

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 7



*Trabajo de Diploma para Optar por el Título de
Ingeniero en Ciencias Informáticas*

**Herramienta de apoyo al proceso de caracterización
psicológica para la entrada a proyectos informáticos**

Autores:

Leticia Valdivia Mena

Yoandri Torres Almarales

Tutores:

Msc. Yadira Palenzuela Fundora

Ing. Yenny Leal González

Ciudad de La Habana, Junio de 2010

“Año 52 de la Revolución”

Datos de Contacto

Msc. Yadira Palenzuela Fundora

Máster en Psicología Educativa 2008, Graduado de la Universidad de la Habana, Facultad de Psicología, 2004. Profesor Asistente de la UCI. Ha impartido asignaturas en nivel de pregrado y postgrado en la UCI. Es miembro de la Sociedad de Psicólogos de Cuba. Actualmente se desempeña como psicóloga organizacional en la empresa Albet Ingeniería y Sistemas como especialista del Grupo de Mercadotecnia y con funciones dentro de la dirección de Desarrollo Organizacional, aplicando los programas de caracterización y participando en las estrategias de selección y capacitación de entrada a proyectos. Su email es ypalenzuela@uci.cu, ypalenzuela@albet.cu.

Ing. Yenny Leal González

Graduada de la Universidad de las Ciencias Informáticas 2008, Facultad 7. Ha impartido pregrado en la UCI, asignaturas del segundo perfil de Informática para la Salud. Es Instructora Recién Graduada de la UCI. Actualmente se desempeña como especialista de mercado en la empresa Albet Ingeniería y Sistemas como miembro del Grupo de Mercadotecnia. Su email es ylgonzalez@uci.cu, ylgonzalez@albet.uci.cu

Resumen

Para la entrada de especialistas a un proyecto informático se realiza el proceso de caracterización psicológica llevado a cabo por la empresa Albet Ingeniería y Sistemas. Los test psicológicos son ejecutados mediante la herramienta Microsoft Excel, que se instala localmente en cada una de las computadoras donde se realizarán. La herramienta usada no permite tabular u obtener los resultados de forma dinámica, por tanto procesar la caracterización se torna engorroso para los profesionales que aplican los test. Por estas razones el objetivo del presente trabajo es desarrollar el análisis, diseño e implementación de una Aplicación para la gestión dinámica de los test psicológicos que den como resultado una caracterización psicológica.

Para el desarrollo de la herramienta se emplean: el lenguaje de programación PHP, el Framework CodeIgniter que utiliza la arquitectura Modelo-Vista-Controlador (MVC), el Sistema Gestor de Bases de Datos MySQL y la metodología de Desarrollo el Proceso Unificado de Desarrollo (RUP). Además; se hace uso del Lenguaje Unificado de Modelado (UML) y Visual Paradigm for UML 2.0 para la creación de los artefactos que se generan durante el ciclo de vida del software.

Esta herramienta permitirá a la empresa contar con un proceso de caracterización psicológica de mayor calidad, facilitando la aplicación de los test, agilizando el proceso de obtención de resultados y garantizando la seguridad de la información contenida.

Palabras Claves: *Proceso de caracterización psicológica, test psicológicos.*

ÍNDICE

Introducción	1
Capítulo I. Fundamentación Teórica	5
1.1 Test Psicológicos	5
1.2.2 Técnicas.....	7
1.2 Estado del Arte	9
1.3 Tendencias y Tecnologías actuales.....	14
1.3.1 Metodologías de desarrollo de software	14
1.3.2 Lenguaje Unificado de Modelado (UML).....	16
1.3.3 Herramienta CASE para el desarrollo de software	16
1.3.4 Sistema gestor de base de datos.....	17
1.3.5 Internet	20
1.3.6 Tecnologías para crear aplicaciones Web	20
1.3.7 Arquitectura.....	24
1.3.8 Herramientas a utilizar.	25
Capítulo 2. Características del Sistema	27
2.1 Objeto de estudio.....	27
2.1.1 Procedimiento para la Caracterización Psicológica	28
2.2 Modelo del Negocio	29
2.2.1 Actores del negocio.....	30
2.2.2. Trabajadores del negocio.....	30
2.2.3. Diagrama de casos de uso del negocio	31
2.2.4. Especificación de los casos de uso del Negocio.	31
2.2.5. Diagrama de actividades.....	32
2.2.6. Diagrama de clases del modelo de objetos.	32
2.3 Especificación de los requisitos de software.....	33
2.3.1 Requerimientos Funcionales.....	33
2.3.2 Requerimientos No Funcionales.	34
2.4 Descripción del Sistema propuesto.....	36
2.4.1 Modelo de Casos de Uso del Sistema	36

2.4.2 Listado de Casos de Usos (CU).....	37
2.4.3 Diagrama de casos de uso del sistema.	39
2.5 Expansión de los Casos de Usos	39
Capítulo 3. Análisis y Diseño del Sistema	49
3.1 Arquitectura.....	49
3.2 Modelo de Análisis.....	50
3.2.1 Diagrama de Clases del Análisis	50
3.3 Modelo de Diseño	53
3.3.1 Estereotipos Web.....	53
3.3.2 Diagrama de Clases del Diseño.....	54
3.3.3 Diagrama de Interacción (Secuencia).....	55
3.4 Descripción de las Clases del Diseño.....	55
3.4.1 Descripción de Clases Controladoras.....	55
3.4.2 Descripción de las Clases Entidades.....	59
Capítulo 4. Implementación	63
4.1 Modelo de datos	63
4.1.1 Descripción de las tablas	64
4.2 Modelo de Implementación.....	66
4.2.1 Diagrama de Despliegue.....	66
4.2.2 Diagrama de Componentes	66
4.3 Tratamiento de errores	70
4.4 Seguridad.....	71
4.5 Estrategias de Codificación. Estándares y estilos a utilizar.....	71
Conclusiones	75
Recomendaciones	76
Referencias Bibliográficas	77
Bibliografía	79
Glosario de Términos	81
Anexos	84
Anexo 1. Diagrama de Clases del Diseño	84
Anexo 2. Diagrama de Secuencia.....	86

Introducción

Las Tecnologías de la Información y las Comunicaciones (TIC) constituyen el hilo conductor de los grandes procesos de cambios de la sociedad actual, encaminada a llamarse Sociedad de la Información. El impacto de las TIC en todos los ámbitos de la vida, hace cada vez más difícil que se pueda actuar eficientemente prescindiendo de ellas, debido a que han impuesto un ritmo acelerado en el desarrollo del mundo. La sociedad ha ido adoptando los conocimientos y las tendencias actuales del desarrollo en la vida diaria, siendo estos necesarios para la productividad y el bienestar.

En Cuba, las empresas realizan grandes cambios dados por la introducción de los avances tecnológicos en busca de una mayor eficiencia y en la omisión a problemas en el desarrollo de software existentes en la actualidad, como parte del perfeccionamiento empresarial. En pos de automatizar los procesos productivos, se requiere de un software con calidad y que a su vez se ajuste a las necesidades del cliente.

En la actualidad existen medios suficientemente poderosos para agilizar el proceso de recolección de datos y la obtención inmediata de resultados. Aún cuando existen innumerables maneras de digitalizar estos procesos, sigue siendo una necesidad recurrente disponer de un mecanismo suficientemente confiable que permita crear formatos con estructuras de datos ajustables, en el menor tiempo, de acuerdo a la dinámica que la vida cotidiana que exige el desarrollo de la sociedad cubana.

El Ministerio de la Informática y las Comunicaciones de Cuba (MIC) es el organismo rector de estas disciplinas en Cuba cuya misión es la de impulsar, facilitar y ordenar el uso masivo de servicios y productos de las Tecnologías de la Información, las Comunicaciones, la Electrónica y la Automatización para satisfacer las expectativas de todas las esferas de la sociedad.

Debido al avance de las tecnologías la empresa Albet Ingeniería y Sistemas realiza transformaciones para aprovechar sus potencialidades. El origen y desarrollo de esta empresa se vincula estrechamente a la Universidad de Ciencias Informáticas (UCI), que con su modelo productivo agrupa más de doce mil profesionales, técnicos y estudiantes. Albet posee los derechos comerciales de todos los productos y servicios que desarrolla la UCI y mediante la alianza con otras prestigiosas entidades ofrece soluciones integrales en la esfera de las tecnologías de la información y las comunicaciones. [1]

Introducción

Las soluciones integrales de Albet contienen desde diseño o transformación organizacional, consultorías gerenciales, consultorías y servicios informáticos. Así como, desarrollo de software a la medida, desarrollo de estrategias de comunicación y campañas comunicacionales, suministro de equipamiento, soluciones de seguridad integral, capacitación y soporte técnico. Especialistas de diferentes instituciones del país, formando un equipo Cuba, gestionan, organizan y dirigen las soluciones conformado por varios proyectos, en dependencia de lo que se contrate. La selección de los especialistas para el desempeño de los diferentes roles en un proyecto, se realiza apoyándose en test psicológicos que lo caracterizan y muestran la capacidad de desempeñar un determinado rol en uno de los proyectos.

La caracterización psicológica de un especialista se realiza actualmente mediante herramientas y técnicas aportadas por el Centro Coordinador para la Formación y el Desarrollo (FORDES), rector de la actividad de Formación y Desarrollo en el MIC. Además de la utilización de otras técnicas e instrumentos confeccionados por especialistas en psicología que tributan a enriquecer la propuesta inicial de caracterización psicológica ofrecida por FORDES, para dotar la evaluación de indicadores que tributen a una correcta selección basada en competencias para los diferentes roles que existen en la gestión de un proyecto informático.

Actualmente la mayoría de los test aplicados por la empresa a los especialistas son ejecutados mediante la herramienta Microsoft Office Excel, más conocido como Microsoft Excel, los cuales se instalan localmente en cada una de las computadoras donde se realizarán. Esto significa un alto esfuerzo y nivel de organización para garantizar los laboratorios y recursos indispensables para efectuarla con éxito. La tecnología usada para aplicar los test no cuenta con el proceso de tabular u obtener la caracterización de forma dinámica. Para obtener los resultados de manera colectiva es necesario ubicar los ficheros de respuesta de cada computadora, en un único ordenador y en las carpetas designadas previamente. Por lo anteriormente expuesto este proceso se torna dificultoso para los profesionales que aplican los test.

Por tal motivo es necesario renovar la tecnología informática utilizada y dotar a la empresa de una herramienta que facilite la aplicación de los test, eliminando estos factores que dificultan y alargan el proceso de caracterización.

Luego de hacer un análisis de la **situación problemática** existente se plantea como **problema científico**: ¿Cómo facilitar la gestión de los test psicológicos aplicados a los especialistas para la entrada a proyectos

de forma que permita su creación, respuesta y obtención de resultados dinámicamente para dar una caracterización psicológica?, dando lugar al **objeto de estudio**: Proceso de Gestión para la caracterización psicológica de especialistas, el cual se enmarca en el **campo de acción**: Proceso de Gestión para la caracterización psicológica de especialistas en la empresa Albet Ingeniería y Sistemas.

Con el propósito de dar solución al problema planteado, se define como **objetivo general**: Desarrollar el análisis, diseño e implementación de una Aplicación para la gestión dinámica de los test psicológicos que den como resultado una caracterización psicológica. En correspondencia con el mismo, se proponen los siguientes **objetivos específicos**:

- ❖ Realizar un estudio sobre formas y mecanismos de test psicológicos proponiendo uno para desarrollar la aplicación.
- ❖ Diseñar una aplicación que soporte un procesamiento de test óptimo.
- ❖ Diseñar una base de datos para gestionar la información de los test y las caracterizaciones psicológicas de forma que garantice la integridad de la información contenida.
- ❖ Implementar una aplicación que soporte un procesamiento de test óptimo.

Para el avance de la investigación y el cumplimiento del objetivo presentado se establecen las siguientes

Tareas de Investigación:

- ✓ Evaluar los sistemas utilizados en Cuba y en el mundo para apoyar el proceso de caracterización psicológica.
- ✓ Valorar las herramientas y tecnologías existentes para este tipo de aplicación.
- ✓ Fundamentar las técnicas para la captura de requisitos y patrones de casos de uso a utilizar.
- ✓ Realizar el modelo de negocio y el levantamiento de requisitos del sistema representado en un lenguaje natural.
- ✓ Realizar el modelo de análisis y diseño de la aplicación.

- ✓ Modelar el flujo de trabajo “Implementación”.
- ✓ Implementar la aplicación propuesta.

El documento está organizado en cuatro capítulos, en los que se fundamenta la solución propuesta, se hace referencia al estudio de las tecnologías, las herramientas utilizadas para el desarrollo de la aplicación, así como el modelado del sistema, el análisis y diseño del sistema y la implementación.

Capítulo I. Fundamentación Teórica.

En este capítulo se hace una breve descripción del estudio para llevar a cabo la investigación sobre el desarrollo de test psicológicos en Cuba y el resto de los países del mundo. Se realizará, además, un estudio de las principales tecnologías y herramientas a utilizar en el desarrollo de la aplicación Web.

Capítulo II. Características del sistema.

En este capítulo se describe las características básicas y fundamentales del sistema informático a desarrollar. En el mismo, se realiza una breve descripción del problema en cuestión, especificándose los requerimientos funcionales que plantea el cliente y que a su vez el sistema debe cumplir. Se da a conocer todo lo referente al modelado del negocio, incluyendo casos de usos del sistema y una breve descripción de cada uno de ellos.

Capítulo III. Análisis y Diseño del sistema.

Este capítulo abarca el desarrollo de la fase de análisis y diseño del sistema, englobando todo lo referente al modelado del diseño específicamente. En el mismo figuran los diagramas de clases del diseño de los casos de uso definidos en el capítulo anterior, así como los diagramas de secuencia correspondientes a dichos casos de uso. Además se representa el modelo de datos con la descripción de sus tablas, el diagrama de clases persistentes y el modelo de despliegue del sistema.

Capítulo IV. Implementación del sistema.

Este capítulo abarca detalles de la fase de elaboración para la implementación del sistema. Se realiza el modelo de implementación con sus respectivos diagramas de componentes pertenecientes a los casos de uso definidos en las iteraciones anteriores.

Capítulo 1. Fundamentación Teórica

Capítulo I. Fundamentación Teórica

El presente capítulo es el resultado de una profunda investigación en el que se exponen los temas relacionados con el objeto de estudio para mayor comprensión del mismo. Se describen los principales conceptos asociados al proceso de caracterización psicológica, necesarios para entender la propuesta de solución. Se hace un análisis de las herramientas, tecnologías, lenguajes y metodologías que pueden ser utilizadas, además se definen las que se utilizan de acuerdo con sus características, ventajas, recursos disponibles y necesidades existentes.

1.1 Test Psicológicos

Un test psicológico es una medida objetiva y normalizada de una muestra de conducta. Son instrumentos especializados que sirven de apoyo al psicólogo al medir conductas específicas. [2]

Clasificación general de los test psicológicos:

La mayor clasificación de los test se da en dos grandes vertientes: [3]

- **Test Psicométricos:** Son test de tipo cuantitativo, ya que miden capacidades, aptitudes, rasgos de personalidad. La selección y formulación de sus diferentes ítems se realiza mediante criterios de validación objetiva. Uno de sus inconvenientes, es que el sujeto puede no responder con sinceridad de forma intencionada, pero suelen presentar alguna escala para controlar ésta. Suelen ser de tipo analítico, de forma que pueden dar una idea del predominio de unas cualidades sobre otras en la personalidad del sujeto.
- **Test Projectivos:** Se caracteriza porque el sujeto examinado proyecta inconscientemente en las respuestas, sus propios rasgos de personalidad (afectos, pulsiones, conflictos pasados y presentes, etc.). Están menos sujetos que los psicométricos a la deformación voluntaria de las respuestas, sin embargo, se les atribuye por parte de sus detractores, menor objetividad, ya que pueden verse afectados por la subjetividad del examinador.

Capítulo 1. Fundamentación Teórica

Los test psicológicos se utilizan con frecuencia para evaluar la idoneidad de un candidato para un puesto de trabajo o para un ascenso. Los psicólogos son los encargados de elaborar, estandarizar y aplicar estos test.

Para que un test sea adecuado tiene que cumplir una serie de requisitos; es decir, ha de ser objetivo, estandarizado, fiable y válido.

Estandarización

Constancia o uniformidad de las condiciones y procedimientos para la administración de un instrumento de evaluación.

Objetividad

La objetividad se refiere principalmente a los resultados del test. La puntuación obtenida en el test o en cada una de las escalas que lo compongan, debe estar libre del juicio del examinador, de manera que una especialista obtenga la misma puntuación independientemente de quién sea la especialista que lo evalúe.

Fiabilidad

La fiabilidad hace referencia a la consistencia de las respuestas a un test. Es decir, si una especialista responde a un test y obtiene una puntuación determinada, debe volver a obtener una puntuación similar si responde al mismo test una semana después. Es normal encontrar una pequeña variación en sus puntuaciones, pero si la diferencia es grande, se considera que el test no es fiable.

Validez

Grado en que un instrumento de evaluación mide lo que pretende medir.

1.2 Proceso de caracterización psicológica

Este proceso consiste en la obtención de atributos competenciales de las especialistas: capacidad intelectual, capacidad de aprendizaje, capacidad de innovación, roles predominantes en el trabajo en equipo y liderazgo. Además de la profundización de las características afines de los especialistas con los perfiles de Investigador y Líder de proyectos, destacando en ambos casos las potencialidades que existan como reservas de cuadros y haciendo mención al área de conocimientos que representa: técnicos, psicosociales o empresariales, para una mayor precisión y certeza en la caracterización.

Capítulo 1. Fundamentación Teórica

1.2.1 Objetivos y Resultados esperados de los test psicológicos:

1.2.1.1 Objetivos:

1. Caracterizar psicológicamente a los especialistas aspirantes a entrar en proyectos, en aspectos tales como capacidad intelectual, capacidad de aprendizaje, capacidad de innovación, liderazgo y roles predominantes en el trabajo en equipo.
2. Identificar especialistas con potencialidades para desempeñarse en los roles líder de proyecto, Investigador y Reserva de Cuadros en áreas Técnicas, Psicosociales y Empresariales.

1.2.1.2 Resultados:

De la implementación adecuada del proceso puede esperarse:

- La obtención de información valiosa y detallada, útil para la toma de decisiones sobre la participación de estos especialistas en distintas tareas.
- Informe individual de caracterización psicológica de los especialistas a los conocimientos que poseen.

1.2.2 Técnicas

Las técnicas seleccionadas para la caracterización psicológica para la entrada a proyectos informáticos pueden describirse del siguiente modo: [4]

Las Leyes:

Es un test que permite revelar algunas particularidades del proceso del pensamiento de la especialista: capacidad para analizar, habilidad para hacer construcciones lógicas, facilidad para establecer enlaces asociativos y estabilidad en la atención. La técnica sugiere la capacidad intelectual de la especialista. Es una prueba para ser administrada en 10 minutos.

