

**APLICACIÓN MÓVIL PARA DIAGNOSTICAR LOS FACTORES DE DESERCIÓN ESTUDIANTIL
EN LA UNIVERSIDAD DE PAMPLONA A TRAVÉS DE MÉTODOS DE RECUPERACIÓN
DE INFORMACIÓN Y NUEVAS TECNOLOGÍAS**

HAIDER ANDRES MIRANDA MENDEZ



UNIVERSIDAD DE PAMPLONA

FACULTAD DE INGENIERÍAS Y ARQUITECTURA

**DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, SISTEMAS Y
TELECOMUNICACIONES**

INGENIERÍA DE SISTEMAS

VILLA DEL ROSARIO

2016

**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, SISTEMAS Y
TELECOMUNICACIONES
INGENIERÍA DE SISTEMAS.**

**TRABAJO PRESENTADO PARA OPTAR POR ÉL TÍTULO DE
INGENIERO DE SISTEMAS**

**TÍTULO: APLICACIÓN MÓVIL PARA DIAGNOSTICAR LOS FACTORES DE DESERCIÓN
ESTUDIANTIL EN LA UNIVERSIDAD DE PAMPLONA A TRAVÉS DE MÉTODOS DE
RECUPERACIÓN DE INFORMACIÓN Y NUEVAS TECNOLOGÍAS**

**FECHA DE INICIO DEL TRABAJO: FEBRERO DE 2016
FECHA DE TERMINACIÓN DEL TRABAJO: MAYO DE 2016**

NOMBRES Y FIRMAS DE AUTORIZACIÓN PARA SUSTENTAR:

**HAIDER ANDRES MIRANDA MENDEZ
AUTOR**

**MSC. PEDRO JOSÉ ARIZA ACEVEDO
DIRECTOR**

**ING-ESP. JOHANN LEONARDO LATORRE JAIMES
DIRECTOR DEL PROGRAMA**

JURADO CALIFICADOR:

**PhD. JOSE GERARDO CHACON
PRESIDENTE**

**MSC. ANDERSON SMITH FLÓREZ FUENTES
OPONENTE**

**ING-ESP JOHANN LEONARDO LATORRE JAIMES
SECRETARIO**

DEDICATORIA

A mi Dios todo poderoso creador de la vida, por ser mi fortaleza mi guía y mi apoyo a lo largo de mi carrera universitaria.

A mi tío papa Noel Segundo Castro Torres D.E.P, que a lo largo de su vida se esmeró por sacar adelante toda la familia, a mi tía mama Roció Miranda que ha sido mi apoyo y motivación terrenal, a mis Padres Pablo Sir y Alba Luz que a pesar de las dificultades siempre han estado a mi lado, a mis hermanos Noel, Rejo, Maye, Braider, Brenda, Aida, Carmen, Maria, Cristian, a mis sobrinos que son mi mayor alegría Camilo, Lauren, Andres, maria, Moises, Rey, issan, a mis abuelos Rosa y Pablo, Mañe y Theolinda y demás familiares.

A mis compañeros de Carrera Ivon y Hubert, a mis profesores de la facultad de Ingenierías y Arquitectura gracias por su apoyo y colaboración.

A mi amiga hermana Tatiana Medina que ha sido mi apoyo y guía a lo largo de mi carrera, a doña Flor a Julián, al Doctor Agenor Mora y familia y demás personas que me acompañaron en este proceso.

A mi amigo incondicional German y su familia, A Juan Manuel, Gerardo Torres gracias por la orientación brindada.

A mis compañeros de trabajo Eduardo, David, Lili y Cami gracias por su apoyo incondicional.

AGRADECIMIENTOS

A Dios creador de la vida, mi familia, mis amigos y compañeros de trabajo, a mi mai que ha estado conmigo incondicional mente.

Agradezco profundamente a La Universidad de Pamplona, mi centro de formación académica, a mi director Pedro Ariza por el apoyo y colaboración en el desarrollo del proyecto de grado, y demás profesores de la institución.

A Las personas que me acompañaron en mi ciclo estudiantil en Pamplona y Villa del Rosario.

RESUMEN

La finalidad de este trabajo ha sido el desarrollar una Aplicación Móvil Para Diagnosticar Los Factores De Deserción Estudiantil En La Universidad De Pamplona A Través De Métodos De Recuperación De Información Y Nuevas Tecnologías, que proporcione las funcionalidades que se requiere para detectar los factores de deserción estudiantil más influyentes en las facultades de la Universidad de Pamplona.

Para este propósito, se dividió el proyecto en dos etapas, La primera implementa el Método Vectorial, utilizado en la recuperación de información con la finalidad de crear un modelado del perfil del estudiante de la Universidad de Pamplona, el cual asociado a unos factores de deserción estudiantil. Este modelado se construye en base a la información suministrada por la oficina de Planeación de la Universidad de Pamplona y registrada en el Sistema para la Prevención de la Deserción en las Instituciones de Educación Superior (SPADIES). A partir de este, se puede conocer que factores de deserción afectan a los estudiantes. La segunda etapa ha sido el de analizar las plataformas móviles existentes, así como las tecnologías que podrían ayudarnos a desarrollar la aplicación, para posteriormente tomar decisiones. Finalmente, el resultado que hemos obtenido ha sido aplicación móvil capaz de realizar, de forma satisfactoria, la tarea para la que fue diseñada.

Palabras claves: recuperación de información, deserción estudiantil, SPADIES, aplicación móvil, JQuery Mobile, Android, PhoneGap.

ABSTRACT

The purpose of this work was to develop a mobile application to diagnose Factors Dropout Student At University Of Pamplona Through Methods Information Retrieval and New Technologies, which provides the functionality required to detect factors dropout most influential in the faculties of the University of Pamplona.

For this purpose, the project was divided into two stages, the first implements the Vector method used in information retrieval in order to create a modeling student of University Of Pamplona, which associated with some factors dropout profile. This modeling is built based on the information provided by the Office of Planning of the University of Pamplona and registered in the system for Dropout Prevention in Higher Education Institutions (SPADIES). From this, you can know which factors affect students dropping out. The second step was to analyze the existing mobile platforms and technologies that could help us develop the application, to subsequently make decisions in this regard. Finally, the result we have obtained mobile application has been able to perform satisfactorily, the task for which it was designed.

Keywords: information retrieval, dropout, SPADIES, mobile application, JQuery Mobile, Android, PhoneGap.

TABLA DE CONTENIDO

INTRODUCCION	17
DESCRIPCION DEL PROBLEMA	19
JUSTIFICACION	20
OBJETIVOS	21
OBJETIVO GENERAL	21
OBJETIVOS ESPECIFICOS	21
CAPITULO 1	22
1. MARCO REFERENCIAL	22
1.1 ANTECEDENTES	22
1.1.1 Deserción Estudiantil	22
1.1.2 Deserción Estudiantil En Colombia	23
2. MARCO CONCEPTUAL	25
2.1 Dispositivos Móviles	25
2.1.1 Clasificación de los dispositivos móviles	26
2.1.2 Smartphone	28
2.2 Aplicativo Móvil	29
2.2.1 Importancia	30
2.2.2 Impacto de Aplicativos Móviles	30
2.2.3 Ventajas de Uso	30
2.3 Sistemas Operativos Para Dispositivos Móviles	31
2.3.1 Android	32
2.3.1.1 Breve Historia	32
2.3.1.2 Versiones	33
2.3.1.3 Características	33
2.3.2 IOS	34
2.3.2.1 Historia	35
2.3.2.2 Características	36
2.3.3 Windows Phone	36

2.3.3.1	Breve Historia	36
2.4	HTTP	37
2.5	HTML5	38
2.5.1	Nuevos Elementos	39
2.5.2	CSS 3	40
2.5.3	Web App	40
2.5.4	Mobile Web	41
2.5	PHP	41
2.6	MySQL	42
2.7	Editores de Código	43
2.7.1	Sublime Text 3	43
2.7.2	Brackets	44
2.7.3	Notepad++	44
2.8	Frameworks	45
2.8.1	Frameworks de Programación en Android	45
2.8.1.2	API nativa de Android por Google	46
2.8.1.3	Resumen General	50
2.8.2	AppInventor	51
2.8.2.1	Resumen General	54
2.8.3	PhoneGap	55
2.8.3.1	Resumen General	56
2.8.4	Jquery Mobile	57
2.8.4.1	Resumen General	58
2.8.5	Otros frameworks	60
2.9	Recuperación De Información	61
2.9.1	Modelos de RI	61
2.9.1.2	Modelo Booleano	62
2.9.1.3	Modelo Vectorial	64
2.9.1.4	Modelo Probabilístico	66
CAPITULO 2		68

3.	MARCO METODOLÓGICO	68
3.1	Diseño de Estudio	68
3.2	Deserción Estudiantil en la Universidad de Pamplona	69
3.2.1	Facultad de Ciencias Agrarias	72
3.2.2	Facultad de Artes y Humanidades	73
3.2.3	Facultad de Ciencias Básicas	74
3.2.4	Facultad de Ciencias Económicas y Empresariales	75
3.2.5	Facultad de Educación	76
3.2.6	Facultad de Ingenierías y Arquitectura	77
3.2.7	Facultad de Salud	78
3.3	Factores de Deserción en la Universidad de Pamplona	79
3.3.1	Facultad de Ciencias Agrarias	79
3.3.2	Facultad de Artes y Humanidades	80
3.3.2	Facultad de Ciencias Básicas	81
3.3.2	Facultad de Ciencias Económicas y Empresariales	82
3.3.2	Facultad de Educación	83
3.3.2	Facultad de ingenierías y Arquitectura	84
3.3.2	Facultad de Salud	85
3.4	Construcción del Perfil de las Facultades	86
3.4.1	Sistema para la Prevención de la Deserción en las Instituciones de Educación Superior (SPADIES)	87
3.4.2	Base de Datos propuesta FDEUPA	88
3.4.3	proceso de aplicación del Modelo Vectorial	89
3.4.3.1	Aplicación del M.V	90
3.4.4	Resultado Perfil Del Estudiante	93
3.5	Diseño del Aplicativo	94
3.5.1	Entorno de Desarrollo	94
3.5.1.1	Entorno del Hardware	94
3.5.1.2	Configuración Entorno del Software	95
3.5.1.3	Arquitectura	98

3.5.1.4 Contenido	100
3.5.1.5 Arquitectura de la Aplicación	100
3.5.1.6 Módulos De Aplicación	102
3.5.1.7 Interfaces Gráficas	103
CAPITULO 3	106
4. PRUEBAS Y RESULTADOS	106
4.1 Perfil de la Facultad	106
4.1.1 Factor Bajo Rendimiento Académico	107
4.1.2 Factor Social	108
4.1.3 Factor Económico	109
4.1.4 Factor Conocimientos Escolares Adquiridos	110
4.1.5 Factor Áreas De Conocimiento	111
4.1.6 Cambio De Lugar De Residencia	112
4.1.7 Institucional	113
CONCLUSIONES	114
RECOMENDACIONES	115
BIBLIOGRAFIA	116

LISTA DE GRAFICOS

Grafico 1: : Número de estudiantes matriculado por facultad de la UPA desde el I Sem. 2014 hasta el II Sem. 2014.....	70
Grafico 2: : Número de estudiantes desertores por facultad de la UPA desde el I Sem. 2014 hasta el II Sem. 2014.....	71
Grafico 3: Deserción en la facultad de Ciencias Agrarias UPA desde el I Sem. 2010 hasta el II Sem. 2014	72
Grafico 4: Deserción en la facultad de Artes y Humanidades UPA desde el I Sem. 2010 hasta el II Sem. 2014.....	73
Grafico 5: Deserción en la facultad de Ciencias Básicas UPA desde el I Sem. 2010 hasta el II Sem. 2014	74
Grafico 6: : Deserción en la facultad de Ciencias Económicas y Empresariales UPA desde el I Sem. 2010 hasta el II Sem. 2014.....	75
Grafico 7: Deserción en la facultad de Educación UPA desde el I Sem. 2010 hasta el II Sem. 2014	76
Grafico 8: Deserción en la facultad de Ingenierías y Arquitectura UPA desde el I Sem. 2010 hasta el II Sem. 2014.....	77
Grafico 9: Deserción en la facultad Salud UPA desde el I Sem. 2010 hasta el II Sem.2014. 78	
Grafico 10: Factores de Deserción en la facultad de Ciencias Agrarias UPA desde el I Sem. 2014 y II Sem. 2014	79
Grafico 11: : Factores de Deserción en la facultad de Artes y Humanidades UPA desde el I Sem. 2014 y II Sem. 2014.....	80
Grafico 12: Factores de Deserción en la facultad de Ciencias Básicas UPA desde el I Sem. 2014 y II Sem. 2014	81
Grafico 13: Factores de Deserción en la facultad de Ciencias Económicas y Empresariales UPA desde el I Sem. 2014 y II Sem. 2014	82
Grafico 14: Factores de Deserción en la facultad de Educación UPA desde el I Sem. 2014 y II Sem. 2014	83
Grafico 15: Factores de Deserción en la facultad de Ingenierías y Arquitectura UPA desde el I Sem. 2014 y II Sem. 2014.....	84
Grafico 16: Factores de Deserción en la facultad de Salud UPA desde el I Sem. 2014 y II Sem. 2014	85

LISTA DE FIGURAS

Figura 1: Logo de Android	32
Figura 2: Porcentajes de distribuciones de Android utilizadas actualmente.....	33
Figura 3: Logo de IOS	34
Figura 4: Logo de Windows Phone.....	36
Figura 5: Logo de HTML5	38
Figura 6: : Logo de CSS3.....	40
Figura 7: Logo de PHP.....	41
Figura 8: Logo de MySQL.....	42
Figura 9: Logo de Sublime Text 3.....	44
Figura 10: Logo Brackets	44
Figura 11: Logo Notepad++	45
Figura 12: Icono de Android Developers.....	46
Figura 13: Interfaz de Eclipse para crear un nuevo proyecto en “Android”	47
Figura 14: : Carpetas creadas en el IDE para el manejo del proyecto en “Android”	47
Figura 15: Carpeta “src” del proyecto “Android” expandida	48
Figura 16: : Carpeta “gen” del proyecto “Android” expandida.....	48
Figura 17: Carpeta “res” del proyecto “Android” expandida.....	49
Figura 18: Icono de Android App Inventor	51
Figura 19: Interfaz del “App Inventor Designer”	53
Figura 20: Interfaz del “App Inventor Blocks Editor”	54
Figura 21: Icono del framework “PhoneGap”	55
Figura 22: Icono del framework “jQuery Mobile”	58
Figura 23: Modelo booleano: consulta ‘modelo AND booleano AND NOT vectorial’	64
Figura 24: Modelo vectorial: espacio tridimensional definido por el vocabulario {procesamiento, lenguaje, natural}.....	65
Figura 25: Construcción del perfil de la Facultad	87
Figura 26: : Sistema para la Prevención de la Deserción en las Instituciones de Educación Superior -SPADIES.....	88
Figura 27: Variables que utiliza SPADIES	89
Figura 28: Base de Datos FDEUPA	90
Figura 29: Diagrama de Flujo FDEUPA.....	91
Figura 30: PhoneGap Instalado en Eclipse	98
Figura 31: Directorio del Aplicativo Móvil FDEUPA.....	101
Figura 32: Capas del Aplicativo Móvil FDEUPA.....	102
Figura 33: Pagina de presentación de Datos del Aplicativo Móvil FDEUPA	103
Figura 34: Página de inicio del Aplicativo Móvil FDEUPA.....	105
Figura 35: reportes BD del Aplicativo Móvil FDEUPA.....	105

Figura 36: Reportes Cálculos del Aplicativo Móvil FDEUPA	106
Figura 37: : Modelado Factor Bajo Rendimiento Academico del Aplicativo Móvil FDEUPA	108
Figura 38: Modelado Factor Social del Aplicativo Móvil FDEUPA	109
Figura 39: Modelado Factor Económico del Aplicativo Móvil FDEUPA	110
Figura 40: : Modelado Factor Conocimientos escolares Adquiridos del Aplicativo Móvil FDEUPA.....	111
Figura 41: : Modelado Factor Áreas de Conocimiento del Aplicativo Móvil FDEUPA.....	112
Figura 42: Modelado Factor Cambio de Lugar de Residencia del Aplicativo Móvil FDEUPA	113
Figura 43: Modelado Factor Institucional del Aplicativo Móvil FDEUPA	114

LISTA DE TABLAS

Tabla 1: Ventas de teléfonos inteligentes en todo el mundo a usuarios finales por sistema operativo Fuente: Gartner (febrero de 2016)..	30
Tabla 2: Clasificación de los Modelos de Recuperación de Información según Dominich. .	60
Tabla 3: Matriz de frecuencia Estudiante vs Factor de Deserción	91
Tabla 4: Cálculo de IDF para los Factores de Deserción	91
Tabla 5: Cálculo de TF*IDF y Norma	92

LISTA DE ECUACIONES

Ecuación (1).....	64
Ecuación (2).....	64
Ecuación (3).....	65
Ecuación (4).....	66
Ecuación (5).....	91
Ecuación (6).....	92
Ecuación (7).....	92
Ecuación (8).....	92

INTRODUCCION

En la actualidad es imposible imaginarnos un mundo sin tecnologías de la información y de las comunicaciones (TIC). Su uso cada vez más generalizado ha mejorado la vida de mucha gente e impulsado el crecimiento económico en muchos países.

Las tecnologías de información se están renovando continuamente por lo que se ha visto la posibilidad de desarrollar aplicaciones para los dispositivos móviles utilizando nuevas tecnologías web como lo son HTML5, CSS3, JavaScript, JQuery Mobile y empaquetarlas a través de Frameworks como Phonegap, dentro de una aplicación nativa para la visualización, obteniendo como resultado aplicaciones híbridas que funcionan en los distintos sistemas operativos móviles.

Según estadísticas del Ministerio de Educación Nacional (MEN), de cada cien estudiantes que ingresan a una institución de educación superior cerca de la mitad no logra culminar su ciclo académico y obtener su titulación.[1] En la actualidad el MEN está incitando a todas las universidades del país a que realicen estudios y ajusten sus políticas de mantenimiento del estudiante para que este no deserte, debido a que los cupos que los estudiantes abandonan no pueden ser ocupados por otros estudiantes, sino que por el contrario se pierden, con las consecuencias que este hecho acarrea para la institución, además anima a las instituciones a mantener un buen índice de permanencia dentro de sus programas como un indicador de eficiencia y eficacia institucional.[1]

El grado de deserción estudiantil que se presenta en la Universidad de Pamplona según el reporte de Sistema de Prevención y Análisis de la Deserción en las Instituciones de Educación Superior SPADIES, (2014), es de un 53%, este porcentaje de estudiantes desertaron por causas académicas, sociales, institucionales, económicas, y culturales presentándose esta deserción en los primeros semestres del periodo (2014-1, 2014-2).

La recuperación de información (IR, Information Retrieval) es el área de la ciencia y la tecnología que trata de la adquisición, representación, almacenamiento, organización y acceso a elementos de información. Desde un punto de vista práctico, dada una necesidad de información del usuario, un proceso de IR produce como salida un conjunto de documentos cuyo contenido satisface potencialmente dicha necesidad. [2] La IR posee distintos métodos para realizar las búsquedas entre los más eficientes encontramos el Modelo Booleano, Probabilístico y Vectorial.

La finalidad de este proyecto es construir un aplicativo móvil que diagnostique los factores de deserción estudiantil en la Universidad de Pamplona, a través del modelado de perfiles de usuario utilizando métodos de recuperación de Información y nuevas tecnologías de desarrollo.

DESCRIPCION DEL PROBLEMA

El fenómeno de la deserción estudiantil es un problema de gran complejidad debido no solo al gran número de variables que intervienen sino también por la dificultad para describir al fenómeno. Durante mucho tiempo, la deserción universitaria fue considerada como un fenómeno normal e, incluso, como una muestra de la exigencia de la universidad y de la carrera particular; sin embargo, hoy se ve como un signo de ineficiencia y como un gran costo para el país, los estudiantes y, para las instituciones de educación superior, lo que pasó a convertirse en un problema que hay que entender para poder combatirlo.

Según cifras del ministerio de educación nacional de Colombia – MEN, demuestran que la deserción es un problema muy serio, el 52% de los estudiantes colombianos que empiezan una carrera universitaria, no la concluyen, pues entre los años 1999 y 2004, el 48% en promedio finalizaron sus estudios, mientras que para el año 2012, la deserción en el nivel universitario alcanzó el 45.3%, lo que significa que uno de cada dos estudiantes que ingresa a educación superior no culmina sus estudios.

En la Universidad de Pamplona sede Villa del Rosario no se cuenta con una aplicación móvil que permita realizar el análisis del fenómeno de deserción, estudiando sus factores de incidencia, lo cual dificulta determinar el posible comportamiento futuro de la misma.

JUSTIFICACION

Con el problema expuesto, en este proyecto se plantea desarrollar una aplicación móvil sobre plataforma Android, capaz de organizar los factores de deserción y modelar los perfiles de los estudiantes de la Universidad de Pamplona sede Villa del Rosario. Para crear la aplicación que servirá como herramienta para la evaluación institucional, generando posibles respuestas que permitan mejorar la eficacia y la eficiencia de la institución en lo que concierne a deserción estudiantil universitaria. Conocer el problema de la deserción es un paso adicional para seguir actuando sobre él, ya que la Universidad de Pamplona no ha sido ajena al mismo y viene trabajando con un programa asistencial específico para disminuir los niveles de deserción estudiantil.

La creación de aplicaciones para dispositivos móviles es un campo del desarrollo Web con una creciente demanda en la actualidad. La aparición de distintos estándares de programación, tales como HTML5, CSS3, jQuery Mobile y Javascript ha facilitado el desarrollo y formación de aplicaciones para dispositivos que permiten múltiples formas de interactuar con los usuarios. Al utilizar herramientas como PhoneGap, se extiende el campo de acción, posibilitando el acceso a la creación de aplicaciones que funcionan de la misma manera que las aplicaciones nativas de cada sistema como Android, iOS, RIM, Windows Phone, Symbian y WebOS.

OBJETIVOS

OBJETIVO GENERAL

Construir una aplicación móvil para diagnosticar los factores de deserción estudiantil en la Universidad de Pamplona a través de métodos de recuperación de información y nuevas tecnologías.

OBJETIVOS ESPECIFICOS

- Estudiar e Identificar los factores de deserción más relevantes en las Universidades de Colombia
- Adaptar el modelo espacio vectorial utilizado en la recuperación de información para la creación de perfil de los estudiantes de la Universidad de Pamplona.
- Modelar cada uno de los módulos adaptables al sistema usando nuevas tecnologías.
- Implementar los módulos adaptables al sistema sobre plataforma Android aplicando técnicas como lo son las librerías JQuery para la optimización de las vistas y código.

CAPITULO 1

1. MARCO REFERENCIAL

1.1 ANTECEDENTES

1.1.1 Deserción Estudiantil

Las primeras investigaciones en el tema de la deserción estudiantil tomaron como base conceptual la teoría del suicidio de Durkheim (1897), y los análisis costo-beneficio de la educación desde una perspectiva económica. En la primera aproximación se toma a la deserción como análoga al suicidio en la sociedad, de ahí que los centros de educación superior se consideren como un sistema que tiene sus propios valores y estructura social (Spady, 1970), donde es razonable esperar que bajos niveles de integración social aumenten la probabilidad de desertar. Sin embargo, estas investigaciones no pasaron de ser estudios longitudinales y cualitativos que abordaron el problema desde una perspectiva individual (integración social) y algunos factores externos que pudieran afectarla (usos alternativos del dinero invertido en educación).

De ahí en adelante, los desarrollos en el tema de la deserción estudiantil dividieron la investigación en aquellos estudios que apuntaban a la profundización teórica del problema y los interesados en encontrar las causas del fenómeno a través de la evidencia empírica. Aunque actualmente la definición de deserción estudiantil continúa en discusión, existe consenso en precisarla como un abandono que puede ser explicado por diferentes categorías de variables: socioeconómicas, individuales, institucionales y académicas. Sin embargo, la forma de operacionalizarlas depende del punto de vista desde el cual se haga el análisis; esto es, individual, institucional y estatal o nacional. Autores como Tinto (1989) afirman que el estudio de la deserción en la educación superior es extremadamente complejo, ya que implica no sólo una variedad de perspectivas sino también una gama de diferentes tipos de abandono. Adicionalmente, sostiene que ninguna definición puede captar en su totalidad la complejidad de este

fenómeno, quedando en manos de los investigadores la elección de la aproximación que mejor se ajuste a sus objetivos y al problema por investigar. [3]

1.1.2 Deserción Estudiantil En Colombia

La deserción estudiantil es un aspecto sumamente preocupante y a la vez importante para el desarrollo educativo, cultural y social de una región. Colombia es uno de los países que se ve seriamente afectado por este hecho, razón por la cual en las últimas épocas se le ha prestado mayor atención a dicho aspecto y se le ha hecho un seguimiento desde varios puntos de vista.

Un estudio sobre la deserción universitaria de los estudiantes colombianos sostiene que el 52% de los estudiantes universitarios comienzan una carrera universitaria, y no la concluyen, la aterradora cifra de deserción universitaria fue calificada como una verdadera "tragedia nacional" y como una muestra de la "monumental crisis de la educación superior". Según el Ministerio de Educación, de las promociones de estudiantes que terminaron estudios entre 1999 y el 2004, el 48% en promedio finalizaron sus estudios. Es decir, de cada dos estudiantes que se matriculan en un programa de pregrado, solo uno culmina su carrera.

La preocupación es mayor si se tiene en cuenta que el 39,52% de quienes abandonan sus estudios lo tienen que hacer por razones económicas. Otro factor inquietante es la mala calidad de los programas en los centros de formación superior, programas que están a punto de ser cerrados por el incumplimiento de requisitos de calidad. [4]

Desde 2002 – con la Ley 749 de ese año – el Ministerio de Educación Nacional propuso disminuir la deserción estudiantil en la educación superior como parte de la estrategia planteada para aumentar la cobertura, la calidad y la eficiencia educativas. Entre otras acciones, puso en marcha el Sistema para la Prevención de la Deserción en la Educación Superior —SPADIES— ante la necesidad de contar con una visión sectorial e integrada de la problemática de la deserción, a partir de la cual se pudiera disponer de una conceptualización, una medición y una metodología de seguimiento del fenómeno aplicables a todas las Instituciones de Educación Superior – IES– del país. En 2004 se hizo

un concurso público de méritos que fue otorgado al Centro de Estudios sobre Desarrollo Económico –CEDE– de la Universidad de los Andes. Este último desarrolló la herramienta y la implementó desde noviembre de 2004 hasta noviembre de 2011, y la instaló en las instituciones de educación superior. Hubo cuatro interventorías: las dos primeras estuvieron a cargo de la Universidad de Antioquia y las dos últimas fueron realizadas por el Ministerio de Educación Nacional. A lo largo de los años, a la herramienta se le han hecho ajustes con el fin de mejorar la calidad y la utilidad de la información obtenida. Por ejemplo, si bien la razón fundamental para la existencia del SPADIES es evitar la deserción estudiantil, también se ha visto la necesidad de identificar casos de rezago estudiantil, es decir, estudiantes que no desertan, sino que permanecen matriculados sin graduarse.

Actualmente el SPADIES está instalado en el 99% de las instituciones de educación superior del país y hace seguimiento a más de 3 millones de estudiantes. [5]

2. MARCO CONCEPTUAL

Desde el punto de vista institucional, todos los estudiantes que abandonan su educación superior pueden ser clasificados como desertores, por tanto, muchos autores asocian la deserción con los fenómenos de mortalidad académica y retiro forzoso. Se define deserción para el caso, como una situación a la que se enfrenta un estudiante cuando aspira y no logra concluir su proyecto educativo, considerándose como desertor a aquel individuo que siendo estudiante de una institución de Educación Superior no continua su actividad académica durante tres semestres académicos de la carrera. Investigaciones sobre este tema, denominan como “primera deserción” el fenómeno que sucede cuando un estudiante abandona sus estudios durante los tres primeros semestres de su carrera, ya que no se puede determinar si pasado este período de tiempo el individuo retomará o no sus estudios o si decidirá iniciar otro Programa Académico.

Se pretende desarrollar un aplicativo móvil que pueda estar aplicado como herramienta técnica para detectar cuáles son los factores de deserción en la Universidad de Pamplona sede villa del rosario, se tratan a continuación los conceptos que tienen que ver con el desarrollo de software móvil Los cuales darán la pauta como los estándares utilizados tanto en la investigación, en el análisis, diseño, implementación, pruebas y mantenimiento de la aplicación.

2.1 Dispositivos Móviles

No existe un consenso claro a la hora de definir qué es realmente un dispositivo móvil y qué no lo es. Es frecuente que hoy en día este término se utilice para designar únicamente a ciertos modelos de teléfonos móviles con mayores o menores prestaciones. A pesar de ello, un dispositivo móvil no tiene por qué ceñirse solamente al ámbito telefónico. [6]

Buscando ser más rigurosos, se podría denominar dispositivo móvil a todo aparato electrónico que cumple unas características muy básicas:

- Es de reducido tamaño, haciéndolo fácil de transportar.

- Cuenta con una cierta capacidad de computación y almacenamiento de datos.
- Incorpora elementos de E/S básicos (por lo general, pantalla y/o algún tipo de teclado).

Más allá de estas características comunes, los dispositivos móviles forman en la actualidad un grupo sumamente heterogéneo y pueden incorporar casi cualquier componente de hardware y software que amplía y diversifica su función inicial. El más frecuente sin duda es la conexión telefónica (incluyendo servicios como el envío de SMS, MMS, y acceso WAP) o la conexión a Internet. [6]

Igualmente son habituales la cámara fotográfica y de vídeo, pantalla táctil, teclado QWERTY, receptor de radio, Bluetooth, conexión mediante infrarrojos, dispositivos de memoria extraíbles, localizador GPS, acelerómetro, etc. Desde el punto de vista del software, pueden incorporar también un amplio abanico de aplicaciones tales como programas ofimáticos, reproductores de audio y vídeo, organizadores, videojuegos, navegadores web o clientes de correo, entre otros. [6]

2.1.1 Clasificación de los dispositivos móviles

Al igual que ocurre a la hora de definir qué es un dispositivo móvil, la clasificación que se pueda hacer de estos aparatos está sujeta a diferentes valoraciones y a veces no existe un acuerdo amplio para ubicar un dispositivo móvil en una determinada familia. En la década de los 90, tras la aparición de estos primeros dispositivos, establecer clasificaciones más o menos rigurosas era posible debido a que cada aparato estaba claramente definido para una función determinada o para un público concreto. El aumento de las prestaciones y funcionalidades que en la actualidad puede ofrecer cualquier dispositivo móvil dificulta el poder agruparlo dentro de un conjunto determinado. Por ejemplo, un *smartphone* representa una evolución de un teléfono móvil tradicional, esto es, su cometido es ofrecer comunicación telefónica; sin embargo, cuenta con otros servicios adicionales como la conexión a Internet y aplicaciones, servicios propios de un ordenador, cámara de fotos y de vídeo o la posibilidad de reproducir películas o videojuegos. [6]

La clasificación que a aquí se propone utilizar como principal criterio la funcionalidad o servicio principal para la que ha sido diseñado bien el propio dispositivo móvil, o bien aquel dispositivo del que directamente procede y del que supone una evolución mejorada. Tal y como se deduce de todo lo anteriormente explicado, la pertenencia a una categoría concreta no implica en absoluto que el dispositivo no pueda ofrecer otras muchas características propias de otras categorías. [6]

Dicho lo anterior, los dispositivos móviles pueden ser clasificados en los siguientes grupos:

➤ **Dispositivo de comunicación**

Un dispositivo de comunicación es aquel dispositivo móvil cuyo cometido principal es ofrecer una infraestructura de comunicación, principalmente telefónica. Estos dispositivos ofrecen además servicios como el envío de mensajes SMS y MMS, o acceso WAP. En esta categoría se incluiría el tradicional teléfono móvil, precursor indiscutible dentro de los dispositivos móviles, la BlackBerry y el smartphone, que amplía considerablemente las prestaciones del primero mediante pantalla táctil, conexión a Internet o la ejecución de aplicaciones (por ejemplo, iPhone o HTC G1). [6]

➤ **Dispositivo de computación**

Los dispositivos de computación son aquellos dispositivos móviles que ofrecen mayores capacidades de procesamiento de datos y cuentan con una pantalla y teclado más cercanos a un ordenador de sobremesa. Dentro de este grupo encontramos a las PDA, muy populares a finales de los años 90 y que permitían al usuario disponer de un organizador mucho más completo que los ofrecidos por los teléfonos móviles del momento, e incluso en ocasiones la visualización de documentos o acceso a Internet. Por otro lado, dispositivo de computación también es un ordenador portátil o laptop, que dentro de los dispositivos móviles son sin duda los que mayores prestaciones hardware ofrecen (igualando o superando a los de sobremesa) pero también los que tienen, con diferencia, un mayor tamaño, peso y precio. Las calculadoras gráficas pueden ser igualmente incluidas en este grupo de dispositivos de computación. [6]

➤ **Reproductor multimedia**

Un reproductor multimedia es aquel dispositivo móvil que ha sido específicamente diseñado para proporcionar al usuario la reproducción de uno o varios formatos de datos digitales, ya sea audio, vídeo o imágenes. Dentro de estos dispositivos encontramos reproductores de MP3, los DVD portátiles, los eBooks, y en los últimos años los reproductores multimedia de la popular familia iPod de Apple, que ofrecen tanto audio y como vídeo. Estos dispositivos son con frecuencia los de más reducido tamaño y, junto a los teléfonos móviles y smartphones, los más extendidos. [6]

➤ **Grabador multimedia**

Dentro de los dispositivos móviles, un grabador multimedia es aquel dispositivo que posibilita la grabación de datos en un determinado formato digital, principalmente de audio y vídeo. En esta categoría se hallan las cámaras fotográficas digitales o las cámaras de vídeo digital.

