbytes per second</i> (variable)
Framerate	30.303 <i>fps</i> (constante)
Códec de audio	AAC - <i>Advanced Audio Codec</i>
Bitrate (audio)	128 <i>kbytes per second</i> (constante)
Resolución	852x380 píxeles
Bitrate total	757 <i>kbytes per second</i>
Tamaño del archivo	6.51 MB
Duración	1 minuto.

A.1.3. Video 3

Cuadro. A.3 Metadatos del video 3.

Característica	Descripción
Formato	MPEG-4
Códec Video	AVC - <i>Advanced Video Coding</i>
Bitrate (video)	767 <i>kbytes per second</i> (constante)
Framerate	30.303 <i>fps</i> (constante)
Códec de audio	AAC - <i>Advanced Audio Codec</i>
Bitrate (audio)	128 <i>kbytes per second</i> (constante)
Resolución	854x480 píxeles
Bitrate total	969 <i>kbytes per second</i>
Tamaño del archivo	4.63 MB
Duración	40 segundos.

A.1.4. Página Web

El contenido de la página web consiste en un conjunto de animaciones en formato GIF (*Graphics Interchange Format*) junto con un pequeño texto descriptivo que acompaña a cada una. En el siguiente cuadro se detallan los componentes de la página web junto con el tamaño de cada uno en *kilobytes(kB)*.

Cuadro. A.4 Componentes de la página web de prueba.

Componente	Tamaño (kB)
CSS	114.7
Javascript	138.7
GIF	3648,7
HTML	5

Cabe destacar que la página de prueba se ha creado teniendo en cuenta la pantalla y las capacidades de los dispositivos móviles. Al igual que diversas páginas móviles reales, esta página se encuentra optimizada para su uso en dispositivos móviles mediante la carga de componentes necesarios en el momento necesario. Por otra parte, no todas las animaciones GIF se cargan al inicio de la página, se da la posibilidad al usuario de cargarlas a medida que lo deseé.

Apéndice B

Servidor de Configuración

B.1. Conceptos

A continuación definiremos los parámetros básicos de calidad de servicio que fueron variados por el servidor de configuración para generar distintas condiciones de red.

- *Ancho de banda (Bandwidth)*: Mide la cantidad de datos que se pueden enviar a través de un canal de comunicaciones en un período determinado de tiempo. Típicamente es expresado en múltiplos de bits por segundo [DIC15].
- *Retardo (Delay)*: es el tiempo que le toma a un bit de datos viajar a través de la red de un nodo a otro. Se mide en múltiplos o fracciones de segundo. El retardo puede variar ligeramente, dependiendo de la ubicación del par específico de nodos de comunicación [Wik14].
- *Jitter*: Es la variación en el retardo de paquetes. También se mide en múltiplos o fracciones de segundo [DC02].
- *Paquetes perdidos (Packet loss)*: ocurre cuando uno o más paquetes de datos que viajan a través de una red no llegan a su destino. Para medir este parámetro se calcula el porcentaje de paquetes que no llegaron a su destino sobre el total de paquetes enviados [Wik15].

B.2. Variación de parámetros de red

Los parámetros definidos en la sección anterior fueron utilizados para variar las condiciones de la red por medio del siguiente script bash (sh) ejecutado en el servidor

B.2 Variación de parámetros de red

de configuración.

Listing B.1 Script de configuración de parámetros

```
#!/bin/bash
#Nombre del comando de control de trafico
TC=/sbin/tc
#Interfaz de red a la que se le aplica las configuraciones
NI=eth0
# Limite de ancho de banda(en kilobits)
BW=$2
#Delay(en milisegundos)
DELAY=$3
#Jitter(en milisegundos)
JITTER=$4
#Paquetes perdidos(en porcentaje)
PACLOSS=$5
start() {
    $TC qdisc del dev $NI root
    $TC qdisc add dev $NI root handle 1: htb default 1
    $TC class add dev $NI parent 1:1 classid 1:1 htb
    rate $BW ceil $BW
    $TC qdisc add dev $NI parent 1:1 handle 10:
    netem delay ${DELAY} ${JITTER} loss ${PACLOSS}
}
stop() {
    $TC qdisc del dev $NI root
}
show() {
    $TC -s qdisc ls dev $NI
}
case "$1" in
    start)
        echo -n "Comenzando configuracion: "
        start
    stop)
        echo -n "Deteniendo configuracion: "
    esac
```

```
stop
show)
echo "Estado de la configuracion:"
show
*)
pwd=$(pwd)
echo "Uso: tc.bash {start|stop|show}
{BW+kbit} {DELAY+ms} {JITTER+ms} {PL+%}"
esac
exit 0
```

Este script recibe como primer parámetro la acción que se desea realizar, existen tres opciones posibles:

- *start*: al seleccionar esta acción se procederá a ejecutar los comandos de control de tráfico de Linux *tc* y *netem*, utilizando los cuatro parámetros restantes que indican el límite de ancho de banda en kbytes, el delay en milisegundos, el jitter también en milisegundos y el porcentaje de paquetes perdidos. De esta manera la interfaz de red seleccionada queda configurada con las condiciones de red indicadas por los parámetros de entrada del script bash.
- *stop*: esta acción remueve las configuraciones del tráfico de red aplicadas a la interfaz establecida en el script.
- *show*: al escoger esta acción se muestran las configuraciones de red actuales de la interfaz indicada en el script.

Las distintas condiciones de red utilizadas en este experimento son leídas de un archivo y luego pasadas como parámetros de entrada al script. Para generar dichas condiciones de red escogimos diversos valores para los parámetros seleccionados que pueden ser observados en el siguiente cuadro.

B.2 Variación de parámetros de red

Cuadro. B.1 Parámetros de red establecidos en el servidor de configuración

Parámetros	Valores	Unidades
Ancho de banda	350, 750, 1200, 1600	kbps.
Delay	100, 300, 500	ms.
Paquetes perdidos	0, 2,5, 5, 10	%
Jitter	0, 25, 50	ms.

Algunas de estas condiciones de red generadas se corresponden a variaciones encontradas en las redes celulares reales, por ejemplo: conexiones *EDGE* con ancho de banda de 350 *kmps*[ETS15], conexiones con alta pérdida de paquetes (más del 5 %), conexiones 3G básicas (750 *kmps* y superiores), entre otros.

Apéndice C

Detalles de la Aplicación Android

C.1. Prueba Subjetiva

Al iniciar la aplicación, comienza la prueba subjetiva (ver figura C.1). El primer paso es solicitarle al usuario completar datos personales, datos sobre la utilización y costo de los servicios de datos móviles (figuras C.2 y C.3).

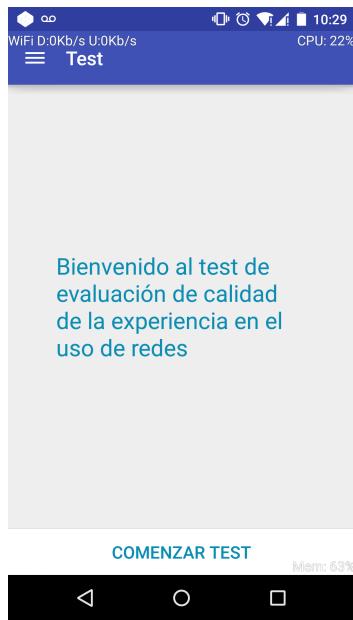


Figura. C.1 Inicio Prueba Subjetiva

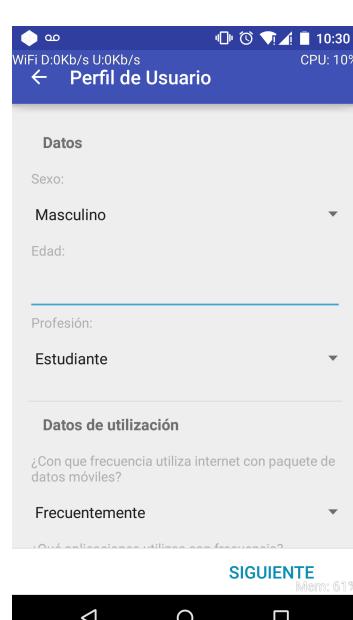


Figura. C.2 Datos (1)



Figura. C.3 Datos (2)

C.1 Prueba Subjetiva

Luego se realiza el proceso de medición de parámetros de red como se ve en las figuras C.4 y C.5. También se obtienen las características y el estado del dispositivo que esta ejecutando la aplicación (figura C.6).

