



**PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL ECUADOR
SEDE AMBATO**
SERÉIS MIS TESTIGOS

UNIDAD ACADEMICA

Tema:

**DESARROLLO DE UN SISTEMA DISTRIBUIDO PARA EVALUAR LOS
CONOCIMIENTOS DE SUFICIENCIA EN INFORMÁTICA EN LA
PUCESA**

Disertación de grado:

Previo a la obtención del título de Ingeniero De Sistemas

Autor:

JOSE DARWIN CASTILLO JUMBO

Asesor:

ING. Msc. PATRICIO MEDINA

Ambato – Ecuador

Mayo 2010

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

SEDE AMBATO

HOJA DE APROBACIÓN

Tema:

DESARROLLO DE UN SISTEMA DISTRIBUIDO PARA EVALUAR LOS
CONOCIMIENTOS DE SUFICIENCIA EN INFORMÁTICA EN LA PUCESA

Autor:

JOSE DARWIN CASTILLO JUMBO

Patricio Medina, Ing. Msc. f. _____

DIRECTOR DE DISERTACIÓN.

Darío Robayo, Ing. f. _____

CALIFICADOR

Galo López, Ing. f. _____

CALIFICADOR

Santiago Acurio, Ing f. _____

DIRECTOR UNIDAD ACADÉMICA

Pedro José Rivadeneira Río, Dr. f. _____

SECRETARIO GENERAL PUCESA

Ambato – Ecuador

Junio 2010

**DECLARACIÓN DE AUTENTICIDAD
Y RESPONSABILIDAD**

Yo, José Darwin Castillo Jumbo portador de la cédula de ciudadanía No. 1104147440 declaro que los resultados obtenidos en la investigación que presento como informe final, previo la obtención del título de Ingeniero de Sistemas son absolutamente originales, auténticos y personales.

En tal virtud, declaro que el contenido, las conclusiones y los efectos legales y académicos que se desprenden del trabajo propuesto de investigación y luego de la redacción de este documento son y serán de mi sola y exclusiva responsabilidad legal y académica.

José Darwin Castillo Jumbo

CI. 1110414744-0

AGRADECIMIENTO

A Dios, por darme vida, salud, inteligencia y la determinación suficiente para alcanzar todas mis metas.

A mis padres por el apoyo brindado, sin el cual no hubiera podido materializar este sueño, y sobre todo por ser mí modelo de esfuerzo y sacrificio, el mismo que deseo reflejar a mis hijos cuando los tenga.

A mí amor y amiga Anita Paola, por su cariño, comprensión y apoyo en la realización de este trabajo.

A la empresa "IT del Ecuador" que me dio la oportunidad de poner en práctica mis capacidades como profesional en el desarrollo de software.

A mí estimado asesor, Ing. Patricio Medina por compartir su experiencia y conocimientos a lo largo de mi vida universitaria y en la preparación de esta Disertación.



DEDICATORIA

Con todo mi afecto a mis queridísimos hermanos Edgar y Arlyn, tan importantes para mí.

A todos quienes comparten mi pasión por ésta carrera, la que anhelo contribuya al desarrollo económico y social de la nación.

Y al más grande ejemplo de humildad, carisma, talento y perseverancia que haya podido tener, mi único ídolo el señor Michael Joseph Jackson, para él mi eterna admiración

Resumen

El objetivo principal de esta Disertación, es desarrollar un sistema distribuido que evalúe exámenes de suficiencia en informática para la “PUCESA” y certificar los conocimientos y capacidades informáticas del estudiantado.

La evaluación es el tema con mayor protagonismo del ámbito educativo, y no porque se trate de un tema nuevo sino porque la sociedad en su conjunto está más consciente de la importancia y las consecuencias de evaluar y ser evaluado.

Los directivos van observando mediante la evaluación la ruta que va tomando la institución, lo que les permite aplicar estrategias que contribuyan a superar los problemas de aprendizaje. Es a través de la evaluación que la institución puede indiscutiblemente proyectar sus planes de mejoramiento en la calidad académica, para ello se utiliza la programación en cuyo proceso de creatividad se busca resolver problemas y también decir que es una parte fundamental, vital para la elaboración de cualquier sistema informático.

Al elaborar este sistema era necesario reunir las características que harían del mismo una fuente de recursos efectivos en la tarea de alcanzar un método evaluativo cada vez más preciso, y estoy seguro que con “CRONOS” lo conseguiremos, ya que luego de investigar durante mucho tiempo y analizar las necesidades que hay que satisfacer inmediatamente, se proporcionará tanto a la institución como al alumnado una herramienta útil, capaz de incrementar la competitividad tan común en esta época de avances acelerados, y cómo no plantear el factor tiempo si es precisamente lo que estuvo presente en la realización de este proyecto, además es algo inevitable en una época tan moderna como la nuestra.

Abstract

The main objective of this dissertation is to develop a distributed system to evaluate computer proficiency exams for the "PUCESA" and certify computer skills and abilities of students.

Evaluation is the issue more prominence in the field of education, not because it is a new theme but because society as a whole is more aware of the importance and consequences of assessment and evaluation.

Emerging Managers by evaluating the path that is taking the institution, allowing them to implement strategies that help overcome learning problems. It is through the evaluation that the institution can definitely design their plans for improvement in academic quality, for this I used the programming in which the process of looking for creativity and problem solving that is a fundamental, vital to the development of any computer system.

In developing this system was needed that would meet the characteristics of the source of effective remedies in the task of reaching assessment method increasingly precise, and I am sure that with "CRONOS" will succeed, because after long investigation and analyze requirements to be met immediately, they provide both the institution and the student a useful tool, capable of enhancing the competitiveness so common in this era of accelerated progress, and how not to raise the factor of time if that is what was present at the execution of this project and it is inevitable in an age so modern as ours.

Índice Tabla De Contenidos

Autoría	iii
Agradecimiento	iv
Dedicatoria	v
Resumen	vi
Abstract	vii
1. CAPITULO I: Problema de Investigación	
1.1. Antecedentes	1
1.2. Planteamiento del problema	2
1.2.1. Formulación del problema	3
1.2.2. Delimitación del problema	3
1.3. Objetivos	4
1.3.1. Objetivo general	4
1.3.2. Objetivos específicos	4
1.4. Justificación	5
2. CAPITULO II: FUNDAMENTO TEORICO	
2.1. Evaluación	6
2.1.1. Concepto	6
2.1.2. Tipo de evaluación	7
2.1.2.1. Formativa	7
2.1.2.2. Diagnostica	7
2.1.3. Principales Actores de la evaluación	8
2.1.3.1. Directivos	8
2.1.3.2. Docentes	8
2.1.3.3. Estudiantes	8
2.1.4. La evaluación como método de aprendizaje	8
2.1.4.1. Objetivos de la evaluación	9
2.1.5. Métodos de puntuación	10
2.1.5.1. Puntuación directa	10
2.1.5.2. Puntuación corregida	10
2.2. Redes locales	11
2.2.1. Conceptos	11

2.2.2.	Protocolos de red	11
2.2.3.	Funcionamiento	12
2.2.3.1.	Funcionamiento físico	12
2.2.3.2.	Funcionamiento lógico	13
2.3.	Sistemas Distribuidos	14
2.3.1.	Características	16
2.3.2.	Modelos de seguridad	18
2.3.3.	Tecnologías	20
2.3.3.1.	Middleware	20
2.3.3.2.	Web Services	21
2.3.4.	Modelos Arquitectónicos	23
2.3.4.1.	Arquitectura lógica	24
2.3.4.2.	Arquitectura física	25
2.4.	Base de datos	30
2.4.1.	Definición	30
2.4.2.	Modelo de base de datos	31
2.4.2.1.	Modelo de jerárquico de datos	31
2.4.2.2.	Modelo de datos en red	31
2.4.2.3.	Modelo relacional de datos	32
2.4.3.	Creación de una base de datos	32
2.4.4.	Seguridad en una base de datos	33
2.4.4.1.	Medidas de seguridad	34
2.5.	Programación	35
2.5.1.	Programación orientada a objetos	35
2.5.1.1.	Conceptos fundamentales	35
2.5.1.2.	Características de la Prog. orientada a objetos	37
2.5.2.	Lenguaje de programación C#	38
2.5.3.	Linq	39
2.6.	Modelado de sistema	41
2.6.1.	UML	41
2.6.2.	Diagramas de casos de uso	42
2.6.3.	Diagramas de secuencia	43
3.	CAPITULO III: DESARROLLO DEL SISTEMA DISTRIBUIDO	
3.1.	Desarrollo del análisis	44
3.1.1.	Requerimientos del sistema	44
3.1.2.	Requerimientos funcionales	45
3.1.3.	Requerimientos no funcionales	48
3.1.4.	Descripción de los casos de uso	48
3.1.5.	Diagramas de casos de uso	67

3.2.	Diseño del sistema	72
3.2.1.	Definición de la arquitectura	72
3.2.2.	Diagramas de clases	74
3.2.3.	Diagramas de secuencias	78
3.2.4.	Diseño de la interfaz del sistema	92
3.2.4.1.	Módulo Administrador	93
3.2.4.2.	Módulo Examinador	98
3.3.	Construcción	100
3.3.1.	Selección de herramientas	100
3.3.2.	Requerimientos de hardware y software del equipo de desarrollo	101
3.3.3.	Codificación	101
3.3.4.	Pruebas	106
3.3.4.1.	Pruebas de unidad	107
3.3.4.2.	Pruebas de aceptación	114
3.3.4.3.	Pruebas de carga	116
3.3.5.	Implementación	119
3.3.5.1.	Requerimientos de hardware y software del equipo implementación	120
3.3.5.2.	Documentación	122
4.	CAPITULO IV: CONCLUSIONES Y RECOMENDACIONES	
4.1.	Conclusiones	123
4.2.	Recomendaciones	124
	Material de referencia	125
	Glosario	128
	Anexos	131

Tabla de gráficos

Gráfico 2.1: Topología red física	13
Gráfico 2.2: Funcionamiento de una red lógica	13
Gráfico 2.3: Arquitectura cliente servidor	26
Gráfico 2.4: Arquitectura de dos niveles	28
Gráfico 2.5: Arquitectura de tres niveles	28
Gráfico 2.6: Arquitectura de n niveles	29
Gráfico 3.1: Diagrama caso de uso escuelas	67
Gráfico 3.2.: Diagrama caso de uso cátedras	68
Gráfico 3.3: Diagrama caso de uso docentes	68
Gráfico 3.4: Diagrama caso de uso alumnos	69
Gráfico 3.5: Diagrama caso de uso exámenes	69
Gráfico 3.6: Diagrama caso de uso bco. preguntas	70
Gráfico 3.7: Diagrama caso de uso reportes	70
Gráfico 3.8: Diagrama caso de uso usuarios	71
Gráfico 3.9: Diagrama caso de uso roles	71
Gráfico 3.10: Diagrama caso de uso examinador	72
Gráfico 3.11: Diagrama de clases entidades de negocio	74
Gráfico 3.12: Diagrama de clases acceso a datos	75
Gráfico 3.13: Diagrama de clases lógica de negocios	76
Gráfico 3.14: Diagrama de clases presentación	77
Gráfico 3.15: Diagrama de clases servicio web	78
Gráfico 3.16: Diagrama de secuencia validar usuario	79
Gráfico 3.17: Diagrama de secuencia crear escuela	79
Gráfico 3.18: Diagrama de secuencia modificar escuela	80
Gráfico 3.19: Diagrama de secuencia eliminar escuela	80
Gráfico 3.20: Diagrama de secuencia ver escuela	81
Gráfico 3.21: Diagrama de secuencia crear cátedra	81
Gráfico 3.22: Diagrama de secuencia modificar cátedra	82
Gráfico 3.23: Diagrama de secuencia eliminar cátedra	82
Gráfico 3.24: Diagrama de secuencia ver cátedra	83
Gráfico 3.25: Diagrama de secuencia crear alumno	83
Gráfico 3.26: Diagrama de secuencia modificar alumno	84
Gráfico 3.27: Diagrama de secuencia eliminar alumno	84
Gráfico 3.28: Diagrama de secuencia ver alumno	85
Gráfico 3.29: Diagrama de secuencia crear docente	85
Gráfico 3.30: Diagrama de secuencia modificar docente	86
Gráfico 3.31: Diagrama de secuencia eliminar docente	86

Gráfico 3.32: Diagrama de secuencia ver docente	87
Gráfico 3.33: Diagrama de secuencia crear examen	87
Gráfico 3.34: Diagrama de secuencia modificar examen	88
Gráfico 3.35: Diagrama de secuencia eliminar examen	88
Gráfico 3.36: Diagrama de secuencia ver examen	89
Gráfico 3.37: Diagrama de secuencia crear pregunta	89
Gráfico 3.38: Diagrama de secuencia modificar pregunta	90
Gráfico 3.39: Diagrama de secuencia eliminar pregunta	90
Gráfico 3.40: Diagrama de secuencia eliminar pregunta	91
Gráfico 3.41: Diagrama de secuencia ver reporte	91
Gráfico 3.42: Diagrama de secuencia Tomar examen	92
Gráfico 3.43: Acceso módulo administrador	93
Gráfico 3.44: pantalla principal	94
Gráfico 3.45: Pantalla administrar escuelas	94
Gráfico 3.46: Pantalla administrar cátedras	95
Gráfico 3.47: Pantalla administrar docentes	95
Gráfico 3.48: Pantalla administrar alumnos	96
Gráfico 3.49: Pantalla administrar banco de preguntas	96
Gráfico 3.50: Pantalla administrar exámenes	97
Gráfico 3.51: Pantalla administrar usuarios	97
Gráfico 3.52: Pantalla administrar roles	98
Gráfico 3.53: Pantalla acceso al módulo examinador	99
Gráfico 3.54: Pantalla examinador información	99
Gráfico 3.55: Pantalla examinador	100
Gráfico 3.56: Transact SQL Creación de una tabla	102
Gráfico 3.57: Transact SQL Creación de un procedimiento almacenado	102
Gráfico 3.58: Transact SQL Creación de un disparador	103
Gráfico 3.59: Explorador de la solución del sistema CRONOS	103
Gráfico 3.60: Código fuente de la capa de acceso a datos	104
Gráfico 3.61: Código fuente de la capa de acceso a datos	104
Gráfico 3.62: Código fuente de la capa de lógica de negocio	105
Gráfico 3.63: Código fuente de la capa de servicios	105
Gráfico 3.64: Código fuente de la capa de presentación	106
Gráfico 3.65: Ejecución de pruebas unitarias	112
Gráfico 3.66: Resultado de pruebas unitarias I	112
Gráfico 3.67: Error de prueba	113
Gráfico 3.68: Resultado de pruebas unitarias II	113
Gráfico 3.69: Prueba de carga, ListarAlumnoByIdCedula	117
Gráfico 3.70: Prueba de carga, ListarExamenPreguntaByIdExamenTest	117

Gráfico 3.71: Gráfico Prueba de carga (contadores)	118
Gráfico 3.72: Prueba de carga (resultados)	118
Gráfico 3.73: Implementación (Estructura física)	119
Gráfico 3.74: Implementación (Elementos del sistema)	120

Tablas

Tabla 3.1: Requerimientos funcionales (Módulo de escuelas)	46
Tabla 3.2: Requerimientos funcionales (Módulo de cátedras)	46
Tabla 3.3: Requerimientos funcionales (Módulo de exámenes)	46
Tabla 3.4: Requerimientos funcionales (Módulo de preguntas)	47
Tabla 3.5: Requerimientos funcionales (Módulo de docentes)	47
Tabla 3.6: Requerimientos funcionales (Módulo de alumnos)	47
Tabla 3.7: Requerimientos funcionales (Módulo de seguridades)	47
Tabla 3.8: Requerimientos funcionales (Módulo de reportes)	48
Tabla 3.9: Requerimientos funcionales (Módulo examinador)	48
Tabla 3.10: Caso de uso (Autenticar Usuario)	49
Tabla 3.11: Caso de uso (Administrar escuela)	50
Tabla 3.12: Caso de uso (Crear escuela)	50
Tabla 3.13: Caso de uso (Modificar escuela)	51
Tabla 3.14: Caso de uso (Eliminar escuela)	51
Tabla 3.15: Caso de uso (Ver escuela)	52
Tabla 3.16: Caso de uso (Administrar Cátedra)	52
Tabla 3.17: Caso de uso (Crear Cátedra)	53
Tabla 3.18: Caso de uso (Modificar Cátedra)	53
Tabla 3.19: Caso de uso (Eliminar Cátedra)	54
Tabla 3.20: Caso de uso (Ver Cátedra)	54
Tabla 3.21: Caso de uso (Administrar Alumno)	55
Tabla 3.22: Caso de uso (Crear Alumno)	55
Tabla 3.23: Caso de uso (Modificar Alumno)	56
Tabla 3.24: Caso de uso (Eliminar Alumno)	56
Tabla 3.25: Caso de uso (Ver Alumno)	57
Tabla 3.26: Caso de uso (Administrar Docente)	57
Tabla 3.27: Caso de uso (Crear Docente)	58
Tabla 3.28: Caso de uso (Modificar Docente)	58
Tabla 3.29: Caso de uso (Eliminar Docente)	59
Tabla 3.30: Caso de uso (Ver Docente)	59
Tabla 3.31: Caso de uso (Administrar Preguntas)	60
Tabla 3.32: Caso de uso (Crear Pregunta)	60
Tabla 3.33: Caso de uso (Modificar Pregunta)	61
Tabla 3.34: Caso de uso (Eliminar Pregunta)	61
Tabla 3.35: Caso de uso (Ver Pregunta)	62
Tabla 3.36: Caso de uso (Administrar Exámenes)	62
Tabla 3.37: Caso de uso (Crear Examen)	63

Tabla 3.38: Caso de uso (Modificar Examen)	63
Tabla 3.39: Caso de uso (Eliminar Examen)	64
Tabla 3.40: Caso de uso (Ver Examen)	64
Tabla 3.41: Caso de uso (Agregar pregunta a un examen)	65
Tabla 3.42: Caso de uso (Eliminar Pregunta de un examen)	65
Tabla 3.43: Caso de uso (Reporte examen)	66
Tabla 3.44: Caso de uso (Tomar examen)	66
Tabla 3.45: Caso de uso (Ver resultado del examen)	67
Tabla 3.46: Paquete del sistema (Examinador)	100
Tabla 3.47: Requerimientos de hardware del equipo de desarrollo	101
Tabla 3.48: Requerimientos de software del equipo de desarrollo	108
Tabla 3.49: Prueba Unitaria GrabarAlumno	107
Tabla 3.50: Prueba Unitaria GrabarExamen	107
Tabla 3.51: Prueba Unitaria GrabarExamenPregunta	108
Tabla 3.52: Prueba Unitaria ListarAlumnoByCedula	108
Tabla 3.53: Prueba Unitaria ObtenerUltimoIdExamen	108
Tabla 3.54: Prueba Unitaria ListarProfesionalByCedula	109
Tabla 3.55: Prueba Unitaria ListarEscuela	109
Tabla 3.56: Prueba Unitaria ListarCatedraByIDEscuela	109
Tabla 3.57: Prueba Unitaria ListarPreguntaByIDCatedra	110
Tabla 3.58: Prueba Unitaria ObtenerUltimoIdExamen	110
Tabla 3.59: Prueba Unitaria ListarRespuestaPracticaByIdPregunta	110
Tabla 3.60: Prueba Unitaria ListarRespuestaEnlaceByIDPregunta	111
Tabla 3.61: Prueba Unitaria ListarRespuestaDecisionByIDPregunta	111
Tabla 3.62: Prueba Unitaria ObtenerNombreCatedraById	111
Tabla 3.63: Prueba Aceptación Modulo Administrador	115
Tabla 3.64: Prueba Aceptación Modulo Examinador	115
Tabla 3.65: Prueba de carga	116
Tabla 3.66: Requerimientos de hardware y software del Servidor BDD	121
Tabla 3.67: Requerimientos de hardware y software del Servidor Web	121
Tabla 3.68: Requerimientos de hardware y software de PC	121

CAPITULO I

1 Problema de Investigación

1.1 Antecedentes

Las instituciones y empresas en un inicio utilizaban el computador para realizar trabajos rápidos y sistemáticos, sin embargo, este uso se ha venido extendiendo al área educativa, ante lo cual se han planteado diversas propuestas para el desarrollo de programas que permitan optimizar el proceso de evaluación de exámenes de suficiencia y demás.

Si consideramos a la evaluación como el control de las situaciones que ocurren, la modificación de conducta o la adquisición de una competencia en el estudiantado, es importante que las instituciones educativas cuenten con los procedimientos e instrumentos idóneos para juzgar el grado en que se dan los cambios, tanto al final del proceso como durante el mismo.

Mediante la evaluación se puede conocer hasta qué punto los alumnos han modificado su conducta como un resultado planeado y directo de la acción educativa. El proceso de la evaluación incluye una serie continua e interrelacionada de decisiones relativas a la instrucción que buscan incrementar la calidad del

conocimiento de los alumnos. Sin embargo, esta efectividad depende en gran medida de la calidad de información dada por la evaluación sobre la cual se habrán de basar las decisiones que normen cada etapa de dicho proceso. De esta situación se desprende la primordial importancia que adquiere la evaluación dentro del proceso de acreditar alumnos de calidad.

Las instituciones educativas de nuestros días sienten la necesidad de contar con medios idóneos y seguros para establecer hasta qué punto los estudiantes alcanzan las metas educativas preestablecidas, en otros términos, cómo llegar a una justa y válida evaluación del conocimiento.

1.2 Planteamiento del problema

La escuela de Ingeniería de Sistemas de la “Pontificia Universidad Católica del Ecuador Sede Ambato”, debido a la falta de un sistema distribuido que permitan la generación y evaluación de exámenes individualizados en forma automatizada, está obligada a elaborar un examen de suficiencia informática único para ser empleado en diferentes niveles e implica los siguientes sub problemas.

- El estudiantado no es conceptuado de una forma óptima y veraz.
- Posibles errores de puntuación del examen por parte del docente.
- Respuesta no inmediata en la entrega de la nota adquirida en el examen.
- Desestimación de la seguridad de la información en la construcción de sistemas.

1.2.1 Formulación del problema

¿De qué manera pueden los avances tecnológicos vigentes en la realización de sistemas distribuidos, mejorar y modernizar la metodología de la evaluación educativa en la PUCESA?

1.2.2 Delimitación del problema

La elaboración de este proyecto se debe a las necesidades imperiosas de la PUCESA, de tener mayor competitividad y efectividad en servicio de la educación.

Para esto se desarrollara un sistema distribuido para evaluar los conocimientos de suficiencia en informática, que ayudaran a la toma de decisiones, por parte de sus directivos y de quienes imparten la cátedra de informática. Para esto el desarrollo del proyecto engloba lo siguiente:

- Será elaborado para la “PUCESA”
- Implementado para que funcione en la red local de la “PUCESA”.
- Desarrollado con la tecnología .NET y con un lenguaje de programación orientado a objetos.
- El sistema distribuido será desarrollado con la arquitectura n capas con consumo de Web Services.
- Base de datos de los docentes (administradores) y el estudiantado (usuarios), para identificarles.
- Evaluará exámenes personalizados con preguntas de decisión simple, múltiple, enlace y prácticas.

Para el desarrollo del proyecto se toman a consideración los siguientes temas para la toma de exámenes.

- Básico e intermedio de Windows.
- Básico e intermedio de Microsoft Power Point y Microsoft Project.
- Intermedio y avanzado de Microsoft Word y Microsoft Excel.

1.3 Objetivos

1.3.1 Objetivo general

Desarrollar un sistema distribuido que evalúe exámenes de suficiencia en informática para la “PUCESA” y certificar los conocimientos y capacidades informáticas del estudiantado.

1.3.2 Objetivos específicos

- Conceptuar al estudiantado según sus verdaderos conocimientos y capacidades.
- Eliminar errores humanos en la puntuación de una prueba.
- Dar a conocer la calificación en el espacio y tiempo que el estudiante tome el examen.
- Incentivar la creatividad de los estudiantes en la construcción de sistemas distribuidos tomando en cuenta la seguridad de la información.

1.4 Justificación

La seguridad y confiabilidad son necesidades elementales en las medidas con fines de evaluación, por lo que debe tomarse en cuenta el carácter práctico de los instrumentos de medición, dentro de este marco de utilidad tienen importancia el factor de ahorro de tiempo y las características que facilitan la administración.

Por eso la elaboración de un sistema distribuido, para la evaluación de exámenes de suficiencia en informática en la “PUCESA”, suministrando pruebas personalizadas, es de mucha importancia ya que será una de las primeras instituciones educativas en la provincia en desarrollarla e implementarla.

Referente a lo local, el estudio de sistemas distribuidos ayudara a que los estudiantes de la Escuela de Sistemas de la “Pontifica Universidad Católica del Ecuador Sede Ambato” consideren la importancia de proteger infraestructuras, crear y preservar la confiabilidad de la información, estas son razones que por sí solas justifican la realización del presente proyecto.

CAPITULO II

2 Fundamento teórico

2.1 Evaluación

“Una vida sin exámenes, no merece ser vida” Sócrates.

La evaluación es el tema con mayor protagonismo del ámbito educativo, y no porque se trate de un tema nuevo en absoluto, sino porque administradores, educadores, padres, alumnos y toda la sociedad en su conjunto son más conscientes que nunca de la importancia y las consecuencias del hecho de evaluar o de ser evaluado. Existe una gran necesidad de alcanzar alta calidad educativa, de aprovechar adecuadamente los recursos, el tiempo y los esfuerzos, por otra parte, el nivel de competencia entre los individuos e instituciones también es mayor.

2.1.1 Concepto de la evaluación

En su acepción más próxima se trata de un instrumento que trata de medir el comportamiento y/o conocimiento humano, por lo tanto esto implica y exige un proceso minucioso en su elaboración.

En esencia una evaluación constituye una especie de estímulo ante el cual se pone a prueba a los individuos para que estos manifiesten sus conocimientos o habilidades.

2.1.2 Tipos de evaluación

2.1.2.1 Formativa

Se caracteriza por no tener calificación, sino una apreciación de la calidad del trabajo académico realizado, pues es la que nos permite determinar en cada segmento o tramo del Curso o Carrera los resultados obtenidos, para realizar los ajustes y adecuaciones necesarias para llegar al éxito, con la excelencia que demanda la sociedad actual en estos tiempos posmodernos.

2.1.2.2 Diagnóstica

Nos enseña cuales son los conocimientos previos acerca de un tema, determina fortalezas y limitaciones. Su función primordial: ubicar/clasificar.

2.1.3 Principales Actores De La Evaluación

2.1.3.1 Directivos

Los directivos van observando mediante la evaluación la ruta que va tomando la institución, lo que les permite aplicar estrategias que contribuyan a superar los problemas de aprendizaje. Es a través de la evaluación que la institución puede indiscutiblemente proyectar sus planes de mejoramiento en la calidad académica.

2.1.3.2 Docentes

Son aquellos que indican hacia dónde va el proceso de aprendizaje por eso la misión del docente y sus responsabilidades exige preparación lustrada y formación

minuciosa, de esta manera está actualizando sus instrumentos de evaluación, para tener resultados más efectivos y tomar decisiones a tiempo, es importante que el docente no solamente se centre en qué aprenden los estudiantes, sino también en cómo lo aprenden, teniendo en cuenta que ellos sean cada vez más conscientes de sus propios procesos.

Evaluar adecuadamente no es fácil; se trata de una destreza que los docentes deben desarrollar en su formación inicial y continua, para que logren realizar un uso significativo de esta práctica en el aula.

2.1.3.3 Estudiantes

Por medio de la evaluación el estudiante puede auto verificar los logros y dificultades que presenta, en directa relación con lo anterior, es necesario considerar que un estudiante puede llegar a rechazar un tema de estudio solamente por las notas que obtiene en él, pues llega a convencerse de que ‘es malo en tal área’. De allí la importancia de entregar una buena retroalimentación, que ayude al estudiante a entender sus logros, sus dificultades y las formas de mejorar su aprendizaje. Solamente de esta manera entenderá que su mala calificación ‘no es perpetua’, sino que existe la posibilidad de progresar si se mejora lo que el docente sugiere.

2.1.4 La evaluación como método de aprendizaje

Lo primero que se debe hacer al preparar la enseñanza es **fijar ciertos criterios de evaluación**, que especifiquen qué se espera que los estudiantes aprendan. En

segundo lugar, **planificar escenarios de evaluación coherentes con la expectativa** definida -esto es, una serie de actividades, organizadas según una lógica que va desde lo más cercano a lo más complejo-, que permitan monitorear los logros de los estudiantes con respecto a los criterios definidos. Junto con ello, durante el desarrollo del escenario se debe entregar **retroalimentación efectiva y tomar decisiones pedagógicas pertinentes**, en base a la evidencia de evaluación que el docente analiza.

Para prevenir el fracaso es necesario evaluar muy bien, pero con finalidades formativas y formadoras. Sin identificar las dificultades de cada estudiante y sus posibles causas, no hay posibilidad de ayudarles a superarlas y, sin enseñarles a autoevaluarse para que ellos mismos sean capaces de encontrar caminos para salvar los obstáculos, difícilmente aprenderán a aprender, en conclusión según sean los exámenes planteados se promoverá la enseñanza de contenidos muy diversos y con metodologías distintas.

2.1.4.1 Objetivos de la evaluación

- Evaluar el nivel del proceso individual
- Permite medir la eficiencia del docente
- Motiva al estudiante a aprender
- Facilita la enseñanza inmediata
- Sirve de diagnóstico

- Mide la capacidad para resolver problemas
- Muestra adecuadamente los contenidos de la enseñanza
- Permite tomar dediciones a los actores involucrados en la misma

2.1.5 Métodos de puntuación

En cuanto al cálculo de las puntuaciones de una evaluación, debemos decir que cuando calificamos una prueba, podemos obtener dos tipos de puntuaciones:

2.1.5.1 Puntuación directa

La puntuación directa se obtiene otorgando un valor a cada uno de los ítems contestados correctamente.

2.1.5.2 Puntuación corregida

Este método es usado cuando se pretende eliminar los efectos de “buena suerte” en respuestas correctas por parte del estudiante, a continuación la fórmula que se emplean para hallar la puntuación corregida cuando no se tiene en cuenta las omisiones: $P = A - (E / N - 1)$.

A = Numero de acierto.

E = Numero de errores.

N = Numero de alternativas del ítem.

P = Puntuación corregida.

2.2 Redes Locales

2.2.1 Concepto

Una red local de computadores es una agrupación de componentes informáticos con la naturaleza de poder compartir información y recursos entre ellos, con el objetivo de abaratar costos y facilitar el trabajo en grupo: (compartir archivos, periféricos como impresoras, acceso a internet, bases de datos, gestión de la seguridad).

2.2.2 Protocolos de red

Un protocolo de red es un conjunto de normas que regulan la comunicación (establecimiento, mantenimiento y cancelación) entre varios computadores, existen dos tipos de protocolos: protocolos de bajo nivel y protocolos de red.

Los protocolos de bajo nivel controlan la forma en que las señales se transmiten por el medio físico, mientras que los protocolos de red organizan la información para su transmisión por el medio físico a través de los protocolos de bajo nivel, entre los más importantes se cita los siguientes:

- **AppleTalk.**- protocolo de comunicación para computadores de Apple Macintosh y viene incluido en su sistema operativo.
- **TCP/IP.**- Es realmente un conjunto de protocolos, donde los más conocidos son TCP (Transmisión Control Protocolo, o protocolo de control de transmisión) e IP (Internet Protocolo, o protocolo internet).

2.2.3 Funcionamiento

En una red local podemos dividir el funcionamiento en dos partes, físico y lógico.

2.2.3.1 Funcionamiento físico

Una red local funciona de varias maneras como las que se presentan a continuación:

- **Una topología de bus** usa un solo cable backbone que debe terminarse en ambos extremos. Todos los hosts se conectan directamente a este backbone.
- **La topología de anillo** conecta un host con el siguiente y al último host con el primero. Esto crea un anillo físico de cable.
- **La topología en estrella** conecta todos los cables con un punto central de concentración.
- **Una topología en estrella extendida** conecta estrellas individuales entre sí mediante la conexión de hubs o switches. Esta topología puede extender el alcance y la cobertura de la red.
- **Una topología jerárquica** es similar a una estrella extendida. Pero en lugar de conectar los hubs o switches entre sí, el sistema se conecta con un computador que controla el tráfico de la topología.
- **La topología de malla** se implementa para proporcionar la mayor protección posible para evitar una interrupción del servicio. El uso de una topología de malla en los sistemas de control en red de una planta nuclear sería un ejemplo excelente.

- **La topología broadcast** significa que cada host envía sus datos hacia todos los demás hosts del medio de red. No existe una orden que las estaciones deban seguir para utilizar la red. Es por orden de llegada.

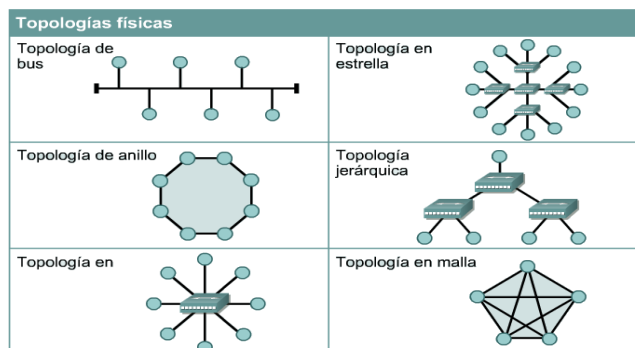


Gráfico 2.1: Topología red física (1)

2.2.3.2 Funcionamiento lógico

Aunque dos computadoras estén conectadas físicamente, hace falta establecer una conexión lógica entre ellas para que se lleve a cabo la comunicación y esto se hace asignando direcciones conocidas como direcciones IP, la topología lógica de una red es la forma en que los hosts se comunican a través del medio. Los dos tipos más comunes de topologías lógicas son broadcast y transmisión de tokens.

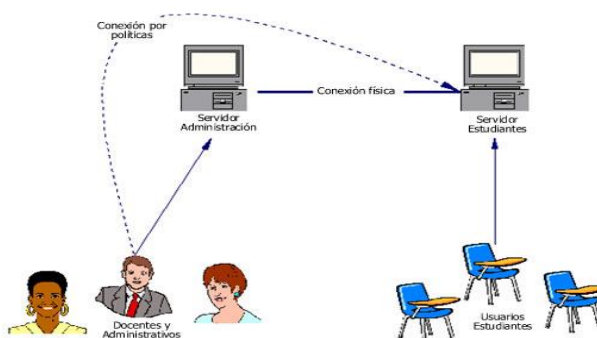


Gráfico 2.2 Funcionamiento de una red lógica (2)

(1) <http://www.monografias.com/trabajos30/redes-de-datos/redes-de-datos.shtml>

(2) <http://www.eduteka.org/RedEscolarDatos.php>

2.3 SISTEMAS DISTRIBUIDOS

Mucho tiempo ha transcurrido ya desde que en la década de los ochenta la tecnología de la información comenzara tímidamente ocupar el lugar que hoy en día posee. En aquellos días, que tal vez algunos desarrolladores veteranos recuerden con cierta nostalgia, la mayoría de los sistemas informáticos estaban constituidos por enormes sistemas centralizados que ejecutaban todas las aplicaciones dentro de sí mismos, y en el mejor de los casos los usuarios interactuaban con el sistema mediante terminales tontos, que eran simples pantallas con teclado que nos permitían enviar caracteres al sistema central y mostrar las respuestas, también basadas en caracteres que éste monstruo generaba y enviaban a través de un cable.

La aparición de los computadores personales constituyó la primera revolución, ahora los usuarios disponían de un procesador y de un sistema de almacenamiento autónomo, capaz de procesar datos e incluso de mostrar Gráficos por pantalla. Surgieron multitud de aplicaciones de escritorio que independizaron a los usuarios de los grandes sistemas y el coste del hardware se redujo a niveles tan aceptables que empresas que no hubieran podido afrontar la inversión de un sistema informático centralizado comenzaron a introducir en sus procesos de negocio la gestión informática de los datos.

Pero sin embargo los modos de ejecución de los sistemas no habían cambiado. De la ejecución de un gran sistema centralizado se pasó a la ejecución de sistemas más pequeños e independientes pero al fin y al cabo estos sistemas seguían siendo islas incomunicadas entre sí. Por otro lado, cuando los datos eran generados por un

computador central no existía manera de tratarlos localmente y toda la potencia de un computador personal se desperdiciaba por decirlo de alguna manera.

El avance de las tecnologías de redes se comenzó a trazar un horizonte donde los sistemas se comunican entre sí y en que los procesos de estos se distribuirían en diferentes computadores, cada uno con características que les permitieran aumentar la eficacia y la disponibilidad del sistema. Se comenzó a separar la lógica de los sistemas para situarlas en el nivel más conveniente y conceptos como “cliente” y “servidor” fueron cobrando cada vez más sentido. Tras algunos años de indecisión los protocolos de red se estandarizaron y hacia mediados de la década de los noventa Internet se convirtió en la primera revolución del siglo XXI, provocando no sólo un vuelco en las relaciones sociales y económicas sino también, por supuesto, un cambio completo de paradigma en la arquitectura de los sistemas distribuidos.

Los sistemas se convierten así en sistemas distribuidos, sin arriesgarme a proporcionar una definición técnica que puede encontrarse fácilmente en Internet, informalmente un sistema distribuido es aquel cuyo objetivo final se alcanza mediante la ejecución de diversos procesos independiente que por lo general se encuentran en computadores dispersos y que de una forma u otra se pasan datos entre ellos mediante protocolos de comunicación bien establecidos, de manera tal que el usuario de un sistema distribuido bien diseñado debería percibir un sistema de computación único e integrado.

2.3.1 Características

Independientemente de su arquitectura todos los sistemas distribuidos comparten ciertas características que los diferencian notablemente de los centralizados y de las aplicaciones o sistemas de escritorio. Conocer esas características y evaluar su impacto es muy importante para poder escoger el modelo más adecuado.

A continuación una lista de los aspectos primordiales que han de evaluarse a la hora de construir un sistema distribuido.

- **Concurrencia.** De igual forma que en los sistemas centralizados, los sistemas distribuidos serán utilizados por numerosos usuarios concurrentemente. Aspectos como las transacciones, los bloqueos de recursos o el uso de la CPU de los computadores a los que acceden muchos usuarios son determinantes a la hora de diseñar una arquitectura con la misma eficacia.
- **Topología de la red.** A pesar de que a día de hoy los anchos de banda cada vez son más amplios, el tráfico de red puede ser importante que condicione el tiempo de respuesta del sistema. La forma de distribuir los procesos del sistema tendrá que tomar en consideración el tipo de red que soportara el tráfico de datos.
- **Ubicación de la lógica:** Dado que un sistema distribuido intervienen varios procesos, será necesario decidir cuál de los posibles procesos físicos se sitúa cada componente lógico del sistema. En muchas ocasiones la ubicación de los

componentes lógicos impacta sobre el rendimiento sobre la reutilización del código o sobre la facilidad de programación.

- **Homogeneidad de las plataformas.** En un sistema distribuido los sistemas operativos involucrados o los lenguajes de desarrollo utilizados pueden ser un factor a tener en cuenta a la hora de decidir algunos aspectos importantes, como por ejemplo el modo de pasar datos entre procesos. La utilización de estándares puede ser muy útil a la hora de crear sistemas distribuidos que permanezcan abiertos a diversos sistemas heterogéneos, pero si las plataformas son similares es posible alcanzar mejor rendimiento sacrificando interoperabilidad.
- **Seguridad.** Un sistema distribuido mantiene procesos que de una forma u otra están a la escucha en una red, lo que aumenta la vulnerabilidad del sistema. Será necesario establecer políticas de seguridad que impidan el acceso no autorizado a los procesos. Pedir al usuario un nombre y una contraseña al iniciar el sistema es probable que no sea suficiente.

No existe una solución única para todos los problemas. A pesar de que en la actualidad se priman ciertas arquitecturas sobre otras, la realidad es que la decisión final dependerá de todos los factores anteriormente citados y de otros que a veces no se tienen presentes pero que también poseen su importancia, como el coste total del sistema o la complejidad de la solución respecto al problema planteado.

Con los sistemas distribuidos el mero análisis funcional de las características de un sistema ya no es suficiente y es necesario también considerar la distribución y

estructura de los procesos involucrados. La tarea del arquitecto o desarrollador consistirá precisamente en tomar la decisión de diseño más adecuada para resolver el problema ajustándose a los requerimientos y tomando en cuenta todos los factores implicados.

A continuación algunas de las ventajas que dan los sistemas distribuidos.

- Disponibilidad. La operación es factible en una configuración reducida cuando algunos nodos están temporalmente no disponibles.
- Configuración más flexible. una aplicación puede configurarse de distintas maneras, seleccionando el número apropiado de nodos para una instancia dada.
- Costo reducido. con frecuencia un sistema distribuido es más barato que un centralizado.

2.3.2 Modelos de seguridad

La seguridad es la capacidad del sistema para proteger datos, servicios y recursos de usuarios no autorizados.

El fin de la seguridad es garantizar la protección o estar libres de todo peligro o daño, y que en cierta manera es infalible. Como esta característica, particularizando para el caso de sistemas operativos o redes de computadores, es muy difícil de conseguir

según la mayoría de expertos, imposible, se suaviza la definición de seguridad y se pasa hablar de fiabilidad (probabilidad de que un sistema se comporte tal y como se espera) más que seguridad, por tanto, se habla de sistemas fiables en lugar de sistemas seguros.

