



UNIVERSIDAD JOSÉ CARLOS MARIÁTEGUI

VICERRECTORADO DE INVESTIGACIÓN

**FACULTAD DE INGENIERÍA Y
ARQUITECTURA**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E
INFORMÁTICA**

TRABAJO DE SUFICIENCIA PROFESIONAL

**REPLICACIÓN EN APLICACIONES DISTRIBUIDAS Y
SU APLICACIÓN AL SISTEMA ACADÉMICO EN
EDUCACIÓN TÉCNICA PRODUCTIVA
CETPRO “CÉSAR VALLEJO” DE ILO**

PRESENTADO POR

BACHILLER HEBER JESUS CHAVEZ CHOQUE

PARA OPTAR TÍTULO PROFESIONAL DE

INGENIERO DE SISTEMAS E INFORMÁTICA

MOQUEGUA – PERÚ

2016

RESUMEN

Actualmente las empresas están relacionadas con el avance tecnológico; la gran influencia de este desarrollo se debe al internet, llegando cada vez a más personas con el negocio digital, que viene siendo importante por las diferentes opciones de comprar y vender productos de cualquier país y en cualquier momento. Debido a esto las operaciones en los bancos a nivel mundial tuvieron la necesidad de acoplarse y decidieron inventar medidas de seguridad como: firmas electrónicas, huellas digitales y certificados digitales. Asimismo, apreciamos que las empresas hoy en día, se renuevan constantemente con tecnología variable, buscando ser distintos a otras y perdurar en el tiempo. Por tal motivo, el desarrollo de los negocios sufrió un gran cambio, logrando tener más llegada a nuevas compañías y un próspero desarrollo de software más eficaz, competente, optimiza el trámite de matrícula y manejo de notas, un sistema de gestión de matrículas, basándose en procedimientos almacenados. A raíz de esto surge la interacción óptima en distintas compañías a través de la comunicación ofrecida por el internet, ya sea compañías particulares, del estado o transnacionales.

Palabras claves: Simulación, prototipo, interfaz, matrícula, procedimiento almacenado, entorno distribuido, modelo iterativo.

CONTENIDO

Portada	
Página de Jurado	II
Reflexión Inicial	III
Dedicatoria	IV
Agradecimientos	V
Resumen	VI
Contenido	VII
Índice de Tablas	X
Índice de Figuras	XI

CAPÍTULO I

Introducción	1
--------------	---

CAPÍTULO II

OBJETIVOS Y METODOLOGÍA

2.1 Objetivo General	2
2.2 Objetivos específicos	2
2.2.1 Justificación	3

CAPÍTULO III

DESARROLLO DEL TEMA

3.1 Conceptos Generales	4
3.2 Metodología	4
3.2.1 Método	4
3.2.2 Técnica	5
3.2.3 Instrumentos	5
3.3 Descripción Operativa del Proyecto	6
3.3.1 Visión	6
3.3.2 Alcances	7

3.3.3	Historia y Escenarios	17
3.3.3.1	Escenario N° 1: Requerimiento de Matrícula	17
3.3.3.2	Escenario N° 2: Gestión Matrícula	19
3.3.3.3	Escenario N° 3: Autorizar Matrícula	21
3.3.3.4	Escenario N° 4: Administra Información Académica	23
3.3.3.5	Escenario N° 5: Entrega Reporte de Notas	25
3.3.3.6	Escenario N° 6: Registra Matrícula	27
3.3.3.7	Escenario N° 7: Entregar Notas	29
3.3.3.8	Escenario N° 8: Administra Información Académica	30
3.3.3.9	Escenario N° 9: Ingresar Notas	32
3.4	Marco teórico	37
3.4.1	Tecnología Informática IT	37
3.4.1.1	Cliente/Servidor	37
3.4.1.2	Arquitectura	38
3.4.1.3	Características	40
3.4.1.4	Desventajas de la Arquitectura Cliente/Servidor	41
3.4.2	Aplicaciones Distribuidas	42
3.4.2.1	Concepto de Computación Distribuida	42
3.4.2.2	Objetivos	43
3.4.3	Replicación de Entornos Distribuido	43
3.4.3.1	Inicios de la Replicación de Datos	46
3.4.4	Como saber si se Necesita Replicación	46
3.4.5	Consideraciones Generales Para Replicación	48
3.4.6	Arquitecturas Comunes de Replicación	58
3.4.7	Elementos de Replicación	49
3.4.7.1	Replicación Sincrónica Two Phase Commit	50
3.4.7.2	Replicación Asincrónica	51
3.4.8	Detección y Resolución de Conflictos	54
3.4.8.1	Mecanismo de Control de Concurrencia	55
3.4.8.2	Conflictos en la Modificación de Datos	56
3.4.8.3	Detección de Conflictos	57
3.4.9	Distribución vs Replicación	58
3.5	Representación de Resultados	59

3.5.1 Evaluación del Sistema	59
3.5.1.1 Implementación Funcional	59
3.5.1.2 Test y Pruebas	60
3.5.1.3 Estabilización	60
3.5.1.4 Control de Calidad	60
3.5.1.5 Desarrollo de Planes	60
3.5.1.6 Despliegue	60

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones	61
4.2. Recomendaciones	62
Bibliografía	63
Apéndice	65

ÍNDICE DE TABLAS

Tabla 1. Fases del Proyecto y sus Entregables	8
Tabla 2. Definición de Roles	9
Tabla 3. Definición de Roles	10
Tabla 4. Detalle de Fases y Costos	11
Tabla 5. Herramientas Estándar, Modelado y Desarrollo	13
Tabla 6. Especificaciones de Hardware, Servidor	14
Tabla 7. Especificaciones de Software, Servidor	15
Tabla 8. Especificaciones de Hardware, Usuario Final	16
Tabla 9. Especificaciones de Software, Usuario Final	16

ÍNDICE DE FIGURAS

Figura 1. Método Iterativo e Incremental	5
Figura 2. Línea de Tiempo del Proyecto diagrama de Gantt	12
Figura 3. Diagrama Caso de Uso - Requerimiento de Matrícula	17
Figura 4. Especificación de Casos de Uso - Requerimiento de Matrícula	17
Figura 5. Diagrama de Estado - Requerimiento de Matrícula	18
Figura 6. Diagrama de Secuencia - Requerimiento de Matrícula	18
Figura 7. Diagrama de Actividades - Requerimiento de Matrícula	18
Figura 8. Diagrama Caso de Uso - Gestionar Matrícula	19
Figura 9. Especificación de Casos de Uso - Gestionar Matrícula	19
Figura 10. Diagrama de Estado - Gestionar Matrícula	20
Figura 11. Diagrama de Secuencia - Gestionar Matrícula	20
Figura 12. Diagrama de Actividades - Gestionar Matrícula	21
Figura 13. Diagrama Caso de Uso - Autorizar Matrícula	21
Figura 14. Especificación de Casos de Uso - Autorizar Matrícula	22
Figura 15. Diagrama de Estado - Autorizar Matrícula	22
Figura 16. Diagrama de Secuencia - Autorizar Matrícula	22
Figura 17. Diagrama de Actividades - Autorizar Matrícula	23
Figura 18. Diagrama Caso de Uso - Adm. Información Académica	23
Figura 19. Especificación de Casos de Uso-Adm. Información Académica	23
Figura 20. Diagrama de Estado - Adm. Información Académica	24
Figura 21. Diagrama de Secuencia - Adm. Información Académica	24
Figura 22. Diagrama de Actividades - Adm. Información Académica	24
Figura 23. Diagrama Caso de Uso - Entrega Reporte de Notas	25
Figura 24. Especificación de Casos de Uso - Entrega Reporte de Notas	25
Figura 25. Diagrama de Estado - Entrega Reporte de Notas	25
Figura 26. Diagrama de Secuencia - Entrega Reporte de Notas	26
Figura 27. Diagrama de Actividades - Entrega Reporte de Notas	26
Figura 28. Diagrama de Caso de Uso - Registra Matrícula	27
Figura 29. Especificación de Casos de Uso - Registra Matrícula	27
Figura 30. Diagrama de Estado - registra matrícula	27
Figura 31. Diagrama de Secuencia - registra matrícula	28
Figura 32. Diagrama de Actividades - registra matrícula	28
Figura 33. Diagrama Caso de Uso - entrega de notas	29

Figura 34. Especificación de Casos de Uso - Reportar Notas	29
Figura 35. Diagrama de Estado - Entrega de Notas	29
Figura 36. Diagrama de Secuencia - Entrega de Notas	30
Figura 37. Diagrama de Actividades - Entrega de Notas	30
Figura 38. Diagrama Caso de Uso - Adm. Información Académica	30
Figura 39. Especificación de Casos de Uso-Adm. Información Académica	31
Figura 40. Diagrama de Estado - Adm. Información Académica	31
Figura 41. Diagrama de Secuencia - Adm. Información Académica	31
Figura 42. Diagrama de Actividades - Adm. Información Académica	32
Figura 43. Diagrama Caso de Uso - Ingreso de Notas	32
Figura 44. Especificación de Casos de Uso - Ingresar Notas	33
Figura 45. Diagrama de Estado - Ingreso de Notas	33
Figura 46. Diagrama de Secuencia - Ingreso de Notas	33
Figura 47. Diagrama de Actividades - Ingreso de Notas	34
Figura 48. Diagrama Caso de Uso - Proceso de Registro de Matrícula	34
Figura 49. Diagrama Caso de Uso - Proceso de Control de Notas	35
Figura 50. Diagrama de Despliegue - Matrícula y Notas	35
Figura 51. Diagrama de Clases - Matrícula y Notas	36
Figura 52. Arquitectura Distribuida	43
Figura 53. Enfoque de la Distribución	45
Figura 54. Distribución	50
Figura 55. Consolidación de Datos	50
Figura 56. Two – Phase Commit	51

CAPÍTULO I

INTRODUCCIÓN

En la actualidad las bases de datos relacionales son más utilizadas en muchos campos de aplicaciones para la investigación. Su forma estructural de datos básica nos indica que, la tabla relacional, es conveniente para muchas aplicaciones frecuentes. Los ejemplos de consideración práctica son los sistemas de apoyo al diseño industrial, las bases de datos científicos, la base multimedia. Las bases de datos orientadas a objetos pretenden solucionar los problemas, añadiendo las siguientes características: Acogen como ejemplo de datos el de los lenguajes dirigidos a objetos, aceptando así el uso de estructuras de datos tan complicadas y eliminando en su mayoría las barreras entre el desarrollo de aplicaciones y la gestión de datos. Admiten la ampliación con nuevos tipos de datos complejos, permitiendo añadir operaciones arbitrarias sobre ellos. Estas características mencionadas han estimulado el desarrollo de diversos sistemas dirigidos a objetos. Por consiguiente, la actual emergencia de aplicaciones web que necesiten estructuras de datos adaptables está estimulando, un gran interés en sistemas de mantenimiento, como es el caso de las bases de datos dirigidas a objetos.

1. CAPÍTULO II

OBJETIVOS

2.1. Objetivo general

Crear una solución que permitan gestionar de forma rápida y optima el trámite de matrícula y manejo de notas de la institución técnica productiva CETPRO “César Vallejo” considerándose necesario la elaboración de un sistema de gestión de matrículas, basándose en procedimientos almacenados y el empleo de técnicas de replicación en entornos distribuidos, aplicaciones n-Niveles y el manejo de la metodología UML, para el diseño de sistemas.

2.2. Objetivos Específicos

- a) **Diseñar de la aplicación:** Diseño ventanas de usuario o formularios propuestos, diagramas UML, diagramas E-R.
- b) **Establecer modelo de datos:** Unificar nombres de campos y tablas de la base de datos, diccionario de datos, diseño lógico de BD.
- c) **Determinar plataforma Tecnológica:** Herramientas para preparación de datos y motor de base.
- d) **Asegurar:** celeridad, protección de información o datos a tiempo real.
- e) **Brindar:** a través de un software de formación de aplicaciones cliente/Servidor la mejor interfaz gráfica para que sea manejable y confiable para todas las personas que utilizaran este tipo de aplicación y será un beneficio propio.

2.2.1. Justificación

Las dificultades actuales del CETPRO César Vallejo, son principalmente el llenado de datos, proceso y la información de los estudiantes. En la actualidad para el contenido de gestión académica se usan algunos tipos de archivos en Excel o manuscritos, esto se ve restringido por ciertas irregularidades y el tiempo que se hace en las consultas son muy extensos, estos datos son revisados por los usuarios en Excel.

Entre los diversos problemas que hay en la actualidad:

- ✓ Información incongruente y repetida.
- ✓ La información que se registra en ese momento está ligada a la seguridad, permitiendo el acceso todas las personas que entran al computador.
- ✓ El personal que tiene la misión de llamar a cada uno de los clientes (padres u tutor), necesitan la información exacta de los clientes con los datos actualizados y completos.
- ✓ Dentro de las diferentes funciones de la secretaria es de generar las matrículas del alumnado, esta información la realiza a través del programa Word, hojas de cálculo (Excel), copiando la información de acuerdo con sus necesidades.
- ✓ Se requiere un control de los pagos por pensiones y matrículas.

3. CAPÍTULO III

DESARROLLO DEL TEMA

3.1. Conceptos Generales

3.2. Metodología

3.2.1. Método

Se basa en el método iterativo e incremental, presentado las siguientes características:

- ✓ Es un desarrollo ascendente basado en una abstracción de clases en aumento.
- ✓ No se ajusta a la realidad, sino en la manera en que las personas entiendan y asimilen la realidad.
- ✓ Se puede utilizar muchas veces (reutilizable)
- ✓ Su base es el reconocimiento de objetos, definición y estructuración.

Las diferentes actividades utilizadas para el desarrollo del sistema de matrículas nos brindan una guía para:

- ✓ Identificar actividades que se darán en la formulación de proyecto de sistema de información.
- ✓ Unir criterios en la estructuración para el desarrollo de sistemas informáticos.
- ✓ Brindar indicadores de supervisión y revisión.

