

Universidad de las Ciencias Informáticas  
Facultad 5



## **REINGENIERÍA PARA SISTEMAS DE EVALUACION TEÓRICA**

**Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas.**

**Autor:** Juan Carlos Quevedo Lussón

**Tutores:** Ing Fernando Jiménez López  
Universidad de las Ciencias Informáticas  
[fjimenez@uci.cu](mailto:fjimenez@uci.cu)

Ing Alain Hernández Castillo  
Universidad de las Ciencias Informáticas  
[alainhc@uci.cu](mailto:alainhc@uci.cu)

Mayo, 2007

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 5 de la Universidad de Ciencias Informáticas para que haga el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del \_\_\_\_\_.

Juan Carlos Quevedo Lussón

Nombre completo del autor

Fernando Jiménez López

Nombre completo del primer tutor

Alain Hernandez Castillo

Nombre completo del segundo tutor

## **Agradecimientos**

A mis padres por apoyarme en todo. Por entenderme cuando lo necesité. A mi abuela del alma. A Andrés que ha sido un segundo padre para mí. A mis tíos y mi hermano que me han facilitado la vida tan lejos de casa.

A mis compañeros de universidad, que me enseñaron muchas cosas en 5 años. A mis grandes amigos y amigas que tanto me aconsejaron y me ayudaron mientras pudieron.

A mis tutores, que tanto se esforzaron porque este trabajo saliera.

A Leyanis por entenderme tanto.

A Vivian por ayudarme a plantearme metas en la vida.

A la Revolución por haberme puesto esta posibilidad a mi alcance.

En fin a todos los que de una forma u otra han colaborado para que yo llegara hasta aquí.

## Dedicatoria

*A mi familia y a los amigos que siempre me brindaron su ayuda  
incondicional.*

## Resumen

El presente trabajo está enmarcado en la modernización de un sistema para evaluar teóricamente a los aspirantes a obtener licencia de conducción, aunque pudiera ser usado para cualquier otra área en que se necesite evaluar los conocimientos teóricos. Tiene como objetivo realizar una reingeniería al sistema de evaluación teórica desarrollada por el proyecto Auto-Teórico de la Facultad 5 de la Universidad de las Ciencias Informáticas, para mejorar el funcionamiento de dicho sistema, las clases y componentes obtenidos podrán servir de *framework* para desarrollar sistemas similares.

El presente documento recoge los resultados del trabajo investigativo realizado. Se explican los conceptos relacionados con el mismo, se realiza un diseño de la propuesta del sistema, y se dejan algunas recomendaciones para el mejoramiento futuro del mismo.

# Índice

<b>AGRADECIMIENTOS.....</b>	<b>I</b>
<b>DEDICATORIA .....</b>	<b>II</b>
<b>RESUMEN .....</b>	<b>III</b>
<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA.....</b>	<b>5</b>
1.1 INTRODUCCIÓN.....	5
1.2 OBJETO DE ESTUDIO Y CAMPO DE ACCIÓN. ....	6
1.3 REINGENIERÍA .....	7
1.4 TENDENCIAS Y TECNOLOGÍAS ACTUALES. ....	9
1.4.1 <i>Aplicaciones genéricas</i> .....	9
1.4.2 <i>Modularidad</i> .....	10
1.4.3 <i>Extensibilidad</i> .....	10
1.4.4 <i>Sistemas Gestores de Bases de Datos (SGBD)</i> .....	11
1.5 METODOLOGÍAS PROPUESTAS. ....	18
1.5.1 <i>UML (Unified Modeling Language)</i> .....	19
1.5.2 <i>Rational Rose</i> .....	20
1.6 LENGUAJES DE PROGRAMACIÓN .....	21
1.7 ANÁLISIS DE OTRAS SOLUCIONES EXISTENTES. ....	25
1.8 PROPUESTA DE SOLUCIÓN TÉCNICA .....	28
1.9 CONCLUSIONES .....	30
<b>CAPÍTULO 2 MODELO DEL NEGOCIO.....</b>	<b>31</b>
2.1 INTRODUCCIÓN.....	31
2.2 MODELO DEL NEGOCIO ACTUAL .....	31
2.3 REGLAS DEL NEGOCIO A CONSIDERAR .....	32
2.4 ACTORES DEL NEGOCIO .....	33
2.5 DIAGRAMA DE CASOS DE USO DEL NEGOCIO.....	33
2.6 TRABAJADORES DEL NEGOCIO .....	34

2.7	CASOS DE USO DEL NEGOCIO .....	35
2.7.1	<i>Caso de uso Realizar Examen Teórico</i> .....	35
2.7.2	<i>Caso de uso Gestionar Reclamación de Inscripción</i> .....	38
2.7.3	<i>Caso de uso Gestionar Inscripción</i> .....	40
2.7.4	<i>Caso de uso Gestionar Reclamación de Examen</i> .....	42
2.8	MODELO DE OBJETOS .....	43
2.9	CONCLUSIONES .....	44
<b>CAPÍTULO 3 REQUISITOS .....</b>		<b>45</b>
3.1	INTRODUCCIÓN.....	45
3.2	ACTORES DEL SISTEMA A AUTOMATIZAR .....	45
3.3	PAQUETES Y SUS RELACIONES .....	46
3.4	DIAGRAMA DE CASOS DE USO DEL SISTEMA A AUTOMATIZAR .....	47
3.4.1	<i>Casos de uso a desarrollar en la primera iteración del sistema</i> .....	47
3.4.2	<i>Descripción y expansión de los casos de uso</i> .....	48
3.5	DEFINICIÓN DE LOS REQUISITOS FUNCIONALES Y NO FUNCIONALES .....	70
3.5.1	<i>Requisitos funcionales</i> .....	70
3.5.2	<i>Requisitos no funcionales</i> .....	73
3.6	CONCLUSIONES .....	74
<b>CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA .....</b>		<b>75</b>
4.1	INTRODUCCIÓN.....	75
4.2	DIAGRAMA DE CLASES DEL DISEÑO .....	75
4.2.1	<i>Arquitectura de 3 capas</i> .....	75
4.2.2	<i>Capa Aplicación</i> .....	76
4.2.3	<i>Capa Negocio</i> .....	86
4.2.4	<i>Capa Acceso a Datos</i> .....	93
4.3	REALIZACIÓN DE LOS CASOS DE USO .....	94
4.3.1	<i>Generar examen</i> .....	94
4.3.2	<i>Realizar examen</i> .....	95
4.3.3	<i>Calificar examen</i> .....	96
4.3.4	<i>Conectar con examinador</i> .....	96
4.3.5	<i>Almacenar datos de examen</i> .....	97
4.3.6	<i>Buscar</i> .....	98

4.3.7	<i>Cancelar examen</i> .....	99
4.3.8	<i>Asignar puesto de examen</i> .....	100
4.3.9	<i>Chequear datos de puesto de trabajo</i> .....	101
4.3.10	<i>Visualizar datos de examen</i> .....	101
4.3.11	<i>Autenticar usuario</i> .....	102
4.4	PRINCIPIOS DE DISEÑO .....	102
4.4.1	<i>Interfaz de usuario del Examinador</i> .....	103
4.4.2	<i>Interfaz de usuario del Cliente</i> .....	103
4.5	DISEÑO DE LA BASE DE DATOS .....	104
4.5.1	<i>Modelo lógico de datos</i> .....	104
4.5.2	<i>Modelo físico de datos</i> .....	105
4.6	MODELO DE IMPLEMENTACIÓN. ....	110
4.6.1	<i>Diagrama de paquetes</i> . ....	111
4.6.2	<i>Diagrama de componentes Capa Acceso a datos</i> . ....	112
4.6.3	<i>Diagrama de componentes Capa de Aplicación</i> . ....	113
4.6.4	<i>Diagrama de componentes Capa de Negocio</i> . ....	115
4.6.5	<i>Diagrama de componentes binarios</i> . ....	117
4.7	DIAGRAMA DE DESPLIEGUE .....	117
4.8	CONCLUSIONES .....	118
	<b>CONCLUSIONES GENERALES</b> .....	<b>119</b>
	<b>RECOMENDACIONES</b> .....	<b>120</b>
	<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	<b>A</b>
	<b>GLOSARIO DE TÉRMINOS</b> .....	<b>B</b>



## Introducción

Hoy en día es muy conocido el proceso que se lleva a cabo para adquirir la licencia de conducción. Uno de los pasos de este proceso consiste en el examen teórico, para evaluar los conocimientos de los temas referentes a las regulaciones del tránsito que poseen los aspirantes.

Normalmente este examen se realiza mediante una prueba escrita en la que el examinado deberá responder una serie de preguntas de temas del tránsito, lo cual puede resultar tedioso en ocasiones, tanto para el los evaluados como para los evaluadores, que deben calificar todos esos exámenes, lo cual puede dar a lugar a errores en la evaluación de los mismos.

Así es que la Universidad de las Ciencias Informáticas (UCI), institución educacional dedicada además a la producción de software, ha entrado en colaboración con la empresa SIMPRO (Simuladores Profesionales), con el objetivo de desarrollar el sistema CISMA para automatizar el proceso de examinación teórica que se realiza para la obtención de la licencia de conducción. Este sistema funciona a través de una aplicación que muestra una interfaz gráfica con videos de situaciones del tránsito con sus preguntas correspondientes, con la que interactúa el evaluado cuando realiza el examen. Además cuenta con otros módulos encargados del registro de los examinados, chequeo de la realización de los exámenes y administración del sistema.

El sistema existente actualmente es muy dependiente de todos esos módulos para su funcionamiento, si falta uno de ellos no funciona correctamente, por lo que cada vez que un cliente necesita eliminar o adicionar un nuevo módulo, se debe reprogramar gran parte del sistema y dar una versión nueva del producto, lo que dificulta en gran medida la implantación del sistema en otras áreas.

De ahí el surgimiento de este trabajo como necesidad de dar respuesta a las situaciones antes expuestas, por lo que el **problema** a resolver consiste en: ¿Cómo eliminar el alto acoplamiento

de los módulos del sistema, de forma tal que solo con un mínimo de esfuerzo de programación, pueda satisfacer las necesidades de cualquier cliente de añadir o eliminar módulos al mismo?

El **objeto de estudio** de este trabajo lo constituye el sistema Evaluador Teórico existente y el **campo de acción** se enmarca en los procesos de evaluación automatizados en dicho sistema y la integración de los módulos del mismo.

El **objetivo general** del presente trabajo de diploma es realizar una reingeniería al sistema para evaluar teóricamente a las personas interesadas en obtener licencia de conducción, desarrollado por el proyecto Auto Teórico de la facultad 5, para obtener un nuevo sistema que permita su usos en cualquier área que requiera una calificación de los conocimientos de la teoría relacionada con dicha actividad, dividido en módulos con bajo nivel de acoplamiento entre ellos y con la plataforma de almacenamiento de datos.

Para dar cumplimiento al objetivo se propone la realización de las siguientes **tareas**:

- Análisis de cómo se encuentran en el ámbito internacional las tecnologías que se utilizan para llevar a cabo herramientas como la que se pretende modificar.
- Desarrollo de la reingeniería un sistema que permita la evaluación teórica de las personas interesadas.

El desarrollo de esta aplicación permitirá minimizar las constantes actualizaciones a las que estaba sometido el sistema Evaluador Teórico. Además se extenderá y comercializará el nuevo sistema a otros clientes nacionales e internacionales.

Con el propósito de desarrollar las tareas planteadas, se utilizaron los métodos de investigación siguientes:

**Métodos Empíricos:** Se utilizaron en la recogida de información y recopilación de los datos relacionados con el producto.

- **Observación:** Para conocer los detalles fundamentales las NTIC (nuevas tecnologías de la informática y las comunicaciones).
- **Entrevista:** Para conocer como funcionan todos los fenómenos que se analizan en el presente trabajo.

**Métodos Teóricos:**

- **Analítico sintético:** La revisión bibliográfica se realizó sobre un conjunto de libros, revistas, publicaciones, monografías y documentos en soporte electrónico, que se encuentran situados en Páginas Web, Internet, Trabajos de Curso, Revistas, etc. y se usó para sintetizar todas las citas, apuntes, datos, etc. tomados al respecto.
- **Inductivo deductivo:** Partiendo del análisis de las generalidades permite llegar a los puntos particulares de este trabajo.

El presente documento se estructura en cuatro capítulos:

**Capítulo 1:** Describe la situación actual, se hace referencia a sistemas automatizados existentes en la realización del exámenes teóricos. Se describen las tendencias y tecnologías actuales a considerar en la selección de aquellas que se van a utilizar en la implementación de la aplicación. Finalmente se plantea dicha selección a modo de propuesta, con cada uno de sus aspectos debidamente fundamentado.

**Capítulo 2:** En este capítulo se realiza el modelado del negocio actual para realizar exámenes teóricos. Se describe el patrón que se sigue para estos tipos de exámenes y la solución técnica propuesta para el sistema

**Capítulo 3:** En este capítulo se exponen los requisitos funcionales y no funcionales del sistema desarrollado. Se definen y describen los casos de uso del sistema que se realizarán en la primera iteración.

**Capítulo 4:** Se realiza la construcción del sistema en los flujos de diseño e implementación, afirmando las bases para una posterior programación de los módulos diseñados.

# Capítulo 1 Fundamentación Teórica

## ***1.1 Introducción.***

La revolución de las Nuevas Tecnologías de la Información y las Comunicaciones (NTIC), con la incorporación de la computadora a los medios electrónicos, continúa asombrando a las personas.

Con el tiempo cada vez son más las personas que dan un paso al frente y comienzan a utilizar las nuevas herramientas digitales, aprovechando las mejoras con respecto a los "métodos tradicionales" y llevando cada tarea a un nivel superior, con más calidad, con posibilidades únicas.

La humanidad es testigo de los cambios de una era industrial a otra donde la supremacía de individuos, organizaciones y naciones pasa por el manejo inteligente de la información. Por lo que se refiere a las tecnologías de los sistemas de información, entendiendo como tales las tecnologías de ordenadores, telecomunicaciones y automatización de oficinas y servicios tradicionales.

En este capítulo se pretende abordar los aspectos y conceptos generales relacionados con el tema de los sistemas de evaluación teórica, una descripción del objeto de estudio y el campo de acción del trabajo, además de una reseña del estado del arte del tema a tratar, tecnologías y tendencias actuales que se usan para el desarrollo de productos como el que se quiere obtener. Se pretende en este capítulo dejar sentadas las bases teóricas para una correcta implementación. Además se explicará las herramientas seleccionadas para dar solución a la problemática propuesta.

## **1.2 Objeto de estudio y campo de acción.**

La empresa SIMPRO es una institución perteneciente a las FAR, que se dedica a la producción de simuladores tanto para la exportación, aportando ganancias significativas al país, como para la utilización a nivel nacional.

La Universidad de Ciencias Informáticas en conjunto con SIMPRO comenzó a hacer gestiones para integrar a un grupo de estudiantes en sus tareas aprovechando las oportunidades y la tecnología que ofrece el centro. Fue así que surgió el proyecto “Simpro” en la Facultad 5. Desde entonces el grupo de trabajo del proyecto ha desarrollado un software, con el objetivo de evaluar las habilidades de los aspirantes a obtener la licencia de conducción, dividido en un Evaluador Teórico y un Evaluador Práctico; de manera tal que el sistema desarrollado con nombre Evaluador Teórico, tiene como objetivo principal informatizar el proceso de realización de exámenes teóricos para acceder al permiso o licencia de conducción.

Este sistema está dividido en varios módulos que pueden variar según la petición del cliente para el cual se realiza. Estos módulos pueden ser:

- **Empleado Bancario:** Donde se chequeará el pago del examen por los aspirantes a obtener licencia de conducción (en los países que se cobre por realizar este examen).
- **Área de Registro:** Donde se realizará el registro de los datos primarios del aspirante.
- **Área de Salud:** Donde se actualizan los parámetros biométricos obtenidos mediante el chequeo médico aplicado al aspirante a obtener licencia de conducción.
- **Módulo de Supervisión:** Desde este módulo se tiene un acceso a los exámenes realizados en los distintos puestos de trabajo, y se chequean una serie de variables como

son tiempo de ejecución del examen, pregunta por la que se encuentra, una imagen del video que se está visualizando en cada puesto, al finalizar cada examen permite imprimir un reporte con los datos del aspirante a obtener la licencia de conducción y su calificación en el examen realizado.

- **Módulo de Examen:** En este módulo se realiza el examen, permite visualizar los videos con las preguntas, da opción de responder las mismas y de repetir el video.
- **Módulo de Administración:** Desde el cual se pueden obtener reportes a partir de todos los datos almacenados por el sistema, administrar las situaciones a presentar en el examen, configurar la cantidad de preguntas de cada temario y otros datos relativos al examen como tiempo máximo de ejecución permitido.

Como se puede ver son varios los módulos existentes en el sistema, aunque no todos son necesarios: se incluyen o eliminan a petición del cliente que compre el producto. De esos módulos algunos pueden funcionar sin la presencia de los otros, como un sistema en sí mismo.

Por lo anteriormente expuesto se debe realizar un estudio de cómo lograr un sistema que pueda asimilar cuantos módulos se le quieran incluir o eliminar y que además pueda tener módulos que funcionen independientemente de la presencia de los otros. El sistema debe ser configurable para cualquier tipo de evaluación teórica, no solamente la que se realiza para obtener licencia de conducir.

### **1.3 Reingeniería**

La reingeniería del software ha sido y es un tema muy importante para muchas empresas y organismos que tienen que seguir manteniendo sus aplicaciones porque sus desarrollos han sido

costosos y adaptados a sus necesidades, lo que en muchos casos hace que no existan aplicaciones comerciales similares. El inconveniente es que estos sistemas con el paso de los años presentan un aspecto obsoleto, mostrando pantallas y diseños ya descartados.

La reingeniería de sistemas tiene por finalidad reestructurar o transformar viejos sistemas en aplicaciones más fáciles de mantener, con entornos más agradables e integradas en nuevas plataformas de hardware/software. **[Álvarez 2004]**.

El Instituto de Ingeniería de software (SEI) desarrolló una definición de Reingeniería como: “la transformación sistemática de un sistema existente dentro de una nueva forma de realizar mejoramientos de calidad en unas operaciones, capacidad del sistema, funcionalidad, rendimiento o capacidad de evolución a bajo costo, agendas o riesgos para el cliente”.**[Carmona 2004]**

La reingeniería casi siempre implica cambiar la forma de un programa y mejorar su documentación. En este caso, la funcionalidad del programa no es cambiada; sólo su forma es modificada. En otros casos, la reingeniería va más allá de la forma e incluye rediseñar para cambiar la funcionabilidad del programa para buscar mejores requerimientos de usuario. En este trabajo de diploma la funcionalidad no es cambiada pero se enriquece permitiendo que el software sea escalable, por lo que se aplican ambas variantes de la reingeniería.

La siguiente lista muestra las razones por las que es aplicable la reingeniería al Sistema Evaluador Teórico existente:

- Frecuentes fallas de producción (fiabilidad cuestionable).
- Problemas de rendimiento.
- Problemas de integración del sistema.
- Código de baja calidad.



- Dificultad al cambio.
- Dificultad para probar.
- Incremento de problemas del sistema.

## ***1.4 Tendencias y tecnologías actuales.***

En el contexto mundial actual existen tendencias en el desarrollo de software que se van imponiendo para lograr productos de alta calidad, y bajos costos de elaboración. La reusabilidad de código, extensibilidad del producto, genericidad de las aplicaciones, son algunos ejemplos de estas tendencias. Se pueden citar otras como el usar varios sistemas gestores de bases de datos o programar aplicaciones que sean independientes de la plataforma de almacenamiento de datos, el uso de una metodología de desarrollo de software como el Proceso Unificado Racional (**RUP**) o la Programación Extrema (**XP**).

### **1.4.1 Aplicaciones genéricas**

La tendencia a realizar aplicaciones genéricas cada vez es mayor, debido a las múltiples facilidades que brinda, construir aplicaciones capaces de ser adaptadas a diferentes realidades es de vital importancia para el desarrollo de software. De esta forma, es posible adaptarse rápidamente a los cambios que ocurren en las organizaciones y reutilizar una aplicación en un amplio espectro de realidades.

La genericidad es un recurso importante de factorización y reusabilidad en la programación, las funciones y los tipos genéricos soportados por la programación funcional son una buena base para introducir conceptos de reutilización de código.

Sin duda alguna realizar aplicaciones genéricas por estos días es la mejor opción para lograr un exitoso desarrollo de software.

### **1.4.2 Modularidad**

Un problema complejo no puede solucionarse de una sola vez, por eso una de las técnicas más utilizadas en la programación es la de dividir este problema en sub-problemas más sencillos, que den la solución al problema original. Para resolver estos sub-problemas se necesita crear y trabajar con subprogramas llamados módulos, de ahí el nombre de programación modular.

La programación modular consta de varias secciones divididas de forma que interactúan a través de llamadas a procedimientos, que integran el programa en su totalidad. El programa principal coordina las llamadas a los módulos secundarios y pasa los datos necesarios en forma de parámetros.

De ahí la importancia de este tipo de programación, es que sus módulos son independientes, es decir pueden ser modificados o reemplazados sin afectar el resto del programa o pueden ser reutilizado en otros programas.

La ventaja de utilizar el diseño modular es que sus módulos pueden escribirse y probarse por separados y son más fáciles de mantener y documentar.

### **1.4.3 Extensibilidad**

La extensibilidad de un programa o aplicación informática es la capacidad que presentan de adaptarse a los cambios de especificación.

Para programas pequeños realizar cambios de los requisitos, de los algoritmos, de la representación de los datos, de las técnicas de implementación, etc., no es una tarea difícil; pero a medida que el software crece comienza a ser cada vez más difícil de adaptar, aunque la extensibilidad se pueden aplicar con pequeños programas, su importancia sólo se refleja con claridad en los grandes proyectos.

Los principios esenciales de la extensibilidad son:

**Simplicidad del diseño:** una arquitectura simple siempre será más fácil de adaptar a los cambios que una compleja.

**Descentralización:** cuanto más autónomos sean los módulos, existe una mayor probabilidad de que un cambio afecte a un solo módulo, o a un número pequeño de módulos, en lugar de provocar una reacción en cadena de cambios en el sistema completo.

#### 1.4.4 Sistemas Gestores de Bases de Datos (SGBD)

Los sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.[Wikipedia 1]

El propósito de los sistemas de gestión de bases de datos es el de manejar de manera sencilla y ordenada un conjunto de información almacenada en las bases de datos, para ello deben tener ciertos objetivos:

- **Abstracción de la información.** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.
- **Independencia.** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Redundancia mínima.** Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no

obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.

- **Consistencia.** En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- **Seguridad.** La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra asegurada frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- **Integridad.** Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.
- **Respaldo y recuperación.** Los SGBD deben proporcionar una forma eficiente de realizar copias de seguridad de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.
- **Control de la concurrencia.** En la mayoría de entornos (excepto quizás el doméstico), lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. Y es también frecuente que dichos

accesos se realicen de forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.

- **Tiempo de respuesta.** Lógicamente, es deseable minimizar el tiempo que el SGBD tarda en darnos la información solicitada y en almacenar los cambios realizados.

### **Ventajas de los sistemas de gestión de bases de datos:**

- Facilidad de manejo de grandes volúmenes de información.
- Gran velocidad en muy poco tiempo.
- Independencia del tratamiento de información.
- Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.
- No hay duplicidad de información, comprobación de información en el momento de introducir la misma.
- Integridad referencial al terminar los registros.

### **Desventajas:**

- El costo de actualización del hardware y software son muy elevados.
- Costo (salario) del administrador de la base de datos es costoso.
- El mal diseño de esta puede originar problemas a futuro.
- Un mal adiestramiento a los usuarios puede originar problemas a futuro.
- Si no se encuentra un manual del sistema no se podrán hacer relaciones con facilidad.
- Generan campos vacíos en exceso.
- El mal diseño de seguridad genera problemas en esta.

Los SGBD se clasifican en libres y comerciales, los libres son aquellos que no necesitan de pago de una licencia para su uso, los comerciales son los que se debe comprar una licencia para poder instalarlos y usarlos de forma legal. Entre los SGBD libres o gratuitos se encuentran principalmente MySQL, PostgreSQL, Firebird, en el grupo de los comerciales se hallan Microsoft SQL Server, dBase, Oracle, Paradox.

## **MySQL**

**MySQL** es un sistema de gestión de base de datos, multihilo y multiusuario con más de seis millones de instalaciones. **[Schumacher]**

Este sistema fue programado casi completamente en ANSI C y es desarrollado por una empresa privada lo que facilita que la licencia sea dual: gratuita para la comunidad y comercial para empresas que quieran incluirlo en productos privativos. MySQL es usado principalmente para el trabajo en la web en muchos sitios en Internet, funciona sobre múltiples plataformas. Como rasgos distintivos este sistema presenta múltiples motores de almacenamiento de datos como MyISAM, InnoDB, Merge, etc.

## **Firebird**

Firebird es un sistema de administración de base de datos de código abierto, basado en la versión 6 de Interbase, cuyo código fue liberado por Borland en 2000. **[Wikipedia 2]**

Es multiplataforma, y actualmente puede ejecutarse en los sistemas operativos: Linux, HP-UX, FreeBSD, Mac OS, Solaris y Microsoft Windows. Ejecutable pequeño, con requerimientos de hardware bajos. Arquitectura Cliente/Servidor sobre protocolo TCP/IP. Soporta transacciones y claves foráneas. Es medianamente escalable. Posee una buena seguridad basada en usuarios y roles. Presenta diferentes arquitecturas, entre ellas el Firebird empotrado (embedded server) que permite ejecutar aplicaciones monousuario en ordenadores sin instalar el software Firebird, bases

de datos de sólo lectura, para aplicaciones que corran desde dispositivos sin capacidad de escritura, como CD-Roms. En Firebird existen controladores ODBC, OLEDB y JDBC. Tiene requisitos de administración bajos, siendo consideradas sus bases de datos como libres de mantenimiento, al margen de la realización de copias de seguridad y restauraciones periódicas. Firebird brinda pleno soporte del estándar SQL-92, tanto de sintaxis como de tipos de datos y un Completo lenguaje para la escritura de disparadores (*triggers*) y procedimientos almacenados. Y además tiene capacidad de almacenar elementos BLOB (*binary large objects*).

### **Microsoft SQL Server**

SQL Server es un completo Sistema de Gestión de Base de Datos Relacional con todas las posibilidades que tienen estas aplicaciones; un servidor disponible en varias ediciones en el que es posible utilizar un lenguaje, Transact-SQL, para programar procedimientos almacenados y desencadenadores, aparte de poder definirse tablas, índices, vistas, etc.

MS SQL Server, es un potente motor de bases de datos de alto rendimiento, capaz de integrarse con herramientas de desarrollo como Visual Studio .NET, incorporando un modelo de objetos totalmente programable (SQL-DMO) con el cual se pueden desarrollar aplicaciones que usen componentes de SQL Server.

Microsoft SQL Server 2000 es una versión del sistema de gestión de bases de datos relacionales (SGBD), cliente servidor, basado en un lenguaje de consulta estructurada (SQL, Structured Query Language), que aprovecha la sólida base establecida por su predecesor SQL Server 6.5. y 7. Como la mejor base de datos para Windows NT, MS SQL Server 2000 es el SGBDR ideal para un amplio campo de clientes corporativos y fabricantes independientes de software. Las necesidades y requisitos del cliente, han dado lugar a innovaciones significativas en SQL Server 2000, entre las que se incluyen la facilidad de uso, escalabilidad, fiabilidad, y almacenamiento de datos. Proporcionando soporte para un conjunto de características que aportan las siguientes ventajas:

- **Facilidad de instalación, distribución y utilización:** SQL Server incluye un conjunto de herramientas administrativas y de desarrollo que mejoran la capacidad para instalar, distribuir, administrar y utilizar SQL Server entre varios sitios.
- **Escalabilidad:** Puede utilizarse el mismo motor de base de datos a través de plataformas que van desde equipos portátiles que ejecutan Microsoft Windows® 9X, 200X o hasta grandes servidores con varios procesadores que ejecutan Microsoft Windows NT®, Enterprise Edition.
- **Almacenamiento de datos:** SQL Server incluye herramientas para extraer y analizar datos resumidos para el proceso analítico en línea (OLAP, Online Analytical Processing). SQL Server incluye también herramientas para diseñar gráficamente las bases de datos y analizar los datos mediante preguntas en lenguaje normal.
- **Integración del sistema con otro software de servidor:** SQL Server se integra con el correo electrónico, Internet y Windows.