A grandes rasgos se entiende que mantener un sistema seguro o fiable consiste básicamente en garantizar tres aspectos: confidencialidad, integridad y disponibilidad.

Algunos estudios integran la seguridad dentro de una propiedad más general de los sistemas, la confiabilidad entendida como el nivel de calidad del servicio ofrecido. Consideran la disponibilidad como un aspecto al mismo nivel que la seguridad y no como parte de ella, por lo que dividen esta última en solo las dos facetas restantes, confidencialidad e integridad. Éste trabajo no seguirá esta corriente por considerarla minoritaria.

- Confidencialidad. Nos dice que los objetos de un sistema han de ser accedidos únicamente por elementos autorizados a ellos, y que esos elementos no van a convertir esa información en disponible para otras entidades.
- Integridad. Significa que los objetos solo pueden ser modificados por elementos autorizados y de una manera controlada.

- Disponibilidad. Indica que los objetos del sistema tienen que permanecer accesibles a elementos autorizados; es el contrario de la negación de servicio.

2.3.3 Tecnologías

2.3.3.1 Middleware

El término “Middleware” se utiliza para describir una amplia variedad de software, es por esto, que es difícil dar una definición exacta y concreta de este término.

Si echamos un vistazo a la literatura informática nos encontraremos con múltiples definiciones, que van desde una capa software entre el sistema operativo (y/o red) y las aplicaciones hasta el “pegamento” capaz de conectar dos aplicaciones.

Una posible definición genérica podría ser aquel software que permite que múltiples procesos (aplicaciones) que se ejecutan en una o en varias máquinas interactúen o se comuniquen entre sí. Este software, por tanto, facilita la comunicación en un sistema distribuido.

Podemos dividir los servicios middleware en tres categorías principales: servicios de integración, servicios de aplicación y servicios de gestión. Los primeros permiten la comunicación entre aplicaciones. Los segundos facilitan el acceso a los servicios distribuidos y a la red, uno de los más importantes es el middleware de acceso a datos, el cual permite a las aplicaciones comunicarse y acceder a bases de datos ya sean éstas locales o remotas. Por último, los terceros gestionan e integran múltiples

tipos de middleware con servicios de valor añadido, como la recuperación de errores del sistema, optimización de recursos o el tratamiento de excepciones.

En este apartado nos centraremos en los servicios middleware para la integración de aplicaciones las cuales ofrecen interfaces a una gran variedad de aplicaciones.

Dentro de esta categoría se encuentran: RPC (Remote Procedure Calls) o llamadas a procedimiento remoto, Distributed Computing Environment (DCE) de la Open Software Foundation (OSF) basada en RPCs, el modelo de componentes de objetos (COM, Component Object Model) de Microsoft, las JavaBeans y las Enterprise Java Beans (EJB) de Sun Microsystems, RMI (Remote Method Invocation) o invocación remota de métodos, y los Web Services.

2.3.3.2 Web Services

Los Web Services son un mecanismo para construir sistemas distribuidos que puedan compartir e intercambiar información entre ellas utilizando como protocolo de comunicación SOAP. Paralelamente, alrededor de los web services existen una serie de protocolos y mecanismos adicionales para facilitar tareas tales como el descubrimiento de Web Services distribuidos a lo largo de la Red, o UDDI (Universal Description, Discovery and Integration of Web Services), así como una descripción de su uso WSDL (Web Services Description Languages) que sirve de ayuda a los programadores para conocer la manera adecuada de hacer uso del WS.

Es un error pensar que los Web Services constituyen la herramienta ideal para el diseño e implementación de cualquier sistema de información distribuido, de hecho, son herramientas similares a otras tecnologías asentadas hace años, tipo CORBA, RMI, DCOM etc., pero en la actualidad parecen ser el punto de referencia en estándares de comunicación de aplicaciones informáticas.

Realmente, cuando se hace uso de un Web service desde una aplicación cliente no se tiene ninguna referencia a objeto remoto alguno. Simplemente, es un intercambio de mensajes XML en una petición-respuesta clásica. En cambio, mediante tecnologías verdaderamente de objetos distribuidos se posee una referencia al objeto remoto y siempre se puede saber el estado de dicho objeto durante su vida útil, incluso pasar esta referencia a otro programa, es lo que en computación distribuida se denomina sistema con estado.

En el ámbito de los sistemas distribuidos sin necesidad de mantener estado es dónde se empieza a vislumbrar el potencial de los Web Services, dado que los tiempos y complejidad de desarrollo son menores a los de otras tecnologías de objetos distribuidos.

Los Web Services se adaptan perfectamente a esta situación ya que operaran normalmente sobre protocolos como HTTP, SMTP...y difícilmente se pueden encontrar cerrados estos puertos ya que se utilizan a su vez para otros servicios básicos como son la Web y e-mail respectivamente.

Las ventajas y desventajas de los Web Services son:

Ventajas

- Pueden trabajar en cualquier entorno que se permita el acceso de HTTP puerto 80. O sea, la mayoría de sitios. Sin necesidad de reconfigurar firewalls para activar puertos específicos.
- Se consiguen sistemas distribuidos bajamente acoplados.
- Desarrollo rápido de sistemas en comparación con otras tecnologías distribuidas

Desventajas

- Un poco más lento que otras tecnologías
- Al apoyarse en HTTP, pueden esquivar medidas de seguridad basadas en firewall cuyas reglas tratan de bloquear la comunicación entre programas.
- Existe poca información de servicios web para algunos lenguajes de programación.

2.3.4 Modelos Arquitectónicos

El modelo arquitectónico de un sistema distribuido trata sobre la colocación de sus partes y la relación entre ellas.

2.3.4.1 Arquitectura lógica

La arquitectura lógica describe la organización lógica de los componentes de un sistema distribuido.

Un componente lógico es una unidad modular, sustituible, que define las interfaces requeridas y ofrecidas en otros componentes lógicos. Los sistemas distribuidos se forman a base de componentes lógicos interrelacionados entre sí. Los conectores son mecanismos que hacen de mediadores de la comunicación, coordinación o cooperación entre componentes. A partir de interrelacionar componentes lógicos y conectores se pueden construir diferentes estilos de sistemas distribuidos:

- Arquitecturas por capas. Los componentes lógicos se organizan por capas. Un componente lógico de la capa L_i sólo puede invocar operaciones en la capa de bajo L_{i+1} . En este estilo, las invocaciones van de las capas superiores a las capas inferiores, mientras que los resultados van de las capas inferiores hacia las capas superiores.
- Arquitecturas orientadas a componentes. Cada objeto representa un componente lógico y los componentes lógicos están interconectados mediante invocaciones remotas (RPC). Este estilo y el estilo de la arquitectura por capas son los más usados en sistemas distribuidos de gran escala.

- Arquitectura orientada a los datos. En los sistemas distribuidos la localidad de los datos es un factor que afecta al rendimiento del sistema. Algunos sistemas de ficheros distribuidos, o las aplicaciones web distribuidas, deben organizar sus componentes lógicos en función de la localidad de los datos. Esta arquitectura también es utilizada cuando se quieren gestionar datos que están distribuidos de una cierta manera, por ejemplo, cuando el volumen de datos es muy grande, puede ser interesante que el componente lógico que procesa los datos esté allí donde estén los datos para no tener que enviarlas mediante la red.
- Arquitecturas orientadas a eventos. Los componentes básicamente se comunican mediante eventos. Este estilo tiene como uno de sus paradigmas más representativos lo que se conoce como paradigma de publicación-suscripción.
- En estos sistemas los componentes publican eventos que sólo son recibidos por aquellos componentes que están suscritos a ellos. La principal ventaja de estos sistemas es el desacoplamiento de los componentes lógicos.

2.3.4.2 Arquitectura Física

Es una clasificación muy extendida de los sistemas distribuidos en función de la ubicación, jerarquía o relación entre los componentes lógicos.

Ciente – Servidor

Esta arquitectura es la que estamos acostumbrados a utilizar en entornos distribuidos. Históricamente ha sido la más usada, y todavía lo es hoy en día. La web es un ejemplo de arquitectura cliente-servidor.

En el modelo cliente-servidor hay dos tipos de componentes:

- Clientes: hacen peticiones de servicio. Normalmente, los clientes inician la comunicación con el servidor.
- Servidores: proveen servicios. Normalmente, los servidores esperan recibir peticiones. Una vez han recibido una petición, la resuelven y devuelven el resultado al cliente.

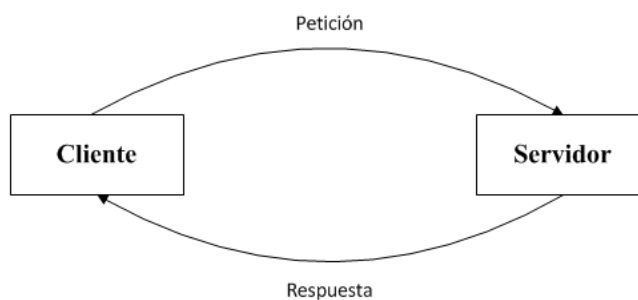


Gráfico: 2.3 Arquitectura cliente servidor

Arquitectura por niveles

Hay que resaltar el uso del término “nivel” y no el de capa porque no significa lo mismo. El término capa se utiliza para referenciar a las distintas partes en que un

sistema se divide desde un punto de vista lógico; mientras que “nivel” corresponde a la forma física en que se organiza una aplicación.

En las arquitecturas basadas en niveles, la funcionalidad está distribuida entre distintas plataformas u computadores. La interfaz reside en el computador del usuario, los servicios funcionales pueden estar en uno o más computadores, y los datos o los sistemas propietarios están en plataformas adicionales. Es muy habitual hablar de arquitecturas en capas en la bibliografía relacionada con las arquitecturas de sistemas de información. Las arquitecturas en niveles más habituales son: la de dos, tres y n niveles.

La arquitectura de dos niveles está formada por tres componentes distribuidos en dos niveles (nivel cliente y nivel servidor). Los tres componentes son los siguientes:

- Interfaz usuario del sistema
- Lógica de negocio
- Gestión de datos (servicio de datos y ficheros)

La interfaz de usuario está ubicada en el cliente. La gestión de la base de datos está ubicada en el servidor. La capacidad de procesamiento está repartida entre el cliente y el servidor. El gráfico 2.4 muestra la arquitectura de dos niveles.

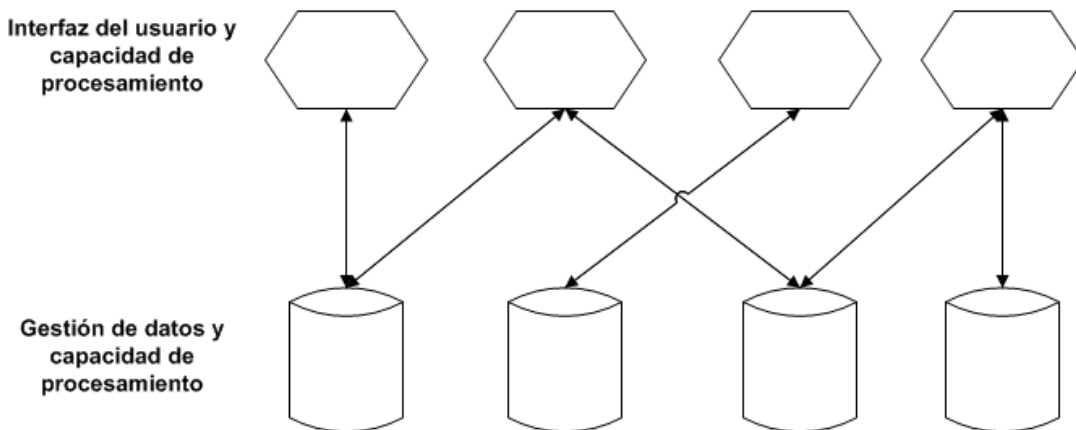


Gráfico 2.4: Arquitectura de dos niveles

La arquitectura en dos estratos mejora la usabilidad, la escalabilidad y la flexibilidad del sistema.

La **arquitectura de tres niveles** es una evolución de la arquitectura de dos estratos y ubica el tercer estrato entre la interfaz de usuario (cliente) y el gestor de datos (servidor). Este tercer estrato proporciona la capacidad de procesamiento, el gráfico 2.6 muestra esta arquitectura.

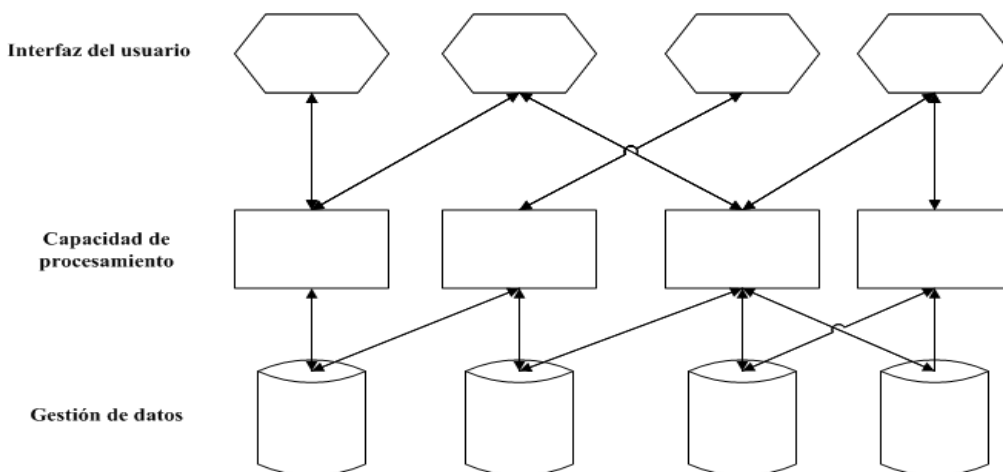


Gráfico2.5: Arquitectura de tres niveles

Comparándola con la arquitectura de dos estratos, la arquitectura de tres estratos mejora el rendimiento, la flexibilidad, la facilidad de mantenimiento de la aplicación, la reusabilidad y la escalabilidad.

La **arquitectura n niveles** también es una evolución de la arquitectura de dos niveles, aquí cabe mencionar que son muchos los expertos que consideran que un sistema distribuido es de n niveles cuando está dividido en tres niveles o más.

Un cuarto nivel puede ser el resultado de la división del nivel de la capa de negocio o procesamiento en dos, estos podrían ser un servidor web y de aplicaciones. Ver el Gráfico 2.6.

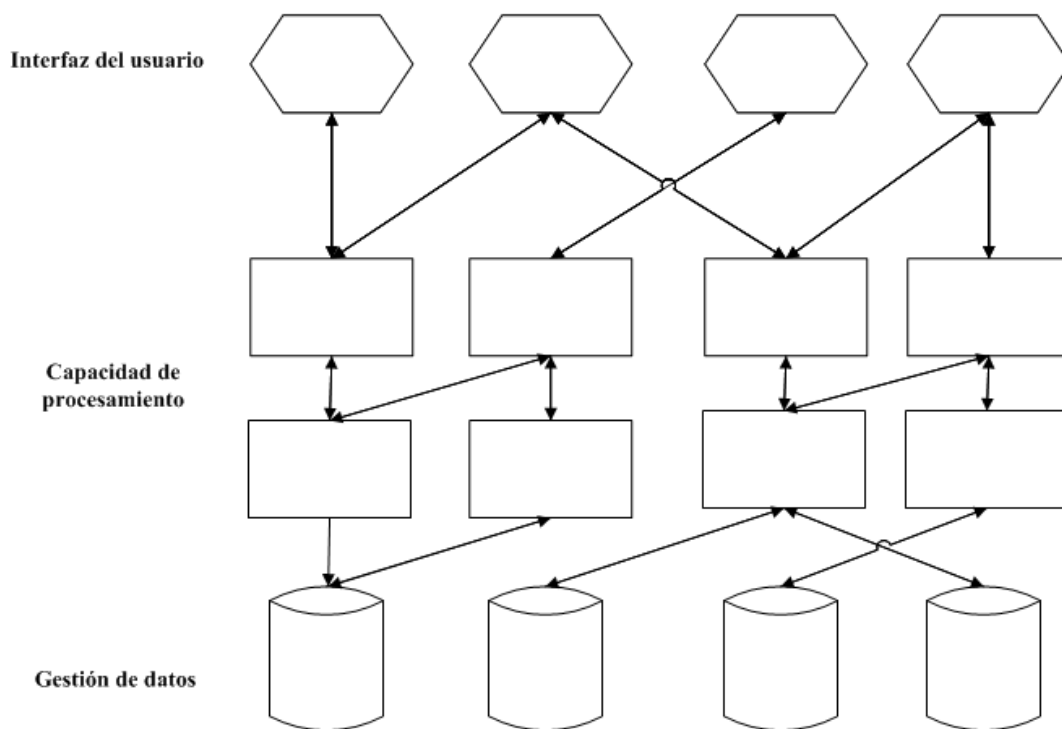


Gráfico 2.6: Arquitectura de n niveles

La arquitectura de tres y/o n niveles mejora el rendimiento, la flexibilidad, la facilidad de mantenimiento de la aplicación, la reusabilidad y la escalabilidad.

Los sistemas distribuidos ofrecen la solución más optimizada para grandes sistemas que requieren alta concurrencia o máxima reutilización de código.

Los procesos se ejecutan en máquinas dedicadas que se configura de la manera más adecuada para ofrecer los servicios que requiere cada parte del sistema.

En suma, los sistemas cliente-servidor tradicionales deben escogerse sólo en casos muy específicos en los que el rendimiento o la simplicidad son actores determinantes. Cuando necesitamos compartir funcionalidad, mejorar la distribución o sencillamente facilitar el mantenimiento, es necesario optar por un modelo mucho más abierto y más distribuido.

2.4 Base de Datos

2.4.1 Definición

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

2.4.2 Modelos de base de datos

Los tres modelos lógicos principalmente de bases de datos son el jerárquico, de redes y el relacional. Cada modelo lógico tiene ciertas ventajas de procesamiento y también ciertas ventajas de negocios.

2.4.2.1 Modelo jerárquico de datos

Una clase de modelo lógico de bases de datos que tiene una estructura arborescente. Un registro subdivide en segmentos que se interconectan en relaciones padre e hijo y muchos más. Los primeros sistemas administradores de bases de datos eran jerárquicos.

Puede representar dos tipos de relaciones entre los datos: relacionados de uno a uno y relaciones de uno a muchos.

2.4.2.2 Modelo de datos en red

Es una variación del modelo de datos jerárquico. De hecho las bases de datos pueden traducirse de jerárquicas en redes y viceversa con el objeto de optimizar la velocidad y la conveniencia del procesamiento. Mientras que las estructuras jerárquicas describen relaciones de muchos a muchos.

2.4.2.3 Modelo relacional de datos

Es el más reciente de estos modelos y es el que se utilizará para desarrollar el sistema distribuido, este modelo supera algunas de las limitaciones de los otros dos anteriores. El modelo relacional de datos representa todos los datos en la base de datos como sencillas tablas de dos dimensiones llamadas relaciones. Las tablas son semejantes a los archivos planos, pero la información en más de un archivo puede ser fácilmente extraída y combinada.

2.4.3 Creación de una base de datos

Para crear una base se deben realizar dos ejercicios de diseño: lógico y uno físico. El diseño lógico de una base de datos es un modelo abstracto de la base de datos desde una perspectiva de negocios, mientras que el diseño físico muestra como la base de datos se ordena en realidad en los dispositivos de almacenamiento de acceso directo.

El diseño físico de la base de datos es llevado a cabo por los especialistas en bases de datos, mientras que el diseño lógico requiere de una descripción detallada de las necesidades de información de los negocios actuales usuarios finales de la base. Idealmente, el diseño de la base será una parte del esfuerzo global de la planeación de datos a nivel institucional.

El diseño lógico de la base de datos describe como los elementos en la base de datos han de quedar agrupados.

El proceso de diseño identifica las relaciones entre los elementos de datos y la manera más eficiente de agruparlos para cumplir con los requerimientos de información. El proceso también identifica elementos redundantes y los agrupamientos de los elementos datos que se requieren para programas de aplicaciones específicas.

Objetivos en el uso de un sistema de gestión de base de datos:

- Oportunidad, asociado a la eficiencia y eficacia
- Disponibilidad, permitiendo la accesibilidad de datos
- Consistencias (oportunidad más disponibilidad), como calidad de datos
- Integridad, en el nivel de los datos así como el sistema

Existen cuatro productos básicos en los Sistemas de Gestión de Base de Datos para sistemas Host (propietarios). Estos son:

- Repositorio/diccionario/directorio/enciclopedia
- Desarrollador de aplicaciones (Prototipo, Lenguaje(s) de programación, basado en interpretes)
- Lenguaje de consulta

2.4.4 Seguridad en la base de datos

Cuando hablamos de seguridad, es proteger la Base de Datos contra accesos no autorizados. Se llama también privacidad e incluye aspectos de:

- Políticas de la empresa, niveles de información pública y privada
- Controles de tipo físico, acceso a las instalaciones
- Identificación de usuarios: voz, retina del ojo, etc.
- Controles de sistema operativo

En relación al SGBD (Sistema Gestor de Base de Datos), debe mantener información de los usuarios, su tipo, los accesos y operaciones permitidas a éstos.

2.4.4.1 Medidas de seguridad

- Físicas: Controlar el acceso al equipo. Tarjetas de acceso, etc.
- Personal: Acceso sólo del personal autorizado. Evitar sobornos, etc.
- SO: Seguridad a nivel de SO
- SGBD: Uso herramientas de seguridad que proporcione el SGBD. Perfiles de usuario, vistas, restricciones de uso de vistas, etc.

Un aspecto importante de la seguridad es el de impedir la pérdida de información, la cual puede producirse por diversas causas: fenómenos naturales, guerras, errores de hardware o de software, o errores humanos. La solución es una sola: mantener la información respaldada, de preferencia en un lugar lejano.

2.5 Programación

La programación es un proceso de creatividad en donde se lo escribe en un lenguaje de programación, para resolver problemas, también podría decir que es parte fundamental, vital para la elaboración de cualquier sistema informático.

2.5.1 Programación orientada a objetos

Durante años los programadores nos hemos dedicado a construir aplicaciones muy parecidas que resolvían una y otra vez los mismos problemas, a raíz de esto surge la maravillosa programación orientada a objetos, (POO)

La POO no es para nada difícil de aprender y expresarla, pero si es una manera especial de pensar, a veces subjetiva de quien la programa, de manera que la forma de hacer las cosas puede ser diferente según el programador, lo difícil no es programar orientado objetos sino programar aprovechando todas las ventajas de la POO.

2.5.1.1 Conceptos fundamentales

Entre los conceptos más importantes tenemos los siguientes:

- **Clase:** definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas.

- **Herencia:** (por ejemplo, herencia de la clase D a la clase C). Es la facilidad mediante la cual la clase D hereda en ella uno de los atributos y operaciones de C, como si esos atributos y operaciones hubiesen sido definidos por la misma D. Por lo tanto, puede usar los mismos métodos y variables públicas declaradas en C. Los componentes registrados como “privados” también se heredan, pero como no pertenecen a la clase, se mantienen escondidos al programador y sólo pueden ser accedidos a través de otros métodos públicos.
- **Método:** Algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un “mensaje”. Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un “evento” con un nuevo mensaje para otro objeto del sistema.
- **Mensaje:** una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.
- **Propiedades o atributo:** contenedor de un tipo de datos asociados a un objeto (o a una clase de objetos), que hace los datos visibles desde fuera del objeto y esto se define como sus características predeterminadas, y cuyo valor puede ser alterado por la ejecución de algún método.

2.5.1.2 Características de la programación orientada a objetos

Hay un cierto acuerdo sobre exactamente qué características de un método de programación o lenguaje le definen como “orientado a objetos”, pero hay un consenso general en que las características siguientes son las más importantes:

- **Abstracción:** Denota las características esenciales de un objeto, donde se captura su comportamiento. Cada objeto en el sistema sirve como modelo de un “agente” abstracto que puede realizar trabajo, informar y cambiar su estado, y “comunicarse” con otros objetos en el sistema sin revelar cómo se implementan estas características. Los procesos, las funciones o los métodos pueden también ser abstraídos y cuando lo están, una variedad de técnicas son requeridas para ampliar una abstracción.
- **Encapsulamiento:** Significa reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes del sistema. Algunos autores confunden este concepto con el principio de ocultación, principalmente porque se suelen emplear conjuntamente.
- **Polimorfismo:** Comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O dicho de otro

modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado.

- **Herencia:** Las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el encapsulamiento permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que volver a implementarlo.

2.5.2 Lenguaje de programación C#

Durante las últimas dos décadas, C y C++ han sido los lenguajes más ampliamente utilizados para el desarrollo de software comercial de negocios. Mientras que ambos lenguajes proporcionan al programador una vasta cantidad de control al detalle, esta flexibilidad se refleja en costo de productividad. En comparación con un lenguaje como Visual Basic, las aplicaciones C y C++ equivalentes con frecuencia tardan más en desarrollarse. Debido a la complejidad y largos ciclos de tiempo relacionados con estos lenguajes, muchos programadores C y C++ han estado buscando un lenguaje que ofrezca mejor balance entre poder y productividad.

En la actualidad, existen lenguajes que elevan la productividad sacrificando la flexibilidad que los programadores C y C++ con frecuencia requieren. Dichas soluciones restringen demasiado al desarrollador, por ejemplo, al omitir mecanismos que permiten el control de código de bajo nivel y proporcionan capacidades del menor común denominador. No interoperan fácilmente con los sistemas existentes, y no siempre se combina bien con las prácticas de programación Web actuales.

C# es la solución Microsoft para este problema. C# es un moderno lenguaje orientado a objetos que permite que los programadores desarrollen rápidamente una amplia gama de aplicaciones para la nueva plataforma Microsoft .NET. Debido a su elegante diseño orientado a objetos, C# es una fabulosa opción para crear una amplia gama de componentes, desde objetos de negocios de alto nivel hasta aplicaciones de nivel del sistema. Debido a que están soportados por la plataforma .NET, estos componentes se pueden convertir en servicios Web, permitiéndoles ser invocados a través de Internet, desde cualquier lenguaje que se ejecute en cualquier sistema operativo.

2.5.3 Linq

LINQ es una extensión a los lenguajes del .NET Framework (C# y Visual Basic .NET).

El objetivo de LINQ es el proveer a este par de lenguajes de programación la posibilidad de hacer consultas nativas, es decir que ya no lidiaremos en un futuro inmediato con SqlCommand, SqlDataReader, y demás objetos.

LINQ no solo fue creado para suprimir el problema del miss-match entre las bases de datos relacionales y los lenguajes orientados a objetos, sino que se puede realizar consultas a cualquier clase que implemente la interfaz, ejemplo IEnumerable, por ejemplo:

```
class Program
{
    static void Main(string[] args)
    {
        List<Persona> lstPersonas = new List<Persona>()
        {
            new Persona{ _nombre="Jose", _apellido="Castillo", _edad=24},
            new Persona{ _nombre="Ana", _apellido="Obando", _edad=20},
            new Persona{ _nombre="Arlyn", _apellido="Castillo", _edad=6},
            new Persona{ _nombre="Michael", _apellido="Jackson", _edad=50}
        };

        IEnumerable<Persona> query = from persona in lstPersonas
            where persona.edad > 20
            orderby persona.nombre ascending
            select persona ;

        foreach (Persona p in query)
            Console.WriteLine("{0}, {1}, {2}", p._nombre, p._apellido);
    }
}
```

```
        Console.ReadLine();  
    }  
}
```

Para lograr todo esto LINQ utiliza una característica nueva del framework.Net, los tipos anónimos son una característica del .Net Framework que permite crear tuplas o tipos de datos inexistentes automáticamente en tiempo de compilación, suena bastante complicado, pero es muy sencillo. Lo que ocurre es que si se necesitara un objeto que contenga las propiedades “Nombre” y “Apellido”, por ejemplo para obtener los nombres y apellidos de una lista de personas guardadas en una base de datos, normalmente lo que tendríamos que hacer antes de todo es crear una clase o estructura con estas dos propiedades, con los tipos anónimos esto se acabó, basta con asignar las propiedades que necesitemos a un objeto inicializador con la palabra reservada **var** y el compilador automáticamente creará esta clase con sus respectivas propiedades en tiempo de ejecución.

2.6 Modelado de sistemas

2.6.1 UML

UML (El Lenguaje Unificado de Modelado) es un lenguaje de modelado visual para sistemas. Los sistemas de software modernos normalmente requieren el soporte de herramientas. Los diagramas UML son legibles por las personas y las computadoras pueden mostrarlos fácilmente, con ello, UML proporciona una sintaxis visual que podemos utilizar para construir modelos. En el presente trabajo utilizaremos los

diagramas de casos de uso, los diagramas de clases y los diagramas de secuencias como herramientas de modelado.

2.6.2 Diagramas de casos de uso

“Ningún sistema se encuentra aislado. Cualquier sistema interesante interactúa con actores humanos o mecánicos que lo utilizan con algún objetivo y que esperan que el sistema funcione de forma predecible. Un caso de uso especifica el comportamiento de un sistema o de una parte de éste, y es una descripción de un conjunto de secuencias de acciones, incluyendo variantes, que ejecuta un sistema para producir un resultado observable de valor para un actor.

Los casos de uso se emplean para capturar el comportamiento deseado del sistema en desarrollo, sin tener que especificar cómo se implementa ese comportamiento. Los casos de uso proporcionan un medio para que los desarrolladores, los usuarios finales del sistema y los expertos del dominio lleguen a una comprensión común del sistema.” (Booch et al., 2006, p. 243).

2.6.3 Diagramas de secuencia

Un diagrama de interacción muestra una interacción, que consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.

Un diagrama de secuencia es un diagrama de interacción que resalta la ordenación temporal de los mensajes.

“Los diagramas de secuencia son diagramas UML que se utilizan para modelar los aspectos dinámicos de los sistemas. De esta manera pueden utilizarse para modelar un flujo de control particular de un caso de uso.” (Booch et al., 1999, p. 211).

CAPITULO III

3 DESARROLLO DEL SISTEMA DISTRIBUIDO

3.1 Desarrollo del análisis

Analizando las diferentes causas que permiten centrarse en el problema planteado, y que algunos de ellos tienen la inmediata necesidad de involucrar a los docentes en el uso de las nuevas tecnologías de información y, tomando en cuenta la acumulación de conocimientos que los estudiantes traen consigo en la utilización de sistemas computacionales; se realizó un acercamiento previo en base a entrevistas individuales a docentes y alumnos, para recoger criterios de los diferentes estamentos que se involucrarían en la investigación del presente proyecto, es así que la encuesta empleada está clasificada de la siguiente manera:

Encuesta dirigida a los docentes que laboran en la Escuela de Ingeniería de Sistemas de la PUCESA, que fue estructurada con preguntas cerradas y abiertas de manera que permitió alcanzar los objetivos de la investigación

Encuesta dirigida a los estudiantes, estructurada con preguntas cerradas y abiertas con la finalidad de identificar cómo se viene evaluando el proceso de enseñanza aprendizaje en la Escuela de Ingeniería de Sistemas en la PUCESA. Ver Anexo 1.

Para la aplicación de las encuestas primeramente fue necesario definir la muestra bajo la cual se trabajaría, por lo que se procedió de una población estudiantil de 250,

fueron 45 estudiantes que en forma aleatoria fueron seleccionados para responder la encuesta

En cuanto a las encuestas del personal docente, por ser un universo pequeño se decidió encuestar a 10 docentes. Ver Anexo 2.

ENCUESTADOS	MUESTRA
Docentes	10
Estudiantes	45
TOTAL	55

Elaborado por: José Castillo.

3.1.1 Requerimientos del Sistema

Arlow (2006) afirma: La ingeniería de requisitos es un término utilizado para describir las actividades implicadas en la obtención, documentación y mantenimiento de un conjunto de requerimientos para un sistema de software. (p. 78) Se trata de descubrir lo que los diferentes stakeholders¹ necesitan que el sistema realice.

Podemos definir un requisito como una especificación de lo que se debería implementar, existen básicamente dos tipos de requisitos:

Funcionales: el comportamiento que el sistema debe ofrecer.

No funcionales: una propiedad específica o una restricción del sistema.

Los requisitos o requerimientos son la base de todo sistema. Específicamente, son sentencias que el sistema debe brindar. (Arlow, 2006, p.79)

3.1.2 Requerimientos funcionales

A continuación se detallan los requerimientos funcionales que deberá cubrir el sistema agrupados por módulos funcionales:

Módulo de Escuelas	
Código	Descripción
ESC01	El sistema debe registrar o actualizar los datos de la escuela: código (autogenerado), nombre, director, secretaria, descripción.
ESC02	El sistema debe eliminar escuela que no tengan cátedras o alumnos asociados.
ESC03	El sistema debe buscar escuelas en función de filtros: nombre y director.

Tabla 3.1: Requerimientos funcionales (Módulo de escuelas)

Módulo de Cátedras	
Código	Descripción
CAT01	El sistema debe registrar o actualizar los datos de la escuela: código, nombre, contenido, descripción, indicador a la escuela que pertenece.
CAT02	El sistema debe eliminar cátedras que no tengan exámenes o preguntas asociadas.
CAT03	El sistema debe buscar cátedras en función de filtros: código y nombre.

Tabla 3.2: Requerimientos funcionales (Módulo de cátedras)

Módulo de Exámenes	
Código	Descripción
EXA01	El sistema debe registrar o actualizar los exámenes: código, título, docente, instrucciones, tipo de puntuación, control de tiempo.
EXA02	El sistema debe eliminar exámenes que no tengan preguntas asociadas.
EXA03	El sistema debe buscar exámenes en función de filtros: código y título.

Tabla 3.3: Requerimientos funcionales (Módulo de exámenes)

Módulo de Preguntas	
Código	Descripción
PRG01	El sistema debe registrar o actualizar los datos de las preguntas: tipo de pregunta (decisión, enlace, practica), enunciado, ítems de respuestas, puntaje.
PRG02	El sistema debe eliminar preguntas que no tengan exámenes asociados.
PRG03	El sistema debe buscar preguntas en función de filtros: enunciado, tipo de pregunta, puntaje.

Tabla 3.4: Requerimientos funcionales (Módulo de preguntas)

Módulo de Docentes	
Código	Descripción
DOC01	El sistema debe registrar o actualizar los datos de las preguntas: código (autogenerado), nombre, apellido, título, cedula y teléfono.
DOC02	El sistema puede eliminar docentes.
DOC03	El sistema debe buscar docentes en función de filtros: nombre y apellido.

Tabla 3.5: Requerimientos funcionales (Módulo de docentes)

Módulo de Alumnos	
Código	Descripción
ALU01	El sistema debe registrar o actualizar los datos del alumno: código (autogenerado), nombre, apellido, título, cedula, teléfono, indicador a la escuela que pertenece.
ALU02	El sistema debe eliminar alumnos que no tengan exámenes asociados.
ALU03	El sistema debe buscar alumnos en función de filtros: nombre y apellido.

Tabla 3.6: Requerimientos funcionales (Módulo de alumnos)

Módulo de Seguridades	
Código	Descripción
SGU01	El sistema debe registrar o actualizar los datos de Usuario: tipo de usuario, usuario, clave, nombres y apellidos, indicador del rol asignado.
SGU 02	El sistema debe puede eliminar usuarios
SGU 03	El sistema debe buscar usuarios en función de filtros: nombre y apellido.
SGU 04	El sistema debe registrar los datos del rol: nombre, ítems de las pantallas del sistema.
SGU 05	El sistema debe puede eliminar roles
SGU 06	El sistema debe buscar roles en función de filtros: nombre.

Tabla 3.7: Requerimientos funcionales (Módulo de seguridades)

Módulo de Reportes	
Código	Descripción
RPT01	El sistema debe generar un reporte en función de filtros: escuela, cátedra, examen y alumno.
RPT 02	El sistema en el reporte debe mostrar los siguientes campos: pregunta, respuestas, valor de la pregunta, puntaje conseguido y nota del examen.

Tabla 3.8: Requerimientos funcionales (Módulo de reportes)

Módulo Examinador	
Código	Descripción
EXD01	El sistema debe mostrar todos los exámenes disponibles para el alumno.
EXD02	El sistema debe permitir navegar hacia adelante por todas las preguntas que contenga el examen seleccionado

Tabla 3.9: Requerimientos funcionales (Módulo examinador)

3.1.3 Requerimientos no funcionales

A continuación se detallan los requerimientos no funcionales que tendrá el sistema:

- El sistema debe tener una interfaz de uso intuitiva y sencilla.
- El sistema será desarrollado en el lenguaje orientado a objetos.
- El acceso al sistema será controlado con nombres de usuario y contraseñas.
- Se definirá una política de permisos para acceder a las diversas funcionalidades del sistema.

3.1.4 Descripción de los Casos de Uso

Antes de realizar los diagramas y dibujos del sistema era necesario que se le asigne un nombre, después de unos días en búsqueda del nombre perfecto, la decisión fue “CRONOS”, ¿Por qué?, porque CRONOS es el Dios del tiempo en la mitología

griega y es justamente el tiempo lo que estuvo siempre presente en la realización de este proyecto.

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un usuario y el sistema, cuando el usuario emplea el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad, a continuación se presenta todos los casos de usos encontrados durante el análisis del desarrollo de CRONOS.

Caso Uso:
Autenticar Usuario
Código:
CU001
Actores
Docente
Precondiciones
Descripción
El Docente debe ingresar sus datos de usuario y clave. El sistema valida los datos del docente El sistema muestra la pantalla principal con el menú de opciones
Observaciones
Punto 2: Si los datos no son correctos el sistema mostrará un mensaje de no autorizado.

Tabla 3.10: Caso de uso (Autenticar Usuario)

Caso Uso:
Administrar escuela
Código:
CU002
Actores
Docente
Precondiciones
El usuario debe haberse autenticado en el sistema
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona escuelas 2. EL sistema verifica si el docente tiene permisos para escuelas 3. El sistema muestra la pantalla principal de escuelas, con sus respectivas opciones (Crear, Modificar, Eliminar, Ver).
Observaciones
Punto 2: Si el docente no tiene permisos para Escuelas, el sistema mostrar un mensaje indicando esto.

Tabla 3.11: Caso de uso (Administrar escuela)

Caso Uso:
Crear escuela
Código:
CU003
Actores:
Docente
Precondiciones
El docente debe estar situado en Administrar escuelas
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona la opción Crear 2. El sistema muestra los campos necesarios para crear una escuela 3. El docente ingresa la información necesario para crear una escuela 4. El docente presiona el botón grabar 5. El sistema verifica que exista información en los campos obligatorios 6. El sistema guarda la información e indica que la operación se ha realizado con éxito..
Observaciones
Punto 5: Si los campos obligatorios están vacíos, el sistema señalará que el campo es requerido

Tabla 3.12: Caso de uso (Crear escuela)

Caso Uso:
Modificar escuela
Código:
CU004
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar escuelas
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona la escuela a modificar 2. El docente selecciona la opción modificar. 3. El sistema muestra la información de la escuela seleccionada. 4. El docente realizar los cambios en la información de la escuela 5. El docente presiona el botón grabar 6. El sistema verifica que exista información en los campos obligatorios 7. El sistema guarda la información e indica que la operación se ha realizado con éxito.
Observaciones
Punto 6: Si los campos obligatorios están vacíos, el sistema señalará que el campo es requerido

Tabla 3.13: Caso de uso (Modificar escuela)

Caso Uso:
Eliminar escuela
Código:
CU005
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar escuelas
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona la escuela a eliminar 2. El docente selecciona la opción eliminar. 3. El sistema muestra la información de la escuela seleccionada. 4. El docente presiona el botón grabar 5. El sistema elimina la información e indica que la operación se ha realizado con éxito.
Observaciones

Tabla 3.14: Caso de uso (Eliminar escuela)

Caso Uso:
Ver escuela
Código:
CU006
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar escuelas
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona la Escuela a eliminar 2. El docente selecciona la opción visualizar. 3. El sistema muestra la información de la escuela seleccionada. 4. El sistema bloquea los campos de la cátedra impidiendo su modificación e eliminación
Observaciones

Tabla 3.15: Caso de uso (Ver escuela)

Caso Uso:
Administrar cátedra
Código:
CU007
Actores
Docente
Precondiciones
El docente debe haberse autenticado en el sistema
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona la cátedra 2. EL sistema verifica si el docente tiene permisos para la cátedra 3. El sistema muestra la pantalla principal de cátedras, con sus respectivas opciones (Crear, Modificar, Eliminar, Ver).
Observaciones
Punto 2: De no tener permisos el sistema mostrar un mensaje indicando esto.