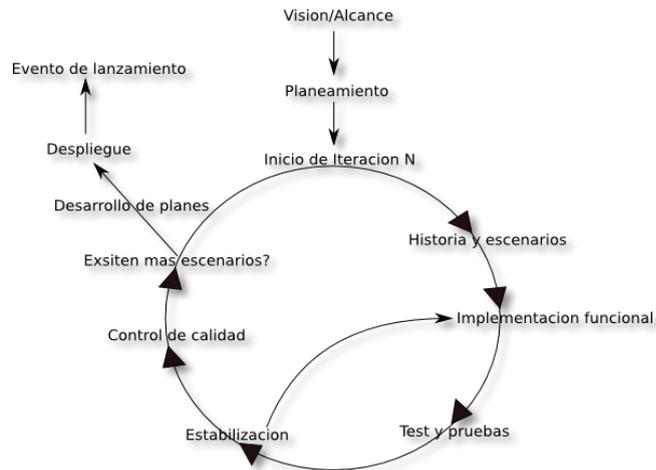


Figura 1. Método Iterativo e Incremental
Fuente: Pressman, 2005

3.2.2. Técnicas

3.2.2.1. Programación orientada a objetos (POO)

Es un tipo de programación que usa objetos y sus relaciones para crear programas y aplicaciones informáticas. Se basa en diferentes técnicas, abarcando herencia, abstracción, polimorfismo y encapsulamiento. Su utilización se volvió famoso al inicio de la década de los años 90. Actualmente, existe diferentes lenguajes de programación que se adaptan la orientación a objetos. Nuestra metodología se dirige a utilizar las técnicas del POO¹ (Yourdon, E., (1991) Object-Oriented Design.).

3.2.3. Instrumentos

- a) **Observación:** Para la realización de nuestra metodología necesitamos conocer e interpretar que es la observación.

Es una técnica que consiste en observar atentamente el fenómeno, hecho o caso, tomar información y registrarla para su posterior análisis. Hay dos clases de observación: la Observación no científica y la observación

¹ La Programación Orientada a Objetos - es un paradigma de programación que usa objetos y sus interacciones.

científica. La diferencia entre ambas está en la intencionalidad: observar científicamente significa observar con un objetivo claro, definido y preciso: el investigador sabe qué es lo que desea observar y para qué quiere hacerlo, lo cual implica que debe preparar meticulosamente la observación. Observar no científicamente significa observar sin intención, sin objetivo definido y, por tanto, sin preparación previa (Puente, 2000:1)

- b) **Rational Rose Enterprise 2003:** Herramienta CASE² para modelar todos los componentes del proceso de desarrollo de aplicaciones.
- c) **NetBeans IDE 8.2:** Esta herramienta la usamos para realizar la codificación de nuestra aplicación en lenguaje Java³.
- d) **MySQL Server:** Se usó esta herramienta para crear nuestra base de datos relacional.
- e) **SQLyog:** Herramienta CASE, para administrar la base de datos.

3.3. Descripción Operativa del Proyecto

Tiene una síntesis del proyecto, equipo de trabajo, definición del problema, tamaño del proyecto, roles y responsabilidades.

3.3.1. Visión

Nos da una descripción del propósito, objetivos y alcance total del proyecto, instaurando los entregables que serán realizados y usados durante la ejecución del proyecto.

² Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora.

³ Es un lenguaje de programación de propósito general, concurrente, orientado a objetos

3.3.2. Alcances

Con el fin de tener clara la presentación, se ha dividido en dos fases; alcances del proyecto y alcances específicos de la solución para la institución CETPRO “César Vallejo”.

El Plan de Desarrollo del Software detalla de manera general, el plan que será utilizado para el desarrollo del Sistema de Matrícula. En el proceso de desarrollo en lo concerniente a la Visión se definen las características del producto a desarrollar desde la perspectiva del cliente.

Para la versión 1.0 del plan de desarrollo del software, se enfoca principalmente en la identificación de los roles y responsabilidades, también como de los requisitos brindados por los participantes en el trámite de matrícula, en el inicio del proyecto y durante la fase de inicio se dará la primera versión.

El avance del proyecto tendrá un seguimiento continuo e iterativo que ocasionará ajustes, que serán de mucha importancia pues permitirán mantenernos en una constante retroalimentación, y por tal obtener un mejor producto final en cada proceso de iteración del producto.

La implementación de la plataforma requiere de la ejecución de ciertas actividades, éstos se encuentran detallados a continuación.

Tabla 1
Fases del Proyecto y sus Entregables

Fase	Entregables
Visión –alcance (planeamiento)	Documento visión-alcance aprobado, planeamiento.
Iteraciones (Análisis, Diseño y Desarrollo del Sistema)	Historias/escenarios, diagramas E-R, vista de formularios, diagramas UML. Ejecutables.
Doc. del Sistema (Desarrollo de planes de contingencia, despliegue, soporte, formación)	Plan de contingencia, plan de despliegue, plan de soporte, plan de formación.
Despliegue (implementación y mantenimiento de la aplicación.)	Todas las pruebas de conectividad y planes anteriormente diseñados

Fuente: Elaboración Propia.

- ✓ **Definición de roles.** - Se identifican roles en el desarrollo del proyecto, que serán cubiertos en algunos casos por miembros de este equipo de trabajo. Esos roles son los siguientes:

Tabla 2
Definición de Roles

Rol	Responsabilidades	Responsables
Director de proyecto	<ul style="list-style-type: none"> - Conducción del Proyecto - Responsable de brindar cumplimiento a definición y tiempo de entrega de las especificaciones establecidas. - Establecer la arquitectura. - Garantizar los procesos y servicios administrativos. - Establecer las especificaciones funcionales de los servicios a realizar. 	Heber Chávez
Dirección del Programa	<ul style="list-style-type: none"> - Asegurar que el equipo concrete la visión formulada del proyecto. - Responsable de este rol, y por ello debe tener conocimiento de la realidad del cliente y así como el criterio para priorizar los ítems a implementar. 	Heber Chávez
Desarrollador	<ul style="list-style-type: none"> - Ejecutar la implantación y completar las funcionalidades del producto tanto en arquitectura como en diseño. 	Wilmer Carbajal

Fuente: Elaboración Propia.

Tabla 3
Definición de Roles

Rol	Responsabilidades	Responsables
Encargados de pruebas	- Validan la calidad y la adecuada operación de los servicios entregados, así como la documentación.	Heber
	- Inician la operación en el momento en que se autoriza y apertura el proceso de desarrollo.	Chávez
Educación de Usuario	- Define las necesidades de instrucción.	Heber
	- Calidad de la información que fundamenta del proyecto.	Chávez
Dirección de lanzamiento	- Dispone y distribuye los recursos necesarios para que el proyecto sea ejecutado.	
	- Planifica las actividades externas al equipo humano de trabajo, tales como la instrucción de los usuarios ó la conceptualización de la plataforma de hardware y software.	Wilmer Carbajal

Fuente: Elaboración Propia.

- ✓ **Planeamiento.** - Esta parte del proyecto implica tener uno o varios objetivos a realizar junto con las acciones requeridas, a fin de concluirse exitosamente el proyecto a realizar. Contiene: Desarrollo de planes estimaciones iniciales de costos, uso de recursos humanos, materiales,

descripción, Instalación, plan de trabajo, diagrama de Gantt, Procesos de negocio.

Estimaciones iniciales: Costos y Gastos

Tabla 4

Detalle de Fases y Costos

PROYECTO	FASE	ENTREGABLE	MONTO S/.	
SISTEMA DE CONTROL DE MATRICULA/ NOTAS	Planteamiento del Sistema	Informe de Planteamiento del Sistema		
	Total Fase		894.00	894.00
	Análisis del Sistema	Informe de Análisis del Sistema		
	Total Fase		344.00	344.00
	Diseño del Sistema	Informe del Diseño del Sistema		
	Total Fase		1,090.00	1,090.00
	Desarrollo del Sistema	Informe del Desarrollo del Sistema		
	Total Fase		1,878.67	1,878.67
	Documentación del sistema	Informe de Documentación del sistema		
	Total Fase		1,878.67	1,878.67
	Implementación del sistema	Informe de Implementación del Sistema		
	Total Fase		880.00	880.00
	Mantenimiento del sistema	Informe de Mantenimiento del Sistema		
	Total Fase		192.00	192.00
TOTAL FASES			5,366.67	
Reserva de Contingencia			1,000.00	
Reserva de Gestión			1,000.00	
PRESUPUESTO TOTAL DEL PROYECTO			7,366.67	

Fuente: Elaboración Propia.

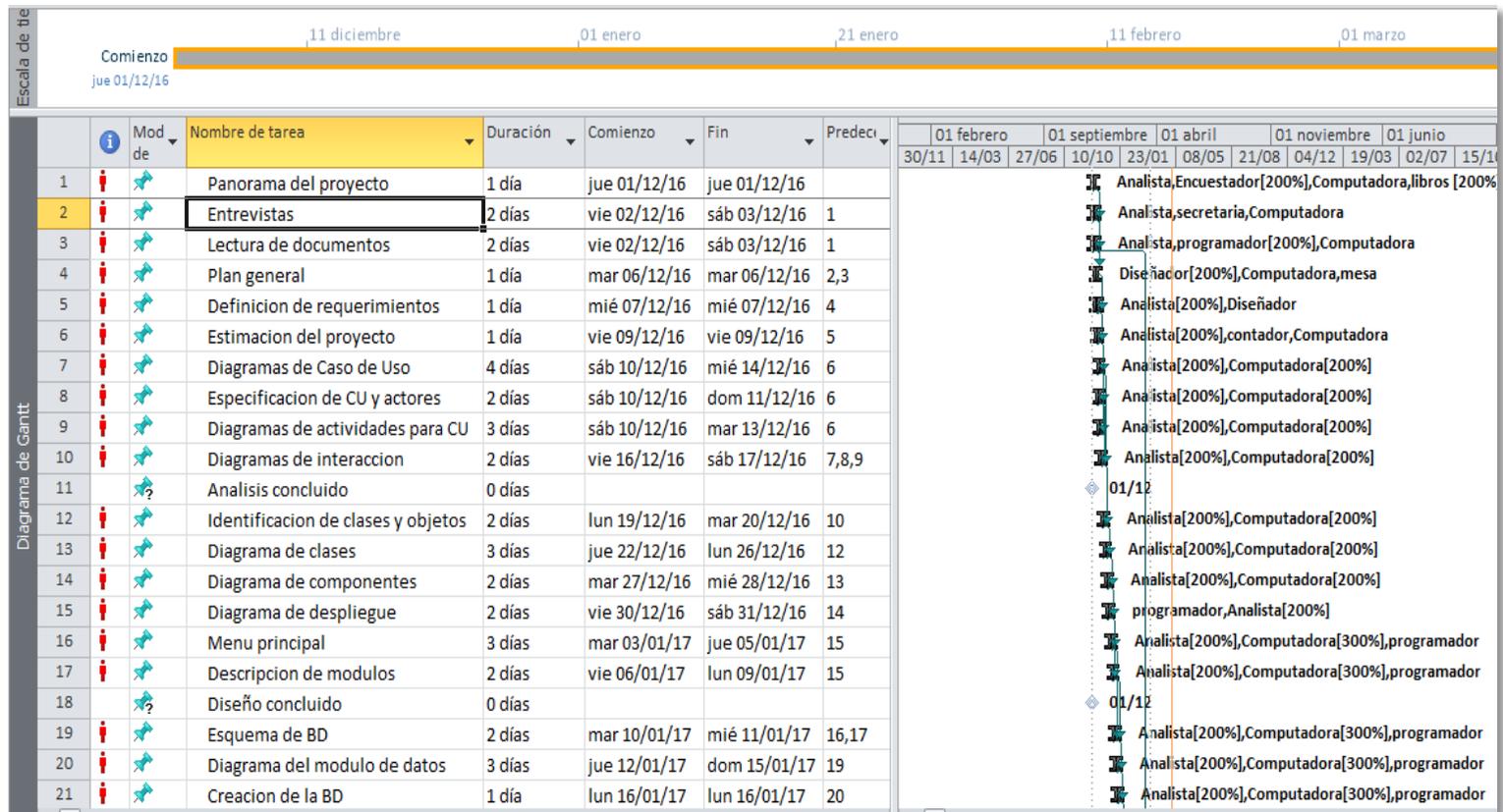


Figura 2. Línea de Tiempo del Proyecto Diagrama de Gantt
Fuente: Elaboración Propia

- ✓ **Plataforma de desarrollo.** - En esta parte describiremos el entorno de software común en el cual se desenvuelve la programación de nuestra aplicación.

Tabla 5
Herramientas Estándar

Tipo	Producto y Versión	Comentarios
Procesador de Textos	Microsoft Word 2010	Procesador de texto, empleado para centralizar todos los datos y documentación.
CASE modelado UML	Rational Rose 2000	Empleado para el boceto de todos nuestros diagramas.
Gestión de proyectos	Microsoft Project	Utilizado para la programación de actividades del proyecto.
Diseño IDE java	NetBeans IDE 8.2	Empleado para el boceto de interfaz de usuario.
Desarrollo de reportes	ireportes 5.0	Difusión de reportes impresos del sistema.
Servidor web wamservser 2.3	Servicio de base de datos: Mysql	Aplicación utilizada para asentar y estructurar un servidor web basado en software libre.
Modelado de base de datos	SQLyog	Empleado para el modelamiento de BD

Fuente: Elaboración Propia

- ✓ **Dimensionamiento de hardware y software.** - Se debe dimensionar tanto en la máquina servidor como en las máquinas clientes, buscando las características apropiadas.
- ✓ **Servidor.** - En esta parte se describe el hardware y el software de la máquina servidor.

Tabla 6
Especificación de Hardware Servidor de Aplicaciones

Tipo	Características	Comentario
Torre 	Modelo: PowerEdge T610	Ideal para Pymes.
	Núcleos: 3 núcleos	Los procesadores de 3 núcleos son los más adecuados para las aplicaciones de oficina.
	Tecnologías optimizadas para el uso de energía	Esquematizadas para disminuir el consumo de la energía é incrementar a su vez la capacidad de rendimiento.
	Memoria: 16 GB	Por el costo.
	Disco Duro: 1 TB	Por el costo.
	Capacidad de virtualización: si	Proporciona una óptima capacidad de máquinas virtuales por servidor.

Fuente: Elaboración Propia

Tabla 7*Especificación de Software Servidor de Aplicaciones*

Tipo	Producto y Versión	Comentarios
Sistema operativo	Windows 2008 server 64 bits	Desarrollado con una óptima productividad para las actividades empresariales. Centraliza los procesadores basados en las características del servidor.
Servidor web	Apache 2	Procesa una aplicación del lado del servidor con el cliente, brindando una solución en cualquier lenguaje o aplicación.
Motor de base de datos	Mysql 5.5	Contribuye a una eficiente dirección y optimización de la Base de datos.
Servidor FTP		Admite desplazar uno o más archivos con seguridad entre los ordenadores suministrando así, seguridad y orden de los archivos, así como el dominio de la transferencia.

Fuente: Elaboración Propia

- ✓ **En los Usuarios finales.** - En esta parte se describe el hardware y el software de las máquinas clientes.

