

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
FACULTAD 1**



**Aplicación web para la gestión de notas en Nova 360**

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

**Autor: Aideé Mora Urdaneta**

**Tutor: Ing. Ivaniet Díaz Romeu**

**Ing. Yosel L Vera González**

**LA HABANA, JUNIO DEL 2018**



*“La revolución no se lleva a la boca para vivir con ella, se lleva en el corazón para morir con ella”*

*Che Guevara*

**Declaración de autoría**

Declaro que yo Aideé Mora Urdaneta, con carnet de identidad 94090629876 soy la autora principal del trabajo de diploma titulado Aplicación web para la gestión de notas en Nova 360 y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firman la presente a los \_\_\_\_días del mes de junio del año 2018

\_\_\_\_\_  
Aideé Mora Urdaneta

Autora

\_\_\_\_\_  
Ing. Ivaniel Díaz Romeu

Tutor

\_\_\_\_\_  
Ing. Yosel L Vera González

Tutor

**Datos de contacto**

Ing. Ivaniel Díaz Romeu

Universidad de las Ciencias Informáticas, La Habana, Cuba

correo: ivaniet@uci.cu

Ing. Yosel Lázaro Vera González

Universidad de las Ciencias Informáticas, La Habana, Cuba

correo: ylvera@uci.cu

## **Agradecimientos**

A mi **abuela** por ser la mejor abuela del mundo, por creer siempre en mí aun cuando yo misma no creía que podía, por darme siempre toda la serenidad y la calma para afrontar los problemas de una manera diferente. Gracias por todos estos años de lucha de esfuerzo, sacrificio, de entrega incondicional hacia mí, no alcanza los agradecimientos, ni las palabras para expresar que hoy no hubiese podido lograr este sueño sin tu apoyo, ejemplo de mujer, de madre, de amiga y de abuela.

A mi **madre** que tan solo con 20 años decidió convertirse en la mejor madre que conozco, por dejar todos sus sueños aun lado y convertir mis sueños y mis logros en los suyos. Por ser mi apoyo, mi sostén, mi amiga, por batallar horas y horas por darme una vida mejor. Gracias por todo el esfuerzo que has hecho por sacarme hacia adelante por guiarme, por creer siempre en tu hija, por enseñarme que las mejores batallas se logran con la perseverancia. Gracias por hoy poder decir que me siento orgullosa de mi madre por ser la mejor guerrera que conozco.

A mi tía **Norka** que a pesar de no encontrarse físicamente con nosotros, sé que este donde este se siente hoy muy orgullosa de mí.

A mi **papá** por su apoyo, por su perseverancia ante la vida, por demostrarme que cuando hay ganas y fuerzas de vivir todo es posible.

A **Abreuni** por su apoyo incondicional durante todos estos años de estudio, por sus consejos, por su dedicación, por demostrarme que la distancia no significa ausencia ni olvido. Gracias por todo tu cariño y por la lealtad que solo los padres les enseñan a sus hijos.

A mi tío **José** por su apoyo incondicional que aunque hoy este lejos de nosotros sé que siente un orgullo inmenso.

A mi **hermana** que espero que algún día seas tú la que estés parada frente a nosotros dedicando tu trabajo de diploma.

A mi novio **Javier**, por regalarme mis mejores años de universidad que aunque hayan sido regulares, buenos o malos, siempre has estado ahí para mí. Gracias por tu apoyo incondicional, por hacer que mis problemas, mis sueños y mis metas sean tuyas también. Hoy solo tengo palabras de agradecimiento hacia ti y aunque ya tengas tu título quiero compartir el mío contigo porque sin tu ayuda nada hubiese sido posible.

A **Ifraín** por quererme como una hija más, por su apoyo, dedicación. Gracias por su esfuerzo, su entrega y sacrificio en estos años de estudio.

A mis padrinos **Cira** y **Gilberto** por todo el cariño brindado hacia mí.

A **Dania** y a **Marlene**, por demostrarme que no necesitamos tener lazos sanguíneos para comportarnos como familia. Gracias **Dania** por correr junto conmigo ante cualquier llamado, por ser la mejor hermana que mi abuela pudiera tener tan cerca.

A mis amigas **Yisi**, **Claudia**, **Dareina** por regalarnos tantos años de amistades en las buenas y las malas.

A mi hermano **Maikel** por ser el hermano que me dio la Uci. Gracias por todos los momentos que pasamos juntos.

A mis amigos de la infancia **Pavel** y **Frank** por tantos años de amistad gracias por siempre estar pendiente de mí.

A mis suegros **Odalys** y **Piñeiro** por su apoyo hacia mí.

A mis tutores **Ivaniet** y **Yosel**, por la ayuda brindada por su dedicación entrega y por sus consejos. A **Ivaniet** por todas esas horas que le robé de su tiempo, por estar ahí siempre que lo necesite más que un tutor me llevo un amigo. Infinitamente agradecida con los dos.

A mis **compañeros** de aula por ser la familia más cercana que tuve durante estos 5 años, la familia **UCI**, me los llevo en el corazón a todos, en especial, al **Lillo** por sus horas de repasos, a mi **Coco** porque me siento feliz de que hoy nos gradúemos juntos, a **David** por su apoyo y ayuda ante cualquier llamado, a **Manuel** por su amistad, a **José** por su amistad y tantas fiestas que pasamos juntos, a **Mauro** por hacernos reír en los momentos más tensos. A **Daniela** y **Elena** por haberme brindado su apoyo cuando más las necesite.

A **José Carlo** por su ayuda, entrega y dedicación durante mi carrera.

A todos los **profesores** que durante estos 5 años me ayudaron a mi formación profesional. Gracias por todo.

## **Dedicatoria**

A las dos personas más importantes de mi vida, a mi razón de ser, mi **Abuela** y mi **Madre** por tanta dedicación, entrega y por tanto esfuerzo. A las dos guerreras de mi vida, a dos mujeres que no se han cansado de luchar por darme un futuro mejor. Este es mi pequeña muestra de amor y gratificación hacia ustedes.

**Resumen**

La computación en la nube es uno de los modelos computacionales de mayor crecimiento, permitiendo el desarrollo de nuevas aplicaciones para el almacenamiento en línea. En la actualidad existen nubes públicas que permiten el almacenamiento de la información de forma gratuita mediante aplicaciones de código abierto que permiten: administrar nubes públicas y privadas, mantener varios directorios sincronizados en tiempo real en varios ordenadores y brindar a los usuarios la posibilidad de gestionar notas, para facilitar el manejo de la información almacenada. Cuba en aras de ganar en soberanía tecnológica y socio-adaptabilidad, está inmersa en un proceso de migración a software libre donde la distribución cubana de GNU/Linux Nova es la distribución seleccionada para ser desplegada en los Organismos de Administración Central del Estado (OACE). Como parte de este proceso de migración surge Nova 360 que es una solución informática basada en *Owncloud* que posibilita la sincronización con múltiples dispositivos de datos, calendarios, tareas y otros. El objetivo del presente trabajo de diploma es desarrollar una Aplicación web para la gestión de notas en Nova 360. Para guiar el proceso de elaboración de la propuesta de solución se utilizó la metodología de desarrollo Variación de AUP para la UCI, en la implementación se utilizaron herramientas y tecnologías libres. La evaluación de la propuesta de solución se realizó a partir de la aplicación de pruebas de software y la técnica de ladov que garantizan el correcto funcionamiento de la aplicación y demostraron la satisfacción del cliente hacia la solución desarrollada.

**Palabras claves:** computación en la nube, notas, Nova 360, *Owncloud*, software libre



## Índice

|   |          |
|---|----------|
| <b>Introducción .....</b>   | <b>1</b> |
| <b>Capítulo 1: Fundamentación teórica sobre la gestión de notas en aplicaciones web .....</b> | <b>5</b> |
| <b>Introducción .....</b>   | <b>5</b> |
| 1.1 Conceptos fundamentales.....  | 5        |
| 1.1.1 Nota .....  | 5        |
| 1.1.2 Cloud Computing.....  | 5        |
| 1.1.3 OwnCloud .....  | 5        |
| 1.1.4 Personalización de software .....   | 5        |
| 1.2 Nova 360.....   | 6        |
| 1.3 La gestión de notas en aplicaciones informáticas.....                                     | 6        |
| 1.4 Aplicaciones informáticas de gestión de notas.....  | 7        |
| 1.4.1 QownNotes .....   | 7        |
| 1.4.2 OwnNote .....   | 8        |
| 1.4.3 Notes.....  | 8        |
| 1.4.4 Quicknote .....   | 8        |
| 1.4.5 Note Station.....   | 9        |
| 1.4.6 Análisis comparativo de las aplicaciones informáticas de gestión de notas .....         | 9        |
| 1.5 Metodología de desarrollo de software .....   | 11       |
| 1.5.1 Metodología Variación de AUP para la UCI.....   | 11       |
| 1.6 Lenguajes y herramientas para el modelado de la solución .....                            | 12       |
| 1.6.1 Lenguaje de modelado: UML 2.5.....  | 12       |
| 1.6.2 Herramientas de modelado: Visual Paradigm 8.0.....                                      | 12       |
| 1.7 Herramientas y tecnologías de implementación .....  | 12       |
| 1.7.1 Lenguajes de programación .....   | 13       |
| 1.7.2 Entorno de Desarrollo Integrado.....  | 14       |
| 1.7.3 Sistema de Gestión de Base de Datos (SGBD).....   | 15       |
| 1.7.4 Servidor de aplicaciones web .....  | 15       |