## XML

XML proviene de un lenguaje desarrollado por IBM en los años setenta, llamado GML (General Markup Language), que surgió por la necesidad que tenía la empresa de almacenar grandes cantidades de información. Este lenguaje gustó a la **ISO**, por lo que en 1986 trabajaron para normalizarlo, creando **SGML** (Standard General Markup Language), capaz de adaptarse a un gran abanico de problemas. A partir de él se han creado otros sistemas para almacenar información.

**XML**, sigla en inglés de Extensible Markup Language (Lenguaje de Marcas Extensible), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (**W3C**). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos



(de la misma manera que **HTML** es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. **[Wikipedia 3]**

XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Ventajas de XML:

- Comunicación de datos. Si la información se transfiere en XML cualquier aplicación podría escribir un documento de texto plano con los datos que estaba manejando en formato XML y otra aplicación recibir esta información y trabajar con ella.
- Migración de datos. Si trabajamos en formato XML sería muy sencillo mover datos de una base de datos a otra.
- Aplicaciones Web. Con XML hay una sola aplicación que maneja los datos y para cada navegador podemos tener una hoja de estilo o similar para aplicarle el estilo adecuado.

La tecnología XML busca dar solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible. Que la información sea estructurada quiere decir que se compone de partes bien definidas, y que esas partes se componen a su vez de otras partes. Entonces se tiene un árbol de pedazos de información.

Una etiqueta consiste en una marca hecha en el documento, que señala una porción de este como un elemento, un pedazo de información con un sentido claro y definido. Las etiquetas tienen la forma <nombre>, donde nombre es el nombre del elemento que se está señalando.

A continuación se muestra un ejemplo para entender la estructura de un documento XML:

```
<?xml version="1.0"?>
<!DOCTYPE MENSAJE SYSTEM "mensaje.dtd">
<mensaje>
  <remitente>
    <nombre>Alfredo Reino</nombre>
    <mail>alf@ibium.com</mail>
  </remitente>
  <destinatario>
    <nombre>Juan Carlos Quevedo</nombre>
    <mail>jquevedo@gmail.com</mail>
  </destinatario>
  <asunto>Hola Juan Carlos</asunto>
  <texto>
    <párrafo>¿Hola que tal? Hace <énfasis>mucho</énfasis> que
    no escribes. A ver si llamas y quedamos para tomar algo. </párrafo>
  </texto>
</mensaje>
```

### **1.5 Metodologías propuestas.**

El Proceso Unificado es un proceso de desarrollo de software (conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software). Es un marco

de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos.

El Proceso Unificado está basado en componentes. Utiliza el lenguaje unificado de modelado (UML) para preparar todos los esquemas de un sistema de software. De hecho, UML es una parte esencial de RUP, sus desarrollos fueron paralelos. No obstante los verdaderos aspectos definitorios del proceso unificado se resumen en tres fases claves: dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental. [ **Corporation 2004**]

### **1.5.1 UML (Unified Modeling Language)**

UML (Unified Modeling Language) o Lenguaje de Modelación Unificado es un lenguaje gráfico para detallar, construir, visualizar y documentar las partes o artefactos (información que se utiliza o produce mediante un proceso de software). Pueden ser artefactos: un modelo, una descripción que comprende el desarrollo de software que se basen en el enfoque Orientado a Objetos. UML usa procesos de otras metodologías, aprovechando la experiencia de sus creadores, eliminó los componentes que resultaban de poca utilidad práctica y añadió nuevos elementos. [ **Histchfeld**]

UML es un lenguaje más expresivo, claro y uniforme que los anteriores definidos para el diseño Orientado a Objetos, que no garantiza el éxito de los proyectos pero si mejora sustancialmente el desarrollo de los mismos, al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios.

De forma general las principales características son: [ **Caro 2004**]

- Lenguaje unificado para la modelación de sistemas.
- Tecnología orientada a objetos.
- El cliente participa en todas las etapas del proyecto.
- Corrección de errores viables en todas las etapas.

- Aplicable para tratar asuntos de escala inherentes a sistemas complejos. de misión crítica, tiempo real y cliente/servidor.

UML es desde finales de 1997, un lenguaje de modelado orientado a objetos estándar, de acuerdo con el Object Management Group, siendo utilizado diariamente por grandes organizaciones como: Microsoft, Oracle, Rational.

### **1.5.2 Rational Rose.**

Existen herramientas Case de trabajo visuales como el Analise, el Designe, el Rational Rose, que permiten realizar el modelado del desarrollo de los proyectos, en la actualidad la mejor y más utilizada en el mercado mundial es Rational Rose y es la que se utiliza en la modelación de este proyecto. Rational Rose es la herramienta de modelación visual que provee el modelado basado en UML **[Rational]**

La Corporación Rational ofrece un Proceso Unificado Racional (RUP) para el desarrollo de los proyectos de software, desde la etapa de Ingeniería de Requerimientos hasta la de Pruebas. Para cada una de estas etapas existe una herramienta de ayuda en la administración de los proyectos, Rose es la herramienta del Rational para la etapa de análisis y diseño de sistemas. **[Rational]**

Rational Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros de equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Una de las grandes ventajas de Rose es que utiliza la notación estándar en la arquitectura de software(UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común, además los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto.

## **1.6 Lenguajes de Programación**

Al desarrollarse las primeras computadoras electrónicas, se vio la necesidad de programarlas, es decir, de almacenar en memoria la información sobre la tarea que iban a ejecutar.

Los lenguajes más primitivos fueron los lenguajes de máquina, ya que el hardware se desarrolló antes del software, y además cualquier software finalmente tiene que expresarse en el lenguaje que maneja el hardware.

El primer gran avance que se dio, fue la abstracción dada por el Lenguaje Ensamblador, y con él, el nacimiento de las primeras herramientas automáticas para generar el código máquina. Esto redujo los errores triviales, que son sumamente engorrosos y difíciles de detectar, pero fáciles de cometer. Sin embargo, aún aquí es fácil para el programador perderse y cometer errores de lógica, pues debe bajar al nivel de la forma en que trabaja el CPU, y entender bien todo lo que sucede dentro de él.

Con el desarrollo en los 50s y 60s de algoritmos de más elevado nivel, y el aumento de poder del hardware, empezaron a entrar al uso de computadoras científicos de otras ramas; ellos conocían mucho de Física, Química y otras ramas similares, pero no de Computación, y por supuesto, les era sumamente complicado trabajar con lenguaje Ensamblador en vez de fórmulas. Así, nació el concepto de Lenguaje de Alto Nivel, con el primer compilador de FORTRAN (FORmula TRANslation), que, como su nombre indica, inició como un "simple" esfuerzo de traducir un lenguaje de fórmulas, al lenguaje ensamblador y por consiguiente al lenguaje de máquina. A partir de FORTRAN, se han desarrollado innumerables lenguajes, que siguen el mismo concepto: buscar la mayor abstracción posible, y facilitar la vida al programador, aumentando la productividad, encargándose los compiladores o intérpretes de traducir el lenguaje de alto nivel, al lenguaje de computadora.

## **El lenguaje C**

C es un lenguaje de programación diseñado por Dennis Ritchie, de los Laboratorios Bell, y se instaló en un PDP-11 en 1972; se diseñó para ser el lenguaje de los Sistemas Operativos UNIX1. A su vez, UNIX es un Sistema Operativo desarrollado por Ken Thompson, quién utilizó el lenguaje ensamblador y un lenguaje llamado B para producir las versiones originales de UNIX, en 1970. C se diseñó para superar las limitaciones de B. C es un lenguaje maduro de propósitos generales que se desarrolló a partir de estas raíces; su definición aparece en 1978 en el apéndice "C Reference Manual" del libro "The C Programming Language", de Brian W. Kernighan y Dennis M. Ritchie (Englewood Cliffs, Nueva Jersey, Prentice-Hall 1978), pero el estándar recomendable más reciente apareció en junio de 1983, en el documento de los Laboratorios Bell titulado "The C Programming Language-Reference Manual", escrito por Dennis M. Ritchie.

## **El lenguaje C++**

C++ es un lenguaje de programación, diseñado a mediados de los años 1980, por Bjarne Stroustrup, como extensión del lenguaje de programación C.

Actualmente existe un estándar, denominado ISO C++, al que se han adherido la mayoría de los fabricantes de compiladores más modernos. Existen también algunos intérpretes como ROOT ([enlace externo](#)).

Las principales características del C++ son el soporte para programación orientada a objetos y el soporte de plantillas o programación genérica (templates). Se puede decir que C++ es un lenguaje que abarca tres paradigmas de la programación: la programación estructurada, la programación genérica y la programación orientada a objetos.

Además posee una serie de propiedades difíciles de encontrar en otros lenguajes de alto nivel:

- Posibilidad de redefinir los operadores (sobrecarga de operadores)
- Identificación de tipos en tiempo de ejecución.

C++ está considerado por muchos como el lenguaje más potente, debido a que permite trabajar tanto a alto como a bajo nivel, sin embargo es a su vez uno de los que menos automatismos trae (obliga a hacerlo casi todo manualmente al igual que C) lo que dificulta mucho su aprendizaje.

El nombre C++ fue propuesto por Rick Masciatti en el año 1983, cuando el lenguaje fue utilizado por primera vez fuera de un laboratorio científico. Antes se había usado el nombre "C con clases". En C++, "C++" significa "incremento de C" y se refiere a que C++ es una extensión de C.

### **El lenguaje Java**

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los 90. A diferencia de los lenguajes de programación convencionales, que generalmente están diseñados para ser compilados a código nativo, Java es compilado en un bytecode que es interpretado (usando normalmente un compilador JIT), por una máquina virtual Java.

El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos mucho más simple y elimina herramientas de bajo nivel como punteros.

La independencia de la plataforma de Java, significa que programas escritos en este lenguaje pueden ejecutarse igualmente en cualquier tipo de hardware. Es lo que significa ser capaz de escribir un programa una vez y que pueda ejecutarse en cualquier dispositivo, tal como reza el axioma de Java, "write once, run everywhere" ("una vez escrito, corre dondequiera").

Para ello, se compila el código fuente escrito en lenguaje Java, para generar un código conocido como "bytecode": instrucciones máquina simplificadas específicas de la plataforma Java. Esta

pieza está “a medio camino” entre el código fuente y el código máquina que entiende el dispositivo destino. El *bytecode* es ejecutado entonces en la máquina virtual (VM), un programa escrito en código nativo de la plataforma destino (que es el que entiende su hardware), que interpreta y ejecuta el código.

Aunque cada vez la tecnología Java se acerca más y más al PC de mesa, las aplicaciones Java han sido relativamente raras para uso de escritorio, por varias razones:

- Las aplicaciones Java pueden necesitar gran cantidad de memoria física.
- La Interfaz Gráfica de Usuario (GUI) no sigue de forma estricta la Guía para Interfaces Humanas (Human Interface Guidelines), así como tampoco aquella a la que estamos habitualmente acostumbrados. La apariencia de las fuentes no tiene las opciones de optimización activadas por defecto, lo que hace aparecer al texto como si fuera de baja calidad.
- Las herramientas con que cuenta el JDK no son suficientemente potentes para construir de forma simple aplicaciones potentes. Aunque el uso de herramientas como Eclipse, un IDE con licencia libre de alta calidad, facilita enormemente las tareas de desarrollo.
- Hay varias versiones del Entorno en Tiempo de Ejecución de Java, el JRE. Es necesario tener instalada la versión adecuada.
- Las aplicaciones basadas en la Web están tomando la delantera frente a aquellas que funcionan como entidades independientes. Las nuevas técnicas de programación producen aplicaciones basadas en un modelo en red cada vez más potentes.

## **El lenguaje C#**

C# es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA e ISO.



Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET el cual es similar al de Java aunque incluye mejoras derivadas de otros lenguajes (más notablemente de Delphi y Java). C# fue diseñado para combinar el control a bajo nivel de lenguajes como C y la velocidad de programación de lenguajes como Visual Basic.

C#, como parte de la plataforma .NET, está normalizado por ECMA desde diciembre de 2001 (ECMA-334 "Especificación del Lenguaje C#"). El 7 de noviembre de 2005 acabó la beta y salió la versión 2.0 del lenguaje que incluye mejoras tales como tipos genéricos, métodos anónimos, iteradores, tipos parciales y tipos anulables. Ya existe la versión 3.0 de C# en fase de beta destacando los tipos implícitos y el LINQ (Language Integrated Query).

Aunque C# forma parte de la plataforma .NET, ésta es una interfaz de programación de aplicaciones; mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Aunque aún no existen, es posible implementar compiladores que no generen programas para dicha plataforma, sino para una plataforma diferente como Win32 o UNIX.

### ***1.7 Análisis de otras soluciones existentes.***

Los accidentes de tránsito son una de las principales causas de muerte en todo el mundo. Por eso, hay gran preocupación por desarrollar nuevas tecnologías con el fin de prevenirlos y disminuir sus consecuencias.

El examen teórico de conducción es un trámite que se realiza por muchas personas en el mundo diariamente, sirve para medir los conocimientos de cada aspirante a obtener la licencia de conducir. Con la tendencia que existe de informatizar cada servicio o actividad de la vida diaria este trámite no pasa desapercibido a esta inclinación de la tecnología. En varios lugares del mundo se automatiza el proceso de evaluación teórica para obtener licencia de conducción.

En Cuba actualmente no existe ninguna aplicación que automatice el proceso de evaluación para obtener licencia de conducción. Por otra parte a nivel internacional en Suecia, se automatiza el proceso de evaluación teórica, mediante el sistema ProviT, en la página web <http://www.ore.vv.se/provit/main.htm> se puede acceder a una versión demo de dicho sistema. El examen se hace según los siguientes requisitos:[Sin\_autor]

- El examen se hace con un ordenador. Hay varias opciones, pero sólo es correcta una de las respuestas para cada pregunta.
- Para las categorías A, B y para una licencia de tractor, dispondrá de 50 minutos para responder 70 preguntas.
- El examen incluye 5 preguntas de “prueba” que no se contabilizan en el resultado final. Esto se hace con la intención de encontrar preguntas adecuadas para utilizarlas en exámenes teóricos en el futuro.
- Cuando haya finalizado el examen, se mostrará inmediatamente en la pantalla el resultado.
- Para aprobar en las categorías A, B y una licencia de tractor, debe obtener 52 puntos. En las otras categorías de licencia se requieren 44 puntos para aprobar el examen. Se le dará un punto por cada respuesta correcta.


Analizando la aplicación demo existente de dicho producto se puede concluir que las preguntas del examen teórico aparecen en un texto y acompañadas visualmente de imágenes que ayudan a comprender su significado. Tienen varias opciones para responder pero una sola es la correcta. En el sitio web donde se muestra la versión demo no aparece ninguna referencia a si el software es o no gratuito.

← Back    → Forward    Overview    Highlight

**Practise question 1(2)**  
Answer the question by choosing letter A, B, C or D. You can see an enlarged image of any picture simply by clicking on it. Click on the Forward-button to move to the next question.

Which alternative shows the correct maximum speed limit for the vehicles shown in pictures A-D (each caravan/trailer is equipped with a braking system)?

<b>A</b>	Picture A 70 km/h
<b>B</b>	Picture B 80 km/h
<b>C</b>	Picture C 100 km/h
<b>D</b>	Picture D 110 km/h



**Ilustración 1. Interfaz gráfica de la versión demo de ProVIIT.**

En Chile, en la comunidad municipal de Huechuraba, también se aplica el examen por computadora, seleccionándose una cantidad aleatoria según sea la categoría del examen a realizar. El examen se realiza en una página web donde se aparecen las preguntas y se pueden responder en la misma. En la página web <http://www.huechuraba.cl/webint/transito.htm> se encuentra disponible información acerca de los trámites a realizar para examinarse y acerca de los diferentes tipos de examen que se aplican.

En Perú, también se usa un sistema automatizado para la realización del examen teórico. Una versión demo del sistema puede ser hallada en la dirección web: [www.brevetesperu.com](http://www.brevetesperu.com).

Actualmente en Republica Dominicana y en Guatemala se aplica la primera versión del sistema evaluador teórico CISMA, el cual es nuestro objeto de estudio.

El mismo sistema aplica como algo novedoso la proyección de videos con situaciones del tránsito para a raíz de las mismas plantear una interrogante al aspirante a obtener la licencia de conducción. La pregunta se muestra en la pantalla al finalizar el video en forma de texto, y en una audición. Las preguntas tienen solo dos formas de responder: Verdadero o Falso, y las respuestas se hacen mediante dos botones que presenta la interfaz gráfica del software, esto se

debe a que la población de estos dos países son en su mayoría analfabetos y se quería tener un sistema que no fuese complicado de manipular por parte de los aspirantes a obtener licencia de conducción, el sistema cuenta además con varios módulos que fueron descritos en el epígrafe 1.2.

Los otros sistemas encontrados no explotan mucho la posibilidad de brindar información gráfica y auditiva de las preguntas del examen, el único uso que le dan a elementos gráficos es mediante imágenes con señalizaciones del tránsito. Otra deficiencia que tienen los sistemas que aparecen a nivel internacional es una interfaz gráfica bastante deficiente en cuanto a diseño, con poca estética visual. Lo positivo de estos sistemas es que las preguntas tienen varias opciones para responder, no solo verdadero y falso como el sistema CISMA.

También existen diversos sitios en Internet que siguen la óptica de test evaluativos, donde los usuarios realizan una especie de examen para ver si es óptimo el nivel de conocimientos que poseen con respecto a las leyes del tránsito, pero sólo con este propósito.

### **1.8 Propuesta de solución técnica**

Para solucionar la problemática planteada en este trabajo de diploma, se construye un sistema que sirva para la evaluación de conocimientos teóricos, y que además constituya un *framework* para facilitar el desarrollo de otros sistemas que tengan el mismo objetivo. Con módulos bien definidos, basado en interfaces de comunicación entre ellos que permitan su posterior uso en otras aplicaciones. Este sistema es genérico, con el objetivo de ampliar el área de aplicación del mismo a cualquier entidad que necesite de un sistema parecido para automatizar el proceso de evaluación de conocimientos teóricos y aumentar al mismo tiempo el mercado de venta del sistema y por consiguiente las ganancias que puede aportar por concepto de comercialización.

La metodología usada para desarrollar el proyecto es RUP, que garantiza la elaboración de todas las fases de un producto de software orientado a objeto. RUP no es simplemente un proceso, sino un marco de trabajo extensible que puede ser adaptado para diferentes áreas de aplicación, diferentes organizaciones o proyectos específicos. Esta metodología entre sus múltiples ventajas ayuda a asegurar la producción de software de alta calidad con un costo y tiempo predecible para el usuario.

En vistas de que se quiere realizar un software de calidad y siguiendo metodologías de desarrollo de software como RUP, donde se usa la programación orientada a objetos, debemos desechar el lenguaje C, ya que este no incorpora el paradigma de Programación Orientado a Objetos (**POO**), el lenguaje de programación Java, nos brinda facilidades a la hora de programar, pues es multiplataforma, y podríamos obtener un software que corriera en cualquier sistema operativo, pero tiene los inconvenientes de que las aplicaciones de escritorio son poco comunes en este lenguaje de programación, debido a lo lentas que resultan y otras inconveniencias que posee el mismo, si a esto se le añade que las máquinas virtuales de java para compilar el código de nuestra aplicación poseen licencia comercial (deben ser pagadas por su uso), esto conlleva a desechar también este lenguaje de programación, en el caso del C#, la no existencia de compiladores para este lenguaje que no generen los programas para la plataforma .NET, y el hecho de que los compiladores de este lenguaje de programación también poseen licencia comercial, indican que la solución factible para programar el software deseado es hacerlo en C++, si se programa usando los tipos básicos de este lenguaje, se haría más factible una migración a software libre, ya que las versiones actuales del sistema solo corren en plataforma Win32.

Para la realización de este trabajo se tiene en cuenta que el sistema debe ser independiente de la plataforma de almacenamiento de los datos, por lo que se crea un sistema de clases de acceso a datos para la manipulación de los mismos, que pueden estar en cualquier tipo de servidor de bases de datos, en la primera versión se usarán los gestores MySQL, Firebird y

Microsoft SQL Server 2000, además del uso de ficheros XML para cuando el sistema deba correr como una aplicación en modo solitario (solamente se ejecuta el módulo del examen, con vistas a preparar a los aspirantes).

## **1.9 Conclusiones**

En el presente capítulo se realizó un análisis del panorama de la evaluación teórica para obtener licencia de conducción y de la automatización de este proceso y sus trámites relacionados.

Además se realizó un análisis de las metodologías usadas para el desarrollo de aplicaciones, de las herramientas CASE para modelar, sistemas gestores de bases de datos y lenguajes de programación a usar para realizar el software deseado.

## **Capítulo 2 Modelo del negocio**

### ***2.1 Introducción***

En este capítulo se realiza un análisis del proceso que se lleva a cabo para realizar el examen teórico para cualquier actividad en la que este tipo de examinación se haga necesario, para facilitar una mayor comprensión del mismo. Se parte de la base del proceso a realizar para obtener licencia de conducción

Se presenta el modelo del negocio con el objetivo de entender el plano en que se emplaza el sistema y por tanto contribuir a la comprensión de los requisitos del sistema que se desprenden de este contexto. Se determinan los actores y trabajadores del negocio y se plantean las principales reglas del negocio a tener en cuenta en el sistema a desarrollar. Finalmente se muestran los casos de uso del negocio detallados y una representación del modelo de objetos.

### ***2.2 Modelo del negocio actual***

En la mayoría de las entidades que se necesita realizar una evaluación teórica de cualquier tema, no se sigue un proceso definido, pero partiendo de los pasos establecidos para realizar el examen teórico de conducción, se pretende generalizar algunos de los mismos que pueden ser aplicables en cualquier proceso de evaluación teórica que se requiera.

En la examinación teórica para licencia de conducción automatizada en el sistema Evaluador Teórico se realiza un examen médico para determinar si la persona está lista para conducir sin perjuicio para su salud ni para la sociedad, siendo este examen el primer paso de este proceso de evaluación. Una vez realizado este chequeo médico, el cliente se presenta en la entidad donde se realiza el examen teórico para inscribirse, se le toman sus datos y los resultados del examen médico, si estos son válidos, se registra y se le informa de la fecha en que debe

presentarse a realizar el examen, cuando el cliente se presenta nuevamente a realizar el examen se le entrega el temario y se le informa de las características del mismo, al terminar de responder las preguntas, entrega el temario, posteriormente se le califica el mismo y se le informa su evaluación, ya sea aprobado o no, después de esto, si el cliente tiene alguna inconformidad con los resultados del examen puede dirigirse al personal encargado de estas reclamaciones donde se tramita su solicitud de revisión.

Los pasos anteriores pueden ser aplicables a cualquier área que necesite evaluar los conocimientos teóricos para desempeñar una labor determinada. Solamente cambia el contenido del examen para cada entidad, pero los pasos en sí mismos pueden ser generalizados, son muchos los procesos y labores que requieren de conocimientos teóricos para poder desempeñar correctamente las actividades y responsabilidades que derivan de su contenido de trabajo, ejemplo de estas labores son la manipulación de equipos para la perforación de pozos petroleros, el trabajo con equipos pesados, el manejo de químicos, entre otros procesos donde un desconocimiento de cómo realizar estas labores puede ocasionar una catástrofe de gran magnitud, resultando en perjuicio de la salud de los trabajadores implicados y en pérdidas para la entidad afectada, por ello es importante tener una forma de evaluar a las personas interesadas en desempeñar las labores descritas anteriormente, para determinar su idoneidad para estas actividades.

### ***2.3 Reglas del negocio a considerar***

Se identificaron varias reglas que debe seguir el sistema que se desarrolle para respetar y garantizar las restricciones existentes en el negocio:

1. El cliente para ser inscrito con el objetivo de realizar el examen teórico debe haber aprobado el examen médico.
2. Si no está inscrito no puede realizar el examen teórico.



3. El tiempo de duración del examen, la nota mínima de aprobado, así como la cantidad de preguntas del temario, pueden ser configurados por los responsables de aplicar el examen, en dependencia del tipo de examen
4. El cliente puede reclamar su evaluación en el examen si no está de acuerdo con la misma.
5. El cliente puede reclamar en caso de que existan errores en su inscripción.
6. Los empleados de la entidad encargada de realizar el examen teórico no podrán variar la nota del examen que se obtenga a través del sistema

## 2.4 Actores del negocio

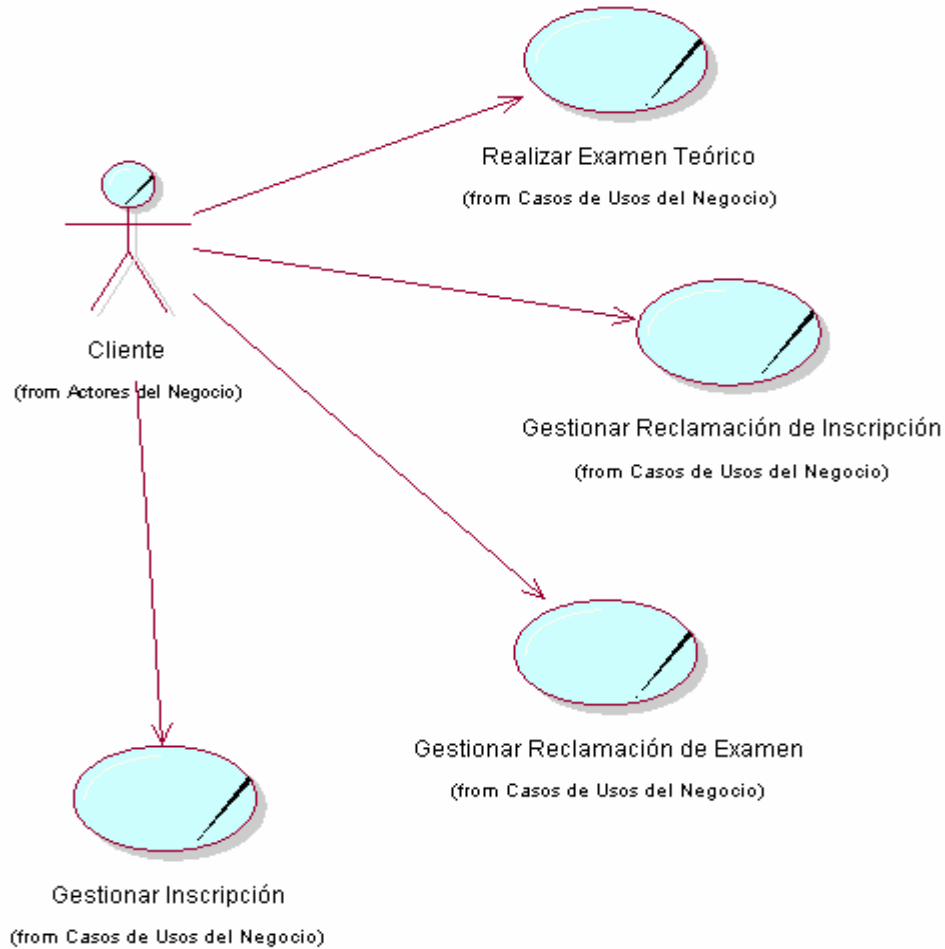
En el negocio analizado existe un solo actor que es quien se beneficia con el mismo: el cliente.

Tabla 1. Descripción de los actores del negocio

Nombre del actor	Descripción
Cliente	Persona interesada en realizar un examen teórico escrito, y que luego de realizar el examen práctico también, con resultados satisfactorios en ambos exámenes, obtiene la licencia de conducción de autos. El cliente es el actor que se beneficia en este proceso.

## 2.5 Diagrama de casos de uso del negocio

La siguiente imagen representa el diagrama de casos de uso del negocio, identificados al analizar todo el proceso del negocio actual. Se identifican cuatro casos de uso: Gestionar Inscripción, Realizar Examen Teórico, Gestionar Reclamación de Inscripción y Gestionar Reclamación de Examen.