Tabla 8
Especificación de Hardware Estaciones de Usuario Final

Tipo	Característica	Comentario
PC Desktop	Modelo compatible, Core 2 Duo, FSB's de 800, 1066 y 1333MHz	Accede a desarrollar modificaciones en el futuro.
	Lector óptico DVD	Necesario
	Microprocesador de 2 núcleos	Excelente para utilizar en las máquinas de clientes.
	Memoria de 4GB	Fundamental
	Disco de 350 GB	Fundamental
	Monitor LED 14 pulgadas	

Fuente: Elaboración Propia

Tabla 9
Especificación de Software Estaciones de Usuario Final

Tipo	Producto y versión	Comentarios
Sistema operativo	Windows Seven 64 bits	Accede a desarrollar modificaciones en el futuro.
Firewall	Windows,	Protege al equipo. Excelente para utilizar tanto en el cliente como en el servidor.
Antivirus	Kaspersky, Panda	Obligatorio para el cliente y el servidor.

Fuente: Elaboración Propia

3.3.3. Historia y Escenarios

Está compuesto por escenarios específicos del sistema expresado como casos de uso y su solución con el sistema de software, diagrama inicial E-R que involucra del escenario, así como los diagramas de secuencia, diagramas de clases, diagramas de actividad.

3.3.3.1. Escenario N° 1. Requerimiento de Matrícula

UML:

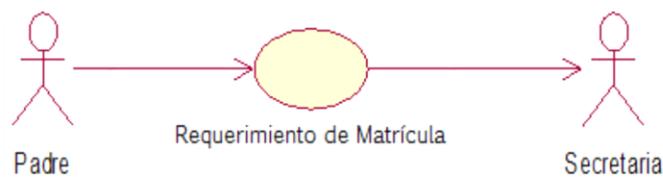


Figura 3. Caso de Uso - Requerimiento de Matrícula
Fuente: Elaboración Propia

Identificador	CU – Requerimiento de matrícula	
Versión	V 1.0	
Autor	Heber Chávez Choque	
Descripción	El sistema lleva el control de alumnos matriculados, transferidos, reasignados, etc. se realiza la búsqueda.	
Actores	Padre de familia, secretaria.	
Secuencia	1	Padre de familia presenta formato con datos del alumno.
		1.1 NO encontrado en sistema
		1.1.1 Secretaria realiza el registro en sistema.
		1.2 SI es encontrado, se envía observación por diversos motivos – reasignación – transferido – retorno – etc.
	1.2.1 Secretaria observa requerimiento	
	2	Observaciones entregadas al padre de familia, para ser subsanadas en el caso de haber estudiado años atrás en la institución.
Excepciones	1	Se registra al alumno observado, con autorización del director, previo compromiso del padre de familia en regularizar para que no pierda el año.
		1.1 Si el requerimiento fuese posterior a la fecha de inicio de labores.
Frecuencia	10 veces al mes	
Importancia	Importante	
Urgencia	Inmediata	
Comentarios	El requerimiento posterior al inicio escolar, solo se da por motivos de traslados.	

Figura 4. Especificación de Casos de Uso - Requerimiento de Matrícula
Fuente: Elaboración Propia

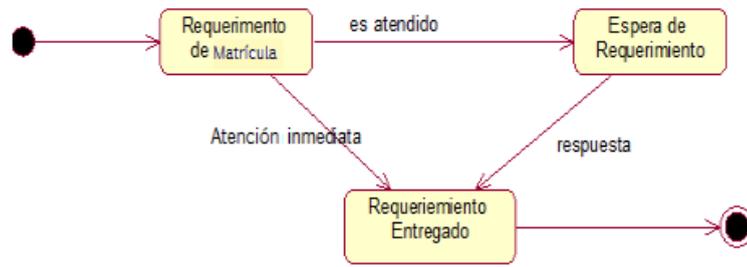


Figura 5. Diagrama de Estado - Requerimiento de Matrícula
Fuente: Elaboración Propia

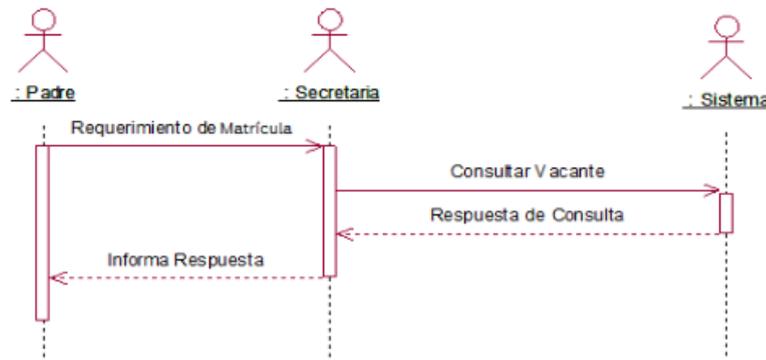


Figura 6. Diagramas de Secuencia - Requerimiento de Matrícula
Fuente: Elaboración Propia

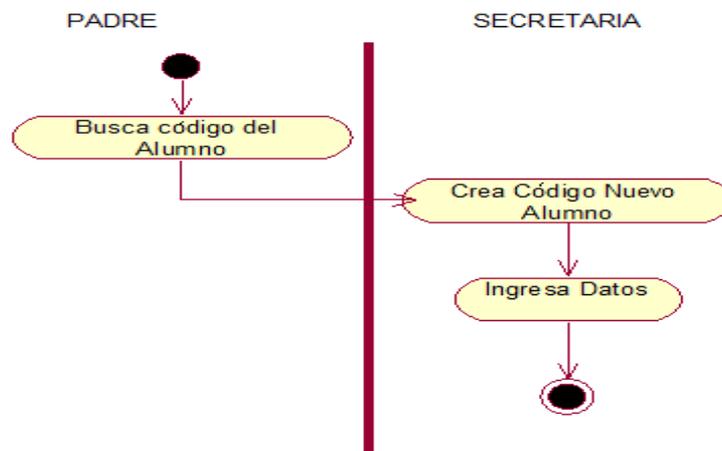


Figura 7. Diagrama de Actividades - Requerimiento de Matrícula
Fuente: Elaboración Propia

3.3.3.2. Escenario N° 2. Gestionar Matrícula

UML:

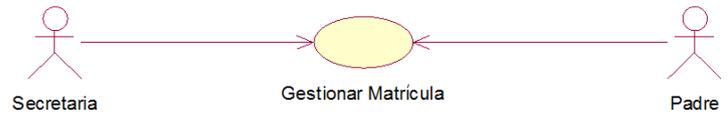


Figura 8. Caso de Uso - Gestionar Matrícula
Fuente: Elaboración Propia

Identificador	CU – Gestionar matrícula	
Versión	V 1.0	
Autor	Heber Chávez Choque	
Descripción	Se registra al alumno en el sistema sea nuevo, transferido y/o retorno a la institución.	
Actores	Padre de familia, Secretaria.	
Secuencia	1	Secretaria valida los documentos presentados por el padre de familia.
	1.1	Ordena los documentos anexos dándoles una prioridad
	2	Búsqueda de datos previos al ingreso del sistema.
	2.1	Actualizar datos si alumno es localizado en sistema.
	3	Crear código nuevo para asignar al alumno matriculado.
Excepciones	1	Cursos aprobados, su pre – requisitos.
Frecuencia	Constante	
Importancia	Importante	
Urgencia	Inmediata	
Comentarios	Este proceso es realizado 2 meses anteriores al inicio escolar.	

Figura 9. Especificación de Casos de Uso - Gestionar Matrícula
Fuente: Elaboración Propia

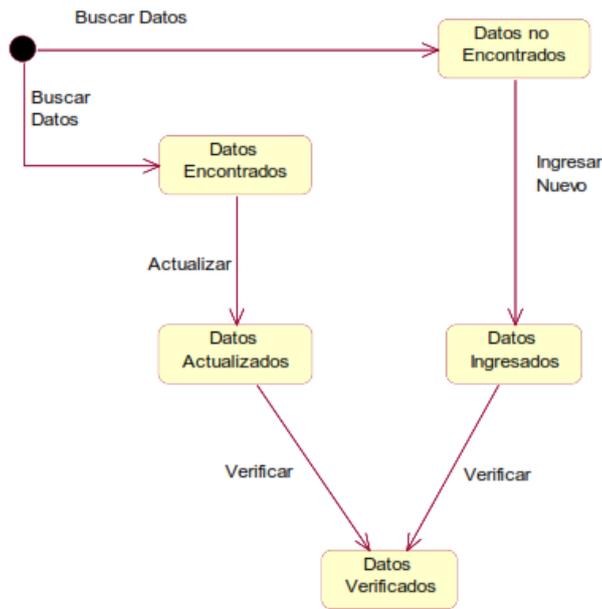


Figura 10. Diagrama de Estado - Gestionar Matrícula
Fuente: Elaboración Propia

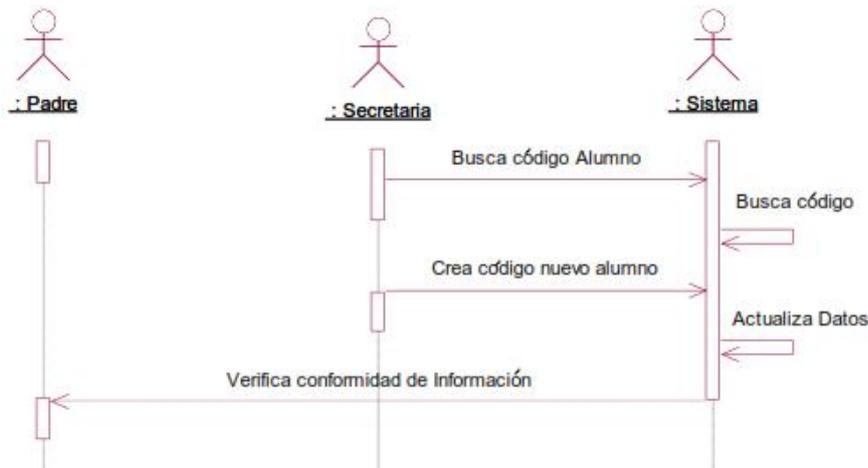


Figura 11. Diagramas de Secuencia - Gestionar Matrícula
Fuente: Elaboración Propia

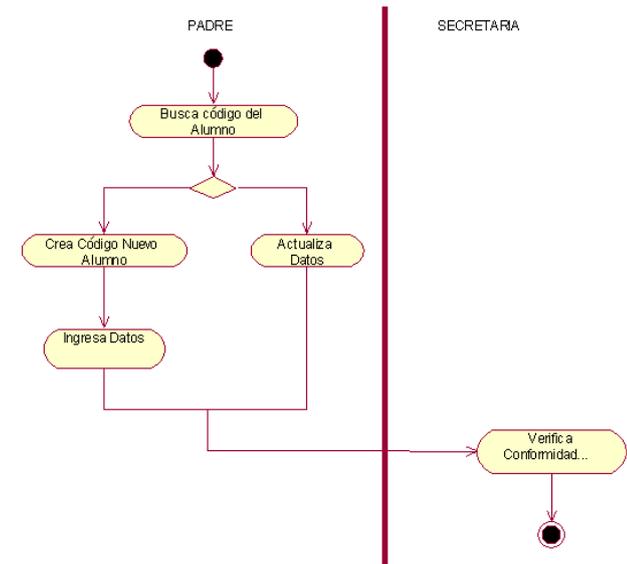


Figura 12. Diagrama de Actividades - Gestionar Matrícula
Fuente: Elaboración Propia

3.3.3.3. Escenario N° 3. Autorizar Matrículas

UML:

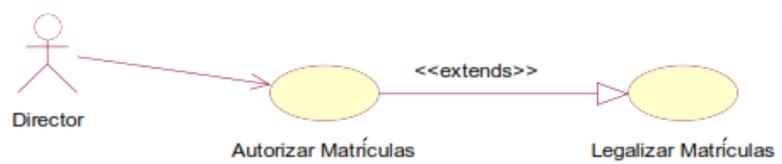


Figura 13. Caso de Uso - Autorizar Matrícula
Fuente: Elaboración Propia

Identificador	CU – Autorizar matrícula	
Versión	V 1.0	
Autor	Heber Chávez Choque	
Descripción	El director autoriza la matrícula y después legaliza la matricula con sello.	
Actores	Director, Sistema	
Secuencia	1	Recibe los formularios de actualización de información del estudiante ya diligenciados.
	1.1	Observa si encuentra errores.
	2	El director autoriza las matrículas para luego legalizar.
Excepciones	1	Acta de compromiso de documentos por regularizar.
Frecuencia	Constante.	
Importancia	Importante.	
Urgencia	Inmediata.	
Comentarios	Se verifican fotocopias de identificación y el historial de nivel académico para anexaras a una carpeta.	

Figura 14. Especificación de Casos de Uso - Autorizar Matrícula
Fuente: Elaboración Propia

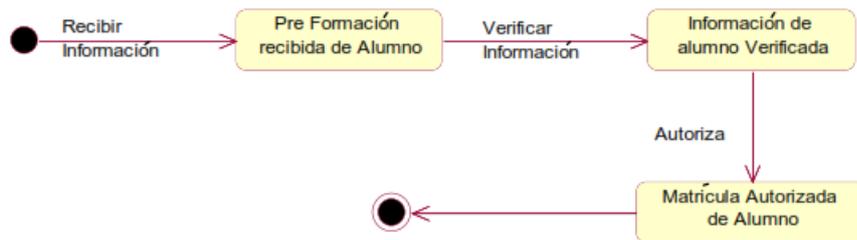


Figura 15. Diagrama de Estado - Autorizar Matrícula
Fuente: Elaboración Propia

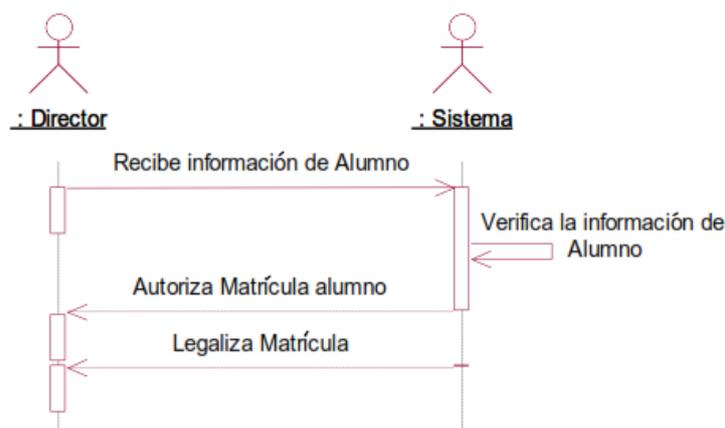


Figura 16. Diagrama de secuencia - Autorizar Matrícula
Fuente: Elaboración Propia



Figura 17. Diagramas de Actividad - Autorizar Matrícula
Fuente: Elaboración Propia

3.3.3.4. Escenario N° 4. Administra Información Académica

UML:



Figura 18: Caso de Uso Administrar - Información Académica
Fuente: Elaboración Propia

Identificador	CU – Administrar Información Académica	
Versión	V 1.0	
Autor	Heber Chávez Choque	
Descripción	El coordinador administra todo tipo de información académica que contempla las notas del estudiante, las observaciones y el historial de años cursados.	
Actores	Coordinador, Director, Profesor.	
Secuencia	1	La información académica también es una función que desempeña el director.
	1.1	Envía notificaciones para actualizar la información.
	2	Genera reportes de observaciones encontradas al profesor.
	3	Coordinador imprime consolidado de notas.
Excepciones	1	Cierra cada termino de año.
Frecuencia	Constante	
Importancia	Importante	
Urgencia	Inmediata	
Comentarios	No debe permitir actualizar el ingreso de notas una vez cerrado el periodo escolar.	

