

**UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA
(UNAN-MANAGUA)
RECINTO UNIVERSITARIO "RUBEN DARIO"
FACULTAD DE CIENCIAS E INGENIERIAS
DEPARTAMENTO DE COMPUTACION**



Seminario de graduación para optar al título de Licenciatura en Ciencias de la Computación.

TEMA:

Programación de aplicaciones para Móviles.

TEMA DELIMITADO:

Aplicación móvil bajo la Plataforma Android para la Agenda de Actividades del Centro Cultural "Pablo Antonio Cuadra", año 2014.

PRESENTADO POR:

- Br. Engels Holson Tenorio.
- Br. Christiane Cano Dávila.
- Br. Mercedes Salazar Silva.

TUTOR: Lic. Roberto Ezequiel Morales Suárez.

FECHA: jueves 11 de diciembre, 2014

Dedicado a:

Son muchas las personas especiales a las que agradezco su apoyo incondicional y compañía en todas las etapas de mi vida. Algunas están aquí conmigo y otras en mi corazón, pero sin importar en donde estén, esta tesis es dedicada con mucho amor y empeño a todos ellos.

A mi Abuela Ruth Mireya Orozco Sánchez que ha sido la persona más fundamental en mi vida, más que abuela, mi Madre, mi Mamita.

A mi Madre Idania Dávila Orozco, A mi Hermana Joyner Cano, que han sido fuerza para no desistir en mi camino.

Sobre todas las cosas, se la dedico a Mi Hijo Esli Alexandre Cano, el motor de mi lucha para avanzar en este ámbito, mi inspiración sin vacilar. Ahora me corresponde regresar un poquito de ese inmenso interés genuino a todos ustedes y que no solo sea el desenlace sino el inicio de una etapa más refrescante.

Br. Christiane Jose Cano Davila.

Dedicado a:

A Dios, por darme la oportunidad de vivir y por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

Mi madre, por darme la vida, quererme mucho, creer en mí y porque siempre me apoyaste, Mamá gracias por darme una carrera para mi futuro, todo esto te lo debo a ti.

Mi abuela, por quererme y apoyarme siempre, esto también se lo debo a ella.

Mis tíos, quienes me apoyaron en este con sus consejos y ejemplos.

Todos mis amigos, por compartir los buenos y malos momentos. Infinitas gracias para aquellos familiares y amigos que no recordé al momento de escribir esto.

Br. Engels Amado Holson Tenorio.

Dedicado a:

La presente tesis la dedico con mucho cariño a mis padres y hermanos, porque ellos siempre han estado a mi lado brindándome su apoyo, amor y sus consejos de perseverancia para culminar mi carrera.

A mi esposo que ha sido el impulso durante toda mi carrera, sus palabras de anhelo y amor incondicional, ha sido amigo y compañero, gracias por tener paciencia en todo momento y apoyarme en esta etapa de estudio y esfuerzo.

A mis amigos y compañeros por compartir sus conocimientos, alegrías y momentos difíciles para hacer de esta experiencia una realidad, quienes de una manera muy gentil y desinteresada estuvieron siempre dispuestos asesorándome.

A Dios como ser supremo y creador de todos los que nos rodea y ser mi Guía y Fortaleza en todo momento de mi vida.

Br. Mercedes del Rosario Salazar Silva

Agradecimientos

Agradecemos infinitamente a todas las personas e inclusive a la entidad que hizo posible la realización de este proyecto:

A nuestros padres y abuelos, que nos brindaron la inspiración para continuar esta etapa rumbo a la profesionalización.

A nuestros docentes, que sin duda alguna, compartieron sus conocimientos y experiencias para formar en nosotros valores profesionales destinados al éxito.

Al director del Centro Cultural PAC Lic. Jesús de Santiago, que mostro fe en nosotros para la realización de este proyecto.

INDICE

ÍNDICE DE FIGURAS.....	10
RESUMEN	14
I INTRODUCCION	1
II JUSTIFICACION.....	2
III OBJETIVOS.....	3
3.1 Objetivo General.....	3
3.2 Objetivos Específicos.....	3
IV MARCO TEORICO	4
4.1 Conozcamos la historia del Centro Cultural “Pablo Antonio Cuadra”	4
4.1.2 Un cimiento para la historia	5
4.2 Sistema	7
4.2.1 Sistemas Operativos.....	7
4.2.1.1 El top de los Sistemas Operativos	8
4.2.2 Sistemas operativos multiprogramados.....	10
4.2.3 Componentes de un sistema operativo	11
4.3 Gestión de procesos	11
4.3.1 Gestión de la memoria principal	12
4.3.2 Gestión del almacenamiento secundario	12
4.3.3 El sistema de entrada y salida	12
4.3.4 Sistema de archivos.....	13
4.3.5 Sistemas de protección	13
4.3.6 Sistema de comunicaciones.....	14
4.3.7 Programas de sistema.....	14
4.3.8 Gestor de recursos.....	14
4.4 Sistemas Operativos Móviles	14
4.4.1 Tipos de Sistemas Operativos Móviles.....	15
4.4.1.1 Windows Phone 7	15
4.4.1.2 Symbian 3	15
4.4.1.3 Bada 1.1	16
4.4.1.4 BlackBerry OS V6.0.....	16

4.4.2 Android.....	16
4.4.3 Componentes del Sistema ANDROID	17
4.4.3.1 Actividades (Activity)	17
4.4.3.2 Servicios	18
4.4.3.3 Receptores de mensajes de distribución (Broadcas Treceiver)	19
4.4.3.4 Proveedores de contenido (Contentprovider).....	19
4.5 Arquitectura	19
4.5.1 Marco de trabajo de aplicaciones	19
4.6 Análisis de sistemas.....	20
4.6.1 Conceptos y Análisis:	20
4.6.2 Objetivos del Análisis.	22
4.7 Lenguaje Unificado de Modelado (UML)	23
4.7.1 Estudio de Diagramas	23
4.7.1.1 Modelamiento de Clases	23
4.7.1.2 Casos de Uso	24
4.7.1.3 Diagramas de Interacción.....	27
4.7.1.4 Diagramas de Estados.....	28
4.7.1.5 Diagramas de Actividades.....	29
4.8 Modelos de Procesos de Software	30
4.8.1 El modelo lineal o modelo en casada	30
4.8.2 El modelo incremental	32
4.8.3 El modelo de construcción de prototipos.....	33
4.8.4 El modelo en espiral	34
4.8.5 Ventajas e inconvenientes de los Modelos	35
4.9 Bases de Datos.....	37
4.9.1 Componentes de una Base de Datos:	38
4.9.2 Tipos de Usuarios en Base de Datos.....	38
4.9.3 Niveles de Abstracción en Base de datos.....	39
4.9.4 DBMS	41
4.9.4.1 Características y Objetos:.....	41
4.9.4.2 Integridad de Datos	43
4.9.5 Control y Administración de Recursos.....	43

4.10 Modelos de Bases de Datos.....	45
4.10.1 Modelo Entidad-Relación.....	46
4.10.1.1 Elementos del modelo entidad-relación	46
4.10.2 Normalización.....	50
4.10.2.1 Modelado de la arquitectura del Sistema.	53
4.10.2.2 Especificaciones del Sistema.	53
4.11 SQL.....	53
4.11.1 Características generales del SQL	54
4.11.2 Manipulación de Bases de Datos SQL SERVER 2008	55
4.11.2.1 Características Principales SQL SERVER 2008	56
4.12 Programación.....	57
4.12.1 Historia.....	57
4.12.2 Léxico y programación	58
4.12.3 Programas y algoritmos	58
4.12.4 Compilación	59
4.12.5 Objetivos de la programación	60
4.13 Entorno de Desarrollo.....	61
4.13.1 JAVA	62
4.13.2 Entorno ECLIPSE	63
4.13.3 Herramientas	64
4.13.4 VISUAL STUDIO / C #	67
4.13.4.1 Crystal Reports	69
4.13.4.2 Web Service	70
4.13.4.3 IIS.....	71
4.14 Estudio de Factibilidad	71
4.14.1 Concepto de Factibilidad	72
4.14.2 Objetivos de un Estudio de Factibilidad	72
4.14.3 Factibilidad Estructura.....	72
4.15 Evaluación de un Proyecto Factible	73
4.15.1 Componentes del Estudio de Proyecto Factible	73
4.15.1.1 Estudio de Mercado.....	73
4.15.1.2 Estudio Técnico.....	73

4.15.1.3 Estudio Financiero.....	73
4.16 Pagina web de encuestas (e-encuesta.com).....	74
4.17 Embarcadero ER Studio.....	75
4.17 Adobe Photoshop.....	75
V DISEÑO METODOLOGICO	76
5.1 Tipo de Investigación	76
5.2 Población y Muestra.....	76
5.3 Técnicas de Recolección de Datos	77
5.4 Modelo de desarrollo.....	78
5.4.1 Modelo Incremental	78
VI ANALISIS DE RESULTADOS.....	82
6.1 Resultados de Encuesta.....	82
6.2 Herramientas utilizadas para el desarrollo del software de escritorio y aplicación móvil ..	84
6.3 Descripción de la aplicación móvil	84
6.4 Descripción del software de agenda	85
6.5 Estudio de factibilidad	87
6.5.1 Factibilidad Técnica:	87
6.5.2 Factibilidad Económica:	88
6.5.3 Factibilidad Operacional.....	88
6.6 Base de Datos Normalizada	89
6.7 Diagramas UML	90
6.7.1 Diagrama de Clase	90
6.7.2 Diagrama de Casos de Uso.....	91
6.7.3 Diagramas de Secuencias	108
6.7.4 Diagramas de Actividades	128
6.8 Diccionario de datos.....	138
6.9 Pantallas de aplicativo móvil.....	141
6.10 Pantallas de software de para administrar información de aplicativo	150
6.11 Pruebas realizadas de la aplicación móvil	156
VII CONCLUSIONES.....	157
VIII RECOMENDACIONES	158
IX BIBLIOGRAFIA.....	159

X ANEXOS	160
10.1 Formato de Encuesta a través de Sitio Web Encuestas.Com	160
10.1 La Encuesta	165

ÍNDICE DE FIGURAS.

Figura 1. Interacción entre el SO con el resto de las partes	8
Figura 2. Componentes de un sistema operativo	11
Figura 3. Clase.....	24
Figura 4. Actor	24
Figura 5. Caso de uso.....	25
Figura 6. Asociación	25
Figura 7. Dependencia o Instancia	26
Figura 8. Generalización.....	26
Figura 9. Objeto/Actor.....	27
Figura 10. Mensaje de otro objeto	27
Figura 11. Mensaje al mismo objeto	28
Figura 12. Comportamiento de un diagrama de estado	29
Figura 13. Modelo de cascada.....	31
Figura 14. Modelo Incremental	32
Figura 15. Modelo de construcción de prototipo.....	33
Figura 16. Modelo en Espiral.....	34
Figura 17. Componentes de una base de datos.....	38
Figura 18. Niveles de base de datos	39
Figura 19. Subsistema de DBMS.....	40
Figura 20. Base de datos.....	41
Figura 21. Datos repetidos en una DBMS	42
Figura 22. Ejemplos de Identidades	47
Figura 23. Diagrama de relación.....	48
Figura 24. Diagrama de relación y cardinalidad	48

Figura 25. Diagrama de cardinalidad (0,n)(1,1)	49
Figura 26. Diagrama de cardinalidad varios a varios.....	49
Figura 27. Dependencia funcional	50
Figura 28. Dependencia funcional transitiva	52
Figura 29. Descargar del IDE Eclipse	64
Figura 30. Aceptar la instalación del IDE Eclipse	65
Figura 31. Reiniciar el Eclipse	65
Figura 32. Abrir el SDK Manager	65
Figura 33. Android SDK Manager	66
Figura 34. Instalación de paquetes.....	67
Figura 35. Ilustración de dos incrementos del proyecto PAC	78
Figura 36. Porcentaje de personas que manejan en sus celulares diferentes sistemas operativos.	82
Figura 37. Porcentaje de personas que se conectan a internet por medio de sus celulares.	82
Figura 38. Porcentajes de personas que instalan aplicaciones en su celular.	83
Figura 39. Porcentajes de personas que desearían tener la aplicación móvil del Centro Cultural PAC.....	83
Figura 40. Base de datos Normalizada.....	89
Figura 41. Diagrama de clase.....	90
Figura 42. Caso de uso comprobación de la conectividad a Internet y al webservice.	91
Figura 43. Caso de uso Descargar la aplicación	93
Figura 44. Caso de uso consulta de la agenda en la aplicación móvil.	94
Figura 45. Diagrama de uso al sistema de escritorio Agenda PAC.....	95
Figura 46. Diagrama de uso Administrar el evento.....	96
Figura 47. Diagrama de uso de gestionar el tipo de evento.	98
Figura 48. Diagrama de uso de gestionar el tipo de evento.	100
Figura 49. Diagrama de uso del ingreso del presentador.....	102
Figura 50. Diagrama de uso del ingreso del tipo de presentador	104
Figura 51. Diagrama de uso del ingreso de los datos de la empresa que pertenece el presentador.....	106
Figura 52. Diagrama de secuencia comprobación de conectividad.	108

Figura 53. Diagrama de secuencia consulta a la agenda móvil PAC.	109
Figura 54. Diagrama de secuencia Conexión al webservice y base de datos.....	110
Figura 55. Diagrama de secuencia consulta de los eventos	111
Figura 56. Diagrama de secuencia validar el usuario del software de escritorio	112
Figura 57. Diagrama de secuencia agregar un nuevo tipo de evento.	113
Figura 58. Diagrama de secuencia agregar un nuevo evento.	114
Figura 59. Diagrama de secuencia agregar un nuevo nombre del lugar	115
Figura 60. Diagrama de secuencia agregar un nuevo presentador.....	116
Figura 61. Diagrama de secuencia agregar un nuevo tipo de presentador.....	117
Figura 62. Diagrama de secuencia agregar un nuevo nombre de la empresa.....	118
Figura 63. Diagrama de secuencia modificar la tabla de eventos.	119
Figura 64. Diagrama de secuencia modificar la tabla lugar	120
Figura 65. Diagrama de secuencia modificar la tabla de empresa.....	121
Figura 66. Diagrama de secuencia modificar la tabla presentador.....	122
Figura 67. Diagrama de secuencia modificar la tabla tipo de presentador.....	123
Figura 68. Diagrama de secuencia eliminar la información del evento.	124
Figura 69. Diagrama de secuencia eliminar el tipo de evento.	125
Figura 70. Diagrama de Secuencia eliminar el presentador.....	126
Figura 71. Diagrama de secuencia eliminar el lugar.....	127
Figura 72. Diagrama de actividades conexión en el móvil.....	128
Figura 73. Diagrama de actividades conexión al Web Service en la móvil.....	129
Figura 74. Diagrama de actividad consulta a la agenda en el móvil.....	130
Figura 75. Diagrama de actividad comprobación a la conexión a la base de datos.	131
Figura 76. Diagrama de actividad agregar un nuevo evento	132
Figura 77. Diagrama de actividad agregar un nuevo presentador.....	133
Figura 78. Diagrama de actividades agregar un nuevo evento.	134
Figura 79. Diagrama de actividades agregar tipo de presentador.....	135
Figura 80. Diagrama de actividades agregar un nuevo lugar.	136
Figura 81. Diagrama de actividades agregar un nuevo nombre de la empresa.	137
Figura 82. Icono de la aplicación móvil.....	141

Figura 83. Menú de eventos.	142
Figura 84. Lista de eventos seleccionados del menú de eventos.	143
Figura 85. Detalle del evento	144
Figura 86. Imagen del evento seleccionado.	145
Figura 87. Sin eventos a mostrar.	146
Figura 88. Menú Agenda PAC.	147
Figura 89. Acerca PAC.	148
Figura 90. Acerca PAC.	149
Figura 91. Pantalla de inicio de sesión.	150
Figura 92. Pantalla principal de software.	151
Figura 93. Lista de eventos.....	152
Figura 94. Pantalla de creación de eventos.....	153
Figura 95. Pantalla de edición de evento.....	154
Figura 96. Lista de catálogos del software.	155
Figura 97. Generador de reporte de Agenda PAC.	155
Figura 98. Reporte de eventos.	156

RESUMEN

Los dispositivos móviles constituyen cada vez más una realidad que ofrece al usuario en un mismo y reducido aparato, funciones de comunicación y procesamiento de datos que van mucho más allá de simples llamadas telefónicas.

En el presente proyecto se muestra el análisis y el diseño de una aplicación móvil aunado con un software de escritorio para el Centro Cultural Pablo Antonio Cuadra.

Primeramente se dan a conocer historia, objetivos, requerimientos y necesidades del Centro Cultural PAC, los cuales fueron apoyo para la creación de la aplicación móvil y el sistema de escritorio.

Luego se detallan los objetivos planteados para estos proyectos y los procedimientos y herramientas utilizadas para el desarrollo de la aplicación móvil y el software de escritorio.

Al final se muestra los resultados del análisis y el desarrollo para estos dos proyectos.

I INTRODUCCION

Con el creciente aumento de la tecnología, ahora no basta solamente con una aplicación que se ejecute en un ordenador, cada vez más los usuarios quieren algo nuevo que llene sus expectativas, es aquí donde surge la necesidad de gestionar movilidad y capacidad con una aplicación móvil, que no dificulte su uso y que permita desplazarse al mismo tiempo.

El desarrollo de las aplicaciones móviles se ha implementado en diversos sectores, desde el sector empresarial y financiero, hasta el entretenimiento, ya sea con fines informativos o de ocio. Fue un 5 de noviembre del 2007, que paso a ser una realidad un sistema operativo móvil que revolucionaria el mercado, Android, desde ese entonces el mundo de la tecnología ha evolucionado teniendo cambios, novedades y actualizaciones.

Cabe señalar que este año, Android se expandió con todo, viéndose superadas las expectativas en el mercado de la telefonía móvil dando un giro radical desde los Smartphone hasta las Tablets. Siendo, por lo tanto, esta plataforma una constante expansión y es precisamente aquí donde queremos incursionar.

El presente trabajo implementa el desarrollo de una aplicación móvil bajo Plataforma Android para el calendario de actividades del Centro Cultural "Pablo Antonio Cuadra"; dicho Centro se destaca por promover la cultura y el aprendizaje a diferentes sectores de la sociedad, resultando ser un aporte extraordinario de rescate a la cultura Nicaragüense y a través de nuestro proyecto abrir más puertas, llegar a más personas que es el propósito esencial de la entidad.

También se implementa el desarrollo de un software de escritorio "Agenda PAC", proporcionando una herramienta al personal para registrar en la base de datos la información de eventos de cada mes.

II JUSTIFICACION.

Desde sus inicios el Centro Cultural "**Pablo Antonio Cuadra**" compromete a Librería Hispamer a hacer el sueño de una entidad que sobresalga por educar a los Nicaragüenses una realidad; es por eso que poco a poco y con mucha dedicación, durante estos 15 años, el señor Jesús de Santiago Director del Centro Cultural Pablo Antonio Cuadra (PAC) ha adquirido lotes de terrenos necesarios para arrancar con el proyecto y es hasta el 04 de Diciembre del 2012, que con la puesta de la primera piedra, se inicia la construcción de este moderno edificio.

Desde entonces, El Centro Cultural "Pablo Antonio Cuadra" brinda a la sociedad Nicaragüense un espacio digno donde puedan encontrar educación, arte y cultura a través de las diversas actividades artísticas y educativas, foros, capacitaciones, talleres y conferencias que se desarrollen en el Centro PAC. Todo ello, nos conlleva a analizar que en este mundo de la tecnología conviene grandemente que el PAC sea parte de él, a través de una aplicación móvil que no solamente sirva de guía sino también de atractivo; más allá de los usuarios que ya son parte de la entidad, también se pretende captar a otros, siendo esto un reto, por lo tanto, esta aplicación será un camino a seguir para generar esta cadena.

III OBJETIVOS

3.1 Objetivo General

- Desarrollar una Aplicación Móvil bajo la Plataforma Android para la Agenda de Actividades del Centro Cultural "Pablo Antonio Cuadra", año 2014.

3.2 Objetivos Específicos

- Analizar las diferentes tecnologías y herramientas utilizadas en el diseño y desarrollo de aplicaciones móviles.
- Determinar las condiciones tecnológicas presentes en el Centro Cultural Pablo Antonio Cuadra.
- Diseñar las interfaces correspondiente para la Aplicación Móvil Agenda de Actividades del Centro Cultural Pablo Antonio Cuadra.
- Programar un mecanismo de control para la agenda basados en la actualización de los datos y calendarización de las actividades.

IV MARCO TEORICO

4.1 Conozcamos la historia del Centro Cultural "Pablo Antonio Cuadra"

El Centro Cultural Pablo Antonio Cuadra, nace en honor a la vida y obra de nuestro gran poeta e ideólogo Pablo Antonio Cuadra Cardenal, con el objetivo de crear un espacio presenciado por diferentes expresiones de cultura y arte Nicaragüense.

Surge de los ideales que tenía nuestro poeta y se retoma su iniciativa e interés por nuestra cultura. Por ello se realiza una inversión para poder crear un lugar en el cual se fomente la cultura a través del desarrollo de diversas actividades que nos identifiquen como un pueblo Nicaragüense unido en el apoyo de este concepto.

Actualmente se está trabajando en la creación de alianzas con las organizaciones, al mismo tiempo de actividades culturales mensualmente. Todo con el objetivo de cumplir los sueños y aspiraciones de nuestro poeta Nicaragüense, rescatando lo relacionado a la música, el teatro, artes plásticas, teniendo presentes las tradiciones y costumbres, que muchos artistas han desarrollado y están desarrollando actualmente.

Pablo Antonio Cuadra, gran poeta y pensador, junto con un grupo de intelectuales (Jesús de Santiago, Carlos Mantica, César Ramírez, Germán Romero, Jaime Incer y Humberto Belli), hace más de 15 años sintieron la necesidad de ofrecer un espacio digno para las diferentes expresiones culturales propias de Nicaragua e Internacionales, donde se desarrolle todo tipo de actividades y capacitaciones que aporten un valor significativo a la sociedad y abierto a todos los sectores políticos, sociales y religiosos. De esta inquietud nace la Fundación CALMECAC, Fundación para el rescate, promoción y divulgación de la Cultura Nicaragüense, cuyo presidente fue Pablo Antonio Cuadra.

Una construcción cuya primera piedra fue de papel, de hojas de las que emana la sabiduría de los grandes escritores Nicaragüenses, es hoy el tercer Centro Cultural más grande de Latinoamérica.

Ubicado en el que puja por ser el corazón de Managua, el Centro Cultural Pablo Antonio Cuadra de Hispamer, fue erigido en honor al gestor de la Vanguardia Nicaragüense y según sus impulsores, nació "con el sueño de ofrecer un espacio digno para las diferentes expresiones culturales y educativas propias de Nicaragua, rescatando y divulgando obras de nuestra cultura, abriéndonos al mundo editorial y desarrollando todo tipo de actividades culturales que aporten valor agregado a los Nicaragüenses".

4.1.2 Un cimiento para la historia

El Centro Cultural PAC tiene como visión convertirse en líder a nivel nacional en el fomento de la cultura y el arte, creando, fomentando e investigando el talento Nicaragüense, además de unir a todas las organizaciones, empresas e individuos para alcanzar un desarrollo cultural en todo el país.

La primera piedra de este edificio no se extrajo de cantera alguna, sino que fue tallada con la pluma de los mejores hacedores de palabra que usaron como materia básica el pensamiento y como preservante de la cultura.

El escritor Julio Valle-Castillo, primer Premio Internacional Pablo Antonio Cuadra, fue el encargado de materializar la original idea de Jesús de Santiago: colocó en un cofre, ejemplares de obras universales y Nicaragüenses de los escritores más representativos, asimismo, los diarios del día y un mapa de Nicaragua con su plataforma marítima completa y su Río San Juan. Una vez que los tesoros estuvieron depositados en el cofre, fueron enterrados y sellados con una lápida que inmortaliza esa semilla que germinará cuando el Centro Cultural Pablo Antonio Cuadra abra sus puertas a los Nicaragüenses.

La inauguración se presenció martes 17 de diciembre 2013, y en ella se reunió gran parte de la intelectualidad nacional, así como iconos de la cultura, cuerpo diplomático e invitados especiales.

El acto fue presidido por Jesús de Santiago, propietario de Hispamer; Francisco Castro, motor impulsor de la obra; señor Carlos Blasco, de la Embajada de España; Pedro

Xavier Solís, nieto de PAC; los escritores Sergio Ramírez, Gioconda Belli y Carlos Tünnermann. También estuvieron monseñor Silvio Báez y Julio Ramírez, director de Crédito de Banpro.

“Te queremos mostrar este edificio que comenzamos a construir el año pasado, el 4 de diciembre, en conmemoración del centenario de Pablo Antonio Cuadra, con todos los sistemas ecológicos de ahorro energético; no es mío ni de mi familia, es propiedad intelectual de todos los Nicaragüenses”, afirmó Jesús de Santiago en su discurso.

Para Sergio Ramírez, la apertura de este Centro Cultural es un gran acontecimiento del que hay pocos ejemplos en Centroamérica, porque es parte de una iniciativa privada y de un empresario con visión de país y de futuro. Además, afirmó que en este espacio convivirán los viejos libros de papel y los nuevos libros digitales.