➤ **Consola portátil**

Una consola portátil es un dispositivo móvil cuya única función es la de proporcionar al usuario una plataforma de juego. Las consolas portátiles fueron, junto a los teléfonos, los primeros dispositivos móviles en convertirse en un producto de masas. Hoy en día representan un importantísimo volumen de ventas dada su gran aceptación en la sociedad y son objeto de auténticas guerras comerciales entre las principales compañías del sector. Algunos ejemplos de esta categoría son la Nintendo DS de Nintendo, o la PSP de Sony. [6]

2.1.2 Smartphone

Dentro de los dispositivos móviles, un smartphone (cuya traducción en español sería “teléfono inteligente”) es una evolución del teléfono móvil tradicional que cuenta con ciertas características y prestaciones que lo acercan más a un ordenador personal que a un teléfono tradicional. Entre dichas características, se puede encontrar una mejora en la

capacidad de proceso y almacenamiento de datos, conexión a Internet mediante Wi-Fi, pantalla táctil, acelerómetro, posicionador geográfico, teclado QWERTY y diversas aplicaciones de usuario como navegador web, cliente de correo, aplicaciones ofimáticas, reproductores de vídeo y audio, etc. incluyendo la posibilidad de descargar e instalar otras nuevas. [6]

A pesar de estas importantes mejoras con respecto a sus predecesores móviles, el reducido tamaño de los smartphones conlleva inexorablemente limitaciones de hardware que los mantienen claramente diferenciados de los ordenadores convencionales. Estas limitaciones se reflejan principalmente en pantallas más pequeñas, menor capacidad del procesador, restricciones de memoria RAM y memoria persistente, y necesidad de adaptar el consumo de energía a la capacidad de una pequeña batería. [6]

Estas limitaciones obligan a tener muy presente la capacidad real del dispositivo a la hora de desarrollar su software, ya sean aplicaciones de usuario o el propio sistema operativo.

2.2 Aplicativo Móvil

Se denomina apps o aplicación al software que se instala en el dispositivo móvil. Se lo llamó como tal desde el inicio del iPhone, la compañía Apple como marketing se usó este nuevo nombre para referirse al software que se encuentra subido a en las tiendas virtuales, ya sea teléfono o tableta y para su instalación se necesita descargarlo e instalar, algunos son gratuitos y otros tienen costos. Estos se integran a las características del equipo, como su cámara, acelerómetro y sistema de posicionamiento global (GPS), etc. [7]

Las aplicaciones son parte del software y siempre se ejecutan sobre algún sistema operativo específico ya sea móvil o no y suelen tener un único objetivo específico. (Revisión de correo, internet, estado de vías, consultas y búsquedas, etc.)[7]

Con ese nombre se designan unas pequeñas aplicaciones que pueden funcionar en el escritorio, el navegador o en ambos, del computador o el dispositivo móvil dependiendo de la plataforma sobre la que funcione el mismo. Se conocen desde hace ya algún tiempo pero últimamente se usan de una forma bastante activa por los diferentes usuarios.

Normalmente funcionan si hay conexión con algún motor de Internet de manera que se convierten en una forma de usar servicios disponibles en la red sin tener que abrir el navegador. [7]

2.2.1 Importancia

Las Aplicaciones móviles permiten el acceso desde un dispositivo móvil a toda la información ya sea corporativa, o de uso personal que quiera movilizar, tal como agendas de clientes, catálogos de productos y precios, lista de teléfonos de empleados o amigos o cualquier información textual. [7]

2.2.2 Impacto de Aplicativos Móviles

Los aplicativos con el creciente desarrollo de dispositivos y tecnologías de redes inalámbricas como Mireles LAN, GSM, GPRS, Bluetooth, ha sido perfeccionada la movilidad de los usuarios, de manera que ya no están obligados a realizar su trabajo en un lugar fijo. Como ejemplo de esto, un inspector de calidad podría ingresar los resultados de su inspección en su dispositivo móvil, y desde el mismo lugar donde realiza éste trabajo, transmitir la información hacia la base de datos de su empresa, vía una interface de red inalámbrica para el almacenamiento permanente o análisis inmediato. [7]

Además con su ayuda con los gráficos, videos y sonidos, la información que se pretenderá explicar es por la manera que se la da más fácil de entenderla y captarla. [7]

2.2.3 Ventajas de Uso

Una aplicación descargada desde una tienda en línea o desde links donde promocionan este app, al dar clic comienza la descarga e instalación dentro del dispositivo Smartphone, para luego poder utilizar con conexión a Internet u offline. Se actualiza periódicamente en sus versiones nuevas para esto requiere estar conectado. [7]

Su visualización está perfectamente diseñada exclusivamente para ser vista en dispositivos móviles, mostrando su contenido expandido a lo largo de su pantalla donde aparecerá la información bien distribuida sin presentar partes del contenido no visibles. [7]

2.3 Sistemas Operativos Para Dispositivos Móviles

Un sistema operativo es un programa o conjunto de programas que en un sistema informático gestiona los recursos de hardware y provee servicios a los programas de aplicación, ejecutándose en modo privilegiado respecto de los demás. Entre algunas de sus características principales tenemos: [8]

- Gobernar y proveer de un ambiente conveniente de trabajo
- Hacer uso eficiente del hardware
- Proveer de una adecuada distribución y asignación de los recursos
- Administrar y controlar la ejecución de programas

Un sistema operativo móvil es un sistema operativo que controla un dispositivo móvil al igual que las computadoras utilizan Windows, Linux o Mac OS, los sistemas operativos móviles son más sencillos y se orientan más a la conectividad inalámbrica, los formatos multimedia y las diferentes maneras de introducir información en ellos. [8]

Según el último estudio realizado por Gartner en febrero de 2016 las ventas de teléfonos inteligentes en todo el mundo a usuarios finales por el sistema operativo indica que Android aumentó 16,6 por ciento en el cuarto trimestre de 2015, para dar cuenta de un 80,7 por ciento del total de los smartphones a nivel mundial seguido por Apple con un 20.6% y con Microsoft cerrando el podio, aunque reduciendo su participación respecto al año pasado, con un 2.8%. Como se evidencia en Tabla 1. [9]

ventas de teléfonos inteligentes en todo el mundo a usuarios finales por sistema operativo en 4Q15 (miles de unidades)				
Operating System	4Q15	4Q15 Market Share (%)	4Q14	4Q14 Market Share (%)
	Units		Units	
Android	325,394.4	80.7	279,057.5	76.0
iOS	71,525.9	17.7	74,831.7	20.4
Windows	4,395.0	1.1	10,424.5	2.8
Blackberry	906.9	0.2	1,733.9	0.5
Others	887.3	0.2	1,286.9	0.4
Total	403,109.4	100.0	367,334.4	100.0

Tabla 1 - Ventas de teléfonos inteligentes en todo el mundo a usuarios finales por sistema operativo Fuente: Gartner (febrero de 2016).

2.3.1 Android



Figura 1: Logo de Android Fuente: https://developer.android.com/images/brand/Android_Robot_100.png.

Hace algunos años, Google decidió que debía expandir su negocio hacia los móviles, así que su mejor estrategia fue crear un sistema operativo móvil propio, gratis y con varios de los más grandes fabricantes de celulares como respaldo, de esta manera nace Android. [8]

Android es un sistema operativo móvil basado en el kernel de Linux, con una interfaz de programación Java, diseñado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros. Desarrollado por la Open Handset Alliance la cual es liderada por Google. [8]

Android permite programar aplicaciones en una variación de Java llamada Dalvik. El sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (GPS, llamadas, SMS, agenda, entre otras.) de una forma muy fácil en un lenguaje de programación muy popular como es Java.[8]

2.3.1.1 Breve Historia

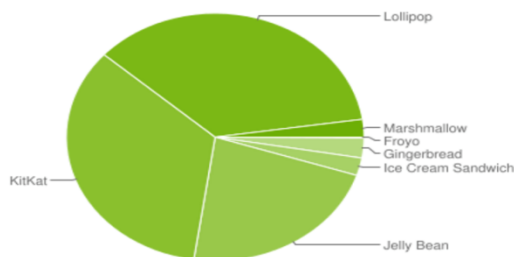
En Julio de 2005, Google adquirió Android Inc., en ese entonces la compañía se dedicaba a la creación de software para teléfonos móviles. Una vez dentro de Google, el equipo desarrolló una plataforma basada en el núcleo Linux para dispositivos móviles, que fue promocionado a fabricantes de dispositivos y operadoras con la promesa de proveer un sistema flexible y actualizable, Google adaptó su buscador y sus aplicaciones para el uso en móviles. [8]

Es el principal producto de la Open Handset Alliance, una alianza comercial de un conglomerado de compañías entre fabricantes y desarrolladores de hardware, software y operadores de servicio, dedicadas al desarrollo de estándares abiertos para dispositivos móviles, algunos de sus miembros son Google, HTC, Dell, Intel, Motorola, Qualcomm, Texas Instruments, Samsung, LG, T-Mobile, Nvidia y Wind River Systems. Google liberó la mayoría del código de Android bajo licencia Apache, una licencia libre y de código abierto.

2.3.1.2 Versiones

Cada actualización del sistema operativo Android es desarrollada bajo un nombre en código de un elemento relacionado con postres, los nombres en código están en orden alfabético. [8]

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.3%
4.1.x	Jelly Bean	16	8.1%
4.2.x		17	11.0%
4.3		18	3.2%
4.4	KitKat	19	34.3%
5.0	Lollipop	21	16.9%
5.1		22	19.2%
6.0	Marshmallow	23	2.3%



Data collected during a 7-day period ending on March 7, 2016.

Figura 2: Porcentajes de distribuciones de Android utilizadas actualmente Fuente:
<http://www.elandroidelibre.com/wp-content/uploads/2016/03/informe-android-marzo.png>

2.3.1.3 Características

- Framework de aplicaciones, que permite el remplazo y la reutilización de los componentes.
- Sistema de notificaciones, esta característica es algo en lo que Android sobresale del resto de sistemas operativos móviles.
- Navegador web integrado, basado en el motor Webkit.
- Sqlite, para almacenamiento de datos.

- Lenguaje de programación Java.
- Soporta diversos formatos multimedia (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
- Soporta HTML, HTML5, Adobe Flash Player, entre otros.
- Incluye un emulador de dispositivos, herramientas para depuración de memoria y plugin para Eclipse.
- Máquina virtual Dalvik, la cual está optimizada para dispositivos móviles, muy similar a Java.
- Telefonía GSM.
- Bluetooth, 3g y Wifi.
- GPS, Cámara, acelerómetro y brújula.
- Tienda de aplicaciones gratuitas o pagadas llamada Google Play.
- Búsqueda por voz versión de Siri. [8]

2.3.2 IOS



Figura 3: Logo de IOS Fuente: <http://wiselimber.com/wp-content/uploads/2015/02/iOS-logo-vector-300x204.gif>.

iOS, anteriormente denominado iPhone OS es un sistema operativo móvil de Apple. Originalmente desarrollado para el iPhone, para luego ser usado en dispositivos como el iPod Touch, iPad y el Apple TV. Apple, Inc. no permite la instalación de iOS en hardware de terceros. Tenía el 26% de cuota de mercado de sistemas operativos móviles vendidos en el último cuatrimestre de 2010, detrás de Google Android y Nokia Symbian. En mayo de 2010 en los Estados Unidos, tenía el 59% de consumo de datos móviles (incluyendo el iPod Touch y el iPad). [8]

La interfaz de usuario de iOS está basada en el concepto de manipulación directa, usando gestos multitáctiles. Los elementos de control consisten de deslizadores, interruptores y botones. La respuesta a las órdenes del usuario es inmediata y provee de una interfaz fluida. La interacción con el sistema operativo incluye gestos como deslices, toques, pellizcos, los cuales tienen definiciones diferentes dependiendo del contexto de la

interfaz. Se utilizan acelerómetros internos para hacer que algunas aplicaciones respondan al sacudir el dispositivo (por ejemplo, para el comando deshacer) o rotarlo en tres dimensiones (un resultado común es cambiar de modo vertical a horizontal). [8]

iOS se deriva de Mac OS X, que a su vez está basado en Darwin BSD, y por lo tanto es un sistema operativo Unix. iOS cuenta con cuatro capas de abstracción: la capa del núcleo del sistema operativo, la capa de "Servicios Principales", la capa de "Medios" y la capa de "Cocoa Touch". [8]

2.3.2.1 Historia

Apple reveló la existencia de iPhone OS en la Macworld Conference & Expo en enero de 2007, aunque el sistema no tuvo un nombre oficial hasta que salió la primera versión beta del iPhone SDK un año más tarde, en marzo de 2008. Antes de esto se consideraba simplemente que el iPhone ejecutaba OS X. A partir de entonces se llamaría iPhone OS. El lanzamiento del iPhone OS tuvo lugar el 29 de junio de 2007. [8]

El interés en el SDK aumentaría en meses siguientes debido al explosivo crecimiento de la plataforma iPhone, que se vio incrementado en septiembre de 2007 del iPod Touch, un dispositivo con las capacidades multimedia del iPhone pero sin la capacidad de hacer llamadas telefónicas. [8]

El 27 de enero de 2010 Steve Jobs, CEO de Apple, anunció el iPad, un dispositivo muy similar al iPod Touch pero con un enfoque más orientado hacia la industria de contenidos. Este dispositivo, apoyado en una pantalla táctil de mayor dimensión, compartiría sistema operativo con sus dos exitosos hermanos, y vendría acompañado de una aplicación oficial para la compra y lectura de libros electrónicos, iBooks. [8]

En de abril de 2010 se estima 185.000 las aplicaciones disponibles para iPhone OS a través de la App Store. El 7 de junio de 2010, durante la presentación del iPhone 4, Steve Jobs anunció que iPhone OS pasaría a ser llamado oficialmente como iOS. El 12 de septiembre de 2012 se presenta iOS 6 durante la presentación del iPhone 5. Tim Cook hace presentación del iPhone 5, iTunes 11 e iOS 6. [8]

2.3.2.2 Características

- Interfaz gráfica está diseñada para el touch screen, con capacidad para gestos, la interfaz se encuentra constituida fundamentalmente de sliders, interruptores y botones, con una respuesta inmediata.
- Notificaciones, cuando hay una notificación, no se interrumpe.
- Soporta acelerómetros.
- Emplea unos 500 MB de almacenamiento, depende del modelo.
- Diversas aplicaciones para gestionar correo, fotos, cámara, mensajes, clima, notas, contactos, reloj, entre otros.
- Soporta multitarea, aunque con algunas limitaciones.
- No dispone de soporte para: Adobe Flash y Java, por este motivo los sitios web con dichas tecnologías no pueden ser visualizados en este sistema operativo. [8]

2.3.3 Windows Phone



Figura 4: Logo de Windows Phone Fuente: <http://onetile.ru/wp-content/uploads/2012/08/windows-phone-8-logo.jpg>

Windows Phone anteriormente conocido como Windows Mobile, es un sistema operativo móvil desarrollado por Microsoft, diseñado para su uso en teléfonos inteligentes. Windows Phone hace parte de los sistemas operativos con interfaz natural de usuario, se basa en el núcleo del sistema operativo Windows CE y posee un conjunto de aplicaciones básicas que utilizan las APIs de Microsoft Windows. [8]

Microsoft tomó la decisión de no hacer compatible Windows Phone con Windows Mobile, por este motivo las aplicaciones existentes no funcionan en Windows Phone, por lo que es necesario desarrollar nuevas aplicaciones, Windows Phone ofrece una nueva interfaz de usuario e integra varios servicios en el sistema operativo. [8]

2.3.3.1 Breve Historia

Windows Phone es el sucesor de la versión del sistema operativo móvil Windows Mobile desarrollado por Microsoft quien comenzó a trabajar desde el 2008 en lo que debía ser un replanteamiento de su sistema operativo, ya que en el año anterior a este había cambiado

la industria de la telefonía celular cuando Nokia introdujo su modelo N95 y nuevos competidores como Apple y Android se interesaban en competir con sus respectivos sistemas operativos móviles. [8]

Después de un intenso trabajo, Microsoft presentó a Windows Phone 7, que se lanzó el 1 de septiembre de 2010, y la versión del SDK estuvo disponible a mediados de septiembre de 2010, fue lanzado en Europa y Asia en octubre del mismo año y en EEUU en noviembre. Inicialmente, Windows Phone 7 estaba destinado para lanzarse durante el 2009, pero luego de varios aplazamientos Microsoft tuvo que desarrollar Windows Mobile 6.5 como una versión de pasajera. [8]

2.4 HTTP

El HTTP (Hypertext Transfer Protocol o Protocolo de Transferencia de Hipertexto) es el protocolo utilizado en las transacciones de internet. Su versión 1.0 surgió en mayo de 1996, y entre sus características podemos encontrar: [10]

- Sigue una arquitectura cliente-servidor, lo más frecuente es que los clientes sean los navegadores de Internet y los servidores sean servidores web.
- Funciona sobre el protocolo de nivel de transporte TCP en el puerto 80.
- No sólo soporta el envío de hipertexto, también trabaja con otros recursos web como imágenes.
- Es un sistema orientado a transacciones, y cada petición/respuesta va acompañada de unas cabeceras que incluyen metadatos sobre la misma.

Este protocolo va a ser importante para la aplicación web a la hora de comunicarse, y para ello será necesario implementar el framework para que permita hacer peticiones HTTP. Se trata de un requisito imprescindible, ya que el cliente para la aplicación web obtiene toda la información a través de Internet, y todas las acciones que se pueden realizar desde la interfaz requieren llamadas al servidor. Existen dos tipos de peticiones necesarias: [10]

GET: petición de la representación de un recurso concreto. El servidor que va a recibir las peticiones del dispositivo cuenta con un API por URL, y en las peticiones de este tipo se encuentran pares clave-valor separados por & que contienen los parámetros. La mayor parte de las peticiones de la aplicación se realizarán con este sistema. [10]

POST: al igual que con GET, con este tipo de petición se pueden pedir recursos, pero tiene otras funciones. Las peticiones POST se suelen emplear para enviar los resultados de un formulario, ya que no tiene la limitación del tamaño máximo por argumento que tiene GET (256 caracteres) y además así no se muestran en la URL los datos enviados, ya que van incluidos en el cuerpo del mensaje. [10]

2.5 HTML5



Figura 5: Logo de HTML5 Fuente: https://upload.wikimedia.org/wikipedia/commons/thumb/6/61/HTML5_logo_and_wordmark.svg/128px-HTML5_logo_and_wordmark.svg.png.

HTML5 (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML (XHTML) (application/xhtml+xml). Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo. [10]

Todavía se encuentra en modo experimental, lo cual indica la misma W3C; aunque ya es usado por múltiples desarrolladores web por sus avances, mejoras y ventajas. Al no ser reconocido en viejas versiones de navegadores por sus nuevas etiquetas, se le recomienda al usuario común actualizar a la versión más nueva, para poder disfrutar de todo el potencial que provee HTML5. [10]

El desarrollo de este lenguaje de marcado es regulado por el Consorcio W3C. [10]

2.5.1 Nuevos Elementos

HTML5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Algunos de ellos son técnicamente similares a las etiquetas <div> y , pero tienen un significado semántico, como por ejemplo <nav> (bloque de navegación del sitio web) y <footer>. Otros elementos proporcionan nuevas funcionalidades a través de una interfaz estandarizada, como los elementos <audio> y <video>. Mejoras en el elemento <canvas>, capaz de renderizar en los navegadores más importantes (Mozilla, Chrome, Opera, Safari e IE) elementos 3D. Algunos elementos de HTML 4.01 han quedado obsoletos, incluyendo elementos puramente de presentación, como y <center>, cuyos efectos son manejados por el CSS. También hay un renovado énfasis en la importancia del scripting DOM para el comportamiento de la web.

Las novedades frente a HTML 4.01 son las siguientes: [10]

- Incorpora etiquetas (canvas 2D y 3D, audio, video) con codecs para mostrar los contenidos multimedia. Actualmente hay una lucha entre imponer codecs libres (WebM + VP8) o privados (H.264/MPEG-4 AVC).
- Etiquetas para manejar grandes conjuntos de datos: Datagrid, Details, Menu y Command. Permiten generar tablas dinámicas que pueden filtrar, ordenar y ocultar contenido en cliente.
- Mejoras en los formularios. Nuevos tipos de datos (eMail, number, url, datetime...) y facilidades para validar el contenido sin Javascript.
- Visores: MathML (fórmulas matemáticas) y SVG (gráficos vectoriales). En general se deja abierto a poder interpretar otros lenguajes XML.
- Drag & Drop. Nueva funcionalidad para arrastrar objetos como imágenes. [10]

2.5.2 CSS 3



Figura 6: Logo de CSS3 Fuente: <http://seeklogo.com/images/C/css3-logo-8724075274-seeklogo.com.gif>

Las hojas de estilo en cascada (en inglés Cascading Style Sheets), CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). CSS 3 es la última versión de los CSS. Esta tiene muchas características nuevas que se convierten mucho más fáciles para configurar el diseño. [7]

Ofrecen la posibilidad de definir las reglas y estilos de representación en diferentes dispositivos, ya sean pantallas de equipos de escritorio, portátiles, móviles, impresoras u otros dispositivos capaces de mostrar contenidos web 2.0. [7]

2.5.3 Web App

Se trata de un sitio web específicamente diseñado para su visualización en las pantallas de los móviles, las cuales además de ser más pequeñas, suelen ser táctiles. Estos dos elementos hay que tenerlos en cuenta a la hora del diseño. [7]

Al tratarse de un sitio web, accedemos a ellos a través del navegador que tengamos instalado en ese dispositivo móvil, simplemente hemos de acceder a su dirección web o URL.

Muchos sitios web móviles se están desarrollando en HTML5/CSS3, el cual es una excelente opción para los desarrolladores que quieren crear un sitio compatible con los dispositivos móviles más modernos. Problemas derivados, como por ejemplo la gran diversidad dispositivos móviles, ya tienen solución en HTML5 mediante código que permiten adaptar el sitio según tamaño y resolución del equipo”.

En resumen, se definirá que un sitio web móvil es multiplataforma (con algunos ajustes adicionales), tecnología más sencilla y estándar, e independiente de los canales de distribución de los fabricantes de OS (los famosos “markets”). [7]

2.5.4 Mobile Web

Se refiere al uso de aplicaciones que se conectan directamente al internet (Browser-Based) desde un Smartphone, un Tablet u otro dispositivo móvil, utilizando directamente redes inalámbricas.

El crecimiento del Mobile Web se ha ido incrementando desde el 2007 con una cantidad enorme de Smartphones y con los Tablets desde el 2010 ya que ambas plataformas proveen un mejor acceso a internet por medio de estas aplicaciones, aún más que las que previas generaciones de teléfonos móviles.

El acceso Mobile Web hoy en día sufre de problemas de usabilidad e interoperabilidad. Los problemas de interoperabilidad surgen de la fragmentación de la plataforma o los dispositivos móviles, los diferentes sistemas operativos móviles y la variedad de browsers y los problemas de usabilidad están centrados básicamente en el pequeño tamaño físico de algunos factores del teléfono, tales como los límites en la resolución de la pantalla tales como la resolución. [7]

2.5 PHP



Figura 7: Logo de PHP Fuente: http://php.net/manual/en/images/c0d23d2d6769e53e24a1b3136c064577-php_logo.png

PHP (PHP Hypertext Pre-processor) es un lenguaje de programación interpretado, diseñado en 1994 para la creación de páginas web dinámicas. Siendo multiplataforma y teniendo un gran parecido con los lenguajes más comunes de programación estructurada permiten que el aprendizaje sea muy corto. Todo esto junto con la facilidad de instalación hizo que PHP cogiera fama muy rápidamente. [10]

Cuando el cliente realiza una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP procesando el script solicitado y generando el contenido de manera dinámica. [10]

Las características más importantes que hacen que PHP destaque son:

- Lenguaje multiplataforma.
- Orientado al desarrollo de web dinámicas con acceso a información de base de datos.
- Soporte con diferentes tipos de servidores de bases de datos, como por ejemplo MySQL.
- Integración con librerías externas hace que la funcionalidad aumente bastante.
- Código invisible al usuario que lo solicita, programación segura y confiable.
- Permite aplicar técnicas de programación orientada a objetos.
- No requiere definición de tipos, sino que decide en periodo de ejecución.
- Manejo de excepciones.
- Uso libre.

Por todo esto se ha elegido su uso como lenguaje para la programación de las distintas funciones y procedimientos. Se han explotado al máximo las posibilidades de este lenguaje creando una programación orientada a objetos para la gestión de la base de datos. De cara al proyecto PHP será el encargado de ejecutar toda la lógica del programa, siendo el controlador que medie entre las vistas y el modelo de bases de datos. [10]

2.6 MySQL



Figura 8: Logo de MySQL Fuente: <https://upload.wikimedia.org/wikipedia/en/thumb/6/62/MySQL.svg/640px-MySQL.svg.png>

MySQL es un sistema de gestión relacional, multihilo y multiusuario con más de seis millones de instalaciones. El objetivo de la empresa en el momento de su creación en 1995 fue seguir con el estándar SQL, sin sacrificar la velocidad, fiabilidad o usabilidad. A lo largo de los años ha ido convirtiéndose en el gestor más popular del mundo por su rendimiento, eficiencia y facilidad de uso.

Sus características más importantes son:

- Implementación multihilo.
- Soporte en el tipo de datos.
- Portabilidad entre sistemas.
- Buen nivel de seguridad.

Su punto fuerte es una gran aceptación, debido a la cual existen infinidad de librerías para una gran cantidad de lenguajes de programación. En cuanto a la licencia, MySQL es de uso libre y código abierto pero Oracle tiene el copyright sobre ese código.

Hay varios tipos de gestión dentro del propio MySQL, entre los que destacan dos, MyISAM y InnoDB. En el caso de MyISAM, tecnología usada por defecto, se da una rápida lectura al utilizar el motor no transaccional ya que no tiene que hacer comprobaciones de integridad referencial ni bloquear las tablas por ausencia de atomicidad, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación.

En segundo caso, InnoDB, es la tecnología opuesta, soporta las transacciones tipo ACID y bloqueo de registros junto con la integridad referencial, ofreciendo una fiabilidad y consistencia a cambio de un rendimiento menor.

2.7 Editores de Código

Es un procesador de textos orientado para escribir código fuente de aplicaciones en general en lenguajes de programación. Generalmente los editores de código soportan varios lenguajes y son capaces de abrir varios archivos a la vez, resaltar su sintaxis y ofrecer ayudas contextuales a la hora de escribir o visualizar el código fuente de las aplicaciones. [11].

2.7.1 Sublime Text 3



Figura 9: Logo de Sublime Text 3 Fuente: https://pruebas012015.files.wordpress.com/2015/06/sublime_text_logo.png

Sublime Text es un editor de texto y editor de código fuente está escrito en C++ y Python para los plugins. Desarrollado originalmente como una extensión de Vim, con el tiempo fue creando una identidad propia, por esto aún conserva un modo de edición tipo vi llamado Vintage mode.

Se puede descargar y evaluar de forma gratuita. Sin embargo no es software libre o de código abierto y se debe obtener una licencia para su uso continuado, aunque la versión de evaluación es plenamente funcional y no tiene fecha de caducidad. [11]

2.7.2 Brackets



Figura 10: Logo Brackets Fuente: <https://pruebas012015.files.wordpress.com/2015/06/brackets.png>

Brackets es un editor de código abierto para el diseño y desarrollo web construido sobre tecnologías como HTML, CSS y JavaScript. El proyecto fue creado y es mantenido por Adobe, y se distribuye bajo una licencia MIT.

Brackets se diferencia de los demás editores gracias a la facilidad de mostrar el código específico de acuerdo al contexto usado, esto mediante su interfaz tipo Quick Edit UI.

Brackets te permite trabajar directamente en el navegador editando el código al instante, estableciendo breakpoints y moviéndose con fluidez entre las diferentes vistas de código y del mismo navegador. El editor de texto dentro de Brackets se encuentra basado en CodeMirror. [12]

2.7.3 Notepad++



Figura 11: Logo Notepad++ Fuente: https://pruebas012015.files.wordpress.com/2015/06/notepad_logo.png

Notepad++ es un editor gratuito de código fuente y un sustituto de Notepad que soporta varios lenguajes. Funciona en entorno MS Windows y su uso está regulado por la licencia GPL.

Basado en un potente componente de edición llamado Scintilla, Notepad++ está escrito en C++ y usa directamente la API de Win32 y STL, lo que asegura una gran velocidad de ejecución y un menor tamaño del programa. Mediante la optimización de tantas rutinas como sea posible sin perder la interfaz amigable con el usuario, Notepad++ intenta reducir las emisiones mundiales de dióxido de carbono. Al usar una menor potencia de CPU, el PC puede frenar y reducir el consumo de energía, dando como resultado un medio ambiente más ecológico. [13]

2.8 Frameworks

“Los frameworks o marcos de trabajo, son esquemas (un esqueleto, un patrón) para el desarrollo y/o implementación de una aplicación. Definen, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar. Los frameworks incluyen programas de apoyo, compiladores, bibliotecas de código, interfaz de programación de aplicaciones (API) y un conjunto de herramientas que reúnen a todos los diferentes componentes para permitir el desarrollo de un proyecto o solución.” [14]

2.8.1 Frameworks de Programación en Android

El entorno de trabajo oficial de Android es desarrollado por Google y cuenta con una página web en la que se explica, de manera detalla y con algunos ejemplos, el uso del API mediante el lenguaje de programación “Java”. Sin embargo, a pesar de que esta es la opción más estable, completa y recomendable, existen en la red múltiples opciones que permiten a los desarrolladores programar aplicaciones usando otros métodos y lenguajes de programación alternos a Java. [15]

A continuación se presentan cuatro frameworks diferentes con los cuales se pueden realizar aplicaciones en Android. Dos de ellas, la nativa y AppInventor, utilizan el API

nativo de Android (Aunque el segundo lo utiliza mediante la unión de bloques visuales), mientras que los dos restantes, PhoneGap y JQuery Mobile, utilizan HTML5, hojas de diseño CSS, y JavaScript en conjunto con múltiples librerías.

