



UNIVERSIDAD TÉCNICA DEL NORTE

**FACULTAD DE INGENIERIA EN CIENCIAS
APLICADAS**

**ESCUELA DE INGENIERIA EN SISTEMAS
COMPUTACIONALES.**

TEMA: Sistema web para la gestión del fútbol barrial en la ciudad de Ibarra utilizando herramientas libres.

AUTOR: JOFFRE P. AGUAS R.

DIRECTOR: MSC. ING. RODRÍGO NARANJO

IBARRA – ECUADOR

2012

CERTIFICACIÓN

El señor JOFFRE PAOLO AGUAS RODRÍGUEZ portador del número de cédula 1003212360 ha trabajado en el desarrollo del proyecto de tesis “Sistema web para la gestión del fútbol barrial en la ciudad de Ibarra utilizando herramientas libres”, previo a la obtención del título de Ingeniero en Sistemas Computacionales, trabajo que lo realizo con interés profesional y responsabilidad, es todo cuanto puedo decir en honor a la verdad.

Msc. Ing. Rodrigo Naranjo

DIRECTOR DE TESIS

CERTIFICACIÓN

Ibarra, 14 de Mayo de 2012

Señores

UNIVERSIDAD TÉCNICA DEL NORTE

Presente

De mis consideraciones.-

Siendo auspiciantes del proyecto de tesis del Egresado JOFFRE PAOLO AGUAS RODRÍGUEZ con CI: 1003212360 quien desarrolló su trabajo con el tema “Sistema web para la gestión del fútbol barrial en la ciudad de Ibarra utilizando herramientas libres, me es grato informar que se han superado con satisfacción las pruebas técnicas y la revisión de cumplimiento de los requerimientos funcionales, por lo que se recibe el proyecto como culminado y realizado por parte del egresado JOFFRE PAOLO AGUAS RODRÍGUEZ. Una vez que hemos recibido la capacitación y documentación respectiva, nos comprometemos a continuar utilizando el mencionado aplicativo en beneficio de nuestra empresa/institución.

El egresado JOFFRE PAOLO AGUAS RODRÍGUEZ puede hacer uso de este documento para los fines pertinentes en la Universidad Técnica del Norte.

Atentamente,

Lc. Fernando Checa
Presidente
FEDELIGAS IMBABURA



UNIVERSIDAD TÉCNICA DEL NORTE

CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, JOFFRE PAOLO AGUAS RODRÍGUEZ, con cédula de identidad Nro. 1003212360, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6 en calidad de autor de la obra o trabajo de grado denominado: **SISTEMA WEB PARA LA GESTIÓN DEL FÚTBOL BARRIAL EN LA CIUDAD DE IBARRA UTILIZANDO HERRAMIENTAS LIBRES** que ha sido desarrollado para optar por el título de INGENIERO EN SISTEMAS COMPUTACIONALES en la Universidad Técnica del Norte, quedando la universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

Firma:

Nombre: JOFFRE PAOLO AGUAS RODRÍGUEZ.

Cédula: 1003212360.

Ibarra, a los 8 días del mes de Mayo de 2012.



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital institucional determina la necesidad de disponer los textos completos de forma digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual ponemos a disposición la siguiente investigación:

DATOS DE CONTACTO	
CEDULA DE IDENTIDAD	1003212360
APELLIDOS Y NOMBRES	AGUAS RODRÍGUEZ JOFFRE PAOLO
DIRECCIÓN	EL CHOTA
EMAIL	golazoaguitas@hotmail.com
TELÉFONO FIJO	062562045
TELÉFONO MOVIL	091890105

DATOS DE LA OBRA	
TITULO	“SISTEMA WEB PARA LA GESTIÓN DEL FÚTBOL BARRIAL EN LA CIUDAD DE IBARRA UTILIZANDO HERRAMIENTAS LIBRES.”
AUTOR	AGUAS RODRÍGUEZ JOFFRE PAOLO
FECHA	14 DE MAYO DEL 2012
PROGRAMA	PREGRADO
TITULO POR EL QUE	INGENIERÍA EN SISTEMAS COMPUTACIONALES
DIRECTOR	MSC. ING. RODRÍGO NARANJO

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, AGUAS RODRÍGUEZ JOFFRE PAOLO, con cedula de identidad Nro. 1003212360, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad

Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y el uso del archivo digital en la biblioteca de la universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión, en concordancia con la Ley de Educación Superior Artículo 143.

.....

Firma

Nombre: AGUAS RODRÍGUEZ JOFFRE PAOLO

Cédula: 1003212360

Ibarra a los 14 días del mes de abril del 2012

DEDICATORIA

Este trabajo realizado con dedicación va dedicado con amor y cariño a mis padres Joel y Laura por el apoyo que me brindaron desde el inicio de mi carrera, a mis hijos Jisdany y Haitana que han sido los que me motivan y dan fuerzas para superarme y salir adelante.

También a Cristina quien me ha brindado su apoyo en esos momentos duros de mi vida, por todos ellos que aportaron de una u otra manera a culminar con éxito una etapa más de aprendizaje.

JOFFRE AGUAS

AGRADECIMIENTO

En primer lugar a **Dios** Creador de todo lo existente y guía de mi vida, quien con su voluntad ha permitido que llegue a la meta que un día me propuse. Que me da la oportunidad de seguir creciendo mentalmente, y poner siempre a las personas indicadas en el transcurrir de mi vida.

A **Mis padres** Joel Aguas, quien me ha brindado todos sus conocimientos desde los inicios de mi vida y lo más importante, el ejemplo de llevar una vida digna de ser un hombre a admirar; padre, estaré siguiendo siempre tus pasos; Laura Rodríguez, quien me ha dado su cariño, atenciones, recuerdos y alegrías desde mi niñez y por estar siempre pendiente de mí, a ambos por el apoyo incondicional que me dieron a lo largo de la carrera y a lo largo de mi vida.

A **Mis hermanos** Por sus consejos y apoyo y por los buenos tiempos que hemos vivido, que siempre estarán en mis pensamientos.

A **Mi familia** Porque siempre me han apoyado, aconsejado y brindado todo el cariño que ha sido fundamental en mi vida; familia, este logro es de todos.

A **Mi asesor** Ingeniero Rodrigo Naranjo, por su excelente asesoría y dirección en mi trabajo de investigación.

A **Mis amigos** Que sin duda alguna, sus consejos, experiencias y sobre todo, su apoyo y paciencia, contribuyeron en todos mis éxitos.

Como olvidarme de la institución que me abrió las puertas a la educación superior la **UNIVERSIDAD TÉCNICA DEL NORTE** y a los grandes profesionales que se encuentran en las aulas de la escuela en ingeniería en sistemas impartiendo sus conocimientos.

En general A todas aquellas personas que de una u otra forma, colaboraron o participaron en mi formación como persona y profesional, hago extensivo mi más sincero agradecimiento.

JOFFRE AGUAS

ÍNDICE DE CONTENIDOS

ÍNDICE DE CONTENIDOS.....	VI
ÍNDICE DE TABLAS.....	XII
ÍNDICE DE GRÁFICOS.....	XIV
RESUMEN.....	XVIII
SUMARY.....	XIX
1. Capítulo I. INTRODUCCIÓN	
1.1 PROBLEMA.....	1
1.2 SITUACIÓN ACTUAL.....	2
1.3 PROSPECTIVA.....	2
1.4 FORMULACIÓN DEL PROBLEMA.....	2
1.4.1 ¿A QUIEN AFECTA?.....	3
1.4.2 ¿QUE IMPACTO CAUSA?.....	3
1.4.3 ¿SOLUCIÓN EXITOSA?.....	3
1.5 OBJETIVOS.....	4
1.5.1 OBJETIVO GENERAL.....	4
1.5.2 OBJETIVOS ESPECÍFICOS.....	4
1.6 ALCANCE.....	5
1.7 JUSTIFICACIÓN DEL PROYECTO.....	7
2. CAPITULO II. MARCO TEÓRICO	
2.1 HERRAMIENTAS DE SOFTWARE LIBRE.....	9
2.1.1 ¿QUÉ ES EL SOFTWARE LIBRE?.....	9
2.1.2 ¿QUÉ ES UN SERVIDOR WEB?.....	11
2.1.2.1 ¿QUÉ ES APACHE?.....	11
2.1.2.2 ¿QUÉ ES XAMMP?.....	12
2.1.3 PHP.....	18
2.1.4 NETBEANS.....	20

2.2 SISTEMAS WEB	23
2.3 BASES DE DATOS	26
2.3.1 QUE ES POSTGRES.....	26
2.4 ENTREGABLES DEL PROYECTO	38
2.5 ORGANIZACIÓN DEL PROYECTO	42
2.5.1 PARTICIPANTES DEL PROYECTO.....	42
2.6 ADMINISTRACIÓN DE PROCESOS	43
2.6.1 PLAN DE PROYECTO.....	43
2.6.2 PLAN DE FASES.....	43
2.6.2.1 OBJETIVOS DE CADA ITERACIÓN.....	45
2.6.2.2 ENTREGAS.....	45
2.6.2.3 CRONOGRAMA DEL PROYECTO.....	46
2.6.2.4 PLANEANDO FASES.....	48
2.6.3 CALENDARIO DEL PROYECTO.....	51
2.6.4 SEGUIMIENTO Y CONTROL DEL PROYECTO.....	61
2.6.4.1 CONTROL DE PLAZOS.....	61
2.6.4.2 CONTROL DE CALIDAD.....	61
2.6.4.3 GESTIÓN DE RIESGOS.....	61
2.6.4.4 GESTIÓN DE CONFIGURACIÓN.....	61

3. Capítulo III. DISEÑO DE LA SOLUCIÓN

3.1 MODELOS DE CASOS DE USO	62
3.1.1 INTRODUCCIÓN.....	62
3.1.2 PROPÓSITO.....	62
3.1.3 ALCANCE.....	62
3.1.4 PANORAMA GENERAL.....	62
3.1.5 CASO DE USO OBTENER INFORMACIÓN DE EQUIPO.....	63
3.1.6 CASO DE USO OBTENER INFORMACIÓN DE JUGADOR....	64
3.1.7 CASO DE USO OBTENER INFORMACIÓN DE LIGA BARRIAL.....	65
3.1.8 CASO DE USO OBTENER INFORMACIÓN DE TRANSFERENCIA.....	66
3.2 ESPECIFICACIÓN DE CASOS DE USO	67
3.2.1 ESPECIFICACIÓN CASO DE USO ASIGNACIÓN DE USUARIOS AL SISTEMA.....	68

3.2.2	ESPECIFICACIÓN DE CASO DE USO INFORMACIÓN DE LIGAS.....	70
3.2.3	ESPECIFICACIÓN DE CASO DE USO INFORMACIÓN DE EQUIPOS.....	72
3.2.4	ESPECIFICACIÓN DE CASO DE USO INFORMACIÓN DE JUGADORES.....	74
3.2.5	ESPECIFICACIÓN DE CASO DE USO NOTICIAS.....	77
3.2.6	ESPECIFICACIÓN DE CASO DE USO INFORMACIÓN TRANSFERENCIAS.....	79
3.3	USUARIOS.....	81
3.3.1	ADMINISTRADOR DEL SISTEMA.....	81
3.3.2	CONSULTOR DEL SISTEMA.....	82
3.3.3	USUARIOS EXTERNOS.....	82
3.3.4	NECESIDADES DE INTERESADOS Y USUARIOS.....	83
4.	Capitulo IV. ELABORACIÓN DE LA ARQUITECTURA DEL SOFTWARE.....	85
4.1	INTRODUCCIÓN.....	85
4.1.1	PROPÓSITO.....	85
4.1.2	ALCANCE.....	85
4.1.3	REFERENCIAS.....	86
4.1.4	RESUMEN.....	86
4.2	REPRESENTACIÓN DE LA ARQUITECTURA.....	86
4.3	OBJETIVOS Y RESTRICCIONES DE LA ARQUITECTURA.....	86
4.3.1	OBJETIVOS.....	86
4.3.2	RESTRICCIONES.....	87
4.4	VISTA DINÁMICA.....	87
4.4.1	INTRODUCCIÓN.....	87
4.4.2	DIAGRAMAS DE SECUENCIA.....	87
4.5	VISTA LÓGICA.....	89
4.5.1	INTRODUCCIÓN.....	89
4.5.2	DESCOMPOSICIÓN DE SUBSISTEMAS.....	89

4.6 VISTA DE DESPLIEGUE	90
4.6.1 SERVIDOR DE APLICACIONES.....	90
4.6.2 COMPUTADORES INTERNOS.....	90
4.6.3 COMPUTADORES EXTERNOS.....	91
4.7 DISEÑO EN CAPAS	91
4.7.1 CAPA INTERFACE DE USUARIO.....	91
4.7.2 CAPA LÓGICA DEL NEGOCIO.....	92
4.8 VISTA DE DATOS	92
4.8.1 MODELO RELACIONAL.....	92
4.8.2 MODELO FÍSICO.....	92
4.9 TAMAÑO Y DESEMPEÑO	93
4.9.1 TIEMPO DE RESPUESTA EN EL ACCESO A LA BASE DE DATOS.....	93
4.9.2 TIEMPO DE RESPUESTA DE TRANSACCIONES.....	93
4.9.3 ESPACIO EN DISCO PARA SERVIDOR DE BASE DE DATOS.....	93
4.10 CALIDAD	93
4.10.1 USABILIDAD.....	93
4.10.2 EFICIENCIA.....	94
4.10.3 SEGURIDAD.....	94
4.10.4 CONFIABILIDAD.....	94
4.10.5 MANTENIMIENTO.....	94
4.11 ARQUITECTURA DEL PROYECTO	94
4.11.1 INTRODUCCIÓN.....	94
4.11.2 PAQUETES DE ANÁLISIS.....	95
4.11.3 VISTA DINÁMICA.....	98
4.11.4 DIAGRAMAS DE SECUENCIA.....	99
4.11.5 VISTA DE PROCESOS.....	103
4.11.5.1 DIAGRAMA DE CLASE: CLASE SUBSISTEMA EQUIPOS.....	103
4.11.5.2 DIAGRAMA DE CLASE: CLASE SUBSISTEMA JUGADORES.....	104

4.11.5.3	DIAGRAMA DE CLASE: CLASE SUBSISTEMA TRANSFERENCIAS.....	105
4.11.5.4	DIAGRAMA DE CLASE: SISTEMA COMPLETO.....	106
4.12LISTA DE RIESGOS.....		107
4.12.1	INTRODUCCIÓN.....	107
4.12.1.1	PROPÓSITO.....	107
4.12.1.2	ALCANCE.....	107
4.12.1.3	REVISIÓN GENERAL.....	108
4.12.2	RIESGOS.....	109
4.12.2.1	RIESGO 1: LA BASE DE DATOS QUE SE PRETENDE UTILIZAR (POSTGRESQL) ES POCO CONOCIDA POR EL DESARROLLADOR.....	109
4.12.2.2	RIESGO 2: SE PUEDE PRESENTAR UN TRABAJO EN LA INSTITUCIÓN PARA PODER ORDENAR INFORMACIÓN NECESARIA QUE SERVIRÁ AL PROYECTO	110
4.12.2.3	RIESGO 3: NO PRESENTAR ACTUALIZADOS LOS DATOS DE LOS JUGADORES.....	111
4.12.2.4	RIESGO 4: ORDENADOR ES ATACADO POR VIRUS.....	112
4.12.2.5	RIESGO 5: MODELO VISTA CONTROLADOR NO SE ACOPLA A LA METODOLOGÍA RUP.....	113
4.12.2.6	RIESGO 6: INCOMPATIBILIDAD CON NAVEGADORES DE INTERNET Y/O CONFIGURACIONES ESPECÍFICAS EN LAS COMPUTADORAS CLIENTES.....	114
4.12.2.7	RIESGO 7: DESCONOCER EL PROCESO DE INSCRIPCIONES Y COMO SE VIENEN REALIZANDO LOS CAMPEONATOS EN LA INSTITUCIÓN.....	115
4.13PROTOTIPO DE INTERFACE DE USUARIO.....		117
4.13.1	INTRODUCCIÓN.....	117
4.13.1.1	PROPÓSITO.....	117

4.13.1.2	DESCRIPCIÓN.....	117
4.13.2	ARCHIVOS DE CONFIGURACIÓN.....	117
4.13.3	MAPA DE NAVEGACIÓN.....	118
4.13.4	ESTRUCTURA DE LAS PAGINAS.....	119
5.	CONCLUSIONES Y RECOMENDACIONES.....	131
	5.1 CONCLUSIONES.....	131
	5.2 RECOMENDACIONES.....	133
6.	BIBLIOGRAFÍA.....	134
7.	ANEXOS.....	141

ÍNDICE DE TABLAS

Tabla 1.- Plan de fases.....	43
Tabla 2.- Descripción de Hito.....	44
Tabla 3.- Objetivos de cada iteración.....	45
Tabla 4.- Entregas.....	45
Tabla 5.- Esfuerzo-horario contra fases del RUP.....	49
Tabla 6.- Calendario del proyecto Fase de Inicio.....	51
Tabla 7.- Calendario de proyectos. Fase de Elaboración.....	53
Tabla 8.- calendario del proyecto. Fase de construcción Iteración I.....	55
Tabla 9.- calendario del proyecto. Fase de construcción Iteración II.....	57
Tabla 10.- calendario del proyecto. Fase de construcción Iteración III.....	59
Tabla 11.- Caso de uso obtener información de equipo de fútbol.....	63
Tabla 12.- Caso de uso obtener información de jugadores de fútbol.....	64
Tabla 13.- Caso de uso obtener información de jugadores de fútbol.....	65
Tabla 14.- Caso de uso obtener información de transferencias de jugadores.....	66
Tabla 15.- Resumen casos de uso.....	67
Tabla 16.- Perfiles de usuario administrador del sistema.....	81
Tabla 17.- Perfiles de usuario consultor del sistema.....	82
Tabla 18.- Perfiles de usuario externo del sistema.....	82
Tabla 19.- Necesidades de interesados y usuarios.....	83
Tabla 20.- Descripción de los paquetes de análisis.....	96
Tabla 21. Estructura de la página Index.php.....	199
Tabla 22.- Estructura del formulario Validar usuario.....	121

Tabla 23.- Estructura del formulario Inicio.php.....	122
--	-----

TABLAS ANEXOS

Tabla A. 1.- Tabla que permite llevar el control de campeonatos.....	A141
Tabla A. 2.- Tabla que permite llevar el control de categorías.....	A142
Tabla A. 3.- Tabla que contiene datos sobre la directiva y contactos de cada liga...	A143
Tabla A. 4.- Tabla que permite llevar el control de los equipos.....	A145
Tabla A. 5.- Tabla que permite llevar el control de las tablas de posiciones.....	A147
Tabla A. 6.- Tabla que permite llevar el control de los estadios.....	A148
Tabla A. 7.- Tabla que permite llevar el control de las etapas.....	A149
Tabla A. 8.- Tabla que permite llevar el control de los goleadores.....	A150
Tabla A. 9.- Tabla que permite llevar el control de los jugadores.....	A151
Tabla A. 10.- Tabla que permite llevar el control de las ligas.....	A153
Tabla A. 11.- Tabla que permite llevar el control de las noticias.....	A154
Tabla A. 12.- Tabla que permite llevar el control de las series.....	A155
Tabla A. 13.- Tabla que permite llevar el control de las transferencias.....	A156
Tabla A. 14.- Tabla que permite llevar el control de las cuentas de usuarios.....	A158

ÍNDICE DE GRÁFICOS

Ilustración 1. Modelo global del sistema.....	6
Ilustración 2. Modelo vista controlador (MVC).....	8
Ilustración 3. Herramientas de software libre.....	9
Ilustración 4. Selección del lenguaje de instalación de Xampp.....	14
Ilustración 5. Aceptación de términos y condiciones.....	14
Ilustración 6. Selección del directorio de instalación.....	15
Ilustración 7. Selección de Iconos del escritorio y servicios.....	15
Ilustración 8: Instalación completada.....	16
Ilustración 9. Panel de control.....	16
Ilustración 10: Página principal del Xampp.....	17
Ilustración 11: Selección de Idioma de descarga de Netbeans.....	20
Ilustración 12: Bienvenida a la instalación.....	21
Ilustración 13: Imagen de localización del jdk.....	21
Ilustración 14: Finalización de la instalación.....	22
Ilustración 15: Mapa de los sistemas web.....	24
Ilustración 16: Funcionamiento sistemas web.....	26
Ilustración 17: Selección de idioma de Postgres.....	28
Ilustración 18: Bienvenida a la instalación de Postgres.....	29
Ilustración 19: Términos y condiciones.....	29
Ilustración 20: Opciones de instalación.....	30
Ilustración 21: Configuración del servicio.....	30

Ilustración 22: Configuración de password.....	31
Ilustración 23: Aceptación de usuario y password.....	31
Ilustración 24: Inicialización de Clúster.....	32
Ilustración 25: Activación de lenguajes de procedimientos.....	33
Ilustración 26: Habilitación de módulos contrib.....	33
Ilustración 27: Instalación de Postgres.....	34
Ilustración 28: Progreso de la instalación.....	34
Ilustración 29: Finalización de la instalación.....	35
Ilustración 30: Localización de pgAdmin III.....	36
Ilustración 31: PgAdmin III.....	36
Ilustración 32: Autenticación.....	37
Ilustración 33.- Cronograma de trabajo.....	46
Ilustración 34.- Fases del RUP.....	48
Ilustración 35.- Recursos utilizados en las fases del RUP en el tiempo.....	50
Ilustración 36.- Esfuerzo respecto de los flujos de trabajo.....	50
Ilustración 37.- Caso de uso obtener información de equipo.....	63
Ilustración 38.- Caso de uso obtener información de jugador.....	64
Ilustración 39.- Caso de uso obtener información de liga barrial.....	65
Ilustración 40.- Caso de uso obtener información de transferencias.....	66
Ilustración 41.- Recorte tabla usuarios.....	68
Ilustración 42.- Recorte del formulario insertar usuarios.....	69
Ilustración 43.- Recorte tabla liga.....	70
Ilustración 44.- Recorte del formulario insertar ligas.....	71

Ilustración 45.- Recorte tabla equipo.....	72
Ilustración 46.- Recorte formulario equipo.....	73
Ilustración 47.- Recorte de la tabla jugador.....	74
Ilustración 48.- Recorte formulario ingresar jugador.....	75
Ilustración 49.- Recorte tabla noticia.....	77
Ilustración 50.- Recorte formulario noticia.....	78
Ilustración 51.- Recorte tabla transferencia.....	79
Ilustración 52.- Recorte formulario transferencia.....	80
Ilustración 53.- Diagrama de secuencia: Arquitectura MVC.....	88
Ilustración 54.- Diagrama de despliegue: Modelo MVC.....	90
Ilustración 55.- Diagrama de distribución de Capas del sistema.....	91
Ilustración 56.- Dependencias entre paquetes del sistema web.....	98
Ilustración 57.- Diagrama de secuencia Caso de Uso: Asignación de usuarios al sistema.....	99
Ilustración 58.- Diagrama de secuencia Caso de Uso: Información de ligas.....	99
Ilustración 59.- Diagrama de secuencia Caso de Uso: Información de equipos.....	100
Ilustración 60.- Diagrama de secuencia Caso de Uso: Ingreso de jugadores.....	101
Ilustración 61.- Diagrama de secuencia Caso de Uso: Noticias.....	101
Ilustración 62.- Diagrama de secuencia Caso de Uso: Información de transferencias..	102
Ilustración 63.- Diagrama de clase: Clase subsistema equipos.....	103
Ilustración 64.- Diagrama de clase: Clase subsistema jugadores.....	104
Ilustración 65.- Diagrama de clase: Clase subsistema Transferencias.....	105
Ilustración 66.- Diagrama de clases del sistema completo.....	106
Ilustración 67.- Mapa de sitio y navegación.....	118

Ilustración 68.- Mapa- sitio. Php.....	119
Ilustración 69.- Estructura de la página Index.php.....	120
Ilustración 70.- Prototipo de interface de usuario: Index.php.....	120
Ilustración 71.- Estructura de la parte del formulario de validación de usuarios.....	121
Ilustración 72.- Prototipo de interface de usuario: Validar usuario.....	121
Ilustración 73.- Estructura de la página Inicio.php.....	122
Ilustración 74.- Prototipo de interfaz de usuario: Inicio.php.....	123

RESUMEN

El siguiente trabajo contiene un material que ha sido recopilado a lo largo de los años ya que los registros que pertenecen a las ligas barriales y parroquiales de Imbabura se han ido incrementando de una forma muy rápida por lo que se ha creído necesario y fundamental realizar un sistema informático web en el que se automatizara el proceso de inscripciones y manejo de fichajes de los jugadores así como los partidos de fútbol que se vienen dando semana tras semana en los diferentes escenarios deportivos.

El sistema brindara transparencia en el proceso deportivo que viene dando año a año en la federación deportiva de ligas barriales, se espera que con este sistema se despejen todas las dudas que existen en los jugadores y directivos de todos los equipos.

Se espera que el sistema sea implantado en todas las provincias para que sea transparente a nivel de provincias y luego tener un solo sistema que maneje los pases de todos los jugadores que son afiliados a las ligas barriales que es una entidad importante donde salen jugadores que representan al país en los torneos internacionales.

Al momento existen 5 torneos que se realiza en la FEDERACIÓN DE LIGAS BARRIALES, que son: campeonato sub-12, campeonato femenino, campeonato Master, campeonato Sénior y el campeonato de interligas que es uno de los más importantes ya que los equipos que más se han destacado en todo el año de campeonato en cada una de las ligas a la que pertenecen.

Ha quedado resuelto el problema y se ha cumplido con los objetivos y metas por lo que se puede decir que se ha puesto en marcha el sistema y está funcionando correctamente a favor de los jugadores y equipos.

SUMMARY

The following paper contains a material that has been gathered over the years as the records belonging to the neighborhood and parish leagues Imbabura have been increasing in a very fast so it was thought necessary and essential to make a web information system which would automate the process of registration and management signings of players and football matches that are taking place every week in various sports venues.

The system would provide transparency in the sport that has been giving every year in the neighborhood league sports federation, it is expected that this system will clear all the doubts that exist in the players and managers of all teams.

It is expected that the system is implemented in all provinces to be transparent to the provincial level and then have a single system that handles passes all players who are affiliated to the neighborhood links is an important entity where players come representing the country in international tournaments.

At the moment there are 5 tournaments taking place in the neighborhood LINKS FEDERATION, which are his-12 championship, championship female Master Championship, Senior Championship and the interleague championship as one of the most important since most teams highlighted throughout the championship year in each of the leagues to which they belong.

To become resolved the problem and has met the goals and objectives so we can say that it has launched the system is working properly for players and teams.

SUMMARY

The following paper contains a material that has been gathered over the years as the records belonging to the neighborhood and parish leagues Imbabura have been increasing in a very fast so it was thought necessary and essential to make a web information system which would automate the process of registration and management signings of players and football matches that are taking place every week in various sports venues.

The system would provide transparency in the sport that has been giving every year in the neighborhood league sports federation, it is expected that this system will clear all the doubts that exist in the players and managers of all teams.

It is expected that the system is implemented in all provinces to be transparent to the provincial level and then have a single system that handles passes all players who are affiliated to the neighborhood links is an important entity where players come representing the country in international tournaments.

At the moment there are 5 tournaments taking place in the neighborhood LINKS FEDERATION, which are his-12 championship, championship female Master Championship, Senior Championship and the interleague championship as one of the most important since most teams highlighted throughout the championship year in each of the leagues to which they belong.

To become resolved the problem and has met the goals and objectives so we can say that it has launched the system is working properly for players and teams.

1. PROBLEMA

1.1 ANTECEDENTES

FEDELIGAS federación de ligas barriales de la ciudad de Ibarra es una institución que se dedica a la organización del deporte barrial y constituye un gran aporte para la formación de futuros profesionales en el mundo del deporte cuya meta es hacer quedar muy en alto el nombre de la provincia de Imbabura.

Esta institución está estructurada por:

- Presidente.
- Secretario.
- Tesorero.

Y se compone de diferentes ligas barriales como:

- San Miguel de Ibarra.
- Las palmas.
- Jesús del gran poder.
- Priorato.
- San Pablo.
- Alpachaca.
- La Esperanza.
- San Antonio.
- Antonio Ante.
- Caranqui.

Cada una de estas ligas está conformada por 2 categorías A y B... Cada una de estas con sus respectivos equipos.

1.2 SITUACIÓN ACTUAL

FEDELIGAS en la actualidad no se encuentra automatizado en el proceso general de la gestión del fútbol como son: inscripciones, pases, equipos, mejores jugadores.

Cuenta con una pequeña base de datos en Access donde se registran equipos y jugadores.

La institución no cuenta con un sistema informático web donde se promocionen jugadores ya que los medios de comunicación poco y nada se integran a este semillero de deportistas que se han perdido por falta de promoción en medios.

PROSPECTIVA

Si no damos solución a este problema no tendremos en el futuro talentos de nuestra provincia que nos han hecho quedar bien en el fútbol nacional e internacional, y las personas no estarán informadas de la calidad de jugadores que tenemos en nuestra provincia.

Brindar transparencia en el proceso selectivo de jugadores.

No existiría competitividad de nuestros deportistas en relación a otros que están triunfando en el fútbol internacional.

1.3 FORMULACIÓN DEL PROBLEMA.

FEDELIGAS mantiene sus registros en una base de datos que no es tan funcional y dinámica como lo es Access y no promocionan los nuevos talentos que nacen semana tras semana en los escenarios deportivos de la ciudad.

No existe nada de información sobre transferencias de jugadores a los diferentes clubs ni el tipo de pase ya sea a préstamo o definitivo.

Es difícil hacer un seguimiento a los diferentes deportistas para conocer en que clubes ha desarrollado sus actividades como deportista.

No existe la debida información en cuanto a noticias que se debe conocer en los diferentes torneos que se desarrollan en la institución.

1.3.1 ¿A QUIÉN AFECTA?

