



Facultad 2

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

**APLICACIÓN WEB PARA EL PROCESAMIENTO DE ENCUESTAS DE
IDENTIFICACION DE ESTUDIANTES POTENCIALMENTE TALENTOSOS EN
INFORMÁTICA**

Autor:

Luis Mario Barrios Robainas

Tutor:

MSc. Dayana Caridad Tejera Hernández

La Habana, 2015.

“Año 57 de la Revolución”

"Una de las cosas más fascinantes de los programadores es que no puedes saber si están trabajando o no sólo con mirarlos. A menudo están sentados aparentemente tomando café, chismorreando o mirando a las nubes. Sin embargo, es posible que estén poniendo en orden todas las ideas individuales y sin relación que pululan por su mente"

Charles M. Strauss

Dedicatoria

Quiero dedicar este trabajo principalmente a las personas más allegadas: mi mamá, mi papá, mi hermano, mi tío y a mi familia en general, por apoyarme en todo momento. También a aquellas personas que no se encuentran hoy día presentes, y que deseaban ver este sueño hecho realidad.

Declaración de Autoría

Declaramos ser autores de la presente tesis y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Firma del Autor

Luis Mario Barrios Robainas

Firma del Tutor

M.Sc. Dayana Caridad Tejera Hernández

Datos de Contactos

M.Sc. Dayana Caridad Tejera Hernández, Ing.

Graduada de Ingeniería en Ciencias Informáticas en 2007. Se desempeña actualmente como metodóloga del Departamento Metodológico Central de Ingeniería y Gestión de Software (IGSW) de la UCI, como profesora principal de las asignaturas Ingeniería de Software 1 y 2 (IS1 e IS2) y como jefa de la disciplina Ingeniería y Gestión de Software. En 9 años de labor docente (incluidos 2 años de alumna ayudante), ha impartido las asignaturas IS1, IS2, Gestión de Software, Metodología de la Investigación Científica e Historia de la Informática. También ha impartido cursos de posgrado. Se desempeñó como jefa del departamento de IGSW y PP y luego como asesora de calidad en la facultad 3. Cumplió misión en Venezuela, en el despliegue de la solución informática para la cadena de tiendas PDVAL. Más adelante fue coordinadora de 5to año en la Dirección de Planificación de la Vicerrectoría de Formación. Posee categoría docente principal de Profesor Asistente. Se graduó de la maestría en Tecnología educativa, e-learning y gestión del conocimiento de la Universidad de las Islas Baleares, España.

Correo electrónico: dtejera@uci.cu

Agradecimientos

Gracias a mi mamá Saray, por los muchos sacrificios que ha tenido que hacer por mí y por mi familia, pero más que eso le agradezco por ser mi mamita linda, TE QUIERO MUCHO AUNQUE NUNCA TE LO DIGA. Gracias a mi papá Rolando por esta vida tan maravillosa que me ha dado, a mi hermano Rolandito por apoyarme en todo lo que ha podido, a mi tío Miguel que más que un tío ha sido siempre un padre. Gracias, todo lo que soy se lo debo a ustedes.

Le agradezco a la persona más importante en este proceso de tesis, sin duda alguna sin su presencia, su confianza y dedicación para conmigo, estos últimos meses en la universidad, nunca hubiese podido terminar satisfactoriamente, por lo que es válida la aclaración y el reconocimiento que se merece: A mi tutora Dayana Caridad Tejera Hernández, mil gracias por pasar estos momentos difíciles ayudándome.

Gracias también a todas las bellas personas que conocí aquí en la UCI, especialmente a los más allegados a mí que siempre me extendieron la mano en las situaciones difíciles que me presentó la vida. Nunca voy a olvidar mi primer año, ese año donde todo te parece grande y estrepitoso, mi primer grupo 7101, quienes juntos dimos los primeros pasos como universitarios y donde también estaban mis primeros compañeros con los que compartí los buenos y malos momentos. A mi gente de la UCI gracias por soportarme todos estos años y por compartir tantas experiencias únicas en nuestra vida. Gracias a Alex mi hermano y mi eterno compañero de cuarto, al Piter y al Franco mis otros dos brothers, al Indio, al Riki, a Ivey, a Randy, a Jeffrey, al Joe, a Victor, a Mario Agustín mi tocayo, a Marla mi hermanita que aunque está lejos le agradezco por alarme las orejas y ponerme a estudiar, gracias por estar ahí siempre y por ser mis amigos, mis hermanos, mi familia aquí, estos cinco años, les deseo éxitos y prosperidad en esta vida y en la que viene.

Gracias a todas aquellas personas que de una forma u otra me han apoyado siempre, me han guiado por un buen camino y aconsejado de forma correcta, pues sin ellos no sería quien soy. Mencionarlos a todos creo que sería imposible pues son muchos los que de una forma u otra han contribuido a la causa de hacer un hombre de bien con mi persona. Por último agradecer también a esas personas que se incorporan a nuestras vidas y se convierten en personas especiales, solo para dejarte una enseñanza y decirte cuando realmente lo necesitas: "TODO VA A SALIR BIEN".

Resumen

El proyecto Talenmático de la facultad 5 de la Universidad de las Ciencias Informáticas (UCI) desarrolla el Sistema Inteligente De Identificación y Clasificación de Talentos (SIDICT). Para proveer a este sistema la información necesaria sobre los indicadores que define si un estudiante es potencialmente talentoso o no, el proyecto diseñó un grupo de encuestas. Estas deben ser aplicadas en principio a todos los estudiantes, por lo cual su procesamiento les resultará engorroso a los profesores. Para su aplicación en el Sistema de Encuestas (SE) de la UCI deben ser procesados los ficheros que este genera.

La presente investigación tiene como objetivo desarrollar una aplicación web que permita minimizar el esfuerzo empleado por los profesores para procesar las encuestas del proyecto Talenmático, para la identificación de estudiantes potencialmente talentosos en la UCI. Para el desarrollo del sistema se utilizó la metodología Proceso Unificado de Desarrollo y Lenguaje de Modelado Unificado. Fue utilizado el entorno de desarrollo JBoss Developer Studio, unido a Java como lenguaje de programación, PostgreSQL como Sistema Gestor de Base de Datos y Visual Paradigm para el modelado. Junto a estas herramientas se incorporaron frameworks y librerías. Todos los elementos anteriores giraron alrededor del patrón Modelo-Vista-Controlador. Luego de concluir el desarrollo se obtuvo como resultado un componente funcional y con atributos de seguridad.

Palabras claves: identificación, clasificación, talento, Sistema de Gestión de Encuestas, SIDICT.

Contenido

Contenido	8
Introducción.....	9
Capítulo 1. Fundamentación Teórica.....	13
Capítulo 2. Diseño e implementación de la propuesta.....	30
Capítulo 3: Validación.	53
Conclusiones.....	59
Recomendaciones.....	59
Bibliografía	60
Anexos	63
Anexo 1: Instrumento para la autoidentificación de estudiantes potencialmente talentosos.....	63
Anexo 2: Instrumento para la identificación de estudiantes potencialmente talentosos, por sus condiscípulos.....	65
Anexo 3: Instrumento para la identificación de estudiantes potencialmente talentosos por los profesores.	66
Anexo 4: Descripción de los indicadores definidos por el proyecto Talemático	69
Anexo 5: Encuesta realizada al actual Director en función de Informatización.	71

Introducción

El siglo XXI ha sido denominado como el momento de la Tecnología de la Información y las Comunicaciones (TIC), alcanzando niveles de desarrollo impresionantes. Constantemente se hace alusión a la sociedad de la información, concebida como una forma de desarrollo económico y social. Las TIC juegan un papel central en la actividad económica, en la creación de riquezas y en la definición de la calidad de vida y las prácticas culturales de los ciudadanos. Su uso e introducción en los diferentes niveles y sistemas pedagógicos, tienen un impacto significativo en el desarrollo del aprendizaje de los estudiantes y en el fortalecimiento de sus competencias para el trabajo instructivo, ya que favorecerán en gran medida su inserción en la sociedad del conocimiento. [1]

A lo largo de la historia de la humanidad, el desarrollo del talento y la creatividad ha sido de gran importancia. Las personas con alto desempeño han hecho progresar la sociedad y mejorar las condiciones de vida. Los países que cuentan con un alto potencial científico dedican cuantiosos recursos a la investigación sobre el tema del talento, obteniendo amplios resultados en este campo. Ocupa un lugar destacado Japón con 4.1 científicos por cada mil habitantes, le siguen Israel con 3.8, Estados Unidos con 3.7, la Unión Europea con 2.7 y en América Latina, Cuba con 1.8. Estos datos dan la medida de los niveles de atención que se le presta a los sujetos talentosos en el mundo y de la relación que se puede establecer entre el potencial científico de una nación y la atención e importancia que se le concede al desarrollo del talento, de ahí la necesidad de estimular su desarrollo. [2]

En la Universidad de las Ciencias Informáticas (UCI) actualmente no se han logrado sistematizar actividades para la detección y estimulación del talento universitario, aun cuando existen diferentes espacios que propician el impulso del potencial de sus educandos. En la Facultad 5 de la UCI se trabaja en un proyecto de innovación pedagógica aprobado por el Grupo de Proyectos del Centro de Innovación y Calidad de la Educación (CICE), denominado: "Modelo de Intervención Educativa para la Estimulación del Talento Informático en la UCI", identificado con el acrónimo de Talenmático. Su objetivo es sistematizar un Modelo Educativo que permita brindar una atención científicamente fundamentada, a los estudiantes potencialmente talentosos en informática (EPTI), para su identificación, selección, estimulación y orientación. [3] Uno de los resultados del trabajo de sus miembros, fue la definición de los indicadores (Anexo 4:

Descripción de los indicadores definidos por el proyecto Talemático) a tener en cuenta para la identificación de EPTI. Como parte del proyecto, se desarrolla el Sistema Inteligente De Identificación y Clasificación de Talentos (SIDICT) [4], el cual necesita tener información de los estudiantes relacionada con estos indicadores.

Talenmático cuenta con tres encuestas diseñadas para la obtención de esta información, a partir de los criterios de cada estudiante, de sus compañeros de aula y sus profesores; por lo que el número de personas a encuestar es muy grande. Algunos miembros del proyecto ya han realizado estas encuestas en sus grupos y en proyectos de la facultad 5 que le han sido asignados. Sin embargo estos alegan que el procesamiento de forma manual de estos datos les ocupa tiempo considerable y les resulta tedioso lo que implica un alto esfuerzo humano. Además el rigor y la veracidad de la información extraída no es totalmente confiable.

La UCI cuenta con un Sistema de Encuestas (SE <http://encuesta.uci.cu>) que es utilizado con fines de interés de la institución, que pudiera ser la solución a esta situación. Este sistema no fue desarrollado en Cuba por lo que no se puede implementar ningún servicio web para consumo de los resultados obtenidos en las encuestas, sin embargo, este permite exportar los resultados en formato de EXCEL.

Por lo anteriormente planteado surge como **problema de investigación:** ¿Cómo minimizar el esfuerzo empleado por los profesores para procesar las encuestas del proyecto Talemático para la identificación de estudiantes potencialmente talentosos en la UCI?

Siendo el Objeto de estudio de la investigación:

La gestión de encuestas.

Y el Campo de acción:

Se enmarca en los sistemas de procesamiento de encuestas para la identificación de estudiantes potencialmente talentosos en la Universidad de las Ciencias Informáticas.

Y como Objetivo General:

Desarrollar una aplicación web que permita minimizar el esfuerzo empleado por los

profesores en el procesamiento de las encuestas del proyecto Talenmático para la identificación de estudiantes potencialmente talentosos en la UCI.

Para desarrollar satisfactoriamente el sistema se define un conjunto de **tareas de investigación** que permiten cumplir el objetivo general propuesto, estas son:

1. Definición de los conceptos relacionados con los sistemas de gestión de la información de estudiantes potencialmente talentosos.
2. Descripción del estado actual de la gestión de encuestas y de los sistemas para el procesamiento de encuestas.
3. Desarrollo de la implementación y diseño de un sistema web para el procesamiento de encuestas de los EPTI en la UCI.
4. Validación de la propuesta enfocada en la solución el problema de investigación definido.

Métodos utilizados en la investigación:

Métodos teóricos

Permiten tener un conocimiento general sobre el estado actual del tema escogido, su evolución y conceptos generales. Se plantea utilizar los siguientes métodos:

- **Analítico sintético:** Este método permitirá analizar las teorías y los documentos referentes al tema de la investigación, facilitando de esta forma la extracción de los elementos más importantes relacionados con el objeto de estudio. Además a partir del análisis detallado de cada uno de los documentos previamente mencionados.
- **Análisis documental:** Permitirá seleccionar las ideas informativamente relevantes de un documento a fin de expresar su contenido sin ambigüedades para recuperar la información en él contenida. La aplicación de este método será de gran ayuda porque a partir del análisis realizado a los documentos que exponen con mucho detalle el flujo que sigue el proceso que representa el objeto de la investigación, contribuirá a la fácil comprensión del mismo.

Métodos empíricos

Para enriquecer la investigación se propone utilizar métodos empíricos, que describen y explican las características fenomenológicas del objeto, representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional.

- **Entrevista:** La aplicación de este método permitirá identificar los requisitos funcionales y no funcionales que debe cumplir el sistema intimados por el usuario. Ver Anexo 6.
- **Encuestas:** Se aplicarán a miembros del Proyecto Talenmático con el objetivo de validar si el sistema permite la accesibilidad de la información necesaria.

El presente trabajo está estructurado en tres capítulos, tal y como se describe a continuación:

Capítulo 1. Fundamentación teórica: En el presente capítulo se tratan los aspectos teóricos que dan sustento a la investigación, tales como: sistemas similares vinculados al campo de acción enunciado, selección de las herramientas, tecnologías y proceso de desarrollo.

Capítulo 2. Diseño e implementación de la propuesta: Este capítulo abarca el diseño de cada uno de los artefactos creados en correspondencia con la propuesta del sistema, así como la presentación de los aspectos relacionados con los procesos de implementación.

Capítulo 3: Validación: En el presente capítulo se efectúan pruebas a la aplicación en aras de comprobar su calidad y se detallan los aspectos que se tuvieron en cuenta para su desarrollo. Además se presenta la validación de la propuesta por los miembros del proyecto Talenmático.

Este documento posee, además, **Conclusiones generales, Recomendaciones, Bibliografías, Referencias bibliográficas, Glosario de términos y Anexos.**

Capítulo 1. Fundamentación Teórica

1.1 Introducción

En este capítulo se tratan los principales conceptos asociados al talento y a los indicadores definidos para su identificación. Además se estudia la gestión de encuestas y específicamente a su procesamiento, como medio para la obtención de la información necesaria para la identificación de Estudiantes Potencialmente Talentosos en Informática. Por último se describen las metodologías, lenguajes, tecnologías y herramientas utilizadas para el desarrollo de la solución propuesta.

1.2 Definición de Conceptos

1.2.1 Definiciones de talento

Existen muchas definiciones de talento en general. Según [5] el talento es: “una configuración psicológica de la personalidad de naturaleza cognitivo-afectiva, que integra de manera dinámica las capacidades generales y especiales del individuo, con una fuerte energía motivacional manifiesta en los planos intra e interpersonal, expresando un alto nivel de desempeño creativo en un área de interés”.

El talento también se puede definir como: “el resultado de la integración funcional de la inteligencia y las capacidades especiales del ser humano con el desarrollo de fuertes intereses en áreas donde se halla profundamente comprometido en lo emocional”. [6]

El talento es una especial capacidad intelectual o aptitud que una persona tiene para aprender las cosas con facilidad o para desarrollar con mucha habilidad una actividad. Se puede considerar que el talento informático puede ser concebido como una configuración psicológica de determinado nivel de complejidad, pero que al mismo tiempo, es muy específico en lo particular, del estudiante o profesional de las Ciencias Informáticas. Ello no significa en modo alguno, que cualquier otro profesional que no sea especialista en informática, no disponga del mismo. [3]

El análisis de las definiciones anteriores permitió definir al talento como: un fenómeno social presente en individuos con capacidades y habilidades cognitivas, emocionales y psicosociales ampliamente desarrolladas. En la presente investigación se asume el concepto de talento definido por el proyecto Talenmático.

1.2.2 Definición de talento informático

Según la organización o la época han existido diferentes tipos de talentos. Por ejemplo, existe el talento comercial, el talento directivo, el talento de líder, talento técnico, talento administrativo y así dependiendo de sus funciones, objetivos y aporte de valor a la organización. [3]

De acuerdo con [3] no existe una definición de talento asociado al área de la informática, por lo que en esta investigación se asume la propuesta por estos actores:

“Formación psicológica compleja sustentada en una elevada motivación profesional, que le permite al estudiante potencialmente talentoso desarrollar altos niveles de creatividad a partir del compromiso e interacción con el ambiente social-tecnológico en que se desenvuelve, obteniendo resultados relevantes socialmente válidos en una, o varias áreas de la informática”.