Capítulo 1. Fundamentación Teórica

Inventario de estilos decisionales:

Es una técnica que refleja el modo habitual que las especialistas toman decisiones. A través de la combinación de las dimensiones orientación de valores (centrismo en lo social, lo técnico, el trabajo o las especialistas cuando se toman decisiones) y tolerancia a la ambigüedad (grado en que las especialistas necesitan del control y estructuración de sus espacios vitales y son propensas a encontrar las situaciones psicológicamente incómodas) se obtienen 4 estilos decisionales que son: Directivo, Analítico, Conceptual y Conductual. Es una prueba que puede sugerir aptitudes para el liderazgo.

Test de autopercepción de Belbin:

Técnica para evaluar esencialmente habilidades de trabajo en equipo. Identifica ocho tipos de trabajadores en equipo, que son: TC trabajador de la compañía, PR presidente, FO formador, IN introductor, IR investigador de recursos, ME monitor evaluador, TE trabajador de equipo y AP acabador perfeccionador. Además permite identificar aptitudes para el liderazgo.

Cuestionario de estilos cognitivos:

Esta es una técnica que ofrece una valoración sobre los modos en que las especialistas procesan la información que perciben, si son más sensoriales o intuitivos, sentimentales o reflexivos. En este sentido se brindan cuatro estilos resultantes de la combinación de estos procesos. La prueba sugiere tendencias del sujeto hacia la proyección futura, la ingeniosidad, creatividad, pragmatismo y formas particulares de tomar decisiones.

Matriz de Fortalezas y Debilidades:

En esta prueba se le presenta al sujeto un conjunto de atributos personales que él deberá evaluar en sí mismo como fortalezas o debilidades. Permite tomar el referente del propio sujeto y realizar un cruce con los resultados de otras técnicas aplicadas.

Capítulo 1. Fundamentación Teórica

1.2 Estado del Arte

Ámbito internacional:

En las búsquedas realizadas a nivel internacional se ha constatado la existencia de diversos sistemas utilizados para apoyar el proceso de caracterización psicológica realizando la selección de personal basada en competencias.

Psycowin:

Es la aplicación líder en el mercado que apoya funciones de Recursos Humanos como: Reclutamiento y Selección, Capacitación, Desarrollo Organizacional, Planes de Sucesión, Compensaciones. Desarrollado por la empresa mexicana PSW Global Solutions®. Permite evaluar al personal a través de Pruebas Psicométricas Automatizadas en forma modular e interactiva.

Con esta herramienta se puede construir modelos de evaluación enfocados a competencias, basados en la misión de la organización y también se puede estandarizar perfiles de puesto con base en los lineamientos corporativos.

Además Psycowin permite homologar fácilmente los criterios de evaluación, diagnosticar áreas de oportunidad y competencias sobresalientes, e implementar planes de desarrollo con base en el diagnóstico. [5]

Características Principales:

- Ambiente Web y cliente servidor.
- 14 pruebas psicométricas.
- Integración de datos curriculares, fotografía y resumen del proceso de incorporación.
- Inventario de 16 perfiles de puesto clásico y 44 perfiles organizacionales.
- Modelo genérico de 27 competencias.
- Modelo de administración del talento.
- Reportes estadísticos de psicometría.
- Reporte gráfico comparativo.
- Aplicación electrónica de pruebas.

Capítulo 1. Fundamentación Teórica

Este sistema a pesar de las funcionalidades que presenta, no cubre las necesidades actuales de informatización del proceso de caracterización psicológica llevado a cabo por Albet; pues entre los test psicométricos empleados para caracterizar al personal no figura ninguno de los utilizados por Albet para evaluar psicológicamente al personal. Además de que Albet solo necesita un modelo genérico para los atributos competenciales: capacidad intelectual, capacidad de aprendizaje, capacidad e innovación, roles predominantes en el trabajo en equipo y liderazgo; mientras que el sistema Pscowin ofrece un modelo genérico de 27 competencias. Y la funcionalidad característica del sistema de integración de datos curriculares, fotografía y resumen de proceso de incorporación no cubre ninguna de las funcionalidades requeridas para satisfacer las necesidades de informatización.

Belbin:

Belbin Associates es una organización liderada por el Dr. Meredith Belbin, experto reconocido a nivel mundial en todos los temas relacionados con el trabajo en equipo. El Dr. Belbin y su equipo han desarrollado diversas herramientas que facilitan la aplicación de la Metodología Belbin de roles de equipo, metodología que surge como resultado de una larga y minuciosa investigación llevada a cabo en el Henley Management College, Reino Unido, entre los años 1971 y 1979 y reconocida mundialmente como una metodología clave en la gestión de equipos de trabajo. [6]

BELBIN® se constituyó en Cambridge, Reino Unido, en 1988 y actualmente cuenta con delegaciones en prácticamente todo el mundo, lo que permite que tanto la metodología Belbin como las herramientas creadas para su aplicación estén disponibles en numerosos idiomas. Belbin Associates trabaja con veinte de las cien primeras empresas del mundo y en sectores tan variados como: instituciones públicas, telecomunicaciones, gran consumo, banca, seguros, distribución, etc. [7]

Herramientas desarrolladas por Belbin Associates:

➤ *Licencia Be e-interplace*

e-interplace es un software de última generación que permite la corrección de los distintos cuestionarios Belbin, produciendo una serie de informes individuales, de equipo y de puestos, basados todos ellos en los roles de equipo (comportamiento laboral). Teniendo en cuenta que las especialistas constituyen el valor más importante de una organización, es crucial que en el contexto laboral su organización sea

Capítulo 1. Fundamentación Teórica

óptima con el fin de obtener el mejor resultado posible. Las especialistas han de desempeñar el trabajo adecuado, colaborar con los compañeros apropiados y comportarse de acuerdo con sus capacidades más idóneas. [8]

La herramienta integra datos sobre especialistas y puestos de trabajo, los cuáles se obtienen a través de unos cuestionarios estandarizados que incluyen un inventario de autopercepción, valoraciones de evaluadores, valoraciones de los requisitos del puesto y evaluaciones del puesto.

Debido a que es principalmente una herramienta de asesoramiento y orientación para ayudar a las especialistas y a los equipos a comprenderse mejor a sí mismos y a los demás, parte del proceso implica conocimiento personal. El punto fuerte de e-interplace radica en que sus informes se basan en un consenso del comportamiento observado. La finalidad de las valoraciones de los evaluadores es complementar la autopercepción del candidato con un punto de vista objetivo. Estas valoraciones aportan información 360º sobre los roles de equipo del individuo. Este cuestionario lo deberán completar especialistas que trabajan o han trabajado recientemente con el candidato. La herramienta requiere como mínimo cuatro evaluadores. Los evaluadores pueden ser colegas, colaboradores o jefes que conocen bien al candidato desde el punto de vista laboral. [9]

La empresa Belbin Associates solicita por la licencia de esta herramienta el pago de 2.950,00 euros por una versión limitada de 100 pasos y 4.450,00 euros por una versión limitada de 200 pasos.

➤ *Belbin online*

Belbin online es un inventario de autopercepción y hasta 6 valoraciones de los evaluadores para particulares y equipos pequeños, que constituye la manera más sencilla y eficaz de obtener perfiles de Roles de Equipo Belbin. Los perfiles se obtienen por correo electrónico luego de haber completado el(los) cuestionario(s) vía Web y haber pagado su costo.

Belbin Associates solicita el pago de 35,00 euros por un informe de roles de equipo para particulares; y para equipos pequeños 35,00 euros por especialista para un informe completo.

A pesar de que las herramientas desarrolladas por la organización Belbin Associates presentan muchas funcionalidades, no cubren las necesidades actuales de informatización del proceso de caracterización psicológica llevado a cabo por Albet; pues entre los test psicométricos utilizados en estas herramientas

Capítulo 1. Fundamentación Teórica

figura sólo el Test de Autopercepción de Belbin empleado por Albet en el proceso de caracterización. Además de que la organización solita un pago por el uso de las herramientas desarrolladas por la misma.

Hum&Select

El sistema Hum&Select es un paquete computacional especialmente diseñado para satisfacer las necesidades del proceso de Reclutamiento, Selección y Evaluación Psicométrica de Personal en las Empresas. [10]

El Hum&Select está diseñado para trabajar en Red Local o vía Internet. Al utilizar el Hum&Select se puede aplicar, evaluar, graficar e interpretar las pruebas psicométricas utilizadas rápidamente, ya que la aplicación de los psicométricos también se puede realizar de manera interactiva vía Web y los reportes se generan en formato HTML, lo que significa que se podrán ver o imprimir, enviarlos inmediatamente por correo electrónico o subirlos a la página de Inter/Intranet de la organización.

Este sistema fue desarrollado por HUM&SOFTWARE, empresa especializada en el desarrollo de sistemas de información para Recursos Humanos, el cuál solicita por la versión completa el pago de \$2,500.00 USD para adquirir la licencia monousuario para la instalación en 1 computadora, incluyendo 8 meses gratuitos de los Servicios Web y 8 meses de la Póliza de Actualización Tecnológica; \$400.00 USD por año para el autoregistro de información curricular y aplicación de las pruebas de manera interactiva vía Internet y \$250.00 USD por año para la póliza de actualización tecnológica.

A pesar de las funcionalidades que presenta el sistema Hum&Select, no cubre las necesidades actuales de informatización del proceso de caracterización psicológica llevado a cabo por Albet; pues entre los test psicométricos empleados para caracterizar al personal no figura ninguno de los utilizados por Albet para evaluar psicológicamente al personal. Además de que la empresa solita un pago por el uso de las herramientas desarrolladas por la misma.

Capítulo 1. Fundamentación Teórica

Ámbito nacional:

En Cuba existen sólo dos herramientas que emplean test psicológicos en el proceso de caracterización psicológica para la selección de personal.

Selsoft

Aplicación para la selección y evaluación de los Recursos Humanos desarrollada en 1998 por la Empresa de Desarrollo de Software para la gestión empresarial en Cuba, Desoft. Producto diseñado para la evaluación integral de los requisitos exigidos por el perfil psicológico en el proceso de selección de personal y evaluación de los puestos de trabajo. La evaluación psicológica se realiza mediante la utilización de un grupo de test psicométricos de personalidad, aptitud, habilidad, inteligencia y preferencias vocacionales, que permite crear el expediente del candidato con la hoja de vida y los resultados de los test aplicados facilitando la evaluación integral del mismo. [11]

Selsoft fue desarrollado utilizando el Lenguaje *Visual Basic Versión 3.0* y Bases de Datos en *Access 2.0*, soportados sobre Medio Ambiente *Windows 3.x* ó *Windows 95*.

La empresa Desoft solicita el pago de \$ 541.53 MN.

A pesar de las funcionalidades que presenta el sistema *Selsoft*, no cubre las necesidades actuales de informatización del proceso de caracterización psicológica llevado a cabo por Albet; pues entre los test psicométricos empleados para caracterizar al personal no figura ninguno de los utilizados por Albet para evaluar psicológicamente al personal. Además de que es una herramienta sin conectividad que no garantiza la agilidad de la aplicación de los test eliminando los factores que dificultan el proceso de caracterización psicológica.

Programa de Caracterización Psicología. Sistema MIC.

En el MIC está establecido un programa para la caracterización psicológica a especialistas usando técnicas y herramientas aportadas por el FORDES. La primera edición de este programa caracterizó a los recién graduados del año 2007 que ingresaron al MIC, lográndose la evaluación psicológica de 258 jóvenes. Este proceso se ha continuado realizando cada año con los recién graduados de nuevo ingreso al MIC. El programa es ejecutado mediante la herramienta Microsoft Office Excel, que permite tabular las

Capítulo 1. Fundamentación Teórica

respuestas de los especialistas agrupando dicho resultado en una base de datos por provincias apreciándose la caracterización del mismo.

Todas las empresas asociadas al MIC emplean este programa para la entrada de recién graduados a las mismas como son: Copextel, Etecsa, RadioCuba, Gedeme, FORDES, etc. En la universidad la empresa Albet S.A. emplea algunos de los test psicométricos desarrollados en este programa para caracterizar a los especialistas de nuevo ingreso a un proyecto informático.

A pesar de las funcionalidades que presenta este programa, tiene algunas limitaciones que provocan la necesidad de este trabajo, pues es una herramienta sin conectividad y no cuenta con el proceso de tabular u obtener los resultados de forma dinámica, necesario para agilizar el proceso de caracterización psicológica.

1.3 Tendencias y Tecnologías actuales

1.3.1 Metodologías de desarrollo de software

Durante los últimos años se han desarrollado dos corrientes en lo referente a las metodologías de desarrollo de software, las llamadas “pesadas o tradicionales” y las llamadas “ligeras o ágiles”. Las primeras se basan en la idea de conseguir el objetivo común por medio de orden y documentación. Estas se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir y las herramientas y notaciones que se usaran. Estas propuestas han demostrado ser más efectivas y necesarias en proyectos de gran envergadura.

Mientras que las segundas tratan de lograrlo por medio de la comunicación directa e inmediata entre aquellos que intervienen en el proceso. Además que brindan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Esta propuesta está mostrando su efectividad en proyectos con requisitos muy cambiantes o poco claros cuando se exige reducir los tiempos de desarrollo manteniendo una alta calidad. Son especialmente orientadas para proyectos pequeños.

1.3.1.1 Rational Unified Process (RUP)

Capítulo 1. Fundamentación Teórica

El Proceso Unificado de Rational (RUP) es una propuesta de proceso para el desarrollo de software orientado a objetos que utiliza Unified Model Language (UML) para describir todo el proceso. Es una metodología basada en componentes o sea que el sistema de software en construcción está formado por componentes de software interconectados a través de interfaces bien definidas. Se basa en casos de uso para describir lo que se espera del software y está muy orientado a la arquitectura del sistema, documentándose lo mejor posible. Constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Tiene como características distintivas estar **Dirigido por Casos de Uso**, **Centrado en Arquitectura** y ser **Iterativo e Incremental**.

✓ **Dirigido por Casos de Uso.**

Los Casos de Uso son los encargados de dirigir el proceso de software, ellos representan pedazos de funcionalidades que en conjunto le dan vida al producto final, capturan los requisitos necesarios permitiendo que cliente y programadores lleguen a un acuerdo para definir la realización del producto. Los Casos de Uso enlazan cada una de las partes que conforman el producto a realizar.

✓ **Centrado en Arquitectura.**

La Arquitectura en el software es importante para tomar decisiones como: organización que tendrá el software, los elementos estructurales, el comportamiento y estilo que guiará la organización prevista. Con una Arquitectura bien definida se puede comprender mejor el sistema que se pretenda conformar, organizar su desarrollo, qué evoluciones podría tener en una próxima versión, así como, permitir reutilización de código existente.

✓ **Iterativo e Incremental.**

Esta característica se refiere a que el proyecto completo se subdivide en mini-proyectos (iterativo) y que al ir avanzando se van haciendo varias versiones que incorporan cambios y arreglos (incremental).

La vida de un producto de Software según RUP:

Capítulo 1. Fundamentación Teórica

El Proceso Unificado establece que desde el nacimiento de un sistema de software hasta su muerte, suceden en el tiempo un conjunto de ciclos, cada uno de los cuales concluye con la entrega de una versión del sistema. Cada ciclo se divide en cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada fase según la complejidad del sistema se divide en iteraciones, termina con un hito, en el cual los directivos, basados en el conjunto de artefactos obtenidos, toman una serie de decisiones para la continuación o no en la siguiente fase. [12]

1.3.2 Lenguaje Unificado de Modelado (UML)

UML es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema de software con tecnología orientado a objetos, proporciona una forma estándar de escribir los planos de un sistema, cubriendo tanto los conceptos de los procesos del negocio y funciones del sistema, como las acciones concretas: las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes de software reutilizables. [13]

Sirve para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real mediante el uso de conjunto de notaciones y diagramas. UML se utiliza además para detallar los artefactos en el sistema; para documentar y construir el lenguaje en el que está descrito el modelo.

1.3.3 Herramienta CASE para el desarrollo de software

Cuando se habla de herramientas de modelado en la disciplina de Ingeniería de Software, es importante mencionar las herramientas CASE (Computer Aided Software Engineering) que reemplazan al papel y al lápiz por el ordenador para transformar la actividad de desarrollar software en un proceso automatizado, lo cual es lo mismo que Ingeniería de Software asistida por computadoras. Las herramientas CASE son usadas en algunas de las fases de desarrollo de sistemas de información, incluyendo análisis, diseño y programación. Su objetivo fundamental es proveer un lenguaje para describir el sistema general que sea lo suficientemente explícito para generar todos los programas necesarios. Algunas de las herramientas CASE más conocidas son: Rational Rose Enterprise, Visual Paradigm (VP), Enterprise Architect (EA), entre otras.

Capítulo 1. Fundamentación Teórica

Estas herramientas ofrecen muchos beneficios para todos los involucrados en un proyecto, por ejemplo, líder del proyecto, analistas, arquitectos, desarrolladores y otros.

- ✓ Los Analistas pueden capturar los requisitos con un modelo de casos de uso.
- ✓ Los Diseñadores/Arquitectos pueden producir el modelo de diseño para enunciar la interacción entre los objetos o los subsistemas de la misma o de diferentes capas.
- ✓ Los Desarrolladores pueden transformar los modelos en una aplicación que funcione, y buscar un subconjunto de clases y métodos y asimilar el entendimiento de cómo lograr interfaces con ellos.

Dentro de las herramientas CASE mencionadas anteriormente, para el desarrollo de la solución propuesta se decidió utilizar Visual Paradigm (VP).

Visual Paradigm (VP)

Es una poderosa herramienta CASE para la representación visual de UML. Está diseñada para usuarios como: los ingenieros de software, analistas del sistema, analistas del negocio y los arquitectos, que están interesados en la construcción de sistemas de software confiables. El ambiente de Visual Paradigm en conjunto con UML proporciona medios para realizar el análisis y el diseño del sistema orientado a objeto. Permite crear diversos tipos de diagramas en un ambiente totalmente visual. Esta herramienta tiene unas características gráficas muy importantes como son:

- ✓ Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- ✓ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ✓ Capacidades de ingeniería directa e inversa.
- ✓ Modelo y código que permanece sincronizado en todo el ciclo de desarrollo. [14]

1.3.4 Sistema gestor de base de datos

Un Sistema Gestor de Bases de Datos (SGBD) o DBMS (DataBase Management System) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de

Capítulo 1. Fundamentación Teórica

datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. [15]

Algunos ejemplos de SGBD son Oracle, DB2, PostgreSQL, MySQL, MS SQL Server, etc.

Las características de un Sistema Gestor de Base de Datos SGBD son: [16]

- **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 con seguridad 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.

Capítulo 1. Fundamentación Teórica

- **Respaldo y recuperación.** Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo 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. Dentro de los sistemas gestores de base de datos inicialmente mencionados, para el desarrollo de la solución propuesta se decidió utilizar MySQL en su versión 5.1.

MySQL

MySQL es un gestor de bases de datos SQL en una implementación Cliente-Servidor que consta de un servidor y diferentes clientes (programas/librerías). Podemos agregar, acceder, y procesar datos grabados en una base de datos. Actualmente el gestor de base de datos juega un rol central en la informática, como única utilidad, o como parte de otra aplicación.