- A los usuarios de la institución.
- A los periodistas.
- A los jugadores.
- A los dirigentes de ligas barriales.
- A los dirigentes de equipos.
- A los empresarios encargados de los jugadores.

1.3.2 ¿QUÉ IMPACTO CAUSA?

- Pérdida de tiempo en la búsqueda de transferencias y de información sobre los jugadores, clubs y campeonatos.
- Perdida de fichas y pases.
- Existen controles manuales que producen que la información se pierda lo cual no permite la libre comercialización de jugadores.
- Poca información acerca de los clubs que forman parte de las diferentes ligas barriales.

1.3.3 ¿SOLUCIÓN EXITOSA?

- Implementar una solución informática de calidad que cumpla con todas las normas del desarrollo de software.
- Obtener información de manera rápida y ordenada de jugadores, transferencias, noticias, campeonatos, equipos.

- Mostrar los escenarios de encuentros, con la hora y fecha exacta a través de la página web que tendrá el sistema.
- Cada usuario podrá observar los datos que requiera y los permisos o restricciones que son asignados a cada uno de ellos.
- Obtener información como tablas de posiciones y tablas de goleadores de los diferentes torneos, categorías y campeonatos desarrollados.

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERAL

Crear un sistema informático web donde se apoye la gestión de campeonatos de fútbol barrial en la ciudad y donde se registre el proceso deportivo de los diferentes jugadores en las diferentes ligas, promocionar jugadores.

1.4.2 OBJETIVOS ESPECÍFICOS

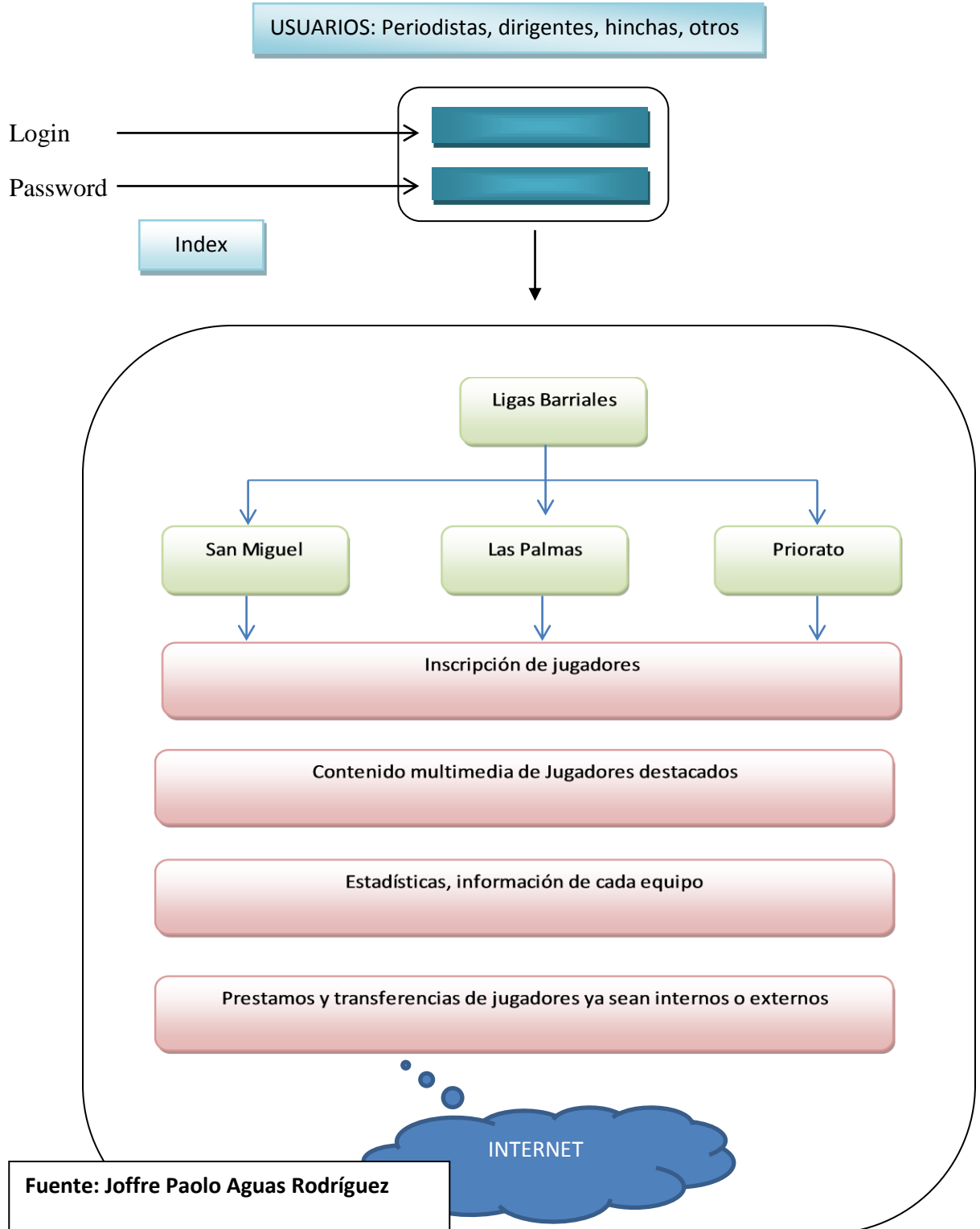
- Organizar la información de equipo y ligas barriales en los diferentes campeonatos.
- Organizar la información de todos los deportistas que participan en los diferentes torneos.
- Agregar al sistema web una base de datos grande que no se caiga con el acceso múltiple de usuarios como lo es Postgres.

- Someter al sistema a pruebas de conectividad transmitiendo jugadas y los mejores goles de tal manera que las personas que visiten esta página puedan acceder a los perfiles de los mejores futbolistas.
- Escoger un buen servidor de aplicaciones libre que no caiga la aplicación aunque tenga un gran número de visitas.

1.5 ALCANCE

- El sistema informático web tendrá imágenes, en los cuales los usuarios podrán apreciar la buena calidad de jugadores a promocionar.
- El sistema primero será implementado en una red interna para ser sometido a pruebas y subir toda la información posible.
- Se podrá migrar a otras tecnologías.

Ilustración 1. Modelo global del sistema.



1.6 JUSTIFICACIÓN DE PROYECTO

El presente sistema informático web tendrá contenidos de una investigación profunda de la vida futbolística de los deportistas imbabureños de años atrás.

El sistema informático web apoyara a FEDELIGAS:

- Mediante la inscripción de equipos y jugadores por parte de los secretarios que son los que van a digitar los datos correspondientes.
- El sistema brindara a los jugadores la posibilidad de salir a debutar en torneos nacionales e internacionales y crecer como futbolistas.
- Estadísticas de los partidos que se vienen dando semana tras semana, tablas de posiciones, goleadores, calendario de encuentros.

El sistema informático web constara de 2 tipos de usuarios:

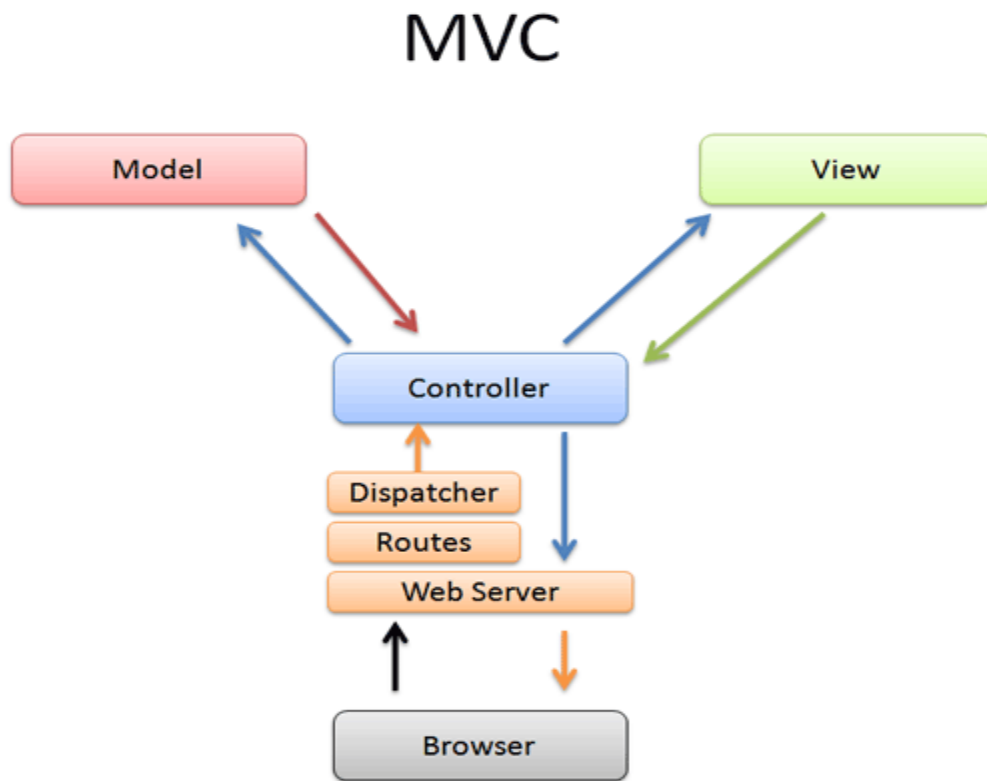
1. Administrador del sistema que tendrá acceso a todos los lugares como:
 - Inscripciones de equipos y jugadores.
 - Equipos y pases.
 - Links para subir videos y reportajes de los mejores deportistas de la fecha.
2. Usuario final que tendrá acceso:
 - Solo para ver los mejores jugadores de la semana.
 - Contactos donde localizarlos.
 - Equipo en el que debuta el jugador.
 - Recorrido del jugador.
 - Calendario de los próximos torneos.

En el sistema informático web estaremos usando base de datos postgres, herramientas de desarrollo libre como Netbeans se estará usando estas tecnologías de desarrollo libre por que no tienen costo alguno.

Este sistema informático web va dirigido a los diferentes tipos de usuarios como:

- Periodistas: Que son los que se encargan de difundir información al país, los mismos que tendrán acceso a estadísticas.
- Dirigentes de cada equipo que tendrán que ir alimentando la base de datos con estadísticas de los jugadores que tienen en sus filas.
- Hinchas tienen la posibilidad de enterarse de lo que sucede en los escenarios deportivos en lo que respecta a sus equipos favoritos.
- Usuarios internacionales que se conectaran con el sistema informático web mediante un browser ya que la aplicación contara con un servidor de aplicaciones el mismo que facilitara el acceso a la aplicación.

Ilustración 2. Modelo vista controlador (MVC)



Fuente: www.google.com/imagenes/mvc

2.1 HERRAMIENTAS DE SOFTWARE LIBRE

Ilustración 3. Herramientas de software libre



Fuente: Manual apache, netbeans y postgres

2.1.1 ¿QUÉ ES EL SOFTWARE LIBRE?

Software Libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

- La libertad de usar el programa, con cualquier propósito (**libertad 0**).
- La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades (**libertad 1**). El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias, con lo que puedes ayudar a tu vecino (**libertad 2**).
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (**libertad 3**). El acceso al código fuente es un requisito previo para esto. [1] www.soft-libre.net/docs/pgarrido.pdf

Un programa es software libre si los usuarios tienen todas estas libertades. Así pues, se debería tener la libertad de distribuir copias, sea con o sin modificaciones, sea gratis o cobrando una cantidad por la distribución, a cualquiera y a cualquier lugar. El ser libre de hacer esto significa (entre otras cosas) que no se tiene que pedir o pagar permisos.

También se debería tener la libertad de hacer modificaciones y utilizarlas de manera privada en tu trabajo u ocio, sin ni siquiera tener que anunciar que dichas modificaciones existen. Si se publica cambios, no se tiene por qué avisar a nadie en particular, ni de ninguna manera en particular. [1]

La libertad para usar un programa significa la libertad para cualquier persona u organización de usarlo en cualquier tipo de sistema informático, para cualquier clase de trabajo, y sin tener obligación de comunicárselo al desarrollador o a alguna otra entidad específica. [2] www.utpl.edu.ec/ried/images/pdfs/uso.pdf

La libertad de distribuir copias debe incluir tanto las formas binarias o ejecutables del programa como su código fuente, sean versiones modificadas o sin modificar (distribuir programas de modo ejecutable es necesario para que los sistemas operativos libres sean fáciles de instalar). Está bien si no hay manera de producir un binario o ejecutable de un programa concreto (ya que algunos lenguajes no tienen esta capacidad), pero se debe tener la libertad de distribuir estos formatos si se encuentra o desarrolla la manera de crearlos.

Para que las libertades de hacer modificaciones y de publicar versiones mejoradas tengan sentido, se debe tener acceso al código fuente del programa. Por lo tanto, la posibilidad de acceder al código fuente es una condición necesaria para el software libre. [3] www.asle.ec/portal/

Para que estas libertades sean reales, deben ser irrevocables mientras no se haga nada incorrecto; si el desarrollador del software tiene el poder de revocar la licencia aunque no se le haya dado motivos, el software no es libre.

Son aceptables, sin embargo, ciertos tipos de reglas sobre la manera de distribuir software libre, mientras no entren en conflicto con las libertades centrales. Por ejemplo, *copyleft* es la regla que implica que, cuando se redistribuya el programa, no se pueden

agregar restricciones para denegar a otras personas las libertades centrales. Esta regla no entra en conflicto con las libertades centrales, sino que más bien las protege.

'*Software libre*' no significa '*no comercial*'. Un programa libre debe estar disponible para uso comercial, desarrollo comercial y distribución comercial. El desarrollo comercial del software libre ha dejado de ser inusual; el software comercial libre es muy importante. [3]

Pero el software libre sin copyleft también existe. Creemos que hay razones importantes por las que es mejor usar copyleft, pero si tus programas son software libre sin ser copyleft, los podemos utilizar de todos modos.

Cuando se habla de software libre, es mejor evitar términos como: 'regalar' o 'gratis', porque esos términos implican que lo importante es el precio, y no la libertad.

2.1.2 ¿QUÉ ES UN SERVIDOR WEB?

Se podría definir un servidor WEB como una aplicación que permite acceder a los recursos contenidos en algunos de los directorios del ordenador que la alberga a usuarios remotos que realizan sus peticiones mediante el protocolo HTTP.

Por tanto, instalar un servidor web no es otra cosa que instalar y configurar un programa en una unidad o directorio de un ordenador cualquiera.

2.1.2.1 ¿QUÉ ES APACHE?

Bajo este nombre suele hacerse referencia a Apache Software Foundation, organización norteamericana que se autodefine con el objetivo de «facilitar ayuda organizativa, legal y financiera para los proyectos de desarrollo de software tipo Open Source (código abierto)». [5] es.wikipedia.org/wiki/Servidor_HTTP_Apache

Uno de los proyectos más populares de Apache es el desarrollo y suministro de forma gratuita y libre de un software de servidor HTTP, conocido también como el servidor Apache.

Para realizar la instalación de apache bajo Windows se ha optado por hacerlo mediante XAMPP, que nos permitirá realizar la instalación de un modo muy sencillo de los tres componentes, de tal modo que tengamos funcionando el sistema en muy poco tiempo.

2.1.2.2 ¿QUÉ ES XAMMP?

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. [6] es.wikipedia.org/wiki/XAMPP

DISPONIBILIDAD

Fácil de usar y capaz de interpretar páginas dinámicas. Actualmente xampp está disponible para Microsoft Windows, gnu/Linux, Solaris, y macos x.

INSTALACIÓN

Para instalar xampp sólo se requiere descargar y extraer xampp, es todo. No hay cambios en el registro de Windows (a no ser que se utilice la versión de xampp para el instalador de Windows) y no es necesario editar ningún archivo de configuración.

[7] www.apachefriends.org/es/xampp.html

DONDE GUARDARLO

Al correr la instalación, lo primero que se tiene que decidir es dónde guardarlo (recomiendo dejarlo en la carpeta que aparece por defecto **c:\xampp**). La ubicación es importante porque ahí es donde se ubicarán físicamente los sitios que se quiera mostrar en el servidor. [7]

ICONOS Y OPCIONES DEL SERVICIO

El siguiente paso es para la creación de los íconos en el escritorio y qué opciones como servicio quieres instalar. Por defecto estas opciones están sin marcar. Si se quiere que cada vez que se prenda la computadora corra el servidor, se debe seleccionar que instale Apache, MySQL y FileZilla como servicios.

Caso contrario, se puede dejarlos sin señalar. Se van a instalar igual pero se deberá activar las opciones de forma manual cada vez que se necesite correr el servidor.

PANEL DE CONTROL

Luego de un par de siguientes y hacer clic en finalizar, el instalador va a preguntar si se desea abrir el panel de control. Desde esa nueva ventana es que se va a poder prender y apagar el servidor Apache y MySQL (PHP se inicia cuando se inicia Apache) cada vez que se lo necesite. [7]

LISTO!!!

Ya está instalado. Ahora sólo se tiene que guardar los archivos del sitio dentro de la carpeta **c:\xampp\htdocs** para que puedan ser vistos desde **http://localhost/** en los navegadores.

Por ejemplo si yo tengo el archivo **c:\xampp\htdocs\index.php**, lo voy a poder visualizar desde Internet Explorer o Firefox sin problemas escribiendo **http://localhost/index.php**.

SU COSTO

XAMMP es una compilación de software libre es gratuito y libre de ser copiado conforme a los términos de la licencia. Eso si este software es distribuido gratuitamente con muchos beneficios pero sin ninguna garantía.

PASOS PARA LA INSTALACIÓN

1. Selección del lenguaje

Ilustración 4. Selección del lenguaje de instalación de Xampp.



2. Aceptación de términos y condiciones

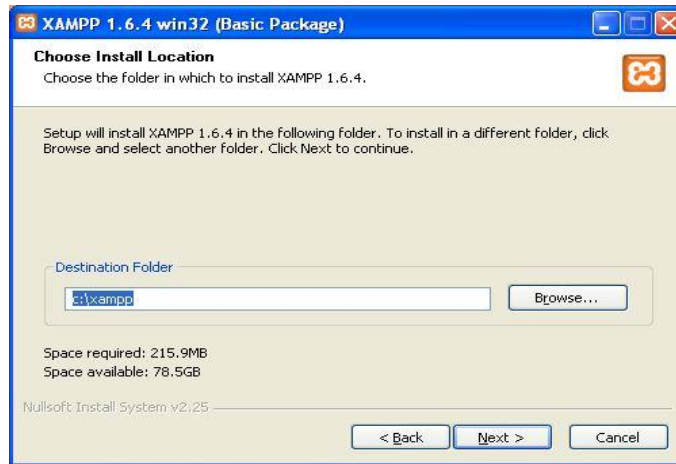
Ilustración 5. Aceptación de términos y condiciones



Recomendaciones para realizar la instalación donde sugiere que se cierren las aplicaciones que se tenga abierto en ese momento ya que el sistema estará realizando actualizaciones y cambios en el sistema. [8] www.nosolocodigo.com/como-instalar-xampp-en-windows.

3. Selección del directorio de instalación.

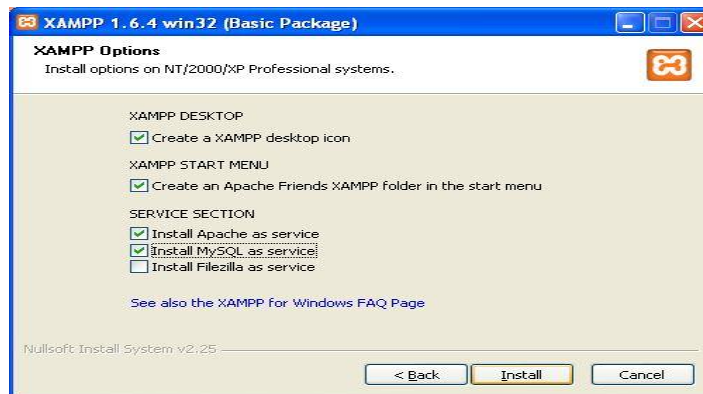
Ilustración 6. Selección del directorio de instalación.



Seleccione la carpeta donde quiere instalar el servidor se recomienda instalar en el directorio que ofrece por defecto es decir “C:\xampp”.

4. Selección de Iconos del escritorio.

Ilustración 7. Selección de Iconos del escritorio y servicios.



Se escoge opciones adicionales, por ejemplo, si se quiere que se ejecuten como servicios, en cuyo caso cada vez que se inicie el sistema operativo se ejecutarán. Se puede seleccionar, además, dónde se mostrará los iconos de inicio de Xampp, desde dónde se puede iniciar el resto de servidores.

5. Instalación completa...

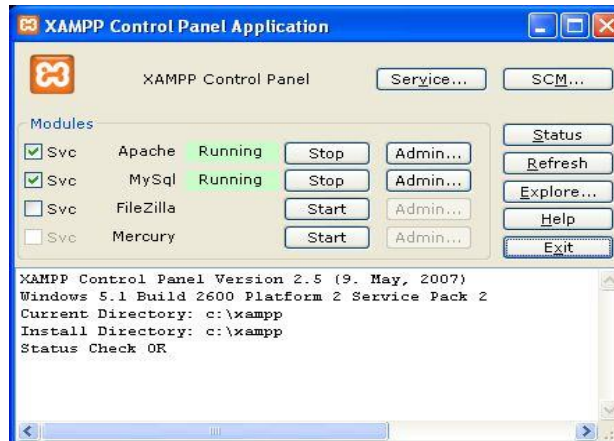
Ilustración 8: Instalación completada



Si todo ha ido bien, mostrará una ventana de consola informando de que todo ha ido bien, seguida de una ventana informando de la instalación completa.

POSIBLES OPCIONES DE CONFIGURACIÓN Y ADMINISTRACIÓN

Ilustración 9. Panel de control

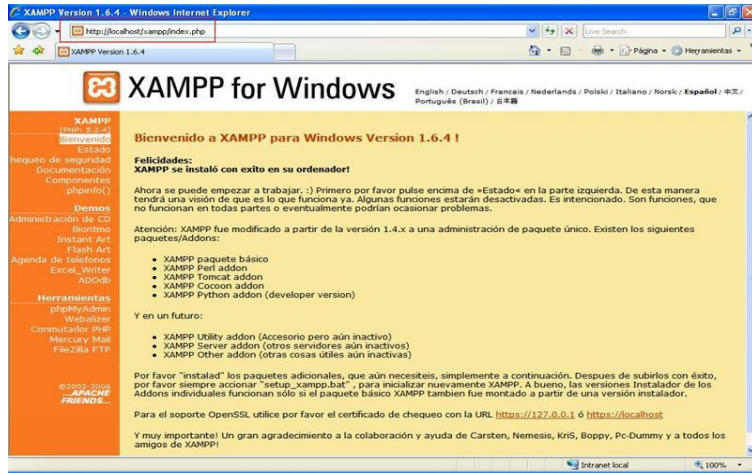


La ilustración 9, muestra el momento en que se inicia el programa servidor de aplicaciones. Se da clic en Start para iniciar cada uno de los servicios que contiene el servidor o en stop para pararlos cuando ya no los vayamos a utilizar. [8]

Al colocar el visto en la casilla “Svc” estamos diciendo que queremos que se inicien los servicios al iniciar el equipo como parte de los servicios del sistema operativo.

PARA REVISAR SI LA INSTALACIÓN ES CORRECTA BASTA PONER EL NAVEGADOR [HTTP://127.0.0.1](http://127.0.0.1) O [HTTP://LOCALHOST](http://localhost)

Ilustración 10: Página principal del xampp.



2.1.3 PHP

INTRODUCCIÓN

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+. [9] php.net/manual/es/index.php

PHP es un acrónimo recursivo que significa **PHP Hypertext Preprocessor** (Procesador de Hipertexto).

Fue creado originalmente por Rasmus Lerdorf en 1994; Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. [9]

VENTAJAS

- Es un lenguaje multiplataforma.
- Completamente orientado a la web.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.

- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.

DESVENTAJAS

- Consume muchos recursos (y por tanto es ligeramente más lento) llamando y ejecutando una función que ejecutando código que encuentra en línea, embebido en el script. Por eso no tiene sentido usar funciones salvo que efectivamente se vaya a utilizar esa porción de código varias veces.

2.1.4 NETBEANS

INTRODUCCIÓN

NetBeans es un IDE, Entorno Integrado de Desarrollo (por sus siglas en inglés) que nos permite crear aplicaciones no solo en el lenguaje de programación Java, sino también en una gran variedad de lenguajes en los cuales se está creando el soporte. [10] es.wikipedia.org/wiki/NetBeans

INSTALACIÓN DE NETBEANS

Lo primero que hay que hacer es ir a la página de descarga de Netbeans <http://www.netbeans.org/downloads/>. Allí aparecen todas las descargas para el IDE, lo primero que hay que hacer es cambiar la página de descarga al español, seleccionando español en la parte de lenguaje de IDE. [11] netbeans.org/community/releases/68/install_es.html

Ilustración 11: Selección de Idioma de descarga de Netbeans.



Después dar clic en el botón download de la columna que se quiere instalar, aparece una ventana con la descarga del archivo, el cual se debe guardar en un lugar donde sea fácil ubicarlo. Ya con el paquete descargado solo se tiene que ejecutarlo dándole clic al archivo, la ejecución puede ser algo lenta así que hay que esperar un poco, cuando al fin se ejecute aparecerá una ventana con el asistente de instalación como la siguiente:

Ilustración 12: Bienvenida a la instalación.



Esta pantalla se muestran los paquetes que se van a instalar se da clic en siguiente para continuar con la instalación.

En la siguiente ventana nos pregunta si aceptamos los términos de licencia, se le da en siguiente.

Después preguntara dónde se quiere instalar el IDE y la ubicación jdk, es conveniente dejarle la ubicación así como está. Le da clic en siguiente:

Ilustración 13: Imagen de localización del jdk



El asistente muestra la configuración básica por defecto del servidor glash Fish, no es necesario hacer ningún cambio así que se le da en siguiente, siguiente y luego instalar.

Ilustración 14: Finalización de la instalación.



Después que termina solo se tiene que dar clic en terminar y ya está instalado Netbeans.

2.2 SISTEMAS WEB

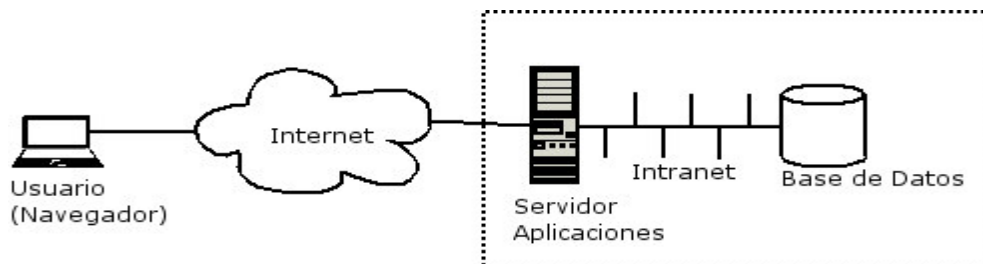
Las aplicaciones web se han convertido en pocos años en complejos sistemas con interfaces de usuario cada vez más parecidas a las aplicaciones de escritorio, dando servicio a procesos de negocio de considerable envergadura y estableciéndose sobre ellas requisitos estrictos de accesibilidad y respuesta. Esto ha exigido reflexiones sobre la mejor arquitectura y las técnicas de diseño más adecuadas. En este documento se pretende dar un breve repaso a la arquitectura de tales aplicaciones y a los patrones de diseño más aplicables. [12] sistemasweb.com/

En los últimos años, la rápida expansión de Internet y del uso de intranets corporativas ha supuesto una transformación en las necesidades de información de las organizaciones. En particular esto afecta a la necesidad de que:

1. La información sea accesible desde cualquier lugar dentro de la organización e incluso desde el exterior.
2. Esta información sea compartida entre todas las partes interesadas, de manera que todas tengan acceso a la información completa en cada momento.

Estas necesidades han provocado un movimiento creciente de cambio de las aplicaciones tradicionales de escritorio hacia las aplicaciones web, que por su particularidad, cumplen a la perfección con las necesidades mencionadas anteriormente. Por tanto, los sitios web tradicionales que se limitaban a mostrar información se han convertido en aplicaciones capaces de una interacción más o menos sofisticada con el usuario. Inevitablemente, esto ha provocado un aumento progresivo de la complejidad de estos sistemas y, por ende la necesidad de buscar opciones de diseño nuevas que permitan dar con la arquitectura óptima que facilite la construcción de los mismos.[13] www.eticaygestion.org/documentos/e-business/4.pdf

Ilustración 15: Mapa de los sistemas web.



Los sistemas web tienen la siguiente topología: el usuario mediante un navegador de internet previamente instalado accede a los datos que se encuentran en un servidor de aplicaciones que se encuentra en una red interna y que contiene una base de datos a los cuales el usuario accede de acuerdo a sus necesidades.