El doctor Carlos Tünnermann se encargó de hacer un breve recorrido por la obra y el legado de PAC, de quien afirmó: es el que con mayor acierto ha dibujado los rasgos que caracterizan el modo de ser Nicaragüense, “calificado por Ernesto Cardenal como el más Nicaragüense de los poetas”. Asimismo, manifestó que durante la intervención Nicaragua dio dos grandes testimonios de nacionalismo: Sandino en la montaña y Pablo Antonio Cuadra en sus poemas nicaragüenses.

Por su lado, el señor Julio Ramírez, en representación del Banco de la Producción, Banpro, entidad que creyó en la viabilidad del proyecto y otorgó el crédito necesario para la construcción del edificio, manifestó que el Centro Cultural PAC será un punto para que jóvenes y adultos conozcan su encomiable contribución a la cultura.

Asimismo, manifestó que en esas instalaciones “la familia podrá compartir el acceso a juguetes didácticos, materiales escolares, bibliotecas electrónicas, talleres de pintura y artesanía; en su auditorio central se ofrecerán revistas culturales, conciertos y ciclos de cine”. También agradeció “en nombre de Banpro Grupo Promerica por la oportunidad de compartir con todos ustedes este sueño convertido en realidad”.

Se hizo hincapié en detalles como un espacio cultural concebido con un concepto innovador, con lucernarios naturales y a pesar de que en la primera planta funcionará la

librería Hispamer, con área para juguetes didácticos, también habrá auditorio para 250 personas, cafetería, Museo PAC, Galería de artesanía y pintura. Espacio para actividades infantiles, Oficina de Intur, Turoperadoras, Instrumentos musicales y Editorial.

El Centro Cultural PAC tiene como objetivos:

- Promover la diversidad de la cultura Nicaragüense a través de foros, talleres, congresos, exposiciones, presentaciones de libros, cine, actividades infantiles, seminarios, etc.
- Crear espacios en los que se desarrollen actividades y proyectos de apoyo a los ciudadanos con capacidades diferentes.
- Trabajar con compromiso y disposición junto a otras organizaciones de carácter cultural, empresarial y gubernamental.
- Promover el desarrollo mental y creativo de los niños con actividades y talleres que sean de ayuda en su desarrollo artístico.
- Potenciar a los artistas y autores nacionales e internacionales.
- Organizar y diseñar una Agenda cultural y turística del país.

4.2 Sistema

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 restantes y anteriores próximos y viceversa.

4.2.1 Sistemas Operativos

Uno de los propósitos del sistema operativo que gestiona el núcleo intermediario consiste en gestionar los recursos de localización y protección de acceso del hardware, hecho que alivia a los programadores de aplicaciones de tener que tratar con estos detalles. La mayoría de aparatos electrónicos que utilizan microprocesadores para funcionar, llevan incorporado un sistema operativo (teléfonos móviles, reproductores de DVD, computadoras, radios, enrutadores, etc.). En cuyo caso, son manejados mediante

una interfaz gráfica de usuario, un gestor de ventanas o un entorno de escritorio, si es un celular, mediante una consola o control remoto si es un DVD y, mediante una línea de comandos o navegador web si es un enrutador.



Figura 1. Interacción entre el SO con el resto de las partes

La modernización de la informática es llevada a cabo a finales de los 80, cuando la filosofía de estructura básica de funcionamiento de los grandes computadores se rediseñó a fin de llevarla a los hogares y facilitar su uso, cambiando el concepto de computador multiusuario, (muchos usuarios al mismo tiempo) por un sistema monousuario (únicamente un usuario al mismo tiempo) más sencillo de gestionar. (O'brien, 2006)

4.2.1.1 El top de los Sistemas Operativos

Según el ingeniero Julio Vélez, para la gran mayoría solo existen sistemas operativos tales como Android y Windows, sin embargo, antes de explotar un poco más la idea del SO Android, se mencionan algunos Sistemas Operativos más, entre ellos:

OS X Mavericks: El sistema operativo más reciente de Apple (antes llamado Mac OS X) no solo evolucionó visualmente. Aunque se sigue basando en Unix tomando como eje el NeXTSTEP con Mach kernel, las mejoras en el rendimiento de aplicaciones como

Safari, iCloud, Calendar, Finder y un mejor manejo del CPU con reducción de uso de energía han mantenido encantados a usuarios de equipos nuevos y moderadamente antiguos (desde 2007).

Apple decidió lanzarlo en forma gratuita y su interfaz es más amable con iOS y Apple TV. ¿Desventajas? Hay que tener una Mac, considerablemente más costosa que una PC común. (Velez, 2014)

Windows 7: Aunque fue lanzado hace más de 4 años, Windows 7 es el preferido de millones de usuarios, incluso ante la presencia del actual Windows 8. ¿Las razones? Microsoft implementó un Kernel híbrido y arquitecturas IA-32 y x86-64 (32 y 64 bits). Después del decepcionante Windows Vista, los entusiastas de PC quedaron encantados con un SO que retomaba lo mejor de los Windows previos a Vista pero con una interfaz visual mucho más llamativa y con guiños a sistemas táctiles y móviles.

A la fecha, Microsoft ofrece gran soporte y constantes mejoras automáticas. Desventajas: A fin de dar mayor fuerza a Windows 8 y Windows Phone, Microsoft podría dejar de brindar soporte de forma inesperada. (Velez, 2014)

Ubuntu 13.10: Sigue siendo un SO basado en Unix con entorno Unity de uso complejo y que exige elevados conocimientos de informática, pero sin duda la versión Saucy Salamander de Ubuntu sigue siendo la favorita de miles de usuarios.

No existen virus, la interfaz visual es estupenda, nube de más de 5GB y un centro de software con miles de aplicaciones y juegos. Lamentablemente se requiere un nivel avanzado de uso de codecs y conocimientos de programación de intermedios a avanzados. (Velez, 2014)

IOS 7: En esta lista combinamos sistemas operativos móviles y para computadoras por igual, y la versión más reciente de iOS no podía faltar en la lista. Aunque criticada por muchos detractores de Apple, lo cierto es que la nueva interfaz visual y simplificación de uso de muchos apps intrínsecos de la reciente versión han derivado en comentarios positivos de usuarios frecuentes de iPhone, iPad y iPod touch, además de atraer a nuevos usuarios (en su mayoría, disidentes de Android).

Muchos siguen quejándose del sistema de contraseñas "delatador" e insisten que Apple terminó copiando a Windows Phone. (Velez, 2014)

Linux Mint: Es básicamente una versión simplificada y más amable con el usuario de Ubuntu. Muchas aplicaciones ya están preinstaladas, es libre de virus y pesa muy poco, dando mejor rendimiento a tu máquina. Y aunque es sumamente estable, jamás tendrá una interfaz visual como la de la versión Ubuntu.

Windows 8: Se caracteriza por una interfaz visual más llamativa y estable, además de ser homogénea tanto en computadoras como tabletas (especialmente la Surface Pro).

Sí requiere recorrer una nueva curva de aprendizaje, pero no es nada del otro mundo. Lo malo: es muy vulnerable a virus, cuesta trabajo comprenderse sin un menú de inicio, y pesa más de 4 GB (vs solo 500 MB de Ubuntu). (Velez, 2014)

4.2.2 Sistemas operativos multiprogramados

Surge un nuevo avance en el hardware: el hardware con protección de memoria. Lo que ofrece nuevas soluciones a los problemas de rendimiento:

- Se solapa el cálculo de unos trabajos con la entrada/salida de otros trabajos.
- Se pueden mantener en memoria varios programas.
- Se asigna el uso de la CPU a los diferentes programas en memoria.

Debido a los cambios anteriores, se producen cambios en el monitor residente, con lo que éste debe abordar nuevas tareas, naciendo lo que se denomina como Sistemas Operativos multiprogramados, los cuales cumplen con las siguientes funciones:

- Administrar la memoria.
- Gestionar el uso de la CPU (planificación).
- Administrar el uso de los dispositivos de E/S.

Cuando desempeña esas tareas, el monitor residente se transforma en un sistema operativo multiprogramados. (O'brien, 2006)

4.2.3 Componentes de un sistema operativo

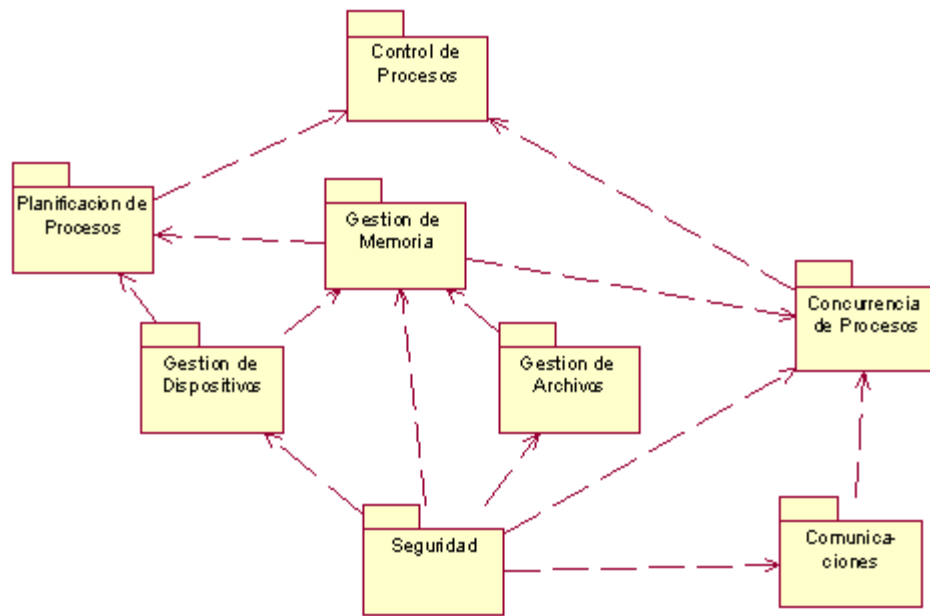


Figura 2. Componentes de un sistema operativo

4.3 Gestión de procesos

Un proceso es simplemente, un programa en ejecución que necesita recursos para realizar su tarea: tiempo de CPU, memoria, archivos y dispositivos de E/S. El sistema operativo es el responsable de:

- Crear y destruir procesos.
- Parar y reanudar procesos.
- Ofrecer mecanismos para que los procesos puedan comunicarse y se sincronicen.

La gestión de procesos podría ser similar al trabajo de oficina. Se puede tener una lista de tareas a realizar y a estas fijarles prioridades alta, media, baja por ejemplo. Debemos comenzar haciendo las tareas de prioridad alta primero y cuando se terminen seguir con las de prioridad media y después las de baja.

Una vez realizada la tarea se tacha. Esto puede traer un problema que las tareas de baja prioridad pueden que nunca lleguen a ejecutarse y permanezcan en la lista para siempre. Para solucionar esto, se puede asignar alta prioridad a las tareas más antiguas.

4.3.1 Gestión de la memoria principal

La memoria es una gran tabla de palabras o bytes que se referencian cada una mediante una dirección única. Este almacén de datos de rápido acceso es compartido por la CPU y los dispositivos de E/S, son volátil y pierde su contenido ante fallos del sistema. El sistema operativo es el responsable de:

- Conocer qué partes de la memoria están siendo utilizadas y por quién
- Decidir qué procesos se cargarán en memoria cuando haya espacio disponible
- Asignar y reclamar espacio de memoria cuando sea necesario

4.3.2 Gestión del almacenamiento secundario

Un sistema de almacenamiento secundario es necesario, ya que la memoria principal (almacenamiento primario) es volátil y además muy pequeña para almacenar todos los programas y datos. También es necesario mantener los datos que no convenga mantener en la memoria principal. El sistema operativo se encarga de:

- Planificar los discos.
- Gestionar el espacio libre.
- Asignar el almacenamiento.
- Verificar que los datos se guarden en orden.

4.3.3 El sistema de entrada y salida

Consiste en un sistema de almacenamiento temporal (caché), una interfaz de manejadores de dispositivos y otra para dispositivos concretos. El sistema operativo debe gestionar el almacenamiento temporal de E/S y servir las interrupciones de los dispositivos de E/S.

4.3.4 Sistema de archivos

Los archivos son colecciones de información relacionada, definidas por sus creadores. Éstos almacenan programas (en código fuente y objeto) y datos tales como imágenes, textos, información de bases de datos, etc. El sistema operativo es responsable de:

- Construir y eliminar archivos y directorios.
- Ofrecer funciones para manipular archivos y directorios.
- Establecer la correspondencia entre archivos y unidades de almacenamiento.
- Realizar copias de seguridad de archivos.

Existen diferentes sistemas de archivos, es decir, existen diferentes formas de organizar la información que se almacena en las memorias (normalmente discos) de los ordenadores. Por ejemplo, existen los sistemas de archivos FAT, FAT32, ext3, NTFS, XFS, etc.

Desde el punto de vista del usuario estas diferencias pueden parecer insignificantes a primera vista, sin embargo, existen diferencias muy importantes.

Por ejemplo, los sistemas de ficheros FAT32 y NTFS, que se utilizan fundamentalmente en sistemas operativos de Microsoft, tienen una gran diferencia para un usuario que utilice una base de datos con bastante información ya que el tamaño máximo de un fichero con un sistema de archivos FAT32 está limitado a 4 gigabytes, sin embargo, en un sistema NTFS el tamaño es considerablemente mayor. (O'Brien, 2006).

4.3.5 Sistemas de protección

Mecanismo que controla el acceso de los programas o los usuarios a los recursos del sistema. El Sistema operativo se encarga de:

- Distinguir entre uso autorizado y no autorizado.
- Especificar los controles de seguridad a realizar.
- Forzar el uso de estos mecanismos de protección.

4.3.6 Sistema de comunicaciones

Para mantener las comunicaciones con otros sistemas es necesario poder controlar el envío y recepción de información a través de las interfaces de red.

También hay que crear y mantener puntos de comunicación que sirvan a las aplicaciones para enviar y recibir información, y crear y mantener conexiones virtuales entre aplicaciones que están ejecutándose localmente y otras que lo hacen remotamente.

4.3.7 Programas de sistema

Son aplicaciones de utilidad que se suministran con el sistema operativo pero no forman parte de él. Ofrecen un entorno útil para el desarrollo y ejecución de programas, siendo algunas de las tareas que realizan:

- Manipulación y modificación de archivos.
- Información del estado del sistema.
- Soporte a lenguajes de programación.
- Comunicaciones.

4.3.8 Gestor de recursos

Como gestor de recursos, el sistema operativo administra:

- La unidad central de procesamiento (donde está alojado el microprocesador).
- Los dispositivos de entrada y salida.
- La memoria principal (o de acceso directo).
- Los discos (o memoria secundaria).
- Los procesos (o programas en ejecución).

4.4 Sistemas Operativos Móviles

Un sistema operativo móvil es un sistema operativo que controla un dispositivo móvil al igual que los PCs utilizan Windows o Linux entre otros. Sin embargo, los sistemas operativos móviles son mucho más simples y están más orientados a la conectividad

inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos.

Algunos de los sistemas operativos utilizados en los dispositivos móviles están basados en el modelo de capas.

4.4.1 Tipos de Sistemas Operativos Móviles

4.4.1.1 Windows Phone 7

Microsoft ha apostado por sus servicios en la Nube o de Cloud-Computing. Te recomendamos esta opción si eres fiel usuario de Windows Live.

- Valoración: Buena.
- Ventajas: Supone una gran mejora respecto a su anterior versión, el Windows Mobile 6.5.

4.4.1.2 Symbian 3

Es el sistema con el que funcionan, de momento, los teléfonos del fabricante Nokia. No está a la altura de sus principales competidores, pero sigue siendo el líder del sector.

- Valoración: Aceptable.
- Ventajas: Funciona correctamente y es fácil de usar.
- Inconvenientes: Es el que menos opciones ofrece en el correo electrónico, una pega importante si hablamos de un Smartphone.
- Aplicaciones: De buena calidad, aunque sin llegar a despertar el mismo interés que las de sus competidores.
- Teléfonos: En todos los que actualmente fabrica Nokia, aunque parece que dentro de poco irá dejando sitio al sistema de Windows.
- Futuro: Nokia vende tal número de teléfonos que, pese a no derrochar calidad, Symbian sigue siendo el líder de los sistemas, el más extendido. El pacto entre Nokia y Microsoft para incorporar Windows Phone a los futuros Smartphone hace pensar que Symbian quedará relegada para los modelos Nokia de gama baja.

- Resultados del test de la OCU: Bueno en configuración inicial, archivos multimedia y aplicaciones. Aceptable en Internet y actualizaciones-copias de seguridad. Malo en la gestión del correo electrónico.

4.4.1.3 Bada 1.1

Es un sistema operativo creado por Samsung exclusivamente para su serie de teléfonos Wave.

- Valoración: Aceptable.
- Ventajas: Junto con iOS, es el único compatible con Outlook. Suele funcionar en teléfonos que son más baratos.
- Inconvenientes: Es inestable y da muchos errores, especialmente en el correo electrónico. Limitada compatibilidad en formatos de vídeo.

4.4.1.4 BlackBerry OS V6.0

Este sistema está pensado especialmente para dar servicio a las empresas y profesionales.

- Valoración: Aceptable.
- Ventajas: Un teclado físico muy cómodo para escribir. En España, los operadores de telefonía móvil se hacen cargo de todos los trámites y se lo ponen fácil al usuario: algunos operadores tienen tarifas específicas para BlackBerry.
- Inconvenientes: Tiene una configuración inicial realmente complicada, en especial la del correo electrónico, que depende de servidores propios de BlackBerry e incluye una cuota mensual. Su teclado físico no es lo mejor para navegar.

4.4.2 Android

Android es un sistema operativo basado en el kernel de Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas,

inicialmente desarrollado por Android, Inc. Google respaldó económicamente y más tarde compró esta empresa en 2005.

Android fue presentado en 2007 junto la fundación del Open Handset Alliance: un consorcio de compañías de hardware, software y telecomunicaciones para avanzar en los estándares abiertos de los dispositivos móviles.

El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008. Android ha visto numerosas actualizaciones desde su liberación inicial. Estas actualizaciones al sistema operativo base típicamente arreglan bugs y agregan nuevas funciones. Generalmente cada actualización del sistema operativo Android es desarrollada bajo un nombre en código de un elemento relacionado con postres en orden alfabético.

La reiterada aparición de nuevas versiones que, en muchos casos, no llegan a funcionar correctamente en el hardware diseñado para versiones previas, hacen que Android sea considerado uno de los elementos promotores de la obsolescencia programada.

Android ha sido criticado muchas veces por la fragmentación que sufren sus terminales al no ser soportado con actualizaciones constantes por los distintos fabricantes.

Se creyó que esta situación cambiaría tras un anuncio de Google en el que comunicó que los fabricantes se comprometerán a aplicar actualizaciones al menos 18 meses desde su salida al mercado, pero esto al final nunca se concretó y el proyecto se canceló. Actualmente hay una aplicación disponible en Android 2.2 que intenta disminuir la fragmentación por parte de Google. (Soriano, 2012).

4.4.3 Componentes del Sistema ANDROID

4.4.3.1 Actividades (Activity)

Son las encargadas de mostrar la interfaz de usuario e interactuar con él. Responden a los eventos generados por el usuario (pulsar botones, etc.). Heredan de la clase Activity.

El aspecto de la actividad se aplica pasando un objeto View (Encargado de dibujar una parte rectangular en la pantalla, pueden contener más objetos View, además todos los componentes de la interfaz (botones, imágenes, etc.) heredan de View) al método `Activity setContentView ()`, que es el método encargado de dibujar la pantalla.

Normalmente las vistas ocupan toda la pantalla, pero se pueden configurar para que se muestren como flotantes. Las actividades también pueden llamar a componentes que se mostrarán sobre su View (como diálogos o menús).

Por cada pantalla distinta hay una actividad distinta, normalmente las aplicaciones tienen una actividad fijada como punto de entrada.

Por ejemplo:

Una aplicación que lee el correo tendrá las siguientes actividades:

- ✓ `RecibidosActivity`: muestra el listado de mensajes recibidos.
- ✓ `LeerMensajeActivity`: muestra el contenido de un mensaje.
- ✓ `CrearMensajeActivity`: recibe como parámetro los datos necesarios, si no hay, muestra el formulario para rellenarlos y envía el mensaje.

4.4.3.2 Servicios

No tienen interfaz visual y se ejecutan en segundo plano, se encargan de realizar tareas que deben continuar ejecutándose cuando nuestra aplicación no está en primer plano. Todos los servicios extienden de la clase `Service`.

Continuando con el ejemplo anterior, la aplicación de correo tendrá un servicio que comprobará y descargará nuevos correos. Es posible lanzar o conectar con un servicio en ejecución con la interfaz que proporciona la clase `Service`.

Los servicios disponen de un mecanismo para ejecutar tareas pesadas sin bloquear la aplicación ya que se ejecutan en un hilo distinto.

4.4.3.3 Receptores de mensajes de distribución (BroadcastReceiver)

Simplemente reciben un mensaje y reaccionan ante él, extienden de la clase BroadcastReceiver, no tienen interfaz de usuario, pero pueden lanzar Actividades como respuesta a un evento o usar NotificationManager para informar al usuario.

Android habitualmente lanza muchas notificaciones de sistema (llamadas entrantes, nuevos correos, nuevos sms, etc.). Si ponemos como ejemplo la aplicación del correo mencionada anteriormente, esta tendría un BroadcastReceiver escuchando el mensaje nuevo_correo, que lanzaría el servicio cada vez que detectara uno. Cuando esto sucediera, se mandaría un aviso a la barra del sistema para alertar al usuario.

4.4.3.4 Proveedores de contenido (ContentProvider)

Ponen un grupo de datos a disposición de distintas aplicaciones, extienden de la clase ContentProvider para implementar los métodos de la interfaz, pero para acceder a esta interfaz se ha de usar una clase llamada ContentResolver.

Con esta clase se permite acceder al sistema de ficheros, bases de datos SQLite o cualquier otra fuente de datos unificada.

Un lector de correo podría disponer de un ContentProvider para acceder a la bandeja de entrada y los datos del mensaje. (Soriano, 2012)

4.5 Arquitectura

4.5.1 Marco de trabajo de aplicaciones

Los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework).

Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.

Bibliotecas: Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema.

Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.

Runtime de Android: Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java.

Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente.

Núcleo Linux: Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores.

El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software. (Soriano, 2012)

4.6 Análisis de sistemas

4.6.1 Conceptos y Análisis:

Es un conjunto o disposición de procedimientos o programas relacionados de manera que juntos forman una sola unidad. Un conjunto de hechos, principios y reglas clasificadas y dispuestas de manera ordenada mostrando un plan lógico en la unión de las partes. Un método, plan o procedimiento de clasificación para hacer algo. También es un conjunto o arreglo de elementos para realizar un objetivo predefinido en el procesamiento de la Información. Esto se lleva a cabo teniendo en cuenta ciertos principios:

- Debe presentarse y entenderse el dominio de la información de un problema.
- Defina las funciones que debe realizar el Software.

- Represente el comportamiento del software a consecuencias de acontecimientos externos.
- Divida en forma jerárquica los modelos que representan la información, funciones y comportamiento.

El proceso debe partir desde la información esencial hasta el detalle de la Implementación.

La función del Análisis puede ser dar soporte a las actividades de un negocio, o desarrollar un producto que pueda venderse para generar beneficios. Para conseguir este objetivo, un Sistema basado en computadoras hace uso de seis (6) elementos fundamentales:

- Software, que son Programas de computadora, con estructuras de datos y su documentación que hacen efectiva la logística metodología o controles de requerimientos del Programa.
- Hardware, dispositivos electrónicos y electromecánicos, que proporcionan capacidad de cálculos y funciones rápidas, exactas y efectivas (Computadoras, Censores, maquinarias, bombas, lectores, etc.), que proporcionan una función externa dentro de los Sistemas.
- Personal, son los operadores o usuarios directos de las herramientas del Sistema.
- Base de Datos, una gran colección de informaciones organizadas y enlazadas al Sistema a las que se accede por medio del Software.
- Documentación, Manuales, formularios, y otra información descriptiva que detalla o da instrucciones sobre el empleo y operación del Programa.
- Procedimientos, o pasos que definen el uso específico de cada uno de los elementos o componentes del Sistema y las reglas de su manejo y mantenimiento. (kendall, 2006)

Un Análisis de Sistema se lleva a cabo teniendo en cuenta los siguientes objetivos en mente:

- Identifique las necesidades del Cliente.
- Evalúe que conceptos tiene el cliente del sistema para establecer su viabilidad.
- Realice un Análisis Técnico y económico.
- Asigne funciones al Hardware, Software, personal, base de datos, y otros elementos del Sistema.
- Establezca las restricciones de presupuestos y planificación temporal.
- Cree una definición del sistema que forme el fundamento de todo el trabajo de Ingeniería.

Para lograr estos objetivos se requiere tener un gran conocimiento y dominio del Hardware y el Software, así como de la Ingeniería humana (Manejo y Administración de personal), y administración de base de datos. (kendall, 2006)

4.6.2 Objetivos del Análisis.

✓ **Identificación de Necesidades.**

Es el primer paso del análisis del sistema, en este proceso el Analista se reúne con el cliente y/o usuario (un representante institucional, departamental o cliente particular), e identifican las metas globales.

A su vez, se analizan las perspectivas del cliente, sus necesidades y requerimientos, sobre la planificación temporal y presupuestal, líneas de mercadeo y otros puntos que puedan ayudar a la identificación y desarrollo del proyecto.