**Ilustración 2. Diagrama de casos de uso del Negocio**

## 2.6 Trabajadores del negocio

**Tabla 2. Descripción de los trabajadores del negocio**

Nombre del trabajador	Descripción
Registrador	Es el encargado de inscribir los datos de las personas interesadas en realizar el examen teórico

Examinador	Es el responsable de chequear que exista capacidad para que el cliente realice el examen teórico, chequear que esté inscrito para examinarse, entregarle el examen a realizar y una vez culminado este, darle una evaluación e informar al cliente de la misma
Supervisor de reclamaciones	Es el responsable de atender las solicitudes de reclamación ante inconformidades por parte del cliente en el registro de sus datos o en los resultados de su examen y darle una respuesta del proceso de reclamación.

## 2.7 Casos de uso del negocio

A continuación se describe cada caso de uso del negocio y se presenta una imagen del diagrama de actividades de cada caso de uso.

### 2.7.1 Caso de uso Realizar Examen Teórico

Tabla 3. Descripción del caso de uso Realizar Examen Teórico.

<b>Nombre del Caso de Uso</b>	Realizar Examen Teórico.
<b>Actores</b>	Cliente(inicia)
<b>Propósito</b>	Realizar al cliente el examen Teórico.
<b>Resumen</b>	El caso de uso se inicia cuando el cliente se presenta en el Área del examen teórico para realizar la prueba. Es atendido por el examinador que es el encargado de tomarle sus datos y revisar que el mismo esté registrado en el listado de pendientes a evaluar en esa fecha. En caso de que no, se le informa al cliente que no está inscrito para realizar el examen ese día y se culmina el caso de uso, en caso de encontrarse registrado se le entrega el temario

	<p>del examen, se le explican los requisitos y características de la evaluación. Al culminar de responder las preguntas entrega el temario al examinador, el cual lo revisa y le informa al cliente de la evaluación alcanzada. Posteriormente el examen es archivado en el registro de exámenes realizados.</p>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

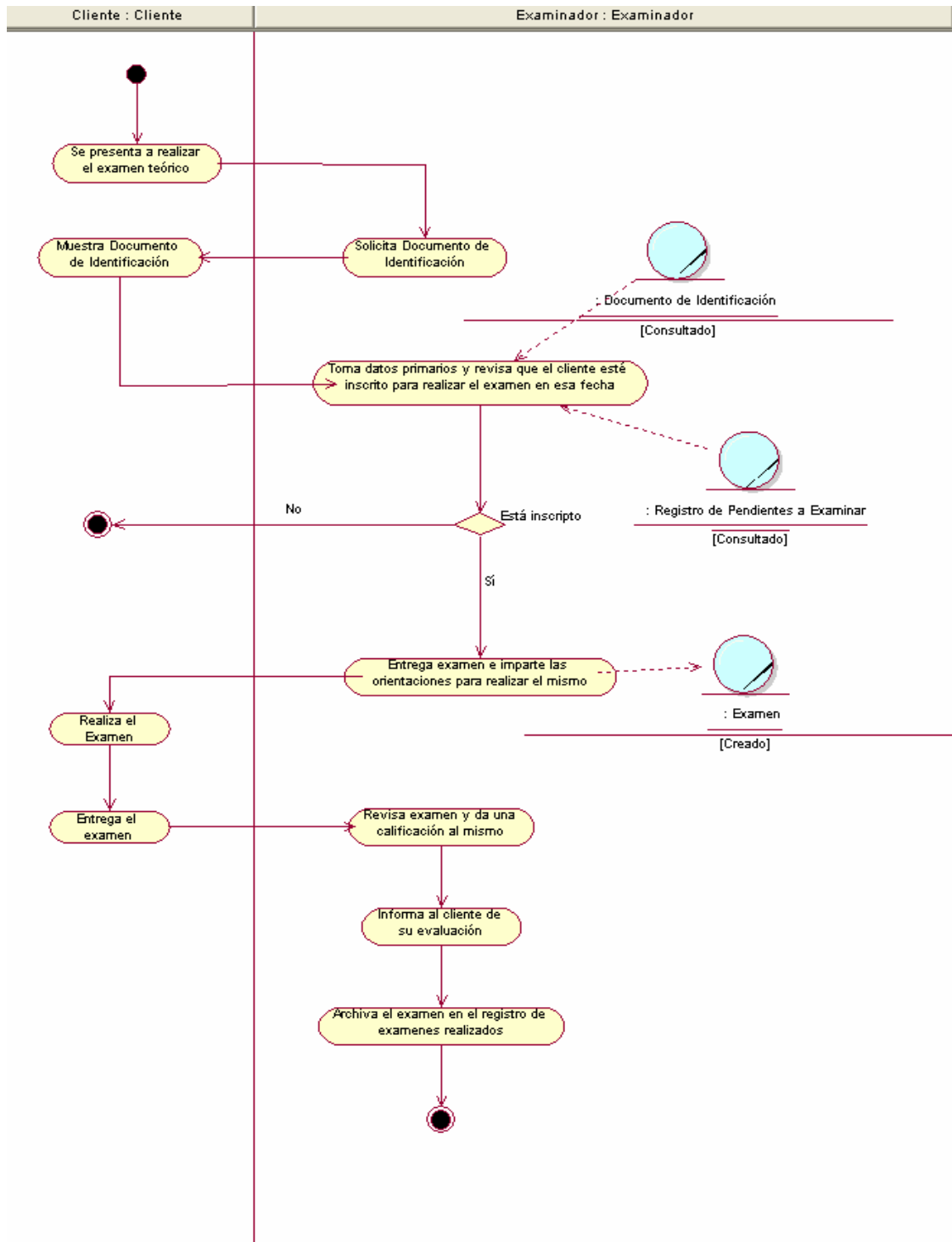
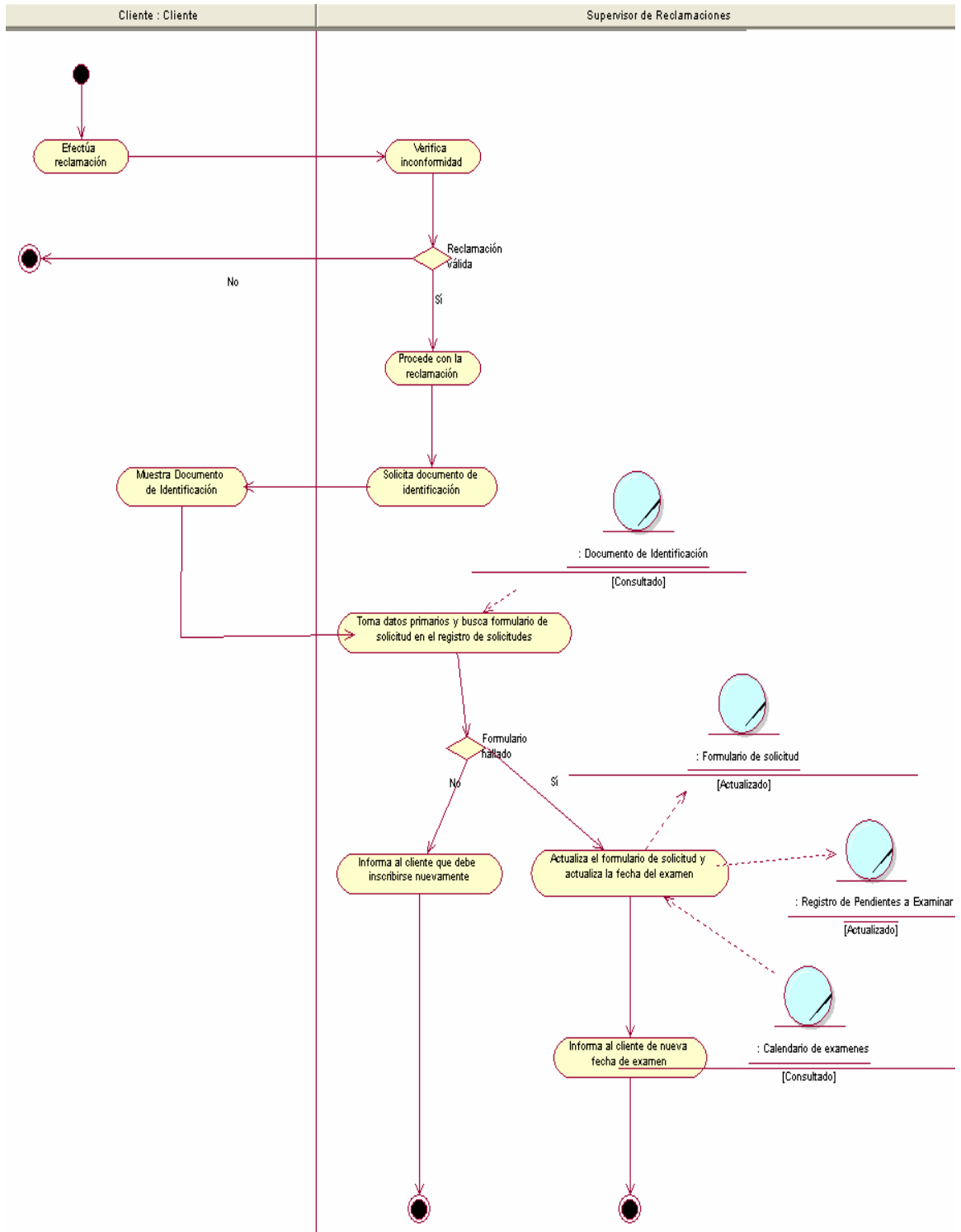


Ilustración 3. Diagrama de actividades del caso de uso Realizar Examen Teórico

### 2.7.2 Caso de uso Gestionar Reclamación de Inscripción

Tabla 4. Descripción del caso de uso Gestionar Reclamación de Inscripción.

<b>Nombre del Caso de Uso</b>	Gestionar Reclamación de Inscripción.
<b>Actores</b>	Cliente(inicia)
<b>Propósito</b>	Reclamar ante inconformidad en la inscripción
<b>Resumen</b>	El caso de uso se inicia cuando el cliente se presenta en el Área de Reclamación para efectuar una reclamación de inscripción, porque no aparece registrado para realizar el examen o por existir errores en los datos del cliente. Es atendido por el Supervisor de Reclamaciones, quien se encarga de tomar los datos de la reclamación, verificando si la inconformidad es válida, si es así, procede con la reclamación, buscando el formulario de solicitud correspondiente al cliente, si es encontrado el formulario, se revisan los datos de la inscripción y se actualizan en caso de ser necesario, se actualiza el registro de pendientes a evaluar y se le informa al cliente la nueva fecha en que puede presentarse a realizar el examen, y así concluye el caso de uso, en caso de no aparecer el formulario de solicitud se le informa al cliente que debe inscribirse nuevamente para realizar el examen, concluyendo de esta forma el caso de uso.



**Ilustración 4. Diagrama de actividades del caso de uso Gestionar Reclamación de Inscripción**

### 2.7.3 Caso de uso Gestionar Inscripción

Tabla 5. Descripción del caso de uso Gestionar Inscripción.

<b>Nombre del Caso de Uso</b>	Gestionar Inscripción.
<b>Actores</b>	Cliente(inicia)
<b>Propósito</b>	Inscribirse para realizar el examen teórico
<b>Resumen</b>	El caso de uso se inicia cuando el cliente se presenta en el Área de Inscripción para efectuar una inscripción, con el objetivo de realizar el examen teórico. Es atendido por el Registrador, quien le pide los resultados del examen médico, verificando si los mismos son válidos, si es así, solicita el documento de identificación del cliente, elabora el formulario de solicitud, inscribe al cliente en el registro de pendientes de examinar, archiva el formulario de solicitud en el registro de solicitudes, revisa en el calendario de exámenes y busca la fecha más cercana donde exista capacidad para realizar el examen, le informa al cliente la fecha en que debe presentarse a realizar el examen, concluyendo de esta forma el caso de uso.



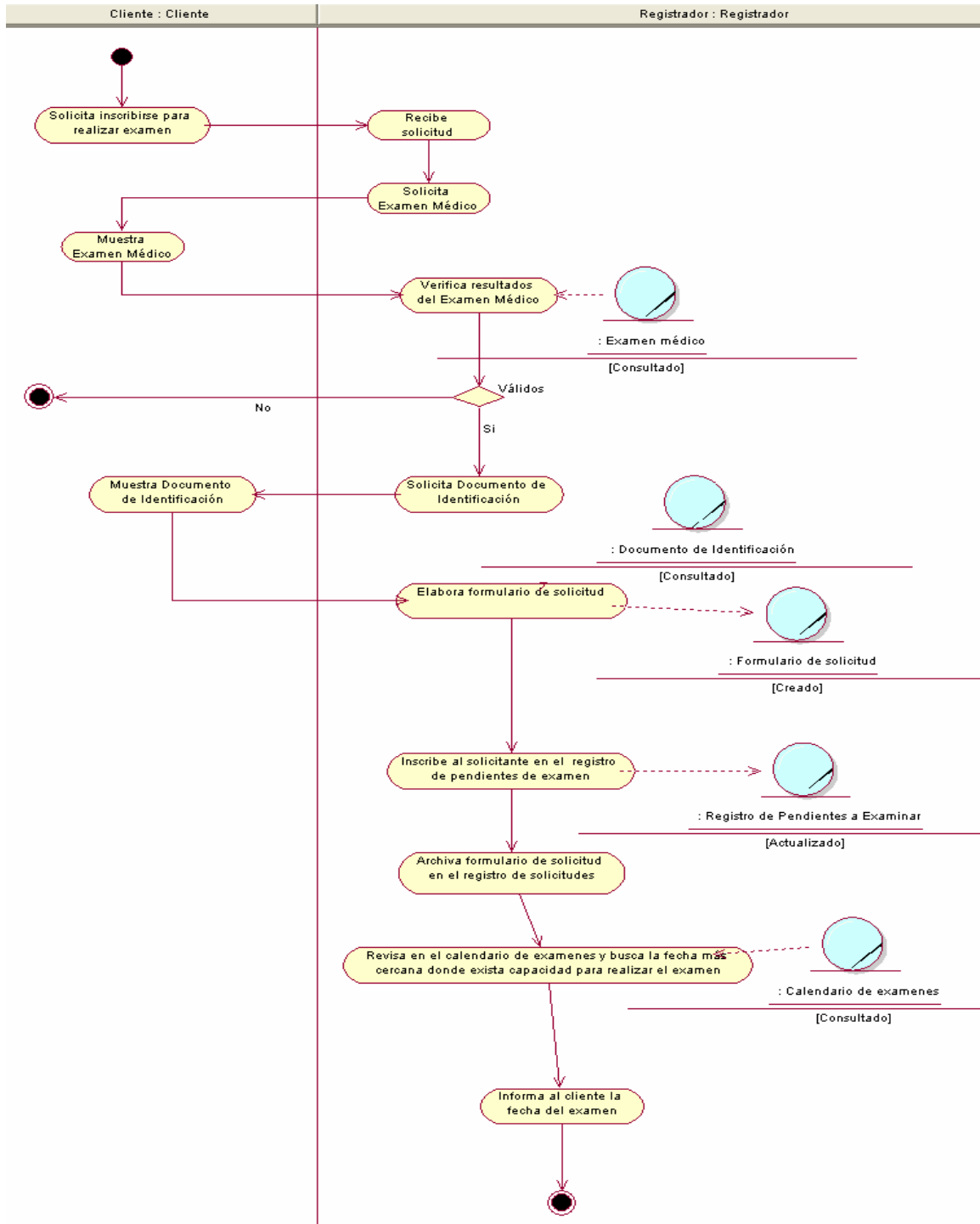


Ilustración 5. Diagrama de actividades del caso de uso Gestionar Inscripción

#### 2.7.4 Caso de uso Gestionar Reclamación de Examen

Tabla 6. Descripción del caso de uso Gestionar Reclamación de Examen.

<b>Nombre del Caso de Uso</b>	Gestionar Reclamación de Examen
<b>Actores</b>	Cliente(inicia)
<b>Propósito</b>	Reclamar ante inconformidad con la nota del examen.
<b>Resumen</b>	El caso de uso se inicia cuando el cliente se presenta en el Área de Reclamación para reportar una inconformidad ante la nota de su examen. Es atendido por el Supervisor de Reclamaciones, quien le pide el documento de identificación, busca en el registro de exámenes realizados el que corresponde al cliente, si es hallado el examen, lo revisa y le informa al cliente de los errores cometidos en el mismo y así concluye el caso de uso.

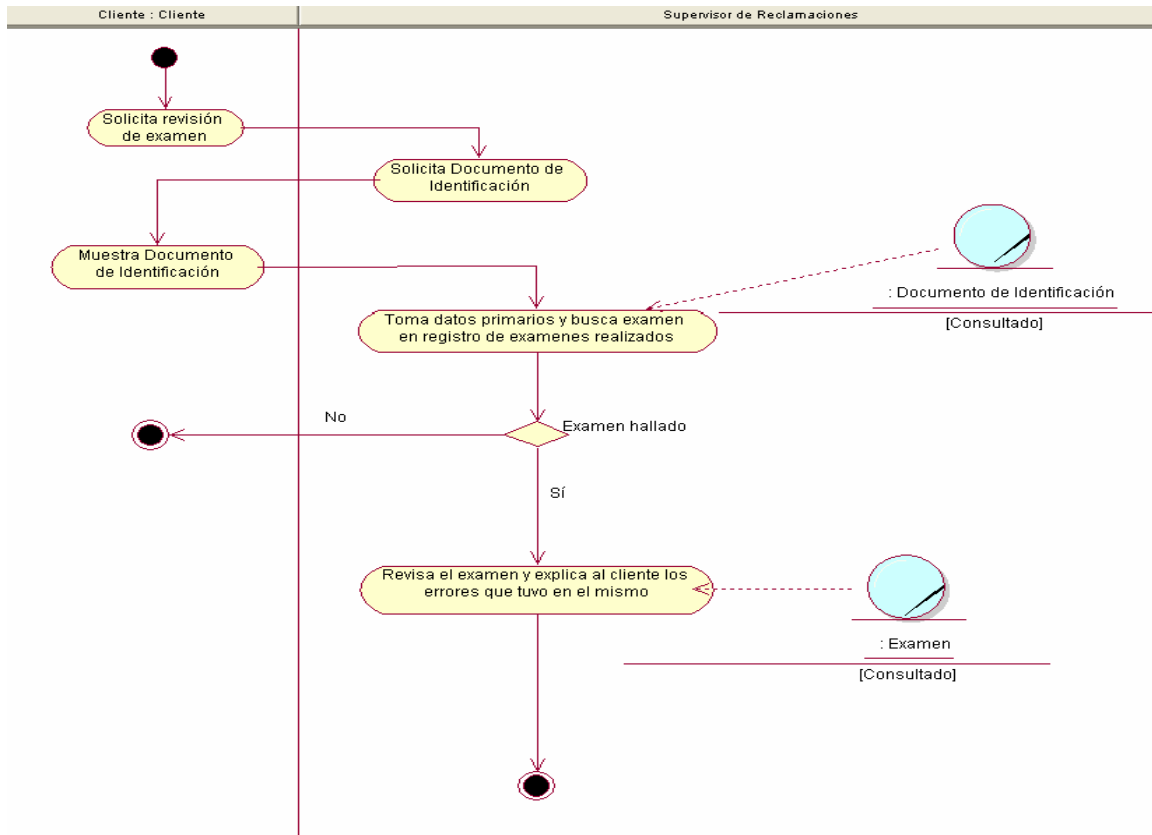
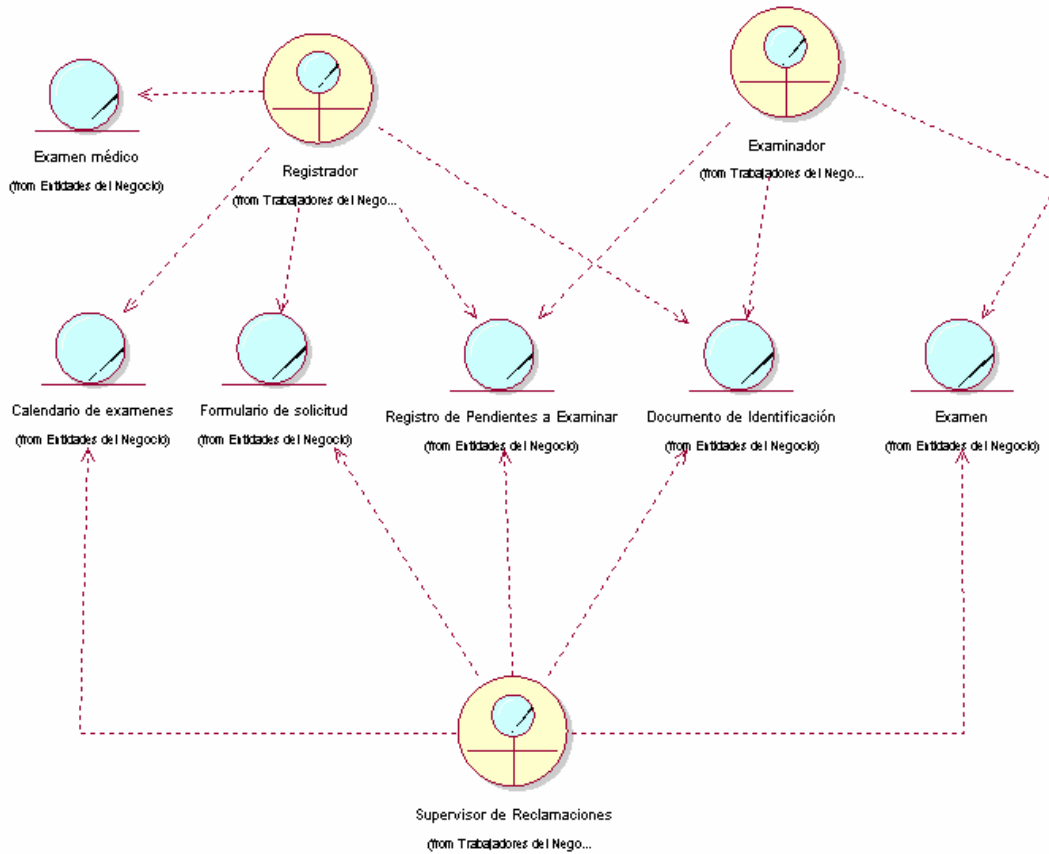


Ilustración 6. Diagrama de actividades del caso de uso Gestionar Reclamación de Examen

## 2.8 Modelo de objetos

La siguiente imagen muestra la estructura del modelo de objetos del negocio, donde se establecen las relaciones existentes entre los trabajadores del negocio y las entidades que son utilizadas en el mismo.



**Ilustración 7. Modelo de Objetos del Negocio**

## 2.9 Conclusiones

En este capítulo se abordó todo el tema del Modelado del Negocio, los procesos que se siguen para la evaluación teórica de licencia de conducción, que pueden ser generalizados a cualquier tipo de evaluación de conocimientos teóricos. Además se realizó el análisis de los casos de uso del negocio, mediante un diagrama de actividades y una descripción para cada uno de ellos.

## Capítulo 3 Requisitos

### 3.1 Introducción

En este capítulo se determinan los actores del sistema y se definen los casos de uso del sistema obtenido. Además se realiza una captura de los requisitos que debe llevar el sistema a elaborar, desglosándolos en requerimientos funcionales y no funcionales.

### 3.2 Actores del sistema a automatizar

A partir de los actores y trabajadores del negocio, definidos en el capítulo 2, se obtuvieron los actores del sistema. La siguiente tabla muestra la definición de los mismos.

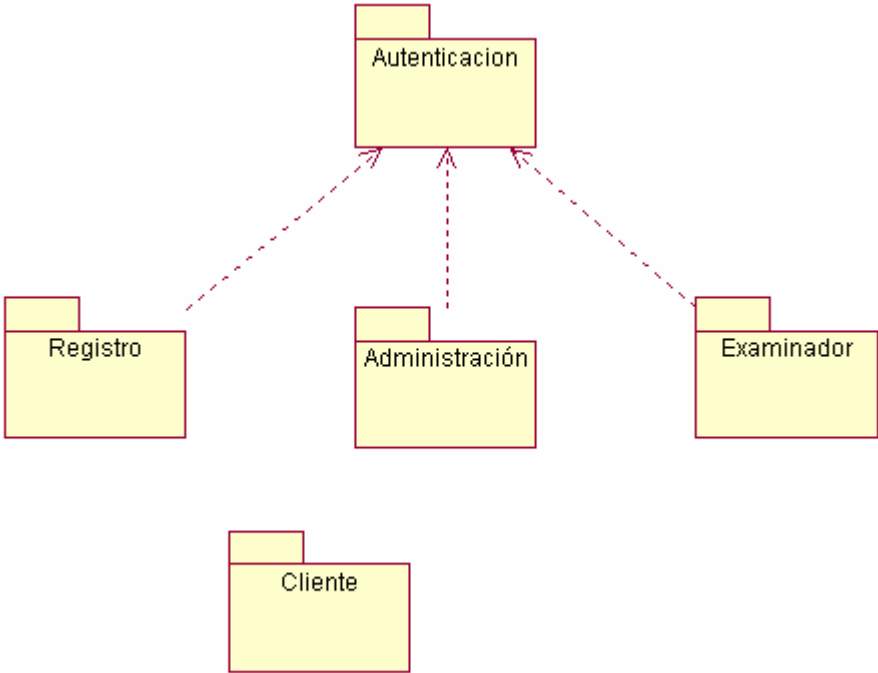
**Tabla 7. Definición de actores del sistema a automatizar**

Nombre del actor	Descripción
Cliente	Es la persona interesada en realizar el examen teórico, con el objetivo de acreditar sus conocimientos en una materia determinada, por ejemplo el código del tránsito.
Administrador	Es el encargado de configurar las diferentes opciones del examen. Emite los reportes. Administra las preguntas que se incluirán en los temarios. Administra los usuarios del sistema.
Registrador	Es el encargado de registrar los datos primarios de los clientes en el sistema. Estos datos serán nombre, apellidos, número de identificación y otros que pueden ser definidos según las características de la empresa que vaya a usar el sistema para sus procesos de evaluación teórica.
Examinador	Es el encargado de atender a cada cliente, orientarle en cual puesto de trabajo realizará su examen, chequear la

	realización del mismo, y poner a punto desde su puesto de trabajo cada máquina del cliente una vez concluido el examen.
--	-------------------------------------------------------------------------------------------------------------------------

**3.3 Paquetes y sus relaciones**

A continuación se muestran los paquetes en que se han organizado los casos de uso del sistema. Se creó un paquete por cada aplicación y un paquete para encapsular el caso de uso de Autenticación. La ilustración 8 muestra los paquetes y sus relaciones.



**Ilustración 8. Relaciones entre los paquetes de casos de uso del sistema**

### 3.4 Diagrama de casos de uso del sistema a automatizar

El modelado de casos de uso es la técnica más efectiva para modelar los requisitos del sistema. Los casos de uso se utilizan para modelar el funcionamiento o que desea el cliente que haga el sistema. Utilizando las facilidades que brinda el UML, se capturan los requisitos funcionales del sistema y se representan mediante un diagrama de casos de uso.

#### 3.4.1 Casos de uso a desarrollar en la primera iteración del sistema

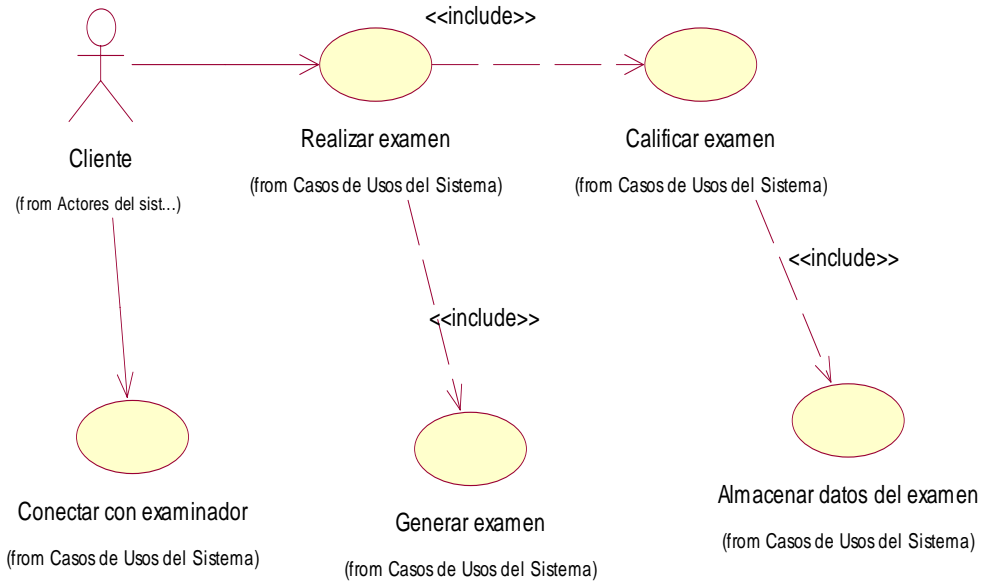
En la tabla siguiente se muestran los casos de uso que se desarrollarán en la primera iteración del sistema. Aunque el resto de los casos de uso se describen, serán estos que se listan a continuación los que incluirá la primera versión del sistema, porque incluyen las funcionalidades principales para realizar el examen teórico.