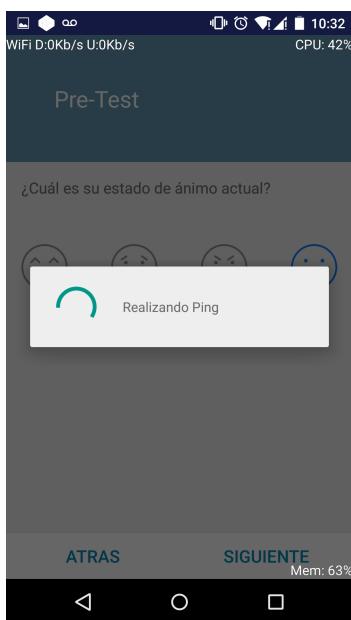


Figura. C.4 Test PING



Figura. C.5 Test IPERF

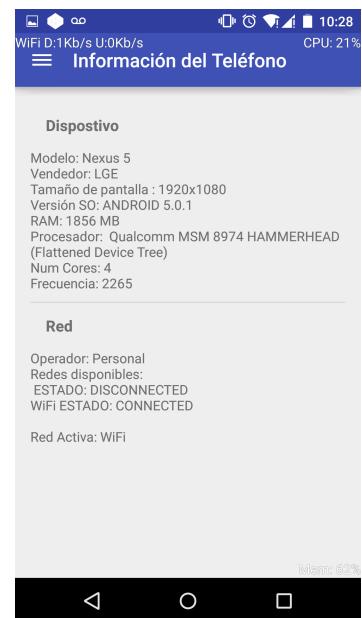


Figura. C.6 Características del dispositivo

C.1 Prueba Subjetiva

Posteriormente se procede a realizar el test de navegación web (figura C.7). En este test el usuario observa una página web con texto y animaciones (figura C.8) y luego responde una serie de preguntas acerca de la calidad del contenido y la velocidad de carga del mismo (figura C.9). Esta evaluación se realiza utilizando una escala predefinida de puntuaciones (escala absoluta MOS).

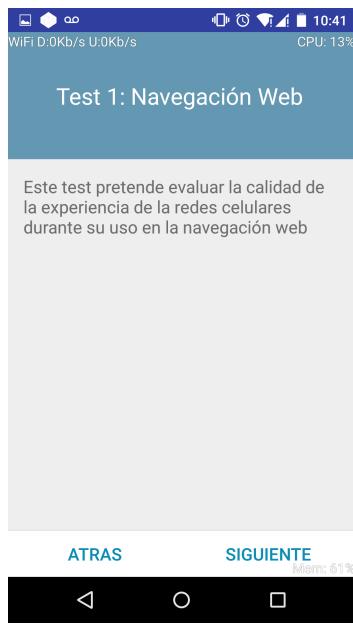


Figura. C.7 Test de Navegación Web



Luego del trabajo

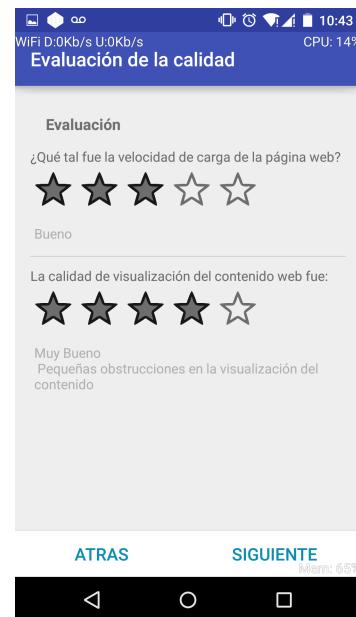


Figura. C.9 Evaluación de la Navegación Web

C.1 Prueba Subjetiva

Para culminar la prueba subjetiva se realiza el test de calidad de video (figura C.10). En este test el usuario debe observar un video que se le asigna aleatoriamente de entre los tres videos posibles que se encuentran en el servidor transmisor y que se pueden observar en las figuras C.12, C.13 y C.14. Una vez finalizada la visualización del video, el usuario procede a realizar la evaluación del mismo. La evaluación contiene un conjunto de preguntas acerca de la calidad del contenido desplegado en la pantalla, el retardo de reproducción del video y la cantidad de pausas durante la reproducción (figura C.11).

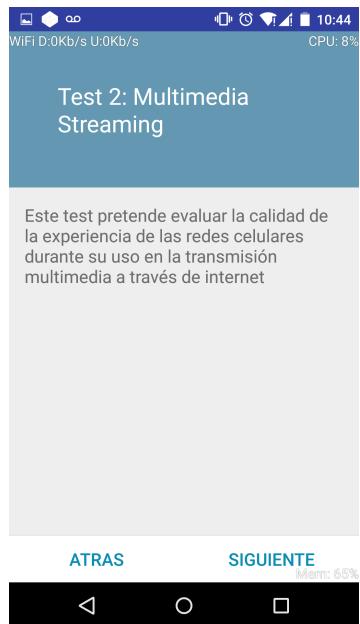


Figura. C.10 Inicio del Test de Video

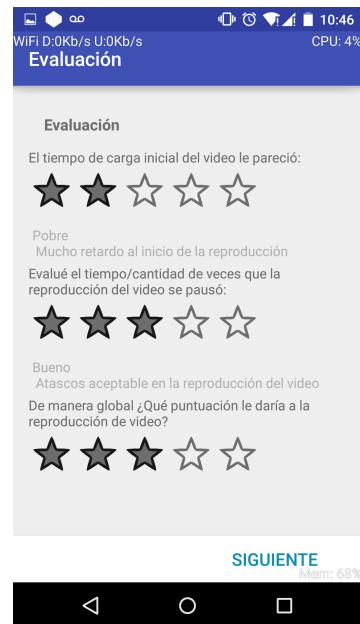


Figura. C.11 Evaluación del Video



Figura. C.12 Video 1



Figura. C.13 Video 2

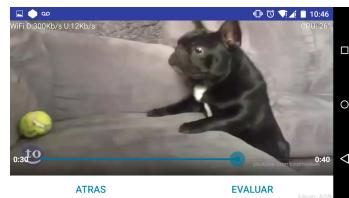


Figura. C.14 Video 3

C.1 Prueba Subjetiva

Al finalizar la prueba subjetiva se envían los datos recolectados por la aplicación al servidor de persistencia para ser almacenados (figura C.15) y se le agradece al usuario por su colaboración (figura C.16).

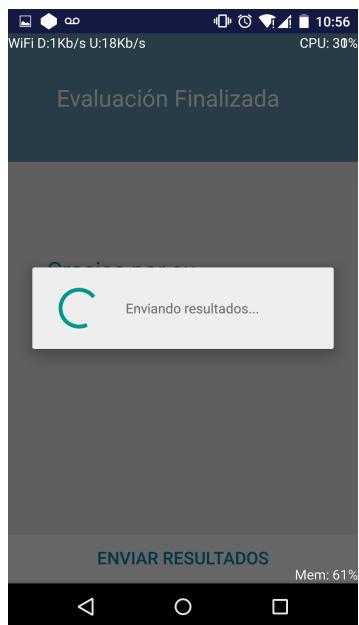


Figura. C.15 Envío de Datos

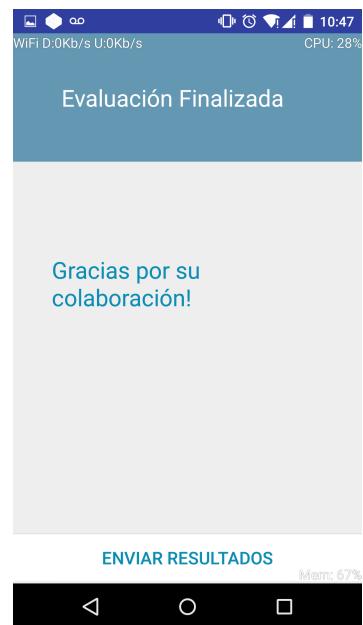


Figura. C.16 Fin de la Prueba Subjetiva

C.2. Herramientas de Medición del Agente Móvil

C.2.1. Uso de CPU

El *uso de CPU (CPU usage)* se define como el porcentaje de tiempo que la CPU del dispositivo se encuentra procesando instrucciones de programas sin tener en cuenta el tiempo que se encuentra esperando por procesos de entrada/salida (E/S) o el tiempo en el que se encuentra inactiva (*idle*). Para obtener los valores correspondientes a los tiempos de procesamiento, esperas por E/S y/o estados inactivos, se realiza una lectura al archivo /proc/stat de Linux [Ker14]. A partir de los valores del archivo, se calcula el valor porcentual de tiempo que la CPU destina a procesamiento sobre el total de tiempo transcurrido. Este proceso de lectura se lleva a cabo cada 500 ms en un hilo separado. Además de ello, estos valores porcentuales son promediados (media geométrica) con el fin de obtener el uso promedio de CPU.

C.2.2. Uso de RAM

El uso de la memoria RAM está definido como el porcentaje utilizado de la RAM sobre la cantidad total disponible. Con la finalidad de obtener la cantidad total de memoria RAM del dispositivo, se realiza una lectura al archivo /proc/meminfo, propio de Linux. A parte de ello, se hace uso del servicio Android **ACTIVITY_SERVICE**, la clase **ActivityManager** y el método *getMemoryInfo()* [Goo14a] con la finalidad de obtener la cantidad de memoria RAM disponible. Al disponer de estos datos es posible calcular el uso de RAM con la siguiente fórmula:

$$Uso\ RAM = \frac{total\ RAM - RAM\ disponible}{total\ RAM}$$

De la misma forma que el uso de la CPU, este valor es calculado cada 500 ms por un hilo separado. Además de ello, estos valores son promediados (media aritmética) con el fin de obtener el uso promedio de RAM.