El usuario interactúa con las aplicaciones web a través del navegador. Como consecuencia de la actividad del usuario, se envían peticiones al servidor, donde se aloja la aplicación y que normalmente hace uso de una base de datos que almacena toda la información relacionada con la misma. El servidor procesa la petición y devuelve la respuesta al navegador que la presenta al usuario. Por tanto, el sistema se distribuye en tres componentes: el navegador, que presenta la interfaz al usuario; la aplicación, que se encarga de realizar las operaciones necesarias según las acciones llevadas a cabo por éste y la base de datos, donde la información relacionada con la aplicación se hace persistente. Esta distribución se conoce como el modelo o arquitectura de tres capas. [13]

En la mayoría de los casos, el navegador suele ser el presentador de información (modelo de cliente delgado), y no lleva a cabo ningún procesamiento relacionado con la lógica de negocio. No obstante, con la utilización de applets, código JavaScript y DHTML la mayoría de los sistemas se sitúan en un punto intermedio entre un modelo de cliente delgado y un modelo de cliente grueso (donde el cliente realiza el procesamiento de la información y el servidor sólo es responsable de la administración de datos). No obstante el procesamiento realizado en el cliente suele estar relacionado con aspectos de la interfaz (como ocultar o mostrar secciones de la página en función de determinados eventos) y nunca con la lógica de negocio. [13]

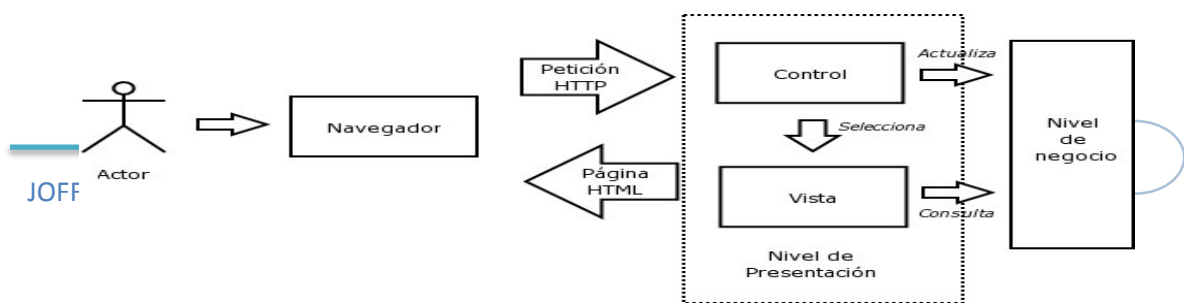
En todos los sistemas de este tipo a cada una de las capas de despliegue comentadas, podemos dividir la aplicación en tres áreas o niveles:

1. **Nivel de presentación:** Es el encargado de generar la interfaz de usuario en función de las acciones llevadas a cabo por el mismo.
2. **Nivel de negocio:** Contiene toda la lógica que modela los procesos de negocio y es donde se realiza todo el procesamiento necesario para atender a las peticiones del usuario.
3. **Nivel de administración de datos:** encargado de hacer persistente toda la información, suministra y almacena información para el nivel de negocio.

Uno de los patrones que ha demostrado ser fundamental a la hora de diseñar aplicaciones web es el **Modelo-Vista-Control (MVC)**. Este patrón propone la separación en distintos componentes de la interfaz de usuario (vistas), el modelo de negocio y la lógica de control. Una vista es una “fotografía” del modelo (o una parte del mismo) en un determinado momento. [14] es.wikipedia.org/wiki/Modelo_Vista_Controlador

En una aplicación web, las vistas serían las páginas HTML que el usuario visualiza en el navegador. A través de estas páginas el usuario interactúa con la aplicación, enviando eventos al servidor a través de peticiones HTTP. En el servidor se encuentra el código de control para estos eventos, que en función del evento concreto actúa sobre el modelo convenientemente. Los resultados de la acción se devuelven al usuario en forma de página HTML mediante la respuesta HTTP. [13]

Ilustración 16: Funcionamiento sistemas web.



2.3 BASES DE DATOS

Postgres, desarrollada originalmente en el Departamento de Ciencias de la Computación de la Universidad de California en Berkeley, fue pionera en muchos de los conceptos de bases de datos relacionales orientadas a objetos que ahora empiezan a estar disponibles en algunas bases de datos comerciales. [15]

structio.sourceforge.net/guias/servidor_OpenBSD/postgresql.html

El Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos conocido como PostgreSQL está derivado del paquete Postgres escrito en Berkeley. PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, perl y python). [15]

2.3.1 ¿QUÉ ES POSTGRES?

Los sistemas de mantenimiento de Bases de Datos relacionales tradicionales (DBMS) soportan un modelo de datos que consisten en una colección de relaciones con nombre, que contienen atributos de un tipo específico. En los sistemas comerciales actuales, los tipos posibles incluyen numéricos de punto flotante, enteros, cadenas de caracteres, cantidades monetarias y fechas. Está generalmente reconocido que este modelo será inadecuado para las aplicaciones futuras de procesado de datos. El modelo relacional sustituyó modelos previos por ser simple. Sin embargo, como se ha mencionado, esta simplicidad también hace muy difícil la implementación de ciertas aplicaciones. Postgres ofrece una potencia adicional sustancial al incorporar los siguientes cuatro

conceptos adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema:

- ❖ Clases
- ❖ Herencia
- ❖ Tipos
- ❖ Funciones

Otras características aportan potencia y flexibilidad adicional:

Restricciones (Constraints)

Disparadores (triggers)

Reglas (rules)

Integridad transaccional

Estas características colocan a Postgres en la categoría de las Bases de Datos identificadas como objeto-relacionales. Estas son diferentes de las referidas como orientadas a objetos, que en general no son bien aprovechables para soportar lenguajes de Bases de Datos relacionales tradicionales. Postgres tiene algunas características que son propias del mundo de las bases de datos orientadas a objetos. De hecho, algunas Bases de Datos comerciales han incorporado recientemente características en las que Postgres fue pionera. . [16] www.elcodigok.com.ar/2008/07/que-es-postgresql/

INSTALACIÓN DE POSTGRESQL

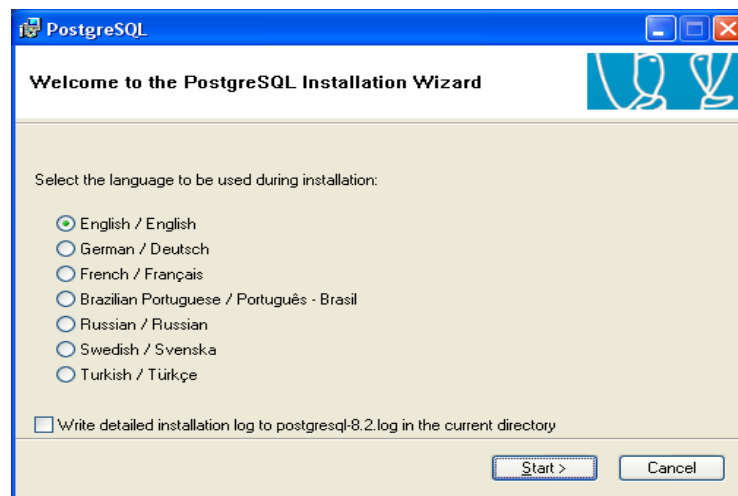
Consideraciones iniciales:

El usuario con el cual vamos a instalar el Postgresql debe tener privilegios de administrador.

Iniciando la instalación

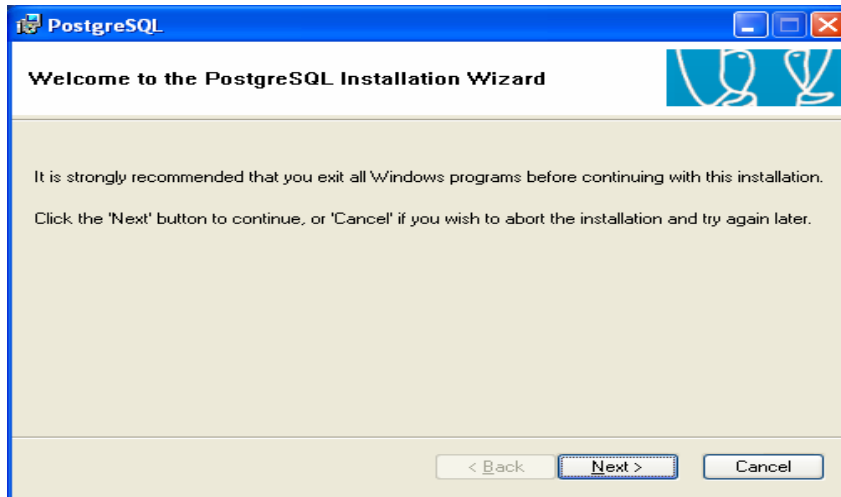
Escoger el idioma de instalación de postgres

Ilustración 17: Selección de idioma de postgres



El asistente de instalación del postgresql nos da la bienvenida.

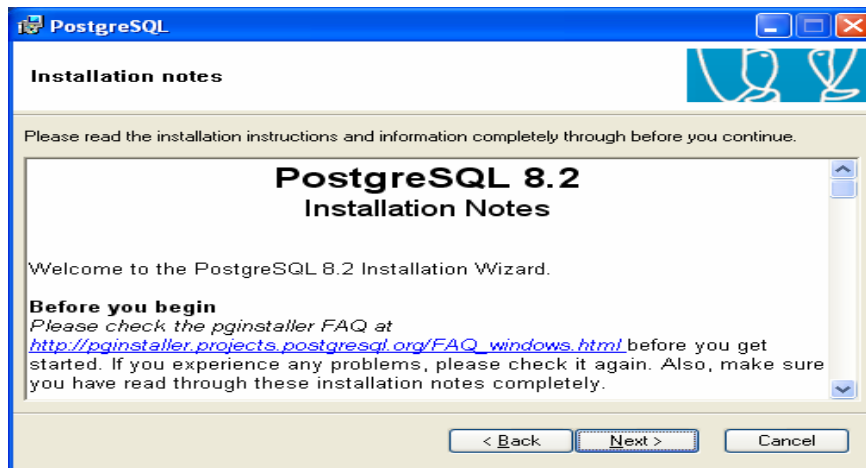
Ilustración 18: Bienvenida a la instalación de Postgres.



Es recomendable cerrar los demás programas que estemos utilizando para continuar con la instalación.

Notas de instalación

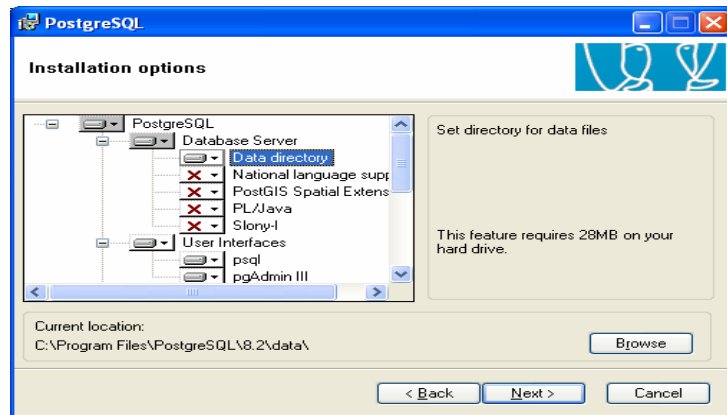
Ilustración 19: Términos y condiciones



Leer los términos y las condiciones y si está de acuerdo puede seguir con la instalación de la base de datos Postgres.

Opciones de instalación

Ilustración 20: Opciones de instalación

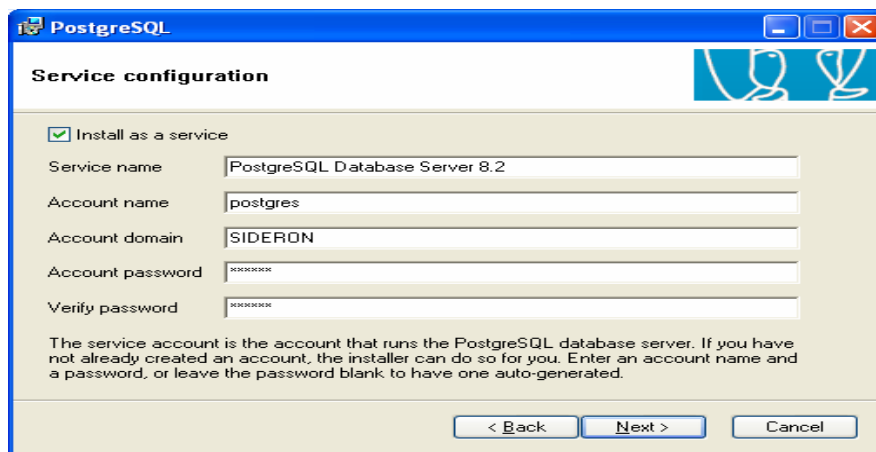


En esta ventana es importante mantener todas las opciones que aparecen predeterminadas. La opción de PostGIS debe aparecer deshabilitada para la instalación.

En la parte inferior de la ventana muestra la ubicación donde se guardaran los archivos de instalación de postgresql. Se puede buscar otra ubicación, pero es recomendable mantener la que se nos muestra.

Configuración del servicio

Ilustración 21: Configuración del servicio



Se va a realizar la configuración del servicio, en esta ventana se debe ingresar datos, algunos aparecen predeterminados por lo que es recomendable mantenerlos así. Los que no aparecen llenos se debe ingresarlos, como el password.

Al pulsar el botón de Next, se desplegará la siguiente ventana.

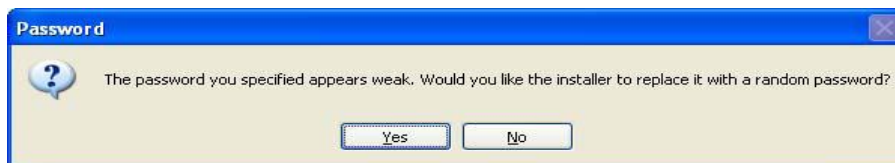
Ilustración 22: Configuración de password



El mensaje nos indica que la cuenta que hemos digitado no existe. Además nos pregunta si quiere crear la cuenta.

Al dar pulsar la opción Yes se desplegara la siguiente ventana.

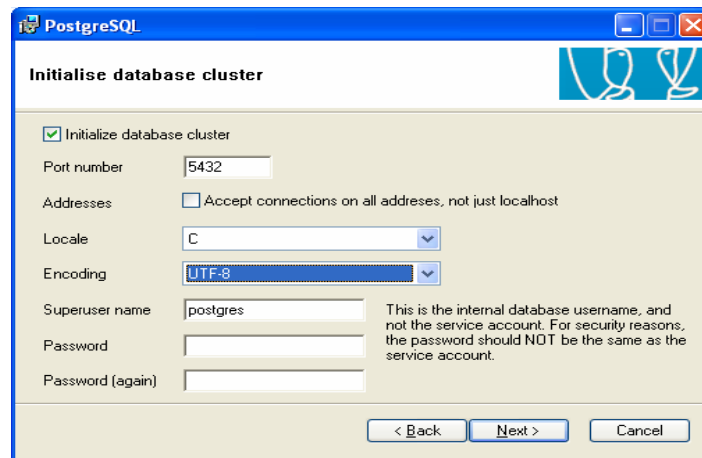
Ilustración 23: Aceptación de usuario y password



El mensaje indica que se ha introducido un password con poca seguridad. Nos presenta la opción de crearnos un password aleatorio. Preferiblemente se debe escoger que no para mantener el password que ya habíamos digitado.

Inicializar el Clúster de la base de datos

Ilustración 24: Inicialización de Clúster



Vamos a inicializar el clúster de la base de datos. En esta ventana debemos especificar lo siguiente:

Port Number: No es necesario modificar el predeterminado.

Addresses: Se debe mantener sin chequear.

Locale: C

Encoding: UTF-8

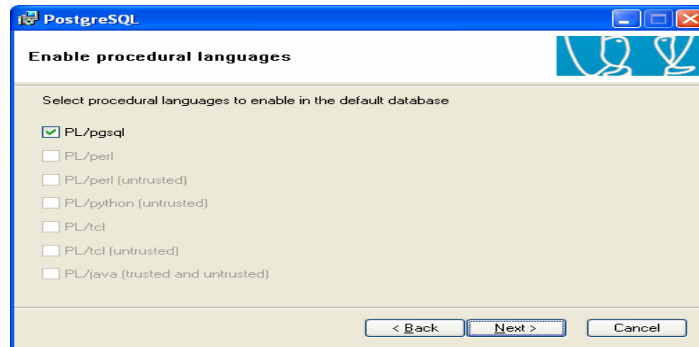
Super user name: Puede ser modificado pero preferiblemente mantener el superuser postgres

Password: Se debe introducir un password.

Habilitar lenguajes procedimentales

En esta ventana es necesario chequear la opción PL/pgsql.

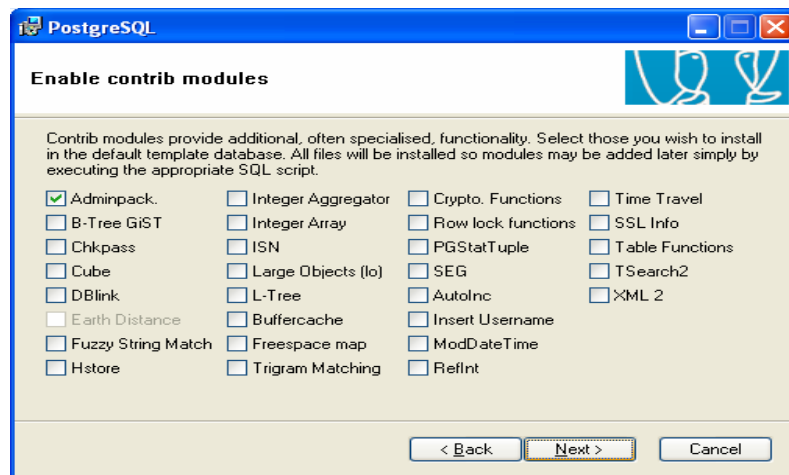
Ilustración 25: Activación de lenguajes de procedimientos



Habilitar los módulos contrib

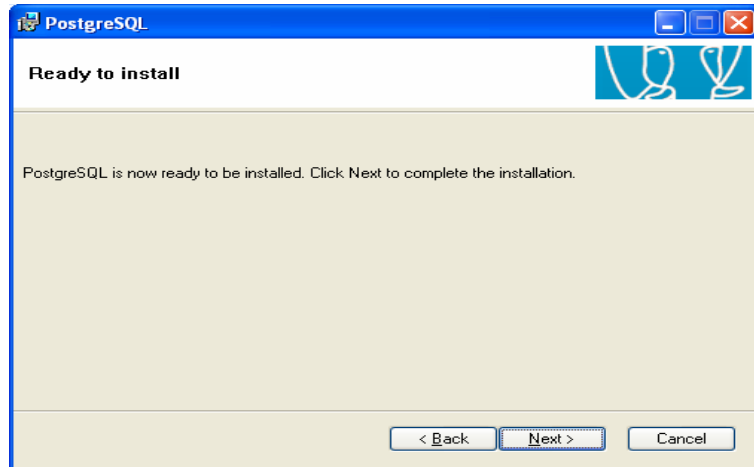
En esta ventana, se deben mantener las opciones que aparecen chequeadas predeterminadamente. No es necesario marcar ninguna otra opción.

Ilustración 26: Habilitación de módulos contrib.



Listo para instalar

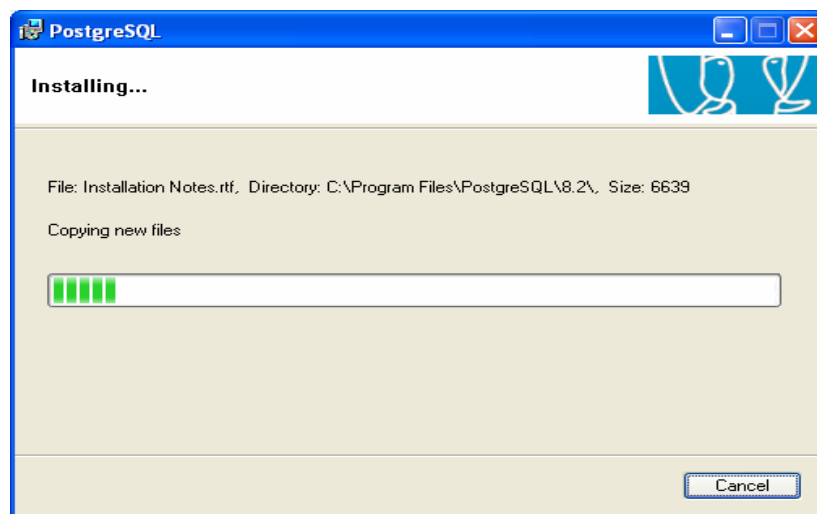
Ilustración 27: Instalación de Postgres.



Listo para instalar, solamente se debe pulsar el botón Next y la instalación dará inicio.

Instalando.

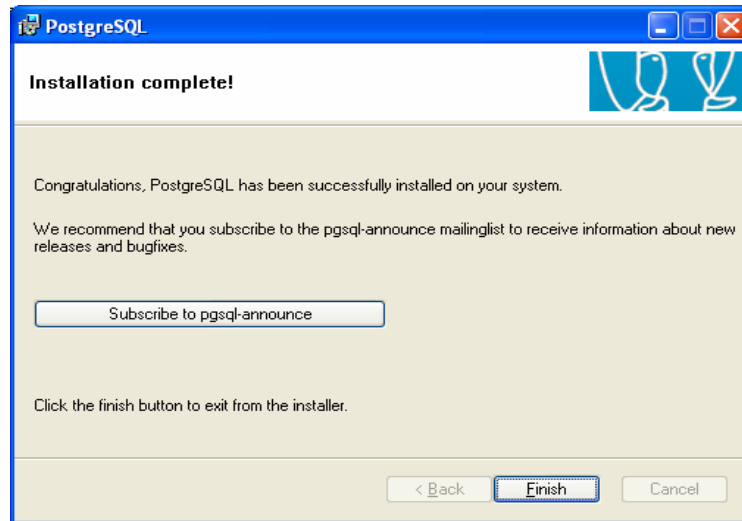
Ilustración 28: Progreso de la instalación.



En esta pantalla muestra el estado de la instalación.

Instalación completa

Ilustración 29: Finalización de la instalación.



Felicitaciones, hemos instalado exitosamente Postgresql. Es recomendable suscribirse en línea.

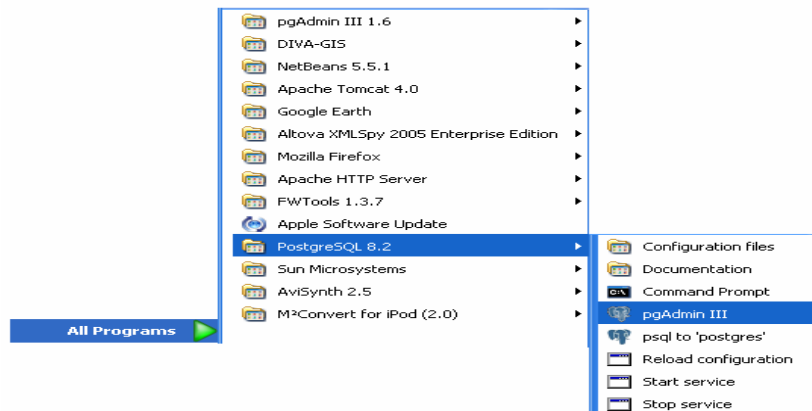
CONSIDERACIONES INICIALES

En el momento de nombrar la base de datos o el rol preferiblemente solo introducir nombres completamente en minúscula.

Ingresamos en el programa PgAdmin III

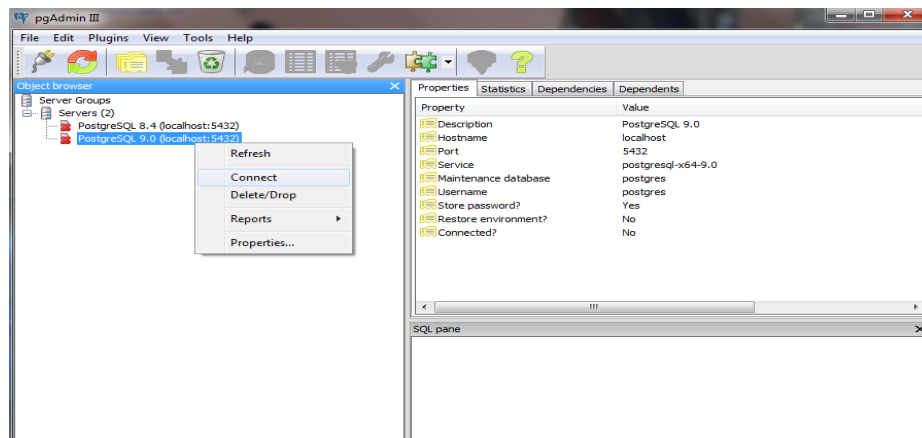
Pulsar el botón de Inicio y escoger todos los programas, luego elegir la carpeta de Postgresql y por último el programa PgAdmin III.

Ilustración 30: Localización de PgAdmin III



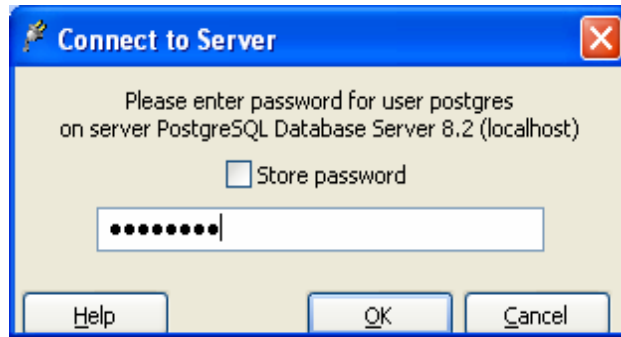
Se desplegará la siguiente interfaz del programa PgAdmin. En la parte izquierda de la pantalla observamos el servidor de PostgreSQL con una equis roja que significa que está desconectado. Para conectarlo dar clic derecho en el nombre del servidor y seleccionaremos la opción connect que se despliega en el menú contextual.

Ilustración 31: PgAdmin III.



Para acceder a la base de datos se debe digitar el password de acceso que definimos durante la instalación (Inicializar el clúster de la base de datos).

Ilustración 32: Autenticación.



2.4 ENTREGABLES DEL PROYECTO

Se continuara indicando cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables. Esta lista constituye la configuración de RUP desde la perspectiva de artefactos, y que proponemos para este proyecto.

Se debe destacar que de acuerdo a la filosofía de RUP (y de todo proceso iterativo e incremental), todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo al término del proceso se podría tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de completitud y estabilidad de los artefactos. Esto será indicado más adelante cuando se presenten los objetivos de cada iteración.

- **Caso de negocio.**

Este artefacto proporciona la información necesaria desde un punto de vista empresarial para determinar si vale la pena invertir en este proyecto o no. Para un producto de software comercial, el caso de negocio debe incluir un conjunto de suposiciones sobre el proyecto y el orden de magnitud de rendimiento de capital invertido si estas suposiciones son ciertas.

- **Plan de Desarrollo del Software**

Es el presente documento.

- **Modelo de Casos de Uso del Negocio**

Es un modelo de las funciones de negocio vistas desde la perspectiva de los actores externos (Agentes de registro, solicitantes finales, otros sistemas etc.) permite situar al sistema en el contexto organizacional haciendo énfasis en los objetivos en este ámbito. Este modelo se representa con un Diagrama de Casos de Uso usando estereotipos específicos para este modelo.

- **Modelo de Objetos del Negocio**

Modelo que describe la realización de cada caso de uso del negocio, estableciendo los actores internos, la información que manipulan y los flujos de trabajo asociados al caso de uso del negocio. Para la representación de este modelo se utilizan Diagramas de Colaboración (para mostrar actores externos, internos y las entidades (información) que manipulan, un Diagrama de Clases para mostrar gráficamente las entidades del sistema y sus relaciones, y Diagramas de Actividad para mostrar los flujos de trabajo.

- **Glosario**

Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología consensuada.

- **Modelo de Casos de Uso**

El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas. Se representa mediante Diagramas de Casos de Uso.

- **Visión**

Este documento define la visión del producto desde la perspectiva del cliente, especificando las necesidades y características del producto. Constituye una base de acuerdo en cuanto a los requisitos del sistema.

- **Especificaciones de Casos de Uso**

Para los casos de uso que lo requieran se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, post-condiciones, flujo de eventos, requisitos no-funcionales asociados. También, para casos de uso cuyo flujo de eventos sea complejo podrá adjuntarse una representación gráfica mediante un Diagrama de Actividad.

- **Especificaciones Adicionales**

Este documento capturará todos los requisitos que no han sido incluidos como parte de los casos de uso y se refieren requisitos no-funcionales globales. Dichos: requisitos incluyen: requisitos legales o normas, aplicación de estándares, requisito; de calidad del producto, tales como: confiabilidad, desempeño, etc., u otro; requisitos de ambiente, tales como: sistema operativo, requisitos de compatibilidad etc.

- **Prototipos de Interfaz de Usuario**

Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema. Estos prototipos se realizarán como: dibujos a mano en papel, dibujos con alguna herramienta gráfica o prototipos ejecutables interactivos, siguiendo ese orden de acuerdo al avance del proyecto. Sólo los de este último tipo serán entregados al final de la fase de Elaboración, los otros serán desechados.

- **Modelo de Análisis y Diseño**

Este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación), de acuerdo al avance del proyecto.

- **Modelo de Datos**

Previendo que la persistencia de la información del sistema será soportada por un a base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un Diagrama de Clases (donde se utiliza un profile UML para Modelado de Datos, para conseguir la representación de tablas, claves, etc.).

- **Modelo de Implementación**

Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema. (Este modelo es sólo una versión preliminar al final de la fase de Elaboración, posteriormente tiene bastante refinamiento).

- **Solicitud de Cambio**

Los cambios propuestos para los artefactos se formalizan mediante este documento. Mediante este documento se hace un seguimiento de los defectos detectados, solicitud de mejoras o cambios en los requisitos del producto. Así se provee un registro de decisiones de cambios, de su evaluación de impacto, y se asegura que éstos sean conocidos por el equipo de desarrollo.

- **Plan de Iteración**

Es un conjunto de actividades y tareas ordenadas temporalmente, con recursos asignados, dependencias entre ellas. Se realiza para cada iteración, y para todas las fases.

- **Evaluación de Iteración**

Este documento incluye la evaluación de los resultados de cada iteración, el grado en el cual se han conseguido los objetivos de la iteración, las lecciones aprendidas y los cambios a ser realizados.

- **Lista de Riesgos**

Este documento incluye una lista de los riesgos conocidos y vigentes en el proyecto, ordenados en orden decreciente de importancia y con acciones específicas de contingencia o para su mitigación.

- **Manual de Instalación**

Este documento incluye las instrucciones para realizar la instalación del producto.

- **Material de Apoyo al Usuario Final**

Corresponde a un conjunto de documentos y facilidades de uso del sistema, incluyendo: Guías del Usuario, Guías de Operación, Guías de Mantenimiento y Sistema de Ayuda en Línea

- **Producto**

Los ficheros del producto empaquetados y almacenados en un CD con los mecanismos apropiados para facilitar su instalación. El producto, a partir de la primera iteración de la fase de Construcción es desarrollado incremental e iterativamente.