**Tabla 8. Casos de uso a desarrollar en la primera iteración**

Referencia	Caso de uso	Prioridad
<b>CUS 1</b>	Realizar examen	Crítico.
<b>CUS 2</b>	Generar examen	Crítico
<b>CUS 3</b>	Calificar examen	Crítico
<b>CUS 4</b>	Conectar con examinador	Crítico
<b>CUS 5</b>	Almacenar datos de examen	Crítico
<b>CUS 8</b>	Buscar	Crítico
<b>CUS 9</b>	Cancelar examen	Crítico
<b>CUS 10</b>	Asignar puesto de examen	Crítico
<b>CUS 11</b>	Chequear datos de puesto de trabajo	Crítico
<b>CUS 12</b>	Visualizar datos de examen	Crítico
<b>CUS 13</b>	Inicializar clientes	Crítico
<b>CUS 18</b>	Autenticar usuario	Crítico.

**3.4.2 Descripción y expansión de los casos de uso.**

**3.4.2.1 Paquete Cliente.**



**Ilustración 9. Diagrama de casos de uso del paquete Cliente**

**Tabla 9. Casos de uso del paquete Cliente**

Referencia	Caso de uso	Prioridad
<b>CUS 1</b>	Realizar examen	Crítico
<b>CUS 2</b>	Generar examen	Crítico
<b>CUS 3</b>	Calificar examen	Crítico
<b>CUS 4</b>	Conectar con examinador	Crítico
<b>CUS 5</b>	Almacenar datos de examen	Crítico

**Tabla 10. Descripción del caso de uso Realizar examen.**

<b>CUS 1</b>	Realizar examen
<b>Actores</b>	Cliente(inicia).



<b>Propósito</b>	Realizar el examen teórico.
<b>Resumen</b>	El caso de uso inicia cuando el cliente da click en la imagen para iniciar el examen. Se le van mostrando los diferentes videos que constituyen las preguntas del examen, las cuales son respondidas por el cliente, hasta finalizar el examen, momento en el que se le muestra la calificación obtenida en el mismo.
<b>CU asociados</b>	CUS 2, CUS 3,
<b>Referencias</b>	R8, R12, R16, R17, R18, R30, R31, R32
<b>Precondiciones</b>	La aplicación cliente debe encontrarse lista para ser usada
<b>Post condiciones</b>	Se culmina correctamente el examen teórico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El cliente da click en la imagen para iniciar el examen.	1.1 El sistema visualiza el video de presentación del examen 1.2 Muestra el video correspondiente a la pregunta a responder. 1.3 Muestra los datos del examen (tiempo transcurrido, cantidad total de preguntas, pregunta actual) 1.4 Envía al puesto del instructor el resumen de los datos del examen. 1.5 Visualiza las posibles respuestas a la pregunta planteada.

2. El cliente responde la pregunta.	2.1 El sistema ejecuta los pasos 1.2 al 1.5
3 El cliente culmina su examen.	3.1 El sistema invoca al caso de uso Calificar examen. 3.2 El sistema visualiza el video de culminación de examen. 3.3 El sistema muestra los resultados del examen al cliente. 3.4 El sistema envía al instructor los datos del examen completos.
<b>Flujo Alternativo de Eventos</b>	
Acción 2 2.2 El cliente decide visualizar nuevamente el video	El sistema reproduce nuevamente el video anterior.

**Tabla 11. Descripción del caso de uso Generar examen.**

<b>CUS 2</b>	Generar examen .<<include>>
<b>Actores</b>	CUS 1
<b>Propósito</b>	Generar las preguntas del examen
<b>Resumen</b>	El caso de uso se inicia cuando el cliente va a

	comenzar a realizar el examen. Se genera el mismo de forma aleatoria, según las características del tipo de examen a realizar.	
<b>CU asociados</b>		
<b>Referencias</b>	R6, R7, R8	
<b>Precondiciones</b>	La aplicación cliente debe encontrarse lista para ser usada	
<b>Post condiciones</b>	Se genera el temario a realizar.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>		
1. El cliente comienza el examen	1.1 El sistema busca una pregunta para el examen. 1.2 Chequea que no se repitan preguntas. 1.3 Chequea que la pregunta se correspondan con las adecuadas para el tipo de examen que se realiza. 1.4 Se adiciona la pregunta al temario 1.5 Cuando se complete la cantidad de preguntas configuradas para el tipo de examen se devuelve el temario completo.	

**Tabla 12. Descripción del caso de uso Calificar examen.**

<b>CUS 3</b>	Calificar examen .<<include>>	
<b>Actores</b>	CUS 1	
<b>Propósito</b>	Calificar el examen realizado, mostrar los resultados del mismo y enviar la notificación al instructor.	
<b>Resumen</b>	El caso de uso se inicia cuando el cliente concluye su examen.	
<b>CU asociados</b>	CUS 5	
<b>Referencias</b>	R13, R31	
<b>Precondiciones</b>	Debe haber concluido el examen.	
<b>Post condiciones</b>	Se muestran los resultados obtenidos en el examen.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El cliente responde la ultima pregunta del examen	1.1 El sistema realiza una calificación de las preguntas 1.2 Hace un cómputo de la nota alcanzada en el examen. 1.3 Se envía la notificación de que finalizó el examen al examinador. 1.4 Se muestran los resultados del examen al cliente.	

**Tabla 13. Descripción del caso de uso Conectar con examinador.**

<b>CUS 4</b>	Conectar con examinador
<b>Actores</b>	Cliente (inicia)
<b>Propósito</b>	Conectarse con el examinador y recibir los datos de las

	preguntas que debe incluirse en cada examen.
<b>Resumen</b>	El caso de uso se inicia cuando el cliente ejecuta la aplicación para realizar el examen
<b>CU asociados</b>	
<b>Referencias</b>	R40
<b>Precondiciones</b>	La aplicación del examinador debe encontrarse en ejecución.
<b>Post condiciones</b>	Se establece la conexión con el examinador y se reciben los datos de las preguntas y sus respuestas.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El cliente ejecuta la aplicación de examen.	<p>1.1 El sistema intenta conectarse con el examinador.</p> <p>1.2 Cuando recibe confirmación de conexión el sistema se prepara para recibir los datos de las preguntas.</p> <p>1.3 El sistema recibe los datos de las preguntas existentes en la Base de Datos.</p> <p>1.4 El sistema almacena los datos recibidos en las estructuras de almacenamiento de datos</p>

**Tabla 14. Descripción del caso de uso Almacenar datos de examen.**

<b>CUS 5</b>	Almacenar datos de examen .<<include>>
<b>Actores</b>	CUS 3

<b>Propósito</b>	Almacenar en la BD los datos de un examen
<b>Resumen</b>	El caso de uso se inicia cuando se culmina de calificar un examen.
<b>CU asociados</b>	
<b>Referencias</b>	R32
<b>Precondiciones</b>	Debe haberse concluido el examen y su calificación.
<b>Post condiciones</b>	Se almacenan en la BD los datos del examen realizado
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Culmina la calificación del examen.	1.1 El sistema almacena cada pregunta con la respuesta seleccionada por el cliente en la BD.

3.4.2.2 Paquete Registro

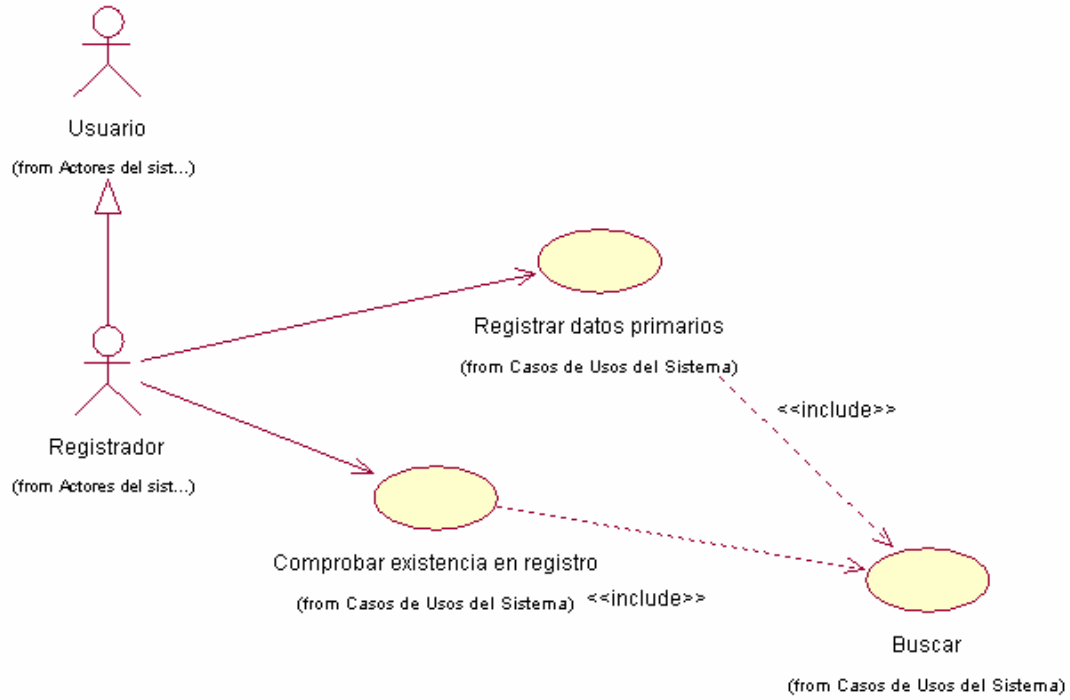


Ilustración 10. Diagrama de casos de uso del paquete Registro

Tabla 15. Casos de uso del paquete Registro.

Referencia	Caso de uso	Prioridad
<b>CUS 6</b>	Registrar datos primarios	Secundario.
<b>CUS 7</b>	Comprobar existencia en registro	Secundario
<b>CUS 8</b>	Buscar	Crítico

Tabla 16. Descripción del caso de uso Registrar datos primarios.

<b>CUS 6</b>	Registrar datos primarios
<b>Actores</b>	Registrador(inicia).
<b>Propósito</b>	Registrar los datos de un cliente para realizar el examen teórico.

<b>Resumen</b>	El caso de uso inicia cuando el Registrador inscribe los datos de un cliente para ser evaluado.
<b>CU asociados</b>	CUS 8
<b>Referencias</b>	R1, R3
<b>Precondiciones</b>	La aplicación de registro debe encontrarse lista para ser usada
<b>Post condiciones</b>	Se registran correctamente los datos del cliente

**Tabla 17. Descripción del caso de uso Comprobar existencia en registro.**

<b>CUS 7</b>	Comprobar existencia en registro
<b>Actores</b>	Registrador(inicia)
<b>Propósito</b>	Comprobar si un cliente determinado ya está registrado en la BD
<b>Resumen</b>	El caso de uso inicia cuando el Registrador solicita la búsqueda de un cliente en la BD.
<b>CU asociados</b>	CUS 8
<b>Referencias</b>	R2, R3, R4, R5
<b>Precondiciones</b>	La aplicación de registro debe encontrarse lista para ser usada
<b>Post condiciones</b>	Se realice una búsqueda del cliente mostrando los resultados de la misma.

**Tabla 18. Descripción del caso de uso Buscar.**

<b>CUS 8</b>	Buscar.<<include>>
<b>Actores</b>	CUS 6, CUS 7, CUS 12
<b>Propósito</b>	Realizar búsqueda de información en la BD
<b>Resumen</b>	El caso de uso inicia cuando se solicita la búsqueda de un



	cliente en la BD.	
<b>CU asociados</b>		
<b>Referencias</b>	R3	
<b>Precondiciones</b>		
<b>Post condiciones</b>	Se realiza una búsqueda devolviendo los resultados de la misma.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. Se selecciona realizar una búsqueda.	1.1 El sistema realiza la búsqueda según los parámetros especificados 1.2 Devuelve los resultados obtenidos en la búsqueda.	

3.4.2.3 Paquete Examinador

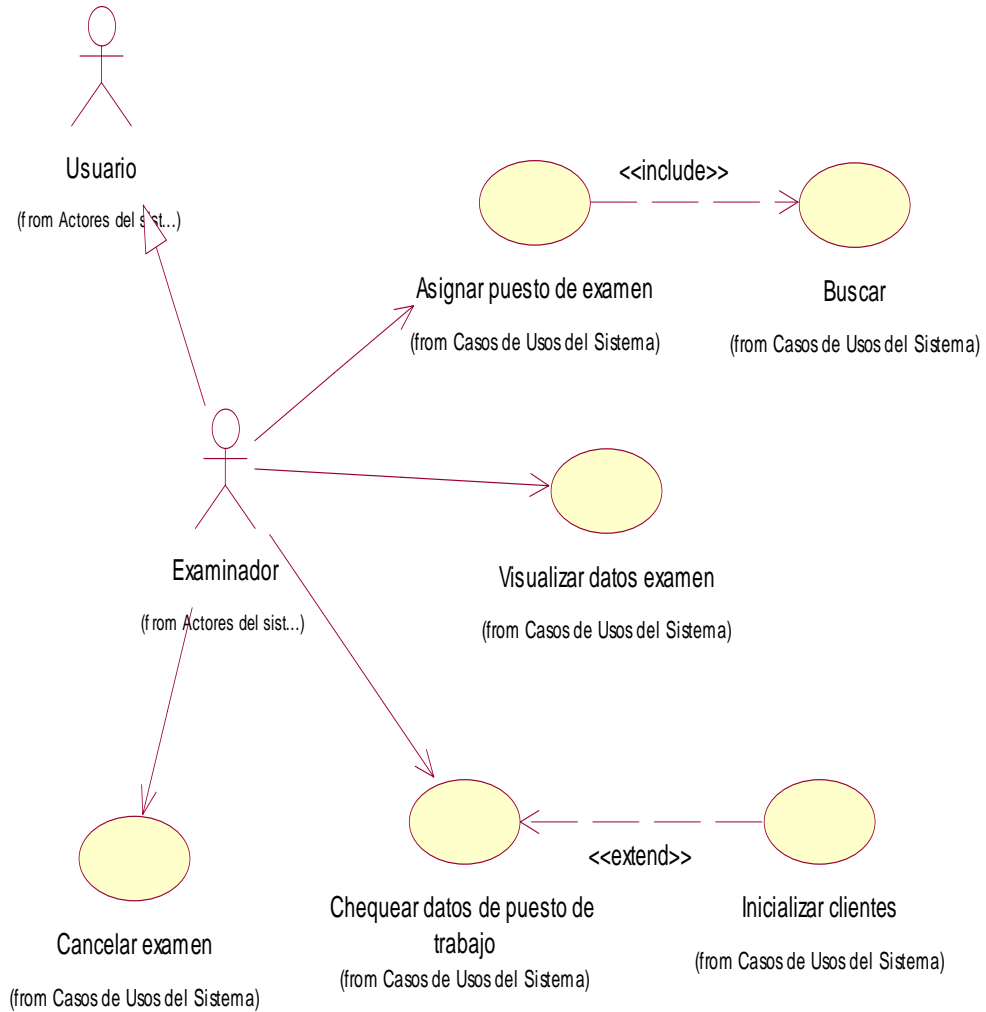


Ilustración 11. Diagrama de casos de uso del paquete Examinador

Tabla 19. Casos de uso del paquete Examinador.

Referencia	Caso de uso	Prioridad
<b>CUS 9</b>	Cancelar examen	Crítico
<b>CUS 10</b>	Asignar puesto de examen	Crítico

<b>CUS 11</b>	Chequear datos de puesto de trabajo	Crítico
<b>CUS 12</b>	Visualizar datos de examen	Crítico
<b>CUS 13</b>	Inicializar clientes<<extend>>	Crítico

**Tabla 20. Descripción del caso de uso Cancelar examen.**

<b>CUS 9</b>	Cancelar examen	
<b>Actores</b>	Examinador(inicia)	
<b>Propósito</b>	Cancelar el proceso de realización de un examen	
<b>Resumen</b>	El caso de uso se inicia cuando se selecciona la opción cancelar examen, se procede a calificar el mismo y se poner el puesto de trabajo a punto para ser usado nuevamente por otro cliente.	
<b>CU asociados</b>		
<b>Referencias</b>	R27	
<b>Precondiciones</b>	En el puesto de trabajo a cancelar, debe estarse ejecutando un examen.	
<b>Post condiciones</b>	Se cancela el examen que se está realizando y se pone a punto el puesto de trabajo.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. Selecciona cancelar examen, haciendo click en un puesto de trabajo que esté activo.	1.1 Muestra una ventana de confirmación.	

2. Acepta cancelar el examen.	<p>2.1 El sistema envía al puesto de trabajo seleccionado el comando de cancelar el examen.</p> <p>2.2 Muestra el reporte de finalización de examen.</p> <p>2.3 Envía el comando de disponible al puesto de trabajo.</p>

**Tabla 21. Descripción del caso de uso Asignar puesto de trabajo.**

<b>CUS 10</b>	Asignar puesto de trabajo
<b>Actores</b>	Examinador (inicia)
<b>Propósito</b>	Asignar un puesto de trabajo a un cliente para ejecutar el examen en él
<b>Resumen</b>	El caso de uso se inicia cuando se presenta un cliente a realizar el examen, el examinador determina de los puestos de trabajo disponibles en cual de ellos se va a realizar, y asigna los datos del cliente a dicho puesto de trabajo.
<b>CU asociados</b>	CUS8,
<b>Referencias</b>	R3, R5, R36, R38

<b>Precondiciones</b>	Debe existir al menos un puesto de trabajo disponible para realizar el examen.
<b>Post condiciones</b>	Se pone a punto el puesto de trabajo para que se pueda ejecutar el examen, asignándole los datos del cliente que ejecutará el examen allí.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Selecciona asignar cliente, haciendo click en un puesto de trabajo que esté disponible.	1.1 El sistema muestra la interfaz para añadir los datos del cliente.
2. El examinador entra los datos del cliente y presiona el botón Buscar.	2.1 Busca los clientes que coincidan con los datos entrados. 2.2 El sistema muestra los resultados de la búsqueda.
3. El examinador selecciona los datos del cliente, de los resultados mostrados en la búsqueda y presiona el botón Asignar.	3.1 El sistema asigna los datos del cliente al puesto de trabajo. 3.2 El sistema actualiza el estado del puesto de trabajo poniéndolo en estado de espera 3.3 El sistema envía los datos del examen al puesto de trabajo

<b>Flujo Alternativo de Eventos</b>	
<p>Acción1: No existe puesto de trabajo disponible, se le informa al cliente que debe dirigirse a otro puesto de examinador, o esperar porque se libere algún puesto de trabajo.</p>	

**Tabla 22. Descripción del caso de uso Chequear datos de puesto de trabajo.**

<b>CUS 11</b>	Chequear datos de puesto de trabajo
<b>Actores</b>	Examinador(inicia)
<b>Propósito</b>	Visualizar los datos de los diferentes puestos de trabajo que controla el examinador.
<b>Resumen</b>	El caso de uso se inicia cuando se inicializa la aplicación del examinador, y continúa de forma cíclica, hasta que esta es cerrada. En este intervalo se pueden recibir intentos de conexión de los puestos de trabajo.
<b>CU asociados</b>	CUS 13
<b>Referencias</b>	R19, R20, R21, R22, R24
<b>Precondiciones</b>	El examinador debe tener puestos de trabajo asignados para chequear.
<b>Post condiciones</b>	Se visualizan los datos de los puestos de trabajo asignados al examinador.
<b>Flujo Normal de Eventos</b>	

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Abre la aplicación y se autentica	1.1 Muestra los diferentes puestos de trabajo asignados al examinador y sus estados.

**Tabla 23. Descripción del caso de uso Visualizar datos del examen.**

<b>CUS 12</b>	Visualizar datos del examen
<b>Actores</b>	Examinador(inicia)
<b>Propósito</b>	Visualizar los resultados del cliente en un examen.
<b>Resumen</b>	El caso de uso se inicia cuando se notifica que en un puesto de trabajo se ha concluido el examen y el Examinador hace click para mostrar los resultados.
<b>CU asociados</b>	
<b>Referencias</b>	R13, R14
<b>Precondiciones</b>	Debe haber concluido el examen en un puesto de trabajo.
<b>Post condiciones</b>	Se muestran los resultados del examen
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>

1. Selecciona visualizar resultados, haciendo click en un puesto de trabajo que esté finalizado.	1.1 El sistema muestra el reporte de finalización de examen. 1.2 Se envía el comando de disponible al puesto de trabajo.
--------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------

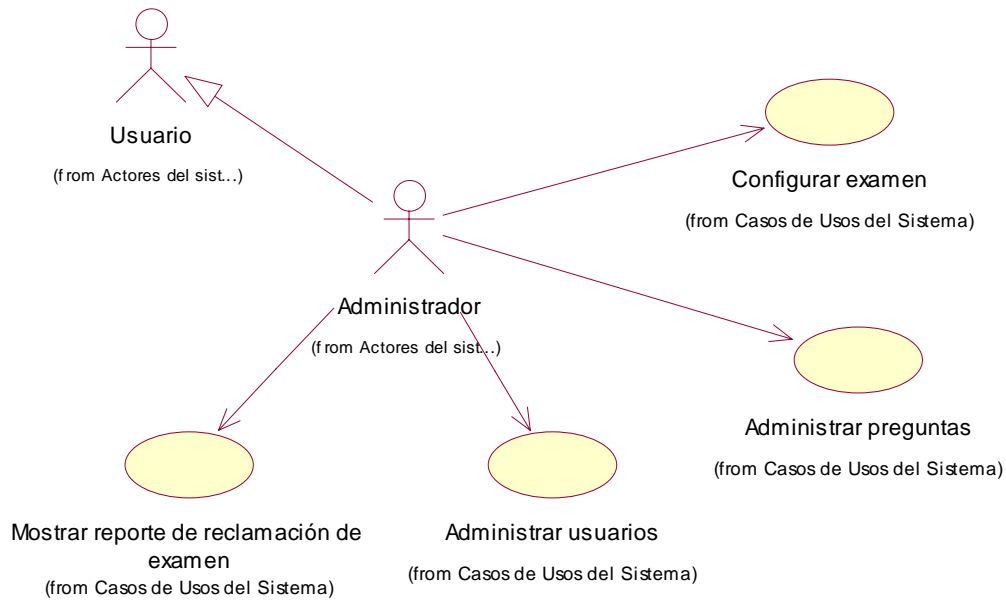
**Tabla 24. Descripción del caso de uso Inicializar clientes.**

<b>CUS 13</b>	Inicializar clientes <<extend>>
<b>Actores</b>	CUS 10 (inicia)
<b>Propósito</b>	Recibir las conexiones de los clientes y almacenarlos en la lista de clientes conectados.
<b>Resumen</b>	El caso de uso se inicia se recibe la petición de conexión de un cliente. El sistema debe enviar al cliente la confirmación de que se aceptó la conexión y almacena el identificador del puesto de trabajo en su lista de conexiones
<b>CU asociados</b>	
<b>Referencias</b>	R41
<b>Precondiciones</b>	
<b>Post condiciones</b>	Se le envía la confirmación de conexión al puesto del cliente y se añade a la lista de clientes conectados.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>



<p>1. Chequea el estado de un puesto de trabajo y recibe una petición de conexión</p>	<p>1.1 El sistema envía la confirmación de conexión aceptada en caso de tener capacidad para atender este puesto de trabajo.</p>
---------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------

**3.4.2.4 Paquete Administración**



**Ilustración 12. Diagrama de casos de uso del paquete Administración.**

**Tabla 25. Casos de uso del paquete Administración**

Referencia	Caso de uso	Prioridad
<b>CUS 14</b>	Mostrar reporte de reclamación de examen	Secundario
<b>CUS 15</b>	Administrar usuarios	Secundario

<b>CUS 16</b>	Administrar preguntas	Secundario
<b>CUS 17</b>	Configurar examen	Secundario

**Tabla 26. Descripción del caso de uso Mostrar reporte de reclamación de examen**

<b>CUS 14</b>	Mostrar reporte de reclamación de examen	
<b>Actores</b>	Administrador	
<b>Propósito</b>	Muestra el reporte de reclamación de un examen	
<b>Resumen</b>	El caso de uso se inicia cuando se desea visualizar el reporte de reclamación de un examen, porque el cliente no esté de acuerdo con la nota obtenida en el mismo.	
<b>CU asociados</b>		
<b>Referencias</b>	R42	
<b>Precondiciones</b>	Debe existir un examen a mostrar	
<b>Post condiciones</b>	Se muestran los resultados del reporte seleccionado.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. Selecciona visualizar reporte	1.1 El sistema muestra la interfaz para entrar los datos del reporte.	

2. Entra el número del examen realizado y presiona el botón Mostrar	2.1 Busca los datos del reporte. 2.2 Muestra el reporte con las preguntas realizadas en el examen y la calificación obtenida en el mismo
---------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------

**Tabla 27. Descripción del caso de uso Administrar usuario**

<b>CUS 15</b>	Administrar usuarios
<b>Actores</b>	Administrador(inicia)
<b>Propósito</b>	Gestionar los usuarios que trabajarán en el sistema
<b>Resumen</b>	El caso de uso se inicia cuando el Administrador necesita añadir, eliminar o modificar los usuarios del sistema
<b>CU asociados</b>	
<b>Referencias</b>	R9, R10
<b>Precondiciones</b>	El Administrador tiene que estar autenticado satisfactoriamente por el sistema.
<b>Post condiciones</b>	Se realiza la operación seleccionada y el sistema queda en condiciones de usarse nuevamente.

**Tabla 28. Descripción del caso de uso Administrar preguntas**

<b>CUS 16</b>	Administrar preguntas
<b>Actores</b>	Administrador (inicia)
<b>Propósito</b>	Gestionar las preguntas que se incluirán en los exámenes a aplicar

<b>Resumen</b>	El caso de uso se inicia cuando el Administrador necesita añadir, eliminar o modificar las preguntas que conforman los exámenes.
<b>CU asociados</b>	
<b>Referencias</b>	R8, R37, R41
<b>Precondiciones</b>	El Administrador tiene que estar autenticado satisfactoriamente por el sistema.
<b>Post condiciones</b>	Se realiza la operación seleccionada y el sistema queda en condiciones de usarse nuevamente.

**Tabla 29. Descripción del caso de uso Configurar examen**

<b>CUS 17</b>	Configurar examen
<b>Actores</b>	Administrador (inicia)
<b>Propósito</b>	Configurar los datos del examen tales como tiempo de ejecución, cantidad de preguntas, valor mínimo de aprobado, etc.
<b>Resumen</b>	El caso de uso se inicia cuando se quiere añadir una nueva configuración de realización de examen o modificar una existente
<b>CU asociados</b>	
<b>Referencias</b>	R25
<b>Precondiciones</b>	El Administrador debe estar autenticado en el sistema.
<b>Post condiciones</b>	Se configuran los datos de realización de un examen tipo.

### 3.4.2.5 Paquete Autenticación.



Ilustración 13. Diagrama de casos de uso del paquete Autenticación.

Tabla 30. Casos de uso del paquete Autenticación

Referencia	Caso de uso	Prioridad
<b>CUS 18</b>	Autenticar usuario	Crítico.