---

|  |           |
|--|-----------|
| 1.7.5 Servicios web .....  | 16        |
| 1.8 Conclusiones del capítulo .....  | 17        |
| <b>Capítulo 2: Análisis y diseño de la aplicación web para la gestión de notas en Nova 360.....</b>            | <b>18</b> |
| <b>Introducción .....</b>  | <b>18</b> |
| 2.1 Descripción del contexto del negocio de la propuesta de solución .....                                     | 18        |
| 2.1.1 Modelo conceptual .....  | 18        |
| 2.1.2 Diccionario de datos .....   | 19        |
| 2.2 Requisitos .....   | 20        |
| 2.2.1 Obtención de requisitos de software.....   | 20        |
| 2.2.3 Especificación de requisitos de software .....   | 21        |
| 2.2.4 Descripción de requisitos de software mediante Historias de usuario.....                                 | 23        |
| 2.2.5 Validación de requisitos de software .....   | 24        |
| 2.2.6 Administración de requisitos de software.....  | 26        |
| 2.3 Análisis y Diseño.....   | 27        |
| 2.3.1 Diseño de clases .....   | 27        |
| 2.3.2 Patrones de diseño de software .....   | 29        |
| 2.3.3 Modelo de datos .....  | 30        |
| 2.3.4 Diseño arquitectónico .....  | 31        |
| 2.4 Conclusiones del capítulo .....  | 32        |
| <b>Capítulo 3: Implementación y evaluación de la aplicación web para la gestión de notas en Nova 360 .....</b> | <b>33</b> |
| <b>Introducción .....</b>  | <b>33</b> |
| 3.1 Implementación .....   | 33        |
| 3.1.1 Estándares de codificación.....  | 33        |
| 3.1.2 Ejemplo de la interfaz gráfica de usuario de la aplicación web para la gestión de notas....              | 34        |
| 3.1.3 Diagrama de despliegue.....  | 35        |
| 3.2 Pruebas de software para la evaluación de la aplicación de gestión de notas .....                          | 37        |
| 3.2.1 Tipos de pruebas de software .....   | 37        |

---

|  |           |
|--|-----------|
| 3.2.2 Métodos de prueba de software .....                          | 38        |
| 3.2.3 Técnicas de prueba de software.....                          | 38        |
| 3.3 Aplicación de las pruebas de software .....                    | 38        |
| 3.3.1 Pruebas internas .....                                       | 38        |
| 3.3.2 Pruebas de aceptación.....                                   | 44        |
| 3.4 Evaluación de la satisfacción de los potenciales usuarios..... | 44        |
| 3.5 Conclusiones del capítulo .....                                | 46        |
| <b>Conclusiones generales .....</b>                                | <b>47</b> |
| <b>Recomendaciones.....</b>  | <b>48</b> |
| <b>Referencias bibliográficas .....</b>                            | <b>49</b> |
| <b>Anexos .....</b>  | <b>57</b> |

---

## Índice de figuras

|   |    |
|---|----|
| Figura 1. Modelo conceptual .....   | 18 |
| Figura 2. Prototipo elemental de interfaz gráfica de usuario del RF. Añadir nota..... | 24 |
| Figura 3. Diseño de caso de prueba del RF. 1 Añadir Nota .....                        | 25 |
| Figura 4. Descripción de las variables .....  | 26 |
| Figura 5. Matriz de trazabilidad de Requisitos-Diseño de casos de pruebas .....       | 27 |
| Figura 6. Diagrama de clases del diseño de la propuesta de solución.....              | 28 |
| Figura 7. Aplicaciones de los patrones GRASP e Inyección de dependencias .....        | 30 |
| Figura 8. Modelo de datos .....   | 31 |
| Figura 9. Modelo Vista Controlador .....  | 31 |
| Figura 10. Arquitectura de la Solución.....   | 32 |
| Figura 11. Estándar de codificación.....  | 34 |
| Figura 12. Interfaz gráfica de usuario para gestionar notas .....                     | 35 |
| Figura 13. Diagrama de Despliegue .....   | 36 |
| Figura 14. Método createNote del RF1 Adicionar Nota.....                              | 39 |
| Figura 15. Grafo de flujo.....  | 39 |
| Figura 16. Resultados de las pruebas funcionales .....                                | 41 |
| Figura 17. Prueba funcional con el Insomnia Rest.....                                 | 42 |
| Figura 18. Prueba funcional con el Rest Client.....                                   | 42 |
| Figura 19. Diseño de caso de prueba para la Prueba de integración.....                | 43 |

---

## Índice de tablas

|  |    |
|--|----|
| Tabla 1 .Definición de criterios y su ponderación.....                               | 10 |
| Tabla 2. Tabla comparativa de las aplicaciones informáticas de gestión de notas..... | 10 |
| Tabla 3. Diccionario de datos del concepto Notas .....                               | 19 |
| Tabla 4. Listado de requisitos funcionales .....                                     | 22 |
| Tabla 5. Listado de requisitos no funcionales .....                                  | 23 |
| Tabla 6. Descripción de la Historia de Usuario del RF.1 Añadir nota.....             | 23 |
| Tabla 7. Caso de prueba para el camino 5 del método createNote .....                 | 40 |
| Tabla 8. Cuadro Lógico de ladov.....   | 44 |
| Tabla 9. Resultados de la escala de satisfacción .....                               | 46 |

---

## Introducción

Los volúmenes de información digital que se han generado hasta la actualidad aumentan de forma exponencial cada año. Según estimaciones de Sogeti, compañía del Grupo Capgemini (dedicada al estudio de almacenamiento de información), este volumen alcanzará en el 2020 los 35,2 Zettabytes (ZB)<sup>1</sup> (Sogeti, 2017). La gestión de esta información y la necesidad de poder acceder a documentos electrónicos desde cualquier sitio geográfico y garantizar su compartición y manipulación por diferentes personas y programas, en múltiples dispositivos electrónicos ha contribuido al surgimiento del *cloud computing* (computación en la nube) (Leva, 2013).

Actualmente, la Computación en la Nube es uno de los modelos computacionales de mayor crecimiento, permitiendo el desarrollo de nuevas aplicaciones para el almacenamiento en línea. Opera en tres niveles: Software como Servicio (*Software as a Service*, SaaS), Infraestructura como Servicio (*Infrastructure as a Service*, IaaS), y Plataforma como Servicio (*Platform as a Service*, PaaS). Los resultados, publicados en el informe “CloudSleuth, 2013”, reflejan que más de medio millón de *test* de rendimiento llevados a cabo durante el último año entre los principales proveedores de servicios de la computación en la nube en el mundo, demuestran que proveedores de software en todos los países se han lanzado a usar (SaaS).

Las aplicaciones que se suministran en este modelo de servicio son accesibles a través de un navegador web y el usuario no tiene control sobre ellas, aunque en algunos casos se le permite realizar algunas configuraciones. En este nivel, una empresa brinda el mantenimiento, soporte y operación que usará el cliente durante el tiempo que haya contratado el servicio (Axel Springer España S.A., 2016).

El desarrollo de la computación en la nube presente en aplicaciones de software libre es cada vez mayor. Herramientas informáticas como: *Eucalyptus*, *OpenNebula*, *Openstack*; *Cloud Foundry* constituyen alternativas libres al uso de software privativo implementado en diferentes tecnologías, como: *Dropbox*, *Box*, *One Driver*, *Google Driver*. Grupos de personas sin fines lucrativos y mediante el trabajo colaborativo aumentan constantemente el desarrollo de programas libres (Alicante, 2017).

En la actualidad existen nubes públicas que permiten el almacenamiento de la información de forma gratuita mediante aplicaciones de código abierto que permiten: administrar nubes públicas y privadas, mantener varios directorios sincronizados en tiempo real en varios ordenadores y brindar a los usuarios la posibilidad de gestionar notas (texto de cualquier longitud), para facilitar el manejo de la

---

<sup>1</sup> 1 ZB es igual a 1 trillón de GB

información almacenada; ya que el cerebro humano tiene su propia limitación para ser eficiente recordando citas y tareas (Redrejo, 2012).

La gestión de notas permite escribirlas, editarlas, eliminarlas, listarlas, añadirle colores a la nota, así como organizarla en secciones o grupos, le da la posibilidad al usuario de organizarlas en su agenda digital para saber cuándo estarán vigentes o para recibir avisos. Existen aplicaciones informáticas basadas en tecnologías libres para la gestión de notas como: *Simplenotes*, *Gumnotes*, *Springpad* y privativas como *EverNote* y *OneNote* (Genbenta, 2017).

Cuba, en aras de ganar en soberanía tecnológica y socio-adaptabilidad, así como garantizar la informatización de todas las esferas de la sociedad, está inmersa en un proceso de migración a software libre donde la distribución cubana de GNU/Linux Nova es la distribución seleccionada para ser desplegada en los Organismos de Administración Central del Estado (OACE). Esta distribución es desarrollada por el proyecto Nova, perteneciente al Centro de Software Libre (CESOL) de la Universidad de las Ciencias Informáticas (UCI).

Como parte de este proceso de migración surge Nova 360 que es una solución informática basada en *Owncloud* (herramienta que permite gestionar servicios en una nube personalizada de forma segura y con la garantía de la seguridad de los datos almacenados) que posibilita la sincronización con múltiples dispositivos de datos, calendarios, tareas y otros. Su desarrollo está centrado en brindar una solución que integre diferentes tecnologías de Nova como: NovaDroid (personalización cubana de Android disponible para ser utilizada en los dispositivos móviles).