Antes de su reunión con el analista, el cliente prepara un documento conceptual del proyecto, aunque es recomendable que este se elabore durante la comunicación Cliente – analista, ya que de hacerlo el cliente solo de todas maneras tendría que ser modificado, durante la identificación de las necesidades.

Cabe señalar, que la necesidad identificada en el Centro Cultural tiene que ver con el aumento de interés y el índice de asistentes al lugar y esto nos conlleva al diseño y desarrollo así como implementación de la Aplicación ANDROID. (Kendall, 2006)

4.7 Lenguaje Unificado de Modelado (UML)

El Lenguaje de Modelamiento Unificado (UML - Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software.

UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables. (Scott, 1999).

4.7.1 Estudio de Diagramas

Contempla el estudio de tres diagramas:

- Modelamiento de clases
- Casos de uso
- Diagrama de interacción

4.7.1.1 Modelamiento de Clases

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso.

Un diagrama de clases está compuesto por los siguientes elementos:

- Clase: atributos, métodos y visibilidad.
- Relaciones: Herencia, Composición, Agregación, Asociación y Uso.

Elementos

- **Clase**

Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase). A través de ella podemos modelar el entorno en estudio (una Casa, un Auto, una Cuenta Corriente, etc.). (Scott, 1999).

En UML, una clase es representada por un rectángulo que posee tres divisiones:

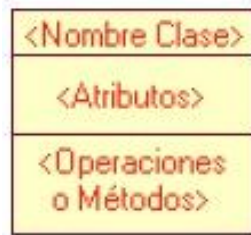


Figura 3. Clase

- **Superior:** Contiene el nombre de la Clase
- **Intermedio:** Contiene los atributos (o variables de instancia) que caracterizan a la Clase (pueden ser private, protected o public).
- **Inferior:** Contiene los métodos u operaciones, los cuales son la forma como interactúa el objeto con su entorno (dependiendo de la visibilidad: private, protected o public).

4.7.1.2 Casos de Uso

El diagrama de casos de uso representa la forma en cómo un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso).

Un diagrama de casos de uso consta de los siguientes elementos:

- Actor.
- Casos de Uso.
- Relaciones de Uso, Herencia y Comunicación.

Elementos

- **Actor:**

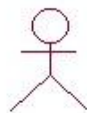


Figura 4. Actor

Una definición previa, es que un **Actor** es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.

Como ejemplo a la definición anterior, tenemos el caso de un sistema de ventas en que el rol de Vendedor con respecto al sistema puede ser realizado por un Vendedor o bien por el Jefe de Local. (Scott, 1999)

- **Caso de Uso:**

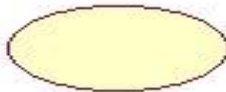


Figura 5. Caso de uso

Es una operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

- **Relaciones:**

- **Asociación**



Figura 6. Asociación

Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple.

- **Dependencia o Instanciación**

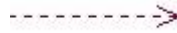


Figura 7. Dependencia o Instancia

Es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se instancia (se crea). Dicha relación se denota con una flecha punteada.

- **Generalización**



Figura 8. Generalización

Este tipo de relación es uno de los más utilizados, cumple una doble función dependiendo de su estereotipo, que puede ser de **Uso** (<<uses>>) o de **Herencia** (<<extends>>).

Este tipo de relación está orientado exclusivamente para casos de uso (y no para actores).

Extends: Se recomienda utilizar cuando un caso de uso es similar a otro (características).

Uses: Se recomienda utilizar cuando se tiene un conjunto de características que son similares en más de un caso de uso y no se desea mantener copiada la descripción de la característica.

De lo anterior cabe mencionar que tiene el mismo paradigma en diseño y Modelamiento de clases, en donde está la duda clásica de **usar** o **heredar**.

4.7.1.3 Diagramas de Interacción

El diagrama de interacción, representa la forma en cómo un Cliente (Actor) u Objetos (Clases) se comunican entre sí en petición a un evento. Esto implica recorrer toda la secuencia de llamadas, de donde se obtienen las responsabilidades claramente.

Dicho diagrama puede ser obtenido de dos partes, desde el Diagrama Estático de Clases o el de Casos de Uso (son diferentes).

Los componentes de un diagrama de interacción son:

- Un Objeto o Actor.
- Mensaje de un objeto a otro objeto.
- Mensaje de un objeto a sí mismo.

Elementos

- **Objeto/Actor:**

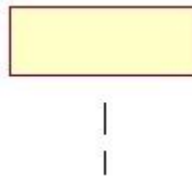


Figura 9. Objeto/Actor

El rectángulo representa una instancia de un Objeto en particular, y la línea punteada representa las llamadas a métodos del objeto.

- **Mensaje a Otro Objeto:**

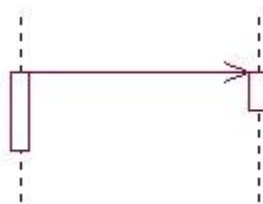


Figura 102. Mensaje de otro objeto

Se representa por una flecha entre un objeto y otro, representa la llamada de un método (operación) de un objeto en particular.

- **Mensaje al Mismo Objeto:**

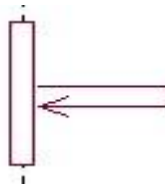


Figura 11. Mensaje al mismo objeto

No solo llamadas a métodos de objetos externos pueden realizarse, también es posible visualizar llamadas a métodos desde el mismo objeto en estudio. (Scott, 1999).

4.7.1.4 Diagramas de Estados

Los diagramas de máquinas de estado son útiles para describir el comportamiento de clases y sistemas que han sido concebidos haciendo uso de un modelo de estados.

En un modelo de estados se identifican las situaciones en la que el comportamiento o capacidad de respuesta con cualitativamente diferentes, así como los eventos o condiciones bajo las que se pasa de una situación a otra (transiciones de estados). (DRAKE, 2013).

Los diagramas de estados son intensivamente utilizados en:

- Sistemas de tiempo real y críticos.
- La descripción de sistemas reactivos.
- La descripción de sistemas basados en protocolos.

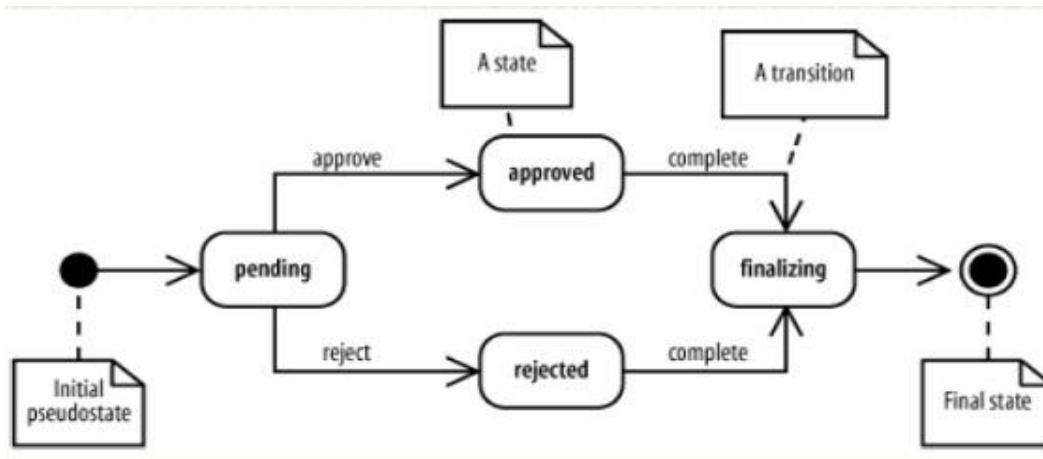


Figura 3. Comportamiento de un diagrama de estado

Los diagramas de estado describen gráficamente los eventos y los estados de los objetos; los diagramas de estado son útiles, entre otras cosas, para indicar los eventos del sistema en los casos de uso.

Un evento es un acontecimiento importante a tomar en cuenta para el sistema.

Un estado es la condición de un objeto en un momento determinado: el tiempo que transcurre entre eventos.

Una transición es una relación entre dos estados, e indica que, cuando ocurre un evento, el objeto pasa del estado anterior al siguiente. (DRAKE, 2013).

En UML, los estados se representan mediante óvalos. Las transiciones se representan mediante flechas con el nombre del evento respectivo. Se acostumbra poner un estado inicial (círculo negro).

4.7.1.5 Diagramas de Actividades

Un diagrama de actividad es un caso especial de un diagrama de estados (otro diagrama de UML, que discutiremos más adelante en la materia) en donde todos o al menos la mayoría de los estados son estados de acciones y en donde todas o al menos la mayoría de las transiciones son disparadas por la finalización de las acciones que las alimentan.

Un diagrama de actividad está asociado a la implementación de un caso de uso. El propósito de este diagrama es enfocarse en los flujos manejados por el procesamiento interno (en contraposición con eventos externos). Se debe usar diagrama de actividad en situaciones donde todos o la mayoría de los eventos representan la finalización de acciones generadas internamente (esto es, flujo de control procedural). Este tipo de diagrama no es adecuado en situaciones donde ocurren eventos asincrónicos.

4.8 Modelos de Procesos de Software

4.8.1 El modelo lineal o modelo en cascada

Es el más antiguo de todos los modelos de Ingeniería del Software. El modelo lineal presenta una estructura secuencial (de ahí el nombre de Modelo en cascada) formada por seis fases o etapas:

- Análisis del Sistema
- Análisis de Requisitos de Software
- Diseño
- Codificación
- Prueba
- Mantenimiento

Las fases incluyen dentro de sí determinadas tareas que clasifican de una forma clara el trabajo a realizar.

El desarrollo de las fases, como he mencionado antes, se produce de manera secuencial. Una vez se produce el análisis tanto del Sistema como de los requisitos del software demandado por el cliente, (fases en las que la intervención del cliente es absolutamente necesaria), se procede a la fase de diseño de la arquitectura global del software.

Un diseño elaborado de forma cuidadosa llevará a una rápida codificación. Tras haber traducido el programa a un lenguaje comprensible para el ordenador, se comprueban los elementos de forma individual y más tarde de manera homogénea (todos los sistemas a la vez).

Una vez entregado el software al cliente, la fase de Mantenimiento comprenderá las actualizaciones y las correcciones de errores que sean necesarias en el programa.

El Modelo en cascada no permite retroceder (más tarde analizaremos las ventajas e inconvenientes de todos los modelos en común), por lo que se hace estrictamente necesario que al final de cada fase el analista de sistemas o, en su caso, el programador, verifique y valide todo el trabajo realizado, ya que un error no detectado a tiempo podría perjudicar gravemente la fecha de entrega del software a nuestro cliente. (Sommerville, 2005)

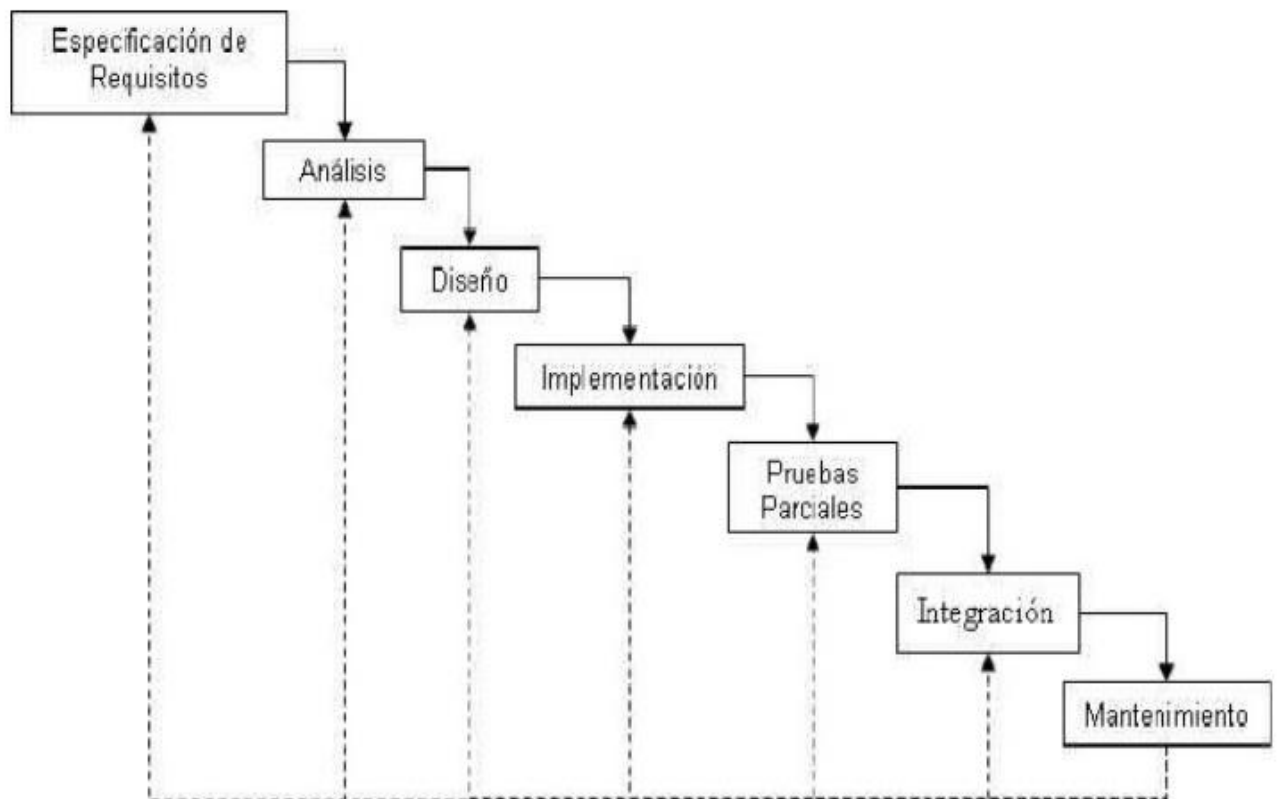


Figura 13. Modelo de cascada

4.8.2 El modelo incremental

El modelo incremental es una evolución del modelo de cascada; viene a suplir el problema de no poder retroceder en las fases de desarrollo del software. Es, por tanto, un modelo no secuencial.

El funcionamiento es sencillo. Comienza con el análisis de los requisitos, tras el cual se prepara un primer diseño.

La novedad de este modelo respecto del anterior, es la introducción de iteraciones para "bifurcar" diseños. Es decir, este modelo ofrece la posibilidad de comenzar un diseño, arquitectura, estructura, etc. del software, que de no convencer al cliente (o al propio programador) es rechazado y se comienza con una segunda iteración (o un segundo diseño), sin necesidad de realizar un nuevo análisis de requisitos. Pueden realizarse tantas iteraciones (también llamadas incrementos) como sean necesarias. (Sommerville, 2005)

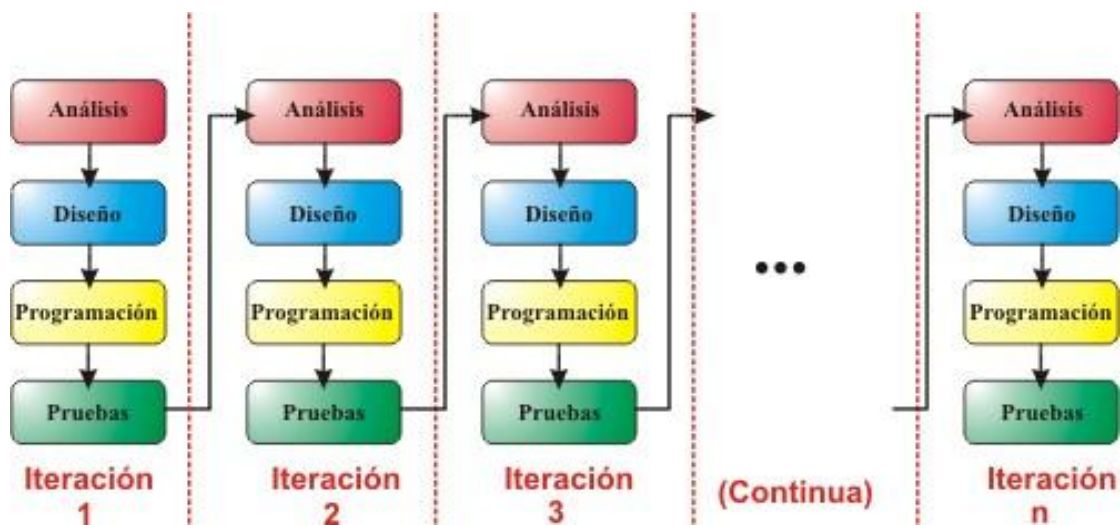


Figura 44. Modelo Incremental

4.8.3 El modelo de construcción de prototipos

Este modelo no secuencial, basado en la construcción de simulaciones o modelos ejecutables de aplicaciones más extensos, persigue un objetivo principal: la participación directa del cliente en la construcción del software requerido.

Las fases son similares a las del modelo en cascada: es necesario un análisis previo de los requisitos tanto del sistema como del cliente, se concibe la arquitectura del sistema y se realiza el diseño del software. Sin embargo, se incluye un elemento hasta ahora no utilizado, que consiste en el diseño rápido de un prototipo que se mostrará al cliente para que evalúe el trabajo realizado. (Sommerville, 2005)

El prototipo es una versión reducida del programa completo; es una "fachada virtual" que mostramos al cliente (que carece de la posibilidad de ser utilizada de la forma en que lo haríamos con el software final).

Tras recoger los requisitos tanto del cliente como del sistema, se comienza con el diseño rápido del prototipo; el diseño completo obedece al previo diseño de pequeños prototipos específicos para funciones individuales. Más tarde, estos diseños serán unidos en uno sólo.

Después, se procede a la construcción del mismo. Éste prototipo es el que mostraremos al cliente para que lo evalúe y considere cambios en él, aunque no se trate de una versión definitiva.

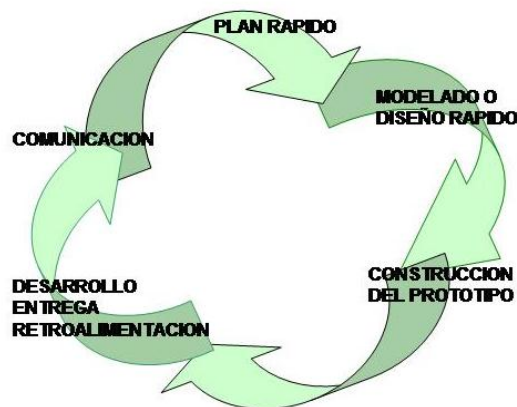


Figura 5. Modelo de construcción de prototipo

4.8.4 El modelo en espiral

Este modelo, también no secuencial, es algo más complejo que los anteriores, aunque incluye un elemento muy útil e importante en el desarrollo del software: análisis de riesgos. El modelo en espiral concreta cuatro fases:

- Planificación
- Análisis de Riesgos
- Ingeniería (Construcción del prototipo)
- Evaluación por el cliente

Si ésta última fase es afirmativa, el modelo continúa con la estructura del Ciclo de vida Clásico. Si el cliente no está satisfecho con el resultado, se cubre otra banda de la espiral y se vuelve a la primera fase (de planificación). (Sommerville, 2005)



Figura 6. Modelo en Espiral

4.8.5 Ventajas e inconvenientes de los Modelos

Podemos considerar al modelo en cascada como el más sencillo de utilizar, aunque también podríamos dudar de su eficacia dado el alto número de inconvenientes que presenta, siendo el principal el que se trate de un modelo secuencial.

Por otra parte, este modelo exige tareas de profundización en el análisis de requisitos del sistema. Si este sistema no es bien conocido, o es difícil de analizar, esta fase puede alargarse demasiado. Ninguno de los modelos es perfecto; el modelo incremental añade la posibilidad de utilizar iteraciones para doblegar el diseño y contemplar varias posibilidades hasta elegir una.

Es un modelo completamente interactivo, que permite trabajar con él en situaciones en las que los cambios de opinión estén a la orden del día. Cada incremento es un paso más en el desarrollo del software final, lo que nos permite cambiar entre iteraciones (o bien proceder a la entrega del software a nuestro cliente como si se tratara de "fascículos semanales").

Esta ventaja es también el principal inconveniente; no en todas las situaciones de desarrollo de software podemos permitirnos la división del trabajo en incrementos, ni tampoco periodificar la entrega de los mismos. Además, aunque la mayoría de las veces el software se puede fragmentar y podemos trabajar con un conjunto de programas determinado, pueden darse situaciones en las que sea imposible ejecutar una iteración sin la existencia de otra que cumpla una función complementaria.

Los prototipos (cambiando de modelo), son una herramienta muy eficaz para imaginar el software completo de una forma rápida y sencilla. De esta forma, incluso observando el prototipo podemos descubrir requerimientos del software en los que antes no habíamos reparado.

Mejora también el proceso de introducción de cambios en el desarrollo de los programas. En el modelo incremental podíamos recurrir a las iteraciones, pero resultará más sencillo (y sobre todo, más visual) realizar éstas modificaciones sobre el prototipo en cuestión.

Además, ésta operación puede realizarse bajo la supervisión del cliente, lo que hace a éste modelo más interactivo aún que su predecesor. (Sommerville, 2005)

Sin embargo, los prototipos tienen un gran problema en contraposición a sus ventajas:

La rapidez con la que se diseñan y construyen puede llevar a errores que no sean detectados en la fase de prueba y acaben integrándose en el producto final. Además, un prototipo es una representación casi exacta del programa final, carente de contenido real. Pero esto es algo que el cliente desconoce; si tiene prisa, puede creer que nuestro trabajo está mucho más avanzado de lo que creía (a pesar de que el prototipo sea tan sólo la fachada de un edificio sin paredes ni escaleras) y puede optar por adelantar la fecha de entrega; al final, el pobre programador es el que paga las consecuencias haciendo horas extras y, además, si se acelera demasiado la construcción del sistema final volvemos al problema de la inclusión de errores no detectados a tiempo.

Por raro que sea, o difícil de entender, el modelo en espiral parece entender los problemas de los anteriores e intentar subsanarlos.

Si en el modelo anterior utilizábamos prototipos para hacernos una idea del software final, en éste modelo los utilizaremos también para hacernos una idea detallada de cuáles son los errores que tiene, o podría tener el programa durante su funcionamiento (lo que antes llamábamos análisis de riesgos).

Esta manera de enfocar el diseño del software permite al cliente evaluar los factores de riesgo que le comunica el prototipo de análisis de riesgo, y según esta información tomar una decisión u otra. Esto hace que el modelo en espiral sea todavía más interactivo que los anteriores.

En cada fase se evalúa el trabajo terminado y, si nos dan el visto bueno, continuamos "girando" en la espiral hasta que llegamos a la evaluación del cliente, la cual decidirá si continuamos en el modelo clásico o volvemos a la primera fase del modelo en espiral.

Todo éste análisis de riesgos (que tan útil parece ser) no parece fácil de utilizar; un análisis de riesgos detallado utilizado sin experiencia podría sobre valorar (o subestimar) los errores que se presenten, haciendo imposible en paso a la siguiente fase (y entonces sí que nos meteríamos en una verdadera espiral sin fin, cosa que al cliente no

debe hacerle mucha gracia). Éste problema genera otro adicional, y es que viendo estas situaciones, será difícil convencer al cliente para que acepte un proyecto realizado bajo las condiciones de éste modelo. (Sommerville, 2005)

4.9 Bases de Datos

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), y por ende se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos.

Existen programas denominados sistemas gestores de bases de datos, abreviado DBMS, que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada.

Las propiedades de estos DBMS, así como su utilización y administración, se estudian dentro del ámbito de la informática.

Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas; También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

Aunque las bases de datos pueden contener muchos tipos de datos, algunos de ellos se encuentran protegidos por las leyes de varios países. Por ejemplo en España, los datos personales se encuentran protegidos por la Ley Orgánica de Protección de Datos de Carácter Personal (LOPD) y en México por la Ley Federal de Transparencia y Acceso a la Información Pública Gubernamental. (Project, 2013)

4.9.1 Componentes de una Base de Datos:

- ✓ **Hardware:** Constituido por dispositivos de almacenamiento como discos, tambores, cintas, etc.
- ✓ **Software:** Es el DBMS o sistema administrador de base de datos.
- ✓ **Datos:** Los cuales están almacenados de acuerdo a la estructura externa y van a ser procesados para convertirse en información.

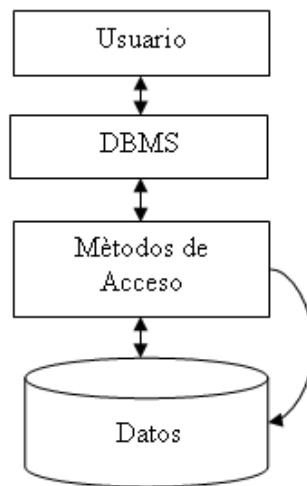


Figura 7. Componentes de una base de datos

4.9.2 Tipos de Usuarios en Base de Datos

- ✓ **Usuario final:** Es la persona que utiliza los datos, esta persona ve datos convertidos en información.
- ✓ **Desarrollador de aplicaciones:** Es la persona que desarrolla los sistemas que interactúan con la base de datos.
- ✓ **DBA:** Es la persona que asegura integridad, consistencia, redundancia, seguridad este es el administrador de base de datos quien se encarga de realizar el mantenimiento diario o periódico de los datos.