Tabla 31. Descripción del caso de uso Autenticar usuario

<b>CUS 18</b>	Autenticar usuario
<b>Actores</b>	Usuario(inicia)
<b>Propósito</b>	Permitir acceso del usuario a las aplicaciones del sistema
<b>Resumen</b>	El caso de uso inicia cuando el usuario entra el nombre de usuario y la contraseña para acceder al sistema, validándose los datos entrados y permitiendo o no la entrada al sistema.
<b>CU asociados</b>	
<b>Referencias</b>	R11
<b>Precondiciones</b>	Existencia de un usuario y contraseña válidos en la BD.
<b>Post condiciones</b>	Se garantiza la entrada del usuario al sistema. Se activa el

	módulo deseado.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Introduce nombre de usuario y contraseña, presiona el botón Entrar	1.1 El sistema encripta la contraseña 1.2 El sistema comprueba que los datos entrados son correctos. 1.3 El sistema permite al usuario acceder a las opciones de la aplicación.
2. El usuario inicia su sesión. Con el acceso concedido se abre la ventana del puesto de trabajo correspondiente.	
<b>Flujo Alternativo de Eventos</b>	
Acción 2	En caso que la autenticación se realice de forma incorrecta, el sistema informa de un error, pasando a la Acción 1.1

### **3.5 Definición de los requisitos funcionales y no funcionales**

#### **3.5.1 Requisitos funcionales**

Tabla 32. Requisitos funcionales del sistema.

Referencia	Función
------------	---------

R1	Introducir los datos primarios para inscribir al aspirante a evaluarse
R2	Comprobar existencia en el registro del cliente a evaluar
R3	Buscar los datos de un cliente
R4	Mostrar los datos de un cliente en el registrador
R5	Mostrar mensaje en caso de que no aparezca ningún cliente en la búsqueda
R6	Buscar preguntas de forma aleatoria
R7	Chequear que las preguntas no se repitan
R8	Generar el examen
R9	Administrar usuarios en el sistema
R10	Asignar los privilegios a los usuarios
R11	Permitir autenticación de los usuarios en las diferentes aplicaciones.
R12	Mostrar reporte de evaluación al cliente al concluir el examen.
R13	Notificar al examinador de la culminación de un examen.
R14	Mostrar reporte de evaluación al examinador
R15	Permitir el envío de los datos de cada examen entre las aplicaciones del Examinador y el Cliente por la red.
R16	Visualizar el tiempo de ejecución del examen en el cliente
R17	Mostrar numero de pregunta actual en el cliente
R18	Mostrar cantidad total de preguntas a realizar en el cliente
R19	Mostrar tiempo de ejecución en el examinador
R20	Mostrar numero de pregunta actual en el examinador
R21	Mostrar cantidad total de preguntas a realizar en el examinador

R22	Mostrar en el examinador una imagen correspondiente al video que se visualiza en el cliente.
R23	Asignar los puestos de evaluación que debe atender cada examinador.
R24	Mostrar los puestos de trabajo que atiende cada examinador según las asignaciones realizadas.
R25	Configurar los tipos de exámenes.
R26	Mostrar de forma gráfica los puestos disponibles para realizar el examen y el estado de los mismos.
R27	Cancelar la realización de un examen.
R28	Visualizar los videos que representan las preguntas escogidas en cada examen.
R29	Visualizar las posibles respuestas de las preguntas.
R30	Responder cada pregunta.
R31	Emitir resultado del examen.
R32	Registrar los datos de cada examen en la BD.
R33	Definir el tipo de almacenamiento de las preguntas y de los datos.
R34	Administrar las preguntas a evaluar en el examen.
R35	Reproducir nuevamente un video determinado del examen.
R36	Asignar los clientes a los puestos de trabajo disponibles
R37	Visualizar reporte de reclamación.
R38	Permitir el envío de comandos para cancelar examen, o alistar puesto de trabajo desde el examinador a un puesto de trabajo.
R39	Mostrar mensaje de error en caso de que el cliente a registrar exista en la BD



R40	Permitir a los clientes conectarse al examinador.
R41	Permitir al examinador aceptar las peticiones de conexión de los clientes.
R42	Mostrar reporte de reclamación de examen.

### 3.5.2 *Requisitos no funcionales*

#### **Resolución de pantalla**

La resolución de pantalla será de 800 x 600 píxeles. La profundidad de color será de 24 bits.

#### **Interfaz gráfica de usuario**

La interfaz gráfica de la aplicación de realización del examen, estará compuesta por un área donde se reproduzcan los videos, otra con las posibles respuestas a cada pregunta, otra área de información con los datos del examen (cantidad de preguntas, número actual de las mismas, tiempo transcurrido del examen). En el caso de la aplicación del instructor la misma debe contar con nueve áreas, una para puesto de trabajo, en las mismas se visualizará toda la información referente a los puestos de trabajo que debe atender el instructor.

#### **Implementación**

La base de datos del sistema, será implementada en 3 sistemas gestores de bases de datos diferentes y además debe usarse XML para el caso de que la aplicación de realización de examen corra como módulo de entrenamiento.

El sistema se programará en C++ standard, usando los tipos y librerías básicos, para facilitar la migración a sistemas operativos de software libre.

La comunicación de datos entre los puestos de evaluación y el instructor, se realizarán mediante conexión por sockets, usando protocolo TCP/IP.

#### **Sistema Operativo**

La primera versión del sistema será para ser usada en Windows.

**Seguridad**

La seguridad del sistema la brindará el sistema gestor de bases de datos que se use, y en el caso de almacenarse la información en ficheros XML, la seguridad se implementará mediante encriptación de los datos.

**3.6 Conclusiones**

En el capítulo que finaliza se presentaron los casos de uso y los actores que se definieron para el sistema, agrupándolos en 5 paquetes para su mejor comprensión (Cliente, Registro, Administración, Examinador, Autenticación), con un total de 15 casos de uso. Además se presentaron los requisitos funcionales y no funcionales del sistema

Se realizó la descripción de cada caso de uso en formato expandido, y gracias a la culminación de este flujo de trabajo se puede comenzar la construcción del sistema, tratando de que se cumplan los requisitos y funcionalidades planificadas en este capítulo.

## Capítulo 4 Construcción de la solución propuesta

### 4.1 Introducción

El presente capítulo muestra la construcción de la solución propuesta, a través de los flujos de trabajo de diseño e implementación. Previamente se presenta el modelo de diseño, donde son expuestas las realizaciones de los casos de uso en el capítulo anterior, mediante diagramas de clases del diseño y diagramas de interacción. Además se presentan los diagramas de clases persistentes, así como el modelo de datos obtenido a partir de estos. Se muestra también el modelo de implementación con los diagramas de componentes definidos.

Se usa una arquitectura de 3 capas, divididas en:

- Capa de aplicación
- Capa de negocio
- Capa de acceso a datos

Se aplican patrones de diseño como el Row Data Gateway, en la capa de acceso a datos, el cual consiste en un objeto que actúa como puerta de acceso a un record en la fuente de datos (base de datos), existiendo una instancia del mismo por cada fila que exista en la tabla de la base de datos.

### 4.2 Diagrama de clases del diseño

#### 4.2.1 Arquitectura de 3 capas

La figura siguiente muestra la estructura en 3 capas del sistema.

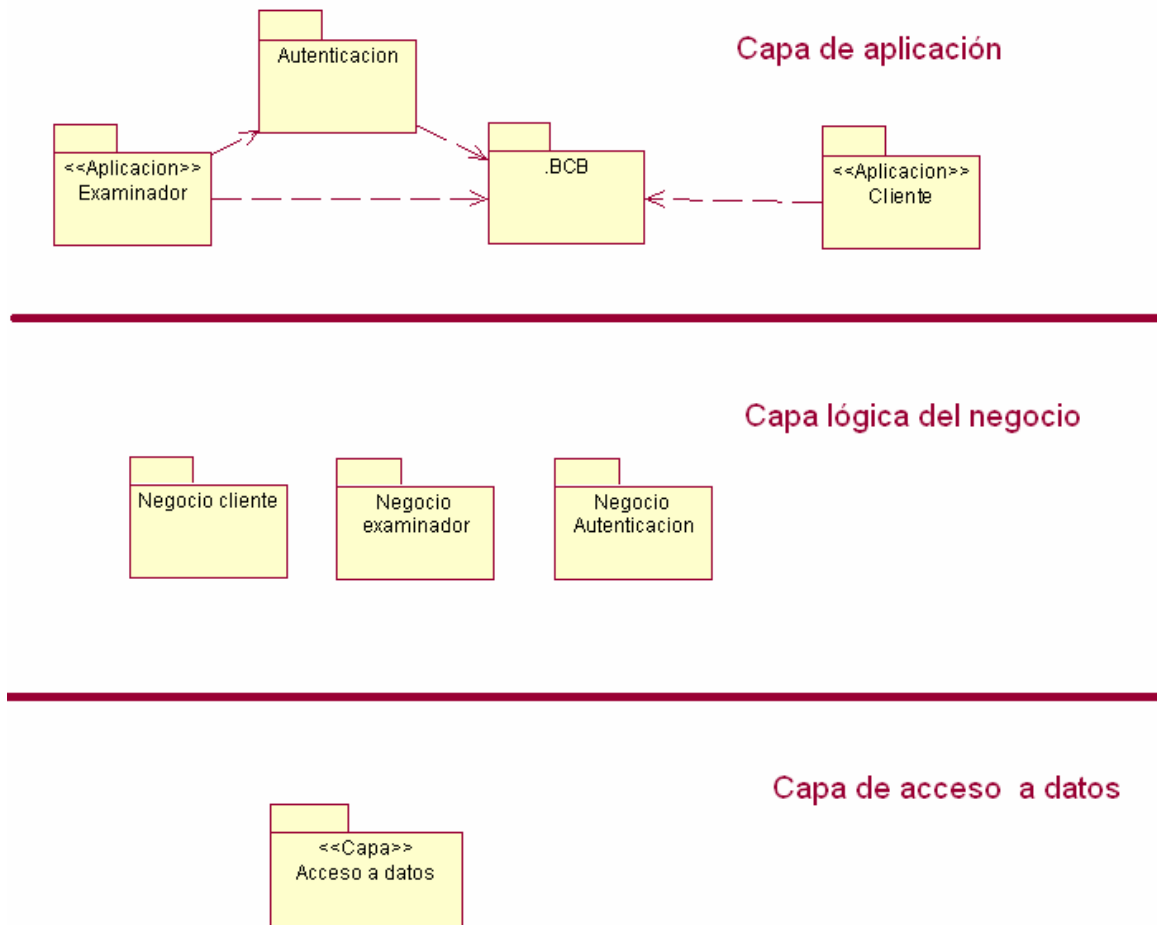


Ilustración 14. Arquitectura de 3 capas del sistema.

## 4.2.2 Capa Aplicación

### 4.2.2.1 Paquete .BCB

La siguiente imagen muestra el diagrama de clases del paquete .BCB en la Capa de Aplicación. Donde se relacionan todos los elementos de la capa de presentación o aplicación.

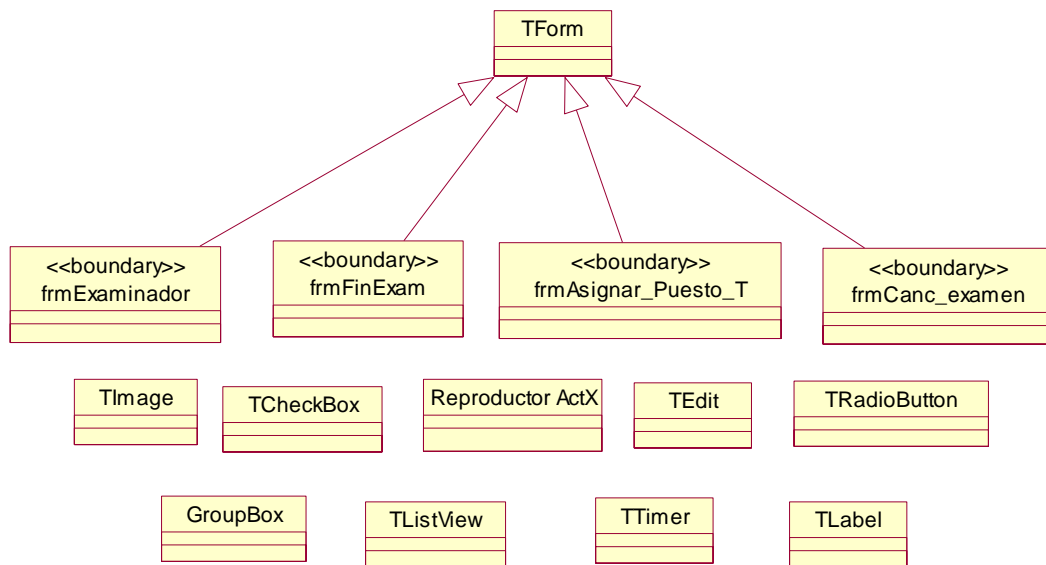


Ilustración 15. Diagrama de clases del paquete .BCB en la Capa de Aplicación

#### 4.2.2.2 Paquete Autenticación

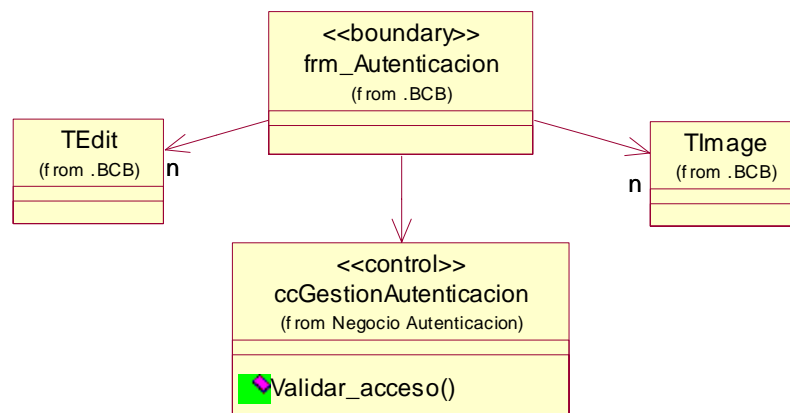


Ilustración 16. Diagrama de clases del paquete Autenticación en la Capa de Aplicación

**Descripción de las clases que conforman el diagrama:**

A continuación se describen las clases que conforman el diagrama del paquete de autenticación, solamente se describen las clases que no son del paquete .BCB.

**Tabla 33. Descripción de la clase ccGestionAutenticación**

<b>Nombre: ccGestionAutenticación</b>	
<b>Tipo de clase: controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre:	Validar acceso
Descripción:	Esta responsabilidad se encarga de validar si los datos pasados como parametro son correctos para garantizar la autenticación del usuario en el sistema.

4.2.2.3 Paquete Cliente

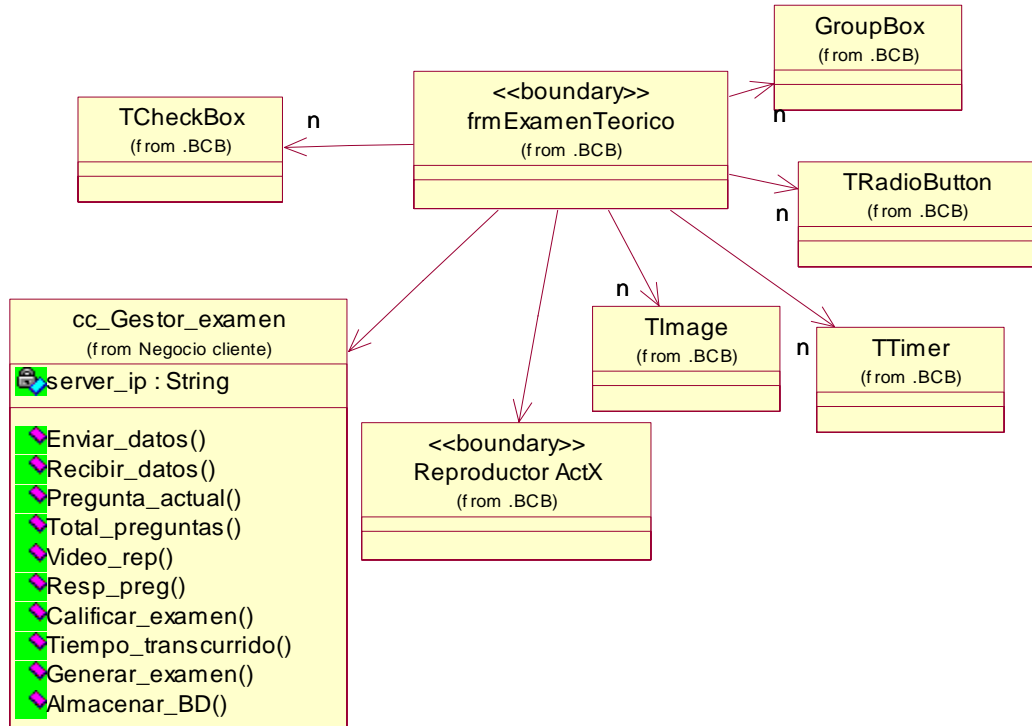


Ilustración 17. Diagrama de clases del paquete Cliente en la Capa de Aplicación

Descripción de las clases que conforman el diagrama:

A continuación se describen las clases que conforman el diagrama del paquete del cliente, solamente se describen las clases que no son del paquete .BCB.

Tabla 34. Descripción de la clase cc\_Gestor\_examen

<b>Nombre:</b> cc_Gestor_examen	
<b>Tipo de clase:</b> controladora	
<b>Atributo</b>	<b>Tipo</b>
server_ip	String
<b>Para cada responsabilidad:</b>	
Nombre:	Descripción:

Enviar_datos	Esta responsabilidad se encarga de enviar los datos de un examen al puesto del examinador, por la red, para que sean mostrados allá.
Recibir_datos	Se usa para recibir los comandos que el examinador envía por la red.
Pregunta_actual	Se usa para devolver la pregunta actual del examen que se está realizando
Total_preguntas	Se usa para devolver la cantidad de preguntas que conforman un examen.
Video_rep	Esta funcionalidad devuelve el video que le corresponde reproducir al examen.
Resp_preg	Permite responder una pregunta determinada.
Calificar_examen	Este metodo realiza una calificación del examen que se está realizando, emitiendo un resultado del mismo
Tiempo_transcurrido	Devuelve el tiempo que ha transcurrido en un examen.
Generar_examen	Genera el examen que se va a realizar.
Almacenar_BD	Almacena los datos del examen en la base de datos.

#### 4.2.2.4 Paquete Examinador

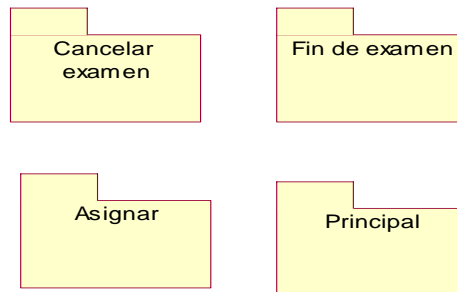
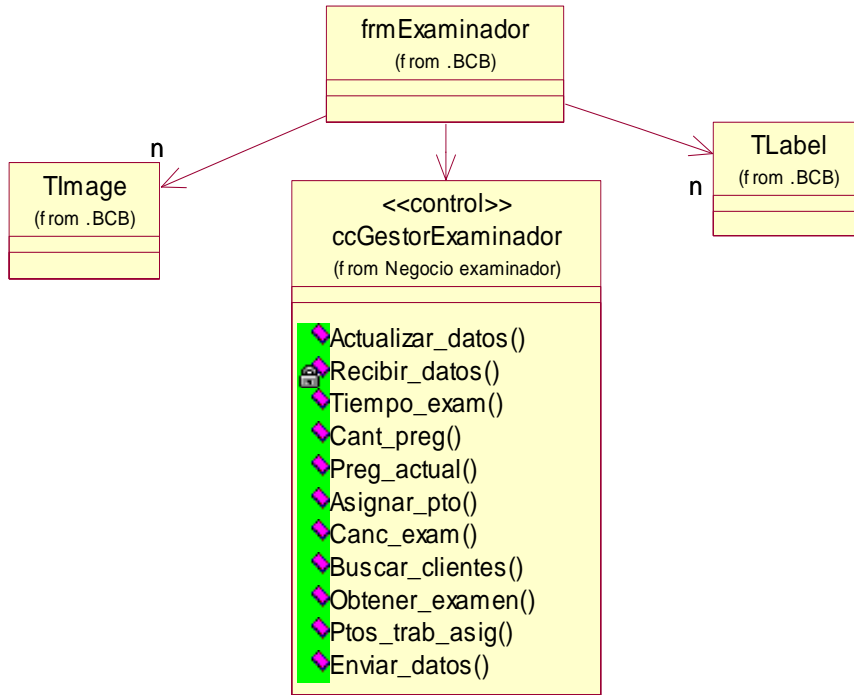


Ilustración 18. Diagrama de clases del paquete Examinador en la Capa de Aplicación

##### 4.2.2.4.1 Vista Principal





**Ilustración 19. Diagrama de clases de la Vista Principal del paquete Examinador en la Capa de Aplicación**

**Descripción de las clases que conforman el diagrama:**

A continuación se describen las clases que conforman el diagrama del paquete examinador, en la vista principal, solamente se describen las clases que no son del paquete .BCB.

**Tabla 35. Descripción de la clase ccGestorExaminador**

<b>Nombre: ccGestorExaminador</b>	
<b>Tipo de clase: controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre:	Descripción:
Actualizar_datos	Esta responsabilidad se encarga de actualizar todos los datos de los exámenes que se

	están monitoreando en cada puesto de trabajo.
Recibir_datos	Se usa para recibir por la red los datos que envían los puestos de trabajo
Enviar_datos	Esta funcionalidad se usa para enviar los distintos comandos a los puestos de trabajo.
Tiempo_examen	Para devolver el tiempo que lleva un examen determinado.
Cant_preg	Devuelve la cantidad total de preguntas de un examen.
Preg_actual	Devuelve el numero de la pregunta actual de un examen determinado
Asignar_pto	Permite la asignación de un cliente a un puesto de trabajo para que realice un examen .
Canc_exam	Permite cancelar el examen en un puesto de trabajo
Buscar_clientes	Permite realizar una búsqueda de los clientes según un criterio de búsqueda pasado por parámetro
Obtener_examen	Permite visualizar los datos de un examen que acabe de concluir en un puesto de trabajo, cargandolos de la BD.

4.2.2.4.2 Vista Cancelar examen

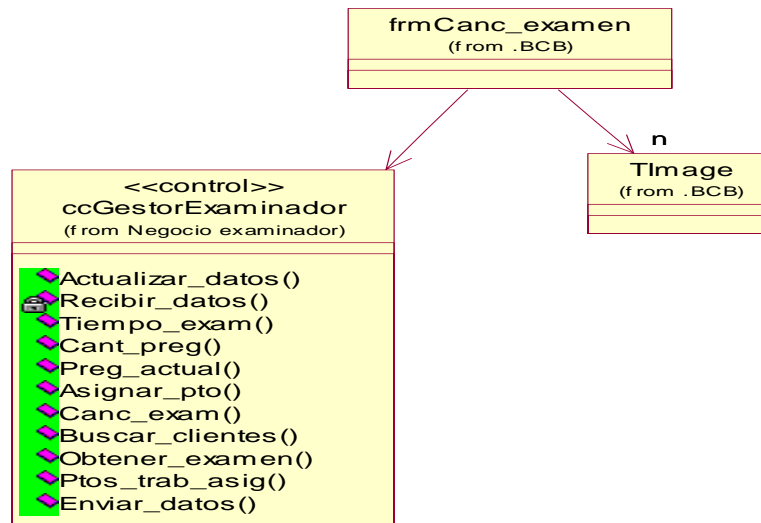
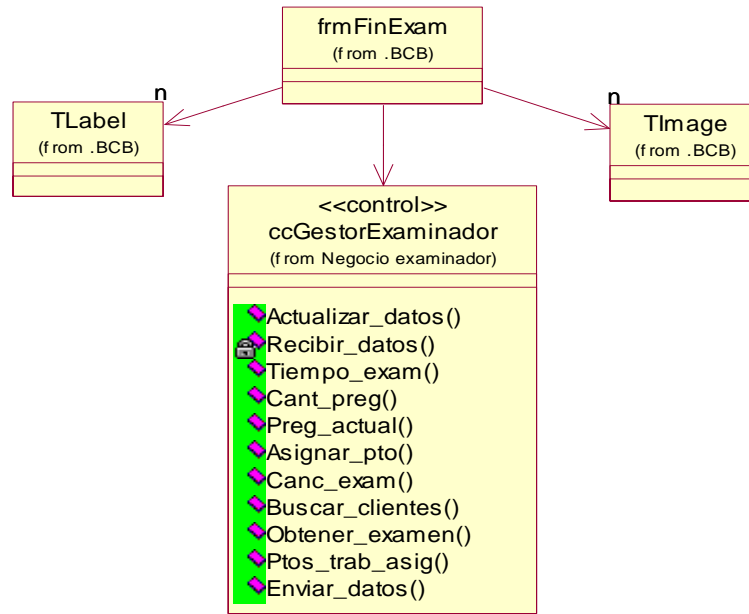


Ilustración 20. Diagrama de clases de la Vista Cancelar examen del paquete Examinador en la Capa de Aplicación

4.2.2.4.3 Vista Fin de examen



**Ilustración 21. Diagrama de clases de la Vista Fin de examen del paquete Examinador en la Capa de Aplicación**

**4.2.2.4.4 Vista Asignar**

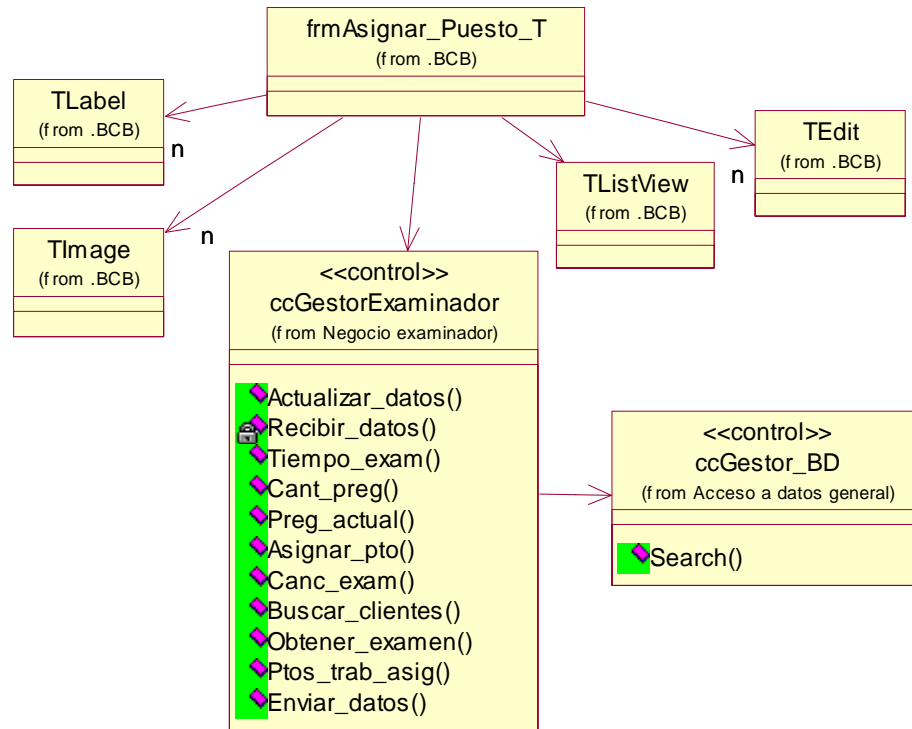


Ilustración 22. Diagrama de clases de la Vista Asignar del paquete Examinador en la Capa de Aplicación

**Descripción de las clases que conforman el diagrama:**

A continuación se describen las clases que conforman el diagrama del paquete examinador, en la vista Asignar, solamente se describe la clase ccGestorBD, ya que las otras son del paquete .BCB o ya han sido descritas.