1.2.3 Indicadores del talento.

El proceso de identificación constituye el primer paso para abordar la atención educativa de los alumnos con talento y pretende determinar su capacidad y ritmo de aprendizaje como indicadores indispensables para poder ofrecer respuestas educativas que tengan en cuenta sus necesidades.[7]

Para identificar a estudiantes talentosos en la investigación de [8] se utilizaron como indicadores: Dominio de conocimientos, hábitos y habilidades estipulados para el grado o nivel; intereses marcados hacia la asignatura, motivación, esfuerzo; dinámica de aprendizaje (rapidez, facilidad, profundidad en el proceso de asimilación, etc.); independencia cognoscitiva; creatividad.

En [9] se define un instrumento para la identificación de estudiantes potencialmente talentosos que utiliza indicadores definidos para diferentes áreas del conocimiento. Este está conformado por una categoría de habilidades consideradas generales que agrupa los indicadores correspondientes al nivel de conocimiento y habilidades de aprendizaje, las características de creatividad y productividad, junto a las características socio afectivas y cuatro categorías de habilidades específicas: a) lenguaje y escritura, b) lógica y matemática, c) ciencias naturales, y d) ciencias sociales.

Para el talento informático los indicadores definidos en el proyecto Talenmático están

ubicados en tres dimensiones: Cognitiva-Instrumental, Afectiva-Emocional y Psico-Social. Estos indicadores pueden encontrarse en el Anexo 4: Descripción de los indicadores definidos por el proyecto Talemático Teniendo en cuenta que estos fueron los indicadores definidos por el proyecto y que se encuentran en el área de la informática, fueron los seleccionados para esta investigación.

En el siguiente acápite se tratará la gestión de encuestas teniendo en cuenta que este es el método utilizado en el proyecto Talemático para la obtención de la información respecto a los indicadores del talento informático.

1.2.4 Gestión de encuestas

La encuesta es una técnica de adquisición de información de interés sociológico, que se realiza mediante un cuestionario, a través del cual se puede conocer la opinión o valoración del sujeto seleccionado en una muestra sobre un asunto dado. El cuestionario cuenta con una estructura lógica y rígida que permanece inalterada. El encuestado lee previamente el cuestionario y lo responde por escrito. Para facilitar la evaluación de los resultados por métodos estadísticos, se determinan las posibles variantes de respuestas estándares. [10]

Diferentes autores han propuesto pasos para el desarrollo de encuestas. En [11] se proponen los siete pasos del ciclo o proceso de implementación de una encuesta (Ver Figura 1: Ciclo de ejecución de una encuesta).

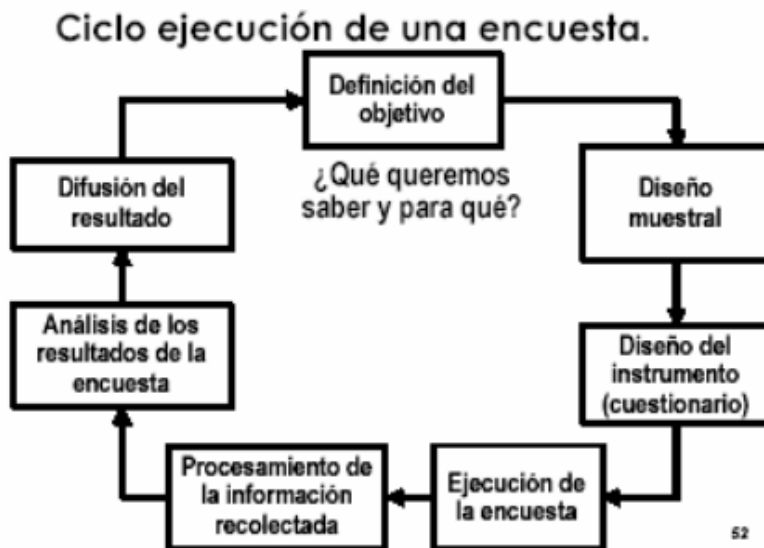


Figura 1: Ciclo de ejecución de una encuesta

Fuente: Tomado de [11].

Según [12] los pasos para el ciclo son:

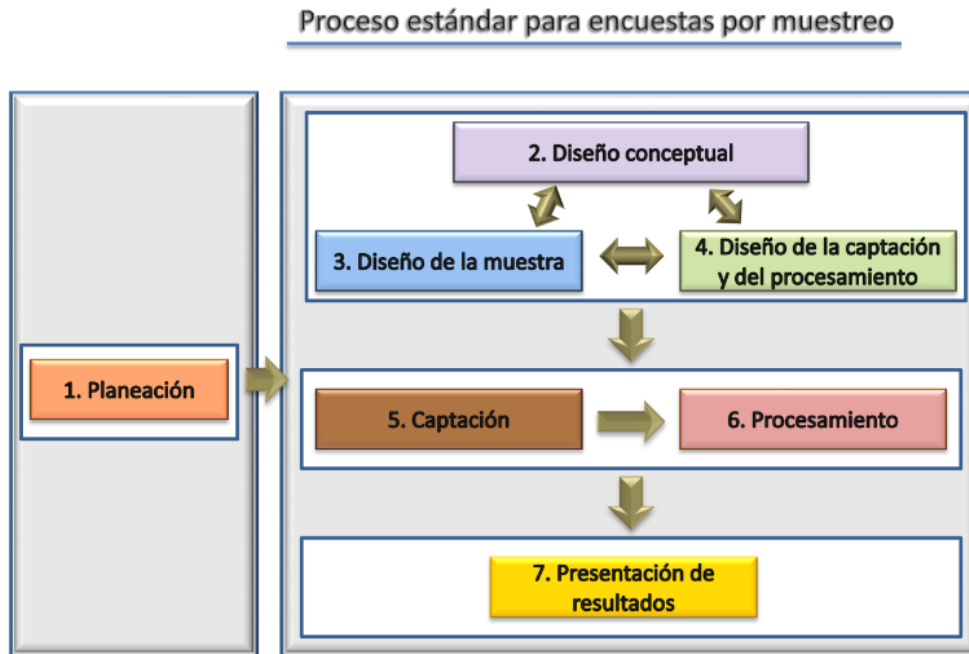


Figura 2: Proceso estándar para encuestas por muestreo.

Fuente: Tomado de [12]

En los dos procedimientos el procesamiento es uno de los pasos. En esta fase se organizan los datos captados, en archivos de información, preparados para su explotación estadística, lo cual implica, según las modalidades tecnológicas, la captura y codificación manual o automatizada de los datos individuales, además de la validación de la información para eliminar inconsistencias y resolver, en lo posible, problemas de omisiones de datos o multirrespuestas. [12]

Procesamiento de la información colectada estará en función a la metodología definida previamente, donde debe prevalecer un enfoque objetivo e imparcial a fin de evitar problemas en el trabajo.

Requisitos esenciales para procesar información fiable: [11]

- ✓ Asegurar la independencia en el diseño, ejecución y análisis de encuestas.

- ✓ Asegurar la integridad de las encuestas y su análisis.
- ✓ No permitir la manipulación de datos para fines políticos.

1.3 Análisis de sistemas homólogos

En la búsqueda se encontraron muchos sistemas de gestión de encuesta asociados al área empresarial centrados más bien en la gestión de recursos humanos, que en la gestión del talento académico como tal.

En el ámbito internacional

En [13] se referencian varios sistemas de gestión y procesamiento de encuestas.

1.3.1 RotatorSurvey

Es un software profesional que permite diseñar, procesar y analizar encuestas de todo tipo. Dirigida a Investigadores, Agencias de Investigación de Mercados y Empresas de Opinión Pública que buscan alta calidad, máxima productividad, trabajo en equipo y bajo costo. Permite construir cuestionarios en papel, tabular y digitar los datos, asegurar la calidad y confiabilidad de todo el proceso y controlar al máximo las inconsistencias. [14]

El análisis realizado al sistema RotatorSurvey permitió conocer sus características y funcionalidades. Esta aplicación a pesar de los avances tecnológicos está basada en el uso del papel. Según [14] sigue siendo el medio más usado para realizar encuestas profesionales.

1.3.2 NetQuest

Survey Manager es una solución de software desarrollada íntegramente por Netquest para la gestión de encuestas avanzadas a través de Internet. Cubre todas las fases del proceso de creación y lanzamiento de una encuesta online. Este no requiere ninguna instalación, se usa a través de Internet. Desarrollada íntegramente por Netquest y se encuentra en constante actualización y adaptación a las demandas del sector.[15]

El sistema antes mencionado preocupamos por hacer que las encuestas sean fáciles de responder para el encuestado. Pensamos que preguntar información de forma lógica e intuitiva permite obtener datos de mayor calidad y mejorar al mismo tiempo las tasas de respuesta. Por ese motivo, hemos ideado múltiples preguntas interactivas que facilitan la tarea y mejoran los datos obtenidos

1.3.3 TGI TECNOLOGÍAS PARA LA GESTIÓN INDUSTRIAL S.A.

QSS software para la toma de encuestas en terreno. QSS 2.0 (Quick Survey System) es un software que permite la toma de encuestas en terreno utilizando computadores de bolsillo con sistema operativo Windows for Pocket PC. QSS busca reducir significativamente el tiempo que transcurre entre la captura de los datos y la disponibilidad de los resultados, a la vez que reduce los errores en la toma de encuestas.[16]

En el ámbito nacional

1.3.4 Sistema de Encuesta (LimeSurvey)[17]

En la UCI se utiliza un software producido por LimeSurvey que es capaz de aplicar encuestas a la población en el momento y el lugar que se requiera, con las funcionalidades precisas, garantizando un trabajo con la máxima calidad posible. Se puede instalar donde el usuario final quiera y tantas veces como necesite, sin coste alguno.

1.3.5 Sistema de Gestión de Encuestas[18]

Desarrollado en la UCI como resultado de una tesis de diplomado. Para la elaboración del sistema se utilizó como metodología Proceso Unificado de Desarrollo de Software (RUP) , la cual define Lenguaje Unificado de Modelado (UML), para la modelación del sistema se utilizó Visual Paradigm (VP), como lenguaje de programación Java y como entorno de desarrollo integrado NetBeans. Actualmente en la UCI se utiliza el 1.3.4 Sistema de Encuesta (LimeSurvey) .

1.4 Tendencias y tecnologías actuales a considerar

En este acápite se describen las tecnologías a utilizar en el proceso de desarrollo de la solución: metodología de desarrollo de software, lenguaje de modelado, patrones de arquitectura y diseño, servidor de base de datos, tecnologías y herramientas a utilizar en el proceso.

1.4.1 Metodología de desarrollo de software

El Proceso Unificado de Desarrollo (RUP) es una de las metodologías de desarrollo de software más utilizada en el mundo para el análisis, implementación y documentación de sistemas orientados a objetos. Durante todo el ciclo de desarrollo, se administran los requisitos, con el objetivo de mantener la trazabilidad de los requerimientos funcionales,

siendo así el proceso de desarrollo guiado por casos de uso. Permite la reutilización de estos requerimientos para otros sistemas o funcionalidades del propio software. RUP es centrado en la arquitectura, por lo que es fundamental identificar los casos de uso de mayor prioridad e importancia basándose en este criterio. [19]

El ciclo de vida de RUP se caracteriza por : [20]

- **Dirigido por casos de uso:** Los casos de uso reflejan lo que el cliente necesita y desea, lo cual se capta al modelar el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo, ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.
- **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema en la que el equipo de desarrollo y los clientes deben de estar de acuerdo. RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura.
- **Iterativo e Incremental:** Cada fase se desarrolla en iteraciones, una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla algunos más que de otros. Es práctico dividir el trabajo en partes más pequeñas o mini-proyectos, donde cada uno de ellos es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

Dentro de sus disciplinas gestiona el control de cambios, que permite mantener al equipo trabajando en los mismos artefactos, en cualquier momento del desarrollo del producto. En su modelación RUP define como sus principales elementos a los trabajadores, sus responsabilidades, los artefactos y los flujos de actividades. Los trabajadores son los propietarios de elementos o artefactos y se encargan de realizar las actividades, las cuales describen cómo una tarea es realizada por un trabajador y a su vez, manipulan elementos. RUP realiza varias iteraciones en número variable en las que se hacen mayor o menor hincapié en las distintas actividades por lo que se han definido en nueve flujos o disciplinas. [19]

Ventajas que ofrece trabajar con RUP [21]

- Permite crear software que cumplan las expectativas de los usuarios a través de

las especificaciones de requisitos.

- Permite llevar un seguimiento detallado en cada una de las fases de desarrollo.

Lenguaje Unificado de Desarrollo (UML)

UML es un lenguaje usado para especificar, visualizar, construir y documentar los componentes de un sistema orientado a objetos. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad, está apoyado en gran manera por el OMG (Object Management Group).[22]

Además ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Permite modelar sistemas utilizando técnicas orientadas a objetos, es independiente aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.[23]

Patrón de arquitectura

Los patrones arquitectónicos, son patrones de software que ofrecen soluciones a problemas de arquitectura de software. Especifican un conjunto predefinido de subsistemas con sus responsabilidades y una serie de recomendaciones para organizar los distintos componentes. Expresan un esquema organizativo estructural fundamental para sistemas de software. Resuelven problemas arquitectónicos, adaptabilidad a requerimientos cambiantes, modularidad, acoplamiento, entre otros. La solución que se plantea es la creación de patrones de llamadas entre objetos (similar a los patrones de diseño), decisiones y criterios arquitectónicos, empaquetado de funcionalidad.[22]

Un patrón de diseño expresa esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas automatizados. Es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular. Identifica clases, instancias, roles, colaboraciones y la distribución de responsabilidades. Constituye una solución estándar para un problema común de programación, una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios, un proyecto o estructura de implementación que logra una finalidad

determinada, son en fin, una manera más práctica de describir ciertos aspectos de la organización de un programa. [22]

Patrón Modelo-Vista-Controlador (MVC)

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón de llamada y retorno MVC, se ve frecuentemente en aplicaciones web, donde la vista es la página HTML (Hyper Text Markup Language) y el código es el que provee los datos dinámicos a la página. El modelo lo constituyen los datos con los que trabaja la aplicación y la lógica de negocio y el controlador es el responsable de recibir los eventos de entrada desde la vista. [24]

Modelo: Es el componente encargado del acceso a datos, representando las estructuras de datos del sistema. Típicamente el modelo de clases contendrá funciones para consultar, insertar y actualizar información de la base de datos.

Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: Responde a eventos y actúa como intermediario entre el modelo, la vista y cualquier otro recurso necesario para generar una página.

La utilización del patrón MVC facilita la realización de cambios en la aplicación puesto que cuando se realiza un cambio de bases de datos, programación o interfaz de usuario solo se maneja uno de los componentes. Además se puede modificar uno de los componentes sin conocer cómo funcionan los otros.

Arquitectura Cliente Servidor

La arquitectura Cliente – Servidor (C/S) es una forma de dividir y especializar programas y equipos de cómputo de forma que la tarea que cada uno de ellos realice se efectúe con la mayor eficiencia posible y permita simplificar las actualizaciones y mantenimiento del sistema. En este modelo, el papel del cliente es iniciar el diálogo, enviando peticiones al servidor conforme a algún protocolo asimétrico, pidiéndole que actúe, que le informe, o ambas cosas. El servidor es quien responde las solicitudes del cliente. [22]

Servidor de Base de Datos

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Entre sus principales características se encuentran: [25]

Extensible: El código fuente está disponible para todos sin costo. Si un equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales. Esto es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden PostgreSQL todos los días.

Multiplataforma: PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de Windows está actualmente en estado beta de pruebas. Diseñado para ambientes de alto volumen.

Lenguaje de programación

Java es una plataforma de software desarrollada por un grupo de ingenieros de SunMicrosystem en 1991. La plataforma Java consta de las siguientes partes: [26]

- El lenguaje de programación, Java.
- La máquina virtual de Java o JRE, que permite la portabilidad en ejecución.
- El API Java, una biblioteca estándar para el lenguaje.

Java incorpora parte de la sintaxis de C, C++, Smalltalk, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros. Tiene una arquitectura neutral, portátil y robusta. Posee las estructuras mínimas de un lenguaje de programación tradicional, sin añadir ninguna estructura más. Una de las características más importantes es que los programas “ejecutables”, creados por el compilador de Java, son independientes de la arquitectura. Se ejecutan indistintamente en una gran variedad de equipos con diferentes microprocesadores y sistemas operativos. Proporciona numerosas comprobaciones en

compilación y tiempo de ejecución y elimina la necesidad de liberación explícita de memoria, convirtiéndolo en un lenguaje robusto. [26]

Este lenguaje es utilizado de manera horizontal en el desarrollo del sistema, pues puede estar presente en las diferentes capas de la aplicación. Con Java se pueden programar aplicaciones web dinámicas, con acceso a bases de datos, utiliza XML, con cualquier tipo de conexión de red entre cualquier sistema.[26]

Tecnologías utilizadas en el proceso de desarrollo

En este epígrafe se exponen las tecnologías utilizadas en el proceso de desarrollo, agrupadas en el Modelo, la Vista y el Controlador, según el patrón arquitectónico seleccionado.