Las personas tienen acceso DBMS se clasifican de la siguiente manera:

- ✓ Usuarios Ingenuos: Son aquellos que interactúan con el sistema por medio de aplicaciones permanentes.
- ✓ Usuarios sofisticados: Son aquellos con la capacidad de acceder a la información por medios de lenguajes de consulta.
- ✓ Programadores de aplicación: Son aquellos con un amplio dominio del DML capaces de generar nuevos módulos o utilerías capaces de manejar nuevos datos en el sistema.
- ✓ Usuarios especializados: Son aquellos que desarrollan módulos que no se refieren precisamente al manejo de los datos, si no aplicaciones avanzadas como sistemas expertos, reconocimientos de imágenes, procesamiento de audio y demás (Project, 2013).

4.9.3 Niveles de Abstracción en Base de datos

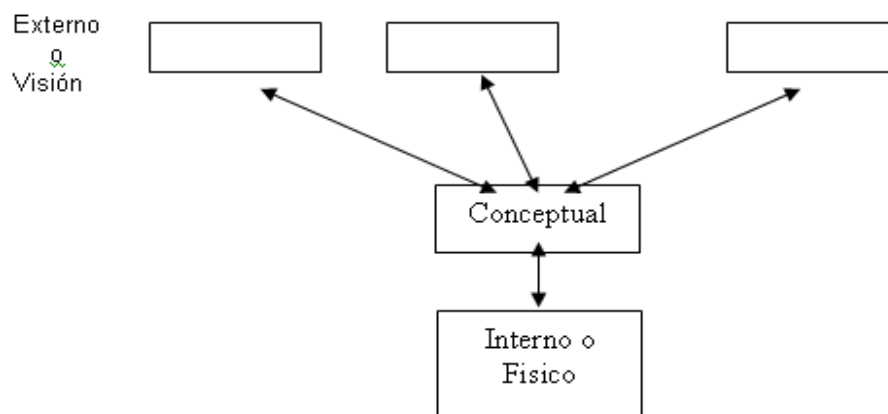


Figura 8. Niveles de base de datos

Externo: Esa es la visión del usuario final, se ve cómo se maneja los datos ya convertidos en información.

Es aquel en el que se presenta al usuario final y que puede combinaciones o relaciones entre los datos que conforman a la base de datos global.

Conceptual: Se ve como está estructurada la Base Datos, equipos de campo tiene como están estructurados los registros.

Es aquel en el que se definen las estructuras lógicas de almacenamiento y las relaciones que se darán entre ellas. Ejemplos comunes de este nivel son el diseño de los registros y las ligas que permitirán la conexión entre registros de un mismo archivo, de archivos distintos incluso, de ligas hacia archivos. (Project, 2013)

Interno: Se ve cómo se almacena los datos físicamente; Es aquel en el que se determinan las características de almacenamiento en el medio secundario. Los diseñadores de este nivel poseen un amplio dominio de cuestiones técnicas y de manejo de hardware.

Los Sistemas Gestores de Bases de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema Gestor de Bases de Datos y DataBase Management System, su expresión inglesa.

Subsistema de un DBMS

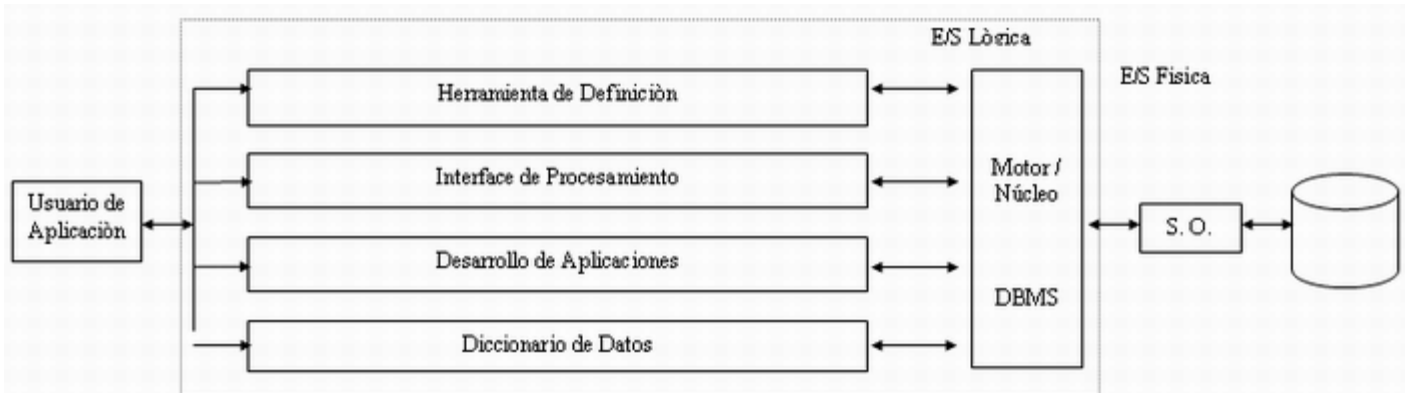


Figura 9. Subsistema de DBMS

Herramientas de definición: Permite definir y modificar la estructura de la Base de Datos, a este nivel definimos lo que se conoce como "Esquema " que es la definición total de base de datos, es que definimos la estructura de la tabla, los tipos de campos, las restricciones para los campos.

- *Subes quema*: manejo de vistas de datos, de niveles externos.
- *Esquema*: manejo de niveles conceptuales.

Interface de Procesamiento: Me provee de las facilidades de actualización, despliegue y visualización de datos.

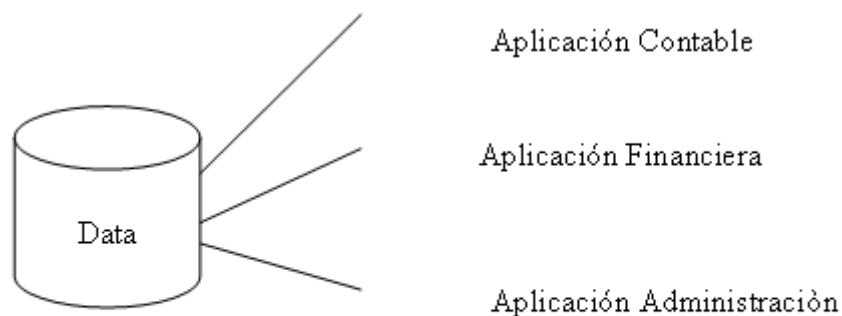
Desarrollo de Aplicaciones: Me permite generar una aplicación por Ej.: generadores de formas, pantalla, código, herramientas case, etc.

Diccionario de Datos: Este es el componente al subsistema con el que interactúan directamente el DBA, le proporciona niveles de consulta y reportes útiles para su trabajo de administración. Es la descripción de la estructura de Base de Datos y relaciones entre datos, y programas.

4.9.4 DBMS

4.9.4.1 Características y Objetos:

- Independencia de Datos: El DBMS me provee una independencia de mis datos vs. las aplicaciones.
- Cambio en datos no implica cambio en programas y viceversa (Menor coste de mantenimiento).



Se cambia las aplicaciones y no se afecta la Data

Figura 10. Base de datos

- Minimizar redundancia (Datos repetidos): Desperdicio de Espacio de Almacenamiento.

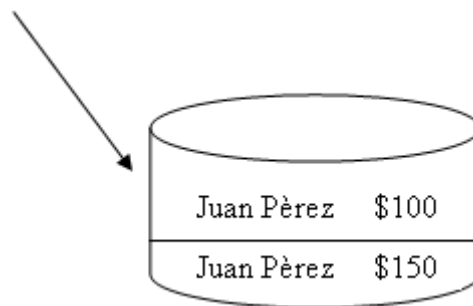


Figura 11. Datos repetidos en una DBMS

Independencia de datos es proteger nuestro programa de aplicaciones frente a las modificaciones en la estructura de datos y viceversa, ya sea en forma física o lógica.

- Independencia Física: Es protección a los programas de aplicación debido a cambios en la estructura de archivos, con cambios en las características de los campos. Ej.: cambio de clave primaria a secundaria.
- Independencia Lógica: Protección a los programas de aplicación cuando se modifica el esquema.

Redundancia, datos repetidos y distribuidos en cualquier parte. El efecto que ocasiona la redundancia es tener inconsistencia de datos y desperdicio de espacio de almacenamiento.

Esta se presenta cuando se repiten innecesariamente datos en los archivos que conforman la base de datos.

- Inconsistencia de Datos: Dato que está en lugar con un valor y encuentra en otro lugar con otro valor. Ej.: se actualiza el archivo cliente pero no se actualiza el archivo de transacciones.

Ocurre cuando existe información contradictoria o incongruente en la base de datos. (Project, 2013)

4.9.4.2 Integridad de Datos

Integridad: Conjunto de seguridades que son utilizadas para mantener los datos correctos.

Ocurre cuando no existe a través de todo el sistema procedimientos uniformes de validación para los datos.

- Fuente de error: Estas fuentes de error se origina si el programa de entrada de datos no está validado. Ej.: fallas de hardware, actualizaciones incompletas, defectos del software, inserción de datos no válidos, errores humanos.

Una técnica que usa el BDMS de una entrada de datos no válida es la validación.

Validación: Es proteger los datos, validar los datos en la entrada de datos. Existen tipos de validaciones:

- **Tipo de Dato:** Se define un campo como carácter o char y no puede ingresar números enteros.
- **Valor de Dato:** Se define un valor entero se puede especificar un rango y no se puede pasar de ese valor.

4.9.5 Control y Administración de Recursos

El DBMS debe proveer al DBA de todos los mecanismos para control y administración de recursos. Para que el DBA con integridad de datos, recuperación de errores e implementación de seguridad.

Establecimiento de Relaciones entre Datos

El BDMS debe proveer los recursos para el establecimiento de relaciones entre los datos, cuales son las relaciones: 1 -> 1, 1 -> n, n -> n.

Ciclo de vida de las operaciones de Base de datos

Etapas:

- Planificación del Proyecto
- Definición del Sistema
- Recolección y Análisis de los Requisitos
- Diseño de la Base de Datos
- Selección del SGDB / DBMS
- Diseño de la Aplicación
- Prototipo
- Implementación
- Conversión y Carga de datos
- Prueba
- Mantenimiento

Estas etapas no son estrictamente secuenciales de hecho hay que repetir algunas de las etapas varias veces haciendo lo que se conoce como "Ciclos de re-alimentación".

Planificación del Proyecto:

Esta etapa con lleva la planificación de cómo se puede llevar a cabo las etapas de ciclo de vida de la manera más eficiente, hay tres componentes principales:

- El trabajo que se va a realizar.
- El recurso para llevarlo a cabo.
- El dinero para pagar todo ello.

Definición del Sistema

En esta etapa se especifica el ámbito y los índices de la aplicación de la base de datos así como con que otros sistemas interactúan. También hay que determinar quiénes son los usuarios y las áreas de la aplicación.

Diseño de Base de datos:

En esta etapa se crea un esquema conceptual de la base de datos. Se desarrollan las especificaciones hasta el punto en que puede comenzar la implementación. Durante esta etapa se crean modelos detallados de las vistas de usuario y sobre todo las relaciones entre cada elemento del sistema, documentando los derechos de uso y manipulación de los diferentes grupos de usuarios.

Si parte de la información necesaria para crear algún elemento establecido ya se encuentra implementado en otro sistema de almacenamiento hay que documentar que relación existirá entre uno y otro y detallar los sistemas que eviten la duplicidad o incoherencia de los datos.

El diseño consta, como se vio anteriormente, de tres fases: El diseño global o conceptual, el diseño lógico y el modelo físico.

Esta etapa consta de tres fases: Diseño conceptual, diseño lógico, diseño físico de la Base de Datos. (Project, 2013)

4.10 Modelos de Bases de Datos

Un modelo de base de datos es un tipo de modelo de datos que determina la estructura lógica de una base de datos y de manera fundamental determina el modo de almacenar, organizar y manipular los datos. (Marquez, 2008)

Entre los modelos lógicos comunes para bases de datos se encuentran:

- Modelo jerárquico
- Modelo en red
- Modelo relacional
- Modelo entidad–relación
- Modelo entidad–relación extendido
- Modelo de objetos
- Modelo documental
- Modelo entidad–atributo–valor

- Modelo en estrella

Los modelos físicos de datos incluyen:

- Índice invertido
- Fichero plano

Otros modelos lógicos pueden ser:

- Modelo asociativo
- Modelo multidimensional
- Modelo multivalor
- Modelo semántico
- Base de datos XML

4.10.1 Modelo Entidad-Relación

Este modelo es solo y exclusivamente un método del que disponemos para diseñar estos esquemas que posteriormente debemos de implementar en un gestor de BBDD (bases de datos). Este modelo se representa a través de diagramas y está formado por varios elementos.

Este modelo habitualmente, además de disponer de un diagrama que ayuda a entender los datos y como se relacionan entre ellos, debe de ser completado con un pequeño resumen con la lista de los atributos y las relaciones de cada elemento. (Osorio, 2008)

4.10.1.1 Elementos del modelo entidad-relación

Entidad

Las entidades representan cosas u objetos (ya sean reales o abstractos), que se diferencian claramente entre sí.

Para poder seguir un ejemplo durante el artículo añadiré ejemplos sobre un taller mecánico, donde se podría crear las siguientes entidades:

- **Coches** (objeto físico): Contiene la información de cada taller.

- **Empleado** (*objeto físico*): Información de los trabajadores.
- **Cargo del empleado** (*cosa abstracta*): Información de la función del empleado.

Estas entidades se representan en un diagrama con unos rectángulos, como los siguientes.

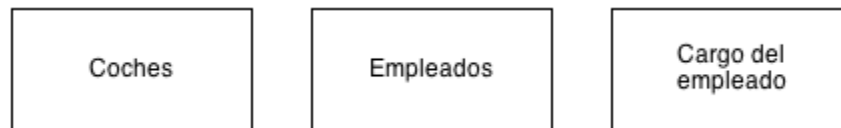


Figura 12. Ejemplos de Identidades

Atributos

Los atributos definen o identifican las características de entidad (**es el contenido de esta entidad**). Cada entidad contiene distintos atributos, que dan información sobre esta entidad. Estos atributos pueden ser de distintos tipos (numéricos, texto, fecha...).

Siguiendo el ejemplo de antes podemos analizar los atributos de nuestra entidad "**Coches**", que nos darán información sobre los coches de nuestro supuesto taller.

Unos posibles atributos serían los siguientes: Número de chasis, matrícula, DNI del propietario, marca, modelo y muchos otros que complementen la información de cada coche. (Genbeta, 2013)

Relación

Es un vínculo que nos permite definir una dependencia entre varias entidades, es decir, nos permite exigir que varias entidades compartan ciertos atributos de forma indispensable.

Por ejemplo, los empleados del taller (de la entidad "**Empleados**") tienen un cargo (según la entidad "**Cargo del empleado**"). Es decir, un atributo de la entidad "Empleados" especificará que cargo tiene en el taller, y tiene que ser idéntico al que ya existe en la entidad "Cargo del empleado".

Las relaciones se muestran en los diagramas como rombos, que se unen a las entidades mediante líneas. (Genbeta, 2013)



Figura 13. Diagrama de relación

Relaciones de cardinalidad

Podemos encontrar distintos tipos de relaciones según como participen en ellas las entidades. Es decir, en el caso anterior cada empleado puede tener un cargo, pero un mismo cargo lo pueden compartir varios empleados.

Esto complementa a las representaciones de las relaciones, mediante un intervalo en cada extremo de la relación que especifica cuantos objetos o cosas (de cada entidad) pueden intervenir en esa relación.

Uno a uno: Una entidad se relaciona únicamente con otra y viceversa. Por ejemplo, si tuviésemos una entidad con distintos chasis y otra con matrículas deberíamos de determinar que cada chasis solo puede tener una matrícula (y cada matrícula un chasis, ni más en ningún caso). (Genbeta, 2013)



Figura 14. Diagrama de relación y cardinalidad

Uno a varios o varios a uno: Determina que un registro de una entidad puede estar relacionado con varios de otra entidad, pero en esta entidad existir solo una vez. Como ha sido en el caso anterior del trabajador del taller.



Figura 15. Diagrama de cardinalidad (0,n)(1,1)

Varios a varios: Determina que una entidad puede relacionarse con otra con ninguno o varios registros y viceversa. Por ejemplo, en el taller un coche puede ser reparado por varios mecánicos distintos y esos mecánicos pueden reparar varios coches distintos. (Genbeta, 2013)



Figura 16. Diagrama de cardinalidad varios a varios

Los indicadores numéricos indican el primero el número mínimo de registros en una relación y posteriormente el máximo (si no hay límite se representa con una "n").

Claves

Es el atributo de una entidad, al que le aplicamos una restricción que lo distingue de los demás registros (no permitiendo que el atributo específico se repita en la entidad) o le aplica un vínculo (exactamente como comentábamos en las relaciones).

Estos son los distintos tipos:

Superclave: Aplica una clave o restricción a varios atributos de la entidad, para así asegurarse que en su conjunto no se repitan varias veces y así no poder entrar en dudas al querer identificar un registro.

Clave primaria: Identifica inequívocamente un solo atributo no permitiendo que se repita en la misma entidad. Como sería la matrícula o el número de chasis de un coche (no puede existir dos veces el mismo).

Clave externa o clave foránea: Este campo tiene que estar estrictamente relacionado con la clave primaria de otra entidad, para así exigir que exista previamente ese clave.

Anteriormente hemos hablado de ello cuando comentábamos que un empleado indispensablemente tiene que tener un cargo (que lo hemos representado numéricamente), por lo cual si intentásemos darle un cargo inexistente el gestor de bases de datos nos devolvería un error. (Genbeta, 2013).

4.10.2 Normalización

El proceso de normalización de **bases de datos** consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional.

Las bases de datos relacionales se normalizan para:

- Evitar la redundancia de los datos.
- Disminuir problemas de actualización de los datos en las tablas.
- Proteger la integridad de los datos.

En el modelo relacional es frecuente llamar tabla a una relación, aunque para que una tabla sea considerada como una relación tiene que cumplir con algunas restricciones:

- Cada tabla debe tener su nombre único.
- No puede haber dos filas iguales. No se permiten los duplicados.
- Todos los datos en una columna deben ser del mismo tipo.

Dependencia funcional



Figura 17. Dependencia funcional

B es funcionalmente dependiente de **A**.

Una dependencia funcional es una conexión entre uno o más atributos. Por ejemplo si se conoce el valor de DNI tiene una conexión con Apellido o Nombre.

Las dependencias funcionales del sistema se escriben utilizando una flecha, de la siguiente manera:

FechaDeNacimiento \rightarrow Edad

De la normalización (lógica) a la implementación (física o real) puede ser sugerible tener éstas dependencias funcionales para lograr la eficiencia en las tablas.

Propiedades de la Dependencia funcional

Existen 3 axiomas de Armstrong:

Dependencia funcional Reflexiva

Si "y" está incluido en "x" entonces $x \rightarrow y$

A partir de cualquier atributo o conjunto de atributos siempre puede deducirse él mismo. Si la dirección o el nombre de una persona están incluidos en el DNI, entonces con el DNI podemos determinar la dirección o su nombre.

Dependencia funcional aumentativa

$x \rightarrow y$ Entonces $xz \rightarrow yz$

DNI \rightarrow nombre

DNI, dirección \rightarrow nombre, dirección

Si con el DNI se determina el nombre de una persona, entonces con el DNI más la dirección también se determina el nombre y su dirección. (Osorio, 2008)

Dependencia funcional transitiva



Figura 18. Dependencia funcional transitiva

Sean X , Y , Z tres atributos (o grupos de atributos) de la misma entidad. Si Y depende funcionalmente de X y Z de Y , pero X no depende funcionalmente de Y , se dice entonces que Z depende transitivamente de X . Simbólicamente sería:

$X \rightarrow Y \rightarrow Z$ entonces $X \rightarrow Z$

FechaDeNacimiento \rightarrow Edad

Edad \rightarrow Conducir

FechaDeNacimiento \rightarrow Edad \rightarrow Conducir

Entonces tenemos que FechaDeNacimiento determina a Edad y la Edad determina a Conducir, indirectamente podemos saber a través de FechaDeNacimiento a Conducir (En muchos países, una persona necesita ser mayor de cierta edad para poder conducir un automóvil, por eso se utiliza este ejemplo).

"C será un dato simple (dato no primario), B, será un otro dato simple (dato no primario), A, es la llave primaria (PK). Decimos que C dependerá de B y B dependerá funcionalmente de A."

Propiedades deducidas

Unión

$x \rightarrow y$ y $x \rightarrow z$ entonces $x \rightarrow yz$

✓ **Pseudo-transitiva**

$x \rightarrow y$ y $wy \rightarrow z$ entonces $wx \rightarrow z$

✓ **Descomposición**

$x \rightarrow y$ y z está incluido en y entonces $x \rightarrow z$

4.10.2.1 Modelado de la arquitectura del Sistema.

Cuando queremos dar a entender mejor lo que vamos a construir en el caso de edificios, herramientas, aviones, maquinas, se crea un modelo idéntico, pero en menor escala (más pequeño).

Sin embargo cuando aquello que construiremos es un software, nuestro modelo debe tomar una forma diferente, deben representar todas las funciones y subfunciones de un sistema. Los modelos se concentran en lo que debe hacer el sistema no en como lo hace, estos modelos pueden incluir notación gráfica, información y comportamiento del sistema.

4.10.2.2 Especificaciones del Sistema.

Es un documento que sirve como fundamento para la ingeniería hardware, software, base de datos, e ingeniería humana. Describe la función y rendimiento de un sistema basado en computadoras y las dificultades que estarán presentes durante su desarrollo. Las especificaciones de los requisitos del software se producen en la terminación de la tarea del análisis.

En conclusión un proyecto de desarrollo de un sistema de Información comprende varios componentes o pasos llevados a cabo durante la etapa del análisis, el cual ayuda a traducir las necesidades del cliente en un modelo de sistema que utiliza uno más de los componentes: Software, hardware, personas, base de datos, documentación y procedimientos.

4.11 SQL

El SQL es un lenguaje estándar de programación para el acceso a bases de datos; se utiliza para acceder y manipular datos en cualquier base de datos del mercado, como

por ejemplo, para las bases de datos MySQL, Oracle, DB2, SQL Server, Access. (Meza, 2013)

El SQL es un lenguaje estructurado y un estándar ANSI para el acceso y manipulación de los datos de cualquier base de datos.

El SQL se compone de sentencias SQL, cada una con una utilidad diferente, como por ejemplo:

- Creación de una base de datos (CREATE DATABASE)
- Creación de una tabla (CREATE TABLE)
- Creación de una vista (CREATE VIEW)
- Creación de un índice de una tabla (CREATE INDEX)
- Creación de procedimientos almacenados (CREATE PROCEDURE)
- Creación de disparadores (CREATE TRIGGER)
- Consultar los datos almacenados en una tabla (SELECT)
- Insertar datos en una tabla (INSERT)
- Modificar datos ya existentes en una tabla (UPDATE)
- Borrar datos almacenados en una tabla (DELETE)
- Dar permiso de acceso a los datos de una tabla (GRANT)
- Eliminar permisos de acceso a datos de una tabla (REVOKE)
- Finalizar la transacción de una sentencia SQL (COMMIT)
- Retroceder la transacción de una sentencia SQL (ROLLBACK).

4.11.1 Características generales del SQL

El SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales y permite así gran variedad de operaciones.

Es un lenguaje declarativo de "alto nivel" o "de no procedimiento" que, gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros y no a registros individuales permite una alta productividad en codificación y la orientación a objetos.

De esta forma, una sola sentencia puede equivaler a uno o más programas que se utilizarían en un lenguaje de bajo nivel orientado a registros. SQL también tiene las siguientes características:

- **Lenguaje de definición de datos:** El LDD de SQL proporciona comandos para la definición de esquemas de relación, borrado de relaciones y modificaciones de los esquemas de relación.
- **Lenguaje interactivo de manipulación de datos:** El LMD de SQL incluye lenguajes de consultas basado tanto en álgebra relacional como en cálculo relacional de tuplas.
- **Integridad:** El LDD de SQL incluye comandos para especificar las restricciones de integridad que deben cumplir los datos almacenados en la base de datos.
- **Definición de vistas:** El LDD incluye comandos para definir las vistas.
- **Control de transacciones:** SQL tiene comandos para especificar el comienzo y el final de una transacción.
- **SQL incorporado y dinámico:** Esto quiere decir que se pueden incorporar instrucciones de SQL en lenguajes de programación como: C++, C, Java, PHP, Cobol, Pascal y Fortran.
- **Autorización:** El LDD incluye comandos para especificar los derechos de acceso a las relaciones y a las vistas. (Meza, 2013)

4.11.2 Manipulación de Bases de Datos SQL SERVER 2008

SQL SERVER es un sistema administrador de Base de Datos Relacional, Cliente – Servidor, que permite una mayor escalabilidad de explorar objetos de Base de Datos y la integración de secuencias de los comandos en la base de Datos OLTP y OLAP.

Contiene las versiones 2000, 2005, 2008, R2 y 2012, esta última versión fue presentada en este año. (Marquez, 2008)

En sus últimas dos versiones SQL SERVER facilita una plataforma integral empresarial con procedimientos analíticos integrados en la cual se incluye:

- El procesamiento Analítico en Línea (OLAP).