- **Documento de arquitectura de software**

Este producto de trabajo proporciona una visión general arquitectónica completa del sistema, mediante una serie de vistas arquitectónicas diferentes para representar diferentes aspectos del sistema.

2.5 ORGANIZACIÓN DEL PROYECTO

2.5.1 PARTICIPANTES DEL PROYECTO.

Responsable del Proyecto. Como este es un proyecto de tesis el responsable de que el sistema tenga su correcto funcionamiento es el Egresado Joffre Paolo Aguas Rodríguez.

Responsables de Control y Seguimiento, Se designa al Director de Tesis Ingeniero Rodrigo Naranjo.

Programador Será el egresado Joffre Paolo Aguas Rodríguez en sistemas que desarrollará el proyecto.

Analista de Sistemas. Será el egresado Joffre Paolo Aguas Rodríguez quien desarrollará el proyecto.

2.6 ADMINISTRACIÓN DE PROYECTOS

2.6.1 PLAN DEL PROYECTO

En esta sección se presenta la organización en fases e iteraciones y el calendario del proyecto.

2.6.2 PLAN DE FASES

El desarrollo se llevara a cabo en base a fases con una o más iteraciones en cada una. La siguiente tabla muestra los tiempos y el número de iteraciones de cada fase.

Tabla 1.- Plan de fases

FASE	NRO. ITERACIONES	DURACIÓN
Fase de Inicio	1	3 semanas
Fase de Elaboración	1	6 semanas
Fase de Construcción	3	15 semanas
Fase de Transición	1	3 semanas

Los hitos de cada fase se muestran en la siguiente tabla:

Tabla 2.- Descripción de Hito

DESCRIPCIÓN	HITO
-------------	------

Fase de Inicio	En esta fase se desarrollara los requisitos del sistema o producto final desde la perspectiva del usuario, los principales casos de uso serán identificados, caso de negocio, realización del glosario, identificación inicial de riesgos y el plan de desarrollo marcan el final de esta fase.
Fase de Elaboración	En esta fase se analizan los requisitos y se desarrolla un prototipo de arquitectura, al final de esta fase todos los casos de uso que corresponden a requisitos que serán implementados en la fase de construcción deben estar analizados y diseñados. La revisión y aceptación del prototipo de la arquitectura del sistema marca el final de esta fase.
Fase de Construcción	Durante esta fase se termina de analizar y diseñar todos los casos de uso. El sistema o producto final se construye en 3 iteraciones las cuales son aplicadas a pruebas. Se comienza la elaboración del material de apoyo del usuario. Esta fase termina con la capacidad operacional del producto listo para ser entregado al usuario para pruebas.
Fase de Transición	En esta fase de transición se instala el sistema en el PC principal de la oficina de Fedeligas Imbabura el mismo que estará sujeto a cambios según se lo requieran los usuarios finales. Esta fase termina con la entrega de toda la documentación del proyecto con el material de apoyo al usuario.

2.6.2.1 OBJETIVOS DE CADA ITERACIÓN

A continuación se detalla en la siguiente tabla los objetivos de cada iteración.

Tabla 3.- Objetivos de cada iteración.

FASE(NÚMERO DE ITERACIONES)	OBJETIVOS DE CADA ITERACIÓN.
Inicio(1)	Establecer los requisitos del sistema, casos de uso relacionados a todos los procesos del sistema.
Elaboración(1)	Completar prototipo de interfaz de usuario, diagramas de colaboración y actividades. Arquitectura que se va a utilizar. Elaboración de la base de datos y los diagramas de clases.
Construcción(3)	Desarrollo y pruebas del sistema, manuales de usuario.
Transición(1)	Instalación del sistema, Carga de la base de datos y puesta en marcha del sistema.

2.6.2.2 ENTREGAS

Se entregaran en la fase de construcción lo siguiente.

Tabla 4.- Entregas

Versión full	Será realizado a partir de la primera iteración hasta la culminación de tercera iteración para ser sometido a pruebas de usuarios en red.
--------------	---

2.6.2.3 CRONOGRAMA DEL PROYECTO

Aquí se presenta un calendario de las principales tareas del proyecto incluyendo las fases de Inicio y Elaboración. El proceso de implementación RUP se caracteriza por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del proyecto con lo cual los artefactos son elaborados y generados muy tempranamente en el proyecto pero se desarrollan en mayor o menor grado de acuerdo a la fase e iteración del proyecto.

Ilustración 33.- Cronograma de trabajo.

N	Actividad	Duración (Semanas)	Inicio
1	Investigación	6	7 marzo
2	Capacitación	8	15 Abril
3	Desarrollo	10	18 Junio
4	Pruebas	3	4 Agosto
5	Instalación	1	3 Septiembre
6	Documentación	8	24 Septiembre

Se puede hacer mención de las tres características esenciales que definen al RUP:

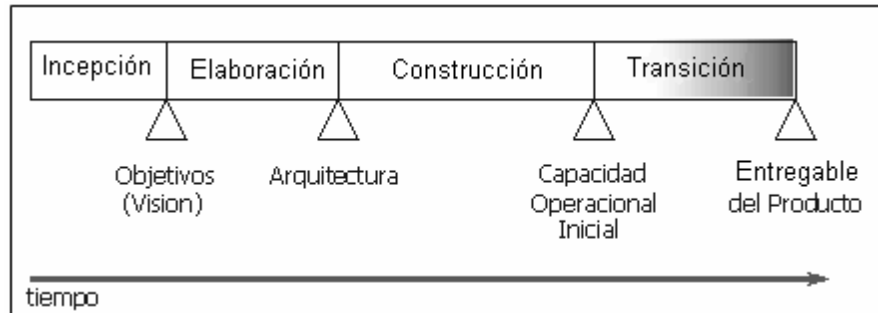
- **Proceso Dirigido por los Casos de Uso:** Con esto se refiere a la utilización de los Casos de Uso para el desenvolvimiento y desarrollo de las disciplinas con los artefactos, roles y actividades necesarias. Los Casos de Uso son la base para la implementación de las fases y disciplinas del RUP.

Un Caso de Uso es una secuencia de pasos a seguir para la realización de un fin o propósito, y se relaciona directamente con los requerimientos, ya que un Caso de Uso es la secuencia de pasos que conlleva la realización e implementación de un Requerimiento planteado por el Cliente.

- **Proceso Iterativo e Incremental:** Es el modelo utilizado por RUP para el desarrollo de un proyecto de software. Este modelo plantea la implementación del proyecto a realizar en Iteraciones, con lo cual se pueden definir objetivos por cumplir en cada iteración y así poder ir completando todo el proyecto iteración por iteración, con lo cual se tienen varias ventajas, entre ellas se puede mencionar la de tener pequeños avances del proyectos que son entregables al cliente el cual puede probar mientras se está desarrollando otra iteración del proyecto, con lo cual el proyecto va creciendo hasta completarlo en su totalidad. Este proceso se explica más adelante a detalle.

- **Proceso Centrado en la Arquitectura:** Define la Arquitectura de un sistema, y una arquitectura ejecutable construida como un prototipo evolutivo. Arquitectura de un sistema es la organización o estructura de sus partes más relevantes. Una arquitectura ejecutable es una implementación parcial del sistema, construida para demostrar algunas funciones y propiedades.

Ilustración 34.- Fases del RUP



El ciclo de vida del software del RUP se descompone en cuatro fases secuenciales (Ilustración 34). En cada extremo de una fase se realiza una evaluación (actividad: Revisión del ciclo de vida de la finalización de fase) para determinar si los objetivos de la fase se han cumplido. Una evaluación satisfactoria permite que el proyecto se mueva a la próxima fase.

2.6.2.4 PLANEANDO LAS FASES

El ciclo de vida consiste en una serie de ciclos, cada uno de los cuales produce una nueva versión del producto, cada ciclo está compuesto por fases y cada una de estas fases está compuesta por un número de iteraciones, estas fases son:

1. Concepción, Inicio o Estudio de oportunidad
 - Define el ámbito y objetivos del proyecto.
 - Se define la funcionalidad y capacidades del producto.

2. Elaboración
 - Tanto la funcionalidad, el dominio del problema se estudian en profundidad.
 - Se define una arquitectura básica.
 - Se planifica el proyecto considerando recursos disponibles.

3. Construcción

- El producto se desarrolla a través de iteraciones donde cada iteración involucra tareas de análisis, diseño e implementación.
- Las fases de estudio y análisis sólo dieron una arquitectura básica conforme se construye (se permiten cambios en la estructura).
- Gran parte del trabajo es programación y pruebas.
- Se documenta tanto el sistema construido como el manejo del mismo.

4. Transición

- Se libera el producto y se entrega al usuario para un uso real.
- Se incluyen tareas de empaquetado atractivo, instalación, configuración, entrenamiento, soporte, mantenimiento, etc.
- Los manuales de usuario se completan y refinan con la información anterior.

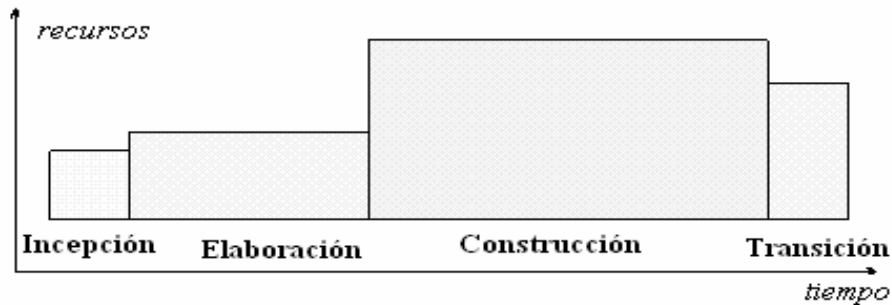
Aunque esto varía considerablemente dependiendo del proyecto, un ciclo de desarrollo inicial típico para un proyecto de tamaño mediano debe anticipar la distribución siguiente el esfuerzo y horario:

Tabla 5.- Esfuerzo-horario contra fases del RUP

	Concepción	Elaboración	Construcción	Transición
Esfuerzo	~5 %	20 %	65 %	10%
Horario	10 %	30 %	50 %	10%

Lo cual se puede representar gráficamente como se muestra en la ilustración 35.

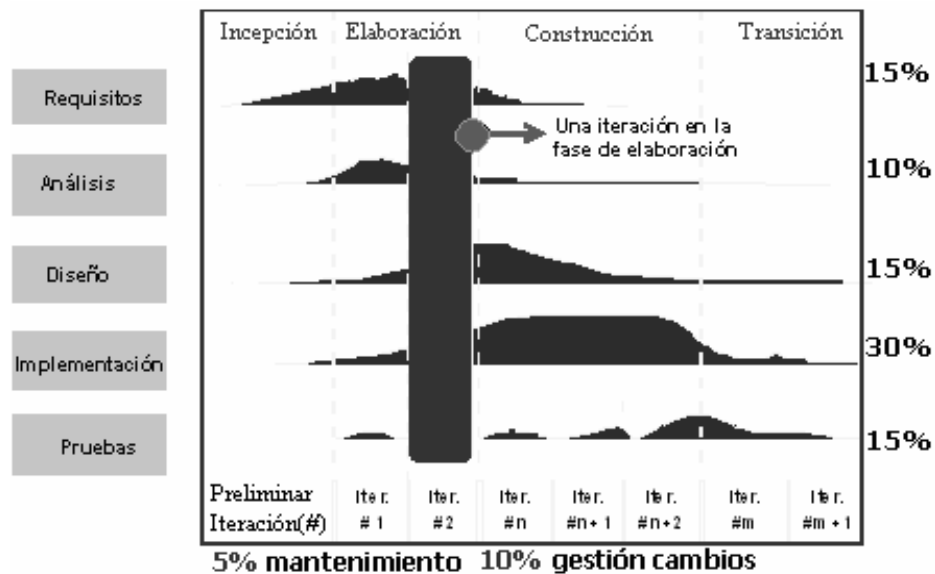
Ilustración 35.- Recursos utilizados en las fases del RUP en el tiempo.



En un ciclo evolutivo, las fases de concepción y elaboración serían considerablemente más pequeñas. Algunas herramientas que pueden automatizar una cierta porción del esfuerzo de la fase de Construcción pueden atenuar esto, haciendo que la fase de construcción sea mucho más pequeña que las fases de concepción y elaboración juntas.

Cada paso con las cuatro fases produce una generación del software. A menos que el producto "muera", se desarrollará nuevamente repitiendo la misma secuencia las fases de concepción, elaboración, construcción y transición, pero con diversos énfasis cada fase.

Ilustración 36.- Esfuerzo respecto de los flujos de trabajo



2.6.3 CALENDARIO DEL PROYECTO

Para este proyecto se ha establecido el siguiente calendario. La fecha de aprobación

indica cuándo el artefacto en cuestión tiene un estado de completitud suficiente para someterse a revisión y aprobación, pero esto no quita la posibilidad de su posterior refinamiento y cambios.

Tabla 6.- Calendario del proyecto Fase de Inicio.

Disciplinas / Artefactos generados o modificados durante la Fase de Inicio	Comienzo	Aprobación
REQUISITOS		
Glosario	Semana 1 01/03/2011	Siguiente Fase
Visión	Semana 1 01/03/2011	Semana 3 18/03/2011
Modelo de Casos de Uso	Semana 3 16/03/2011	Siguiente Fase
Especificaciones Adicionales	Semana 3 17/03/2011	Siguiente Fase
ANÁLISIS		
Modelo de Objetos	Semana 3 17/03/2011	Siguiente Fase
Especificaciones de casos de uso.	Semana 3 17/03/2011	Siguiente Fase
Modelo Entidad - Relación	Semana 2 08/03/2011	Siguiente Fase
Documento de arquitectura del software.	Semana 3 17/03/2011	Siguiente Fase
DISEÑO		
Mapa de navegación	Semana 3 17/03/2011	Siguiente Fase
Prototipos de interfaz (Plantillas).	Semana 3 17/03/2011	Siguiente Fase
Modelo de diseño de clases.	Semana 1 04/03/2011	Siguiente Fase
Modelo de datos (modelo relacional.)	Semana 1 04/03/2011	Siguiente Fase
IMPLEMENTACIÓN		

Modelo de despliegue	Semana 3 17/03/2011	Siguiente Fase
Modelo de Implementación	Semana 3 17/03/2011	Siguiente Fase
Integración del sistema	Semana 4 24/03/2011	Siguiente Fase
Manual de usuario	Semana 10 30/04/2011	Siguiente Fase
PRUEBAS		
Casos de Pruebas Funcionales	Semana 4 24/03/2011	Siguiente Fase
Solicitud de cambios	Semana 4 24/03/2011	Siguiente Fase
GESTIÓN DE CAMBIOS Y CONFIGURACIÓN		
Depósito de proyecto	Semana 3 17/03/2011	Siguiente Fase
Solicitud de cambios	Semana 8 14/04/2011	Siguiente Fase
GESTIÓN DEL PROYECTO		
Lista de riesgos	Semana 1 04/03/2011	Semana 3 17/03/2011
Desarrollar el caso de negocio.	Semana 1 04/03/2011	Semana 3 17/03/2011
Plan de desarrollo de software.	Semana 1 04/03/2011	Siguiente Fase
Plan de iteración.	Semana 2 08/03/2011	Semana 3 17/03/2011
Evaluación de iteración.	Semana 10 30/04/2011	Siguiente Fase
AMBIENTE		
Personalizar el proceso de desarrollo.	Semana 1 04/03/2011	Siguiente Fase
Guía de artefactos.	Semana 27 04/08/2011	Siguiente Fase

Tabla 7.- Calendario de proyectos. Fase de Elaboración.

Disciplinas / Artefactos generados o modificados durante la	Comienzo	Aprobación
---	----------	------------

Fase de Elaboración		
REQUISITOS		
Glosario	Semana 1 01/03/2011	Semana 8 14/04/2011
Visión	Semana 1 01/03/2011	Aprobado
Modelo de Casos de Uso	Semana 3 16/03/2011	Semana 7 07/04/2011
Especificaciones Adicionales	Semana 3 17/03/2011	Semana 7 07/04/2011
ANÁLISIS		
Modelo de Objetos	Semana 3 17/03/2011	Siguiente Fase
Especificaciones de casos de uso.	Semana 3 17/03/2011	Siguiente Fase
Modelo Entidad - Relación	Semana 2 08/03/2011	Semana 9 19/04/2011
Documento de arquitectura del software.	Semana 3 17/03/2011	Semana 9 19/04/2011
DISEÑO		
Mapa de navegación	Semana 3 17/03/2011	Semana 9 19/04/2011
Prototipos de interfaz (Plantillas).	Semana 3 17/03/2011	Semana 9 19/04/2011
Modelo de diseño de clases.	Semana 1 04/03/2011	Semana 7 07/04/2011
Modelo de datos (modelo relacional.)	Semana 1 04/03/2011	Semana 7 07/04/2011
IMPLEMENTACIÓN		
Modelo de despliegue	Semana 3 17/03/2011	Siguiente Fase
Modelo de Implementación	Semana 3 17/03/2011	Siguiente Fase
Integración del sistema	Semana 4 24/03/2011	Siguiente Fase

Manual de usuario	Semana 10 30/04/2011	Siguiente Fase
-------------------	-------------------------	----------------

PRUEBAS		
Casos de Pruebas Funcionales	Semana 4 24/03/2011	Siguiente Fase
Solicitud de cambios	Semana 4 24/03/2011	Siguiente Fase
GESTIÓN DE CAMBIOS Y CONFIGURACIÓN		
Depósito de proyecto	Semana 3 17/03/2011	Semana 6 02/04/2011
Solicitud de cambios	Semana 8 14/04/2011	Siguiente Fase
GESTIÓN DEL PROYECTO		
Lista de riesgos	Semana 1 04/03/2011	Aprobado
Desarrollar el caso de negocio.	Semana 1 04/03/2011	Aprobado
Plan de desarrollo de software.	Semana 1 04/03/2011	Semana 5 29/03/2011
Plan de iteración.	Semana 2 08/03/2011	Aprobado
Evaluación de iteración.	Semana 10 30/04/2011	Siguiente Fase
AMBIENTE		
Personalizar el proceso de desarrollo.	Semana 1 04/03/2011	Siguiente Fase
Guía de artefactos.	Semana 27 04/08/2011	Siguiente Fase

Tabla 8.- calendario del proyecto. Fase de construcción Iteración I.

Disciplinas / Artefactos generados o modificados durante la Fase de Construcción (Iteración 1)	Comienzo	Aprobación
---	-----------------	-------------------

REQUISITOS		
Glosario	Semana 1 01/03/2011	Aprobado
Visión	Semana 1 01/03/2011	Aprobado
Modelo de Casos de Uso	Semana 3 16/03/2011	Aprobado
Especificaciones Adicionales	Semana 3 17/03/2011	Aprobado
ANÁLISIS		
Modelo de Objetos	Semana 3 17/03/2011	Semana 13 16/05/2011
Especificaciones de casos de uso.	Semana 3 17/03/2011	Semana 13 16/05/2011
Modelo Entidad - Relación	Semana 2 08/03/2011	Aprobado
Documento de arquitectura del software.	Semana 3 17/03/2011	Aprobado
DISEÑO		
Mapa de navegación	Semana 3 17/03/2011	Aprobado
Prototipos de interfaz (Plantillas).	Semana 3 17/03/2011	Aprobado
Modelo de diseño de clases.	Semana 1 04/03/2011	Aprobado
Modelo de datos (modelo relacional.)	Semana 1 04/03/2011	Aprobado
IMPLEMENTACIÓN		
Modelo de despliegue	Semana 3 17/03/2011	Semana 11 01/05/2011
Modelo de Implementación	Semana 3 17/03/2011	Semana 12 19/05/2011
Integración del sistema	Semana 4 24/03/2011	Semana 12 19/05/2011
Manual de usuario	Semana 10 30/04/2011	Siguiente Fase

PRUEBAS		
Casos de Pruebas Funcionales	Semana 4 24/03/2011	Siguiente Fase
Solicitud de cambios	Semana 4 24/03/2011	Siguiente Fase
GESTIÓN DE CAMBIOS Y CONFIGURACIÓN		
Depósito de proyecto	Semana 3 17/03/2011	Aprobado
Solicitud de cambios	Semana 8 14/04/2011	Siguiente Fase
GESTIÓN DEL PROYECTO		
Lista de riesgos	Semana 1 04/03/2011	Aprobado
Desarrollar el caso de negocio.	Semana 1 04/03/2011	Aprobado
Plan de desarrollo de software.	Semana 1 04/03/2011	Aprobado
Plan de iteración.	Semana 2 08/03/2011	Aprobado
Evaluación de iteración.	Semana 10 30/04/2011	Siguiente Fase
AMBIENTE		
Personalizar el proceso de desarrollo.	Semana 1 04/03/2011	Siguiente Fase
Guía de artefactos.	Semana 27 04/08/2011	Siguiente Fase

Tabla 9.- calendario del proyecto. Fase de construcción Iteración II.

Disciplinas / Artefactos generados o modificados durante la Fase de Construcción (Iteración 2)	Comienzo	Aprobación
REQUISITOS		
Glosario	Semana 1 01/03/2011	Aprobado
Visión	Semana 1 01/03/2011	Aprobado
Modelo de Casos de Uso	Semana 3 16/03/2011	Aprobado
Especificaciones Adicionales	Semana 3 17/03/2011	Aprobado
ANÁLISIS		
Modelo de Objetos	Semana 3 17/03/2011	Aprobado
Especificaciones de casos de uso.	Semana 3 17/03/2011	Aprobado
Modelo Entidad - Relación	Semana 2 08/03/2011	Aprobado
Documento de arquitectura del software.	Semana 3 17/03/2011	Aprobado
DISEÑO		
Mapa de navegación	Semana 3 17/03/2011	Aprobado
Prototipos de interfaz (Plantillas).	Semana 3 17/03/2011	Aprobado
Modelo de diseño de clases.	Semana 1 04/03/2011	Aprobado
Modelo de datos (modelo relacional.)	Semana 1 04/03/2011	Aprobado
IMPLEMENTACIÓN		
Modelo de despliegue	Semana 3 17/03/2011	Aprobado
Modelo de Implementación	Semana 3 17/03/2011	Aprobado

Integración del sistema	Semana 4 24/03/2011	Aprobado
Manual de usuario	Semana 10 30/04/2011	Siguiente Fase
PRUEBAS		
Casos de Pruebas Funcionales	Semana 4 24/03/2011	Siguiente Fase
Solicitud de cambios	Semana 4 24/03/2011	Siguiente Fase
GESTIÓN DE CAMBIOS Y CONFIGURACIÓN		
Depósito de proyecto	Semana 3 17/03/2011	Aprobado
Solicitud de cambios	Semana 8 14/04/2011	Siguiente Fase
GESTIÓN DEL PROYECTO		
Lista de riesgos	Semana 1 04/03/2011	Aprobado
Desarrollar el caso de negocio.	Semana 1 04/03/2011	Aprobado
Plan de desarrollo de software.	Semana 1 04/03/2011	Aprobado
Plan de iteración.	Semana 2 08/03/2011	Aprobado
Evaluación de iteración.	Semana 10 30/04/2011	Siguiente Fase
AMBIENTE		
Personalizar el proceso de desarrollo.	Semana 1 04/03/2011	Siguiente Fase
Guía de artefactos.	Semana 27 04/08/2011	Siguiente Fase

Tabla 10.- calendario del proyecto. Fase de construcción Iteración III.

Disciplinas / Artefactos generados o modificados durante la Fase de Construcción (Iteración 3)	Comienzo	Aprobación
REQUISITOS		
Glosario	Semana 1 01/03/2011	Aprobado
Visión	Semana 1 01/03/2011	Aprobado
Modelo de Casos de Uso	Semana 3 16/03/2011	Aprobado
Especificaciones Adicionales	Semana 3 17/03/2011	Aprobado
ANÁLISIS		
Modelo de Objetos	Semana 3 17/03/2011	Aprobado
Especificaciones de casos de uso.	Semana 3 17/03/2011	Aprobado
Modelo Entidad - Relación	Semana 2 08/03/2011	Aprobado
Documento de arquitectura del software.	Semana 3 17/03/2011	Aprobado
DISEÑO		
Mapa de navegación	Semana 3 17/03/2011	Aprobado
Prototipos de interfaz (Plantillas).	Semana 3 17/03/2011	Aprobado
Modelo de diseño de clases.	Semana 1 04/03/2011	Aprobado
Modelo de datos (modelo relacional.)	Semana 1 04/03/2011	Aprobado
IMPLEMENTACIÓN		
Modelo de despliegue	Semana 3 17/03/2011	Aprobado
Modelo de Implementación	Semana 3 17/03/2011	Aprobado

Integración del sistema	Semana 4 24/03/2011	Aprobado
Manual de usuario	Semana 10 30/04/2011	Siguiente Fase
PRUEBAS		
Casos de Pruebas Funcionales	Semana 4 24/03/2011	Siguiente Fase
Solicitud de cambios	Semana 4 24/03/2011	Semana 24 16/07/2011
GESTIÓN DE CAMBIOS Y CONFIGURACIÓN		
Depósito de proyecto	Semana 3 17/03/2011	Aprobado
Solicitud de cambios	Semana 8 14/04/2011	Semana 24 16/07/2011
GESTIÓN DEL PROYECTO		
Lista de riesgos	Semana 1 04/03/2011	Aprobado
Desarrollar el caso de negocio.	Semana 1 04/03/2011	Aprobado
Plan de desarrollo de software.	Semana 1 04/03/2011	Aprobado
Plan de iteración.	Semana 2 08/03/2011	Aprobado
Evaluación de iteración.	Semana 10 30/04/2011	Siguiente Fase
AMBIENTE		
Personalizar el proceso de desarrollo.	Semana 1 04/03/2011	Siguiente Fase
Guía de artefactos.	Semana 27 04/08/2011	Siguiente Fase

2.6.4 SEGUIMIENTO Y CONTROL DEL PROYECTO

2.6.4.1 CONTROL DE PLAZOS

El calendario del proyecto tendrá un seguimiento y evaluación semanal por el jefe de proyecto.

2.6.4.2 CONTROL DE CALIDAD

Los defectos detectados en las revisiones y formalizados también en una Solicitud de Cambio tendrán un seguimiento para asegurar la conformidad respecto de la solución de dichas deficiencias. Para la revisión de cada artefacto y su correspondiente garantía de calidad se utilizarán las guías de revisión incluidas en RUP.

2.6.4.3 GESTIÓN DE RIESGOS

A partir de la fase de Inicio se mantendrá una lista de riesgos asociados al proyecto y de las acciones establecidas como estrategia para mitigarlos o acciones de contingencia. Esta lista será evaluada al menos una vez en cada iteración.

2.6.4.4 GESTIÓN DE CONFIGURACIÓN

Se realizará una gestión de configuración para llevar un registro de los artefactos generados y sus versiones. También se incluirá la gestión de las Solicitudes de Cambio y de las modificaciones que éstas produzcan, informando y publicando dichos cambios para que sean accesibles a todo los participantes en el proyecto. Al final de cada iteración se establecerá un registro del estado de cada artefacto, estableciendo una versión, la cual podrá ser modificada sólo por una Solicitud de Cambio aprobada.

3. DISEÑO DE LA SOLUCIÓN.

3.1 MODELO DE CASOS DE USO

3.1.1 INTRODUCCIÓN.

Muestra las interrelaciones entre el sistema y su ambiente, además sirve como un contrato entre el cliente y los diseñadores. Es considerado esencial al iniciar las actividades de análisis, diseño y prueba; este modelo es realizado en la disciplina de Requerimientos.

3.1.2 PROPÓSITO.

Modelar la funcionalidad del sistema agrupándola en descripciones de acciones ejecutadas por un sistema para obtener un resultado.

3.1.3 ALCANCE.

Los Casos de Uso son la base para la implementación de las fases y disciplinas del RUP. Es una secuencia de pasos a seguir para la realización de un fin o propósito, y se relaciona directamente con los requerimientos, ya que un Caso de Uso es la secuencia de pasos que conlleva la realización e implementación de un Requerimiento planteado por el Cliente.

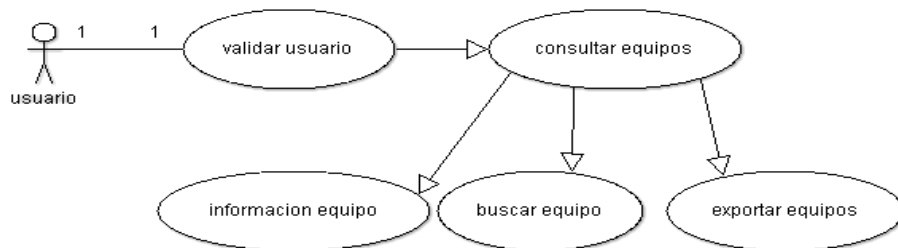
3.1.4 PANORAMA GENERAL.

A continuación se detalla los casos de uso, se empieza con el nombre del caso de uso y se da una breve descripción indicando que función cumple.

3.1.5 CASO DE USO OBTENER INFORMACIÓN DE EQUIPO

Indica la manera como el usuario podrá obtener información de un equipo de fútbol.

Ilustración 37.- Caso de uso obtener información de equipo.



Breve explicación del caso de uso mencionado.

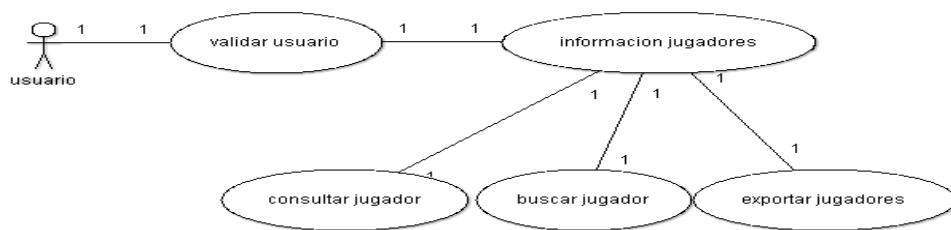
Tabla 11.- Caso de uso obtener información de equipo de fútbol.