En los despliegues realizados en instituciones del país, los usuarios han mostrado interés en el uso de las aplicaciones para la gestión de notas contenidas en diferentes variantes de Nova. En el caso de Nova, se emplea la aplicación *Bijiben* (desarrollada por GNOME, entorno de escritorio e infraestructura de desarrollo para sistemas operativos GNU/Linux) utilizada para tomar notas con un enfoque minimalista, ofreciendo facilidad de uso, búsqueda, edición de texto (Negrita, Cursiva, Tachado) y el uso de viñetas (Esteve, 2017). En NovaDroid se emplea la aplicación *Omni Notes* utilizada como un recordatorio de tarea, que entre sus funciones permite importar y exportar copias (Rosillo, 2017).

La utilización de las herramientas de gestión de notas, *Bijiben* y *Omni Notes* por parte de los usuarios en los OACE, ha generado inconformidades debido a que pierden tiempo al unificar las notas contenidas en los diferentes dispositivos (computadora de escritorio y móviles) con los que trabajan, y no tienen un lugar común donde centralizar todas las notas de forma tal que se facilite el acceso a las mismas desde la web, la aplicación de escritorio o la aplicación móvil.

Dada la situación problemática anteriormente planteada, se define como **problema de la investigación**: ¿Cómo garantizar la gestión de notas en Nova 360? Se define como **objeto de estudio**: aplicaciones web para la gestión de notas que permitan la sincronización con clientes externos enmarcado en el **campo de acción**: aplicación web para la gestión de notas en Nova 360.

El presente trabajo tiene como **objetivo general**: Desarrollar una aplicación web que permita la gestión de notas en Nova 360. Para garantizar el cumplimiento del objetivo general se determinaron los siguientes **objetivos específicos**:

1. Elaborar el marco teórico de la investigación acerca de las aplicaciones web para la gestión de notas que permitan la sincronización con clientes externos.
2. Diseñar una aplicación web que permita la gestión de notas en Nova 360.
3. Implementar una aplicación web que permita la gestión de notas en Nova 360.
4. Evaluar la aplicación web para la gestión de notas en Nova 360 mediante la aplicación de pruebas de software y la técnica de ladov.

Se definen como **preguntas científicas**:

1. ¿Cuáles son los presupuestos teóricos que fundamentan el proceso de gestión de notas en Nova 360?
2. ¿Qué aspectos se deben tener en cuenta para diseñar una aplicación web para la gestión de notas en Nova 360?
3. ¿Cuáles son las herramientas y tecnologías más adecuadas para implementar una aplicación web que permita la gestión de notas en Nova 360?
4. ¿Qué pruebas de software aplicar para la evaluación de la aplicación web de gestión de notas en Nova 360?

Para el desarrollo de la investigación se utilizaron los siguientes **métodos de la investigación científica**:

**Métodos teóricos:**

- **Análisis y síntesis**: se utilizó para el estudio y análisis de diferentes fuentes bibliográficas con el objetivo de obtener un amplio conocimiento acerca de las características y peculiaridades de herramientas informáticas para la gestión de notas.
- **Modelación**: se utilizó para modelar los diagramas de análisis, diseño e implementación que forman parte de la aplicación web para la gestión de notas en Nova 360.

**Métodos empíricos:**

- **Entrevista**: se utilizó para conocer el contexto del proyecto Nova 360, las herramientas y



tecnologías compatibles con el entorno de desarrollo de dicho proyecto, los sistemas que se integran con Nova 360 y las características que debe tener una aplicación para ser instalada en dicho software (Anexo 1).

- **Observación:** se empleó para verificar cómo se realiza la gestión de notas en aplicaciones informáticas basadas en *Owncloud*, que facilitan la sincronización con clientes externos (Anexo 2).
- **Encuesta:** se aplicó para la identificación de la información a procesar en el negocio informatizado, la obtención de requisitos y las restricciones de diseño existentes en la propuesta de solución (Anexo 11) y la evaluación de índice de satisfacción de los potenciales usuarios hacia la solución desarrollada (Anexo 13).

El presente trabajo de diploma está estructurado en: resumen, introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos.

**Capítulo 1. Fundamentación teórica sobre la gestión de notas en aplicaciones web:** en este capítulo, se definen los conceptos fundamentales que abordan el problema planteado, el estudio sobre las principales características de las aplicaciones web para la gestión de notas que permitan la sincronización con clientes externos, así como la metodología de desarrollo de software y las diferentes herramientas que se van a utilizar en la investigación.

**Capítulo 2. Análisis y diseño de la aplicación web para la gestión de notas en Nova 360:** en el capítulo se evidencia el modelo conceptual elaborado para comprender el contexto del negocio a informatizar. Se definen los requisitos de la aplicación web mediante la especificación de requisitos de software, su descripción a través de historias de usuario así como su validación y administración. Contiene el diagrama de clases del diseño concebido sobre la base de los patrones de diseño que muestra las restricciones de implementación de las funcionalidades de la aplicación. Presenta el modelo de datos que describe la estructura de persistencia de la información en la aplicación web y el diseño arquitectónico definido para la propuesta de solución.

**Capítulo 3. Implementación y evaluación de la aplicación web para la gestión de notas en Nova 360:** este capítulo se enfoca en la implementación de la aplicación web a partir de los estándares de codificación definidos para la propuesta de solución y los resultados del análisis y diseño. Se presenta el diagrama de despliegue, las pruebas de software realizadas y se evalúa la satisfacción de los usuarios hacia la aplicación web desarrollada a través de la técnica de ladov.

## **Capítulo 1: Fundamentación teórica sobre la gestión de notas en aplicaciones web**

### **Introducción**

El presente capítulo contiene los conceptos fundamentales que abordan el problema planteado, el estudio sobre las principales características de las aplicaciones web para la gestión de notas que permitan la sincronización con clientes externos, así como la metodología de desarrollo de software y las diferentes herramientas que se van a utilizar en la investigación.

### **1.1 Conceptos fundamentales**

#### **1.1.1 Nota**

La Real Academia de la Lengua Española define la palabra nota como: “Apunte que se hace para recordar algo.”

Nota es una seña o marca identificatoria que se aplica sobre algo con el objetivo de individualizarlo. Permite recordar tareas, brindando la posibilidad al usuario de organizarlas (Pérez, y otros, 2012).

#### **1.1.2 Cloud Computing**

La computación en la nube (*Cloud Computing*) es un modelo o paradigma para la utilización de los recursos informáticos que está completamente basado en Internet. Dichos recursos normalmente son aplicaciones de software, también pueden ser almacenamiento de archivos, bases de datos, correos electrónicos que residen en servidores remotos, de modo que el usuario puede acceder a ellos desde cualquier lugar del mundo, siempre que cuente con un navegador y una conexión a Internet (Cibernat, 2017).

#### **1.1.3 OwnCloud**

*OwnCloud* es una aplicación de código abierto gratuita, desarrollada en lenguaje PHP y JavaScript, que puede ser utilizada tanto para el almacenamiento en la nube, como para compartir archivos e incluso para sincronización de datos. Esta herramienta permite crear y administrar una nube personalizada sin tener que afrontar los costos de las nubes comerciales. Entre sus principales características se destacan el rendimiento, la seguridad y la calidad (Leva, 2013).

#### **1.1.4 Personalización de software**

La Real Academia de la Lengua Española define la palabra personalizar como: “Dar carácter personal a algo”.

Personalización de software es el desarrollo de sistemas, aplicaciones, programas, sitios web, páginas web o cualquier software, de acuerdo con las especificaciones del cliente, es decir, se construye y forma a las necesidades únicas y gustos del cliente (Gesoft, 2015).

## **1.2 Nova 360**

En la entrevista realizada a especialistas de CESOL (Anexo 1) se verificó que Nova 360 es una personalización de *Owncloud* que integra tecnologías para la sincronización de archivos, contactos, calendarios, tareas y notas entre dispositivos heterogéneos. Enfocada a lograr una máxima compatibilidad entre Nova y NovaDroid. Es compatible con los sistemas operativos: Linux, Windows y Mac.

Características de Nova 360:

- Calendario: permite compartir calendarios y eventos con otros usuarios.
- Visor de documento: brinda la funcionalidad para manejar documentos sin necesidad de descargarlos en una máquina.
- Galerías: comparte galerías de fotos con direcciones de correos electrónicos definidas por el usuario.
- Directorio de temas: posee estilo propio, utilizando colores, íconos, fuentes e imágenes.

Utilidades de Nova 360:

- Sincronización de datos: posibilita la sincronización de fotos, contactos, archivos.
- Compartir datos: brinda acceso a terceros a compartir sus datos.
- Acceder a datos propios: permite almacenar una gran variedad de tipos de archivos como fotos, archivos, contactos y carpetas.

Sistemas informáticos que se integran con Nova 360:

- Nova NAS: permite el almacenamiento de la información en el servidor a través de un protocolo NFS (*Network File System*, Sistemas de Archivos de Red).
- Nova Unificado: es una personalización de Nova para la administración centralizada de servicios telemáticos. Permite la autenticación utilizando el servicio LDAP (*Lightweight Directory Access Protocol*, Protocolo Ligero Simplificado de Acceso a Directorios) y el envío de notificaciones empleando el servicio de correos.

## **1.3 La gestión de notas en aplicaciones informáticas**

La forma en la que los usuarios organizan sus notas y la realización de apuntes en libretas, cuadernos y libros ha evolucionado de una manera extraordinaria con el uso de las tecnologías de la información y las comunicaciones (apps.ownCloud, 2017).

La autora de la investigación mediante el método empírico de la observación (Anexo 2) pudo conocer cómo se realiza la gestión de notas en las siguientes aplicaciones informáticas: QownNotes,

*OwnNote*, *Notes*, *Quicknote* y *Note Station*. A continuación se reflejan los resultados obtenidos de la observación realizada.