C.2.3. Red

Estado de la Red

Android proporciona servicios básicos para el control del estado de la red. Entre ellos se encuentran: **CONNECTIVITY_SERVICE** y **WIFI_SERVICE**. Estos servicios representan los servicios para administración de conexiones de datos celulares o WiFi, respec-

C.2 Herramientas de Medición del Agente Móvil

tivamente. El servicio **CONNECTIVITY_SERVICE** junto a la clase **ConnectivityManager** [Goo14b] permiten realizar diferentes funciones relacionadas a las conexiones del dispositivo:

- Monitoreo del estado de la conexión del dispositivo. Verificar a que tipo de red está actualmente conectado el dispositivo (WiFi/Celular).
- Monitoreo de cambios en el estado de la conexión del dispositivo (conectado, desconectado, en proceso de conexión, entre otros).

El servicio **WIFI_SERVICE** junto a la clase **WifiManager** [Goo14g] permiten las siguientes funciones básicas:

- Listar las conexiones WiFi configuradas.
- Obtener el nivel de señal de la red WiFi conectada.

Además de los servicios citados anteriormente, se hace uso del servicio **TELEPHONY_SERVICE** junto a la clase **TelephonyManager** [Goo14c] con la finalidad de monitorear el nivel de señal dentro de la red celular. Cabe destacar que tanto la señal WiFi como la señal del dispositivo son medidas en base a un número que indica la cantidad de barras de señal, similares a la mostradas en la interfaz de cualquier dispositivo. Por tanto, el nivel de señal es un número que va de 0 (recepción nula) a 4 (recepción excelente).

Tráfico de la Red

Para el monitoreo del tráfico de las conexiones del dispositivo, Android dispone de la clase **TrafficStats** [Goo14d]. La cual proporciona estadísticas como: bytes recibidos y transmitidos, paquetes transmitidos y recibidos, sobre todas las interfaces de comunicación disponibles del dispositivo (WiFi o Celular, en este caso). Así también permite recolectar el consumo de datos filtrados por las aplicaciones instaladas en los dispositivos. Dentro del agente móvil, esta clase es revisada cada 500 ms con la finalidad de obtener las velocidades de bajada y de subida instantáneas del dispositivo. Para ello se calcula la diferencia de bytes entre un intervalo de tiempo (en este caso, cada 500 ms). Por tanto, se tiene:

$$\text{tasa de bytes/seg} = \frac{\text{bytes}_2 - \text{bytes}_1}{t_2 - t_1}$$

Los valores de bytes son transformados a bits (1 byte = 8 bits) para obtener la tasa de datos en bits/seg.

Parámetros de Red

Los parámetros de red medidos por el agente móvil son: la latencia (*round-to-trip time RTT*), jitter, ancho de banda efectivo (sobre TCP) y el porcentaje de paquetes perdidos. Debido a que Android no ofrece de manera nativa herramientas de medición de estos parámetros, se hicieron uso de herramientas Linux comúnmente utilizadas para llevar a cabo dichas mediciones: PING e IPERF. Por medio de la herramienta PING se obtienen la latencia de la red y un estimativo del porcentaje de perdidas de paquetes. Un ejemplo de una ejecución de PING se muestra en el código de abajo.

Listing C.1 Ejecución de PING

```
system/bin/ping -a -c 20 <IP>
```

Por otra parte, IPERF permite realizar una medición extensiva de los parámetros restantes (ancho de banda, jitter, paquetes perdidos). Para ello, se utiliza el IPERF en modo cliente tanto para TCP como para UDP. Un ejemplo de una ejecución de IPERF se muestra en el código de abajo.

Listing C.2 Ejecución de IPERF TCP

```
iperf -c <IP> -r -x CSM -f k -t 10 -p 10000
```

Listing C.3 Ejecución de IPERF UDP

```
iperf -c <IP> -u -r -x CSM -f k -b 2M -p 10001
```

En donde, el parámetro **-r** indica que el IPERF operará en modo *tradeoff*, de manera que tanto el cliente como el servidor pueden enviar flujos de datos al otro y así medir los parámetros tanto en bajada como en subida.

C.2.4. Parámetros a nivel de aplicación

Al momento de la creación de las pruebas subjetivas, se crearon un pseudo navegador web y un reproductor de video, utilizando las APIs nativas proporcionadas por Android: *WebView*, *WebViewClient* y *WebChromeClient* [Goo14f] y *VideoView* y *MediaPlayer* [Goo14e], respectivamente. Se utilizaron ambas para obtener un mayor control sobre las estadísticas a medir para cada contexto de uso. Así, las clases *WebView*, *WebViewClient* y *WebChromeClient* permiten obtener el tiempo transcurrido desde el inicio de la búsqueda de una página web hasta su carga completa, como así también verificar la cancelación de carga de la misma. Por tanto, el tiempo que una página tarda en ser cargada por completo puede obtenerse así:

C.2 Herramientas de Medición del Agente Móvil

$$\text{tiempo de carga web} = \text{tiempo fin de carga} - \text{tiempo inicio de carga}.$$

Para el caso del video streaming, la clase `VideoView` y, en particular, la clase nativa `Media Player` permiten obtener un control detallado de los eventos disparados durante la reproducción de un video. En este caso los eventos que nos interesan son: inicio y fin de la carga inicial, inicio y finalización de un evento de buffering y el evento de cancelación. Así, cada uno de los parámetros medidos se obtienen como sigue:

$$\text{tiempo de carga inicial} = \text{tiempo fin de carga} - \text{tiempo inicio de carga}$$

$$\text{tiempo de buffering} = \text{tiempo fin de buffering} - \text{tiempo inicio de buffering}$$

Por otra parte, el agente lleva una cuenta de la cantidad de ocurrencias de eventos de rebuffing y un promedio del tiempo de buffering transcurrido en cada evento de rebuffing disparado.

Apéndice D

Interpretación de gráficos

D.1. Diagrama de Cajas - Boxplot

En un *diagrama de cajas (o Boxplot) y bigotes*, la caja va del 25° percentil¹ hasta el 75° percentil de los datos, esto es también conocido como el *rango intercuartil (interquartile range - IQR)*. La línea dentro de la caja indica la mediana de los datos, es decir, el 50° percentil de los datos. Los bigotes comienzan desde el borde de la caja y se extienden hasta el punto más alejado que se encuentra dentro del rango igual a 1.5 veces el rango inter-cuartil. Si cualquier otro punto de dato se encuentra fuera de dicho rango, se lo considera como un *punto extremo (outlier)* y es representado por medio de puntos [Cha12]. Esto se puede observar en la figura D.1.

Cabe destacar que todos los gráficos presentados en este trabajo han sido realizados por medio del lenguaje de programación estadístico R [R C15] junto al paquete *ggplot2* [Wic09].

¹El percentil es una medida no central usada en estadística que indica, una vez ordenados los datos de menor a mayor, el valor de la variable por debajo del cual se encuentra un porcentaje dado de observaciones en un grupo de observaciones.

D.1 Diagrama de Cajas - Boxplot

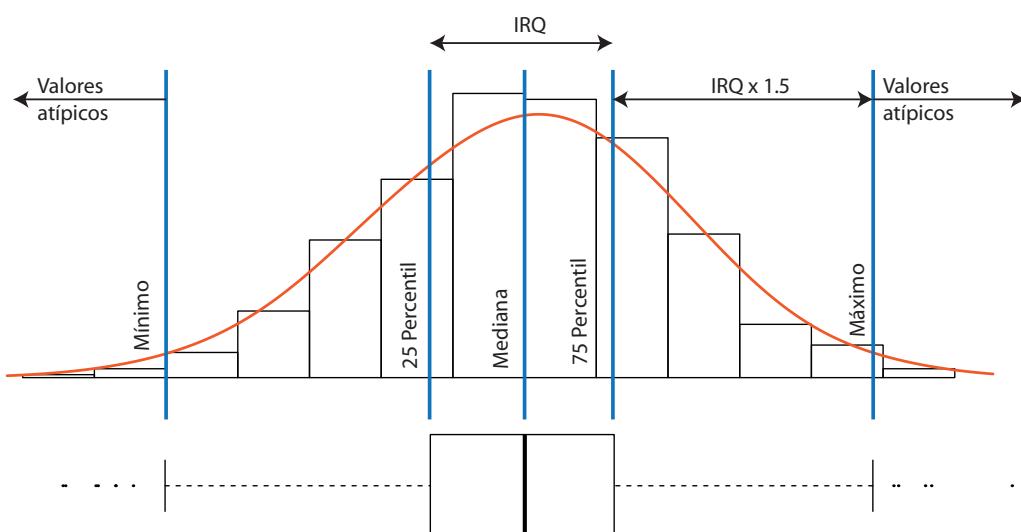


Figura. D.1 Diagrama de caja comparado con un histograma y una curva de densidad [Cha12].