De manera que las siglas de MySQL son Structured Query Language: Lenguaje de Consulta Estructurado. MySQL al ser una implementación Cliente-Servidor consta de un servidor y diferentes clientes (programas/librerías). Se puede agregar, acceder, y procesar datos grabados en una base de datos. Actualmente los gestores de base de datos juegan un rol central en la informática, como única utilidad, o como parte de la aplicación.

Ventajas:

- ✓ Veloz en la accesibilidad a los datos introducidos en las distintas tablas independientes que forman las bases de datos de este lenguaje.
- ✓ Es muy rápido, confiable, robusto y fácil de usar tanto para volúmenes de datos grandes como pequeños (siempre, claro está, comparada con las de su categoría).

Capítulo 1. Fundamentación Teórica

- ✓ Además tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios.
- ✓ Es software libre, por tanto, se puede obtener y trabajar sin preocupaciones, utiliza la licencia GPL para definir que se puede o no hacer.
- ✓ Es muy utilizado para el desarrollo de aplicaciones y páginas Web.
- ✓ Corre tanto en PCs con Sistema Operativo Windows o Linux.

1.3.5 Internet

Internet consiste en el conjunto de redes: redes de ordenadores y equipos físicamente unidos mediante cables que conectan puntos de todo el mundo. Gigantesca Red que se difumina en ocasiones porque los datos pueden transmitirse vía satélite, o a través de servicios como la telefonía celular, o porque a veces no se sabe muy bien a dónde está conectada. Al contrario de lo que se piensa comúnmente, Internet no es sinónimo de World Wide Web (WWW, o "la Web").

La WWW consiste desplegar gráficos y usar el mouse para "navegar" por lugares en Internet. Anteriormente el acceso era complicado y poco interactivo: en nuestras pantallas sólo se mostraban textos y se debían usar instrucciones complicadas o programas manejados con el teclado. Internet incluye aproximadamente 5000 redes en todo el mundo y más de 100 protocolos distintos basados en TCP/IP (Protocolo de Transferencia y Control), que se configura como el protocolo de la red. Los servicios disponibles en la red mundial de PC, han avanzado mucho gracias a las nuevas tecnologías de transmisión de alta velocidad, como DSL (Línea de Abonado Digital) y Wireless (Comunicación Inalámbrica).

1.3.6 Tecnologías para crear aplicaciones Web

Una Aplicación Web es una especialización de las aplicaciones Cliente/Servidor, están comúnmente estructuradas como una aplicación en tres capas, son conocidas como aplicaciones servidor, siendo su objetivo primordial la separación de la lógica de negocios de la lógica de diseño. Las tres capas son: presentación, negocio y datos.

Capítulo 1. Fundamentación Teórica

Ventajas:

- Permite a los clientes o usuarios migrar de sistema operativo o cambiar de hardware libremente sin afectar el funcionamiento de las aplicaciones servidor.
- No se requieren complicadas combinaciones Hardware/Software para utilizar estas aplicaciones, los requerimientos mínimos consisten en un cómputo con un buen navegador web.

Facilita el trabajo a larga distancia, permite trabajar desde cualquier PC con conexión a Internet.

Desventajas:

- Necesidad de conexión permanente y rápida a Internet hacen que el acceso a estas aplicaciones no esté al alcance de todos.
- Elementos de interacción muy limitados.

Servidor Web Apache

Servidor Web hecho por excelencia, su configuración, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. Apache muestra los beneficios del trabajo cooperativo dentro de Internet. Apache, siendo un programa desarrollado en un ambiente software libre, es reconocido en muchos ámbitos empresariales y tecnológicos, por las siguientes razones:

- Corre en una multitud de Sistemas Operativos, por lo que es prácticamente universal.
- Tecnología de código fuente abierta y gratuita.
- Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para este servidor, que son adaptables.
- Trabaja con PHP, PERL y otros lenguajes de script.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.

Capítulo 1. Fundamentación Teórica

Lenguajes de Programación

Al mundo encontrarse inmerso en un desarrollo continuo, son muchos los lenguajes para programar Aplicaciones Web que han ido surgiendo, en la actualidad estos lenguajes se dividen en:

- Lenguajes del lado del Servidor: son los lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y son enviados al cliente en un lenguaje comprensible.
- Lenguajes del lado del Cliente: independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio.

Entre los diferentes lenguajes de programación se encuentra el JavaScript, utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario.

Los lenguajes del lado del servidor más utilizados en el mundo por los desarrolladores de aplicaciones Web son: PHP, ASP, PERL (Practical Extracting and Reporting Language o Lenguaje de Programación para Extraer Información de Archivos de Texto) y JSP (JavaServer Pages). En el caso de las Aplicaciones que están integradas a Servicios Web, el lenguaje más usado es el XML, constituyendo el formato estándar para los datos que se va a intercambiar.

PHP

PHP (Hypertext Preprocessor) es un lenguaje script (no se compila para conseguir códigos máquina si no que existe un intérprete que lee el código y se encarga de ejecutar las instrucciones que contiene éste código), para el desarrollo de páginas web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto, y a que es de Open Source (código abierto), es el más popular y extendido en la web.

Este es un lenguaje de programación clásico: con variables, sentencias condicionales, bucles, funciones, entre otras. Además de ser gratuito, trabaja del lado del servidor y permite interactuar con muchos gestores de Bases de Datos.

Capítulo 1. Fundamentación Teórica

Justificación de su uso

- ✓ Dado que PHP es un lenguaje multiplataforma se ha tomado para programar la aplicación correspondiente, pues permite utilizar el mismo código fuente, pudiendo ser compilado y ejecutado en las diferentes versiones de Unix, Windows (95,98, NT, ME, 2000, XP, entre otras) y Macs.
- ✓ Además puede utilizar la mayoría de los gestores de Base de Datos que se utilizan en la actualidad: MS SQL, Oracle, Informix, PostgreSQL, y por supuesto MySQL.
- ✓ Es Open Source, por tanto, el usuario no necesita pertenecer a una compañía para poder hacer los arreglos que crea pertinente en funciones que no trabajen bien, ni requiere de pagos por las actualizaciones de cada versión.
- ✓ Es muy sencillo de aprender.
- ✓ Utiliza gran cantidades de extensiones que le permiten expandir su potencial y así igual sucede con las librerías, cuando fácilmente podrá crear una API para ésta. Algunas ya vienen implementadas y permiten el manejo de gráficos, archivos PDF, Flash, Cybercash, calendarios, XML, IMAP, POP, entre otras.
- ✓ Es muy rápido y generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz.
- ✓ Permite las técnicas de Programación Orientada a Objetos (POO).

HyperText Markup Language (HTML)

Es un lenguaje de programación para crear páginas web conocido como código HTML, consiste en un lenguaje de marcas de hipertexto. Permite la inclusión de textos, imágenes fijas y móviles, vídeo, archivos, entre otros. Lenguaje en el que se escriben los documentos que se visualizarán a través de WWW.

JavaScript

JavaScript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con JavaScript se pueden crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y

Capítulo 1. Fundamentación Teórica

ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

Framework

El Framework es una aplicación genérica incompleta y configurable a la que se pueden añadir las últimas piezas para construir una aplicación concreta. Los objetivos principales que persigue un Framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Codelgniter

Codelgniter es un Framework para desarrollo de aplicaciones en PHP. Es de código abierto, tiene una interfaz simple y un acceso a sus librerías bien estructurado.

Es liviano, bastante fácil de utilizar, altamente configurable y cuenta con un gran grupo de desarrollo en Internet. Codelgniter PHP es adecuado si se necesita:

- Un rendimiento excepcional.
- Amplia compatibilidad con el estándar de las cuentas de hosting que ejecutar una gran variedad de versiones y configuraciones de PHP.
- Un marco que exija casi cero configuraciones.
- Un marco que no requiera que usted use líneas de comandos.
- Un marco que no le pida que se adhiera a la codificación de normas restrictivas.
- Evitar complejidad, favoreciendo soluciones simples.
- Documentación exhaustiva.

1.3.7 Arquitectura

Modelo Cliente/Servidor

La arquitectura Cliente/Servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que, definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e

Capítulo 1. Fundamentación Teórica

información. Se establece así un enlace de comunicación transparente entre los elementos que conforman la estructura. Entre las principales características de la arquitectura Cliente/Servidor, se pueden destacar las siguientes:

- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Ventajas del Modelo:

- El servidor no necesita potencia de procesamiento, parte del proceso se reparte a los clientes.
- Se reduce el tráfico de red considerablemente.

Modelo-Vista-Controlador (MVC)

Es una de las guías más usadas para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados:

- **Modelo:** representa los datos de la aplicación y las reglas de negocio.
- **Vistas:** representa a los formularios de entrada y salida de la información.
- **Controlador:** representa al conjunto de controladores o páginas servidoras que procesan las peticiones de los usuarios y controla el flujo de ejecución del sistema.

1.3.8 Herramientas a utilizar.

Diseño Gráfico

Existen muchas herramientas que permiten realizar el Diseño Gráfico de una Aplicación Web, entre ellas se encuentra el **Macromedia Fireworks**, herramienta fácil de maniobrar para cualquier tipo de gráficos,

Capítulo 1. Fundamentación Teórica

por estar integrado al Dreamweaver se hace más fácil su uso, por lo que simplifica el trabajo con imágenes en páginas Web. El **Adobe Photoshop CS3** durante cualquier tipo de desarrollo permite al usuario desplegar sus conocimientos con total libertad.

Diseño de Interfaz

Macromedia MX es uno de los editores de desarrollo Web más utilizado a nivel profesional para la creación de sitios Web. Su amplio abanico de herramientas permite crear desde la más simple página Web personal hasta el sitio Web más completo y complejo para una gran empresa y utilizar casi todos los recursos de la Web. Consiste en un editor de HTML para el diseño, codificación y desarrollo de páginas, sitios y aplicaciones Web; permite entre otras cosas la edición visual, es decir, crear páginas rápidamente sin tener la necesidad de escribir una línea de código. Entre otros códigos que permite teclear se encuentran: JavaScript, PHP, CSS, XML y más. Soporta tecnologías como Microsoft ASP, Microsoft ASP.NET, JSP y PHP. Es importante conocer que actualmente el paquete completo de Macromedia fue comprado por la compañía Adobe. Lanzando este año la última versión del Dreamweaver dentro de la Suite Adobe CS3, con el nombre de **Adobe Dreamweaver CS3**

phpDesigner

phpDesigner es un IDE y editor PHP completo, totalmente gratuito que trabaja con HTML, CSS y JavaScript y dispone de un entorno claro y cómodo para el programador. Ayuda a que el código fuente sea más claro y a simplificar el tiempo gracias a sus múltiples herramientas y utilidades específicas para PHP. phpDesigner incluye las librerías más comunes de PHP, así como los scripts y códigos prediseñados más utilizados. Además permite utilizar la función de autocompletar el código mientras se escribe y colorea el texto para hacer más clara su lectura.

A modo de conclusión, en este capítulo se profundizó en el conocimiento de conceptos y definiciones necesarios para una mejor comprensión del trabajo. Se determinó que los sistemas similares existentes a nivel internacional y nacional, no cubren las necesidades actuales de informatización en Albet. Se realizó un análisis de las tecnologías involucradas en el desarrollo del sistema propuesto, fundamentándose también la elección del lenguaje de programación PHP, el sistema gestor de bases de datos MySQL 5.1, la metodología RUP y las herramientas phpDesigner 6.0, Visual Paradigm 6.4 y Macromedia Dreamweaver 8.

Capítulo 2. Características del Sistema

Capítulo 2. Características del Sistema

En el presente capítulo se hace la descripción de la propuesta de solución, para ello se detallan los procesos del negocio que tienen que ver con el objeto de estudio. Se abordan aspectos del modelo de negocio como sus actores y trabajadores. Se muestran los diagramas de casos de uso del negocio, así como el diagrama de actividades, el de objeto del negocio y las descripciones de los casos de uso del negocio. Se definen los requerimientos funcionales y los no funcionales, así como los actores y casos de uso del sistema, incluyendo su descripción.

2.1 Objeto de estudio

La empresa Albet Ingeniería y Sistemas, posee los derechos comerciales de todos los productos y servicios que desarrolla la UCI y mediante la alianza con otras prestigiosas entidades ofrece soluciones integrales en la esfera de las tecnologías de la información y las comunicaciones. Los mejores especialistas de las diferentes instituciones del país, formando un equipo Cuba, gestionan, organizan y dirigen las soluciones conformado por varios proyectos, en dependencia de lo que se contrate. La selección de los especialistas para el desempeño de los diferentes roles en un proyecto, se realiza apoyándose en test psicológicos que lo caracterizan y muestran la capacidad de desempeñar un determinado rol en uno de los proyectos.

Este proceso de caracterización consiste en la obtención de atributos competenciales de los especialistas: capacidad intelectual, capacidad de aprendizaje, capacidad de innovación, roles predominantes en el trabajo en equipo, liderazgo y motivación. Además de la profundización de las características afines de los especialistas con los perfiles de Investigador y Líder de proyectos, destacando en ambos casos las potencialidades que existan como reservas de cuadros y haciendo mención al área de conocimientos que representa: técnicos, psicosociales ó empresariales, para una mayor precisión y certeza en la caracterización.

A raíz de los recursos técnicos con los que se cuenta actualmente para aplicar los test psicológicos que dan como resultado una caracterización psicológica de los especialistas, se hace tediosa la obtención de los resultados de los test debido a que la forma en que se realiza hasta el momento es muy engorrosa y le resta mucho tiempo al proceso de caracterización, efectuado cada cierto intervalo de tiempo.

Capítulo 2. Características del Sistema

Aparentemente este proceso de caracterización, pudiese parecer sencillo, pero una vez planificado su aplicación se hace difícil su coordinación, garantizar los recursos indispensables para realizarlo con éxito; y posteriormente de la ejecución de los test, el procesamiento para la caracterización se torna engorroso.

El proceso de Aplicación y Procesamiento de los test actualmente es muy dificultoso para los profesionales encargados, estos son ejecutados a través de la herramienta Microsoft Excel, que el especialista a caracterizar realiza y salva sus respuestas con un número que lo identifica. Luego el profesional que aplica los test tiene que ubicar cada uno de los ficheros de respuesta de cada una de las computadoras utilizadas, en una misma computadora y en las carpetas designadas previamente según el identificador, para obtenerlos de manera colectiva.

El problema de la obtención de una caracterización psicológica radica en poder generarlas de forma dinámica y eficiente, pues esta tecnología usada para aplicar los test no cuenta con el proceso de tabular u obtener los resultados de forma dinámica.

A continuación se brinda una explicación más detallada de los diferentes procesos involucrados en la caracterización psicológica de especialistas para la entrada a proyectos informáticos.

2.1.1 Procedimiento para la Caracterización Psicológica

➤ *Aplicación (Responsable: Psicólogos)*

- Durante toda la aplicación deben estar solo las especialistas, el Psicólogo asignado.
- Una vez en el local reservado, se realiza la presentación del Psicólogo a cargo y se procede a dar una breve explicación sobre el objetivo de las pruebas que realizarán y como será el proceso de aplicación.
- Seguidamente se procede a dar la explicación del primer instrumento, asegurando que se entienda por parte del especialista, lo que tiene que hacer, en este momento pueden realizarse las aclaraciones pertinentes, en aras de la comprensión de la instrumento. Antes de comenzar cada ejercicio la especialista debe preguntar si aún existen dudas.

Capítulo 2. Características del Sistema

- En la medida que vayan completando cada prueba, el Psicólogo debe revisarla someramente con el objetivo de detectar algún error en el llenado de la técnica. De detectar alguno se llama la atención a la especialista para su corrección. Solo después que la prueba haya sido realizada por todos y revisada por el Psicólogo, este puede dar continuidad al proceso según el orden, siguiendo el mismo protocolo con cada una.
- Debe prestársele especial atención a la salva de cada prueba realizada y al final del proceso recuperar de cada máquina toda la información.

➤ *Caracterización (Responsable: Psicólogos)*

- Revisa la actualización automática de la BD obteniendo la correspondencia o no con los perfiles conformados.
- Finalmente elabora el Informe Individual de Caracterización Psicológica. Para ello realiza una integración de aquellas características que más se repiten, cruzando el reporte de varios instrumentos.

2.1.2 Objeto de automatización

En el presente trabajo se pretende automatizar el proceso de caracterización psicológica, de forma tal que todos los de datos de los especialistas a caracterizar así como su caracterización, se guarden en una base de datos, garantizando la integridad de la información contenida. Esto debe facilitar el trabajo de los profesionales que aplican los test y contribuir a que puedan obtener los resultados de los mismos de forma dinámica, y así mejorar el proceso de caracterización.

2.2 Modelo del Negocio

Un sistema, generalmente es complejo, por pequeño que sea. Por tal motivo se hace necesario dividirlo en fragmentos para poder comprenderlo y gestionar su complejidad. Estos fragmentos se pueden representar mediante modelos que permitan obtener sus características esenciales. Una técnica para la especificación de los requisitos más importantes del sistema, que brinda soporte al negocio, es el modelo del negocio, con lo cual se fortalece la idea de que sea el propio negocio el que determine los requisitos.

Capítulo 2. Características del Sistema

De ahí, que en el campo del software también resulte útil la creación de modelos que organicen y presenten los detalles importantes de problemas reales que se relacionan con el sistema informático a construir. Estos modelos deben cumplir una serie de propiedades, entre ellas la de ser coherentes y relacionados. Uno de los modelos útiles previo al desarrollo de un software es el modelo del negocio.

La modelación del negocio tiene como objetivos:

- Comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema.
- Entender los problemas actuales de la organización e identificar las mejoras potenciales.
- Asegurar que los consumidores, usuarios finales y desarrolladores tengan un buen entendimiento y obtención de los requerimientos del sistema, que va a soportar la organización.

Como los procesos están claramente definidos se logra visualizar quienes son las especialistas que lo inician, quienes son los beneficiados, y al mismo tiempo quienes desarrollan las actividades en cada uno de estos procesos; solo es necesario modelar el negocio propuesto, decidiéndose realizar el mismo.

2.2.1 Actores del negocio

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados.

Actor	Descripción
Especialista	Individuo externo que interactúa con el negocio y se beneficia de este. Cuyo interés es realizar los test psicológicos y obtener una caracterización.

Tabla 2. 1 Descripción de los actores del negocio

2.2.2. Trabajadores del negocio

Un trabajador del negocio representa a especialistas o sistemas dentro del negocio que son los que realizan las actividades que están comprendidas dentro de un caso de uso. Estos trabajadores están

Capítulo 2. Características del Sistema

dentro de la frontera del negocio, son los que en un futuro se convertirán en usuarios del sistema que se quiere construir.

Trabajador	Descripción
Psicólogo	Empleado responsable de aplicar, procesar los test psicológicos y elaborar un informe de caracterización psicológica.

Tabla 2. 2 Descripción de los trabajadores del negocio.