**Tabla 36. Descripción de la clase ccGestorBD**

<b>Nombre: ccGestorBD</b>	
<b>Tipo de clase: controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre:	Descripción:

Search	Esta responsabilidad se encarga de buscar según el parámetro especificado y en la tabla que se le especifique de la BD, devuelve una lista de objetos que cumplan con la condición pasada como parámetro.
--------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 4.2.3 Capa Negocio

#### 4.2.3.1 Paquete Negocio cliente

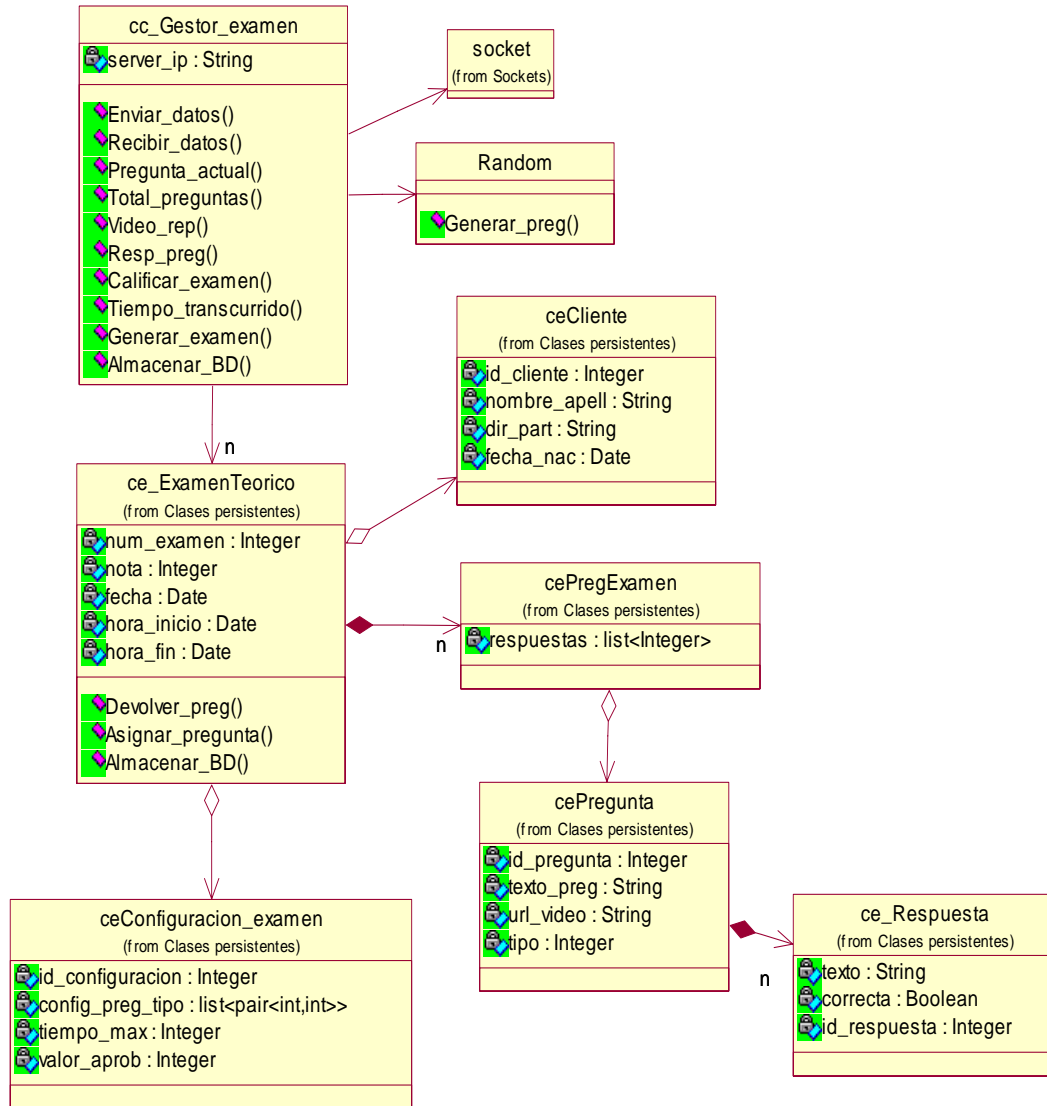


Ilustración 23. Diagrama de clases del paquete Negocio cliente

### Descripción de las clases que conforman el diagrama:

A continuación se describen las clases que conforman el diagrama del paquete Negocio cliente.

**Tabla 37. Descripción de la clase Random**

<b>Nombre: Random</b>	
<b>Tipo de clase: entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	<b>Descripción:</b>
Generar_preg	Se encarga de generar una pregunta para el examen teorico, esta generación se hace de forma aleatoria.

**Tabla 38. Descripción de la clase ce\_ExamenTeorico**

<b>Nombre: ce_ExamenTeorico</b>	
<b>Tipo de clase: entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
num_examen	Integer
nota	Integer
fecha	Date
hora_inicio	Date
hora_fin	Date
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	<b>Descripción:</b>
Devolver_preg	Devuelve la pregunta que le corresponde responder en ese momento al cliente, esto se hace mediante una lista devolviendo el primer elemento hasta que todas estén respondidas.
Asignar_pregunta	Permite asignar una de las preguntas generadas al examen

Tabla 38. Descripción de la clase cePregExamen

<b>Nombre: cePregExamen</b>	
<b>Tipo de clase: entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
respuestas	list<Integer>
<b>Para cada responsabilidad:</b>	
Nombre:	Descripción:

Tabla 39. Descripción de la clase ceConfiguracion\_examen

<b>Nombre: ceConfiguracion_examen</b>	
<b>Tipo de clase: entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
id_configuracion	Integer
config_preg_tipo	list<pair<Integer,Integer>>
tiempo_max	Integer
valor_aprob	Integer

Tabla 40. Descripción de la clase cePregunta

<b>Nombre: cePregunta</b>	
<b>Tipo de clase: entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
id_pregunta	Integer
texto_preg	String
url_video	String
tipo	Integer

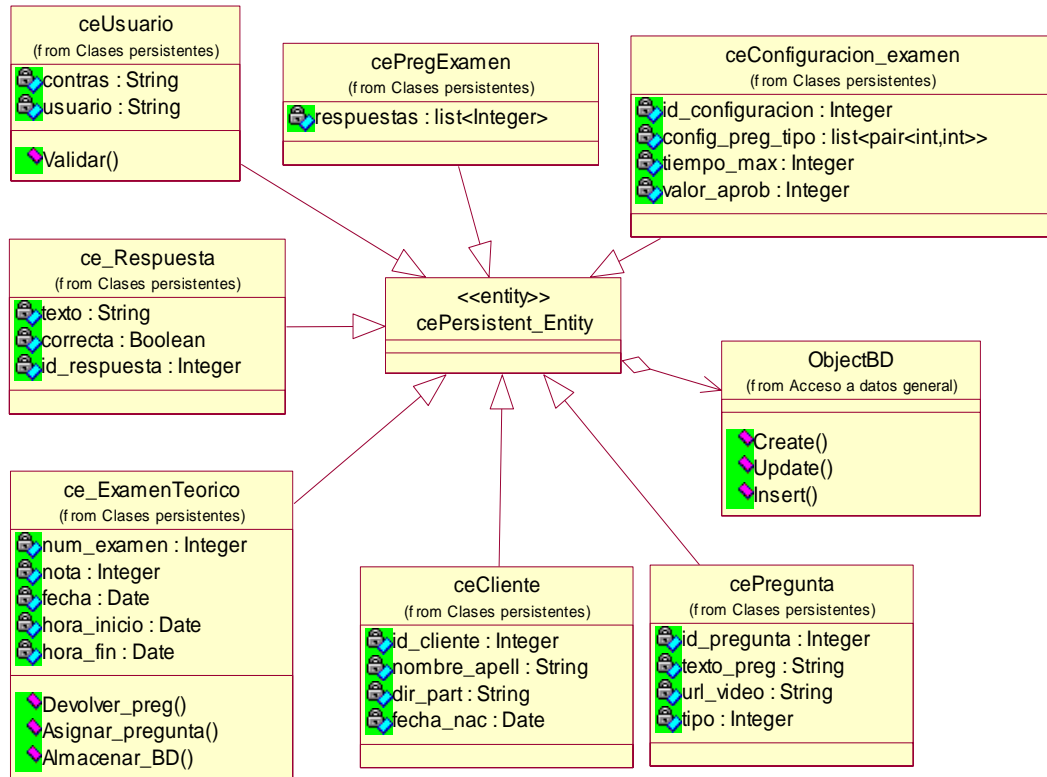
Tabla 41. Descripción de la clase ce\_Cliente

<b>Nombre: ce_Cliente</b>	
<b>Tipo de clase: entidad</b>	
<b>Atributo</b>	<b>Tipo</b>



id_cliente	Integer
nombre_apell	String
dir_part	String
fecha_nac	Date
usuario	String

**4.2.3.2 Paquete Herencia persistencia**



**Ilustración 24.**

**Diagrama de clases del paquete Herencia persistencia**

**Descripción de las clases que conforman el diagrama:**

A continuación se describen las clases que conforman el diagrama del paquete Herencia persistencia. Aquí aparece la clase cePersistent\_Entity que es una clase que agrupa a todos los

objetos que van a ser persistentes en la BD, esta clase tiene una instancia de la clase ObjectBD y de ella heredan todas las clases que serán persistentes.

**Tabla 42. Descripción de la clase cePersistent\_Entity**

<b>Nombre: cePersistent_Entity</b>	
<b>Tipo de clase: entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre:	Descripción:

**Tabla 43. Descripción de la clase ceUsuario**

<b>Nombre: ceUsuario</b>	
<b>Tipo de clase: entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
contras	String
usuario	String
<b>Para cada responsabilidad:</b>	
Nombre:	Descripción:
Validar	Es la encargada de realizar la validación del usuario y la contraseña en la base de datos.

**Tabla 44. Descripción de la clase ObjectBD**

<b>Nombre: ObjectBD</b>	
<b>Tipo de clase: entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre:	Descripción:

Create	Es la encargada de a partir de un identificador realizar la búsqueda de los datos en la BD y devuelve el objeto con todos sus datos
--------	-------------------------------------------------------------------------------------------------------------------------------------

4.2.3.3 Paquete Negocio Examinador

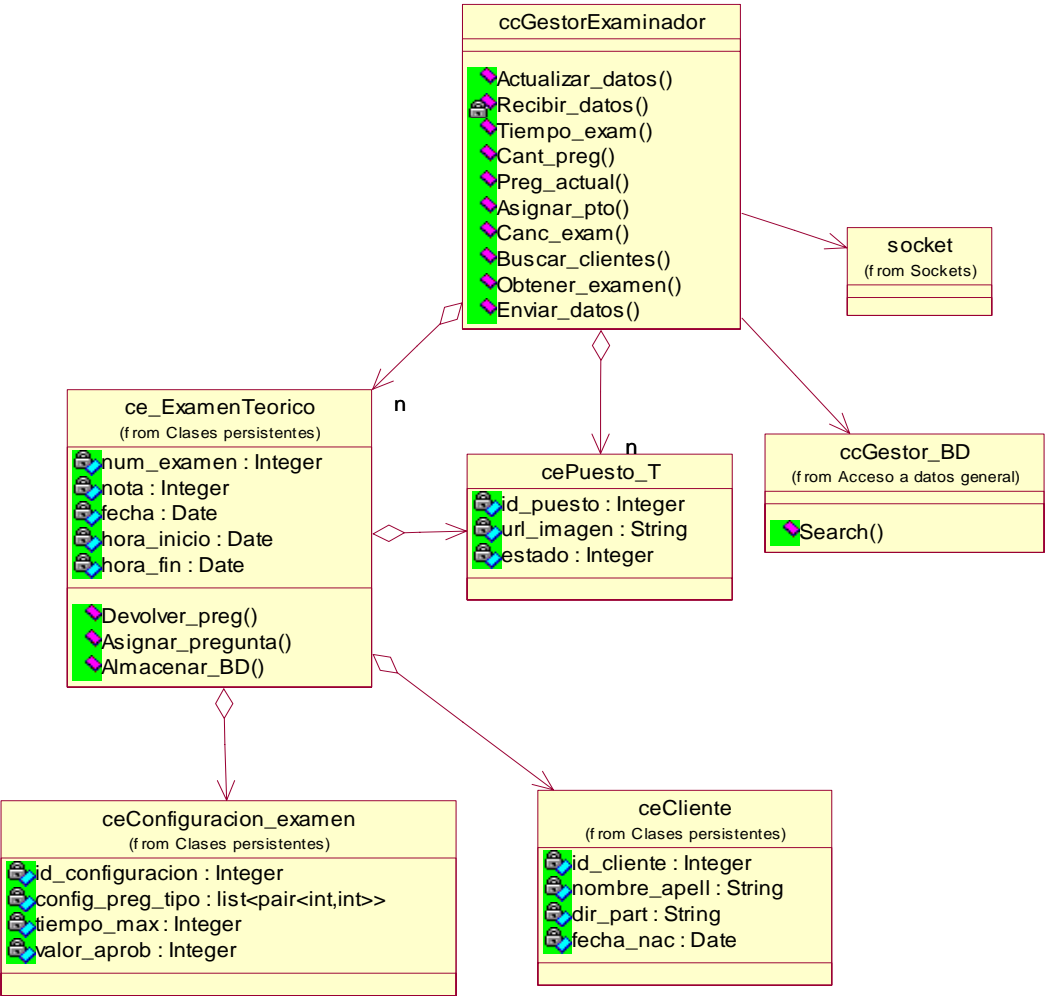


Ilustración 25. Diagrama de clases del paquete Negocio examinador

**Descripción de las clases que conforman el diagrama:**

A continuación se describen las clases que conforman el diagrama del paquete Negocio Examinador

**Tabla 45. Descripción de la clase cePuesto\_T**

<b>Nombre: cePuesto_T</b>	
<b>Tipo de clase: entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
id_puesto	Integer
url_imagen	String
estado	Integer
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	<b>Descripción:</b>
Obtener_examen	Esta funcionalidad devuelve el identificador del examen que se realiza en ese puesto de trabajo
Asignar_examen	Permite asignar un identificador de examen a un puesto de trabajo

4.2.4 Capa Acceso a Datos

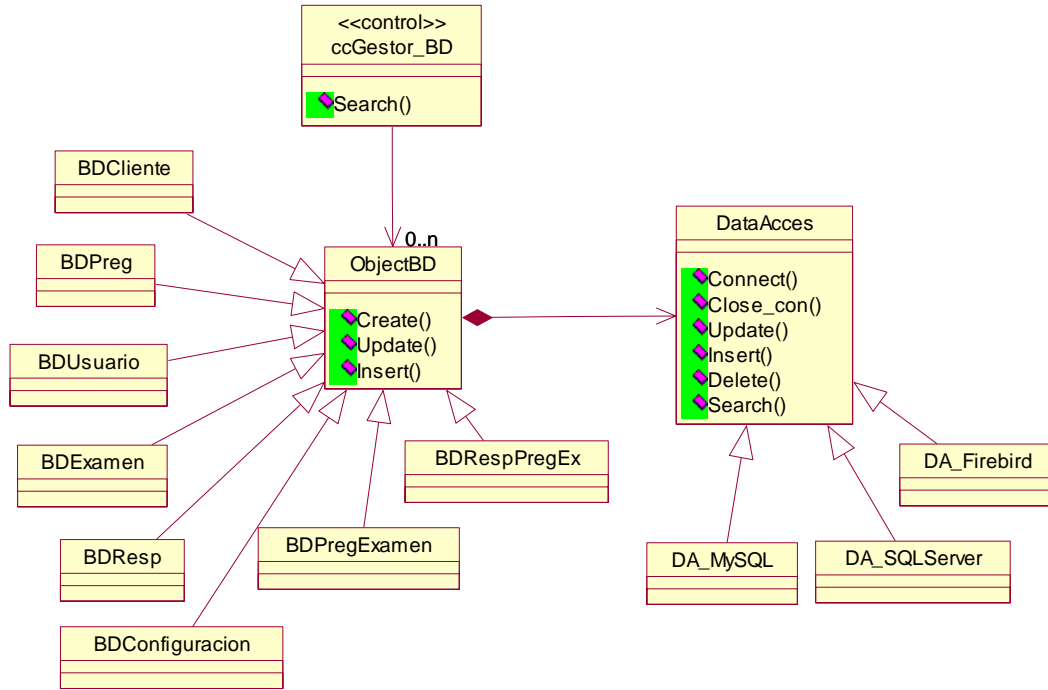


Ilustración 26. Diagrama de clases Capa Acceso a Datos

Descripción de las clases que conforman el diagrama:

A continuación se describen las clases que conforman el diagrama de la capa de Acceso a Datos.

Tabla 46. Descripción de la clase DataAcces

<b>Nombre: DataAcces</b>	
<b>Tipo de clase: entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre:	Descripción:
Connect	Esta funcionalidad es la encargada de establecer la conexión al servidor de bases de datos

Close_con	Permite cerrar la conexión con la base de datos
Update	Permite actualizar datos en la base de datos
Insert	Permite la inserción de datos en una tabla de la base de datos
Delete	Permite la eliminación de una tupla en la base de datos
Search	Permite realizar búsquedas en la BD.

### 4.3 Realización de los casos de uso

En los siguientes epígrafes se representan las realizaciones de caso de uso de los que se desarrollan en la primera iteración, mediante un diagrama de secuencia. No se representan los diagramas de clase de cada caso de uso, porque se estructuraron por paquetes, los cuales fueron representados en los epígrafes anteriores.

#### 4.3.1 Generar examen

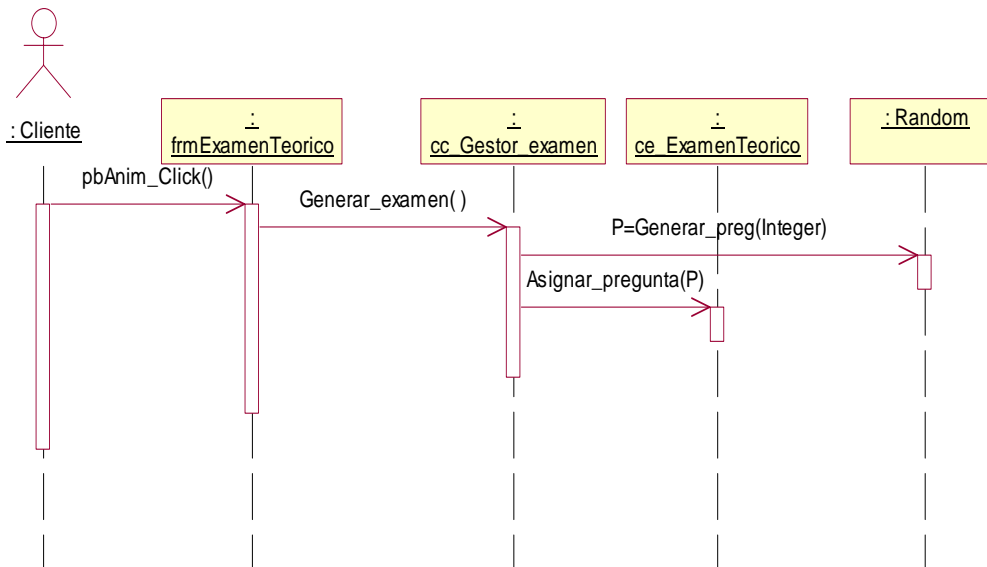


Ilustración 27.

Diagrama de secuencia del caso de uso Generar examen

### 4.3.2 Realizar examen

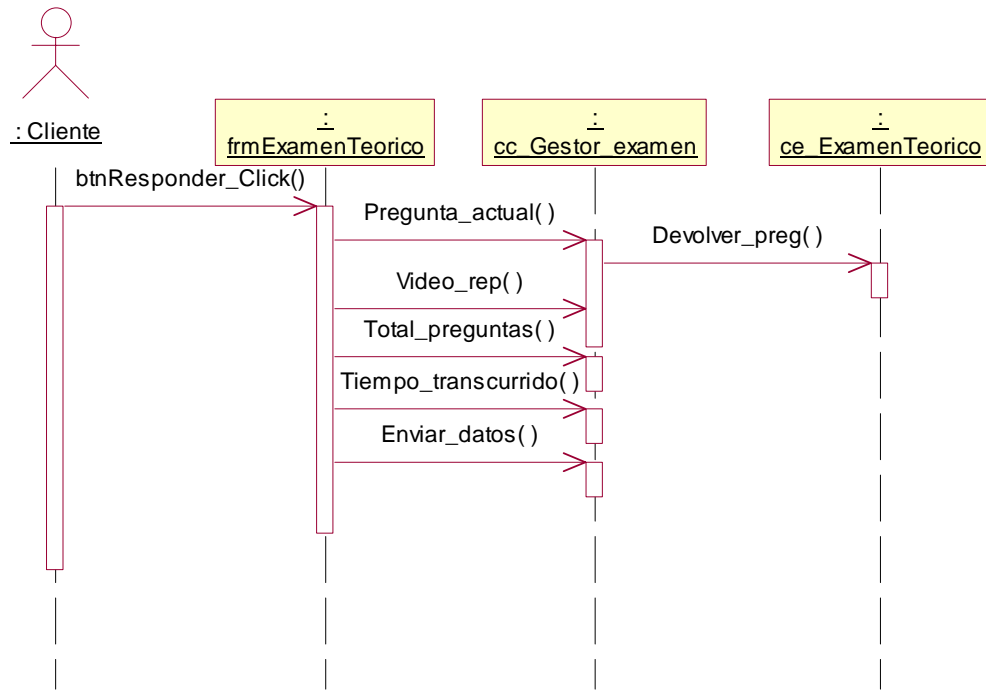


Ilustración 28. Diagrama de secuencia del caso de uso Realizar examen

### 4.3.3 Calificar examen

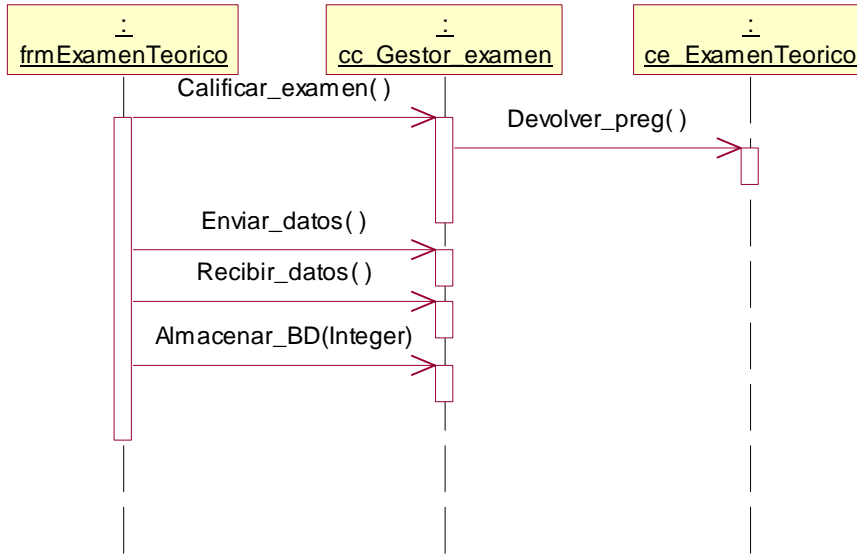


Ilustración 29. Diagrama de secuencia del caso de uso Calificar examen

### 4.3.4 Conectar con examinador

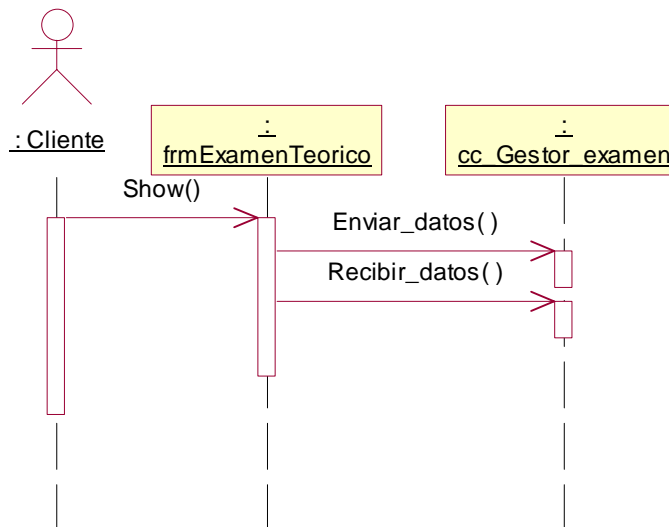




Ilustración 30. Diagrama de secuencia del caso de uso Conectar con examinador

#### 4.3.5 Almacenar datos de examen

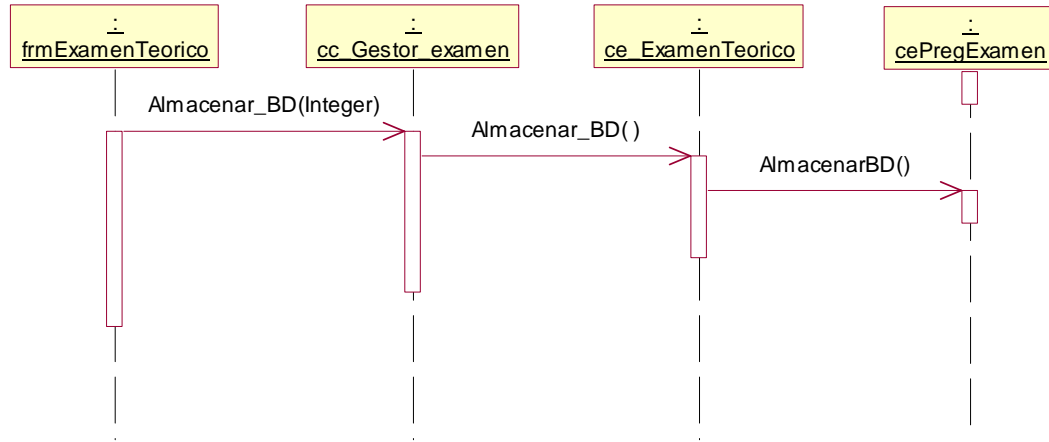


Ilustración 31. Diagrama de secuencia del caso de uso Almacenar datos de examen

### 4.3.6 Buscar

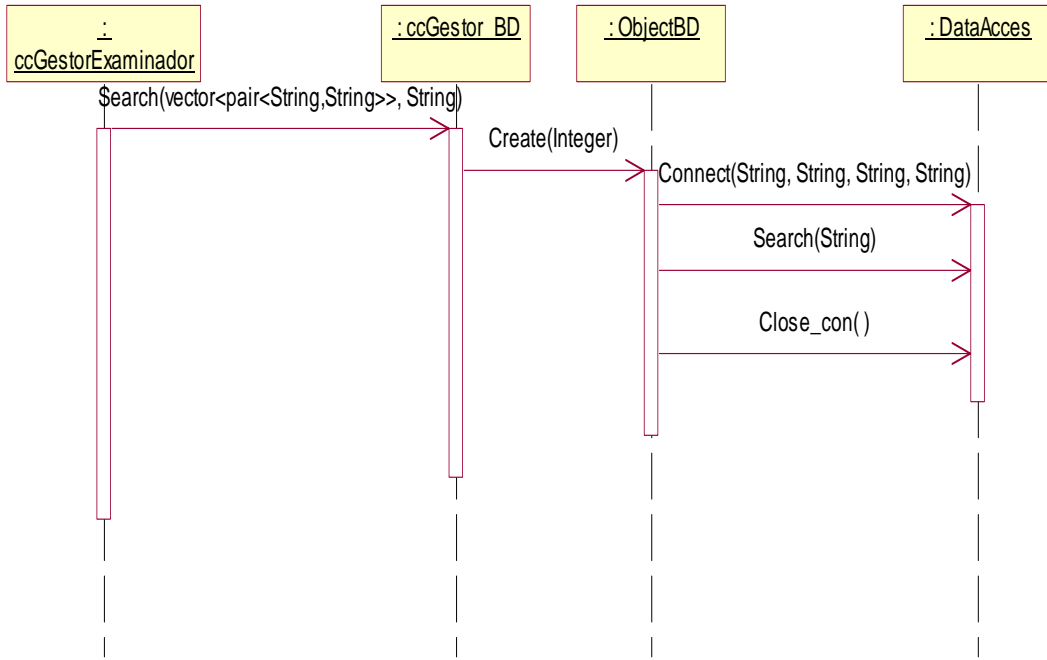


Ilustración 32.