2.8.1.2. API nativa de Android por Google

Es la alternativa oficial presentada por Google para realizar aplicaciones en la plataforma de Android. “Utiliza el lenguaje de programación “Java” en conjunto con comandos propios que permiten acceder a las funcionalidades del teléfono y ejecutar diferentes acciones y eventos”. [16]



Figura 12: Icon Android Developers Fuente: <http://i2.wp.com/www.artit-k.com/wp-content/uploads/2013/05/Android-Developer-logo.png?resize=622%2C130>

Lo recomendable para el desarrollo de aplicaciones utilizando este API es la instalación de una plataforma integral de desarrollo (IDE) como Eclipse o NetBeans, que deben ser configurados para poder funcionar con el entorno de Android. Una vez superado el proceso de instalación y configuración, la plataforma permite la creación de nuevos proyectos Android. Al crear una aplicación se debe especificar el nombre de la misma y las versiones del sistema operativo que serán soportadas por ella.

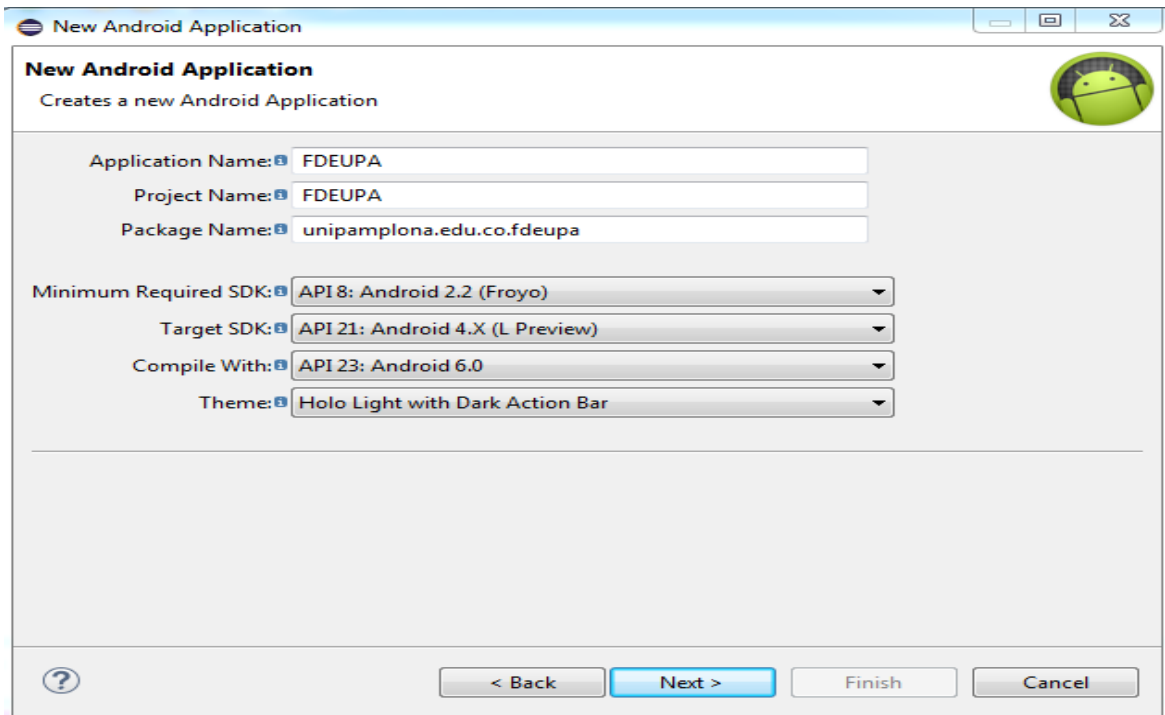


Figura 13: Interfaz de Eclipse para crear un nuevo proyecto en “Android” Fuente: Propia

Finalizada la creación de la aplicación, se presenta en una pestaña del IDE el proyecto con todas las carpetas y archivos que lo componen:

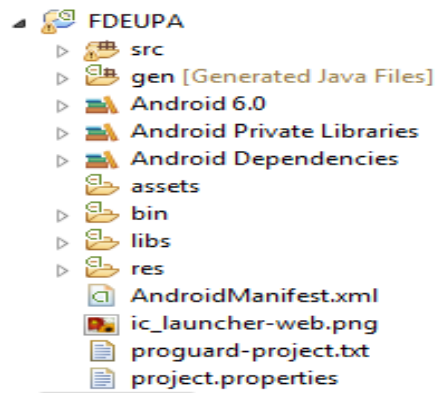


Figura 14: Carpetas creadas en el IDE para el manejo del proyecto en “Android” Fuente: Propia

Entre estos archivos y carpetas se debe destacar el `AndroidManifest.xml` en el cual se declaran todos los permisos y accesos que tendrá la aplicación a desarrollar, cuales son los componentes que se utilizan, las versiones del sistema operativo compatibles, entre otros.

Por otra parte, cada una de las carpetas que componen el proyecto contienen información específica que en conjunto permiten construir la aplicación. A continuación se explican sus principales propiedades y el tipo de archivos que contienen: [15]

➤ Carpeta /src/

En esta carpeta se encuentra el código fuente de la aplicación a desarrollar. Contiene todas las clases donde se implementan las funcionalidades y eventos que hacen servir la aplicación, como las actividades, los servicios, los broadcast receivers y los Content Providers:

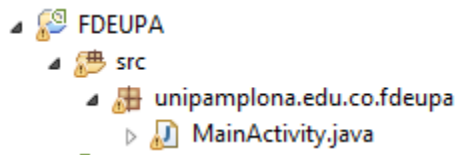


Figura 15: Carpeta “src” del proyecto “Android” expandida Fuente: Propia

➤ Carpeta /gen/

Contienen un conjunto de archivos con código generado automáticamente por el sistema cada vez que se compila el proyecto. En dicha carpeta se destaca el archivo R.java, en la cual se encuentran una serie de constantes con el ID de todos los recursos utilizados en la aplicación (elementos que forman la interfaz gráfica, imágenes, videos, sonidos, etc.)

Debido a que el código de esta carpeta es generado de manera automática, no se deberían modificar los archivos contenidos en ella bajo ninguna circunstancia.

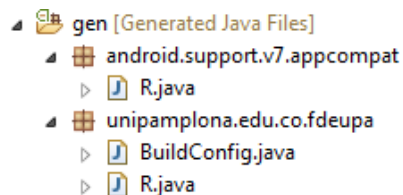


Figura 16: Carpeta “gen” del proyecto “Android” expandida Fuente: Propia

➤ Carpeta /assets/

Contiene ficheros auxiliares utilizados por la aplicación (y que se incluirán en su propio paquete), como por ejemplo ficheros de configuración, de datos, etc.

A diferencia de los archivos que se encuentran en la carpeta /res/, a los elementos que se carguen en este directorio no se les genera ID dentro del archivo R.java, por lo que se accede a ellos por la ruta de su ubicación (como cualquier archivo del sistema).

➤ Carpeta /bin/

Al igual que la carpeta /gen/, esta es generada automáticamente por el sistema y contiene los elementos compilados de la aplicación y otros ficheros auxiliares, entre los que se puede destacar el archivo con extensión “.apk” , que es el ejecutable de la aplicación que se instalará en el dispositivo.

➤ Carpeta /libs/

Contiene todas las librerías auxiliares utilizadas por la aplicación, por ejemplo la librería de GoogleMaps para acceder a la cartografía. Generalmente son archivos escritos en java con extensión “.jar”.

➤ Carpeta /res/

Cuenta con todos los ficheros de recursos utilizados en el proyecto, es decir, imágenes, sonidos, vídeos, cadenas de texto, etc. Se encuentra distribuida en múltiples subcarpetas que permiten organizar los diferentes recursos:

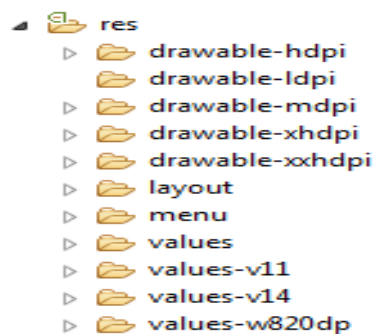


Figura 17: Carpeta “res” del proyecto “Android” expandida Fuente: Propia

➤ /res/drawable

En esta carpeta se encuentran todas las imágenes y diferentes elementos gráficos de la aplicación. Se suele dividir en varias subcarpetas para diferentes resoluciones y densidades de pantalla.

➤ /res/layout

Está formada por archivos XML, en los cuales se definen los elementos que conforman la interfaz gráfica de usuario. En estos archivos se declaran los botones, imágenes, seleccionadores y el resto de los elementos con los que el usuario va a interactuar y cuyas acciones serán tratadas e implementadas por las clases programadas en la carpeta /src/

- /res/menú

Contiene archivos de definición XML con la configuración de los menús de la aplicación.

- /res/values

Contiene otros ficheros XML de recursos de la aplicación, entre los que se destacan las cadenas de texto (strings.xml), los estilos (styles.xml), colores (colors.xml) y los arreglos de valores (arrays.xml).

Algunas carpetas poseen un sufijo “-v”, la cual es usada para definir recursos independientes destinados para versiones concretas del sistema operativo.

Adicionalmente a estas carpetas, el programador puede agregar nuevas carpetas en el caso de que se utilicen otros recursos, y entre ellas se pueden destacar:

- /res/anim/: Definición de animaciones
- /res/raw/: Contiene definiciones adicionales, usualmente en un formato diferente a XML
- /res/xml/: Contiene otros ficheros XML de datos utilizados por la aplicación.

2.8.1.3 Resumen General

- Características y Fortalezas
 - ✓ Gratuito y de código fuente completamente abierto.
 - ✓ Representa la alternativa de desarrollo con la mayor cantidad de funciones para acceder a accesorios y hardware del dispositivo como cámara, GPS, acelerómetro, etc.

- ✓ Es la alternativa con la mejor y más extensa documentación para ayudar al programar a desarrollar aplicaciones
 - ✓ Aplicaciones construidas utilizando Java.
- Debilidades
- ✓ Solo disponible para Android. No es multiplataforma.
 - ✓ No cuenta con su propio IDE.
 - ✓ Requiere de mucha preparación para iniciar la programación.
 - ✓ Requiere conocimientos en el lenguaje de programación Java.
- Requerimientos para el Desarrollo
- ✓ Instalar un IDE. El recomendado es “Eclipse”
 - ✓ Instalar el SDK de Android.
 - ✓ Instalar el ADT Plugin en el IDE.
 - ✓ Instalar el JDK (Java Development Kit)
 - ✓ Opcional: Instalar el AVD manager si se desean utilizar emuladores de dispositivos.

2.8.2 AppInventor



Figura 18: Icono de Android App Inventor Fuente: <http://e-forma.kzgunea.eus/pluginfile.php/4174/course/section/2417/APP%20inventor.jpg>

Es un entorno de desarrollo visual web, que mediante el uso de bloques al estilo de piezas de rompecabezas, permite construir aplicaciones que no requieren un uso extensivo de las librerías de Android. Al estar formado por bloques visuales, el framework permite obtener aplicaciones funcionales, sin la necesidad de escribir líneas de código en algún lenguaje de programación. “Fue anunciado en 2010 por Google, y está basado en la biblioteca Open Blocks Java, del MIT”. [17]

A pesar de encontrarse en fase de desarrollo (Beta), AppInventor proporciona un amplio acceso a componentes y funciones nativas del sistema, entre las cuales se pueden

destacar el GPS, acelerómetro y datos de orientación, mensajes de texto, conversión habla a texto, datos de contacto, almacenamiento permanente, servicios Web, Twitter, entre otros. [17]

Durante el proceso de desarrollo, las aplicaciones construidas con AppInventor no son almacenadas en el disco duro del computador sino que se encuentran en la red, por lo tanto se puede acceder a ellas desde cualquier equipo con acceso a internet y que tenga instalado el SDK de Google. Una vez finalizado el proceso de desarrollo, el framework permite compilar la aplicación y crear un archivo ejecutable “.apk” que podrá ser instalado en un dispositivo con Android. [17]

La programación se realiza mediante el uso de dos interfaces:

➤ “App Inventor Designer”

Contiene todos los elementos y componentes que serán usados en la aplicación, tanto visible como no visible. Los componentes visibles son aquellos que permiten armar la interfaz gráfica de usuario (botones, seleccionadores, imágenes, etc), y los no visuales son los encargados de acceder a herramientas del teléfono y/o controlar eventos (Sensores, cámara, relojes, reproductores de sonido, etc). [17]

Esta interfaz está dividida en 4 secciones diferentes:

- a) *Palette*: Contiene todos los elementos y herramientas soportadas por el framework.
- b) *Viewer*: Está formada por una ventana que simula la pantalla de un dispositivo móvil, en la cual se arrastran todos los componentes que se desean utilizar (tanto visuales como no visuales).
- c) *Components*: Es una lista en la que se encuentran todos los elementos colocados en el Viewer. Permite renombrar, editar y borrar los componentes.

d) *Properties*: Se presentan todas las propiedades editables y configurables de un elemento seleccionado. Por ejemplo, para un representador de texto se tiene la opción de dar color a la letra, cambiar el tamaño de la misma, etc.

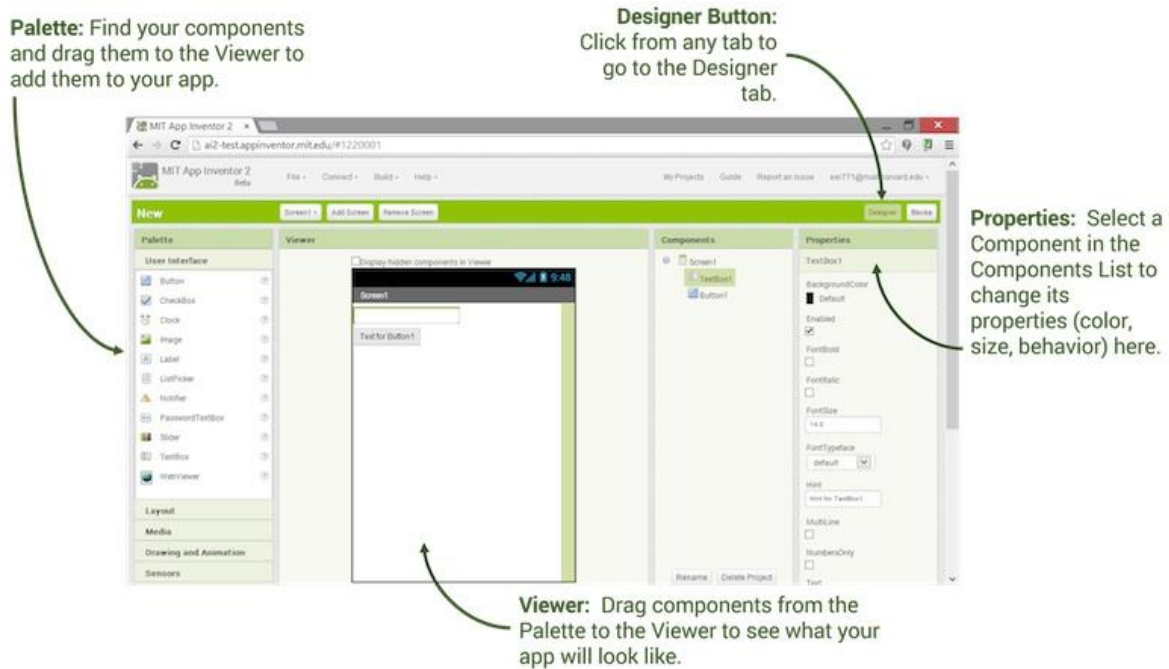


Figura 19: Interfaz del “App Inventor Designer” Fuente: <http://appinventor.mit.edu/explore/sites/all/files/SetupAI2/DesignTab.jpg>

➤ “App Inventor Blocks Editor”

Está constituido por un conjunto de bloques enlazados unos con otros, los cuales definen el comportamiento de la aplicación. Estos elementos trabajan de manera similar a las piezas de un rompecabezas, donde dos bloques solo pueden ser unidos si la salida de uno encaja con la entrada del otro. Los bloques representan los componentes que fueron definidos en el “App Inventor Designer” y las funciones que se pueden realizar con ellas. Además de los componentes, existen múltiples piezas que permiten desarrollar la lógica de la aplicación, como comparadores, condicionantes, etc. [17]

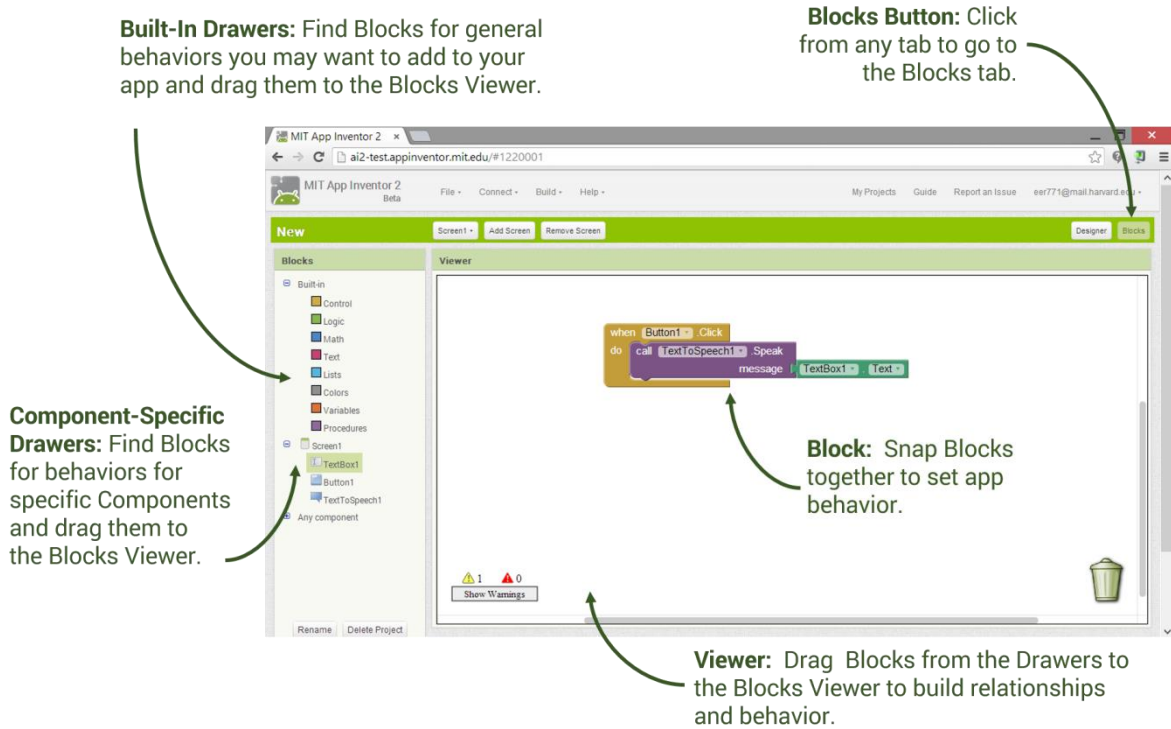


Figura 20: Interfaz del “App Inventor Blocks Editor” Fuente: <http://appinventor.mit.edu/explore/sites/all/files/SetupAI2/BlocksTab.png>

2.8.2.1 Resumen General

➤ Características y Fortalezas

- ✓ Gratuito
- ✓ Es una aplicación web a la que se puede acceder desde cualquier ordenador sin necesidad de instalarlo.
- ✓ Es sencillo de manejar y puede ser utilizado fácilmente por cualquier persona sin necesidad de que tenga algún conocimiento en programación.
- ✓ Cuenta con su propia herramienta para construir la interfaz gráfica de manera sencilla.
- ✓ Cuenta con un emulador Android y también permite conectar dispositivos reales mediante cable o por WI-FI.
- ✓ Utiliza bloques visuales como piezas de rompecabezas para desarrollar la lógica de la aplicación.
- ✓ El IDE puede correr en cualquier sistema operativo con Java

- ✓ Estudio comparativo de alternativas y frameworks de programación, para el desarrollo de aplicaciones móviles en entorno Android.

- Debilidades
 - ✓ Solo disponible para Android. No es multiplataforma.
 - ✓ Cuenta con funciones limitadas.
 - ✓ No se puede acceder al IDE sin conexión a internet.

- Requisitos para comenzar el desarrollo
 - ✓ Tener una cuenta en Google (con la cual se accede a las aplicaciones del usuario).
 - ✓ Instalar Java Runtime.
 - ✓ Instalar el AppInventor Setup el cual cuenta con el emulador y las funciones que utiliza.
 - ✓ Un explorador de internet para acceder a la página del framework.

2.8.3 PhoneGap



Figura 21: Icono del framework "PhoneGap" Fuente: <http://www.javiercantos.net/image.axd?picture=%2F2014%2F04%2Fphonegap-logo.png>

“Es un framework que permite desarrollar aplicaciones móviles para múltiples plataformas, usando los lenguajes de programación HTML5, CSS y JavaScript en conjunto con comandos propios para acceder a funcionalidades del teléfono. Es uno de los frameworks más utilizados en la red con más de 400 mil desarrolladores.”. [18]

El núcleo del motor de PhoneGap es 100% de código abierto, en virtud del proyecto de Apache Cordova. PhoneGap proporciona una interfaz de programación de aplicaciones (API) que permite acceder a múltiples de las funcionalidades nativas del sistema operativo utilizando JavaScript. “El API de PhoneGap se encarga de traducir los comandos JavaScript

y realizar la comunicación con el sistema operativo nativo. A pesar de que las aplicaciones desarrolladas en PhoneGap utilizan HTML, CSS y JavaScript, el producto final es un archivo genérico que puede ser traducido a múltiples extensiones para trabajar en diferentes sistemas operativos.” [18]

En algunas ocasiones, el programador requiere hacer uno de funciones, comandos y elementos que no se encuentran desarrollados de forma nativa en el API de PhoneGap. Por dicha razón, el framework da la posibilidad de construir los llamados “Plugins”, los cuales permiten construir (generalmente utilizando código nativo) aquellas funciones de interés que no están desarrolladas. Aunque pueden ser de ayuda, se deben utilizar con mucho cuidado ya que pueden acabar con la compatibilidad multiplataforma.

Para programar en PhoneGap es necesario contar con una plataforma integral de desarrollo como Eclipse, donde se debe tener instalado tanto el Android SDK, como el Plugin ADT. Se debe crear un nuevo proyecto Android de igual manera como se explicó para la programación Nativa, pero a diferencia de este, la distribución del código y los recursos de la aplicación no se realiza de igual manera.

En el caso de PhoneGap, la carpeta /assets/ almacenan todos los archivos y recursos que serán utilizados por la aplicación. Los archivos “.html” contienen los comandos para implementar la interfaz gráfica de usuario (desarrollada con HTML5), junto con los comandos de diseño CSS y el tratado de eventos y acciones del sistema en JavaScript. La carpeta /assets/res/ contiene todos los recursos a utilizar (imágenes, sonidos, videos, etc), mientras que la carpeta /aseets/lib/, cuenta con las librerías utilizadas por la aplicación. Una librería que no puede faltar es “Cordova.js”, la cual es descargada desde la página oficial del framework y contienen todas las funciones del manejo de utilidades que se pueden implementar con PhoneGap. A dicha librería se le debe invocar en cada uno de los archivos “HTML” desarrollados. [18]

2.8.3.1 Resumen General

- Características y Fortalezas
 - ✓ Gratuito y de código fuente completamente abierto.

- ✓ Es el framework, entre los estudiados, con la mayor cantidad de plataformas móviles compatibles. Además el mismo código desarrollado es compatible con todas las plataformas.
- ✓ Acceso a elementos y hardware del dispositivo mediante JavaScript, como cámara, GPS, acelerómetro, etc.
- ✓ Se integra con el código fuente original
- ✓ Aplicaciones construidas utilizando lenguajes web muy comunes
- Debilidades
 - ✓ La aplicación se hace usando el motor de plataforma web, no objetos individuales de la interfaz de usuario nativo.
 - ✓ No cuenta con su propio IDE.
 - ✓ Acceso limitado a accesorios y hardware del dispositivo
 - ✓ Requiere conocimientos en los lenguajes de programación HTML5, CSS y JavaScript
- Requerimientos para el Desarrollo
 - ✓ Instalar un IDE. El recomendado por el framework es “Eclipse”
 - ✓ Instalar el SDK de Android.
 - ✓ Instalar el ADT Plugin en el IDE.
 - ✓ Descargar las librerías de Cordova y adjuntarlas al proyecto que se cree.
 - ✓ Instalar el JDK (Java Development Kit)
 - ✓ Opcional: Instalar el AVD manager si se desean utilizar emuladores de dispositivos.

2.8.4 JQuery Mobile



Figura 22: Icono del framework “Jquery Mobile” Fuente: <http://demos.jquerymobile.com/1.4.5/assets/img/jquery-logo.png>

“Es un framework basado en HTML5, JavaScript y CSS, que permite desarrollar aplicaciones soportadas por la mayoría de los sistemas operativos móviles, mediante la implementación de un único código.” [19]

Cuenta con un completo API con opciones de configuración globales, eventos y métodos, que permiten aplicar secuencias de comandos, generar páginas dinámicas, e incluso construir aplicaciones nativas con herramientas como PhoneGap (que permite acceder a los elementos de los equipos).

“jQuery Mobile está desarrollado por la fundación jQuery UI, y está basado en las mismas librerías JavaScript que JQuery, lo que permite a desarrolladores con dominio de esta, poder programar aplicaciones móviles de manera mucho más rápida y sin necesidad de aprender un nuevo lenguaje ni manejo de funciones.” [20]

El framework incluye un sistema de navegación en AJAX, que permite transiciones animadas de página y un conjunto básico de widgets para construir la interfaz de usuario: páginas, cuadros de diálogo, barras de herramientas, botones listviews con iconos, elementos de formulario, y más.

La programación se realiza de manera similar a PhoneGap, donde se necesita una plataforma integral de desarrollo como Eclipse, con el Plugin ADT y el SDK ya instalado. Se crea un nuevo proyecto en Android y se almacenan todos los recursos y paginas en la carpeta /assets/ al igual que en PhoneGap. Al estar escrita en HTML5, las páginas de la aplicación tendrán extensión “.html”, en las cuales se encuentran los comandos HTML, JavaScript y CSS.

Por otra parte, el framework cuenta con múltiples librerías escritas en JavaScript y CSS que permiten dar funcionalidad y diseño a la aplicación. La página no recomienda descargar dichas librerías ya que especificando su ruta en los archivos HTML se puede acceder a ellas a través de una conexión a internet, sin embargo, si se desea que la aplicación funcione correctamente sin conexión a la red, se deben descargar y guardarlas en el proyecto.[20]

2.8.4.1 Resumen General

- Características y Fortalezas
 - ✓ Gratuito y de código libre.

- ✓ Es compatible con múltiples plataformas, permitiendo desarrollar un único código compatible con todas ellas.
 - ✓ Cuenta con librerías de diseño que permiten obtener una aplicación estética e intuitiva de manera fácil y rápida.
 - ✓ Aplicaciones construidas utilizando lenguajes web muy comunes
- Debilidades
- ✓ La aplicación se desarrolla usando el motor de plataforma web, no objetos individuales de la interfaz de usuario nativo.
 - ✓ No cuenta con su propio IDE.
 - ✓ Requiere del uso de librerías de PhoneGap para tener acceso a accesorios y hardware del dispositivo móvil.
 - ✓ Requiere conocimientos en los lenguajes de programación HTML5, CSS y JavaScript
- Requerimientos para el Desarrollo
- ✓ Instalar un IDE. El recomendado es “Eclipse”
 - ✓ Instalar el SDK de Android.
 - ✓ Instalar el ADT Plugin en el IDE.
 - ✓ Descargar las librerías de jQuery Mobile o llamarlas especificando su ruta web.
 - ✓ Instalar el JDK (Java Development Kit)
 - ✓ Opcional: Instalar el AVD manager si se desean utilizar emuladores de dispositivos.
 - ✓ Opcional: Descargar las librerías de Cordova y adjuntarlas al proyecto que se cree en el caso de hacer uso de accesorios del dispositivo.

2.8.5 Otros frameworks

Además de los frameworks anteriormente explicados, existen otras alternativas utilizadas por los desarrolladores para la programación de aplicaciones en Android, a continuación se nombran algunos de ellos:

- Corona: Es un framework multiplataforma utilizado principalmente para desarrollar juegos, que se caracteriza por permitir construir aplicaciones de alta calidad en tiempos bastante reducidos. Utiliza el lenguaje de programación ULA. [21]
- Dojo Mobile: Es un frameworks basando en HTML5 y JavaScript, que permite el desarrollo rápido de aplicaciones web para móviles con un aspecto nativo. Es similar a jQuery Mobile. [22]
- Flash Builder: Es un software que brinda un entorno de desarrollo para crear juegos y aplicaciones utilizando el lenguaje ActionScript y el marco Flex de código abierto. Flash Builder incluye herramientas profesionales de evaluación, como perfiladores, monitoreo de red, y soporte de pruebas unitarias. [23]
- Sencha Touch: Al igual que muchos de los frameworks mencionados, Sencha Touch permite la creación de aplicaciones de manera personalizada y rápida utilizando HTML5 y JavaScript. [24]
- Appcelerator Titanium: Es uno de los frameworks más completos que se encuentran en la red. Permite realizar aplicaciones móviles utilizando principalmente códigos en lenguaje JavaScript y JSON, aunque se puede hacer uso de Ruby, Python y scripts PHP. Al igual que otros frameworks, cuenta con una completa biblioteca de funciones que elimina la necesidad de escribir códigos en lenguaje nativo (aunque proporcionan una manera de hacerlo si desea ampliar la funcionalidad). [25]

2.9 Recuperación De Información

La Recuperación de Información (IR, Information Retrieval) es el área de la ciencia y la tecnología que trata de la adquisición, representación, almacenamiento, organización y acceso a elementos de información. Desde un punto de vista práctico, dada una necesidad de información del usuario, un proceso de IR produce como salida un conjunto de documentos cuyo contenido satisface potencialmente dicha necesidad. Esta última puntualización es de suma importancia, ya que la función de un sistema de IR no es la de devolver la información deseada por el usuario, sino únicamente la de indicar qué documentos son potencialmente relevantes para dicha necesidad de información. [2] El ejemplo más popular de un sistema de recuperación de información es el de los motores de búsqueda en Internet tales como Google [26], Bing [27] o Yahoo [28].

2.9.1 Modelos de RI

El diseño de un Sistema de Recuperación de Información (SRI) se realiza bajo un modelo, donde queda definido “cómo se obtienen las representaciones de los documentos y de la consulta, la estrategia para evaluar la relevancia de un documento respecto a una consulta y los métodos para establecer la importancia (orden) de los documentos de salida”. [29]

Existen varias propuestas de clasificación de modelos, una de las síntesis más completas la realiza Dominich en cinco grupos: [30]

Modelo	Descripción
Modelos clásicos	Incluye los tres más comúnmente citados: booleano, espacio vectorial y probabilístico.
Modelos alternativos	Están basados en la Lógica Fuzzy.
Modelos lógicos	Basados en la Lógica Formal. La recuperación de información es un proceso inferencial.

Modelos basados en la interactividad	Incluyen posibilidades de expansión del alcance de la búsqueda y hacen uso de retroalimentación por la relevancia de los documentos recuperados.
Modelos basados en la Inteligencia Artificial	Bases de conocimiento, redes neuronales, algoritmos genéticos y procesamiento del lenguaje natural.

Tabla 2 - Clasificación de los Modelos de Recuperación de Información según Dominich. Fuente: Dominich, S. 'A unified mathematical definition of classical information retrieval'. *Journal of the American Society for Information Science*, 51 (7), 2000. p. 614-624.

En los modelos clásicos, los tres coinciden en tomar en consideración términos de indización (las palabras que representan a todos los documentos de la colección almacenada en una BD). Todos ellos se basan en la asignación de pesos a cada término en cada documento, de tal forma que el peso de un término resulta un indicador de la relevancia de ese término en el documento; sin tener en cuenta, sin embargo, la posible correlación entre términos diferentes del mismo documento (es decir, se asume la independencia entre términos). [31]

2.9.1.2 Modelo Booleano

El modelo booleano es un modelo de recuperación simple basado en la teoría de conjuntos y en el álgebra booleana. Los documentos se representan mediante términos índice. La relevancia es binaria: un documento es relevante o no. Las consultas son expresiones booleanas del tipo: [32]

- 1 palabra: un documento es relevante si y solo si contiene la palabra
- AND: los documentos deben contener todas las palabras
- OR: los documentos deben contener alguna palabra
- A BUTNOT B: los documentos deben ser relevantes para A pero no para B

Supongamos, por ejemplo, que queremos tener acceso a aquellos documentos que contengan los términos modelo y booleano, pero que no contengan el término vectorial. La consulta asociada a esta necesidad podría ser: [33]

modelo AND booleano AND NOT vectorial

Mientras que el conjunto de documentos relevantes se correspondería gráficamente con el área rayada de la figura 23.