2.2.3. Diagrama de casos de uso del negocio

Un caso de uso del negocio representa a un proceso de negocio, por lo que se corresponde con una secuencia de acciones que producen un resultado observable para ciertos actores del negocio. Desde la perspectiva de un actor individual, define un flujo de trabajo completo que produce resultados deseables.

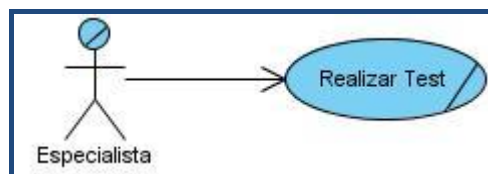


Fig. 2. 1 – Diagrama de Casos de Uso del Negocio

2.2.4. Especificación de los casos de uso del Negocio.

Caso de Uso:	Realizar Test
Actores:	Especialista (inicia)
Trabajadores:	Psicólogo
Resumen: El caso de uso se inicia cuando el especialista se presenta en el local donde se aplicará el test. El psicólogo le indica cuál es su identificador y el puesto de trabajo donde realizará el test. Luego el psicólogo explica el objetivo del test y el especialista procede a realizarlo. El caso de uso finaliza cuando el especialista termina y el psicólogo guarda las respuestas.	

Tabla 2. 3 – Descripción del Caso de Uso del Negocio: Realizar Test

Capítulo 2. Características del Sistema

2.2.5. Diagrama de actividades.

Los casos de uso del negocio consisten en secuencias de actividades que, en conjunto, producen un resultado para el actor del negocio. El proceso consiste de un flujo básico de una o más alternativas. La estructura del flujo se describe gráficamente con la ayuda de un diagrama de actividad.

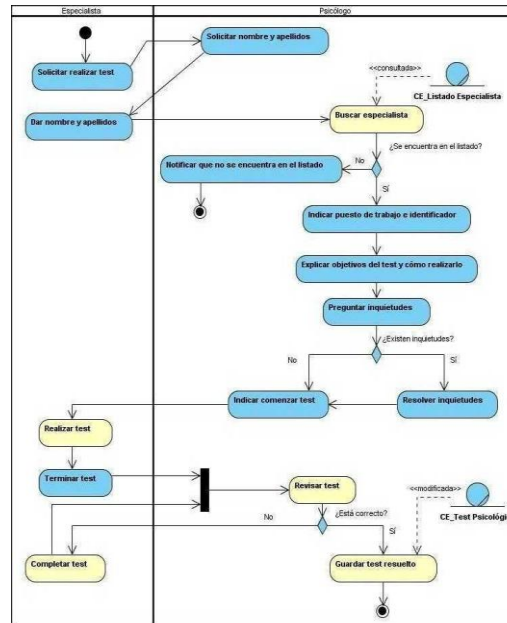


Fig. 2. 2 – Diagrama de Actividades del Caso de Uso del Negocio: Realizar Test

2.2.6. Diagrama de clases del modelo de objetos.

El modelo de objetos del negocio se describe a través de un diagrama de clases, en el cual se representa la relación entre los trabajadores y las entidades del negocio, su participación dentro del negocio y la relación entre ellos.

Las entidades del negocio son objetos que los trabajadores del negocio inspeccionan, manipulan, producen o utilizan durante la realización de los casos de uso del negocio.

Capítulo 2. Características del Sistema

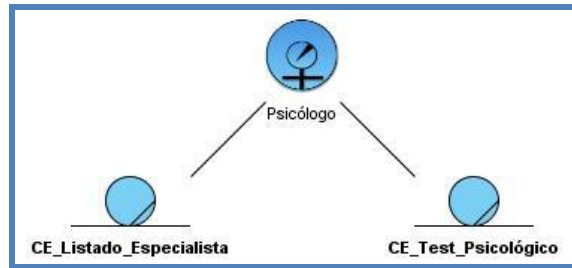


Fig. 2. 3 – Diagrama de Objetos

2.3 Especificación de los requisitos de software

Los requerimientos de software son condiciones o capacidades que debe tener el sistema para satisfacer las necesidades de un cliente, estos deben ser especificados por escrito.

2.3.1 Requerimientos Funcionales.

Los requerimientos funcionales son aquellos requisitos que, desde el punto de vista de las necesidades del usuario, debe cumplir el sistema y que están fuertemente ligados a las opciones del programa.

RF 1 – El sistema debe permitir autenticarse a los usuarios.

RF 1.1- Mostrar un mensaje de alerta en caso de no poder entrar al sistema porque no cuenta con los permisos necesarios.

RF 2 – El sistema debe permitir Registrar un Especialista.

RF 2.1 – Al registrar a la especialista cada uno tiene un identificador.

RF 2.2 – Mostrar un mensaje de alerta en caso de que el Especialista a registrar ya este registrado.

RF 3 – Debe permitir Modificar los datos de un Especialista.

RF 4 - El sistema debe permitir Buscar un Especialista.

RF 4.1- Mostrar un mensaje de alerta en caso de no encontrarse el especialista.

RF 5 - El sistema debe permitir Listar los Especialistas que han realizado los test.

Capítulo 2. Características del Sistema

RF 6 - El sistema debe permitir publicar los test a realizar por los especialistas.

RF 7 - El sistema debe permitir Obtener un Reporte de la Caracterización Psicológica de un Especialista.

RF 7.1- Mostrar un mensaje de alerta en caso de no encontrarse el especialista.

RF 8 - El sistema debe posibilitar al especialista la selección de los distintos test que debe realizar mostrando los datos y contenido de los mismos (nombre, objetivo, las secciones y preguntas a realizar).

RF 9 - El sistema debe permitir seleccionar y mostrar un test solicitado.

RF 10 - El sistema debe permitir la validación de las respuestas de los test:

RF 10.1 - Mostrar un mensaje de alerta en caso de quedar alguna pregunta sin responder.

RF 11 - El sistema debe permitir salvar los test luego de haberlos completado.

RF 11.1- Mostrar un mensaje de alerta en caso de quedar alguna pregunta sin resolver.

2.3.2 Requerimientos No Funcionales.

Los requerimientos no funcionales son características que describen alguna forma o restricción para la realización de algún requerimiento (funcionalidad) o conjunto de ellas e inclusive todos los requerimientos. Se consideran los atributos del sistema, propiedades que debe tener el producto. A continuación se muestran los requerimientos no funcionales:

RNF 1- Requerimiento de Apariencia o interfaz externa:

RNF 1.1 Identificación de colores y formatos con acciones del sistema.

RNF 1.2 La interfaz debe ser de fácil comprensión en su funcionamiento permitiendo la utilización del sistema sin mucho entrenamiento.

RNF 2- Usabilidad:

Capítulo 2. Características del Sistema

RNF 2.1- El sistema podrá ser usado por cualquier especialista que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.

RNF 3- Rendimiento:

RNF 3.1- Tiempos de respuestas no mayor de 2 segundos, al igual que la velocidad de procesamiento de la información.

RNF 4- Soporte:

RNF 4.1- Se requiere un servidor de bases de datos.

RNF 5- Portabilidad:

RNF 5.1- El sistema funciona sobre cualquier plataforma (Windows y/o Linux).

RNF 6- Confiabilidad:

RNF 6.1- La aplicación tiene soporte para recuperación ante fallos y errores.

RNF 7- Funcionalidad:

RNF 7.1- Reducir al mínimo el tiempo en que carga el sistema.

RNF 8- Software:

RNF 8.1- Admite cualquier navegador.

RNF 8.2- PHP Designer 6.1

RNF 8.3- Apache 2.0.x

RNF 8.4- PHP 5.0.x

RNF 8.5 - MySQL 5.1

Capítulo 2. Características del Sistema

RNF9- Hardware:

RNF 9.1 - Pentium 600 MHz o superior.

RNF 9.2- 128 MB de memoria RAM.

RNF 9.3 - 20 GB de disco duro.

2.4 Descripción del Sistema propuesto

Para dar cumplimiento al objetivo propuesto al inicio del trabajo, y teniendo en cuenta todos los requerimientos planteados, el sistema que se propone debe contener todas las funcionalidades previstas.

Para utilizar todas las funcionalidades del sistema, se considera la existencia de dos roles, un usuario será: Psicólogo o Especialista. El sistema tendrá una autenticación previa para prevenir que cada uno de los usuarios pueda realizar solo operaciones para las que tienen permisos.

En la base de datos se llevará el control de todos esos cambios, y de los test que se hayan realizado guardando las respuestas en la misma.

El Psicólogo tiene a su disposición Listar los Especialistas que han realizado los test, Buscar al Especialista, Obtener un Reporte de la Caracterización Psicológica del mismo, Publicar un Test y Elaborar un informe según la caracterización psicológica.

En resumen, con esta herramienta de apoyo para la caracterización psicológica se pretende lograr agilizar el proceso de la misma.

2.4.1 Modelo de Casos de Uso del Sistema

El Modelo de Casos de Uso del Sistema es aquel en el que se representan los actores del Sistema, los Casos de Uso y sus relaciones.

Capítulo 2. Características del Sistema

Haciendo uso de las facilidades que brinda UML, se puede proceder a la captura de los requisitos funcionales del sistema y determinar cómo será utilizado desde el punto de vista del actor, para así construirlo sobre la base de sus necesidades.

Actor	Descripción
Usuario	Va a ser todo aquel personal que va a tener algún rol en el sistema, es decir, los que van a poder trabajar con la aplicación.
Especialista	Representa a la persona que podrá acceder al sistema para realizar los test indicados, podrá registrarse en el sistema y modificar sus datos.
Psicólogo	Representa a la persona encargada de entrar al sistema bajo el permiso de psicólogo y administrador del sistema.

2.4.2 Listado de Casos de Usos (CU)

CU - 1	Autenticar Usuario
Actor	Usuario (Psicólogo, Especialista)
Descripción	Permite que solamente puedan acceder a la aplicación todos aquellos que trabajarán directamente con la misma, dándole acceso a operar en el sistema, según las funcionalidades a las que tiene permiso.
Referencias	RF 1

CU - 2	Registrar Especialista
Actor	Especialista
Descripción	Permite que el especialista que tiene que realizar el test se registre en el sistema.
Referencias	RF 2

CU - 3	Modificar Especialista
---------------	------------------------

Capítulo 2. Características del Sistema

Actor	Psicólogo
Descripción	Permite modificar los datos del especialista.
Referencias	RF 3

CU – 4	Buscar Especialista
Actor	Psicólogo
Descripción	Se permitirá buscar a un especialista y visualizar sus datos.
Referencias	RF 4

CU – 5	Listar Especialista
Actor	Psicólogo
Descripción	Se permitirá listar todos los especialistas que han realizado los test.
Referencias	RF 5

CU – 6	Publicar Test
Actor	Psicólogo
Descripción	Se permitirá publicar el test que debe realizar el especialista.
Referencias	RF 6

CU – 7	Obtener Reporte Caracterización Psicológica
Actor	Psicólogo
Descripción	Se permitirá obtener la caracterización psicológica del especialista luego de que este haya culminado la realización del test.
Referencias	RF 7

CU – 8	Realizar Test
Actor	Especialista
Descripción	Permite al especialista responder el test al cual ha accedido.
Referencias	RF 8

Capítulo 2. Características del Sistema

2.4.3 Diagrama de casos de uso del sistema.

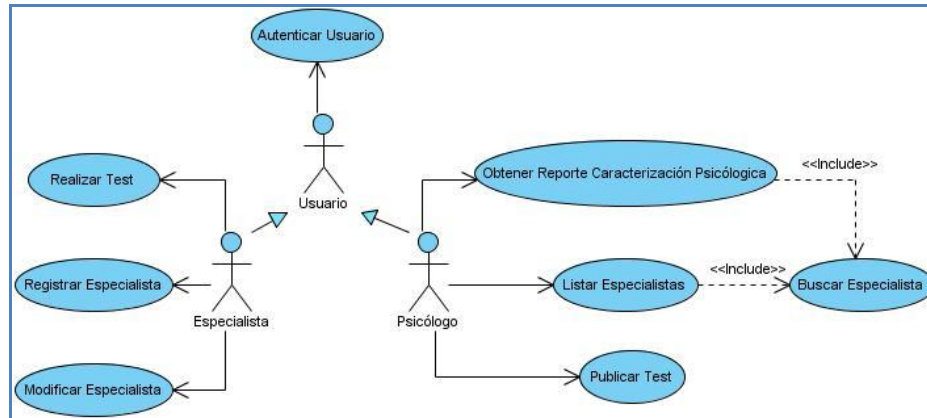


Fig. 2. 4 – Diagrama de Casos de Uso del Sistema

2.5 Expansión de los Casos de Usos

Mediante los casos de uso expandidos se describe paso a paso la secuencia de eventos que los actores utilizan para completar un proceso a través del sistema. Este sería el último paso en el análisis, para pasar a la construcción de la solución propuesta. A continuación se muestran la descripción de los principales casos de uso, para ver la descripción de los demás, remitirse al expediente de proyecto.

Descripción del Caso de Uso "Autenticar Usuario"

Caso de Uso:	Autenticar Usuario
Actores:	Usuario (Especialista, Psicólogo)
Resumen:	Permite que solamente puedan acceder a la aplicación todos aquellos que trabajarán directamente con la misma, dándole acceso a operar en el sistema, según los vínculos a los que tienen permiso.
Precondiciones	
Referencias	RF 1
Prioridad	Crítico
Flujo Normal de Eventos	
Sección "Autenticar Usuario"	

Capítulo 2. Características del Sistema

Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el usuario accede a la aplicación.	
	2. Muestra la Interfaz Autenticar Usuario con un formulario con los datos a llenar para la autenticación: Usuario Contraseña
3. El usuario introduce los datos.	4. Valida los datos introducidos.
	5. Si los datos son correctos, muestra la Interfaz Principal para el rol que desempeña en la aplicación con todas las funcionalidades a las que tiene acceso. Culmina CU.
Prototipo de Interfaz	
Flujos Alternos	
Sección “Autenticar Usuario”	
Acción del Actor	Respuesta del Sistema
	4.1 Si los datos son incorrectos, muestra un mensaje de error “ Usuario o contraseña incorrecta ”. Y regresa a la interfaz Autenticar Usuario. Ir al paso 2 del flujo normal de eventos.
Prototipo de Interfaz	
Poscondiciones	El usuario queda autenticado en el sistema.

Descripción del Caso de Uso “Registrar Especialista”

Caso de Uso:	Registrar Especialista
Actores:	Especialista (inicia)

Capítulo 2. Características del Sistema

Resumen:	El CU se inicia cuando se accede a la opción registrar especialista. El caso de uso termina cuando el especialista queda registrado en el sistema.	
Precondiciones		
Referencias	RF 2	
Prioridad	Crítico	
Flujo Normal de Eventos		
Sección “Registrar Especialista”		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso inicia cuando el actor accede a la opción “Registrar”.		
	<p>2. Muestra los datos predeterminados y brinda la posibilidad de introducir los datos del especialista:</p> <ul style="list-style-type: none"> • Nombre • Primer Apellido • Segundo Apellido • Carnet de Identidad • Usuario • Contraseña • Ubicación laboral 	
<p>3. Introduce los datos del especialista</p> <ul style="list-style-type: none"> • Nombre • Primer Apellido • Segundo Apellido • Carnet de Identidad 		

Capítulo 2. Características del Sistema

<ul style="list-style-type: none"> • Usuario • Contraseña • Ubicación laboral 	
	<ol style="list-style-type: none"> 4. Valida los datos introducidos. <ol style="list-style-type: none"> a. En caso de datos incorrectos. Ver Flujo Alternativo: "Datos Incorrectos."
	<ol style="list-style-type: none"> 5. Agrega el especialista.
	<ol style="list-style-type: none"> 6. Muestra el mensaje de información "Se ha registrado satisfactoriamente".
	<ol style="list-style-type: none"> 7. Finaliza el CU.
Prototipo de Interfaz	
Flujos Alternos	
Alternativa "Datos Incorrectos."	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Muestra el mensaje: "Datos Incorrectos" y señala los datos incorrectos.
<ol style="list-style-type: none"> 2. Corrige los datos incorrectos. 	<ol style="list-style-type: none"> 3. Pasa al paso 4 del Flujo Normal de Eventos.
Prototipo de Interfaz	
Poscondiciones	El especialista queda registrado en el sistema.

Descripción del Caso de Uso "Modificar Especialista"

Caso de Uso:	Modificar Especialista
Actores:	Psicólogo (inicia)
Resumen:	El CU se inicia cuando se accede a la opción modificar especialista. El caso de uso termina cuando los datos de la especialista son modificados.
Precondiciones	El usuario se debe haber autenticado.
Referencias	RF 3
Prioridad	Crítico

Capítulo 2. Características del Sistema

Flujo Normal de Eventos	
Sección "Modificar Especialista"	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor selecciona un especialista y accede a la opción Modificar Datos.	
	2. Muestra la Interfaz Modificar Datos con los datos del especialista seleccionado.
3. Realiza los cambios necesarios.	4. Valida los datos introducidos. a. En caso de datos incorrectos. Ver Flujo Alterno: "Datos Incorrectos."
	5. Modifica en la Base de Datos.
	6. Muestra el mensaje de información "Se han modificado los datos correctamente."
	7. Finaliza el CU.
Prototipo de Interfaz	
Flujos Alternos	
Alternativa "Datos Incorrectos."	
Acción del Actor	Respuesta del Sistema
	1. Muestra el mensaje: "Datos Incorrectos" y señala los datos incorrectos.
2. Corrige los datos incorrectos.	3. Pasa al paso 5 del Flujo Normal de Eventos.
Prototipo de Interfaz	
Poscondiciones	Los datos del especialista son modificados y guardados en el sistema.

Descripción del Caso de Uso "Obtener Reporte Caracterización Psicológica"

Caso de Uso:	Obtener Reporte Caracterización Psicológica
Actores:	Psicólogo

Capítulo 2. Características del Sistema

Resumen:	El CU se inicia cuando se accede a la opción Obtener Reporte Caracterización Psicológica, el sistema brinda la posibilidad de seleccionar uno de los especialistas y muestra la caracterización psicológica del mismo.	
Precondiciones:	El usuario se debe haber autenticado.	
Referencias	RF 7	
Prioridad	Crítico	
Flujo Normal de Eventos		
Sección "Obtener Reporte Caracterización Psicológica"		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso inicia cuando el actor accede a la opción Obtener Reporte Caracterización Psicológica.		
	2. Muestra la Interfaz Obtener Reporte Caracterización Psicológica con un listado de los especialistas que han realizado los test.	
3. Selecciona el especialista del cual desea la caracterización psicológica.	4. Muestra la caracterización psicológica del especialista seleccionado.	
Prototipo de Interfaz		
Poscondiciones	Se obtiene un reporte de la caracterización de un especialista.	

Descripción del Caso de Uso "Realizar Test"

Caso de Uso:	Realizar Test
Actores:	Especialista
Resumen:	El CU se inicia cuando se accede a la interfaz principal del actor y el sistema brinda la posibilidad de seleccionar uno de los test para realizarlo. El actor completa el test al que ha accedido, luego de completar el test, el sistema guarda las respuestas, el caso de uso termina.
Precondiciones:	El usuario se debe haber autenticado.