CASO DE USO	DESCRIPCIÓN DE CASO DE USO
Validar usuario	Se encarga de validar los usuarios para que puedan ingresar a módulos a ellos permitidos.
Consultar equipos	Permite seleccionar el equipo, realizar tareas sobre equipos de fútbol.
Información equipo	Permite verificar la información completa del equipo de fútbol.
Buscar equipo	Permite realizar la búsqueda de equipos de fútbol que el usuario necesite consultar.
Exportar equipo	Realiza la exportación de todos los equipos de fútbol a un archivo de Microsoft Excel a manera de reporte.

3.1.6 CASO DE USO OBTENER INFORMACIÓN DE JUGADOR

Cumple con la función de indicar la manera como el usuario podrá obtener información de un determinado jugador de fútbol.

Ilustración 38.- Caso de uso obtener información de jugador.



A continuación se muestra una breve explicación del caso de uso mencionado.

Tabla 12.- Caso de uso obtener información de jugadores de fútbol.

CASO DE USO	DESCRIPCIÓN DE CASO DE USO
Validar usuario	Se encarga de validar los usuarios para que puedan ingresar a módulos a ellos permitidos.
Consultar jugador	Permite seleccionar el jugador, realizar tareas sobre jugador de fútbol.
Información jugadores	Permite verificar la información completa del jugador de fútbol.
Buscar jugador	Permite realizar la búsqueda de jugadores de fútbol que el usuario necesite consultar.
Exportar jugadores	Realiza la exportación de todos los jugadores de fútbol a un archivo de Microsoft Excel a manera de reporte.

3.1.7 CASO DE USO OBTENER INFORMACIÓN DE LIGA BARRIAL

Cumple con la función de indicar la manera como el usuario podrá obtener información de una determinada liga barrial.

Ilustración 39.- Caso de uso obtener información de liga barrial.



A continuación se muestra una breve explicación del caso de uso mencionado.

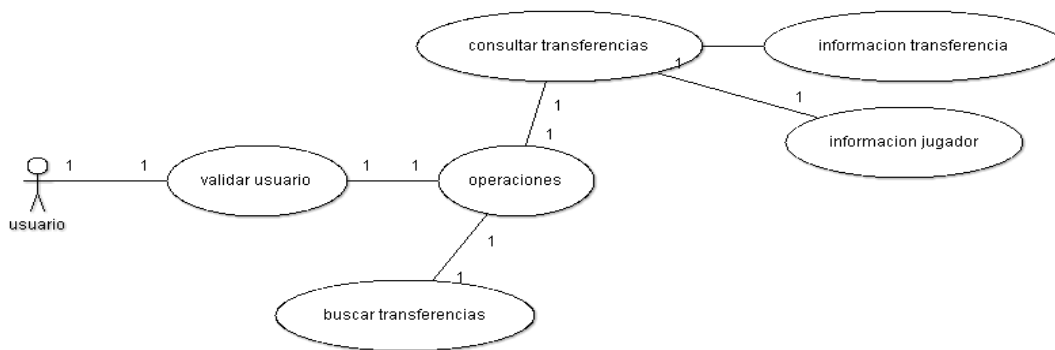
Tabla 13.- Caso de uso obtener información de jugadores de fútbol.

CASO DE USO	DESCRIPCIÓN DE CASO DE USO
Validar usuario	Se encarga de validar los usuarios para que puedan ingresar a módulos a ellos permitidos.
Consultar liga	Permite seleccionar la liga barrial.
Información liga	Permite verificar la información completa de las distintas ligas barriales.

3.1.8 CASO DE USO OBTENER INFORMACIÓN DE TRANSFERENCIA

Cumple con la función de indicar la manera como el usuario podrá obtener información de una determinada transferencia de un jugador.

Ilustración 40.- Caso de uso obtener información de transferencias.



A continuación se muestra una breve explicación del caso de uso mencionado.

Tabla 14.- Caso de uso obtener información de transferencias de jugadores.

CASO DE USO	DESCRIPCIÓN DE CASO DE USO
Validar usuario	Se encarga de validar los usuarios para que puedan ingresar a módulos a ellos permitidos.
Consultar transferencia	Permite seleccionar el jugador, realizar tareas sobre transferencias de jugadores.
Información transferencia	Permite verificar la información completa del equipo anterior y actual al que pertenece el jugador y el tipo de pase.
Información jugador	Permite verificar la información completa del jugador de fútbol.
Operaciones	Permite seleccionar transferencias y jugadores para ser tratados según lo requieran los usuarios.
Buscar transferencias	Permite realizar la búsqueda de una determinada transferencia realizada a un jugador de fútbol.

3.2 ESPECIFICACIÓN DE CASOS DE USO

Esta sección contiene las principales especificaciones de los casos de uso del sistema los cuales serán detallados a continuación.

Tabla 15.- Resumen casos de uso

NRO.	CASO DE USO
1	Asignación de usuarios al sistema.
2	Información de ligas.
3	Información de equipos.
4	Información de jugadores.
5	Noticias.
6	Información transferencias.
7	Modificar jugadores.
8	Modificar equipos
9	Modificar usuarios del sistema.
10	Ingreso datos ligas.
11	Buscar transferencias.
12	Buscar jugadores.
13	Buscar equipos.
14	Buscar usuarios.
15	Exportar Jugadores.
16	Exportar equipos.

NOTA: A continuación se muestran los principales casos de uso que servirán para el análisis durante todo el proyecto los demás se encuentran en el CD del proyecto en la carpeta Anexos/Documentos/especificaciones de caso de uso.doc

3.2.1

ESPECIFICACIÓN DE CASO DE USO: ASIGNACIÓN DE USUARIOS AL SISTEMA.

3.2.1.1 DESCRIPCIÓN

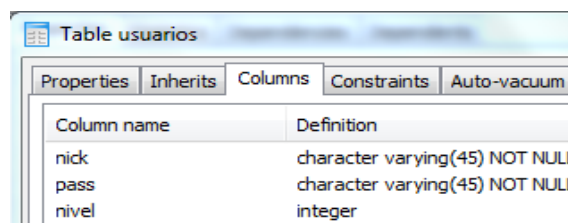
Tiene como función asignar usuarios al sistema para que estos puedan ingresar a páginas a él asignadas.

3.2.1.2 FLUJO DE EVENTOS

Flujo Básico.

- a) El empleado ingresa a la página principal del sistema que es **index.php**.
- b) Buscar la sección de ingreso al sistema, para seguir con el ingreso de usuario y clave para la validación de usuarios los mismos que serán consultados primero en la base de datos fede en la tabla usuarios, para la creación de usuarios es necesario ingresar con un usuario que tiene como nivel administrador ya que solo este tiene permisos especiales de creación, modificación, edición y eliminación.

Ilustración 41.- Recorte tabla usuarios.



Column name	Definition
nick	character varying(45) NOT NULL
pass	character varying(45) NOT NULL
nivel	integer

- c) Luego que es usuario es validado vamos a insertar nuevo usuario del sistema el mismo que será asignado a diferentes secciones del sistema.
- d) Ingresar el nombre de usuario el mismo que será almacenado en la columna Nick que es donde se registran todos los nombres de usuarios creados.

- e) Ingresar la clave de usuario el mismo que será almacenado en la columna pass que es donde se registran todas las claves o contraseñas de usuarios creados.
- f) Seleccionar el nivel de usuario el mismo que será almacenado en la columna nivel que es donde se registran todos los niveles al que el usuario puede ingresar de usuarios creados.

Ilustración 42.- Recorte del formulario insertar usuarios.

The image shows a web form for adding users. It consists of three input fields: 'Usuario' (text), 'Clave' (text), and 'Nivel' (dropdown menu). Below the fields are two buttons: 'Insertar' and 'Restablecer'.

Flujos alternativos.

N/A

3.2.1.3 PRECONDICIONES

- Información del usuario

La información debe estar correctamente ingresada ya que si no, va ser imposible ingresar a este módulo en el sistema ya que por lo menos debe existir un usuario administrador registrado para seguir ingresando más usuarios.

3.2.1.4 POS CONDICIONES

N/A

3.2.2

ESPECIFICACIÓN DE CASO DE USO: INFORMACIÓN DE LIGAS.

3.2.2.1 DESCRIPCIÓN

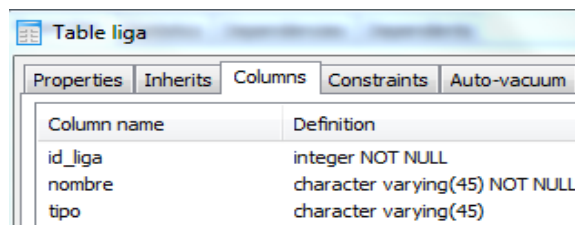
Tiene como función mostrar todo el proceso que corresponde a las ligas barriales que pertenecen a la provincia de Imbabura.

3.2.2.2 FLUJO DE EVENTOS

Flujo Básico.

- a) El administrador ingresa al módulo de inserción de Ligas, es decir que aquí vamos a crear una nueva liga barrial en la provincia de Imbabura este proceso de creación se da en la tabla liga.

Ilustración 43.- Recorte tabla liga.



Column name	Definition
id_liga	integer NOT NULL
nombre	character varying(45) NOT NULL
tipo	character varying(45)

- b) El Id_liga es la clave primaria de la tabla ligas el cual identifica el número de liga que se va a crear, dicho identificador se irá incrementando a medida que son creadas las ligas barriales.
- c) A continuación se ingresa el nombre de liga barrial que necesitamos crear la misma que será almacenada en la columna nombre de la tabla liga.
- d) Luego ingresamos el tipo de liga barrial como es una liga nueva deberá ingresar como “PRELIGA” por qué solo contiene una sola serie.

Ilustración 44.- Recorte del formulario insertar ligas.

Id_liga:

Nombre

Tipo:

Flujos alternativos.

N/A

3.2.2.3 PRECONDICIONES

- Información id_liga
Con este identificador de liga que se autoincrementa evitamos que existan ligas duplicadas.

3.2.2.4 POS CONDICIONES

Recibir la información de la tabla ligas para cuando el usuario quiera ingresar la información acerca de las personas que conforman la directiva de una determinada liga.

3.2.3

ESPECIFICACIÓN DE CASO DE USO INFORMACIÓN DE EQUIPOS.

3.2.3.1 DESCRIPCIÓN

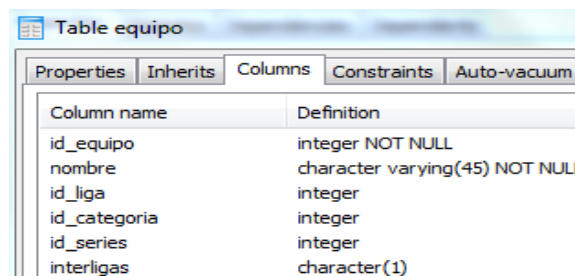
La finalidad es mostrar todo el proceso de información de equipos que pertenecen a una determinada liga barrial de la provincia ya sea este de serie A, B u otra diferente.

3.2.3.2 FLUJO DE EVENTOS

Flujo Básico.

- a) El administrador ingresa al módulo de inserción de equipos, es decir que aquí vamos a crear un nuevo equipo que pertenecerá a una determinada liga barrial en la provincia de Imbabura este proceso de creación se da en la tabla equipo.

Ilustración 45.- Recorte tabla equipo.



Column name	Definition
id_equipo	integer NOT NULL
nombre	character varying(45) NOT NULL
id_liga	integer
id_categoria	integer
id_series	integer
interligas	character(1)

- b) El Id_equipo es la clave primaria de la tabla equipo el cual identifica el número de equipo que se va a crear, dicho identificador se irá incrementando a medida que son creados los equipos en las determinadas ligas barriales.
- c) A continuación debemos ingresar el nombre del equipo que estamos creando dicha información será guardada en la columna nombre.
- d) Luego vamos a seleccionar el nombre de la liga a la que pertenecerá el equipo recién creado, dicha información será almacenada en la columna id_liga.
- e) Seleccionar la categoría a la que pertenece el equipo creado dicha información será almacenada en la columna id_categoria.

- f) Seleccionar la serie a la que va a pertenecer dicho equipo.
- g) Seleccionar si pertenecerá o formaran parte de los equipos que representara a su liga en el campeonato interligas.

Ilustración 46.- Recorte formulario equipo.

id_equipo:

Nombre:

liga: ▼

Categoría: ▼

Flujos alternativos.

N/a

3.2.3.3 PRECONDICIONES

Información de id_liga

Debemos registrar la información de la liga y de la serie a la que pertenecerá dicho equipo para ver si se encuentra registrado.

Información id_equipo

Dicha información se incrementara automáticamente cada vez que se cree un equipo en nuestra base de datos.

3.2.3.4 POS CONDICIONES

Recibir la información por parte del sistema de la liga barrial para averiguar su existencia.

3.2.4 **ESPECIFICACIÓN DE CASO DE USO: INGRESO DE JUGADORES.**

3.2.4.1 DESCRIPCIÓN

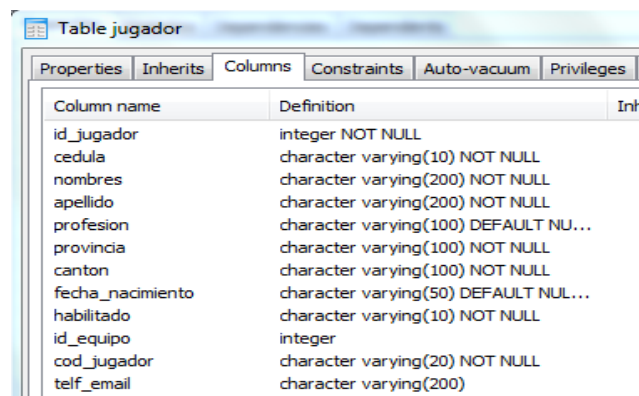
La finalidad es mostrar todo el proceso de información de jugadores que pertenecen a un equipo de una determinada liga barrial de la provincia ya sea este de serie A, B u otra diferente.

3.2.4.2 FLUJO DE EVENTOS

Flujo Básico.

- a) El administrador ingresa al módulo de inserción de jugadores, es decir que aquí vamos a crear un nuevo jugador que pertenecerá a un determinado equipo de liga barrial en la provincia de Imbabura este proceso de creación se da en la tabla jugadores.

Ilustración 47.- Recorte de la tabla jugador.



Column name	Definition	Inf
id_jugador	integer NOT NULL	
cedula	character varying(10) NOT NULL	
nombres	character varying(200) NOT NULL	
apellido	character varying(200) NOT NULL	
profesion	character varying(100) DEFAULT NU...	
provincia	character varying(100) NOT NULL	
canton	character varying(100) NOT NULL	
fecha_nacimiento	character varying(50) DEFAULT NUL...	
habilitado	character varying(10) NOT NULL	
id_equipo	integer	
cod_jugador	character varying(20) NOT NULL	
telf_email	character varying(200)	

- b) El Id_jugador es la clave primaria de la tabla jugador el cual identifica el número de jugador que se va a registrar, dicho identificador se irá incrementando a medida que son creados los jugadores en los determinados equipos.

- c) Ingresar la cedula de Identidad sin utilizar guion “-” o pasaporte si pertenece a otro país.
- d) Ingresar los datos que se encuentran en la cedula de identidad como son: nombres, apellidos, provincia, cantón, profesión, fecha de nacimiento las cuales serán almacenadas en sus respectivos campos de tabla jugador.
- e) Seleccionar la liga, categoría y el equipo al que pertenece el jugador creada dicha información será almacenada en su respectiva columna.
- f) Ingrese el código del jugador anteponiendo la letra que representa a cada liga seguida del id_jugador.
- g) Ingrese el teléfono o e-mail del jugador.

Ilustración 48.- Recorte formulario ingresar jugador.

Id_jugador:
Cedula:
Nombres
Apellido:
Profesion:
Provincia:
Canton:
Fecha de Nacimiento:
Habilitado:
Liga:
CATEGORIA:
Equipo:
Codigo jugador:
Telf / E-mail:

Flujos alternativos.

N/A.

3.2.4.3 PRECONDICIONES

Información id_jugador

Dicha información se incrementara automáticamente cada vez que se registre un jugador en nuestra base de datos.

3.2.4.4 POS CONDICIONES

N/A

3.2.5

ESPECIFICACIÓN DE CASO DE USO: NOTICIAS.

3.2.5.1 DESCRIPCIÓN

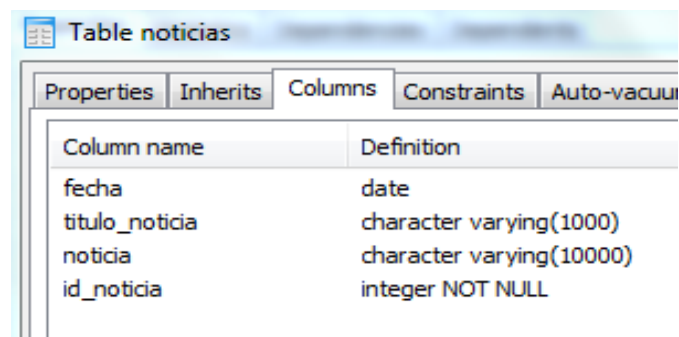
La finalidad es mostrar todo el proceso de información de las noticias más importantes de lo que sucede en las ligas barriales semana tras semana.

3.2.5.2 FLUJO DE EVENTOS

Flujo Básico.

- a) El administrador ingresa al módulo de inserción de noticias, es decir que aquí se va a crear una nueva noticia sobre algún acontecimiento que se dará en los próximos días.

Ilustración 49.- Recorte tabla noticia.



Column name	Definition
fecha	date
titulo_noticia	character varying(1000)
noticia	character varying(10000)
id_noticia	integer NOT NULL

- b) El Id_noticia es la clave primaria de la tabla noticias el cual identifica el número de noticia que se va a publicar, dicho identificador se irá incrementando a medida que se publiquen noticias.
- c) La fecha nos da el sistema tomando referencia a la fecha del sistema es decir la actual.
- d) Ingrese el título de la noticia.
- e) Ingrese la descripción de la noticia que será almacenada en la columna noticia.

Ilustración 50.- Recorte formulario noticia.

id_noticia:

Fecha

Título de la noticia:

Noticia:

Flujos alternativos.

3.2.5.3 PRECONDICIONES

N/A

3.2.5.4 POS CONDICIONES

N/A

3.2.6

ESPECIFICACIÓN DE CASO DE USO: INFORMACIÓN TRANSFERENCIAS.

3.2.6.1 DESCRIPCIÓN

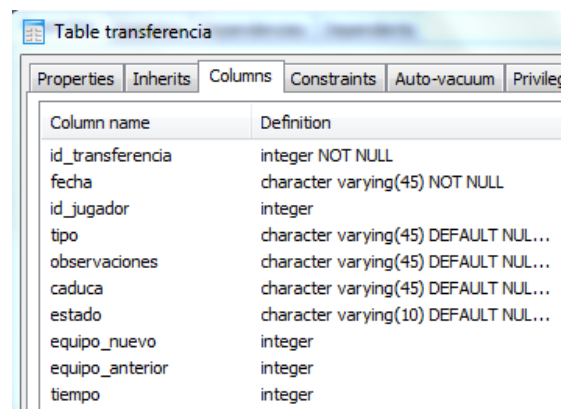
La finalidad es mostrar todo el proceso de información de las transferencias de los jugadores en las diferentes ligas barriales.

3.2.6.2 FLUJO DE EVENTOS

Flujo Básico.

- a) El administrador ingresa al módulo de operaciones, es decir que aquí se va a crear una nueva transferencia de algún jugador fichado en una determinada liga barrial.

Ilustración 51.- Recorte tabla transferencia.



Column name	Definition
id_transferencia	integer NOT NULL
fecha	character varying(45) NOT NULL
id_jugador	integer
tipo	character varying(45) DEFAULT NUL...
observaciones	character varying(45) DEFAULT NUL...
caduca	character varying(45) DEFAULT NUL...
estado	character varying(10) DEFAULT NUL...
equipo_nuevo	integer
equipo_anterior	integer
tiempo	integer

- b) El Id_transferencia es la clave primaria de la tabla transferencia el cual identifica el número de transferencia que se va a registrar, dicho identificador se irá incrementando a medida que se realicen transferencias.
- c) La fecha nos da el sistema tomando referencia a la fecha del sistema es decir la actual.

- d) Seleccionar el id_jugador que es el que código de selección para realizar la transferencia de un jugador.
- e) Seleccionar el id_equipo que es el que código del equipo al que pertenecerá el jugador una vez realizada la transferencia.
- f) Ingrese el código del equipo al que estaba perteneciendo el jugador.
- g) Seleccione el tipo de transferencia si es definitivo o préstamo.
- h) Ingrese 1 si selecciono préstamo en el ítem anterior o 0 si es definitivo.
- i) Seleccione “interno” si el jugador va a jugar en la misma liga o de “liga a liga” si va a pertenecer a otra liga barrial.
- j) Seleccione “a” si el estado es activo o “p” si no está debutando en la actualidad.

Ilustración 52.- Recorte formulario transferencia.

Id_transferencia:

Fecha:

ID JUGADOR:

Equipo nuevo:

equipo_anterior:

Tipo:

Tiempo:

Observaciones:

Caduca:

Estado:

Flujos alternativos.

N/A

3.2.6.3 PRECONDICIONES

Id_jugador

El jugador debe constar en la base de datos para poder realizar una transferencia.

3.2.6.4 POS CONDICIONES

Id_equipo

Actualizar la tabla jugador ya que dicho jugador ya dejara de pertenecer al dicho club.

3.3 USUARIOS

En esta sección se encuentra la descripción de cada usuario del sistema.

3.3.1 ADMINISTRADOR DEL SISTEMA

Tabla 16.- Perfiles de usuario administrador del sistema.

Representante	Joffre Paolo Aguas Rodríguez.
Descripción	Administrador de aplicaciones web y bases de datos de la federación de ligas barriales y parroquiales de la provincia de Imbabura.
Tipo	Usuario experto.
Responsabilidades	Mantener toda la infraestructura de hardware y software en perfecto funcionamiento. Esto incluye la instalación, configuración, administración de claves de acceso a los representantes de cada equipo. Inserción, Modificación, eliminación de datos. Realización de transferencias.
Criterio de éxito	Sistema en funcionamiento.
Implicación	Administración del sistema y bases de datos.
Entregables	Crea, modifica y elimina: usuarios, contraseñas, ingreso de ligas, equipos, jugadores, transferencias, noticias.
Comentarios	N/A

3.3.2 CONSULTOR DEL SISTEMA

Tabla 17.- Perfiles de usuario consultor del sistema.

Representante	Representantes de los diferentes equipos inscritos.
Descripción	Consultor de datos referentes a las ligas barriales, equipos y jugadores.
Tipo	Usuario final.
Responsabilidades	Consultar información referente a sus equipos, jugadores y ligas barriales.
Criterio de éxito	Información correcta.
Implicación	Ver datos en correcto orden.
Entregables	N/A
Comentarios	N/A

3.3.3 USUARIOS EXTERNOS

Tabla 18.- Perfiles de usuario externo del sistema.

Representante	Periodistas, usuarios externos.
Descripción	Consultor de datos referentes a los equipos, jugadores y noticias de lo que sucede en el deporte barrial.
Tipo	Usuario final.
Responsabilidades	Consultar datos de ligas barriales para hacerlas públicas en medios.
Criterio de éxito	Información correcta.
Implicación	Ver datos en correcto orden.
Entregables	N/A
Comentarios	N/A

3.3.4 NECESIDADES DE INTERESADOS Y USUARIOS

Tabla 19.- Necesidades de interesados y usuarios.

NECESIDAD	PRIORIDAD	INTERESES	SOLUCION ACTUAL	SOLUCIONES PROPUESTAS
Poseer información completa de jugadores.	ALTA	Poder realizar inserciones, modificaciones de los diferentes equipos y jugadores de una manera ágil.	Fichas de inscripción.	Desarrollar un sistema de información y administración de jugadores y equipos que pertenecen a cada liga barrial de Imbabura.
Actualización de datos sobre jugadores.	ALTA	Evitar que exista información obsoleta de jugadores y equipos.	Fichas de inscripción.	Utilización de dicho software para realizar la modificación de los datos anteriormente ingresados.
Registro de transferencias de jugadores.	ALTA	Obtener información de transferencias que se encuentran en la base de datos para poder conocer quién es el dueño de los pases de un determinado jugador.	Documentos realizados en Word.	Creación de un módulo de realización de transferencias y luego imprimirla para que quede sellado y constatado que el jugador pertenece a dicho club.
Tabla de posiciones de los equipos de serie A y B.	ALTA	Conocer resultados de equipos que vienen participando en los diferentes torneos.	Fixture de campeonato.	Recolectar los datos que se dan semana tras semana en escenarios deportivos e ingresar al sistema para conocer cuál es su posición en un

				determinado campeonato.
Tabla de goleadores.	ALTA	Conocer los goles que vienen marcando los jugadores en los diferentes escenarios.	Fixture de campeonato.	Recolectar los datos que se dan semana tras semana en escenarios deportivos e ingresar al sistema para conocer cuál es el jugador que viene marcando más goles en el campeonato.
Calendario de encuentros de las diferentes fechas en el fútbol barrial.	ALTA	Conocer las fechas en que los equipos tendrán que enfrentarse.	Fixture de campeonato.	Entregar información sobre próximos encuentros.
Realización del sistema con software libre.	MEDIA	Utilizar políticas gubernamentales sobre la utilización de software libre y verificar el funcionamiento para futuras aplicaciones.	Microsoft Access.	Utilizar herramientas como: Ide de programación Netbeans. Lenguaje: php Base de datos: Postgres.

4. ELABORACIÓN DE LA ARQUITECTURA DEL SOFTWARE

4.1 INTRODUCCIÓN

Uno de los desarrollos más importantes dentro de la construcción del software es el desarrollo de la arquitectura de software, que permite representar la estructura del sistema, sirviendo de comunicación entre las personas involucradas en el desarrollo y ayudando a realizar diversos análisis que orienten el proceso de toma de decisiones.

La plantilla de este documento se basó en las especificaciones de RUP (Rational Unified Process) para el documento de arquitectura de software.

Este documento provee al usuario especializado una vista de la arquitectura del sistema de administración de bienes informáticos.

4.1.1 PROPÓSITO

Este documento proporciona una descripción de la arquitectura del sistema, haciendo uso de diversas visiones arquitectónicas para representar diversos aspectos del sistema. Se realiza con el fin de documentar las decisiones de arquitectura significativas que se han tomado en el sistema. Y mostrar el funcionamiento de la arquitectura MVC que se usará.

4.1.2 ALCANCE

El presente documento se concreta, de manera puntual, en el desarrollo de la vista lógica y de implementación correspondientes. Así como también a la identificación de los diferentes componentes pertenecientes a cada vista.

Lo que concierne a los diferentes componentes externos se anexa una descripción centralizada de los mismos, indicándose referencias para su consulta.

4.1.3 REFERENCIAS

- Artefacto Modelo de casos de uso.
- Artefacto Glosario.

4.1.4 RESUMEN

En las siguientes secciones de este documento se detalla la arquitectura del software a desarrollar. Para ello se presenta de manera clara, empleando un lenguaje sencillo y directo, así como gráficos y vistas de acuerdo a la metodología utilizada.

4.2 REPRESENTACIÓN DE LA ARQUITECTURA

El presente documento presenta la arquitectura como una serie de vistas:

- Vista de casos de uso.
- Vista de despliegue.
- Vista de implementación.

Los modelos han sido desarrollados usando y el lenguaje UML y el diseñador de diagramas ArgoUML una herramienta de software libre.

4.3 OBJETIVOS Y RESTRICCIONES DE LA ARQUITECTURA

4.3.1 OBJETIVOS

- Utilizar la arquitectura MVC, que cuenta con un funcionamiento efectivo, es decir, que provea las funciones y propiedades especificadas.
- Analizar el diseño para verificar el alto grado de acoplamiento entre cada uno de los componentes, para así, permitir la manipulación y reemplazo de los mismos con facilidad que anticipe mudanzas posibles y permita el trabajo a futuro.
- Desarrollar la vista lógica de los componentes que formarán la arquitectura del sistema de gestión de ligas barriales relacionando su estructura con este documento.

4.3.2 RESTRICCIONES

- Los requerimientos de rendimiento estipulados en el Documento de Visión, deben de ser considerados como parte de la arquitectura del sistema a implementar.
- Está aplicación será construida bajo el patrón MVC (Modelo- Vista- Controlador), por lo que toda su funcionalidad cumplirá con este esquema.
- El sistema será una aplicación web, debido a que la información debe llegar a los usuarios de una forma transparente y concreta.
- Debido al sistema web su navegador de internet debe tener activado las funciones de flash y JavaScript por sus contenidos multimedia.

4.4 VISTA DINÁMICA

4.4.1 INTRODUCCIÓN

La vista dinámica tiende a analizarse como pequeñas piezas del sistema, es como escenarios individuales de operaciones. Representa las interacciones de los objetos de un sistema.