Figura 19. Especificación de Casos de Uso - Administrar Información Académica
Fuente: Elaboración Propia

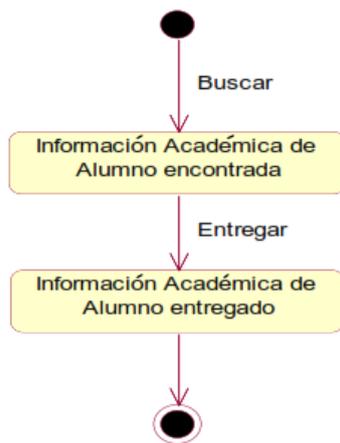


Figura 20. Diagrama de Estado - Administrar Información Académica
Fuente: Elaboración Propia

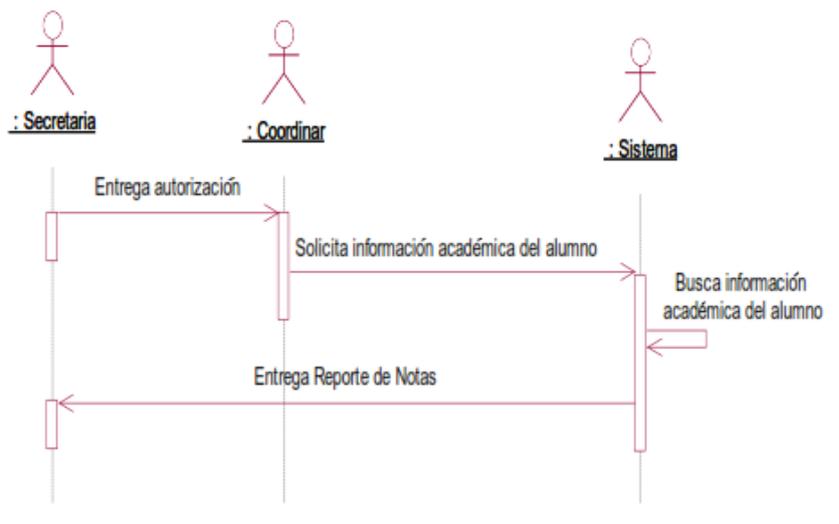


Figura 21. Diagramas de Secuencia - Administrar Información Académica
Fuente: Elaboración Propia

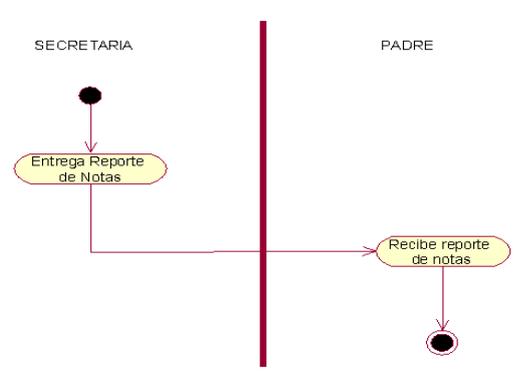


Figura 22. Diagrama de Actividades - Administrar Información Académica
Fuente: Elaboración Propia

3.3.3.5. Escenario N° 5. Entrega Reporte de Notas

UML:



Figura 23. Caso de Uso Entregar - Reporte de Notas.
Fuente: Elaboración Propia

Identificador	CU – Entrega de reporte de notas
Versión	V 1.0
Autor	Heber Chávez Choque
Descripción	El coordinador es quien se encarga de entregar los reportes de notas para que sean ingresados al sistema de registro.
Actores	Coordinador, Secretaria, Padre de familia.
Secuencia	1 Coordinador entrega los reportes de notas a la secretaria para que los registre.
	1.1 Secretaria valida información, observa.
	2 Secretaria ingresa notas al sistema académico.
	3 Secretaria imprime reporte de notas del alumno.
4 Secretaria entrega notas al padre de familia.	
Excepciones	1 Termino de cada bimestre.
Frecuencia	bimestral
Importancia	Importante
Urgencia	Inmediata
Comentarios	Los reportes deben estar soportado con copias de la información suministrada.

Figura 24. Especificación de Casos de Uso - Entregar Reporte de Notas
Fuente: Elaboración Propia

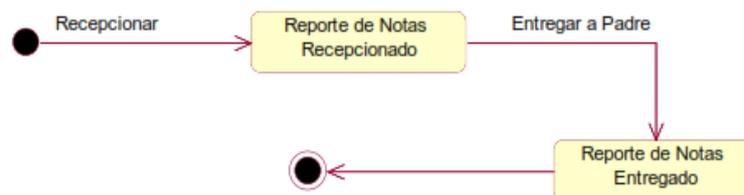


Figura 25. Diagrama de Estado - Entregar Reporte de Notas.
Fuente: Elaboración Propia

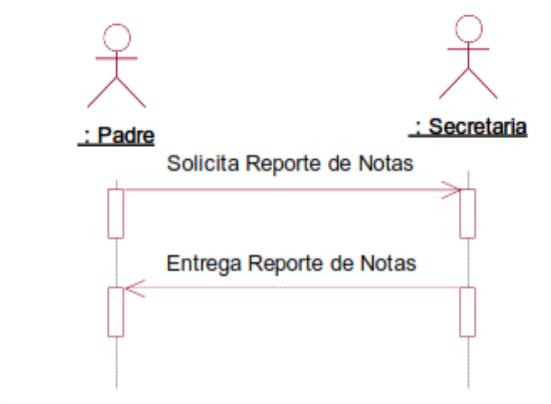


Figura 26. Diagramas de Secuencia - Entregar Reporte de Notas.
Fuente: Elaboración Propia

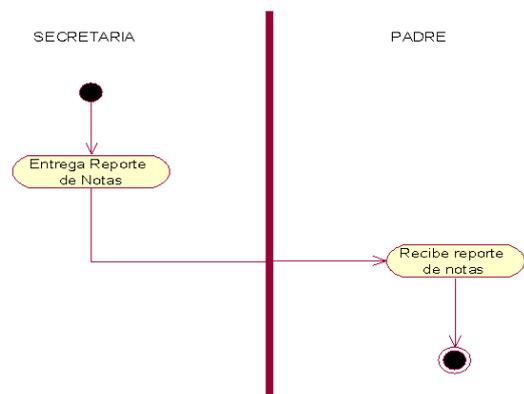


Figura 27. Diagrama de Actividades - Entregar Reporte de Notas.
Fuente: Elaboración Propia

3.3.3.6. Escenario N° 6. Registra Matrícula

UML:

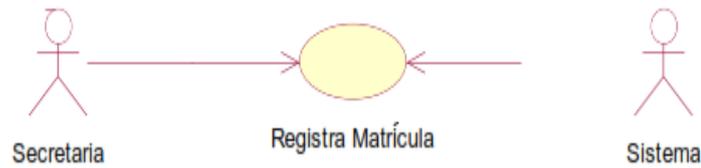


Figura 28. Caso de Uso - Registra Matrícula
Fuente: Elaboración Propia

Identificador	CU – Registrar Matrícula	
Versión	V 1.0	
Autor	Heber Chávez Choque	
Descripción	Secretaria registra matrícula del alumno en sistema.	
Actores	Secretaria, Sistema.	
Secuencia	1	La secretaria realiza el análisis de lo recopilado a través de las herramientas de Word y Excel, consolidando resumen.
	1.1	Actualiza datos del alumno periódicamente en el sistema.
	2	Los datos se almacenan en la base de datos del sistema.
Excepciones	1	Culminación del resumen de matrícula para ingresar al sistema.
Frecuencia	Constante	
Importancia	Importante	
Urgencia	Inmediata	
Comentarios	Mayor fluidez al inicio del año escolar.	

Figura 29. Especificación de Casos de Uso - Registra Matrícula
Fuente: Elaboración Propia



Figura 30. Diagrama de Estados - Registra Matrícula
Fuente: Elaboración Propia

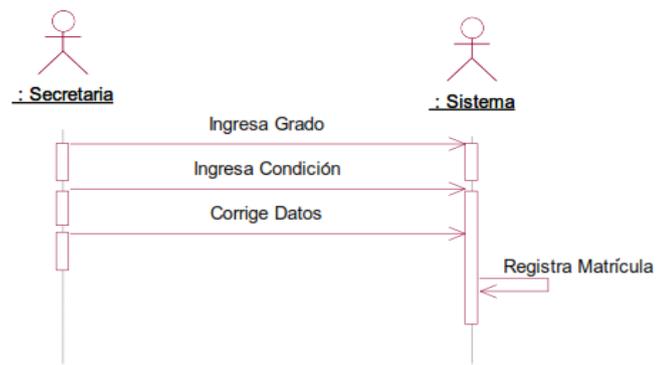


Figura 31. Diagramas de Secuencia - Registra Matrícula
Fuente: Elaboración Propia

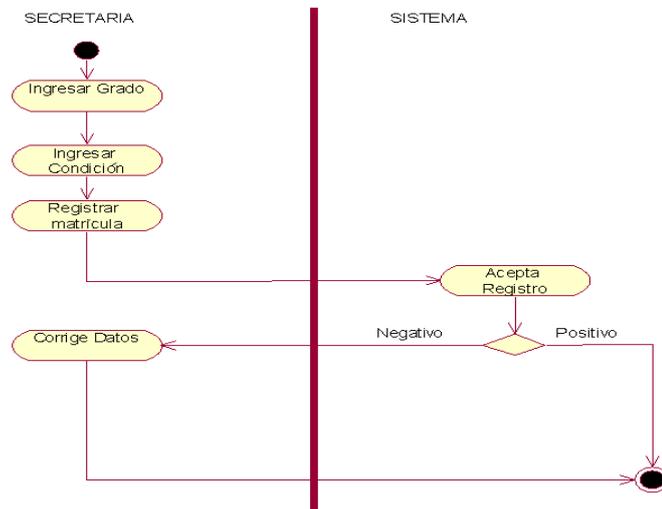


Figura 32. Diagrama de Actividades - Registra Matrícula
Fuente: Elaboración Propia

3.3.3.7. Escenario N° 7. Entregar notas

UML:

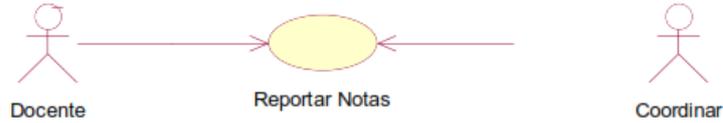


Figura 33. Caso de Uso Entregar Notas
Fuente: Elaboración Propia

Identificador	CU – Reportar Notas
Versión	V 1.0
Autor	Heber Chávez Choque
Descripción	El docente lleva el control de notas en planillas y se las entrega al estudiante.
Actores	Docente, Coordinador.
Secuencia	1 Esta entrega de notas que realiza el docente hacia los estudiantes es de forma informativa.
	2 Docente entrega las planillas de notas al coordinador.
	3 Coordinador, valida información para ser entregada a secretaria.
Excepciones	1 Planilla de notas incompleta no es ingresada, observaciones de alumnos con inasistencias.
Frecuencia	Constante.
Importancia	Importante
Urgencia	Inmediata
Comentarios	Mayor fluidez al inicio del año escolar.

Figura 34. Especificación de Casos de Uso - Reportar Notas
Fuente: Elaboración Propia



Figura 35. Diagrama de Estado Entregar Notas
Fuente: Elaboración Propia

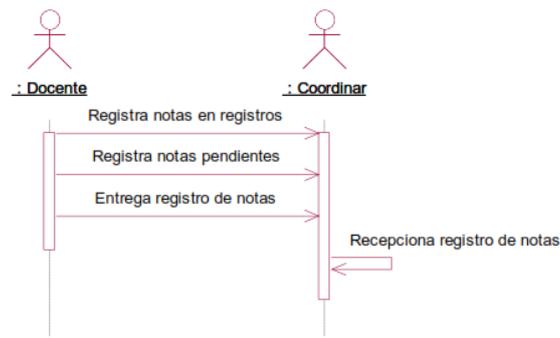


Figura 36. Diagramas de Secuencia Entregar Notas
Fuente: Elaboración Propia

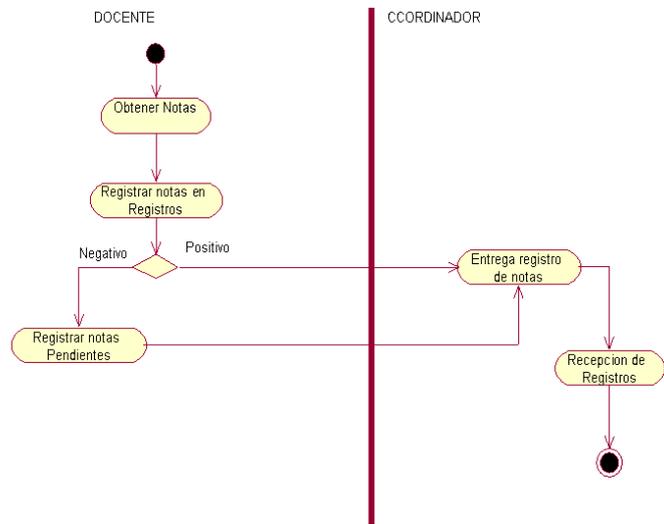


Figura 37: Diagrama de Actividades - Entregar Notas
Fuente: Elaboración Propia

3.3.3.8. Escenario N° 8. Administra Información Académica

UML:



Figura 38. Caso de Uso Administrar - Información Académica
Fuente: Elaboración Propia

Identificador	CU – Administrar Información Académica
Versión	V 1.0
Autor	Heber Chávez Choque
Descripción	El coordinador administra todo tipo de información académica en donde considera las notas del estudiante, las observaciones y el historial de años cursados.
Actores	Coordinador.
Secuencia	1 La información académica también es una función que desempeña el director.
	2 Coordinador, registra datos del docente.
	3 Coordinador, asigna cursos al docente y horarios.
	4 Seguimiento de notas registradas al sistema académico.
Excepciones	1 Ninguna.
Frecuencia	Constante.
Importancia	Importante
Urgencia	Inmediata
Comentarios	Mayor fluidez al inicio del año escolar.