Apéndice E

Influencia de parámetros sobre el QoE

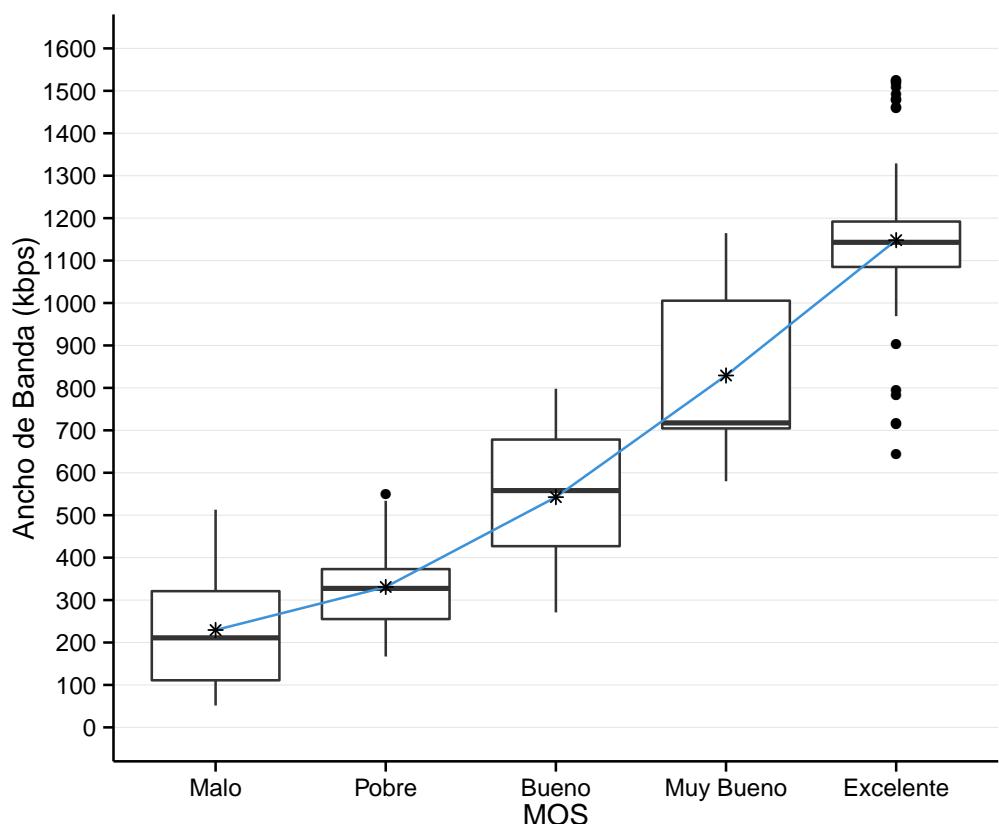


Figura. E.1 Variación del ancho de banda de acuerdo al MOS para el contexto de video streaming.

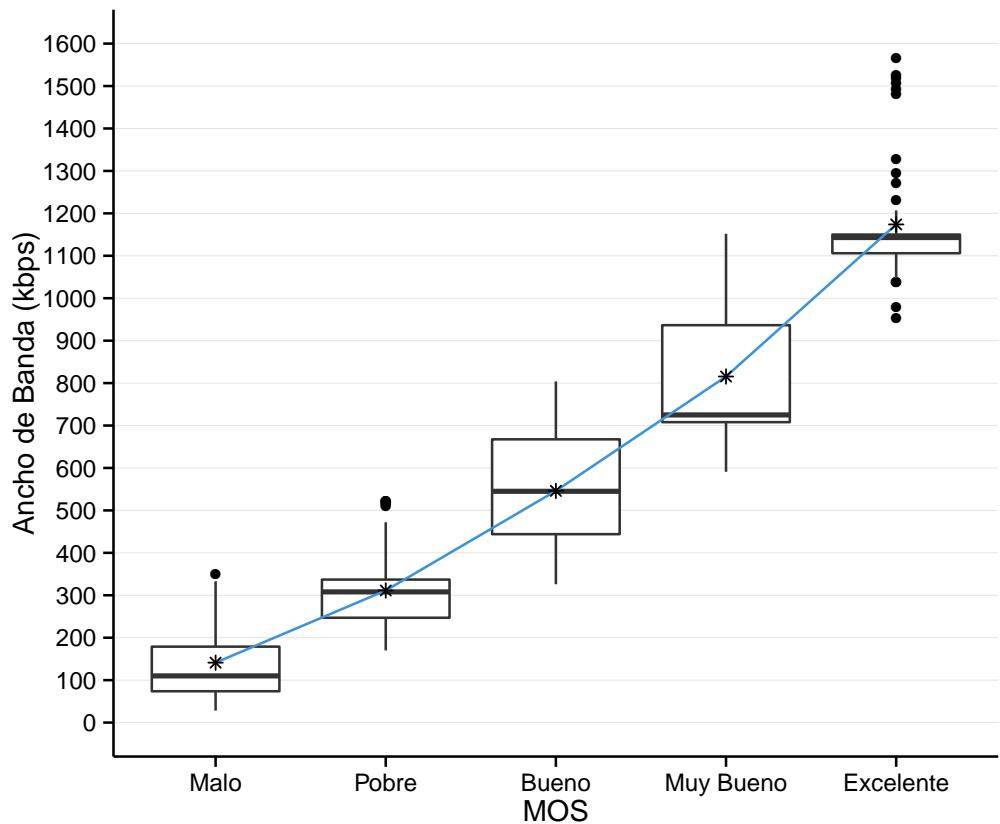


Figura. E.2 Variación del ancho de banda de acuerdo al MOS para el contexto de navegación web.

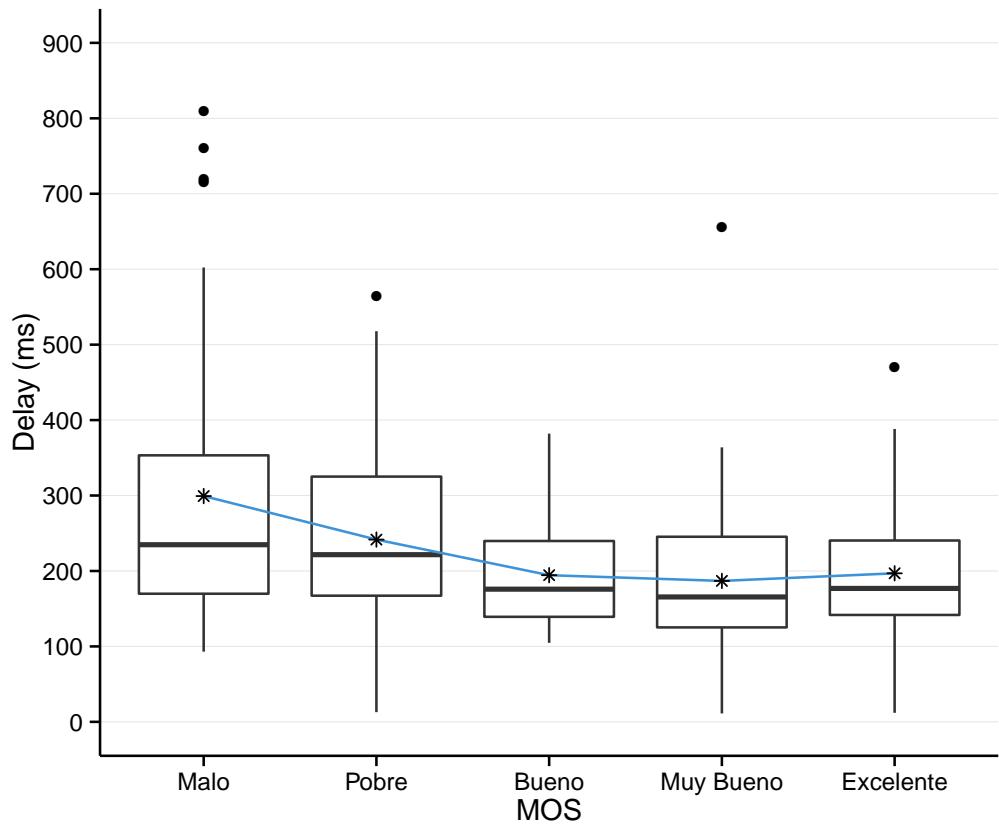


Figura. E.3 Variación del delay de acuerdo al MOS para el contexto de video streaming.