Diagrama de secuencia del caso de uso Conectar con examinador

### 4.3.7 Cancelar examen

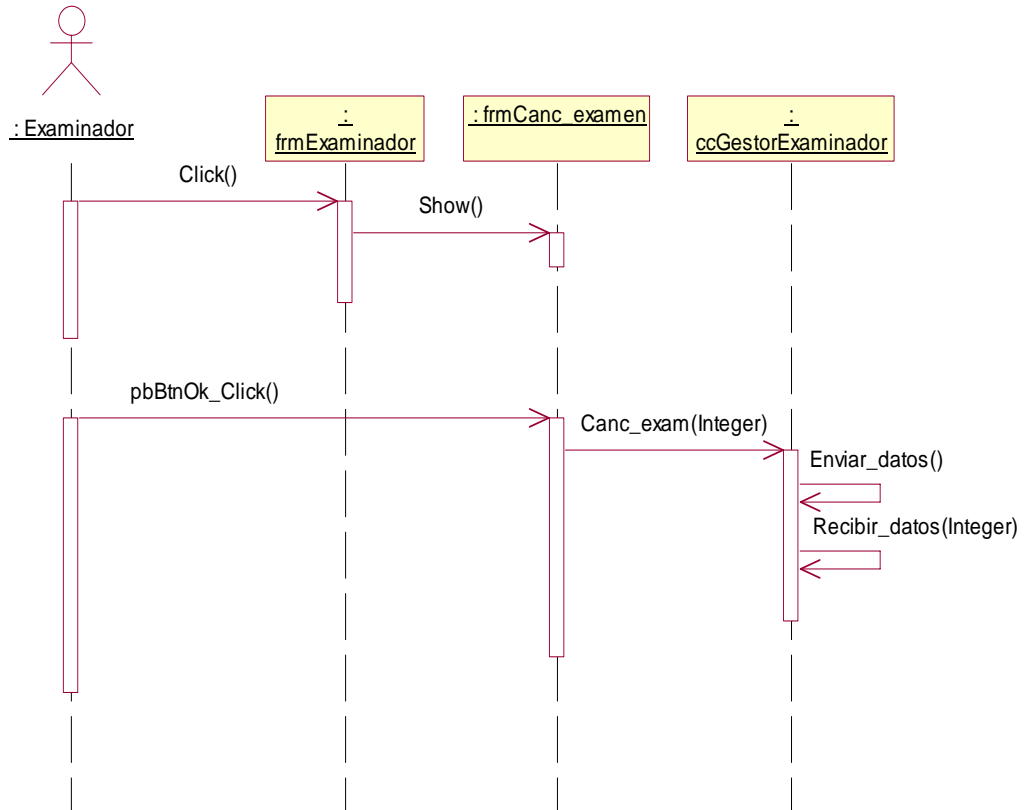


Ilustración 33. Diagrama de secuencia del caso de uso Cancelar examen

#### 4.3.8 Asignar puesto de examen

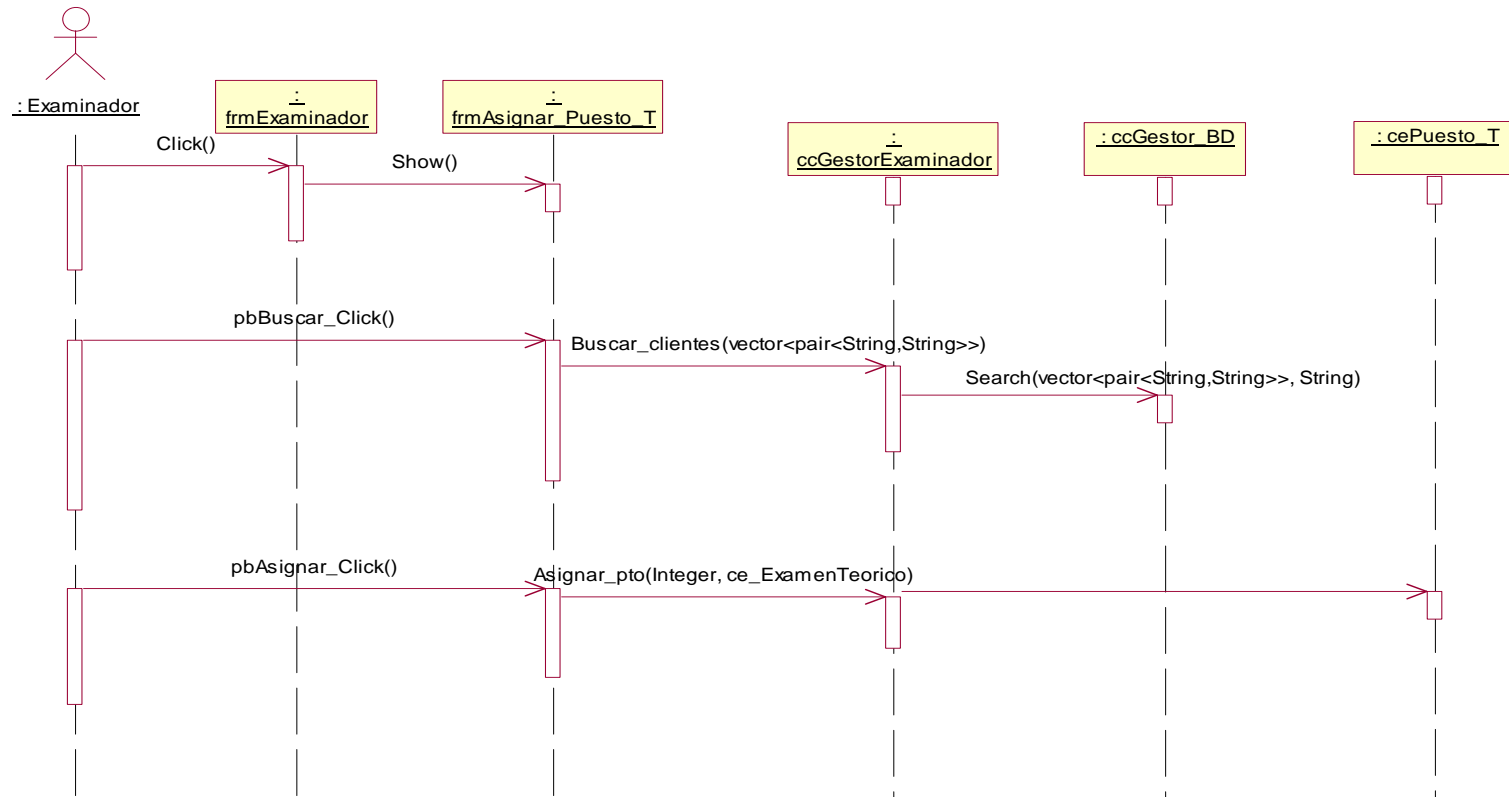


Ilustración 34. Diagrama de secuencia del caso de uso Asignar puesto de examen

### 4.3.9 Chequear datos de puesto de trabajo

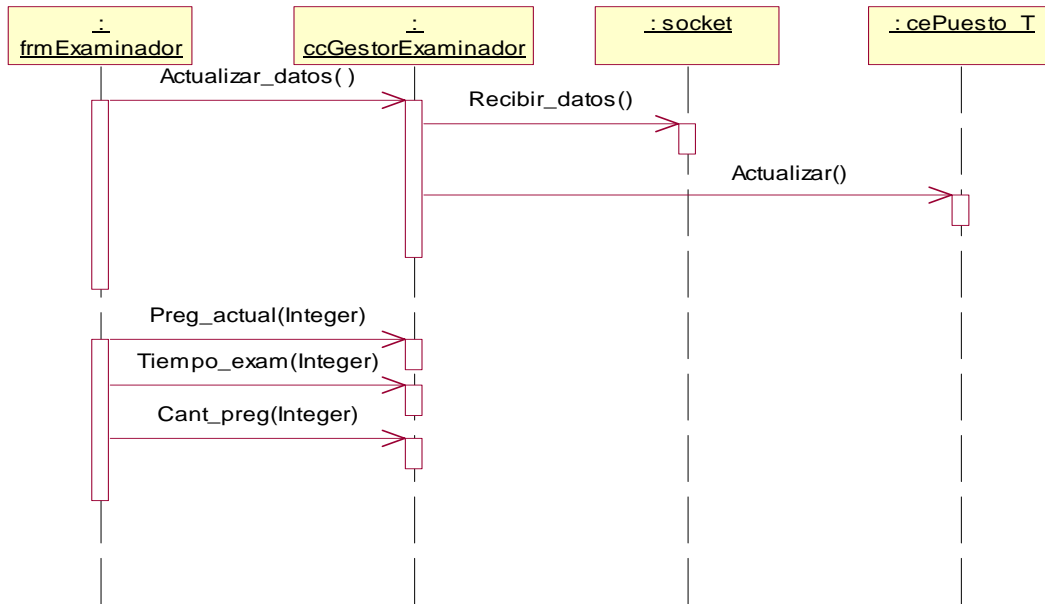


Ilustración 35. Diagrama de secuencia del caso de uso Chequear datos de puesto de trabajo

### 4.3.10 Visualizar datos de examen

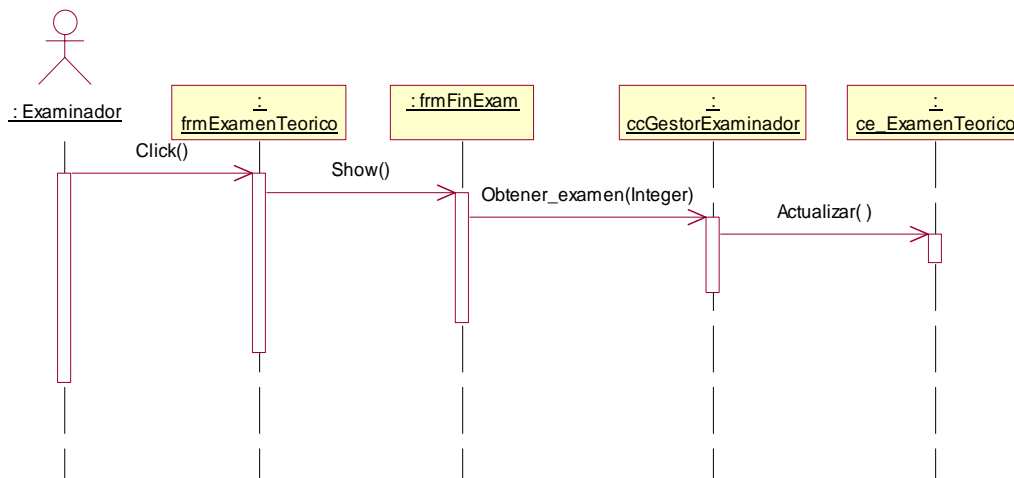


Ilustración 36. Diagrama de secuencia del caso de uso Visualizar datos de puesto de trabajo

### 4.3.11 Autenticar usuario

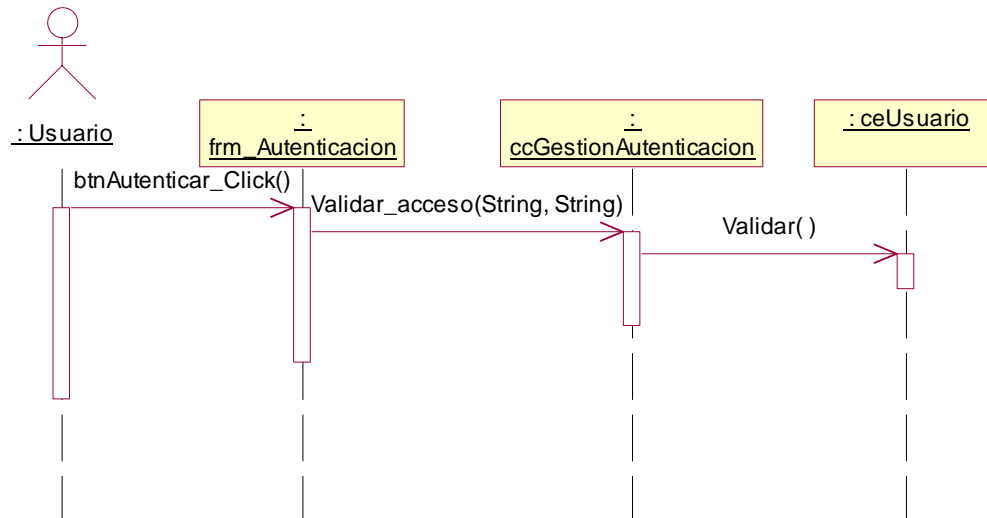


Ilustración 37. Diagrama de secuencia del caso de uso Autenticar usuario

## 4.4 Principios de diseño

La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso, es por eso que uno de los aspectos más relevantes de la usabilidad de un sistema es la consistencia de su interfaz de usuario.

Para el diseño de la interfaz del sistema se tuvieron en cuenta aspectos necesarios, que garanticen la comodidad por parte del usuario, teniendo presente la organización de la información que se muestra y su distribución en la pantalla. La interfaz gráfica del usuario es el medio por el cual este interactúa con el sistema, por lo que esta debe ser lo más amigable posible y lograr que se sienta identificado con la misma. Las pantallas del sistema contienen la información necesaria para evitar la sobrecarga, además de mantener las opciones principales en el mismo lugar de la interfaz para una mejor interacción y adaptabilidad del cliente con la

aplicación. El la interfaz de usuario del sistema es comprensible, con colores agradables y poco llamativos para no perder concentración

#### 4.4.1 Interfaz de usuario del Examinador

Se mantiene la interfaz que tenía el sistema anterior para el puesto del examinador, esta interfaz de usuario se corresponde con una versión para un sistema de evaluación teorica para obtener licencia de conducción, en caso de que se implante el software en otro contexto, la misma puede variar con el objetivo de adecuarse al entorno donde se aplique el software. La interfaz gráfica del examinador se muestra en la siguiente imagen:

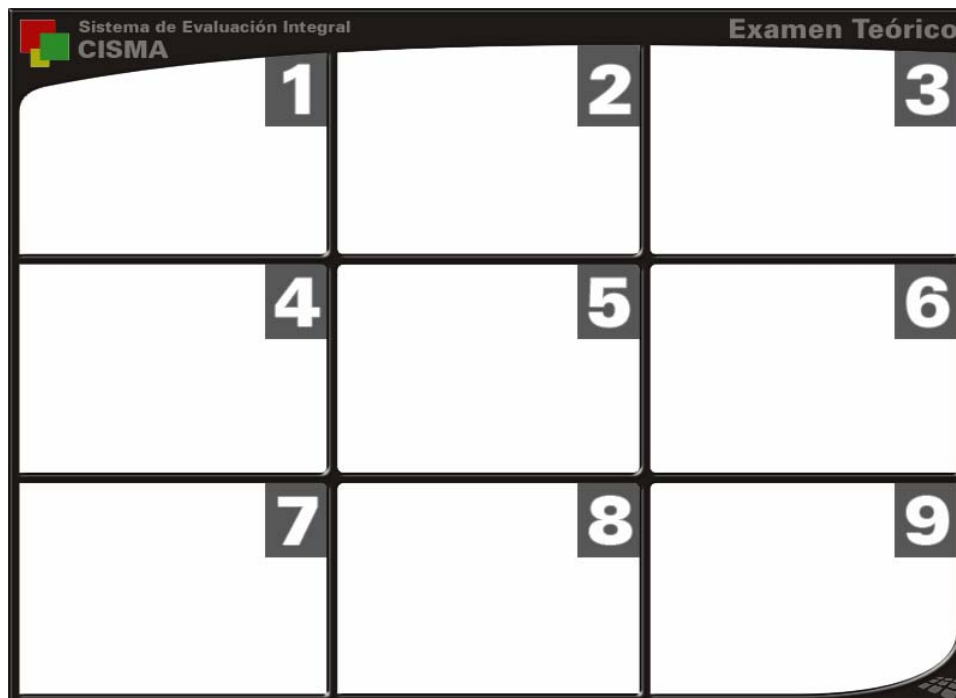


Ilustración 38. Interfaz de usuario del Examinador

#### 4.4.2 Interfaz de usuario del Cliente

La interfaz de usuario del sistema anterior para el puesto de trabajo del cliente, varía en algunos detalles: Se eliminan los botones Si y No para responder las preguntas, ya que las preguntas actuales pueden tener más de una respuesta. Además se añade en el área de reproducción del

video un panel, que será donde se muestren las distintas respuestas posibles de una pregunta y las opciones para responder la pregunta.



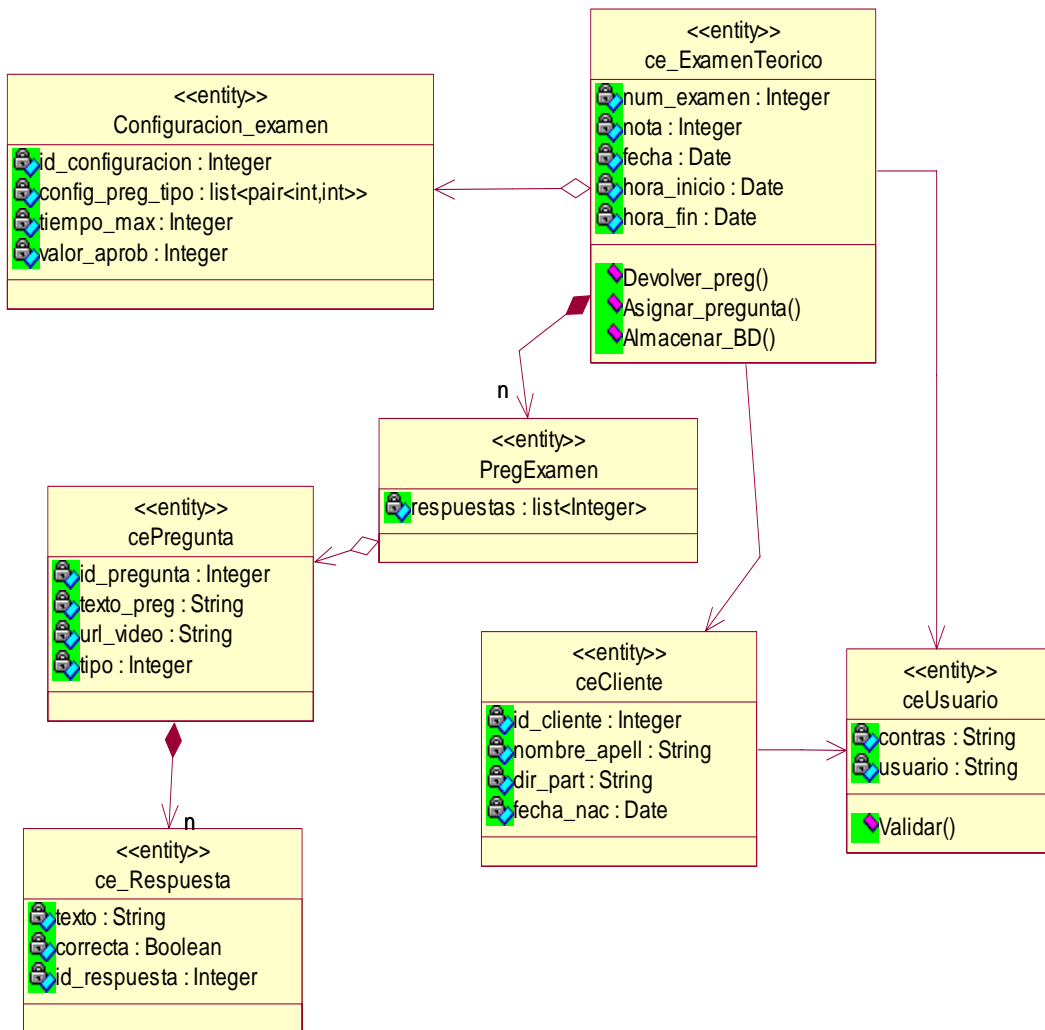
Ilustración 39. Interfaz de usuario del Cliente

## **4.5 Diseño de la base de datos**

### **4.5.1 Modelo lógico de datos**

La siguiente imagen muestra el modelo lógico de los datos, representado en el diagrama de clases persistentes del sistema.





**Ilustración 40. Modelo lógico de datos (Diagrama de clases persistentes)**

#### 4.5.2 Modelo físico de datos

En la siguiente imagen se muestra el modelo físico de los datos, que será la estructura que tendrá la Base de Datos que soporte el sistema.

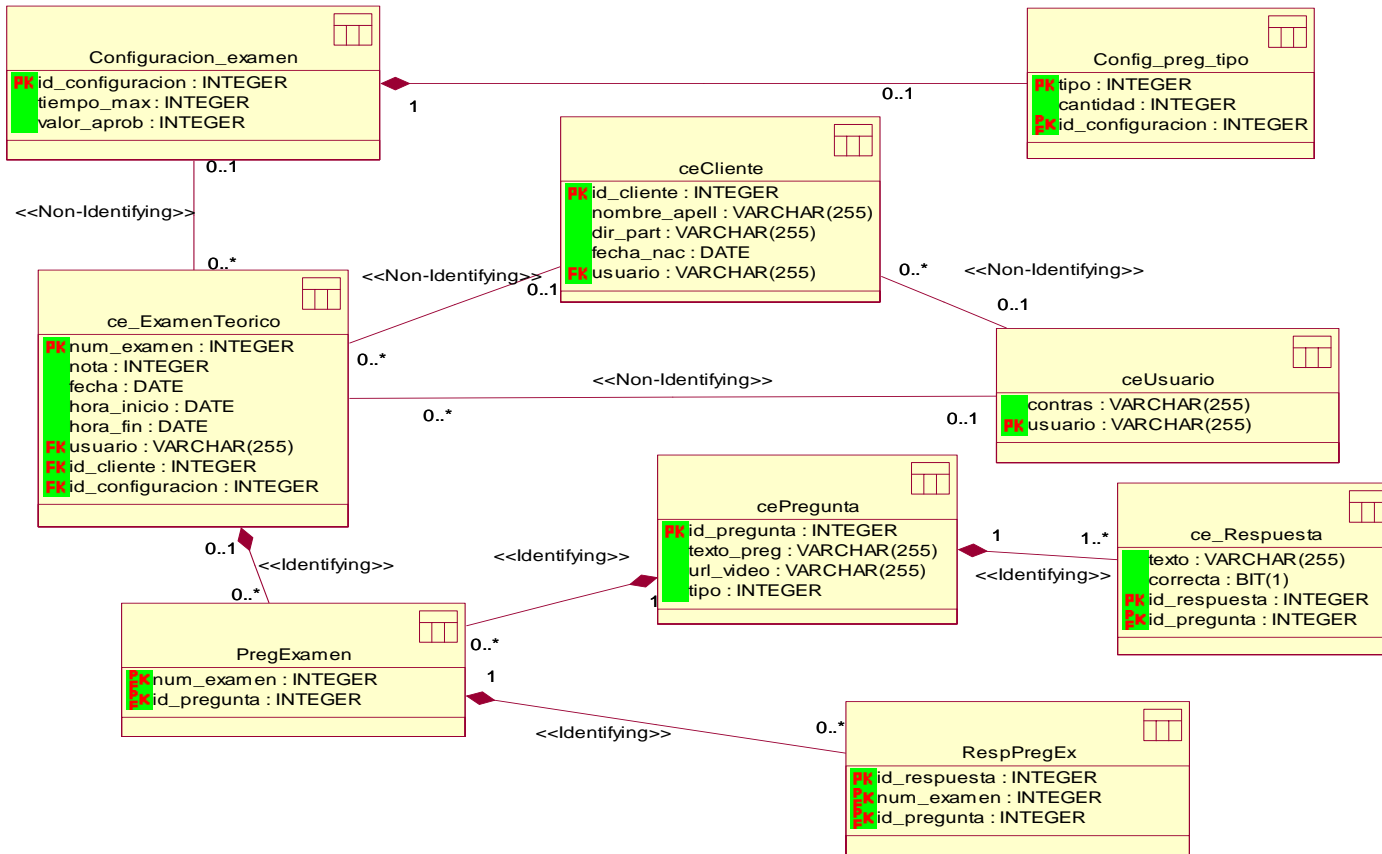


Ilustración 41. Modelo físico de datos

#### 4.5.2.1 Explicación de las tablas de la base de datos.

**Tabla 47. Descripción de la tabla ceCliente**

Nombre:	ceCliente.	
<b>Descripción:</b>	En esta tabla se almacenan los datos correspondientes a los clientes.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>id_cliente</b>	<b>Integer</b>	Identificador de la tabla
<b>nombre_apell</b>	<b>Varchar(255)</b>	Nombre y apellidos del cliente
<b>dir_part</b>	<b>Varchar(255)</b>	Dirección particular del cliente
<b>fecha_nac</b>	<b>Date</b>	Fecha de nacimiento del cliente
<b>usuario</b>	<b>Varchar(255)</b>	Contiene el nombre del usuario que lo registró en el sistema

**Tabla 48. Descripción de la tabla Configuracion\_examen**

Nombre:	Configuracion_examen	
<b>Descripción:</b>	En esta tabla se almacenan los datos correspondientes a las configuraciones que usarán los exámenes	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>id_configuracion</b>	<b>Integer</b>	Identificador de la tabla
<b>tiempo_max</b>	<b>Integer</b>	Tiempo máximo de duración del examen
<b>valor_aprob</b>	<b>Integer</b>	Valor mínimo a alcanzar en el examen para alcanzar la categoría de aprobado.

**Tabla 49. Descripción de la tabla Config\_preg\_tipo**

Nombre:	Config_preg_tipo	
<b>Descripción:</b>	En esta tabla se almacenan los datos correspondientes a las cantidad de preguntas de cada tipo que se usarán para cada examen, es una tabla débil de Configuración_examen.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>id_configuracion</b>	<b>Integer</b>	Identificador de la tabla, referencia un

		identificador de la configuración
<b>tipo</b>	<b>Integer</b>	Tipo de pregunta que se usará, tipo es una categoría de las preguntas
<b>cantidad</b>	<b>Integer</b>	Cantidad de preguntas de un tipo que se incluirá en la configuración.

**Tabla 50. Descripción de la tabla cePregunta**

Nombre:	cePregunta	
<b>Descripción:</b>	En esta tabla se almacenan los datos correspondientes a las preguntas que se usarán para los exámenes	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>id_pregunta</b>	<b>Integer</b>	Identificador de la tabla
<b>texto_preg</b>	<b>Varchar(255)</b>	Texto que acompaña a la pregunta
<b>url_video</b>	<b>Varchar(255)</b>	Nombre del video que representa a la pregunta
<b>Tipo</b>	<b>Integer</b>	Tipo de la pregunta, las configuraciones de exámenes tienen las preguntas por tipo.

**Tabla 51. Descripción de la tabla ce\_ExamenTeorico**

Nombre:	ce_ExamenTeorico	
<b>Descripción:</b>	En esta tabla se almacenan los datos correspondientes a los exámenes que se realizan	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>id_configuracion</b>	<b>Integer</b>	Referencia a la configuración que se usará en cada examen
<b>num_examen</b>	<b>Integer</b>	Identificador de la tabla
<b>nota</b>	<b>Integer</b>	Puntuación alcanzada en el examen
<b>fecha</b>	<b>Date</b>	Fecha en que se realiza el examen
<b>hora_inicio</b>	<b>Date</b>	Hora en que comienza el examen
<b>hora_fin</b>	<b>Date</b>	Hora en que se culmina el examen

<b>usuario</b>	<b>Varchar(255)</b>	Nombre del examinador que le asignó el examen
<b>id_cliente</b>	<b>Integer</b>	Identificador del cliente que realiza el examen

**Tabla 52. Descripción de la tabla ceRespuesta**

Nombre:	ceRespuesta	
<b>Descripción:</b>	En esta tabla se almacenan los datos correspondientes a las posibles respuestas a mostrarle al cliente por cada pregunta. Es débil de la tabla pregunta.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>id_respuesta</b>	<b>Integer</b>	Identificador de la tabla
<b>id_pregunta</b>	<b>Integer</b>	Identificador de la tabla, referencia a la pregunta propietaria de la respuesta
<b>correcta</b>	<b>Bit(1)</b>	Determina si la respuesta es correcta o no
<b>texto</b>	<b>Varchar(255)</b>	Texto de la respuesta

**Tabla 53. Descripción de la tabla PregExamen**

Nombre:	PregExamen	
<b>Descripción:</b>	En esta tabla se almacenan los datos correspondientes a las preguntas que conforman cada examen. Es débil de la tabla examen	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>num_examen</b>	<b>Integer</b>	Identificador de la tabla, es el numero del examen
<b>id_pregunta</b>	<b>Integer</b>	Identificador de la tabla, referencia a la pregunta propietaria de la respuesta

**Tabla 54. Descripción de la tabla RespPregEx**

Nombre:	RespPregEx
<b>Descripción:</b>	En esta tabla se almacenan los datos correspondientes a las respuestas que selecciona el cliente para cada pregunta del examen. Es débil de la tabla PregExamen

<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>num_examen</b>	<b>Integer</b>	Identificador de la tabla, es el numero del examen
<b>id_pregunta</b>	<b>Integer</b>	Identificador de la tabla, referencia a la pregunta propietaria de la respuesta
<b>id_respuesta</b>	<b>Integer</b>	Identificador de la tabla, representa el identificador de la respuesta seleccionada por el cliente.