Las aplicaciones de gestión de notas analizadas poseen diferentes opciones para que el usuario cree, edite, modifique y elimine sus notas. Poseen una interfaz visual en la que el usuario no realiza acciones innecesarias para la gestión de sus notas. Brindan la posibilidad de guardar todo tipo de información. No se basan únicamente para dispositivos móviles, de escritorio y web, sino que contienen un sistema de almacenamiento en la nube, que permiten la conexión con los diferentes dispositivos y en múltiples plataformas. Garantizan que el usuario pueda tener organizado sus notas ya sean escritas o de imagen, a las que pueden acceder mediante su cuenta en la nube. En el epígrafe 1.4 se caracterizan las aplicaciones informáticas para la gestión de notas observadas anteriormente.

## **1.4 Aplicaciones informáticas de gestión de notas**

### **1.4.1 QownNotes**

*QownNotes* (Anexo 3) es un editor de texto libre, multiplataforma que incluye soporte para el lenguaje de marcado Markdown y la integración de *Ownccloud*. Las notas se almacenan como archivos de texto sin formato. *QownNotes* está disponible en varios idiomas tales como: Inglés, Alemán, Francés, Polaco, Chino, Japonés, Ruso, Portugués, Húngaro, Holandés y Español. Cuenta con un gestor de lista de tareas para GNU/Linux, Mac y Windows. Permite recibir notificaciones sobre las modificaciones externas de la nota actual y conectarse al servidor de Cloud<sup>2</sup> para obtener más funciones como el control de versiones y la papelera (Amoedo, 2017).

Características de *QownNotes* (Amoedo, 2017):

- Permite la sincronización de notas sobre dispositivos (escritorio y móviles) con su propio cliente de sincronización *CloudCloud* o *Nextcloud*.
- Todas las notas eliminadas se pueden recuperar de su *Ownccloud*.
- Proporciona soporte para el cifrado AES-256 de notas (las notas solo se podrán cifrar desde *QownNotes*).
- Todos los paneles se pueden ubicar donde el usuario quiera.
- La interfaz de usuario de la aplicación es personalizable (permite aumentar o disminuir el tamaño de fuente).
- Permite exportar notas a PDF (*Portable Document Format*, Formato de Documento Portable).
- Etiquetado jerárquico de notas y subcarpetas de estas.

---

<sup>2</sup> Un servidor cloud es una potente infraestructura virtual o física que almacena y procesa información y aplicaciones.

#### **1.4.2 OwnNote**

*OwnNote* (Anexo 4) es un editor completo de nota de estilo EverNote/OneNote que se integra con la aplicación web *OwnNote* instalada como parte de la solución de nube empresarial, *Owncloud* (Inc, 2017).

Características de *OwnNote* (Inc, 2017):

- Libre de anuncios.
- Agrupa notas según la categoría.
- Guarda notas como archivos HTML si se configura el servidor.
- Múltiples formatos de fuente, negrita, cursiva, viñeta, tachado, listas enumeradas, así como alineación y sangría.
- Inserción automática de enlace.
- Insertar imágenes redimensionables integradas con BASE64<sup>3</sup>.

#### **1.4.3 Notes**

*Notes* (Anexo 5) es una aplicación de nota que permite guardar notas, listas de compras y enlaces. Ofrece la posibilidad al usuario de ordenar las notas por categorías. Incluye soporte para la edición del lenguaje de marcado Markdown (apps.ownCloud, 2017).

Características de *Notes* (apps.ownCloud, 2017):

- Soporte para GNU/Linux.
- Soporte para el lenguaje de marcado Markdown.
- Soporte para sistemas operativos como Android, NemoMobile<sup>4</sup>.

#### **1.4.4 Quicknote**

*Quicknote* (Anexo 6) es una aplicación de nota que permite notas rápidas, le brinda la posibilidad al usuario de crear y administrar notas de manera fácil (Apps, 2017).

Características de *Quicknote* (Apps, 2017):

- Notas simples, con formato simple.
- Mantiene las notas organizadas.
- Permite tener decenas de notas organizadas de una manera agradable.

---

<sup>3</sup> La codificación Base 64 se basa en el uso de los caracteres US-ASCII (no acentuados) para codificar cualquier tipo de información mediante un código de 8 bits.

<sup>4</sup> Distribución de Linux para dispositivos móviles.

#### **1.4.5 Note Station**

*Note Station* (Anexo 7) es una aplicación de nota totalmente gratuita, que no presenta limitaciones en cuanto a espacio y tamaño, brindando la posibilidad de adjuntar la mayor cantidad posible de archivos adjuntos (ceatic.ujaen.es, 2017).

Características de *Note Station* “según Ramírez, 2017 ”:

- Añade nuevas características a las notas.
- Permite gestionar tareas especificando la prioridad (fecha de vencimiento, recordatorio).
- Permite adjuntar el tamaño de fuente por defecto en la nota.
- Exporta notas como archivos HTML individuales para que otros usuarios que no tengan la aplicación *Note Station* puedan leerlos.

#### **1.4.6 Análisis comparativo de las aplicaciones informáticas de gestión de notas**

En el siguiente epígrafe se realiza un análisis comparativo de las aplicaciones informáticas de gestión de notas estudiadas anteriormente, mediante el método QSOS (*Qualification and Selection of Open Source Software*, Calificación y Selección de Software de Código Abierto). La finalidad de este análisis tiene como objetivo conocer si las aplicaciones analizadas cumplen con las necesidades existentes para la gestión de notas en Nova 360, expuestas en la problemática de la investigación. A continuación se muestra la aplicación del método.

##### **Aplicación del método QSOS**

Este método propone cuatro fases para la evaluación y selección de software de código abierto: Definición, Evaluación, Calificación y Selección, permitiendo establecer un nivel de detalle en el análisis de aplicaciones de software libre en función de las necesidades de cada proceso de desarrollo de software (Ramos, y otros, 2012).

Fase de Definición: Se establece el marco de referencia para la búsqueda de la información relacionada con las necesidades existentes en el proyecto de desarrollo de software a desarrollar. El estudio de aplicaciones informáticas para la gestión de nota en Nova 360, realizado en la investigación está enmarcado en herramientas basadas en owncloud.

Fase de Evaluación: Consiste en realizar una caracterización de los software analizados. En el epígrafe 1.4, se describen las aplicaciones informáticas estudiadas.

Fase de Calificación: Consiste en la ponderación de los criterios definidos para realizar la comparación de las aplicaciones informáticas analizadas. En la Tabla 1 se describen los criterios establecidos, en correspondencia con las necesidades del cliente y las características del entorno de desarrollo del proyecto Nova 360.

Tabla 1 .Definición de criterios y su ponderación

(Fuente: elaboración propia)

| Criterios de análisis                               | Puntuación  |  |
|---|---|--|
|   | No cubierto<br>0                                      | Totalmente cubierto<br>1                           |
| Gestionar notas por categoría                       | No gestiona notas por categoría                       | Gestiona nota por categoría                        |
| Clasificar notas por categoría                      | No clasifican notas por categoría                     | Clasifican notas por categoría                     |
| Basada en software Libre                            | No está basada en software libre                      | Están basadas en software libre                    |
| Compatibilidad owncloud 10.0.2                      | No compatible con owncloud 10.0.2                     | Compatibilidad con owncloud 10.0.2                 |
| Permitir la sincronización de notas desde Nova      | No permite la sincronización de notas desde Nova      | Permite la sincronización de notas desde Nova      |
| Permitir la sincronización de notas desde NovaDroid | No permite la sincronización de notas desde NovaDroid | Permite la sincronización de notas desde NovaDroid |
| Basada en tecnología web                            | No basada en tecnología web                           | Basada en tecnología web                           |

Fase de Selección: Se realiza la comparación de los diferentes software analizados mediante los criterios definidos en la fase de calificación. En la Tabla 2 se evidencia la comparación de aplicaciones informáticas de gestión de notas estudiadas.

Tabla 2. Tabla comparativa de las aplicaciones informáticas de gestión de notas.

(Fuente: elaboración propia)

| Criterios de análisis                               | Sistemas informáticos de gestión de notas |         |       |           |              |
|---|---|---------|-------|-----------|--------------|
|   | QownNote                                  | OwnNote | Notes | Quicknote | Note Station |
| Gestionar notas por categoría                       | 0   | 1       | 0     | 0         | 0            |
| Agrupar nota por categoría                          | 0   | 1       | 1     | 0         | 0            |
| Basada en software Libre                            | 1   | 1       | 1     | 1         | 1            |
| Compatibilidad owncloud 10.0.2                      | 1   | 0       | 0     | 0         | 0            |
| Permitir la sincronización de notas desde Nova      | 0   | 0       | 0     | 0         | 0            |
| Permitir la sincronización de notas desde NovaDroid | 0   | 0       | 0     | 0         | 0            |
| Basada en tecnología web                            | 0   | 1       | 1     | 1         | 1            |

La comparación anterior (Tabla 2) evidencia que las aplicaciones estudiadas no cubren totalmente (valor 1) las solicitudes del cliente reflejadas en los criterios analizados. Se demuestra la importancia de desarrollar una nueva solución ya que las aplicaciones investigadas no satisfacen las necesidades existentes para la gestión de notas en Nova 360. El estudio realizado permitió definir requisitos funcionales y características que deben tener las notas, así como las limitaciones que poseen estas aplicaciones, tomadas en cuenta en el diseño de la propuesta de solución para evitar errores en la gestión de notas y fortalecer las funcionalidades de la solución.