Capítulo 2. Características del Sistema

Referencias	RF 8	
Prioridad	Crítico	
Flujo Normal de Eventos		
Sección “Realizar Test”		
Acción del Actor	Respuesta del Sistema	
<p>1. El caso de uso inicia cuando el actor se autentica y se encuentra en la interfaz a la que tiene acceso.</p>		
	<p>2. Muestra un listado con los test a realizar.</p>	
<p>3. Selecciona uno de los test:</p> <p>a. <i>Test Las Leyes.</i> Ver Sección “ Test Las Leyes”.</p> <p>b. <i>Test Inventario de Estilos Decisionales.</i> Ver Sección “ Test Inventario de Estilos Decisionales”.</p> <p>c. <i>Test de Autopercepción de Belbin</i> Ver Sección “ Test Inventario de Autopercepción de Belbin ”.</p> <p>d. <i>Test Cuestionario de Estilos Cognitivos</i> Ver Sección “ Test Inventario de Estilos Cognitivos ”</p> <p>e. <i>Test Matriz de Fortalezas y Debilidades</i> Ver Sección “ Test Inventario Matriz de Fortalezas y Debilidades ”.</p>		
Sección “Test Las Leyes”		
Acción del Actor	Respuesta del Sistema	

Capítulo 2. Características del Sistema

4. Responde cada pregunta del test.	5. Valida que todas las preguntas estén respondidas. a. Si hay alguna pregunta que no esté respondida. Ver Flujo Alternativo "Respuestas Incompletas".
6. Guarda el test.	7. Salva las respuestas en la base de datos.
	8. Finaliza el CU.
Sección "Test Inventario de Estilos Decisionales"	
Acción del Actor	Respuesta del Sistema
4. Responde cada pregunta del test dando los valores 1, 2, 4 y 8 a cada una de las posibles respuestas.	5. Valida que todas las preguntas estén correctamente respondidas. a. Si hay alguna pregunta que no esté respondida. Ver Flujo Alternativo "Respuestas Incompletas". b. Si hay alguna que no tenga los valores indicados en las respuestas o se repitan los valores. Ver Flujo Alternativo: "Valores Incorrectos."
6. Guarda el test.	7. Salva las respuestas en la base de datos.
	8. Finaliza el CU.
Sección "Test de Autopercepción de Belbin"	
Acción del Actor	Respuesta del Sistema
4. Responde cada pregunta dando un valor de 10 puntos distribuidos entre las posibles respuestas.	5. Valida que cada pregunta tenga un total de 10 puntos y que en general todas sumen 70 puntos. a. Si en general las respuestas no suman 70 puntos. Ver Flujo Alternativo "Respuestas Incorrectas".

Capítulo 2. Características del Sistema

6. Guarda el test.	7. Salva las respuestas en la base de datos.
	8. Finaliza el CU.
Sección “Test Inventario de Estilos Cognitivos”	
Acción del Actor	Respuesta del Sistema
4. Responde cada pregunta del test marcando con una x la respuesta que considere.	5. Valida que todas las preguntas estén respondidas. a. Si hay alguna pregunta que no esté respondida. Ver Flujo Alternativo “Respuestas Incompletas”.
6. Guarda el test.	7. Salva las respuestas en la base de datos.
	8. Finaliza el CU.
Sección “Test Matriz de Fortalezas y Debilidades”	
Acción del Actor	Respuesta del Sistema
4. Responde cada pregunta del test dando valores del 1 al 5 como fortalezas o debilidades.	5. Valida que todas las preguntas estén respondidas. a. Si hay alguna pregunta que no esté respondida. Ver Flujo Alternativo “Respuestas Incompletas”. b. Si hay alguna que no este respondida correctamente. Ver Flujo Alternativo: “Respuestas Incorrectas”.
6. Selecciona de todas las preguntas aquellas que constituyen sus cinco debilidades y sus cinco fortalezas.	
7. Guarda el test.	8. Salva las respuestas en la base de datos.
	9. Finaliza el CU.
Prototipo de Interfaz	
Flujos Alternos	

Capítulo 2. Características del Sistema

Alternativa "Respuestas Incorrectas."	
Acción del Actor	Respuesta del Sistema
	1. Muestra el mensaje: "Debe completar correctamente la pregunta" y muestra las preguntas que no están correctamente respondidas.
2. Corrige las preguntas.	3. Pasa al paso 6 del Flujo Normal de Eventos.
Alternativa "Respuestas Incompletas."	
Acción del Actor	Respuesta del Sistema
	1. Muestra el mensaje: "Debe completar todas las preguntas" y muestra las preguntas que no están respondidas.
2. Completa las preguntas.	3. Pasa al paso 6 del Flujo Normal de Eventos.
Alternativa "Valores Incorrectos."	
Acción del Actor	Respuesta del Sistema
	1. Muestra el mensaje: "Debe completar todas las preguntas con los valores indicados y estos no se pueden repetir." y muestra los valores incorrectos.
2. Corrige las preguntas.	3. Pasa al paso 6 del Flujo Normal de Eventos.
<i>Prototipo de Interfaz</i>	
Poscondiciones	Las respuestas del test son salvadas en el sistema.

A modo de conclusión en este capítulo se hizo la descripción de la propuesta de solución, detallando los procesos del negocio que tienen que ver con el objeto de estudio. Se abordaron aspectos del modelo de negocio como sus actores y trabajadores. Se mostraron los diagramas de casos de uso del negocio, así como el diagrama de actividades, el de objeto del negocio y las descripciones de los casos de uso del negocio. Se definieron los requerimientos funcionales y los no funcionales, así como los actores y casos de uso del sistema, incluyendo su descripción.

Capítulo 3. Análisis y Diseño del Sistema

Capítulo 3. Análisis y Diseño del Sistema

Cada paso durante el ciclo de desarrollo del proyecto cumple un importante papel para obtener el producto final, en este caso se desarrolla el Flujo de Trabajo de Análisis y Diseño. Se explica la estructura del sistema, la arquitectura a utilizar, así como la representación gráfica de los diagramas de clases del análisis, de interacción y de clases del diseño con estereotipos web.

3.1 Arquitectura

Modelo-Vista-Controlador (MVC) Fue diseñada para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales son que el modelo, las vistas y los controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el modelo se refleje automáticamente en cada una de las vistas. Este modelo de arquitectura presenta varias ventajas:

- Hay una clara separación entre los componentes de un programa; que permite implementarlos por separado.
- Hay un API (Interfaz de Programación de Aplicaciones) muy bien definido; cualquiera que use el API, podrá reemplazar el modelo, la vista o el controlador, sin aparente dificultad.
- La conexión entre el modelo y sus vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

Al incorporar el modelo de arquitectura MVC a un diseño, las piezas de un programa se pueden construir por separado y luego unirlos en tiempo de ejecución. Si uno de los Componentes, posteriormente, se observa que funciona mal, puede reemplazarse sin que las otras piezas se vean afectadas. Este escenario contrasta con la aproximación monolítica típica de muchos programas Java. Todos tienen un Marco (Frame) que contiene todos los elementos, un controlador de eventos, y una gran cantidad de cálculos y la presentación del resultado. Ante esta perspectiva, hacer un cambio aquí no es nada trivial.

Definición de las partes:

Capítulo 3. Análisis y Diseño del Sistema

El **Modelo** es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El modelo no tiene conocimiento específico de los controladores o de las vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el modelo y sus vistas, y notificar a las vistas cuando cambia el modelo.

La **Vista** es el objeto que maneja la presentación visual de los datos representados por el modelo. Genera una representación visual del modelo y muestra los datos al usuario. Interactúa con el modelo a través de una referencia al propio modelo.

El **Controlador** es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del modelo o por alteraciones de la vista. Interactúa con el modelo a través de una referencia al propio modelo.

3.2 Modelo de Análisis.

El modelo de análisis es un modelo de objetos conceptual que analiza los requisitos mediante su refinamiento y estructuración y permite razonar sobre los aspectos internos del sistema, incluidos sus recursos compartidos. Además ofrece un mayor poder expresivo, una mayor formalización y proporciona una estructura centrada en el mantenimiento de aspectos tales como la flexibilidad ante los cambios y la reutilización. Esta estructura no solo es útil para el mantenimiento de los requisitos, sino también que se utiliza como entrada en las actividades de diseño e implementación. El principal artefacto que se genera durante este flujo de trabajo es la realización de los casos de uso del análisis que describen como se refinan estos en términos de colaboraciones dentro del modelo de análisis y además de cómo se ejecuta un caso de uso determinado en términos de las clases del análisis y de sus objetos.

3.2.1 Diagrama de Clases del Análisis

El diagrama de clases del análisis se realiza para cada caso de uso del sistema y muestra las clases participantes, y sus relaciones. Es un artefacto en el que se representan los conceptos en un dominio del problema y representa las cosas del mundo real. En los diagramas de clases se identifican tres tipos de clases: interfaz, controladoras y entidades.

Capítulo 3. Análisis y Diseño del Sistema

Las clases interfaz se utilizan para modelar la interacción entre el sistema y sus actores, es decir, usuarios y sistemas externos. Esta interacción a menudo implica recibir información y peticiones de los usuarios y sistemas externos. Cada clase de interfaz debe asociarse con al menos un actor.

Las *clases de control* representan coordinación, secuencia, transacciones y control de otros objetos, se usan con frecuencia para encapsular el control de un caso de uso en concreto. También se utilizan para representar derivaciones y cálculos complejos, como la lógica del negocio. Los aspectos dinámicos del sistema se modelan con clases de control, debido a que ellas manejan y coordinan las acciones y los flujos de control principales, y delegan trabajo a otros objetos, es decir, objetos de interfaz y de entidad.

Las *clases entidad* se utilizan para modelar la información que posee una vida larga y que es a menudo persistente, además, modelan el comportamiento asociado a algún fenómeno o concepto, como una especialista, un objeto del mundo real, o un suceso del mundo real. Las clases entidad suelen mostrar una estructura de datos lógica y contribuyen a comprender de qué información depende el sistema.

A continuación se muestran los diagramas de clases del análisis de los principales casos de uso.

- Caso de Uso Autenticar Usuario.

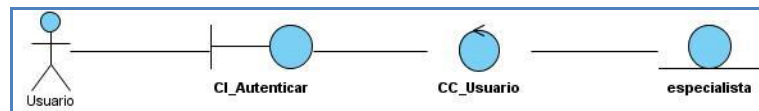


Fig. 3.1. Diagrama de Clases del Análisis Autenticar Usuario.

- Caso de Uso Registrar Especialista.

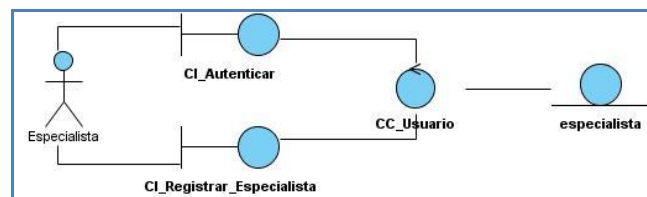


Fig. 3.2. Diagrama de Clases del Análisis Registrar Especialista.

- Caso de Uso Modificar Especialista.

Capítulo 3. Análisis y Diseño del Sistema

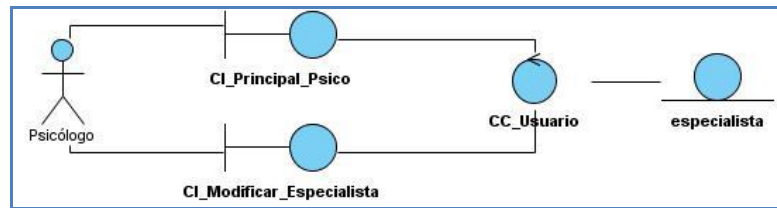


Fig. 3.3. Diagrama de Clases del Análisis Modificar Especialista.

- Caso de Uso Obtener Reporte Caracterización Psicológica.

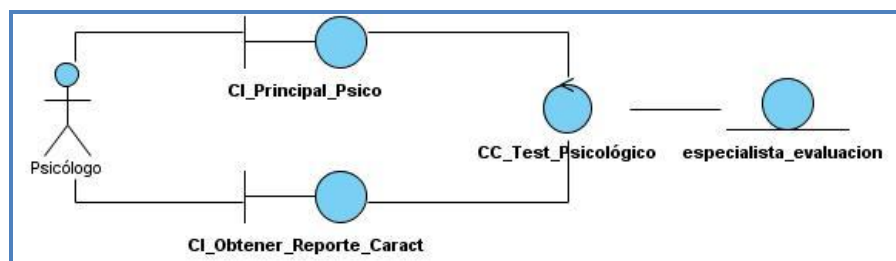


Fig. 3.7. Diagrama de Clases del Análisis Obtener Reporte Caracterización Psicológica.

- Caso de Uso Realizar Test.

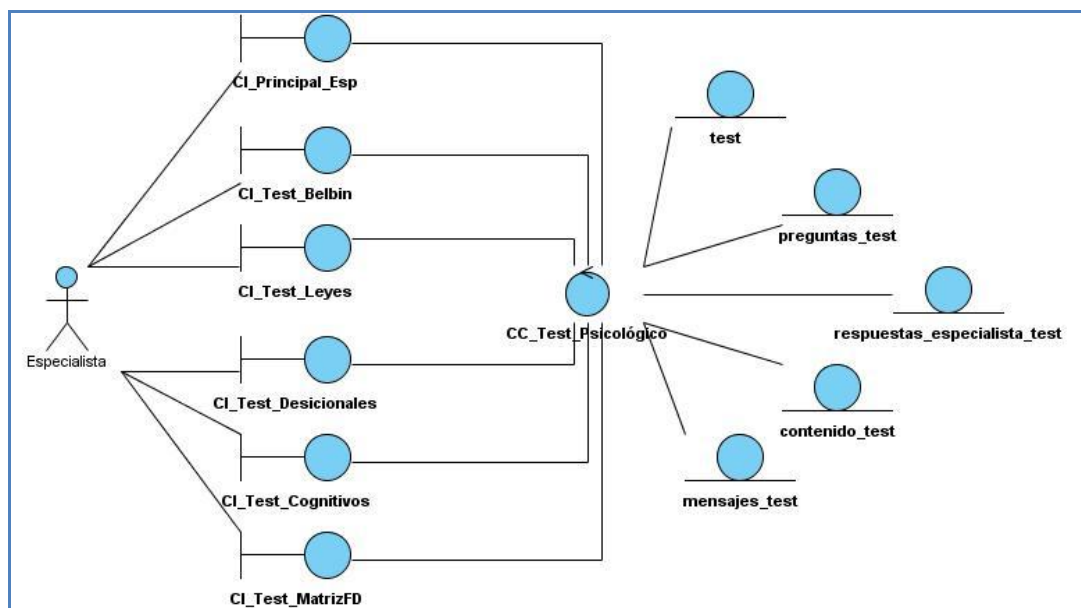


Fig. 3.9. Diagrama de Clases del Análisis Realizar Test.

Capítulo 3. Análisis y Diseño del Sistema

3.3 Modelo de Diseño

El modelo de diseño es un modelo que contribuye a establecer una arquitectura estable y sólida, y crear un plano del modelo de implementación. Es un modelo de objetos que describe la realización de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema que desarrolla. Se representa por un sistema de diseño que denota el subsistema de nivel más alto del modelo. La utilización de otro subsistema es, entonces, una forma de organización del modelo de diseño en porciones más manejables. [15]

3.3.1 Estereotipos Web.

En el diseño aplicaciones Web, Jim Conallen propone en 1998, una extensión al UML, a esta extensión se le conoce hoy en día como los estereotipos web, conformadas por Server page, Client Page, Form, Frame set, y Target. De todas las extensiones, presenta como elementos más significativos tres clases UML, que son los estereotipos Server Page, Client Page, y Form. En la Tabla 3.1 están representadas las relaciones que se establecen entre los tres elementos claves mencionados anteriormente.

Al ser RF una aplicación web, la misma se modelará con estereotipos definidos para este tipo de aplicaciones, los estereotipos web, lo cual proporcionará una mayor comprensión de las funcionalidades del sistema y el poder distinguir qué atributos, operaciones y relaciones están activas en el servidor y cuáles están activas cuando el usuario está interactuando con la página en el navegador cliente.

A continuación se brinda una explicación de cómo son usados estos estereotipos en el diseño de la propuesta del sistema y qué representa cada cual:



sp_<NombreClaseServidora>

<<Server Page>>:

Representa la clase que tiene código que se ejecuta en el servidor, la cual se encarga de construir (build) o generar el resultado HTML y/o realizar peticiones a la capa inferior.

Capítulo 3. Análisis y Diseño del Sistema



cl_<NombreClaseCliente>

<<Client Page>>:

Es una página Web con formato XHTML. Mezcla de datos, presentación y lógica. Son interpretadas por el navegador. Sus atributos son las variables declaradas dentro del script que son accesibles para cualquier función dentro de la página. Cada página cliente es construida por una sola página de servidor.



form_<NombreFormulario>

<<FormHTML>>:

Es una colección de elementos de entrada que están contenidos en la página cliente. Sus atributos son los elementos de entrada del formulario (input boxes, text areas, radio buttons, check boxes, hidden fields, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST y se comunican con las páginas servidores mediante submit.

3.3.2 Diagrama de Clases del Diseño.

Un diagrama de clases del diseño se utiliza para modelar la vista de diseño estática de un sistema. Estos contienen clases, interfaces, colaboraciones y relaciones de dependencia, generalizaciones y asociación. Son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa. Brindan además un mayor acercamiento a la forma y al contenido de la aplicación web propuesta, por lo que para realizar estos diagramas de clases se tuvo en cuenta la extensión UML para Web.

En el diagrama de clases del diseño se muestra la estructura estática del sistema, se crea el diseño conceptual de la información que se manejará en el sistema, así como los componentes que se encargarán del funcionamiento y la relación entre ellos. ([Ver Anexo 1](#))

Casos de Uso arquitectónicamente significativos:

- ✓ Autenticar Usuario.
- ✓ Registrar Especialista.
- ✓ Modificar Especialista.

Capítulo 3. Análisis y Diseño del Sistema

- ✓ Buscar Especialista.
- ✓ Listar Especialista.
- ✓ Publicar Test.
- ✓ Obtener Reporte Caracterización Psicológica.
- ✓ Realizar Test.

3.3.3 Diagrama de Interacción (Secuencia)

Los diagramas de secuencia y los diagramas de colaboración (ambos llamados diagramas de interacción) son dos de los cinco tipos de diagramas de UML que se utilizan para modelar los aspectos dinámicos de los sistemas. Un diagrama de interacción consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.

Los diagramas de secuencia son de gran importancia en el diseño de un sistema debido a que permiten observar las interacciones que ocurren entre los distintos objetos que participan en un escenario determinado. ([Ver Anexo 2](#))

3.4 Descripción de las Clases del Diseño

3.4.1 Descripción de Clases Controladoras

Nombre: CC_Usuario	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Buscar_Especialista()
Descripción:	Busca los especialistas registrados en el sistema.
Nombre:	Registrar_Especialista()
Descripción:	Registra especialistas en el sistema.