- Minería de Datos (OLAP).
- Las Herramientas de gestión y administración.
- El almacenamiento de datos y desarrollo de informes.
- SQL SERVER facilitara a las empresas a construir y desarrollar sin complicaciones
- aplicaciones de inteligencia empresarial robustas y controlar el costo en el desarrollo de estas aplicaciones. Permite a realizar los siguientes aspectos:
- Desarrollar e innovar aplicaciones empresariales.
- Optimizar la productividad de los TI, reduce la complejidad en la creación y administración de la aplicación de base de datos.
- Aumentar las capacidades de los programadores con un entorno de desarrollo Flexible y actual.
- Compartir datos a través de múltiples plataformas y aplicaciones.

4.11.2 1 Características Principales SQL SERVER 2008

- Comprensión de datos.
- Paralelismo de tablas particionada.
- Optimización de Star Join Query.
- Integración optimizada con el sistema Office.
- Reporting Services Mejorado.
- Las consultas distribuidas en el motor de base de datos.
- Los altos niveles de seguridad, encriptación de datos.
- Programabilidad en el motor de Base de Datos.
- Desarrollo de base de datos espejos.
- Integración en XML de datos Relacionados.
- Arquitectura de interfaz virtual.

4.12 Programación

Proceso de diseñar, codificar, depurar y mantener el código fuente de programas computacionales. El código fuente es escrito en un lenguaje de programación.

El propósito de la programación es crear programas que exhiban un comportamiento deseado; el proceso de escribir código requiere frecuentemente conocimientos en varias áreas distintas, además del dominio del lenguaje a utilizar, algoritmos especializados y lógica formal.

Programar no involucra necesariamente otras tareas tales como el análisis y diseño de la aplicación (pero sí el diseño del código), aunque sí suelen estar fusionadas en el desarrollo de pequeñas aplicaciones.

Existe un error común que trata por sinónimos los términos 'lenguaje de programación' y 'lenguaje informático'.

Los lenguajes informáticos engloban a los lenguajes de programación y a otros más, como por ejemplo HTML (lenguaje para el marcado de páginas web que no es propiamente un lenguaje de programación, sino un conjunto de instrucciones que permiten estructurar el contenido de los documentos). (Gual, 1985)

4.12.1 Historia

Para crear un programa, y que la computadora lo intérprete y ejecute las instrucciones escritas en él, debe usarse un lenguaje de programación. En sus inicios las computadoras interpretaban sólo instrucciones en un lenguaje específico, del más bajo nivel, conocido como código máquina, siendo éste excesivamente complicado para programar.

De hecho sólo consiste en cadenas de números 1 y 0 (sistema binario). Para facilitar el trabajo de programación, los primeros científicos que trabajaban en el área decidieron reemplazar las instrucciones, secuencias de unos y ceros, por palabras o letras provenientes del inglés; las codificaron y crearon así un lenguaje de mayor nivel, que se conoce como Assembly o lenguaje ensamblador.

Por ejemplo, para sumar se usa la letra A de la palabra inglesa add (sumar). En realidad escribir en lenguaje ensamblador es básicamente lo mismo que hacerlo en lenguaje máquina, pero las letras y palabras son bastante más fáciles de recordar y entender que secuencias de números binarios.

A medida que la complejidad de las tareas que realizaban las computadoras aumentaba, se hizo necesario disponer de un método sencillo para programar. Entonces, se crearon los lenguajes de alto nivel.

Mientras que una tarea tan trivial como multiplicar dos números puede necesitar un conjunto de instrucciones en lenguaje ensamblador, en un lenguaje de alto nivel bastará con solo una.

Una vez que se termina de escribir un programa, sea en ensamblador o en un lenguaje de alto nivel, es necesario compilarlo, es decir, traducirlo a lenguaje máquina. (Gual, 1985)

4.12.2 Léxico y programación

La programación se rige por reglas y un conjunto más o menos reducido de órdenes, expresiones, instrucciones y comandos que tienden a asemejarse a una lengua natural acotada (en inglés); y que además tienen la particularidad de una reducida ambigüedad.

Cuanto menos ambiguo es un lenguaje de programación, se dice, es más potente. Bajo esta premisa, y en el extremo, el lenguaje más potente existente es el binario, con ambigüedad nula (lo cual lleva a pensar así del lenguaje ensamblador).

En los lenguajes de programación de alto nivel se distinguen diversos elementos entre los que se incluyen el léxico propio del lenguaje y las reglas semánticas y sintácticas.

4.12.3 Programas y algoritmos

Un algoritmo es una secuencia no ambigua, finita y ordenada de instrucciones que han de seguirse para resolver un problema. Un programa normalmente implementa (traduce a un lenguaje de programación concreto) uno o más algoritmos.

Los programas suelen subdividirse en partes menores, llamadas módulos, de modo que la complejidad algorítmica de cada una de las partes sea menor que la del programa completo, lo cual ayuda al desarrollo del programa. (Pedro Garcia Sevilla, 2000)

Se han propuesto diversas técnicas de programación cuyo objetivo es mejorar tanto el proceso de creación de software como su mantenimiento. Entre ellas, se pueden mencionar las siguientes:

- Programación declarativa.
- Programación estructurada.
- Programación modular.
- Programación orientada a objetos.

4.12.4 Compilación

El programa escrito en un lenguaje de programación (fácilmente comprensible por el programador) es llamado programa fuente y no se puede ejecutar directamente en una computadora. La opción más común es compilar el programa obteniendo un módulo objeto, aunque también puede ejecutarse en forma más directa a través de un intérprete informático.

El código fuente del programa se debe someter a un proceso de traducción para convertirlo en lenguaje máquina, código esté directamente ejecutable por el procesador. A este proceso se le llama compilación.

Habitualmente la creación de un programa ejecutable (un típico.exe para Microsoft Windows o DOS) conlleva dos pasos. El primer paso se llama compilación (propriadamente dicho) y traduce el código fuente escrito en un lenguaje de programación almacenado en un archivo a código en bajo nivel (normalmente en código objeto, no directamente a lenguaje máquina).

El segundo paso se llama enlazado en el cual se enlaza el código de bajo nivel generado de todos los ficheros y subprogramas que se han mandado compilar y se añade el código de las funciones que hay en las bibliotecas del compilador para que el

ejecutable pueda comunicarse directamente con el sistema operativo, traduciendo así finalmente el código objeto a código máquina, y generando un módulo ejecutable.

Estos dos pasos se pueden hacer por separado, almacenando el resultado de la fase de compilación en archivos objetos (un típico.obj para Microsoft Windows, DOS o para Unix); para enlazarlos en fases posteriores, o crear directamente el ejecutable. (Pedro Garcia Sevilla, 2000)

4.12.5 Objetivos de la programación

La programación debe perseguir la obtención de programas de calidad. Para ello se establece una serie de factores que determinan la calidad de un programa. Algunos de los factores de calidad más importantes son los siguientes:

- **Corrección:** Un programa es correcto si hace lo que debe hacer tal y como se estableció en las fases previas a su desarrollo. Para determinar si un programa hace lo que debe, es muy importante especificar claramente qué debe hacer el programa antes de desarrollarlo y, una vez acabado, compararlo con lo que realmente hace.
- **Claridad:** Es muy importante que el programa sea lo más claro y legible posible, para facilitar así su desarrollo y posterior mantenimiento.

Al elaborar un programa se debe intentar que su estructura sea sencilla y coherente, así como cuidar el estilo en la edición; de esta forma se ve facilitado el trabajo del programador, tanto en la fase de creación como en las fases posteriores de corrección de errores, ampliaciones, modificaciones, etc. Fases que pueden ser realizadas incluso por otro programador, con lo cual la claridad es aún más necesaria para que otros programadores puedan continuar el trabajo fácilmente.

Algunos programadores llegan incluso a utilizar Arte ASCII para delimitar secciones de código. Otros, por diversión o para impedir un análisis cómodo a

otros programadores, recurren al uso de código ofuscado. (Pedro Garcia Sevilla, 2000)

- **Eficiencia:** Se trata de que el programa, además de realizar aquello para lo que fue creado (es decir, que sea correcto), lo haga gestionando de la mejor forma posible los recursos que utiliza. Se hace referencia al tiempo que tarda en realizar la tarea para la que ha sido creado y a la cantidad de memoria que necesita, pero hay otros recursos que también pueden ser de consideración al obtener la eficiencia de un programa, dependiendo de su naturaleza (espacio en disco que utiliza, tráfico de red que genera, etc.).
- **Portabilidad:** Un programa es portable cuando tiene la capacidad de poder ejecutarse en una plataforma, ya sea hardware o software, diferente a aquella en la que se elaboró.

La portabilidad es una característica muy deseable para un programa, ya que permite, por ejemplo, a un programa que se ha desarrollado para sistemas GNU/Linux ejecutarse también en la familia de sistemas operativos Windows. Esto permite que el programa pueda llegar a más usuarios más fácilmente.

4.13 Entorno de Desarrollo

Un entorno de desarrollo integrado, llamado también IDE (sigla en inglés de integrated development environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. (Lee, 2012)

Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, PHP, Python, Java, C#, Delphi, Visual Basic, etc. En

algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto, como es el caso de Smalltalk u Objective-C.

4.13.1 JAVA

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.

Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra.

Su sintaxis es la de C++ "simplificada". Los creadores de Java partieron de la sintaxis de C++ y trataron de eliminar de este todo lo que resultase complicado o fuente de errores en este lenguaje.

Java es un lenguaje orientado a objetos, aunque no de los denominados puros; en Java todos los tipos, a excepción de los tipos fundamentales de variables (int, char, long...) son clases. Sin embargo, en los lenguajes orientados a objetos puros incluso estos tipos fundamentales son clases, por ejemplo en Smalltalk.

El código generado por el compilador Java es independiente de la arquitectura: podría ejecutarse en un entorno UNIX, Mac o Windows. El motivo de esto es que el que realmente ejecuta el código generado por el compilador no es el procesador del ordenador directamente, sino que este se ejecuta mediante una máquina virtual. Esto permite que los Applets de una web pueda ejecutarlos cualquier máquina que se conecte a ella independientemente de qué sistema operativo emplee (siempre y cuando el ordenador en cuestión tenga instalada una máquina virtual de Java).

4.13.2 Entorno ECLIPSE

Eclipse es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores.

Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).

Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent o Azureus.

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project, cubriendo casi todas las áreas de Model Driven Engineering. (Lee, 2012)

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios. (Lee, 2012)

4.13.3 Herramientas

Developer Tools

- Download
- Installing the SDK
- Exploring the SDK
- NDK

Workflow

Tools Help

Revisions

Extras

Samples

ADK

Get the Android SDK

The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android.

Download the SDK for Windows

Other platforms | System requirements

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r20.0.3-windows.zip	90379469 bytes	cd895c79201f7f02507eb3c3868a1c5e
	installer_r20.0.3-windows.exe (Recommended)	70495456 bytes	cf23b95d0c9cd57fac3c3be253171af4
Mac OS X (intel)	android-sdk_r20.0.3-macosx.zip	58218455 bytes	07dc88ba2c0817ef178a665d002831bf
Linux (i386)	android-sdk_r20.0.3-linux.tgz	82616305 bytes	0d53c2c31d6b5d0cf7385bccd0b06c27

Except as noted, this content is licensed under Creative Commons Attribution 2.5. For details and restrictions, see the Content License

Figura 19. Descargar del IDE Eclipse

1. **Equipo** > Propiedades (botón derecho del ratón) > Administración avanzada del sistema > Variables de entorno >> Editar la variable PATH añadiendo al final, añadiendo (;) en caso de no estar ya puesto, la ruta SDK Manager de Android.
2. **Configurar** el plugin de Android para Eclipse: Esta acción se realiza desde Eclipse, llevando a cabo una actualización de software personalizada. La manera de realizarlo dependerá de la versión de Eclipse utilizada, pero en todos los casos es muy parecido. Se puede obtener más información en developer.android.com/sdk/eclipse-adt.html
 - Help > Install New Software > Pulsar en Add
 - Nombre: ADT (Android Developer Tools)
 - dl-ssl.google.com/android/eclipse
 - OK



Figura 20. Aceptar la instalación del IDE Eclipse



Figura 21. Reiniciar el Eclipse

Hay que instalar los paquetes necesarios para los diferentes dispositivos. Al reiniciar Eclipse sale una pantalla en la que se puede elegir qué instalar.

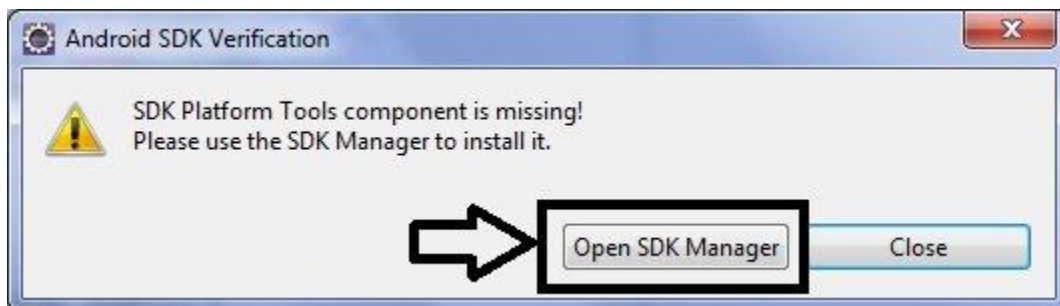


Figura 22. Abrir el SDK Manager

3. a. Windows > Android SDK and AVD Manager > A la izquierda aparece Available Packages.

b. Seleccionar todos los que aparezcan a la derecha. Son bastantes MB, y dependiendo de la conexión pueden tardar más o menos. Quizás no sean todos necesarios, pero no está de más tenerlos instalados por si algún día son necesarios.

c. Para que se instalen hay que pulsar:

Install Selected. Después seleccionar en "Accept All" y pulsar en "Install".

d. Después de tenerlo instalado, es bueno realizar estos pasos para comprobar si existen actualizaciones. (Lee, 2012)

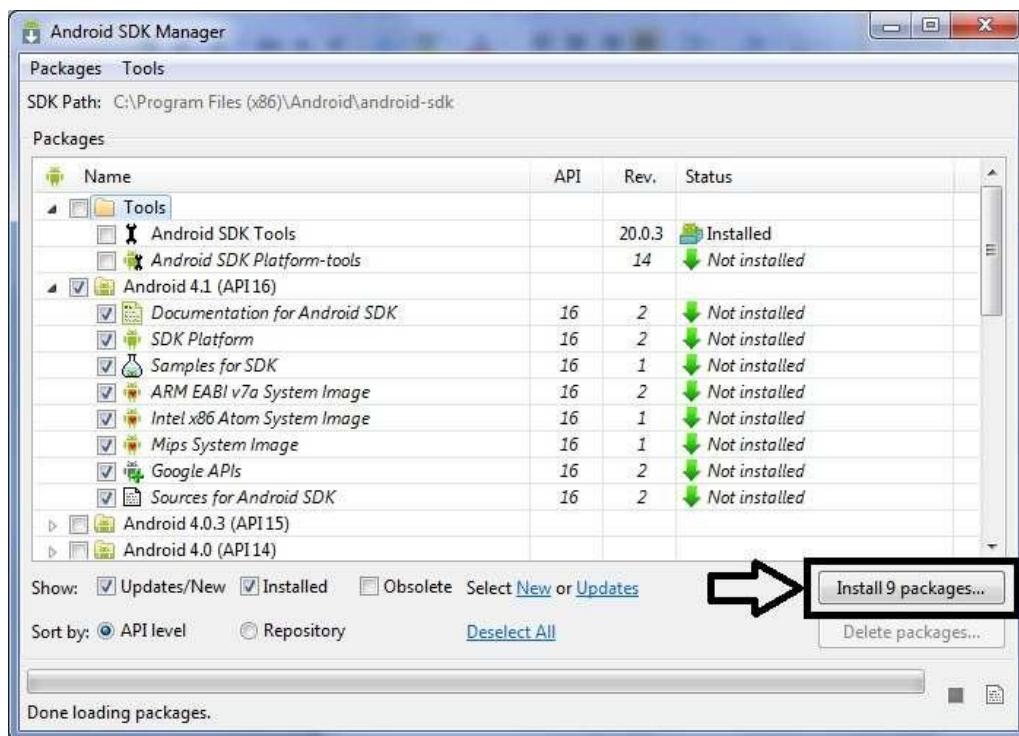


Figura 23. Android SDK Manager

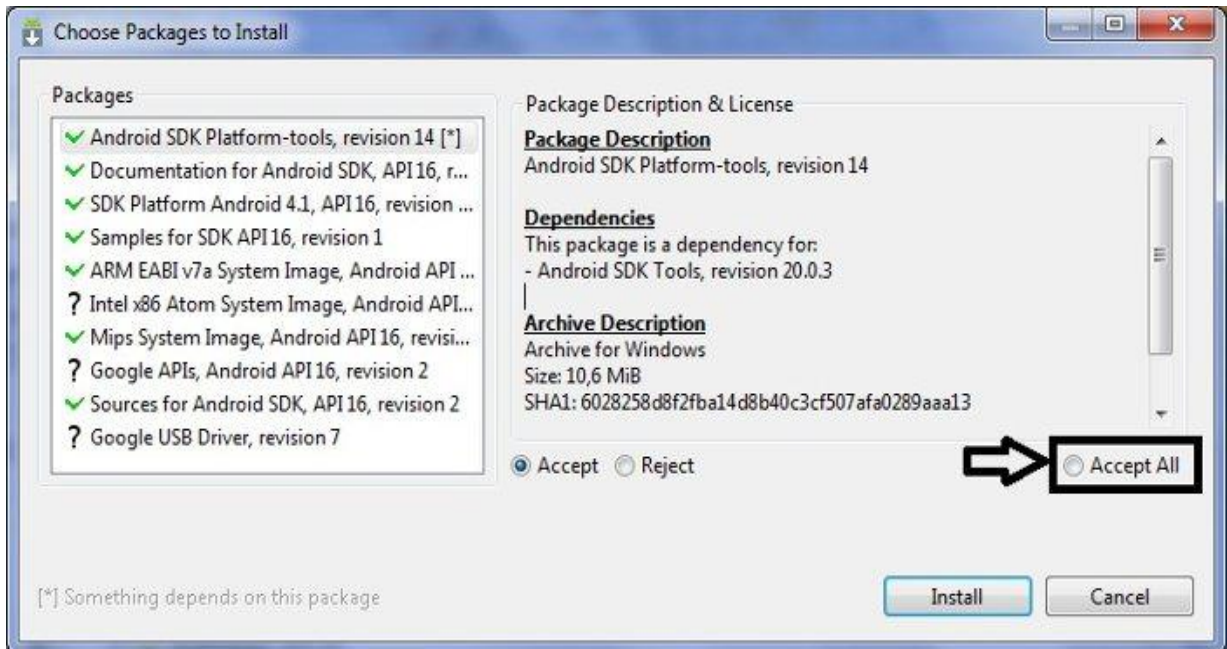


Figura 24. Instalación de paquetes.

En conclusión, Eclipse es un paquete IDE de código abierto cuyos proyectos están enfocados en construir una plataforma de desarrollo extensiva, runtimes y frameworks de aplicaciones para construir, desplegar y manejar software a través del ciclo de vida completo del software.

Además de Java, Eclipse puede manejar perfectamente proyectos en muchos lenguajes de programación populares como C, C++, Perl, PHP, Python, Ruby, Fortran, Haskell, JavaScript, y muchos otros.

La primera versión de Eclipse fue creada a partir de los restos del proyecto Visual Age de IBM Canadá, quien gestionó la formación de un consorcio que supervisaría el desarrollo del software autónomo Eclipse, hecho en Java. (Lee, 2012)

4.13.4 VISUAL STUDIO / C

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta múltiples lenguajes de programación tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby, PHP; al igual que entornos de desarrollo web como ASP.NET MVC, Django, etc., a lo cual

sumarle las nuevas capacidades online bajo Windows Azure en forma del editor Monaco.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos, consolas, etc. (Somasegar, 2013)

Ahora bien, cuando hablamos de C#, este es un lenguaje orientado a objetos elegante y con seguridad de tipos que permite a los desarrolladores compilar diversas aplicaciones sólidas y seguras que se ejecutan en .NET Framework.

Puede utilizar C# para crear aplicaciones cliente de Windows, servicios Web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos. Visual C# proporciona un editor de código avanzado, cómodos diseñadores de interfaz de usuario, depurador integrado y numerosas herramientas más para facilitar el desarrollo de aplicaciones basadas el lenguaje C# y .NET Framework.

Como lenguaje orientado a objetos, C# admite los conceptos de encapsulación, herencia y polimorfismo. Todas las variables y métodos, incluido el método Main que es el punto de entrada de la aplicación, se encapsulan dentro de definiciones de clase. Una clase puede heredar directamente de una clase primaria, pero puede implementar cualquier número de interfaces. (Somasegar, 2013)

Los métodos que reemplazan a los métodos virtuales en una clase primaria requieren la palabra clave override como medio para evitar redefiniciones accidentales. En C#, una struct es como una clase sencilla; es un tipo asignado en la pila que puede implementar interfaces pero que no admite la herencia.

Además de estos principios básicos orientados a objetos, C# facilita el desarrollo de componentes de software a través de varias construcciones de lenguaje innovadoras, entre las que se incluyen las siguientes:

- Firmas de métodos encapsulados denominadas delegados, que habilitan notificaciones de eventos con seguridad de tipos.
- Propiedades, que actúan como descriptores de acceso para variables miembro privadas.
- Atributos, que proporcionan metadatos declarativos sobre tipos en tiempo de ejecución.
- Comentarios en línea de documentación XML.
- Language-Integrated Query (LINQ) que proporciona funciones de consulta integradas en una gran variedad de orígenes de datos. (Somasegar, 2013)

4.13.4.1 Crystal Reports

Crystal Reports es una aplicación de inteligencia empresarial utilizada para diseñar y generar informes desde una amplia gama de fuentes de datos (bases de datos). Varias aplicaciones, como Microsoft Visual Studio, incluyen una versión OEM de Crystal Reports como una herramienta de propósito general para informes/reportes. Crystal Reports se convirtió en el escritor de informes estándar cuando Microsoft lo liberó con Visual Basic.

Crystal Reports ha formado parte de Visual Studio desde 1993, y ahora es el estándar de elaboración de informes de Visual Studio. Se incluye en todas las copias de Visual Studio Professional y se integra directamente en el entorno de desarrollo.

Crystal Reports para Visual Studio incorpora la posibilidad de crear contenido interactivo con calidad de presentación al entorno de Windows. Con Crystal Reports para Visual Studio, puede crear informes complejos y profesionales en un programa basado en GUI.

Después puede conectar el informe a casi todos los orígenes de base de datos, así como a datos proxy, como un conjunto de resultados (por ejemplo, un ADO.NET DataSet).

Los asistentes del diseñador de GUI le permiten establecer fácilmente los criterios de formato, agrupamiento y gráficos, etc.

Puede almacenar el informe en una aplicación Web o para Windows, con uno de los controles de visores de Crystal Reports para Visual Studio. La presentación de informes, tanto en clientes Windows como en HTML 3.2 ó 4.0, es muy interactiva y proporciona funciones como la profundización en gráficos, la exploración de informes y la búsqueda de texto.

Crystal Reports para Visual Studio incluye un SDK extenso. Puede utilizarlo para interactuar con el informe mediante programación en tiempo de ejecución, usando uno de los cuatro modelos de objetos posibles:

- CrystalReportViewer, el modelo de objetos más sencillo.
- ReportDocument, el modelo de objetos más completo.
- ReportClientDocument, el modelo de objetos más completo. Este modelo de objetos está disponibles con Crystal Reports 2008 o con un servidor RAS.
- InfoObject, un modelo de objetos muy eficaz para la programación y configuración de informes en el marco de Crystal Reports Server o BusinessObjects Enterprise.

4.13.4.2 Web Service

Un servicio web (en inglés, Web Service o Web services) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet, la interoperabilidad se consigue mediante la adopción de estándares abiertos.

Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

Es una máquina que atiende las peticiones de los clientes web y les envía los recursos solicitados. (Casati, 2004)

4.13.4.3 IIS

Internet Information Services o IIS es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows. Originalmente era parte del Option Pack para Windows NT.

Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS.

Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Este servicio convierte a una PC en un servidor web para Internet o una intranet, es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente.

Se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas. Por ejemplo, Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl. (Casati, 2004)

4.14 Estudio de Factibilidad

Es necesario analizar los objetivos de la organización para determinar la aplicabilidad de un proyecto que permita el alcance de las metas organizacionales, es por ello que este estudio permite la utilización de diversas herramientas que ayuden a determinar la infraestructura tecnológica y la capacidad técnica que implica la implantación del sistema en cuestión, así como los costos, beneficios y el grado de aceptación que la propuesta genera en la institución. (kendall, 2006)

4.14.1 Concepto de Factibilidad

El estudio de factibilidad es el análisis que realiza una empresa para determinar si el negocio que se propone será bueno o malo, y cuáles serán las estrategias que se deben desarrollar para que sea exitoso. Según el Diccionario de la Real Academia Española, la Factibilidad es la "cualidad o condición de factible". (Kendall, 2006)

4.14.2 Objetivos de un Estudio de Factibilidad

La determinación de factibilidad en general de un proyecto solicitado significa el encontrar cuáles son los objetivos organizacionales, y luego determinar si el proyecto sirve para mover el negocio hacia sus objetivos.