Tabla 3.16: Caso de uso (Administrar Cátedra)

Caso Uso:
Crear cátedra
Código:
CU008
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar cátedras
Descripción
<ol style="list-style-type: none"> 1. El sistema muestra los campos necesarios para crear una cátedra 2. El sistema lista las escuelas existentes para que la cátedra sea asignado a una de ellas. 3. El docente ingresa la información necesario para crear una cátedra 4. El docente presiona el botón grabar 5. El sistema verifica que exista información en los campos obligatorios 6. El sistema guarda la información e indica que la operación se ha realizado con éxito.
Observaciones
Punto 2: Si no existe escuelas en el sistema, éste no permitirá crear la cátedra
Punto 5: Si los campos obligatorios están vacíos, el sistema señalará que el campo es requerido

Tabla 3.17: Caso de uso (Crear Cátedra)

Caso Uso:
Modificar cátedra
Código:
CU009
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar cátedras
Descripción
<ol style="list-style-type: none"> 2. El docente selecciona la cátedra a modificar 3. El docente selecciona la opción modificar. 4. El sistema muestra la información de la cátedra seleccionada. 5. El docente realizar los cambios en la información de la cátedra 6. El docente presiona el botón grabar 7. El sistema verifica que exista información en los campos obligatorios 8. El sistema guarda la información e indica que la operación se ha realizado con éxito.
Observaciones
Punto 7: Si los campos obligatorios están vacíos, el sistema señalará que el campo es requerido

Tabla 3.18: Caso de uso (Modificar Cátedra)

Caso Uso:
Eliminar Cátedra
Código:
CU010
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar cátedras
Descripción
<ol style="list-style-type: none"> 2. El docente selecciona la cátedra a eliminar 3. El docente selecciona la opción eliminar. 4. El sistema muestra la información de la cátedra seleccionada. 5. El docente presiona el botón grabar 6. El sistema elimina la información e indica que la operación se ha realizado con éxito.
Observaciones

Tabla 3.19: Caso de uso (Eliminar Cátedra)

Caso Uso:
Ver cátedra
Código:
CU011
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar cátedras
Descripción
<ol style="list-style-type: none"> 2. El docente selecciona la cátedra a visualizar 3. El docente selecciona la opción Ver. 4. El sistema muestra la información de la cátedra seleccionada. 5. El sistema bloquea los campos de la cátedra impidiendo su modificación e eliminación
Observaciones

Tabla 3.20: Caso de uso (Ver Cátedra)

Caso Uso:
Administrar alumnos
Código:
CU012
Actores
Docente
Precondiciones
El usuario debe estar situado en Administrar alumnos
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona la alumno 2. EL sistema verifica si el docente tiene permisos para administrar alumnos 3. El sistema muestra la pantalla principal de alumnos, con sus respectivas opciones (Crear, Modificar, Eliminar, Ver).
Observaciones
Punto 2: De no tener permisos el sistema mostrar un mensaje indicando esto.

Tabla 3.21: Caso de uso (Administrar Alumno)

Caso Uso:
Crear alumno
Código:
CU013
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar alumnos
Descripción
<ol style="list-style-type: none"> 1. El sistema muestra los campos necesarios para crear un alumno 2. El sistema lista las escuelas existentes para que el alumno sea asignado a una de ellas. 3. El docente ingresa la información necesario para crear una alumno 4. El docente presiona el botón grabar 5. El sistema verifica que exista información en los campos obligatorios 6. El sistema guarda la información e indica que la operación se ha realizado con éxito.
Observaciones
Punto 2: Si no existe escuelas en el sistema, éste no permitirá crear la cátedra
Punto 5: Si los campos obligatorios están vacíos, el sistema señalará que el campo es requerido

Tabla 3.22: Caso de uso (Crear Alumno)

Caso Uso:
Modificar alumno
Código:
CU014
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar alumnos
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona el alumno a modificar 2. El docente selecciona la opción modificar. 3. El sistema muestra la información del alumno seleccionada. 4. El docente realizar los cambios en la información del alumno 5. El docente presiona el botón grabar 6. El sistema verifica que exista información en los campos obligatorios 7. El sistema guarda la información e indica que la operación se ha realizado con éxito.
Observaciones
Punto 6: Si los campos obligatorios están vacíos, el sistema señalará que el campo es requerido

Tabla 3.23: Caso de uso (Modificar Alumno)

Caso Uso:
Eliminar alumno
Código:
CU015
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar alumnos
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona el alumno a eliminar 2. El docente selecciona la opción eliminar. 3. El sistema muestra la información del alumno seleccionada. 4. El docente presiona el botón grabar 5. El sistema elimina la información e indica que la operación se ha realizado con éxito.
Observaciones

Tabla 3.24: Caso de uso (Eliminar Alumno)

Caso Uso:
Ver alumno
Código:
CU016
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar alumnos
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona el alumno a visualizar 2. El docente selecciona la opción Ver. 3. El sistema muestra la información del alumno seleccionada. 4. El sistema bloquea los campos del alumno impidiendo su modificación e eliminación
Observaciones

Tabla 3.25: Caso de uso (Ver Alumno)

Caso Uso:
Administrar Docente
Código:
CU017
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar alumnos
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona el docente 2. EL sistema verifica si el docente tiene permisos para el docente 3. El sistema muestra la pantalla principal de Docentes, con sus respectivas opciones (Crear, Modificar, Eliminar, Ver).
Observaciones
Punto 2: De no tener permisos el sistema mostrar un mensaje indicando esto.

Tabla 3.26: Caso de uso (Administrar Docente)

Caso Uso:
Crear Docente
Código:
CU018
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar alumnos
Descripción
<ol style="list-style-type: none"> 1. El sistema muestra los campos necesarios para crear un docente 2. El docente ingresa la información necesario para crear un docente 3. El docente presiona el botón grabar 4. El sistema verifica que exista información en los campos obligatorios 5. El sistema guarda la información e indica que la operación se ha realizado con éxito.
Observaciones
Punto 4: Si los campos obligatorios están vacíos, el sistema señalará que el campo es requerido

Tabla 3.27: Caso de uso (Crear Docente)

Caso Uso:
Modificar docente
Código:
CU019
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar docentes
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona el docente a modificar 2. El docente selecciona la opción modificar. 3. El sistema muestra la información del docente seleccionado. 4. El docente realizar los cambios en la información del docente 5. El docente presiona el botón grabar 6. El sistema verifica que exista información en los campos obligatorios 7. El sistema guarda la información e indica que la operación se ha realizado con éxito.
Observaciones
Punto 6: Si los campos obligatorios están vacíos, el sistema señalará que el campo es requerido

Tabla 3.28: Caso de uso (Modificar Docente)

Caso Uso:
Eliminar docente
Código:
CU020
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar docentes
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona el docente a eliminar 2. El docente selecciona la opción eliminar. 3. El sistema muestra la información del docente seleccionado. 4. El docente presiona el botón grabar 5. El sistema elimina la información e indica que la operación se ha realizado con éxito.
Observaciones

Tabla 3.29: Caso de uso (Eliminar Docente)

Caso Uso: Ver docente
Código: CU021
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar docentes
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona el docente a visualizar 2. El docente selecciona la opción Ver. 3. El sistema muestra la información del docente seleccionado. 4. El sistema bloquea los campos del docente impidiendo su modificación e eliminación
Observaciones

Tabla 3.30: Caso de uso (Ver Docente)

Caso Uso:
Administrar preguntas
Código:
CU022
Actores
Docente
Precondiciones
El usuario debe haberse autenticado en el sistema
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona la Preguntas 2. EL sistema verifica si el docente tiene permisos para administrar preguntas 3. El sistema muestra la pantalla principal de preguntas, con sus respectivas opciones (Crear, Modificar, Eliminar, Ver).
Observaciones
Punto 2: De no tener permisos el sistema mostrar un mensaje indicando esto.

Tabla 3.31: Caso de uso (Administrar Preguntas)

Caso Uso:
Crear Pregunta
Código:
CU023
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar preguntas
Descripción
<ol style="list-style-type: none"> 1. El sistema muestra los campos necesarios para crear una pregunta 2. El docente ingresa la información necesario para crear una pregunta 3. El docente presiona el botón grabar 4. El sistema verifica que exista información en los campos obligatorios 5. El sistema guarda la información de la Pregunta dejando el historial y muestra un mensaje indicando que la operación se ha realizado con éxito.
Observaciones
Punto 4: Si los campos obligatorios están vacíos, el sistema señalara que el campo es requerido

Tabla 3.32: Caso de uso (Crear Pregunta)

Caso Uso:
Modificar pregunta
Código:
CU024
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar preguntas
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona el pregunta a modificar 2. El docente selecciona la opción modificar. 3. El sistema muestra la información de la pregunta seleccionado. 4. El docente realizar los cambios en la información de la pregunta 5. El docente presiona el botón grabar 6. El sistema verifica que exista información en los campos obligatorios 7. El sistema guarda la información e indica que la operación se ha realizado con éxito.
Observaciones
Punto 6: Si los campos obligatorios están vacíos, el sistema señalará que el campo es requerido

Tabla 3.33: Caso de uso (Modificar Pregunta)

Caso Uso:
Eliminar pregunta
Código:
CU025
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar docentes
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona la pregunta a eliminar 2. El docente selecciona la opción eliminar. 3. El sistema muestra la información de la pregunta seleccionado. 4. El docente presiona el botón grabar 5. El sistema elimina la información e indica que la operación se ha realizado con éxito.
Observaciones

Tabla 3.34: Caso de uso (Eliminar Pregunta)

Caso Uso:
Ver pregunta
Código:
CU026
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar preguntas
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona la pregunta a visualizar 2. El docente selecciona la opción Ver. 3. El sistema muestra la información de la pregunta seleccionada. 4. El sistema bloquea los campos de la pregunta impidiendo su modificación y eliminación
Observaciones

Tabla 3.35: Caso de uso (Ver Pregunta)

Caso Uso:
Administrar exámenes
Código:
CU027
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar docentes
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona la opción Exámenes 2. EL sistema verifica si el docente tiene permisos para administrar exámenes 3. El sistema muestra la pantalla principal de Exámenes, con sus respectivas opciones (Crear, Modificar, Eliminar, Ver).
Observaciones
Punto 2: De no tener permisos el sistema mostrar un mensaje indicando esto.

Tabla 3.36: Caso de uso (Administrar Exámenes)

Caso Uso:
Crear examen
Código:
CU028
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar exámenes
Descripción
<ol style="list-style-type: none"> 1. El sistema muestra los campos necesarios para crear un examen 2. El docente ingresa la información necesario para crear un examen 3. El docente presiona el botón grabar 4. El sistema verifica que exista información en los campos obligatorios 5. El sistema guarda la información e indica que la operación se ha realizado con éxito.
Observaciones
Punto 4: Si los campos obligatorios están vacíos, el sistema señalará que el campo es requerido

Tabla 3.37: Caso de uso (Crear Examen)

Caso Uso: Modificar examen
Código: CU029
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar exámenes
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona el examen a modificar 2. El docente selecciona la opción modificar. 3. El sistema muestra la información del examen seleccionado. 4. El docente realizar los cambios en la información del examen 5. El docente presiona el botón grabar 6. El sistema verifica que exista información en los campos obligatorios 7. El sistema guarda la información e indica que la operación se ha realizado con éxito.
Observaciones
Punto 6: Si los campos obligatorios están vacíos, el sistema señalará que el campo es requerido

Tabla 3.38: Caso de uso (Modificar Examen)

Caso Uso:
Eliminar examen
Código:
CU030
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar exámenes
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona la examen a eliminar 2. El docente selecciona la opción eliminar. 3. El sistema muestra la información del examen seleccionado. 4. El docente presiona el botón grabar 5. El sistema elimina la información e indica que la operación se ha realizado con éxito.
Observaciones

Tabla 3.39: Caso de uso (Eliminar Examen)

Caso Uso:
Ver examen
Código:
CU031
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar exámenes
Descripción
<ol style="list-style-type: none"> 1. El docente selecciona el examen a visualizar 2. El docente selecciona la opción Ver. 3. El sistema muestra la información del examen seleccionada. 4. El sistema bloquea lo campos del Examen impidiendo su modificación y eliminación
Observaciones

Tabla 3.40: Caso de uso (Ver Examen)

Caso Uso:
Agregar pregunta a un examen
Código:
CU032
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar exámenes Debe existir un examen
Descripción
<ol style="list-style-type: none"> 1. El sistema permite buscar preguntas agrupadas por la cátedra del examen 2. El docente selecciona la o las preguntas 3. El docente presiona el botón Agregar preguntas. 4. El sistema muestra un mensaje de operación realizada con éxito.
Observaciones

Tabla 3.41: Caso de uso (Agregar pregunta a un examen)

Caso Uso:
Eliminar Pregunta de un examen
Código:
CU033
Actores
Docente
Precondiciones
El docente debe estar situado en Administrar exámenes Debe existir un examen creado
Descripción
<ol style="list-style-type: none"> 1. El sistema muestra las preguntas que contiene el examen seleccionado 2. El docente selecciona la pregunta y la elimina visualmente 3. El docente presiona el botón Grabar 4. El sistema muestra un mensaje de operación realizada con éxito.
Observaciones

Tabla 3.42: Caso de uso (Eliminar Pregunta de un examen)

Caso Uso:
Reporte examen
Código:
CU034
Actores
Docente
Precondiciones
El usuario debe estar situado en Administrar docentes
Descripción
<ol style="list-style-type: none"> 1. El sistema muestra las escuelas, cátedras, exámenes y alumnos del sistema 2. El Docente selecciona la escuela, cátedra, examen y alumno 3. El Docente pulsa el botón Ver Examen
Observaciones

Tabla 3.43: Caso de uso (Reporte examen)

Caso Uso:
Tomar examen
Código:
CU035
Actores
Alumno Docente
Precondiciones
El alumno debe haberse autenticado en el sistema
Descripción
<ol style="list-style-type: none"> 1. El sistema muestra todos los exámenes disponibles para el alumno 2. El alumno selecciona el examen 3. El docente provee al Alumno el código del examen 4. El alumno ingresa el código del examen 5. El alumno empieza el examen 6. El alumno navega el examen pregunta por pregunta 7. El alumno termina el examen.
Observaciones
Punto 4: Si el código del examen es incorrecto, el sistema mostrar un mensaje avisando esto.
Punto 6: Si el examen es temporizado, el sistema bloquea el examen, dándolo por terminado.

Tabla 3.44: Caso de uso (Tomar examen)

Caso Uso:
Ver resultado del examen
Código: CU035
Actores
Alumno
Precondiciones
El alumno debe haberse autenticado en el sistema
Descripción
<ol style="list-style-type: none"> 1. El Sistema muestra todos los exámenes disponibles para el alumno. 2. El Alumno selecciona el examen. 3. El Alumno pulsa el botón ver examen. 4. El Sistema muestra el resultado del examen.
Observaciones

Tabla 3.45: Caso de uso (Ver resultado del examen)

3.1.5 Diagramas de casos de uso

A continuación se muestra los diagramas de casos de uso que tendrá el sistema, éstos han sido agrupados por los módulos encontrados en los requerimientos funcionales

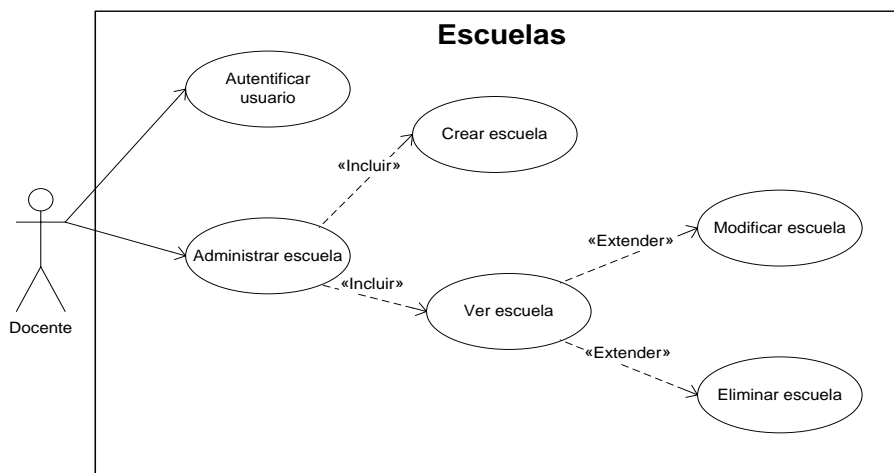


Gráfico 3.1: Diagrama caso de uso escuelas

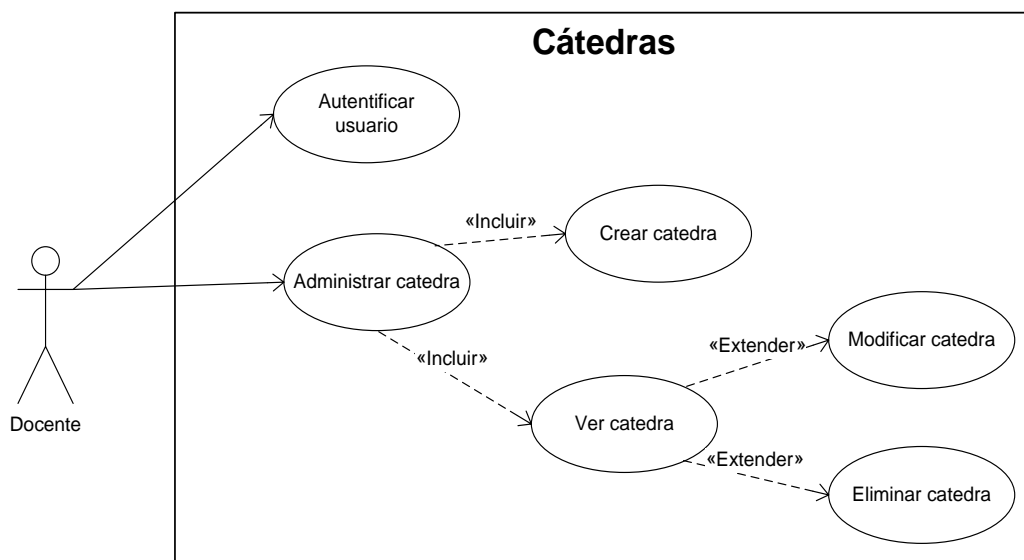


Gráfico 3.2.: Diagrama caso de uso cátedras

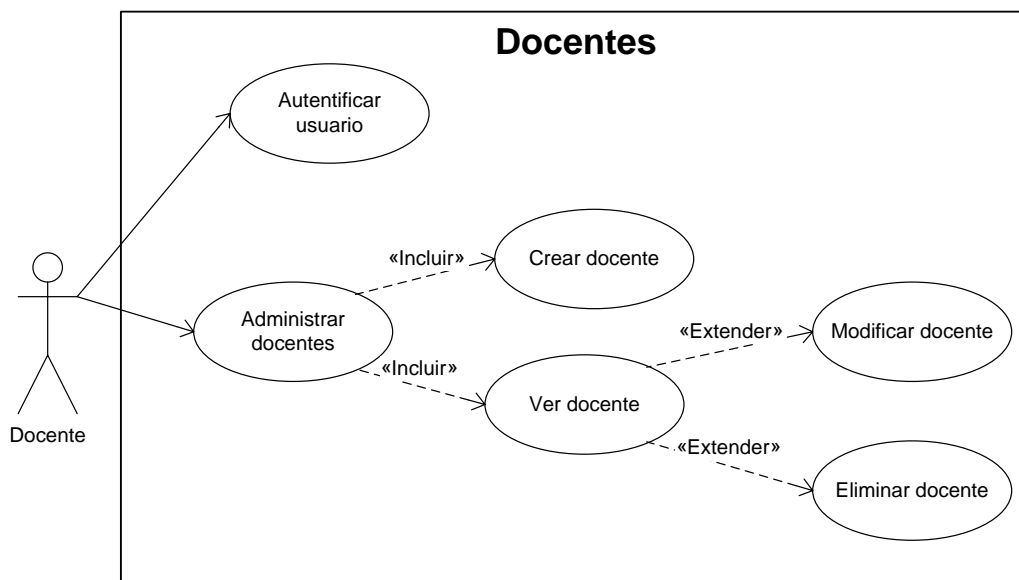


Gráfico 3.3: Diagrama caso de uso docentes

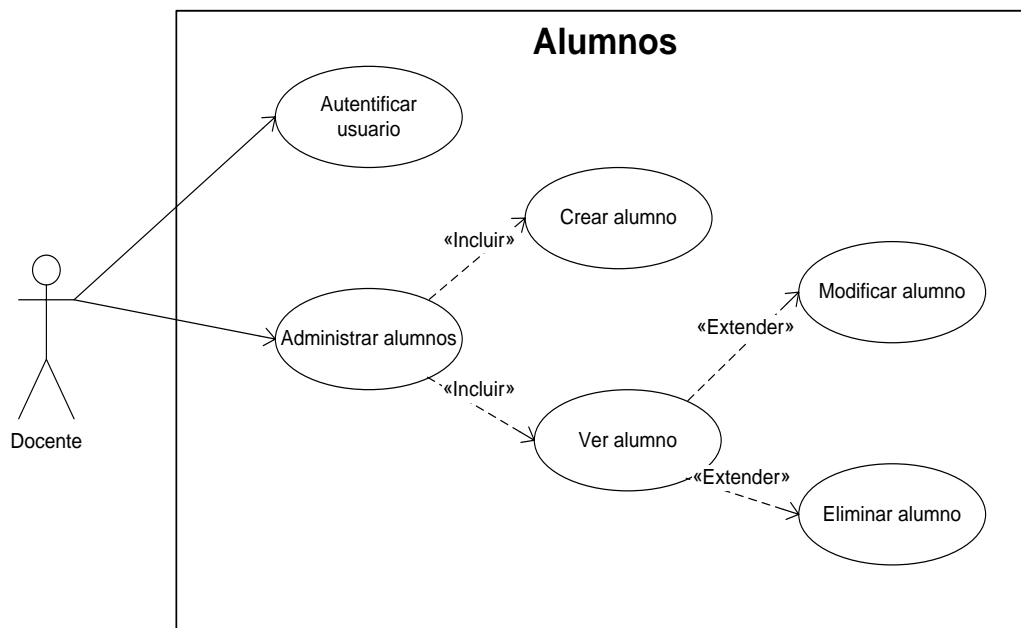


Gráfico 3.4: Diagrama caso de uso alumnos

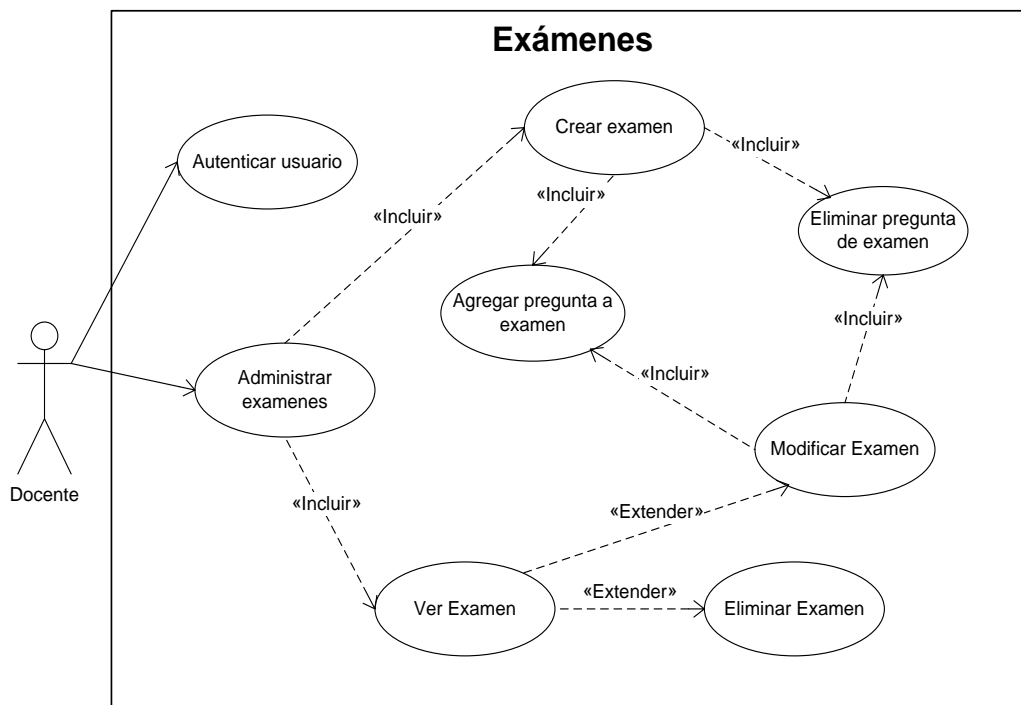


Gráfico 3.5: Diagrama caso de uso exámenes

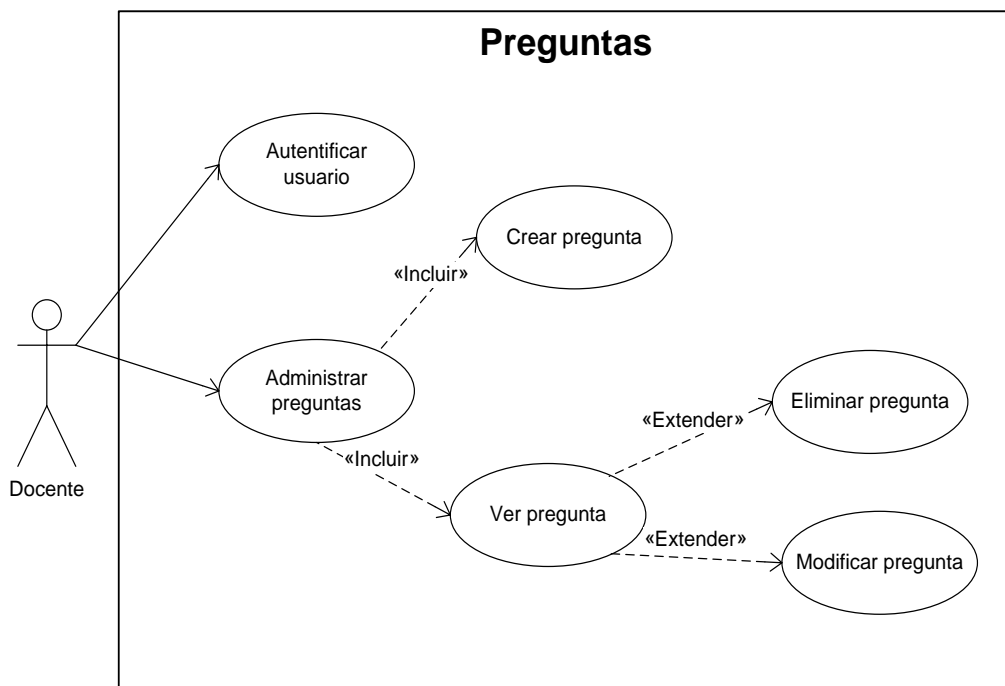


Gráfico 3.6: Diagrama caso de uso bco. preguntas

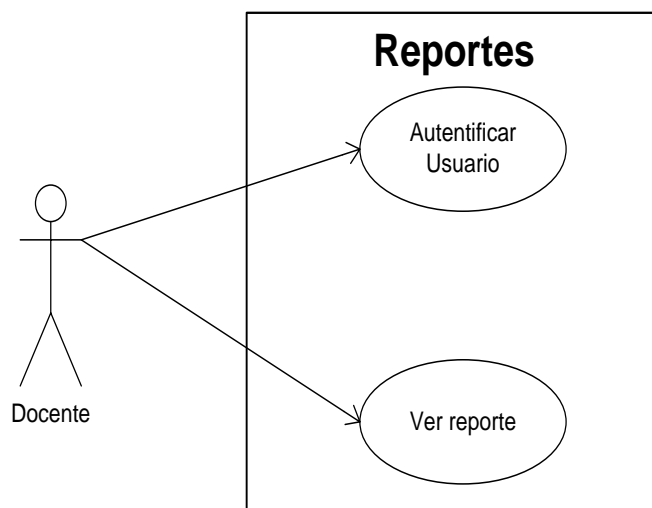


Gráfico 3.7: Diagrama caso de uso reportes

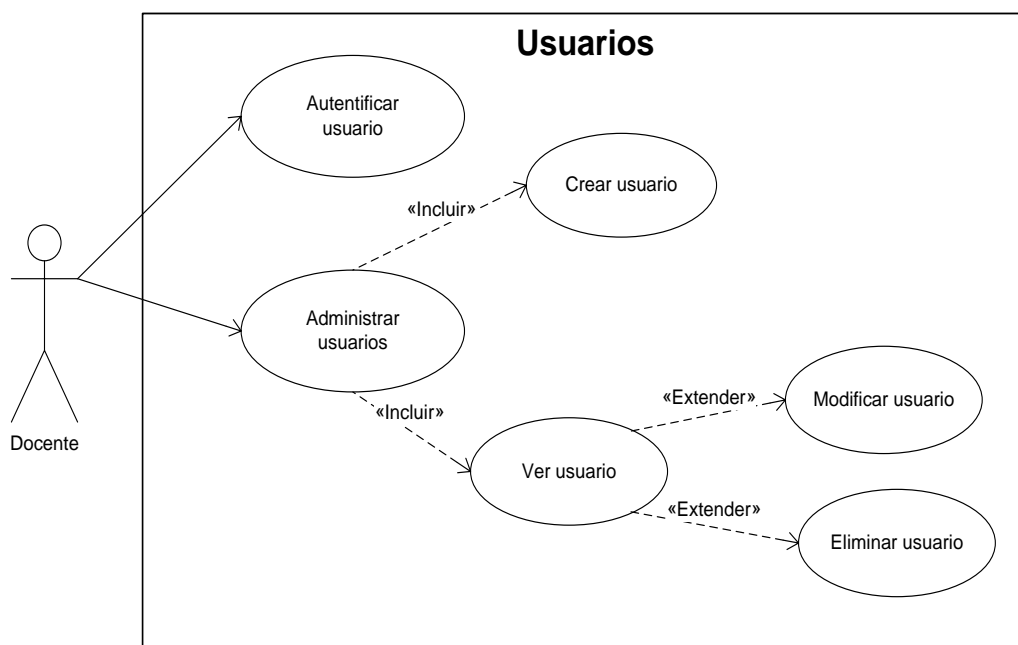


Gráfico 3.8: Diagrama caso de uso usuarios

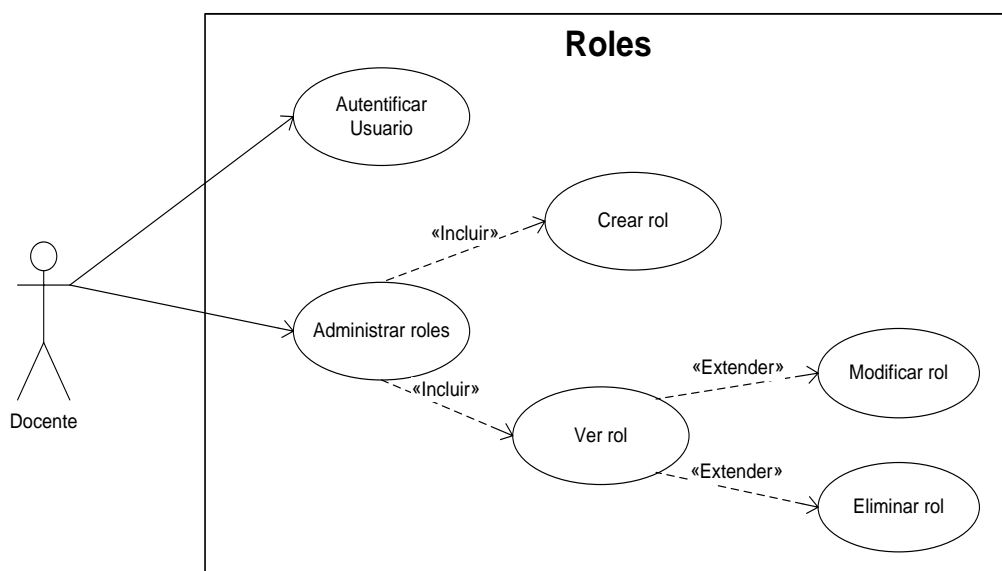


Gráfico 3.9: Diagrama caso de uso roles

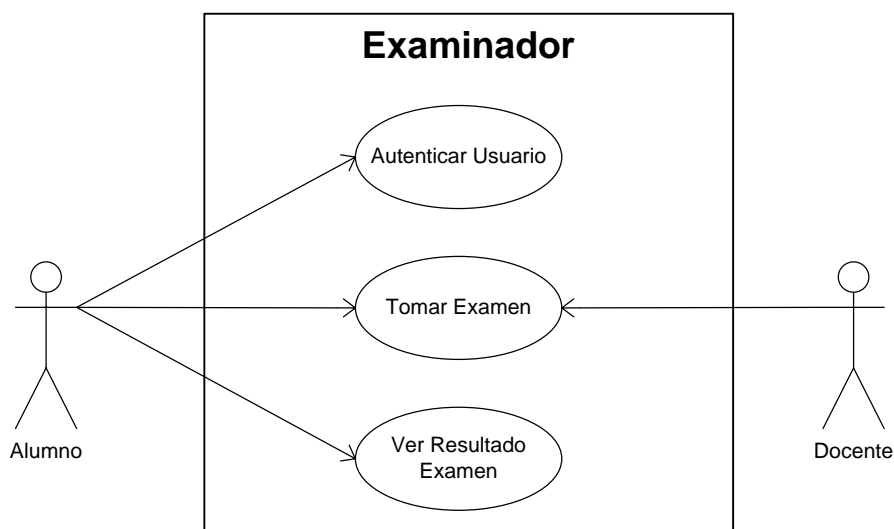


Gráfico 3.10: Diagrama caso de uso examinador

3.2 Diseño del sistema

3.2.1 Definición de la arquitectura

La arquitectura del sistema estará basada en N capas y N niveles, ésta arquitectura es muy eficiente ya que permite separar la funcionalidad del acceso a datos, lógica de negocios, servicios web y de la presentación. Esta separación lógica y física del sistema hace que el soporte a múltiples clientes sea fácil de implementar, probar y mantener.

A continuación una pequeña pero importante descripción de cada una de las capas del sistema distribuido CRONOS.

- La **capa de presentación** es la capa que contiene todo aquello con lo que el usuario puede interactuar. Además, contiene todos los elementos que constituyen la interfaz con el usuario, como por ejemplo los formularios o pantallas del sistema y los controles que estos contienen.
- La **capa de servicios** es una capa de abstracción que se crea entre la capa de presentación y la capa de negocios, y que agrupa funcionalidad de la capa de negocios para ser expuesta a la capa de presentación.
- La **capa de negocios** es la capa donde se maneja la lógica del negocio. Contiene la funcionalidad del sistema que interactúa con las peticiones realizadas en la capa de presentación y con los datos provistos por la capa de datos, ya sea para consultas o actualizaciones.
- La **capa de datos** es la capa donde se manejan los mecanismos para manipular la información. Esta permite conectarse a una base de datos relacional y a los esquemas de datos de cada aplicación a través de gestores de bases de datos que recibirán solicitudes de almacenamiento o recuperación de información desde la capa de negocios.

Los principales beneficios al desarrollar un sistema distribuido con esta arquitectura son:

- **Abstracción.** Las capas permiten cambios que se realicen en un nivel abstracto. Podemos incrementar o disminuir el nivel de abstracción usado en cada capa jerárquica.
- **Aislamiento.** El estilo de la arquitectura de capas permite aislar los cambios en tecnologías a ciertas capas para reducir el impacto en el sistema total.
- **Rendimiento.** Distribuir las capas entre múltiples niveles puede incrementar la estabilidad, la tolerancia a fallos y el rendimiento.

3.2.2 Diagramas de clases

- **Entidades de Negocio:** Es donde residen los datos e interactúa con todas las capas del sistema.

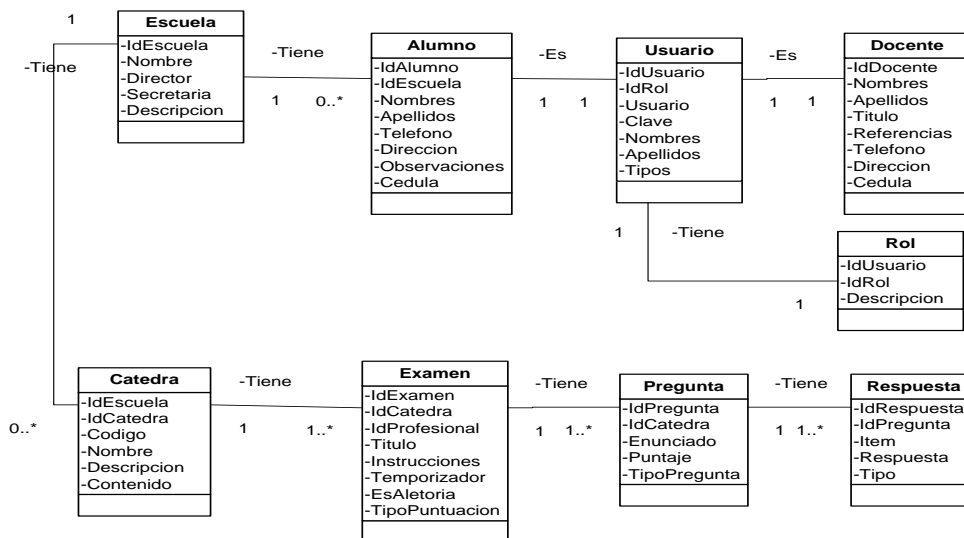


Gráfico 3.11: Diagrama de clases entidades de negocio

- **Acceso a Datos:** Se encarga de acceder y actualizar los datos y está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocios.

La capa de acceso a datos del sistema distribuido, dispone de una clase abstracta denominada *AccesoDatosBase*, ésta clase permite la comunicación con la base de datos, para la ejecución de sentencias Transact SQL, para una mejor comprensión de las clases de esta capa, ver el gráfico 3.12.

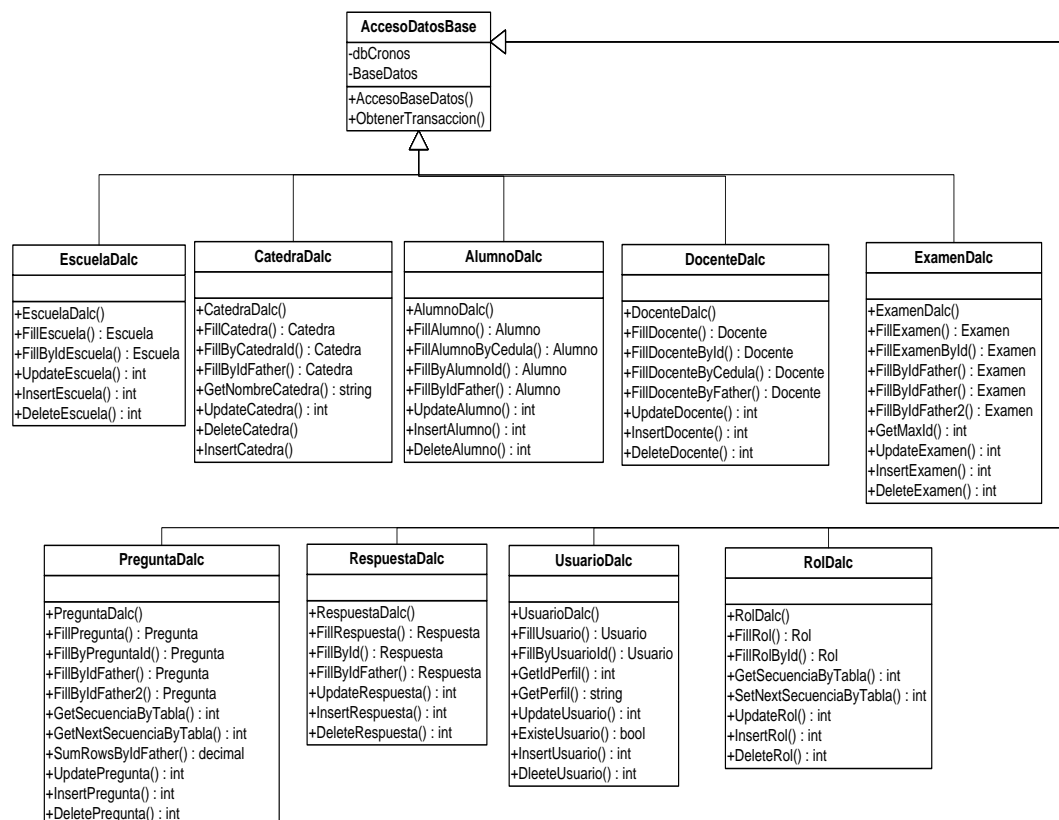


Gráfico 3.12: Diagrama de clases acceso a datos

- Lógica de negocio:** Aquí se establecen las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación mediante un servicio web, también se comunica con la capa de acceso a datos, para solicitar al gestor de bases de datos, para almacenar y recuperar información.