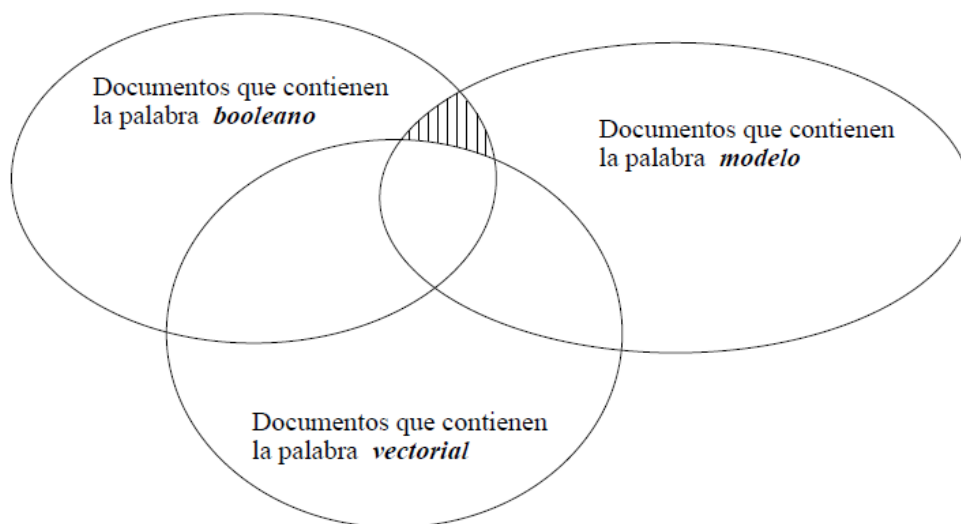


Figura 23: Modelo booleano: consulta 'modelo AND booleano AND NOT vectorial' Fuente: [33]

Dada su simplicidad y formalismo claro, el modelo booleano recibió gran interés en el pasado y fue adoptado por muchos de los sistemas bibliográficos comerciales. Es de las primeras ideas que a uno se le ocurren, y es la opción preferida para manejar texto en una BD relacional.

Sin embargo, es bastante malo para IR. Su estrategia de recuperación se basa en un criterio de decisión binario (i.e., un documento se relevante o no relevante) sin ninguna noción de escala. Por lo tanto, da lo mismo que un documento contenga 1 o 100 veces las palabras de la consulta, o que cumpla una o todas las cláusulas de un OR. Tampoco considera la posibilidad de una coincidencia parcial de un documento (ej. que cumpla con

casi todas las cláusulas de un AND). Es, por tanto, más un modelo de recuperación de datos que de información. [32]

Además, a muchos usuarios les resulta complicado expresar sus consultas mediante expresiones booleanas. Por ello, en la mayoría de los casos las expresiones formuladas actualmente por los usuarios son bastante simples. Para usuarios expertos puede ser adecuado, o con mejores interfaces. [32]

2.9.1.3 Modelo Vectorial

Para dar solución a los problemas planteados por el modelo booleano, el modelo vectorial plantea un marco formal diferente en el que se permite tanto la asignación de correspondencias parciales, como la existencia de grados de relevancia en base a los pesos de los términos en consultas y documentos. [33]

En este nuevo modelo, ambos, consultas y documentos, son representados mediante vectores dentro de un espacio multidimensional definido por los propios términos, de tal forma que cada uno de los términos (diferentes) del sistema —es decir, cada uno de los términos del vocabulario— define una dimensión. De este modo, un vocabulario de tamaño t define a un espacio t –dimensional donde un documento d_j es representado como un vector [33]

$$\vec{d}_j = (w_{1j}, w_{2j}, \dots, w_{tj}) \quad (1)$$

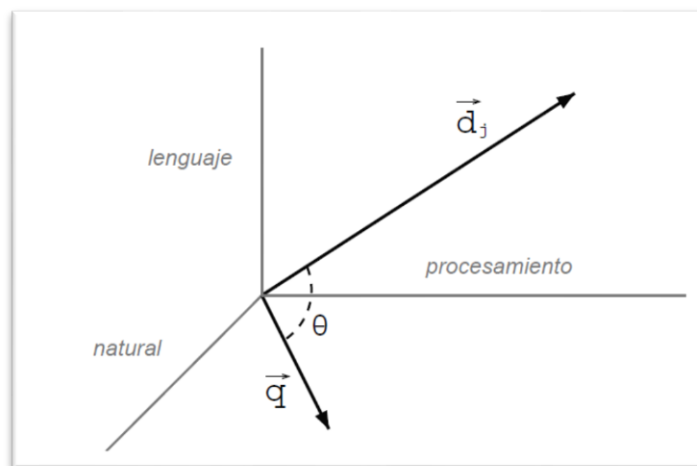


Figura 24: Modelo vectorial: espacio tridimensional definido por el vocabulario {procesamiento, lenguaje, natural}
Fuente: [33]

y, paralelamente, una consulta q es representado como un vector

$$\vec{q} = (w_{1q}, w_{2q}, \dots, w_{tq}) \quad (2)$$

Siendo $w_{ij} \geq 0$ y $w_{iq} \geq 0$ los pesos del término t_i —el i -ésimo término del vocabulario— en el documento d_i y la consulta q , respectivamente. [33]

Desde un punto de vista geométrico, si ambos vectores, consulta y documento, están próximos, es factible asumir que el documento es similar a la consulta —en otras palabras, el documento es posiblemente relevante. Por lo tanto, a mayor proximidad entre ambos vectores, mayor relevancia del documento. En concreto, el modelo vectorial plantea medir la similitud entre un documento d_j y una consulta q en base a la proximidad entre sus vectores correspondientes, en lugar de basarse en criterios de inclusión/exclusión como en el caso del modelo booleano. A su vez, dicha proximidad entre vectores es medida en base al coseno del ángulo θ que tales vectores forman. Llegados a este punto cabe decir que, al asumir que los términos están incorrelados, esto nos permite suponer que las dimensiones son ortogonales, simplificando notablemente los cálculos. La figura 24 muestra un ejemplo gráfico para un vocabulario mínimo de tres términos {procesamiento, lenguaje y natural} que define un espacio tridimensional. De esta forma,

$$sim(d_j, q) = \cos(\theta) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \times |\vec{q}|} = \frac{\sum_{i=1}^t w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^t w_{ij}^2} \times \sqrt{\sum_{i=1}^t w_{iq}^2}} \quad (3)$$

siendo $|\vec{d}_j|$ y $|\vec{q}|$ las normas de los vectores documento y consulta, respectivamente.

El modelo vectorial no se limita, pues, a comprobar si los términos especificados en la consulta están o no presentes en el documento, como en el caso del modelo booleano, sino que la similitud entre ambos se calculan en base a los pesos de los términos involucrados, permitiendo de este modo, por un lado, la existencia de correspondencias parciales, y por otro, el cálculo de grados de similaridad o relevancia conforme a los cuales los documentos pueden ser devueltos por orden de mayor a menor relevancia, facilitando

notablemente el trabajo del usuario, que puede concentrar sus esfuerzos en los primeros documentos devueltos —aquellos más relevantes— o incluso definir umbrales de relevancia por debajo de los cuales un documento no es tenido en consideración. [33]

Sin embargo, antes de calcular el grado de similaridad entre los vectores, es necesario calcular los pesos de los términos. Dichos pesos pueden ser calculados de múltiples maneras, si bien el esquema de pesos *tf – idf* y sus derivados se han convertido en los más populares. [33] En el esquema *tf – idf* básico el peso w_{ij} de un término i en un documento j viene dado por la fórmula

$$w_{ij} = tf_{ij} \times idf_i \quad (4)$$

A la serie de fórmulas para el cómputo de pesos formada por este esquema inicial y las variantes a las que da lugar se las denomina esquemas *tf – idf*.

Los buenos resultados obtenidos con el modelo vectorial, unidos a la simplicidad a nivel de concepto e implementación, su bondad a la hora de aceptar consultas en lenguaje natural, y su capacidad para permitir correspondencias parciales y ordenamiento por relevancia, han hecho de este modelo una de las principales bases sobre la que se han desarrollado gran parte de los experimentos y sistemas en todo el ámbito de la Recuperación de Información [33]. Sus buenas características, unidas al hecho de que sea uno de los modelos de representación más utilizados, le han convertido frecuentemente en el sistema de referencia respecto al cual comparar resultados a la hora de desarrollar nuevos modelos de recuperación [33].

2.9.1.4 Modelo Probabilístico

El modelo probabilístico fue formulado por Stephen Robertson y Sparck Jones en 1977. Este modelo se basa en que en el proceso de RI es intrínsecamente impreciso. Dentro del propio proceso, hay determinados aspectos que son no deterministas, por ejemplo: [34]

- La representación que hace una consulta de la necesidad de información del usuario.
- La representación de los documentos en el sistema.

Teniendo esto en cuenta, el modelo probabilístico postula que la mejor manera de poder representar esto es mediante la teoría de probabilidades. Este modelo intenta estimar la probabilidad de que, dada una consulta q , un documento d sea relevante para esa consulta. Esto se denota como: $P(Rel|d)$ En el modelo se intenta obtener un conjunto de documentos relevantes (denominado R), que deberá maximizar la probabilidad de relevancia. Un documento se considera relevante si su probabilidad de ser relevante $P(Rel|d)$, es mayor que la probabilidad de no ser relevante, $P(noRel|d)$. [34]

Teniendo en cuenta que la probabilidad de relevancia de un documento es estimada a partir de las probabilidades de los términos que lo componen, podemos calcular el grado de similitud de un documento con una consulta.

El modelo probabilístico se basa en un proceso iterativo. Este proceso se inicia con un primer conjunto de documentos relevantes, que es paulatinamente recalculado en función de la información que proporciona el usuario de aquellos documentos que considera relevantes y no relevantes. [34]

La principal ventaja de este modelo consiste en que constituye un modelo teórico importante que permite representar el proceso de RI. Además, el conjunto resultante proporciona una ordenación de los documentos en base a su probabilidad de relevancia.

Dentro de sus desventajas, cabe destacar la necesidad de iniciar el modelo a partir de una primera estimación del conjunto de documentos relevantes, y el hecho de que no se tiene en cuenta el número de veces que cada término aparece en un documento a la hora de estimar su probabilidad de relevancia. [34]

CAPITULO 2

3. MARCO METODOLÓGICO

3.1 Diseño de Estudio

El estudio se realizara en tres fases que comprenden la adquisición de información sobre la deserción estudiantil en la Universidad de Pamplona, luego la construcción del perfil del estudiante y por último el modelado del aplicativo móvil.

La primera fase consiste en la recopilación de información existente sobre la tasa de deserción estudiantil en las facultades de la Universidad de Pamplona en el I y II semestre, del año 2014, dicha información será solicitada a las oficinas de Planeación Universitaria, Registro y Control Académico y Bienestar Universitario de la Universidad de Pamplona.

Una segunda fase que comprende la construcción del perfil del estudiante a través de métodos de recuperación de información, y estadísticos de factores de deserción asociados a la Universidad de Pamplona.

La tercera y última fase corresponderá al modelado del aplicativo móvil que determinara los factores de deserción más influyentes en las facultades de la Universidad de Pamplona, utilizando nuevas tecnologías como lo son JQuery Mobile y PhoneGap.

La investigación es descriptiva y sigue el método inductivo, pues, se busca detallar características en forma general y a partir del análisis de la situación particular de los fenómenos de deserción y permanencia académica al interior de las Facultades de la Universidad de Pamplona.

Las fuentes de información serán principalmente, la suministrada por Planeación Universitaria, Registro y Control Académico y Bienestar Universitario, y la que se encuentra registrada en el Sistema para la Prevención de la Deserción en las Instituciones de Educación Superior (SPADIES).

Se hará uso de otras fuentes de información relacionada con el tema, como tesis de grado, estudios realizados por otras universidades y documentos del Ministerio de Educación Nacional y del Consejo Nacional de Acreditación.

3.2 Deserción Estudiantil en la Universidad de Pamplona

La Universidad de Pamplona nació en 1960, como institución privada, bajo el liderazgo de Presbítero José Faría Bermúdez. En 1970 fue convertida en Universidad Pública del orden departamental, mediante el decreto No 0553 del 5 de Agosto de 1970 y en 1971 el Ministerio de Educación Nacional la facultó para otorgar títulos profesionales según Decreto No. 1550 del 13 de Agosto.

Durante los años sesenta y setenta, la Universidad creció en la línea de formación de licenciados y licenciadas, en la mayoría de las áreas que debían ser atendidas en el sistema educativo: Matemáticas, Química, Biología, Ciencias Sociales, Pedagogía, Administración Educativa, Idiomas Extranjeros, Español – Literatura y Educación Física.

En los años ochenta la Institución dio el salto hacia la formación profesional en otros campos del saber, etapa que inició a finales de esa década con el Programa de Tecnología de Alimentos.

Posteriormente en los años noventa fueron creados en los campos de las Ciencias Naturales y Tecnológicas, los Programas de Microbiología con énfasis en Alimentos, las Ingenierías de Alimentos y Electrónica y la Tecnología en Saneamiento Ambiental. En el campo de la Ciencias Socioeconómicas, el programa de Administración de Sistemas, inicialmente como tecnología y luego a nivel profesional.

Hoy, la Universidad ha ampliado significativamente su oferta educativa logrando atender nuevas demandas de formación profesional, generadas en la región o en la misma evolución de la ciencia, el arte, la técnica y las humanidades. Cumple esta tarea desde todos los niveles de la Educación Superior: pregrado, posgrado y educación continuada, y en todas las modalidades educativas: presencial, a distancia y con apoyo virtual; lo cual, le

ha permitido proyectarse tanto en su territorio como en varias regiones de Colombia y del Occidente de nuestro país vecino y hermano Venezuela.

Pese a que la educación superior en los últimos años se han caracterizado por aumentos de cobertura e ingreso de estudiantes nuevos, el número de alumnos que logra culminar sus estudios es bajo, dejando entrever que una gran parte de estos los abandona, primordialmente en los primeros semestres.

Como información preliminar tenemos que, durante el periodo comprendido entre el I semestre del 2014 y el II semestre 2014, Las Facultades de la Universidad de Pamplona presentaron el comportamiento que se mostrará a continuación en cuanto al total de matriculados y el número de estudiantes que abandonaron su programa académico por algún motivo.

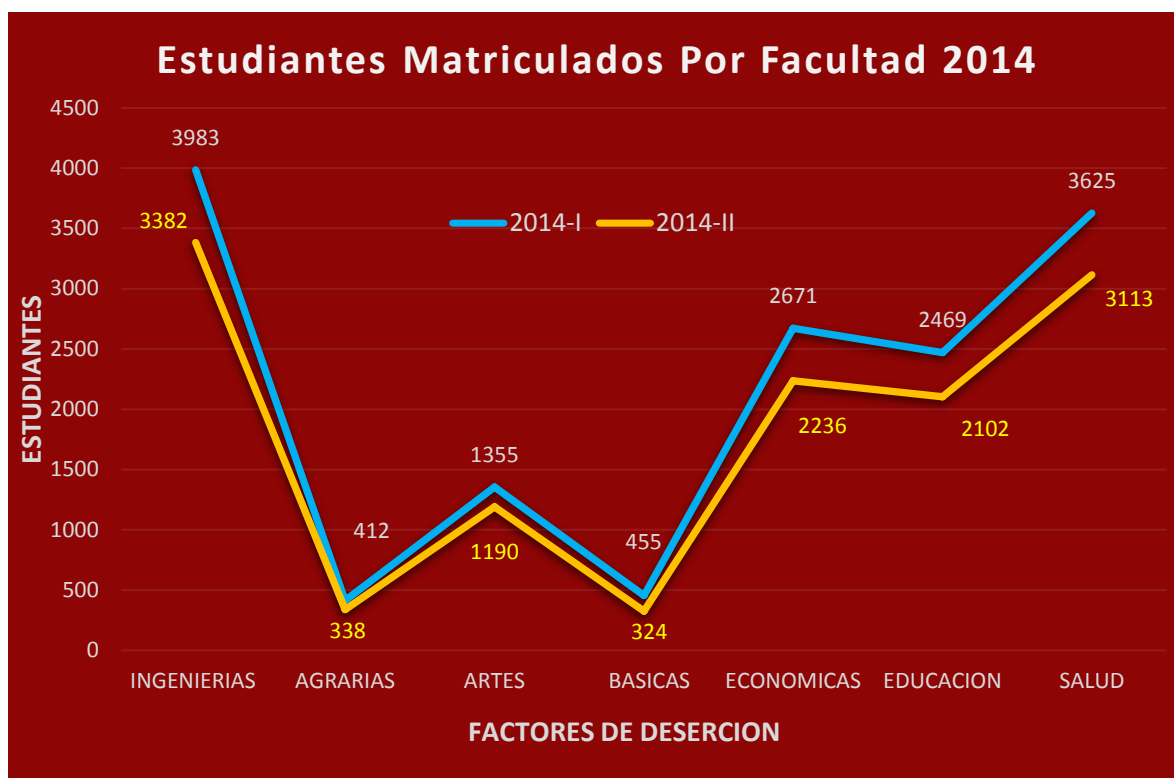


Gráfico 1: Número de estudiantes matriculados por facultad de la UPA desde el I Sem. 2014 hasta el II Sem. 2014
Fuente: SPADIES UNIVERSIDAD DE PAMPLONA 2014.

En el Grafico 1, podemos observar el número aproximado de estudiantes que ingresaron en los semestres I y II del 2014 por facultad. Se observa un comportamiento regular en el número de matrículas.

Mientras que la deserción presenta algunas variaciones que van desde un mínimo de 6.675 % para el II Sem. 2014 frente a un máximo de 7,785% para el I Sem. 2014. Dicho comportamiento se puede observar de mejor manera en el siguiente gráfico:

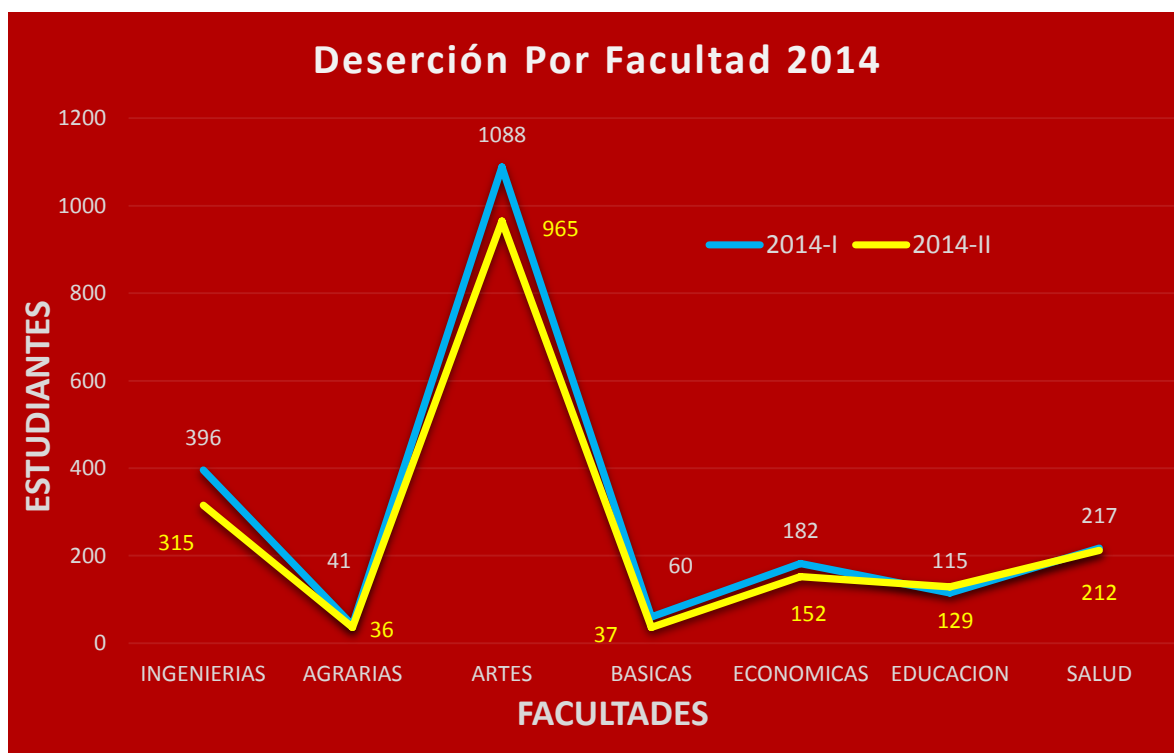


Grafico 2: Número de estudiantes desertores por facultad de la UPA desde el I Sem. 2014 hasta el II Sem. 2014 Fuente: SPADIES UNIVERSIDAD DE PAMPLONA 2014.

El II Sem. 2014 presenta el menor nivel de deserción, 6,675% (1846 desertores), mientras que el I Sem. 2014 reporta el nivel más alto 7,785% (2153 desertores). El total de alumnos que presentó algún tipo de deserción durante estos dos semestres es de 3999 estudiantes.

En cuanto a las Facultades, cada una presenta un comportamiento diferente, siendo el de la Facultad de Artes y Humanidades el que presenta un mayor porcentaje de deserción

acumulada para los periodos de estudio. A continuación se presenta el análisis detallado en cifras para cada una de las facultades.

3.2.1 Facultad de Ciencias Agrarias

La Facultad de Ciencias Agrarias de la Universidad de Pamplona, orienta sus actividades académicas a la formación de profesionales y tecnólogos integrales con compromiso social y ambiental; promoviendo la formación del talento humano y el desarrollo tecnológico como elementos fundamentales para el desarrollo sustentable y el mejoramiento de la calidad de vida en el sector agropecuario y la sociedad colombiana.

Con un promedio de 390 alumnos matriculados por semestre, presenta una deserción promedio del 1,925% para los dos semestres que comprende este estudio.

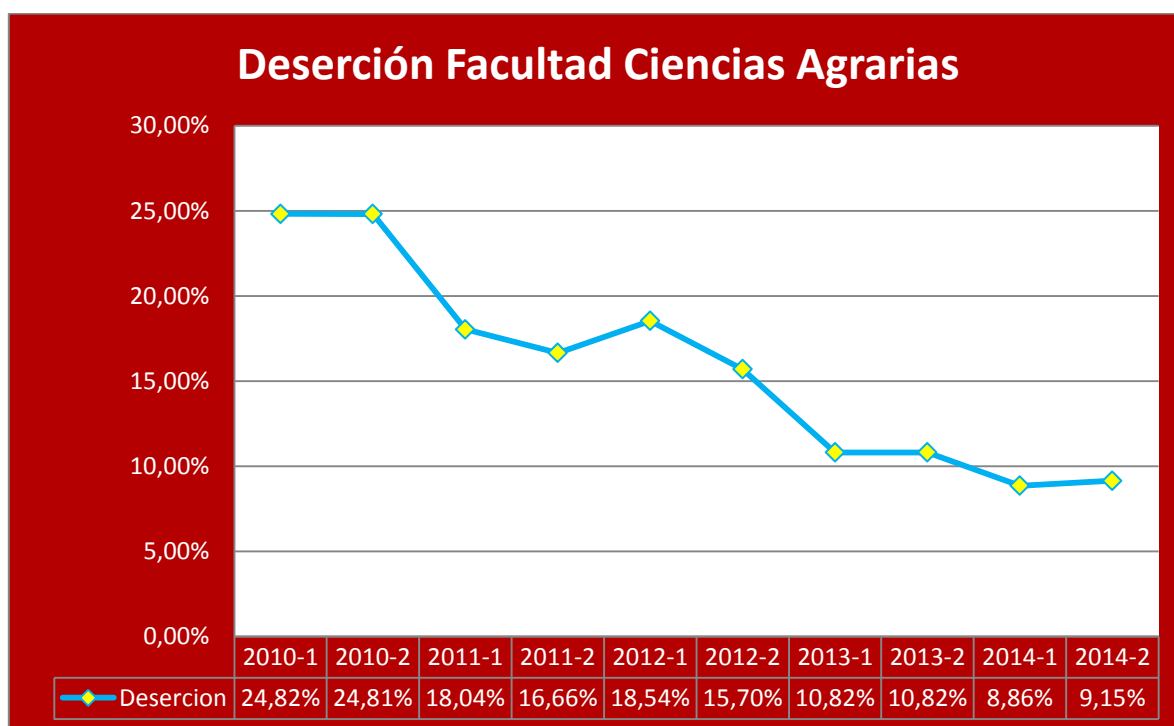


Grafico 3: Deserción en la facultad de Ciencias Agrarias UPA desde el I Sem. 2010 hasta el II Sem. 2014 Fuente: SPADIES UNIVERSIDAD DE PAMPLONA 2014.

En el Grafico 3 se puede observar que el comportamiento de la deserción al interior de la Facultad de Ciencias Agrarias es regular, pero ha venido presentando un decremento durante los semestres objeto de este estudio.

3.2.2 Facultad de Artes y Humanidades

La Facultad de Artes y Humanidades fue creada por el Acuerdo No. 051 del 16 de julio de 1999 del Honorable Consejo Superior de la Universidad de Pamplona y está conformada actualmente por los programas de pregrado de Artes Visuales, Comunicación Social, Derecho, Licenciatura en Educación Artística y Cultural, Música y Filosofía y de posgrado la Especialización en Educación Artística y la Maestría en Paz, Desarrollo y Resolución de Conflictos cuenta además con sedes en Pamplona, Cúcuta y Villa del Rosario.

Con un promedio de 1.419 alumnos matriculados por semestre, presenta una deserción promedio del 28,13% para los dos semestres que comprende este estudio.

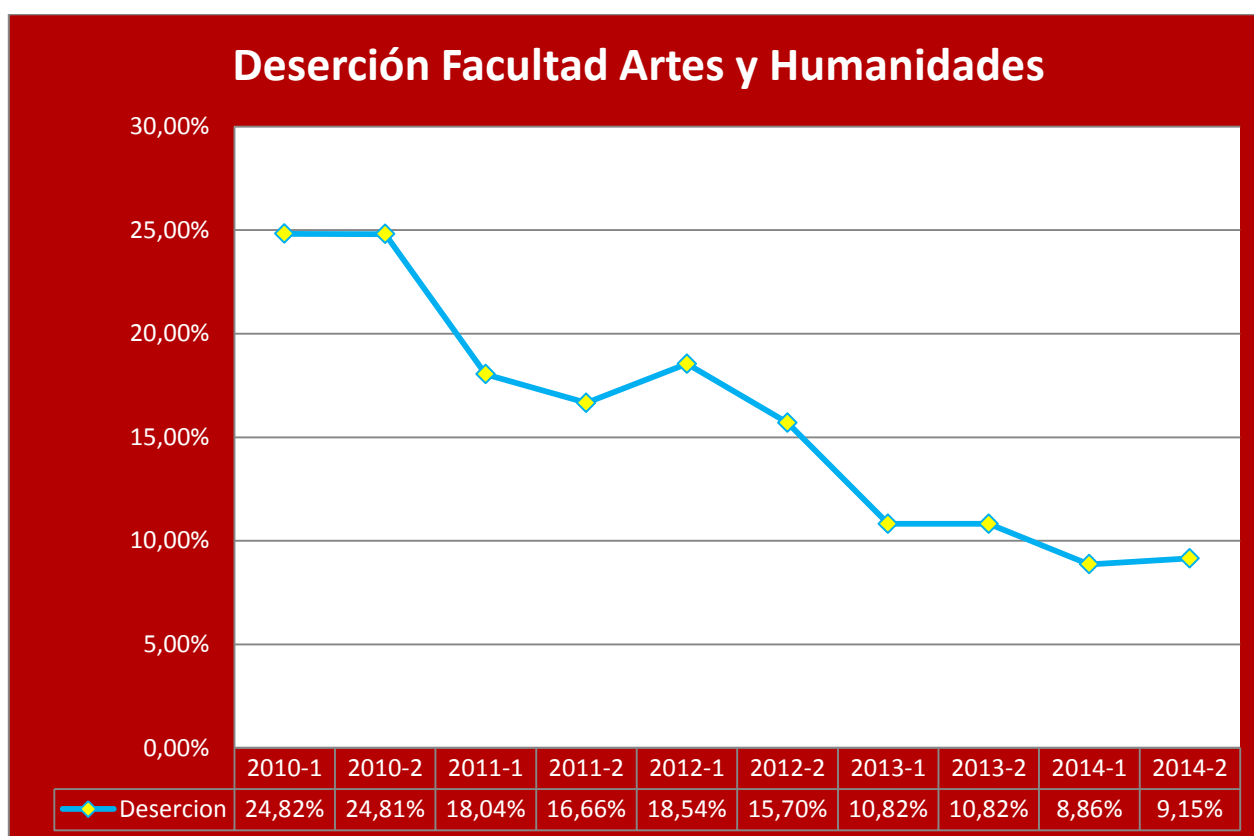


Grafico 4: Deserción en la facultad de Artes y Humanidades UPA desde el I Sem. 2010 hasta el II Sem. 2014 Fuente: SPADIES UNIVERSIDAD DE PAMPLONA 2014.

En el Grafico 4 se puede observar que el comportamiento de la deserción al interior de la Facultad de Artes y Humanidades es normal, pero ha venido presentando un decremento durante los semestres objeto de este estudio.

3.2.3 Facultad de Ciencias Básicas

Facultad de Ciencias Básicas, efectivamente para ofrecer programas en formación docente, programas de licenciatura en Química – Biología, Biología - Química, en Matemáticas – Física y en Física - Matemáticas, esto consolidó una comunidad académica y científica para ese momento donde se integraban profesores y docentes para desarrollar temáticas de investigación en el área de la docencia, pero también en el área de formación en las diferentes disciplinas como la Química, Biología, Física y Matemáticas.

Con un promedio de 430 alumnos matriculados por semestre, presenta una deserción promedio del 25,63% para los dos semestres que comprende este estudio.

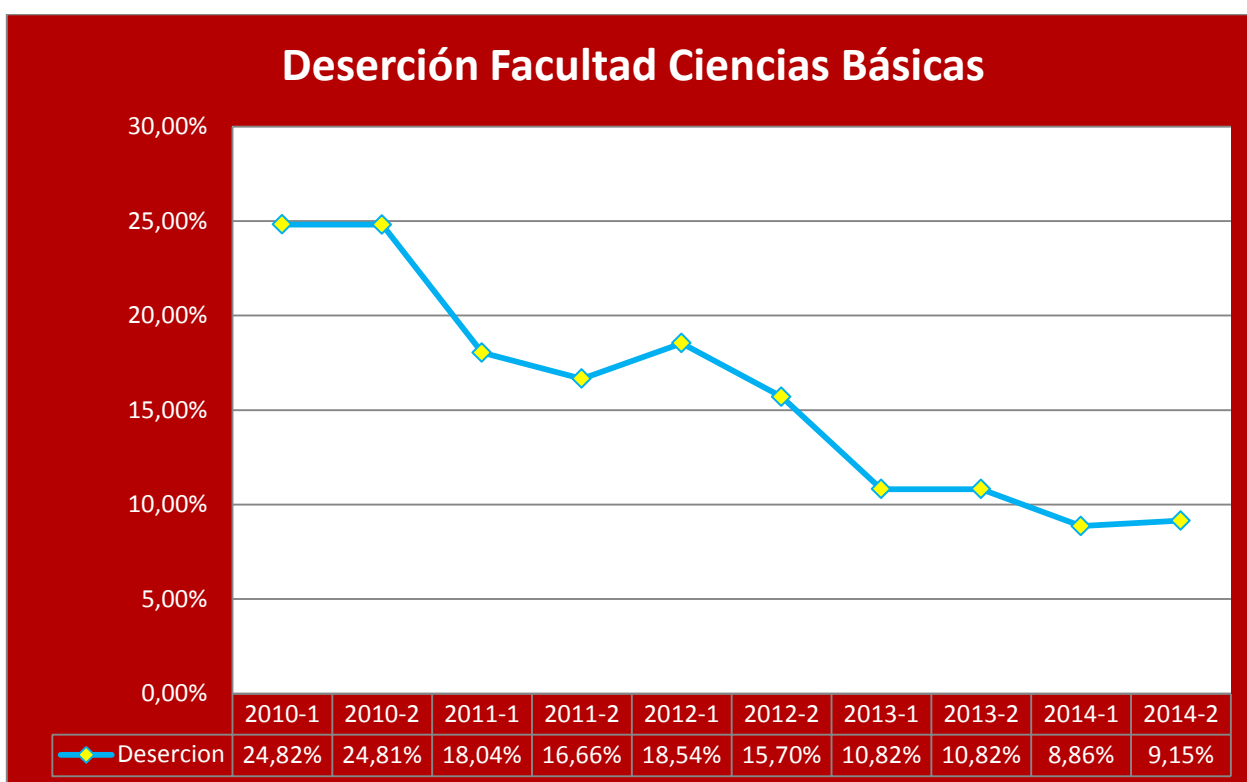


Grafico 5: Deserción en la facultad de Ciencias Básicas UPA desde el I Sem. 2010 hasta el II Sem. 2014 Fuente: SPADIES
UNIVERSIDAD DE PAMPLONA 2014.