Hay varios objetivos aceptables para los proyectos de sistema que estos incluyen, pero no están limitados, a:

- Reducción de errores y mayor precisión en los procesos.
- Reducción de costos mediante la eliminación de recursos no necesarios.
- Integración de todas las áreas y subsistemas
- Actualización y mejoramiento de los servicios a clientes o usuarios.
- Hacer un plan de producción y comercialización.
- Aceleración en la recopilación de los datos.
- Reducción en el tiempo de procesamiento y ejecución de las tareas.
- Automatización óptima de procedimientos manuales.

4.14.3 Factibilidad Estructura

Para Coss, un estudio de factibilidad tiene una estructura definida y el orden de los contenidos está orientado a:

- a) Presentar un resumen de proyecto.
- b) Elaborar cada parte por separado.
- c) Fundamentar cada parte en las anteriores.
- d) Elaborar conclusiones y recomendaciones.

4.15 Evaluación de un Proyecto Factible

Un proyecto factible, es el que ha aprobado cuatro evaluaciones básicas:

- ✓ Evaluación financiera.
- ✓ Evaluación ambiental.
- ✓ Evaluación socioeconómica.
- ✓ Evaluación técnica.

La aprobación de cada evaluación se le puede llamar viabilidad; estas viabilidades se deben dar al mismo tiempo para alcanzar la factibilidad de un proyecto.

4.15.1 Componentes del Estudio de Proyecto Factible

4.15.1.1 Estudio de Mercado

Según Kendall y Kendall, tiene como finalidad determinar si existe o no, una demanda que justifique la puesta en marcha de un programa de producción de ciertos bienes o servicios, en un espacio de tiempo.

(kendall, 2006).

4.15.1.2 Estudio Técnico

Tiene por objeto proveer información, para cuantificar el monto de las inversiones y costos de las operaciones relativas en esta área.

4.15.1.3 Estudio Financiero

Ordenar y sistematizar la información de tipo monetario que proporcionaron las etapas anteriores.

Consiste en identificar y ordenar todos los ítems de inversiones, costos e ingresos que pueden deducirse de los estudios previos. Sin embargo, y debido a que no se ha proporcionado toda la información necesaria para la evaluación, en esta etapa deben definirse todos aquellos elementos que debe suministrar el propio estudio financiero. El caso clásico es el cálculo del monto que debe invertirse en capital de trabajo o el valor de desecho del proyecto. (kendall, 2006)

4.16 Pagina web de encuestas (e-encuesta.com)

e-encuesta le proporciona la tecnología necesaria para que usted mismo pueda crear, distribuir y analizar encuestas de todo tipo, desde la más sencilla a la más profesional.

→ e-encuesta es una aplicación web propiedad de WEBTOOLS S.L.

WEBTOOLS S.L. es un proveedor de aplicaciones web (SAAS). La empresa ofrece soluciones útiles y eficientes a todo tipo de organizaciones. WEBTOOLS S.L. pone a su disposición herramientas web basadas en las últimas tendencias del mercado y orientadas a satisfacer sus necesidades específicas. Nuestros productos facilitan el trabajo a profesionales de diversos sectores optimizando el uso del tiempo y de los recursos disponibles. Nuestras oficinas están en Madrid, si bien proporcionamos un servicio accesible desde cualquier parte del mundo.

→ Nuestra filosofía se basa en un uso eficiente de la tecnología

Se trata de una nueva forma de comercialización del software: Los costes se comparten entre todos los clientes de la herramienta, de forma que el usuario dispone en todo momento de software avanzado y actualizado a un precio muy razonable.

→ Nos dirigimos a empresas e instituciones de cualquier tamaño o sector

WEBTOOLS S.L. se dirige a todo tipo de empresas y organizaciones: grandes, pequeñas y medianas empresas, instituciones académicas, ONGs, organismos públicos, institutos de investigación, y a profesionales del área de marketing, ventas, recursos humanos, formación, investigadores, profesionales de la educación y estudiantes.

→ Nuestro equipo se compone de profesionales especializados

Se trata de un equipo joven, multidisciplinar, con experiencia en el medio, que apuesta por las grandes ventajas que ofrece internet en el campo de los servicios a empresas.

4.17 Embarcadero ER Studio

Es una poderosa herramienta CASE, para efectuar el modelado lógico y físico de bases de datos relaciones y multidimensionales.

El ER/Studio ofrece:

- Entorno de diseño dirigido por el modelo.
- Soporte al ciclo de vida completo de las base
- Gestión de modelos empresarial-
- Capacidades de comunicación empresariales
- Almacén de datos y soporte a la integración
- Diseño de calidad de base de datos

4.17 Adobe Photoshop.

Adobe Photoshop es un editor de gráficos de adobe system. Usado principalmente para el |retoque de fotografías y gráficos, su nombre en español significa literalmente "taller de fotos". Es líder mundial del mercado de las aplicaciones de imágenes y domina este sector de tal manera que su nombre es ampliamente empleado como sinónimo para la edición de imágenes en general.

Actualmente forma parte de la familia Adobe Creative Suite y es desarrollado y comercializado por Adobe System Incorporated inicialmente para computadores Apple pero posteriormente también para plataformas PC con sistema operativo Windows.

Su distribución viene en diferentes presentaciones, que van desde su forma individual hasta como parte de un paquete, siendo éstos: Adobe Creative Suite Design Premium y Versión Standard, Adobe Creative Suite Web Premium, Adobe Creative Suite Production Studio Premium y Adobe Creative Suite Master Collection.

Características de Photoshop:

- Editor de gráficos ráster
- Editor de gráficos vectoriales

- Licencia: Software propietario
- Escrito en: C++
- Idiomas: más de 25

Las funcionalidades de Photoshop pueden ser extendidas empleando add-ons o plugins, especialmente filtros para realizar diferentes efectos en las imágenes. (Alegsa, 2010).

V DISEÑO METODOLOGICO

5.1 Tipo de Investigación

El presente proyecto está orientado a una investigación aplicada o bien tipo de estudio Aplicativo, cuyo objetivo es lograr un nuevo conocimiento destinado a solucionar problemas prácticos.

El objetivo de la realización del proyecto con referencia al tipo de estudio, es que a través de la realidad transformemos la necesidad creando nuevas tecnologías.

De modo que se hace hincapié en un estudio transversal ya que a través de este se recolectan datos con el propósito de analizar su incidencia y su interrelación en un momento dado, es decir, puntualizar y no hacer algo complejo organizativamente, permitiéndonos diferenciar los efectos y las causas que se pueden presentar en poco tiempo, basándonos en lo que se efectúa en el tiempo presente.

5.2 Población y Muestra

Se puede enfatizar que, el universo de estudio con características definitivas al proyecto planteado, consta de 22 Centros Culturales existentes en Managua. Partiendo de este universo, nuestra muestra es el Centro Cultural "Pablo Antonio Cuadra" ubicando en el Edificio Hispamer Managua.

5.3 Técnicas de Recolección de Datos

- **Entrevista:** La entrevista fue el medio apropiado que encontramos para saber las necesidades, requerimientos y los puntos necesarios para poder implementar la aplicación móvil y el software de escritorio en el centro PAC. Parte de estos puntos fueron dados por el Lic. Jesús de Santiago Blanco, Director general del PAC, la cual se hizo en la fecha 14-03-2014.

Se fueron haciendo visitas consuetudinarias, para entrevistar a las diferentes áreas involucradas en el Centro Cultural PAC, los cuales nos dieron detalles de puntos que les gustaría ver en la aplicación.

- **Encuesta:** De la misma forma, al diseñar la encuesta y confeccionar el cuestionario hay que tomar en cuenta los recursos (tanto humanos como materiales) de los que se disponen, tanto para la recopilación como para la lectura de la información, para así lograr un diseño funcionalmente eficaz. Por lo tanto se hizo uso del tipo de encuesta sobre hechos y de opinión ya que tuvimos como objetivo averiguar lo que el público en general pensaba acerca de una determinada necesidad o lo que considera debía hacerse en una implementación concreta.

5.4 Modelo de desarrollo

5.4.1 Modelo Incremental



Figura 25. Ilustración de dos incrementos del proyecto PAC

Se utilizó este modelo para el desarrollo de la aplicación, debido a que nos permitió conocer y rediseñar los requerimientos que exigía nuestro cliente como tal para llegar a un resultado final y mantenerlo.

Para el desarrollo del software de agenda y aplicación Android, se utilizaron dos incrementos, ya que este modelo de forma interactiva, o sea con cada incremento se entrega parcialmente el producto con ampliación y mejoras de la versión previamente implementada.

Primer incremento:

En el primer incremento se analizó el desarrollo del software de escritorio "Agenda PAC", porque el Centro Cultural no cuenta con un software de escritorio y base de datos para almacenar la información de los eventos.

Análisis:

En la primera fase del modelo incremental se describe la necesidad del personal del Centro Cultura PAC de tener toda la información almacenada en una base de datos, tales como: Tipos de presentadores, nombre de los presentadores y de que empresa pertenecen, la hora y fecha de los eventos y el nombre del evento, el tipo de de entrada.

Diseño:

En esta etapa se hizo el diseño de los diagramas de UML, para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo del software.

Se diseñó la base de datos, los procedimientos de captura de datos.

Se diseñó las pantallas de manera amigable para procesar la información (catálogos, reportes, conexión de la ruta de la base de datos y una pantalla para el ingreso de información: tipo de evento, nombre del evento, hora y fecha, nombre de los representantes, tipo de entrada, fecha de ingreso y un campo donde se activa los eventos).

Codificación:

En esta etapa se codifico la fase del diseño con la finalidad de que el software de agenda funcionara de acuerdo a las necesidades y requerimientos establecidos previamente.

Se validó el ingreso de información por parte del usuario para garantizar la veracidad de los datos.

Se codifico para que el software hiciera su respectiva conexión con el servidor de base de datos.

Prueba:

Esta es la última etapa del modelo incremental en probar el software para localizar y corregir errores dentro del software.

Estas pruebas se llevan a cabo con los usuarios involucrados con el centro cultural y con los programadores.

El personal del Centro Cultural PAC quedo satisfecho con el software, ya que es de fácil manejo cada de uno de los procesos a aplicar para la administración de la información.

Segundo incremento:

En el segundo incremento es el desarrollo de la aplicación móvil.

Análisis:

El director del centro PAC, nos dio a conocer la necesidad del mismo, que es inmiscuir el Centro Cultural PAC en el avance tecnológico, de hacer el calendario de eventos de cada mes electrónicamente y que este directamente en contacto con el cliente. Esto es para disminuir gastos de papel, envió masivos de correos, y no solamente presentarlo en pantallas LCD en el edificio.

Diseño:

Se elaboró diagramas de uml, para interpretar la funcionalidad de la aplicación móvil.

Se diseñó el interfaz de usuario que consta de 3 pantallas (El inicio de la aplicación, la segunda donde presenta los diferentes tipos de eventos y la tercera donde presenta una imagen representativa del evento, nombre del evento, nombre de los presentadores, hora y fecha, tipo de entrada.

Codificación:

En esta etapa de la codificación de decidió desarrollarlo con la plataforma eclipse instalando sus respectivas herramientas (ADT para construir una interfaz de usuario de aplicación, depurar y la exportación firmado paquetes de aplicación APKS para su distribución y El SDK que proporciona las bibliotecas de las API).

Para la consulta con el cliente y con el servidor, el Webservice se desarrolló en Visual Basic 2010 C#, que por medio de parámetros se pueda acceder a la información solicitada.

Prueba:

En esta última etapa se hizo las pruebas con algunos visitantes del Centro Cultural PAC, instalando la aplicación en sus celulares y al mirar los usuarios la interfaz de la aplicación dieron buenos comentarios, de la facilidad de navegación y entendimiento de la misma.

VI ANALISIS DE RESULTADOS

6.1 Resultados de Encuesta



Figura 26. Porcentaje de personas que manejan en sus celulares diferentes sistemas operativos.

¿Con qué frecuencia se conecta a Internet a través del móvil?

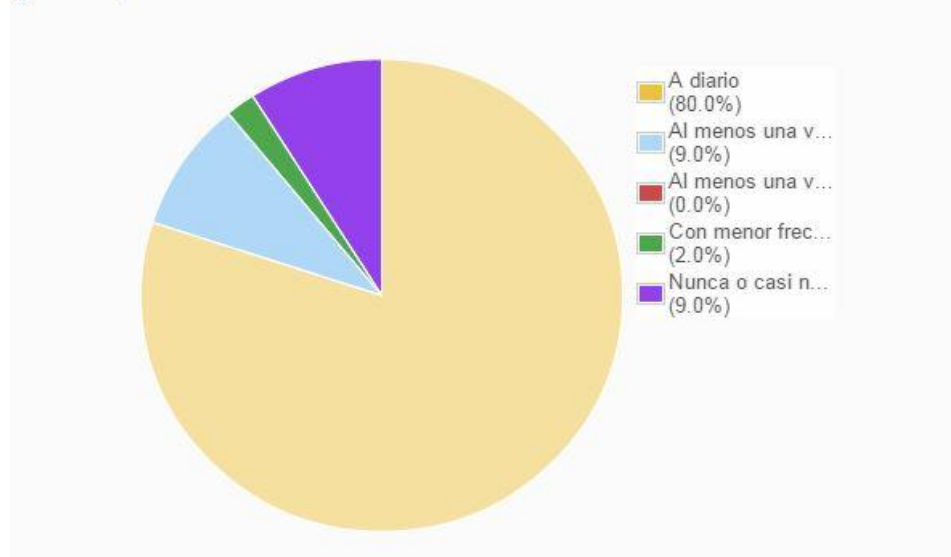


Figura 27. Porcentaje de personas que se conectan a internet por medio de sus celulares.

¿Cuántas aplicaciones tienes instaladas en su celular?

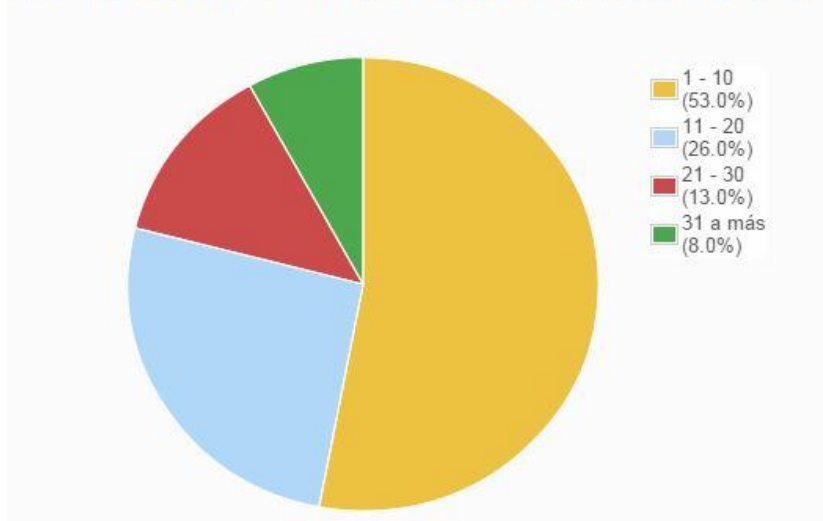


Figura 28. Porcentajes de personas que instalan aplicaciones en su celular.

Si hubiese una aplicación móvil para consultar los eventos que realiza el PAC, ¿Usted la utilizaría?

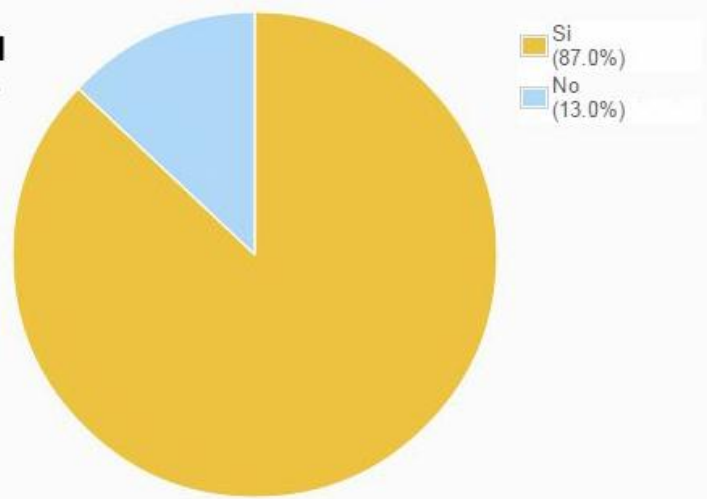


Figura 29. Porcentajes de personas que desearían tener la aplicación móvil del Centro Cultural PAC.

6.2 Herramientas utilizadas para el desarrollo del software de escritorio y aplicación móvil

Las herramientas utilizadas para el desarrollo del software de escritorio Agenda PAC y la aplicación móvil.

- * **Eclipse:** Utilizado como entorno de desarrollo para la aplicación móvil en android.
- * **Android development tools (ADT):** Es un plugin para el IDE Eclipse que amplía las capacidades del eclipse.
- * **Ksoap2:** Utilizo de librería de servicio web en android.
- * **SQL SERVER 2008:** Utilizado para la creación de la base de datos que utiliza el software y la aplicación móvil.
- * **Microsoft Visual Studio 2010:** Utilizado para el desarrollo del software de escritorio para los usuarios del Centro Cultural PAC y para la creación de los diseños de diagrama UML. Desarrollándose también el Webservice en lenguaje C# es el servicio que comunica con el servidor y los dispositivos móviles.
- * **Embarcadero:** Utilizado para el modelado de la base de datos y modelo entidad –relación.
- * **Photoshop:** Utilizado para rediseñar el logo del Centro Cultural PAC y algunas gráficas.
- * **Microsoft office 2010:** Utilizado para la elaboración del documento.

6.3 Descripción de la aplicación móvil

La aplicación móvil creada para el Centro Cultural PAC fue desarrollada para la plataforma Android, entorno de desarrollo Eclipse, creada en la versión 4.2 pero siendo factiblemente ejecutada en versiones 2.3.3 a superior. Esta, a su vez, consume un Webservice hecho en C#, que es la que muestra la información al dispositivo móvil.

El modo en que trabaja la aplicación móvil consiste en mostrar los eventos del mes del Centro PAC. Cuando se ingresa a la aplicación, la primera pantalla que se muestra es la lista de todos los eventos del mes, entre ellos, talleres de pintura (adultos y niños), lectura, arte, poesía, presentación de libros, orquesta, y cuenta cuentos .Al seleccionar

de la lista uno de ellos, se mostrara el listado de los temas que presentaran para determinada sección, una vez seleccionado ya el tema del segundo listado se mostrara el detalle del evento tales como: imagen, descripción, expositor, lugar, precio, hora.

Por lo tanto nuestra aplicación, sin duda alguna, brinda:

- ✓ Más innovación a los servicios que ofrece el centro PAC, pasando así de lo monótono a lo novedoso, ya que su interactivo es rápido, ligero, sencillo pero llamativo.
- ✓ Automatiza la información del Centro, evitando el uso de broshure o correos masivos que no captan la atención inmediata de los visitantes.
- ✓ Aumenta el interés de los visitantes actuales como también la captación de nuevos clientes, que según la encuesta en su mayoría utilizan Smartphone.

6.4 Descripción del software de agenda

El software de agenda registra los tipos de eventos, nombre del evento, los presentadores, hora y fecha, el lugar donde se realiza y tipo de entrada. En este caso de que el evento tenga un precio el sistema permite digitar el valor monetario, y si es un evento que es gratuito y la entrada es libre el sistema permite digitar en formato de texto. También, los eventos tienen una celda donde permite desactivar o activar por tratarse de un evento que ya se presentó en fecha anterior o si es actual.

Tiene la opción de catálogos donde se registra los datos de los presentadores, del lugar donde se realizan los diferentes eventos, los tipos de presentadores, a que empresa trabajan, esto es importante para el personal como consulta.

El software de agenda genera un reporte por fecha.

Hacemos hincapié que el objetivo del desarrollo de este software, fue porque se vio la necesidad de que el Centro Cultural PAC tuviese una herramienta para registrar en la base de datos por medio de una interfaz de usuario amigable, y para hacer funcionar la aplicación móvil y así dar información rápida y personalizada a los clientes.

Beneficios:

- ✓ La flexibilidad al manejar gran volumen y diversidad de información con rapidez, oportuna y precisa, lo que ofrece una mejor herramienta de trabajo al personal, que facilitará sus labores.
- ✓ Capacidad de registrar y almacenar datos de los eventos en la base de datos.
- ✓ Reducir costo de papelería.

6.5 Estudio de factibilidad

6.5.1 Factibilidad Técnica:

En la factibilidad técnica consistió en evaluar la tecnología existente en el Centro Cultural PAC, tanto como hardware y software y se determinó que los equipos cumplen los requerimientos para el desarrollo e implementación para las dos aplicaciones, tales como:

Dos servidores:

- Servidor Web en la nube
- Servidor de Base de Datos.

Ahora bien, hacemos una breve reseña de los equipos que técnicamente aplicamos en el desarrollo del proyecto:

Equipo	Características	Cantidad
Computadora de escritorio(Servidor de base de datos - Hispamer)	Procesador: XEON 2.7 GHz 1.26 GHz	1
	Memoria RAM de 8 GB	
	Capacidad Disco Duro 230 GB	
Servidor web-(HOSTGATOR)	Espacio en disco y ancho de banda ilimitado	1
Pc Intel Pentium	Procesador: Pentium Dual Core CPU E5800@ 3.20GHz 3.19 GHz	1
	Memoria RAM de 4 GB	
	Capacidad Disco Duro 500 GB	
Laptop COMPAQ Presagio CQ43_416LA Notebook	Procesador: Intel core i-3-380M	2
	Memoria RAM de 4 GB	
	Capacidad Disco Duro 500 GB	
Smartphone Samsung Galaxy Y Duos S6102	Tipo TFT touchscreen capacitivo, 256K colores	1
	Tamaño 240 x 320 pixels, 3.14 pulgadas	
	Soporte multitouch	
	OS: Android OS, v2.3 Gingerbread	
Celular Huaewi Ascend Y200	Tipo TFT LCD IPS, 260K colores	1
	Tamaño 240 x 320 pixels, 3.14 pulgadas	
	OS: Android OS, 2.3.6 Gingerbread	
Thomsom (Modem)	Modelo Tg585v7	1

Tabla 1. Recursos de hardware que se utilizó para la aplicación móvil y el software de agenda.

6.5.2 Factibilidad Económica:

Presentamos los costos de los equipos técnicos para el desarrollo y prueba de la aplicación móvil y software de escritorio.

Equipos y materiales			
Servidor de base de datos	1	\$ 900.00	\$ 900.00
Servidor web	1	\$ 402.00	\$ 402.00
Laptop COMPAQ	2	\$ 550.00	\$ 1,100.00
Pc Intel	1	\$ 480.00	\$ 480.00
Samsung Galaxy Y Dúos S6102	1	\$ 160.00	\$ 160.00
Thomsom	1	\$ 25.00	\$ 25.00
Huaewi Ascend Y200	1	\$ 200.00	\$ 200.00
Libro: Android 4 (Desarrollo de aplicaciones)	1	\$ 57.73	\$ 57.73
Recursos humanos			
Pago desarrolladores	3	\$ 800.00	\$ 2,400.00
Software			
Eclipse	3	sin costo	\$ -
Visual Studio 2010	1	\$ 2,000.00	\$ 2,000.00
ER/Studio 8	1	Sin costo	\$
Photoshop CS5	1	\$ 200	\$ 200.00
Gran Total			\$ 7,927.73

Tabla 2. Costos Totales Aplicación móvil y software de escritorio

6.5.3 Factibilidad Operacional

Comprender una determinación de posibilidad que un nuevo sistema se use como se supone y que sea efectivo, por lo tanto, ¿Operativamente cuál es la factibilidad de la aplicación PAC?

Con la aplicación móvil, nuestro objetivo es automatizar todo el proceso por decirlo así monótono o rustico de publicación de las actividades del Centro Cultural PAC. Operativamente es 100% factible porque agiliza el proceso de captación del cliente.

La necesidad de un software de agenda de escritorio llevo la aceptación de parte de los usuarios, que de manera sencilla y amigable, cubre los requerimientos, expectativas y proporcionando las herramientas para almacenar la información de los eventos que se realizan cada mes, tanto las pantallas como el reporte generada por el sistema.