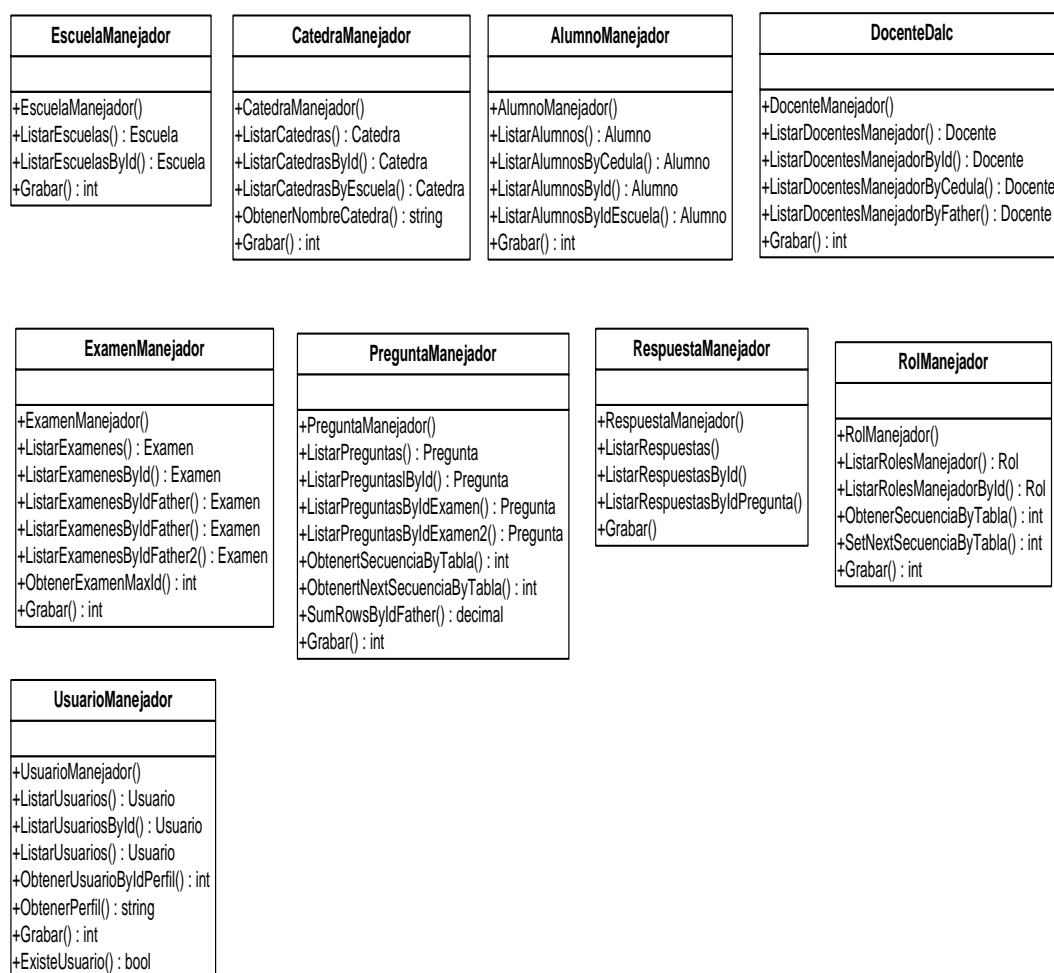


Gráfico3.13: Diagrama de clases lógicas de negocios

- Presentación:** Esta capa es la que ve e interactúa con el usuario, la capa de presentación ofrece a los usuarios información, sugerencias, acciones y captura los datos de entrada a través del teclado y el ratón.

La capa de presentación también debe ser independiente de la tecnología de interfaz de usuario y plataforma, aunque este requerimiento es muy complicado de cumplir y en la mayoría de las ocasiones es imposible obtener una independencia total.

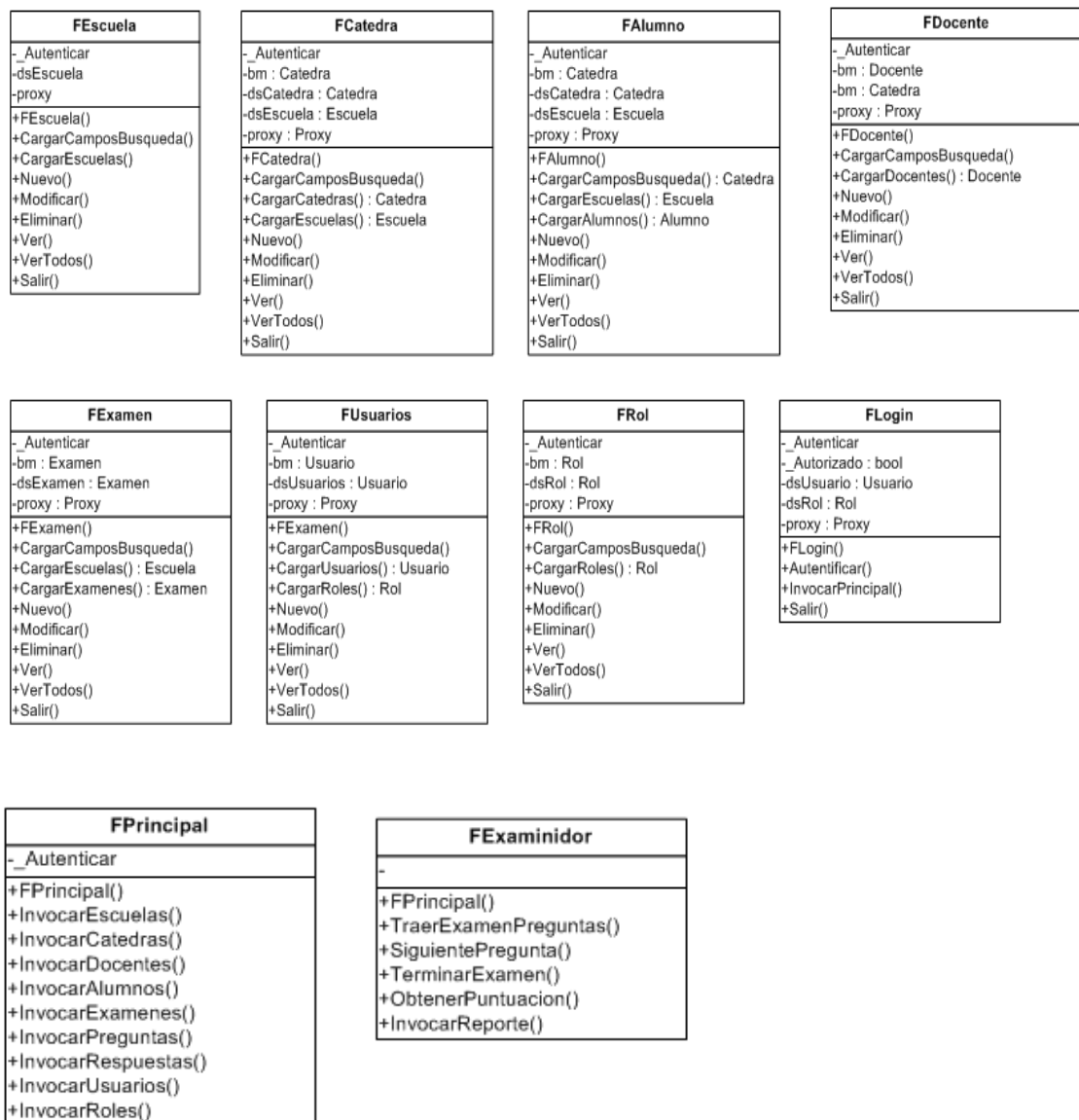


Gráfico 3.14: Diagrama de clases presentación

- **Servicio Web:** Actúa como enlace entre la capa de la lógica de negocio y la capa de presentación.

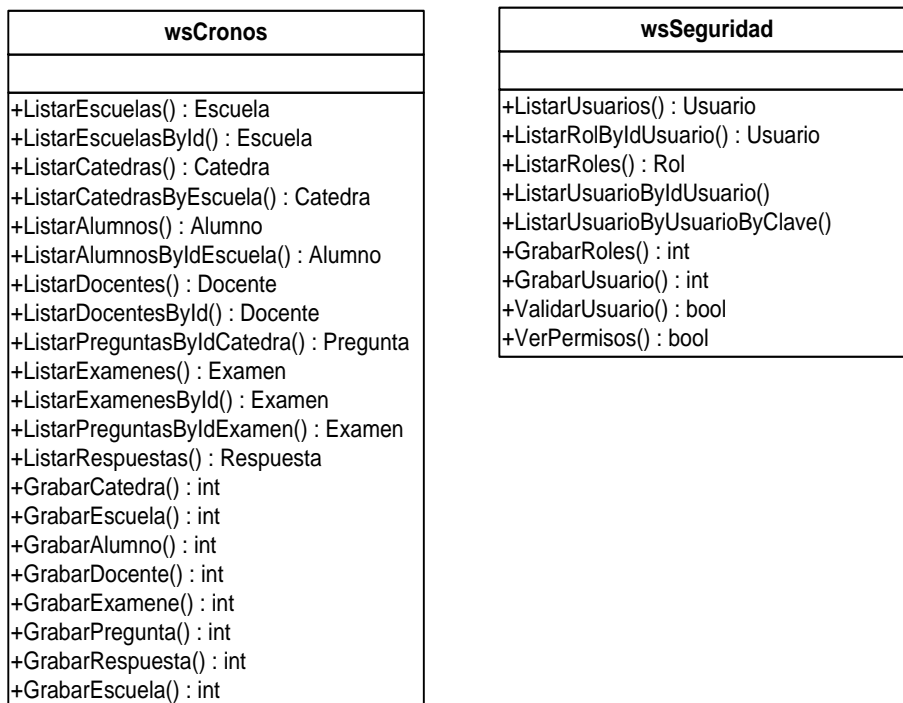


Gráfico 3.15: Diagrama de clases servicio web

3.2.3 Diagramas de secuencia

Con los diagramas de secuencia podemos delinear gráficamente los casos de uso del sistema.

Los siguientes gráficos muestran los principales diagramas de secuencia del sistema distribuido “CRONOS”.