Capítulo 3. Análisis y Diseño del Sistema

Nombre:	Modificar ()
Descripción:	Modifica datos de los especialistas.
Nombre:	Listar_Especialista()
Descripción:	Lista todos los especialistas que se encuentran en el sistema.
Nombre:	Autenticar_usuario()
Descripción:	Utilizado para que el usuario se autentique.
Nombre:	Ver_usuario()
Descripción:	Ver los datos del usuario.
Nombre:	Cargar_registrar()
Descripción:	Carga la vista con los datos que debe registrar el usuario.
Nombre:	Eliminar()
Descripción:	Elimina el usuario del sistema.
Nombre:	VerModificartest()
Descripción:	Cargar la vista para modificar la clave de los test.
Nombre:	VerMoficar()
Descripción:	Ver los datos a modificar del usuario seleccionado.
Nombre:	Listar_Test()
Descripción:	Lista todos los test que existen en el sistema.

Capítulo 3. Análisis y Diseño del Sistema

Nombre:	Modificar_clave()
Descripción:	Modifica la contraseña de los test.
Nombre:	Modificar_evaluacion()
Descripción:	Modifica la evaluación del especialista en caso de que se desee agregar una mejor descripción del resultado obtenido.
Nombre:	ver()
Descripción:	Muestra los resultados de la búsqueda según los criterios insertados.
Nombre:	Ver_resultados()
Descripción:	Muestra los resultados obtenidos de los test realizados por el especialista seleccionado.

Nombre: CC_Test_Psicologico

Tipo de clase: Controladora

Para cada responsabilidad:

Nombre:	Publicar_Test()
Descripción:	Publica el test para que el especialista pueda realizarlo.
Nombre:	Mostrar_usuario()
Descripción:	Muestra la pagina principal del usuario.
Nombre:	Mostrar_administrador()

Capítulo 3. Análisis y Diseño del Sistema

Descripción:	Muestra la pagina principal del administrador.
Nombre:	Cargar_des()
Descripción:	Carga las preguntas del test Inventario de estilos decisionales.
Nombre:	Cargar_belb()
Descripción:	Carga las preguntas del test Belbin.
Nombre:	Cargar_cogv()
Descripción:	Carga las preguntas del test Estilos Cognitivos.
Nombre:	Cargar_leyes()
Descripción:	Carga las preguntas del test Las leyes.
Nombre:	Cargar_matrizFD()
Descripción:	Carga las preguntas del test Matriz de Fortalezas y Debilidades.
Nombre:	Guardar_respuesta_belbin()
Descripción:	Guarda en la base de datos las respuestas dadas al test de Autopercepción de Belbin.
Nombre:	Guardar_respuesta_cognitivos()
Descripción:	Guarda en la base de datos las respuestas dadas al test Estilos Cognitivos.
Nombre:	Guardar_respuesta_decisionales()
Descripción:	Guarda en la base de datos las respuestas dadas al test de Inventario de Estilos Decisionales.

Capítulo 3. Análisis y Diseño del Sistema

Nombre:	Guardar_MatrisFD()
Descripción:	Guarda en la base de datos las respuestas dadas al test Matriz de Fortalezas y debilidades.
Nombre:	Tabular_belbin()
Descripción:	Tabula las respuestas del test Belbin.
Nombre:	Tabular_decisionales()
Descripción:	Tabula las respuestas del test Inventario de estilos decisionales.
Nombre:	Tabular_leyes()
Descripción:	Tabula las respuestas del test Las leyes.
Nombre:	Tabular_cognitivos()
Descripción:	Tabula las respuestas del test Estilos Cognitivos.
Nombre:	Tabular_matrizFD()
Descripción:	Tabula las respuestas del test Matriz de Fortalezas y Debilidades.
Nombre:	Validar_password()
Descripción:	Valida que la contraseña insertada para acceder al test es correcta.

3.4.2 Descripción de las Clases Entidades

Nombre: rol	
Tipo de clase: Entidad	
Atributo	Tipo

Capítulo 3. Análisis y Diseño del Sistema

id	int
valor	varchar

Nombre: especialista	
Tipo de clase: Entidad	
Atributo	Tipo
id	int
id_rol	int
nombre	varchar
apellidos	varchar
ci	varchar
centro	varchar
usuario	varchar
clave	varchar

Nombre: respuestas_especialista_test	
Tipo de clase: Entidad	
Atributo	Tipo
id	int
id_especialista	int
respuestas	varchar
id_preguntas_test	int
respuestas_aux	varchar

Nombre: test	
Tipo de clase: Entidad	
Atributo	Tipo
id	int
nombre	varchar

Capítulo 3. Análisis y Diseño del Sistema

activado	tinyint
----------	---------

Nombre: contenido_test	
Tipo de clase: Entidad	
Atributo	Tipo
id	int
id_test	int
enunciado	varchar

Nombre: preguntas_test	
Tipo de clase: Entidad	
Atributo	Tipo
id	int
id_contenido_test	int
valor	varchar

Nombre: especialista_evaluacion	
Tipo de clase: Entidad	
Atributo	Tipo
id	int
id_especialista	int
id_test	int
resultado	varchar

Nombre: mensaje_test	
Tipo de clase: Entidad	
Atributo	Tipo
id	int
id_test	int
mensaje	varchar

Capítulo 3. Análisis y Diseño del Sistema

En este capítulo se ha completado el flujo de análisis y diseño con la realización de los diagramas de clases del análisis, los diagramas de clases de diseño y los diagramas de secuencia de los casos de uso del sistema, lo que proporciona una idea completa del software, materializando con precisión los requerimientos. Los diagramas y especificaciones de diseño que se proponen constituyen una guía que puede ser fácilmente leída y comprendida por los desarrolladores del sistema.

Capítulo 4. Implementación

El propósito fundamental de este capítulo es definir cómo desarrollar la arquitectura prevista durante el diseño, representando el sistema como un todo. Se implementan las clases definidas en el capítulo anterior en términos de componentes. Se modelan los diagramas de componentes y de despliegue, quedando conformado el modelo de implementación del sistema, dando una visión de cómo quedará construida y distribuida la aplicación. Además, se proporciona una detallada explicación de los estándares de diseño, codificación, tratamiento de errores a utilizar, y se construye el Modelo de datos y se describen cada una de las tablas que lo componen.

4.1 Modelo de datos

Un modelo de datos es un conjunto de conceptos, reglas y convenciones que permiten describir, y manipular los datos de un cierto mundo real que se desea almacenar en la base de datos. Está formado por dos componentes, las propiedades estática y dinámica. La parte estática se refiere a la estructura y la dinámica a las operaciones que se pueden realizar sobre cada objeto.

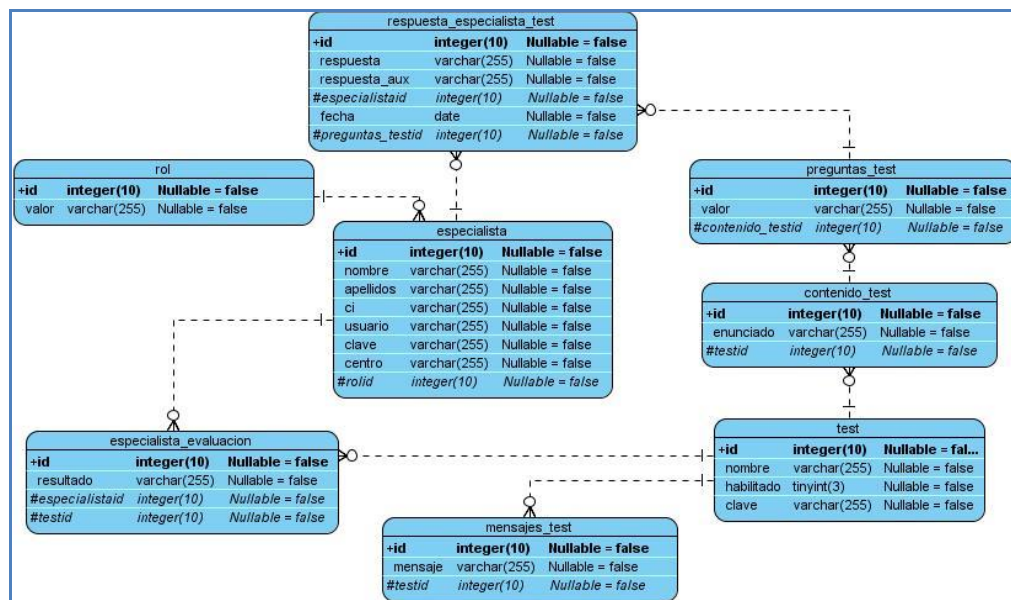


Fig. 4.1 Modelo de Datos

Capítulo 4. Implementación

4.1.1 Descripción de las tablas

Nombre: rol		
Descripción: Tabla que guarda los roles que tendrá la aplicación.		
Atributo	Tipo	Descripción
id	int	Identificador de un rol.
valor	varchar	Rol que posee el usuario en la aplicación.

Nombre: especialista		
Descripción: Tabla que guarda los datos de un especialista.		
Atributo	Tipo	Descripción
id	int	Identificador del especialista
nombre	varchar	Nombre del especialista.
apellidos	varchar	Apellidos del especialista.
ci	varchar	Carnet de Identidad del especialista
usuario	varchar	Usuario del especialista.
clave	varchar	Clave del especialista.
centro	varchar	Centro o ubicación laboral del especialista.

Nombre: respuestas_especialista_test		
Descripción: Tabla que almacena las respuestas de los test realizados por los especialistas.		
Atributo	Tipo	Descripción
id	int	Identificador de la respuesta
id_especialista	int	Identificador del especialista
respuestas	varchar	Valor de la respuesta
id_preguntas_test	int	Identificador de la pregunta
respuestas_aux	varchar	Valor de la respuesta

Nombre: test		
Descripción: Tabla que almacena el nombre y estado del test.		

Capítulo 4. Implementación

Atributo	Tipo	Descripción
id	int	Identificador del test
nombre	varchar	Nombre del test
activado	tinyint	Estado del test

Nombre: contenido_test

Descripción: Tabla que almacena el enunciado de las preguntas de los test.

Atributo	Tipo	Descripción
id	int	Identificador del contenido del test
id_test	int	Identificador del test
enunciado	varchar	Enunciado de la pregunta del test

Nombre: preguntas_test

Descripción: Tabla que almacena los incisos de las preguntas.

Atributo	Tipo	Descripción
id	int	Identificador de la pregunta.
id_contenido_test	int	Identificador del contenido del test.
valor	varchar	Inciso de la pregunta.

Nombre: especialista_evaluacion

Descripción: Tabla que almacena los resultados de la evaluación del especialista luego de la tabulación de los test.

Atributo	Tipo	Descripción
id	int	Identificador de la evaluación del especialista.
id_especialista	int	Identificador del especialista.
id_test	int	Identificador del test.
resultado	varchar	Resultado de la tabulación de los test.

Nombre: mensaje_test

Descripción: Tabla que almacena los mensajes para dar en el reporte de caracterización.

Capítulo 4. Implementación

Atributo	Tipo	Descripción
id	int	Identificador del mensaje
id_test	int	Identificador del test.
mensaje	varchar	Mensaje del reporte de caracterización por cada uno de los test.

4.2 Modelo de Implementación

El modelo de implementación describe cómo los elementos del Modelo de diseño, serán implementados en términos de componentes, describe además cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, así como la dependencia que se establece entre los componentes. [17]

4.2.1 Diagrama de Despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos. Los nodos representan recursos de cómputos: procesadores o dispositivos de hardware. [18]

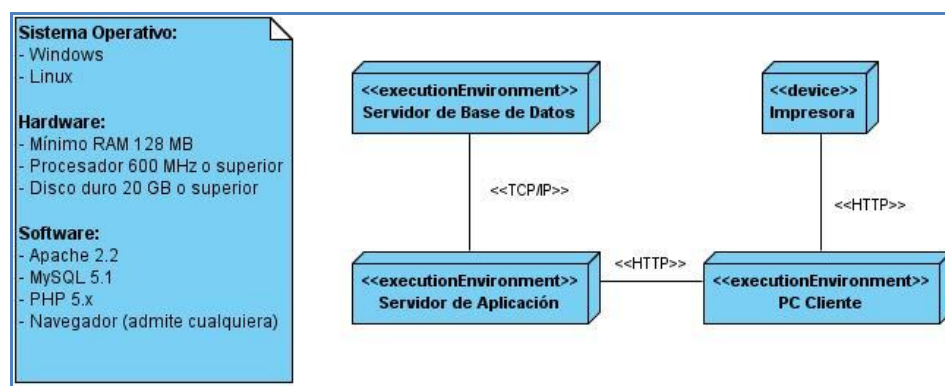


Fig. 4.2 Diagrama de Despliegue

4.2.2 Diagrama de Componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software, sean éstos de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de

Capítulo 4. Implementación

componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. [19]

El sistema cuenta con tres subsistemas de implementación: Vistas, Controladoras y Modelo; estos están estructurados según el patrón arquitectónico MVC. En el subsistema Vistas se encuentran encapsulados los componentes que permiten la interacción directa con los usuarios del sistema. El subsistema Controladoras contiene todas las clases que manipulan los eventos del usuario y realizan peticiones al modelo para mostrarlas en las vistas. El subsistema Modelo, agrupa las clases que interactúan con la base de datos y velan por el cumplimiento de las reglas del negocio.

A continuación son expuestos los Diagramas de Componentes asociados a los subsistemas de implementación identificados. Siguiendo la arquitectura, la estructuración en subsistemas de implementación es la siguiente:

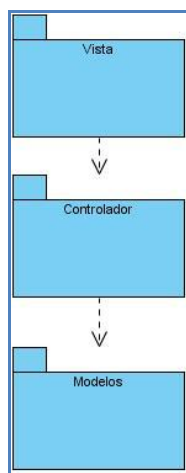


Fig. 4.3 Subsistema de Implementación

Diagrama de Componentes del Subsistema: Vistas

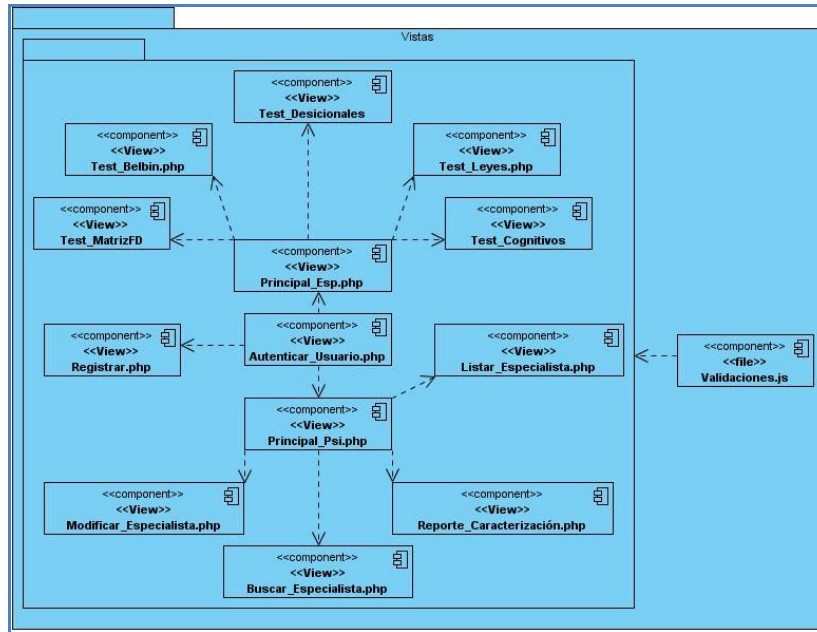


Fig. 4.4 Diagrama de Componentes del Subsistema: Vistas

Diagrama de Componentes del Subsistema: Controlador

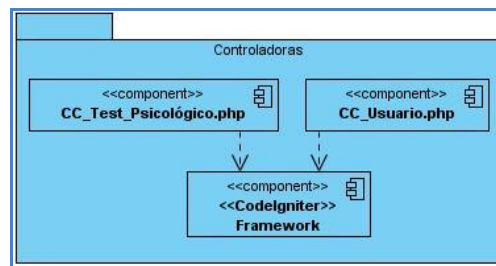


Fig. 4.5 Diagrama de Componentes del Subsistema: Controlador

Diagrama de Componentes del Subsistema: Modelo

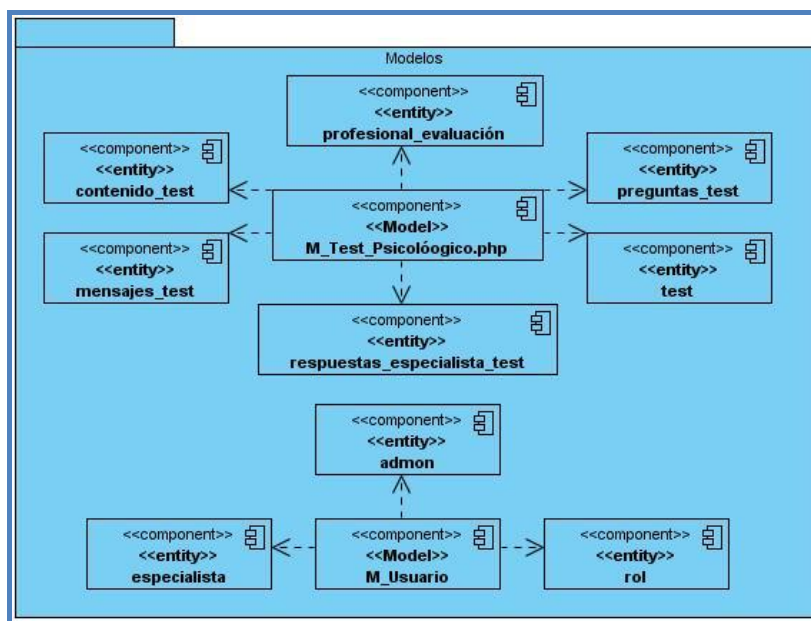


Fig. 4.6 Diagrama de Componentes del Subsistema: Modelo

En el siguiente diagrama se ilustran las relaciones entre los componentes a modo general, pudiéndose apreciar de forma práctica como se implementa el patrón arquitectónico MVC.