Java Enterprise Edition 5 (JEE5)

Es la plataforma que provee SunMicrosystem para dar soporte y desarrollar software para las empresas. El gran éxito de java como plataforma para el desarrollo de aplicaciones se encuentra en esta especificación, que no es más que un conjunto de librerías que establecen un estándar para lograr un producto altamente calificado. Con el lanzamiento de esta edición, la especificación dio un gran paso hacia la excelencia, pues incorporó nuevas tecnologías de punta que no estaban contempladas anteriormente y que indiscutiblemente le incorporan a la plataforma gran robustez y simplicidad a la hora de trabajar, suficientes razones para no dudar en establecerla como base para el desarrollo. [27]

Java virtual machine

La Máquina Virtual Java es el núcleo del lenguaje de programación Java. En ella se encuentra el motor que en realidad ejecuta el programa Java y es la clave de muchas de las características principales de Java, como la portabilidad, la eficiencia y la seguridad. Se encarga de interpretar todo el código java y convertirlo al lenguaje nativo del sistema operativo en uso. La utilización de la máquina virtual java permite la portabilidad de la aplicación.[28]

Tecnologías para la presentación

XHTML

Es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. XHTML es solamente la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas de XML. La utilización de XHTML facilita en la aplicación la edición directa del código y de mantenimiento. [29]

Java Server Face (JSF)

Java Server Faces (JSF) es un framework Java que permite crear interfaces de usuario (UI) para una aplicación web, mediante el uso de componentes reutilizables. Permite el manejo de estados y eventos, así como la asociación entre los datos de la interfaz y los datos de la aplicación web. [30]

Facelets

Facelets es un framework simplificado de presentación, en donde es posible diseñar de forma libre una página web y luego asociarle los componentes JSF específicos. Aporta mayor libertad al diseñador y mejora los informes de errores que tiene JSF. [30]

Ajax4JSF

Ajax4jsf es una librería de código abierto que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas, dotándolas con tecnología Ajax de forma limpia y sin añadir código JavaScript. Mediante el uso de este framework se puede variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de recargarla por completo, realizar peticiones al servidor automáticas, control de cualquier evento de usuario, entre otras. [30]

Richfaces

Es una biblioteca de componentes JSF y un avanzado framework para la integración de Ajax con facilidades en la capacidad de desarrollo de aplicaciones de negocio. Se integra completamente dentro del ciclo de vida JSF. Permite crear interfaces de usuario modernas de manera eficiente y rápida, basadas en componentes listos para usar,

altamente configurables en cuanto a temas y esquemas de colores predefinidos por el propio framework o desarrollados a conveniencia. Richfaces, además, es un proyecto que fue desarrollado con una arquitectura abierta para que fuera compatible con la mayor cantidad de entornos. [30]

Seam UI

Serie de controles Seam altamente integrables con JBossSeam. Es utilizado para complementar los controles JSF incorporados y los controles de otras bibliotecas externas. [30]

Tecnologías para el negocio

Seam

Es un poderoso y moderno framework creado para unificar todas las tecnologías estándares JSF, EJB3, JPA. Es un software de código abierto que asocia diferentes tecnologías y estándares Java, adicionando algunas funcionalidades no contempladas en frameworks anteriores. Automatiza muchas de las tareas comunes y hace un uso extensivo de las anotaciones para poder reducir la cantidad de código XML que es necesario escribir, reduce de manera importante la cantidad total de codificación necesaria. Permite la creación de complejas aplicaciones Web. [31]

Integra además el concepto de workspaces (entornos de trabajo) permitiendo que el usuario tenga en varios tabs o ventanas del navegador actividades del negocio con contextos completamente aislados. Seam integra transparentemente la administración de procesos del negocio vía jBoss-jBPM, haciendo muy fácil implementar y optimizar complejas colaboraciones e interacciones con el usuario. [31]

Tecnologías para el acceso a datos

Hibernate

Hibernate es una herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones. Hibernate es un software libre, distribuido bajo los términos de la licencia

GNU LGPL. [32]

Enterprise Java Bean 3 (EJB 3)

Enterprise JavaBeans (EJB) es una arquitectura de componentes de servidor que simplifica el proceso de construcción de aplicaciones de componentes empresariales distribuidos en Java. Con su utilización es posible escribir aplicaciones escalables, fiables y seguras sin escribir código de infraestructura. La existencia de infraestructura permite un desarrollo más rápido de la parte servidora. Un EJB es un componente software que se ejecuta del lado del servidor en una aplicación multicapa. Los clientes del EJB acceden a él por medio de una interfaz que esconde los detalles de implementación del componente. Esta interfaz debe cumplir la especificación EJB. La especificación fuerza la presencia de unos determinados métodos que permitirán al contenedor de EJBs manejar los componentes y su ciclo de vida. [33]

Herramientas

Visual Paradigm para UML 8.0

Se utilizará el Visual Paradigm 6.0 como herramienta CASE (Computer Aided Software Engineering), esta es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software. Ayuda a una construcción de aplicaciones de calidad más rápida y a menor costo. Permite construir todo tipo de diagramas de clases, generar código desde diagramas y generar documentación. Apoya los estándares más altos de las notaciones de Java y de UML. Soporta aplicaciones web, es fácil de instalar y actualizar. Está diseñado para distintos usuarios entre los que se incluyen ingenieros de software, analistas de sistemas, analistas de negocios, arquitectos y desarrolladores. Está orientado a la creación de diseños y se usa el paradigma de programación orientada a objetos. [34]

Visual Paradigm es una poderosa herramienta para visualizar y diseñar elementos de software, para ello utiliza UML (UML 2.1) y ofrece una gama de facilidades para el modelado de aplicaciones. Está orientada a la creación de diseños usando el paradigma de programación orientada a objetos. Provee soporte para la generación de código, tiene integración con diversos IDE's como NetBeans (de SunMicrosystems), Developer (de Oracle), Eclipse (de IBM), JBuilder (de Borland), así como la posibilidad de realizarse la ingeniería inversa para aplicaciones realizadas en JAVA, .NET, XML e Hibernate. [34]

JBoss Developer Studio 8.0

La plataforma Eclipse consiste en un Entorno de Desarrollo Integrado (IDE, Integrated Development Environment) abierto y extensible. Un IDE es un programa compuesto por un conjunto de herramientas útiles para un desarrollador de software. Como elementos básicos, un IDE cuenta con un editor de código, un compilador/intérprete y un depurador. Eclipse sirve como IDE Java y cuenta con numerosas herramientas de desarrollo de software. Es de gran utilidad porque da soporte a otros lenguajes de programación, como son C/C++, Cobol, Fortran, PHP o Python. A la plataforma base de Eclipse se le pueden añadir extensiones (plug-ins) para extender la funcionalidad. [31]

Dentro de las características que presenta se destacan: poseer un editor visual con sintaxis coloreada, permitir la compilación incremental de código, modificar e inspeccionar valores de variables, avisar de los errores cometidos mediante una ventana secundaria y depurar el código que reside en una máquina remota. Es soportado por los sistemas operativos: [31]

- Linux
- Windows
- Solaris 8 (SPARC/GTK 2)
- Mac OSX –Mac/Carbon

Jboss Tools

JBoss Tools es un conjunto de plug-ins de Eclipse que tiene como objetivo ayudar a los desarrolladores a crear aplicaciones webs de forma rápida y sencilla.[32]

Los módulos de JBoss Tools son:[32]

- RichFaces VE: El editor visual aportado por Exadel proporciona el apoyo para la edición visual de páginas HTML, JSF, JSP y Facelets. También incluye soporte visual para las librerías de componentes JSF incluyendo JBossRichFaces.
- Seam Tools: Incluye soporte para seam-gen, RichFaces VE.
- Hibernate Tools: Soporta el mapeo de archivos, anotaciones y JPA con la ingeniería inversa, completamiento de código, asistentes de proyecto, refactorización, ejecución interactiva de HQL/JPA-QL/Criteria.
- JBoss AS Tools: Fácil de iniciar, detener y debuguear al estar integrado con

Eclipse. También incluye funciones para el despliegue eficaz de cualquier tipo de proyecto en el IDE.

- Drools IDE: Editor de ficheros de reglas, debugueo e inspección de reglas.
- jBPM Tools: Edición del flujo de trabajo del jBPM, motor de procesos BPM.
- JBossWS Tools: Desarrollo, invocación, inspección y pruebas de webservices sobre http con la adición y soporte de características JBossWS.

PgAdmin III

PgAdmin III es una aplicación para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como PervasivePostgres, EnterpriseDB, MammothReplicator y SRA PowerGres. [25]

PgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I y mucho más.[25]

JBoss Server 4.2.2 GA

JBoss Application Server es el servidor de aplicaciones de código abierto más ampliamente desarrollado del mercado. Por ser una plataforma certificada J2EE, soporta todas las funcionalidades de J2EE 1.4 e incluye servicios adicionales como clustering, caching y persistencia. JBoss es ideal para aplicaciones Java y aplicaciones basadas en la web. También soporta Enterprise Java Beans (EJB) 3.0, lo que hace el desarrollo de las aplicaciones mucho más simple. Además, al ser desarrollado con tecnología Java, es multiplataforma. [32]

Sus principales características son:[32]

- Producto de licencia de código abierto sin coste adicional.

- Confiable a nivel de empresa.
- Orientado a arquitectura de servicios.
- Flexibilidad consistente.
- Servicios del middleware para cualquier objeto de Java.
- Soporte completo para Java Management Extensions (JMX).

Conclusiones parciales

En este capítulo se trataron los principales conceptos asociados al talento y a los indicadores que se definieron para su identificación. Se realizó un análisis sobre la gestión de encuestas y específicamente de su procesamiento, a nivel nacional e internacional. Además se investigó esta, como medio para la obtención de la información necesaria para la identificación de Estudiantes Potencialmente Talentosos en Informática. Por último se describieron las metodologías, lenguajes, tecnologías y herramientas que se utilizaron para el desarrollo de la solución propuesta.

Capítulo 2. Diseño e implementación de la propuesta.

Introducción

En el siguiente capítulo se describen las principales características del sistema a desarrollar. Como resultado del impulso de este proceso y para su mejor entendimiento se presenta la descripción del modelo conceptual y de la solución propuesta. Se definen y documentan los requisitos funcionales y no funcionales a tener en cuenta así como los actores y los casos de usos del sistema. Además se presenta la confección de los diagramas de clase del diseño con estereotipos web, de secuencia, el diseño de base de datos del sistema, así como el diagrama de componente y el modelo de despliegue.

2.1 Modelado del contexto.

La metodología RUP propone para el modelado del contexto el modelado de negocio o modelado de dominio. Como no se han podido identificar procesos de negocios bien definidos en este caso vamos a hacer un modelo conceptual porque abarca más información que el modelo de dominio. Un modelo conceptual representa los conceptos en el dominio del problema en el mundo real. El análisis orientado a objetos tiene por finalidad estipular una especificación del dominio del problema y los requerimientos desde la perspectiva de la clasificación por objetos y desde el punto de vista de entender los términos empleados en el dominio. [22]

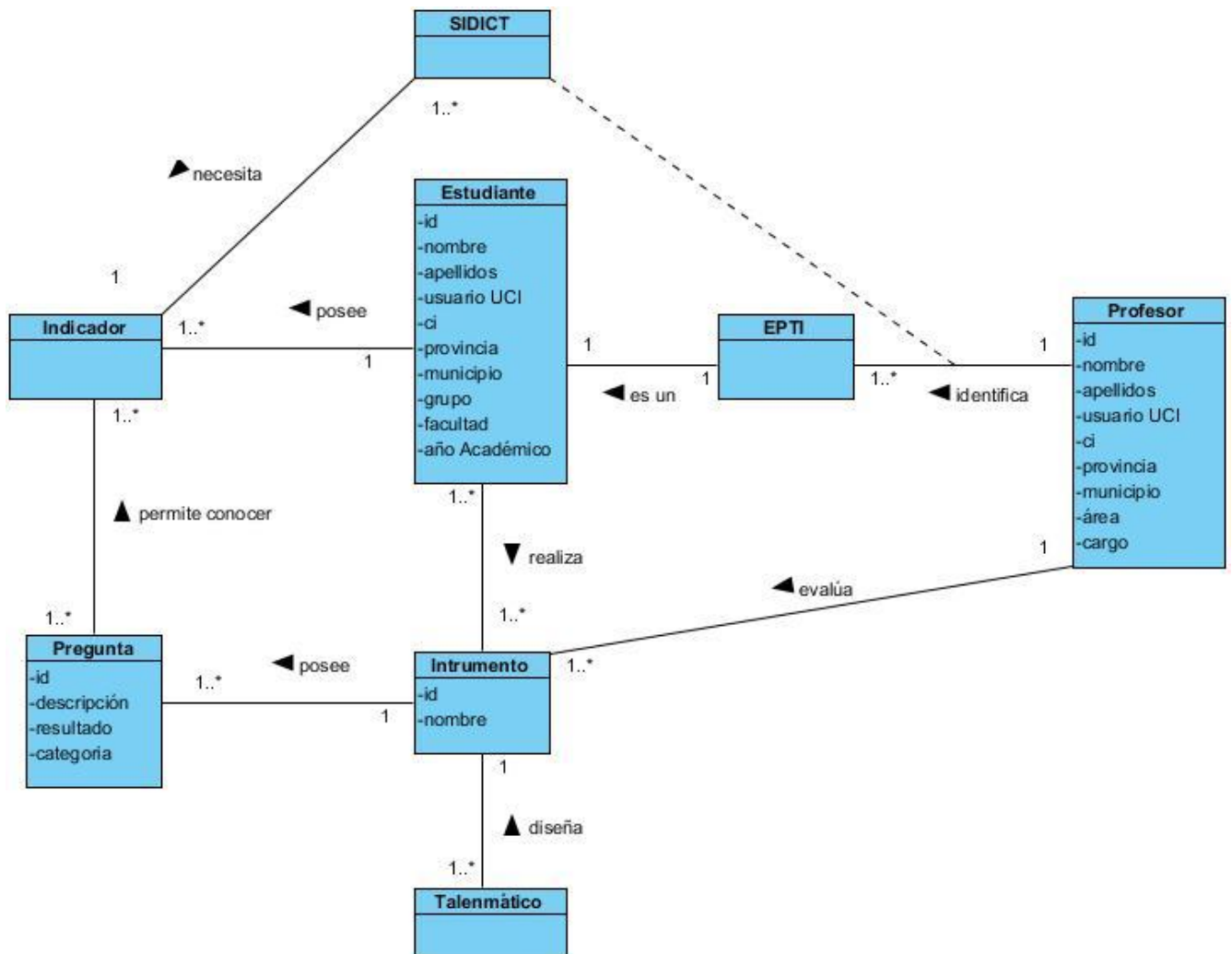


Figura 3: Modelo conceptual.

Fuente: Elaboración propia.

2.1.1 Descripción del modelo conceptual.

EPTI: Estudiante Potencialmente Talentoso en Informática.

Indicador: Son los indicadores definidos por el proyecto Talenmático para la identificación de los posibles talentos.

Instrumento: Son las encuestas que se aplicarán a los estudiantes.

Pregunta: Son las preguntas definidas para cada instrumento.

Talenmático: proyecto de innovación pedagógica que definió los instrumentos a utilizar.

Profesor: Es el especialista que identifica a un estudiante como posible talento.

Estudiante: Es el estudiante al que se le aplica el instrumento para identificar si es un

posible talento.

SIDICT: Sistema Inteligente De Identificación y Clasificación de Talentos.

2.2 Descripción de la solución propuesta

El sistema propuesto cumple el objetivo de desarrollar una aplicación web que permita minimizar el esfuerzo empleado por los profesores, en el procesamiento de las encuestas del proyecto Talenmático para la identificación de estudiantes potencialmente talentosos en la UCI. Para lograr esto el sistema ofrece dos formas de procesar la encuesta: una es manualmente y la otra cargando un fichero Excel en la base de datos. Como resultado de ambas formas, el sistema genera y va actualizando una base de datos que podrá ser utilizada por el sistema SIDICT para la identificación de los estudiantes potencialmente talentosos en la UCI.

La primera forma se concibió para las encuestas ya aplicadas por los profesores a los estudiantes manualmente, sin la utilización del SE. Para ello el usuario encargado deberá insertar primero los datos de la persona que ha llenado la encuesta (estudiante o profesor). En la Figura 4: Listar Persona se muestra una interfaz que permite inserta una persona nueva al sistema, así como ver detalles, editar los detalles de la misma, eliminar la persona e insertar instrumento. Además cuenta con filtros para realizar búsquedas personalizadas.

The screenshot displays a web interface for managing a list of people. It features a search filter section at the top with input fields for Name, Surnames, UCI User, City, Solapin, Province, and Municipality. Below the filters are 'Buscar' and 'Reiniciar' buttons. The main section, titled 'Resultados de Búsqueda de Persona', contains a table with one entry for 'Luis Mario Barrios Robainas Imbarrios'. The table columns include Name, Surnames, UCI User, City, Solapin, Province, Municipality, and Action. The Action column contains buttons for 'Ver', 'Editar', 'Eliminar', and 'Insertar instrumento'. At the bottom of the interface is an 'Insertar Persona' button.