En el Grafico 5 se puede observar que el comportamiento de la deserción al interior de la Facultad de Artes y Humanidades es regular, pero ha venido presentando un decremento durante los semestres objeto de este estudio.

3.2.4 Facultad de Ciencias Económicas y Empresariales

La Facultad de Ciencias Económicas y Empresariales desde su fecha de inicio, se ha destacado en la Universidad por su capacidad de trabajo, visualizada en el cubrimiento de necesidades académicas a la población estudiantil de trece departamentos mediante los CREAD. Adicionalmente se ofertan programas en la sede principal de Pamplona y en la subsede de Villa del Rosario donde están las jornadas diurnas y nocturnas. En esta facultad existen tres departamentos que pertenecen a las ciencias sociales, estos son: Administración, Contaduría Pública y Economía.

Con un promedio de 5.441 alumnos matriculados por semestre, presenta una deserción promedio del 8,35% para los dos semestres que comprende este estudio.

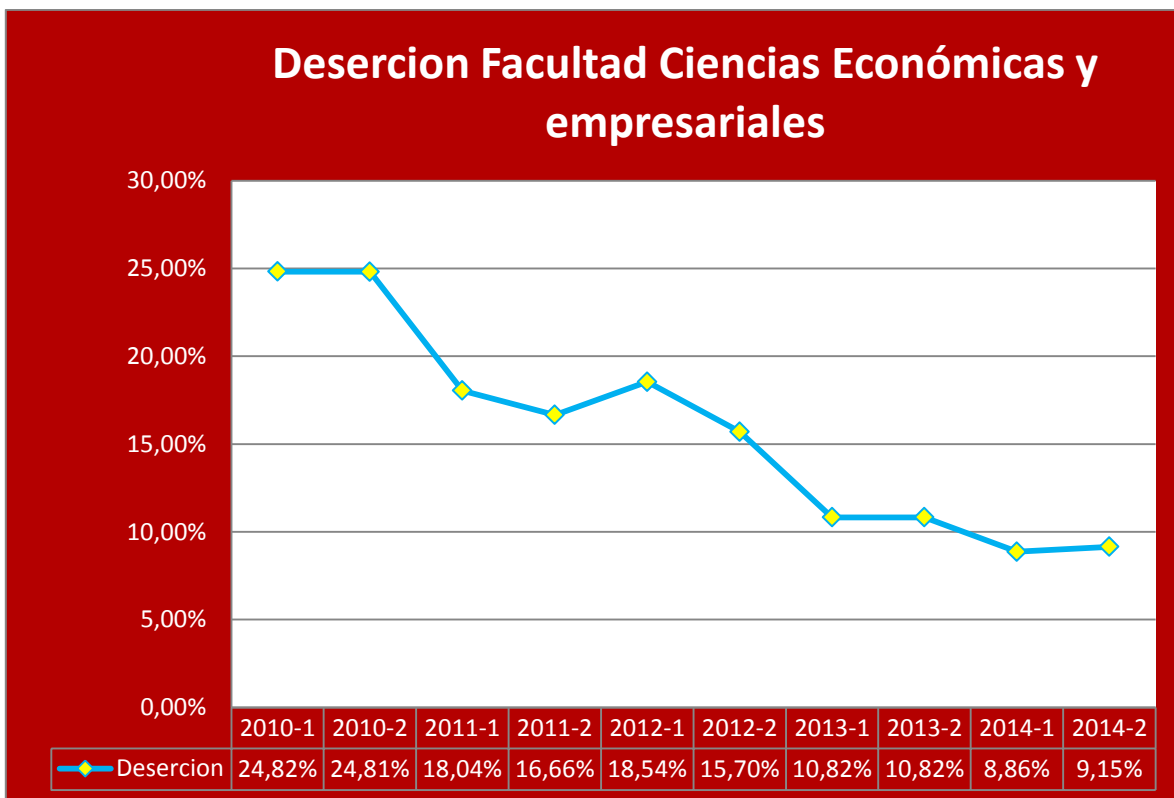


Grafico 6: Deserción en la facultad de Ciencias Económicas y Empresariales UPA desde el I Sem. 2010 hasta el II Sem. 2014 Fuente: SPADIES UNIVERSIDAD DE PAMPLONA 2014.

En el Grafico 6 se puede observar que el comportamiento de la deserción al interior de la Facultad de Ciencias Económicas y Empresariales es usual, pero ha venido presentando un decremento durante los semestres objeto de este estudio.

3.2.5 Facultad de Educación

La historia y el desarrollo de la Facultad de Educación ha estado íntimamente ligada a la historia y el desarrollo de la Universidad y su tarea de formar maestros y maestras ha constituido, sin duda, uno de los referentes más sólidos de la presencia de institucional, tanto en el ámbito regional como nacional. La calidad de sus egresados, en todas las áreas contempladas por la ley, es uno de sus más importantes logros y la mejor expresión del importante trabajo que los docentes adscritos a la Facultad realizan.

Con un promedio de 4.165 alumnos matriculados por semestre, presenta una deserción promedio del 6,10% para los dos semestres que comprende este estudio.

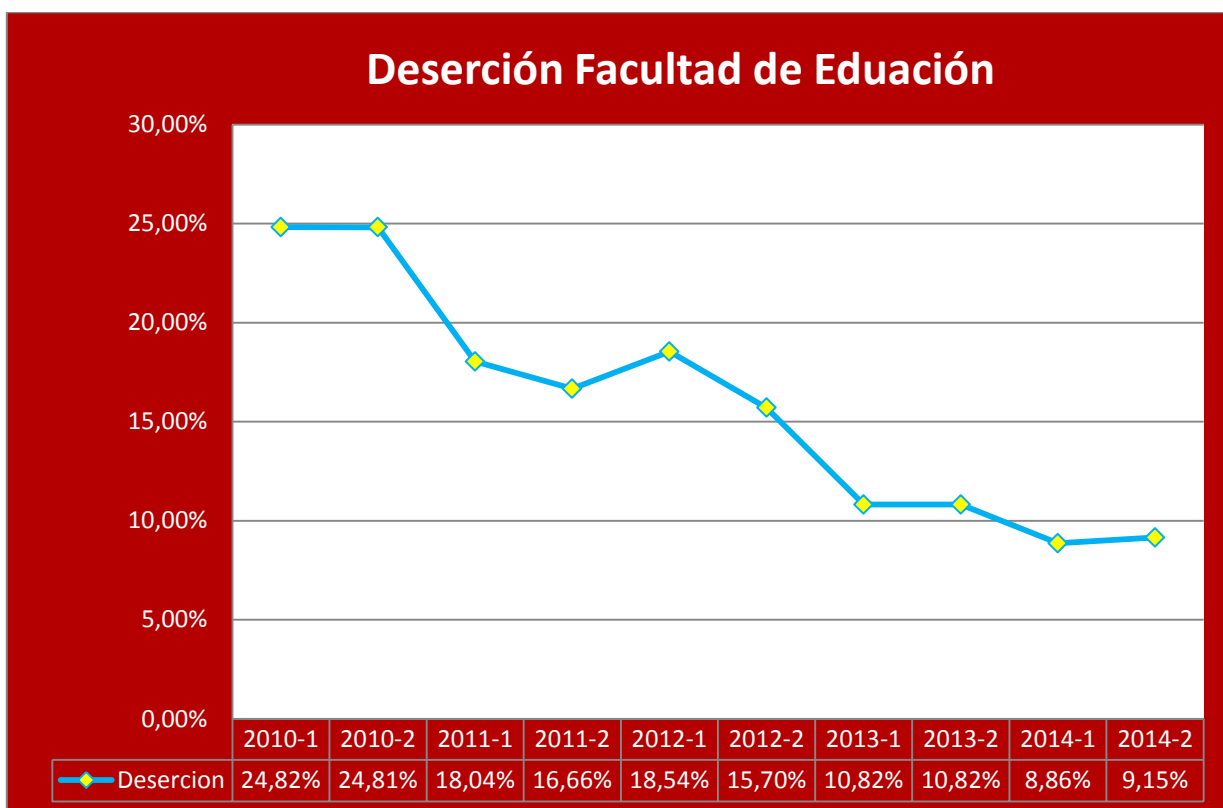


Grafico 7: Deserción en la facultad de Educación UPA desde el I Sem. 2010 hasta el II Sem. 2014 Fuente: SPADIES UNIVERSIDAD DE PAMPLONA 2014.

En el Grafico 7 se puede observar que el comportamiento de la deserción al interior de la Facultad de Educación es regular, pero ha venido presentando un decremento durante los semestres objeto de este estudio.

3.2.6 Facultad de Ingenierías y Arquitectura

A finales de los años noventa se crea la Facultad de Ciencias Naturales y Tecnológicas conformada por los antiguos programas de Ingenierías Electrónica y de Alimentos, el programa de Microbiología con énfasis en Alimentos y las Tecnologías en Saneamiento Ambiental y en Alimentos, así como los programas creados en estos años de Ingenierías Ambiental, de Telecomunicaciones, Mecánica, Mecatrónica, Sistemas, Industrial y el programa profesional de Física. En el año 2003, el programa de Microbiología con Énfasis en Alimentos se traslada a la Facultad de Ciencias Básicas y se incorpora a ésta el programa de Arquitectura, conformándose así la actual Facultad De Ingenierías Y Arquitectura.

Con un promedio de 3.283 alumnos matriculados por semestre, presenta una deserción promedio del 17,77% para los dos semestres que comprende este estudio.

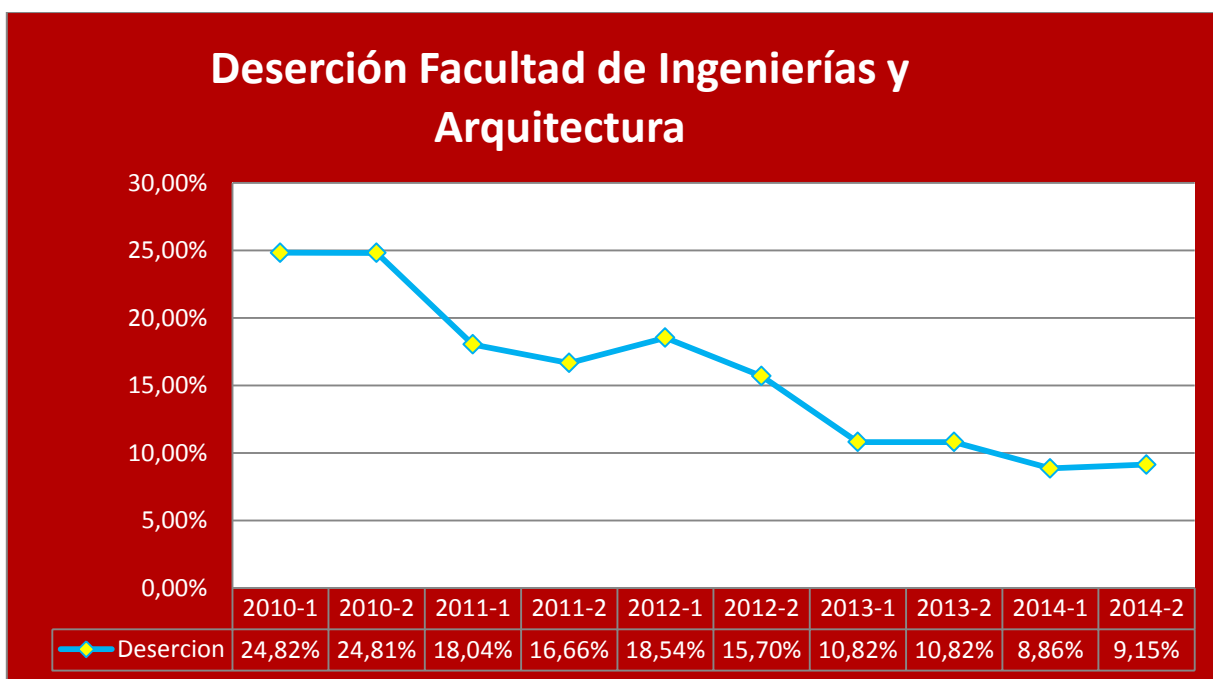


Grafico 8: Deserción en la facultad de Ingenierías y Arquitectura UPA desde el I Sem. 2010 hasta el II Sem. 2014
Fuente: SPADIES UNIVERSIDAD DE PAMPLONA 2014.

En el Grafico 8 se puede observar que el comportamiento de la deserción al interior de la Facultad de Educación es regular, pero ha venido presentando un decremento durante los semestres objeto de este estudio.

3.2.7 Facultad de Salud

La facultad de salud de la Universidad de Pamplona tiene como propósito Formar talento humano en salud con vocación de servicio, competente en la apropiación y transferencia de conocimiento desde la investigación, desarrollo e innovación, comprometido con el desarrollo social sostenible, dirigido a la resolución de situaciones en salud, articuladas a las Políticas de Salud Pública, Atención Primaria en Salud y la atención humanizada.

Con un promedio de 4.291 alumnos matriculados por semestre, presenta una deserción promedio del 12,07% para los dos semestres que comprende este estudio.

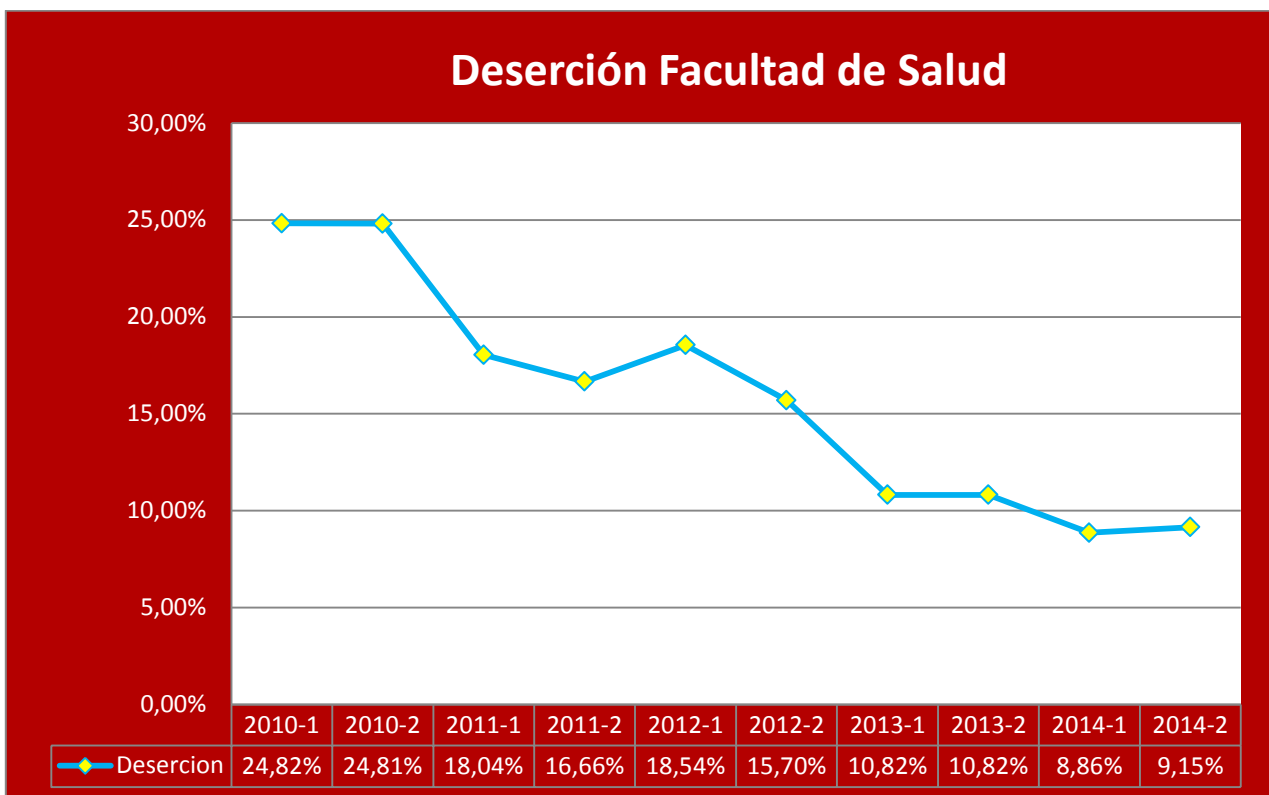


Gráfico 9: Deserción en la facultad de Salud UPA desde el I Sem. 2010 hasta el II Sem. 2014 Fuente: SPADIES UNIVERSIDAD DE PAMPLONA 2014.

En el Gráfico 9 se puede observar que el comportamiento de la deserción al interior de la Facultad de Educación es regular, pero ha venido presentando un decremento durante los semestres objeto de este estudio.

3.3 Factores de Deserción en la Universidad de Pamplona

El Ministerio de Educación Nacional ha liderado, en conjunto con las instituciones de educación superior, el diseño y la operación de una metodología de seguimiento de la deserción estudiantil en educación superior que se concreta en el Sistema para la Prevención y Análisis de la Deserción en las Instituciones de Educación Superior (SPADIES). Esta herramienta permite una visión sectorial del fenómeno, análisis sobre cada una de las variables asociadas a la deserción por cada uno de los estudiantes, así como seguimiento y evaluación a los resultados de las estrategias implementadas para disminuir la deserción.

Dentro de las Facultades de la Universidad de Pamplona los Factores de la deserción son variados, a continuación se presentará un análisis relacionado con este aspecto, obtenido de la información contenida en la Oficina de Planeación, Bienestar Universitario y Admisiones, Registro y Control Académico además la que se encuentra registrada en el Sistema para la Prevención de la Deserción en las Instituciones de Educación Superior (SPADIES).

3.3.1 Facultad de Ciencias Agrarias

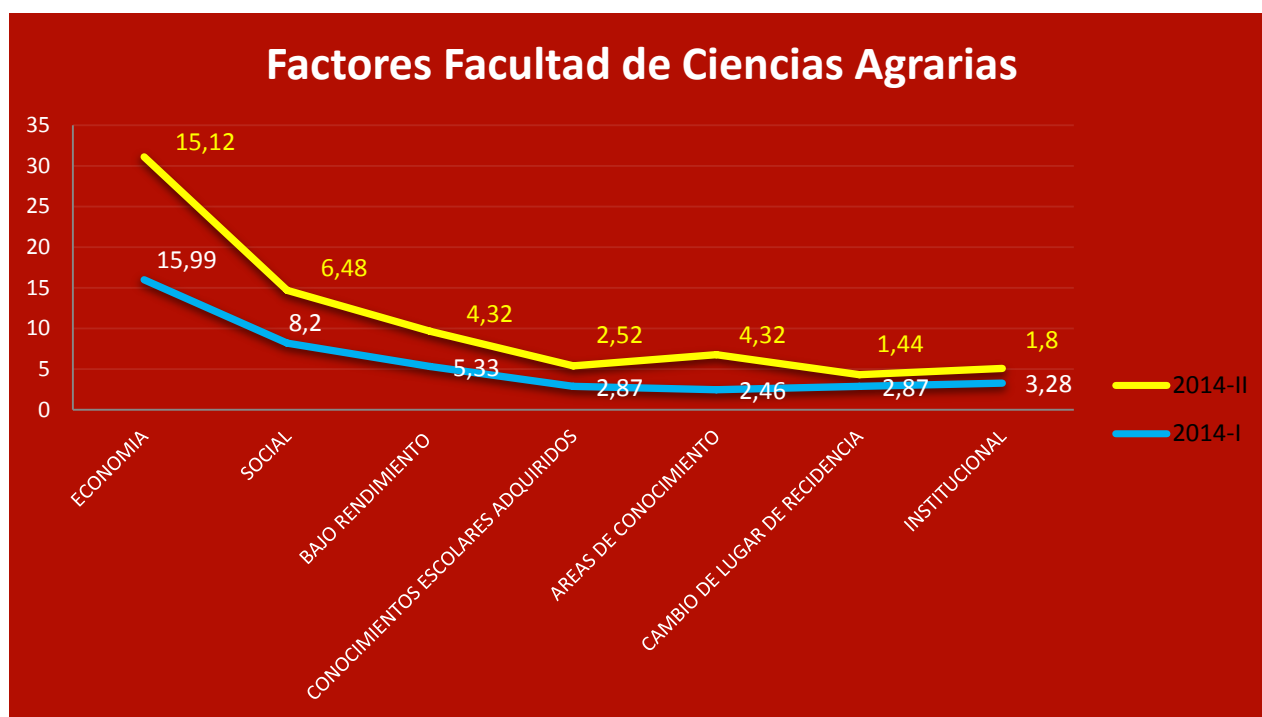


Grafico 10: Factores de Deserción en la facultad de Ciencias Agrarias UPA desde el I Sem. 2014 y II Sem. 2014 Fuente: SPADIES UNIVERSIDAD DE PAMPLONA 2014.

Dentro de los retiros que se oficializan en la UNIVERSIDAD DE PAMPLONA para la facultad de Ciencias Agrarias el motivo de mayor frecuencia está representado en aspectos de tipo Económico con el 39% del total de los registros. Siguiendo el orden de importancia el factor social tiene un peso del 20%, seguido encontramos razones como Bajo rendimiento Académico con el 13%, Áreas de Conocimiento con un 7%, Conocimientos Escolares Adquiridos con un 6%, y por último el factor Institucional y Cambio de lugar de residencia con 7% y 8% respectivamente.

3.3.2 Facultad de Artes y Humanidades

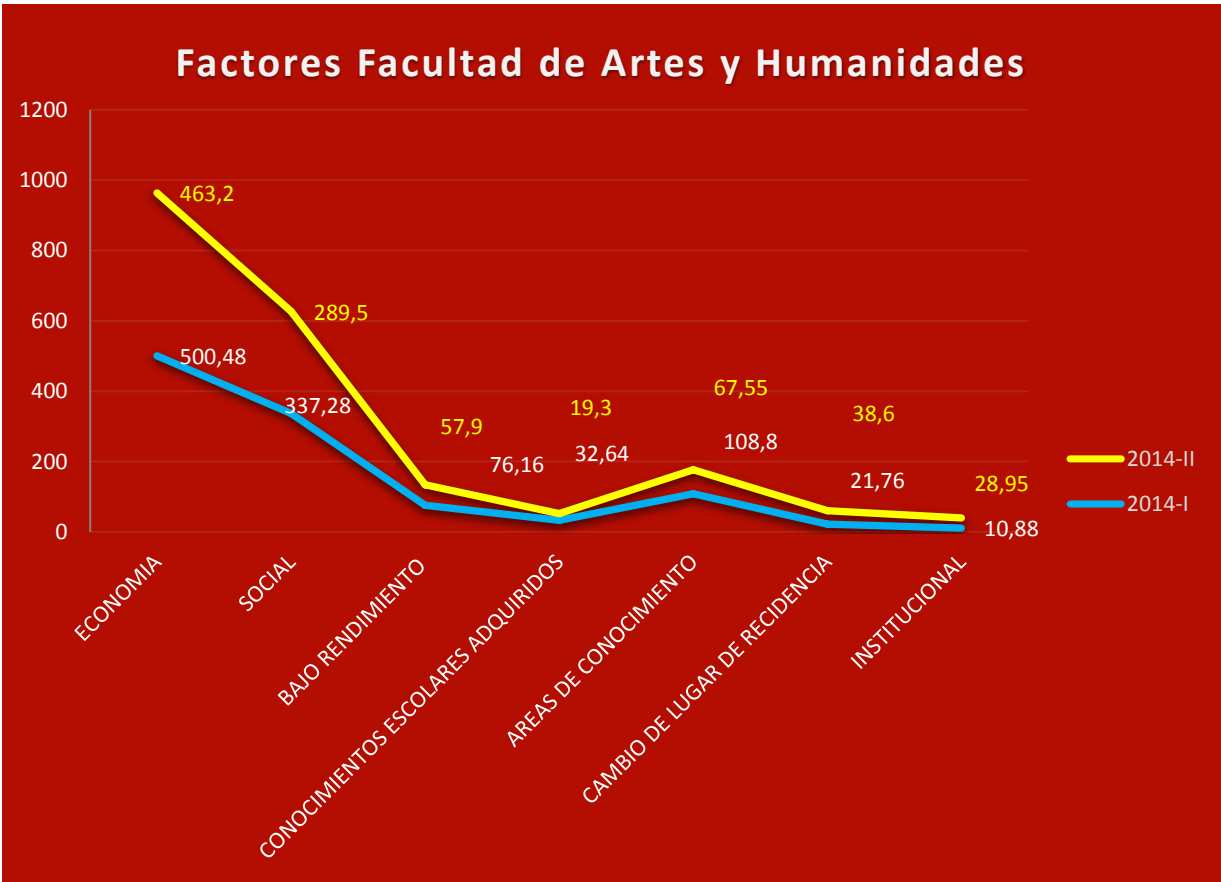


Grafico 11: Factores de Deserción en la facultad de Artes y Humanidades UPA desde el I Sem. 2014 y II Sem. 2014
 Fuente: SPADIES UNIVERSIDAD DE PAMPLONA 2014.

En la Facultad de Artes y Humanidades los motivos de tipo económico son los que se presentan con mayor frecuencia 48%, seguidos del factor social con 30%, el bajo rendimiento académico de los estudiantes también tiene alta incidencia en las causa de

retiro, con 6%; los conocimientos escolares adquiridos tienen una incidencia del 2%, seguido encontramos el factor que evalúa las áreas de conocimiento que tuvo una influencia del 7%, para terminar los factores de cambio de lugar de residencia e institucionales con 4% y 3% respectivamente.

3.3.2 Facultad de Ciencias Básicas

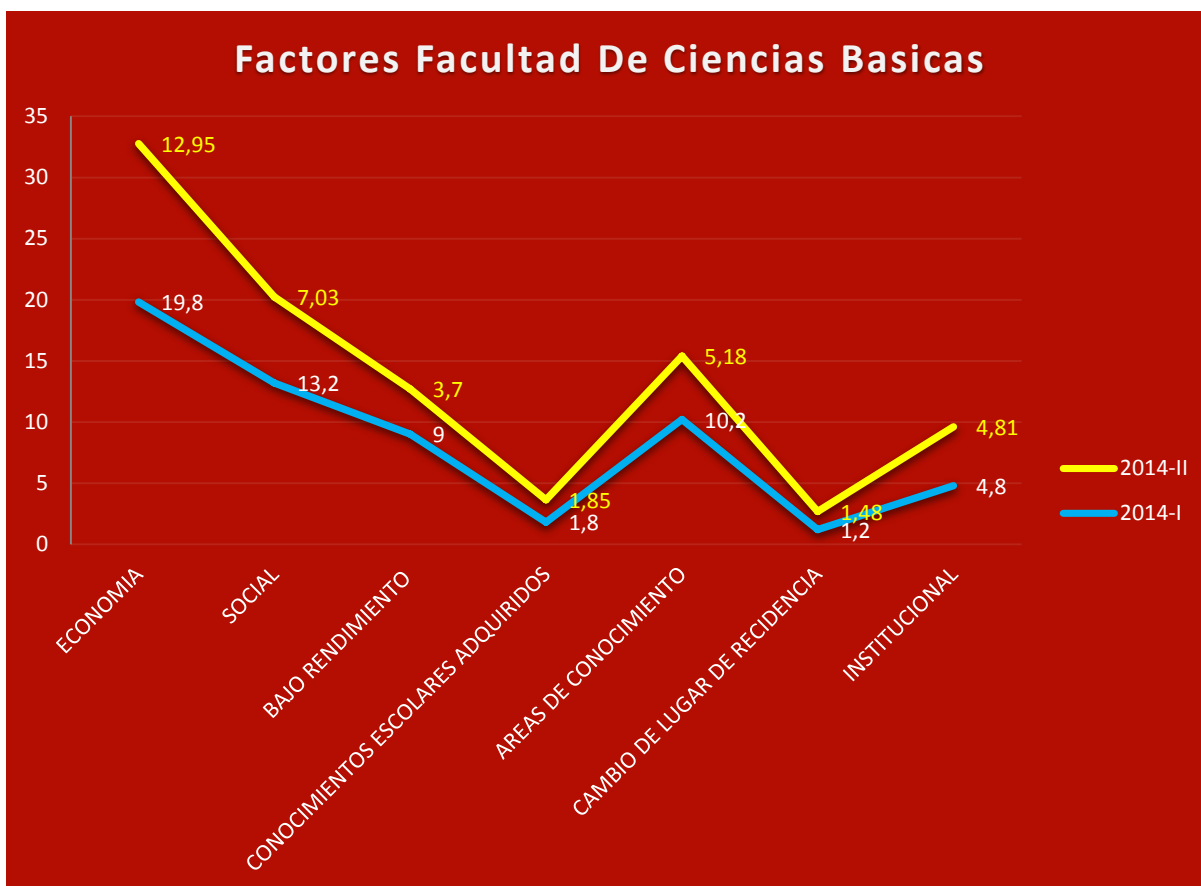


Grafico 12: Factores de Deserción en la facultad de Ciencias Básicas UPA desde el I Sem. 2014 y II Sem. 2014 Fuente: SPADIES UNIVERSIDAD DE PAMPLONA 2014.

Dentro de La facultad de Ciencias Básicas el motivo de mayor frecuencia está representado en aspectos de tipo Económico con el 33% del total de los registros. Siguiendo el orden de importancia el factor social tiene un peso del 22%, seguido encontramos razones como Bajo rendimiento Académico con el 15%, Áreas de

Conocimiento con un 3%, Conocimientos Escolares Adquiridos con un 17%, y por último el factor Institucional y Cambio de lugar de residencia con 2% y 8% respectivamente.