Figura 39. Especificación de Casos de Uso - Administrar Información Académica
Fuente: Elaboración Propia



Figura 40. Diagrama de Estado - Administrar Información Académica
Fuente: Elaboración Propia

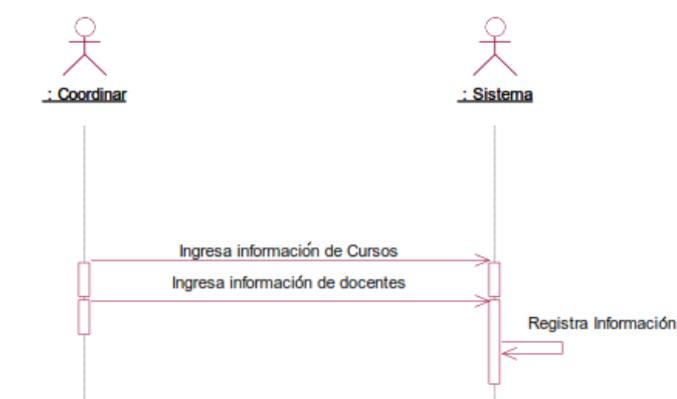


Figura 41. Diagramas de Secuencia - Administrar Información Académica
Fuente: Elaboración Propia

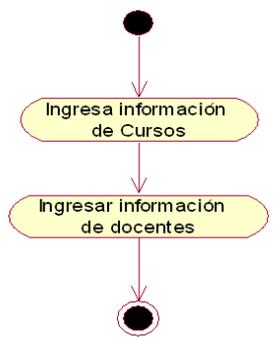


Figura 42. Diagrama de Actividades - Administrar Información Académica
Fuente: Elaboración Propia

3.3.3.9. Escenario N° 9. Ingresar Notas

UML:

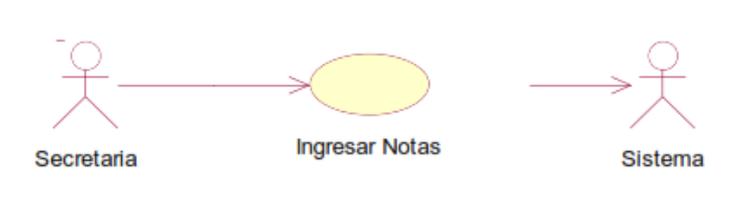


Figura 43. Caso de Uso - Ingresar Notas
Fuente: Elaboración Propia

Identificador	CU – Ingresar Notas
Versión	V 1.0
Autor	Heber Chávez Choque
Descripción	La secretaria ingresa las notas según código del alumno o actualiza notas según la información personal.
Actores	Secretaria, Sistema.
Secuencia	1 Este formato es entregado al coordinador.
	2 Secretaria ingresa la planilla de notas al sistema.
	3 Actualiza notas pendientes, posterior a la entrega.
	4 Registro almacenado en base de datos del sistema.
Excepciones	1 Notas pendientes por regularizar.
Frecuencia	Contante.
Importancia	Importante.
Urgencia	Inmediata.
Comentarios	Mayor fluidez al inicio del año escolar.

Figura 44. Especificación de Casos de Uso - Ingresar Notas

Fuente: Elaboración Propia



Figura 45. Diagrama de Estado - Ingresar Notas

Fuente: Elaboración Propia

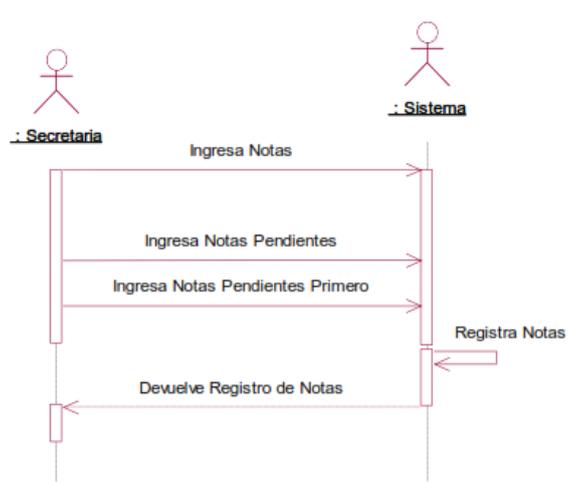


Figura 46. Diagramas de Secuencia - Ingresar Notas

Fuente: Elaboración Propia

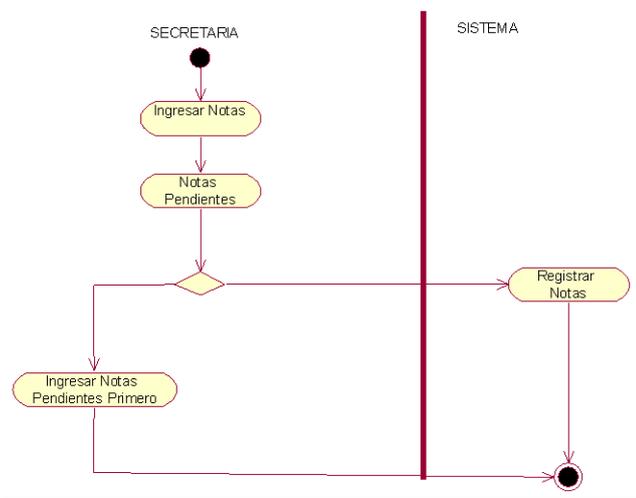


Figura 47. Diagramas de Actividad - Ingresar Notas
Fuente: Elaboración Propia.

Diagramas Caso de Uso - General Registro de Matrícula

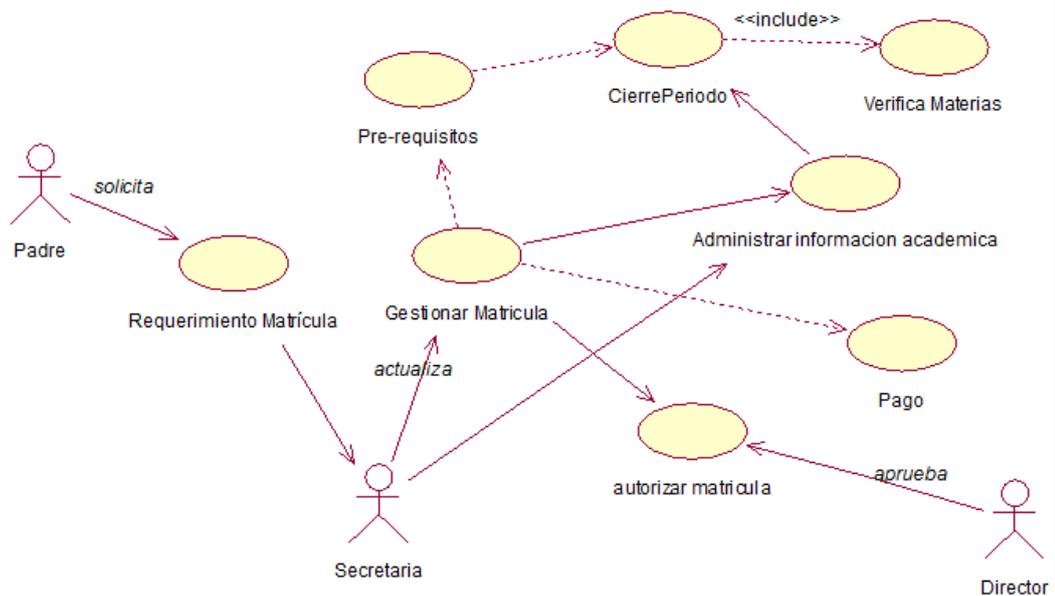


Figura 48. Diagrama de Casos de Uso - Proceso de Registro de Matrícula
Fuente: Elaboración Propia

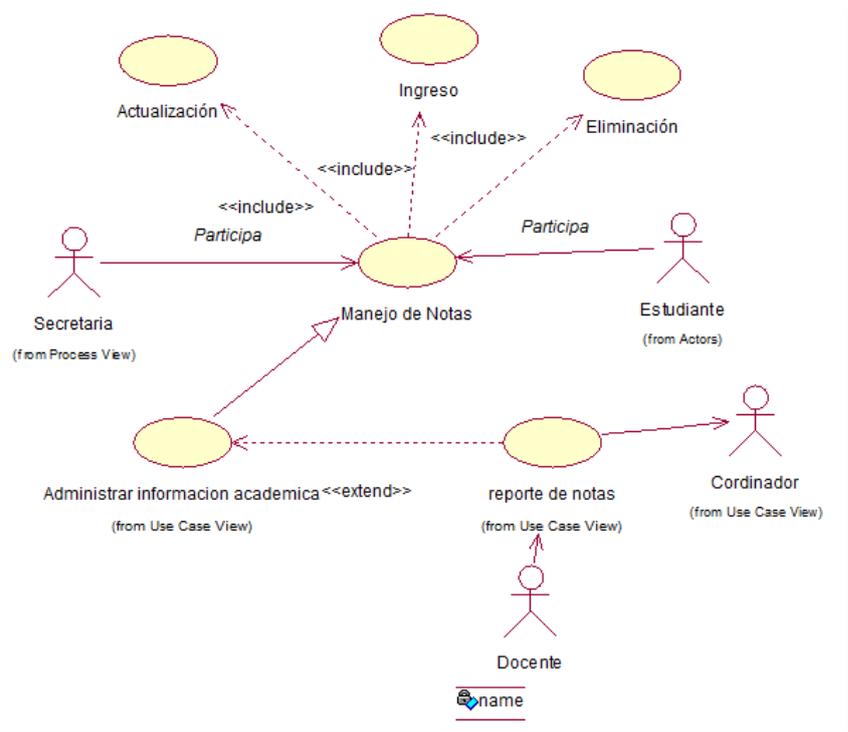


Figura 49. Diagrama de Casos de Uso - Proceso de Control de Notas
Fuente: Elaboración Propia

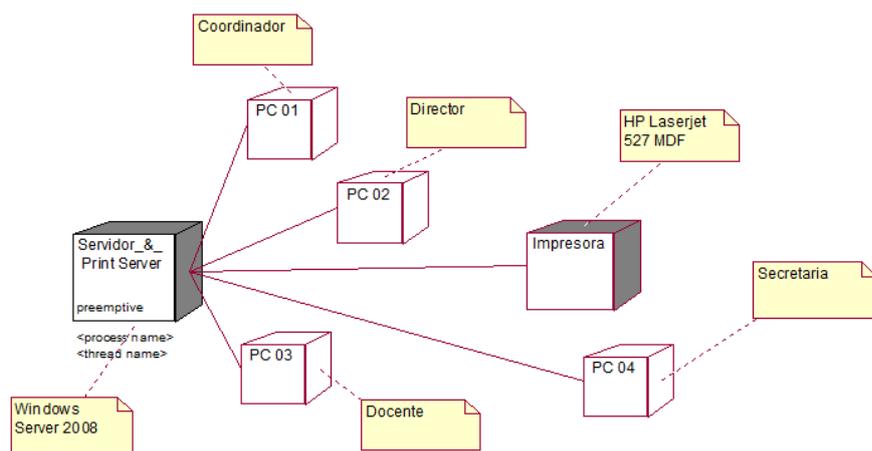


Figura 50. Diagrama de Despliegue - Sistema de Matrícula y Notas
Fuente: Elaboración Propia

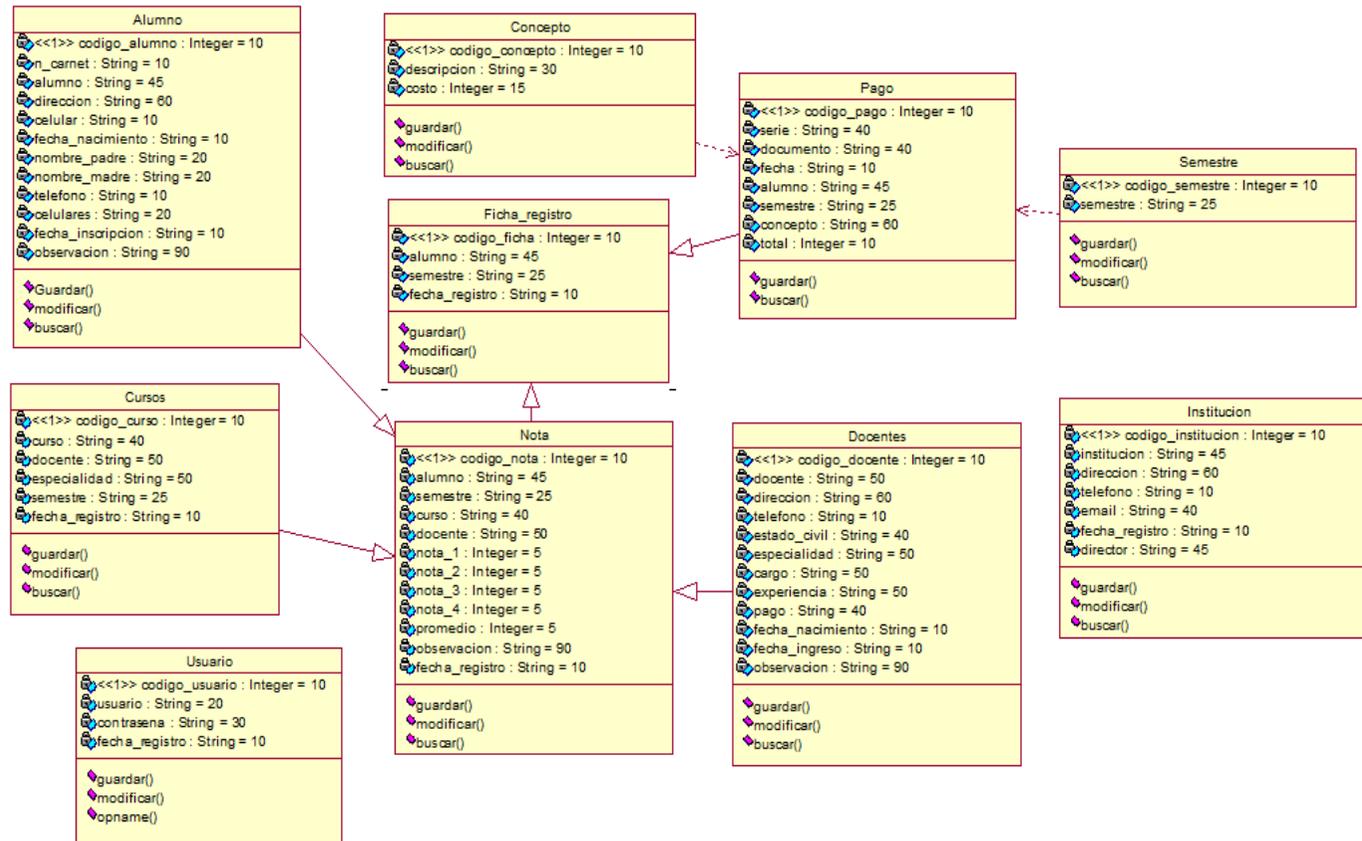


Figura 51. Diagrama de Clases - Sistema de Matrícula y Notas
Fuente: Elaboración Propia

3.4. Marco Teórico

3.4.1. Tecnología Informática IT

3.4.1.1. Cliente/Servidor

El programa de mayor relevancia es aquel que genera los mayores y mejores beneficios (costo/desempeño) a nivel de los servicios. El entorno de cliente/servidor contribuye a la intervención de diversas plataformas que compiten entre sí, tal como los métodos manejadores de bases de datos (SQL), y diversos instrumentos para el progreso de la interfaz gráfica, que representa aquel componente visual que elucida la información que utilizará el consumidor.