Filtro de Búsqueda de Persona							
Nombre	<input type="text"/>						
Apellidos	<input type="text"/>						
Usuario UCI	<input type="text"/>						
Ci	<input type="text"/>						
Solapin	<input type="text"/>						
Provincia	<input type="text"/>						
Municipio	<input type="text"/>						
[Buscar]		[Reiniciar]					
Resultados de Búsqueda de Persona							
Nombre	Apellidos	Usuariouci	Ci	Solapin	Provincia	Municipio	Accion
Luis Mario	Barrios Robainas	Imbarrios	91030824427	EH13573	La Habana	Playa	[Ver] [Editar] [Eliminar] [Insertar instrumento]
[Insertar Persona]							

Figura 4: Listar Persona
Fuente: Elaboración propia.

En la Figura 5: Editar Detalles se muestra una interfaz donde se actualizan los datos de la persona.

Editar Persona

Idpersona*

Nombre

Apellidos

Usuariouci

Ci

Solapin

Provincia

Municipio

* Campos Requeridos

Nombre	Apellidos	Usuario UCI	Ci	Solapin	Provincia	Municipio	Cargo	Area
Leonardo	Boza Jimenez	leoboz	95010422542	T103877	Artemisa	Guanajay	Profesor	Facultad 2. Departamento de Ciencias Sociales y Humanidades

Figura 5: Editar Detalles
Fuente: Elaboración propia.

En la Figura 6: Insertar Estudiante se muestra una interfaz para insertar los datos de un estudiante, asociados a la persona.

Insertar Estudiante

Grupo

A. academico*

Facultad*

* Campos Requeridos

Persona *

Nombre	Apellidos	Usuariouci	Ci	Solapin	Provincia	Municipio
Luis Mario	Barrios Robainas	Imbarrios	91030824427	EH13573	La Habana	Playa

Figura 6: Insertar Estudiante
Fuente: Elaboración propia.

En Figura 7: Insertar Instrumento, Ver Detalles de Instrumento y Editar Instrumento se muestra una interfaz que permite ver los detalles del instrumento, editarlo y crearlos.

Filtro de Búsqueda de Instrumento

Nombre instrumento

Buscar
Reiniciar

Resultados de Búsqueda de Instrumento

Nombre instrumento	Accion
Prueba	Ver Editar

Insertar instrumento

Figura 7: Insertar Instrumento, Ver Detalles de Instrumento y Editar Instrumento

Fuente: Elaboración propia.

La segunda forma de procesar la encuesta es cargando los datos desde un Excel con el formato definido por el SE. Para ello la aplicación permitirá insertar de forma manual una hoja Excel y su contenido, reconociendo las columnas de asociadas a las preguntas de la encuesta y las filas para identificar a cada encuestado. Esta contará además con un botón Agregar Archivo Excel que permitirá cargar un fichero al sistema, copiarlo en el servidor dentro de una carpeta que se creará y esta a su vez contendrá otra con la fecha en la se realizó la operación (Ejemplo: “/AlmacenExcel/ 2015-06-23”). (Ver Figura 8: Insertar Hoja Excel, Ver Detalles de Hoja Excel, Editar Hoja Excel y Agregar Archivo Excel)

Filtro de búsqueda de Hoja Excel

Nombre:

Buscar
Reiniciar


Resultados de búsqueda de Hoja Excel

La búsqueda de Hoja Excel no retorno ningun resultado.

Insertar Hoja Excel Manualmente

Insertar Excel en la Base de Datos «

Agregar Archivo Excel



Examinar...
No se ha seleccionado ningún archivo.
Subir

**Figura 8: Insertar Hoja Excel, Ver Detalles de Hoja Excel, Editar Hoja Excel y
Agregar Archivo Excel**

Fuente: Elaboración propia.

2.3 Requisitos del sistema.

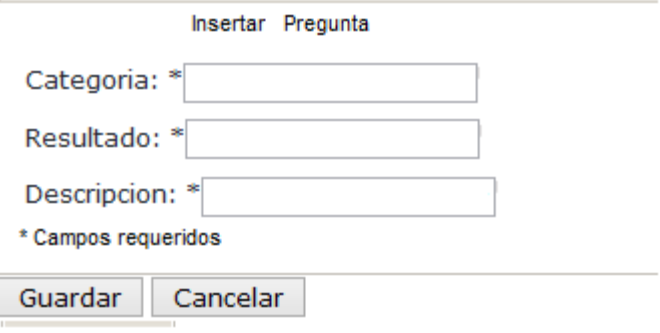
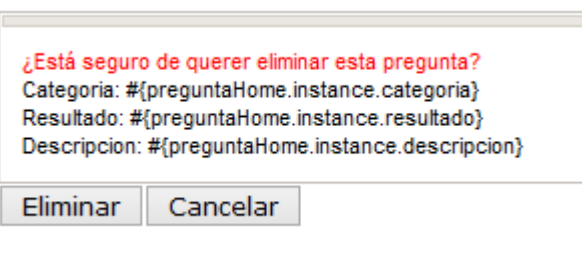
2.3.1 Requisitos Funcionales.

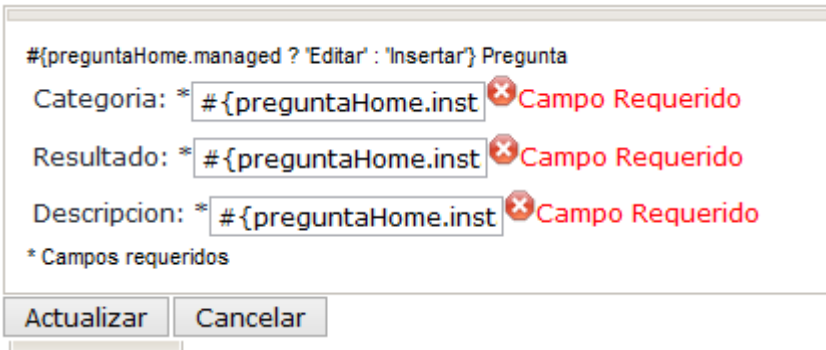
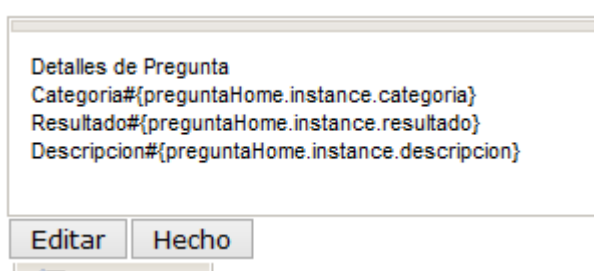
Los requisitos funcionales definen las condiciones o capacidades que el sistema será capaz de realizar (20). Los requisitos definidos para el sistema que se propone en esta investigación se muestran a continuación.

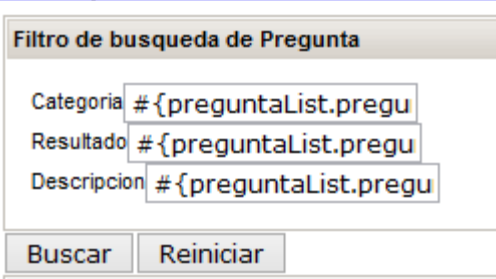
- RF1: Autenticar Usuario
- RF2: Gestionar Usuario
 - RF2.1: Registrar Usuario
 - RF2.2: Modificar Usuario
 - RF2.3: Mostrar Usuario
 - RF2.4: Eliminar Usuario
 - RF2.5: Buscar Usuario
- RF3: Gestionar Estudiante
 - RF3.1: Insertar Estudiante
 - RF3.2: Modificar Estudiante
 - RF3.3: Mostrar Estudiante
 - RF3.4: Eliminar Estudiante
 - RF3.5: Buscar Estudiante
- RF4: Gestionar Profesor
 - RF4.1: Insertar Profesor
 - RF4.2: Eliminar Profesor
- RF7: Importar Datos desde EXEL
- RF4.3: Modificar Profesor
- RF4.4: Mostrar Profesor
- RF4.5: Buscar Profesor
- RF5: Gestionar Instrumento
 - RF5.1: Insertar Instrumento
 - RF5.2: Eliminar Instrumento
 - RF5.3: Modificar Instrumento
 - RF5.4: Mostrar Instrumento
 - RF5.5: Buscar Instrumento
- RF6: Gestionar Pregunta
 - RF6.1: Insertar Pregunta
 - RF6.2: Eliminar Pregunta
 - RF6.3: Modificar Pregunta
 - RF6.4: Mostrar Pregunta
 - RF6.5: Buscar Pregunta

Tabla 1: Descripción de los requisitos funcionales asociados a la agrupación Gestionar Pregunta.

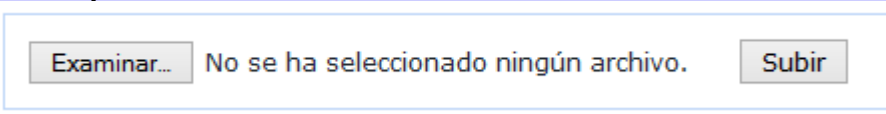
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF 6.1	Insertar pregunta	Inserta una pregunta en el sistema	media	baja

	Prototipo 			
	Campos	Tipos de Datos	Reglas o Restricciones	
	<ul style="list-style-type: none"> Tipo Descripción Resultado 	<ul style="list-style-type: none"> cadena cadena cadena 	El usuario debe estar ya autenticado.	
	Observaciones			
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF 6.2	Eliminar Pregunta	Elimina del sistema la pregunta seleccionada.	media	baja
	Prototipo 			
	Campos	Tipos de Datos	Reglas o Restricciones	
	<ul style="list-style-type: none"> Tipo Descripción Resultado 	<ul style="list-style-type: none"> Cadena Cadena Cadena 	Para eliminar la pregunta primero hay que seleccionarla o buscarla manualmente en la lista de preguntas.	
	Observaciones			
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF 6.3	Modificar Pregunta	Modifica los datos de una pregunta seleccionada	media	baja

	Prototipo 			
	Campos <ul style="list-style-type: none"> Tipo Descripción Resultado 	Tipos de Datos <ul style="list-style-type: none"> Cadena Cadena Cadena 	Reglas o Restricciones Ninguno de los campos puede estar vacíos.	
	Observaciones			
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF 6.4	Mostrar Pregunta	Muestra los datos de una pregunta seleccionada.	media	baja
	Prototipo 			
	Campos <ul style="list-style-type: none"> Tipo Descripción Resultado 	Tipos de Datos <ul style="list-style-type: none"> Cadena Cadena Cadena 	Reglas o Restricciones Para que se muestre una pregunta, el usuario debe escribir algún dato de la misma o buscarla manualmente en la lista de preguntas.	
	Observaciones			
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF 6.5	Buscar Pregunta	Busca en el sistema una pregunta seleccionada.	media	baja

Prototipo 		
Campos	Tipos de Datos	Reglas o Restricciones
<ul style="list-style-type: none"> Tipo Descripción Resultado 	<ul style="list-style-type: none"> Cadena Cadena Cadena 	Para buscar una pregunta en el sistema, se debe insertar datos en los filtros de búsqueda.
Observaciones		

Descripción del requisito funcional Insertar Datos desde Excel.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF 7	Importar Datos desde Excel	Insertar en el sistema la información que genera el SE procesando y guardando los datos en la Base de Datos	Alta	Alta
Prototipo 				
Campos		Tipos de Datos	Reglas o Restricciones	
<ul style="list-style-type: none"> Botón examinar 		<ul style="list-style-type: none"> cadena 	<ul style="list-style-type: none"> Para subir un fichero al servidor, este solo puede ser de tipo *.xls o *.xlsx 	
Observaciones				

2.3.2 Requisitos No Funcionales.

Los requisitos no funcionales no van asociados a casos de uso concretos y consisten en restricciones impuestas por el entorno y tecnologías, especificaciones sobre tiempo de respuesta o volumen de información tratado por una unidad de tiempo, requisitos en cuanto a interfaces, extensibilidad y facilidad de mantenimiento.

2.3.2.1 Usabilidad

- El sistema contiene términos y conceptos obtenidos de la experiencia de las personas que más utilizarán el sistema, para ello los nombres de las interfaces deben corresponderse con los que utilizan los especialistas en esta área. (Ejemplo: Instrumento)
- El comportamiento del sistema no debe provocar sorpresa a los usuarios, o sea, que debe quedarle claro ante un error, cuál fue la causa mediante la descripción del mismo.

2.3.2.2 Seguridad

- Se presentará como interfaz del sistema la autenticación, cada usuario debe ingresar su nombre de usuario único y una clave.

2.3.2.3 Integridad

- Se presentará como interfaz del sistema la autenticación, cada usuario debe ingresar su nombre de usuario único y una clave para acceder a cualquier funcionalidad del sistema.

2.3.2.4 Disponibilidad

- Se garantizará el acceso de cada usuario del sistema las 24 horas del día.

2.3.2.5 Interfaz de usuario

- La interfaz de usuario debe poseer un diseño sencillo con una interfaz amigable donde predominen colores claros, un tamaño de letra de 14, así como imágenes e íconos sugerentes para cada acción que el usuario vaya a realizar.

2.3.2.6 Requisitos de software

En necesario que el servidor esté desplegado con estas características:

- Windows Server 2000 (o superior), Linux/Unix (cualquier distribución) o Mac OS.
- Java v7 (o superior), JBoss v4.2 y Framework Seam v2.1.

- Servidor de base de datos PostgreSQL v9.2.4.1

En las computadoras clientes:

- Un navegador web: Firefox 3.0 (o superior), Opera 8 (o superior), Chrome 6 (o superior), Internet Explorer 9 (o superior), Safari 10 (o superior).

2.3.2.7 Requisitos de hardware

Para un rendimiento aceptable en el uso del sistema cada máquina cliente debe contar con:

- Procesador Intel Pentium IV 2.8 GHz (o superior), o variante AMD Athlon 64x2 (o superior).
- Memoria RAM de 512 Mb (o superior).

El hardware para el servidor debe contar con:

- Procesador Intel Pentium IV 2.8 GHz (o superior), o variante AMD Athlon 64x2 (o superior).
- Memoria RAM de 1 Gb (o superior).

2.4 Descripción de casos de uso:

Un actor es una entidad externa que interacciona con el sistema participando en un caso de uso. Los actores pueden ser gente real, otros ordenadores o eventos externos. Los actores no representan a personas físicas o a sistemas, sino su rol. Esto significa que cuando una persona interactúa con el sistema de diferentes maneras (asumiendo diferentes papeles), estará representado por varios actores. A continuación se muestra Tabla 2: Descripción de los actores que intervienen en los casos de uso.

Actor	Descripción
Usuario	Es cualquier profesor al que se le asigne la responsabilidad de procesar estas encuestas. Posee privilegios para gestionar

	personas, estudiantes, profesores, instrumentos y preguntas. Además puede realizar la carga del Excel en la base de datos así como la gestión de las hojas Excel y su contenido.
Administrador	Incluye los privilegios de usuario, añadiendo las operaciones para gestionar los usuarios.

Tabla 2: Descripción de los actores que intervienen en los casos de uso.

Fuente: Elaboración propia.

El sistema estará confeccionado para que solo personas identificadas como profesores cuyos usuarios tengan permiso, sean los únicos con acceso al mismo, además del Administrador

2.4.1 Diagrama de casos de uso del sistema

Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores. En el contexto de ingeniería del software, un caso de uso 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. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Una relación es una conexión entre los elementos del modelo, por ejemplo la especialización y la generalización son relaciones. Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo. (25)

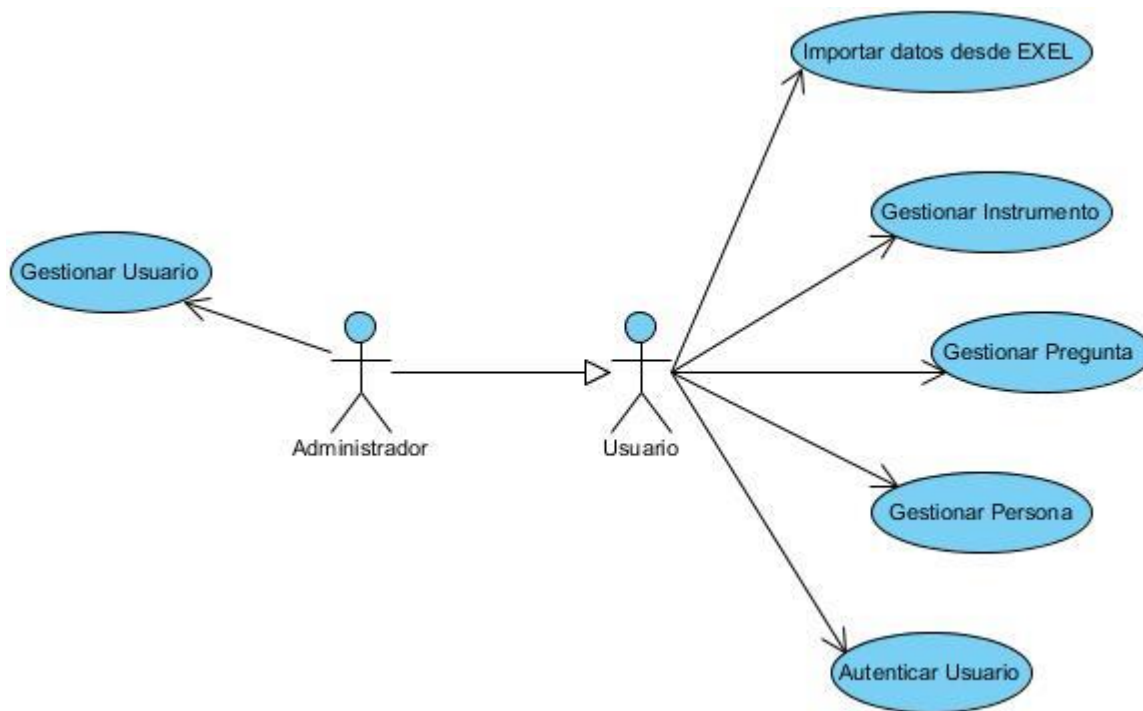


Figura 9: Diagrama de casos de uso.