3.3.2 Facultad de Ciencias Económicas y Empresariales

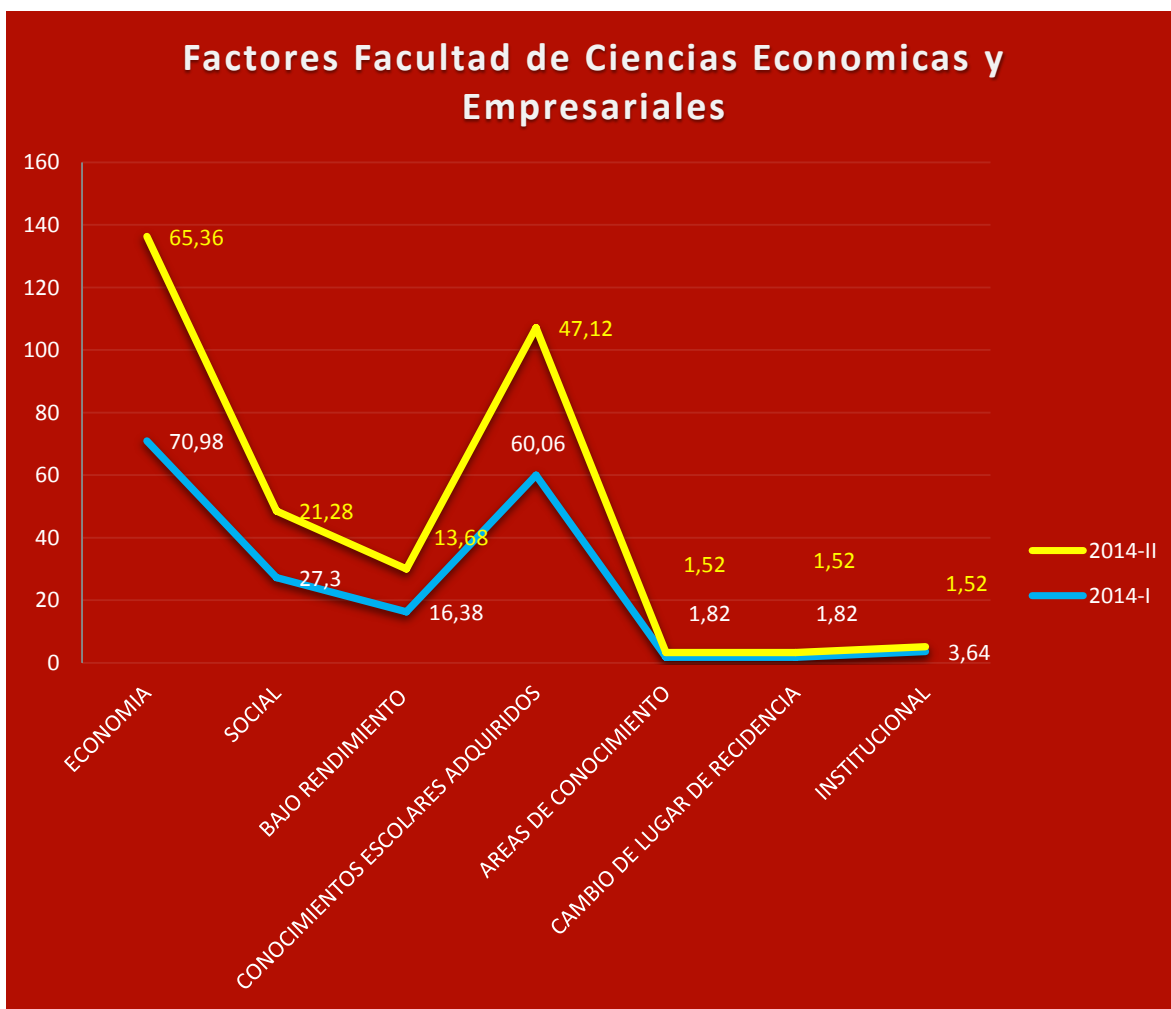


Grafico 13: Factores de Deserción en la facultad de Ciencias Económicas y Empresariales UPA desde el I Sem. 2014 y II Sem. 2014 Fuente: SPADIES UNIVERSIDAD DE PAMPLONA 2014.

En la Facultad de Ciencias Económicas Y Empresariales los motivos de tipo económico son los que se presentan con mayor frecuencia 43%, seguidos del factor social con 14%, el bajo rendimiento académico de los estudiantes también tiene alta incidencia en las causa de retiro, con 9%; los conocimientos escolares adquiridos tienen una incidencia del 31%, seguido encontramos el factor que evalúa las áreas de conocimiento que tuvo una

influencia del 1%, para terminar los factores de cambio de lugar de residencia e institucionales con 1% y 1% respectivamente.

3.3.2 Facultad de Educación

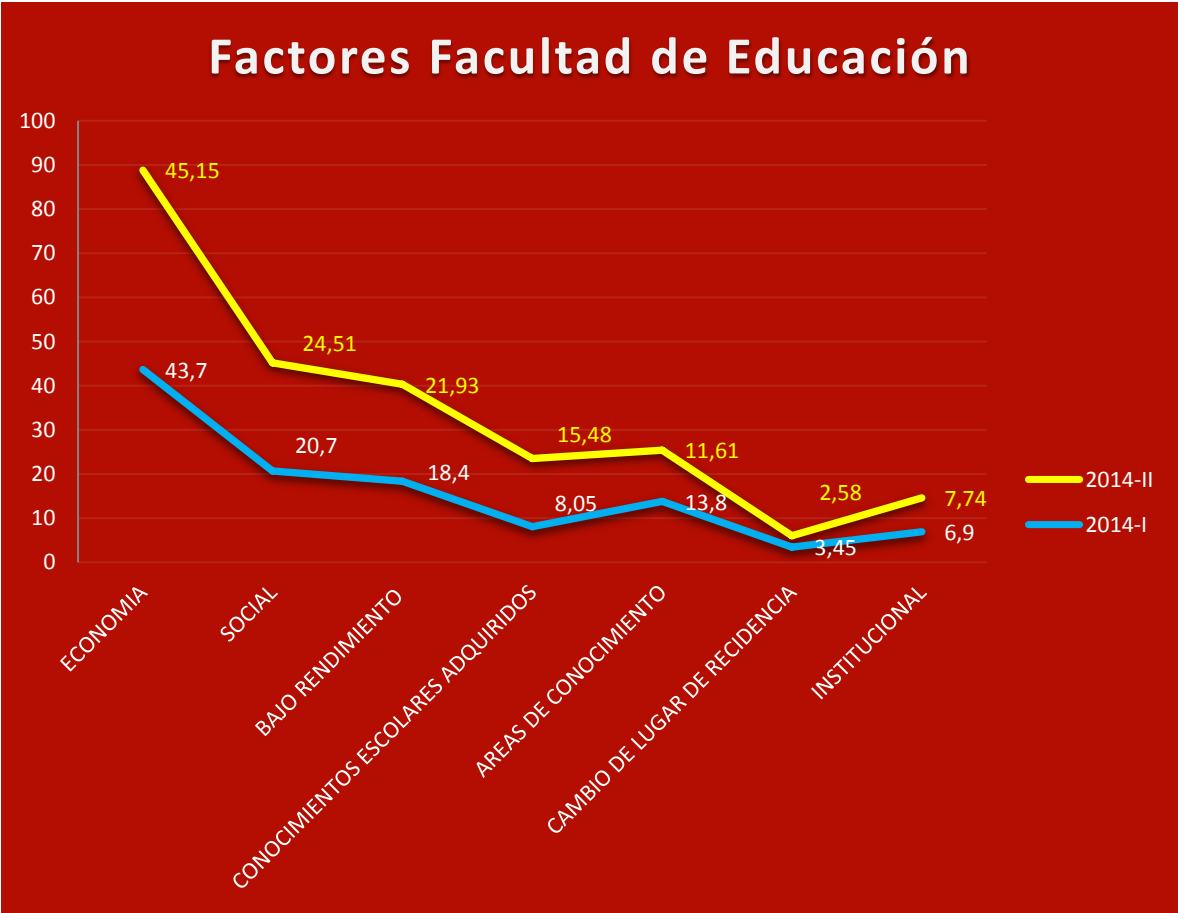


Grafico 14: Factores de Deserción en la facultad de Educación UPA desde el I Sem. 2014 y II Sem. 2014 Fuente: SPADIES UNIVERSIDAD DE PAMPLONA 2014.

En la facultad de Educación el motivo de mayor frecuencia está representado en aspectos de tipo Económico con el 35% del total de los registros. Siguiendo el orden de importancia el factor social tiene un peso del 19%, seguido encontramos razones como Bajo rendimiento Académico con el 17%, Áreas de Conocimiento con un 12%, Conocimientos

Escolares Adquiridos con un 9%, y por último el factor Institucional y Cambio de lugar de residencia con 2% y 6% respectivamente.

3.3.2 Facultad de ingenierías y Arquitectura

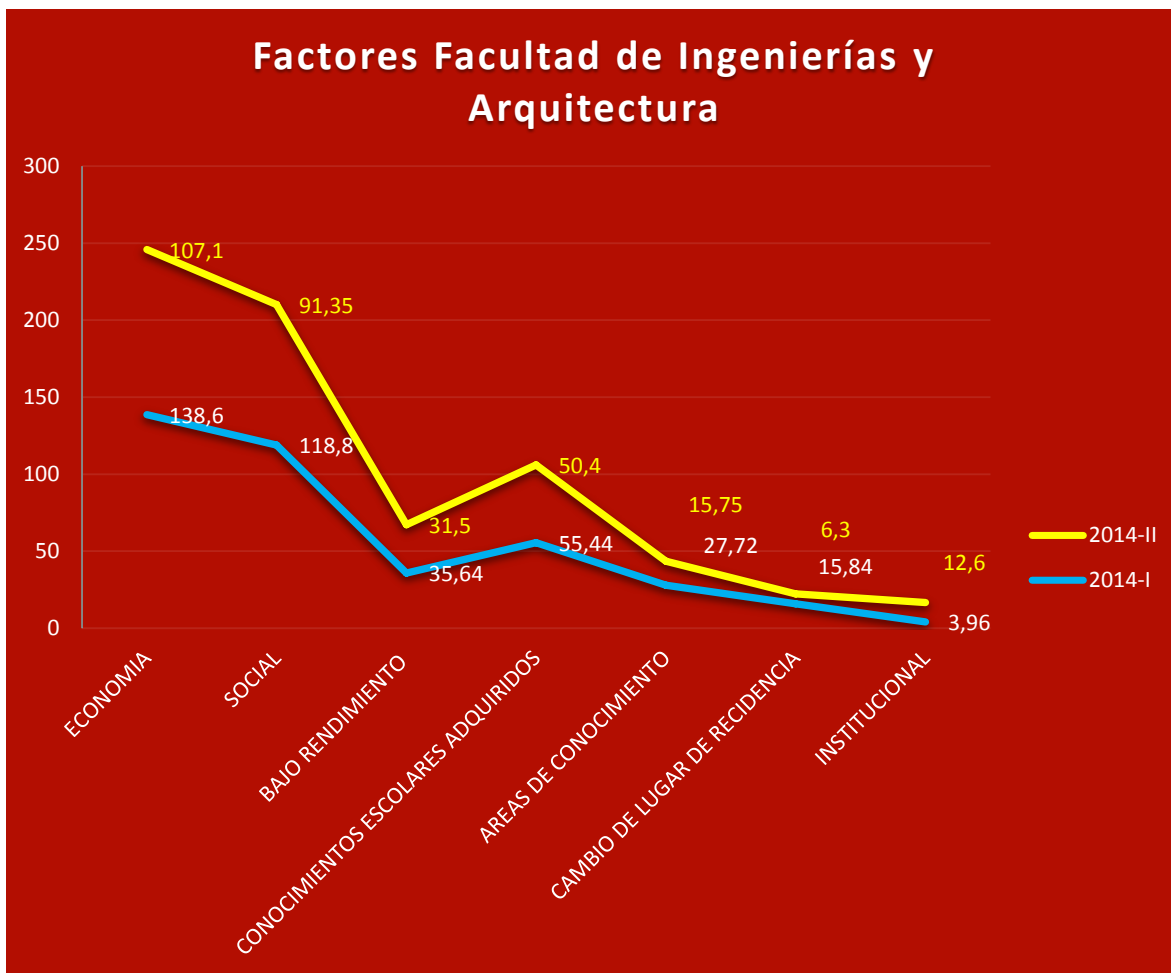


Grafico 15: Factores de Deserción en la facultad de Ingenierías y Arquitectura UPA desde el I Sem. 2014 y II Sem. 2014
Fuente: SPADIES UNIVERSIDAD DE PAMPLONA 2014.

En la Facultad de Ingenierías y Arquitectura los motivos de tipo económico son los que se presentan con mayor frecuencia 34%, seguidos del factor social con 29%, el bajo rendimiento académico de los estudiantes también tiene alta incidencia en las causa de retiro, con 10%; los conocimientos escolares adquiridos tienen una incidencia del 16%, seguido encontramos el factor que evalúa las áreas de conocimiento que tuvo una

influencia del 5%, para terminar los factores de cambio de lugar de residencia e institucionales con 2% y 4% respectivamente.

3.3.2 Facultad de Salud

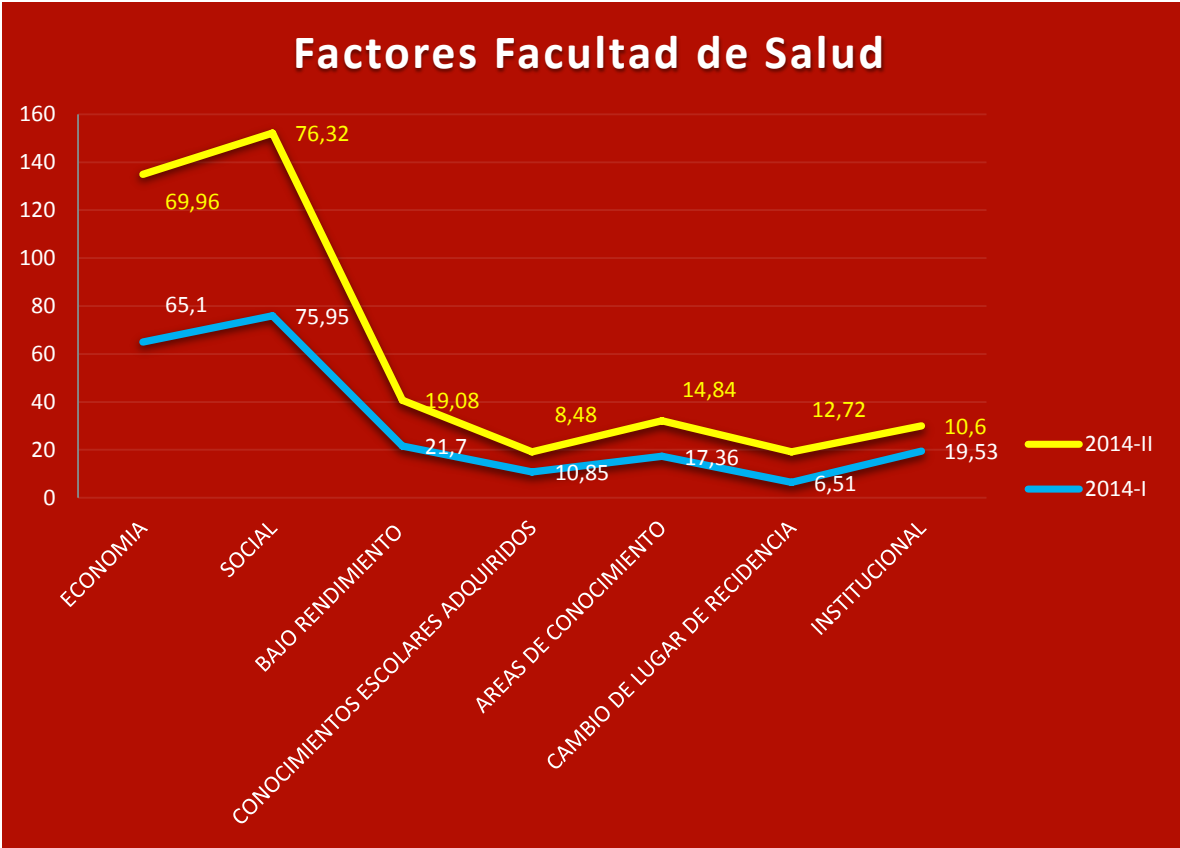


Grafico 16: Factores de Deserción en la facultad de Salud UPA desde el I Sem. 2014 y II Sem. 2014 Fuente: SPADIES UNIVERSIDAD DE PAMPLONA 2014.

En la facultad de Salud el motivo de mayor frecuencia está representado en aspectos de tipo Económico con el 30% del total de los registros. Siguiendo el orden de importancia el factor social tiene un peso del 35%, seguido encontramos razones como Bajo rendimiento Académico con el 10%, Áreas de Conocimiento con un 5%, Conocimientos Escolares Adquiridos con un 8%, y por último el factor Institucional y Cambio de lugar de residencia con 3% y 9% respectivamente.

3.4 Construcción del Perfil de las Facultades

Entre los modelos de RI el modelo que más se adaptó a nuestras necesidades fue el modelo vectorial, el cual nos permite conocer cuáles son los factores más influyentes en los estudiantes de la Universidad de Pamplona. La implementación de este modelo orientado al modelamiento de perfiles de describe a continuación:

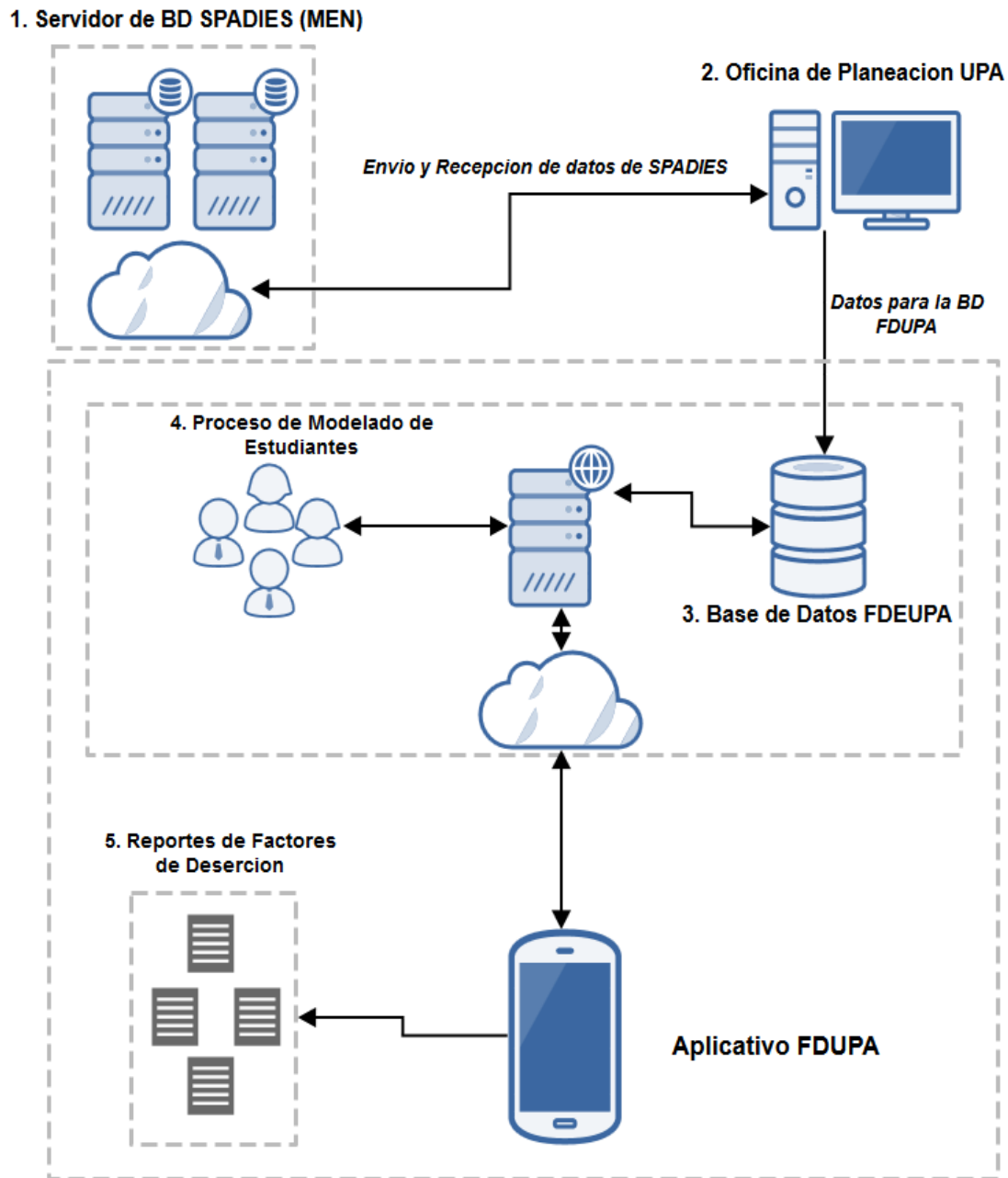


Figura 25: Construcción del perfil de la Facultad Fuente: Propia.

En la figura 25 se describen los pasos que se realizaron para la construcción del perfil de las facultades de la Universidad de Pamplona.

3.4.1 Sistema para la Prevención de la Deserción en las Instituciones de Educación Superior (SPADIES)

El Sistema para la Prevención de la Deserción en las Instituciones de Educación Superior consolida y ordena información que permite hacer seguimiento a las condiciones académicas y socioeconómicas de los estudiantes que han ingresado a la educación superior en el país. De esta manera, permite conocer el estado y evolución de la caracterización y del rendimiento académico de los estudiantes, lo cual es útil para establecer los factores determinantes de la deserción, para estimar el riesgo de deserción de cada estudiante y para diseñar y mejorar las acciones de apoyo a los estudiantes orientadas a fomentar su permanencia y graduación. [35]

EL SPADIES centraliza información proveniente de diferentes fuentes del sector que han resultado tener incidencia en la probabilidad de que un estudiante permanezca o no dentro del trayecto académico.

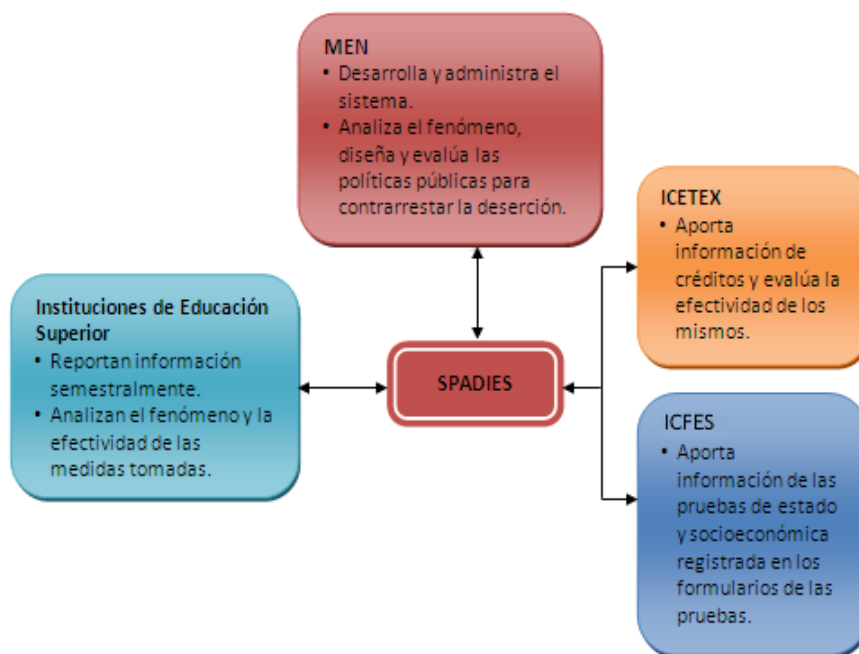


Figura 26: Sistema para la Prevención de la Deserción en las Instituciones de Educación Superior -SPADIES Fuente: [35]

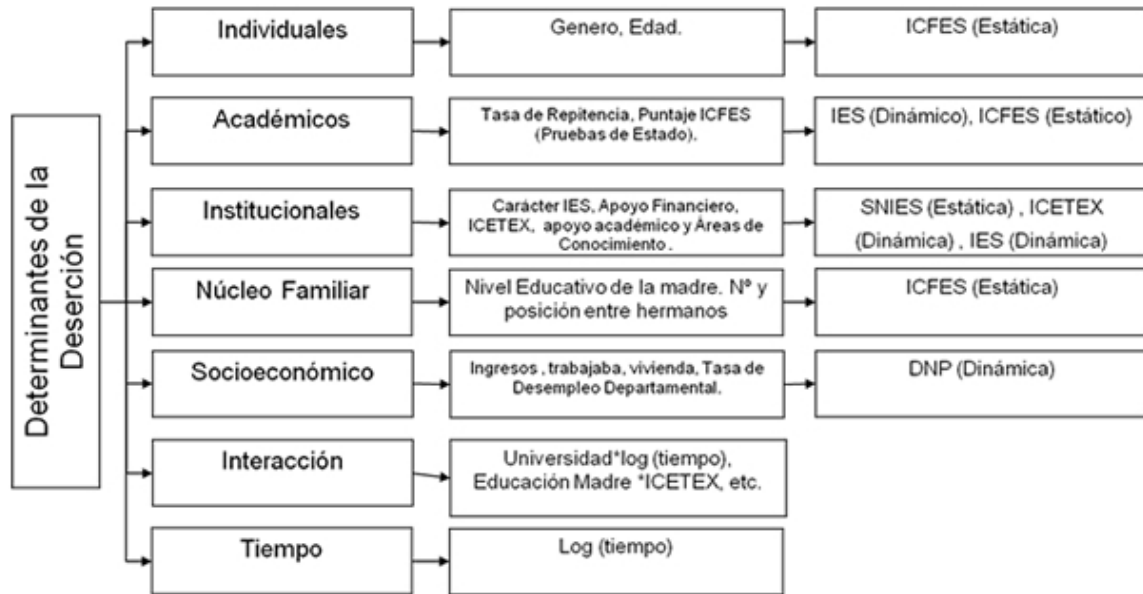


Figura 27: Variables que utiliza SPADIES Fuente: [35]

En la figura 27 se muestran las variables que hacen referencia a algunas condiciones que acompañan al individuo como son su edad, género, el contexto socioeconómico que lo acompaña, la composición de su núcleo familiar, las condiciones académicas al ingresar a la educación superior y el rendimiento que obtiene durante sus estudios, entre otras.

3.4.2 Base de Datos propuesta FDEUPA

La base de datos de SPAIDES que corresponde a la Universidad de Pamplona, es el asiento para la construcción del proceso de extracción de la información y modelamiento del perfil del estudiante, en base a ésta, se diseñó una nueva base de datos la cual será alimentada con la información correspondiente a la deserción estudiantil registrada para los dos semestres del año 2014.

Esta base de datos contiene los factores de deserción asociados a los estudiantes de la Universidad de Pamplona, además de esto cuenta con información básica de acerca de los estudiantes como lo son, nombre, código, facultad, edad entre otras, consta de 5 tablas (facultad, estudiante, factores de deserción, facultad-estudiante y estudiante-deserción) como se evidencia en la figura 28.

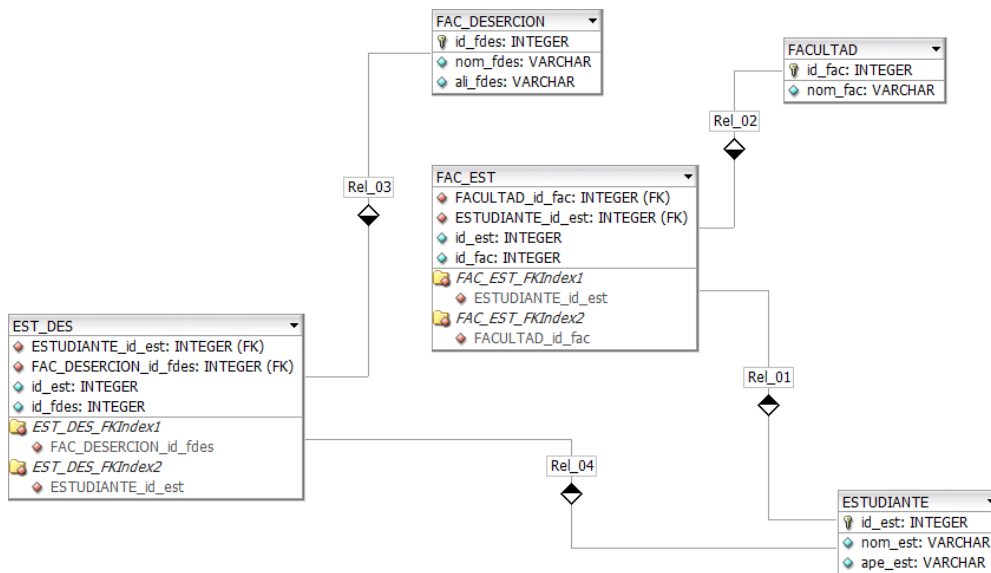


Figura 28: Base de Datos FDEUPA Fuente: propia

3.4.3 proceso de aplicación del Modelo Vectorial

El proceso para modelar el perfil del estudiante, se basa en el modelo vectorial utilizado en la recuperación de información, teniendo en cuenta la frecuencia en que los estudiantes son asociados a un factor de deserción.

3.4.3.1 Aplicación del M.V

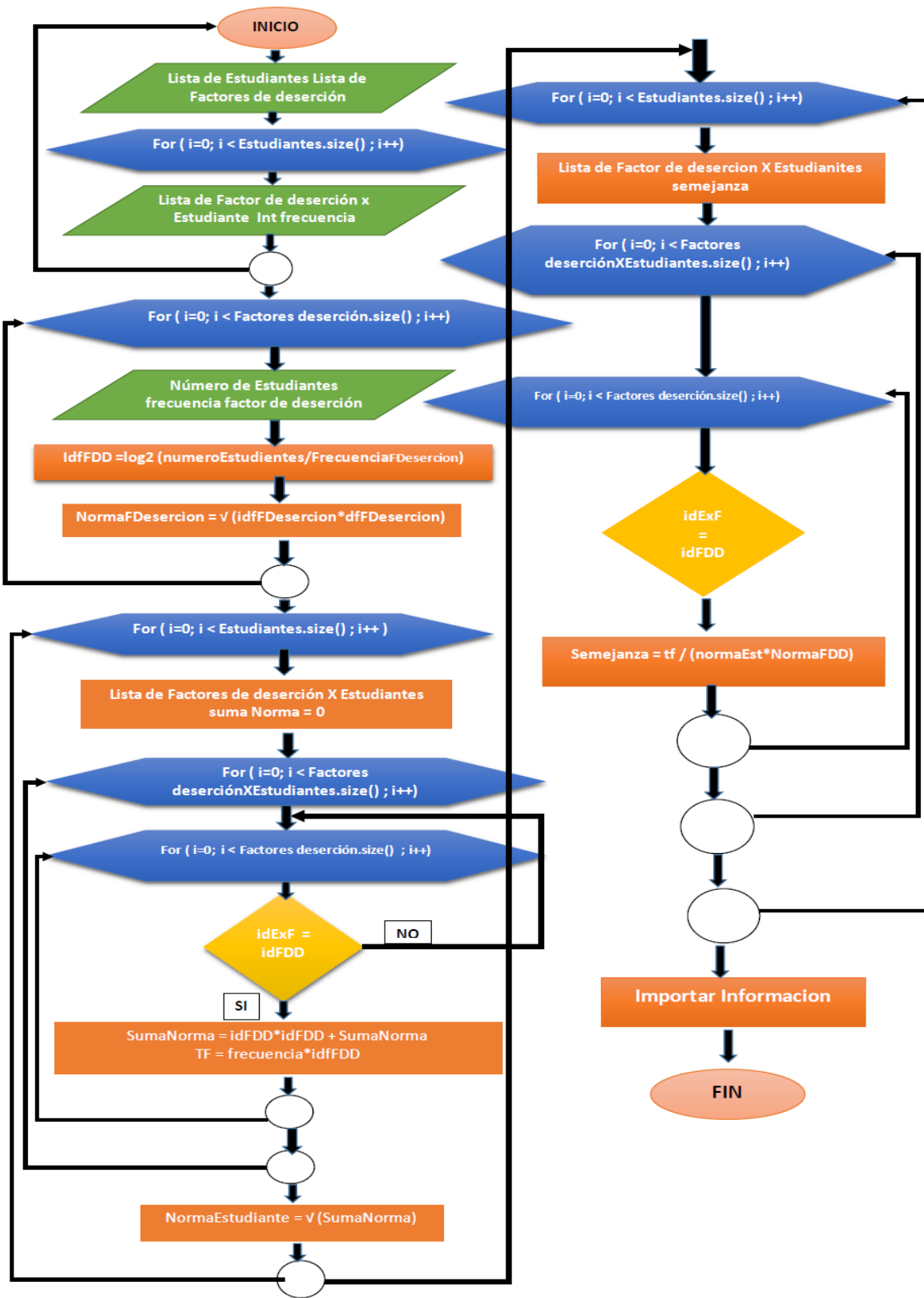


Figura 29: Diagrama de Flujo FDEUPA Fuente: propia

Creamos una lista de *estudiante* de los Estudiantes y Factores de deserción que se encuentran alojados en la base de datos FDUPA. Una vez creadas las listas, realizamos una consulta por cada paciente para conocer los medicamentos asociados a su diagnóstico y con qué frecuencia fueron suministrados. Lo anterior se denominara matriz de frecuencia.

	F_deserción 1	F_deserción ...	F_deserción n
Estudiante 1			
Estudiante ...			
Estudiante m			

Tabla 3 - Matriz de frecuencia Estudiante vs Factor de Deserción Fuente: Propia.