Validar usuario

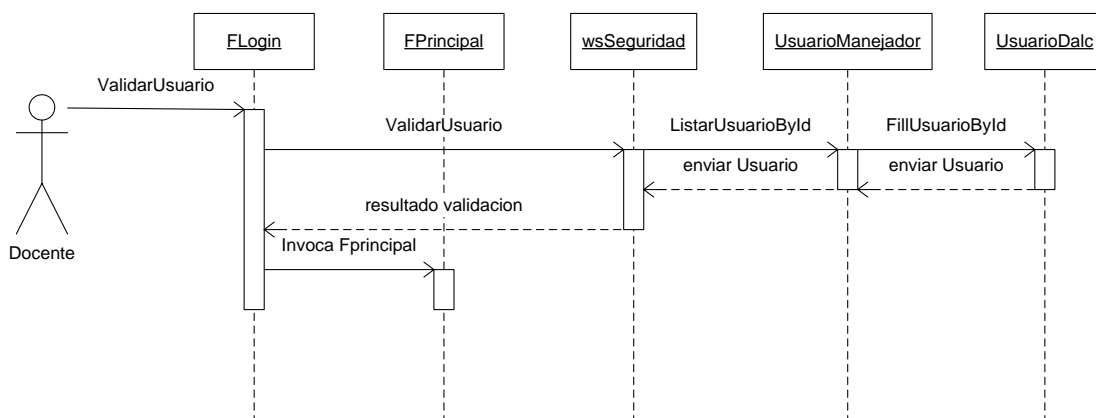


Gráfico 3.16: Diagrama de secuencia validar usuario

Crear escuela

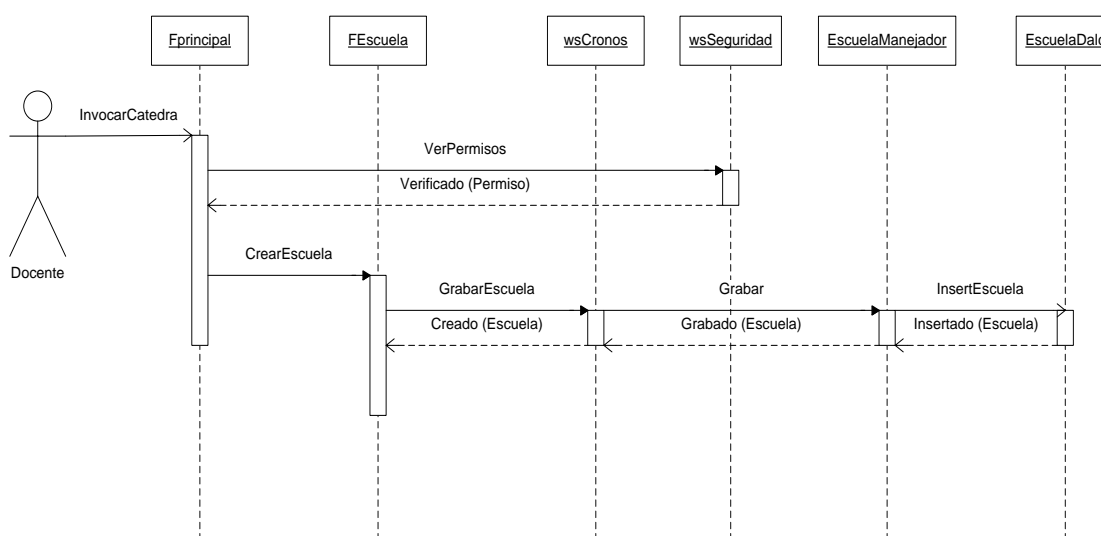


Gráfico 3.17: Diagrama de secuencia crear escuela

Modificar escuela

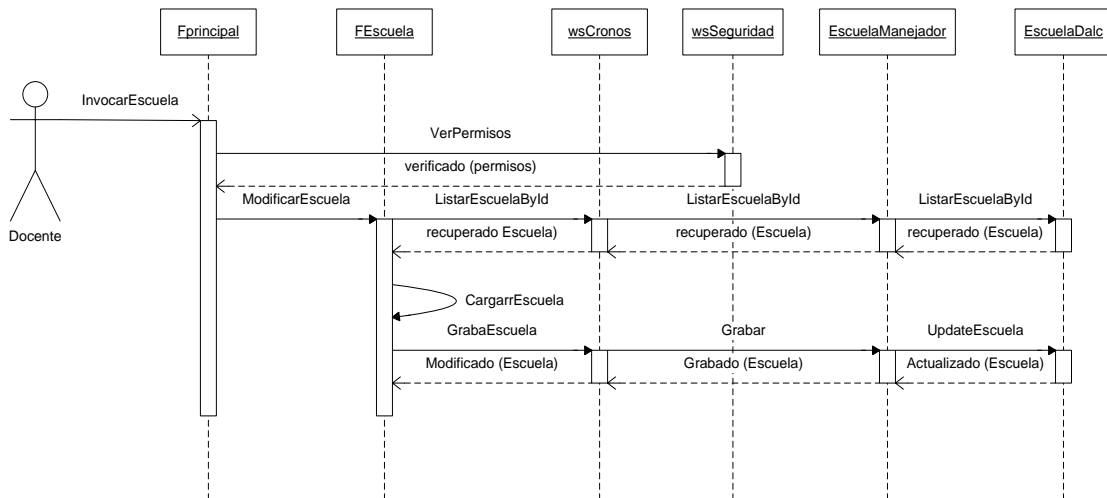


Gráfico 3.18: Diagrama de secuencia modificar escuela

Eliminar escuela

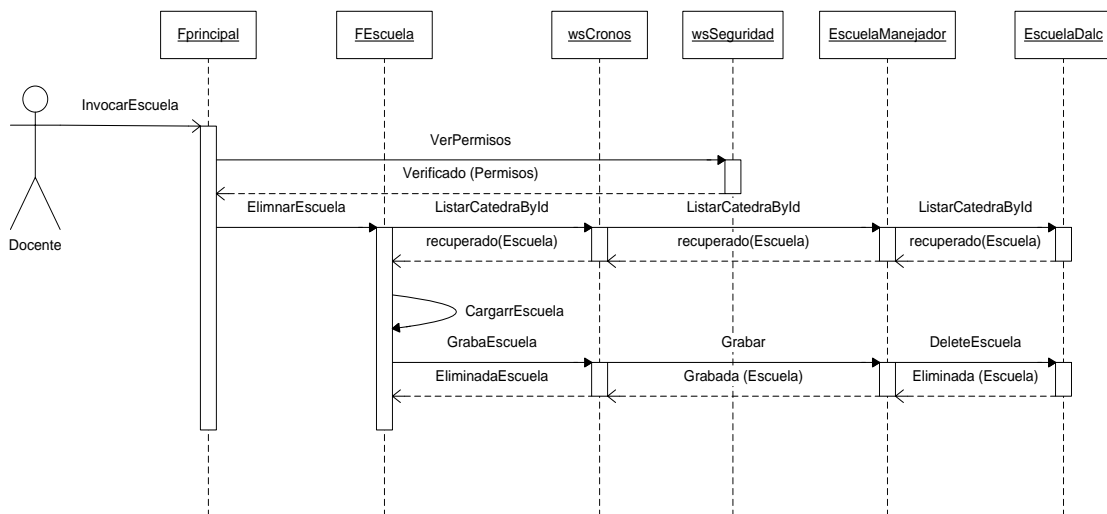


Gráfico 3.19: Diagrama de secuencia eliminar escuela

Ver escuela

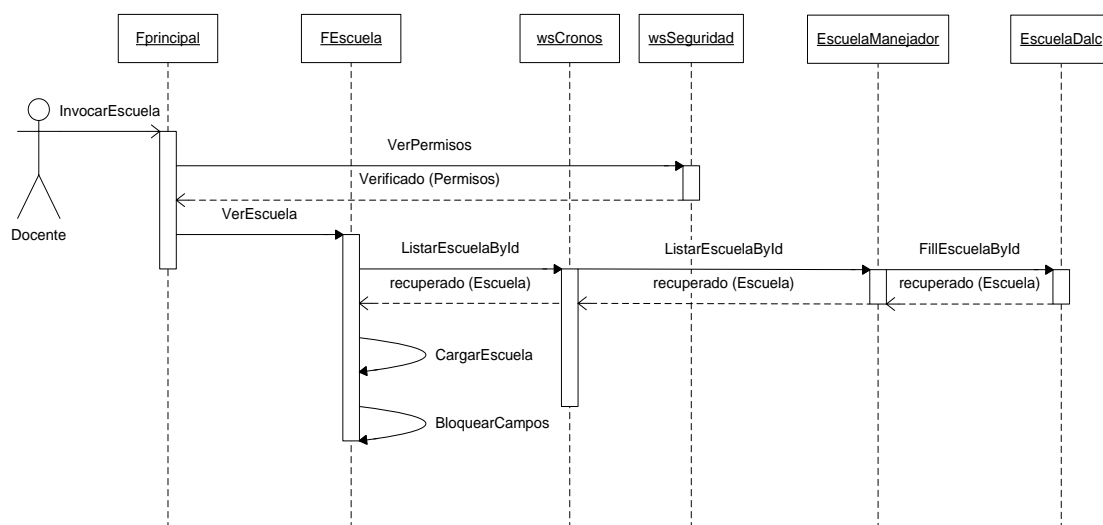


Gráfico 3.20: Diagrama de secuencia ver escuela

Crear cátedra

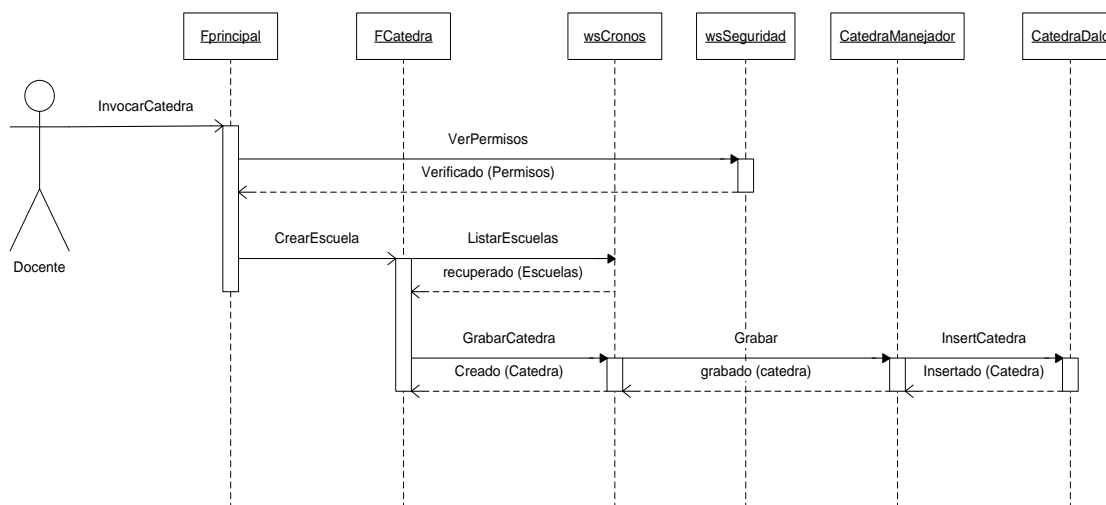


Gráfico 3.21: Diagrama de secuencia crear cátedra

Modificar cátedra

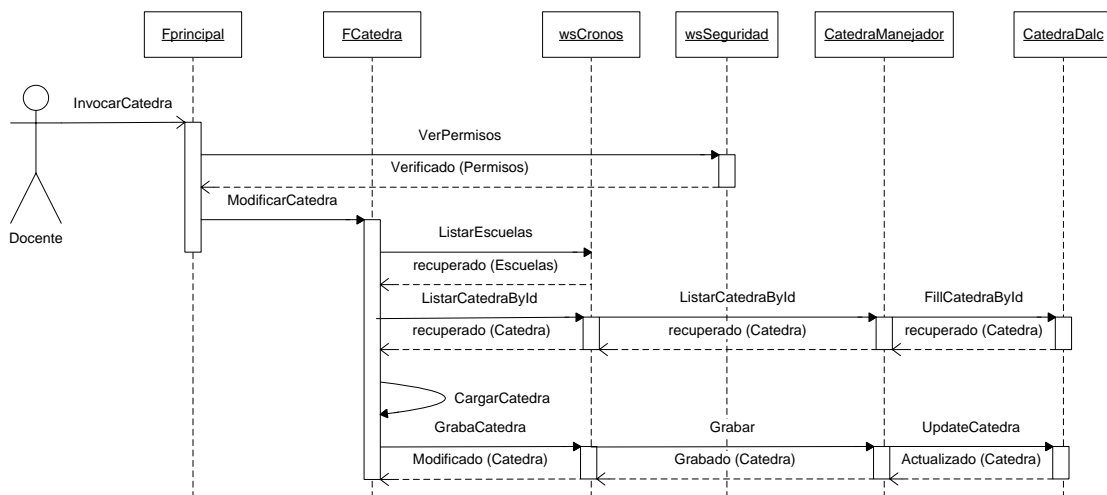


Gráfico 3.22: Diagrama de secuencia modificar cátedra

Eliminar cátedra

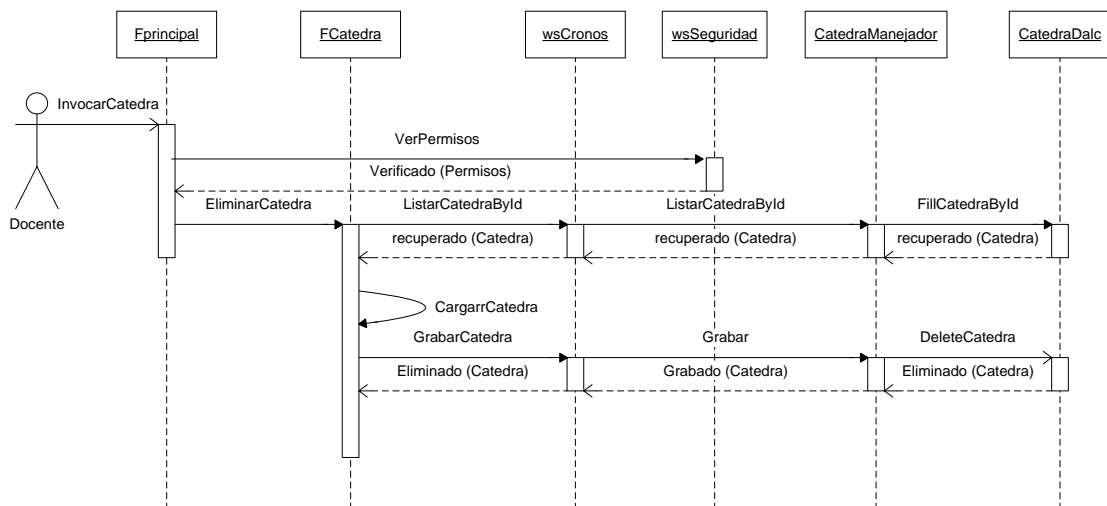


Gráfico 3.23: Diagrama de secuencia eliminar cátedra

Ver cátedra

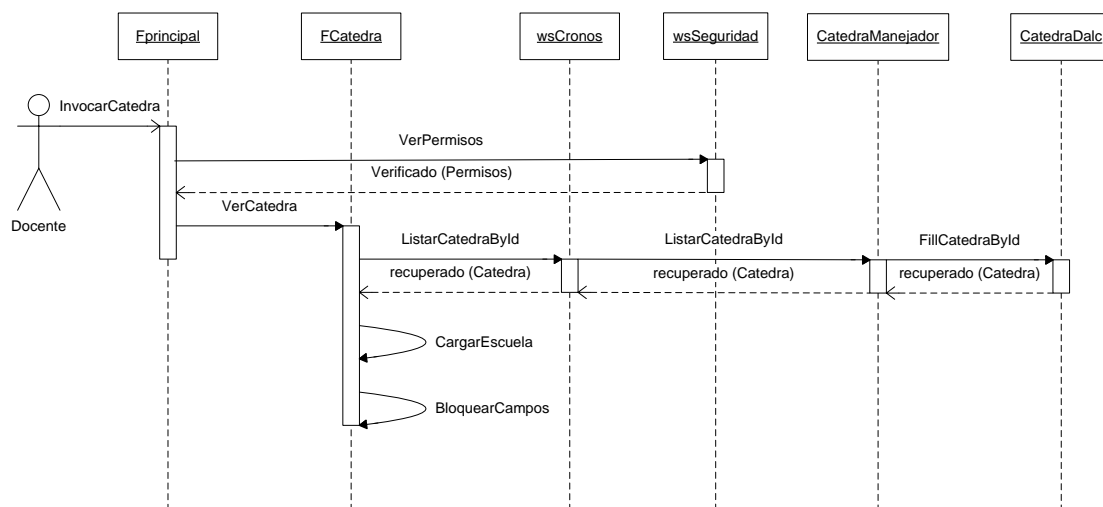


Gráfico 3.24: Diagrama de secuencia ver cátedra

Crear alumno

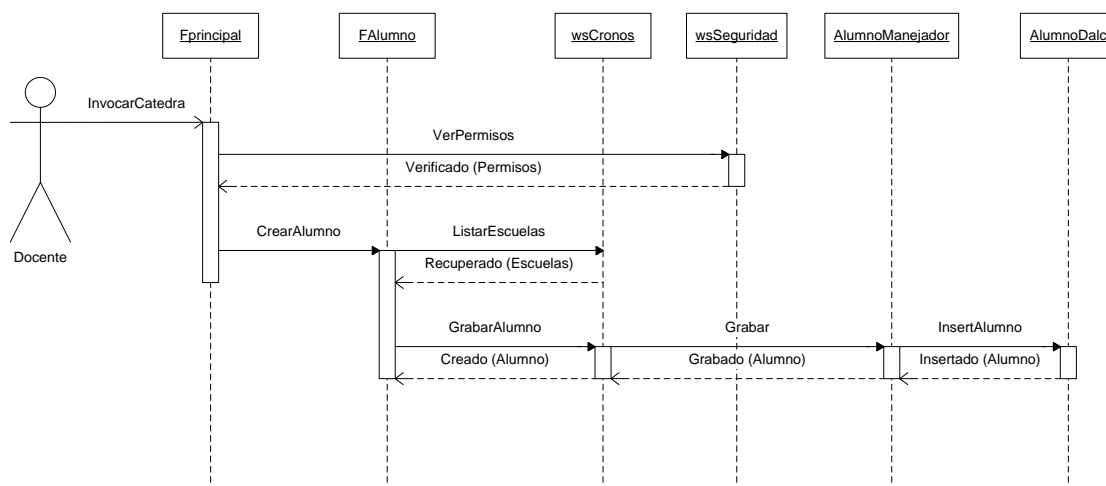


Gráfico 3.25: Diagrama de secuencia crear alumno

Modificar alumno

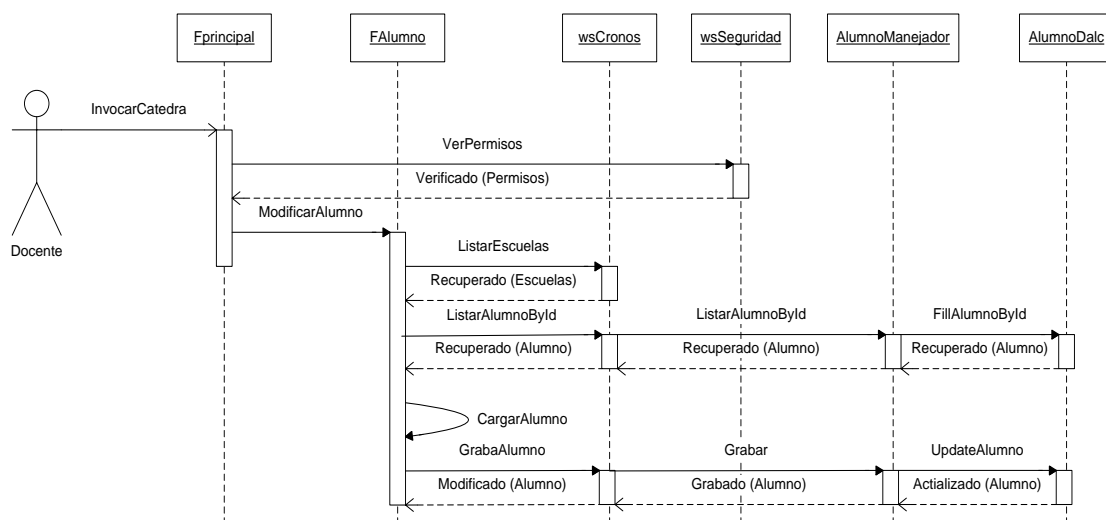


Gráfico 3.26: Diagrama de secuencia modificar alumno

Eliminar alumno

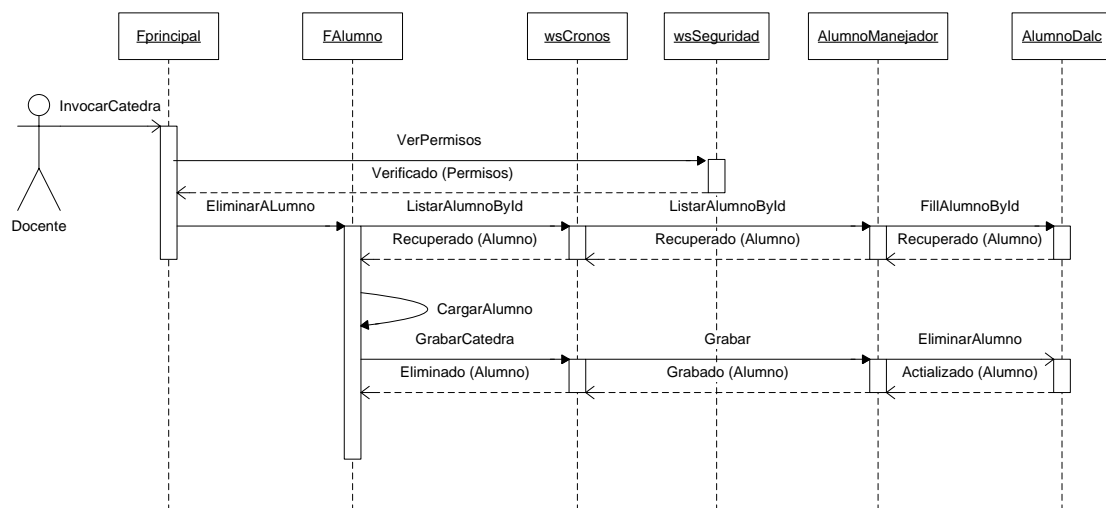


Gráfico 3.27: Diagrama de secuencia eliminar alumno

Ver alumno

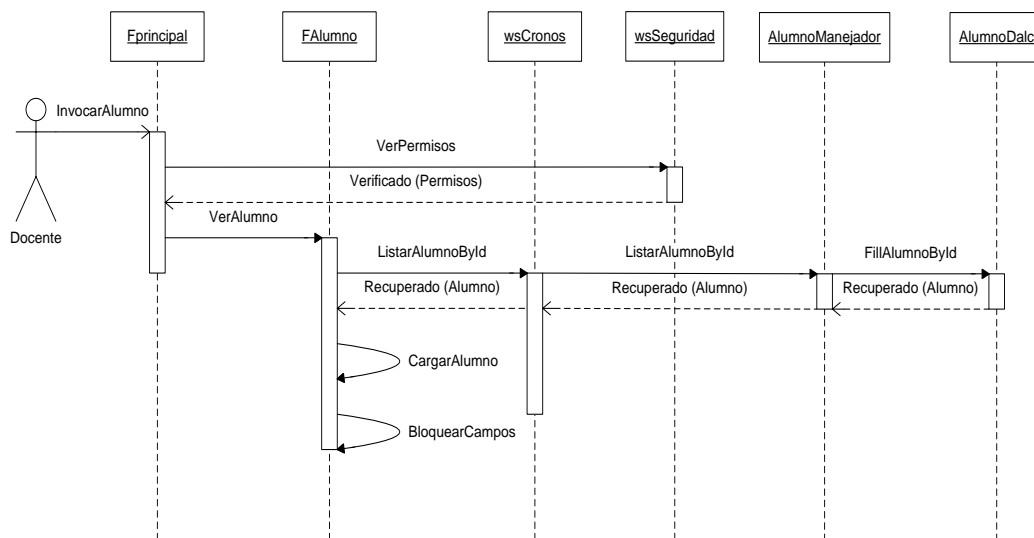


Gráfico 3.28: Diagrama de secuencia ver alumno

Crear docente

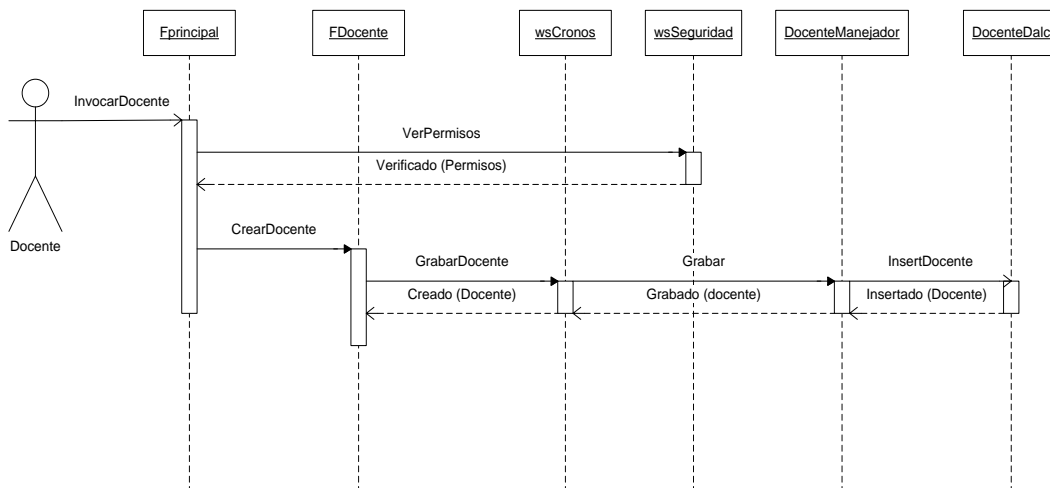


Gráfico 3.29: Diagrama de secuencia crear docente

Modificar docente

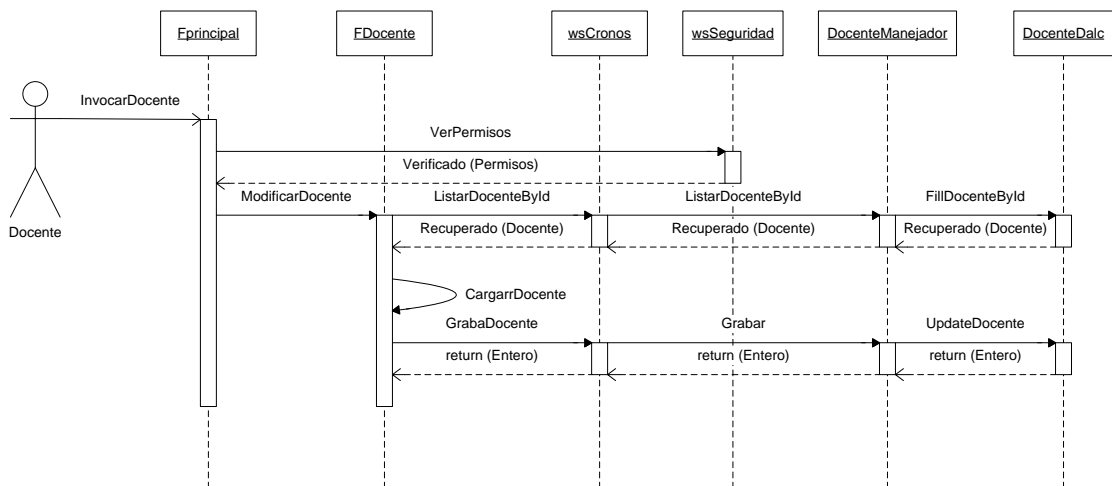


Gráfico 3.30: Diagrama de secuencia modificar docente

Eliminar docente

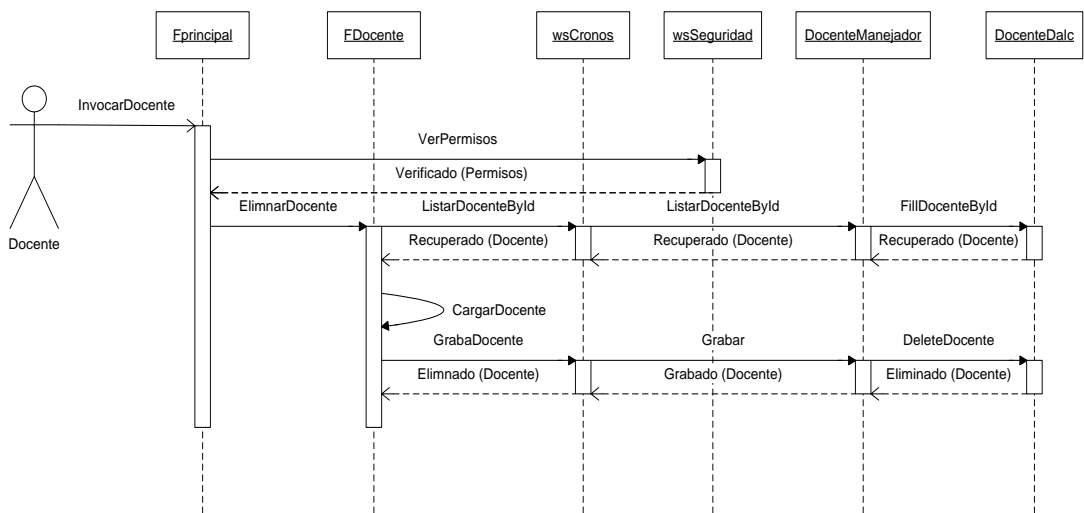


Gráfico 3.31: Diagrama de secuencia eliminar docente

Ver docente

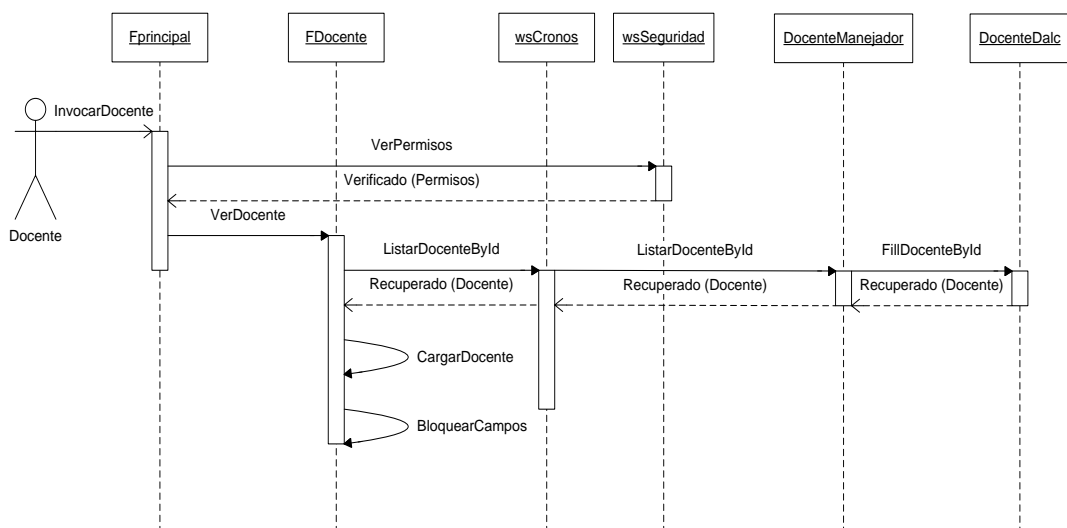


Gráfico 3.32: Diagrama de secuencia ver docente

Crear examen

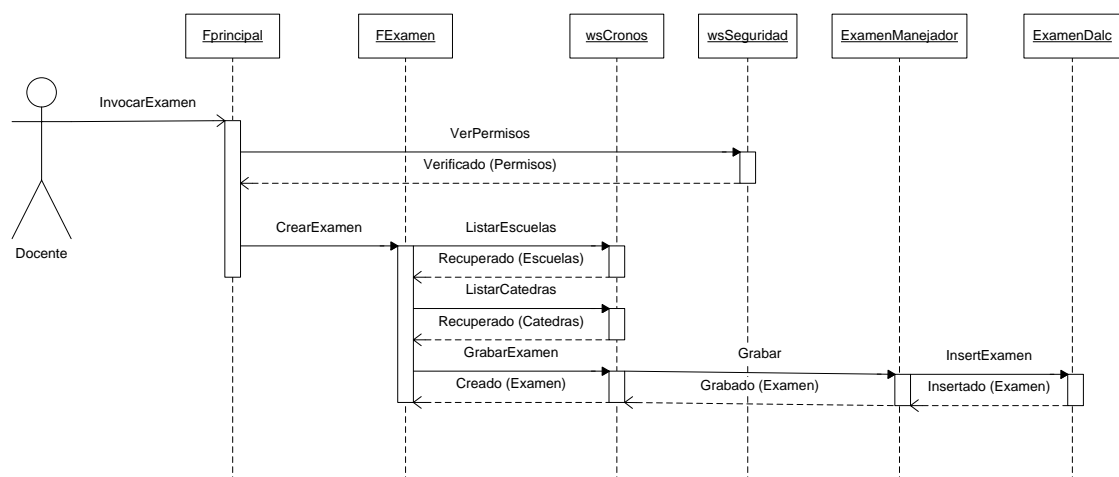


Gráfico 3.33: Diagrama de secuencia crear examen

Modificar examen

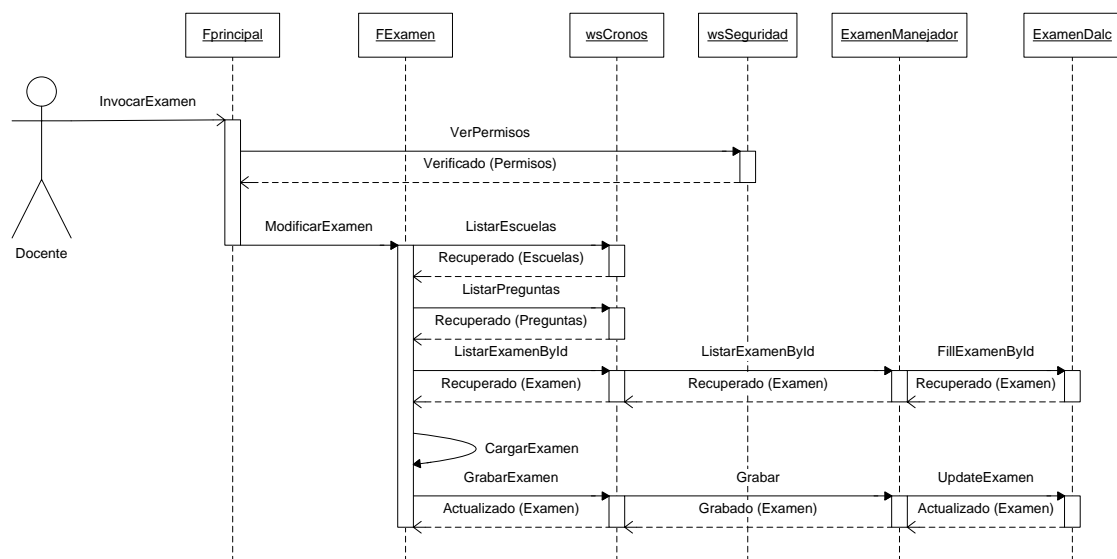


Gráfico 3.34: Diagrama de secuencia modificar examen

Eliminar examen

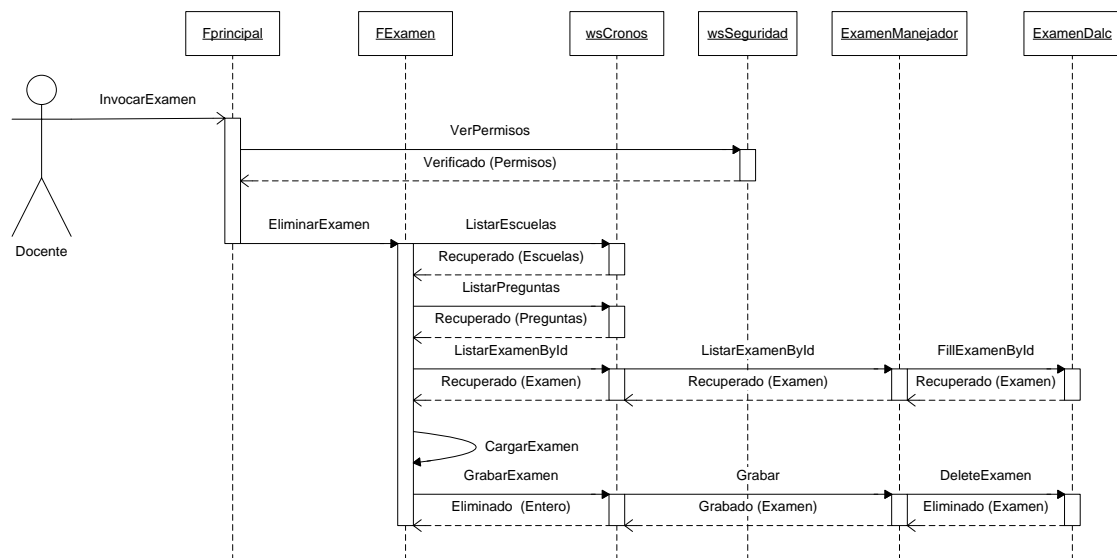


Gráfico 3.35: Diagrama de secuencia eliminar examen

Ver examen

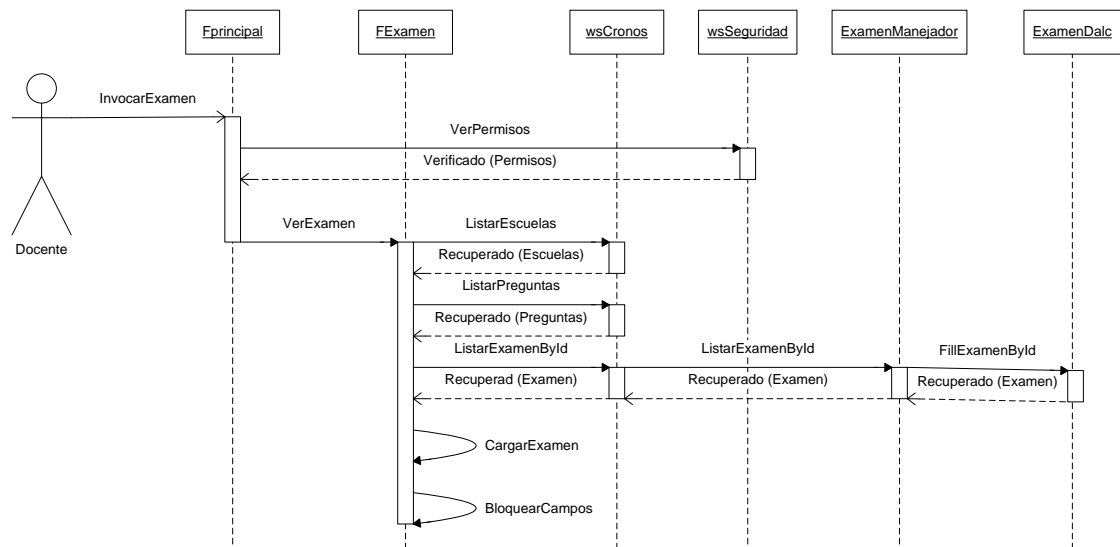


Gráfico 3.36: Diagrama de secuencia ver examen

Crear Pregunta

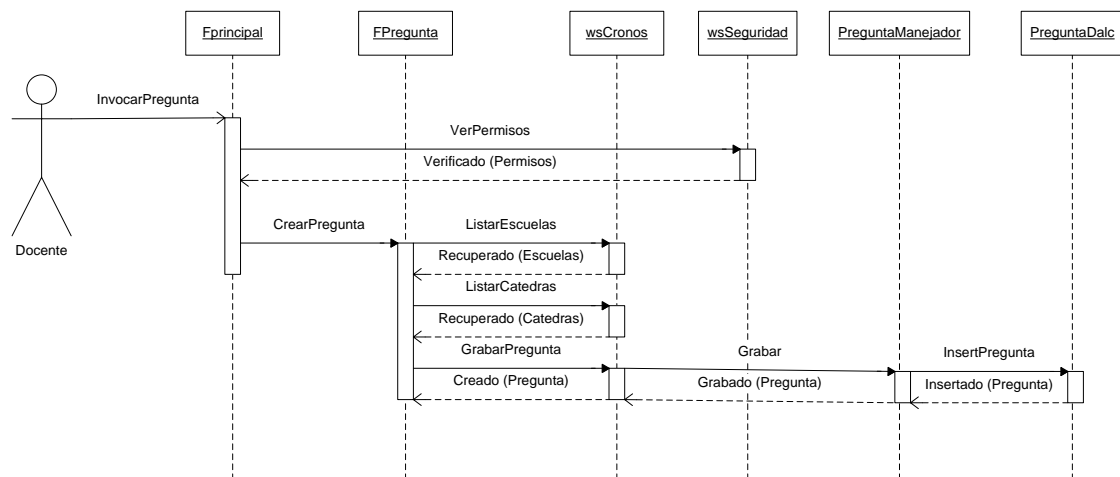


Gráfico 3.37: Diagrama de secuencia crear pregunta

Modificar pregunta

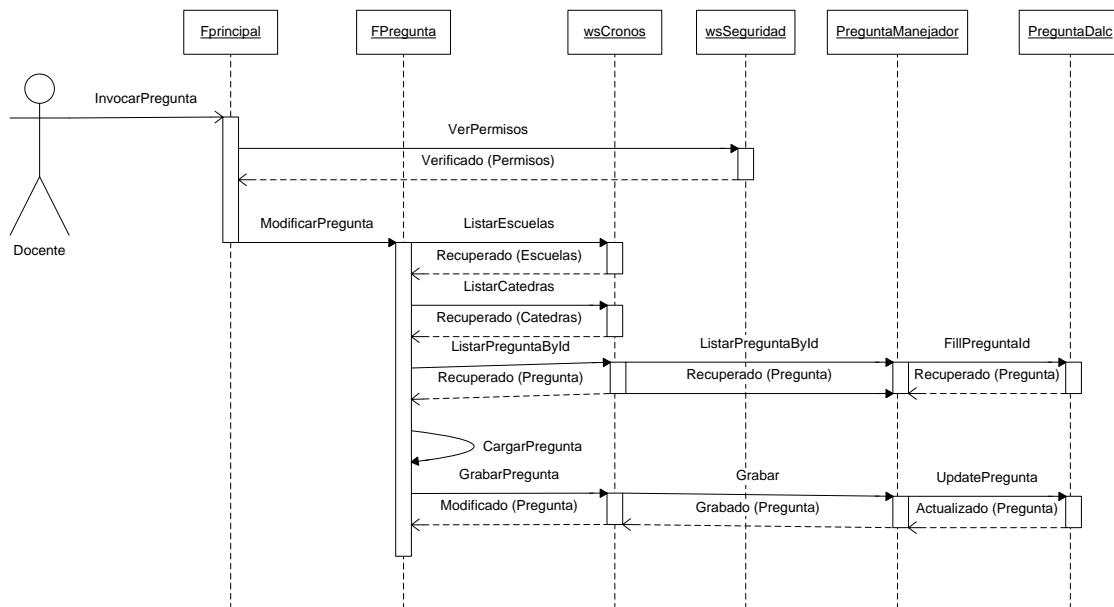


Gráfico 3.38: Diagrama de secuencia modificar pregunta

Eliminar pregunta

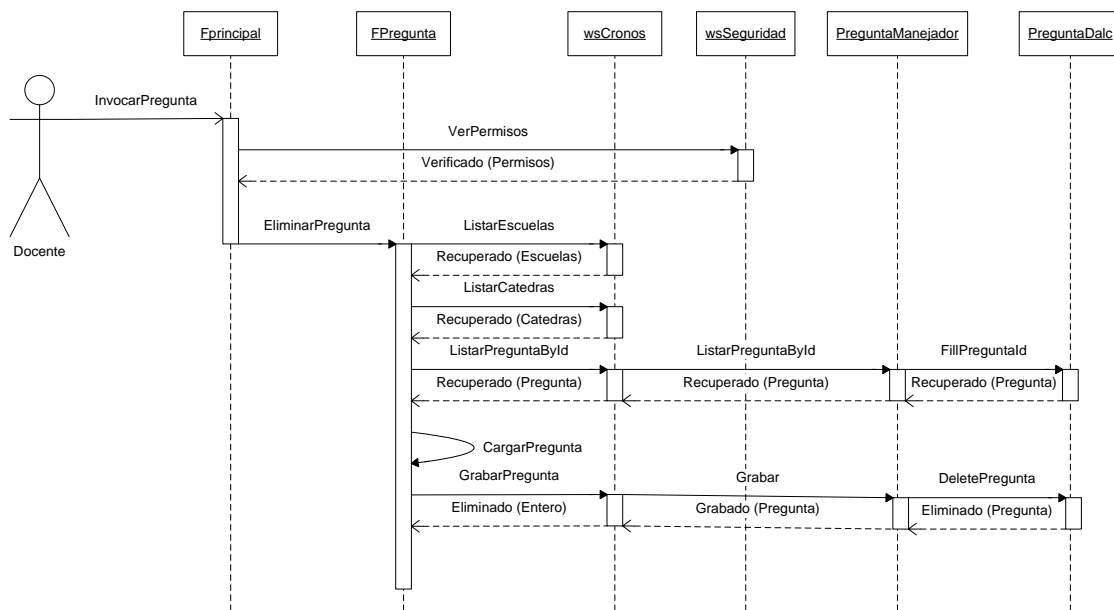


Gráfico 3.39: Diagrama de secuencia eliminar pregunta

Ver pregunta

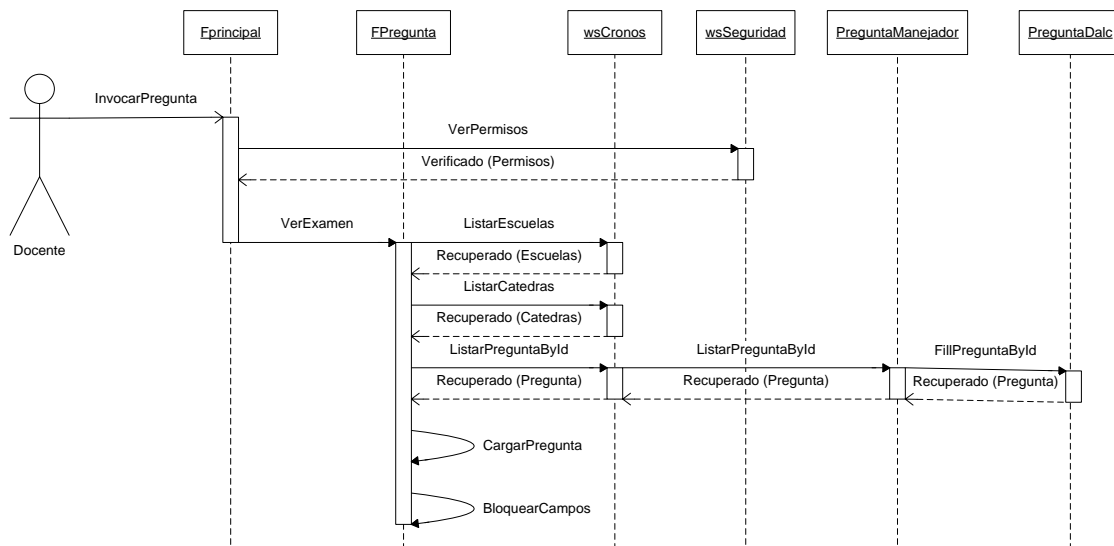


Gráfico 3.40: Diagrama de secuencia eliminar pregunta

Ver reporte

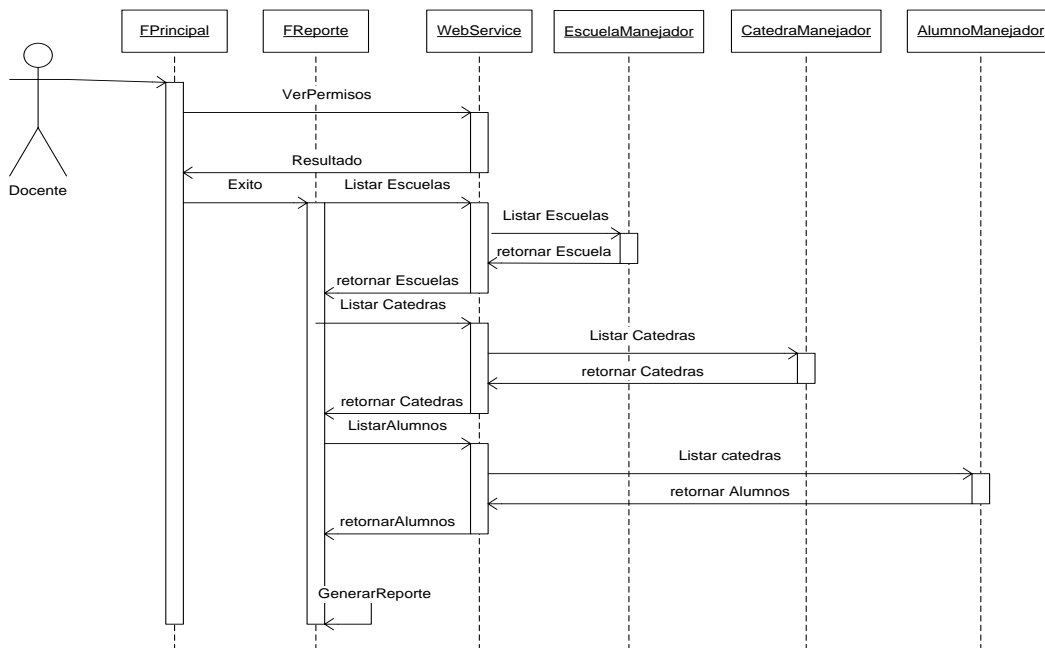


Gráfico 3.41: Diagrama de secuencia ver reporte

Tomar examen

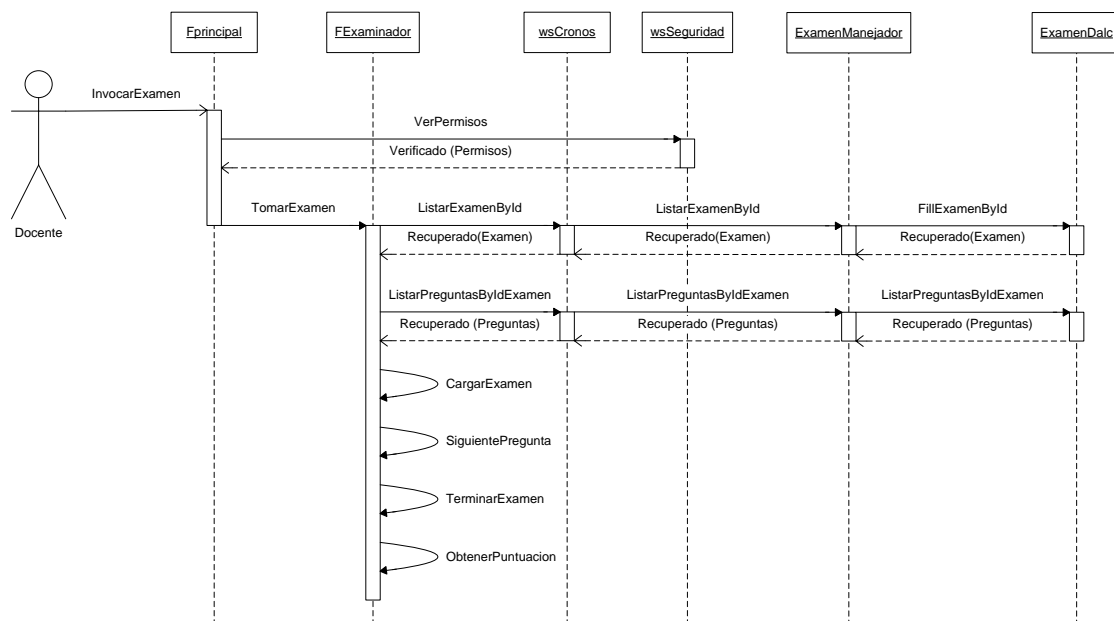


Gráfico 3.42: Diagrama de secuencia Tomar examen

3.2.4 Diseño de la interfaz del sistema

El diseño del sistema, (Capa de presentación). Estos muestran las pantallas con las que interactúa el usuario para realizar de manera atomizada los procesos anteriormente descritos. Así mismo, para los procesos más importantes, se utiliza diagramas de secuencias para comprender la manera en que el sistema realiza las funcionalidades descritas.

El sistema está básicamente dividido en dos grandes módulos de cara al usuario, estos son Administrador y Examinador. Estos dos módulos tienen una pantalla en común que es la de autenticación.

3.2.4.1 Módulo Administrador

El módulo administrador es diseñado para los usuarios que sean docentes, encargados de administrar entidades como por ejemplo (escuelas, docentes, alumnos, exámenes, etc.).

- **Acceso al módulo de administrador**

Para ingresar al sistema, los usuarios deben contar con un usuario y contraseña. Estos datos de acceso le permitirán ingresar al modulo del sistema que haya requerido.

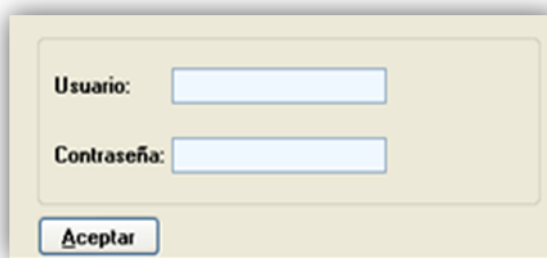
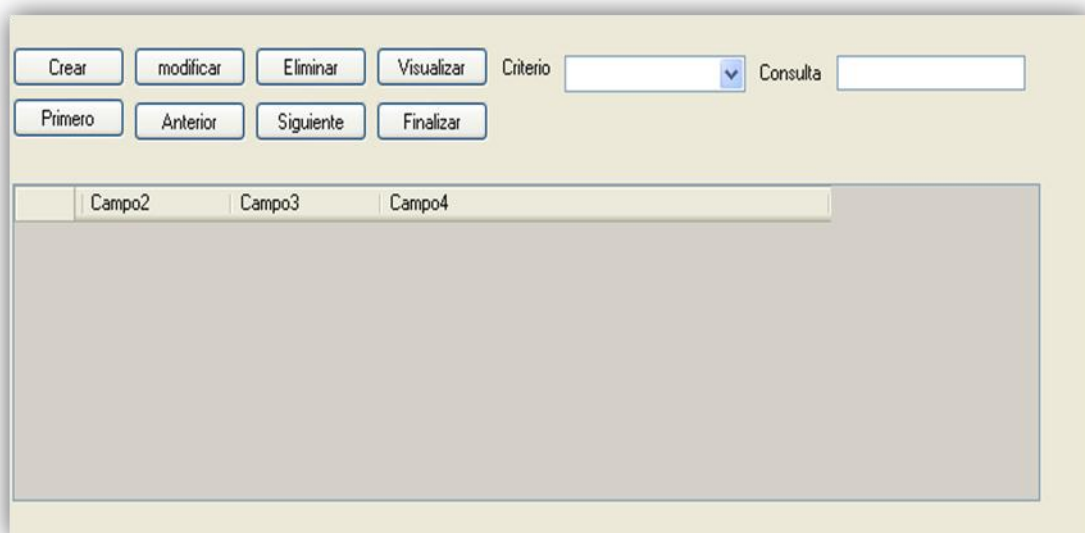
Un formulario de acceso con un fondo beige. Contiene dos campos de texto blancos con bordes azules. El primer campo está etiquetado como 'Usuario:' y el segundo como 'Contraseña:'. Debajo de los campos hay un botón rectangular con el texto 'Aceptar'.

Gráfico 3.43: Acceso módulo administrador

- **Pantalla principal**

La pantalla principal es la que permite ver todas las entidades (Usuarios, Docentes, Alumnos, etc.), admitirá realizar filtros mediante búsquedas y administrarlas.




The screenshot shows a main application window with a light beige background. At the top, there are two rows of buttons: 'Crear', 'modificar', 'Eliminar', and 'Visualizar' in the first row; 'Primero', 'Anterior', 'Siguiete', and 'Finalizar' in the second row. To the right of these buttons is a 'Criterio' dropdown menu and a 'Consulta' text input field. Below the buttons is a table with three columns labeled 'Campo2', 'Campo3', and 'Campo4'. The table area is currently empty.

Gráfico 3.44: pantalla principal

- **Administrar Escuelas**

CRONOS permitirá registrar Escuelas como por ejemplo Ingeniería de sistemas u Optometría



The screenshot shows a form for administering schools. At the top, there are three buttons: 'Nuevo', 'Grabar-Eliminar', and 'Salir'. Below the buttons are four labeled text input fields: 'Nombre:', 'Director:', 'Secretaria:', and 'Descripcion:'. Each label is followed by a white rectangular input box.

Gráfico 3.45: Pantalla administrar escuelas

- **Cátedras**

CRONOS permitirá registrar cátedras como por ejemplo Suficiencia en informática, ingles, física, etc. ofreciendo un listado de las escuelas para su asignación




Gráfico 3.46: Pantalla administrar cátedras

- **Administrar docentes**

CRONOS permitirá registrar docentes habilitados para todas las escuelas. Ver Gráfico 3.44

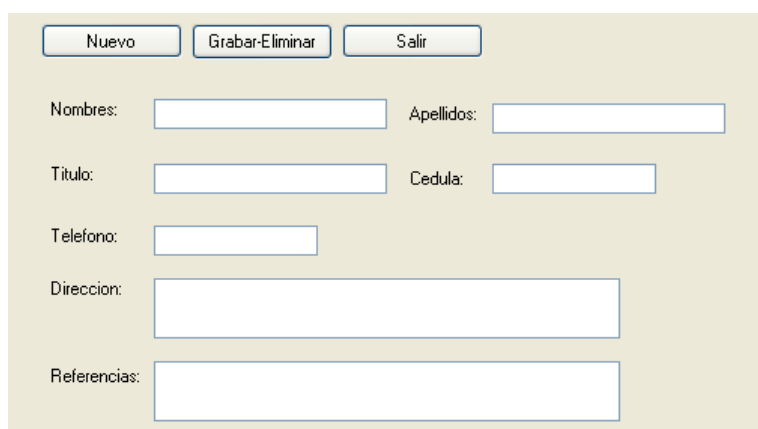


Gráfico 3.47: Pantalla administrar docentes

- **Administrar alumnos**

Para el registro de alumnos CRONOS desplegará las escuelas existentes para que el alumno sea asignado a una.

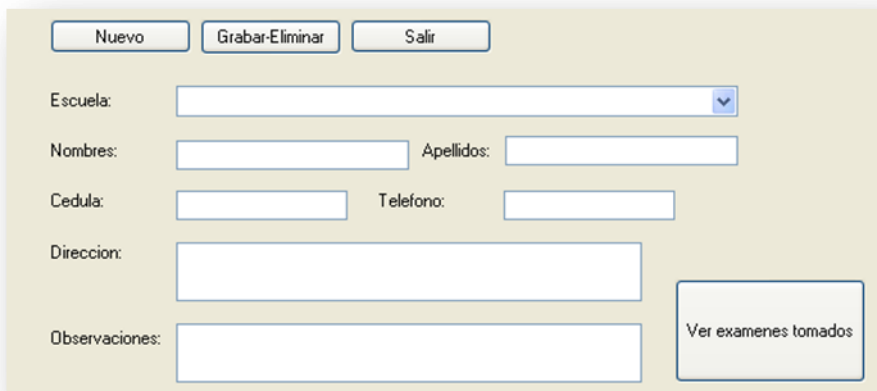


Gráfico 3.48: Pantalla administrar alumnos

- **Banco de preguntas**

Para el registro del banco de preguntas el usuario podrá establecer el tipo de pregunta (decisión, enlace, práctica).

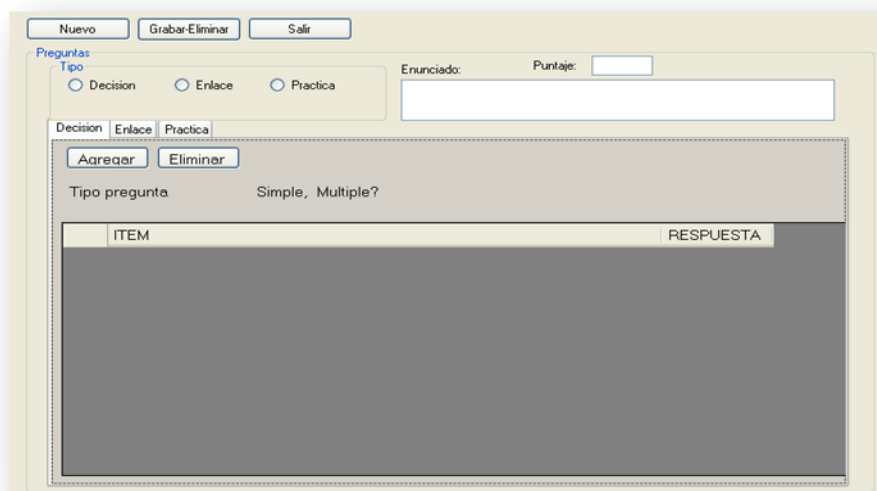


Gráfico 3.49: Pantalla administrar banco de preguntas

- **Administrar exámenes**

Para administrar los exámenes, CRONOS pondrá a disposición todas las preguntas registradas en el banco de preguntas, también presentará los campos necesarios para su respectiva evaluación.

ENUNCIADO	TIPO	PUNTAJE
-----------	------	---------

Gráfico 3.50: Pantalla administrar exámenes

- **Administrar usuarios**

CRONOS mostrará una pantalla para administrar usuarios, listando los docentes, alumnos y roles existentes.

ENUNCIADO	TIPO	PUNTAJE
-----------	------	---------

Gráfico 3.51: Pantalla administrar usuarios

- **Administrar roles**

Para la administración de roles, CRONOS listará todas las pantallas con que cuenta para ser seleccionadas y formar un rol.



Gráfico 3.52: Pantalla administrar roles

3.2.4.2 Módulo Examinador

Solo los usuarios que sean alumnos podrán tomar el examen accediendo a este módulo.

- **Acceso al módulo examinador**

Para ingresar a CRONOS examinador, los usuarios deben contar con un usuario y contraseña. Estos datos de acceso le permitirán ingresar al modulo del sistema que haya requerido.

Gráfico 3.53: Pantalla acceso al módulo examinador

- **Examinador Información**

CRONOS mostrará los datos del usuario (alumno) y la información necesaria de un examen seleccionado por el usuario (alumnos).

Gráfico3.54: Pantalla examinador información

- **Examinador**

Es la evaluación, CRONOS presentará la cátedra, número de pregunta y el cuerpo de todas las alternativas, en caso de que la pregunta sea del tipo (practica) CRONOS invocará al respectivo programa office donde el usuario deberá realizar el desarrollo de la pregunta.

CATEDRA	
ITEM PREGUNTA	
PUNTAJE	ITEMS DE RESPUESTAS PARA PREGUNTAS DE DECISION SIMPLE, MULTIPLE Y ENLACE
TIEMPO RESTANTE	
Finalizar / Siguiente	

Gráfico 3.55: Pantalla examinador

3.3 Construcción

3.3.1 Selección de herramientas

La relación de herramientas software utilizadas para el desarrollo del sistema distribuido “CRONOS” se detalla en la tabla 3.46

Herramienta		Razones de selección
Entorno de desarrollo integrado	Visual Studio 2008	Entorno de desarrollo de fácil uso y completo.
Lenguaje	C#	Sintaxis de programación semejante a C/C++ Único lenguaje 100% orientado a objetos
Entorno de ejecución	.Net	Plataforma de ejecución intermedia
Framework	.Net Framework 3.5	
Gestor base de datos	Microsoft SqlServer 2008	Entorno de administración de fácil uso y completo
Lenguaje de base de datos	Transact-SQL	Lenguaje SQL inherente del producto seleccionado

Tabla 3.46: Paquete del sistema (Examinador)

3.3.2 Requerimientos de hardware y software del equipo de desarrollo

Los requerimientos del hardware y software que se necesitaron para el desarrollo del sistema se detallan en las siguientes tablas:

Dispositivo	Características
Procesador	Intel core duo, Velocidad 2.66 GHz.
Memoria RAM	2GB
Disco Duro	80 GB

Tabla 3.47: Requerimientos de hardware del equipo de desarrollo

Software	Versión
Sistema Operativo	Microsoft Windows XP Service Pack 3
Programas	<ul style="list-style-type: none"> • Microsoft Office 2007 • Herramientas de software detallada en la tabla 3.36

Tabla 3.48: Requerimientos de software del equipo de desarrollo

3.3.3 Codificación

La base de datos del sistema está constituida por tablas, procedimientos almacenados y disparadores, a continuación se muestra los siguientes gráficos para ilustrar la construcción de la base de datos mediante Transact SQL mediante MS SQL server.

```

(local).Cronos - SQLQuery1.sql* Summary
CREATE TABLE [dbo].[EXAMEN] (
  [IDEXAMEN] int IDENTITY(1, 1) NOT NULL,
  [CODIGO] varchar(8) COLLATE Modern_Spanish_CI_AS NOT NULL,
  [IDCATEDRA] int NULL,
  [IDPROFESIONAL] int NULL,
  [TITULO] varchar(80) COLLATE Modern_Spanish_CI_AS NULL,
  [INSTRUCCIONES] varchar(200) COLLATE Modern_Spanish_CI_AS NULL,
  [TEMPORIZADOR] bit NULL,
  [TIEMPO] int NULL,
  [NUMERO_PREGUNTAS] int NULL,
  [PUNTUACION_CORREGIDA] bit NULL,
  [ES_ALEATORIA] bit NULL
)
ON [PRIMARY]
;

```

Gráfico 3.56: Transact SQL Creación de una tabla

```

(local).Cronos - SQLQuery1.sql* Summary
CREATE PROCEDURE [dbo].[LISTARCATEDRABYIDESCUELA]
  @IDESCUELA INT
AS
BEGIN
  SET NOCOUNT ON;

  SELECT  IDCATEDRA,
          IDESCUELA,
          CODIGO,
          NOMBRE,
          CONTENIDO,
          DESCRIPCION
  FROM    CATEDRA
  WHERE   (IDESCUELA = @IDESCUELA)

  ORDER BY NOMBRE;

END

```

Gráfico 3.57: Transact SQL Creación de un procedimiento almacenado

```

(local).Cronos - SQLQuery1.sql* Summary
CREATE TRIGGER [dbo].[EliminarUsuario] ON [dbo].[ESTUDIANTE]
WITH EXECUTE AS CALLER
FOR DELETE
AS
    Declare @CedulaEstudiante VARCHAR(10)

    Select @CedulaEstudiante = d.CEDULA
    From deleted d
    If (@CedulaEstudiante is not null) or (@CedulaEstudiante <> '')
    Begin
        Delete From USUARIO
        Where Clave = @CedulaEstudiante

    end
;

```

Gráfico 3.58: Transact SQL Creación de un disparador

Para la creación del sistema se hace uso de Visual Studio 2008 ya que es una herramienta que facilita enormemente la organización, codificación, depuración y compilación del sistema, en el gráfico 3.59 podemos ver el sistema CRONOS representado en una solución de Visual Studio 2008.

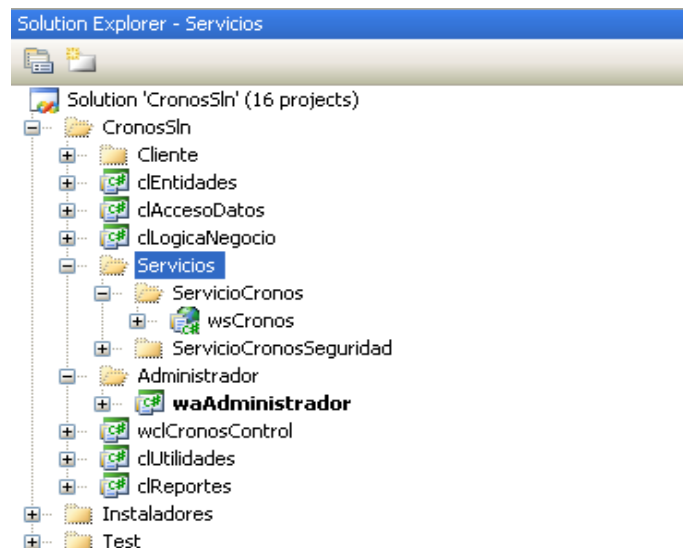


Gráfico 3.59: Explorador de la solución del sistema CRONOS

En la codificación de las clases del sistema se agruparon por proyectos donde cada proyecto representa una capa, las clases están escritas con el lenguaje orientado objetos C# 3, a excepción de la capa entidades de negocio donde se hace uso de los

Datasets. Los gráficos que aparecen a continuación son un ejemplo de lo antes mencionado.

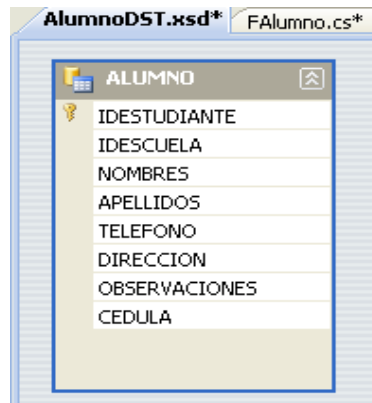


Gráfico 3.60: Código fuente de la capa de acceso a datos

```

/* =====
-- Author:      Jose Castillo
-- Create date: 13\07\2009
-- Description: Acceso de datos a la entidad Alumno
===== */

namespace Pucesa.Cronos.AccesoDatos
{
    public class AlumnoDALC : AccesoDatosBase
    {
        public AlumnoDALC()...
        #region Fill
        /// <summary>
        /// Metodo: Lista todos los Alumnos
        /// </summary>
        /// <returns></returns>
        public AlumnoDST Fill()...

        /// <summary> ...
        public AlumnoDST FillById(int id) ...
        /// <summary> ...
        public AlumnoDST FillByCedula(string cedula) ...
        #endregion

        #region Update
        public int Update(AlumnoDST dsAlumno, IDbTransaction trans) ...
        #endregion
    }
}

```

Gráfico 3.61: Código fuente de la capa de acceso a datos

```

/* =====
-- Author:      Jose Castillo
-- Create date: 21\03\2009
-- Description: Logica de la entidad ALUMNO
===== */

namespace Pucesa.Cronos.Logica
{
    public class Alumno
    {
        public Alumno()
        {
        }

        #region Fill
        /// <summary> ...
        public AlumnoDST Fill()...
        /// <summary>
        /// Metodo: Lista todos los alumnos por su key
        /// </summary>
        /// <param name="id"> Parametro entero</param>
        /// <returns> AlumnoDST </returns>
        public AlumnoDST FillById(int id)...
        /// <summary> ...
        public AlumnoDST FillByCedula(string cedula)...
        #endregion
        #region Update
        public int Update(AlumnoDST dsAlumno)...
        #endregion
    }
}

```

Gráfico 3.62: Código fuente de la capa de lógica de negocio

```

#region Alumno
[WebMethod]
public AlumnoDST ListarAlumno()
{
    return new Alumno().Fill();
}
[WebMethod]
public AlumnoDST ListarAlumnoByID(int ID)
{
    return new Alumno().FillById(ID);
}
[WebMethod]
public AlumnoDST ListarAlumnoByCedula(string cedula)
{
    return new Alumno().FillByCedula(cedula);
}

[WebMethod]
public int GrabarAlumno(AlumnoDST dsAlumno)
{
    return new Alumno().Update(dsAlumno);
}
#endregion

```

Gráfico 3.63: Código fuente de la capa de servicios

```

using Pucesa.Cronos.Interface.Administrador.servCronos;
using Pucesa.Cronos.Utilidades;
using Pucesa.Cronos.Interface.Administrador.Seguridad;

namespace Pucesa.Cronos.Interface.Administrador.Alumno
{
    public partial class FAlumno : Form
    {
        BindingManagerBase bm;
        svCronos proxy = new svCronos();
        EscuelaDST dsEscuela = new EscuelaDST();
        AlumnoDST dsAlumno = new AlumnoDST();
        Autenticacion _Autenticar = null;
        string _invocador = string.Empty;
        string _nombres = string.Empty;
        string _apellidos = string.Empty;
        string _cedula = string.Empty;

        Inicializacion
        Metodos
        Menu
        Propiedades
    }
}

```

Gráfico 3.64: Código fuente de la capa de presentación

3.3.4 Pruebas

Las pruebas de software son los procesos que permiten verificar y revelar la calidad de un software, mediante su utilización se puede identificar posibles fallos de implementación, calidad o usabilidad de un programa de computador.

Pero hay pruebas que necesitan atención específica con el fin de lograr un proceso bien definido que constituye la base para mejorar y estructurar el proceso de pruebas, es así que el sistema CRONOS fue sometido a tres tipos de pruebas que son las siguientes.

3.3.4.1 Pruebas de Unidad

Mediante este tipo de prueba la unidad es probada como tal, como una clase u paquete de servicio o un subsistema.