Fuente: Elaboración propia.

Se detalla a continuación la descripción del caso de uso Gestionar Pregunta.

2.4.1.1 Tabla 4 CU 1. Gestionar Pregunta.

Nombre del caso de uso	Gestionar Pregunta.	
Actores	Usuario.	
Propósito	Permite Insertar, Modificar y Eliminar datos acerca de los Pregunta.	
Resumen	Es aquí donde se insertan, modifican o eliminan los datos de la Pregunta que se almacena en la base de datos.	
Prioridad	media	
Precondiciones	El Usuario del sistema ya autenticado.	
Poscondiciones	Se inserta una pregunta, se modifica o se elimina.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	➤ El Usuario del sistema necesita insertar, eliminar o modificar los datos de un instrumento.	El sistema ejecuta alguna de las siguientes acciones: a) Si decide insertar una pregunta, ir a la sección "Insertar Pregunta". b) Si decide modificar una pregunta, ir a la sección "Editar Pregunta". c) Si decide eliminar una pregunta, ir

		a la sección "Eliminar Pregunta".
Seccion1: "Insertar Pregunta"		
Flujo básico: Insertar Pregunta		
	Actor	Sistema
2.	➤ El Usuario del sistema introduce los datos de una pregunta para realizar su registro en la aplicación.	➤ 2.1 El sistema verifica que el campo de Nombre esté lleno. 2.2 El sistema verifica que esta Pregunta no exista. 2.3 La Pregunta se almacena en el sistema. 2.4 Se muestra un mensaje informándole al Usuario que ya ha sido efectuado el registro de la pregunta y finaliza el caso de uso.
Curso Alterno		
		2.1 Se emite un mensaje para que llene los campos obligatorios 2.3 Si la pregunta existe se muestra un mensaje informativo y finaliza el caso de uso.
Seccion2: "Editar Pregunta"		
Flujo básico: Editar Pregunta		
	Actor	Sistema
2	➤ El Usuario del sistema selecciona la pregunta a modificar.	2.1 El sistema brinda la posibilidad de modificar los datos.
3	1.El Usuario del sistema realiza la actualización deseada.	3.1 Se verifican que los campos obligatorios estén llenos. 3.2 Se actualiza la información y finaliza el caso de uso.
Curso Alternativo		
		3.1 Se emite un mensaje para que llene los campos obligatorios.
Seccion3: "Eliminar Pregunta"		
Flujo básico: Eliminar Pregunta		
	Actor	Sistema
2	➤ El Usuario del sistema selecciona la Pregunta a eliminar.	2.1 El sistema pide confirmación. 2.2 El sistema elimina la pregunta.
Curso Alternativo		
		2.1 Termina el caso de uso si no obtiene confirmación.

Tabla 3: Descripción del caso de uso Gestionar Pregunta

Fuente: Elaboracion propia

2.5 Diagrama de clases del diseño con Estereotipos Web

El diagrama de clase muestra los bloques de construcción de cualquier sistema orientado

a objetos. Los diagramas de clases describen la vista estática del modelo o parte del modelo, describiendo que atributos y comportamientos tienen en lugar de detallar los métodos para realizar operaciones. Los diagramas de Clase son más útiles para ilustrar relaciones entre clases e interfaces. Las generalizaciones, agregaciones, y asociaciones son todas valiosas al reflejar herencias, composición o uso, y conexiones respectivamente. (26)

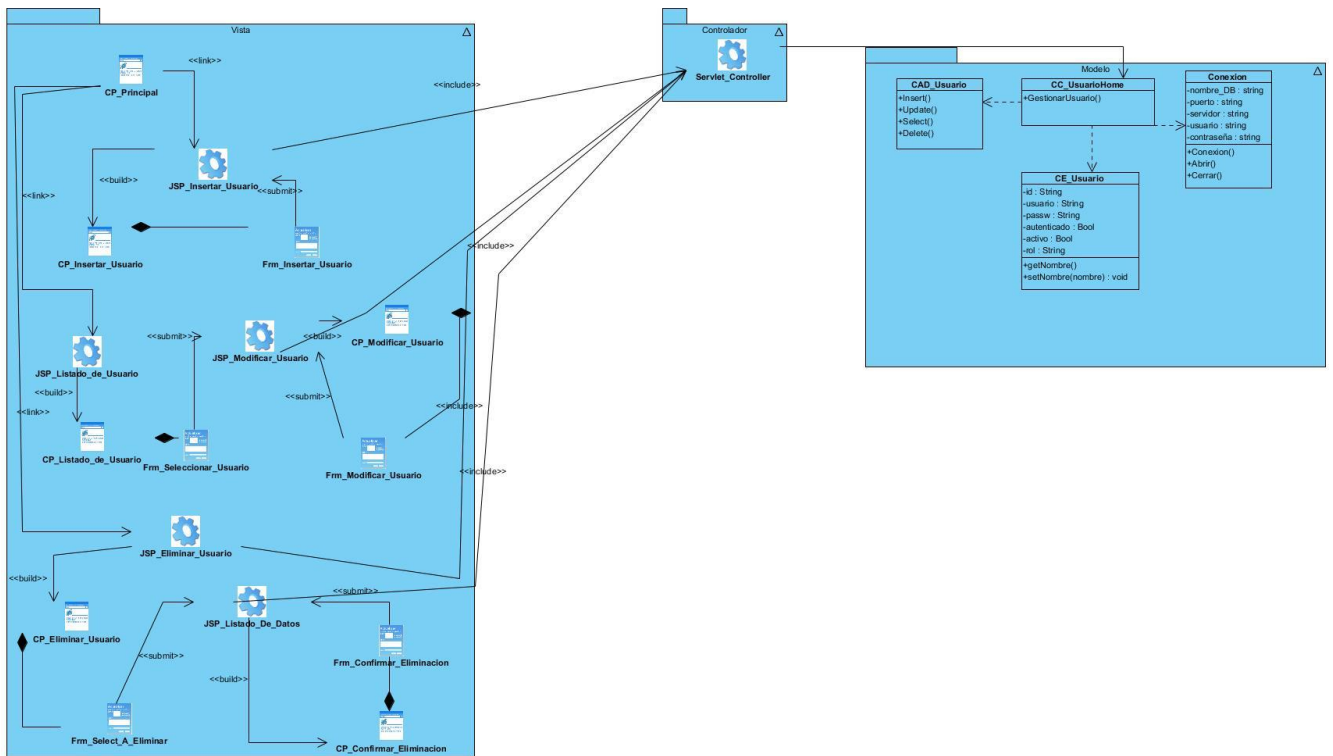


Figura 10: Diagrama de clases de Estereotipos Web Gestionar Usuario

Fuente: Elaboración propia.

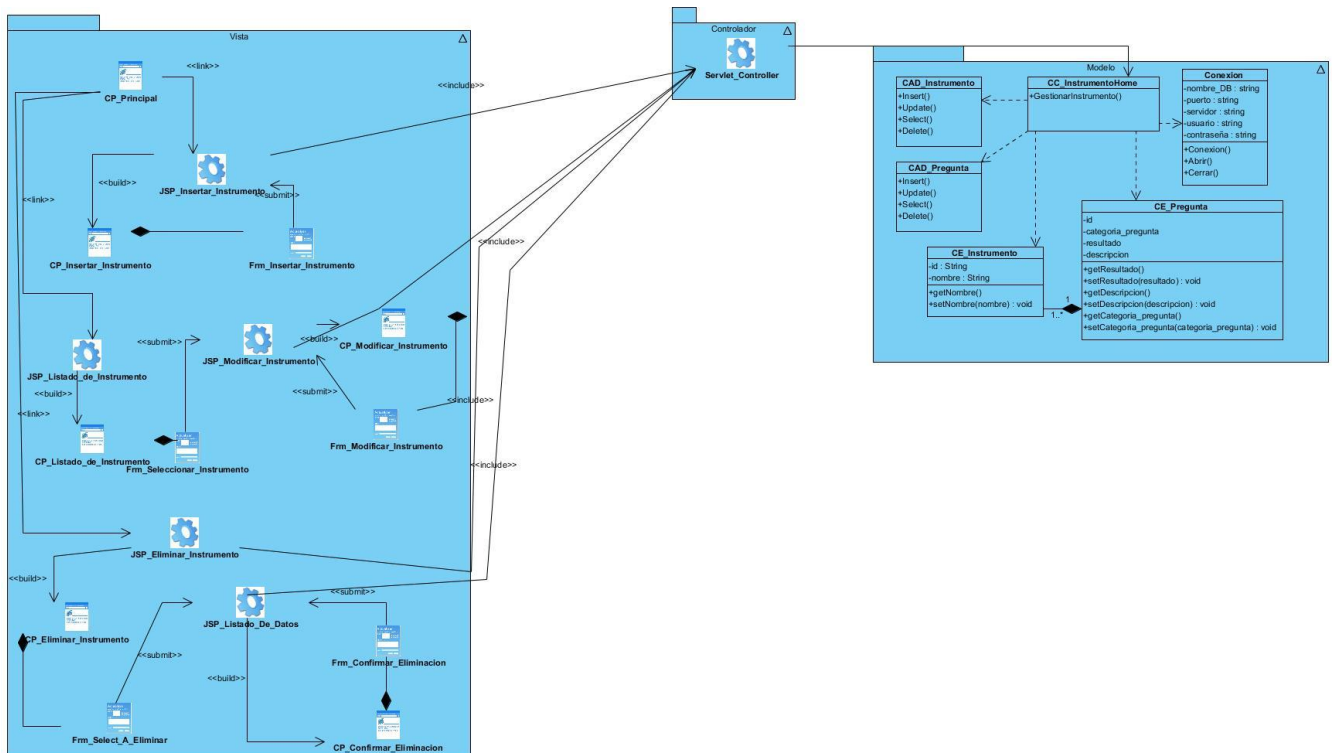


Figura 11: Diagrama de clases de Estereotipo Web Gestionar Instrumento

Fuente: Elaboración propia

Entre sus principales características se encuentran:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, solo 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.

2.6 Patrones de diseño

En una aplicación del alcance que se requiere para cubrir los requisitos del Sistema de Gestión de Estudiantes Potencialmente Talentosos en Informática, que cuenta con numerosas clases, se ha de contar con el empleo de patrones que garanticen un diseño claro y lo más simple posible. Un patrón es un par problema/solución con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones sobre sus compromisos. (21) La mayoría de estos patrones son

impuestos por los frameworks seleccionados, se explican a continuación los más relevantes aplicados en el sistema en cuestión.

Patrones de Asignación de Responsabilidades (GRASP)

Dentro de los patrones GRASP se encuentran: (21)

- **Bajo Acoplamiento:** El acoplamiento mide la fuerza con que una clase está conectada a otra, de esta forma una clase con bajo acoplamiento debe tener un número mínimo de dependencia con otras clases. Este patrón permite realizar un diseño de clases independientes que puedan soportar los cambios de una manera fácil y que a su vez permitan la reutilización. En la práctica, el nivel de acoplamiento no puede evaluarse sin tener en cuenta otros GRASP como la cohesión o el experto.
- **Creador:** consiste en determinar el objeto que debe ser responsable de crear una nueva instancia de la clase.
- **Controlador:** asigna la responsabilidad de gestionar mensajes en el sistema a una clase que represente al sistema global, a un subsistema, manejador de sesión o de eventos de un escenario de caso de uso.
- **Alta Cohesión:** Establece que la información que almacena una clase debe de ser coherente y está en la mayor medida de lo posible relacionada con la clase. La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una clase con alta cohesión, compartirá la responsabilidad de una operación con otras clases. Una clase con baja cohesión, concentrará las responsabilidades de una o muchas operaciones.
- **Experto en Información:** Se aplica para la asignación de responsabilidades a las clases de forma tal que las mismas contengan la información necesaria para poder ejecutar una acción específica, o sea, el controlador frontal le asigna responsabilidades a las clases controladoras para realizar acciones de acuerdo a las peticiones que recibe. El uso de este patrón permite a los objetos valerse de su propia información para hacer lo que se les pide, favorece la existencia de

mínimas relaciones entre las clases, lo que permite contar con un sistema robusto y fácil de mantener.

Este patrón se ve presente en las clases Controller, pues cada una de ellas tiene acceso a todas las instancias de todas las clases entidad de su tipo. Por ejemplo: la clase instrumentoController tiene acceso a todas las instancias de la clase entidad Instrumento. Es por eso que se le asigna a la clase instrumentoController la responsabilidad de listar, insertar, modificar y eliminar una instancia de la clase entidad Instrumento.

La Figura presenta el uso de los patrones GRASP en SIGEPTI:

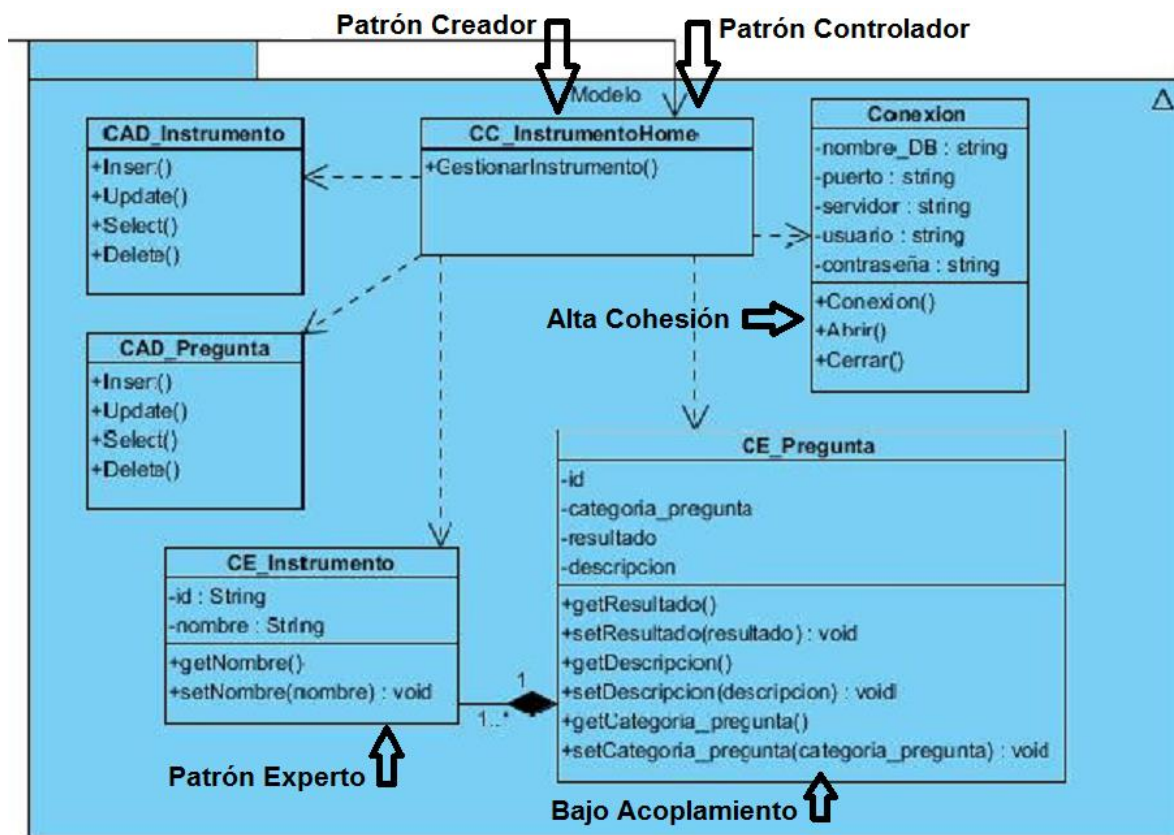


Figura 12: Ejemplo de uso de los patrones GRASP en SIGEPTI

Fuente: Elaboración propia

Patrones del Gang of Four (GOF)