En la plataforma convencional cliente/servidor, el usuario accede al banco de información desde un equipo servidor ó host. En este prototipo, la interfaz del consumidor y el razonamiento de la organización se acoplan en el equipo hardware del cliente, en donde este accede a la regla que permite el funcionamiento del procesador. Dicha señal está ubicada en aquellos componentes que permiten la evolución de la interfaz de programación gráfica (GUI) así como los procedimientos necesarios para la sostenibilidad de la organización.

El componente del dispositivo de un sistema que resuelve las solicitudes de otros elementos, llamados clientes, presenta la ventaja de encontrar y devolver la información en una configuración accesible para los usuarios, por ende, los datos presentados de manera simplificada serán de gran contribución para el demandante final.

Los reglamentos é interfaces de la intercomunicación entre ordenadores no sólo permite el intercambio de datos, sino también compartir recursos de todo tipo, optimizando así elevadas inversiones que son administradas por el cliente para el mejor mensaje con el servidor. Los servidores pueden ejecutar en menor ó mayor medida, labores simples ó significativamente engorrosas. Ahora, sí el

equipo tiene como principal objetivo sostener un determinado software servidor, entonces este término; “servidor”, también será aplicado en la máquina. Por tal razón, antes de brindar respuesta a cualquier interacción, el servidor ya inicio su proceso de transferencia que se justifica y evidencia en el alto índice de soluciones.⁴

Después de emplear un servidor un número finito de veces, cualquier software que origina una demanda y espera respuesta, se denomina “cliente”. El servidor está atento al reclamo de cualquier flujo de datos en clausulado en determinada función. Mientras, que un cliente para poder notificar, reserva al flujo de datos arbitrario y no usado.

3.4.1.2. Arquitectura

La arquitectura está compuesta por dos niveles denominados; “cliente y servidor”. Además, en la construcción de la aplicación cliente/servidor, existe terminología referida al componente donde se ubica el razonamiento de la organización, siendo está el cliente ó el servidor.

Sí el razonamiento de la organización está ubicado en el lado del cliente, será llamado cliente grande pero sí está, al lado del servidor entonces se llamará servidor amplio.

La arquitectura principalmente está basada en el dispositivo que hace posible la comunicación, procesamiento y manejo de la información de manera relevante en el lado del cliente. Por ello, se tiene presente probabilidades como:

- a) Entrada de diferentes clientes al servidor.

⁴ Dordoigne J. (2015). Redes informáticas: Nociones fundamentales. Barcelona: Ediciones ENI.

- b) A mayor ancho de banda de la red, entonces mayor será la capacidad por donde se desplazará la información.
- c) Obstrucción sobre la información que está registrada.

Las soluciones a estas probabilidades son diversas y no está parametrado, por ello puede adquirir matices intermedios, más no extremos.

El conjunto de especificaciones que integran los procesadores muestra un tipo de presentación accesible al usuario para su mejor uso.

El tipo de interfaz de usuario y/o tipo de presentación, está compuesta por varias piezas oculares, mientras que las reglas de la aplicación vigilan la interacción con estas piezas y se desplaza en el equipo del usuario final (computadora cliente).

El tipo de presentación depende básicamente del tipo de tarea que realizará el usuario, así como de la respuesta informática fundamentada en el requerimiento organizacional. Este tipo, impacta a la asignación del software que será asentado y estructurado en el equipo del cliente, así como la demanda del mismo.

Las capas de presentación efectivas son:

- a) **De plataforma específica:** Es un sistema de interacción entre el ordenador y el usuario, que ha sido nombrada por el sistema operativo de la máquina cliente.

En donde, el progreso depende exclusivamente del sistema operativo soportado, así como de las piezas que lo conforman y se puede emplear

componentes como ActiveX⁵ Applets o Plugins para desplazar GUI de otro programa.

Uno de los beneficios de esta GUI es que la manera de interactuar con ella es dinámica, porque se emplean registros ya conocidos por los usuarios.

b) De plataforma independiente la implementación de una GUI de plataforma independiente se desarrolla en lenguaje Java. Por ello, el mismo modulo puede ser instalado en diferentes sistemas operativos.

1. JDBC: esta delimita un patrón de acceso a la base de datos y sus tablas. Además, es empleada en aplicaciones Java.
2. OLE DB: este es un elemento del software Microsoft's Data Access Components (MDAC). Interfaz nativa: la productividad en esta, es la más sobresaliente vista que suministra entrada a las bases de datos y comprende las propiedades de una mejor manera que la ODBC.

Además, si la DBMS no tiene apoyo para la ODBC, entonces puede emplear la OLE DB.

Los conceptos cliente y servidor, asumen que cada pieza se realiza en distintos ordenadores fusionados a través de una red. Sin embargo, se podría dar el suceso que tanto cliente como servidor se desplacen en el mismo equipo. Asimismo, hay que tener presente que un programa puede actuar como servidor, cliente ó los dos a la vez.

3.4.1.3. Características

Este sistema muestra las siguientes propiedades:

⁵ Componente de Software Estándar Microsoft html

- a) Servicio: explica la relación entre los desarrollos que fluyen en diferentes equipos. El procedimiento es:
SERVIDOR. - atiende las solicitudes de los clientes, brindándole solución acorde.
CLIENTE. - es usuario de los beneficios otorgados por el servidor. Por ende, fundamentado en la idea del servicio se genera una división práctica.
- b) Recursos Compartidos: el servidor de manera simultánea; conforme a sus utilidades, brindará respuestas a diferentes clientes a la vez, por ello sobrelleva diversos conectores, en donde actúa como el control de acceso a los recursos demandados por estos.
- c) Transparencia de Ubicación: la ubicación del servidor debe ser de acceso amigable para el cliente.
- d) El cliente y servidor: la comunicación entre ellos está basado en mensajes. Sí se modifica la arquitectura del cliente/servidor a una arquitectura escalable, será indiferente para el cliente. Aquí existen dos tipos de cambios, siendo ellos; crecimiento horizontal y vertical.

El sistema cliente/servidor, debe estar instalado en una red segura y veloz. Actualmente, los negocios son conscientes de la importancia de la red corporativa para la mantener conectadas y comunicadas las diversas áreas de la organización. Desde las oficinas principales de las gerencias corporativas hasta las más pequeña del organigrama organizacional.

3.4.1.4.Desventajas de la Arquitectura Cliente/Servidor

- ✓ Sí la red es compleja, entonces requerirá un dispositivo extraordinario.
- ✓ Uso de protocolos diferenciados.
- ✓ Genera gastos extraordinarios porque se deberá contratar un administrador, sólo sí la aplicación es grande.

3.4.2. Aplicaciones Distribuidas

3.4.2.1. Concepto de Computación Distribuida

Este posee diferentes acepciones. Tales como las señaladas por Microsoft:

- Para los desarrolladores: está herramienta permite operar en múltiples equipos (computadoras) a la vez.
- Para los administradores de Sistema de Tecnología: Las aplicaciones creadas son adaptables para sostener el entorno de cambio del negocio.
- Para Usuarios finales: es amigable y permite acceder a una variedad de información sin requerir complejos procesos, códigos ni diversas claves.

La computación distribuida representa una ventaja en la arquitectura cliente/servidor porque este último; para evitar la centralización, en tiempo real suministra la información y servicios correspondientes entre los usuarios de diversos lugares geográficos y conectados por una misma red. (WAN)

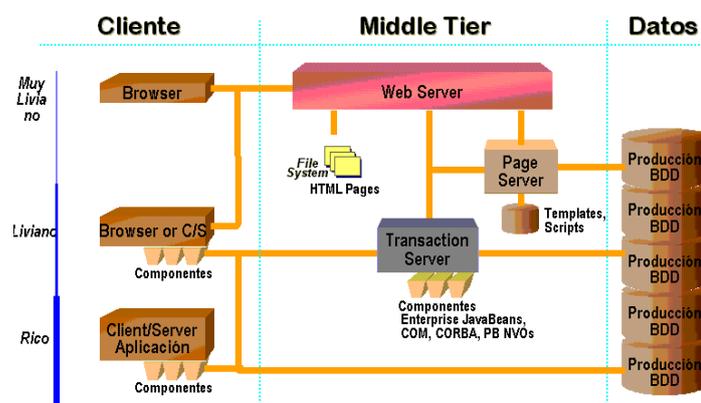


Figura 52. Arquitectura Distribuida
Fuente: Cisneros, 1998

Descripción

En donde, predomina el Cliente, Servidor Intermedio y Fuente de Datos

En esta clase de arquitectura las aplicaciones pueden ser como un navegador ó navegador web, porque se conecta a un servidor web con la finalidad de ser empleada desde otro tipo de herramientas, tales como el cliente/servidor.

La finalidad de la computación distribuida; que es parte de la Middleware, es que la organización independientemente de su ubicación geográfica y, componentes, acceda a sus recursos y servicios en tiempo real.

3.4.2.2. Objetivos

La computación distribuida busca lo siguiente:

- a) Erigir entornos que permita al sistema comunicacional acoplar diversas piezas computacionales para solidificar la plataforma organizacional.
- b) Enlazar todas las demandas y redes existentes en las diversas áreas y estaciones de trabajo de la organización.
- c) Los sistemas informáticos y de minicomputadoras, deben permitir de manera amigable el acceso a los usuarios.
- d) Asegurar la integridad y confiabilidad de los datos a los usuarios ubicados en diferentes zonas geográficas.
- e) Admitir en la organización las aplicaciones y tecnologías cliente/servidor

3.4.3. Replicación de Entornos Distribuidos

Esto considera definir la ubicación de los programas y los datos que formarán parte de la base de información, así como de la estructura de la red de ordenadores.

No se considera una dificultad; la ubicación de programas, sí se tiene la copia de los mismos en cada equipo de la red. Ahora, sí a todos ellos desea tener en un solo equipo que responderá a todas las demandas de las estaciones de trabajo de la organización, entonces se deberá distribuir las relaciones y las tablas por toda la red.⁶

Ahora, si se considera que los programas estén ubicados en un solo equipo, entonces se deberá absolver las siguientes consultas: ¿Cuáles son los criterios

⁶ López J. (2001). Introducción a los sistemas de bases de datos. México: Pearson Educación.

para distribuir las relaciones y las tablas por toda la red? ¿Este enfoque genera el mayor rendimiento? ¿Existe otra mejor alternativa?

Estas interrogantes serán resueltas en los textos posteriores. La compartición presenta las siguientes alternativas; Inexistencia porque cada aplicación y su información distribuyen sólo datos y no programas, por ende, se genera una respuesta de las aplicaciones en cada equipo y los datos se desplazarán por la red. Es así como, se puede conceder a los datos ubicados en una tercera posición. Tal como se elucido líneas anteriores, en este ítem se elegirá el punto intermedio de compartición.⁷

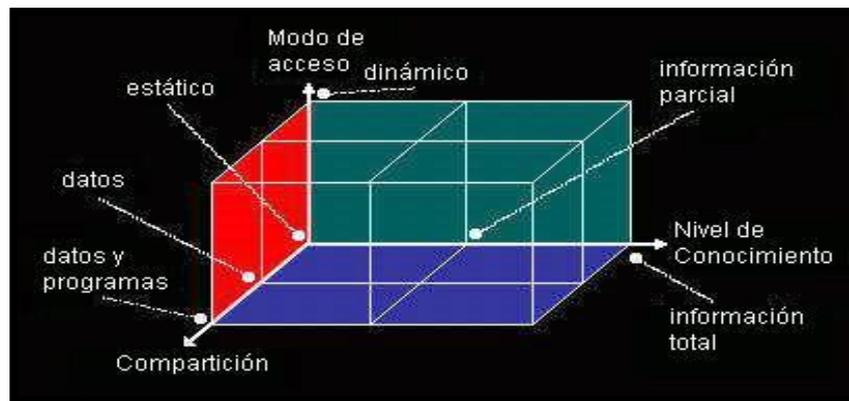


Figura 53. Enfoque de la Distribución
Fuente: Coronel, 2011.

Hay dos alternativas referidas a las características de acceso de datos. Siendo estas las siguientes:

La forma de ingresar a la información solicitada por los usuarios podría ser fija o activa⁸. Es raro, que los sistemas distribuidos puedan catalogarse como fijos. Es así, que su relevancia en el tiempo reside en su dinamismo y el número de cambios. Por ello, está dimensión define la vinculación existente

⁷ Colobran M., Arqués J., & Galindo, E. (2008). Administración de Sistemas Operativos en Red. España: Editorial UOC.

⁸ VV.AA. (2002). Diccionario de Internet. Madrid: Editorial Complutense.

entre el diseño de bases de datos distribuidas y el procesamiento de consultas⁹.

La clasificación tercera es la etapa del entendimiento de las propiedades de una entrada. En esta probabilidad teórica, se puede presentar que los diseñadores no cuenten con la información suficiente sobre como los usuarios ingresan a la base de datos. Dicha eventualidad dificultaría la labor de los diseñadores al carecer de tal testimonio.¹⁰

Por ello, se considera necesario enterarse al máximo detalle la manera de ingreso de los usuarios a la base de datos.