Basándose en las conversaciones con el personal involucrado se demostró que estos no presentan ninguna oposición, por lo que el sistema es factible operacionalmente

6.6 Base de Datos Normalizada

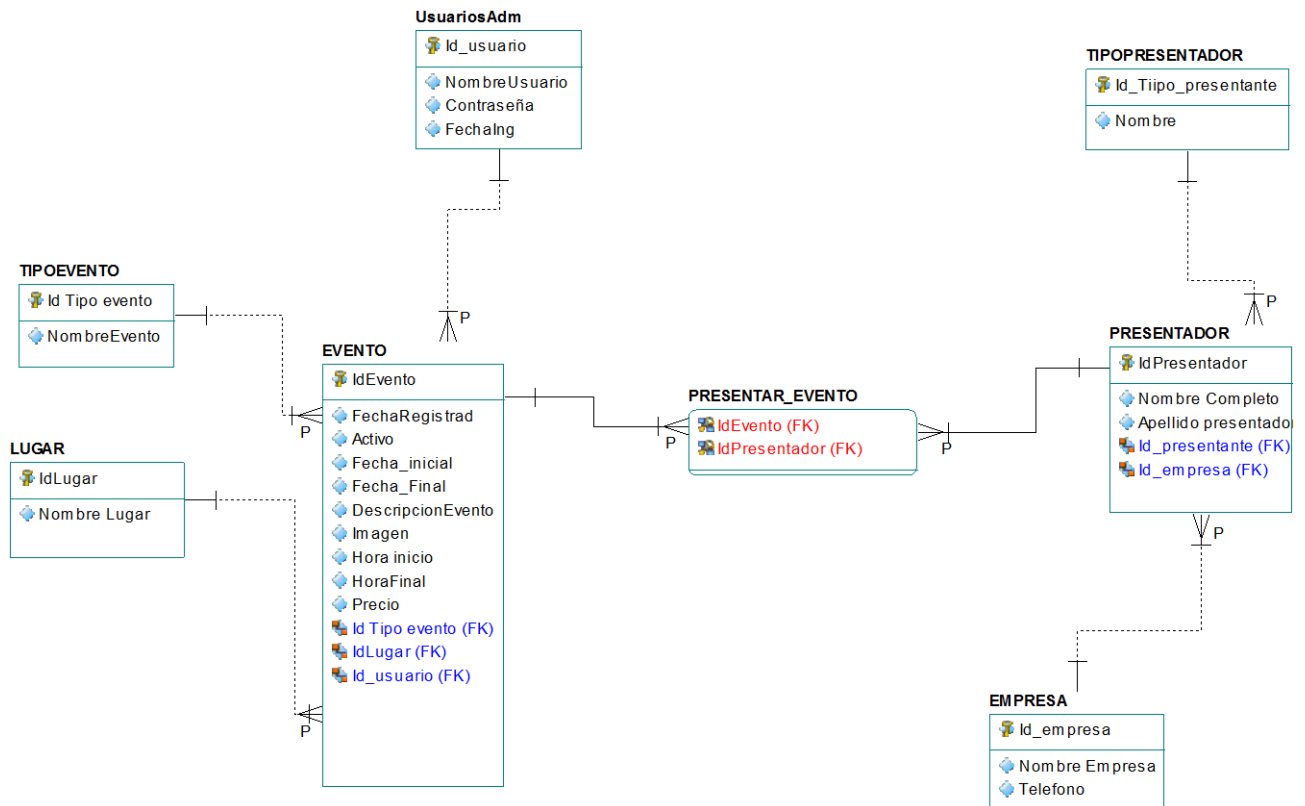


Figura 4030. Base de datos Normalizada

6.7 Diagramas UML

6.7.1 Diagrama de Clase

Fig.

Figura 41. Diagrama de clase

6.7.2 Diagrama de Casos de Uso

a) Caso de uso de la aplicación móvil

■

Figura 42. Caso de uso comprobación de la conectividad a Internet y al webservice.

Caso de uso	Comprobar la conectividad			CU1
Actores	Cliente			
Tipo	Esencial			
Referencias				
Precondición	Tener en su celular Internet y wifi			
Post condición				
Autor	Cientes	Fecha	Versión	1

Propósito
Tener conexión a internet y al Webservice para la consulta de eventos

Resumen
El cliente comprueba la conexión a internet y luego abre la aplicación para conectarse al Webservice

Curso Normal			
1	El cliente empieza a comprobar conexión a internet en su celular		
2	Abre la aplicación en su celular	3	Conectarse al Webservice

Cuadro 1. Especificación y descripción del caso de uso de la comprobación de la conectividad.

Figura 43. Caso de uso Descargar la aplicación

Caso de uso	Descargar la aplicación			CU2
Actores	Cliente			
Tipo	Esencial			
Referencias				
Precondición	Conectividad en su celular Internet y wifi			
Pos condición	Con éxito la conectividad			
Autor	Cientes	Fecha	Versión	1

Propósito
Tener conexión y descargar la aplicación

Resumen
El cliente visita la página web para bajar la aplicación

Curso Normal			
1	El cliente empieza a comprobar conexión a internet en su celular		
2	Visita la página Web	3	Descargar la aplicación "Agenda"

Cuadro 1. Especificación del caso de uso de la descarga de la aplicación móvil

Figura 44. Caso de uso consulta de la agenda en la aplicación móvil.

Caso de uso	Consulta aplicación			CU3
Actores	Cliente			
Tipo	Esencial			
Referencias				
Precondición	Tener conexión a Internet y Webservice			
Post condición	Éxito de conexión			
Autor	Clientes	Fecha	Versión	1

Propósito

Desplegar la consulta de la agenda

Resumen

Al conectarse al Internet y al Webservice se verá la consulta de los diferentes tipos de eventos y al dar clic ver el detalle del evento

Curso Normal

1	Abrir la consulta		
2	Desplegar los tipos de eventos	3	Ver la información de cada evento

Cuadro 2. Especificación y descripción para la consulta de la aplicación móvil.

b) Caso de uso del software de agenda PAC.

Figura 45. Diagrama de uso al sistema de escritorio Agenda PAC.

Caso de uso	Ingreso al sistema			CU4
Actores	Administrador de agenda			
Tipo	Esencial			
Referencias				
Precondición	En el sistema tiene que estar registrado el administrador			
Post condición	El administrador entro al sistema con éxito			
Autor	Administrador de agenda	Fecha	Versión	1

Propósito
Para que el administrador entre con su usuario

Resumen
El usuario va ingresar al sistema validando su usuario y contraseña

Curso Normal			
1	Abrir el sistema		
2	El administrador de agenda ingresa su usuario y contraseña	3	El sistema comprueba la validez de los datos, si no son los correctos se avisa al actor para volver intentar al ingreso.

Cuadro 3. Descripción del ingreso del Sistema Agenda PAC.

8

Figura 46. Diagrama de uso Administrar el evento.

Caso de uso	Administrar el evento			CU5
Actores	Administrador de agenda			
Tipo	Esencial			
Referencias	Esencial			
Precondición	El director del Centro PAC recopilara los eventos a realizarse en el mes			
Post condición	Los eventos quedara registrado en el sistema			
Autor	Administrador de agenda	Fecha	Versión	1

Propósito
Registrar los eventos en el sistema

Resumen
El administrador agregara un nuevo evento, también podrá eliminar y modificar

Curso Normal			
1	Abrir el sistema		
2	El administrador de agenda ingresa su usuario y contraseña	3	El sistema comprueba la validez de los datos, si no son los correctos se avisa al actor para que los corrija
4	Agregar un nuevo evento	5	Busca la existencia en la base de datos
6	el usuario al dar clic a la opción guardar	7	el sistema registra el evento
8	Si el usuario se equivoco puede dar clic a modificar	9	El sistema hace la búsqueda en la base de datos
10	EL administrador da clic en grabar	11	Almacenado con éxito
12	Si hubiese que eliminar el evento el administrador de agenda dará clic en eliminar	13	El sistema hace la búsqueda en la base de datos

Cuadro 5. Especificación y Descripción en administrar el Evento.

8

Figura 47. Diagrama de uso de gestionar el tipo de evento.

Caso de uso	Gestionar el tipo evento			CU6
Actores	Administrador de agenda			
Tipo	Esencial			
Referencias				
Precondición	El tipo de evento se tiene que registrar			
Post condición				
Autor	Administrador de agenda	Fecha	Versión	1
Propósito				
Es clasificar los diferentes tipos de eventos que existen en el Centro Cultural PAC, si hay un nuevo tipo evento se almacenara en la base de datos.				
Resumen				
El administrador de agenda ingresa un nuevo tipo de evento, también puede eliminar o modificar				
Curso Normal				
1	Abrir el sistema			
2	El administrador de agenda ingresa su usuario y contraseña	3	El sistema comprueba la validez de los datos, si no son los correctos se avisa al actor para que los corrija	
4	Agregar un nuevo tipo de evento	5	El sistema busca la existencia en la base de datos	
6	El usuario elige la opción guardar	7	El sistema registra el evento	
8	Si el usuario se equivoco puede elige la opción modificar	9	El sistema hace la búsqueda en la base de datos	
10	El administrador elige la opción de grabar	11	Datos guardados	
12	En caso de eliminar el evento	13	El sistema hace la búsqueda en la base de datos	
		14	El tipo de evento fue eliminado	
Cursos Alternos:				
Si el tipo de evento existe en la base de datos, el sistema envía un mensaje al usuario y no almacenara esa información				
En caso que exista uno o más eventos de la misma categoría, el sistema almacenara esos datos.				

Cuadro 4. Especificación del gestionar el tipo de evento.

8

Figura 48. Diagrama de uso de gestionar el tipo de evento.

Caso de uso	Gestionar el tipo el lugar			CU7
Actores	Administrador de agenda			
Tipo	Esencial			
Referencias				
Precondición	Registrar en el sistema el lugar donde se realizara el evento			
Post condición				
Autor	Administrador de agenda	Fecha	Versión	1
Propósito				
Tener información en qué lugar se realizó el evento				
Resumen				
El administrador de agenda ingresa el nombre del lugar				
Curso Normal				
1	Abrir el sistema			
2	El administrador de agenda ingresa su usuario y contraseña	3		El sistema comprueba la validez de los datos, si no son los correctos se avisa al actor para que los corrija
4	Agregar el nombre del lugar	5		El sistema busca la existencia en la base de datos
6	El usuario elige la opción guardar	7		El sistema registra
8	Si el usuario se equivoco puede elige la opción modificar	9		El sistema hace la búsqueda en la base de datos
10	El administrador elige la opción de grabar	11		Datos guardados
12	En caso de eliminar	13		El sistema hace la búsqueda en la base de datos
		14		El nombre del lugar fue eliminado

Cuadro 7. Especificación y Descripción de gestionar el lugar donde se presenta el evento.

8

Figura 49. Diagrama de uso del ingreso del presentador.

Caso de uso	Gestionar los datos del presentador				CU8
Actores	Administrador de agenda				
Tipo	Esencial				
Referencias					
Precondición					
Post condición	Registra el tipo de presentador				
Autor	Administrador de agenda	Fecha		Versión	1
Propósito					
Resumen					
El administrador de agenda registra el tipo de presentador para llevar el control.					
Curso Normal					
1	Abrir el sistema				
2	El administrador de agenda ingresa su usuario y contraseña	3		El sistema comprueba la validez de los datos, si no son los correctos se avisa al actor para que los corrija	
4	El administrador de agenda ingresa el tipo de presentador	5		El sistema hace la búsqueda en la base de datos	
6	Elige la opción de guardar	6		Almacenado con éxito	
7	Si hay datos que modificar el usuario elige la opción.	8		El sistema guarda la modificación	
9	Si hay que eliminar elige la opción	10		El sistema manda un mensaje de eliminación	

Cursos alternos:
Si existe el tipo de presentador, el sistema no guarda.

Cuadro 8. Especificación del ingreso del presentador.



Figura 50. Diagrama de uso del ingreso del tipo de presentador

Caso de uso	Gestionar el tipo presentador				CU8
Actores	Administrador de agenda				
Tipo	Esencial				
Referencias					
Precondición					
Post condición	Registra el tipo de presentador				
Autor	Administrador de agenda	Fecha		Versión	1
Propósito					
Resumen					
El administrador de agenda registra el tipo de presentador para llevar el control.					
Curso Normal					
1	Abrir el sistema				
2	El administrador de agenda ingresa su usuario y contraseña		3	El sistema comprueba la validez de los datos, si no son los correctos se avisa al actor para que los corrija	
4	El administrador de agenda ingresa el tipo de presentador		5	El sistema hace la búsqueda en la base de datos	
6	Elige la opción de guardar		6	Almacenado con éxito	
7	Si hay datos que modificar el usuario elige la opción.		8	El sistema guarda la modificación	
9	Si hay que eliminar elige la opción		10	El sistema manda un mensaje de eliminación	

Cursos alternos:
Si existe el tipo de presentador, el sistema no guarda.

Cuadro 9. Especificación del ingreso del tipo de presentador.

16

Figura 51. Diagrama de uso del ingreso de los datos de la empresa que pertenece el presentador.

Caso de uso	Gestionar la empresa			CU9
Actores	Administrador de agenda			
Tipo	Esencial			
Referencias				
Precondición	En caso de que el presentador venga de una empresa.			
Post condición	Registra el nombre de la empresa del presentador			
Autor	Administrador de agenda	Fecha	Versión	1
Propósito				
Resumen				
El administrador de agenda registra el nombre de la empresa del presentador para llevar el control.				
Curso Normal				
1	Abrir el sistema			
2	El administrador de agenda ingresa su usuario y contraseña	3		El sistema comprueba la validez de los datos, si no son los correctos se avisa al actor para que los corrija
4	El administrador de agenda ingresa el nombre de la empresa	5		El sistema hace la búsqueda en la base de datos
6	Elige la opción de guardar	6		Almacenado con éxito
7	Si hay datos que modificar el usuario elige la opción.	8		El sistema guarda la modificación
9	Si hay que eliminar elige la opción	10		El sistema manda un mensaje de eliminación

Cuadro 10. Especificación del ingreso de los datos de la empresa que pertenece el presentador

6.7.3 Diagramas de Secuencias

■

Figura 52. Diagrama de secuencia comprobación de conectividad.

78

Figura 53. Diagrama de secuencia consulta a la agenda móvil PAC.

■

Figura 54. Diagrama de secuencia Conexión al webservice y base de datos.

☰

Figura 55. Diagrama de secuencia consulta de los eventos

Fig. 56

Figura 56. Diagrama de secuencia validar el usuario del software de escritorio

¶

Figura 57. Diagrama de secuencia agregar un nuevo tipo de evento.

Fig.

Figura 58. Diagrama de secuencia agregar un nuevo evento.

59

Figura 59. Diagrama de secuencia agregar un nuevo nombre del lugar

Figura 60.

Figura 60. Diagrama de secuencia agregar un nuevo presentador.

Fig.

Figura 61. Diagrama de secuencia agregar un nuevo tipo de presentador.

¶

Figura 62. Diagrama de secuencia agregar un nuevo nombre de la empresa.

78

Figura 63. Diagrama de secuencia modificar la tabla de eventos.

74

Figura 64. Diagrama de secuencia modificar la tabla lugar

78

Figura 65. Diagrama de secuencia modificar la tabla de empresa.

■

Figura 66. Diagrama de secuencia modificar la tabla presentador.

122

Figura 67. Diagrama de secuencia modificar la tabla tipo de presentador.

Fig. 68

Figura 68. Diagrama de secuencia eliminar la información del evento.

Fig.

Figura 69. Diagrama de secuencia eliminar el tipo de evento.

126

Figura 70. Diagrama de Secuencia eliminar el presentador.

127

Figura 71. Diagrama de secuencia eliminar el lugar.

6.7.4 Diagramas de Actividades

❏

Figura 72.Diagrama de actividades conexión en el móvil.

☺

Figura 73. Diagrama de actividades conexión al Web Service en la móvil.

7

Figura 74. Diagrama de actividad consulta a la agenda en el móvil.



Figura 75. Diagrama de actividad comprobación a la conexión a la base de datos.

4

Figura 76. Diagrama de actividad agregar un nuevo evento

à

Figura 77. Diagrama de actividad agregar un nuevo presentador.

3.

Figura 78. Diagrama de actividades agregar un nuevo evento.

3.

Figura 79. Diagrama de actividades agregar tipo de presentador.

Figura 80. Diagrama de actividades agregar un nuevo lugar.



Figura 81. Diagrama de actividades agregar un nuevo nombre de la empresa.

6.8 Diccionario de datos

Nombre	Definición	Atributo	Definición de los atributos	Tipo de dato
PRESENTADOR	La identidad presentador almacén la información del presentador que realizara el evento.	IdPresentador	El código del presentador	INTEGER
		Nombre Completo	Nombre del presentador	VARCHAR(200)
		Apellido presentador	Apellido del presentador	VARCHAR(200)
		Id_Tiipo_presentante	El código del tipo del presentador	INTEGER
		Id_empresa	El código de la empresa que pertenece el presentador	INTEGER

Nombre	Definición	Atributo	Definición de los atributos	Tipo de dato
EVENTO	La identidad Evento presenta a nivel de detalle la información de los eventos que se realizaran en el Centro Cultural P.A.C.	IdEvento	El código del evento	INTEGER
		FechaRegistrad	La fecha que fue registrado el ingreso del evento	DATETIME
		Activo	Este atributo permite desactivar y activar el evento	BIT
		Fecha_inicial	Es la fecha inicio que se realiza el evento	DATE
		Fecha_Final	Es la fecha final que se realiza el evento	DATE
		DescripcionEvento	Es el nombre del evento	VARCHAR(200)
		Imagen	Es la imagen representativa del evento	IMAGE
		Hora inicio	Es la hora de inicio del evento	TIME(7)
		HoraFinal	Es la hora final del evento	TIME(7)
		Precio	El precio es cuando algún evento tiene valor monetario	VARCHAR(50)
		IdTipoEvento	Es el tipo de evento por ejemplo: Presentación de libro, Concierto, Recital de poesía	INTEGER
		IdLugar	Es la código del lugar donde se realiza el evento	INTEGER
Id_usuario	La identificación del personal del Centro Cultural	VARCHAR(3)		

Nombre	Definición	Atributo	Definición de los atributos	Tipo de dato
LUGAR	Esta identidad mencionara los lugares que tiene el Centro Cultural P.A.C para sus eventos.	IdLugar	Es la código del lugar donde se realiza el evento	INTEGER
		Nombre Lugar	Es el nombre del lugar donde se realizan los eventos	VARCHAR(60)

Nombre	Definición	Atributo	Definición de los atributos	Tipo de dato
TIPOEVENTO	Esta identidad clasifica el tipo de evento	Id Tipo evento	Es el tipo de evento por ejemplo: Presentación de libro, Concierto, Recital de poesía	INTEGER
		NombreEvento	Es el nombre del evento	VARCHAR(100)

Nombre	Definición	Atributo	Definición de los atributos	Tipo de dato
TIPOPRESENTADOR	En esta identidad se almacena el tipo de presentador del evento que puede ser: una empresa, autor, grupo musical, cantante, comediante, etc.	Id_Tiipo_presentante	Es el código del tipo de presentador	INTEGER
		Nombre	Es el nombre del tipo de presentador	VARCHAR(50)

Nombre	Definición de la tabla	Atributo	Definición de los atributos	Tipo de dato
EMPRESA	Esta identidad contiene el nombre de la empresa del presentador	Id_empresa	Es el código de la empresa	INTEGER
		Nombre Empresa	Es el nombre de la empresa	VARCHAR(30)
		Telefono	El número telefónico para contactar a la empresa	VARCHAR(8)

Nombre	Definición de la tabla	Atributo	Definición de los atributos	Tipo de dato
UsuariosAdm	Almacena el usuario con su contraseña	Id_usuario	El código de identificación para el personal del Centro PAC	VARCHAR(3)
		NombreUsuario	Es el nombre del usuario	TEXT
		Contraseña	Es la contraseña del usuario	CHAR(7)
		Activo	Si esta activo el usuario o desactivado	BIT
		FechaIng	Es la fecha de ingreso del usuario	DATE

6.9 Pantallas de aplicativo móvil

1) Icono en el celular

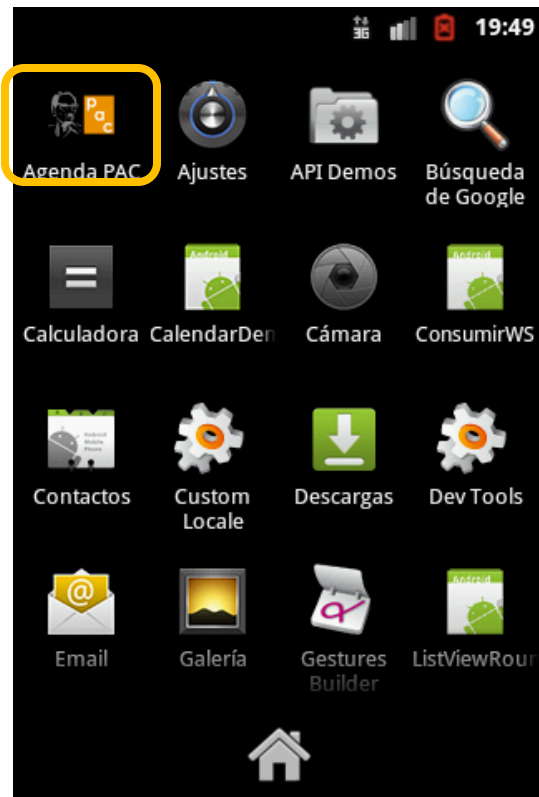


Figura 82. Icono de la aplicación móvil.

2) **Menú de eventos**

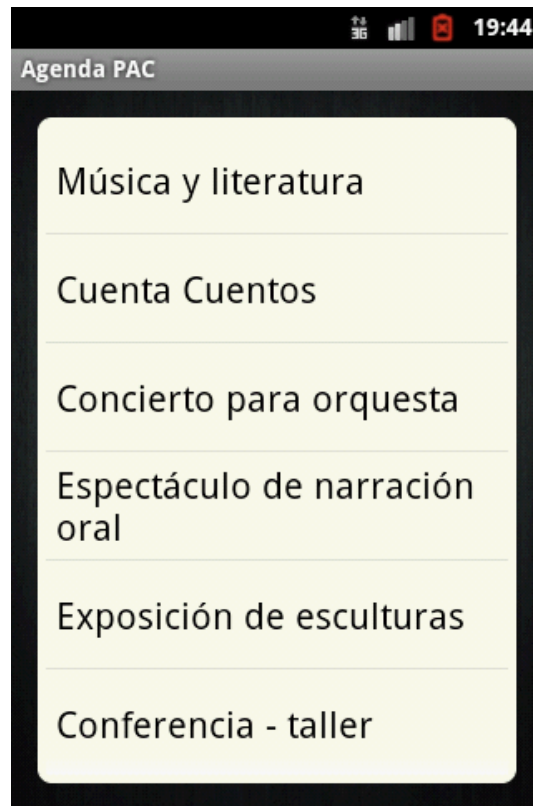


Figura 83. Menú de eventos.

3) Tipos de eventos seleccionados del menú eventos

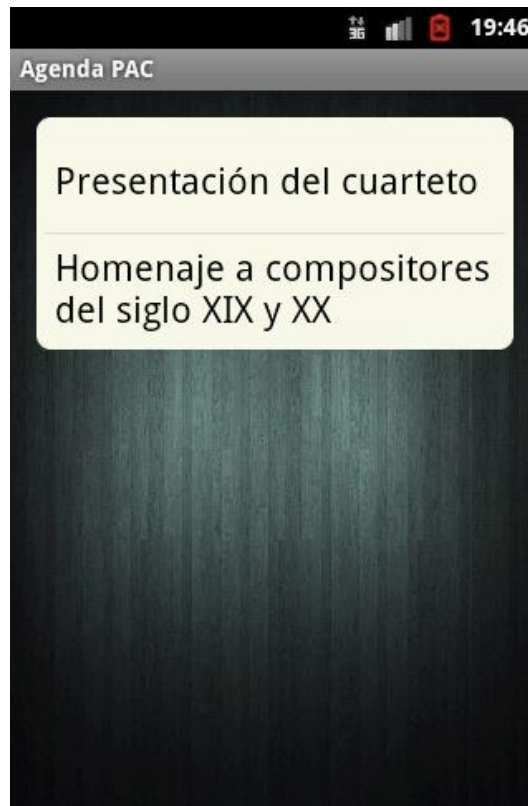


Figura 84. Lista de eventos seleccionados del menú de eventos.

4) **Detalle del evento seleccionado**

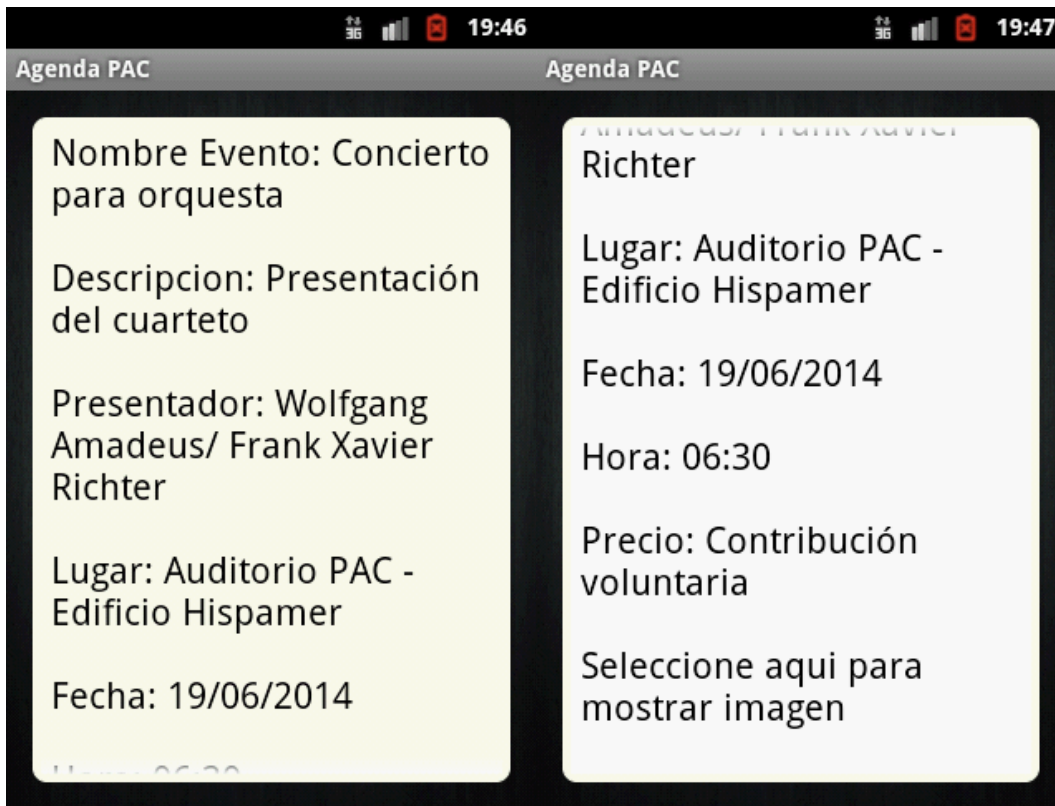


Figura 85. Detalle del evento
(parte1 y parte2).