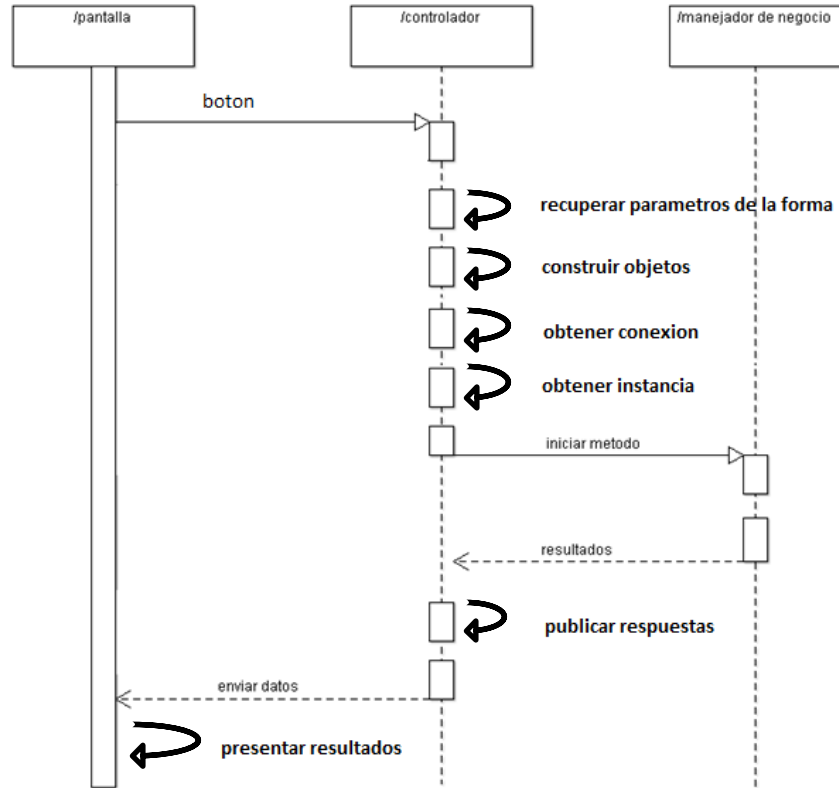
Puede representar como el sistema responderá a las acciones de los usuarios, como los datos son movidos del almacenamiento a la vista del usuario y como los objetos son creados y manipulados; todo lo cual los convierte en los diagramas más usados en proyectos, ya que son los que revelan más directamente características específicas requeridas en el código final.

4.4.2 DIAGRAMAS DE SECUENCIA

Los diagramas de secuencia ilustran la interacción entre objetos y el orden secuencial en el que ocurren dichas interacciones, es decir cómo se comunican los objetos entre sí.

A continuación se detalla la secuencia de interacción del mecanismo implementación MVC.

Ilustración 53.- Diagrama de secuencia: Arquitectura MVC



4.5 VISTA LÓGICA

4.5.1 INTRODUCCIÓN

En esta vista se presentan las unidades lógicas que componen la arquitectura del proyecto de gestión de ligas barriales, ya que el sistema manejará un patrón de arquitectura de software llamado MVC (Modelo - Vista - Controlador). Es un patrón que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

El primer refinamiento realizado consiste en la descomposición en subsistemas. Los subsistemas representan cortes verticales al diseño del sistema. Cada subsistema consiste en el agrupamiento de diferentes funcionalidades relacionadas entre sí y posee la capacidad de funcionar como un sistema en sí mismo.

4.5.2 DESCOMPOSICIÓN DE SUBSISTEMAS

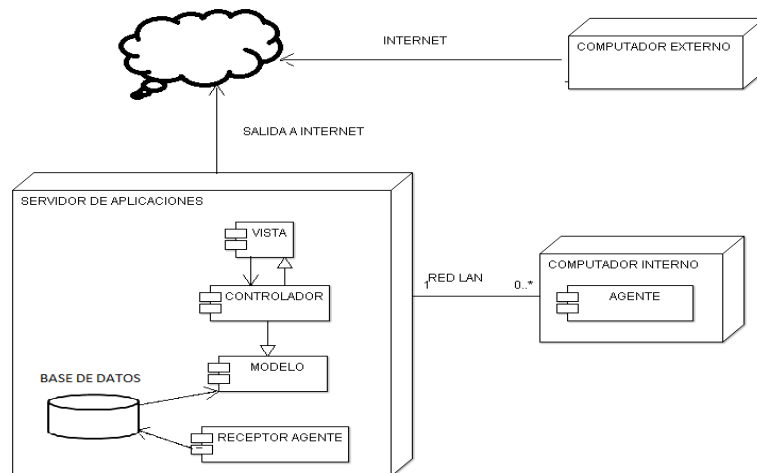
La explicación va relaciona con la información del patrón MVC.

- La capa del modelo define la lógica de negocio (la base de datos pertenece a esta capa), los datos se encuentran en la base de datos postgres los mismos que serán recuperados con el controlador.
- La vista es lo que utilizan los usuarios para interactuar con la aplicación (los gestores de plantillas pertenecen a esta capa). Estas serán creadas en formato php añadiendo los componentes que se requiera para hacer más agradable al usuario. Estas páginas se guardan en varios directorios repartidos por todo el proyecto.
- El controlador es un bloque de código que realiza llamadas al modelo para obtener los datos y se los pasa a la vista para que los muestre al usuario. La comunicación entre la base de datos y el sistema se comunican a través de una clase conexión.php que se la realiza en nuestro proyecto la cual está ubicada en el directorio C:\xampp\htdocs\Fedeligas\Conexion.

4.6 VISTA DE DESPLIEGUE

El presente diagrama de despliegue muestra de manera gráfica los nodos que conforman el sistema, además los describe indicando la localización de las tareas en los nodos físicos. Cada nodo representa un recurso de ejecución como una computadora o servidor y se encuentra conectado con otros mediante un enlace de comunicación.

Ilustración 54.- Diagrama de despliegue: Modelo MVC



4.6.1 SERVIDOR DE APLICACIONES

Sera uno de los computadores de Fedeligas Imbabura donde se encuentra la Base de Datos para lo cual contara con una dirección Ip publica para salir a internet. Los diferentes usuarios tendrán acceso al servidor vía LAN e internet.

4.6.2 COMPUTADORES INTERNOS

Son las computadoras que son utilizadas por cada uno de los usuarios de Fedeligas Imbabura e ingresan al sistema. Están conectadas directamente al servidor principal vía LAN.

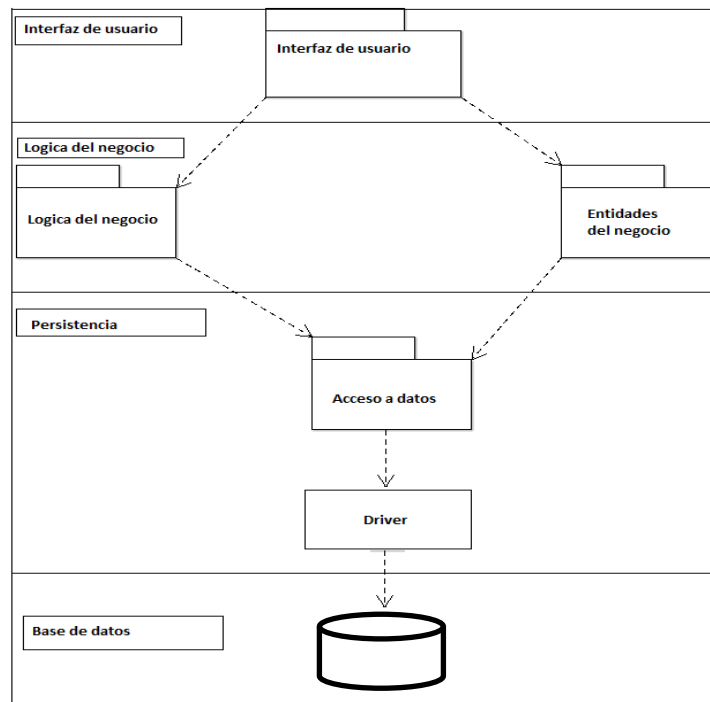
4.6.3 COMPUTADORES EXTERNOS

Son los computadores que se encuentran conectados a internet.

4.7 DISEÑO EN CAPAS

En la siguiente figura podemos observar la distribución de los paquetes en cada una de las capas del sistema.

Ilustración 55.- Diagrama de distribución de Capas del sistema



4.7.1 CAPA INTERFAZ DE USUARIO

Esta capa contiene el paquete Interfaz de Usuario, el cual almacena todas las clases con las cuales el usuario puede interactuar como lo son las ventanas.

4.7.2 CAPA LÓGICA DEL NEGOCIO

Esta capa contiene los paquetes de Servicios de Negocio y Entidades de Negocio. Contiene la lógica para el manejo de las operaciones del negocio.

4.8 VISTA DE DATOS

Se realizará la estructura de los datos desarrollando los siguientes modelos de datos:

4.8.1 MODELO RELACIONAL

En este modelo todos los datos son almacenados en relaciones, y como cada relación es un conjunto de datos, el orden en el que estos se almacenen no tiene relevancia. Este tiene la considerable ventaja de que es más fácil de entender y de utilizar por un usuario no experto.

De manera simple, una relación representa una tabla que no es más que un conjunto de filas, cada fila es un conjunto de campos y cada campo representa un valor que interpretado describe el mundo real. Cada fila también se puede denominar registro y a cada columna también se le puede llamar campo o atributo.

4.8.2 MODELO FÍSICO

Modelo físico de los datos es una representación de un diseño de los datos que considere las instalaciones y los apremios de un sistema de gerencia dado la base de datos. En el ciclo de vida de un proyecto se deriva típicamente de un modelo lógico de los datos, aunque puede ser reverso dirigido de una puesta en práctica dada la base de datos. Un modelo físico completo de los datos incluirá todos los artefactos de la base de datos requeridos para crear relaciones entre las tablas o para alcanzar metas del funcionamiento.

4.9 TAMAÑO Y DESEMPEÑO

4.9.1 TIEMPO DE RESPUESTA EN EL ACCESO A LA BASE DE DATOS

El sistema proveerá accesos a la base de datos con un tiempo de respuesta no mayor a los 3 segundos.

4.9.2 TIEMPO DE RESPUESTA DE TRANSACCIONES

El programa no demorará más de 3 segundos en generar una distribución óptima para los cortes haciendo uso del algoritmo elegido y guardándolo en base de datos.

4.9.3 ESPACIO EN DISCO PARA SERVIDOR DE BASE DE DATOS

El espacio en disco necesario para la parte del Servidor de Base de datos deberá tener como mínimo 1 GB libres en disco para su correcto funcionamiento. La arquitectura seleccionada apoya los requisitos de latencia y capacidad en disco en puesta en práctica de una arquitectura cliente servidor y web. La porción del cliente solo se pone en ejecución en las PC locales de los distintos ambientes dentro de la empresa. Los componentes se han diseñado para asegurarse de obtener requisitos mínimos de disco y memoria en el lado de las PC del cliente.

4.10 CALIDAD

Para un mejor aprovechamiento de la arquitectura de software se dan los siguientes requerimientos de calidad:

4.10.1 USABILIDAD

El sistema permitirá un manejo intuitivo por parte de los usuarios.

4.10.2 EFICIENCIA

El programa no demorará más de 5 minutos en generar una distribución óptima para los cortes haciendo uso del algoritmo elegido.

4.10.3 SEGURIDAD

El sistema permitirá el acceso a funcionalidades dependiendo del perfil del usuario que ingresa al sistema, validando su ingreso a través de una clave.

4.10.4 CONFIABILIDAD

El sistema tendrá en cuenta que la información ingresada en él sea válida, para lo cual mostrará mensajes que expliquen al usuario acerca de los errores que éste pudiera cometer y de aquellos que pueda cometer el mismo sistema.

4.10.5 MANTENIMIENTO

El sistema será flexible, facilitando su mantenimiento futuro.

4.11 ARQUITECTURA DEL PROYECTO

4.11.1 INTRODUCCIÓN

En base a lo que ya se ha visto en el transcurso del documento, la base arquitectónica presentada, se describe a continuación la arquitectura de la aplicación del sistema de gestión de Ligas Barriales.

El desarrollo de un sistema con gran cantidad de software requiere que este sea visto desde diferentes perspectivas, diferentes usuarios tanto del equipo de desarrollo como del usuario final (experto de dominio).

También tenemos que considerar que una arquitectura no debe centrarse específicamente en la estructura y en el comportamiento, sino que abarque temas como el uso, funcionalidad, rendimiento, capacidad de adaptación, reutilización, capacidad para ser comprendida, restricciones, compromisos entre alternativas, así como aspectos estéticos.

4.11.2 PAQUETES DE ANALISIS

Los paquetes de análisis permiten organizar del modelo de análisis en grupos manejables. Se identifican los siguientes paquetes de análisis:

Tabla 20.- Descripción de los paquetes de análisis

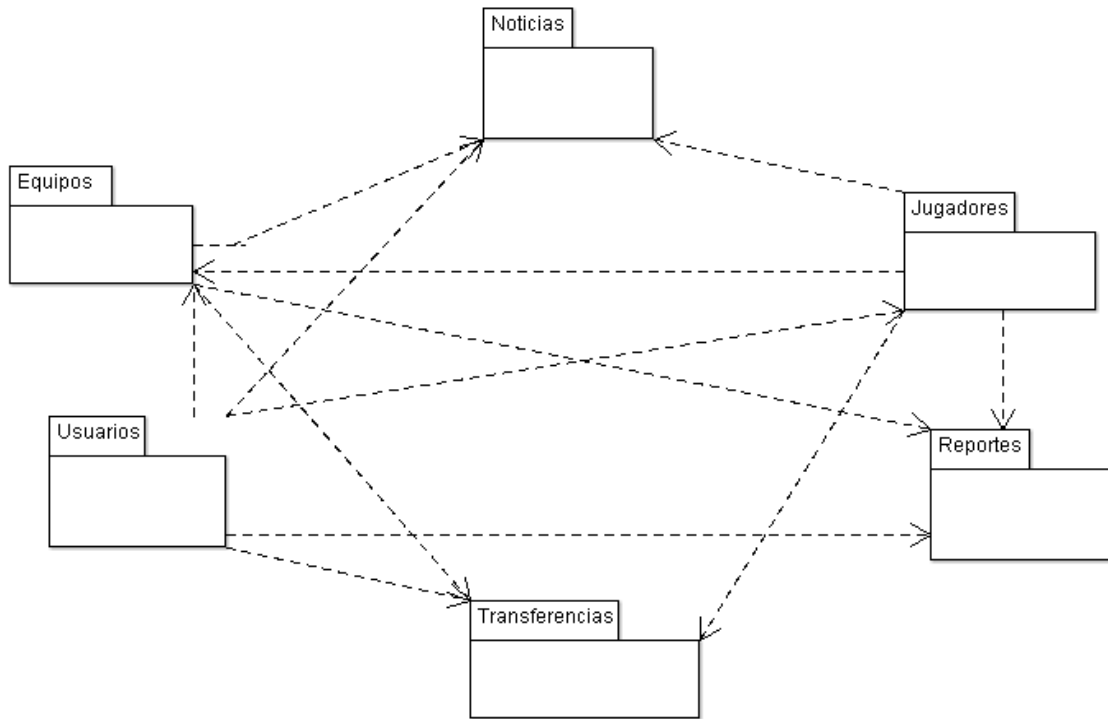
MODULO	PAQUETES DE ANÁLISIS	CASOS DE USO	GRAFICO CASOS DE USO
Sistema web	Equipos	Ingreso equipos.	<pre> graph TD usuario((usuario)) -- 1 --> validar_usuario(validar usuario) validar_usuario -- 1 --> consultarEquipos(consultar equipos) consultarEquipos --> informacionEquipo(información equipo) consultarEquipos --> buscarEquipo(buscar equipo) consultarEquipos --> exportarEquipos(exportar equipos) </pre>
		Actualización equipos.	
		Eliminación equipos.	
	Noticias	Nuevas Noticias.	<pre> graph TD usuario((usuario)) -- 1 --> validar_usuario(validar usuario) validar_usuario -- 1 --> insercion(insercion) insercion --> nuevaNoticia(nueva noticia) nuevaNoticia --> tituloNoticia(titulo noticia) nuevaNoticia --> noticia(noticia) </pre>
	Jugadores	Ingreso jugadores.	<pre> graph TD usuario((usuario)) -- 1 --> validar_usuario(validar usuario) validar_usuario -- 1 --> informacionJugadores(informacion jugadores) informacionJugadores -- 1 --> consultarJugador(consultar jugador) informacionJugadores -- 1 --> buscarJugador(buscar jugador) informacionJugadores -- 1 --> exportarJugadores(exportar jugadores) </pre>
		Actualización jugadores.	
Eliminación jugadores.			

	Usuarios	Ingreso usuarios.	
		Actualización usuarios.	
		Eliminación usuarios.	
	Transferencias	Realizar transferencias.	
Reportes	Reporte de jugadores		
	Reporte de equipos.		
	Reporte usuarios		
	Imprimir transferencias		



En el siguiente grafico se muestran las relaciones de dependencia entre cada uno de los paquetes de análisis.

Ilustración 56.- Dependencias entre paquetes del sistema web.



4.11.3 VISTA DINÁMICA

Los diagramas de secuencia ilustran la iteración entre objetos es decir cómo se comunican los objetos entre sí.

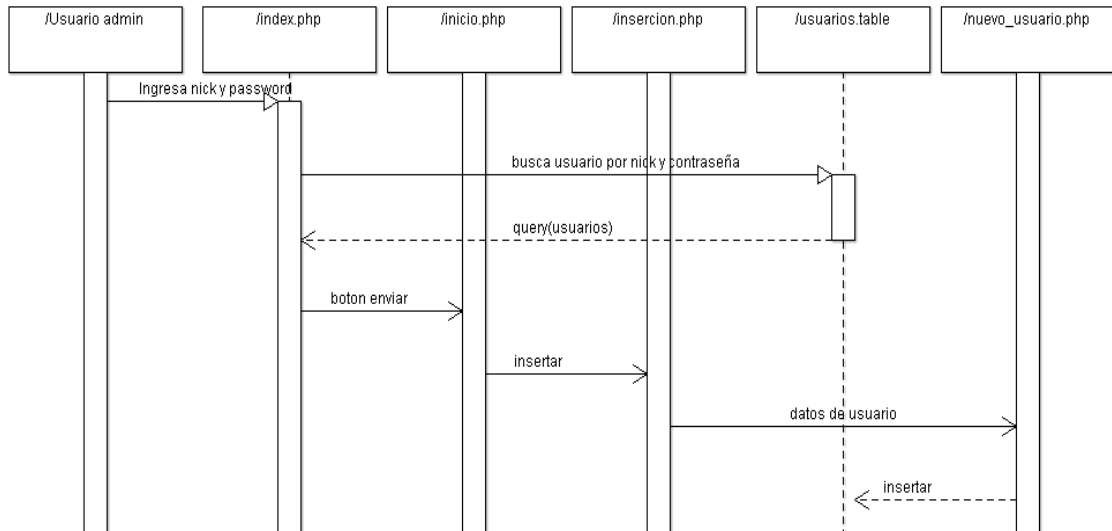
En los casos de uso se modelan las características del sistema y se desarrollan escenarios, los diagramas de secuencia proporcionan un camino a partir de los escenarios para describir las operaciones en una forma más detallada.

A continuación se detallan los diagramas de secuencia considerando los casos de uso de la sección anterior.

4.11.4 DIAGRAMAS DE SECUENCIA

4.11.4.1 Diagrama de secuencia Caso de Uso: Asignación de usuarios al sistema.

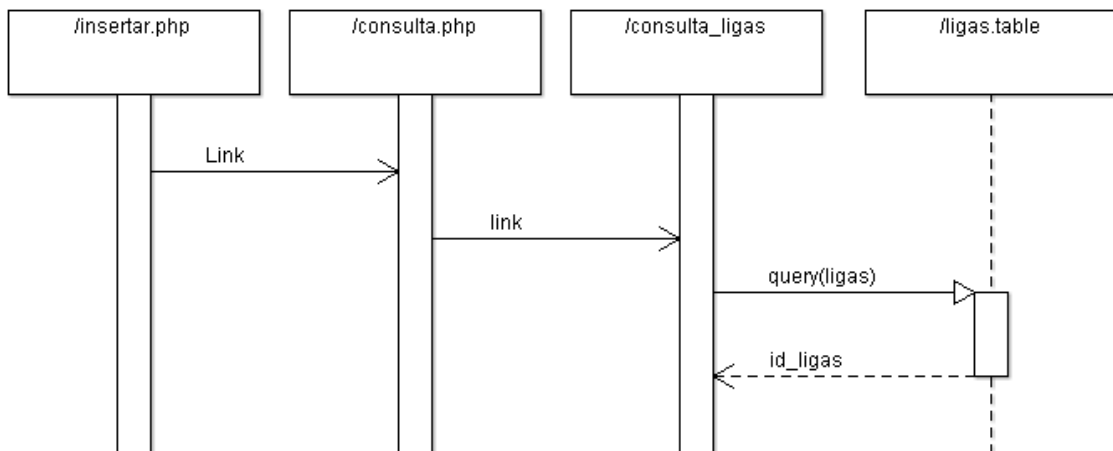
Ilustración 57.- Diagrama de secuencia Caso de Uso: Asignación de usuarios al sistema.



En el siguiente diagrama se analizara como se asigna usuarios al sistema de gestión de ligas barriales y parroquiales de Imbabura.

4.11.4.2 Diagrama de secuencia Caso de Uso: Información de Ligas.

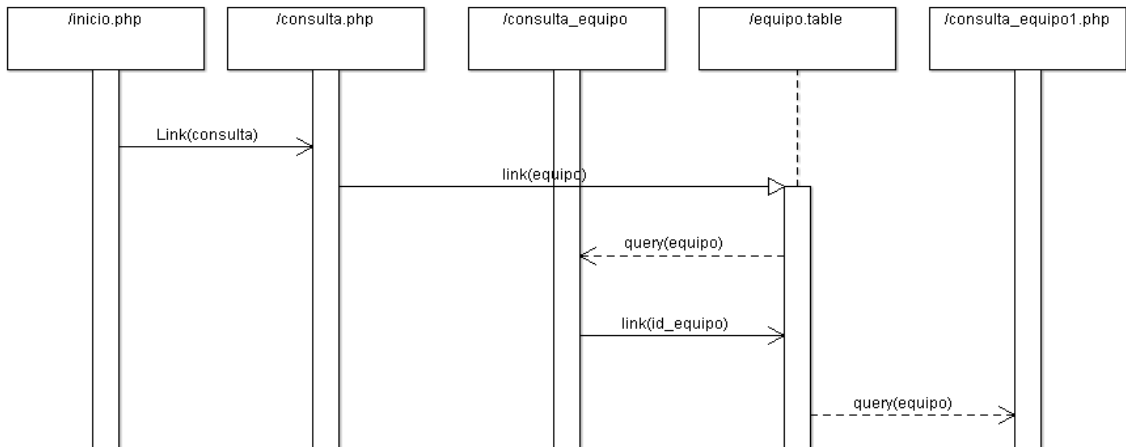
Ilustración 58.- Diagrama de secuencia Caso de Uso: Información de ligas.



En el siguiente diagrama se verá cómo funciona la información sobre las ligas barriales en el sistema.

4.11.4.3 Diagrama de secuencia Caso de Uso: Información de Equipos.

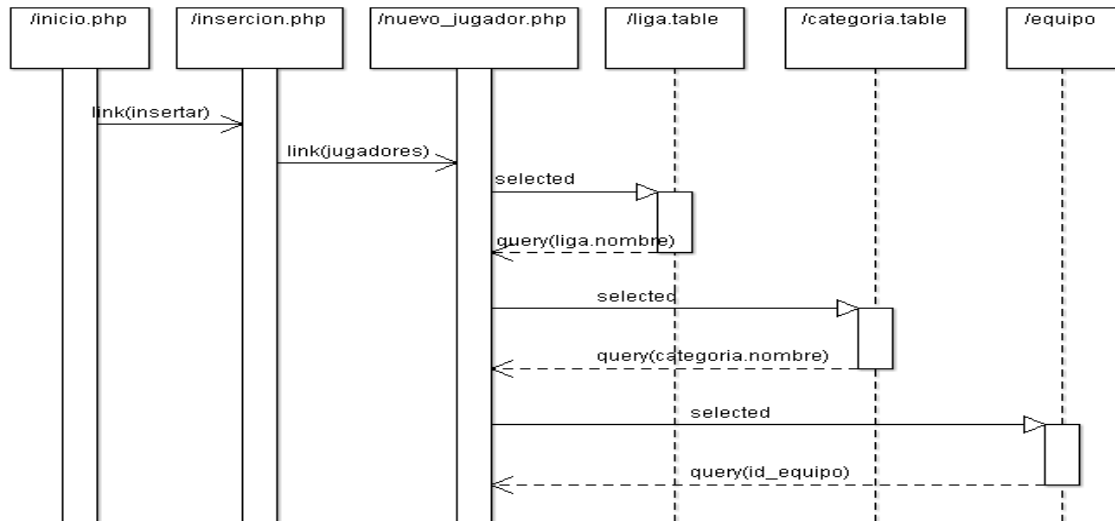
Ilustración 59.- Diagrama de secuencia Caso de Uso: Información de equipos.



En la siguiente ilustración se muestra la información sobre los equipos que pertenecen a todas las ligas barriales de la provincia de Imbabura.

4.11.4.4 Diagrama de secuencia Caso de Uso: Ingreso de jugadores.

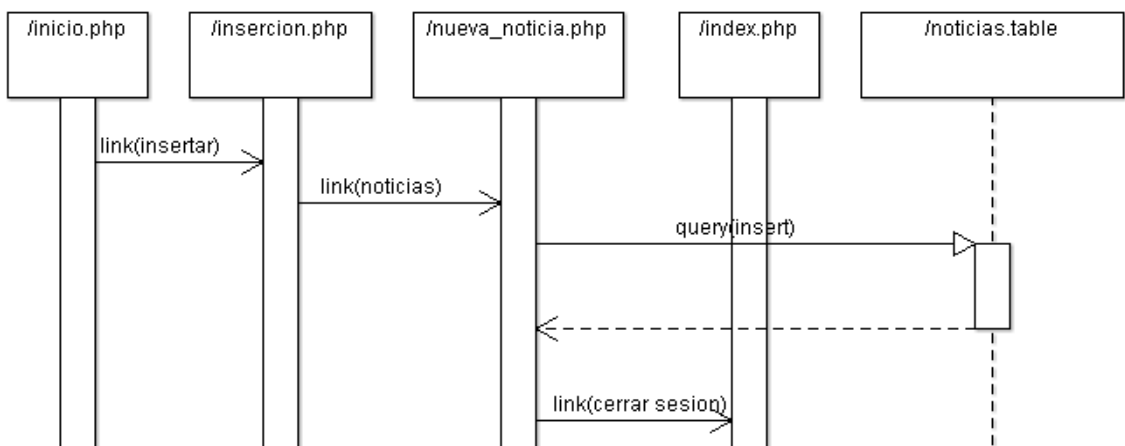
Ilustración 60.- Diagrama de secuencia Caso de Uso: Ingreso de jugadores



En la siguiente ilustración se muestra como se realiza el ingreso de jugadores al sistema.

4.11.4.5 Diagrama de secuencia Caso de Uso: Noticias.

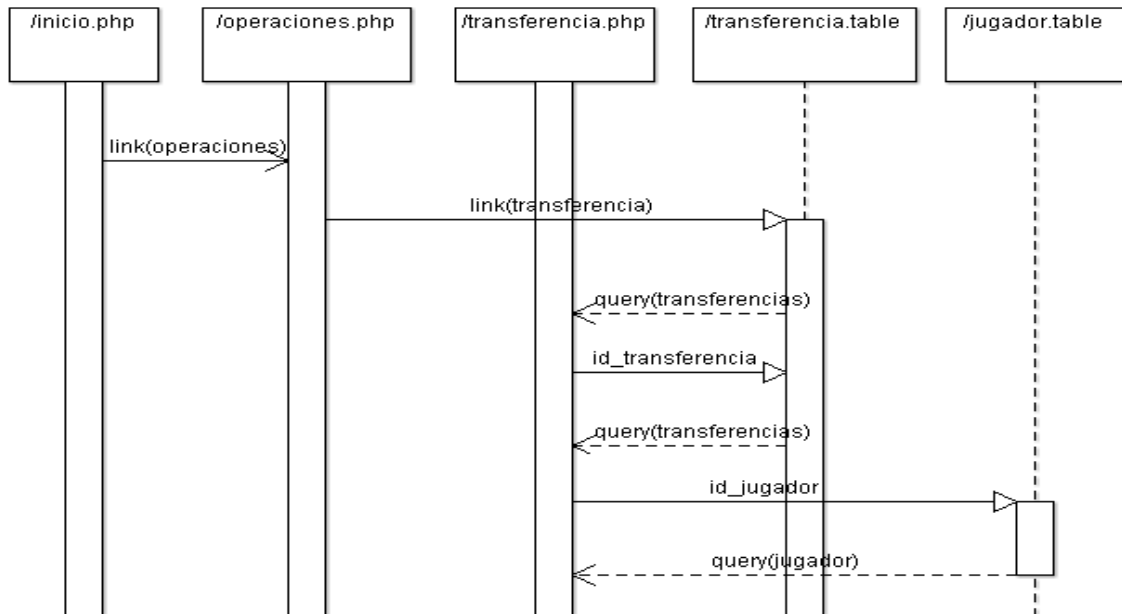
Ilustración 61.- Diagrama de secuencia Caso de Uso: Noticias.



La siguiente información muestra cómo se realiza el ingreso y visualización de nuevas noticias que han sucedido en cada una de las ligas barriales o torneos en general.

4.11.4.6 Diagrama de secuencia Caso de Uso: Información de transferencias.

Ilustración 62.- Diagrama de secuencia Caso de Uso: Información de transferencias.



La siguiente ilustración muestra la como se realiza la recolección de información sobre los pases y transferencias de jugadores en determinadas ligas barriales.