El diseño de base de datos distribuidas tiende a generar los inconvenientes anteriormente expuestos, que no se presentarán si está presente la centralización¹¹.

Es recomendable utilizar la táctica ascendente a partir de un diseño basado en un número de pequeñas bases de datos existentes con la finalidad de consolidarlas en una sola. Por ello, se empezaría de los esquemas conceptuales locales hasta lograr el esquema conceptual global¹².

Para procesar la información requerida por los potenciales usuarios del banco de datos, se deben establecer las condiciones del sistema, así como los objetivos a cumplir considerando los aspectos económicos, seguridad, disponibilidad, flexibilidad, así como los niveles de rendimiento. En donde,

⁹ Cardador L. (2015). Desarrollo de aplicaciones web distribuidas. España: IC Editorial.

¹⁰ Cisneros J. (1998). Panorama sobre base de datos. Un enfoque práctico. México: UABC.

¹¹ Verástegui L. (2001). Introducción a los sistemas de bases de datos. México: Pearson Educación.

¹² Pons, O., Acid, S., Marín, N., Medina, J., & Vila A. (2008). Ventajas de la distribución de datos. En Introducción a los sistemas de base de datos (p. 243). España: Editorial Paraninfo.

los resultados obtenidos sirven de inicio para dos actividades que se desarrollan paralelamente.

La replicación de datos considera más que una copia simple de información, porque su importancia radica en que facilita información actualizada en tiempo real a los usuarios del negocio. Se considera que la replicación de datos provee de amplios beneficios al saber.

3.4.3.1. Inicios de la Replicación de Datos

El origen de la replicación de datos se da a razón que muchas veces algunas unidades del negocio han tomado decisiones fundamentadas en información desactualizada por horas, días ó semanas.

A causa, de que el proceso ha sido casi siempre manual y no automático, el resultado de la distribución copiar y bajar datos es fallido, con información desactualizada y no oportuna que afectó significativamente la toma de decisiones de la organización.

3.4.4. ¿Cómo Saber si se Necesita Replicación?

El administrador de la base de datos es el responsable para diagnosticar en que momento es necesario la replicación como una solución para los requerimientos de negocios y tópicos de tecnología¹³.

Para implementar la replicación, se debe considerar las siguientes directrices:

- ✓ Sí requiere una vista corporativa consolidada de operaciones distribuidas que se desplazan en diversas DBMS.
- ✓ Sí desea disminuir el tráfico de las redes centralizadas.

¹³ Aguilar P. (2011). Políticas de almacenamiento y resguardo de la seguridad (Seguridad Informática). México: Editorial EDITEX.

- ✓ Sí necesita distribuir datos vi direccionales entre un mainframe, cliente/servidor y sistemas desktops con el objetivo de garantizar la entrega confiable de datos a todas las ubicaciones de la organización.
- ✓ Sí está migrando a sistemas mainframes y necesita proveer de espacios determinados para trasladar datos mientras se mantiene a su vez los sistemas sincronizados.
- ✓ Sí necesita proveer de datos a toda la organización en tiempo real.
- ✓ Sí el negocio en tiempo real ejecuta operaciones globales en que no es factible tener todos los datos en un sitio de acceso amigable.
- ✓ Sí requiere mantener el enlace corporativo y brindar control de acceso a la unidad de negocios sobre determinada información, mientras se aíslan las fallas en cualquier parte de la red o en el mainframe.
- ✓ Sí con la finalidad de sincronizar el movimiento de los datos entre sistemas, empleará la técnica datawarehouse o datamart.

Sí bien es cierto, la replicación puede solucionar muchos problemas del proceso distribuido, también es imprescindible reconocer la existencia de dos casos en que la replicación no es recomendable¹⁴, tales como:

- ✓ Existe un significativo número de inscripciones reajustables en múltiples replicas porque tienden a generar conflicto, y por ende se distrae la concentración en solucionar los mismos.
- ✓ La estabilidad de los datos es crítica. - principalmente cuando existen datos vulnerables al cambio continuo, por tal razón será difícil garantizar la estabilidad de los mismos.

¹⁴ Llanos R. (2010). Fundamentos de informática y Programación en C. España: Editorial Paraninfo.

3.4.5. Consideraciones Generales para Replicación

La replicación de datos en una empresa representa más allá de plagiar y trasladar datos¹⁵. Por ello, un sistema de replicación debe tener presente las siguientes instrucciones:

- ✓ **Existencia permanente de datos.** - Los sistemas de replicación deben asegurar confiabilidad al usuario y no poner en riesgo las actividades del negocio a errores del sistema.
- ✓ **Reparto o entrega de Información consistente.** - Los sistemas de replicación deben brindar protección fidedigna a los datos.
- ✓ **Administración fácilmente centralizada.** – los administradores deben tener la capacidad de gestionar oportunamente desde su punto de trabajo a los sistemas de replicación.
- ✓ **Acceso a fuentes de datos heterogéneos.** - tener la capacidad de trasladar los datos a través de diferentes redes de información y usuario.
- ✓ **Autonomía local.** – cada punto que admite datos replicados debe tener la alternativa de deliberar que conjuntos de datos son admitidos y la forma en que estos serán cambiados.

La fase del diseño es la etapa más relevante para definir la creación de un ambiente distribuido. Es de suma importancia que el administrador de la base de datos establezca los datos que ordenará replicar, así como las veces en que se realizará las modificaciones entre las replications, así como brindar solución en las diversas actualizaciones.

3.4.6. Arquitecturas Comunes de Replicación

Los ambientes de replicación son tipificados en base a sus posibles arquitecturas¹⁶. El método de replicación, los clientes se refieren a menudo de

¹⁵ Gallegos J. (2010). Mantenimiento de Sistemas Microinformáticos: Técnicas Básicas. México: EDITEX.

¹⁶ Dordoigne J. (2015). Redes informáticas: Nociones fundamentales. Barcelona: Ediciones ENI.

este término para que pueda ser empelado. Las arquitecturas comunes de replicación:

Distribución. – Se le llama también “*Diseminación de datos*”, quiere decir que los datos son actualizados en una localización central para que después sean replicados a sitios remotos pero solos de lectura.

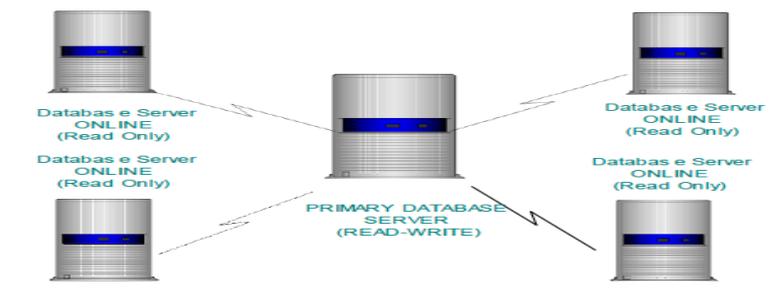


Figura 54. Distribución
Fuente: Verástegui, 2001.

Consolidación: se lleva a cabo en un ambiente donde los datos pueden ser actualizados regionalmente para después ser traídos juntos a repositorios solo de lectura en un servidor central de base de datos.

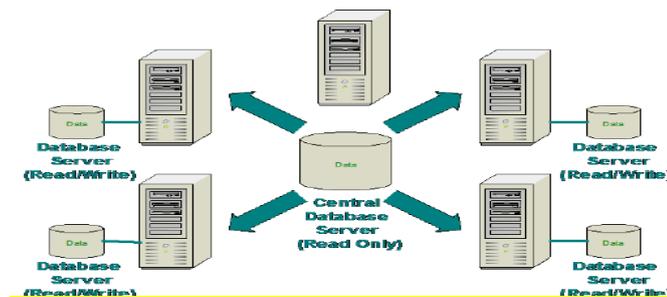


Figura 55. Consolidación de Datos
Fuente: Verástegui, 2001.

3.4.7. Elemento de Replicación

La réplica es una conceptualización de los datos (tablas y columnas) a ser replicadas y una lista a los servidores a los cuales los datos serán replicados.

3.4.7.1. Replicación Sincrónica Two Phase Commit

En la década de los 80, la incorporación de la tecnología two-phase commit, fue la mejor y eficaz manera de distribuir e intercambiar datos. Two-phase commit se encarga de la distribución de datos sincronizados. Cuando los sitios interconectados de distribución estén listos, recién la transacción podrá ser aceptada.

Para la coordinar la aceptación de cada transacción se hace a través del *handshake*.

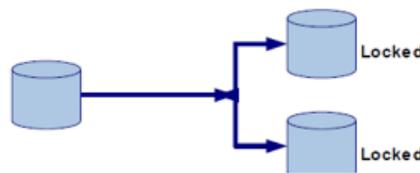


Figura 56. Two – Phase Commit
Fuente: López J., 2001.

En la figura todos los sitios deben estar interconectados para que pueda darse la transacción e el intercambio de datos. Algunas de las operaciones están expuestas a las fallas de los sistemas individuales.

Cuando una corporación necesita absolutamente sincronizar datos distribuidos el two phase commit puede ser conveniente, éste tiene un costo; para aprobar una transacción todos los sitios deben estar distribuidos, la información que se maneja en una corporativa siempre está expuesta a comportamientos individuales de sus componentes. La transacción tiene que esperar cuando algún sitio no esté disponible. Por general, el mecanismo desarrollado de handshaking con mensajes de ida y vuelta entre los sitios para solicitar la aceptación de datos pone un importante cuidado en las redes de la corporación.

En una organización se debería tomar en cuenta el tiempo de distribución de copias actuales de datos. Las localizaciones individuales, al tener acceso a

un backup disponible de información, no requieren que la red ó los sitios remotos estén activos, porque pueden seguir sus actividades locales con el backup disponible. Los sitios distribuidos, tienen como finalidad estar aptos para trabajar con datos que no están sincronizados.

Por ello, se genera la conclusión que los cambios son replicados a nivel de fila a través del mecanismo de TWO PHASE COMMIT.

Los cambios de datos deben aplicarse correctamente en todos los lugares; si en caso en algunos de los sitios no se puede aplicar entonces se genera un rollback de la transacción en todos los sitios, es decir, las bases de datos se devuelven a un estado anterior. Este tipo de replicación es recomendable, para un lugar con redes estables y donde los sitios necesiten sincronización continua.

Igualmente, se plantea el diseño de ambientes de replicación híbridos en la que ciertos modos propaguen sus modificaciones de manera asincrónica o sincrónica. La técnica sincrónica asegura la integridad de datos en base a la disponibilidad y rendimiento del tiempo de respuesta.

3.4.7.2. Replicación Asincrónica

En este tipo de replicación, primero se actualiza la base de datos de origen y después la base de destino. El plazo de renovación; conforme a la configuración, puede ejecutarse en pocos segundos ó en varias horas. Pese a esto; a veces en todos los lugares, los datos están sincronizados.

La replicación asincrónica, en el caso fortuito que se genere una falla ó no esté vigente el sitio, entonces requiere de las demandas locales para avanzar.

La técnica asincrónica extiende la disponibilidad y rendimiento de tiempo de

respuesta. Por tal razón, requiere ser planificado y diseñado para asegurar integridad de datos y fundamentalmente resolver problemas de actualización.¹⁷

A. Diseño de Aplicación con Replicación Asincrónica

Primero, es imprescindible comprender las demandas del negocio, a fin de seleccionar adecuadamente la arquitectura de replicación.

Para establecer las demandas, en la actualidad no hay metodología alguna, más se deben tener presente los siguientes ítems de carácter global:

- ✓ El diseño de la replicación asincrónica debe iniciar con la definición de los datos, ya sea en base a su localización y/o función, así como también conforme a su lugar de origen ó términos de estados actuales. Una vez que se ha definido las características de los datos, llega a ser más fácil entender quién necesita acceso a ellos desde las diferentes localizaciones de la organización.
- ✓ El gestor debe tener la capacidad de elegir los lugares en que los datos serán visibles y a su vez pueden actualizar dicha información.
- ✓ Identificadas las demandas, así como la matriz de datos, entonces se iniciará a planificar la replicación. Para ello, se debe considerar que ingresar a los datos desde un lugar determinado no precisa que esa información este ubicada en ese sitio.
- ✓ En conclusión, solo se debe emplear la replicación cuando un enlace cliente-servidor, presenta una productividad inadecuada hacia un servidor central ó sí tal vez se desea una mayor disponibilidad de ingreso. Asimismo, se recomienda decidir el número de localizaciones a ser sincronizadas, cuando los datos fueron situados en las diferentes ubicaciones de la institución.

Ahora, para las aplicaciones de bases de datos, se elucida con más amplitud

¹⁷ Kroenke D. (2003). Procesamiento de bases de datos: fundamentos, diseño e implementación. México: Pearson Educación.

las consideraciones y actividades para formular la replicación correspondiente.

B. Planificación de Replicación

Para que la replicación sea un éxito, se requiere tener un modelo de trabajo antes de su implementación, ya que es muy similar a proyectar un diseño de base de datos lógico. Por tal razón, se considera tener presente diferentes alcances, incluyendo la decisión de; ¿qué replicar?, ¿quién necesita replicar? y cómo replicarlos?

En consecuencia, se enuncian algunas características importantes que se debe tener presente cuando se planifica replicación:

Cuando hay una distribución de datos, es importante replicar solo los datos precisos y no toda la base de datos. Es necesario entonces tener en cuenta replicar un subconjunto de datos, que es lo que se usó en nuestro Sistema.

- ✓ Se puede usar replicación, si los sitios son solo de lectura y requieren solo información específica de una base de datos. No se debería considerar replicación, si el sitio debe tener capacidades de actualización.
- ✓ La actualización de la información está adaptada por las necesidades de la empresa. Por ejemplo, empresas de marketing, necesitan una información lo más actualizada del momento.
- ✓ El sitio central donde se unifica la información debería ser el elegido a distribuir la información.

C. Determinación del escenario de Replicación

Para establecer el escenario de replicación se debe tener en cuenta siguientes consideraciones:

- ✓ Determinar el objetivo de la replicación de datos (información corporativa, creación de reportes, sistemas de soporte de decisiones)

- ✓ Crear las conexiones de acceso a la red. Advertir que si hay un enlace lento se beneficiaría de tener servidores separados para cada objetivo.
- ✓ Determinar la cantidad de servidores, su estimación ayuda a establecer la carga en el servidor y el tamaño apropiado para el log de transacciones de cada base de datos marcado de replicación.
- ✓ Estimar la frecuencia de replicación, esta característica ayuda a determinar el tiempo adecuado para que cada sitio reciba los datos replicados de la misma manera, la estrategia apropiada de backup¹⁸, el tamaño adecuado para la base de datos de distribución y los requerimientos de balanceo de carga.