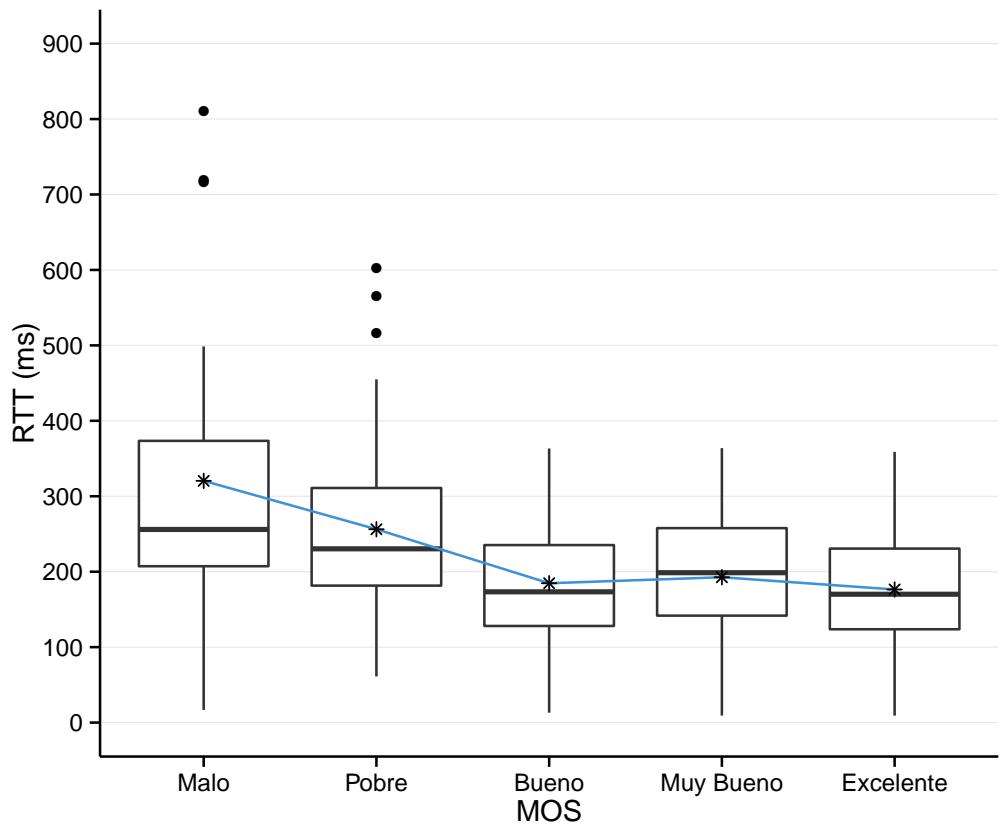


Figura. E.4 Variación del delay de acuerdo al MOS para el contexto de navegación web.

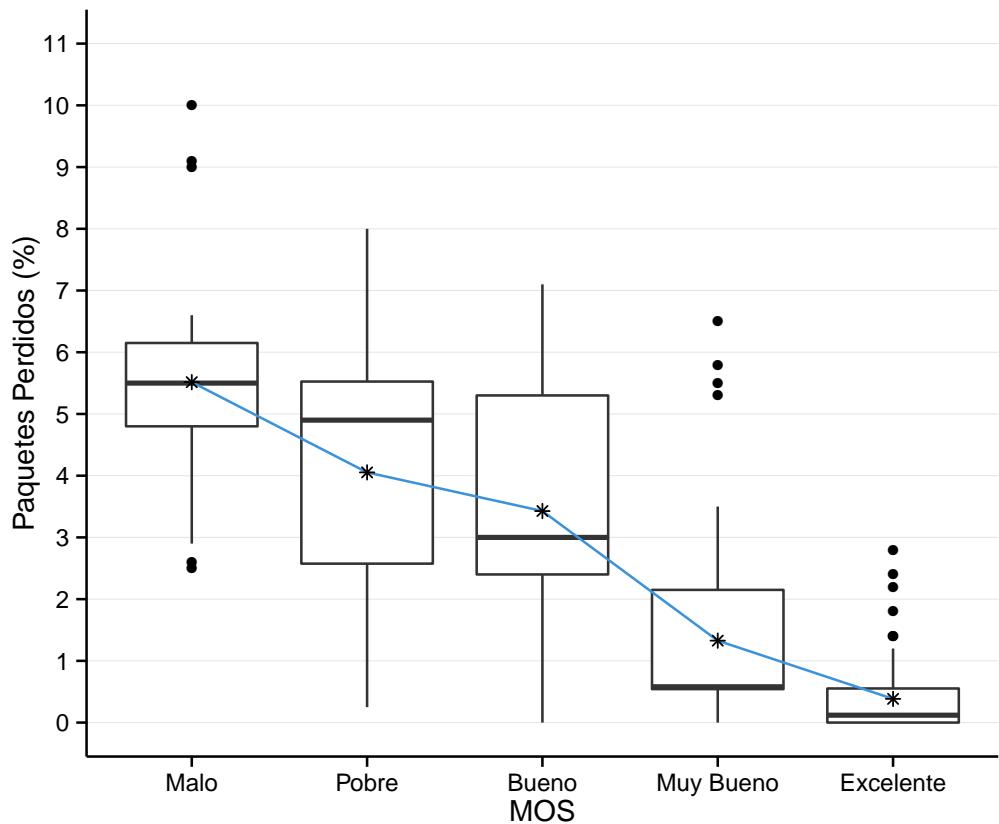


Figura. E.5 Variación de la perdida de paquetes de acuerdo al MOS para el contexto de video streaming.

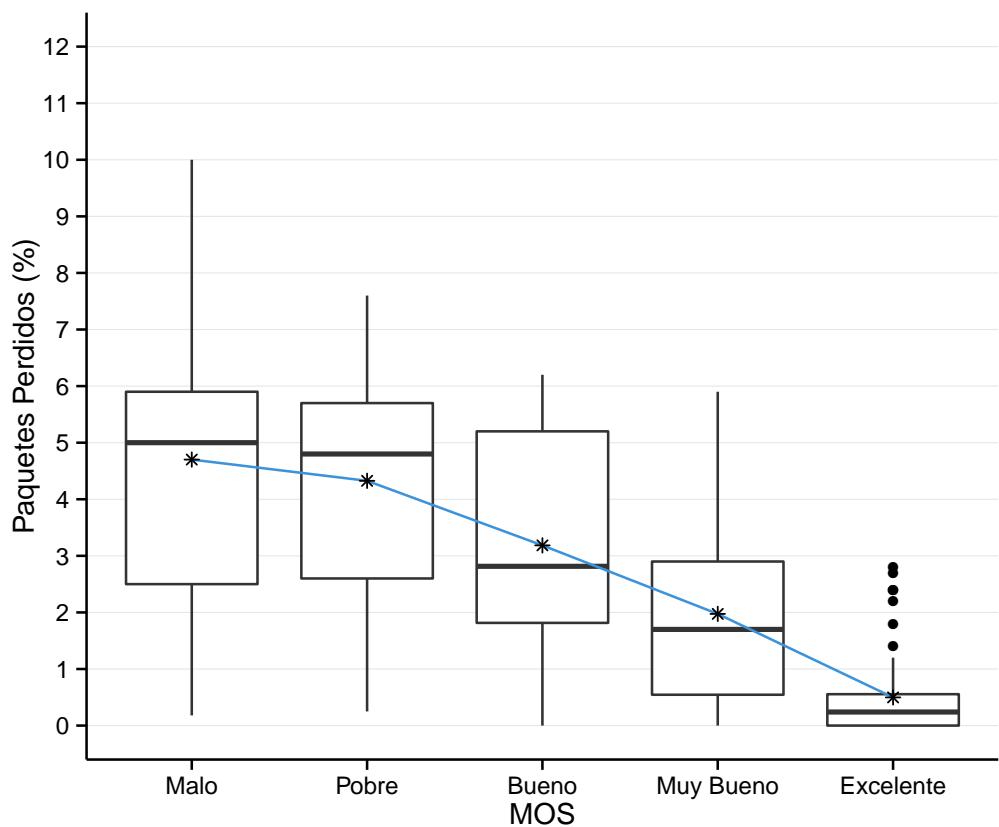


Figura. E.6 Variación de la perdida de paquetes de acuerdo al MOS para el contexto de navegación web.