### **1.5 Metodología de desarrollo de software**

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas que permiten la documentación para el desarrollo de productos. Permiten indicar paso a paso todas las actividades a realizar para lograr el producto informático deseado, delimitando qué personas deben participar en el desarrollo de las actividades y qué papel juegan dentro del mismo. “Se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y notaciones que se usarán” (Penadés, 2016). A continuación se describe la metodología de desarrollo utilizada para la elaboración de la propuesta de solución.

#### **1.5.1 Metodología Variación de AUP para la UCI**

La metodología Variación de AUP para la UCI es una variación de AUP (*Agile Unified Process*, Proceso Unificado Ágil) elaborada en la UCI para estandarizar el proceso de desarrollo de software de la universidad. Logra un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajo. Se adapta a las características de cada proyecto y al ciclo de vida definido para la actividad productiva de la UCI. Esta metodología es usada en el centro CESOL para el cual se desarrolla este trabajo de diploma (Rodríguez, 2015). La misma cuenta con tres fases:

- Inicio: durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
- Ejecución: en esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
- Cierre: en esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.



El presente trabajo de diploma se desarrolla en la fase de Ejecución y transitó por las siguientes disciplinas propuestas en la metodología: Requisitos, Análisis y diseño, Implementación, Pruebas internas y Pruebas de aceptación. Los productos de trabajo generados durante la elaboración de la solución están basados en el Escenario no. 4 ya que se aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio bien definido. El cliente estará siempre acompañando del equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos (Rodríguez,2015).

## **1.6 Lenguajes y herramientas para el modelado de la solución**

En el modelado de la propuesta de solución se utilizaron los siguientes lenguajes y herramientas.

### **1.6.1 Lenguaje de modelado: UML 2.5**

**UML 2.5** (*Unified Modeling Language*, Lenguaje de Modelado Unificado) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema. Ofrece un estándar para describir los modelos, incluyendo aspectos conceptuales como procesos de negocio, funciones del sistema, expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. UML cuenta con un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan (Cabrera, 2012).

### **1.6.2 Herramientas de modelado: Visual Paradigm 8.0**

Visual Paradigm 8.0 es una herramienta CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora) que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software (visual\_paradigm.com, 2017).

Características de Visual Paradigm (visual\_paradigm.com, 2017):

- Licencia gratuita y comercial.
- Disponibilidad en múltiples plataformas (Windows y Linux).
- Diseño centrado y enfocado al negocio que genera un software.
- Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Generador de informe y editor de figuras.

## **1.7 Herramientas y tecnologías de implementación**

En el desarrollo de la propuesta de solución se utilizaron las siguientes herramientas y tecnologías.

### **1.7.1 Lenguajes de programación**

Un lenguaje de programación permite que los seres humanos puedan dar instrucciones a una computadora. Se utilizan principalmente para controlar cómo se comporta una máquina o crear programas informáticos. Es un sistema diseñado y estructurado especialmente para describir el conjunto de acciones consecutivas que un equipo debe ejecutar para garantizar una efectiva comunicación con los equipos de cómputo (Gómez, 2017). Para el desarrollo de la herramienta se seleccionaron los siguientes lenguajes:

#### **Lenguaje de programación en el cliente**

**HTML5** (*Hyper Text Markup Language*, Lenguaje de Marcado de Hipertexto) es un lenguaje de programación utilizado para estructurar y presentar el contenido en la web. Define los nuevos estándares de desarrollo web, rediseñando el código para resolver determinados problemas. No se limita solo a crear nuevas etiquetas o atributos, sino que incorpora muchas características nuevas y proporciona una plataforma de desarrollo para complejas aplicaciones web (Garro, 2015).

**JavaScript** es un lenguaje de programación que usa script (rutinas o guiones) usado para crear interactividad dinámica en los sitios web. Es un lenguaje multi-paradigma basado en prototipos dinámicos. Las capacidades dinámicas de JavaScript incluyen construcción de objetos en tiempo de ejecución. JavaScript es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros. Centrado en el código fuente de la página web (Mdn, 2017).

#### **Lenguaje de programación del lado del servidor**

**PHP 7.0** (*Hypertext Pre-processo*, Pre-procesador de hiper-texto) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser insertado en HTML. Se ejecuta sobre una arquitectura cliente\_servidor. Permite conectarse con base de datos tales como MySQL, Postgres, Oracle, y SQLite. Puede ser usado en diversas plataformas y sistemas operativos entre los que se destacan Windows y Linux (Staurdo Lucho, 2012).

El código fuente escrito en PHP es invisible al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable. Tiene capacidad de expandir su potencial utilizando módulos llamados extensiones. Permite aplicar técnicas de programación orientada a objetos. Manejo de excepciones. No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo de datos que estén manejando en tiempo de ejecución (Christopher, 2016).

## Tecnología en el cliente

**Ajax** son las siglas de *Asynchronous JavaScript And XML*, (*JavaScript* asíncrono y *XML*). Ajax es una nueva técnica que combina los lenguajes de programación. Permite trabajar de forma asíncrona (le da la posibilidad al usuario que la página pueda mostrar otras cosas mientras se espera la respuesta). Ajax se usa particularmente en páginas que manejen gran cantidad de datos. Es usada por gran cantidad de usuarios. Ejemplos de páginas que usan Ajax son Gmail (el correo de Google) o Google Maps (Anyelguti, 2017).

### 1.7.2 Entorno de Desarrollo Integrado

Un entorno de desarrollo integrado (*Integrated Development Environment*, IDE) es un programa que está básicamente compuesto por un conjunto de herramientas, políticas y seguridades que permiten la facilidad y disponibilidad de acceso a la información. Los IDEs proporcionan un único programa en el cual se pueda llevar a cabo todo el desarrollo de un sistema. Aumentan la productividad de los programadores ya que proporcionan componentes necesarios para la creación de interfaces de usuarios y su edición, compilación y depuración. La mayoría de los IDEs tienen autocompletado inteligente de código (Briones, 2016). Para el desarrollo de la propuesta de solución se seleccionó Netbeans 8.2.

**Netbeans 8.2** es un IDE de desarrollo gratuito y de código abierto. Permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio, como aplicaciones web, o para dispositivos móviles. Puede ser instalado en varios sistemas operativos como Windows y Linux (Genbeta, 2017).

Las principales características de Netbeans “según Genbeta, 2017” son:

- Asistente para la creación y configuración de distintos proyectos, incluida la elección de algunos *frameworks* (marcos de trabajo).
- Simplifica la gestión de grandes proyectos con el uso de diferentes vistas, asistentes de ayuda y estructurando la visualización de manera ordenada.
- Herramientas para depurado de errores: el *debugger* (aplicación que permite la ejecución controlada de un programa) que incluye el IDE es útil para encontrar dónde falla el código.
- Optimización de código: Ayuda a optimizar aplicaciones e intentar hacer que se ejecuten más rápido y con el mínimo uso de memoria.
- Acceso a base de datos: permite la conexión con distintos gestores de base datos como por ejemplo Oracle, MySQL, entre otros.

### **1.7.3 Sistema de Gestión de Base de Datos (SGBD)**

**MySQL** (*My Structured Query Language*, Lenguaje de Consulta Estructurado) es un sistema de gestión de base de datos relacionales desarrollado bajo la licencia de software libre GPL (*General Public License*, Licencia General Pública) comercial por la Corporación Oracle. Según (Juan Iruela, 2016) es uno de los SGBD más populares del mundo junto con Microsoft SQL Server y Oracle, que se usan especialmente para entornos web.

MySQL es capaz de almacenar una enorme cantidad de datos e incluye todos los elementos necesarios para la gestión de estos. Desarrolla sus propias aplicaciones de base de datos en la mayor parte de lenguajes de programación utilizados en la actualidad. Es lo suficientemente flexible para trabajar en entornos con gran demanda, tales como aplicaciones web. La arquitectura interna de MySQL está dividida en tres capas: Capa Aplicación (residen las funcionalidades que conectan a MySQL con otros sistemas y lenguajes), Capa Lógica (procesan consultas SQL) y Capa Física (reside la lógica para almacenar y acceder a los datos de la tabla) (Cevallos, 2014).

El gestor de base datos seleccionado fue MySQL Workbench, es un software creado por la empresa Sun Microsystems. Esta herramienta permite modelar diagramas de Entidad-Relación para bases de datos MySQL. Permite elaborar una representación visual de las tablas, vistas, procedimientos almacenados y claves foráneas de la base de datos. Es capaz de sincronizar el modelo en desarrollo con la base de datos real (Amoedo, 2017).

Principales características MySQL Workbench (Amoedo, 2017):

- Libre, distribuida bajo la licencia GPL (*General Public License*, Licencia General Pública)
- Multiplataforma disponible para Windows, GNU/Linux y Mac.
- Permite crear diagramas E-R (Entidad\_Relación).
- Importar archivos SQL.
- Permite generar los scripts SQL a partir del modelo creado.

### **1.7.4 Servidor de aplicaciones web**

Los servidores de aplicaciones web (también conocidos como servidores HTTP) son un tipo de servidor utilizado para la distribución de contenido web en redes internas o en Internet. Es un programa que consiste en ejecutar las actividades de un servidor, realizando el cumplimiento de las peticiones que sean procesadas por el usuario. Permite el cifrado de la comunicación entre el servidor web y el cliente, autenticación HTTP para áreas específicas de una aplicación web y almacenamiento en caché de documentos dinámicos para la respuesta eficiente de solicitudes (Dayerling, 2016).