Luego, se obtiene la tabla de IDF que se obtiene mediante la fórmula:

$$TF * IDF = \log_2 \frac{N}{n_i} \quad (5)$$

Donde

- N = número de Estudiantes
- ni= Frecuencia del factor de deserción en los Estudiantes

Factores Deserción	Ni	N	IDF
Factor 1	Ni1	N1	IDF1
Factor ...	Ni...	N...	IDF...
Factor m	Nim	Nm	IDFm

Tabla 4 - Cálculo de IDF para los Factores de Deserción Fuente: Propia.

Finalmente, se calculan los pesos de los Factores de Deserción en cada Estudiante por TF*IDF, combinando las tablas anteriores.

Factor de Deserción
→

	F_deserción 1	F_deserción ...	F_deserción n	NORMA
Estudiante	Estudiante 1			
	Estudiante ...			
	Estudiante m			

TF * IDF
Tabla 5 - Cálculo de TF*IDF y Norma Fuente: Propia.

En la tabla 5 se agregó la columna norma, que corresponde a la longitud de los vectores y permite normalizar. Se calcula como la raíz cuadrada de la suma del cuadrado de cada uno de los pesos.

$$Norma(Est_1) = \sqrt{F_{des1}^2 + F_{des2}^2 + F_{des3}^2} \quad (6)$$

El Factor de Deserción a consultar también debe pasar por un procedimiento similar. Para hallar la norma de cada Factor de Deserción se tiene en cuenta la siguiente ecuación:

$$Norma(F_{des_m}) = \sqrt{IDF^2} \rightarrow Norma(Fdes_m) = IDF \quad (7)$$

El modo más simple de calcular la similitud entre un Factor de Deserción y un Estudiante, utilizando el modelo vectorial, es realizar el producto escalar de los vectores que los representan ecuación anterior.

$$sim(C, D_i) = \frac{\sum_{j=1}^t w_{qj} * w_{dij}}{\sqrt{\sum_{j=1}^t (w_{qj})^2 * \sum_{j=1}^t (w_{dij})^2}} \quad (8)$$

Dónde:

- $D_i = w_{di1}, w_{di2}, w_{di3}, \dots, w_{dit}$ vector que representa el Estudiante
- $C = w_{q1}, w_{q2}, w_{q3}, \dots, w_{qt}$ vector que representa Los Factores de Deserción

(Este vector solo contiene el peso de un solo Factor de Deserción, ya que se realiza una consulta entre Factor y Estudiante)
- t : Número de Factores en la colección
- w_{dij} : Peso del Factor j en el Estudiante i

La similitud resultante es un valor entre el rango 0 y 1.

3.4.4 Resultado Perfil Del Estudiante

Como resultado se obtienen a los Estudiantes modelados según la frecuencia de afectación de los factores de deserción asociados. Con esta información ya se puede deducir que Estudiantes tienen recurrencia de ciertos factores de deserción, y además de esto se modelarían los factores de deserción estudiantil que más afectan las facultades de la Universidad de Pamplona.

3.5 Diseño del Aplicativo

El propósito del presente apartado es documentar el proceso de programación de la aplicación móvil para detectar los Factores de Deserción estudiantil en la Universidad de Pamplona. En esta fase se implementa el prototipo, se describe el entorno de desarrollo y la arquitectura de los framework de trabajo, PhoneGap/Cordova, JQuery Mobile.

3.5.1 Entorno de Desarrollo

3.5.1.1 Entorno del Hardware

Para el desarrollo del aplicativo Móvil se han utilizado indistintamente dos ordenadores uno personal y el otro un servidor Prestado por la Empresa Inversiones Crear Rama. A continuación se detallan las principales características técnicas:

➤ **Portátil Lenovo B40-20392:**

- ✓ Procesador: Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz, 2401 Mhz
- ✓ Memoria física instalada (RAM): 8GB
- ✓ HDD: 500GB
- ✓ S.O: Microsoft Windows 7 Professional 64 Bits

➤ **HP Z230 Tower Workstation:**

- ✓ Procesador: Intel(R) Xeon(R) CPU E3-1231 v3 @ 3.40GHz, 3401 Mhz,
- ✓ Memoria física instalada (RAM): 8,00 GB
- ✓ HDD: 1 TB
- ✓ S.O: Ubuntu 15,10 64 Bits

Para la depuración y realización de pruebas se ha utilizado un dispositivo móvil y el emulador de Android Virtual Device Manager:

- **Lenovo A2010-I:** procesador MT6735M quad-core 1GHz, GPU Mali-T720, con 1GB de RAM, pantalla de 4,5", Sistema Operativo Android 5.1 (lollipop).

3.5.1.2 Configuración Entorno del Software

A continuación se explica la instalación del entorno de trabajo para desarrollar nuestra aplicación Android. Los principales componentes son, el SDK de Android, el plugin ADT para eclipse, así como el propio IDE Eclipse y la configuración del Framework Phonegap.

1. **Descarga e Instalación de Java JDK:** El equipo debe tener instalada una versión de Java Development Kit, aconsejable la última versión disponible que se puede descargar desde la web de Oracle <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>. Los pasos para la instalación se pueden consultar en la siguiente página web <http://www.luiskano.net/blog/2010/04/28/tutorial-instalar-java-jdk-en-windows-7/>. Para el desarrollo del aplicativo móvil Se ha utilizado la versión 8 update 60 (64 Bits).
2. **Descarga e instalación de Eclipse:** Se ha descargado la versión Eclipse Mars.2 (4.5.2), la versión de 64 bits de Eclipse IDE for Java Developers de la web de eclipse: <https://eclipse.org/downloads/> Una vez descargado, simplemente con descomprimir y ejecutar eclipse.exe se podrá comenzar la aplicación.
3. **Descargar e instalación del SDK de Android:** El SDK de la plataforma Android se puede descargar desde la web de desarrollo de Android <http://developer.android.com>, seleccionando "Use an existing IDE"- "Download the SDK Tools for Windows". Los pasos para la instalación se pueden consultar en la siguiente página web <http://jarroba.com/instalar-el-sdk-de-android-y-vincularlo-con-eclipse/>.
4. **Descargar el plugin Android para Eclipse:** Google pone a disposición de los desarrolladores un plugin para Eclipse llamado Android Development Tools (ADT) que facilita el desarrollo de aplicaciones para la plataforma. Se puede descargar desde las opciones de actualización de Eclipse "Help / Install new software..." indicando la siguiente URL de descarga: <https://dl-ssl.google.com/android/eclipse/> Se deben seleccionar los dos paquetes disponibles "Developer Tools" y "NDK Plugins".

5. **Configurar el plugin ADT:** Una vez instalado el plugin, se deberá configurar indicando la ruta donde ha sido instalado el SDK de Android. Desde la ventana de configuración de Eclipse (Window / Preferences...), en la sección de Android se indica la ruta.

6. **Instalar las Platform Tools y los Platforms necesarios:** Aparte del SDK de Android instalado previamente que contiene las herramientas básicas para desarrollar en Android, también se deben descargar las Platform Tools que contiene herramientas específicas de la última versión de la plataforma, y una o varias plataformas (SDK Platforms) de Android, que no son más que las bibliotecas necesarias para desarrollar sobre cada una de las versiones concretas de Android.

Se aconseja instalar (en Eclipse desde “Window/ Android SDK Manager”) al menos dos versiones, la mínima versión sobre la que se desea soporte para nuestra aplicación y la máxima versión disponible. En nuestro caso se han selecciona la versiones “Android 6.0 (API 23)” y la versión “Android 2.2 (API 8)” como mínima versión. También se ha seleccionado “Android Support Library”, que es una biblioteca que nos permitirá utilizar en versiones antiguas de Android.

7. **Configurar un AVD:** Para realizar pruebas y depurar la aplicación desarrollada no es necesario disponer de un dispositivo físico. Es posible configurar un emulador o dispositivo virtual (Android Virtual Device o AVD) que nos proporciona el SDK de Android. Podremos encontrarlo en Eclipse en “Window -- AVD Manager”.

Es posible indicar la versión de Android, resolución de pantalla, memoria disponible además de otras características hardwares como cámara, gps por ejemplo. A pesar de la disponibilidad de un emulador de Android, debido a la limitación hardware disponible y al consumo de recursos (muchas veces no disponibles) que genera la ejecución del dispositivo virtual de Android se ha

optado para la realización de las pruebas y la depuración del código en un dispositivo físico conectado al PC. Para ello es necesario activar la depuración USB en el móvil e instalar los drivers ADB Interface en Windows. De este modo, al ejecutar la aplicación desde Eclipse, a parte de los AVD configurados, es posible seleccionar el dispositivo móvil conectado al ordenador.

8. Descarga plugin de PhoneGap en Eclipse: La instalación de PhoneGap en Eclipse nos va a otorgar un entorno para desarrollar aplicaciones para dispositivos móviles y con aplicaciones gratuitas. Se puede descargar desde las opciones de actualización de Eclipse, selecciona en “Help”, “Install New Software” y colocas en “location” la siguiente ruta: <http://svn.codespot.com/a/eclipselabs.org/mobile-web-development-with-phonegap/tags/r1.2/download>, Se deben seleccionar los paquetes disponibles.

9. Configuración Plugin PhoneGap: Ya teniendo instalado **PhoneGap en Eclipse** podemos ver, el nuevo icono para crear un nuevo proyecto de PhoneGap (Figura 30).

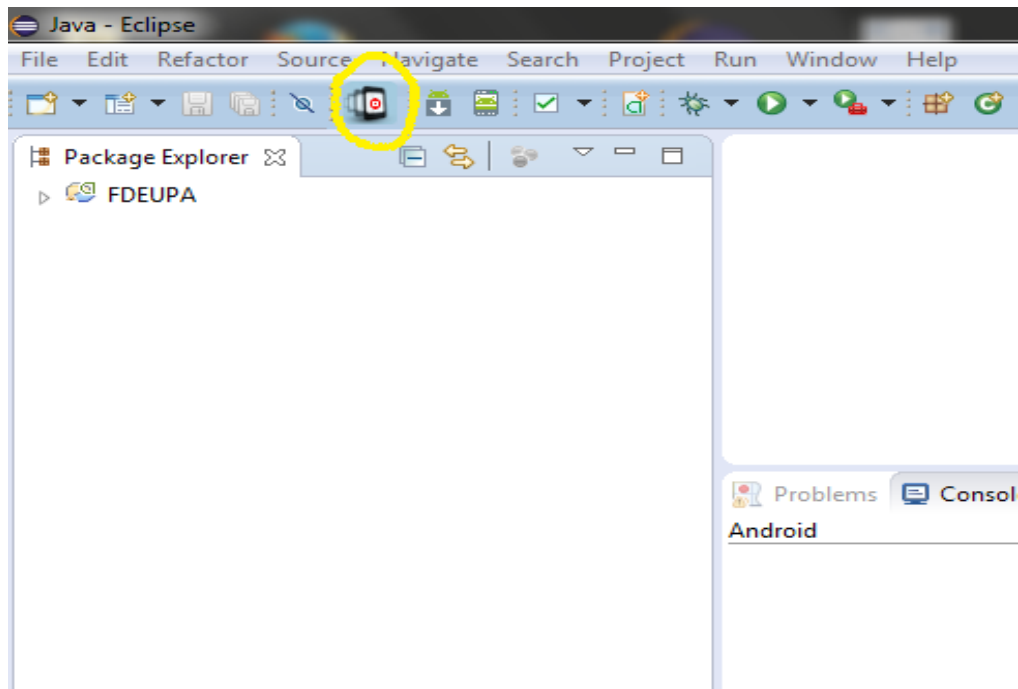


Figura 30: PhoneGap Instalado en Eclipse Fuente: propia

3.5.1.3 Arquitectura

Se introduce rápidamente la arquitectura y los módulos que forman parte de Android y PhoneGap.

- **Elementos Básicos:** La arquitectura Android tiene diversos elementos, que se describen brevemente a continuación. Esta arquitectura es un proceso transparente para el desarrollador de PhoneGap.
 - ✓ **Activity:** Para los desarrollos en Android, el núcleo principal es la clase activity (android.app.Activity). Sin embargo para el desarrollador de PhoneGap, la clase activity es la que instancia los métodos de mismo y extiende las funcionalidades de DroidGap. Además, define la ruta que contiene el documento .html que inicia la aplicación.
 - ✓ **Intents:** Durante el transcurso de una aplicación Android, ésta puede cambiar entre diferentes instancias Activity. A veces pueden existir varias instancias de una actividad en la pila. Los intents permiten lanzar nuevas instancias a través de los métodos habilitado, como: startActivity(), startActivityForResult(), etc. Este proceso es transparente para el desarrollador de Phonegap, encargándose la API de realizar las instancias que sean necesarias.
 - ✓ **Bundle:** El objeto Bundle mantiene información del estado de las actividades instanciadas. Al igual que el punto anterior, PhoneGap se encarga de gestionar los estados siendo transparente para el programador.
 - ✓ **Grupos De Vistas Layout:** Android define las pantallas de la aplicación mediante etiquetas definidas en archivos XML. Phonegap define las vistas mediante HTML5 y CSS.

Servicios: Android interactúa con el sistema operativo y con el hardware interno a través de una serie de gestores, que son responsables de mantener el estado de cada servicio interno del sistema. PhoneGap, a través de su API gestiona los siguientes aspectos del dispositivo móvil, de forma transparente al desarrollador:

- ✓ Accelerometer : Gestiona el sensor de movimiento del dispositivo.
- ✓ Camera : Captura fotos usando la cámara del dispositivo.
- ✓ Capture: Obtiene archivos multimedia usando las aplicaciones de captura.
- ✓ Compass: Obtiene la dirección a la que mira el dispositivo.
- ✓ Connection : Comprueba el estado de la red e información de la red móvil.
- ✓ Contacts: Gestiona la agenda del dispositivo.
- ✓ Device: Obtiene información específica del dispositivo.
- ✓ Events: Eventos nativos a través de JavaScript.
- ✓ File: Sistema de ficheros nativos a través de JavaScript.
- ✓ Geolocation: Permite conocer dónde se encuentra el dispositivo
- ✓ Media: Graba y reproduce ficheros de audio.
- ✓ Notification: Notificaciones visuales, audibles y táctiles.
- ✓ Storage: Acceda a las posibilidades de almacenamiento nativas del dispositivo.

En caso de necesitar alguna funcionalidad que no se implemente en la API de PhoneGap, se puede acceder al gestor de Android mediante la programación de un Plugin. En este caso es necesario desarrollarlo además para el resto de plataformas, si pretendemos mantener la portabilidad de nuestra aplicación entre las diferentes plataformas móviles.

Manifest: Los archivos manifest, en formato XML, deben acompañar a la aplicación, ya que contienen información sobre la misma. En ellos se definen diversos parámetros tales como el nombre de la aplicación, versión, componentes, permisos, etc.

3.5.1.4 Contenido

El desarrollo consta de los siguientes componentes, cuya distribución se ilustra mediante la figura 31.

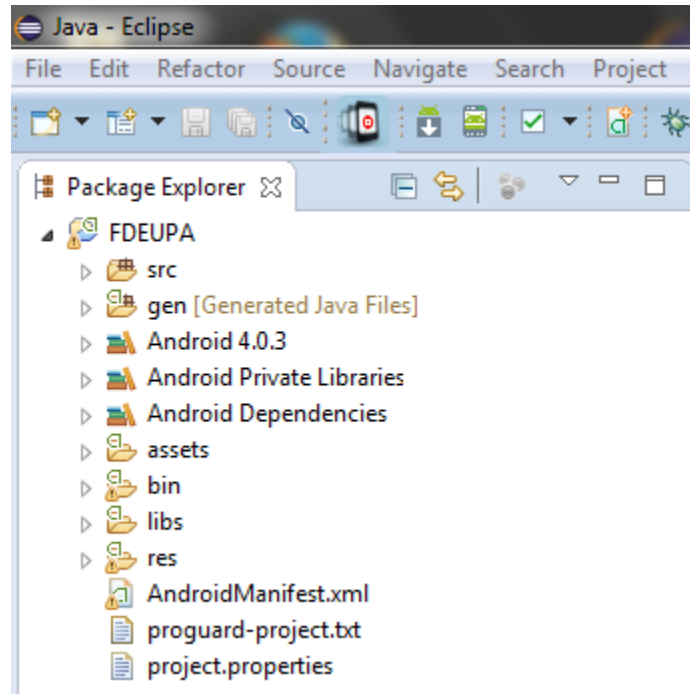


Figura 31: Directorio del Aplicativo Móvil FDEUPA Fuente: propia

- 'assets':** Recursos HTML, CSS y JavaScript. Carpeta de desarrollo.
- 'bin':** Objetos compilados y binarios para la instalación de la aplicación.
- 'gen':** Archivos R.java.
- 'libs':** Librerías Android.jar y Cordova.jar.
- 'res':** Recursos de la aplicación, layouts y values.
- 'src':** Contiene MainActivity.java.

3.5.1.5 Arquitectura de la Aplicación

La aplicación se basa en una arquitectura de tres capas. La ventaja principal de éste estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga

algún cambio, sólo se modifica al nivel requerido sin tener que revisar entre código mezclado.

- **Capa de presentación:** es la que ve el usuario. Presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso. Es la interfaz gráfica y debe tener la característica de ser entendible y fácil de usar para el usuario. Esta capa se comunica únicamente con la capa de negocio.
- **Capa de negocio:** es donde residen los procesos que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Esta capa se comunica con la capa de presentación para recibir las solicitudes y presentar los resultados; y con la capa de datos para solicitar al gestor de base de datos almacenar o recuperar datos de él.
- **Capa de datos:** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por un gestor de bases de datos interno que realiza todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

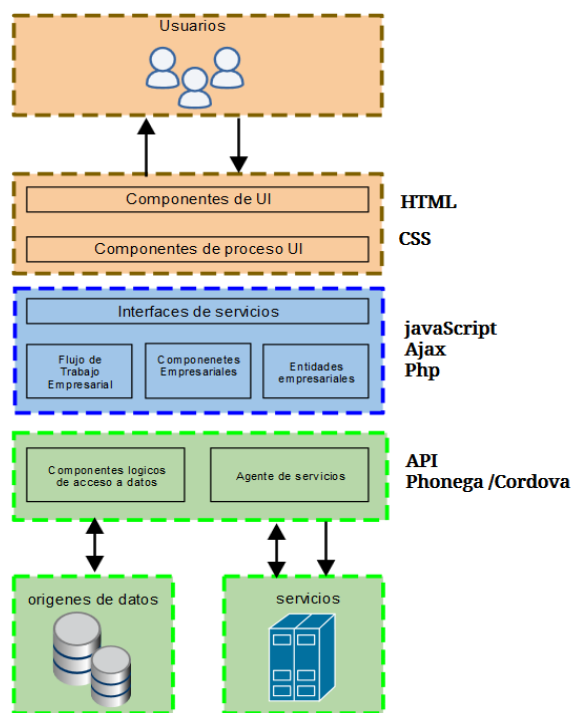


Figura 32: Capas del Aplicativo Móvil FDEUPA Fuente: propia

3.5.1.6 Módulos De Aplicación

Se describen brevemente los diferentes módulos en los que se compone la aplicación en PhoneGap/Cordova:

- **Index.html:** Contiene las funcionalidades de la aplicación en un solo documento .html. Se emplea la tecnología de *jQuery Mobile* para ocultar o mostrar las diferentes pantallas. Esto hace que tengamos todo cargado desde un primer momento en memoria, haciendo que la transición por las diferentes páginas sea más rápida. Se compone de 12 páginas que se van mostrando u ocultando según la lógica de programación.

```
16     </head>
17     <body>
18     <!-- /Inicio -->
19     <div id="inicio" data-role="page"> ... </div><!-- /Fin Inicio -->
62     <!-- /Fin Inicio -->
63     <!-- /Calculos -->
64     <div id="calculo" data-role="page"> ... </div><!-- /Fin Calculos -->
90     <!-- /Fin Calculos -->
91
92     <!-- / Base de Datos-->
93     <div data-role="page" id="bd"> ... </div>
118    <!-- / Fin Base de Datos -->
119
120    <!-- / Listado Clientes-->
121    <div data-role="page" id="lcli" > ... </div>
141    <!-- / Fin Listado Clientes -->
142
143    <!-- / Listado Productos-->
144    <div data-role="page" id="lpro" > ... </div>
162    <!-- / Fin Listado Productos -->
163
164    <!-- / Listado Compras-->
165    <div data-role="page" id="lcom" > ... </div>
184    <!-- / Fin Listado Compras -->
185
186    <!-- / Listado IDF-->
187    <div data-role="page" id="lidf" > ... </div>
206    <!-- / Fin Listado IDF -->
207
208    <!-- / Listado TFIDF-->
209    <div data-role="page" id="ltfidf" > ... </div>
228    <!-- / Fin Listado Norma Clientes -->
229
230    <!-- / Listado Norma Clientes-->
231    <div data-role="page" id="lnorcli" > ... </div>
249    <!-- / Fin Listado Norma Clientes -->
250
251    <!-- / Grafico-->
252    <div data-role="page" id="rgra" > ... </div>
312
313    <!-- / Fin Grafico -->
314    <!-- / info -->
315    <div data-role="page" id="info"> ... </div>
316    <!-- / Fin info --> |
373    </body>
374    </html>
```

Figura 33: Pagina de presentación de Datos del Aplicativo Móvil FDEUPA Fuente: propia

- **Funciones.Js:** Contiene la capa de negocio en un solo documento .js. Se emplean las tecnologías de JavaScript y Ajax para la actualización dinámica de las páginas de la capa de visualización.

- **Componentes Javascript Y Css:** en el aplicativo móvil FDEUPA (Factores de Deserción Estudiantil en la Universidad de Pamplona), se utilizan las siguientes bibliotecas:
 - ✓ **Cordova-1.9.0.js:** Es la biblioteca que emplea la capa de negocios para comunicarse con el dispositivo móvil.

 - ✓ **Jquery-2.2.3.min.js:** Es una biblioteca que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

 - ✓ **Jquery.mobile-1.1.0.min.js:** Es una biblioteca basada en el desarrollo de jQuery, que mejora la interacción de jQuery en dispositivos con pantalla táctil.

 - ✓ **Jquery.mobile-1.1.0.min.css:** Hoja de estilos de la biblioteca jQuery Mobile.

3.5.1.7 Interfaces Gráficas

A continuación se mostraran las interfaces gráficas diseñadas, prestando especial atención a la correspondencia entre los reportes de la base de datos, los cálculos e interfaz principal.

- **Pantalla de Inicio:** como se puede evidenciar en la figura 34 la pantalla principal consta de una barra de navegación, que posee las opciones de información, base

de datos, cálculos y el panel para modelar los factores de deserción estudiantil en las facultades de la Universidad de Pamplona.



Figura 34: Página de inicio del Aplicativo Móvil FDEUPA Fuente: propia

➤ **Reporte de BD:** a través de esta interfaz el usuario podrá realizar los reportes de la información contenida en la BD tal como se muestra en la figura 35.



Figura 35: reportes BD del Aplicativo Móvil FDEUPA Fuente: propia

- **Realizar Cálculos:** a través de esta interfaz el usuario podrá realizar los cálculos asociados al proceso de modelamiento de perfiles tal como se evidencia en la figura 36.



Figura 36: Reportes Cálculos del Aplicativo Móvil FDEUPA Fuente: propia

CAPITULO 3

4. PRUEBAS Y RESULTADOS

Para Evidenciar la efectividad del modelo vectorial, y validar el funcionamiento del aplicativo móvil modelando los factores de deserción que afectan las facultades de la Universidad de Pamplona se necesitó integrar varios lenguajes de programación y nuevas tecnologías:

- **Base de Datos:** la BD se creó a partir de datos suministrados por las oficinas de planeación, Registro y Control Académico y Bienestar universitario, la cual está compuesta por 3945 estudiantes, 7 factores de deserción, y por ultimo 7 facultades. La base de datos se realizó en MySQL y se encuentra alojada en uno de los servidores de la Empresa Inversiones Crear Rama.
- **Entorno de Desarrollo:** el aplicativo móvil fue desarrollado para plataforma Android en el Ambiente de Desarrollo Integrado Eclipse IDE for Java Developers Version: Mars.2 Release (4.5.2), integrando a su vez lenguajes de programación como PHP, JavaScript, Java, HTML, adicional a esto frameworks como PhoneGap, JQuery Mobile, y herramientas para desarrollo WEB como los son Ajax, JSON, CSS3.

4.1 Perfil de la Facultad

Para realizar el modelado del perfil de las facultades con respecto a los factores de deserción el aplicativo FDEUPA se conecta a una BD Remota y luego aplica el modelo vectorial para presentar los factores más influyentes para las facultades.

4.1.1 Factor Bajo Rendimiento Académico

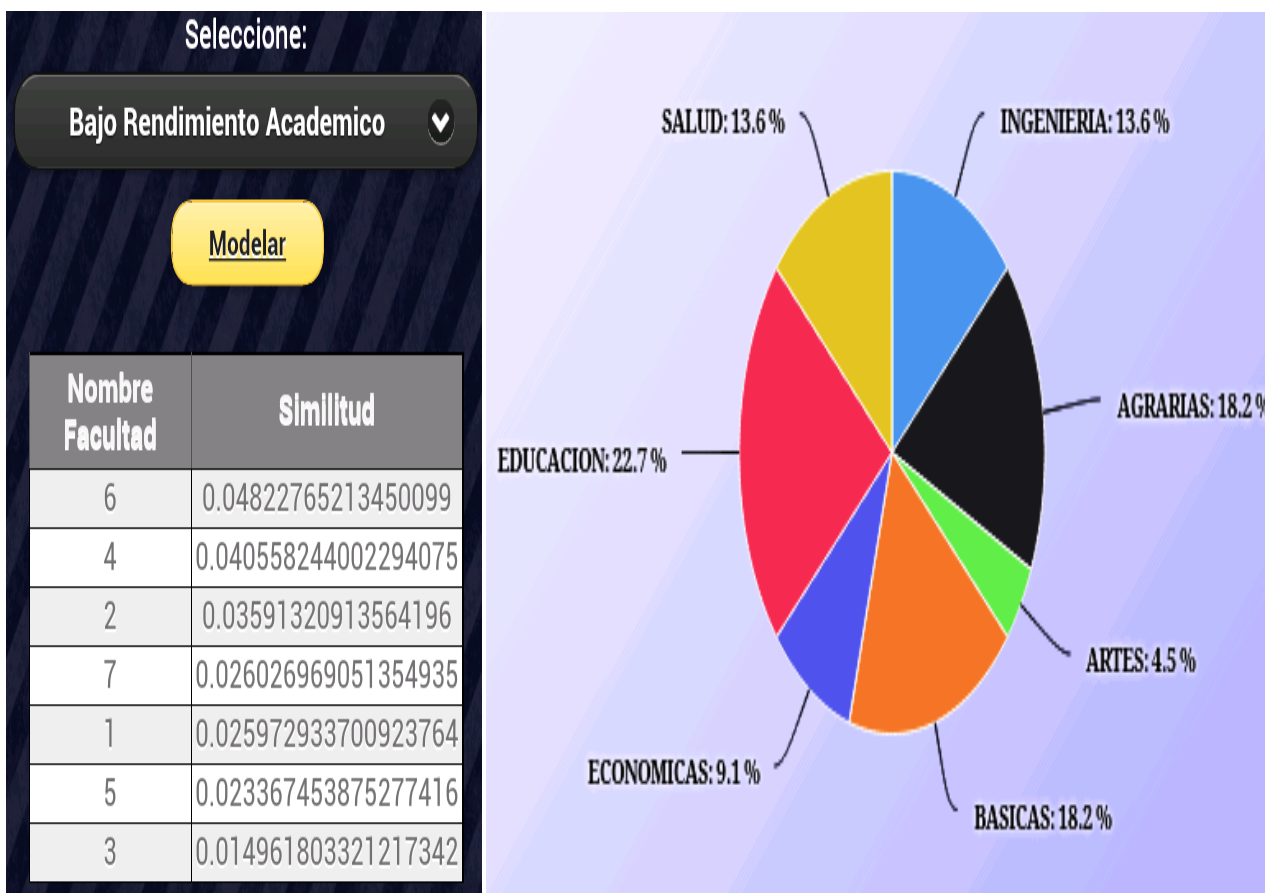


Figura 37: Modelado Factor Bajo Rendimiento Académico del Aplicativo Móvil FDEUPA Fuente: propia

En la figura 37 podemos ver los resultados del Factor de deserción Bajo Rendimiento Académico el cual está asociado a las 7 facultades. En la columna similitud podemos observar el resultado de aplicar el modelo vectorial para este Factor de deserción.

El grafico de la figura 37 nos indica que la facultad a la cual más afecta el factor de deserción Bajo Rendimiento Académico es la de Educación con un 22,7% seguida de Agrarias y Básicas con un 18,2%, Ingenierías y Salud con un peso de 13,6% y por ultimo Económicas y Artes con 9,1% y 4,5% respectivamente.

4.1.2 Factor Social

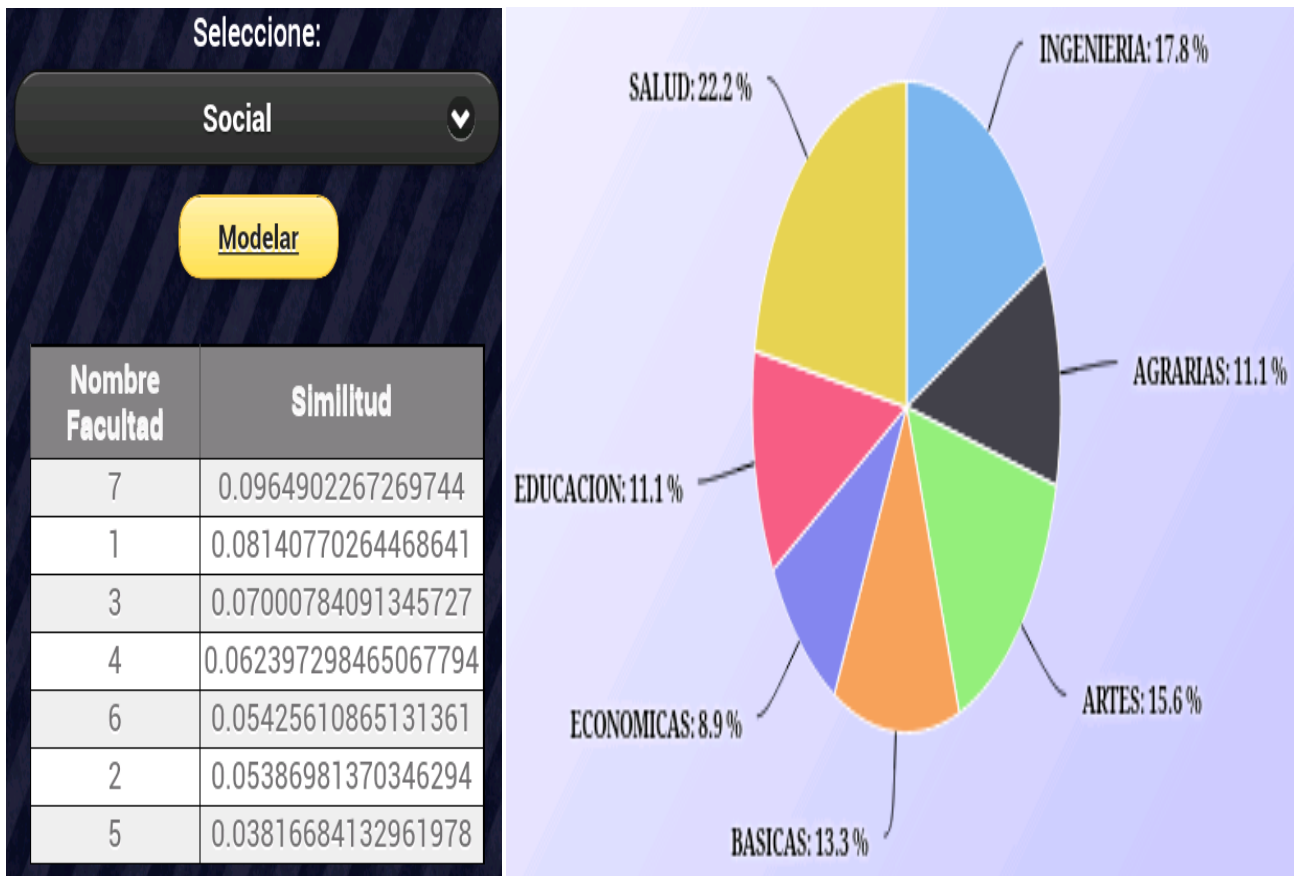


Figura 38: Modelado Factor Social del Aplicativo Móvil FDEUPA Fuente: propia

En la figura 38 podemos ver los resultados del Factor Social el cual está asociado a las 7 facultades. En la columna similitud podemos observar el resultado de aplicar el modelo vectorial para este Factor de deserción.