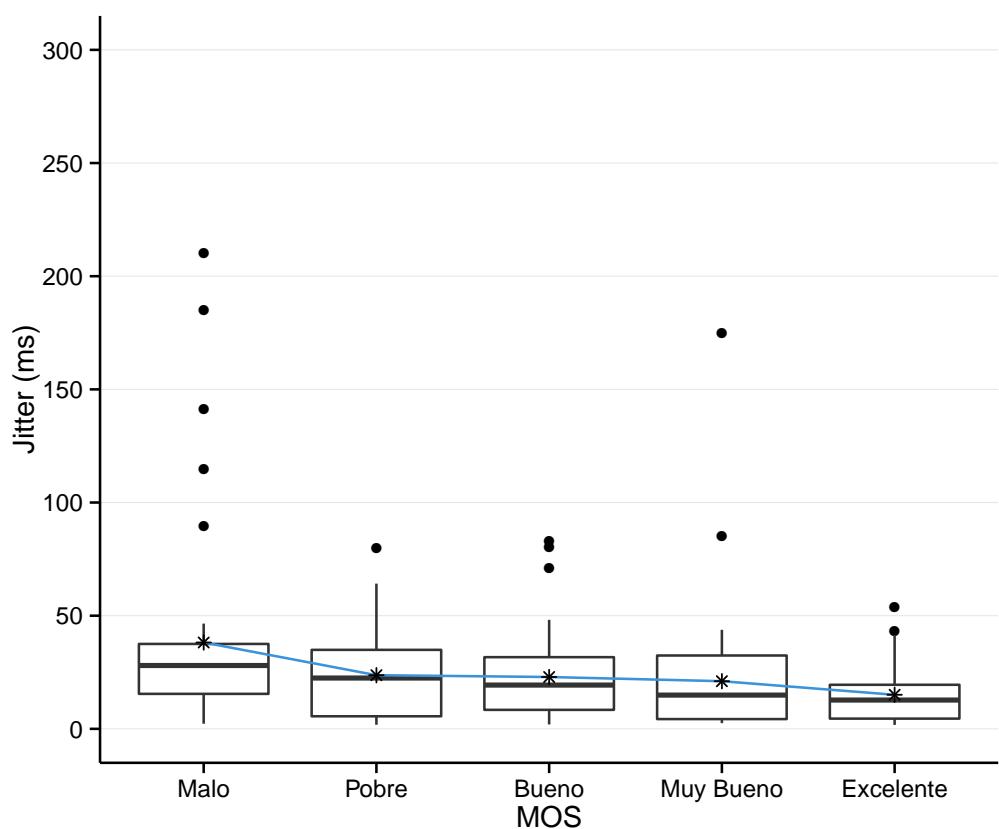


Figura. E.7 Variación del jitter de acuerdo al MOS para el contexto de video streaming.

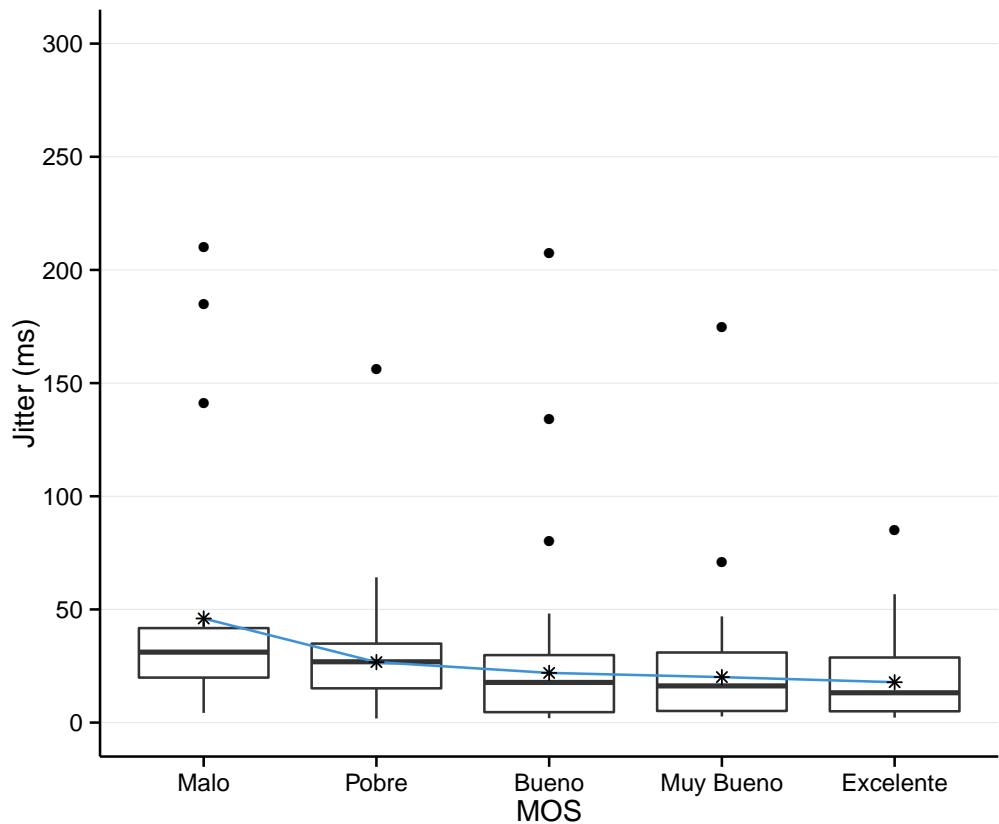


Figura. E.8 Variación del jitter de acuerdo al MOS para el contexto de navegación web.

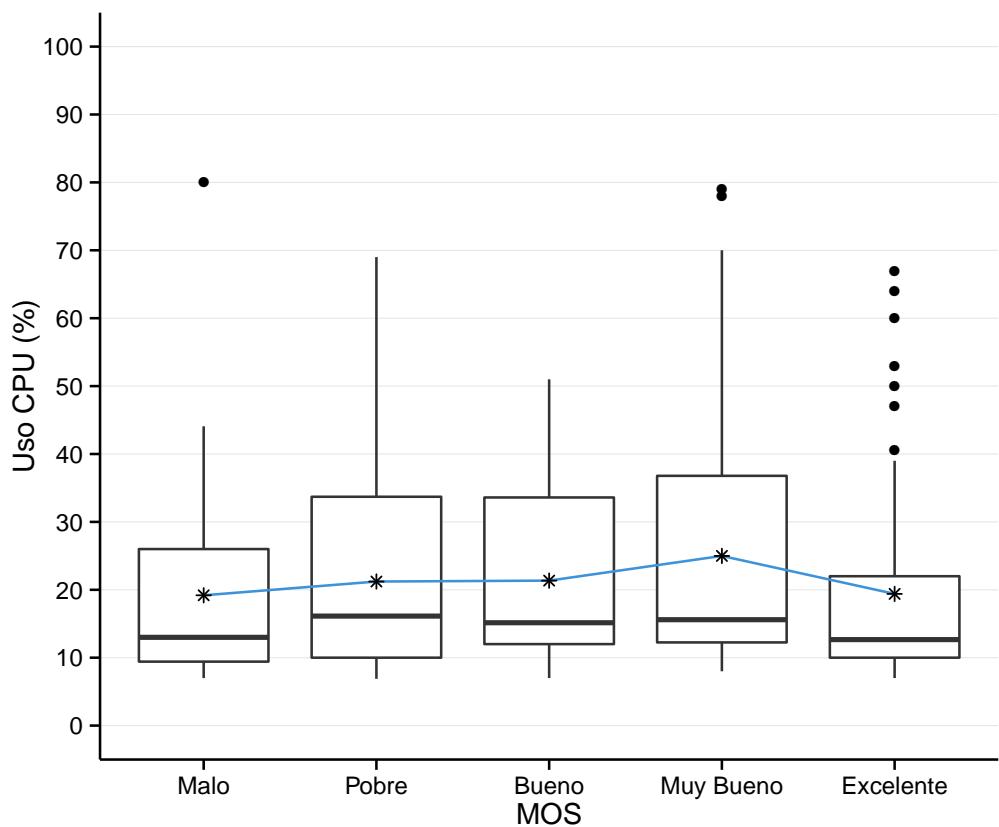


Figura. E.9 Variación del uso de CPU de acuerdo al MOS para el contexto de video streaming.

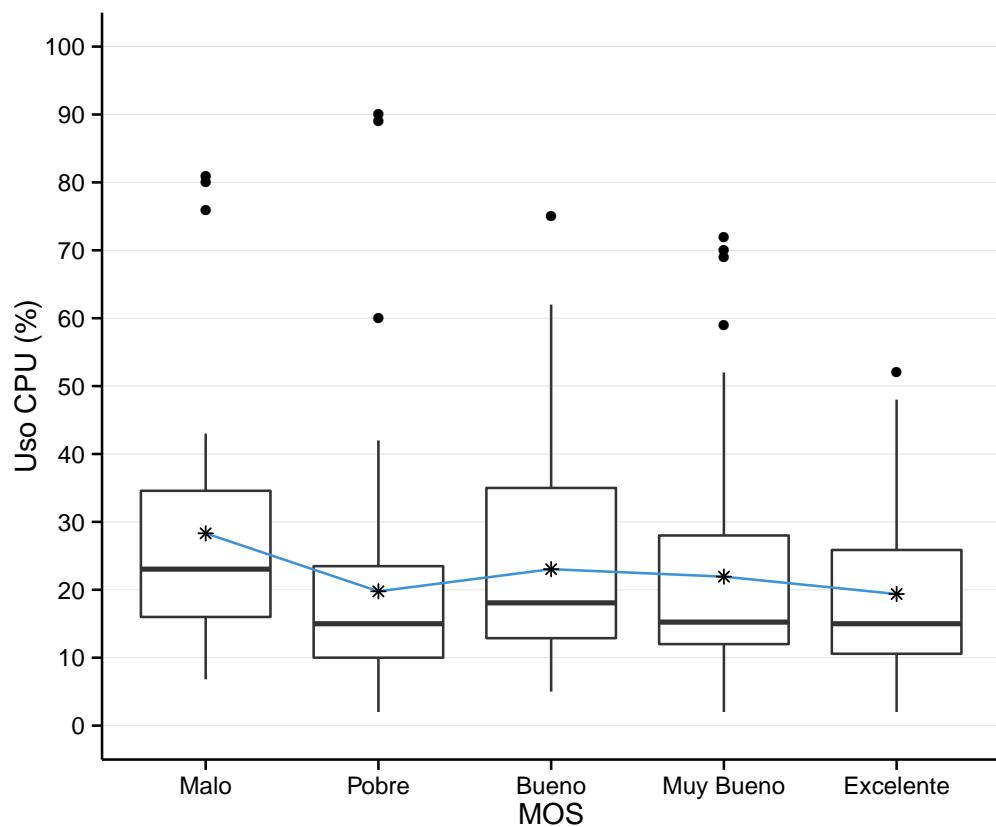


Figura. E.10 Variación del uso de CPU de acuerdo al MOS para el contexto de navegación web.