**Apache 2.5** es un servidor web HTTP de código abierto para la creación de páginas y servicios web. Multiplataforma, gratuito que se destaca por su seguridad y rendimiento. Permite la integración de

nuevos protocolos. El servidor Apache es considerado confiable y de rápida velocidad (Dayerling, 2016).

Características del servidor web Apache (MaxCDN, 2017):

- Soporte de seguridad SSL (Secure Sockets Layer, Capa de Sockets Seguros) y TLS (Transport Layer Security, Seguridad en la Capa de Transporte).
- Puede realizar autenticación de datos utilizando SGDB.
- Puede dar soporte a diferentes lenguajes, como PHP, Python y otros.

### **1.7.5 Servicios web**

Según el consorcio *World Wide Web* (consorcio internacional que genera recomendaciones y estándares para la web) un servicio web es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Estas aplicaciones pueden estar desarrolladas en lenguajes de programación diferentes y ser ejecutadas sobre plataformas diferentes. La interoperabilidad se consigue mediante la adopción de estándares abiertos (Fermín Díaz, 2014).

Los servicios web son calificados como una nueva etapa del desarrollo de los sistemas distribuidos, que permite aprovechar todas las ventajas de trabajar en ambiente Web, unidas a las de contar con una amplia gama de tecnologías que pueden ser utilizadas para el desarrollo de los componentes finales. Por ejemplo el estilo REST (*Representational State Transfer*, Transferencia de Estado Representacional) es una forma ligera de crear servicios web. REST se refiere estrictamente a una colección de principios para el diseño de arquitecturas en red. Se utiliza para describir cualquier interfaz que transmite datos específicos de un dominio sobre HTTP (*Hypertext Transfer Protocol*, Protocolo de Transferencia de Hipertextos) y XML (*eXtensible Markup Language*, Lenguaje de Marcas Extensible) para la transmisión de datos sin la necesidad de contar con una capa adicional como hace SOAP (*Simple Object Access Protocol*, Protocolo Simple de Acceso a Objetos) (Fermín Díaz, 2014).

**Características de REST son (BBVA, 2016):**

- Protocolo cliente/servidor sin estado: cada petición HTTP contiene toda la información necesaria para ejecutarla, lo que permite que ni cliente ni servidor necesiten recordar ningún estado previo para satisfacerla.
- Operaciones más importantes: POST (crear), GET (leer y consultar), PUT (editar) y DELETE (eliminar).
- Los objetos en REST siempre se manipulan a partir de la URI.
- Sistema de capas.

**Ventajas que ofrece REST para el desarrollo (BBVA, 2016):**

- Separación entre el cliente y el servidor: el protocolo REST separa totalmente la interfaz de usuario del servidor y el almacenamiento de datos.
- Visibilidad, fiabilidad y escalabilidad: Se puede migrar a otros servidores o realizar todo tipo de cambios en la base de datos.
- La API REST siempre es independiente del tipo de plataformas o lenguajes: con una API REST se pueden tener servidores PHP, Java, Python o Node.js. Lo único que es indispensable es que las respuestas a las peticiones se hagan siempre en el lenguaje de intercambio de información usado, normalmente XML o JSON.

**1.8 Conclusiones del capítulo**

El análisis de los referentes teóricos acerca del proceso de gestión de notas para Nova 360, así como el estudio de los principales conceptos asociados al problema planteado, permitió sentar las bases para el desarrollo de la investigación y conocer las características del objeto de estudio. La caracterización y comparación de las diferentes aplicaciones de gestión de notas demostró la necesidad de desarrollar una aplicación web ya que las existentes no resuelven las carencias que presenta el proyecto Nova 360. Se definió para la elaboración de la propuesta de solución el empleo de la metodología de software Variación de AUP para la UCI y herramientas libres.

## Capítulo 2: Análisis y diseño de la aplicación web para la gestión de notas en Nova 360

### Introducción

En el capítulo se evidencia el modelo conceptual elaborado para comprender el contexto del negocio a informatizar. Se definen los requisitos de la aplicación web mediante la especificación de requisitos de software, su descripción a través de historias de usuario así como su validación y administración. Contiene el diagrama de clases del diseño concebido sobre la base de los patrones de diseño que muestra las restricciones de implementación de las funcionalidades de la aplicación. Presenta el modelo de datos que describe la estructura de persistencia de la información en la aplicación web y el diseño arquitectónico definido para la propuesta de solución.

### 2.1 Descripción del contexto del negocio de la propuesta de solución

Para describir el contexto del negocio de la propuesta de solución se realizó un modelo conceptual, el cual le permitió a la autora de la investigación tener un mejor dominio del negocio a informatizar.

#### 2.1.1 Modelo conceptual

Un modelo conceptual o modelo de dominio contiene la descripción de cómo se relacionan los conceptos u objetos del mundo real en el contexto de un problema determinado. Es una definición de conceptos, atributos y relaciones entre conceptos que permite representar el conocimiento para diferentes dominios del negocio (Vasquez, 2015). En la Figura 1 se muestra el modelo conceptual de la propuesta de solución.

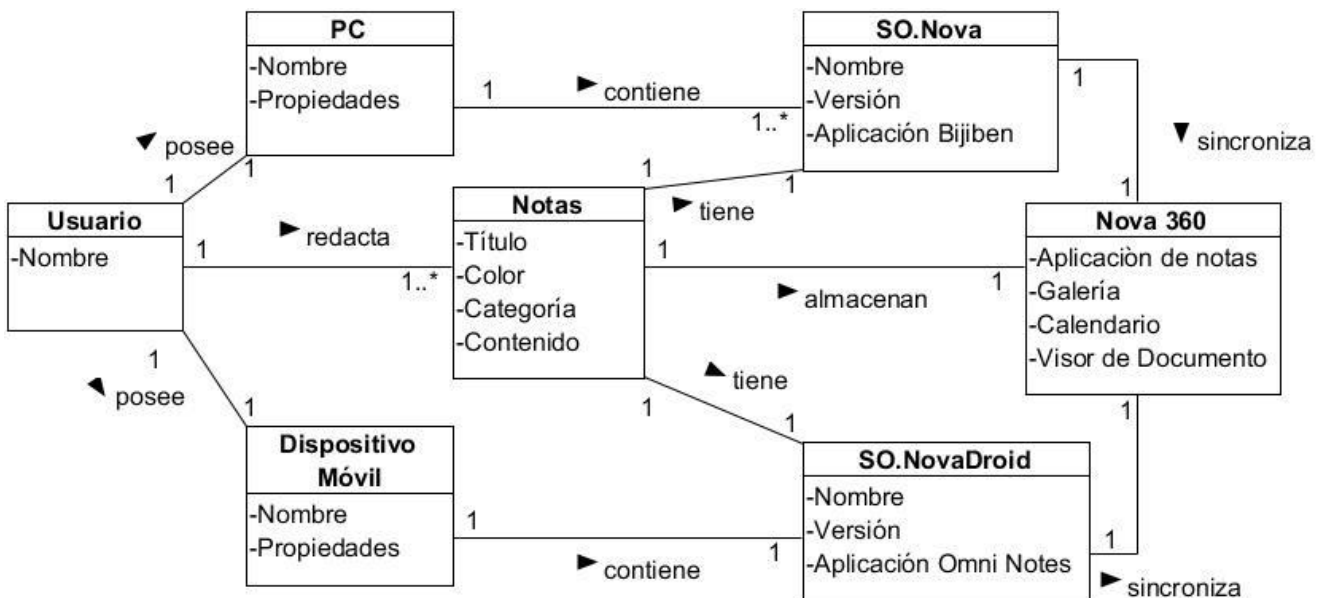


Figura 1. Modelo conceptual  
(Fuente: elaboración propia)

### Conceptos del contexto del negocio:

- **Dispositivo Móvil:** Es un teléfono o un tablet.
- **Notas:** Las notas son apuntes o marcas identificatorias que el usuario realiza para recordar algo.
- **Nova 360:** Es un software que permite la sincronización de archivos, calendarios, tareas y notas.
- **SO. Nova:** Es una distribución cubana de GNU/Linux que contiene la aplicación de notas Bijiben.
- **SO. NovaDroid:** Es una personalización del sistema operativo Android que contiene la aplicación de nota Omni Notes.
- **PC:** Es un ordenador que contiene el sistema operativo Nova capaz de recibir, devolver y procesar las notas del usuario.

### 2.1.2 Diccionario de datos

Un diccionario de datos es un conjunto de información acerca de los conceptos presentes en el dominio de un negocio determinado. Contiene las características de los datos que se van a usar en un sistema informático, así como, detalles sobre las entradas, salidas y almacenamiento de la información. Contribuye a evitar ambigüedades o malas interpretaciones (Arias, y otros, 2012). En la Tabla 3 se presenta el diccionario de datos del concepto Notas.

Tabla 3. Diccionario de datos del concepto Notas

(Fuente: elaboración propia)

| <b>Descripción</b> | Las notas son apuntes o marcas identificatorias que el usuario realiza para recordar algo. |              |                  |         |                        |                     |
|--------------------|--|--------------|------------------|---------|------------------------|---------------------|
| <b>Atributos</b>   |  |              |                  |         |                        |                     |
| Nombre             | Descripción  | Tipo         | ¿Puede ser nulo? | ¿Único? | Restricciones de clase |                     |
|                    |  |              |                  |         | Válidas                | No válidas          |
| Título             | Nombre de la nota  | Alfanumérico | No               | No      | No aplica              | Caracteres extraños |
| Color              | Color de la nota   | Alfanumérico | Sí               | No      | No aplica              | No aplica           |
| Categoría          | Clasificación de la nota por categoría   | Alfanumérico | Sí               | Sí      | No aplica              | Caracteres extraños |
| Contenido          | Es la descripción del contenido de la nota   | Alfanumérico | No               | No      | No aplica              | No aplica           |



## **2.2 Requisitos**

El esfuerzo principal en la disciplina Requisitos es desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto (Rodríguez, 2015). En los siguientes subepígrafos se describen las tareas desarrolladas en la disciplina.