Gang of Four es el nombre con el que se conoce comúnmente a los autores del libro [35] que referencia en el campo del diseño orientado a objetos. En la aplicación se ve presente la utilización de este patrón Template Method con el uso de las etiquetas “ui: define” en todos los archivos XHTML. A continuación se muestra un ejemplo:

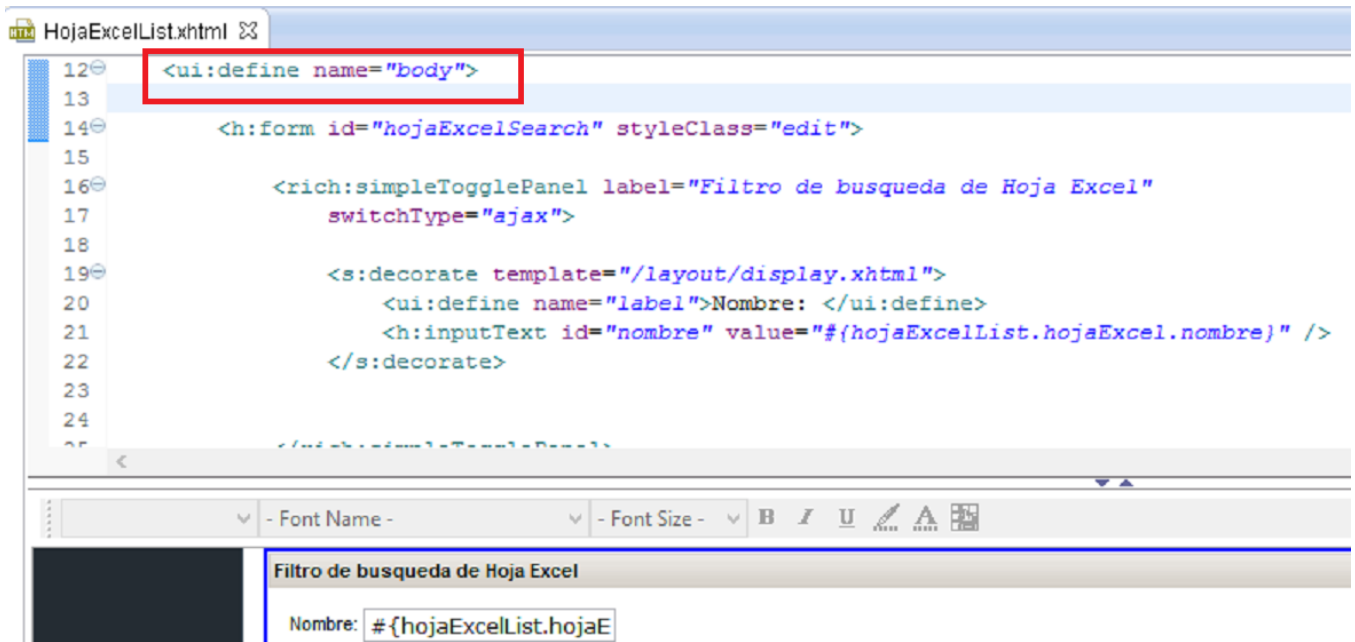


Figura 13: Ejemplo de uso de los patrones GRASP en SIGEPTI.

Fuente: Elaboración propia.

2.6 Diagramas de secuencia

El diagrama de secuencia es un tipo de diagrama usado para modelar interacción entre objetos en un sistema según UML. Muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Mientras que el diagrama de casos de uso permite el modelado de una vista business del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos. (27)

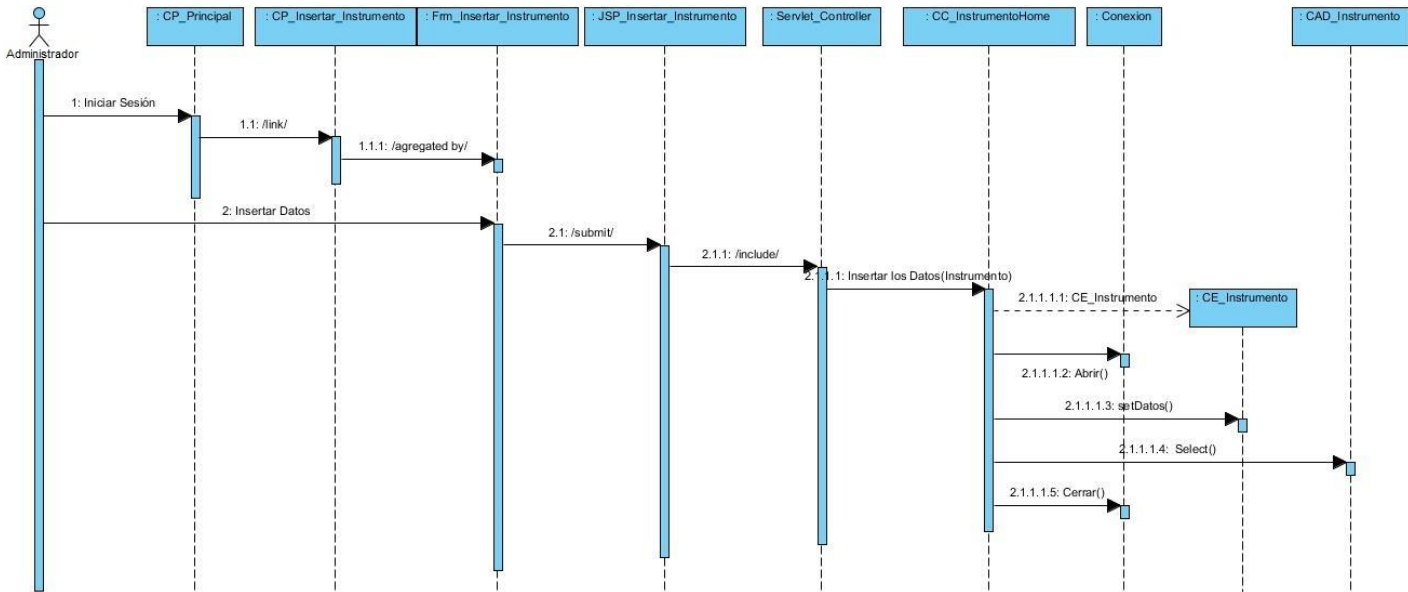


Figura 14: Diagrama de secuencia Insertar Instrumento

Fuente: Elaboración propia.

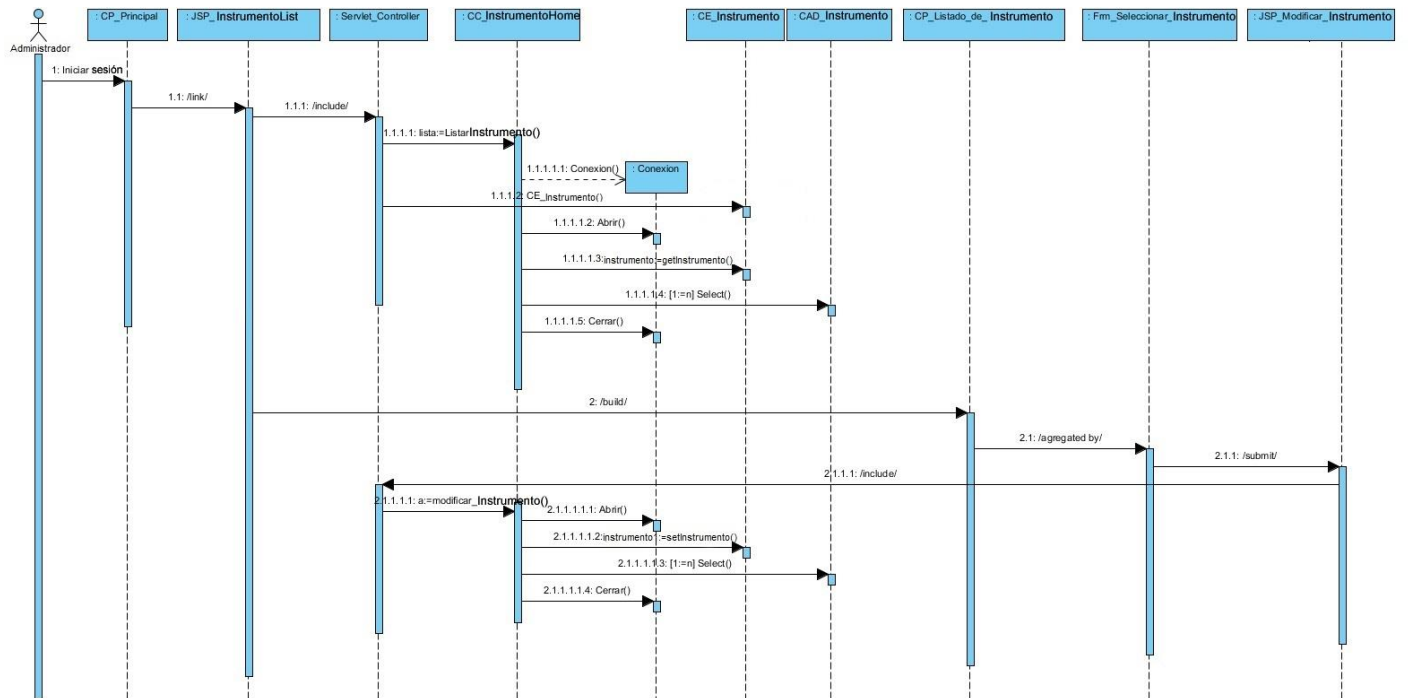


Figura 15: Diagrama de secuencia Modificar Instrumento.

Fuente: Elaboración propia.

2.7 Diseño de base de datos del sistema:

Un modelo de datos describe los datos que apoyan los procesos de una organización e ilustra gráficamente las personas, lugares o cosas acerca de la información capturada y cómo se relaciona entre sí. Refleja con exactitud cómo serán guardados los datos en la base de datos.[36]

La muestra el modelo de datos del SIGEPTI. Contiene cada una de las tablas de la base de datos de la aplicación, así como sus atributos y relaciones. El modelo se encuentra normalizado en Tercera Forma Normal, de las seis formas normales existentes para la normalización de las bases de datos relacionales, por lo que todos los atributos de las tablas del modelo, dependen únicamente de la llave primaria.

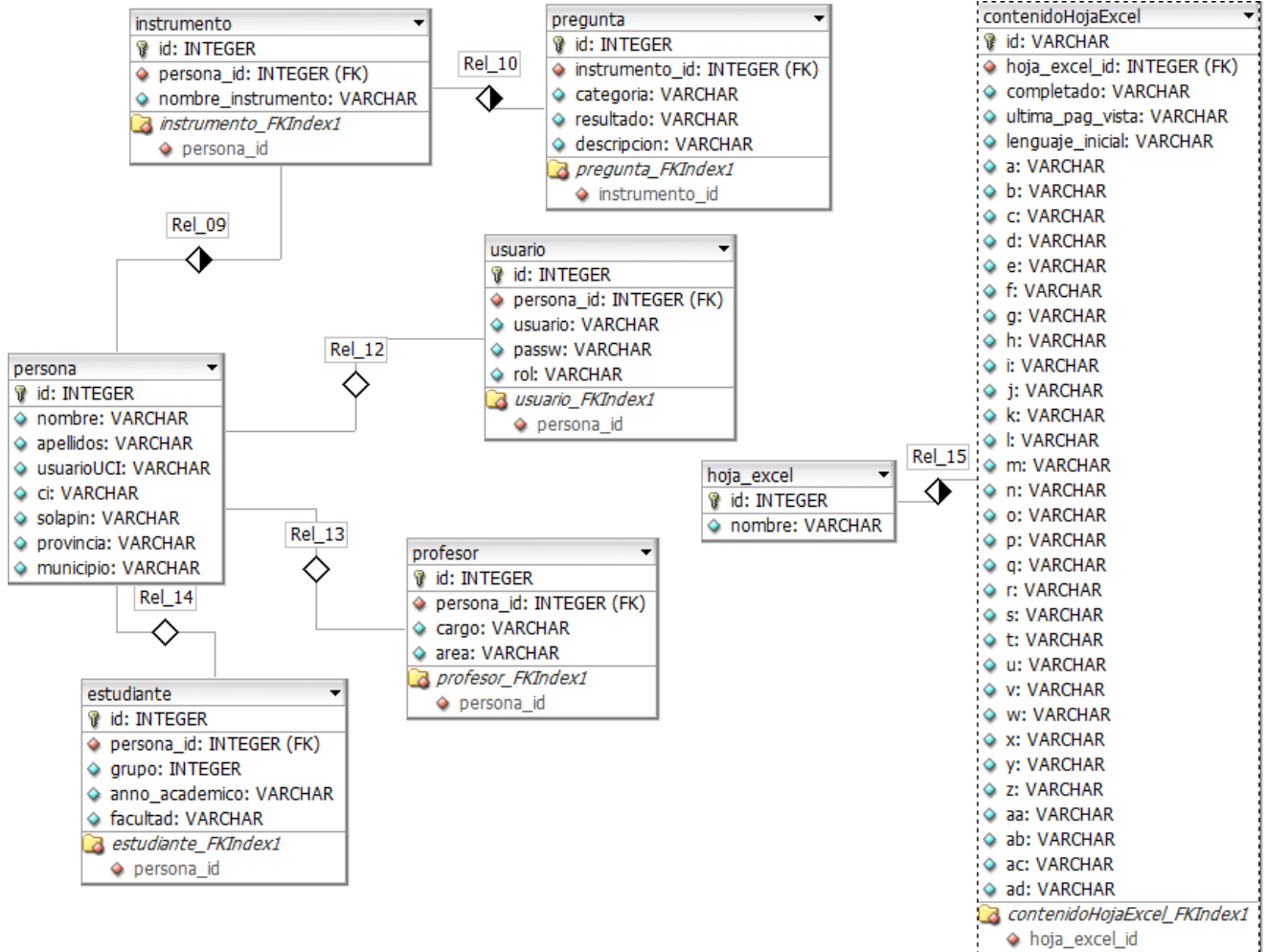


Figura 16. Diseño de Base de Datos

Fuente: Elaboración propia.

2.8 Diagrama de componentes

Un componente es una unidad de composición de aplicaciones de software, que posee un conjunto de interfaces y un conjunto de requisitos, y que ha de poder ser desarrollado, adquirido, incorporado al sistema y compuesto con otros componentes de forma independiente, en tiempo y espacio. (20)

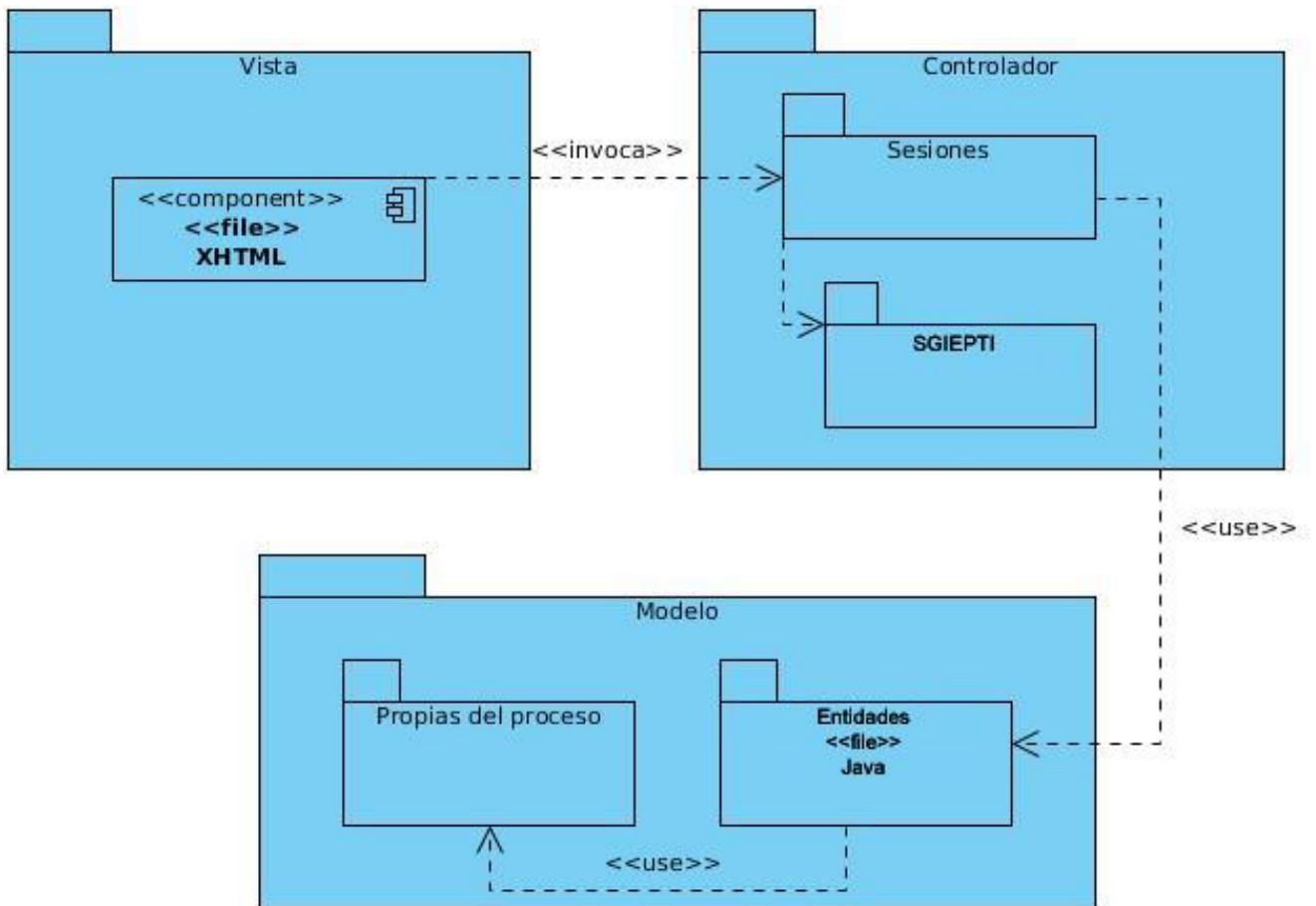


Figura 17: Diagrama de Componente.