**Tabla 55. Descripción de la tabla ceUsuario**

<b>Nombre:</b>	ceUsuario	
<b>Descripción:</b>	En esta tabla se almacenan los datos correspondientes a los usuarios.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>usuario</b>	<b>Varchar(255)</b>	Identificador de la tabla, es el nombre de usuario
<b>contraseña</b>	<b>Varchar(255)</b>	Contraseña con la cual el usuario puede acceder a las distintas aplicaciones.

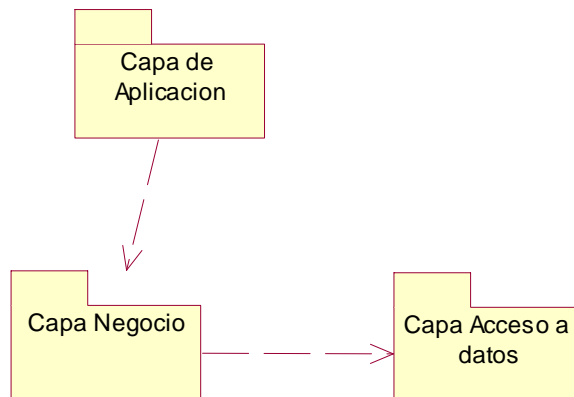
#### **4.6 Modelo de implementación.**

El modelo de implementación describe como se organizan y se relacionan los elementos del modelo del diseño, así como se implementan en términos de componentes, definiendo un componente como el empaquetamiento físico de los elementos de un modelo, como es el caso de las clases del modelo de diseño.

Un Diagrama de Componentes representa la separación de un sistema de software en componentes físicos (por ejemplo archivos, cabeceras, módulos, paquetes, etc.). El Diagrama de Componentes se usa para modelar la estructura del software, incluyendo las dependencias entre los componentes de software, los componentes de código binario, y los componentes ejecutables. En el Diagrama de Componentes se modelan componentes del sistema, a veces

agrupados por paquetes, y las dependencias que existen entre componentes (y paquetes de componentes).

#### 4.6.1 Diagrama de paquetes.



**Ilustración 42. Diagrama de paquetes del modelo de implementación**

4.6.2 Diagrama de componentes Capa Acceso a datos.

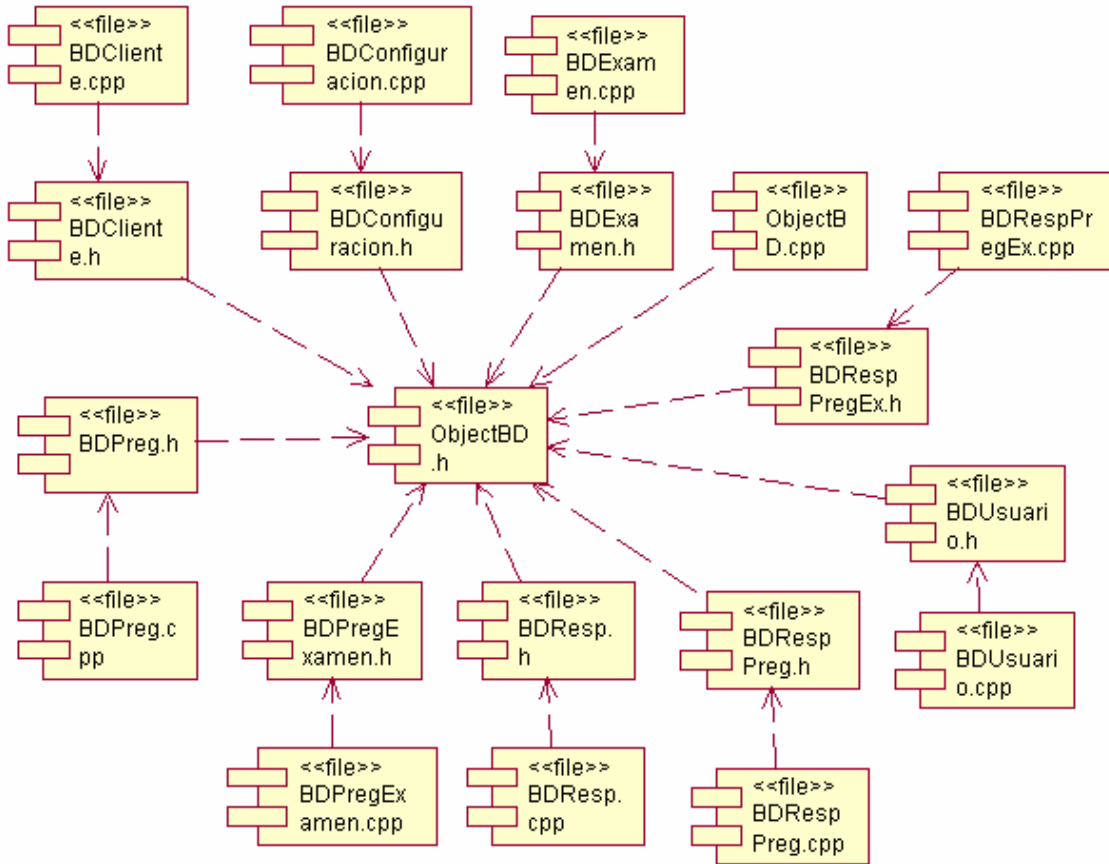


Ilustración 43. Diagrama de componentes de la Capa de Acceso a Datos. Vista ObjectBD



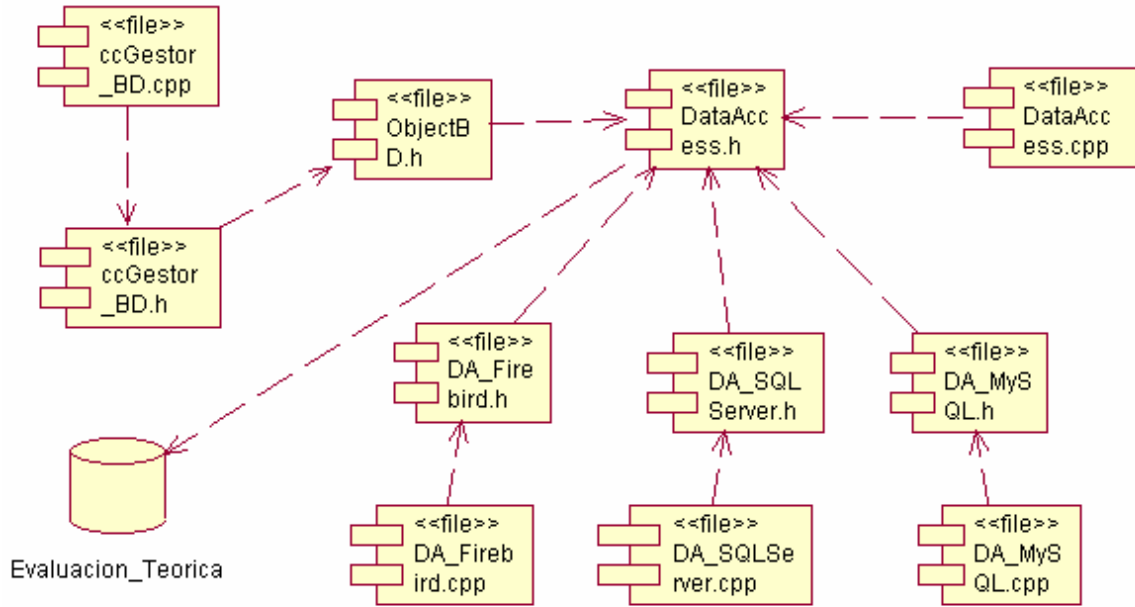


Ilustración 44. Diagrama de componentes de la Capa de Acceso a Datos. Vista Control Acceso BD

### 4.6.3 Diagrama de componentes Capa de Aplicación.

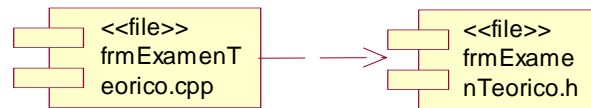
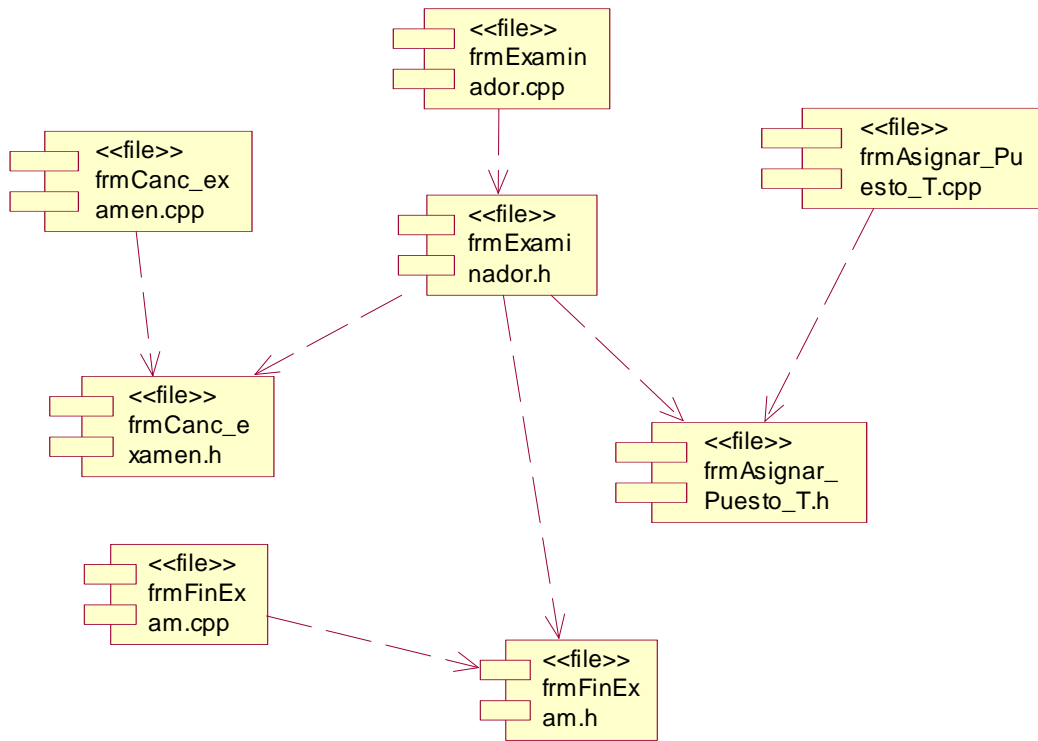


Ilustración 45. Diagrama de componentes Cliente. Vista Capa Aplicación



Ilustración 46. Diagrama de componentes Paquete Capa Aplicación (Vista Autenticación)



**Ilustración 47. Diagrama de componentes Capa Aplicación (Vista Examinador)**

4.6.4 Diagrama de componentes Capa de Negocio.

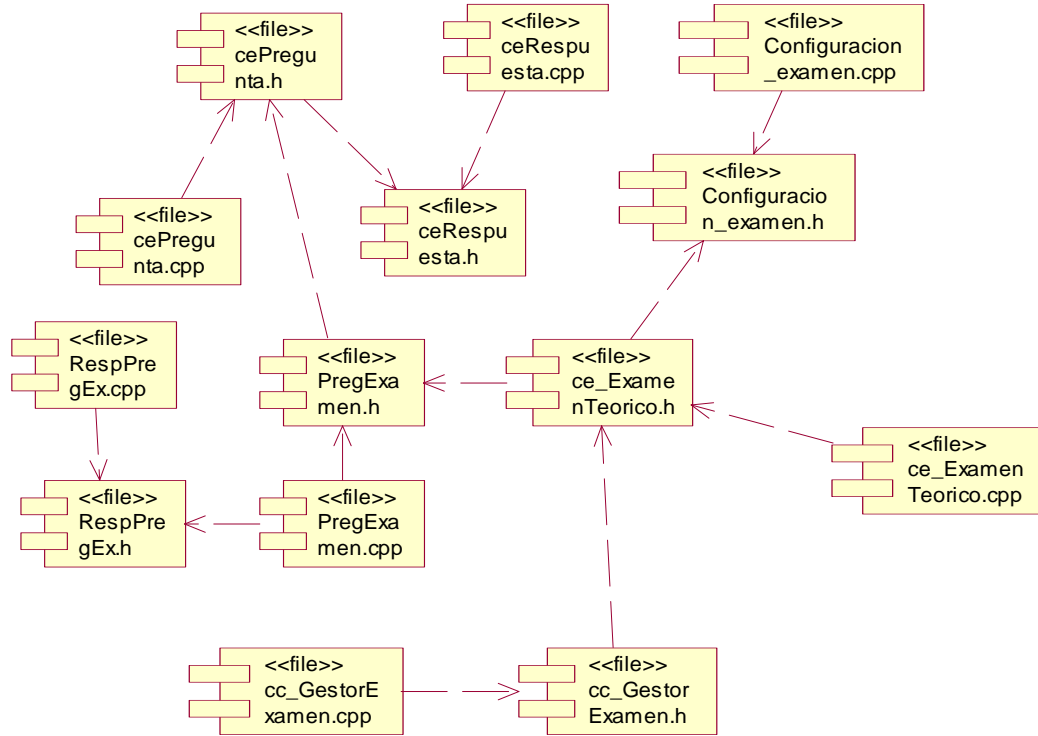
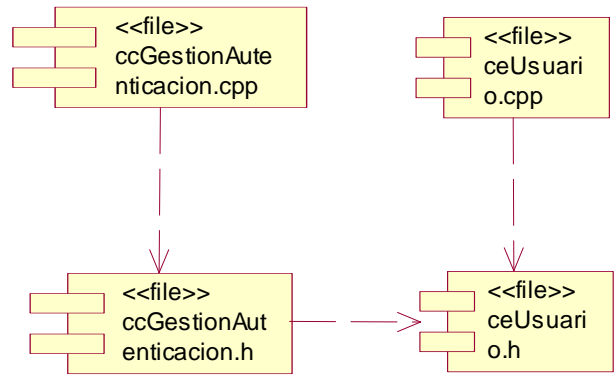
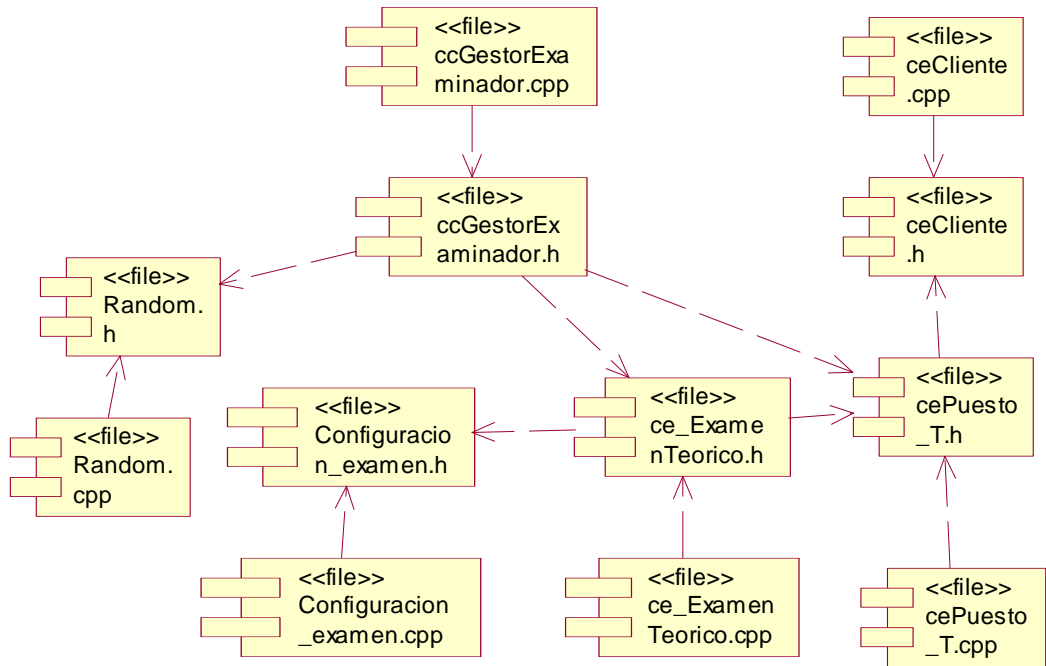


Ilustración 48. Diagrama de componentes Capa Negocio. (Vista Cliente)



**Ilustración 49. Diagrama de componentes Capa Negocio (Vista Autenticación)**



**Ilustración 50. Diagrama de componentes Capa Negocio (Vista Examinador)**

#### 4.6.5 Diagrama de componentes binarios.

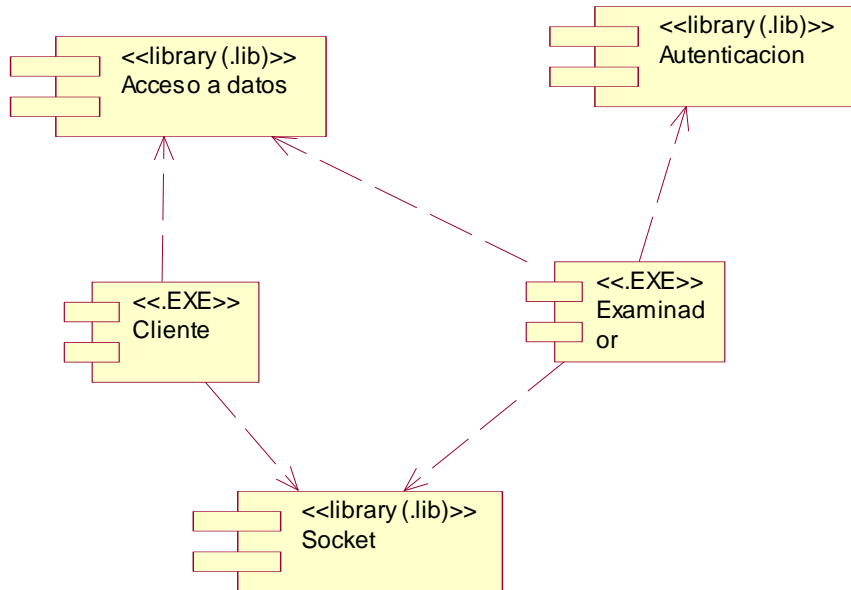
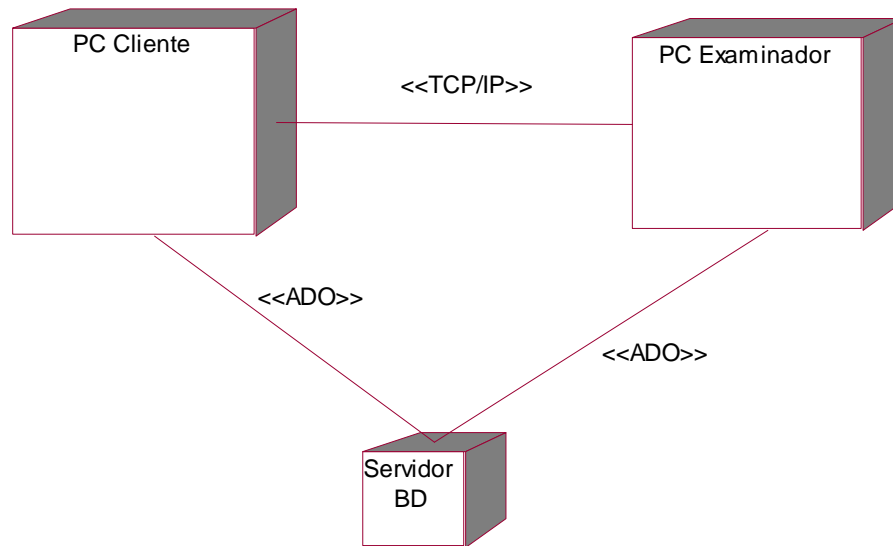


Ilustración 51. Diagrama de componentes binarios

#### 4.7 Diagrama de despliegue

En la ilustración 52 se muestra el Diagrama de despliegue del sistema desarrollado. El mismo cuenta con cinco nodos: uno de ellos para la PC del Cliente que es donde se ejecutará el examen teórico, un puesto para la PC del Examinador, que será el encargado de asignar exámenes y chequear el estado de ejecución de los mismos, entre el puesto del cliente y el examinador debe existir una conexión la cual permitirá controlar las variables de cada examen como son tiempo de ejecución, total de preguntas generadas, video que se reproduce, entre otras. El puesto del Examinador debe conectarse a la base de datos para poder asignar los exámenes y registrarlos.



**Ilustración 52. Diagrama de despliegue**

## **4.8 Conclusiones**

En el capítulo que concluye se desarrollaron los flujos de trabajo de diseño e implementación. Se muestra la realización de cada caso de uso de los seleccionados para la primera iteración del software. Se obtiene además el diseño de la base de datos que soporta el sistema y se describen las tablas que la conforman. En este capítulo se han obtenido todos los elementos posibles para lograr una correcta implementación.

## Conclusiones Generales

A partir de la investigación realizada para la obtención de este Sistema de Evaluación Teórica, utilizando el Proceso Unificado de Desarrollo de Software (RUP), y el Lenguaje de Modelación Unificado (UML) se llega a las siguientes conclusiones:

- A través del estudio que se llevó a cabo se detectaron varios aspectos que incidían en el desempeño del anterior sistema de evaluación teórica.
- Como solución a los problemas encontrados se decide realizarle una reingeniería al sistema existente para resolver los defectos que tenía.
- La herramienta desarrollada, incorpora nuevas funcionalidades al sistema ya existente y brinda la posibilidad de añadir nuevos módulos sin tener que cambiar completamente el sistema.
- Se obtiene una *framework* para el desarrollo de sistemas de evaluación teórica que pueden usarse en cualquier lugar que se necesite evaluar conocimientos teóricos, no solo para obtener licencia de conducción.
- Se facilita la migración a otras plataformas al encapsular bien los diferentes módulos que componen el sistema y separar completamente la interfaz de usuario de la lógica del negocio y ambas de la capa de acceso a datos.

Con la propuesta y el estudio realizado se materializan los objetivos planteados al inicio de esta investigación: Desarrollar una reingeniería a un sistema para la evaluación teórica.

## Recomendaciones

Al finalizar este trabajo quedan algunas recomendaciones que pueden servir de punto de partida para mejorar aún más el sistema obtenido:

- Continuar desarrollando el resto de las aplicaciones que conforman el sistema, seguir el ciclo de desarrollo del producto para obtener el módulo de administración del sistema.
- Refinar el diseño del sistema obtenido, con el objetivo de incorporar nuevas funcionalidades al mismo y obtener un sistema más general y completo.
- Realizar la migración del sistema a software libre, para disminuir los costos de producción y poder comercializarlo con mayores ganancias.
- Estudiar la posibilidad de incluir otros tipos de preguntas en los exámenes, donde se pueda responder con texto redactado por el cliente que se evalúa.



## Referencias bibliográficas

- [**Carmona 2004**]: Carmona, N. J. F. (2004 ). "Reconstrucción de la arquitectura: Una actividad de la reingeniería de software". León, Instituto Tecnológico de León. Retrieved 07/02/2007, from <http://www.monografias.com/trabajos17/reingenieria-software/reingenieria-software.shtml#reing>.
- [**Rational**]: Corporation, R. (2004) "Lo nuevo de Rational Rose 2000". Retrieved 21/01/2007, from <http://www.abists.com.mf/Fabs/Rational/notasTK>
- [**Histchfeld**]: Histchfeld, P. S. C. y. N. "Tutorial de UML." Retrieved 02/11/2007.
- [**Álvarez 2004**]: Juan Carlos Álvarez García, M. M. S., María N. Moreno García (2004). "Metodología de Reingeniería del Software para la remodelación de aplicaciones científicas heredadas", Universidad de Salamanca: 22.
- [**Caro 2004**]: Patricio Salinas Caro, N. H. K. (2004). "Unified Modeling Language." Retrieved 17/02/2007, from [www.dcc.uchile.cl/~psalinas/uml](http://www.dcc.uchile.cl/~psalinas/uml).
- [**Schumacher**]: Robin Schumacher, Arjen Lentz. "Dispelling the Myths". Retrieved 21/01/2007, from <http://dev.mysql.com/tech-resources/articles/dispelling-the-myths.html>.
- [**Sin\_autor**]: Sin\_autor "Examen de conducir - teórico y práctico" Retrieved 23/01/2007, from [http://www.vv.se/templates/page3\\_5517.aspx](http://www.vv.se/templates/page3_5517.aspx)
- [**Wikipedia 1**]: Wikipedia "Firebird " Retrieved 22/01/2007, from <http://es.wikipedia.org/wiki/Firebird>
- [**Wikipedia 2**]: Wikipedia "Sistema de gestión de base de datos" Retrieved 22/01/2007, from <http://es.wikipedia.org/wiki/Firebird>
- [**Wikipedia 3**]: Wikipedia "XML" Retrieved 22/01/2007, from <http://es.wikipedia.org/wiki/XML>

## Glosario de términos

**Actor:** Entidad externa que inicia una actividad en el sistema (actor primario) o que interactúa en cualquier actividad (actores secundarios).

**Arquitectura:** Conjunto de elementos estructurales significativas acerca de la organización de un sistema software, la selección de los elementos estructurales que representan el sistema, y las interfaces entre ellos, junto con su comportamiento, tal y como se especifica en las colaboraciones entre esos elementos, la composición de estos elementos estructurales y de comportamiento en subsistemas progresivamente mayores, y al estilo arquitectónico que guía esta organización: estos elementos y sus interfaces, sus colaboraciones y su composición.

**Clase:** Elemento de software que describe o caracteriza una entidad del mundo real o un entidad del espacio de implementación.

- **CC- Clase controladora**
- **CE- Clase entidad**
- **CI- Clase interfaz**

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

**Diagrama de Clases:** Muestra un conjunto de clases y las relaciones entre ellas. Muestra el diseño de un sistema desde un punto de vista estático

**Diagrama de Casos de Usos del Sistema:** Modelo gráfico que representa a todos los actores que interactúan con el sistema.

**Examen teórico:** Proceso de evaluación mediante el cual se comprueban los conocimientos adquiridos en una materia determinada.

**Evento:** Cambio en el estado de un objeto

**Framework:** En el desarrollo de software, un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros softwares para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

**Modelación del Negocio:** Es una abstracción semánticamente cerrada de un sistema, que especifica el sistema modelado desde un cierto punto de vista y en un determinado nivel de abstracción. Es una vista auto contenida en el sentido de que los desarrolladores tengan una sola interpretación de lo que ocurrirá en el sistema cuando se dispare un evento descrito en el modelo. Abstracción de la realidad bajo estudio. Un modelo de software específico muestra una de las perspectivas sobre la cual un ingeniero puede desarrollar el software.

**Rational Rose (RUP):** Rational Unified Process (Proceso Unificado de Rational), es un proceso de desarrollo de software para planear, construir y administrar el desarrollo de soluciones de negocio.

**Requisito:** Condición o capacidad, necesidad o deseo que debe cumplir un sistema.

**Requisito Funcional:** Requisito que especifica una acción que debe ser capaz de realizar el sistema, sin considerar restricciones físicas. Requisito que especifica comportamiento de entrada/salida de un sistema.

**Requisito no Funcional:** Requisito que especifica propiedades del sistema, como restricciones del entorno o de implementación, rendimiento, dependencias de la plataforma, mantenibilidad, extensibilidad o fiabilidad. Requisito que especifica restricciones físicas sobre un requisito funcional.

**SIMPRO:** Empresa cubana dedicada al desarrollo de simuladores de realidad virtual.