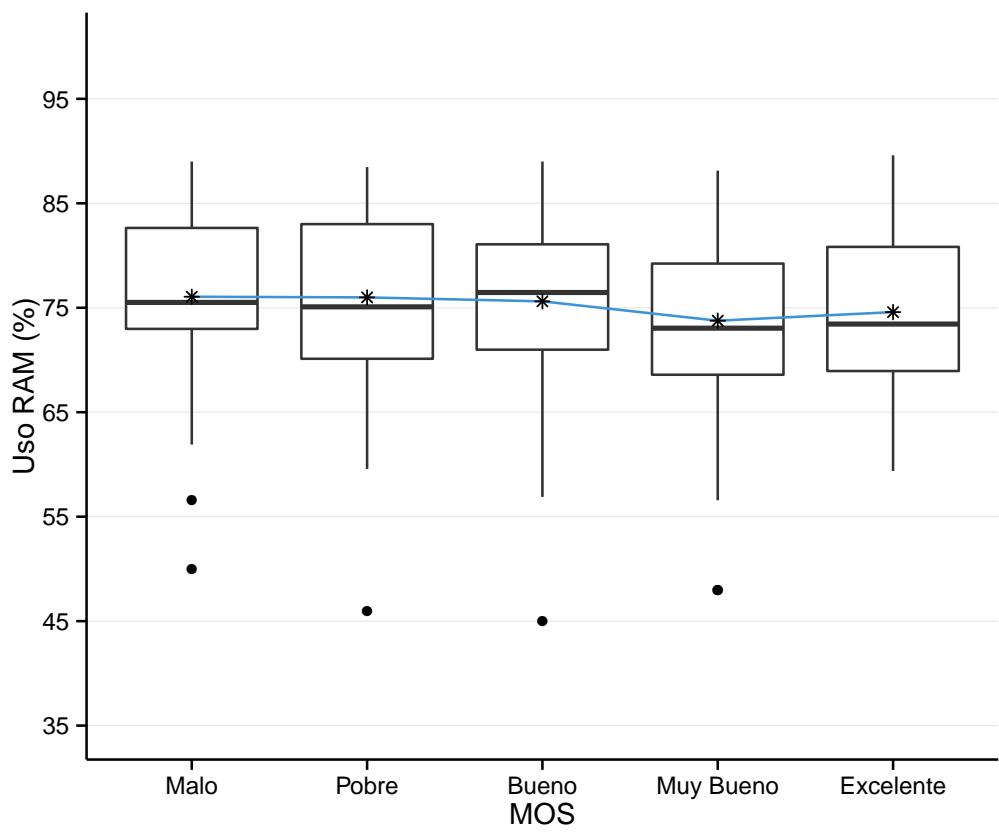


Figura. E.11 Variación del uso de RAM de acuerdo al MOS para el contexto de video streaming.

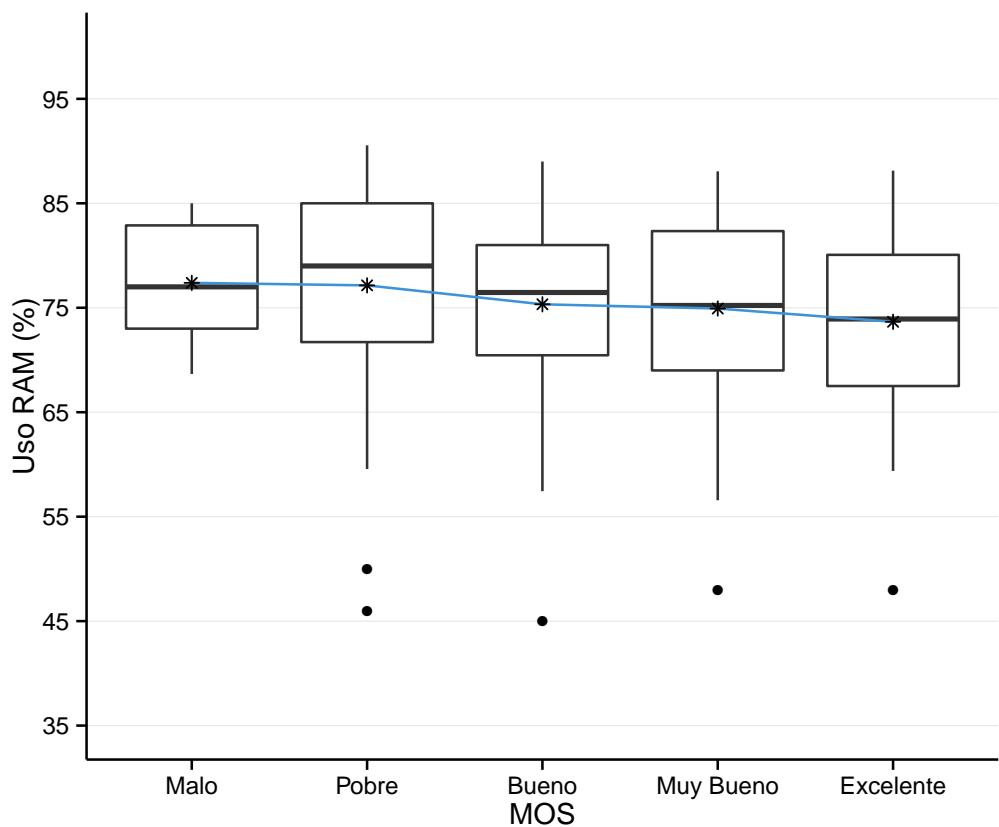


Figura. E.12 Variación del uso de RAM de acuerdo al MOS para el contexto de navegación web.

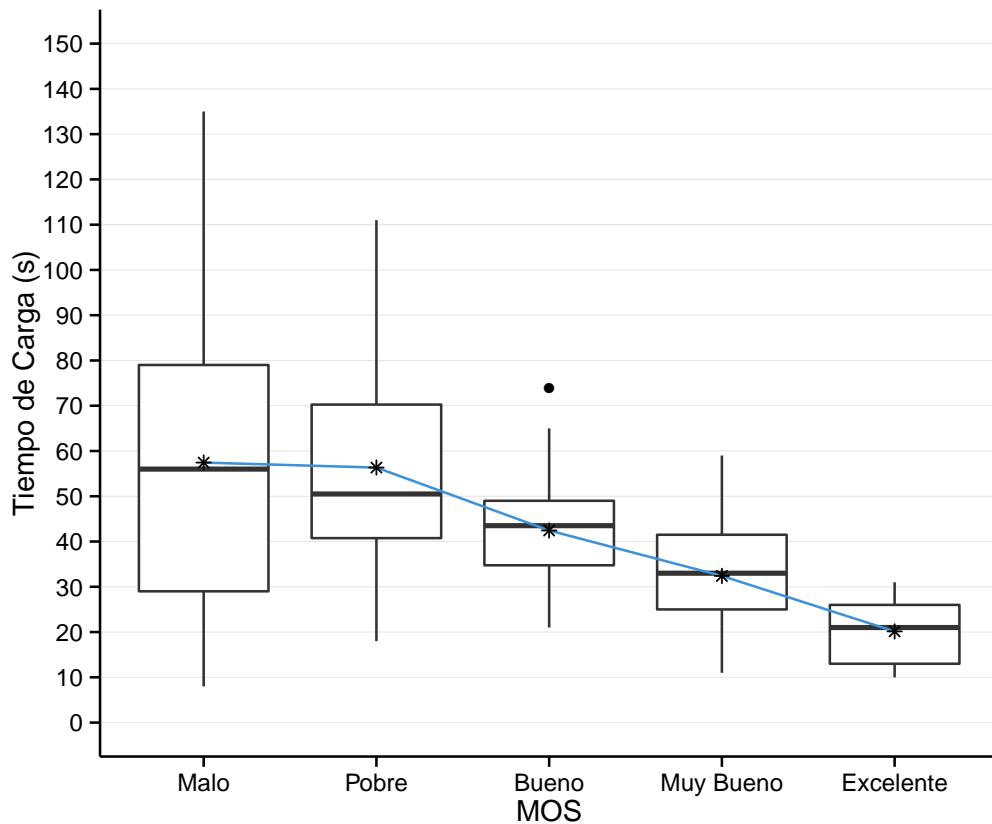


Figura. E.13 Variación del tiempo de carga de página de acuerdo al MOS para el contexto de navegación web.