Fuente: Elaboración propia.

2.9 Modelo de despliegue

Tomando como entrada las especificaciones de los componentes se arroja como salida el modelo de despliegue de la solución, ver Figura 11. El diagrama de despliegue representa

a los nodos y sus relaciones, donde los nodos son objetos físicos en tiempo de ejecución que representan un recurso computacional, generalmente con memoria y capacidad de procesamiento y pueden ser elementos de hardware o software.



Figura 18: Modelo de Despliegue.

Fuente: Elaboración propia.

2.10 Conclusiones parciales

En este capítulo se describieron las principales características del sistema a desarrollar. Como resultado del impulso de este proceso y para su mejor entendimiento se presentó la descripción del modelo conceptual y de la solución propuesta. Se definieron y documentaron los requisitos funcionales y no funcionales a tener en cuenta así como los actores y los casos de usos del sistema. Además se presentó la confección de los diagramas de clase del diseño con estereotipos web, de secuencia, el diseño de base de datos del sistema, así como el diagrama de componente y el modelo de despliegue.

Capítulo 3: Validación.

Introducción

En el presente capítulo se explica el proceso de diseño y aplicación de las pruebas a la aplicación, analizándose las no conformidades encontradas en cada una de las iteraciones hechas, valorándose así el producto de acuerdo a la seguridad, usabilidad, confiabilidad, interfaz de usuario, software y hardware.

3.1 Pruebas

Una vez generado el código fuente, el software debe ser probado para revelar y corregir el máximo de errores posible antes de su entrega al cliente, haciéndolo con la menor cantidad de tiempo y esfuerzo posible, “Las pruebas del software son un elemento para la garantía de la calidad del software y representan una revisión final de las especificaciones, del diseño y de la codificación” [37]. Como objetivo fundamental de la prueba según Pressman en la quinta edición del libro “Ingeniería de Software un enfoque práctico”[37] se tiene el siguiente: descubrir la máxima cantidad de errores no detectados hasta entonces. Aun así, es necesario señalar que las pruebas no pueden asegurar la ausencia de errores, solo puede demostrar que existen defectos en el software.

3.1.1 Estrategia de Pruebas

Se definió una estrategia de pruebas con niveles y métodos de prueba, que permitió solucionar errores que presentaba la aplicación y perfeccionar la solución implementada.

Dentro de los niveles generales de pruebas se encuentran:

- Pruebas Internas.
- Pruebas Cruzadas.
- Pruebas de Liberación.
- Pruebas Piloto.

La estrategia definida se centra en el nivel de Pruebas Internas, donde a su vez se encuentran los niveles:



Figura 19. Niveles de Prueba.

Fuente: Modificado a partir de [37]

Dentro de estos niveles de pruebas, la estrategia elaborada utiliza en el nivel Prueba de Unidad, que es la prueba enfocada a los elementos más pequeños del software. Es aplicable a componentes representados en el modelo de implementación para verificar que los flujos de control y de datos están cubiertos, y que ellos funcionen como se espera[37].

Las pruebas unitarias están basadas en la funcionalidad de los componentes y permiten asegurar que cada uno funciona correctamente por separado. En ellas, los errores están más acotados y son más fáciles de localizar, lo que facilita a los desarrolladores la solución de los mismos permitiendo llegar a la integración con un mayor grado de seguridad del correcto funcionamiento de cada parte del sistema por separado. Para ejecutar las pruebas de unidad se seleccionan las clases de pruebas que fueron diseñadas para cada funcionalidad de forma independiente. Esta actividad tiene como entrada las clases de pruebas, las mismas son ejecutadas por los desarrolladores, y tienen como salida el resultado de las pruebas, que incluye la evaluación de cada una de las funciones que fueron probadas en las clases de prueba, mostrando si devolvieron el valor esperado, el tiempo de ejecución de las mismas y el producto sin los errores encontrados. Como resultado de estas pruebas unitarias se obtiene una versión del producto con todos los errores que fueron detectados. [38]

3.1.2 Método de Prueba. Método de Caja Negra

Se denomina método de caja negra a aquel elemento que es estudiado desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno. En otras palabras, de una caja negra interesa su forma de interactuar con el medio que le rodea, entendiendo qué es lo que hace, pero sin dar importancia a cómo lo hace. Por tanto, de una caja negra deben estar muy bien definidas sus entradas y salidas, es decir, su interfaz; en cambio, no se precisa definir ni conocer

los detalles internos de su funcionamiento. Estas pruebas se realizaron con el objetivo de encontrar: [38]

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las bases de datos externas.
- Errores de inicialización y terminación.

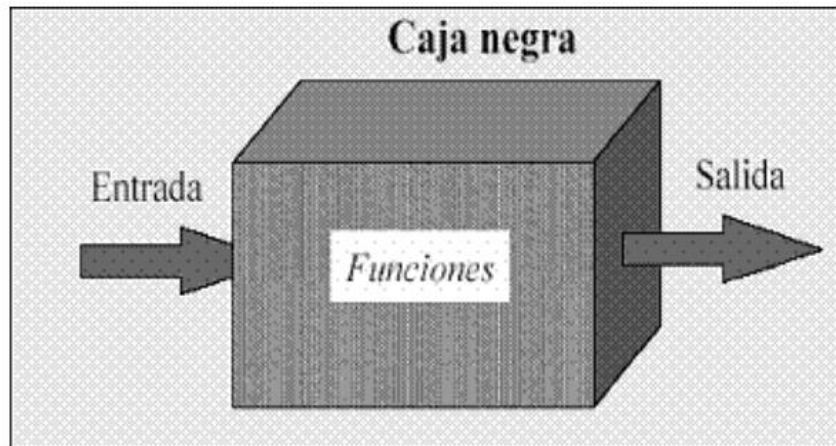


Figura 20. Método de Caja Negra.

Fuente: Modificado a partir de [37]

Para la realización de las pruebas se confeccionaron casos de prueba, que son un conjunto de entradas, condiciones de ejecución y resultados esperados diseñados para un objetivo particular. Los casos de prueba fueron diseñados previamente y utilizados fundamentalmente con el objetivo de determinar si los requisitos especificados fueron cumplidos.

A continuación se muestra el diseño de casos de prueba del caso de uso Gestionar Pregunta.

Diseño de caso de prueba: Caso de uso Gestionar Pregunta.

Condiciones de ejecución

- El usuario debe estar autenticado.

Tabla 4 Descripción de los escenarios del requisito Insertar Pregunta

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1: Insertar Pregunta.	Permite insertar una Pregunta en el sistema.	EP 1.1: "Inserción exitosa"	<ul style="list-style-type: none"> Flujo básico
		EP 1.2: "Faltan datos obligatorios"	<ul style="list-style-type: none"> Flujo alternativo 1.
		EP 1.3: "Ya existe la pregunta"	<ul style="list-style-type: none"> Flujo alternativo 2.

Tabla 5 Casos de prueba del requisito Insertar Pregunta

Id del escenario	Escenario	Variable 1	Variable 2	Variable 3	
		Tipo	Descripción	Resultado	
EP 1.1	Inserción exitosa		Capacidad para identificar, formular y resolver problemas de manera independiente.)		Se muestra un mensaje informándole al Usuario que ya ha sido efectuado el registro de la pregunta y finaliza el caso de uso.
EP 1.2	Faltan datos obligatorios		Capacidad para identificar, formular y resolver problemas de manera independiente.)		Se emite un mensaje para que llene los campos obligatorios.
			Capacidad para identificar, formular y resolver problemas de manera independiente.)		
EP 1.3	Ya existe la pregunta		Capacidad para identificar, formular y resolver problemas de		Se emite un mensaje para que llene los campos obligatorios.

			manera independiente.)		
--	--	--	------------------------	--	--

Tabla 6 Descripción de variables

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Tipo	Campo de texto.	No	Solo acepta letras.
2	Descripción	Campo de texto.	No	Acepto letras y números.
3	Resultado	Campo de texto.	No	Solo acepta letras.

3.1.2 Resultados de las pruebas de Caja Negra

Con el objetivo de encontrar la mayor cantidad de anomalías en la aplicación, se diseñó y ejecutó los diseños de casos de pruebas correspondientes, obteniendo 9 no conformidades en la primera iteración, las cuales las 9 son significativas (1 opción que no funciona, 8 errores técnicos). De las no conformidades detectadas se solucionaron 8 significativas. Para una segunda iteración de pruebas, se encontró 1 no conformidad significativa (1 error técnico), la cual se solucionó. Se ejecutó una tercera iteración obteniéndose un resultado satisfactorio para cada una de las combinaciones de datos por escenario, ya que no se encontraron no conformidades. La siguiente figura muestra estos resultados en forma de gráfica:

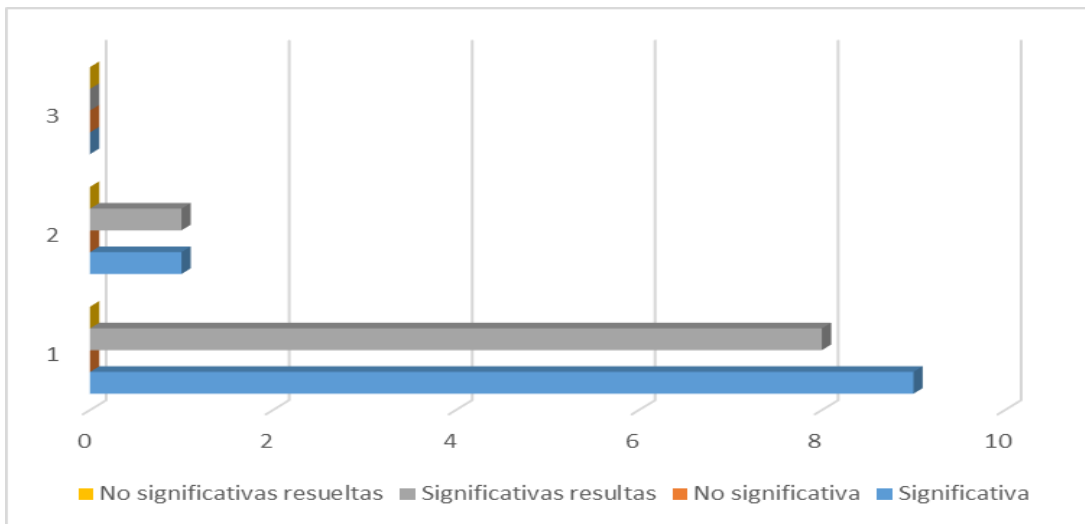


Figura 21. Resultados de las pruebas de Caja Negra.

Fuente: Elaboración propia

3.3 Conclusiones parciales

En el presente capítulo se explicó el proceso de diseño y aplicación de las pruebas a la aplicación, específicamente se utilizó el método de caja negra para verificar la entrada de datos. Se analizaron las no conformidades encontradas en cada una de las iteraciones hechas, valorándose así el producto de acuerdo a la seguridad, usabilidad, confiabilidad, interfaz de usuario, software y hardware.

Conclusiones

A modo de conclusión se puede afirmar:

- ❖ El estudio teórico realizado permitió establecer las relaciones entre el talento informático y sus indicadores, el procesamiento de encuestas para la identificación de estudiantes potencialmente talentosos en esta rama y el uso de metodologías, lenguajes, tecnologías y herramientas para el desarrollo de la aplicación propuesta como solución.
- ❖ Los instrumentos aplicados en la primera etapa de la investigación permitieron corroborar las dificultades que existían en cuanto al procesamiento de encuestas para la identificación de la información de estudiantes potencialmente talentosos.
- ❖ La aplicación web que se propone se caracteriza por ofrecer un alto grado de seguridad para la información que se recopila, permite la integridad de la misma ya que no puede ser libremente modificada pues solamente podrán tener acceso las personas autorizadas, cuenta con una total disponibilidad pues su servicio estará activo las 24 horas, además propicia hacer copias de seguridad y guardar la información en caso de fallos; por su concepción muestra un diseño sencillo y una interfaz amigable para el usuario.

Recomendaciones

Como futuras mejoras al sistema pueden encontrarse:

- ❖ Dotar al sistema de una herramienta para la generación de informes para que el usuario final pueda extraer en PDF u otro formato la relación de los estudiantes.
- ❖ Analizar otras funcionalidades que se le puedan incluir al sistema como la de realizar copias de respaldo para evitar la pérdida de información.

Bibliografía

- [1] VALLEJO PRATTS, Gisela a Gisela Susana ROSADO VÁZQUEZ. *El uso de las Tic's en la educación básica de jóvenes y adultos de comunidades rurales y urbanas del sureste de México*. La Habana, Cuba, 2014. Universidad de las Ciencias Informáticas.
- [2] LORENZO GARCÍA, Raquel a Miguel David ROJAS LÓPEZ. *Talento y Creatividad para la ciencia y la innovación tecnológica*. Medellín, Colombia, 2006. Universidad Nacional de Colombia.
- [3] MENÉNDEZ PÉREZ, Jorge S. a y. *Definición de talento y talento informático en el marco del Proyecto Talenmático*. [Digital] Buenos Aires, 2012. Centro de Estudios Internacionales para el Desarrollo.
- [4] GÁLVEZ, Daniel. *Curso de Sistemas Basados en el Conocimiento*. Villa Clara, Cuba: Universidad Central „Marta Abreu" de las Villas, 1998. T83-T825.
- [5] VERA SALAZAR, C. *Estrategia didáctica para el desarrollo del aprendizaje de los escolares con talento académico de la Educación Primaria, in Tesis en opción al grado de Doctora en Ciencias*. La Habana, Cuba, 2008. Instituto Superior Pedagógico Enrique José Varona.
- [6] CASTELLANOS SIMONS, D., Marta MARTÍNEZ LLANTADA a y. *Talento: concepciones y estrategias para su desarrollo en el contexto escolar*. La Habana, Cuba: Pueblo y Educación, 2009. 978-13-1789-6.
- [7] RODRÍGUEZ, Luis. Identificación y evaluación de niños con talento. In: *La Educación de Niños con Talento en Iberoamérica*. Santiago, Chile: Editorial Trineo S.A., 2004, s. 37–47.
- [8] CASTELLANOS SIMONS, Doris. Diagnóstico del potencial intelectual: La perspectiva del profesor en la identificación del talento. In: *Talento: concepciones y estrategias para su desarrollo en el contexto*. nedatováno, s. 119–127.
- [9] FLANAGAN, Andrea a Violeta ARANCIBIA. Talento Académico: Un Análisis de la Identificación de Alumnos Talentosos Efectuada por Profesores. *Psyche* [online]. 2005, roč. 14, č. 1, s. 121–135 [vid. 24. červen 2015]. Dostupné z: doi:<http://dx.doi.org/10.4067/S0718-22282005000100010>
- [10] ÁLVAREZ DE ZAYAS, Carlos. *Metodología de la investigación científica*. Santiago de Cuba, Cuba, 1995. CENTRO DE ESTUDIOS DE EDUCACIÓN SUPERIOR: Manuel F. Gran.
- [11] D., R. *Diplomado de Gobierno Abierto y Participativo Institucional* [online]. Monterrey, México, 2005. Universidad Virtual del Tecnológico de Monterrey. Dostupné z: http://www.cca.org.mx/funcionarios/cursos/ap066/material/m2met_enc.pdf
- [12] INSTITUTO NACIONAL DE ESTADÍSTICA Y GEOGRAFÍA (MÉXICO). *Proceso estándar para encuestas por muestreo*. Mexico: Instituto Nacional de Estadística y Geografía. 2010.