### **2.2.1 Obtención de requisitos de software**

La obtención de los requisitos de software contribuye a identificar cuáles son los objetivos del sistema informático que se desea desarrollar. Define de qué forma el software puede satisfacer las necesidades del negocio permitiendo detallar la utilidad y alcance que tendrá el mismo (Pressman, 2010). A continuación se describen las fuentes de obtención de requisitos y las técnicas de identificación de requisitos empleados para la elaboración de la propuesta de solución.

#### **Las fuentes de obtención de requisitos consultadas son:**

- Modelo conceptual (Figura 1)
- Análisis de las aplicaciones informáticas de gestión de notas existentes (epígrafes 1.4 y 1.5)
- Especialistas del centro CESOL

#### **Las técnicas de identificación de requisitos aplicadas son:**

##### **Entrevista**

La entrevista se define como un intento sistemático de recoger información de otra persona a través de una comunicación interpersonal que se lleva a cabo por medio de una conversación estructurada. Es de gran utilidad para obtener información cualitativa como opiniones o descripciones subjetivas de actividades. Es una técnica muy utilizada y requiere mayor preparación y experiencia por parte del entrevistador (Guerra, 2017).

La entrevista (Anexo 1) realizada a los 4 especialistas del equipo de desarrollo de Nova 360, le permitió a la autora de la investigación conocer el contexto de este proyecto, así como las herramientas y tecnologías compatibles con el entorno de desarrollo, los sistemas que se integran con Nova 360 y las características que debe tener una aplicación para ser instalada en dicho software.

##### **Observación**

La observación permite obtener información sobre la forma en que se efectúan las actividades. Permite observar la manera en que se llevan a cabo los procesos y verificar que realmente se sigan todos los pasos especificados (Guerra, 2017).

Después de la aplicación de la entrevista y con un dominio previo de las particularidades del contexto del negocio a informatizar, se observó cómo se realiza la gestión de notas en 5 aplicaciones

informáticas basadas en *Owncloud*, que facilitan la sincronización con clientes externos. Dicho estudio le permitió a la autora de la investigación conocer las características que poseen las aplicaciones analizadas, elementos tomados en cuenta para definir la propuesta de solución.

### **Encuesta**

La encuesta (Anexo 11) es una técnica de investigación y recopilación de datos utilizada para obtener información de personas sobre diversos temas. Tiene como objetivo obtener información que se pueda analizar, permite extraer modelos y hacer comparaciones. Los datos suelen obtenerse mediante el uso de procedimientos estandarizados, esto con la finalidad de que cada persona encuestada responda las preguntas en una igualdad de condiciones para evitar opiniones que pudieran influir en el resultado de la investigación o estudio (ARR, 2018).

La encuesta fue realizada a los 4 miembros del equipo de desarrollo del proyecto Nova 360 perteneciente a CESOL, lo que permitió la identificación de la información a procesar en el negocio informatizado, la obtención de requisitos y las restricciones de diseño existentes en la propuesta de solución.

### **Desarrollo de prototipos**

Según Guerra, 2017 los prototipos consisten en versiones reducidas, demos o conjuntos de pantallas (que no son totalmente operativos) de la aplicación a desarrollar. Permiten a los usuarios visualizar cómo éste ayuda a su trabajo. Fomentan el desarrollo de ideas que conducen a la identificación y especificación de requisitos. Contribuye a un mejor entendimiento de los requisitos por parte del equipo de desarrollo de la solución.

La técnica de desarrollo de prototipos le permitió a la autora de la investigación conocer qué características deben tener las interfaces de la aplicación web a desarrollar; así como reflejar la información procesada en el sistema informático. En la Figura 2 se presenta un ejemplo de los prototipos elaborados.

#### **2.2.3 Especificación de requisitos de software**

El objetivo principal de la Especificación de requisitos del software (ERS) es servir como medio de comunicación entre clientes, usuarios y equipo de desarrollado. Contiene tanto las necesidades de clientes y usuarios (necesidades del negocio, también conocidas como requisitos de usuario, requisitos de cliente, necesidades de usuario), como los requisitos que debe cumplir el sistema de software a desarrollar para satisfacer dichas necesidades (requisitos del producto, también conocidos como requisitos de software) (MADEJA, 2018). A continuación se relacionan y describen los requisitos funcionales y no funcionales definidos para la implementación de la propuesta de solución.

### Requisitos funcionales

Los requisitos funcionales (RF) describen las funcionalidades particulares de un software para que sea útil en la realización de los procesos de una organización (Casta, 2015). En la Tabla 4 se listan los requisitos funcionales de la propuesta de solución, su descripción y su complejidad, obtenida mediante el producto de trabajo de Evaluación de Requisitos del expediente del proyecto 5.0 disponible para la actividad productiva de la universidad.

Tabla 4. Listado de requisitos funcionales  
(Fuente: elaboración propia)

| Número | Requisitos                          | Descripción   | Complejidad |
|--------|-------------------------------------|---|-------------|
| RF 1   | Añadir nota                         | La aplicación debe permitir que el usuario pueda añadir una nota.                                       | Alta        |
| RF 2   | Editar nota                         | La aplicación debe permitir que el usuario pueda editar una nota.                                       | Media       |
| RF 3   | Eliminar una nota                   | La aplicación debe permitir que el usuario pueda eliminar una nota.                                     | Baja        |
| RF 4   | Añadir categoría a la nota          | La aplicación debe permitir que el usuario pueda añadirle una categoría a la nota.                      | Media       |
| RF 5   | Añadir color a la categoría         | La aplicación debe permitir que se le pueda añadir un color a la categoría.                             | Media       |
| RF 6   | Buscar notas por categoría          | La aplicación debe permitir que el usuario pueda buscar una nota según la categoría.                    | Media       |
| RF 7   | Buscar notas por contenido de texto | La aplicación debe permitir que el usuario pueda buscar una nota según su contenido.                    | Media       |
| RF 8   | Sincronizar notas desde Nova        | Permitir que las notas puedan sincronizarse con la aplicación web para facilitar el acceso de la misma. | Alta        |
| RF 9   | Sincronizar notas desde NovaDroid   | Permitir que las notas se sincronicen con la aplicación web para facilitar el acceso de la misma.       | Alta        |

### Requisitos no Funcionales

Los requisitos no funcionales (RnF) representan características generales y restricciones que debe cumplir el software (Casta, 2015). Con el objetivo de estandarizar la redacción de los requisitos no funcionales que responden a atributos de calidad de software se utilizó la ISO-9126 e ISO-25010 (*International Organization for Standardization*, Organización Internacional de Normalización). En la Tabla 5 se listan los RnF de la propuesta de solución.

Tabla 5. Listado de requisitos no funcionales

(Fuente: elaboración propia)

| Número | Requisito                      | Descripción   |
|--------|--------------------------------|---|
| RnF 1  | Requisito de interoperabilidad | La aplicación web para la gestión de notas en Nova 360 debe ser compatible con la versión de <i>owncloud</i> 10.0.2.                    |
| RnF 2  | Requisito de confiabilidad     | La aplicación web para la gestión de notas en Nova 360 debe ser capaz de recuperarse después de haberse producido un fallo de software. |
| RnF 3  | Requisito de funcionalidad     | La aplicación web para la gestión de notas en Nova 360 realizará las operaciones indicadas por el usuario en cada momento.              |
| RnF 4  | Requisito de mantenibilidad    | La aplicación web para la gestión de notas en Nova 360 debe ser capaz de garantizar la facilidad de mantenimiento y modificación.       |
| RnF 5  | Requisito de usabilidad        | La aplicación web para la gestión de notas en Nova 360 debe ser fácil de usar y mostrar las interfaces en idioma español.               |

#### 2.2.4 Descripción de requisitos de software mediante Historias de usuario

En la descripción de los requisitos se elaboraron 9 historias de usuarios, las mismas se pueden observar en los (Anexo 8, Anexo 9 y Anexo 10). En el epígrafe se presenta un ejemplo de la descripción de los requisitos realizada mediante este producto de trabajo.

Tabla 6. Descripción de la Historia de Usuario del RF.1 Añadir nota

(Fuente: elaboración propia)

|   |  |                                  |  |
|---|--|----------------------------------|--|
| <b>Número:</b> HU_1   | <b>Nombre del requisito:</b> Añadir nota |                                  |  |
| <b>Programadora:</b> Aideé Mora Urdaneta  |  | <b>Iteración Asignada:</b> 1     |  |
| <b>Prioridad:</b> Alta  |  | <b>Tiempo Estimado:</b> 20 horas |  |
| <b>Riesgo en Desarrollo:</b> no aplica  |  | <b>Tiempo Real:</b> 15 horas     |  |
| <b>Descripción:</b> La aplicación web permite al usuario añadir una nota.<br>1. El usuario selecciona la opción “Nueva nota” en el menú superior.<br>2. Introduce el nombre de la nota.<br>3. Selecciona la categoría para la nota.<br>4. Adiciona la nota. |  |                                  |  |
| <b>Observaciones:</b> Para añadir una nota no es necesario llenar los campos Categoría.   |  |                                  |  |

**Prototipo elemental de interfaz gráfica de usuario:**



Figura 2. Prototipo elemental de interfaz gráfica de usuario del RF. Añadir nota.