Para determinar el plan de implementación, se debe tener en cuenta:

- ✓ El tamaño de la base de datos y logs de transacciones
- ✓ Un horario de replicación (frecuencia)
- ✓ Si en caso se utiliza la actualización en otra parte que no sea en la central, crear métodos y formas de detección y resolución de problemas si se amerita.

3.4.8. Detección y Resolución de Conflictos

Las bases de datos que se encuentran distribuidas presentan determinados factores que establecen su rendimiento con relación a un sistema de red de computadoras. Sin embargo, existen diferentes problemas técnicos que deben ser solucionados para el sistema de base de datos pueda tener una confiabilidad. Los problemas a que se hacen referencia son los que continuación de enumeran¹⁹:

- ✓ Diseño de base de datos distribuidos.
- ✓ Proceso de consultas distribuidas.
- ✓ Administración distribuida.

¹⁸ Término utilizado para describir el proceso de respaldos de una base de datos

¹⁹ Coronel C. (2011). Bases de Datos, Diseño, Implementación y Administración. Panamá: Cengage Learning Editores.

- ✓ Control de concurrencia distribuida.
- ✓ Administración de Deadlocks distribuidos
- ✓ Confiabilidad de los DBMS distribuidos
- ✓ Soporte para sistemas abiertos.
- ✓ Bases de datos heterogéneas

En el momento de replicación de datos, lo más importante es el Control de concurrencia. Dicho factor viola el rendimiento y confiabilidad del proceso de replicación. Por ello, a fin de mantener orden en la integridad de datos, el Control de concurrencia implica la sincronización del acceso a la base de datos distribuida. El Control de concurrencia en una zona distribuido es distinto que para los sistemas centralizados. Se debe tomar en cuenta la consistencia de múltiples copias del BD, porque no solo es un problema de integridad de una base de datos.

Hay muchas maneras para la solución para este problema. Pero las más conocidas de aquellas son: **pesimista y optimista**. La sincronización de la ejecución de los requerimientos del usuario antes de iniciar la ejecución, se le conoce como alternativa pesimista. En cambio, en la alternativa optimista, hace el requerimiento y luego chequea si la ejecución compromete la consistencia de la base de datos. Finalmente se puede decir que existe un tercer tipo de variación que intenta combinar las dos primitivas. Los que son llamados los algoritmos híbridos²⁰.

3.4.8.1. Mecanismos de Control de Concurrencia

En este segmento se hablará de dos tipos de mecanismo: métodos de control de concurrencia pesimista y métodos de control de concurrencia optimista.

²⁰ Price, J. (2004). Sistemas de información gerencial: administración de la empresa digital. México: Pearson Educación.

Inicialmente en su ciclo de vida, los algoritmos pesimistas sincronizan la ejecución concurrente de transacciones. En cambio, los algoritmos optimistas demoran la sincronización de las transacciones hasta su término.

Ambos mecanismos, están conformados por algoritmos basados en lock y en timestamp.

Los locks (seguros) físicos y lógicos en alguna porción ó granulo de la BDD, permiten la sincronización de las transacciones. En donde, una característica relevante, es el tamaño de esa porción (también denominada locking granuly).

3.4.8.2. Conflictos en la Modificación de Datos

Las divergencias potenciales en el cambio de la información podrán ser establecidas a través de la mezcla heterogénea de las demandas del acceso y lugar de objetos- datos. Sí, en las operaciones de la organización se emplea la propagación asincrónica, entonces estará expuestos a riesgos constantes.

Todo ello, sucede cuando los usuarios de diferentes lugares actualizan a la vez, una misma fila a diferentes escalas.

Por ello; al presentarse tal problema, se enumera los potenciales tipos de riesgos y consecuencias que se está expuesto:

Cuando se crea una fila

Los valores para la fila resultante en una falta de unicidad. Otra fila tiene igual clave primaria.

Cuando se remueve una fila

La fila ya ha sido reemplazada

La fila fue cambiada después de ser removida desde la localización original

Cuando se actualiza una fila

La fila anterior fue movida

La fila anterior varió su valor

El nuevo valor de fila es tiene una falta de unicidad

3.4.8.3. Detección de Conflictos

Cuando se trasciende cambios en una zona de replicación pueden pasar diferentes problemas. Por consiguiente, la persona responsable de administrar replicación debe estar capacitada con métodos que permitan la detección y resolución. Cada motor de base de datos tiene sus métodos de detección y resolución.

Tipos de conflictos

Pueden ocurrir 3 tipos de errores, los cuales son:

- ✓ Conflictos de actualización
- ✓ Conflictos de unicidad
- ✓ Conflictos de eliminación

Un **conflicto de actualización** es hallado cuando hay una diferencia entre el valor anterior de la fila replicada y el valor actual en la misma fila en la zona que recibe los datos.

Un **conflicto de unicidad** es hallado cuando una restricción única es quebrantada en las operaciones insert o update en una fila replicada.

Un **conflicto de eliminación** es hallado si una fila cambia en un sitio remoto después que se ha eliminado dicha fila del sitio local. Este tipo de error ocurre cuando el valor anterior de la fila borrada en el sitio local no hace un match (no se ajusta) con el valor actual de la misma fila en el sitio remoto.

A. Tipos de errores

Para identificar conflictos, cada vendedor de base de datos utiliza sus propias técnicas. Los tipos de errores citados; de acuerdo con la presente investigación realizada, son los siguientes:

- ✓ Errores de diseño. – esto sucede cuando un cambio en el diseño maestro difiere con un cambio de diseño en la réplica. Los errores de sincronización y el contenido de las diferentes réplicas empiezan a discrepar
- ✓ Conflictos de sincronización. – pasa cuando los usuarios actualizan el mismo registro en dos réplicas. La sincronización es correcta, pero las modificaciones de una sola tabla son replicadas en las otras.
- ✓ Errores de sincronización. - sucede cuando una modificación en los datos en una réplica no puede ser aplicados a otra réplica puesto que se violaría una restricción, por ejemplo, la integridad referencial. La sincronización es correcta, pero los contenidos de las réplicas no son iguales.

3.4.9. ¿Distribución vs Replicación?

Los datos se distribuyen, tal como se señala a continuación:

Mejora en el rendimiento. – El servidor es gestionado sólo con los datos relacionados al mismo, en donde las tablas son las más aceptablemente cerradas.

Reducción en el costo. – Los servidores chicos son menos onerosos que un servidor monolítico. Sí un servidor presenta errores y/o defectos en un heterogéneo ambiente de servidores, entonces los demás pueden continuar siendo empleados, porque aquel servidor que presentó inconvenientes representa una sola zona de defecto.

Los datos son replicados debido a que permiten:

Soportar distribución de otros datos. – las tablas de datos distribuidos encerrarán las claves de tablas replicadas. Por ello, las tablas de soporte deben ser replicadas a las localizaciones de distribución de datos, y de esta forma

conservar la integridad de los mismos.

Accesibilidad a un mismo dato. – tal como; replicar un resumen de información transaccional en cada servidor, y de esta forma las estadísticas pueden ser originadas por otros servidores.

3.5. Representación de Resultados

3.5.1. Evaluación del Sistema

“La aplicación de la Tecnología Cliente/Servidor con un esquema de Replicación en Bases de Datos asistidos por CASE, permitirá ejecutar un Modelo Informático para la Administración eficaz de los Datos de los alumnos de la I.E.”

En esta investigación la hipótesis planteada que se describe en el párrafo inicial si se cumplió por cuanto:

1. Se creó un esquema de base de datos con la capacidad de soportar sin ningún tipo de modificación el registro de matrícula a la institución educativa seleccionada, este esquema se implementó sobre arquitectura Cliente/Servidor.
2. El modelo de replicación que se usó en CETPRO César Vallejo (Modelo de Consolidación) permite guardar un registro histórico de datos de los estudiantes, de tal manera que la información se unifica en un solo sitio capaz de emitir un sin número de reportes escolares. Se logró mejorar los tiempos y resultados de las diversas etapas de la construcción por usar una correcta metodología orientada a objetos y documentada.

3.5.1.1. Implementación Funcional

Es donde aquí se inicia el trabajo de programación, se guardará todos los códigos utilizados en la programación, Se ejecutará la implementación en base a la documentación presentada anteriormente, se hizo un prototipo inicial el que se irán mejorando constantemente en un proceso iterativo incremental recursivo.

3.5.1.2. Test y Pruebas

En la implementación funcional del software aplicamos exigentes controles de prueba para detectar fallas en el mismo para poder evitar que en el instante de su aplicación se presente un conflicto en algunas de las funciones, por ejemplo, en la introducción de los datos que estarán siendo procesados y automatizados.

Contiene: pruebas unitarias, pruebas funcionales y pruebas de integración, carga de datos iniciales, datos de prueba, etc.

3.5.1.3. Estabilización

Después de realizar los test y las pruebas, Se realizó el afinamiento del prototipo, apariencia, validaciones, etc., La solución pasa a un entorno real de producción, restringido en número de usuarios y en condiciones para llevar un control efectivo de la situación.

3.5.1.4. Control de Calidad

Con la finalidad de determinar la calidad de los productos desarrollados se realizarán diversas pruebas que contemplan monitorear la etapa del progreso del software para asegurar el respeto a los procesos y estándares de garantía de calidad, tales como el control y entregas exigidas y definidas por la organización demandante.

3.5.1.5. Desarrollo de Planes

Los planes se fundamentarán principalmente en los sistemas de información. Planes de mantenimiento, políticas de backups, inventarios, entre otros se encuentran contenidos en los planes de contingencia.

3.5.1.6. Despliegue

En la etapa del despliegue se ejecutarán pruebas de conectividad y se explicará como una aplicación se desplaza a través de la infraestructura. La finalidad del despliegue es definir los componentes específicos que deben tener una aplicación que desplaza por medio de él.

4. CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

Primera. Fundamentados en el uso de métodos UML, procedimientos de replicación y aplicaciones en niveles para el bosquejo de sistemas, se perfeccionó e implementó un sistema académico piloto para la gestión de la educación técnica productiva del CEPTRON “Cesar Vallejo”.

Segunda. La mejor metodología moderna para fundamentar el bosquejo de un proyecto de software es la UML (Unified Modeling Language).

Tercera. Se estudió los distintos tipos de soluciones DBMS y comprobó su compatibilidad con este tipo de aplicaciones.

Cuarta. Se mejoró los procesos y control de la institución estudiada, a través del sistema académico piloto para la gestión de la educación técnica productiva del CEPTRON “Cesar Vallejo”.

Quinta. Se deberá implementar arquitecturas distribuidas basadas en n-capas, sólo si re-empleará la lógica de negocio entre las herramientas y/o cuando se realice la migración a la plataforma Back-End.

4.2. Recomendaciones

Primera. Se sugiere que el proyecto informático a implementar, debe brindar soluciones de acuerdo con el ambiente institucional en que se desarrollará. Por ello, deberá tener presente los siguientes aspectos: Infraestructura de Comunicación y Datos - recursos y afluencia

Segunda. Para otorgar un producto personalizado que satisfaga las expectativas del usuario final del sistema, se sugiere ejecutar previamente; el levantamiento de información con los entes involucrados en los procesos diarios, y políticas internas del sistema estudiado.

Tercera. Para brindar cumplimiento a los entregables definidos en el proyecto de sistema, se sugiere realizar previamente un análisis detallado y puntual a las especificaciones operativas y documentación relacionada.

Cuarta. Se sugiere siempre tener un respaldo para toda la información del negocio en una ubicación diferente al equipo de servidor.

5. REFERENCIAS BIBLIOGRÁFICAS

- Booch, G. (1996). *“Análisis y Diseño Orientado a Objeto”*s. Madrid: Díaz de Santos.
- Informática (2016). *“Diagramas de UML”*. Recuperado de <http://informatica-iutll.blogspot.pe/2013/03/diagramas-de-uml.html>
- Natabloguml (2016). *“Diagrama de secuencia”*. Recuperado de <http://natabloguml.blogspot.pe/2013/12/diagrama-de-secuencia.html>
- Rumbaugh, J. (2010). *“Modelado y diseño orientado a objetos: Metodología OMT”*. México: Prentice Hall
- Rumbaugh, J., Booch, G. & Jacobson, I. (2008). *“El lenguaje unificado de modelado”*. Panamá: Addison Wesley
- Aguilar, P. (2011). Políticas de almacenamiento y resguardo de la seguridad (Seguridad Informática). México: Editorial EDITEX.
- Calderón, C., Dávila, J., & Otero, D. (2013). Replicación en aplicaciones; sistema de administración académica de la UTN. Ecuador: Universidad Técnica del Norte. <http://repositorio.utn.edu.ec/handle/123456789/1105>
- Cegarra, J. (2011). *Metodología de la investigación científica y tecnológica*. Barcelona: Ediciones Díaz de Santos.
- Sparxsystems (2016). *“Diagrama de secuencia”*. Recuperado de http://www.sparxsystems.com.ar/resources/tutorial/dynamic_model.html
- Sparxsystems (2016). *“Diagrama de clase”*. Recuperado de http://www.sparxsystems.com.ar/resources/tutorial/uml2_classdiagram.html
- Sparxsystems (2016). *“Diagrama de caso de uso”*. Recuperado de http://www.sparxsystems.com.ar/resources/tutorial/uml2_usecasediagram.html
- Cisneros J. (1998). *Panorama sobre base de datos*. Un enfoque práctico. México: UABC.
- Coronel C. (2011). *Bases de Datos, Diseño, Implementación y Administración*. Panamá: Cengage Learning Editores.

- Dordoigne J. (2015). *Redes informáticas: Nociones fundamentales*. Barcelona: Ediciones ENI.
- Kroenke D. (2003). *Procesamiento de bases de datos: fundamentos, diseño e implementación*. México: Pearson Educación.
- López J. (2001). *Introducción a los sistemas de bases de datos*. México: Pearson Educación.
- Verástegui L. (2001). *Introducción a los sistemas de bases de datos*. México: Pearson Educación.
- Universidad José Carlos Mariátegui. (2016). “*Manual De Elaboración De Tesis, Trabajo De Suficiencia Profesional Y Artículo Científico*”. ISO 9001:2015.
- Wikispaces. (2016). “*Lenguaje Unificado de Modelado – UML*”. Noviembre 14, 2016, de Wikispaces Sitio web: <https://procesosdesoftware.wikispaces.com/UML>