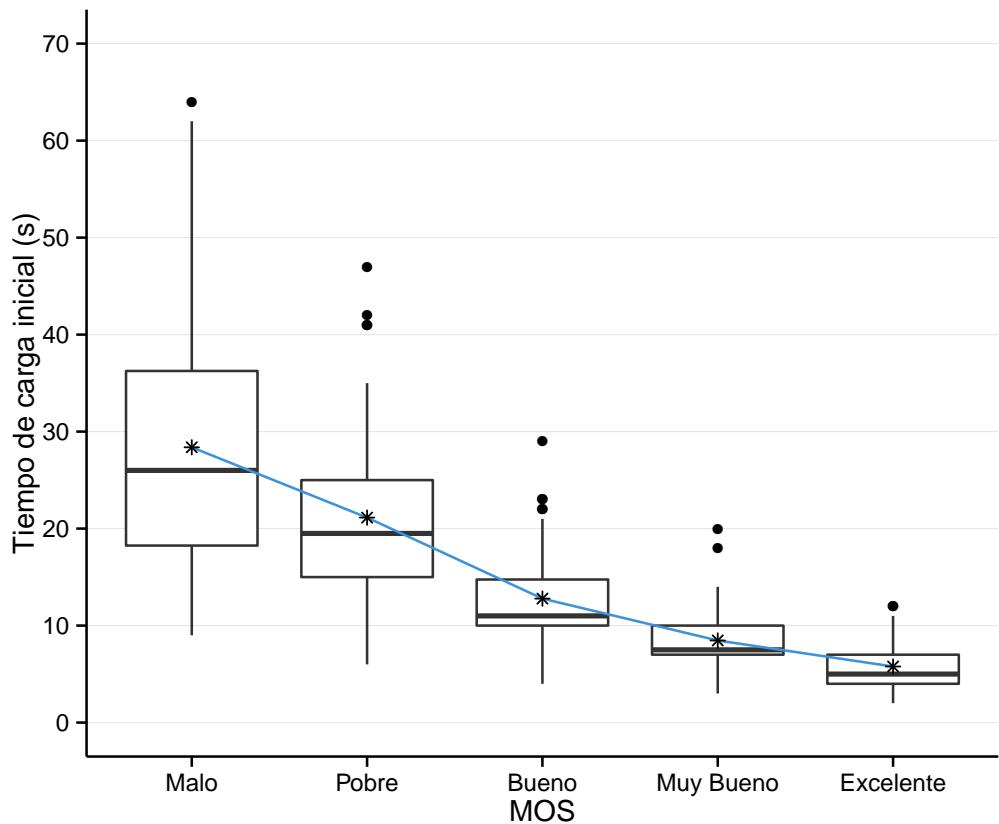


Figura. E.14 Variación de la carga inicial del video de acuerdo al MOS para el contexto de video streaming.

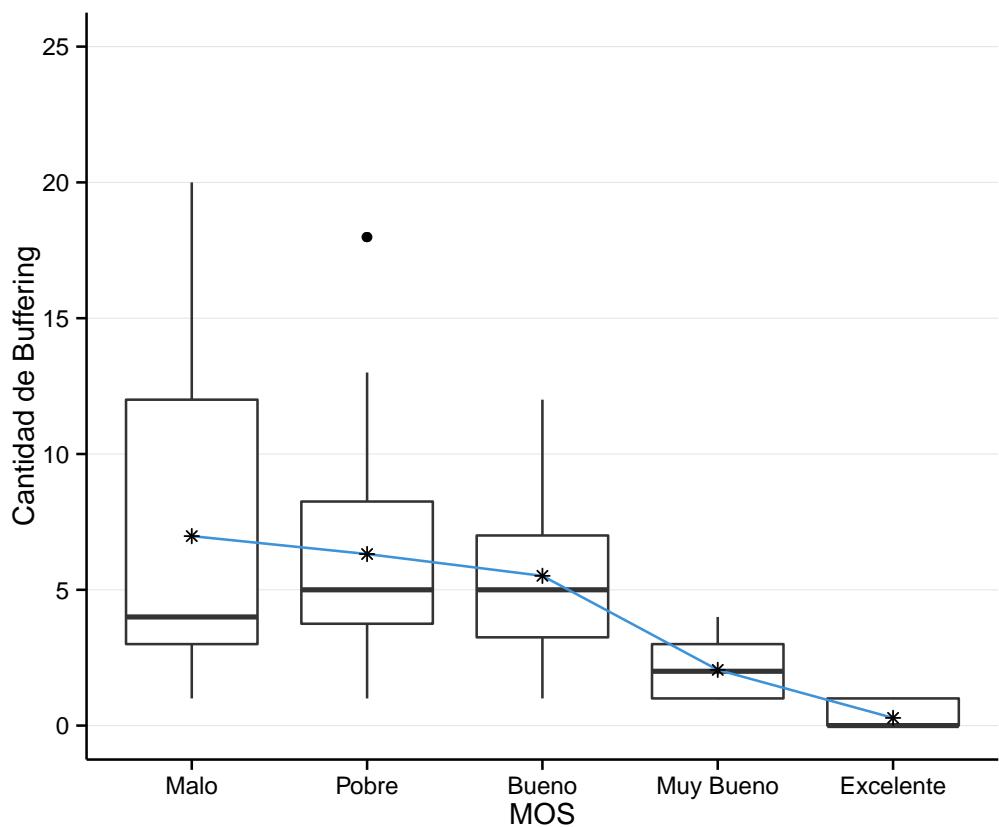


Figura. E.15 Variación de la cantidad de eventos de buffering de acuerdo al MOS para el contexto de video streaming.

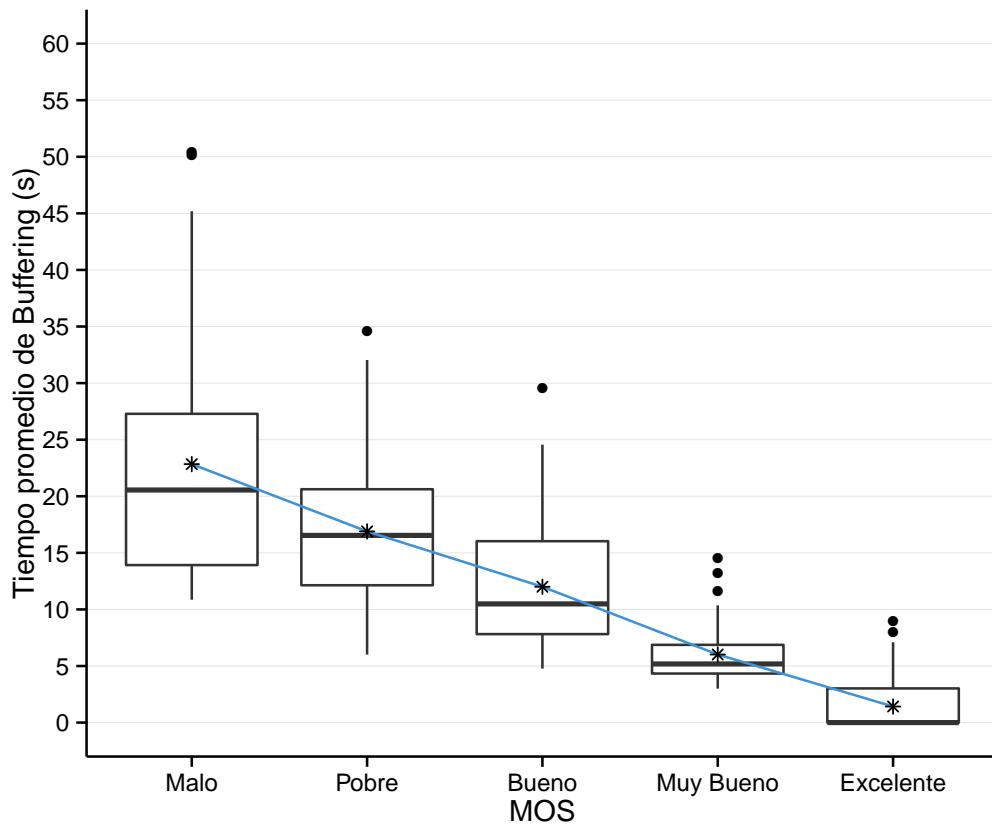


Figura. E.16 Variación del tiempo promedio de eventos de buffering de acuerdo al MOS para el contexto de video streaming.