4.11.5 VISTA DE PROCESOS

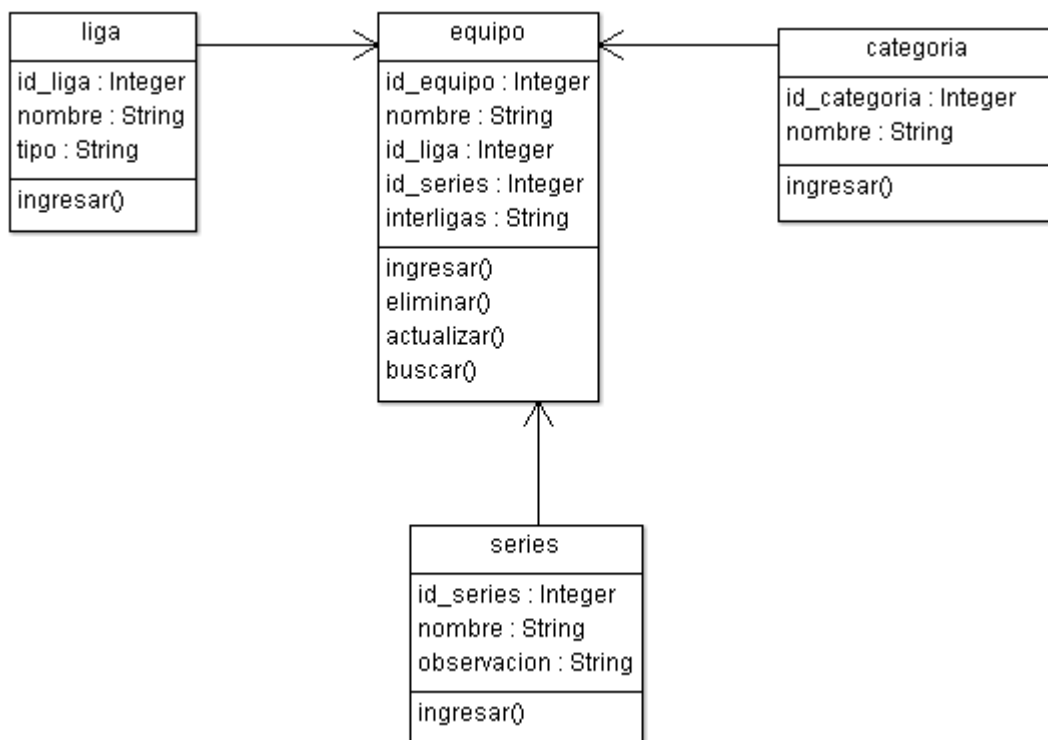
A continuación se muestran los diagramas de clases que utiliza cada caso de uso para desarrollar su proceso.

4.11.5.1 DIAGRAMA DE CLASE: Clase subsistema Equipos.

Se muestran las clases que conforman la inscripción de equipos en el sistema de ligas barriales de Imbabura, los diversos atributos y cómo interactúan entre ellos.

Muestra cómo representar las clases de los equipos que conforman las diferentes ligas barriales y parroquiales.

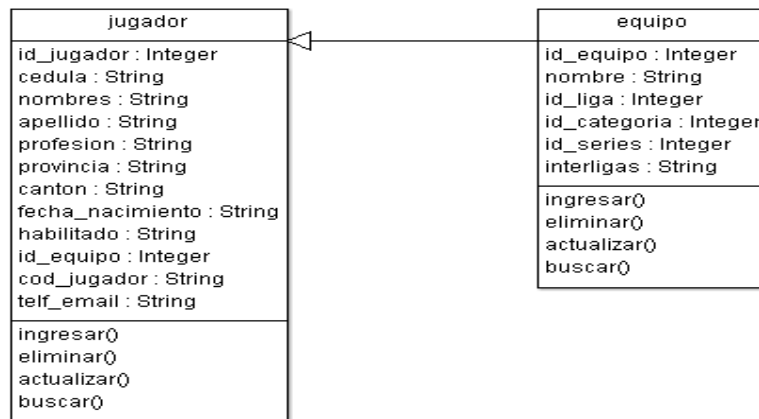
Ilustración 63.- Diagrama de clase: Clase subsistema equipos.



4.11.5.2 DIAGRAMA DE CLASE: Clase subsistema Jugadores.

Se muestran las clases que conforman la inscripción de jugadores en el sistema de ligas barriales de Imbabura, los diversos atributos y cómo interactúan entre ellos. Muestra cómo representar las clases de los jugadores que conforman los diferentes equipos en las ligas barriales y parroquiales.

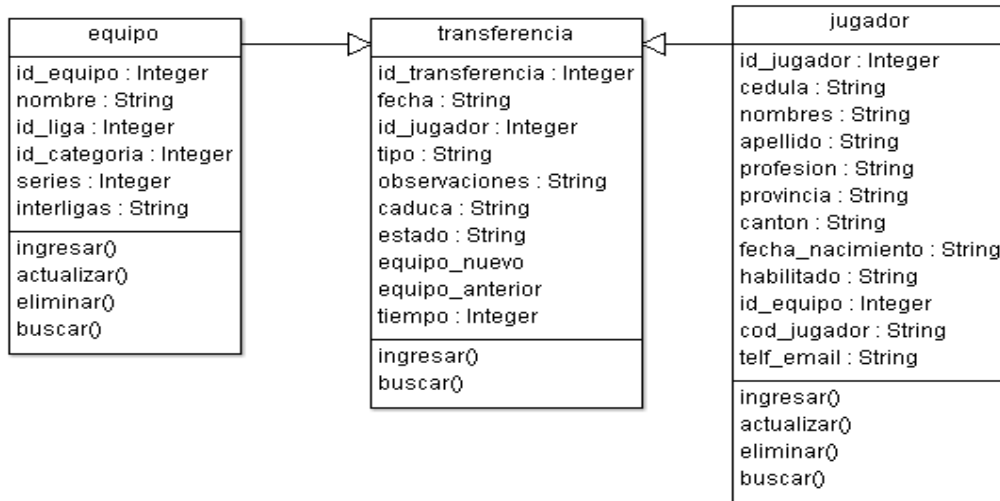
Ilustración 64.- Diagrama de clase: Clase subsistema jugadores.



4.11.5.3 DIAGRAMA DE CLASE: Clase subsistema Transferencias.

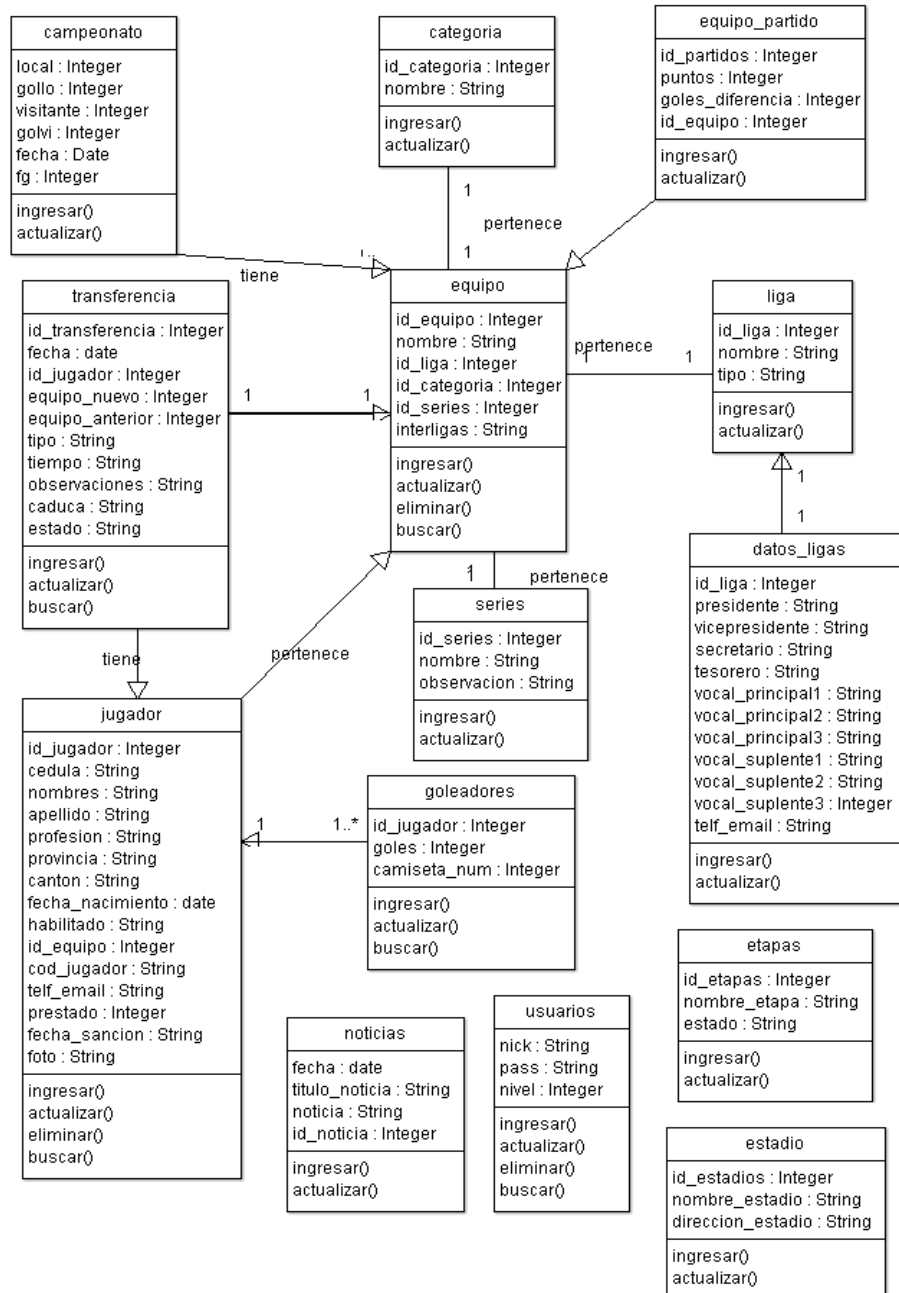
Se muestran las clases que conforman la transferencia de jugadores en el sistema de ligas barriales de Imbabura, los diversos atributos y cómo interactúan entre ellos. Muestra cómo podemos representar las clases de las transferencias de jugadores que conforman los diferentes equipos en las ligas barriales y parroquiales.

Ilustración 65.- Diagrama de clase: Clase subsistema Transferencias.



4.11.5.4 DIAGRAMA DE CLASES: SISTEMA COMPLETO

Ilustración 66.- Diagrama de clases del sistema completo



4.12 LISTA DE RIESGOS

4.12.1 INTRODUCCIÓN

La lista de riesgos permite analizar cada uno de los posibles problemas que se tendrá a lo largo del desarrollo del proyecto, magnitud, descripción, impactos que pueden causar, el indicador que nos permitirá verificar si el riesgo está por producirse o convertirse en un problema, una estrategia de mitigación para evitar que estos riesgos se cumplan, y el plan de contingencia que se dará si el riesgo se llegara a cumplir.

4.12.1.1 PROPÓSITO

El propósito de este documento es describir el proceso de administración de riesgos presentes a lo largo de la elaboración del proyecto, esta lista nos permitirá revisar periódicamente para evitar que uno de ellos se convierta en un problema. Este proceso de administración comprende la identificación, monitoreo y posterior mitigación de los riesgos.

4.12.1.2 ALCANCE

La lista de riesgos tiene como alcance definir varios tipos de riesgos en nuestro caso analizamos diferentes tipos:

- Riesgos de recursos (Organización y persona).
- Riesgos tecnológicos.
- Riesgos de dependencia externa.

4.12.1.3 REVISIÓN GENERAL

En el plan de riesgos que se va a elaborar, se analizara su magnitud que será la que permita clasificar que tan perjudicial puede ser para el proyecto el impacto que puede tener el proyecto, como podemos monitorear determinando riesgos, que está haciendo para reducir impactos de riesgos y en caso de que afecten cual será el plan de contingencia para que el riesgo no se convierta en problema y afecte al proyecto.

4.12.2 RIESGOS

4.12.2.1

LA BASE DE DATOS QUE SE PRETENDE UTILIZAR (POSTGRESQL) ES POCO CONOCIDA POR EL DESARROLLADOR.

- **Magnitud de riesgo o ranking.**
Alto.
- **Descripción.**
Este riesgo se puede presentar por la falta de conocimientos acerca de la utilización de esta herramienta ya que en las aulas no ha sido tan profundizado.
- **Impactos.**
 - Demora al momento de la programación en nivel de base de datos.
 - Demora al momento de realizar la conexión con php.
 - Ingreso de datos.
 - Falta de soporte para el tipo de datos que se necesite.
- **Indicadores**
% información de base de datos= información sobre las maneras de programar la base de datos y realizar la conexión a php/ información en cada tabla y programación de base de datos * 100
Umbral: 100%
- **Estrategia de mitigación.**
Investigar y obtener conocimientos adecuados relacionados al software que se va a utilizar con el fin de asegurar una correcta utilización del mismo.
- **Plan de contingencia.**

Inscribirse en foros con otras personas que manejen estas bases de datos para el momento que se necesite solicitar ayuda o hacerles conocer las inquietudes sobre un determinado problema.

4.12.2.2

SE PUEDE PRESENTAR UN TRABAJO EN LA INSTITUCIÓN PARA PODER ORDENAR INFORMACIÓN NECESARIA QUE SERVIRÁ AL PROYECTO.

- **Magnitud de riesgo o ranking.**
Alto.
- **Descripción.**
Este riesgo se puede presentar por la falta de personal que maneje las tecnologías de la información ya que una sola persona no puede cubrir todas las plazas.
- **Impactos.**
 - Demora al momento de la programación y avances del proyecto.
 - Pérdida del hilo de la programación.
- **Indicadores**
% otras tareas = realización de otras tareas en la federación de ligas barriales/
cada tarea realizada a lo largo del proyecto * 100
Umbral: 90%
- **Estrategia de mitigación.**
Realizar la contratación de personal capacitado para que maneje reportes de información que se encuentra en archivo.
- **Plan de contingencia.**
Pedir a los centros educativos (colegios, universidades) personal que necesite realizar sus pasantías o extensión universitaria para que se encargue de organizar toda la información para que el momento que sea requerida sepan donde se encuentra.

4.12.2.3

NO PRESENTAR ACTUALIZADOS LOS DATOS DE LOS JUGADORES.

- **Magnitud de riesgo o ranking.**
Alto.
- **Descripción.**
Este riesgo se puede presentar por la falta de información actualizada sobre los jugadores inscritos en determinados equipos.
- **Impactos.**
 - Tener datos erróneos.
 - Entorpecer el proceso ya que los pases y transferencias corresponderían a otros equipos.
 - No saber en cual equipo un jugador debuto por última vez.
 - Solicitar transferencias a equipos que cedieron jugadores hace algunos años atrás.
- **Indicadores**
 $\% \text{ información de jugadores} = \frac{\text{información total sobre los jugadores}}{\text{información de jugador de cada equipo}} * 100$

Umbral: 80%
- **Estrategia de mitigación.**
Llamar al jugador y al club en el que aparece como debutante dicho deportista para solucionar los problemas con respecto a su pase, pedir la copia de la carta pase del deportista y en últimas instancias su ficha técnica.

- **Plan de contingencia.**

Solicitar a los jugadores que actualicen sus datos con su respectivo club cada cierto periodo.

4.12.2.4

ORDENADOR ES ATACADO POR VIRUS

- **Magnitud de riesgo o ranking.**

Medio.

- **Descripción.**

Este riesgo se puede presentar por la falta de análisis en busca de virus a los discos duros ya que al pasar conectado a internet pueden estar expuestos al contagio de algún tipo de virus.

- **Impactos.**

- Falta de control en el sistema.
- Reinicio imprevisto de los equipos.
- Cuelgues de computador.
- Fallos en el sistema de archivos y discos duros.

- **Indicadores**

$\% \text{ archivos de sistema infectados} = \text{información necesaria para el correcto funcionamiento} / \text{archivos infectados} * 100$

Umbral: 50%

- **Estrategia de mitigación.**

Realizar la instalación de un buen antivirus y que sea fácil de actualizar periódicamente para contrarrestar el brote de nuevas amenazas, realizar la desinfección del ordenador cada cierto periodo. Desinfectar flash memory o pen drives cada vez que se introduzca en el equipo.

- **Plan de contingencia.**

Solicitar a los jugadores que actualicen sus datos con su respectivo club cada cierto periodo.

4.12.2.5

MODELO VISTA CONTROLADOR NO SE ACOPLA A LA METODOLOGIA RUP.

- **Magnitud de riesgo o ranking.**

Bajo.

- **Descripción.**

Diseñar los objetos de acuerdo al patrón MVC. Y construir la aplicación usando la metodología RUP.

- **Impactos.**

- Diagramas de clase complejos.
- Cambio de arquitectura.

- **Indicadores**

% Acoplamiento de metodología RUP y patrón MVC.

Umbral: 100%

- **Estrategia de mitigación.**

Utilizar un diseñador de diagramas libre que facilite la construcción de la aplicación y que no sea tan complicado su manera de usarlo.

- **Plan de contingencia.**

Realizar los diagramas según el patrón MVC.

4.12.2.6

INCOMPATIBILIDAD CON NAVEGADORES DE INTERNET Y/O CONFIGURACIONES ESPECÍFICAS EN LAS COMPUTADORAS CLIENTES.

- **Magnitud de riesgo o ranking.**

Bajo.

- **Descripción.**

Debido a los diversos navegadores de internet no exista una compatibilidad, ya que deben estar debidamente instalados los complementos flash y JavaScript.

- **Impactos.**

- Mostrar información de manera incorrecta.
- Mostrar páginas web incompletas.
- Reaccionar de una mala manera en las terminales clientes.

- **Indicadores**

$\% \text{ Compatibilidad} = \frac{\text{Navegadores existentes}}{\text{Navegadores compatibles con el sistema}} * 100.$

Umbral: 70%

- **Estrategia de mitigación.**

Buscar componentes que puedan ser compatibles con todos los navegadores y activar todos los complementos necesarios.

- **Plan de contingencia.**

Utilizar navegadores que manejen o gestionen la información multimedia de una manera adecuada.

4.12.2.7

DESCONOCER EL PROCESO DE INSCRIPCIONES Y COMO SE VIENEN REALIZANDO LOS CAMPEONATOS EN LA INSTITUCIÓN.

- **Magnitud de riesgo o ranking.**

Bajo.

- **Descripción.**

Debido a la falta de conocimientos sobre los campeonatos que se vienen organizando en la federación de ligas barriales de Imbabura y reglamentos se podría estar confundido en la manera de ir estructurando los campeonatos.

- **Impactos.**

- Datos erróneos en la estructuración de campeonatos.
- Juegos repetidos entre equipos.
- Incumplimiento de fechas.
- Partidos simultáneos.

- **Indicadores**

% Conocimiento de campeonatos y sus reglas.

Umbral: 30%

- **Estrategia de mitigación.**

Ponerse al tanto del funcionamiento de los procesos de inscripción y realizar cambios con la persona encargada de organizar los campeonatos.

- **Plan de contingencia.**

Antes de realizar el sistema realizar pruebas con el proceso de inscripciones y el de campeonatos.

4.13 PROTOTIPO DE INTERFAZ DE USUARIO

4.13.1 INTRODUCCIÓN

El siguiente artefacto que se detalla es una parte importante del proyecto ya que es la que interactúa de forma visible con el usuario.

Se debe desarrollarlo con elementos adecuados, ya que mediante este el usuario mantiene comunicación con el sistema desarrollado.

4.13.1.1 PROPÓSITO

Dar a conocer la plantilla que se utilizara en el desarrollo del sistema así como también los archivos de configuración.

4.13.1.2 DESCRIPCIÓN

Este documento presenta los siguientes aspectos a los diferentes interesados:

- Archivos y configuraciones que son importantes para la personalización de interfaces gráficas.
- Mapa de sitio.
- Diseño de la plantilla estándar.

4.13.2 ARCHIVOS DE CONFIGURACIÓN

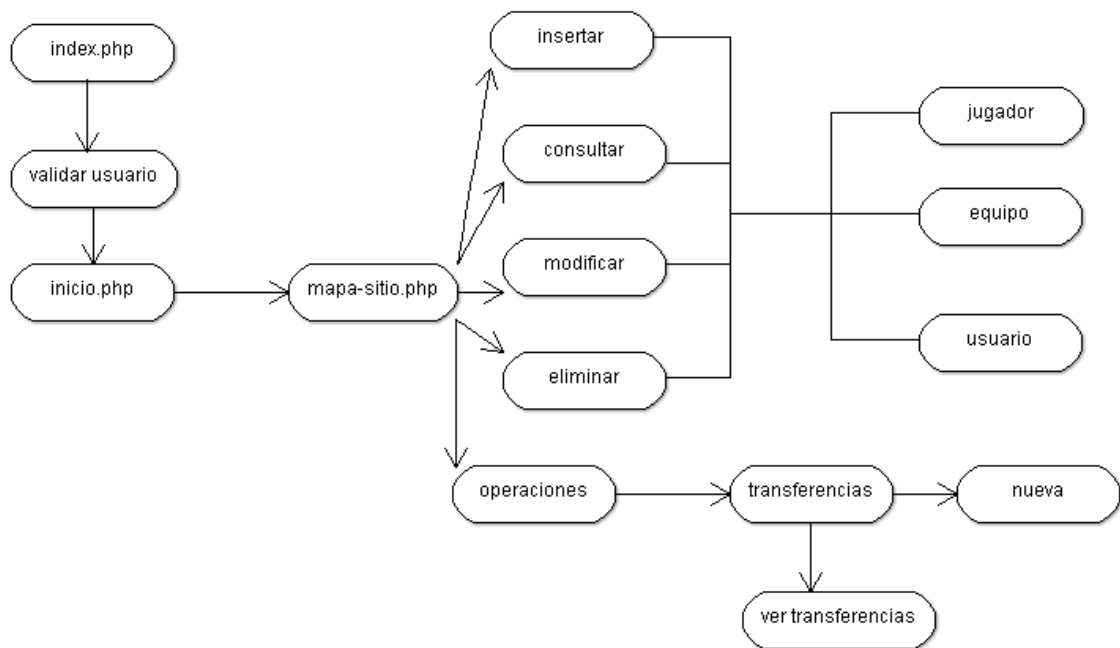
Para configurar la interfaz gráfica se trabaja sobre la capa de la vista del patrón MVC la misma que son trabajadas con los siguientes componentes:

- La presentación **PHP** que consta de código en formato php.
- Código HTML dentro de las paginas PHP.
- Declaraciones <meta>
- Títulos de páginas para indexar la búsqueda en páginas.
- Código JavaScript.
- Hojas de estilos css.

4.13.3 MAPA DE SITIO Y NAVEGACIÓN

Partiendo de los componentes del sistema a continuación se muestra un mapa de sitio de navegación.

Ilustración 67.- Mapa de sitio y navegación.

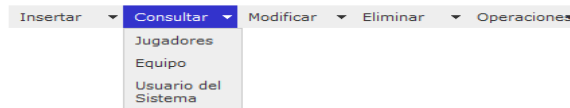


En la ilustración 66 se observa la manera como está estructurado el proyecto en un mapa de sitio que nos muestra la aplicación en una forma global para que el usuario no se pierda con las diferentes páginas web que contiene el mismo.

Ilustración 68.- Mapa- sitio.php



Mapa del Sitio



[Volver](#) | [Portada](#) | [Mapa del sitio](#)

Sistema de Inscripciones Realizado por [Joffre Aguas](#)

4.13.4 ESTRUCTURA DE LAS PAGINAS

4.13.4.1 Index.php

Es la página principal del sistema de Inscripción de Ligas Barriales y Parroquiales de Imbabura.

Tabla 21. Estructura de la página Index.php

NOMBRE COMPONENTE DE LA PAGINA	DESCRIPCION
Title	Nombre de la página.
shortcut icon	Icono de la institución que se muestra en la barra de navegación.
embed src	Banner de la institución.
menu	Despliega los diversos menús que se mostraran en el sistema.
href	Link a páginas de publicidad que auspician a Fedeligas Imbabura.
img	Contiene imágenes del sistema.

BODY	Muestra el contenido de las diferentes noticias, esta varía de acuerdo a la noticia seleccionada.
------	---

Ilustración 69.- Estructura de la página Index.php

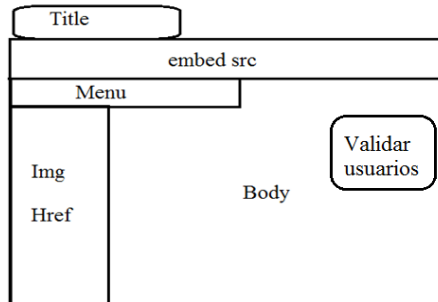


Ilustración 70.- Prototipo de interfaz de usuario: Index.php



4.13.4.2 Validar Usuarios.

En esta sección de la página índice los usuarios son validados para poder ingresar al sistema.

Tabla 22.- Estructura del formulario Validar usuario

COMPONENTE	DESCRIPCION
Table	Crea una tabla.
Tr	Crea filas
Td	Crea columnas
Input	Crear área de texto donde se introduce el nombre de usuario y contraseña.
Submit	Botón que se encarga de generar acciones
strong	Crea cuadros de texto.

Ilustración 71.- Estructura de la parte del formulario de validación de usuarios.

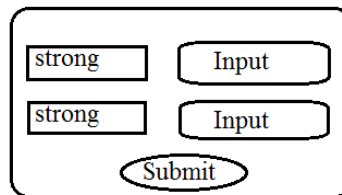


Ilustración 72.- Prototipo de interfaz de usuario: Validar usuario

Usuario:

Clave:

4.13.4.3 Inicio. Php

Es la página que es plantilla para todas las otras páginas que contiene el proyecto es una de las páginas más importantes ya que en esta se muestran los menús para realizar todo tipo de acciones.

Tabla 23.- Estructura del formulario Inicio.php

COMPONENTE	DESCRIPCION
embed src	Banner de la institución.
menu	Despliega los diversos menús que gestionan el sistema de inscripciones.
href	Link a páginas de gestión Fedeligas Imbabura.
BODY	Muestra el contenido de las diferentes noticias, esta varía de acuerdo a la noticia seleccionada.

Ilustración 73.- Estructura de la página Inicio.php.

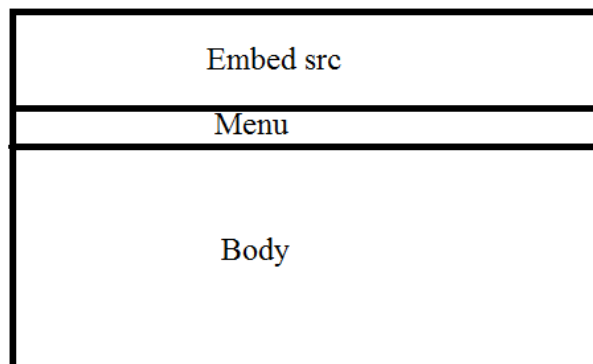
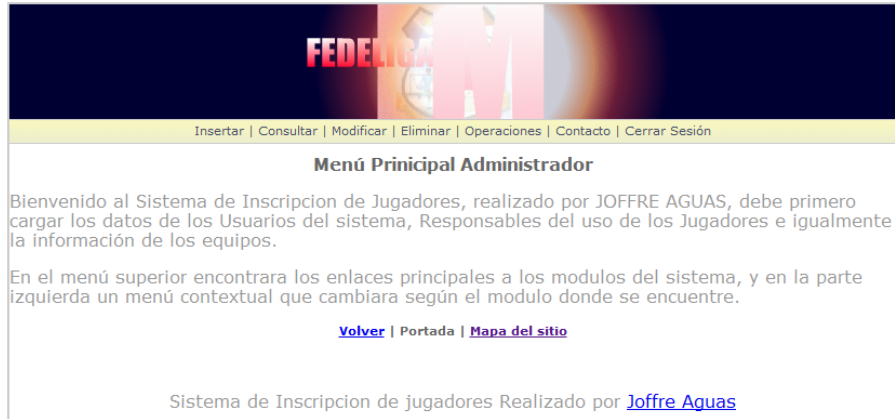


Ilustración 74.- Prototipo de interfaz de usuario: Inicio.php



5. IMPLEMENTACIÓN Y PRUEBAS.

5.1 INTRODUCCIÓN

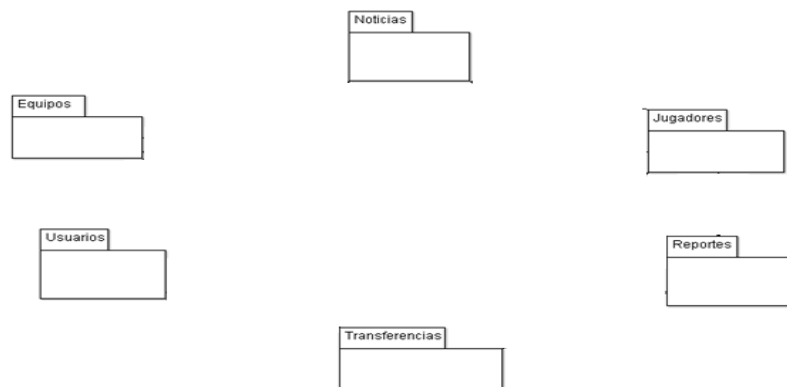
El modelo de implementación se obtuvo identificando varios componentes principales que representan componentes físicos, el modelo que se usará para la construcción del sistema, los cuales serán analizados con anterioridad en el desarrollo del documento de arquitectura de software.

En este capítulo se utilizarán los diagramas de componentes para identificar cómo está conformado el sistema físicamente.

5.2 DEFINICIÓN DE SUBSISTEMAS E IMPLEMENTACIÓN

Se muestran y detallan los subsistemas de implementación.

Ilustración 75.- Subsistemas del módulo de diseño.



5.3 MODULO WEB

5.3.1 SUBSISTEMA DE TRANSFERENCIAS

Está formado de varios componentes, los cuales tienen como propósito la gestión de transferencias de pases de jugadores, su ejecución es llevada a cabo por usuarios administradores del sistema, e incluye las siguientes acciones:

- Transferencias.
- Gestión de Jugadores.
- Gestión de Equipos.
- Gestión de Ligas.
- Tipo.
- Tiempo.
- Observaciones.

5.4 ESTÁNDARES DE PROGRAMACIÓN

5.4.1 INTRODUCCIÓN

Uno de los aspectos más importantes al iniciar un proyecto informático es el desarrollo de normas y políticas que permiten la mejor comprensión de los documentos, código de programación de las bases de datos entre otros recursos.

Si se trata de interpretar los bloques de código de uno o varios proyectos al momento de dar mantenimiento así directamente se perderá tiempo y al momento de realizar cambios los realizara de una manera errónea por lo cual el documento dará gran ayuda para lograr entender la codificación del sistema y administración de pases de los diferentes deportistas que forman parte de la federación de ligas barriales.

5.4.1.1 PROPÓSITO

Este artefacto tiene como objetivo dar a conocer a los interesados los estándares de programación que registran el desarrollo y mantenimiento a la aplicación que se va a realizar.