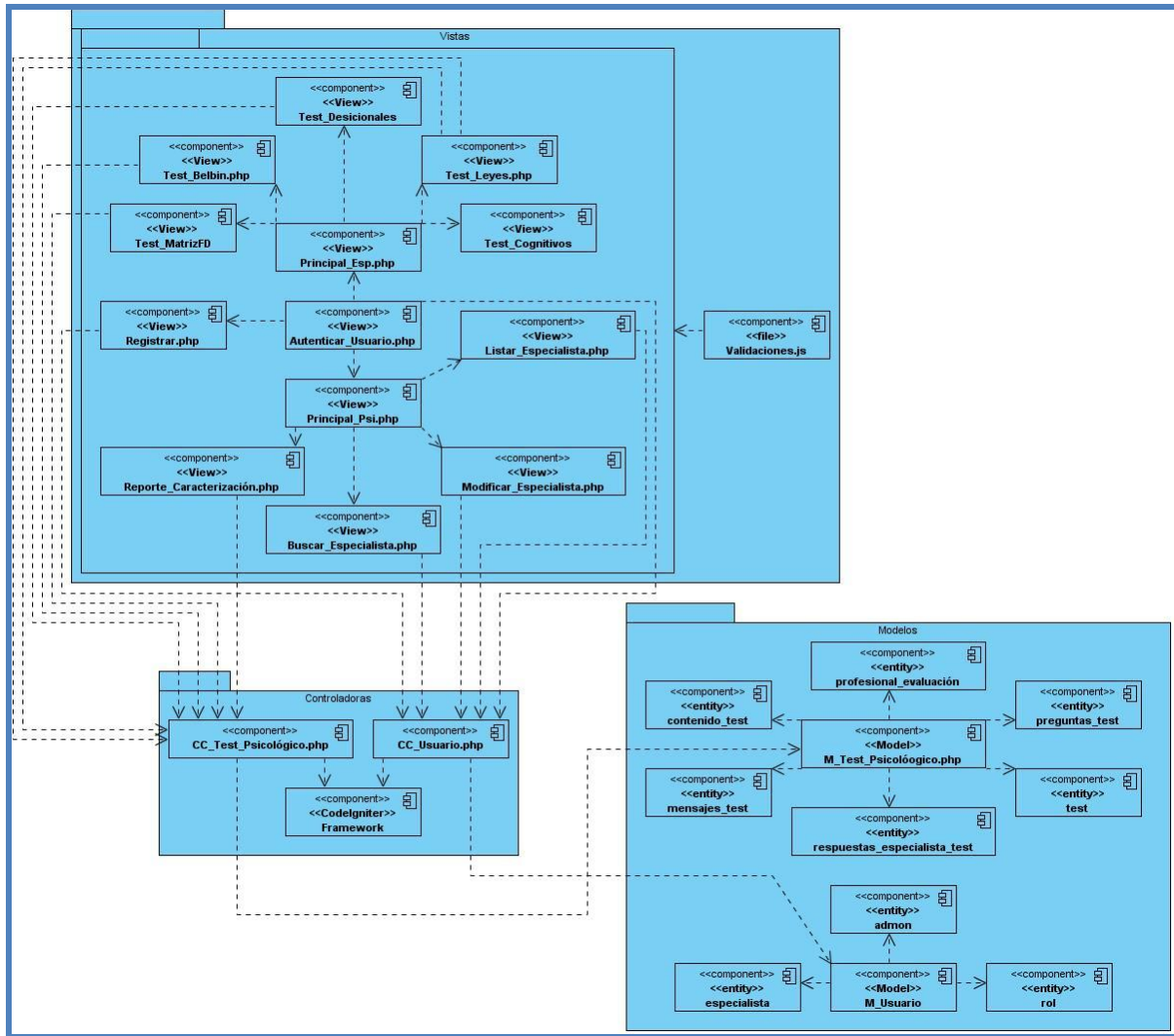


Fig. 4.7 Diagrama de Componentes General

4.3 Tratamiento de errores

En las funcionalidades se tiene en cuenta el tratamiento de todos los posibles errores que puedan aparecer, para garantizar así la integridad y confiabilidad de los datos.

Las funcionalidades también muestran mensajes de confirmación, que son de fácil comprensión para el usuario. Se pueden observar cuando se insertan datos y no se entran los campos que son obligatorios, cuando se registra un paciente que ya existe en la base de datos y cuando se desean modificar los datos

Capítulo 4. Implementación

de un usuario. Se realizan validaciones en el código para la consistencia de los datos entrados por el usuario.

Para depurar los errores se utiliza JavaScript, validando los formularios y evitando consultas a la base de datos sin sentido o que tenga malas intenciones como son las inyecciones SQL, garantizando además que los datos introducidos por los usuarios sean válidos, o les sea posible corregirlos en caso contrario.

4.4 Seguridad

La seguridad de un sistema es de suma importancia para preservar la integridad, autenticidad y confiabilidad de la información. Para el acceso a las funcionalidades del sistema se crearon usuarios con diferentes roles que garantizan niveles de acceso jerárquicos dentro de la aplicación. Cada usuario podrá acceder a las funcionalidades definidas en dependencia del rol que desempeñe dentro del sistema.

4.5 Estrategias de Codificación. Estándares y estilos a utilizar.

Idioma: Se debe utilizar como idioma el español, las palabras no se acentuarán.

Palabras Reservadas: Las palabras reservadas van en minúsculas sin excepción alguna.

Identación		
Objetivos: Lograr una estructura uniforme para los bloques de código así como para los diferentes niveles de anidamiento.		
0 espacios del tabulador desde la izquierda en	require include class	No se empleará ningún espacio del tabulador desde la izquierda para las instrucciones antes mencionadas. Se tomará como inicio de la página la etiqueta PHP <?
1 espacio del tabulador desde la izquierda en	function	Se dejará un espacio del tabulador desde la izquierda en las instrucciones antes mencionadas.
1 espacio del tabulador desde la izquierda en	Inicio y fin de bloque	Se recomienda dejar un espacio del tabulador desde la instrucción anterior para el inicio y fin de

Capítulo 4. Implementación

		bloque {}}. Lo mismo sucede para el caso de las instrucciones If, else, For, While, Do While, Switch, Foreach.
Niveles de anidación	Hasta 5 niveles	Se recomienda emplear hasta 5 niveles de anidación en instrucciones If, For, While.
Aspectos Generales	Los inicios ({} y cierre (}) de ámbito deben estar alineados debajo de la declaración a la que pertenecen y deben evitarse si hay sólo una instrucción. Nunca colocar { en la línea de un código cualquiera, esto requiere una línea propia.	
Comentarios, separadores, líneas, espacios en blanco y márgenes.		
Objetivo: Crear una forma común de comentar el código que sea comprensible con tan sólo leerlo una vez.		
Ubicación de comentarios	Al inicio de cada clase o función.	Se recomienda comentar al inicio de cada clase o función expresando su objetivo.
Separador de instrucciones	Se utiliza el punto y coma (;)	Se recomienda el uso del separador al final de cada línea de instrucción. Ejemplo: \$cadenaTexto = "ejemplo";
Líneas en blanco	Se utilizan antes de una clase o función.	Se recomienda dejar una línea en blanco antes y después de la declaración de una clase o función.
Espacios en blanco	Entre operadores lógicos y aritméticos.	Se recomienda dejar un espacio en blanco entre los operadores para lograr mayor comprensión del código.
Márgenes y líneas de continuidad	Sobre márgenes y líneas de continuidad.	Los márgenes de cada línea de código no deben exceder los 80 caracteres, pero puede exceptuarse si es para terminar la escritura de una palabra. Las líneas de continuación deben estar alineadas entre sí e indentadas respecto al paréntesis abierto.

Capítulo 4. Implementación

Aspectos Generales	Sobre los comentarios	Se debe evitar comentar cada línea de código. Cuando el comentario se aplica a un grupo de instrucciones debe estar seguido de una línea en blanco. En caso de que se necesite comentar una sola instrucción se suprime la línea en blanco o se escribe a continuación de la instrucción.
	Sobre espacios en blanco	No se debe usar espacio en blanco: Después del corchete abierto y antes del cerrado de un arreglo. Después del paréntesis abierto y antes del cerrado.
VARIABLES		
Apariencia de variables	Tendrán un nombre descriptivo.	Los nombres de las variables deberán empezar con minúscula, en caso de que sea un nombre compuesto se pone junto y con mayúscula inicial el segundo nombre. El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma. Ejemplo: \$horaConsulta;
CLASES		
Objetivo: Nombrar las clases e instancias de forma estándar.		
Para nombrar clases.	Se utilizarán nombres descriptivos.	Los nombres se escribirán en minúsculas. Para diferenciar las clases modelos, sus nombres terminarán en “_model”.
Apariencia de clases y objetos.	Se nombrarán en minúsculas.	Los nombres de las clases se escribirán en minúsculas. En caso de nombres compuestos se escriben juntos y con mayúscula inicial el segundo nombre.
Apariencia de atributos.	Se escribirán en minúsculas.	El nombre que se le da a los atributos de las clases debe ser en minúsculas. En caso de nombres complejos se escribirán juntos y con mayúscula inicial el segundo nombre.

Capítulo 4. Implementación

Apariencia de las funciones.	Se escribirán en minúsculas.	Para nombrar las funciones se debe tratar de utilizar verbos que denoten la acción que hace la función. Si son funciones que obtienen un dato se emplea el prefijo get.
Declaración de parámetro en funciones.	Se deben agrupar por tipos. Si alguno puede tomar valor nulo se pone al final.	Los parámetros que se le pasan a las funciones se recomienda sean declarados de forma tal que estén agrupados por el tipo de dato que contienen.
Aspectos Generales	Sobre las clases, los objetos, los atributos y las funciones.	Los nombres deben ser descriptivos con el propósito de conocer su objetivo con tan sólo leerlos.

A modo de conclusión, en este capítulo se ha completado el flujo de implementación y despliegue con la realización del modelo de datos y la descripción de sus principales tablas, los diagramas de componentes y el diagrama de despliegue. Además se proporciona una idea completa de como se lleva a cabo la seguridad del sistema; así como el tratamiento de errores, los estándares y estilos a utilizar como estrategias de codificación.

Conclusiones

Al culminar el presente trabajo de diploma se ha realizado un estudio del proceso de caracterización psicológica que se realiza actualmente en la empresa Albet Ingeniería y Sistemas. Para satisfacer la necesidad de mejorar dicho proceso y valorando las tareas y objetivos propuestos, se llegó a las siguientes conclusiones:

- ✓ El estudio de sistemas similares existentes a nivel nacional e internacional utilizados para apoyar el proceso de caracterización psicológica permitió demostrar que no cumplen con las necesidades de automatización del proceso llevado a cabo por Albet.
- ✓ Se modelaron los flujos de trabajo propuestos por el Proceso Unificado de Desarrollo: Modelado del Negocio, Gestión de Requerimientos, Diseño e Implementación. Obteniéndose los artefactos que se generan en cada uno de estos flujos, para ello se utilizaron como Metodología de Desarrollo de Software el Proceso Unificado de Desarrollo (RUP), el Lenguaje Unificado de Modelado (UML), y Visual Paradigm for UML 6.0 como herramienta CASE.
- ✓ Se realizó la implementación de las funcionalidades propuestas utilizándose las herramientas y tecnologías definidas para el desarrollo de la aplicación como son: el gestor de base de datos MySQL, el lenguaje de programación PHP, el entorno de desarrollo integrado phpDesigner 6.0 y el framework CodeIgniter.
- ✓ Se desarrolló una Aplicación para la gestión dinámica de los test psicológicos utilizados por la empresa Albet para la selección de especialistas que den como resultado una caracterización psicológica.

Recomendaciones

Se recomienda que una vez entregado el sistema al grupo de informatización de Albet:

- Darle continuidad a la investigación desarrollada para su mejoramiento continuo y desarrollar en el futuro un nuevo módulo que permita la incorporación de nuevos test con sus tabulaciones para enriquecer el proceso de caracterización.
- Continuar dando soporte y mantenimiento al sistema.

Referencias Bibliográficas

Referencias Bibliográficas

1. **Portal de Albet.** *Portal de Albet.* [En línea] 2008. [Citado el: 22 de noviembre de 2009.] <http://www.albetsa.com/?q=es/node/299>.
2. **Pacheco, Angel Enrique.** La Psicometría y los test psicológicos. Adaptado de Anne Anastasi. *Instituto Dr. Pacheco de Psicología.* [En línea] 2003 - 2005. [Citado el: 16 de enero de 2010.] <http://www.idpp.org/>.
3. **Dirección de RR. HH. (MIC), Dirección de desarrollo (FORDES).** abril 2009 - junio 2009. *Programa de caraterización psicológica.* abril 2009 - junio 2009.
4. **Clarkin y Hurt S.** Test y escalas de valoración. Evaluación Psicológica. Tratado de Psiquiatría editado por Talbott Ancora. Barcelona 1989. Pág. 223 – 244.
5. **Portal Psycowin.** [En línea] [Citado el: 20 de enero de 2010.] <http://www.psycowin.com/Competencias.aspx>.
6. **Portal Belbin.** [En línea] [Citado el: 16 de enero de 2010.] <http://www.belbin.com/rte.asp?id=128>.
7. Ídem a la 6
8. Ídem a la 6
9. Ídem a la 6
10. **Portal HUM&SOFTWARE.** [En línea] [Citado el: 17 de enero de 2010.] <http://www.humsoftware.com/Paginas/Sel-Index.htm>.
11. **ExpoMatanzas.** [En línea] [Citado el: 20 de enero de 2010.] <http://www.expomatanzas.cu/empresa.php?prd=53&emp=1&catprd=1>.
12. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* Madrid: s.n., 2000.

Referencias Bibliográficas

13. **Schumuller, Joseph.** *Aprendiendo UML en 24 Horas.* 2000.
14. **Visual Paradigm.** [En línea] 2008. [Citado el: 10 de diciembre de 2009.] <http://www.visual-paradigm.com/product/vpuml/>.
15. **Cavsi.** [En línea] [Citado el: 11 de diciembre de 2009.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sqbd/>
16. Ídem a la 15.
17. **UCI.** Flujo de trabajo Análisis & Diseño (Modelo de diseño). Ciudad de la Habana: s.n., 2006.
18. **Jacobson, I.; Booch, G. y Rumbaugh, J.;** “El Proceso Unificado de Desarrollo de software”. 2000.
19. Ídem a la referencia 17
20. **Diagrama de Componentes** [En línea] [Citado el: 28 de Marzo del 2010] <http://tvdi.det.uvigo.es/~avilas/UML/node49.html>

Bibliografía

1. **Arquitectura de Software.** Ventajas del MVC. [Citado el: 20 de marzo de 2010.]
http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rivera_l_a/capitulo2.pdf
2. **Cavsi.** [En línea] [Citado el: 11 de diciembre de 2009.]
<http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>
3. **Dirección de RR. HH. (MIC), Dirección de desarrollo (FORDES).** abril 2009 - junio 2009.
Programa de caracterización psicológica. abril 2009 - junio 2009.
4. **ExpoMatanzas.** [En línea] [Citado el: 20 de enero de 2010.]
<http://www.expomatanzas.cu/empresa.php?prd=53&emp=1&catprd=1>.
5. **Gracia, Joaquín.** Diagramas UML. [En línea] 2005. [Citado el: 21 de enero de 2010.]
<http://www.ingenierossoftware.com/analisisydiseno/uml.php>
6. **Macías Charlie, Orozco Sergio.** Uso de UML en aplicaciones Web. [En línea] 2000. [Citado el: 17 de enero de 2010.]
7. **Ingeniería de Software 1.** Fase de Inicio. Modelo del Negocio. UCI. curso 2008_2009.
Conferencia # 2.
8. **Ingeniería de Software 2.** “Continuación del FT Análisis y Diseño. Modelo de Diseño.” UCI. curso 2008_2009. Conferencia # 1.
9. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* Madrid: s.n., 2000.
10. **Pacheco, Angel Enrique.** La Psicometría y los test psicológicos. Adaptado de Anne Anastasi. *Instituto Dr. Pacheco de Psicología.* [En línea] 2003 - 2005. [Citado el: 16 de enero de 2010.]
<http://www.idpp.org/>.
11. **Portal Belbin.** [En línea] [Citado el: 27 de enero de 2010.]

<http://www.belbin.com/rte.asp?id=262>

12. **Portal de Albet.** [En línea] 2008. [Citado el: 22 de noviembre de 2009.]

<http://www.albetsa.com/?q=es/node/299>

13. **Portal Desoft.** Selsoft. [En línea] 2000. [Citado el: 22 de enero de 2010.]

<http://www.desoft.cu/Productos1/Selsoft/tabid/435/Default.aspx?PageContentID=59>

14. **Portal Diagrama de despliegue** [En línea] [Citado el : 10 de abril del 2010]

<http://virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc>

15. **Portal Hum&Software.** [En línea] [Citado el: 25 de enero de 2010.]

<http://www.humsoftware.com/Paginas/Sel-Index.htm>

16. **Portal Psycowin.** [En línea] [Citado el: 24 de enero de 2010.]

<http://www.psycowin.com/>

17. **Schumuller, Joseph.** *Aprendiendo UML en 24 Horas.* 2000.

18. **Visual Paradigm.** [En línea] 2008. [Citado el: 10 de diciembre de 2009.] <http://www.visual-paradigm.com/product/vpum/>.

Glosario de Términos

Apache: es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1.

Aplicación Web: especialización de las aplicaciones Cliente/Servidor, están comúnmente estructuradas como una aplicación en tres capas, son conocidas como aplicaciones servidor. Arquitectura Cliente/Servidor: es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en elementos independientes que cooperan entre sí para intercambiar información, servicios o recursos.

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

CodeIgniter: es un Framework para desarrollo de aplicaciones en PHP.

Casos de Usos (CU): es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

Estándar de codificación: Es el conjunto de reglas o normas usadas para escribir código y que incluye una gran gama de aspectos dentro del proceso de codificación, logrando que los programas sean más robustos y comprensibles.

HTML: HyperText Markup Language/Lenguaje Hipertexto de Marcado. Lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986. Es un lenguaje de marcas. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII.

HTTP: HyperText Transfer Protocol/ Protocolo de Transferencia de Hipertextos. Modo de comunicación para solicitar páginas Web.

Glosario de Términos

Hardware: Componentes electrónicos, tarjetas, periféricos y equipo que conforman un sistema de computación; se distinguen de los programas (software) porque son tangibles.

Informática: disciplina que estudia el tratamiento automático de la información utilizando dispositivos electrónicos y sistemas computacionales.

Internet: Sistema de redes de computación ligadas entre sí, con alcance mundial, que facilita servicios de comunicación de datos como registro remoto, transferencia de archivos, correo electrónico y grupos de noticias. Internet es una forma de conectar las redes de computación existentes que amplía en gran medida el alcance de cada sistema participante.

MySQL: Es un sistema de gestión de bases de datos relacional que cuentan con todas las características de un motor de BD comercial: transacciones atómicas, triggers, replicación, llaves foráneas entre otras. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar.

MVC: Modelo Vista Controlador.

PHP: Hypertext Preprocessor/Preprocesador de Hipertexto. Es un lenguaje script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas. Con PHP se pueden combinar páginas HTML y scripts. Con el objetivo de crear aplicaciones potentes.

RUP: Rational Unified Process (Proceso Unificado de Desarrollo). Metodología para el desarrollo de Software.

Software: es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.

Sitio Web: es un conjunto de páginas web, típicamente comunes a un dominio de Internet o subdominio en la World Wide Web en Internet.

SGBD: Sistema de Gestión de Bases de Datos. Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.

Glosario de Términos

Test Psicológico: es una medida objetiva y normalizada de una muestra de conducta. Son instrumentos especializados que sirven de apoyo al psicólogo al medir conductas específicas.