La información y resultados de este tipo de prueba se detallan en las siguientes tablas.

Responsable de la prueba	<ul style="list-style-type: none"> • José Castillo
Verificador	<ul style="list-style-type: none"> • Ing. Patricio Medina
Clase	<ul style="list-style-type: none"> • Servicio Web Cronos (servCronos)
Método	<ul style="list-style-type: none"> • GrabarAlumno
Propósito	<ul style="list-style-type: none"> • Crear un alumno
Precondiciones	<ul style="list-style-type: none"> • No existe ningún alumnos creado en la base de datos
Resultado	<ul style="list-style-type: none"> • El método devuelve un valor entero 1 “Éxito”

Tabla 3.49: Prueba Unitaria GrabarAlumno

Responsable de la prueba	<ul style="list-style-type: none"> • José Castillo
Verificador	<ul style="list-style-type: none"> • Ing. Patricio Medina
Clase	<ul style="list-style-type: none"> • Servicio Web Cronos (servCronos)
Método	<ul style="list-style-type: none"> • GrabarExamen
Propósito	<ul style="list-style-type: none"> • Crear un Examen sin preguntas asociadas
Precondiciones	<ul style="list-style-type: none"> • No existe ningún examen creado en la base de datos
Resultado	<ul style="list-style-type: none"> • El método devuelve un valor entero 1 “Éxito”

Tabla 3.50: Prueba Unitaria GrabarExamen

Responsable de la prueba	<ul style="list-style-type: none"> • José Castillo
Verificador	<ul style="list-style-type: none"> • Ing. Patricio Medina
Clase	<ul style="list-style-type: none"> • Servicio Web Cronos (servCronos)
Método	<ul style="list-style-type: none"> • GrabarExamenPregunta
Propósito	<ul style="list-style-type: none"> • Crear un Examen con preguntas asociadas
Precondiciones	<ul style="list-style-type: none"> • Existe tres preguntas creadas
Resultado	<ul style="list-style-type: none"> • Error de conversión de tipo de dato "Error"

Tabla 3.51: Prueba Unitaria GrabarExamenPregunta

Responsable de la prueba	<ul style="list-style-type: none"> • José Castillo
Verificador	<ul style="list-style-type: none"> • Ing. Patricio Medina
Clase	<ul style="list-style-type: none"> • Servicio Web Cronos (servCronos)
Método	<ul style="list-style-type: none"> • ListarAlumnoByCedula
Propósito	<ul style="list-style-type: none"> • Recuperar la información de un alumno por su cédula de identidad
Precondiciones	<ul style="list-style-type: none"> • Existe un alumno creado en la base de datos • Existe un alumno con cédula "1104147440"
Resultado	<ul style="list-style-type: none"> • El método devuelve un registro "Éxito"

Tabla 3.52: Prueba Unitaria ListarAlumnoByCedula

Responsable de la prueba	<ul style="list-style-type: none"> • José Castillo
Verificador	<ul style="list-style-type: none"> • Ing. Patricio Medina
Clase	<ul style="list-style-type: none"> • Servicio Web Cronos (servCronos)
Método	<ul style="list-style-type: none"> • ObtenerUltimoIdExamen
Propósito	<ul style="list-style-type: none"> • Recuperar un valor entero.
Precondiciones	<ul style="list-style-type: none"> • Existe un examen creado en la base de datos con su ID = 1.
Resultado	<ul style="list-style-type: none"> • El método devuelve el valor 1 "Éxito"

Tabla 3.53: Prueba Unitaria ObtenerUltimoIdExamen

Responsable de la prueba	<ul style="list-style-type: none"> • José Castillo
Verificador	<ul style="list-style-type: none"> • Ing. Patricio Medina
Clase	<ul style="list-style-type: none"> • Servicio Web Cronos (servCronos)
Método	<ul style="list-style-type: none"> • ListarProfesionalByCedula
Propósito	<ul style="list-style-type: none"> • Recuperar la información de un docente por su cédula de identidad
Precondiciones	<ul style="list-style-type: none"> • Existe un docente creado en la base de datos • Existe un docente con cédula “1104147441”
Resultado	<ul style="list-style-type: none"> • El método devuelve un registro “Éxito”

Tabla 3.54: Prueba Unitaria ListarProfesionalByCedula

Responsable de la prueba	<ul style="list-style-type: none"> • José Castillo
Verificador	<ul style="list-style-type: none"> • Ing. Patricio Medina
Clase	<ul style="list-style-type: none"> • Servicio Web Cronos (servCronos)
Método	<ul style="list-style-type: none"> • ListarEscuela
Propósito	<ul style="list-style-type: none"> • Recuperar la información de una escuela.
Precondiciones	<ul style="list-style-type: none"> • Existe una escuela creada en la base de datos
Resultado	<ul style="list-style-type: none"> • El método devuelve un registro “Éxito”

Tabla 3.55: Prueba Unitaria ListarEscuela

Verificador	<ul style="list-style-type: none"> • Ing. Patricio Medina
Clase	<ul style="list-style-type: none"> • Servicio Web Cronos (servCronos)
Método	<ul style="list-style-type: none"> • ListarCatedraByIDEscuela
Propósito	<ul style="list-style-type: none"> • Recuperar la información de un docente perteneciente a una escuela.
Precondiciones	<ul style="list-style-type: none"> • Existe una escuela creada con su ID = 1 • Existe una cátedra creada en la base de datos asociada a la escuela con ID = 1
Resultado	<ul style="list-style-type: none"> • El método devuelve un registro “Éxito”

Tabla 3.56: Prueba Unitaria ListarCatedraByIDEscuela

Responsable de la prueba	<ul style="list-style-type: none"> • José Castillo
Verificador	<ul style="list-style-type: none"> • Ing. Patricio Medina
Clase	<ul style="list-style-type: none"> • Servicio Web Cronos (servCronos)
Método	<ul style="list-style-type: none"> • ListarPreguntaByIDCatedra
Propósito	<ul style="list-style-type: none"> • Recuperar la información de una pregunta.
Precondiciones	<ul style="list-style-type: none"> • Existe una cátedra creada en la base de datos con su ID = 1. • Existe tres preguntas creada en la base de datos asociadas a la cátedra con su ID = 1.
Resultado	<ul style="list-style-type: none"> • El método devuelve un registro “Éxito”

Tabla 3.57: Prueba Unitaria ListarPreguntaByIDCatedra

Responsable de la prueba	<ul style="list-style-type: none"> • José Castillo
Verificador	<ul style="list-style-type: none"> • Ing. Patricio Medina
Clase	<ul style="list-style-type: none"> • Servicio Web Cronos (servCronos)
Método	<ul style="list-style-type: none"> • ObtenerUltimoIdExamen
Propósito	<ul style="list-style-type: none"> • Recuperar la información de una pregunta.
Precondiciones	<ul style="list-style-type: none"> • Existe un examen creado en la base de datos con su ID =1.
Resultado	<ul style="list-style-type: none"> • El método devuelve el valor 1. “Éxito”

Tabla 3.58: Prueba Unitaria ObtenerUltimoIdExamen

Responsable de la prueba	<ul style="list-style-type: none"> • José Castillo
Verificador	<ul style="list-style-type: none"> • Ing. Patricio Medina
Clase	<ul style="list-style-type: none"> • Servicio Web Cronos (servCronos)
Método	<ul style="list-style-type: none"> • ListarRespuestaPracticaByIdPregunta
Propósito	<ul style="list-style-type: none"> • Recuperar la información de una pregunta.
Precondiciones	<ul style="list-style-type: none"> • Existe una pregunta creada en la base de datos con su ID = 2. • Existe una pregunta creada en la base de datos asociadas a la pregunta con su ID = 2.
Resultado	<ul style="list-style-type: none"> • El método devuelve un registro “Éxito”

Tabla 3.59: Prueba Unitaria ListarRespuestaPracticaByIdPregunta

Responsable de la prueba	<ul style="list-style-type: none"> • José Castillo
Verificador	<ul style="list-style-type: none"> • Ing. Patricio Medina
Clase	<ul style="list-style-type: none"> • Servicio Web Cronos (servCronos)
Método	<ul style="list-style-type: none"> • ListarRespuestaEnlaceByIDPregunta
Propósito	<ul style="list-style-type: none"> • Recuperar la información de una pregunta.
Precondiciones	<ul style="list-style-type: none"> • Existe una pregunta creada en la base de datos con su ID = 3. • Existe tres respuestas de enlace creada en la base de datos asociadas a la pregunta con su ID = 3.
Resultado	<ul style="list-style-type: none"> • El método devuelve tres registros “Éxito”

Tabla 3.60: Prueba Unitaria ListarRespuestaEnlaceByIDPregunta

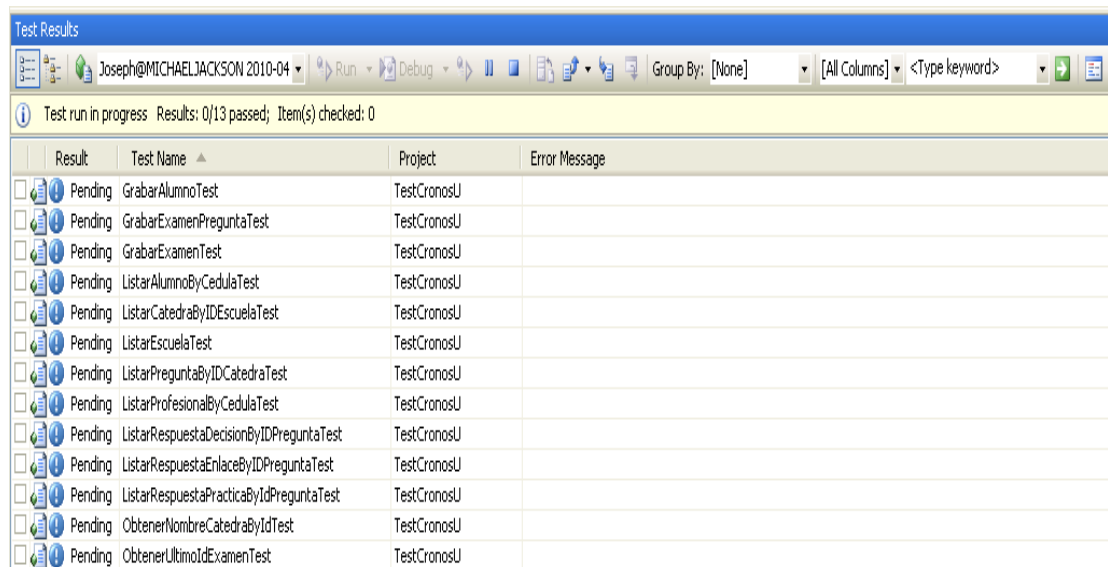
Responsable de la prueba	<ul style="list-style-type: none"> • José Castillo
Verificador	<ul style="list-style-type: none"> • Ing. Patricio Medina
Clase	<ul style="list-style-type: none"> • Servicio Web Cronos (servCronos)
Método	<ul style="list-style-type: none"> • ListarRespuestaDecisionByIDPregunta
Propósito	<ul style="list-style-type: none"> • Recuperar la información de una pregunta.
Precondiciones	<ul style="list-style-type: none"> • Existe una pregunta creada en la base de datos con su ID = 1. • Existe tres respuestas de decisión creadas en la base de datos asociadas a la pregunta con su ID = 1.
Resultado	<ul style="list-style-type: none"> • El método devuelve tres registros “Éxito”

Tabla 3.61: Prueba Unitaria ListarRespuestaDecisionByIDPregunta

Responsable de la prueba	<ul style="list-style-type: none"> • José Castillo
Verificador	<ul style="list-style-type: none"> • Ing. Patricio Medina
Clase	<ul style="list-style-type: none"> • Servicio Web Cronos (servCronos)
Método	<ul style="list-style-type: none"> • ObtenerNombreCatedraById
Propósito	<ul style="list-style-type: none"> • Recuperar el nombre de una cátedra.
Precondiciones	<ul style="list-style-type: none"> • Existe un cátedra creada en la base de datos con su nombre “SUFICIENCIA EN INFORMATICA”
Resultado	<ul style="list-style-type: none"> • El método devuelve el valor “SUFICIENCIA EN INFORMATICA” “Éxito”

Tabla 3.62: Prueba Unitaria ObtenerNombreCatedraById

Visual Studio 2008 nos proporciona un sistema sencillo para poder crear pruebas unitarias que facilita la creación de los proyectos de pruebas y nos genera una estructura básica que tiene una prueba, el gráfico 3.65 muestra la ejecución de las pruebas unitarias.



Test Results

Joseph@MICHAELJACKSON 2010-04 Run Debug

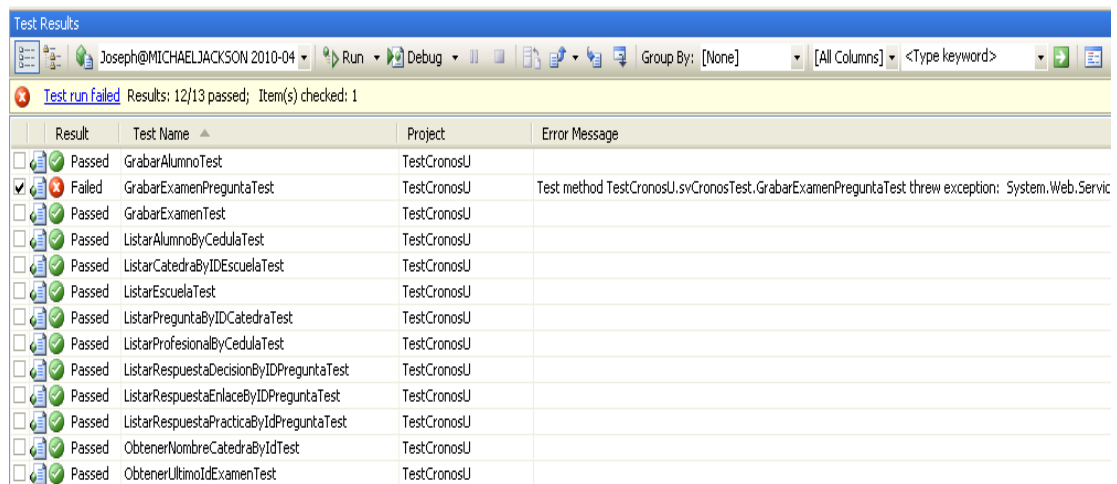
Group By: [None] [All Columns] <Type keyword>

Test run in progress Results: 0/13 passed; Item(s) checked: 0

Result	Test Name	Project	Error Message
Pending	GrabarAlumnoTest	TestCronosU	
Pending	GrabarExamenPreguntaTest	TestCronosU	
Pending	GrabarExamenTest	TestCronosU	
Pending	ListarAlumnoByCedulaTest	TestCronosU	
Pending	ListarCatedraByIDEscuelaTest	TestCronosU	
Pending	ListarEscuelaTest	TestCronosU	
Pending	ListarPreguntaByIDCatedraTest	TestCronosU	
Pending	ListarProfesionalByCedulaTest	TestCronosU	
Pending	ListarRespuestaDecisionByIDPreguntaTest	TestCronosU	
Pending	ListarRespuestaEnlaceByIDPreguntaTest	TestCronosU	
Pending	ListarRespuestaPracticaByIDPreguntaTest	TestCronosU	
Pending	ObtenerNombreCatedraByIdTest	TestCronosU	
Pending	ObtenerUltimoIdExamenTest	TestCronosU	

Gráfico 3.65: Ejecución de pruebas unitarias

En el gráfico 3.66 podemos observar los resultados de las pruebas unitarias.



Test Results

Joseph@MICHAELJACKSON 2010-04 Run Debug

Group By: [None] [All Columns] <Type keyword>

Test run failed Results: 12/13 passed; Item(s) checked: 1

Result	Test Name	Project	Error Message
Passed	GrabarAlumnoTest	TestCronosU	
Failed	GrabarExamenPreguntaTest	TestCronosU	Test method TestCronosU.svCronosTest.GrabarExamenPreguntaTest threw exception: System.Web.Servic
Passed	GrabarExamenTest	TestCronosU	
Passed	ListarAlumnoByCedulaTest	TestCronosU	
Passed	ListarCatedraByIDEscuelaTest	TestCronosU	
Passed	ListarEscuelaTest	TestCronosU	
Passed	ListarPreguntaByIDCatedraTest	TestCronosU	
Passed	ListarProfesionalByCedulaTest	TestCronosU	
Passed	ListarRespuestaDecisionByIDPreguntaTest	TestCronosU	
Passed	ListarRespuestaEnlaceByIDPreguntaTest	TestCronosU	
Passed	ListarRespuestaPracticaByIDPreguntaTest	TestCronosU	
Passed	ObtenerNombreCatedraByIdTest	TestCronosU	
Passed	ObtenerUltimoIdExamenTest	TestCronosU	

Gráfico 3.66: Resultado de pruebas unitarias I

La prueba unitaria GrabarExamenPreguntaTest retorno un error de conversión de datos ubicado en la capa de acceso a datos. Ver gráfico 3.64

Common Results

Test Run: Joseph@MICHAELJACKSON 2010-04-17 22:40:05
 Test Name: GrabarExamenPreguntaTest
 Result: Failed
 Duration: 00:00:00.1860116
 Computer Name: MICHAELJACKSON
 Start Time: 17/04/2010 22:40:08
 End Time: 17/04/2010 22:40:09

Error Message

Test method TestCronosU.svCronosTest.GrabarExamenPreguntaTest threw exception: System.Web.Services.Protocols.SoapException: System.Web.Services.Protocols.SoapException: Server was unable to process request. ---> System.Data.SqlClient.SqlException: Arithmetic overflow error converting numeric to data type numeric. The statement has been terminated. at Pucesa.Cronos.AccesoDatos.ExamenPreguntaDALC.Update(ExamenPreguntaDST dsExamenPregunta, DbTransaction trans) in C:\CronosSh\AccesoDatos\dAccesoDatos\ExamenPreguntaDALC.cs:line 108 at Pucesa.Cronos.Logica.ExamenPreguntaLog.Update(ExamenPreguntaDST dsExamenPregunta) in C:\CronosSh\Logica\logico(LogicaNegocio)\ExamenPreguntaLog.cs:line 65 at wsCronos.svCronos.GrabarExamenPregunta(ExamenPreguntaDST dsExamenPregunta) in C:\CronosSh\Servicios\wsCronos\svCronos.asmx.cs:line 295 End of inner exception stack trace ---

Error Stack Trace

System.Web.Services.Protocols.SoapHttpClientProtocol.ReadResponse(SoapClientMessage message, WebResponse response, Stream responseStream, Boolean asyncCall)
 System.Web.Services.Protocols.SoapHttpClientProtocol.Invoke(String methodName, Object[] parameters)
 TestCronosU.svCronos.svCronos.GrabarExamenPregunta(ExamenPreguntaDST dsExamenPregunta) C:\CronosSh\TestCronos\Unitarias\TestCronosU\Web References\svCronos\Reference.cs : line 1500
 TestCronosU.svCronosTest.GrabarExamenPreguntaTest() C:\CronosSh\TestCronos\Unitarias\TestCronosU\svCronosTest.cs : line 270

Gráfico 3.67: Error de prueba

Una vez corregido el error se procedió a someter una nueva prueba unitaria, teniendo una prueba satisfactoria.

Result	Test Name	Project	Error Message
Passed	GrabarAlumnoTest	TestCronosU	
Passed	GrabarExamenPreguntaTest	TestCronosU	
Passed	GrabarExamenTest	TestCronosU	
Passed	ListarAlumnoByCedulaTest	TestCronosU	
Passed	ListarCatedraByIDEscuelaTest	TestCronosU	
Passed	ListarEscuelaTest	TestCronosU	
Passed	ListarPreguntaByIDCatedraTest	TestCronosU	
Passed	ListarProfesionalByCedulaTest	TestCronosU	
Passed	ListarRespuestaDecisionByIDPreguntaTest	TestCronosU	
Passed	ListarRespuestaEnlaceByIDPreguntaTest	TestCronosU	
Passed	ListarRespuestaPracticaByIDPreguntaTest	TestCronosU	
Passed	ObtenerNombreCatedraByIdTest	TestCronosU	
Passed	ObtenerUltimoIdExamenTest	TestCronosU	

Gráfico 3.68: Resultado de pruebas unitarias II

3.3.4.2 Pruebas de aceptación

Las pruebas de aceptación tienen como fin validar que el sistema cumple los requisitos básicos de funcionamiento esperado y permitir que el usuario determine la aceptación del sistema.

Por este motivo, estas pruebas son realizadas por el usuario final que, durante este periodo de tiempo, debe plantear todas las deficiencias o errores que encuentre antes de dar por aprobado el sistema definitivamente.

Tomando como base a los requerimientos funcionales del sistema se evaluaron los siguientes procesos:

- Creación y registro de escuelas.
- Creación y registro de docentes y alumnos.
- Asignación de roles a usuarios.
- Registro del Banco de preguntas
- Elaboración de exámenes.
- Tomar exámenes con preguntas de decisión, enlace y prácticas de MS Office

Se definieron cinco categorías que fueron utilizadas como resultado de prueba:

- Conforme: el resultado obtenido es conforme al resultado esperado.
- Aceptable: el resultado obtenido indica que el sistema difiere de la especificación pero es aceptable, requiriendo cambio no en el sistema sino en la especificación.
- Tolerable: el resultado obtenido es incorrecto, el sistema funciona y es aceptable pero se deberá definir un tiempo para corregir la falla encontrada.

- Intolerable: el resultado obtenido es incorrecto y la falta debe ser corregida antes de que el sistema culmine la fase de pruebas.
- Error en Prueba: el resultado obtenido es correcto pero el resultado esperado que se especifica en el caso de prueba es incorrecto.

Pruebas de aceptación			
Responsable: José Castillo			
Verificador: Ing. Santiago Acurio			
Módulo: Administrador			
Fecha	Descripción	Defecto corregido	Estado
03-04-10	Errores de ortografía en los textos	X	Conforme
03-04-10	Errores en colores de texto	X	Conforme
03-04-10	Errores en tamaño de texto	X	Conforme
03-04-10	Errores en tamaños de pantallas	X	Conforme
03-04-10	Errores en ubicación de botones	X	Conforme

Tabla 3.63: Prueba Aceptación Modulo Administrador

Pruebas de aceptación			
Responsable: José Castillo			
Verificador: Ing. Santiago Acurio			
Módulo: Examinador			
Fecha	Descripción	Defecto corregido	Estado
04-04-10	Errores de crear perfiles de usuarios	X	Conforme
04-02-09	Errores de ortografía en reportes	X	Conforme
04-07-09	Errores de cerrar aplicación MS Word	X	Conforme
04-07-09	Errores de cerrar aplicación MS PowerPoint	X	Conforme
04-07-09	Errores de cerrar aplicación MS Excel	X	Conforme

Tabla 3.64: Prueba Aceptación Modulo Examinador

Finalmente CRONOS fue calificado como aceptable para su pase a su utilización y culminó con éxito la fase de pruebas de aceptación cuando:

- Todos los procesos de evaluación fueron probados.
- Todos los casos de prueba se efectuaron con éxito.

3.3.4.3 Pruebas de carga

Las pruebas de carga se componen de una serie de pruebas Web o pruebas unitarias que funcionan durante un período de tiempo bajo varios usuarios simulados.

Normalmente este tipo de pruebas están basadas en las aplicaciones bajo cargas pesadas, generalmente usadas en sitios web y en servidores con gran cantidad de datos donde se determina en cuales puntos existen degradaciones del sistema.

El propósito principal de esta prueba de carga es simular el acceso de muchos usuarios a un servidor donde se accede a un servicio web desde un cliente Windows para tomar un examen. La siguiente tabla contiene los datos necesarios para llevar a cabo este tipo de prueba

Responsable de la prueba	<ul style="list-style-type: none"> • José Castillo
Verificador	<ul style="list-style-type: none"> • Ing. Patricio Medina
Escenario	<ul style="list-style-type: none"> • Pucesa
Método	<ul style="list-style-type: none"> • ListarAlumnoByCedula • ListarExamenPreguntaByIdExamen
Precondiciones	<ul style="list-style-type: none"> • Escenario: Equipo de desarrollo • Usuarios: Diez usuarios conectados al servidor Web • Tiempo: Dos minutos

Tabla 3.65: Prueba de carga

Microsoft Visual Studio permite crear y ejecutar pruebas de carga, a continuación los resultados de la prueba de carga.

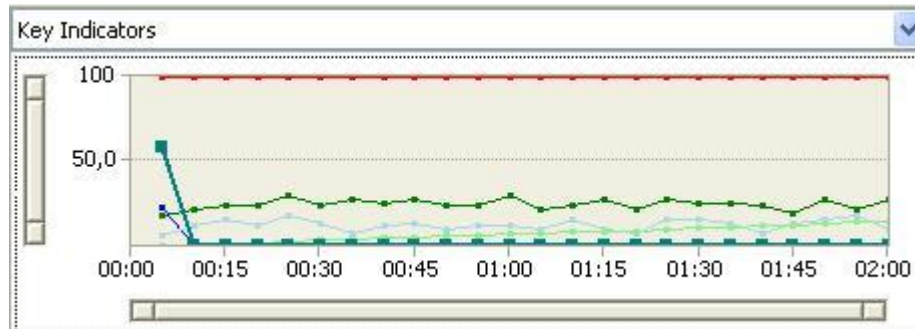


Gráfico 3.69: Prueba de carga, ListarAlumnoByIdCedula

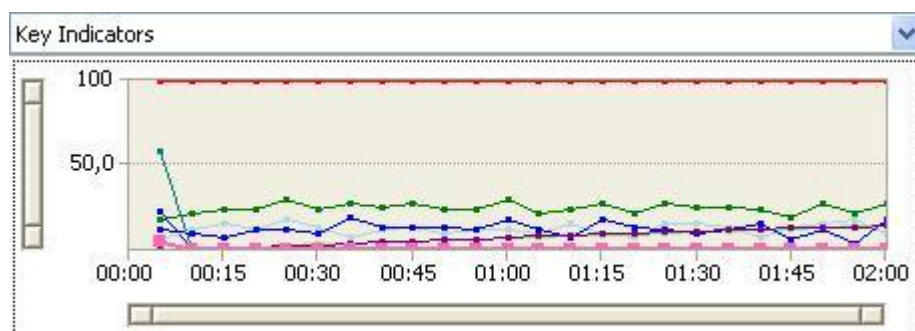


Gráfico 3.70: Prueba de carga, ListarExamenPreguntaByIdExamenTest

Los datos que se muestran en el gráfico 3.71 contienen información sobre el nombre de contador, de la categoría, rango, mínimo, máximo y promedio de lecturas para cada contador.

Counter	Instance	Category	Computer	Color	Range	Min	Max	Avg
<input checked="" type="checkbox"/> Avg. Test Time	ListarAlumnoBy...	LoadTest:Test	MICHAELJACK...		1	0,0059	0,59	0,019
<input checked="" type="checkbox"/> Failed Tests	ListarAlumnoBy...	LoadTest:Test	MICHAELJACK...		0	0	0	0
<input checked="" type="checkbox"/> Failed Tests/Sec	ListarAlumnoBy...	LoadTest:Test	MICHAELJACK...		0	0	0	0
<input checked="" type="checkbox"/> Passed Tests	ListarAlumnoBy...	LoadTest:Test	MICHAELJACK...		1.000	3	151	77
<input checked="" type="checkbox"/> Passed Tests/Sec	ListarAlumnoBy...	LoadTest:Test	MICHAELJACK...		10	0,60	1,80	1,26
<input checked="" type="checkbox"/> Tests/Sec	ListarAlumnoBy...	LoadTest:Test	MICHAELJACK...		10	0,60	1,80	1,26
<input checked="" type="checkbox"/> Total Tests	ListarAlumnoBy...	LoadTest:Test	MICHAELJACK...		1.000	3	151	77
<input checked="" type="checkbox"/> Avg. Test Time	ListarExamenP...	LoadTest:Test	MICHAELJACK...		1	0,0045	0,056	0,0080
<input checked="" type="checkbox"/> Failed Tests	ListarExamenP...	LoadTest:Test	MICHAELJACK...		0	0	0	0
<input checked="" type="checkbox"/> Failed Tests/Sec	ListarExamenP...	LoadTest:Test	MICHAELJACK...		0	0	0	0
<input checked="" type="checkbox"/> Passed Tests	ListarExamenP...	LoadTest:Test	MICHAELJACK...		1.000	6	149	78
<input checked="" type="checkbox"/> Passed Tests/Sec	ListarExamenP...	LoadTest:Test	MICHAELJACK...		10	0,40	2,00	1,24
<input checked="" type="checkbox"/> Tests/Sec	ListarExamenP...	LoadTest:Test	MICHAELJACK...		10	0,40	2,00	1,24
<input checked="" type="checkbox"/> Total Tests	ListarExamenP...	LoadTest:Test	MICHAELJACK...		1.000	6	149	78

Gráfico 3.71: Gráfico Prueba de carga (contadores)

Esta sección detalla la lista de errores que se produjeron durante la prueba. Proporciona información como el tipo de error, el subtipo, y el número de veces el mismo error se produjo durante la prueba, el último mensaje de la pila de errores, y así sucesivamente Ver gráfico 3.69

Test Completed	
Load Test Summary	
Test Run Information	
Load test name	LoadTest1
Description	
Start time	21/04/2010 23:26:51
End time	21/04/2010 23:28:51
Warm-up duration	00:00:00
Duration	00:02:00
Controller	Local run
Number of agents	1
Run settings used	Run Settings1
Overall Results	
Max User Load	10
Requests/Sec	-
Requests Failed	-
Requests Cached Percentage	-
Avg. Response Time (sec)	-
Avg. Content Length (bytes)	-
Tests/Sec	2,50
Tests Failed	0
Avg. Test Time (sec)	0,014
Avg. Transaction Time (sec)	0
Avg. Page Time (sec)	-
Key Statistic: Top 5 Slowest Tests	
Name	Avg. Test Time (sec)
ListarAlumnoByCedulaTest	0,019
ListarExamenPreguntaByIdExamenTest	0,0080

Gráfico 3.72: Prueba de carga (resultados)

3.3.5 Implementación

Una vez concluido con las pruebas realizadas al sistema distribuido está listo para su implementación y es necesario recordar los componentes o elementos que forman al sistema distribuido:

- Una base de datos que manejará la información necesaria del sistema
- Dos servicios web que encapsulan el acceso a datos a la base de datos y que se encargará de proveer los servicios de negocio del sistema.
- Dos aplicaciones Windows que serán las responsables de proveer la interface de usuario para los alumnos y docentes de la PUCESA, estas aplicaciones consumirán los servicios web para realizar todas las operaciones de seguridad y administración del sistema.

Para implementar el sistema CRONOS se requiere de tres, dos o al menos un servidor para alojar los servicios web y la base de datos, y terminales para las aplicaciones Windows, el gráfico 3.73 muestra la estructura física ideal para la implementación del sistema.

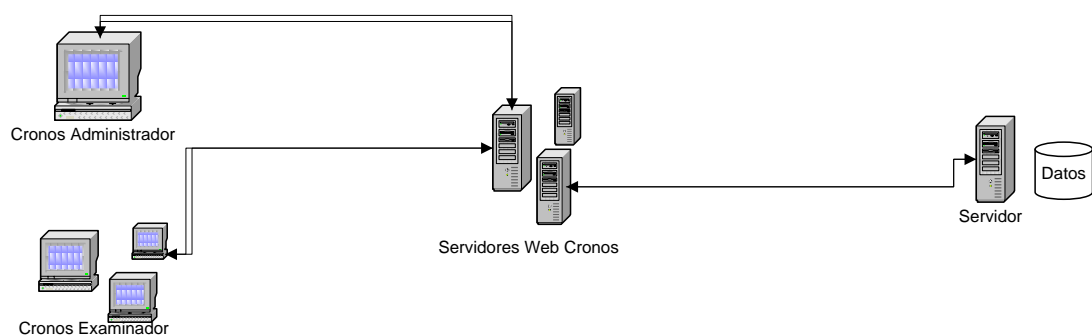


Gráfico 3.73: Implementación (Estructura física para la implementación del sistema)

Es importante indicar que, sólo se puede utilizar los servicios de los web services para acceder a la información que maneja el sistema, porque la base de datos no tiene acceso directo a las aplicaciones Windows.

Para tener una perspectiva aún más clara del sistema distribuido CRONOS, el gráfico 3.74 muestra los elementos en tres niveles interactuando entre sí.

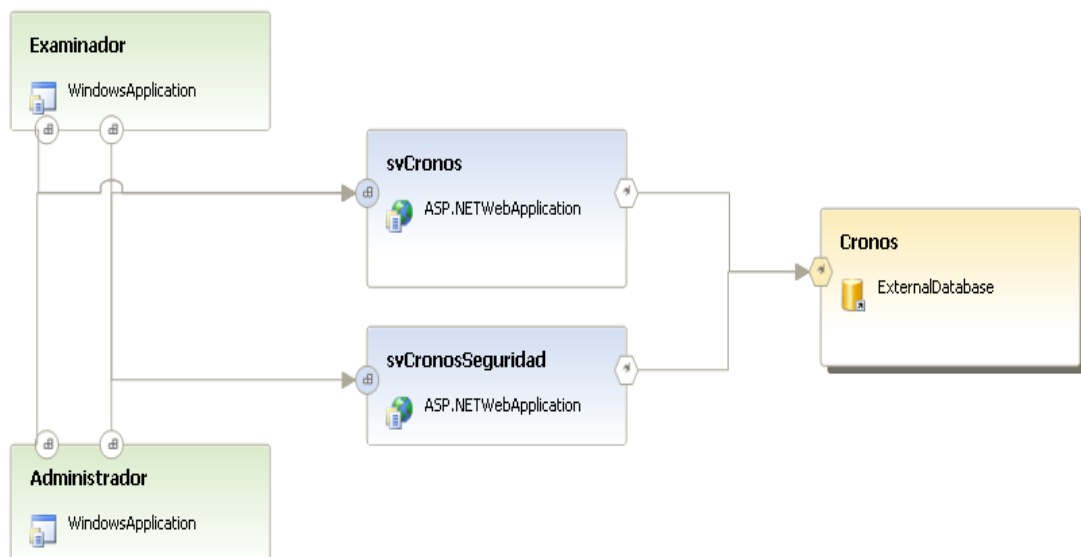


Gráfico 3.74: Implementación (Elementos del sistema)

3.3.5.1 Requerimientos de hardware y software del equipo de implementación

Los requerimientos del hardware y software que se necesitarán para la implementación del sistema se detallan en las siguientes tablas:

Servidor de base de datos

Dispositivo	Características
Procesador	Intel core duo, Velocidad 2.66 GHz
Memoria RAM	2GB
Disco Duro	80 GB
Software	Versión
Sistema Operativo	Microsoft Windows XP Service Pack 3 o superior
Programas	MS SQL Server 2000 o superior

Tabla 3.66: Requerimientos de hardware y software para el Servidor BDD

Servidor Web	
Dispositivo	Características
Procesador	Intel core duo, Velocidad 2.66 GHz
Memoria RAM	2GB
Disco Duro	80 GB
Software	Versión
Sistema Operativo	Microsoft Windows XP Service Pack 3 o superior
Programas	Internet Information Sever 5 o Superior Framework 3.5 Net

Tabla 3.67: Requerimientos de hardware y software del Servidor Web

Computador personal	
Dispositivo	Características
Procesador	Intel core duo, Velocidad 2.66 GHz
Memoria RAM	1 GB
Disco Duro	40GB
Software	Versión
Sistema Operativo	Microsoft Windows XP Service Pack 3 o superior
Programas	MS office Framework 3.5

Tabla 3.68: Requerimientos de hardware y software de PC

3.3.5.2 Documentación

El personal de la Escuela de Sistemas contará con un manual de ayuda. Este manual está clasificado en diferentes secciones que explican las características y funcionalidades del sistema. Además, cuenta con ilustraciones de las interfaces del sistema de modo que el usuario puede identificar claramente los pasos que debe seguir para realizar cada funcionalidad. Ver Anexo 3.

El manual de ayuda considera los siguientes puntos:

Descripción General del Sistema: explicación general de la utilidad de uso del sistema.

Descripción detallada de los módulos del sistema

- ✓ Módulo CRONOS Administrador
- ✓ Módulo de CRONOS Examinador

CAPITULO IV

4 CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

Al finalizar la elaboración de la presente disertación, se determinaron las siguientes conclusiones:

El desarrollo de sistemas distribuidos, permite ir separando física y lógicamente su estructura, de esta manera se mejora tanto su rendimiento como su eficiencia, optimizando recursos y sobrecarga de flujo de datos.

Se ha cumplido con el objetivo de desarrollar el sistema distribuido CRONOS que permite de manera sencilla y segura la generación y realización de evaluaciones diagnósticas y prácticas para las cátedras impartidas en la PUCESA especialmente para "Suficiencia en Informática".

El sector estudiantil se inclina mayoritariamente por la utilización de un sistema de evaluación y el personal docente de la escuela de sistemas de la PUCESA utilizara el sistema como una herramienta, aprovechando todo el potencial que este sistema ofrece en el proceso de enseñanza aprendizaje y particularmente en la evaluación de exámenes.

4.2 RECOMENDACIONES

Se recomienda promover la participación de los docentes en el uso de las nuevas tecnologías de la información que provoque en él, deseo de aprender y utilizar nuevos recursos didácticos en el aula, promover el uso de sistemas que evalúen exámenes personalizados y automatizados.

Es conveniente que el docente, dejando su forma tradicional de evaluar se involucre en la utilización de sistemas para pruebas existentes; así como en la generación de pruebas aleatorias utilizando sistemas informáticos, que le permitan personalizar este instrumento de evaluación.

Extender el uso del sistema CRONOS, no limitándolo a un sistema de evaluación, sino también como una fuente de tareas dirigidas, utilizables ya sea como autoevaluaciones o como trabajos de extra clase, es decir, que gracias a la arquitectura distribuida de CRONOS se podrá ir incorporando nuevos módulos y es aquí cuando los estudiantes entran a familiarizarse con el desarrollo de sistemas distribuidos.

Capacitar al docente en el uso del sistema CRONOS, para que se facilite su manejo y utilización en la elaboración de los exámenes e incentivar a las demás escuelas de la PUCESA al uso del sistema CRONOS.

Material de referencia

Bibliografía

- Díaz Barriga F, Hernández G. "Estrategias docentes para un aprendizaje significativo". 1ª ed. México: McGraw Hill. 1998.
- McDougall D., Squires D. "Como elegir y utilizar software educativo" 1ª ed. Madrid: Ediciones Morata. 1997.
- Gabillaud Gerome. "SQL Transact SQL: Diseño y creación de una base de datos". 1ª ed. Paris: ENI. 2009.
- Robayo Jose Felix "Redes Locales: "Guía Práctica". 1ª ed. Madrid: Anaya Multimedia. 2008.
- Hernández Leal Octavio "C# 3.0 y Linq" 1ª Ed. La Coruña: IDEASPROPIAS 2008.
- Booch G, Rumbaugh J. y Jacobson I. "El lenguaje unificado de modelado Guía del usuario". 2ª ed. Madrid: Pearson Education, 2006.
- Arlow J. y Neustadt I. "UML and the unified process: practical object-oriented analysis and design". Boston: Pearson Education. 2002
- Colouris, George;Dollimore, Jeanand Kindberg,Tim. "Sistemas Distribuidos. Conceptos y Diseño". 1ª ed. Madrid: Iberoamericana España, 2001
- Tanenbaum A, Andrew S. "Sistemas Distribuidos: Principios y Paradigmas", 1ª ed. México: Prentice Hall. 2008

Sitios Web

- OpenCourseWare Universidad de Sevilla, Cristina Granado Alonso y Soledad Domene Martos, 03 /Febrero/ 2007, Fundación William And Flora Hewlett, 10/Agosto/2009.
<http://ocwus.us.es/didactica-y-organizacion-escolar/procesos-de-ensenanza-aprendizaje/Course_listing>
- Maixmail.com Cursos para compartir lo que sabes, Abdiel Cáceres Gonzáles, 10/Julio/2009, Opening Learning, 01/Septiembre/2009
<<http://www.mailxmail.com/curso-informatica-ordenador-cientifico-desarrollo/objetos-oop-programacion-orientada>>
- Monografias.com, Edsel Enrique Uruña León, 03/Enero/2005, 10/Septiembre/2009
<<http://www.monografias.com/trabajos30/redes-de-datos/redes-de-datos.shtml>>
- *EDUTEKA*, EDUTEKA, 01/Marzo/2008, Fundacion Gabriel Piedrahita Uribe, 28/Septiembre/2009
<<http://www.eduteka.org/RedEscolarDatos.php>>
- Maixmail.com Cursos para compartir lo que sabes, Raúl Fernández Aedo, 08/Abril/2009, Opening Learning, 01/Septiembre/2009
<<http://www.mailxmail.com/curso-sistemas-informativos-diseno-creacion-1/sistemas-informaticos-distribuidos-concepto-empresa>>
- Sistemas Distribuidos, Alberto Lafuente, 01/ Marzo/2009, Universidad del País Vasco, 15/Octubre /2009
<<http://www.sc.ehu.es/acwlaroa/SDI/Apuntes/Cap1.pdf>>

GLOSARIO

.NET

Es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años para el desarrollo y ejecución del software en forma de servicios que puedan ser tanto publicados como accedidos a través de Internet de forma independiente del lenguaje de programación,

.NET Framework

Componente integral de Windows que admite la creación, implementación y la ejecución de la siguiente generación de aplicaciones y servicios Web XML.