5.4.1.2 DESCRIPCIÓN

El documento presenta las reglas que permiten estandarizar el desarrollo del “Sistema de gestión de pases e inscripción de jugadores barriales”, utilizando postgres como base de datos y el lenguaje de programación php. Para lo cual se debe tener en cuenta lo siguiente:

- Base de datos.
- Lenguaje de programación.

5.4.2 ESTANDARIZACIÓN DE LA BASE DE DATOS.

5.4.2.1 NORMAS A APLICARSE

- a. Los nombres de bases de datos, tablas deberán ir en minúsculas.
- b. Los nombres de las variables van en singular de forma descriptiva y de manera corta para que sean fáciles de recordar y el esquema se haría más complicado.
- c. Es recomendable usar abreviaturas si el nombre es demasiado largo.
- d. Los nombres de clave primaria deberán ir precedidos de “id_” luego del campo descriptivo con minúsculas y separados con guiones bajos cuando más campos compongan la clave.
- e. Para la creación de índices fki_seguido de las siglas del nombre de la tabla.
- f. Los nombres para las constrains de claves foráneas deberán ir precedidos de “pk_” seguido del nombre de la tabla.
- g. Estilos de las consultas SQL:

La codificación de cualquier sentencia (SELECT, UPDATE, DELETE O INSERT) debe emplear indexación para clarificar la y facilitar la búsqueda de las tablas implicadas (FROM) y las condiciones impuestas (WHERE).

A continuación se muestra un ejemplo SELECT simple. Si es necesario recuperar muchos campos podría agruparse varios por línea:

```
SELECT nombres, apellido  
  
from jugador  
  
where j.id_jugador= 1
```

Todas las sentencias llevaran el estilo del ejemplo

5.4.3 ESTANDARIZACIÓN DEL LENGUAJE DE PROGRAMACIÓN.

Para programar en el lenguaje PHP existe una guía de estándares de desarrollo y programación que define el modo en que un script PHP debe ser realizado.

Estándar 1: Las funciones deben ser llamadas sin espacios entre el nombre de la función, el signo de paréntesis y el primer parámetro; espacios entre cada coma por parámetros y espacios entre el ultimo paréntesis, el signo de paréntesis cerrado y el signo de punto y coma (;).

Estándar 2: El estilo de los comentarios debe ser como el estilo de los comentarios para C (`/* */` ó `//`).

Estándar 3: Cuando se incluya un archivo de dependencia incondicionalmente utilice “`requiere_once`” y cuando sea incondicionalmente “`include_once`”.

Estándar 4: Siempre utilice las etiquetas `<?php ?>` para abrir un bloque de código. No utilice el método de etiquetas cortas porque esto depende de las directivas de configuración del archivo `PHP.ini` y hace que el script no sea tan portable.

Estándar 5: Los nombres de las clases deben empezar con letra mayúscula. Los nombres de las variables y de las funciones pueden empezar con letra minúscula, por si estas tienen más de una palabra, cada nueva palabra debe empezar con letra mayúscula. Si una función en una clase es privada deberá comenzar con el signo de guion mayor para más fácil identificación.

Estándar 6: Los archivos con código PHP, deben ser guardados en formato ASCII utilizando la codificación ISO-8859-1, ya que este formato es el mismo que se utiliza para guardar archivos de texto plano (.txt). La razón por la que se utiliza este estándar es por la que determinados editores HTML (dreamweaver), agregan códigos de carácter extraño de salto de línea (como si se tratara de un archivo binario) y esto puede ocasionar que el intérprete de PHP, encuentre problemas al momento de leer el script.

5.5 CASOS DE PRUEBA

5.5.1 Introducción

Los casos de prueba específica y comunica las condiciones que deben validarse para habilitar una valoración de algunos aspectos concretos del sistema que permitirá ver su comportamiento cuando interactúa con el usuario.

5.5.1.1 Propósito

Su propósito es especificar las pruebas que se realizan al sistema para verificar su funcionamiento.

5.5.1.2 Alcance

Detallar la evaluación que se debe realizar al caso de prueba, que se determina de los casos de uso especificados anteriormente los cuales a continuación se realiza la explicación.

5.5.2 Casos de prueba

5.5.2.1 Caso de prueba para el caso de uso: obtener información de equipo.

Tabla 24.- Caso de prueba para el caso de uso: obtener información de equipo.

Caso de prueba	Obtener información de equipo
Entrada	Se seleccionó anteriormente un equipo de fútbol.
Resultado	Mostrar la información completa del equipo de fútbol.
Condiciones	La información debe ser la correcta y estar actualizada desde la base de datos distribuidos en tablas correspondientes.

5.5.2.2 Caso de prueba para el caso de uso: obtener información de jugadores.

Tabla 25.- Caso de prueba para el caso de uso: obtener información de jugadores.

Caso de prueba	Obtener información de jugadores
Entrada	Se seleccionó anteriormente un jugador de un determinado equipo de fútbol.
Resultado	Mostrar la información completa del jugador
Condiciones	La información debe ser la correcta y estar actualizada desde la base de datos distribuidos en tablas correspondientes.

5.5.2.3 Caso de prueba para el caso de uso: obtener información de liga barrial.

Tabla 26.- Caso de prueba para el caso de uso: obtener información de liga barrial

Caso de prueba	Obtener información de liga barrial
Entrada	Se seleccionó anteriormente una liga

	barrial
Resultado	Mostrar la información completa de la liga barrial
Condiciones	La información debe ser la correcta y estar actualizada desde la base de datos distribuidos en tablas correspondientes.

5.5.2.4 Caso de prueba para el caso de uso: obtener información de transferencia.

Tabla 27.- Caso de prueba para el caso de uso: obtener información de transferencia.

Caso de prueba	Obtener información de transferencia
Entrada	Se seleccionó anteriormente un jugador de un determinado equipo de fútbol.
Resultado	Mostrar la información completa de la transferencia de un jugador.
Condiciones	La información debe ser la correcta y estar actualizada desde la base de datos distribuidos en tablas correspondientes.

6. CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

- ❖ Al terminar con el proceso de desarrollo del sistema se llegó a la conclusión que el objetivo de elaborar un sistema que permita colaborar con la gestión de jugadores y equipos de ligas barriales y parroquiales FEDELIGAS ha sido cumplido con éxito.
- ❖ Se entrega a FEDELIGAS una herramienta que permite manejar aspectos importantes dentro de la institución, llevando el control de todas las actividades que se realizan en la institución.
- ❖ Siendo apoyados en la metodología RUP para la construcción del proyecto se nos ha permitido producir un software de calidad que satisface a las necesidades de los usuarios, posibilitando un avance ordenado a través de sus procesos y actividades. Asimismo, los documentos y modelos propuestos en la metodología facilitaron la documentación de todas las actividades realizadas.
- ❖ Postgres una gran base de datos que agiliza consultas y transacciones de una manera rápida y efectiva colaboro a la producción de este sistema.
- ❖ El uso de herramientas libres es de gran ayuda al momento de producir software, ya que se cumple con las normas establecidas en el ministerio del deporte de uso de software libre.
- ❖ Al momento de utilizar MVC fue de gran ayuda porque nos permite separar muchos aspectos y brinda al programador reutilización de código y flexibilidad que son importantes en la programación orientada a objetos.

- ❖ El proceso de obtener pases y transferencias de jugadores debe ser claro y transparente para que no exista problemas entre equipos de una determinada liga barrial, ya que la clave del éxito en el deporte es brindar a los usuarios la información exacta de los datos de un determinado jugador.

- ❖ El proyecto ofrece las funcionalidades necesarias para convertir este proyecto en un nuevo producto comercial que puede ser utilizado en cualquier torneo real.

- ❖ A diferencia de otras soluciones los datos utilizados por este programa (jugadores, torneos, partidas, resultados, etc.) son almacenados en una única base de datos obteniendo las siguientes ventajas: orden de la información, búsqueda rápida de datos y un respaldo confiable de la información.

6.2 RECOMENDACIONES

- ❖ Invitar a personas que construyen proyectos a utilizar herramientas de desarrollo RUP ya que permite organizar de manera correcta el proyecto y brindar una clara visión de las necesidades de los usuarios.
- ❖ Por ser RUP una herramienta de trabajo extensa se recomienda determinar un alcance durante el desarrollo del proyecto, ya que utiliza documentos que presentan cambios continuos.
- ❖ Utilizar herramientas de software libre para el desarrollo de sistemas por ser flexibles y se adapta a las necesidades del programador.
- ❖ Conjuntamente con el sistema se debe llevar una constancia de lo que sucede en FEDELIGAS en documentos que deben ser entregados a jugadores y equipos de una determinada liga barrial para garantizar transparencia en todos los procesos que dentro de la misma se llevan a cabo.
- ❖ Seguir contribuyendo con herramientas que sigan engrandeciendo el deporte barrial de la provincia y del país en general.
- ❖ Ampliar la funcionalidad de obtención de reportes con información multimedia de jugadores y gráficos que permitan describir en forma visual el desempeño de los participantes de un torneo.
- ❖ Verificar la portabilidad a otros sistemas de bases de datos.
- ❖ Implementar soporte para nuevos tipos de torneos que podrían servir tanto para fútbol como para otros deportes o eventos.

LIBROS

Ian Sommerville, Ingeniería de Software. Sexta Edición.

Addison Wesley. 2002

Aprendiendo UML en 24 horas.

Schmuller, Joseph. Editorial *Prentice Hall*, México, 2000.

Manual del usuario de PostgreSQL

Thomas Lockhart, 2006.

MANUAL BÁSICO DE POSTGRESQL

Escuela colombiana de ingeniería, Julio Garavito, Laboratorio de informática, Bogotá d.

C.

2007

PROGRAMACIÓN WEB II

I.S.C. BERNARDO MARTÍNEZ SANTIAGO, 2009 RIOVERDE; S.L.P.

PHP PRÁCTICO

Matt Doyle, Anaya multimedia, 2010.

Lenguaje unificado del modelado,

Grady Boch, Ivar Jacobson, Editorial Addison Wesley Madrid, 2006.

Tutorial PostgreSQL

2001, John Worsley, Joshua Drake, Andrew Brookins

Xampp con postgres,

Peter Siegel, 2005.

TESIS

Felipe Andrés Pesantez Loja, Freddy Rolando Tello Crespo,
Diseño de un sistema gestor de inscripciones de estudiantes para el oratorio Don Bosco a través de la metodología rup e implementación de un prototipo, 2008.

Gabriel Bucheli,
Aplicación mixta destinada a la vigilancia de parque informático de la red institucional y atención al personal de ilustre municipio de Ibarra, 2010.

PAGINAS WEB

Introducción al UML

<http://www.yoprogramo.com/articulo4>

Modelos para el desarrollo de *software*

http://docentes.usaca.edu.co/wildiaz/INGSOF_MODELOS

Estereotipos para clases

<http://www-gris.det.uvigo.es/~avilas/UML/node38>

Metodología de desarrollo de sistemas basados en el ciclo de vida

<http://comip.mendoza.gov.ar/metodologia%20para%20el%20desarrollo%20de%20sistemas>

Modelos para el desarrollo de software

http://docentes.usaca.edu.co/wildiaz/INGSOF_MODELOS

Software Legal:

<http://www.softwarelegal.org.ar>

OPEN SOURCE

<http://www.opensource.org>

SOFTWARE LIBRE

<http://www.gnu.org/philosophy/free-sw.es>

GLOSARIO

NETBEANS

Es un IDE, Entorno Integrado de Desarrollo.

UML

Unified Modeling Language. Lenguaje Unificado de Modelado, notación estándar para el modelado de sistemas software.

RUP

(Rational Unified Process.) Proceso Unificado de Racional, metodología del proceso de ingeniería de software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización del desarrollo.

CASOS DE USO

Modela la funcionalidad del sistema agrupándola en descripciones de acciones ejecutadas por un sistema para obtener un resultado.

DIAGRAMAS DE CLASE

Muestra las clases (descripciones de objetos que comparten características comunes) que componen el sistema y cómo se relacionan entre sí.

DIAGRAMAS DE SECUENCIA

Muestra una interacción a lo largo del tiempo, además muestra las instancias participantes en una interacción y su “línea de vida”,

ACTORES O ROLES

Son los personajes encargados de la realización de las actividades definidas dentro de los flujos de trabajo de cada una de las disciplinas del RUP.

COPYLEFT

Es la regla que implica que, cuando se redistribuya el programa, no se pueden agregar restricciones para denegar a otras personas las libertades centrales.

PHP

Procesador de hipertexto.

BIBLIOTECAS QT O GTK+.

GTK+ o **The GIMP Toolkit** es un conjunto de bibliotecas multiplataforma para desarrollar interfaces gráficas de usuario (GUI), principalmente para los entornos gráficos GNOME, aunque también se puede usar en el escritorio de Windows, MacOS y otros.

OBJETO-RELACIONALES

Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia.

JDK

Java Development Kit o (**JDK**), es un software que provee herramientas de desarrollo para la creación de programas en Java. Puede instalarse en una computadora local o en una unidad de red.

GLASSFISH

Es un servidor de aplicaciones de software libre desarrollado por Sun Microsystems

INTRANETS

Una **intranet** es una red de ordenadores privados que utiliza tecnología Internet para compartir dentro de una organización parte de sus sistemas de información y sistemas operacionales.

APPLETS

Un applet es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador web. El applet debe ejecutarse en un contenedor, que lo proporciona un programa anfitrión, mediante un plugin, o en aplicaciones como teléfonos móviles que soportan el modelo de programación por 'applets'.

USUARIO

Se denomina así a la persona que tiene derechos especiales en algún servicio de Internet por acreditarse en el mismo mediante un identificador y una clave de acceso, obtenidos con previo registro en el servicio, de manera gratuita o de pago.

FLUJO DE EVENTOS

Al flujo de eventos lo podemos relacionar con un dialogo. Línea a línea, el flujo de eventos indica quien habla y qué dice. La secuencia se suele iniciar con algo dicho por el actor, y se continua intercalando sucesivamente lo hecho por el sistema con lo dicho por el actor. A cada una de estas líneas o indicaciones, le podemos llamar *paso*.

PRIORIDAD

Elemento cualitativo que determina la máxima preferencia; se utiliza en planeación o programación para señalar lo que tiene mayor importancia y que por consiguiente requiere de mayor atención.

ARQUITECTURA DEL SOFTWARE

La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos. Toda arquitectura debe ser implementable en una arquitectura física, que consiste simplemente en determinar qué computadora tendrá asignada cada tarea.

INDICADOR

El instrumento que sirve para mostrar o indicar algo e indica una dirección en cualquier otro contexto.

MITIGACION

Es la reducción de la vulnerabilidad, es decir la atenuación de los daños potenciales sobre la vida y los bienes causados por un evento.

PLAN DE CONTINGENCIA

Un Plan de contingencias es un instrumento de gestión para el buen gobierno de las Tecnologías de la Información y las Comunicaciones en el dominio del soporte y el desempeño.

NOMBRE DE LA TABLA: CAMPEONATO

Descripción: Tabla desarrollada para llevar el control de campeonatos que se realicen en la federación de ligas barriales y parroquiales de Imbabura.

Resumen de columnas:

Tabla A. 1.- Tabla que permite llevar el control de campeonatos

Nombre	Tipo de datos	Valor nulo	Valor default
local	integer	No	
gollo	integer	No	
visitante	integer	No	
golvi	integer	No	
fecha	date	No	
fg	character varying(45)	Si	DEFAULT 1

Detalle de columnas:

local: Número que identifica el equipo local.

gollo: Número de goles del equipo local.

visitante: Número que identifica el equipo local.

golvi: Número de goles del equipo visitante.

fecha: Fecha en la que se jugó dicho partido.

fg: Fecha de juego a la que pertenece ese partido.

NOMBRE DE LA TABLA: CATEGORIA

Descripción: Tabla desarrollada para llevar el control de las categorías que existen en los diferentes torneos realizados a lo largo de los años en la federación de ligas barriales y parroquiales de Imbabura.

Resumen de columnas:

Tabla A. 2.- Tabla que permite llevar el control de categorías.

Nombre	Tipo de datos	Valor nulo	Valor default
id_categoria	integer	No	
nombre	character varying(45)	No	

Detalle de columnas:

id_categoria: Identificador de tabla que Indica el número de categoría.

nombre: Indica el nombre de categoría.

Claves:

categoria_pkey PRIMARY KEY (id_categoria)

NOMBRE DE LA TABLA: DATOS_LIGAS

Descripción: Tabla desarrollada para contener datos sobre la directiva y contactos de cada liga barrial.

Resumen de columnas:

Tabla A. 3.- Tabla que contiene datos sobre la directiva y contactos de cada liga

Nombre	Tipo de datos	Valor nulo	Valor default
id_liga	integer	No	
presidente	character varying(100)	Si	
vicepresidente	character varying(100)	Si	
secretario	character varying(100)	Si	
tesorero	character varying(100)	Si	
vocal_principal1	character varying(100)	Si	
vocal_principal2	character varying(100)	Si	
vocal_principal3	character varying(100)	Si	
vocal_suplente1	character varying(100)	Si	
vocal_suplente2	character varying(100)	Si	
vocal_suplente3	character varying(100)	Si	
telf_email	character varying(200)	Si	

Detalle de columnas:

id_liga: Identificador de la tabla liga.

presidente: Contiene el nombre del presidente de la liga barrial.

vicepresidente:	Contiene el nombre del vicepresidente de la liga barrial.
secretario:	Contiene el nombre del secretario de la liga barrial.
tesorero:	Contiene el nombre del tesorero de la liga barrial.
vocal_principal1:	Contiene el nombre del vocal_principal1 de la liga barrial.
vocal_principal2:	Contiene el nombre del vocal_principal2 de la liga barrial.
vocal_principal3:	Contiene el nombre del vocal_principal3 de la liga barrial.
vocal_suplente1:	Contiene el nombre del vocal_suplente1 de la liga barrial.
vocal_suplente2:	Contiene el nombre del vocal_suplente2 de la liga barrial.
vocal_suplente3:	Contiene el nombre del vocal_suplente3 de la liga barrial.
telf_email:	Contiene el teléfono y el correo electrónico de la liga barrial.

Claves:

pk_idliga	PRIMARY KEY (id_liga),
pk_dato	FOREIGN KEY (id_liga) REFERENCES liga (id_liga) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION

NOMBRE DE LA TABLA: EQUIPO

Descripción: Tabla desarrollada para llevar el control de los equipos que pertenecen a las ligas barriales y parroquiales de Imbabura.

Resumen de columnas:

Tabla A. 4.- Tabla que permite llevar el control de los equipos.

Nombre	Tipo de datos	Valor nulo	Valor default
id_equipo	integer	No	
nombre	character varying(45)	No	
id_liga	integer	Si	
id_categoria	integer	Si	
id_series	integer	Si	
interligas	Carácter(1)	Si	

Detalle de columnas:

id_equipo: Identificador de equipos en esta tabla.

nombre: Indica el nombre de equipo.

id_liga: Identificador de la tabla liga.

id_categoria: Identificador de la tabla categoría.

id_series: Identificador de la tabla series.

interligas: Muestra “s” si pertenece a al campeonato interligas o “n” si no pertenece.

Claves:

equipo_pkey	PRIMARY KEY (id_equipo),
equipo_id_categoria_fkey	FOREIGN KEY (id_categoria) REFERENCES categoria (id_categoria) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION,
equipo_id_liga_fkey	FOREIGN KEY (id_liga) REFERENCES liga (id_liga) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION,
pk_serie	FOREIGN KEY (id_series) REFERENCES series (id_series) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION

Index:

fki_catego	ON equipo USING btree (id_categoria);
fki_equi_liga	ON equipo USING btree (id_liga);

NOMBRE DE LA TABLA: EQUIPO_PARTIDO

Descripción: Tabla desarrollada para llevar el control de las tablas de posiciones de los equipos en cualquier serie.

Resumen de columnas:

Tabla A. 5.- Tabla que permite llevar el control de las tablas de posiciones.

Nombre	Tipo de datos	Valor nulo	Valor default
id_partidos	integer	Si	
puntos	integer	Si	
goles_diferencia	integer	Si	
id_equipo	integer	No	

Detalle de columnas:

id_partidos: Identificador de la tabla partidos.

puntos: Lleva el control de los puntos en la tabla de posiciones.

goles_diferencia: Lleva el control de los goles diferencia en la tabla de posiciones.

id_equipo: Identificador de la tabla equipo.

Claves:

pk_equi PRIMARY KEY (id_equipo),

equipo_partido_id_partidos_fkey FOREIGN KEY (id_partidos)
REFERENCES partidos (id_partidos)
MATCH SIMPLE ON UPDATE NO
ACTION ON DELETE NO ACTION,

pk_eqipar FOREIGN KEY (id_equipo)
REFERENCES equipo (id_equipo)
MATCH SIMPLE ON UPDATE NO
ACTION ON DELETE NO ACTION

Index:

fki_partidos ON equipo_partido USING btree
(id_partidos);

NOMBRE DE LA TABLA: ESTADIO

Descripción: Tabla desarrollada para llevar el control de los diferentes estadios que pertenecen a las ligas barriales y parroquiales de Imbabura.

Resumen de columnas:

Tabla A. 6.- Tabla que permite llevar el control de los estadios.

Nombre	Tipo de datos	Valor nulo	Valor default
id_estadios	integer	No	
nombre_estadio	character varying(200)	Si	DEFAULT NULL::character varying
direccion	character varying(300)	Si	DEFAULT NULL::character varying

Detalle de columnas:

id_estadio: Identificador de tabla.

nombre_estadio: Indica el nombre del estadio.

direccion_estadio: Indica la dirección donde se encuentra situado el estadio.

Claves:

estadio_pkey PRIMARY KEY (id_estadios)

NOMBRE DE LA TABLA: ETAPAS

Descripción: Tabla desarrollada para llevar el control de las etapas del campeonato de cualquier serie.

Resumen de columnas:

Tabla A. 7.- Tabla que permite llevar el control de las etapas.

Nombre	Tipo de datos	Valor nulo	Valor default
id_etapas	integer	No	
nombre_etapa	character varying(45)	Si	DEFAULT NULL::character varying
estado	character (1)	Si	DEFAULT NULL::bpchar,

Detalle de columnas:

id_etapas: Identificador de tabla.

nombre_etapa: Indica el nombre de la etapa.

estado: Indica si la etapa esta activa o si ya se pasó a la segunda etapa.

Claves:

etapas_pkey PRIMARY KEY (id_etapas)

NOMBRE DE LA TABLA: GOLEADORES

Descripción: Tabla desarrollada para llevar el control de los goleadores del campeonato de cualquier serie.

Resumen de columnas:

Tabla A. 8.- Tabla que permite llevar el control de los goleadores.

Nombre	Tipo de datos	Valor nulo	Valor default
id_jugador	integer	No	
goles	integer	Si	
Camiseta_num	integer	No	

Detalle de columnas:

id_jugador: Identificador de tabla jugador

goles: Indica el número de goles que tiene un jugador.

camiseta_num: Indica el número de camiseta que viste un jugador.

Claves:

pk_gol PRIMARY KEY (id_jugador),

pk_goleador FOREIGN KEY (id_jugador) REFERENCES jugador (id_jugador) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION.

NOMBRE DE LA TABLA: JUGADOR

Descripción: Tabla desarrollada para llevar el control de los jugadores del campeonato de cualquier serie.

Resumen de columnas:

Tabla A. 9.- Tabla que permite llevar el control de los jugadores.

Nombre	Tipo de datos	Valor nulo	Valor default
id_jugador	integer	No	
cedula	character varying(10)	No	
nombres	character varying(200)	No	
apellido	character varying(200)	No	
profesion	character varying(100)	Yes	DEFAULT NULL::character varying
provincia	character varying(100)	No	
canton	character varying(100)	No	
fecha_nacimiento	character varying(50)	No	DEFAULT NULL::character varying
habilitado	character varying(10)	No	
id_equipo	integer	No	
cod_jugador	character varying(20)	No	
telf_email	character varying(200)	Si	
prestado	integer	Si	
Fecha_sancion	character varying(50)		
foto	character varying(200)		

Detalle de columnas:

id_jugador:	Identificador de tabla.
cedula:	Indica la cedula del jugador.
nombres:	Indica los nombres completos del jugador.
apellido:	Indica los apellidos completos del jugador.
profesión:	Indica la profesión del jugador.
provincia:	Indica la provincia en donde nació el jugador.
canton:	Indica el cantón donde nació el jugador.
fecha_nacimiento:	Indica la fecha de nacimiento del jugador.
habilitado:	Indica si el jugador está o no habilitado.
id_equipo:	Identificador de la tabla equipo.
cod_jugador.	Indica la el código del jugador de acuerdo a la liga barrial.
telf_email:	Indica el teléfono y correo electrónico del jugador.
Prestado	indica si el jugador está a préstamo en algún equipo.
Fecha_sancion	indica la fecha en la que fue sancionado un jugador.
Foto	indica la foto de un determinado jugador.

Claves:

jugador_pkey	PRIMARY KEY (id_jugador),
jugador_id_equipo_fkey	FOREIGN KEY (id_equipo) REFERENCES equipo (id_equipo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION,
ced_ukey	UNIQUE (cedula),

préstamo FOREIGN KEY (prestado)
 REFERENCES equipo (id_equipo) MATCH
 SIMPLE
 ON UPDATE NO ACTION ON DELETE NO
 ACTION

Index:

fki_juga_equipo ON jugador USING btree (id_equipo);

NOMBRE DE LA TABLA: LIGA

Descripción: Tabla desarrollada para llevar el control de las ligas del deporte barrial.

Resumen de columnas:

Tabla A. 10.- Tabla que permite llevar el control de las ligas.

Nombre	Tipo de datos	Valor nulo	Valor default
id_liga	integer	No	
nombre	character varying(45)	No	
tipo	character varying(45)	Si	

Detalle de columnas:

id_liga: Identificador de tabla.
 nombre: Indica el nombre de la liga barrial.
 tipo: Indica si es liga o pre-liga.

Claves:

liga_pkey PRIMARY KEY (id_liga)

NOMBRE DE LA TABLA: NOTICIAS

Descripción: Tabla desarrollada para llevar el control de las noticias más destacadas del deporte barrial.

Resumen de columnas:

Tabla A. 11.- Tabla que permite llevar el control de las noticias.

Nombre	Tipo de datos	Valor nulo	Valor default
fecha	date	No	
titulo_noticia	character varying(1000)	No	
noticia	character varying(10000)	No	
id_noticia	integer	No	

Detalle de columnas:

fecha: Indica la fecha de publicación de una noticia.

titulo_noticia: Indica el título de la noticia.

noticia: Indica el contenido de la noticia.

id_noticia: Identificador de la tabla.

Claves:

pk_not PRIMARY KEY (id_noticia)

NOMBRE DE LA TABLA: SERIES

Descripción: Tabla desarrollada para llevar el control de las series que tiene cada liga barrial.

Resumen de columnas:

Tabla A. 12.- Tabla que permite llevar el control de las series.

Nombre	Tipo de datos	Valor nulo	Valor default
id_series	integer	No	
nombre	character(1)	No	
observacion	character varying(200)	Si	DEFAULT NULL::character varying,

Detalle de columnas:

id_series: Identificador de tabla.

nombre: Indica el nombre de la serie.

observación: Indica las observaciones que existan en cada serie.

Claves:

series_pkey PRIMARY KEY (id_series)

NOMBRE DE LA TABLA: TRANSFERENCIA

Descripción: Tabla desarrollada para llevar el control de las transferencias de jugadores ya sean definitivos o a préstamo de una o varias ligas.

Resumen de columnas:

Tabla A. 13.- Tabla que permite llevar el control de las transferencias.

Nombre	Tipo de datos	Valor nulo	Valor default
id_transferencia	integer	No	
fecha	varying(45)	No	
id_jugador	integer	No	DEFAULT NULL::character varying,
tipo	character varying(45)	Si	DEFAULT NULL::character varying
observaciones	character varying(45)	Si	DEFAULT NULL::character varying
caduca	character varying(45)	Si	DEFAULT NULL::character varying
estado	character varying(10)	Si	DEFAULT NULL::character varying
equipo_nuevo	integer	No	
equipo_anterior	integer	No	
tiempo	integer	No	

Detalle de columnas:

id_transferencia: Identificador de tabla.

fecha: Indica la fecha en que se realizó una transferencia.

id_jugador: Identificador de tabla jugador que indica el jugador a transferir.

tipo: Indica el tipo de pase si es definitivo o préstamo.

observaciones: Indica si el pase se lo realiza en la misma liga o en otra.

caduca:	Indica el tiempo en que el pase se caduca.
estado:	Indica si el pase del jugador está activo para poder jugar.
equipo_nuevo:	Indicador de tabla equipo que indica el nombre del equipo al que pertenecerá el jugador antes mencionado.
equipo_anterior:	Indicador de tabla equipo que indica el nombre del equipo al que pertenecía el jugador antes mencionado.
tiempo:	Indica el tiempo en el que será prestado el jugador.

Claves:

transferencia_pkey	PRIMARY KEY (id_transferencia),
trans_an_equipo	FOREIGN KEY (equipo_anterior) REFERENCES equipo (id_equipo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION,
trans_nu_equipo	FOREIGN KEY (equipo_nuevo) REFERENCES equipo (id_equipo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION,
transferencia_id_jugador_fkey	FOREIGN KEY (id_jugador) REFERENCES jugador (id_jugador) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION

Index:

fki_transf	ON transferencia USING btree (id_jugador);
------------	--

NOMBRE DE LA TABLA: USUARIOS

Descripción: Tabla desarrollada para llevar el control de los usuarios del sistema.

Resumen de columnas:

Tabla A. 14.- Tabla que permite llevar el control de las cuentas de usuarios.

Nombre	Tipo de datos	Valor nulo	Valor default
nick	character varying(45)	No	
pass	character varying(45)	No	
nivel	integer	No	

Detalle de columnas:

nick: Indica el nombre del usuario.

pass: Indica el password del usuario encriptado y en formato de clave.

nivel: Indica el nivel del usuario.