- [13] *SOFTWARE PARA ENCUESTAS - Promonegocios.net* [online]. [vid. 25. červen 2015]. Dostupné z: <http://www.promonegocios.net/directorio/software-encuestas.html>
- [14] *RotatorSurvey - Software para Encuestas - Aplicación para realizar encuestas* [online]. [vid. 25. červen 2015]. Dostupné z: <http://www.rotatorsurvey.com/index.php>
- [15] *Netquest - Servicios de encuestas online y panel online* [online]. [vid. 25. červen 2015]. Dostupné z: <http://www.netquest.com/es/software/tecnologia-encuestas-online.html>
- [16] *TGI - Mobile Systems* [online]. [vid. 25. červen 2015]. Dostupné z: <http://tgi.cl/>
- [17] *Sistema de Encuestas* [online]. [vid. 25. červen 2015]. Dostupné z: <https://encuesta.uci.cu/>
- [18] *Repositorio Institucional de la UCI: Sistema de Gestion de Encuestas* [online]. [vid. 25. červen 2015]. Dostupné z: http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_03593_10
- [19] FERNÁNDEZ MEDINA, Eduardo. *Ingeniería del Software I Cuarto Curso* [online]. 2007 2006. Dostupné z: <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>
- [20] JACOBSON, Ivar, Grady BOOCH a James RUMBAUGH. El proceso unificado de desarrollo Parte I. In: *El Proceso Unificado de Desarrollo de Software*. B.m.: Addison Wesley, nedatováno.
- [21] KRUP. *The Rational Unified Process: An Introduction*. B.m.: Addison Wesley, 2000.
- [22] LARMAN, Craig. *UML Y PATRONES. Inreoducción al análisis y diseño orientado a objeto*. B.m.: Pearson, nedatováno.
- [23] PONS, C., Roxana GIANDINI a Gabriela PÉREZ. *Desarrollo de software dirigido por modelos. Conceptos teóricos y su aplicación práctica*. 1ra vyd. Argentina: Universidad Nacional de La Plata, 2010. ISBN 978-950-34-0630-4.
- [24] ORTIZ BATISTA, Y. *Sistema integrado de gestión de Ciencia, Tecnología e Innovación en la Universidad de las Ciencias Informáticas*. La Habana, Cuba, 2011. Universidad de las Ciencias Informáticas.
- [25] POSTGRESQL. *What is PostgreSQL?* [online]. 20. květen 2009. Dostupné z: http://wiki.postgresql.org/wiki/FAQ#What_is_PostgreSQL.3F_How_is_it_pronounced.3F_What_is_Postgres.3F
- [26] *¿Qué es Java? Características del lenguaje Java* [online]. 1997 [vid. 20. leden 2015]. Dostupné z: <http://www.iec.csic.es/criptonomicon/java/quesjava.html>
- [27] GHIA, Dustin. *Programación Práctica. JEE5 - Fundamentos* [online]. 31. březen 2011 [vid. 15. prosinec 2014]. Dostupné z: <http://programmabilis.blogspot.com/2011/03/i1-fundamentos-de-jee5.html>
- [28] LATINA, Osmosis. *JVM („Java Virtual Machine“)* [online]. 2000 [vid. 16. únor 2015]. Dostupné z: <http://www.osmosislatina.com/java/basico.htm>

- [29] LIBROSWEB. *HTML Y XHTML* [online]. 2010 [vid. 6. únor 2015]. Dostupné z: http://www.librosweb.es/xhtml/capitulo_1/html_y_xhtml.html
- [30] FERGUSON, Jhon. 2007 [vid. 2. únor 2015]. Dostupné z: http://www.wakaleo.com/public_resources/jsf-jumpstarter.pdf.
- [31] ASSAR, Nirav a Jacob ORSHALICK. *JBOSS SEAM: AGILE RIA DEVELOPMENT FRAMEWORK* [online]. Dostupné z: https://www.redhat.com/f/pdf/JB_Seam_RIAdevFW_2116957_0510_cw_web.pdf
- [32] JBOSS COMMUNITY. *HIBERNATETools. Hibernate Tools for Eclipse and Ant* [online]. [vid. 15. únor 2015]. Dostupné z: <http://www.hibernate.org/subprojects/tools.html>
- [33] UNIVERSIDAD DE LOS ANDES. *Departamento de Sistemas. Enterprise Java Bean 3. Características* [online]. [vid. 15. únor 2015]. Dostupné z: <http://sistemas.uniandes.edu.co/~isis3702/dokuwiki/lib/exe/fetch.php?media=principal:isis3702-ejb3.pdf>
- [34] FREE DOWNLOAD MANAGER. *Visual Paradigm para UML* [online]. březen 2007 [vid. 17. únor 2015]. Dostupné z: http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/
- [35] GAMMA, Erich, Richard HELM, Ralph JOHNSON a John VLISSIDES. *Design Patterns*. 32. vyd. 2005. ISBN 0-201-63361-2.
- [36] NTAULA. *Análisis Y diseño de sistemas. Diagramación: Modelo de datos* [online]. Dostupné z: <http://ntaula0.tripod.com/ads/anamdatos.html>.
- [37] PRESSMAN, Roger S. *Ingeniería de software. Un enfoque practico*. [online]. 1998 [vid. 2. duben 2015]. Dostupné z: En línea
- [38] GONZÁLEZ, Benjamín a Pedro BODA. *Zend Framework*. 2009.

Anexos

Anexo 1: Instrumento para la autoidentificación de estudiantes potencialmente talentosos.

No	Indicadores de medición	Muy Bueno (5)	Bueno (4)	Normal (3)	Malo (2)	Muy Malo (1)
1	Capacidad para identificar, formular y resolver problemas de manera independiente.					
2	Capacidad para evaluar correctamente las soluciones y comunicar los resultados de forma clara y sencilla de manera independiente.					
3	Capacidad para valorar las tareas asignadas en función de sus posibilidades y potencialidades.					
4	Capacidad para el procesamiento efectivo de la información y para extrapolar el conocimiento a otras áreas.					
5	Resultados académico-investigativos-productivos destacados en una o varias áreas.					
6	Capacidad de reestructurar y reconstruir nuevos resultados (modelos, algoritmos, conceptos, tecnologías, técnicas), utilizando criterios propios y diferentes alternativas.					
7	Capacidad de concebir, promover, emprender, prever, ideas y acciones, con criterios propios.					
8	Capacidad de brindar respuestas variadas, modificar ideas, superar la rigidez, reestructurar la información que posee.					
9	Decidido enfrentamiento a subestimaciones, incomprensiones, desconocimiento de la importancia y novedad de lo realizado.					

10	Capacidad para reponerse de las frustraciones, perseverar y no perder la orientación.					
11	Elevado afán de búsqueda de competencias que le permitan destacarse por la eficiencia en su área de desempeño.					
12	Elevado nivel de concentración y disfrute con las tareas hasta concluir las exitosamente.					
13	Se implica en tareas relacionadas con su esfera motivacional, aunque no constituya una obligación formal.					
14	Se implica afectivamente en las tareas convirtiendo su ejecución en una preocupación y ocupación personal.					
15	Naturaleza y calidad de sus selecciones de ocupación del tiempo libre, que promuevan su crecimiento profesional y humano.					
16	Manifiesta propósitos definidos en la vida asociados a su desarrollo profesional.					
17	Responsable en su área de desempeño profesional y evidencia empeño de superación permanente.					
18	Manifiesta la necesidad del reconocimiento moral y material.					
19	Sus decisiones (personales e interpersonales) tienen un alcance social que cumple con los principios éticos de su entorno.					
20	Recibe apoyo y exigencia del entorno familiar y de la comunidad residencial.					
21	Contexto abierto y desafiante (estimulador, crítico-constructivo, que reconoce los logros oportuna y equilibradamente) y que posibilita espacios para exponer resultados.					
22	Se siente parte del contexto socio cultural en el que se desarrolla, porque satisface sus necesidades espirituales.					

23	Estado de salud general satisfactorio con sistema nervioso que permite la transformación de sus potencialidades en resultados relevantes socialmente válidos.					
24	Procesos de obtención de resultados que además de complejos, efectivos y exitosos, se corresponden con las exigencias sociales.					
25	Se empeña en obtener resultados pertinentes y de elevado significado social y, por tanto, reconocidos de forma relevante por la sociedad.					

Anexo 2: Instrumento para la identificación de estudiantes potencialmente talentosos, por sus condiscípulos.

No	Indicadores de medición	Muy Bueno (5)	Bueno (4)	Normal (3)	Malo (2)	Muy Malo (1)
1	Sus resultados académicos en una o varias áreas, se pueden considerar:					
2	Los resultados investigativos en una o varias áreas, se pueden considerar:					
3	Los resultados productivos en una o varias áreas, se pueden considerar:					
4	Se implica en tareas de su interés, aunque no constituya una obligación formal.					
5	Cómo lo valoras en la selección y ocupación del tiempo libre, para su desarrollo profesional y crecimiento humano.					
6	Estimas que tienes propósitos definidos en la vida asociados a su desarrollo profesional.					
7	Lo consideras responsable en su área de desempeño profesional y evidencia empeño de superación permanente.					
8	Cómo valoras sus manifestaciones de necesidad					

	de reconocimiento moral y material.					
9	Estimas que sus decisiones (personales e interpersonales) tienen un alcance social que cumple con los principios éticos de su entorno.					
10	Estimas que es decidido en el enfrentamiento a subestimaciones, incomprendiones, desconocimiento de la importancia y novedad de lo realizado por él.					
11	Cómo valoras el apoyo y exigencia que recibe del entorno familiar y de la comunidad residencial.					
12	El contexto en el que se desarrolla lo consideras abierto y desafiante y que posibilita espacios para exponer resultados.					
13	Consideras que se siente parte del contexto socio cultural en el que se desarrolla, porque satisface sus necesidades espirituales.					
14	Cómo consideras su estado de salud general y su sistema nervioso.					
15	Consideras que sus resultados además de complejos, efectivos y exitosos, se corresponden con las exigencias sociales.					
16	Estimas que sus resultados son pertinentes y de elevado significado social y, por tanto, reconocido de forma relevante por la sociedad.					

Anexo 3: Instrumento para la identificación de estudiantes potencialmente talentosos por los profesores.

No	Indicadores de medición	Muy Bueno (5)	Bueno (4)	Normal (3)	Malo (2)	Muy Malo (1)
1	Cómo consideras que es para identificar, formular y resolver					

	problemas de manera independiente.					
2	Cómo consideras que es para evaluar correctamente las soluciones y comunicar los resultados de forma clara y sencilla de manera independiente.					
3	Cómo consideras que es para valorar las tareas asignadas en función de sus posibilidades y potencialidades.					
4	Cómo consideras que es para el procesamiento efectivo de la información y para extrapolar el conocimiento a otras áreas.					
5	Cómo valoras sus resultados académico-investigativos-productivos.					
6	Cómo valoras su capacidad de reestructurar y reconstruir nuevos resultados (modelos, algoritmos, conceptos, tecnologías, técnicas), utilizando criterios propios y diferentes alternativas.					
7	Cómo valoras su capacidad de concebir, promover, emprender, prever, ideas y acciones, con criterios propios.					
8	Cómo valoras su capacidad de brindar respuestas variadas, modificar ideas, superar la rigidez, reestructurar la información que posee.					
9	Cómo valoras su enfrentamiento a subestimaciones, incomprendiones, desconocimiento de la importancia y novedad de lo realizado por él.					
10	Cómo valoras su capacidad para reponerse de las frustraciones, perseverar y no perder la orientación.					
11	Cómo lo percibes en el afán de búsqueda de competencias que le permitan destacarse por la eficiencia en su área de desempeño.					

12	Cómo valoras su nivel de concentración y disfrute con las tareas hasta concluir las exitosamente.					
13	Cómo es en la implicación en tareas de su interés, aunque no constituya una obligación formal.					
14	Consideras que se implica afectivamente en las tareas convirtiendo su ejecución en una preocupación y ocupación personal.					
15	Cómo valoras la naturaleza y calidad de sus selecciones de ocupación del tiempo libre.					
16	Cómo valoras sus propósitos en la vida asociados a su desarrollo profesional.					
17	Cómo lo percibes en su área de desempeño profesional.					
18	Cómo valoras su empeño de superación permanente.					
19	Cómo lo percibes en sus manifestaciones de necesidades de reconocimiento moral y material.					
20	Cómo valoras sus decisiones (personales e interpersonales) en cuanto al alcance social que cumple con los principios éticos de su entorno.					
21	Cómo valoras el apoyo y exigencia del entorno familiar y de la comunidad residencial, que recibe.					
22	Consideras que el contexto es proclive a su desarrollo y posibilita espacios para exponer resultados.					
23	Cómo valoras el contexto socio cultural en el que se desarrolla.					
24	Cómo percibes su estado de salud general y su con sistema nervioso.					
25	Cómo valoras sus resultados en cuanto a pertinencia, significado social y, por tanto, reconocimiento de forma relevante por la sociedad.					

Sus criterios son muy importantes para nuestro trabajo y serán objeto de un análisis serio y profundo. Le agradecemos su tiempo y colaboración.

Proyecto TALENMÁTICO

Anexo 4: Descripción de los indicadores definidos por el proyecto Talemático

Cognitiva-Instrumental			
Descripción	Peso(w_i)	Incertidumbre(I_i)	Evaluación
Capacidad para identificar y definir problemas en algunas de las áreas de las Ciencias Informáticas.			Ninguno(0), Bajo(1), Medio(2), Alto(3), Muy Alto(4), Extremadamente Alto(5)
Capacidad para utilizar modelos informáticos diferentes para representar y organizar la información.			
Capacidad para formular (y monitorear) estrategias de solución de problemas informáticos.			
Capacidad para evaluar soluciones informáticas de manera independiente.			
Capacidad para interconectar conocimientos informáticos.			
Capacidad para organizar y comunicar un resultado de forma clara y sencilla.			
Reflexión metacognitiva: valora las tareas informáticas asignadas en función de sus capacidades.			
Regulación metacognitiva: manifiesta capacidad de valoración de la calidad de los resultados informáticos a los que arriba.			
Capacidad de manipular con mayor dominio y placer que sus coetáneos lo abstracto conceptual perteneciente al dominio de la informática.			
Capacidad para extrapolar el conocimiento informático en beneficio de otras áreas del saber.			
Capacidad de reestructurar y reconstruir nuevos resultados, utilizando criterios propios y diferentes alternativas.			
Capacidad para detectar, formular y/o resolver nuevos problemas informáticos.			
Capacidad para descubrir relaciones inusuales en lugar de las convencionales.			

Capacidad de promover ideas y protagonizar acciones de liderazgo en ambientes colaborativos de su profesión.			
Capacidad para brindar respuestas variadas, modificar ideas, superar la rigidez, y reestructurar la información que posee.			
Analítico y crítico con las evidencias que recibe y con los criterios contrarios en un ambiente de colaboración efectiva en el dominio de la informática.			
Afectiva-Emocional			
Elevado afán de la búsqueda de competencias que le permitan destacarse en su dominio profesional.			Ninguno(0), Bajo(1), Medio(2), Alto(3), Muy Alto(4), Extremadamente Alto(5)
Alto nivel de concentración y disfruta con las tareas informáticas que desarrolla.			
Dedicación total al cumplimiento de la tarea informática que le ha sido planteada.			
Se implica con facilidad en tareas relacionadas con el dominio de la informática.			
Renuncia manifiesta ante aquellas actividades que lo distraiga de su desempeño profesional			
Elevado vínculo afectivo con el resultado de sus soluciones informáticas.			
Calidad y naturaleza informática de sus selecciones de ocupación del tiempo libre.			
Manifiesta propósitos definidos en la vida asociados a su desarrollo profesional informático.			
Manifiesta ideales y ambiciones elevadas asociados a su desempeño profesional.			
Psico-Social			
Privilegia su desarrollo profesional a los incentivos materiales.			Ninguno(0), Bajo(1), Medio(2), Alto(3), Muy Alto(4), Extremadamente Alto(5)
Experimenta la necesidad del reconocimiento moral y material.			
Su decisión en la solución de problemas informáticos tiene un alcance social que cumple con los principios éticos de su entorno.			

Retador y desafiante: estimulador, crítico-constructivo y que reconoce los logros en el dominio profesional.			
Estimula la imaginación creativa, la búsqueda de lo nuevo y la invención en el dominio informático general.			
Aprovecha de espacios para exponer soluciones a problemas del dominio de la informática.			
Acomete acciones en los procesos de obtención de resultados informáticos que además de complejos, efectivos y exitosos; se corresponden con las exigencias sociales.			
Es pertinente para el empeño de la informatización de nuestra sociedad.			
De elevado significado social y reconocido de forma relevante en la sociedad.			

Anexo 5: Encuesta realizada al actual Director en función de Informatización.

El Director de Informatización es Yunier Saborit Ramírez pero quien se encuentra en funciones es Cesar González Hernández, a quien se entrevistó a continuación:

¿Existe alguna forma de obtener los datos académicos de los estudiantes desde el Sistema de Gestión Universitaria (SGU) mediante un servicio web?

Respuesta: No se ha concebido ningún servicio que permita la obtención de estos datos.

¿Se puede tener acceso a esta información para que otro sistema u aplicación pueda utilizarla?

Respuesta: No, estos datos son muy delicados y por problemas de seguridad no se puede tener acceso a ellos.

¿A nivel de directivos no se puede solicitar permiso para utilizar esta información de forma confidencial y segura?

Respuesta: No existe ningún permiso para ello, la única forma es que se implemente un módulo para el SGU utilizando una simulación de esta información, que posteriormente la Dirección de Informatización se encargaría de incorporar al SGU.