AppleTalk

Es un protocolo propietario que se utiliza para conectar computadoras Macintosh de Apple en redes locales.

Backbone

Es el trecho de mayor capacidad en la red y es donde se conectan varias redes locales.

Base de Datos

Conjunto de información organizada, y relacionada entre si.

Broadcast

Transmisión abierta. Mensajes que se mandan sin destino

BUS

Circuito de interconexión eléctrica para transmitir

Dataset

Se trata de un grupo de datos listo para entrar o salir de la base, justo cuando puede que esté retenido en la zona de Entrada o en la de Salida

CORBA (Common Object Request Broker Architecture)

Es un modelo de componentes estándar para permitir la interoperabilidad entre Componentes de diferentes plataformas, lenguajes, autores o vendedores.

COM (Component Object Model)

Modelo de componentes de Microsoft. Define un estilo arquitectónico para componentes software de Microsoft.

Hardware

Parte física de una computadora

Hub

Un hub es un equipo de redes que permite conectar entre sí otros equipos o dispositivos retransmitiendo paquetes de datos desde cualquiera de ellos hacia todos los demás.

Internet Information Server (IIS)

Es el servidor web de Microsoft con normas de seguridad.

Interface Definition Language (IDL)

Lenguaje para la definición de interfaces. Un IDL ofrece una descripción precisa de los tipos, métodos y excepciones de una interfaz de componente. Los objetos CORBA pueden tener múltiples interfaces compuestas o heredadas de otras.

Linq

Lenguaje Integrado de Consultas de la plataforma .NET

Middleware

Software de interconexión, que proporciona interoperabilidad entre servicios de aplicaciones, como computación de objetos distribuidos, y resuelve algunos aspectos de hardware y diferencias de sistemas operativos en un entorno heterogéneo y distribuido.

Protocolo de red

Es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red

Remote Procedure Call (RPC)

Es una infraestructura cliente/servidor que incrementa la interoperabilidad, portabilidad y flexibilidad de una aplicación permitiendo que esta pueda ser distribuida sobre múltiples plataformas heterogéneas.

Requisito funcional

Algo que el componente software debe hacer.

Requisito no funcional

Una propiedad o cualidad que un componente debe tener.

Servidor

PC configurado con un sistema operativo de servidor, que contiene uno o más de los siguientes componentes periféricos: uno o dos Adaptadores de Host LAN, una o más Tarjetas de Interfaz de LAN (NIC), tarjetas de vídeo, etc.

Software

Parte lógica de una computadora

SOAP (Simple Object Access Protocol)

Protocolo simple basado en XML para intercambiar información estructurada y de tipos en el Web. El protocolo no contiene semántica de aplicación ni de transporte, por lo que resulta muy modular y extensible

SGBD (Sistema Gestor Base Datos)

Interfaz de software, es el responsable de tratar todas las peticiones de información de los usuarios, permitiendo que cada usuario tenga una visión individual de los datos almacenados.

TCP/IP

Protocolo de control de transmisión, protocolo de internet abierto.

Topología

Disposición lógica de una red, donde se muestran los nodos y conmutadores directamente conectados.

Token Ring

Red local en la que el permiso para transmitir es secuencial o en anillo.

Tupla

Dentro de la base de clientes, se refiere a un sólo registro relacionado (por ejemplo) con un individuo o un bien. Los datos que contienen las filas quedan almacenados como grupo de columnas.

Transact SQL

SQL es un lenguaje de consulta para los sistemas de bases de datos relacionales, pero que no posee la potencia de los lenguajes de programación.

UML (Unified Modeling Language)

Lenguaje de modelización de sistemas software desarrollado por OMG

UDDI (Universal Description, Discovery, and Integration)

Es una especificación de una función de directorio para la publicación, localización y enlace de servicios web. En su elaboración participan IBM, Microsoft y Ariba.

Web Service

Unidades de lógica de aplicaciones que proporcionan datos y servicios a otras aplicaciones. Las aplicaciones obtienen acceso a los servicios Web XML mediante protocolos Web estándar y formatos de datos como HTTP, XML y SOAP, con independencia de cómo se implementa cada servicio Web XML. Los servicios Web XML combinan los mejores aspectos del desarrollo basado en componentes y el Web, por lo que son una base fundamental del modelo de programación de Microsoft .NET.

WSDL (Web Services Description Language)

Lenguaje de definición de servicios web propuesto conjuntamente por Microsoft e IBM.

XML (Lenguaje de marcado extensible)

Subconjunto del Lenguaje de marcado generalizado estándar (SGML) optimizado para su uso a través del Web. XML proporciona un método uniforme para describir e intercambiar datos estructurados que es independiente de las aplicaciones o los proveedores.

ANEXOS

Anexo 1

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE AMBATO
ESCUELA DE INGENIERIA DE SISTEMAS

Distinguido docente en proyecciones de buscar soluciones con el uso de un sistema de evaluación en el proceso enseñanza-aprendizaje, le solicito de la manera más comedida se digne responder a la siguiente encuesta de forma precisa y sincera, su aporte será de mucha ayuda para el investigador.

1. ¿En su práctica docente ha utilizado herramientas computacionales?

Si _____

No _____

¿Por qué?

2. ¿Sus pruebas evaluativas son elaboradas con el uso del computador?

Si _____

No _____

¿Por qué?

3. ¿El uso del computador en su práctica docente es?

Procesador de Texto _____

Herramienta de Análisis _____

Fuente de consulta _____

Material Didáctico _____

Distracción _____

No lo usa _____

4. ¿Conoce algún sistema de evaluación?

Si _____

No _____

¿Si su respuesta es sí, puede mencionar algunos de ellos y como lo adquirió?

Internet _____

Compro _____

Facilitaron _____

Copio _____

5. ¿El número de estudiantes a su cargo le permite elaborar pruebas personalizadas?

Si _____

No _____

¿Por qué?

6. ¿Las pruebas generalizadas bajo su criterio provocan la copia entre sus alumnos?

Si _____

No _____

¿Por qué?

7. ¿Cree usted que realizar pruebas aleatorias e individuales evitaría el fraude?

Si _____

No _____

¿Por qué?

8. ¿Estaría usted dispuesto a elaborar manualmente pruebas individuales?

Si _____

No _____

¿Por qué?

9. ¿De disponer de un sistema de exámenes personalizados, utilizaría el mismo en el proceso de enseñanza-aprendizaje?

Si _____

No _____

¿Por qué?

10. ¿Si se aplicara en la escuela de sistemas un sistema informático de pruebas personalizadas, cree que se mejoraría el control de evaluación?

Si _____

No _____

¿Por qué?

11. ¿Le gustaría dominar el funcionamiento de un sistema de evaluación?

Si _____

No _____

¿Por qué?

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR SEDE AMBATO
ESCUELA DE INGENIERIA DE SISTEMAS

Señor Estudiante, solicito a Ud. comedidamente se digne responder a la siguiente encuesta en forma precisa y sincera, su aporte será de mucha ayuda para los encuestadores y sus resultados permitirán contribuir al mejoramiento académico de la PUCESA.

1. ¿Qué porcentaje de docentes utilizan sistemas o herramientas computacionales en el desarrollo de sus evaluaciones?

100% _____ 80% _____ 50% _____ 30% _____ 10% _____

2. ¿Los docentes han utilizado algún software de evaluación en el proceso de aprendizaje?

Si _____

No _____

3. Considera que en el proceso de aprendizaje, sus conocimientos son: medidos o evaluados

Medidos _____

Evaluados _____

4. ¿Desearía que cambien o al menos se modifiquen las modalidades de evaluación?

Si _____

No _____

¿Por qué?

5. ¿Considera usted que existe unidad de criterios entre los docentes de la PUCESA para aplicar la evaluación?

Si _____

No _____

¿Por qué?

6. ¿Le gustaría que los docentes utilicen un sistema informático de evaluación en el proceso de enseñanza?

Si _____

No _____

¿Por qué?

7 ¿Considera que la autoevaluación ayuda a la toma de conciencia del alumno?

Si _____

No _____

¿Por qué?

Anexo 2

1. ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

1.1. Resultado de encuestas

Para poder cumplir con los objetivos planteados en este proyecto, se elaboraron cuestionarios tanto para los docentes como para los estudiantes de la Escuela de sistemas de la PUCESA.

PREGUNTAS DIRIGIDAS A LOS DOCENTES:

Pregunta 1

¿En su práctica docente ha utilizado sistemas computacionales? (sí, no)

CARACTERISTICA	FRECUENCIA	PORCENTAJE
SI	7	70
NO	3	30
TOTAL	10	100

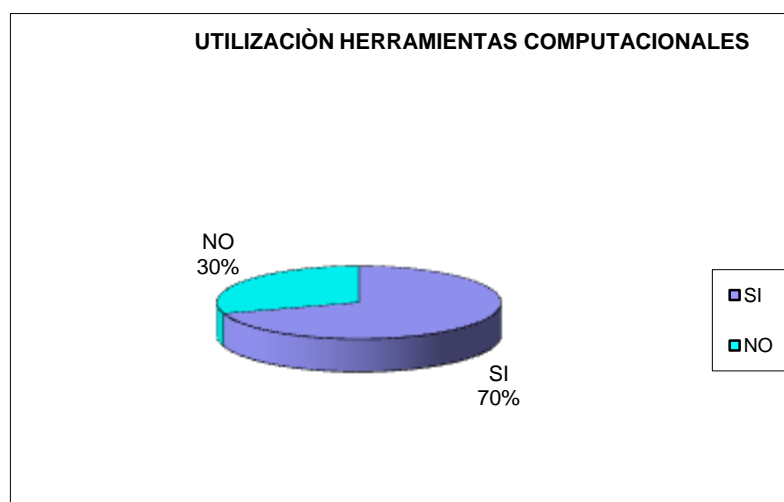


Gráfico 1: Uso de sistemas computacionales.

La investigación muestra que el setenta y cinco por ciento de los docentes han utilizado de alguna manera la computadora en su práctica docente, mientras que el veinticinco por ciento indica que no lo hace.

Pregunta 2

¿Sus pruebas evaluativas son elaboradas con el uso del computador? (sí, no)

CARACTERISTICA	FRECUENCIA	PORCENTAJE
SI	8	80
NO	2	20
TOTAL	10	100

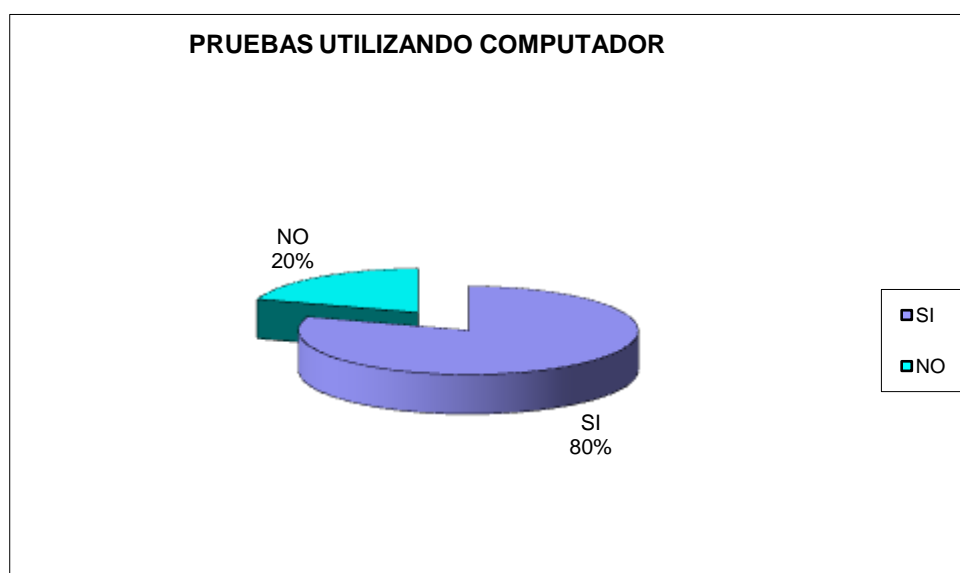


Gráfico 2: Uso del computador en la elaboración de exámenes

Sobre el setenta y dos por ciento de los encuestados indica que utiliza el computador para la elaboración de pruebas mientras que el veinte y ocho por ciento no lo hace.

Pregunta 3

¿El uso del computador en su práctica docente es? (Procesador de Texto, Herramienta de Análisis, Fuente de Consulta, Material Didáctico, Distracción, No lo usa).

CARACTERISTICA	FRECUENCIA	PORCENTAJE
Procesador de Texto	4	40
Herramienta de Análisis	2	20
Fuente de Consulta	1	10
Material Didáctico	1	10
Distracción	1	10
No lo usa	1	10
TOTAL	10	100

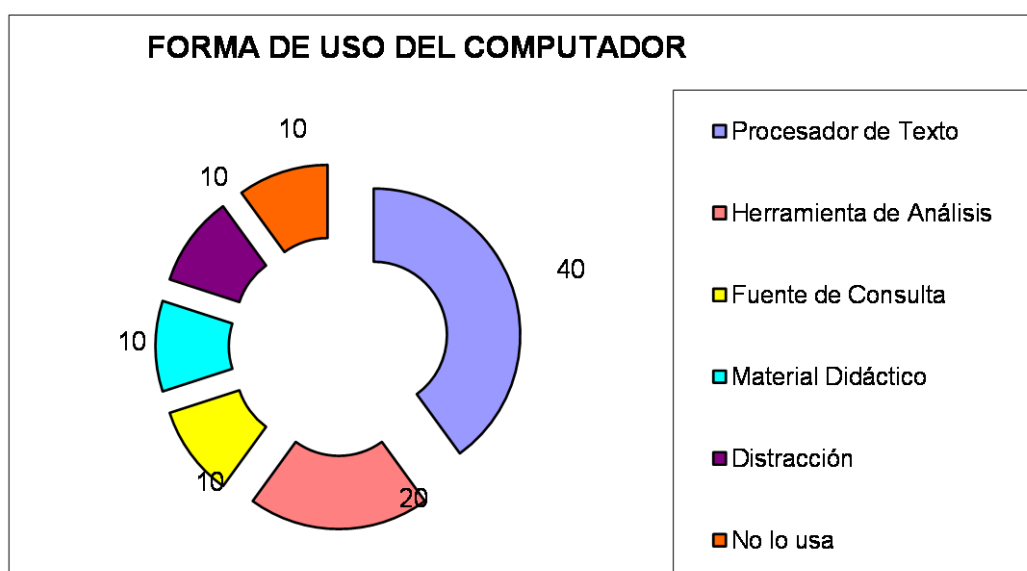


Gráfico 3: Uso del computador en la práctica docente

El setenta por ciento de la muestra es decir la mayoría indica que usa el computador básicamente como procesador de texto, el sesenta y cinco por ciento como fuente de consulta, el cincuenta y ocho por ciento como Material didáctico, en tanto que el treinta y tres por ciento lo usa como Herramienta de Análisis, el veinte y tres como distracción y el diez por ciento no lo utiliza.

Pregunta 4

¿Conoce algún sistema de evaluación? (si, no)

CARACTERISTICA	FRECUENCIA	PORCENTAJE
SI	2	20
NO	8	80
TOTAL	12	100

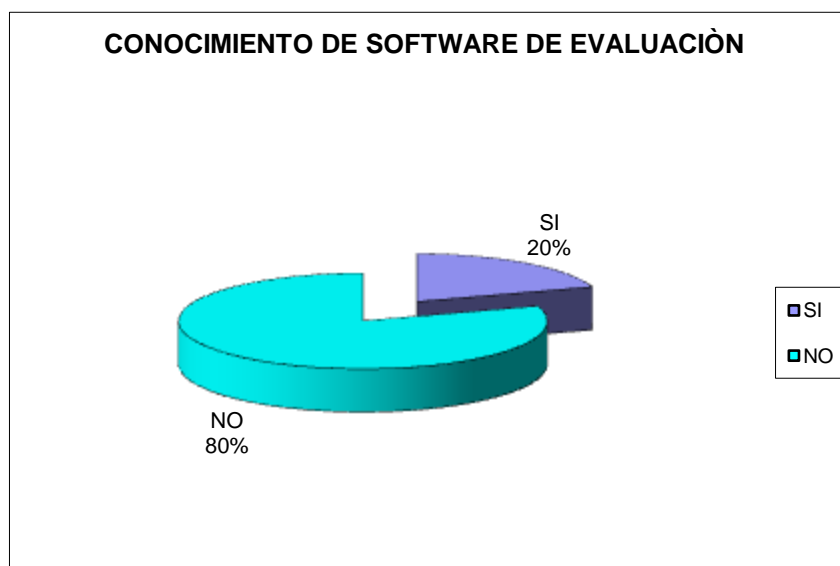


Gráfico 4: Conocimiento de algún sistema evaluativo

Sobre el 28% indica conocer y utilizar software de evaluación en contraste con el casi 72% que refiere el no conocer ni disponer de este tipo de software, se refleja también de las respuestas algunos paquetes utilizados en el medio.

Pregunta 5

¿El número de estudiantes a su cargo le permiten elaborar pruebas personalizadas?

CARACTERISTICA	FRECUENCIA	PORCENTAJE
SI	2	20
NO	8	80
TOTAL	10	100

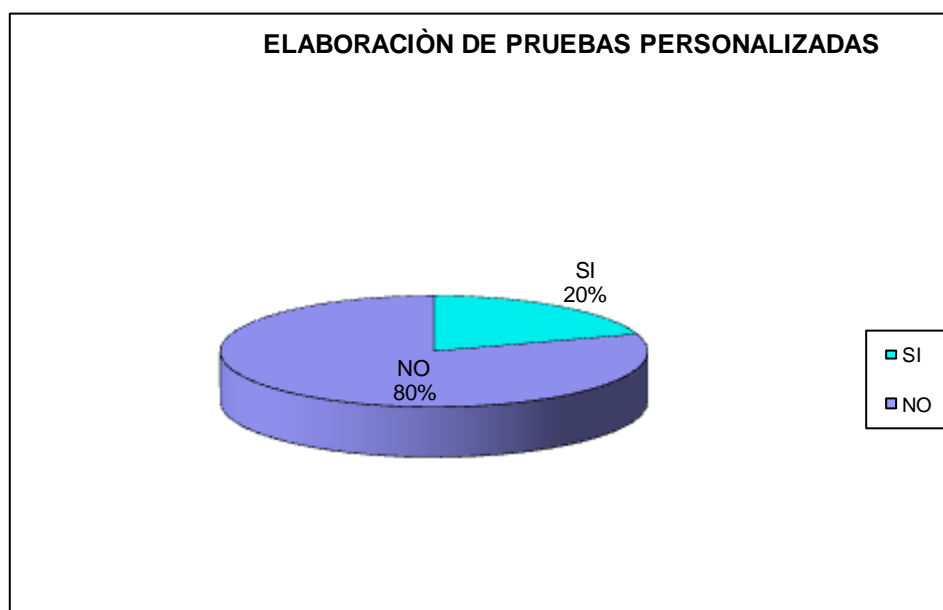


Gráfico 5: Estudiantes y elaboración de exámenes personalizados

La investigación muestra que apenas el 20% de la muestra elabora pruebas personalizadas, mientras que el 80% manifiesta que no lo hace principalmente por el excesivo número de estudiantes.

Pregunta 6

¿Las pruebas generalizadas bajo su criterio provocan la copia entre sus alumnos? (si, no)

CARACTERISTICA	FRECUENCIA	PORCENTAJE
SI	4	40
NO	6	60
TOTAL	10	100

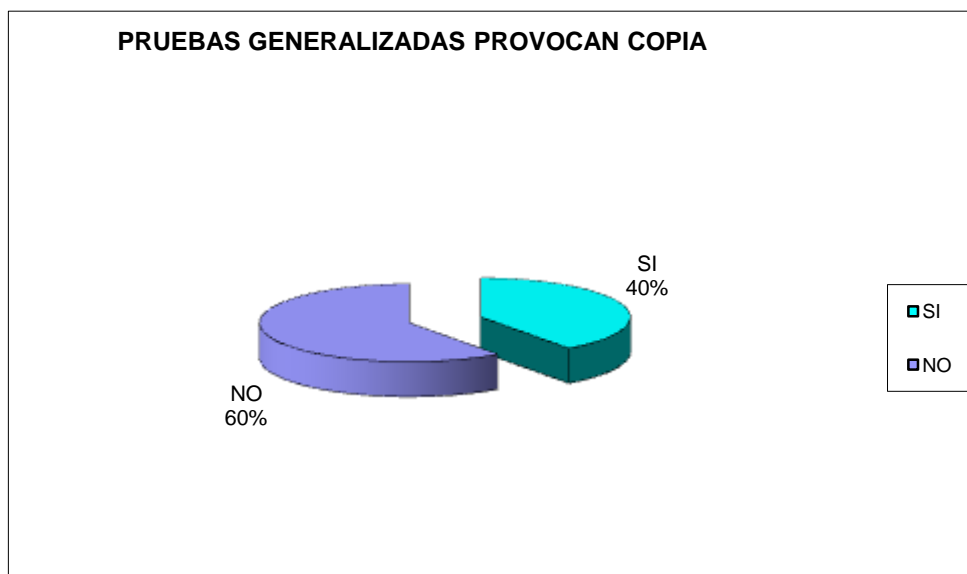


Gráfico 6: Copia entre alumnos

Aproximadamente el 40% opina que las pruebas generalizadas si provocan la copia, mientras que la otra mitad opina que si bien pueden ser un factor que influya no es determinante.

Pregunta 7

¿Cree usted que realizar pruebas aleatorias e individuales evitarían el fraude? (si, no).

CARACTERISTICA	FRECUENCIA	PORCENTAJE
SI	7	70
NO	3	30
TOTAL	10	100

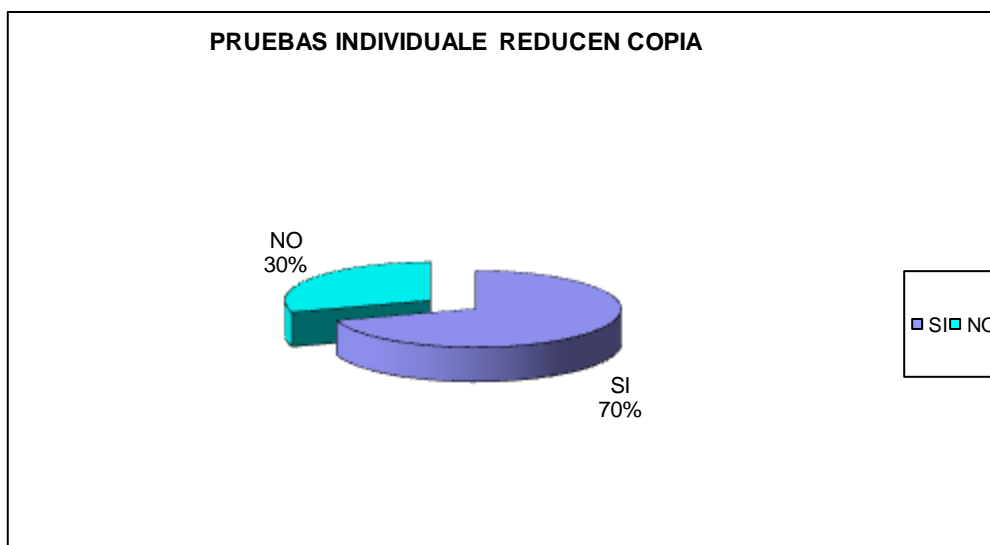


Gráfico 7: Pruebas individuales y el fraude

Sobre el 70% considera que las pruebas aleatorias e individuales reducirían la copia, mientras que el 30% restante considera que no.

Pregunta 8

¿Estaría usted dispuesto a elaborar manualmente pruebas individuales? (si, no).

CARACTERÍSTICA	FRECUENCIA	PORCENTAJE
SI	3	30
NO	7	70
TOTAL	10	100



Gráfico 8: Elaboración manual de exámenes individuales

Se muestra que apenas un 30% de los encuestados estarían dispuestos a elaborar pruebas individuales en forma manual, en contraste con el 70% que no lo haría.

Pregunta 9

¿De disponer un sistema de exámenes personalizados, utilizaría el mismo en el proceso de enseñanza-aprendizaje? (si, no)

CARACTERISTICA	FRECUENCIA	PORCENTAJE
SI	8	80
NO	2	20
TOTAL	10	100

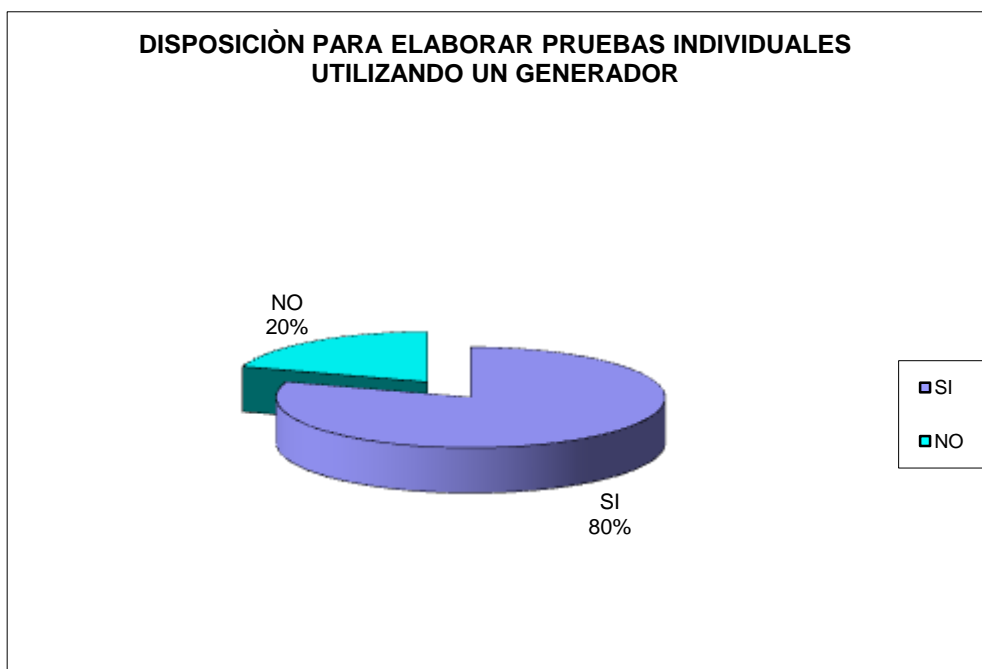


Gráfico 9: Uso del generador de pruebas en el proceso enseñanza - aprendizaje

El 80% de la muestra estaría dispuesto a elaborar pruebas personalizadas utilizando un generador, en contraste con el 20% restante que no lo haría, revelando el interés de los docentes por innovar el proceso de evaluación

Pregunta 10

¿Si se aplicara en la escuela de sistemas un sistema informático de exámenes personalizados, cree que se mejoraría el control de evaluación? (si, no)

CARACTERISTICA	FRECUENCIA	PORCENTAJE
SI	7	70
NO	3	30
TOTAL	10	100



Gráfico 10: Elaborar exámenes utilizando un sistema

La investigación revela que aproximadamente el 70% considera que si se mejoraría el aprendizaje con un sistema informático, mientras que el 30% restante opina lo contrario.

Pregunta 11

¿Le gustaría dominar el funcionamiento de un sistema de evaluación? (si, no)

CARACTERISTICA	FRECUENCIA	PORCENTAJE
SI	9	90
NO	1	10
TOTAL	10	100

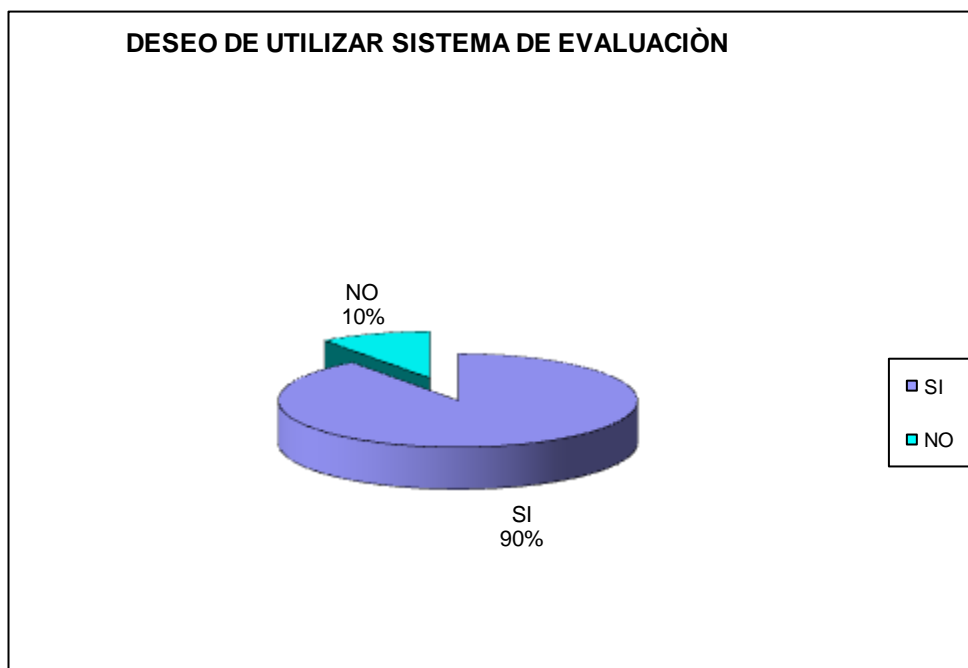


Gráfico 11: Utilizar sistema de evaluación

Casi la totalidad de los encuestados desean dominar el manejo de un software de evaluación, confirmándose así el deseo y la necesidad de utilizar herramientas nuevas de este tipo.

Preguntas dirigidas a los estudiantes

De manera simultánea se encuestó a la muestra estudiantil, de lo cual se presenta a continuación los resultados obtenidos:

Pregunta 1

¿Qué porcentaje de docentes utilizan sistemas o herramientas computacionales en el desarrollo de sus evaluaciones? (100%, 80%, 50%, 30%, 10%)

CARACTERISTICA	FRECUENCIA	PORCENTAJE
100%	2	1
80%	3	9
50%	5	24
30%	5	22
10%	20	44
TOTAL	45	100

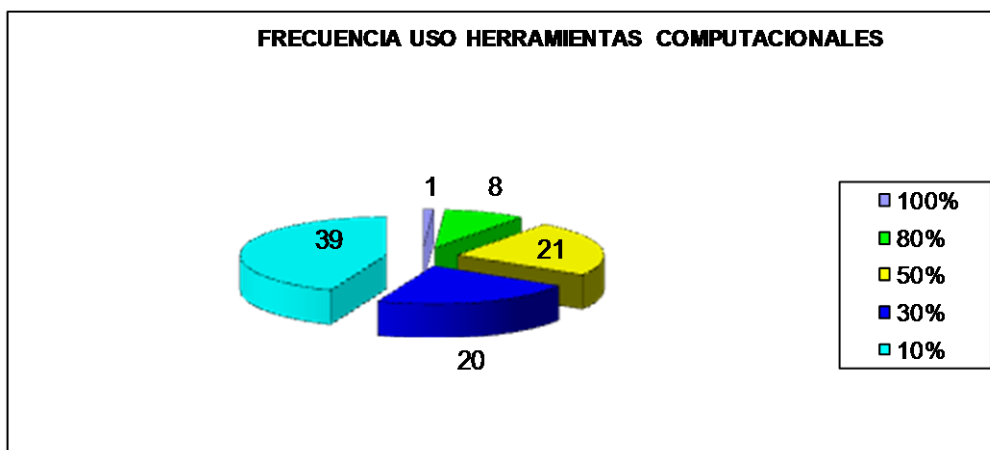


Gráfico 12: Uso de sistemas computacionales en el proceso de evaluación

La investigación muestra que menos del ochenta por ciento de los docentes utilizan herramientas computacionales en sus evaluaciones, lo cual indica un enorme desconocimiento de las herramientas existentes o una mala utilización de las que disponen

Pregunta 2

¿Los docentes han utilizado algún sistema de evaluación en el proceso de aprendizaje? (SI, NO)

CARACTERISTICA	FRECUENCIA	PORCENTAJE
SI	20	45
NO	25	55
TOTAL	45	100

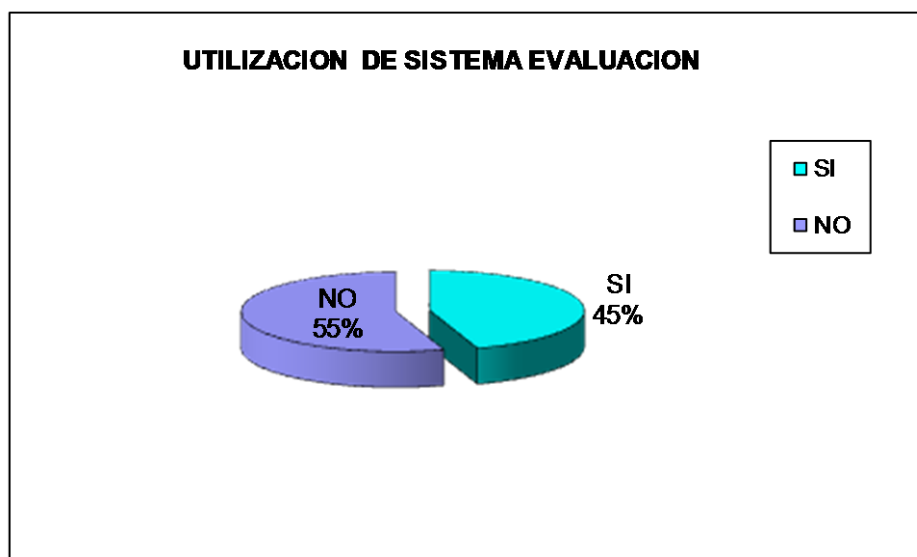


Gráfico 13: Uso de algún sistema de evaluación en el proceso de enseñanza

Sobre el 50% de los estudiantes indican que los docentes no utilizan software de evaluación, confirmando de esta manera la falta de conocimiento en cuanto a sistemas de evaluación que existen en la actualidad para mejorar el proceso de enseñanza – aprendizaje.

Pregunta 3

Considera que en el proceso de aprendizaje, sus conocimientos son: medidos o evaluados.

CARACTERISTICA	FRECUENCIA	PORCENTAJE
MEDIDOS	26	58
EVALUADOS	19	42
TOTAL	45	100

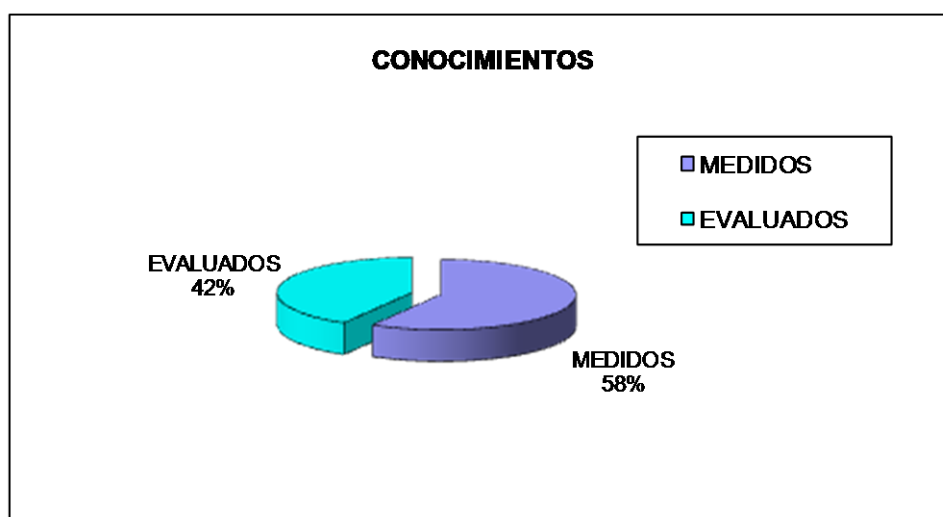


Gráfico 14: Evaluación del conocimiento

El 58% de los encuestados considera que sus conocimientos son únicamente medidos, mientras que el 42% restante cree que si son evaluados

Pregunta 4

¿Desearía que cambien o al menos se modifiquen las modalidades de evaluación?

(Si, no)

CARACTERISTICA	FRECUENCIA	PORCENTAJE
SI	40	89
NO	5	11
TOTAL	45	100

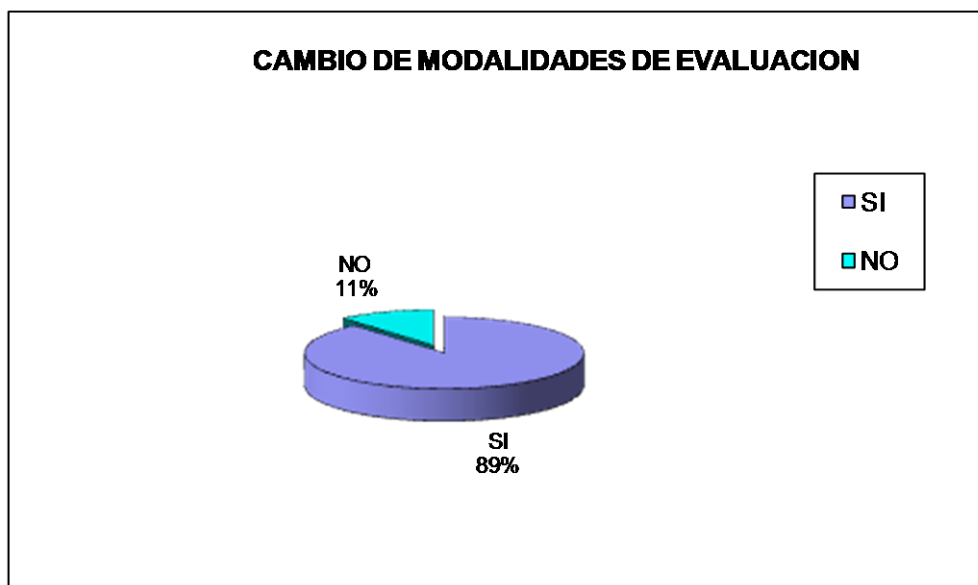


Gráfico 15: Cambios en las modalidades de evaluación

Existe un 89% de estudiantes que considera que se deben cambiar las modalidades de evaluación, y el 11% considera que no, ya que ellos prefieren:

- Evaluación más justa
- Evaluación más real
- Cambiar la rutina de las evaluaciones actuales
- Toma de conciencia alumno/profesor
- Estudiarían para aprender y no para la nota

Pregunta 5

¿Considera usted que existe unidad de criterios entre los docentes de la PUCESA para aplicar la evaluación? (si, no)

CARACTERISTICA	FRECUENCIA	PORCENTAJE
SI	16	37
NO	28	63
TOTAL	45	100

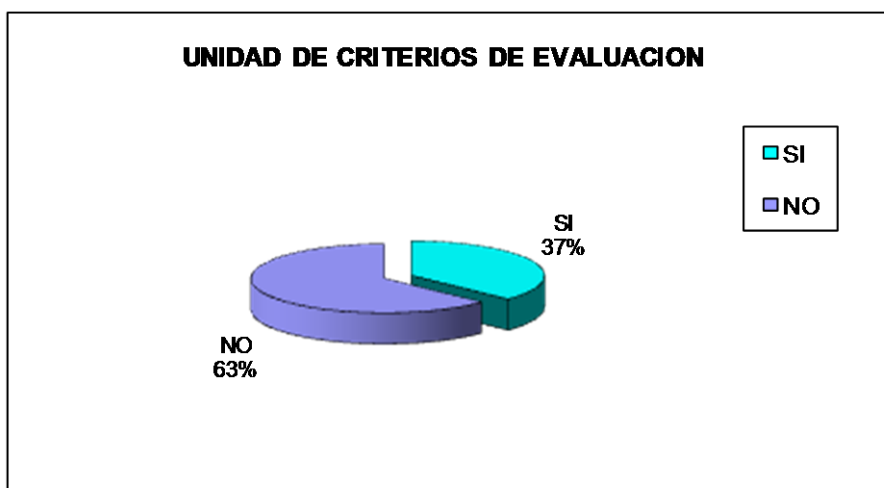


Gráfico 16: Unidad de criterios de evaluación

Sobre el 63% de los encuestados mencionan que no existe unidad en los criterios de evaluación, mientras que el 37% restante opina lo contrario, ya que para los estudiantes:

- Cada uno hace lo que cree
- Muchos no toman en cuenta la opinión del estudiante
- Se apegan al método tradicional
- Las políticas de los docentes se contradicen
- Califican de acuerdo a lo que ellos consideran

Pregunta 6

¿Le gustaría que los docentes utilicen un sistema informático de evaluación en el proceso de enseñanza? (si, no)

CARACTERISTICA	FRECUENCIA	PORCENTAJE
SI	43	99
NO	2	1
TOTAL	45	100

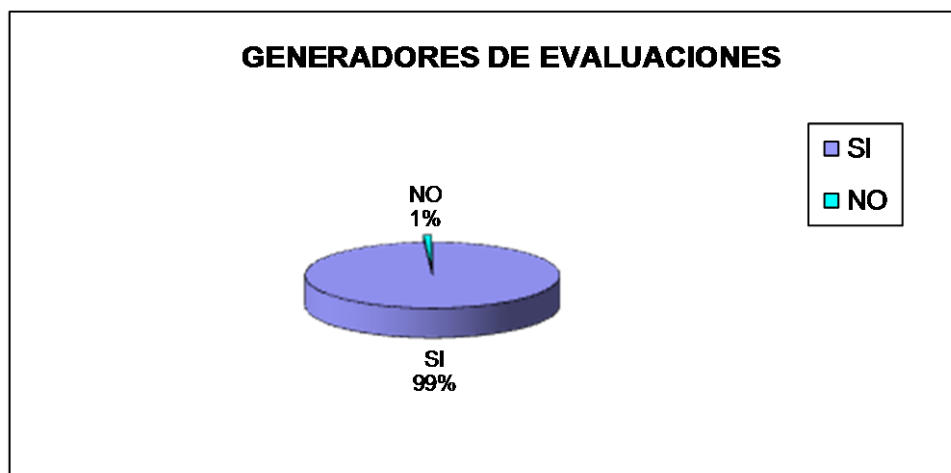


Gráfico 17: Utilización de generadores de evaluaciones

El 99% de la muestra desearía que se utilicen generadores de evaluaciones y el 1% opinan lo contrario.