El grafico de la figura 38 nos indica que la facultad a la cual más afecta el factor Social es la de Salud con un 22,2% seguida de Ingenierías y Artes con un 17,8% y 15,6%, Básicas con un peso de 13,3% y por ultimo Agrarias, Educación y Económicas con 11,1% y 8,9% respectivamente.

4.1.3 Factor Económico

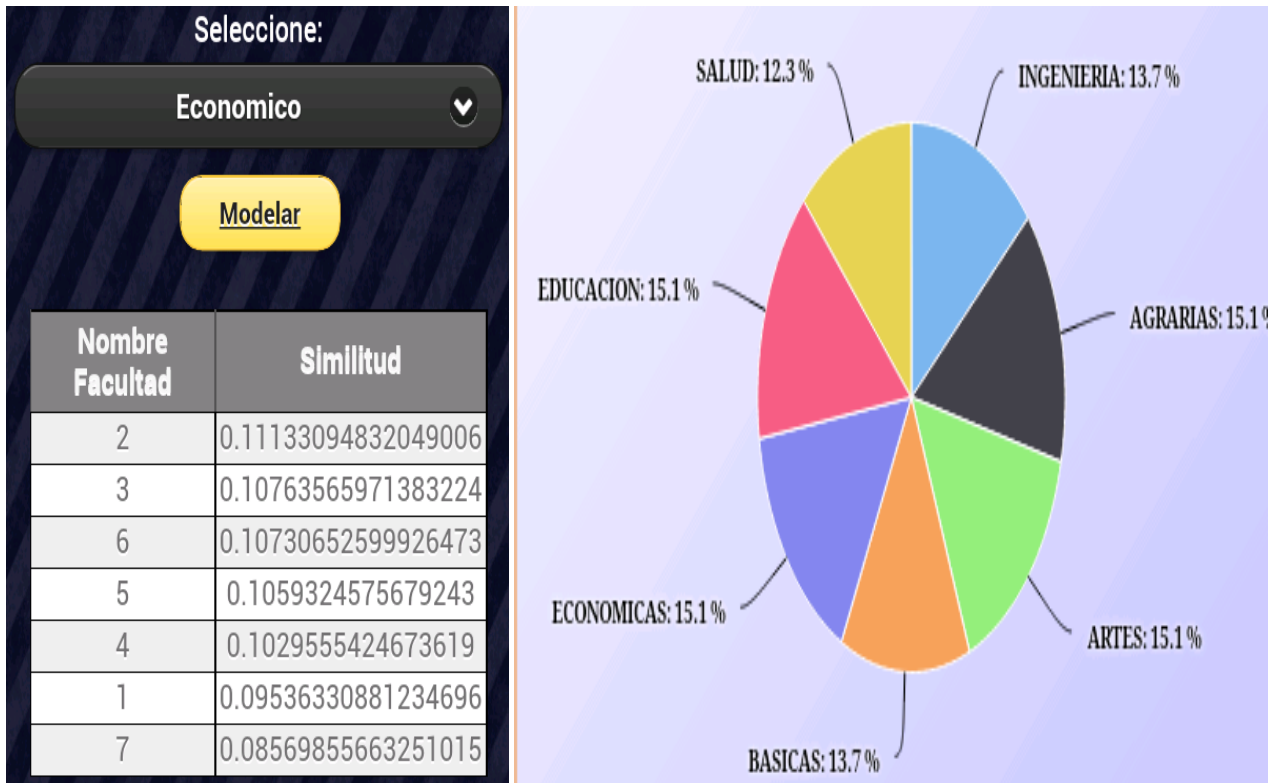


Figura 39: Modelado Factor Económico del Aplicativo Móvil FDEUPA Fuente: propia

En la figura 39 podemos ver los resultados del Factor Económico el cual está asociado a las 7 facultades. En la columna similitud podemos observar el resultado de aplicar el modelo vectorial para este Factor de deserción.

El grafico de la figura 39 que el factor social es uno de los factores más persistentes en las facultades de la Universidad de Pamplona debido a que es el factor por el cual la gran mayoría de estudiantes deserta, los porcentajes de incidencia sobre las facultades de educación, agrarias, económicas y artes tiene un peso de 15,1% mientras que para básicas e ingeniería es 13,7% y por último la facultad de salud con un 12,3% de afectación.

4.1.4 Factor Conocimientos Escolares Adquiridos

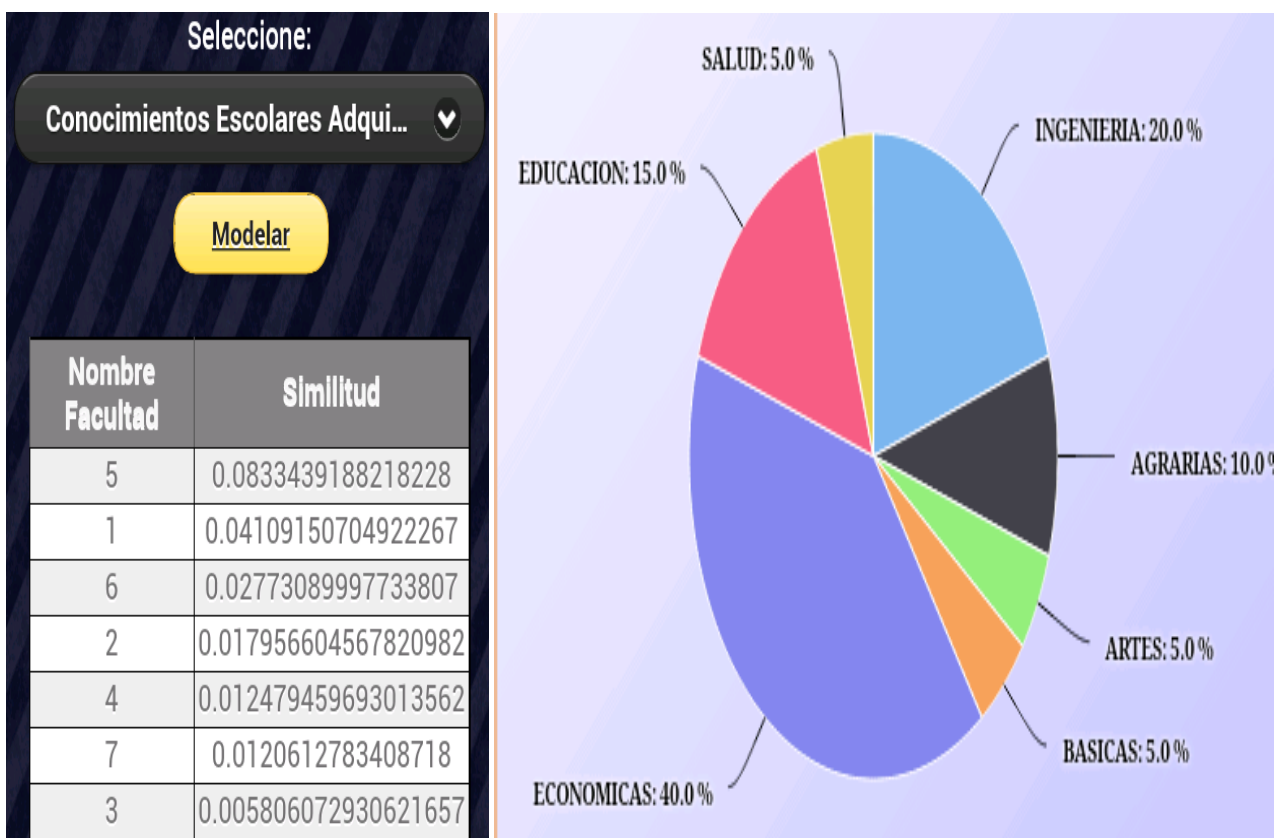


Figura 40: Modelado Factor Conocimientos escolares Adquiridos del Aplicativo Móvil FDEUPA Fuente: propia

En la figura 40 podemos ver los resultados del Factor Conocimientos Escolares Adquiridos el cual está asociado a las 7 facultades. En la columna similitud podemos observar el resultado de aplicar el modelo vectorial para este Factor de deserción.

El grafico de la figura 40 nos muestra que las facultades de economía ingeniería y educación son las más afectadas por este factor con un porcentaje de 40, 20 y 15%. Seguida de Agrarias con un 10%, y por ultimo con un porcentaje de afectación del 5% las facultades de salud, artes y básicas.

4.1.5 Factor Áreas De Conocimiento

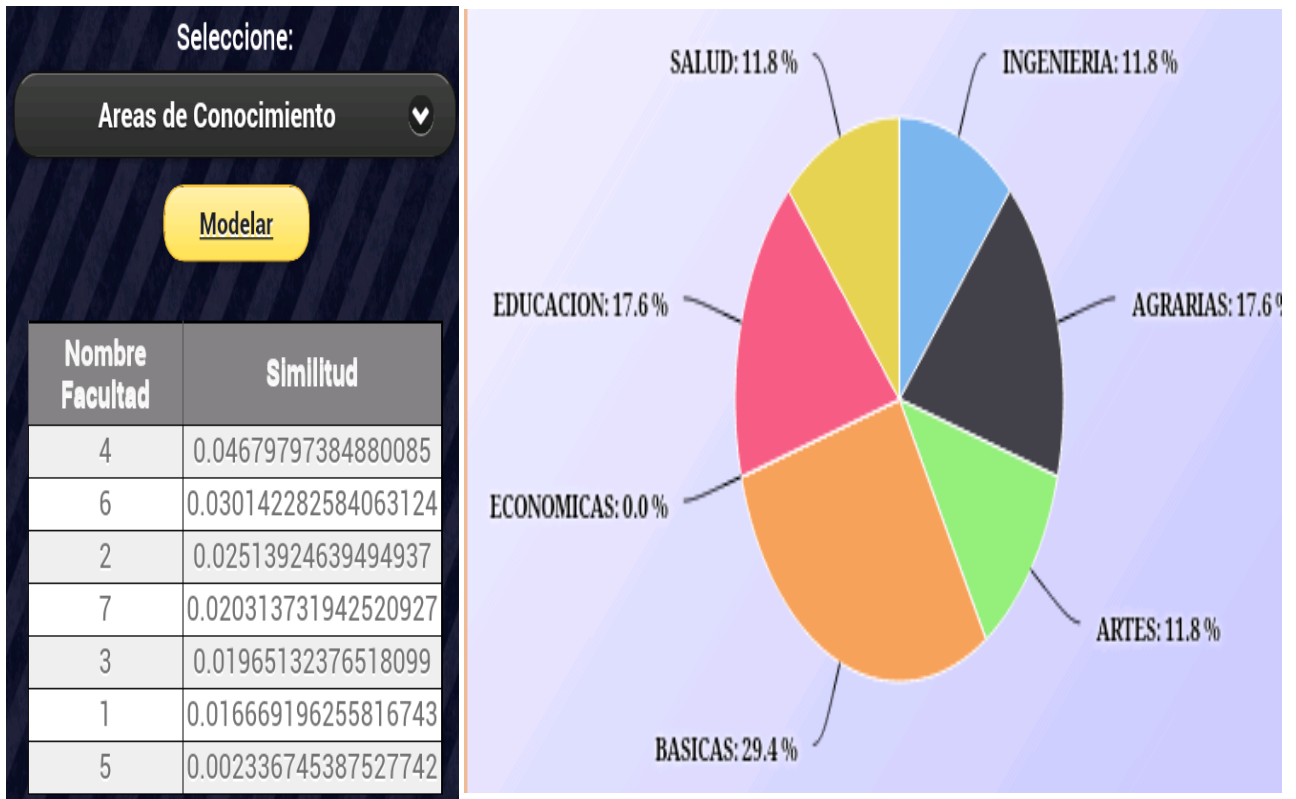


Figura 41: Modelado Factor Áreas de Conocimiento del Aplicativo Móvil FDEUPA Fuente: propia

En la figura 41 podemos ver los resultados del Factor Áreas de Conocimiento el cual está asociado a las 6 facultades. En la columna similitud podemos observar el resultado de aplicar el modelo vectorial para este Factor de deserción.

El grafico de la figura 41 nos muestra que la facultad de Ciencias básicas es la más afectada por este factor con un 29,4%. Seguida de las facultades de Agrarias y Educación con un peso de 17,6%, y por ultimo las Facultades de Artes, Ingeniería y salud con un 11,8% respectivamente.

4.1.6 Cambio De Lugar De Residencia

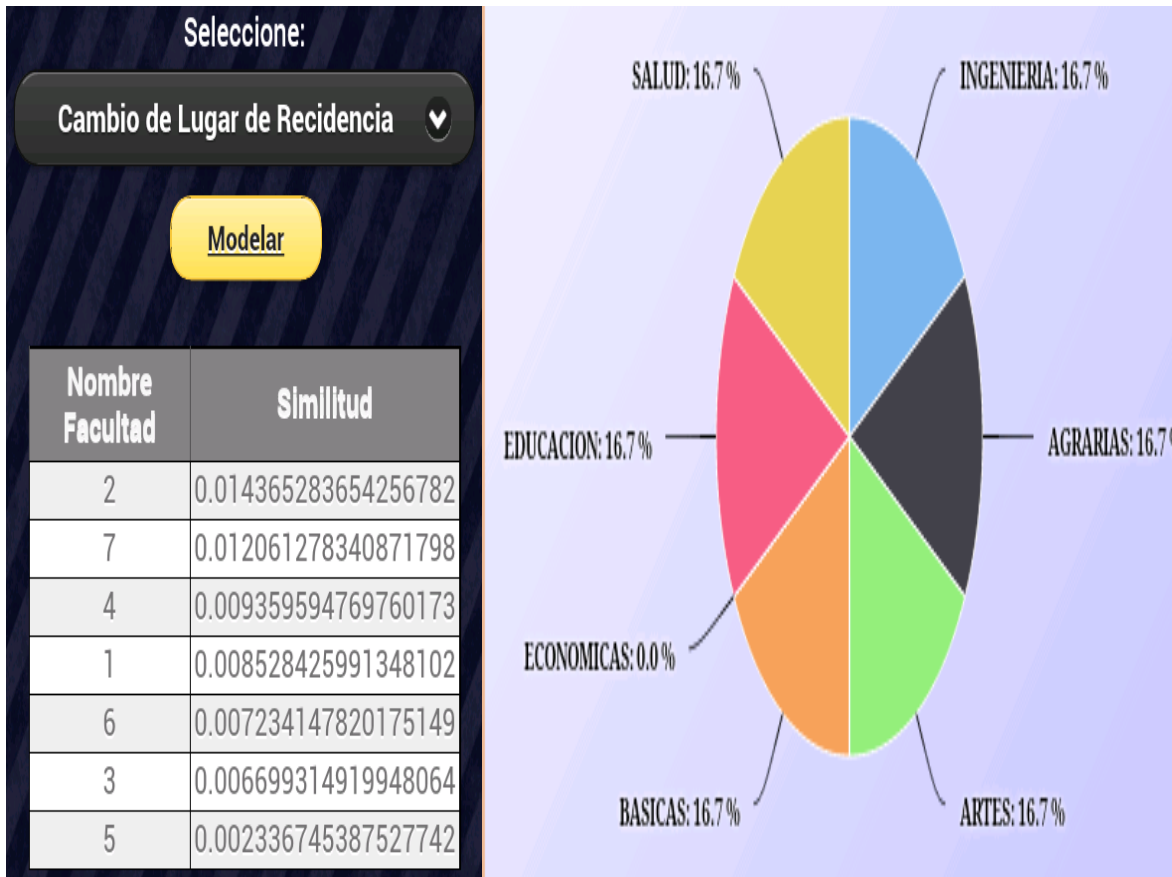


Figura 42: Modelado Factor Cambio de Lugar de Residencia del Aplicativo Móvil FDEUPA Fuente: propia

En la figura 42 podemos ver los resultados del Factor de Cambio de Lugar de Residencia el cual está asociado a las 6 facultades. En la columna similitud podemos observar el resultado de aplicar el modelo vectorial para este Factor de deserción.

El grafico de la figura 42 nos indica que el factor Cambio de Lugar de Residencia es uno de los factores que afecta todas las facultades, con un mismo nivel de incidencia 16,7% esto debido a que son bajos los casos de deserciones por este factor.

4.1.7 Institucional

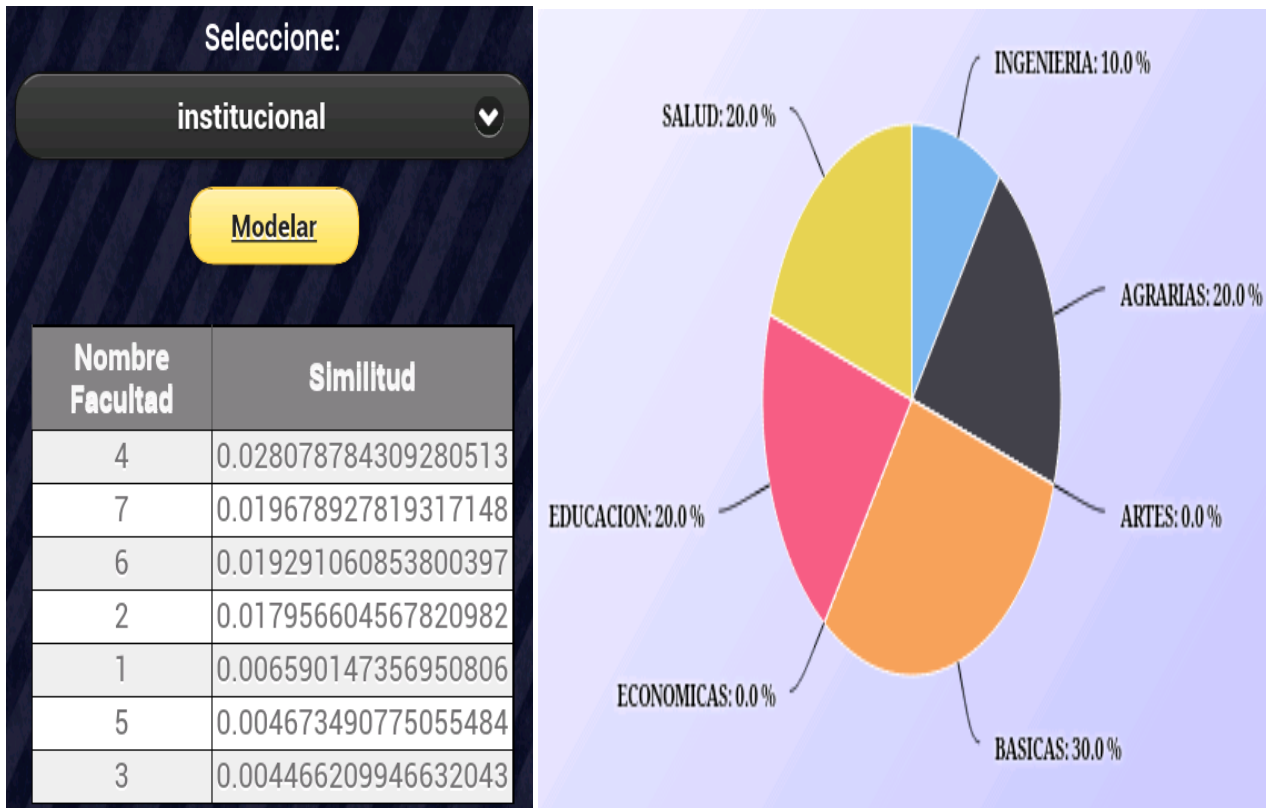


Figura 43: Modelado Factor Institucional del Aplicativo Móvil FDEUPA Fuente: propia

En la figura 43 podemos ver los resultados del Factor Institucional el cual está asociado solo a las 5 facultades. En la columna similitud podemos observar el resultado de aplicar el modelo vectorial para este Factor de deserción.

El grafico de la figura 43 nos muestra que la facultad de Ciencias básicas es la más afectada por este factor con un 30%. Seguida de las facultades de Agrarias, salud y Educación con un peso de 20%, y por ultimo las Facultades de Ingeniería con un 10%.

CONCLUSIONES

En el proyecto realizado se ha logrado desarrollar una aplicación para dispositivos móviles con sistema operativo Android, capaz de modelar el perfil del estudiante para luego indicar cuales son los factores de deserción estudiantil más influyentes en las distintas facultades de la Universidad de Pamplona. El modelamiento fue realizado gracias a la aplicación del modelo vectorial como forma para optimizar el manejo del perfil de un estudiante teniendo en cuenta los factores de deserción asociados a este, obteniendo así resultados satisfactorios al momento de ejecutar el aplicativo móvil FDEUPA.

La información contenida en la base de datos del aplicativo móvil procede de fuentes reales provenientes de las oficinas de Planeación, Bienestar Universitario y Registro y Control Académico de la Universidad de Pamplona, la cual hace referencia a algunas condiciones que acompañan al estudiante como son su edad, género, el contexto socioeconómico que lo acompaña, la composición de su núcleo familiar, las condiciones académicas al ingresar a la educación superior y el rendimiento que obtiene durante sus estudios, entre otras, dicha información se encuentra registrada en el Sistema para la Prevención de la Deserción en las Instituciones de Educación Superior (SPADIES), del cual se obtuvieron datos de los dos periodos académicos del Año 2014.

Una de las principales razones por la que se desarrolló la aplicación en Android, es su alto nivel de utilización en plataformas celulares y tabletas, además de esto se ha obtenido un gran conocimiento sobre desarrollo en dispositivos móviles basados en Android, así como su arquitectura, características, componentes y funcionamiento. Se realizó una completa investigación de desarrollo para dispositivos móviles, encontrándose que se pueden implementar aplicaciones utilizando diferentes tecnologías web como HTML5, CSS3 y JavaScript y con ayuda de Frameworks como JQuery Mobile, PhoneGap.

Finalmente, este proyecto ha demostrado que es viable modelar perfiles de usuarios a través de técnicas de recuperación de información, señalando principalmente el modelo vectorial.

RECOMENDACIONES

La aplicación desarrollada abre la posibilidad de Seguir con el proceso de investigación para el desarrollo para dispositivos móviles utilizando tecnologías web, ya que hoy en día la mayoría de empresas están apostando por HTML5, CSS3 y JavaScript para la realización multimedia o de animaciones, entre estos procesos se puede destacar el uso de frameworks como JQuery Mobile, y Phonegap,. Que facilitan notablemente el desarrollo de aplicaciones móviles, agregando una ventaja más que es la de contar con aplicaciones multiplataforma.

En lo que concierne a modelos de recuperación de información, se podría implementar una combinación entre el modelo vectorial y el probabilístico esto con el fin de agregar otros factores que son sumamente importantes al momento de modelar perfiles de usuario como por ejemplo el factor tiempo. Para con esto poder suministrar consultas más precisas.

BIBLIOGRAFIA

- [1]. Universidad Nacional – ICFES. Estudio de la Deserción estudiantil en la Educación Superior en Colombia. Estado del Arte sobre la retención estudiantil. 2014. Convenio 107/2002. Recuperado el [13 abril 2016].
- [2]. Jesús Vilares. Introducción a la Recuperación de Información. 2012. Disponible en: <http://www.grupolys.org/docencia/In/biblioteca/ir.pdf>. Recuperado el [12 Febrero 2016].
- [3]. Ministerio de Educación Nacional. Deserción estudiantil en la educación superior colombiana Metodología de seguimiento, diagnóstico y elementos para su prevención. 2009. Disponible en: <http://www.mineducacion.gov.co/sistemasdeinformacion/1735/articles-254702_libro_desercion.pdf>. Recuperado el [15 Febrero 2016].
- [4]. Caicedo C. “Colombia; Alto porcentaje de deserción universitaria” Estadística alcanza el 52% y es considerada como una “tragedia nacional”. El Diario La Prensa, junio 20 de 2005. p. 14.
- [5]. Ministerio de Educación Nacional. Portal de SPADIES del Ministerio de Educación Nacional de la República de Colombia: <http://www.mineducacion.gov.co/sistemasdeinformacion/1735/w3-article-254651.html> y del video de capacitación sobre el uso del SPADIES. 2014 Disponible en: <http://SPADIES.sikuani.net/Capacitaciones/SPADIES%2001/iFrame.html>. Recuperado el [19 Abril 2016].
- [6]. Jaime Aranz Tudela. Desarrollo de aplicaciones para dispositivos móviles sobre la plataforma android de google. 2009. Disponible en <http://e-archivo.uc3m.es/bitstream/handle/10016/6506/PFC_Jaime_Aranaz_Tudela_2010116132629.pdf;jsession

nid=956A639F6E76DC49754495CFD63C30E2?sequence=1>. Recuperado el [24 Abril 2016].

[7]. Ramón Núñez del Arco. Aplicativo para dispositivos móviles sobre información turística y hotelera de la ciudad de guayaquil dirigido a turistas nacionales y extranjeros. 2013. Disponible en: <<http://docplayer.es/9183502-Universidad-catolica-de-santiago-de-guayaquil-facultad-de-artes-y-humanidades-ingenieria-en-sistemas-multimedia.html>>. Recuperado [20 Febrero 2016].

[8]. Mónica Lucía Tapia Marroquín. Estudio y desarrollo de aplicaciones para dispositivos móviles android. 2013. Disponible en: <<http://repositorio.utn.edu.ec/bitstream/123456789/2614/1/04%20ISC%20284%20TESIS.pdf>>. Recuperado el [23 Febrero 2016].

[9]. Gartner. Gartner Says Worldwide Smartphone Sales Grew 9.7 Percent in Fourth Quarter of 2015. 2016. Disponible en: <<http://www.gartner.com/newsroom/id/3215217>>. Recuperado el [15 Mayo 2016].

[10]. Javier Espinosa Alfonso. Framework de desarrollo de aplicaciones web multiplataforma. 2012. Disponible en: <http://e-archivo.uc3m.es/bitstream/handle/10016/17139/tfg_javier_espinosa_alfonso_2012.pdf?sequence=1>. Recuperado el [24 Abril 2016].

[11]. Desarrolloweb.com. Editor de código. 2016. Disponible en: <<http://www.desarrolloweb.com/wiki/editor-de-codigo.html>>. Recuperado el [1 Abril 2016].

[12]. Html5facil. Brackets.io. Un editor de texto Open Source por Adobe. 2014. Disponible en: <<http://html5facil.com/tips/brackets-io-un-editor-de-texto-open-source-por-adobe/>>. Recuperado el [5 Mayo 2016].

- [13]. Notepad-plus-plus. Acerca de. 2016. Disponible en: <<https://notepad-plus-plus.org/>>. Recuperado el [10 Abril 2016].
- [14]. Ghatol, Rohit, Patel, Yogesh Beginning. PhoneGap Mobile Web Framework for JavaScript and HTML5. 2012. Disponible en: <<http://www.springer.com/us/book/9781430239031>>. Recuperado el [12 Febrero 2016].
- [15]. Ricardo Jose Iskandar Morine. Estudio comparativo de alternativas y frameworks de programación, para el desarrollo de aplicaciones móviles en entorno Android. 2013. Disponible en:<http://upcommons.upc.edu/bitstream/handle/2099.1/18249/Proyecto_Final_de_Carrera_UPC_-_Ricardo_Iskandar.pdf>. Recuperado el [15 Febrero 2016].
- [16]. Developers A. Android Developers. 2013. Disponible en: <<https://developer.android.com/preview/index.html>>. Recuperado el [20 Febrero 2016].
- [17]. Massachusetts Institute of Technology. MIT AppInventor. 2015. Disponible en: <<http://appinventor.mit.edu/explore/learn.html>>. Recuperado el [14 Febrero 2016].
- [18]. Cordova PhoneGap. PhoneGap Documentation. 2013. Disponible en: <<http://docs.phonegap.com/>>. Recuperado el [26 Febrero 2016].
- [19]. Brad Broulik. Pro jQuery Mobile. 2011. Disponible en: <<http://www.apress.com/9781430239666>>. Recuperado el [18 Febrero 2016].
- [20]. The jQuery Foundation. JQueryMobile. 2013. Disponible en: <<https://jquerymobile.com/resources/>>. Recuperado el [19 Febrero 2016].

- [21]. Condesa. Corona SDK: Otra alternativa para desarrollar juegos en Android. 2011. Disponible en: <<http://androideity.com/2011/08/24/corona-sdk-alternativa-para-desarrollar-juegos-android/>>. Recuperado el [14 Febrero 2016].
- [22]. Jose Angel Zamora. Las mejores alternativas a la programación nativa en Android. 2014. Disponible en: <<http://www.elandroidelibre.com/2014/05/las-mejores-alternativas-a-la-programacion-nativa-en-android.html>>. Recuperado el [23 Febrero 2016].
- [23]. Adobe. Creación de aplicaciones de ADOBE® AIR®. 2013. Disponible en:<http://help.adobe.com/es_ES/air/build/air_buildingapps.pdf>. Recuperado el [12 Marzo 2016].
- [24]. Johanna Rengifo, Carlos Betancourt. Frameworks y herramientas para el desarrollo de aplicaciones orientadas a dispositivos móviles. 2011. Disponible en: <<http://repositorio.utp.edu.co/dspace/bitstream/11059/2490/1/00416R412.pdf>>. Recuperado el [26 Febrero 2016].
- [25]. Carlos Azaustre. ¿qué framework o librería de javascript elegir para mis desarrollos?. 2015. Disponible en: <<https://carlosazaustre.es/blog/frameworks-de-javascript/>>. Recuperado el [12 Marzo 2016].
- [26]. Google.com.co. Google Colombia ofrecido en: Español (Latinoamérica). 2016. Disponible en:<https://www.google.com.co/?gws_rd=ssl>. Recuperado el [12 Marzo 2016].
- [27]. Microsoft. Bing. 2016. Disponible en:<<https://www.bing.com/?setlang=es>>. Recuperado el [1 Marzo 2016].

- [28]. Yahoo. YAHOO!. 2016. Disponible en: <<https://es.search.yahoo.com/>>. Recuperado el [15 Marzo 2016].
- [29]. Villena Román, J. Sistemas de Recuperación de Información. 2014. Disponible en: <<http://www.mat.upm.es/~jmg/doct00/RecupInfo.pdf>>. Recuperado el [21 Marzo 2016].
- [30]. Dominich, S. 'A unified mathematical definition of classical information retrieval'. Journal of the American Society for Information Science. 51 (7). 2000. p. 614-624.
- [31]. Juan Ramón López Rodríguez. Recuperación de información en BDs Documentales. 2012. Disponible en: <<http://docencia.lbd.udc.es/bdd/teoria/tema3/3.-RecuperacionDeInformacion.pdf>>. Recuperado el [10 Marzo 2016].
- [32]. Fidel Cacheda. Introducción a los modelos clásicos de Recuperación de Información. 2008. Disponible en: <<https://revistas.ucm.es/index.php/RGID/article/viewFile/RGID0808110365A/9331>>. Recuperado el [22 Marzo 2016].
- [33]. Jesús Vilares Ferro. Aplicaciones del Procesamiento del Lenguaje Natural en la Recuperación de Información en Español. 2005. Disponible en: <<http://coleweb.dc.fi.udc.es/cole/library/ps/Vil2005a.pdf>>. Recuperado el [17 Marzo 2016].
- [34]. Yunior César Fonseca Reyna. Recuperación de la información: taxonomía de sus modelos. 2012. Disponible en: <<http://www.redalyc.org/pdf/3783/378343676008.pdf>>. Recuperado el [8 Marzo 2016].

[35]. Ministerio de Educación Nacional (MEN). SPADIES. 2012. Disponible en: <<http://www.mineducacion.gov.co/sistemasdeinformacion/1735/w3-article-254668.html>>. Recuperado el [12 Marzo 2016].