UML: Unified Modeling Language/Lenguaje Unificado de Modelado. Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.

WEB (WWW): red de documentos HTML intercomunicados y distribuidos entre servidores del mundo entero.

Anexos

Anexo 1. Diagrama de Clases del Diseño

Diagrama de Clases del Diseño del CU *Autenticar Usuario*

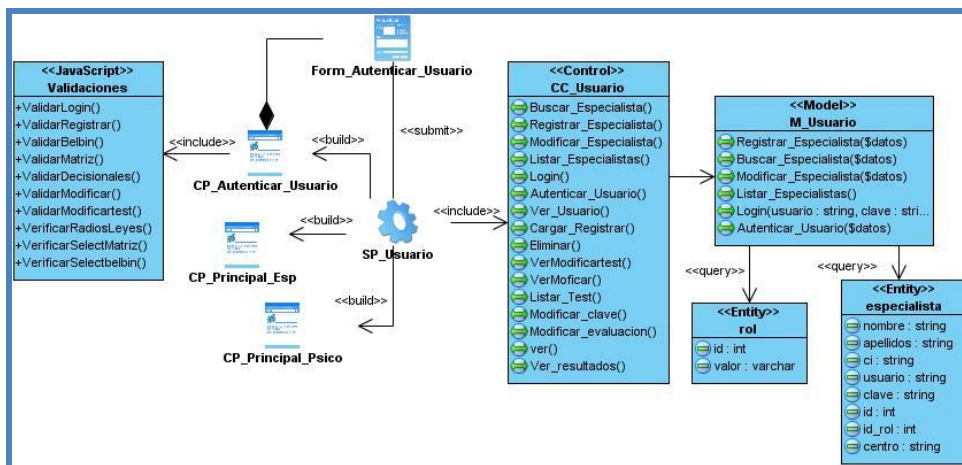


Diagrama de Clases del Diseño del CU *Registrar Especialista*

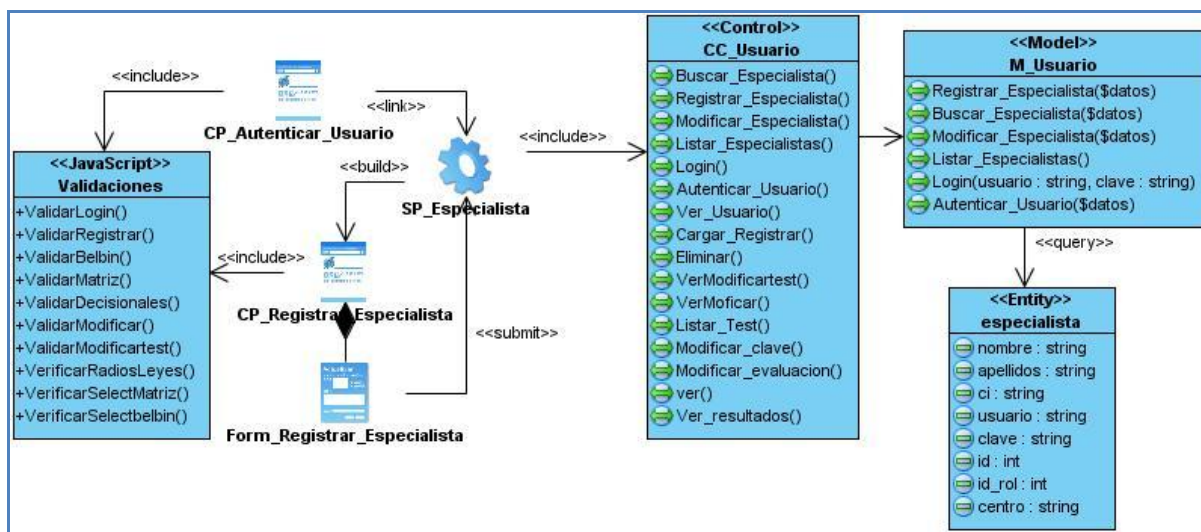


Diagrama de Clases del Diseño del CU **Modificar Especialista**

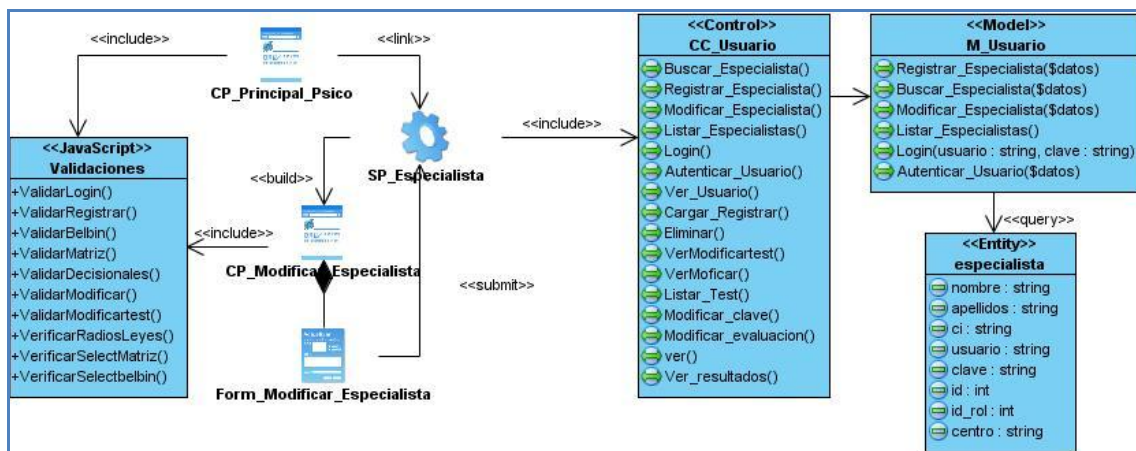


Diagrama de Clases del Diseño del CU **Obtener Reporte Caracterización Psicológica**

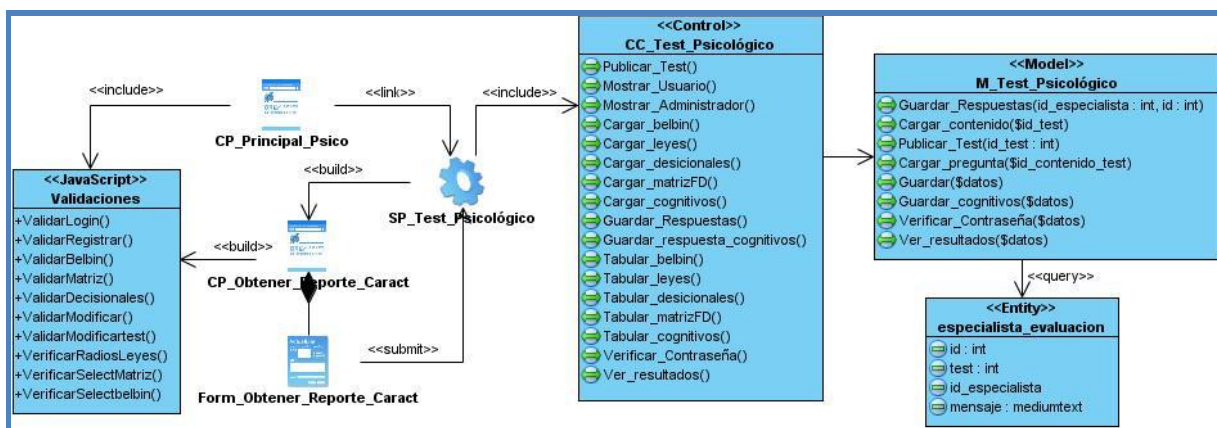
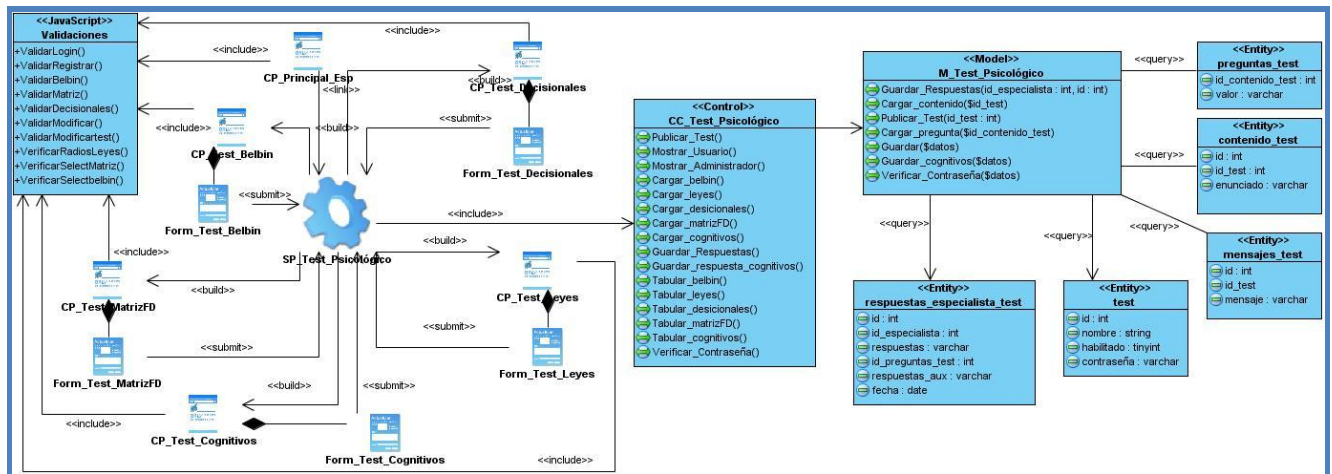


Diagrama de Clases del Diseño del CU **Realizar Test**



Anexo 2. Diagrama de Secuencia

Diagrama de Secuencia del CU *Autenticar Usuario*

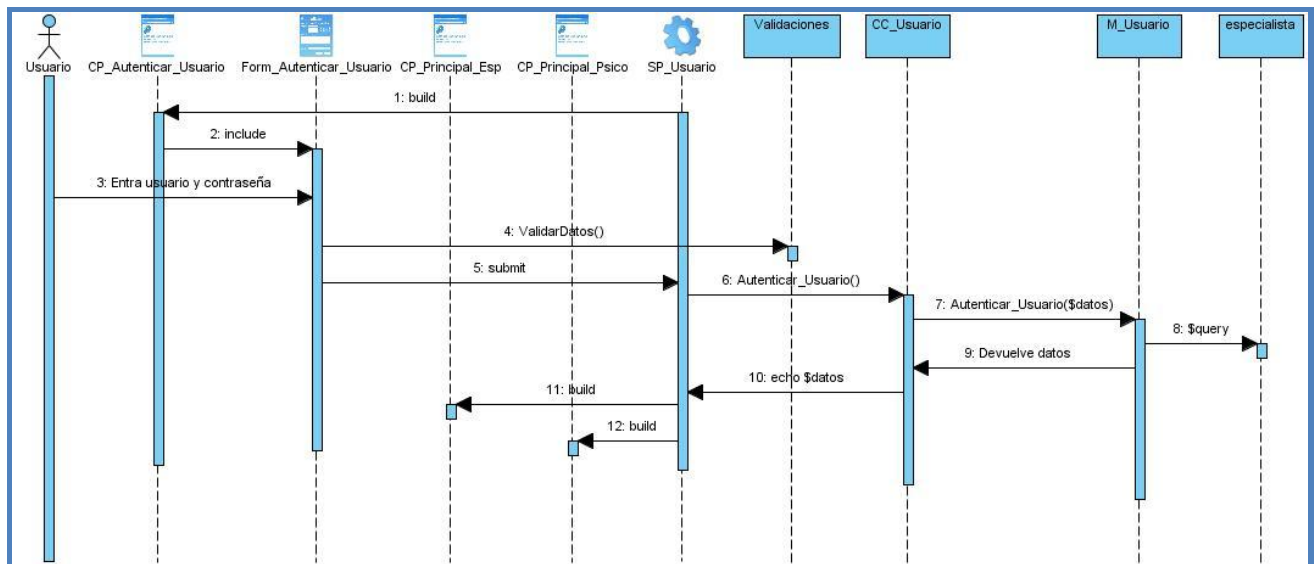


Diagrama de Secuencia del CU **Registrar Especialista**

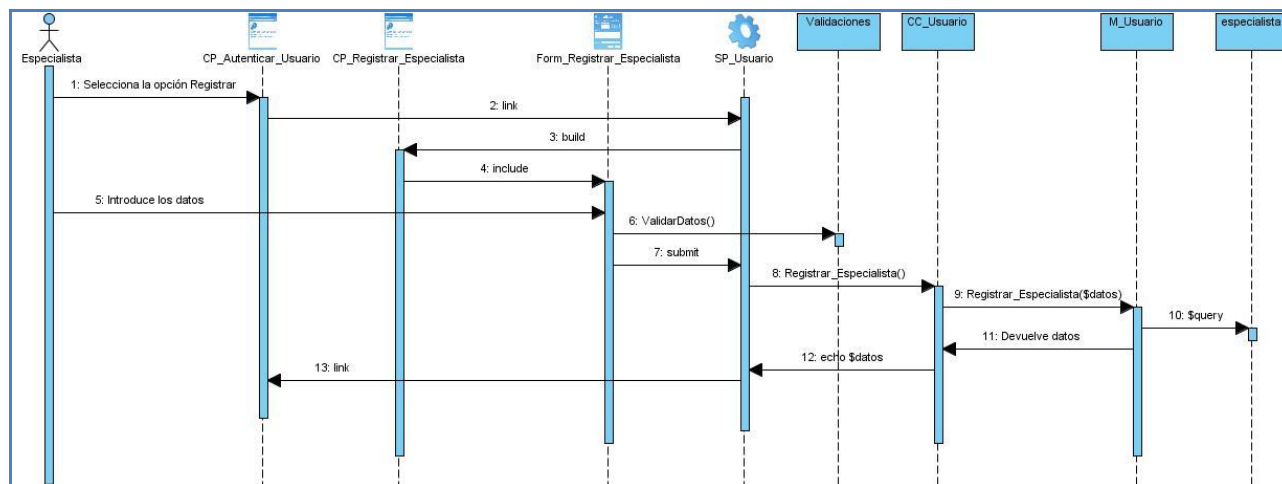


Diagrama de Secuencia del CU **Modificar Especialista**

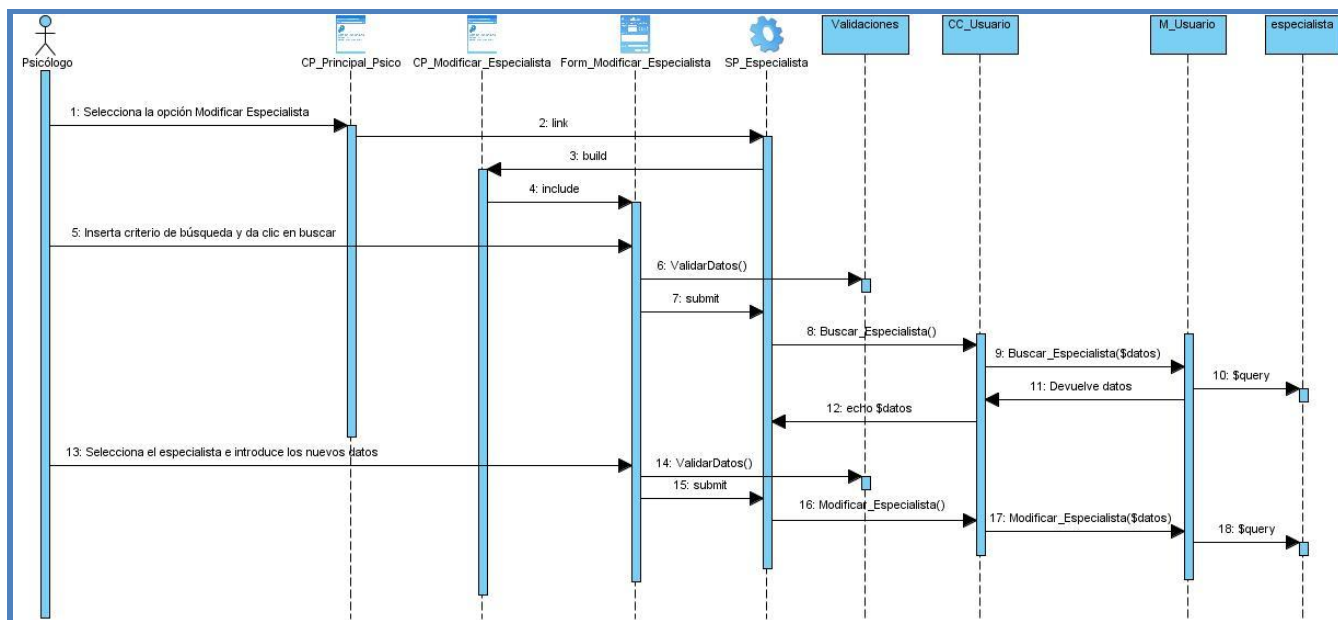


Diagrama de Secuencia del CU *Obtener Reporte Caracterización Psicológica*

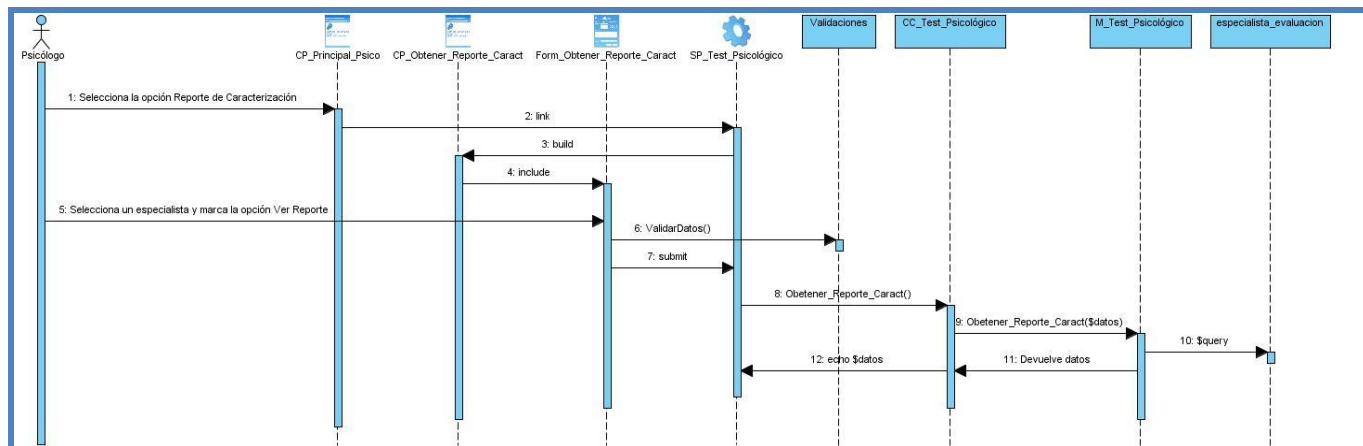


Diagrama de Secuencia del CU *Realizar Test: Escenario Test Las Leyes*