- Más objetividad
- Unidad de criterios
- Facilitarían el trabajo al docente
- Acercamiento a la tecnología
- Más agradable para el alumno

Pregunta 7

¿Considera que la autoevaluación ayuda a la toma de conciencia del alumno? (si, no)

CARACTERISTICA	FRECUENCIA	PORCENTAJE
SI	38	87
NO	7	13
TOTAL	45	100

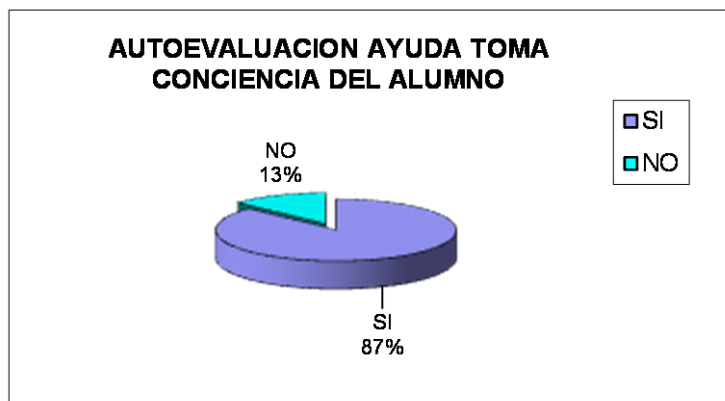


Gráfico 18: Conocimiento de algún sistema evaluativo

El 87% de los estudiantes consideran que la autoevaluación ayuda a la toma de conciencia del alumno, mientras que el 13% considera que no.

- Se aprende sin presión
- Ayuda a detectar falencias
- Fortalece el aprendizaje

Anexo 3

Manual de usuario del Sistema Cronos

1. Introducción

CRONOS es un sistema distribuido que gestiona la evaluación educativa, pensado como herramienta de uso general, utilizable por cualquier persona sin conocimientos de informática.

El sistema CRONOS es totalmente integrado y está constituido de forma modular, (Administrador y Examinador), cada uno de los módulos se van acoplando de acuerdo a las necesidades de los usuarios.

2. Generalidades del módulo administrador

2.1. Usuarios y autorizaciones

Para asignar autorizaciones en CRONOS, se definen tres conceptos principales:

2.1.1. Usuario

Persona a la que se otorga la autorización.

2.1.2. Contraseña

Información de usuario necesaria para su autenticación

2.1.3. Pantalla

Entidad sobre la que se asigna Autorización. Por ejemplo: Una Cátedra, un Docente, etc., con la combinación de estos dos conceptos se puede especificar que el Rol “ABC” puede operar sobre “Exámenes” o que el Rol “DEF” puede operar sobre “Alumnos”.

3. DESCRIPCION DEL MÓDULO: ADMINISTRADOR

3.1. Ingreso al módulo

La pantalla de acceso al módulo le pedirá los datos de usuario y contraseña, Ver gráfico 1, al ingresarla incorrectamente, se presenta el mensaje mostrado en el gráfico 2 y tendrá que volver a digitar la información.

Al ingresar correctamente el usuario y la contraseña presione el botón ACEPTAR o la tecla ENTER, este acceso ha sido otorgado por el administrador del sistema.



Gráfico 1: Acceso al módulo



Gráfico 2: Mensaje de error

3.2. Menús

La forma normal de operar CRONOS es seleccionar opciones que se hallan agrupadas en menús. Aunque el contenido varía, todos los menús de CRONOS tienen una estructura y funcionamiento muy parecidos, de forma que se puede describir un esquema general que cumplirán todos ellos.

3.2.1. Estructura de un menú

Aunque el contenido varía en cada caso, todo el menú de CRONOS es simétrico. La información se agrupa por cabeceras y detalles. Ver gráfico 3.



Gráfico 3: Pantalla de listado

3.2.2. Pantalla

En el apartado anterior se ha visto la estructura básica de los menús de CRONOS, una vez seleccionada la opción que corresponde al trabajo a efectuar, se visualiza un formato de pantalla.

3.3. Pantallas

3.3.1. Estructura de una pantalla

Existen básicamente dos tipos de pantallas que por lo general presentan las mismas secciones y éstas se explican a continuación:

3.3.1.1. Pantalla de listado

Invocar a una pantalla de mantenimiento, permite realizar búsquedas y navegar por los registros de la entidad seleccionada en el menú principal. Ver gráfico 4

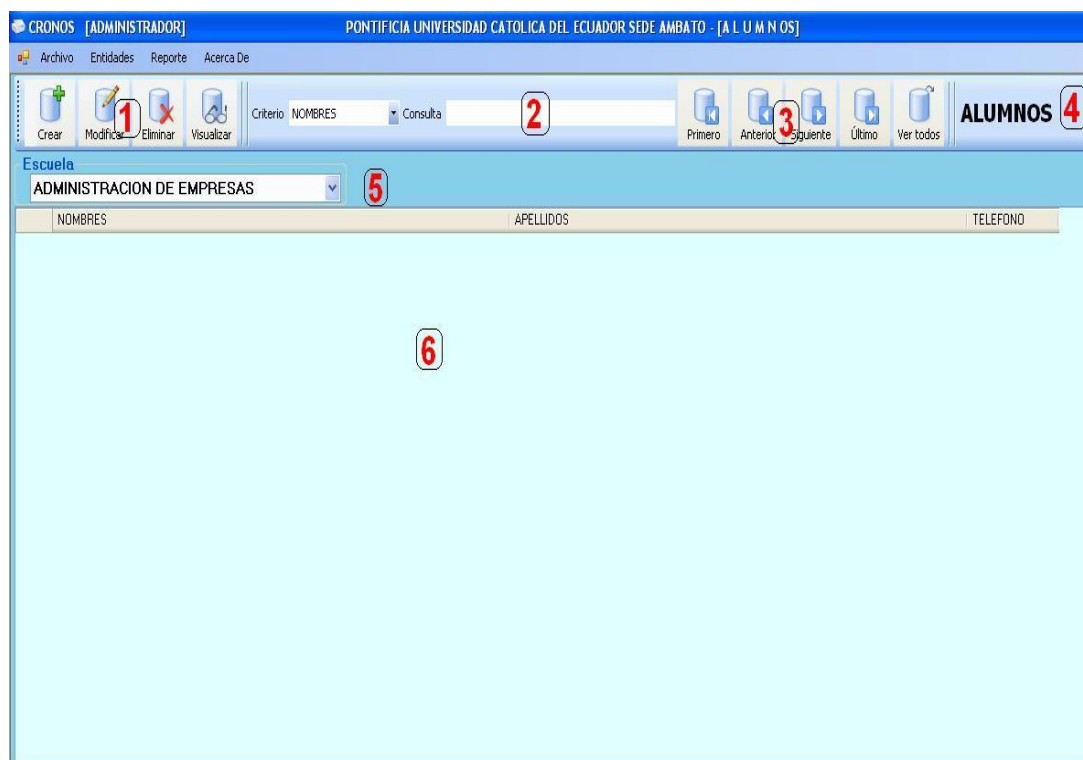


Gráfico 4: Pantalla de listado

- **Sección 1:** Presenta un menú con sus respectivas opciones.
- **Sección 2:** Presenta la funcionalidad de realizar determinadas búsquedas.
- **Sección 3:** Presenta un menú con sus respectivas opciones de navegación.
- **Sección 4:** Indica la entidad actual en que se encuentra el Usuario operando.
- **Sección 5:** Presenta una lista de posibles asociaciones que pertenece la entidad elegida.
- **Sección 6:** Se presenta una lista de todas las entidades existentes.

3.3.1.2. Pantalla de mantenimiento

Con La pantalla de mantenimiento podemos crear, modificar, eliminar y ver una entidad previamente seleccionada por la pantalla de listado. Ver gráfico 5



Gráfico 5: Pantalla de mantenimiento

- **Sección 1:** Presenta un menú con sus respectivas opciones, nuevo, grabar, salir.
- **Sección 2:** Presenta los campos que tiene una entidad.
- **Sección 3:** Presenta un mensaje de aviso que ciertos campos son obligatorios.

3.3.1.3. Campos

Campos de entrada: se pueden introducir datos en ellos.

Normalmente aparecen vacíos en el formato, aunque en ocasiones se muestran valores por omisión que se pueden cambiar. Ej.

Campos protegidos: aparecen con un contenido determinado y no está permitido cambiar la información. A estos campos se los llama campos protegidos y generalmente aparecen menos resaltados.

Campos de selección: Sirven para seleccionar entre varias opciones. Ej.



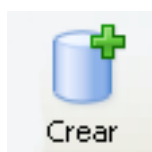
3.3.2. Navegación entre pantallas

La navegación entre pantallas se la puede definir como el fácil acceso para ir de una pantalla a otra. Cuando se trata de navegar de una opción a otra, el Usuario podrá saber en qué opción se encuentra actualmente.

Finalmente el Usuario podrá encontrar pantallas en las que puede ingresar a otra opción para el registro de datos sin salir de la opción actual y por ende sin perder los datos ingresados inicialmente. Por ejemplo, en el Módulo de Exámenes/Banco de preguntas, se puede observar el botón Subir que permite ir a esta opción, registrar las respectivas Preguntas y volver a la pantalla inicial con el botón Bajar.

3.3.3. Botones comunes en una pantalla

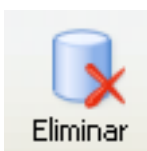
Botones comunes de cada modulo para el menú principal



Muestra una pantalla solicitando todos los datos requeridos para poder crear una nueva entidad.



Muestra una pantalla con los datos para su corrección o modificación de una entidad seleccionada anteriormente.

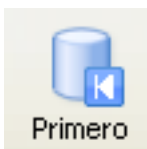


Muestra una pantalla con los datos para su eliminación de una entidad seleccionada anteriormente.

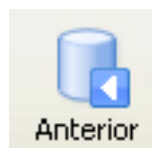


Muestra una pantalla con los datos para su visualización de una entidad

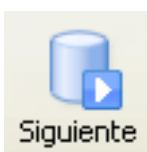
seleccionada anteriormente.



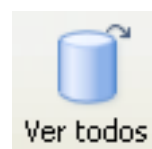
Permite ubicarse en el primer registro en una lista de entidades.



Permite ubicarse en el anterior del registro actual en una lista de entidades.



Permite ubicarse en el siguiente del registro actual en una lista de entidades.



Muestra todos los registros de una lista de entidades.



Muestra una pantalla en blanco para crear un registro de una entidad o agregar un registro en blanco.







Guarda un registro de una entidad



Elimina un registro de una entidad



Permite salir de la pantalla actual, para regresar a la anterior.

-  Permite eliminar un registro de una manera visual.
-  Muestra una pantalla relacionada con la actual.
-  Regresa a la pantalla que invoco a la actual.
-  Permite ver los exámenes tomados por un alumno.

3.4. Entidades

3.4.1. Escuelas

Permite gestionar la información necesaria para trabajar con las Escuelas, son obligatorios los campos (Nombre, Director).

A continuación se muestra las pantallas para las opciones **Crear, Modificar, Eliminar, Visualizar**.



The screenshot shows a web application window titled "Escuela". At the top, there are three icons: a plus sign, a stack of papers, and a power button. Below the icons, the text "Datos Escuela:" is followed by four input fields: "Nombre:", "Director:", "Secretaria:", and "Descripción:". The "Nombre:" and "Director:" fields are the only ones with asterisks, indicating they are required.

Gráfico 6 Crear Escuela

Escuela

Datos Escuela

Nombre: ADMINISTRACION DE EMPRESAS

Director: LCDA. TAIMI ACHILIE

Secretaria: BETY MORA

Descripción: SU MISION PRINCIPAL ES FORMAR A LOS FUTUROS ADMINISTRADORES DE LA NACION Y POR QUE NO DEL.

Gráfico 7 Modificar Escuela

Escuela

Datos Escuela

Nombre: ADMINISTRACION DE EMPRESAS

Director: LCDA. TAIMI ACHILIE

Secretaria: BETY MORA

Descripción: SU MISION PRINCIPAL ES FORMAR A LOS FUTUROS ADMINISTRADORES DE LA NACION Y POR QUE NO DEL.

Gráfico 8 Eliminar Escuela

Escuela

Datos Escuela

Nombre: ADMINISTRACION DE EMPRESAS

Director: LCDA. TAIMI ACHILIE

Secretaria: BETY MORA

Descripción: SU MISION PRINCIPAL ES FORMAR A LOS FUTUROS ADMINISTRADORES DE LA NACION Y POR QUE NO DEL.

Gráfico 9 Ver Escuela

3.4.2. Cátedras

Permite gestionar la información necesaria para trabajar con las cátedras, para esta opción es necesario que exista la entidad Cátedra, todos los campos son alfanuméricos y son obligatorios todos los campos, a continuación se muestra las pantallas para las opciones **Crear, Modificar, Eliminar, Visualizar**.

Cátedra

Datos Escuela

Escuela: ADMINISTRACION DE EM Codigo: 0002

Nombre: DERECHO MERCANTIL

Contenido: DERECHO

Descripción: MUY UTIL

Gráfico 10 Crear Cátedra

Cátedra

Datos Escuela

Escuela: ADMINISTRACION DE EM Codigo:

Nombre:

Contenido:

Descripción:

Gráfico 11 Modificar Cátedra

Cátedra

Datos Escuela

Escuela: ADMINISTRACION DE EM Codigo: 0002

Nombre: DERECHO MERCANTIL

Contenido: DERECHO

Descripción: MUY UTIL

Gráfico 12: Ver Cátedra

Cátedra

Datos Escuela

Escuela: ADMINISTRACION DE EM Codigo: 0002

Nombre: DERECHO MERCANTIL

Contenido: DERECHO

Descripción: MUY UTIL

Gráfico 13: Ver Cátedra

3.4.3. Docentes

Permite gestionar la información necesaria para trabajar con los docentes, estos pueden ser candidatos para ser Usuarios del sistema **Administrador**, los campos obligatorios son (nombres, apellidos, título, cedula y teléfono) y todos los campos son alfanuméricos excepto cedula y teléfono que son numéricos, a continuación se muestra las pantallas para las opciones **Crear, Modificar, Eliminar, Visualizar**.

Docente

Datos Profesional

Nombres: PASCUDO

Apellidos: VELASCO

Título: ADMINISTRADOR

Cedula: 0058762365

Telefono: 2569087

Direccion: SUCRE 412 E IGUERAS

Referencias: MAS DE 20 AÑOS DE EXPERIENCIA EN LA DOCENCIA

Gráfico 14: Crear Docente

Docente

Datos Profesional

Nombres:

Apellidos:

Título:

Cedula:

Telefono:

Direccion:

Referencias:

Gráfico 15: Modificar Docente

Docente

Datos Profesional

Nombres: MICHAEL

Apellidos: JACKSON

Título: ARTISTA

Cedula: 1111111111

Telefono: 098765432

Direccion: 098765432

Referencias: EL MEJOR ARTISTA DE TODOS LOS TIEMPOS

Gráfico 16: Modificar Docente

The screenshot shows a window titled 'Docente' with a toolbar containing a plus sign, a stack of papers, and a power button. The main area is labeled 'Datos Profesional' and contains the following fields:

Nombre:	ANGEL
Apellidos:	ANCHUNDIA
Título:	PEDAGOGO
Cedula:	1104147440
Telefono:	2367549
Direccion:	AV. CEVALLOS Y TOMAS SEVILLA
Referencias:	DOCENTE DEL INSTITUTO TECNICO SUPERIOR "BOLIVAR"

Gráfico 17 Visualizar Docente

3.4.4. Alumnos

Permite gestionar la información necesaria para trabajar con los alumnos, estos son candidatos a ser usuarios para el modulo **Examinador**, a continuación se muestra las pantallas para las opciones **Crear, Modificar, Eliminar, Visualizar**.

The screenshot shows a window titled 'Alumno' with a toolbar containing a plus sign, a stack of papers, and a power button. The main area is labeled 'Datos Alumno' and contains the following fields:

Escuela:	INGENIERIA DE SISTEMAS
Nombres:	JOSE
Apellidos:	CASTILLO
Cedula:	1111111111
Telefono:	222222222
Direccion:	QUITO
Observaciones:	MAMASEMAMACOSA

Gráfico 18 Crear Alumno

The screenshot shows a window titled 'Alumno' with a toolbar containing a plus sign, a stack of papers, and a power button. The main area is labeled 'Datos Alumno' and contains the following fields:

Escuela:	ADMINISTRACION DE EMPRESAS
Nombres:	
Apellidos:	
Cedula:	
Telefono:	
Direccion:	
Observaciones:	

Gráfico 19 Modificar Alumno

Alumno

Datos Alumno

Escuela: INGENIERIA DE SISTEMAS

Nombres: JOSE

Apellidos: CASTILLO

Cedula: 1111111111

Telefono: 222222222

Direccion: QUITO

Observaciones: MAMASEMAMACDOSA

Gráfico 20 Eliminar Alumno

Alumno

Datos Alumno

Escuela: INGENIERIA DE SISTEMAS

Nombres: JOSE

Apellidos: CASTILLO

Cedula: 1111111111

Telefono: 222222222

Direccion: QUITO

Observaciones: MAMASEMAMACDOSA

Gráfico 21 Visualizar Alumno

3.4.5. Exámenes

Permite gestionar la información necesaria para trabajar con los exámenes, un examen está dividido en dos partes:

Cabecera: En esta sección se encuentra información básica de un examen datos de escuela y configuración. Ver gráfico 22

La cabecera del examen cuenta con un campo protegido, (Docente), solo el Usuario con privilegios de administrador podrá modificarlo, los campos (código, título,

nombre, descripción) son obligatorios, existen campos numéricos, alfanuméricos y de selección.

The screenshot shows a window titled 'Examen' with a blue header. Below the header are three icons: a plus sign, a globe with a blue arrow, and a red power button. The main area is divided into two sections:

- Datos Escuela:** Contains a dropdown menu for 'Cátedra' with 'SUFICIENCIA EN INFORMATICA' selected, a text input for 'Código', a text input for 'Título', a text input for 'Docente', and a large text area for 'Instrucciones'.
- Configuracion:** Contains several checkboxes: 'Automática', 'Corregida', and 'Aleatoria'. There is also a 'Control de tiempo' field with a numeric input set to '0' and the unit 'mins', and a 'Calificar sobre' text input.

Gráfico 22 Cabecera de examen

Preguntas: Esta sección depende de la cabecera del examen, si el usuario en la sección de configuración no selecciono la opción automática se presenta una lista de preguntas que el examen tendrá para su evaluación, las preguntas serán solamente de la cátedra seleccionada anteriormente en la cabecera. Ver gráfico 23.

The screenshot shows a section titled 'PREGUNTAS' with a green play button icon. Below the title is a table with the following structure:

ENUNCIADO	TIPO	PUNTAJE

Gráfico 23 Preguntas adheridas al examen

Si el usuario selecciono la opción de automática el sistema mostrar los módulos de la cátedra que tengan preguntas, donde el usuario podrá ingresar el número de preguntas, éste será aleatorio. Ver gráfico 24.

PREGUNTAS		
Preguntas		
Módulo	# Existentes	#A agregar
1	32	
2	7	

Gráfico 24 Preguntas adheridas al examen (Automática)

A continuación se muestra las pantallas para las opciones **Crear, Modificar, Eliminar, Visualizar.**

The screenshot shows a web application window titled 'Examen'. It features a toolbar with a plus sign, a stack of papers, and a power button. Below the toolbar, there are two main sections: 'Datos Escuela' and 'Configuracion'. The 'Datos Escuela' section contains a dropdown menu for 'Catedra' (selected as 'DERECHO MERCANTIL'), a text input for 'Codigo', a text input for 'Titulo', a dropdown for 'Docente' (selected as 'PEDAGOGO ANGEL ANCHUNDIA'), and a text area for 'Instrucciones'. The 'Configuracion' section has three checkboxes: 'Corregida', 'Aleatoria', and 'Control de tiempo' (set to 0 mins). Below these is a label '0 Preguntas'. At the bottom, there is a table with columns 'ENUNCIADO', 'TIPO', and 'PUNTAJE'.

Gráfico 25: Crear examen



Gráfico 26: Modificar examen

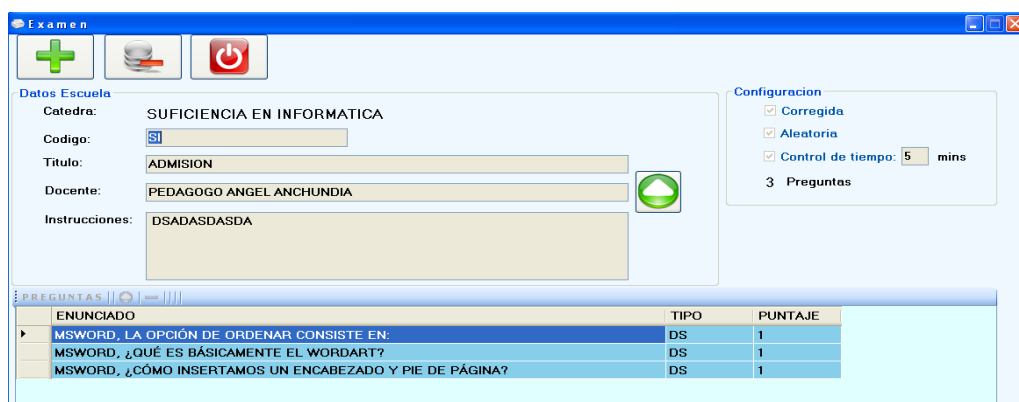


Gráfico 27: Eliminar examen

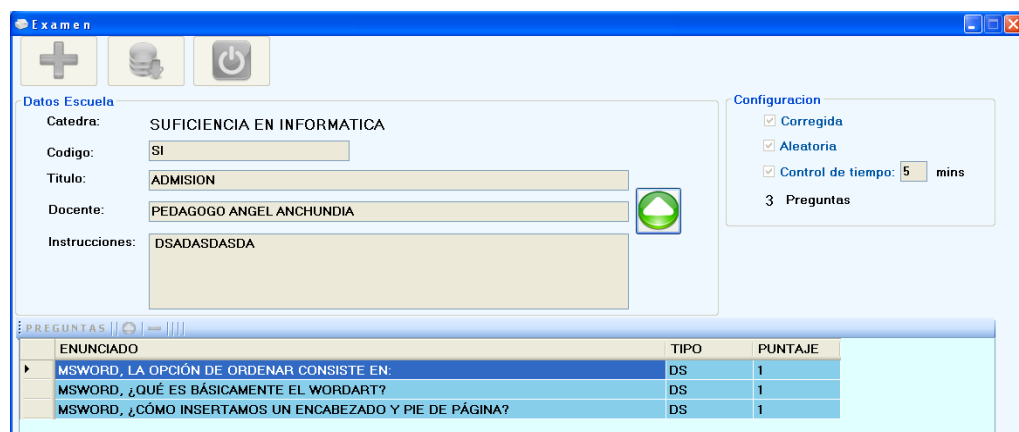


Gráfico 28: Visualizar examen

3.4.6. Banco de Preguntas

Le permite gestionar información sobre preguntas que pueden formar parte de un examen, es muy importante que sepa que existen tres tipos de pregunta (Decisión, Enlace y Práctica) esta última está habilitada cuando se trate preguntas sobre la cátedra “SUFICIENCIA EN INFORMATICA”, son obligatorios los campos (Enunciado y Puntaje), alfanumérico y numérico respectivamente.

A continuación algunas pantallas para ilustrar.



Gráfico 29: Pantalla para pregunta del tipo practica

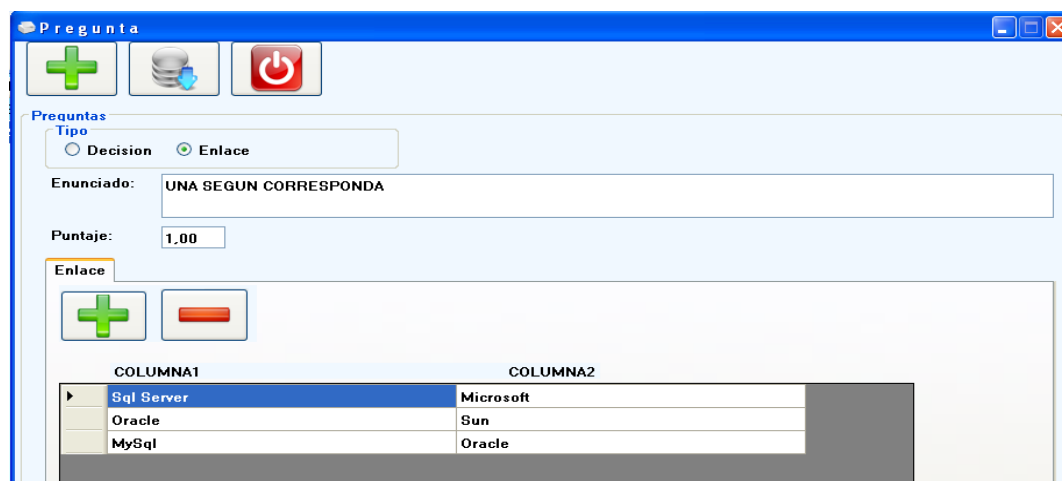


Gráfico 30: Pantalla para pregunta del tipo enlace

3.4.7. Usuarios

Permite gestionar los usuarios para los módulos Administrador y Examinador, los usuarios del sistema CRONOS pueden ser Docentes y Alumnos.

Para la opción modificar solo el rol del usuario, si se modifica el alumno por docente o viceversa, el sistema borrara el usuario actual y crea uno nuevo, A continuación se muestra las pantallas para las opciones **Crear, Modificar, Eliminar, Visualizar**.

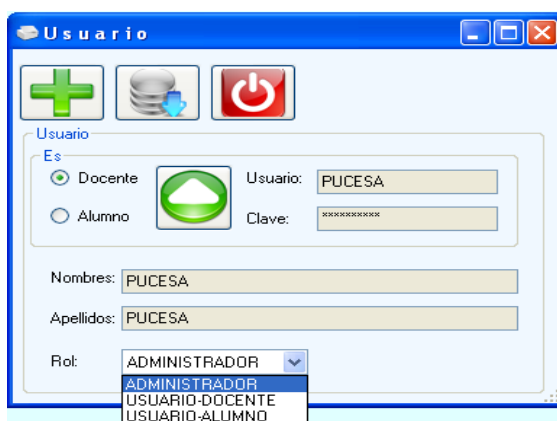


Gráfico 31: Crear Usuario

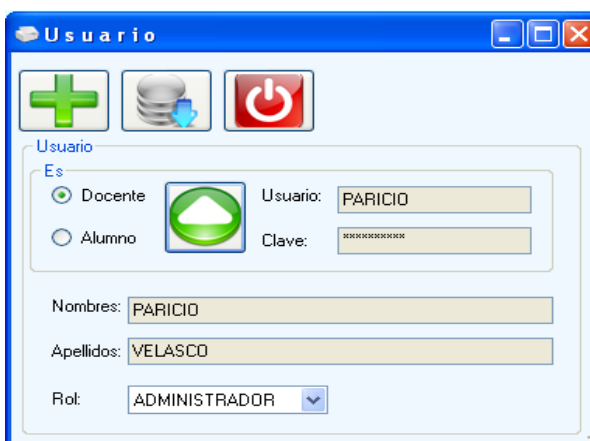


Gráfico 32: Modificar Usuario



Gráfico 33: Eliminar Usuario

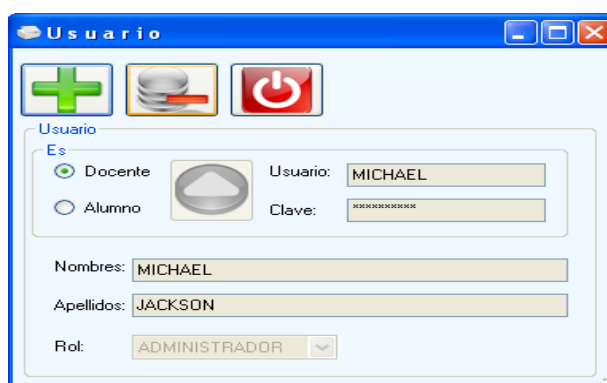


Gráfico 34: Modificar Usuario

3.4.8. Roles

Permite gestionar los roles para los usuarios de los módulos Administrador y Examinador, ver los gráficos para **Crear, Modificar, Eliminar, Visualizar**.

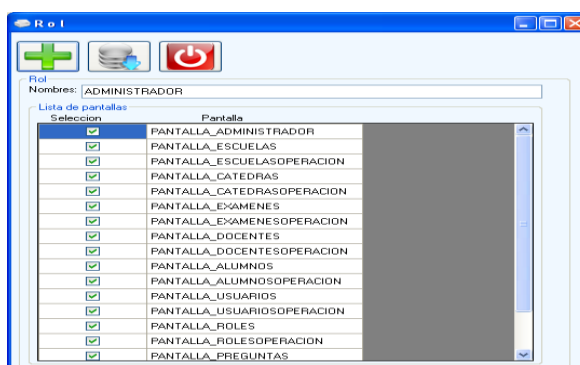


Gráfico 35: Crear Rol

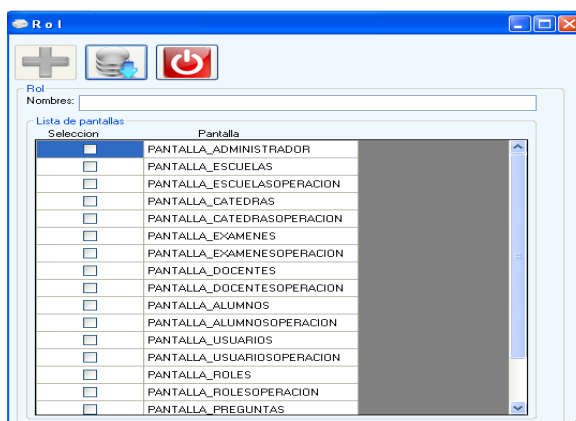


Gráfico 36 Modificar Rol

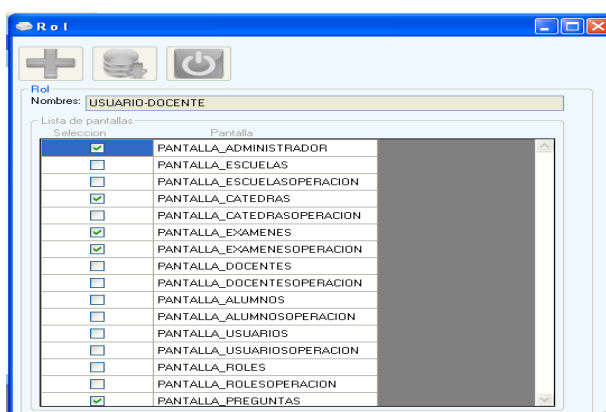


Gráfico 37: Eliminar Rol

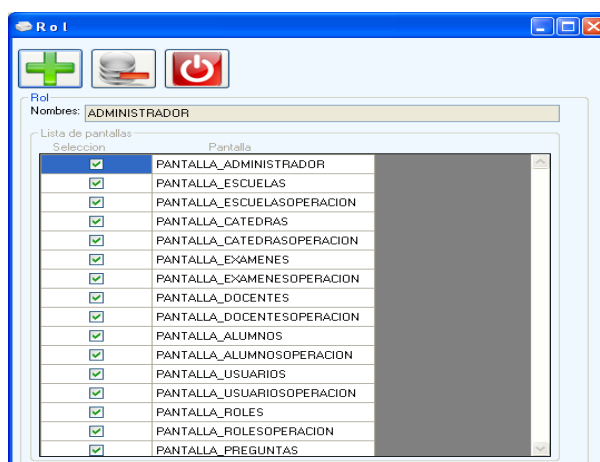


Gráfico 38: Visualizar Rol

3.4.9. Reporte

Reporte examen, permite visualizar e imprimir el examen tomado por un alumno.

La pantalla del gráfico 39, permite seleccionar el usuario para visualizar su examen.



Gráfico 39: Selección de alumno

La gráfico 40, muestra un ejemplo de examen tomado por un alumno, se detallan datos como pregunta, respuesta, valor de la pregunta y puntaje obtenido.


			
ESCUELA:	INGENIERIA DE SISTEMAS		
CATEDRA:	SUFICIENCIA EN INFORMATICA		
EXAMEN:	AUTOMATICO Y ALEATORIO		
ALUMNO:	CARLOS ANALUISA		
VALOR:	7,00		
	NOTA: 4,50		
Pregunta	Respuesta	Valor Pregunta	Puntaje
MSWORD, ESTABLESCA EL TAMANO DE TODO EL TEXTO MAYOR A 20			
Fallida		1,00	0,00
ESTABLESCA EL TEXTO CON FUENTE MENOR A 5			
Excelente		1,00	1,00

Gráfico 40: Reporte Examen

Reporte calificaciones, permite visualizar e imprimir las calificaciones de los alumnos de un examen específico, este reporte muestra la calificación sobre el valor del examen.

La pantalla del gráfico 41 permite seleccionar el examen para posteriormente visualizar las calificaciones de los alumnos, ver gráfico 42.



Gráfico 41: Selección de examen



ESCUELA: INGENIERIA DE SISTEMAS
 CÁTEDRA: SUFICIENCIA EN INFORMATICA
 EXAMEN: EXAMEN PRUEBA
 CALIFICACION SOBRE: 6,00

<u>NOMBRES</u>	<u>APELLIDOS</u>	NOTA	<u>CALIFICACIÓN</u>
JOSE	CASTILLO	3,50 / 6,00	3,50

Gráfico 42: Reporte calificaciones

4. DESCRIPCIÓN DEL MÓDULO EXAMINADOR

4.1. Ingreso al módulo

Al ingresar correctamente el usuario y la contraseña presione el botón ACEPTAR o la tecla ENTER, este acceso ha sido otorgado por el administrador del sistema. Ver gráfico 43

Ingrese la contraseña del usuario. Al ingresarla incorrectamente, se presenta el mensaje mostrado en el gráfico 44 y tendrá que volver a digitar la información.

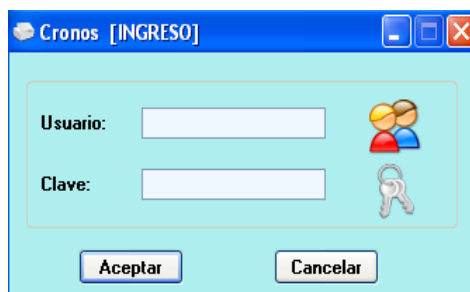


Gráfico 43: Acceso al modulo

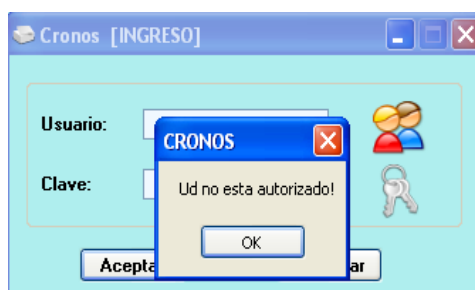


Gráfico 44: Mensaje de error

4.2. Pantallas

En este modulo básicamente existen dos pantallas, Panel y Examinador, a continuación veremos su estructura con sus respectivas secciones:

4.2.1. Estructura de una pantalla de información

La pantalla panel se presenta una vez que el usuario ha ingresado al módulo. Ver gráfico 45, 46

Gráfico 45: Pantalla Información

Gráfico 46 Pantalla Información

- **Sección 1:** Presenta un árbol menú con sus respectivas cátedras y exámenes.
- **Sección 2:** Presenta la información del alumno o del examen.
- **Sección 3:** Permite moverse entre la información de un alumno o examen.
- **Sección 4:** Permite tomar el examen ingresando el código del examen.

4.2.2. Pantalla Examinador

La pantalla examinador sirve para que el alumno navegue pregunta por pregunta y termine el examen. Ver gráfico 47

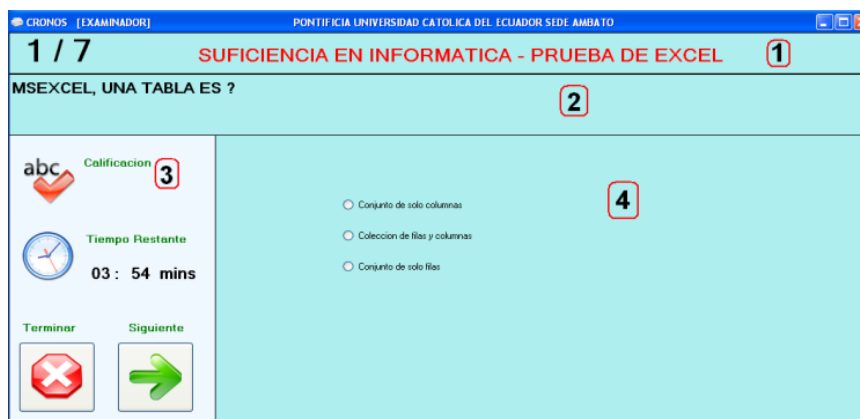


Gráfico 47: Pantalla Examinador

- **Sección1:** Presenta el número de pregunta y el examen que está tomando.
- **Sección2:** Presenta el enunciado de la pregunta.
- **Sección3:** Presenta calificación, tiempo restante, botones (avanzar y terminar)
- **Sección4:** Presenta la pregunta, decisión simple, múltiple y enlace.

4.3. Controles de preguntas

4.3.1. Pregunta decisión simple

El control solo le permite escoger una opción. Ver gráfico 48

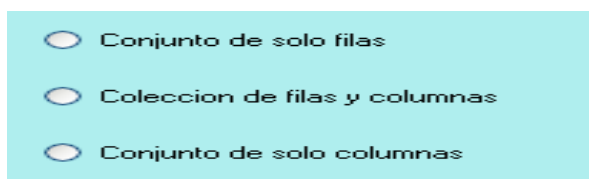


Gráfico 48: Pregunta decisión simple

4.3.2. Pregunta decisión múltiple

El control solo le permite escoger más de una opción. Ver gráfico 49

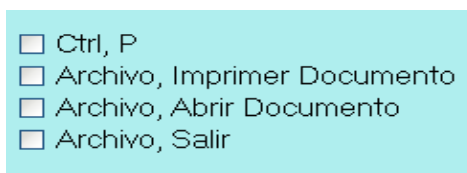


Gráfico 49: Pregunta decisión simple

4.3.3. Pregunta de enlace

El control le permite unir los elementos mediante líneas y existen dos botones para su corrección si es necesario. Ver gráfico 50

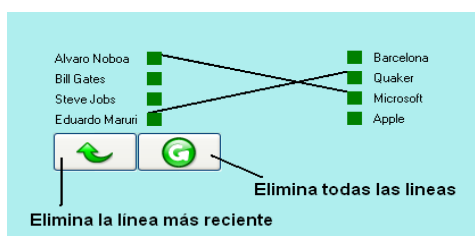


Gráfico 50: Pregunta enlace

4.3.4. Pregunta Practica

Pregunta exclusiva para examen de la cátedra SUFICIENCIA EN INFORMATICA, el sistema abrirá el programa (MS Word, MS Excel, MS PowerPoint) según la pregunta. Ver gráfico 51

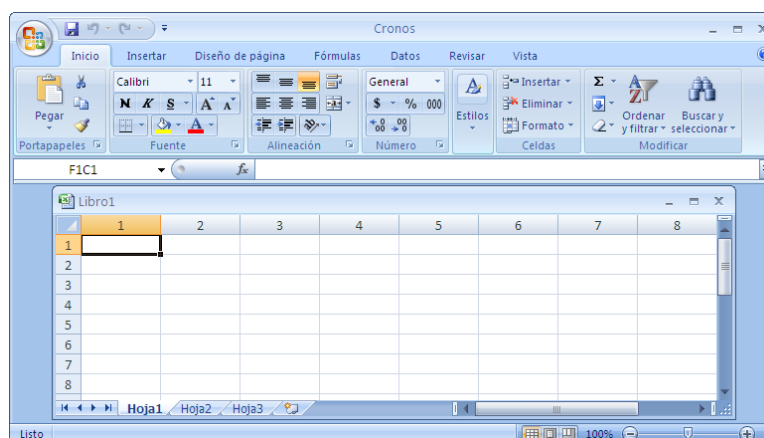


Gráfico 51: Pregunta práctica