5) Imagen del evento seleccionado



Figura 86. Imagen del evento seleccionado.

6) **Evento seleccionado del menú eventos, sin registros**



Figura 87. Sin eventos a mostrar.

7) **Menú de Agenda PAC**



Figura 88. Menú Agenda PAC.

8) **Seleccionando el botón Acerca del Centro PAC**

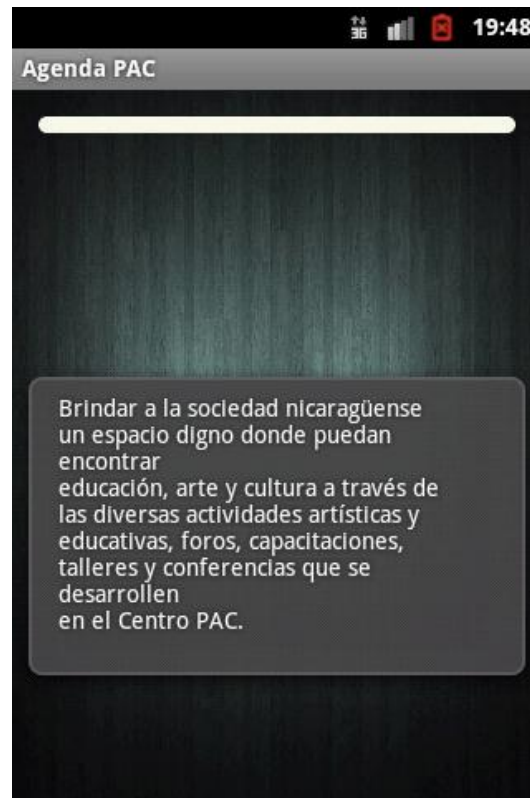


Figura 89. Acerca PAC.

9) **Seleccionando el botón Acerca App**



Figura 90. Acerca PAC.

6.10 Pantallas de software de para administrar información de aplicativo

1) Iniciar sesión

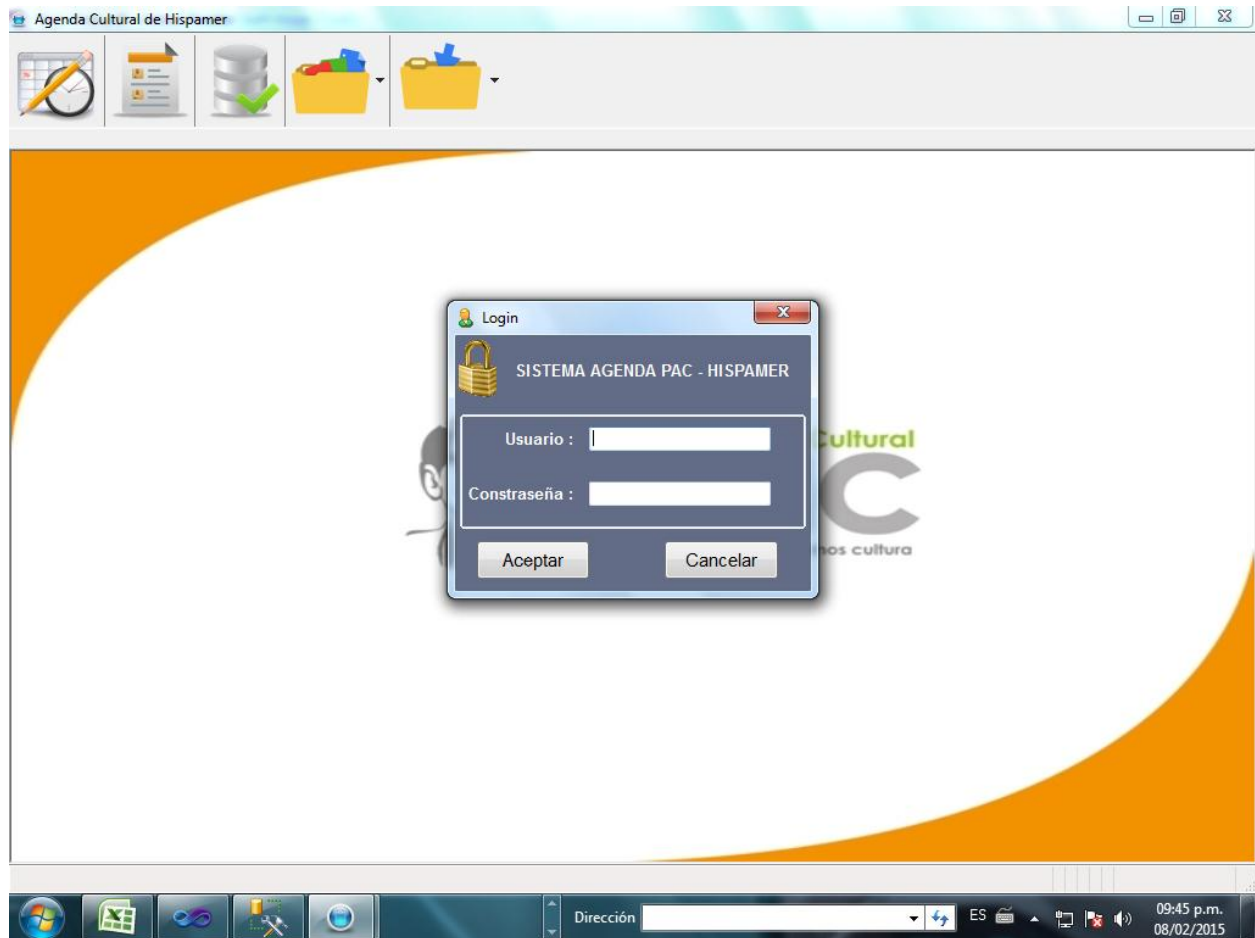


Figura 91. Pantalla de inicio de sesión.

2) Pantalla principal

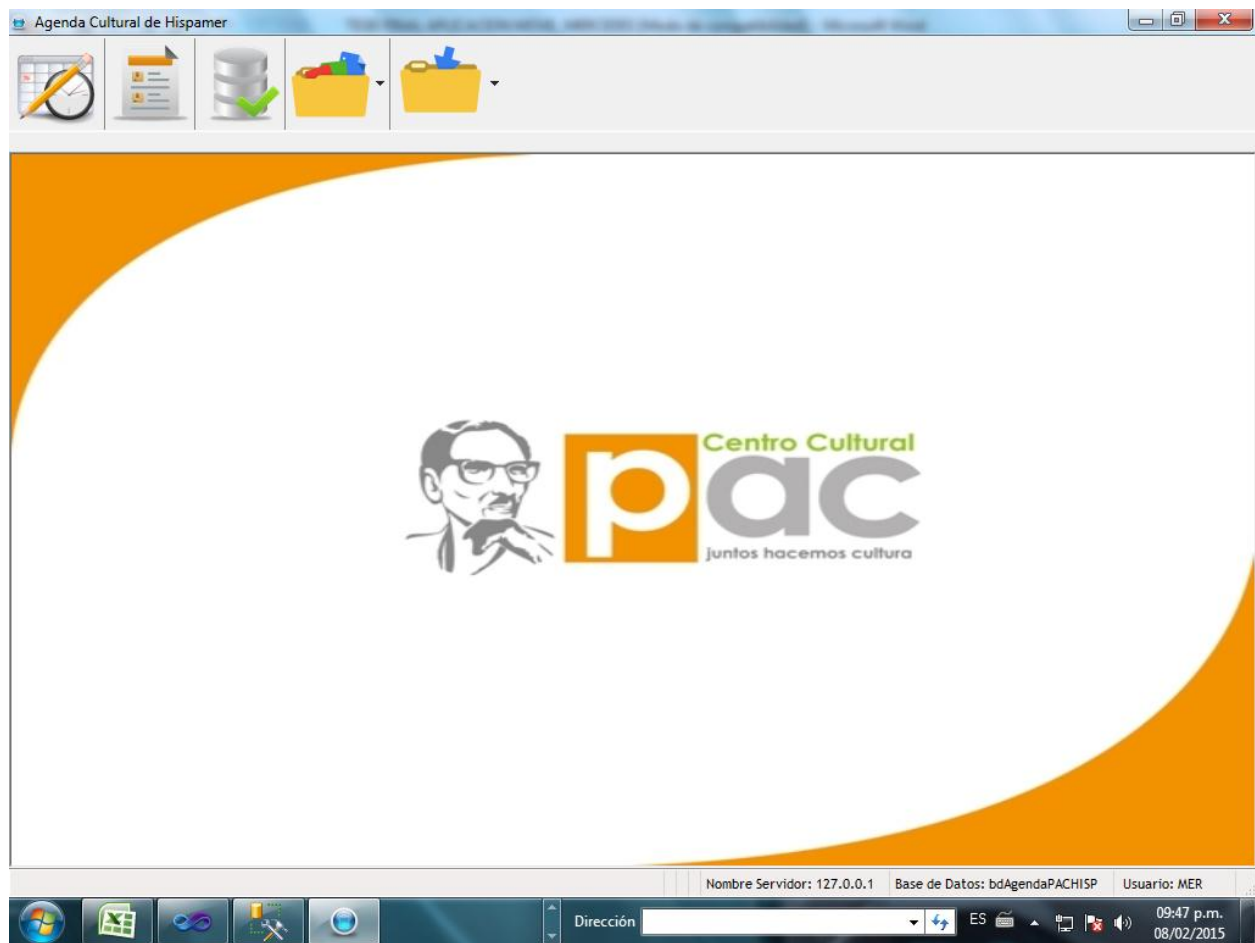


Figura 92. Pantalla principal de software.

3) Formulario principal

Nombre Del Evento	Evento
Música y Literatura	El temblor mágico por el escritor George de Vigne
Presentación de libros	"Miradas de Solentiname: fotografías y reflexiones"
Presentación de libros	100 años de diversidad sexual en Nicaragua
Concierto para orquesta	Presentación del cuarteto
Dibujo y Pintura	Para los adolescentes y adultos
Cuenta Cuentos	Ven a disfrutar cuentos infantiles
Concierto para orquesta	Homenaje a compositores del siglo XIX y XX
Espectáculo de narración oral	Ruta del cuento y la cultura de Managua
Exposición de esculturas	Pinturas del pintor Ernesto Cardenal
Conferencia - taller	Nicaragua Lee, Encuentro internacional para el desarrollo de la lectura presentada por la Lic. María Elena de Farmacia
Clases de yoga	Para adolescentes y adultos
Recital	Juntos y revueltos
Espectáculo de títeres	Barco a la vista, una historia de piratas
Bebe Artista	Niños de menores de 3 años podrán experimentar a través de técnicas secas y acuosas, lápices de colores y plastilina
Clases de yoga	Yoga para padres e hijos
Conferencia	Nuevos enfoques sobre China : La revolución silenciosa
Concierto	Katia Cardenal presenta un concierto "Casa Abierta"

Figura 93. Lista de eventos.

4) Registra la información del evento.

Eventos

Centro Cultural
pac

Estado
 Activo

Insertar Imagen

Tipo de Evento :

Evento :

Autor/
Institución:

Fecha Inicio :

Fecha Final :

Hora Inicio :

Hora Final :

Lugar :

Entrada :

Nombre	Apellido
Victoria	Gonzalez Rivera
Victoria Gonzalez	Karen
Roger	Pérez de la Rocha
Karen	Kampwirth
Conversatorio	Upoli
Tiago	Genoveze
Ruth	Saézn Puerto Rico

Usuario : MER

Figura 94. Pantalla de creación de eventos.

5) Edición evento

Eventos

Centro Cultural
pac

Estado
 Activo

Insertar Imagen

Tipo de Evento : Fiesta Infantil

Evento : Agüizote

Autor/
Institución: (86) Grupo Masaya

Nombre	Apellido
Victoria	Gonzalez Rivera
Victoria Gonzalez	Karen
Roger	Pérez de la Rocha
Karen	Kampwirth
Conversatorio	Upoli
Tiago	Genoveze
Ruth	Saézn Puerto Rico

Fecha Inicio : 25/10/2014

Fecha Final : 08/02/2015

Hora Inicio : 03:00 p.m.

Hora Final : 05:00 p.m.

Lugar : Área Infantil-Edificio Hispamer

Entrada : Libre y Gratuita

Usuario : MER

Figura 95. Pantalla de edición de evento.

6) Catálogos

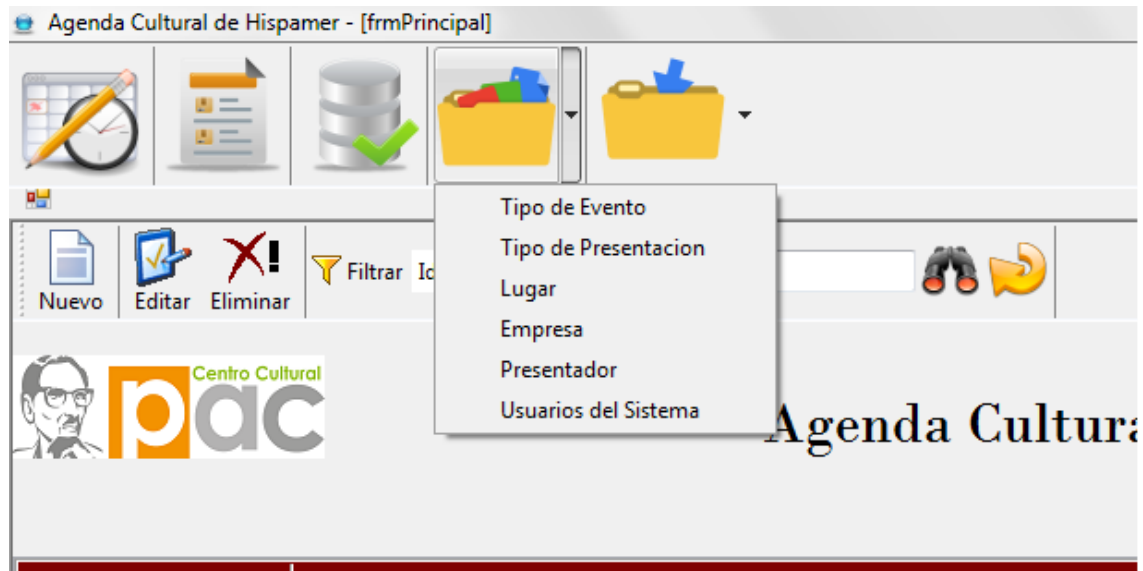


Figura 96. Lista de catálogos del software.

7) Generación de reportes



Figura 97. Generador de reporte de Agenda PAC.

8) Reporte

imagen	Nombre del Evento	Descripción Evento	Presentador	NombreLugar
	Conferencia	Nuevos enfoques sobre China : La revolución silenciosa	Douglas Salamanca/ Cairo Amado	Auditorio PAC -Edifici
	Concierto	Katia Cardenal presenta un concierto "Casa Abierta"	Katia Cardena	Auditorio PAC -Edifici
	Presentación y conversatorio	Documental "El canto de Bosawas"	Cale FG	Auditorio PAC -Edifici
	Encuentro de cineastas	Conversatorio de sobre el presente del cine en Nicaragua	ICARO NICARAGUA/ KINO MANAGU	Auditorio PAC -Edifici

Figura 98. Reporte de eventos.

6.11 Pruebas realizadas de la aplicación móvil

Ciclo	Equipo	Observación
1	Samsung Galaxy Y Dúos S6102	No se mostró ningún inconveniente
2	Samsung Galaxy Y Dúos S6102	No se mostró ningún inconveniente
	Huaewi Ascend Y200	Se mostró inconvenientes en sección lista de eventos
3	Coby kyros mid7120 4g (Tablet)	No se mostró ningún inconveniente, en este ciclo se superaron los inconvenientes
	Huaewi Ascend Y200	
4	Varios equipos	Esta etapa se probó en el centro con varios usuarios y no se mostró inconvenientes

VII CONCLUSIONES

- Se evaluaron las diferentes tecnologías y herramientas para móviles, considerándose Android, con el desarrollo de entorno eclipse, que es la que más se adapta a lo que requerían los clientes.
- Se determinaron las condiciones tecnológicas del centro **PAC**, encontrándose que dicho centro no ha tenido la dicha de aprovechar las oportunidades que las aplicaciones móviles pueden aportar a las empresas, por lo cual se desarrolló la aplicación móvil "Agenda PAC".
- Se diseñaron interfaces que permitan una fácil y rápida navegación a través de la aplicación a cada uno de los temas contenidos en la agenda móvil: presentación de libros, conferencias, exposiciones de arte, talleres, entre otros.
- Se elaboró un mecanismo de control que permita actualizar las calendarizaciones de eventos (lugar, hora, fecha) por parte de los usuarios internos que administran la aplicación.

VIII RECOMENDACIONES

- Realizar una mayor ampliación de la aplicación móvil "Agenda PAC" en base a nuevos requerimientos de la empresa, de forma que enriquezcan la versión actual y provean un buen elemento de promoción y marketing al negocio.
- Desarrollar la aplicación en diferentes plataformas y a si alcanzar los usuarios que cuentan con otro tipo de sistema operativo móvil.

IX BIBLIOGRAFIA

- Casati, F. (2004).** *Web Services*. Canada: ETH Zentrum.
- DRAKE, J. (2013).** *Seminario UM: Diagramas de Estados*. Unican.
- Gual, L. J. (1985).** *Biblioteca practica de la computacion*. Mexico: Oceano.
- kendall, k. y. (2006).** *Analisis y Diseño de Sistemas*. Mexico: Pearson .
- Lee, W.-M. (2012).** *Android 4 Desarrollo de Aplicaciones*. Mexico: ANAYA.
- Marquez, M. P. (2008).** *Bases de Datos y Administracion*. Madrid: San fernando de Hanares.
- Meza, J. A. (2013).** *SQL. Tutorial de SQL usando SQL*, 1-25.
- O´brien, J. A. (2006).** *Sistemas Operativos*. Mexico: Mexico.DF.
- Osorio, F. L. (2008).** *Bases de Datos Relacionales*. Colombia: Fondo Editorial ITM.
- Pedro Garcia Sevilla, F. M. (2000).** *Introduccion a la Programacion*. España: Pearson.
- Scott, K. (1999).** *UML Gota a Gota*. Mexico: pearson.
- Sommerville, I. (2005).** *Ingenieria del Software*. Madrid: Pearson Madrid.
- Soriano, J. E. (2012).** *Programacion en Android*. España: Marcombo S.A.


BIBLIOGRAFÍA CONSULTADA EN INTERNET.


- Alegsa, L. (11 de 10 de 2010).** *Alegsa.com.ar*. Recuperado el 25 de 09 de 2014, de www.alega.com.ar
- Genbeta. (05 de 11 de 2013).** *Genbetadev*. Recuperado el 19 de 07 de 2014, de [Genbetadev: www.genbetadev.com](http://www.genbetadev.com)
- Velez, J. (23 de julio de 2014).** *Alto Nivel*. Obtenido de Alto Nivel: www.altonivel.com
- Somasegar. (01 de 09 de 2013).** *Somasegar blogs*. Recuperado el 11 de 08 de 2014, de Somasegar blogs:
- Project, O. D. (24 de 11 de 2013).** *Wikipedia.org*. Recuperado el 12 de 05 de 2014, de Wikipedia.org:

X ANEXOS

10.1 Formato de Encuesta a través de Sitio Web Encuestas.Com


1. En una escala del 1 al 6, donde 6 es "muy interesante" y 1 es "nada interesante".							Gráficos
6	5	4	3	2	1	Media aritmética	
¿Cómo de interesante es una aplicación móvil de centro PAC-Hispamer para usted?	46% (46)	19% (19)	17% (17)	12% (12)	3% (3)	3% (3)	4,84
Total Respondentes						100	
Filtros aplicados						0	



2. ¿Cuál es el modelo de su celular?				Gráficos
Opción	Distribución	Porcentaje	Total	
Iphone		20%	20	
Blackberry		8%	8	
Samsung		38%	38	
Otro(por favor, especifique) 		34%	34	
Total Respondentes			100	
Filtros aplicados			0	


3. ¿Qué sistema operativo utiliza en su celular?			
Gráficos			
Opción	Distribución	Porcentaje	Total
Windows Phone		18%	18
Android		62%	62
Symbian		0%	0
Otro(por favor, especifique) 		20%	20
Total Respondentes			100
Filtros aplicados			0


4. ¿Con qué frecuencia se conecta a Internet a través del móvil?			
Gráficos			
Opción	Distribución	Porcentaje	Total
A diario		80%	80
Al menos una vez a la semana		9%	9
Al menos una vez al mes		0%	0
Con menor frecuencia		2%	2
Nunca o casi nunca		9%	9
Total Respondentes			100
Filtros aplicados			0

5. ¿Cuántas aplicaciones tienes instaladas en su celular? Gráficos			
Opción	Distribución	Porcentaje	Total
1 - 10		53%	53
11 - 20		26%	26
21 - 30		13%	13
31 a más		8%	8
Total Respondentes			100
Filtros aplicados			0

6. Si hubiese una aplicación móvil para consultar los eventos que realiza cada mes el Centro Cultural Pac - Hispamer, ¿Usted la utilizaría? Gráficos			
Opción	Distribución	Porcentaje	Total
Si		87%	87
No, diga porque 		13%	13
Total Respondentes			100
Filtros aplicados			0

7. ¿Considera usted que la aplicación móvil dirigida hacia el Centro Cultura PAC - Hispamer, seria de útil progreso? Explique su respuesta.			
Opción	Distribución	Porcentaje	Total
 Si		92,05%	81
 No		9,09%	8
Total Respondentes			88
Filtros aplicados			0

8. Pensando en la consulta de los eventos que realiza el Centro Cultural PAC-Hispamer ¿qué requerimiento le gustaría que contenga la aplicación? Gráficos			
Opción	Distribución	Porcentaje	Total
Imagen		63%	63
Descripción del evento		91%	91
Fecha y hora		86%	86
Nombre del Presentador		69%	69
Costo de los talleres		75%	75
(Otro) Especifique 		13%	13
Total Respondentes			100
Filtros aplicados			0

9. Hablando de detalles en diseño en la pantalla ¿Qué sugerencias nos daría?	
	Total
 Hablando de detalles en diseño en la pantalla ¿Qué sugerencias nos daría?	
Total Respondentes	95
Filtros aplicados	0

10. ¿Es usted? Gráficos			
Opción	Distribución	Porcentaje	Total
Mujer		53%	53
Hombre		47%	47
Total Respondentes			100
Filtros aplicados			0

11. ¿Qué edad tiene usted?			Gráficos
Opción	Distribución	Porcentaje	Total
12 - 20		6%	6
21 - 30		26%	26
31 - 40		17%	17
40 a Más		51%	51
		Total Respondentes	100
		Filtros aplicados	0

10.1 La Encuesta

Diseño de Encuesta

ENCUESTA A LOS CLIENTES:

“Aplicación móvil para la agenda de actividades en el Centro Cultural Pablo Antonio Cuadra- Hispamer”

1. ¿Cuál es la marca de su celular?

- Iphone
- Blackberry
- Android

Otro: _____

2. ¿Qué sistema operativo utiliza su celular?

- Windows Phone
- Android
- SYMBIAN

Otro: _____

3. ¿Con qué frecuencia se conecta a internet a través del móvil?

- A diario
- Al menos una vez a la semana
- Al menos una vez al mes
- Con menor frecuencia
- Nunca o casi nunca

4. ¿Cuántas aplicaciones tienes instaladas en su celular?

5. ¿Con qué frecuencia accede a los servicios web (Redes sociales: Facebook, Twitter, pagina web) que proporciona el Centro Cultural PAC-Hispamer?

- A diario
- Al menos una vez a la semana
- Al menos una vez al mes
- Con menor frecuencia
- Nunca o casi nunca

6. Si hubiese una aplicación móvil para consultar los eventos que realiza cada mes el Centro Cultural PAC -Hispamer, ¿Usted la utilizaría?

Sí

No

7. ¿Considera usted que una aplicación móvil dirigida hacia el Centro Cultural PAC - Hispamer, sería de útil progreso?

Sí

No

¿Porque?: _____

8. Pensando en la consulta de los eventos que realiza el Centro Cultural PAC-Hispamer ¿Qué requerimiento le gustaría que contenga la aplicación?

Imagen

Costo de los talleres

Descripción del evento

Fecha y hora

Nombre del Presentador

Otro: _____

Explique _____

9. Hablando de detalles en diseño en la pantalla ¿Qué sugerencias nos daría?

Características del usuario

Es para de determinar el perfil del usuario.

10. ¿Es usted?

Hombre

Mujer

11. ¿Qué edad tiene usted?

Muchas gracias por tu colaboración.