(Fuente: elaboración propia)

### 2.2.5 Validación de requisitos de software

El objetivo principal de la validación de requisitos de software es comprobar que los requisitos definidos en la ERS se corresponden con las necesidades del negocio de los clientes y los usuarios (MADEJA, 2018). A continuación se describen las diferentes técnicas de validación utilizadas para la evaluación de los requisitos de la propuesta de solución.

#### Prototipado de interfaz de usuario

El prototipado de interfaz de usuario es una técnica de representación aproximada de la interfaz de usuario de un software que permite a clientes, usuarios y equipo de desarrollo entender más fácilmente la propuesta de solución a los problemas del negocio a informatizar. Está definido como la creación u obtención de modelos concretos o la aproximación de ideas mediante una serie de tecnologías para demostrar conceptos y probar opciones de diseño, posibilitando la validación y optimización de los mismos en un tiempo e inversión mínima (Idrovo Tapia, y otros, 2014).

Los prototipos de interfaz de usuario realizados a los requisitos obtenidos fueron analizados con el programador, analista y el jefe de proyecto de Nova 360.

#### Revisión técnica formal

La revisión técnica formal (RTF) es una actividad de control de la calidad del software que se realiza por el equipo de desarrollo. Es un proceso de revisión rigurosa el cual tiene como objetivo detectar errores en la función, la lógica o la implementación de cualquier producto de trabajo, así como las desviaciones que se van generando a lo largo del desarrollo del software (Aguirre, y otros, 2013).

Los productos de trabajo generados en las disciplinas Requisitos y Análisis y diseño fueron revisados por la analista y la administradora de la calidad del proyecto de Nova 360.

## Diseño de casos de pruebas

El principal objetivo del diseño de casos de pruebas (DCP) es crear un conjunto de casos de prueba que sean efectivos descubriendo defectos en los programas y mostrando que el sistema satisfaga sus requisitos. Para su diseño se seleccionan las funcionalidades que se están probando, un conjunto de entradas que ejecutan dicha característica, se documentan las salidas esperadas o rasgos de salida y donde sea posible se diseña una prueba automatizada que demuestre que las salidas reales y las esperadas son las mismas (Carvajal, 2016). En el desarrollo de la propuesta de solución se elaboraron 9 diseños de casos de prueba. A continuación en la Figura 3 se presenta el DCP para el requisito funcional 1 Añadir nota.

### Caso de prueba del RF. 1 Añadir nota

**Descripción general:** Permite al usuario añadir una nota.

**Precondiciones:** No aplica.

| Escenario  | Descripción  | Nombre     | Categoría   | Color     | Respuesta del sistema   | Flujo central  |
|--|--|------------|-------------|-----------|---|--|
| EC 1.1 Añadir nota con una categoría existente           | La aplicación permite añadir una nota.                           | Grupo 1503 | Importantes | no aplica | Añade la nota y la visualiza en la interfaz principal del sistema | <ol style="list-style-type: none"> <li>1. Seleccionar la opción "Nueva nota" en el menú superior.</li> <li>2. Introducir el nombre de la nota.</li> <li>3. Seleccionar la categoría para la nota.</li> <li>4. Añadir la nota.</li> </ol>   |
| EC 1.2 Añadir nota con una nueva categoría               | La aplicación permite añadir una nota.                           | Grupo 1503 | Proyecto    | #0000ff   | Añade la nota y la visualiza en la interfaz principal del sistema | <ol style="list-style-type: none"> <li>1. Seleccionar la opción "Nueva nota" en el menú superior.</li> <li>2. Introducir el nombre de la nota.</li> <li>3. Seleccionar una nueva categoría.</li> <li>4. Introducir el nombre para la nueva categoría.</li> <li>5. Seleccionar el color para la nueva categoría.</li> <li>6. Añadir la nota.</li> </ol> |
| EC 1.2 Añadir la nota dejando campos obligatorios vacíos | La aplicación permite verificar que no haya campos obligatorios. | vacio      | vacio       | vacio     | No añade la nota  | <ol style="list-style-type: none"> <li>1. Seleccionar la opción "Nueva nota" en el menú superior</li> <li>2. Dejar campos obligatorios vacíos.</li> </ol>  |
| EC 1.3 Cancelar la operación                             | La aplicación permite cancelar la operación.                     | Grupo 2504 | Esculela    | #000000   | Cancelar la operación   | <ol style="list-style-type: none"> <li>1. Seleccionar la opción nueva nota en el menú superior.</li> <li>2. Introducir el nombre de la nota.</li> <li>3. Seleccionar la categoría para la nota.</li> <li>4. Cancelar la operación.</li> </ol>  |

Figura 3. Diseño de caso de prueba del RF. 1 Añadir Nota

(Fuente: elaboración propia)

### Descripción de las variables

| No | Nombre de variable | Clasificación      | Valor Nulo | Descripción  |
|----|--------------------|--------------------|------------|--|
| 1  | Nombre             | Campo de texto     | No         | Este campo solo puede contener números y letras.   |
| 2  | Categoría          | Campo de selección | Sí         | En este campo el usuario selecciona una categoría. |
| 3  | Color              | Campo de selección | Sí         | En este campo el usuario selecciona un color.      |

Figura 4. Descripción de las variables

(Fuente: elaboración propia)

### 2.2.6 Administración de requisitos de software

La administración de requisitos de software es un proceso que tiene por objetivo comprender y controlar los cambios en los requisitos de software. Comprende las actividades relacionadas con la definición, clasificación, asignación, seguimiento y control de los requisitos durante todo el ciclo de vida de desarrollo de software. Es indispensable para asegurar la calidad de los productos, así como para llevar control y seguimiento de los proyectos (de la Cruz Londoño, y otros, 2015).

En la propuesta de solución se utilizaron las matrices de trazabilidad para la administración de los requisitos de la aplicación web desarrollada, lo que permitió seguir la evolución del ciclo de vida del proyecto. Para ello se definieron los siguientes elementos para la trazabilidad de requisitos en correspondencia con el documento Guía de Trazabilidad establecido por la Dirección de producción de la Universidad de las Ciencias Informáticas para la actividad productiva de software.

- Requisitos con Requisito
- Requisitos con Especificación de requisitos de software
- Requisito con Diagrama de clases del diseño
- Requisitos con Diseño de casos de pruebas

A continuación en la Figura 5 se muestra la matriz de trazabilidad de Requisitos-Diseño de casos de pruebas, elaborada en el desarrollo de la propuesta de solución:

By:  ▾

(9) Requirement

| (9) Requirement                          | DCP 1. Añadir Nota | DCP 4. Añadir categoría a la nota | DCP 5. Añadir color a la nota | DCP 6. Buscar notas por categoría | DCP 7. Buscar notas por contenido ... | DCP 2. Editar Nota | DCP 3. Eliminar una Nota | DCP 8. Sincronizar notas desde Nova | DCP 9. Sincronizar notas desde Nov... |
|--|--------------------|-----------------------------------|-------------------------------|-----------------------------------|---------------------------------------|--------------------|--------------------------|-------------------------------------|---------------------------------------|
| RF.1 Añadir nota                         | ↔                  |                                   |                               |                                   |                                       |                    |                          |                                     |                                       |
| RF.4 Añadir categoría a la nota          |                    | ↔                                 |                               |                                   |                                       |                    |                          |                                     |                                       |
| RF.5 Añadir color a la nota              |                    |                                   | ↔                             |                                   |                                       |                    |                          |                                     |                                       |
| RF.6 Buscar notas por categoría          |                    |                                   |                               | ↔                                 |                                       |                    |                          |                                     |                                       |
| RF.7 Buscar notas por contenido de texto |                    |                                   |                               |                                   | ↔                                     |                    |                          |                                     |                                       |
| RF.2 Editar Nota                         |                    |                                   |                               |                                   |                                       | ↔                  |                          |                                     |                                       |
| RF.3 Eliminar una Nota                   |                    |                                   |                               |                                   |                                       |                    | ↔                        |                                     |                                       |
| RF.8 Sincronizar notas desde Nova        |                    |                                   |                               |                                   |                                       |                    |                          | ↔                                   |                                       |
| RF.9 Sincronizar notas desde NovaDroid   |                    |                                   |                               |                                   |                                       |                    |                          |                                     | ↔                                     |

Figura 5. Matriz de trazabilidad de Requisitos-Diseño de casos de pruebas

(Fuente: elaboración propia)

La disciplina de requisitos permitió definir la propuesta de solución generando especificaciones correctas que permiten describir con claridad las necesidades existentes en el proyecto de Nova 360, sirviendo como una base de apoyo para el análisis y diseño del producto de software que se describe a continuación.

## 2.3 Análisis y Diseño

En esta disciplina, si se considera necesario, los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de estos, y una descripción que sea fácil de mantener y ayude a la estructuración del sistema (incluyendo su arquitectura). Además, en esta disciplina se modela el sistema y su forma para que soporte todos los requisitos, incluyendo los requisitos no funcionales. Los modelos desarrollados son más formales y específicos (Rodríguez, 2015).

### 2.3.1 Diseño de clases

Un diagrama de clases es una estructura estática que describe la estructura de un sistema mostrando las clases, sus atributos, operaciones y las relaciones entre los objetos. Contiene principalmente los requisitos funcionales de un sistema y los servicios que el sistema debe proporcionar a sus usuarios finales (UML, 2018). A continuación se presenta en la Figura 6 el diagrama de clases del diseño de la propuesta de solución.



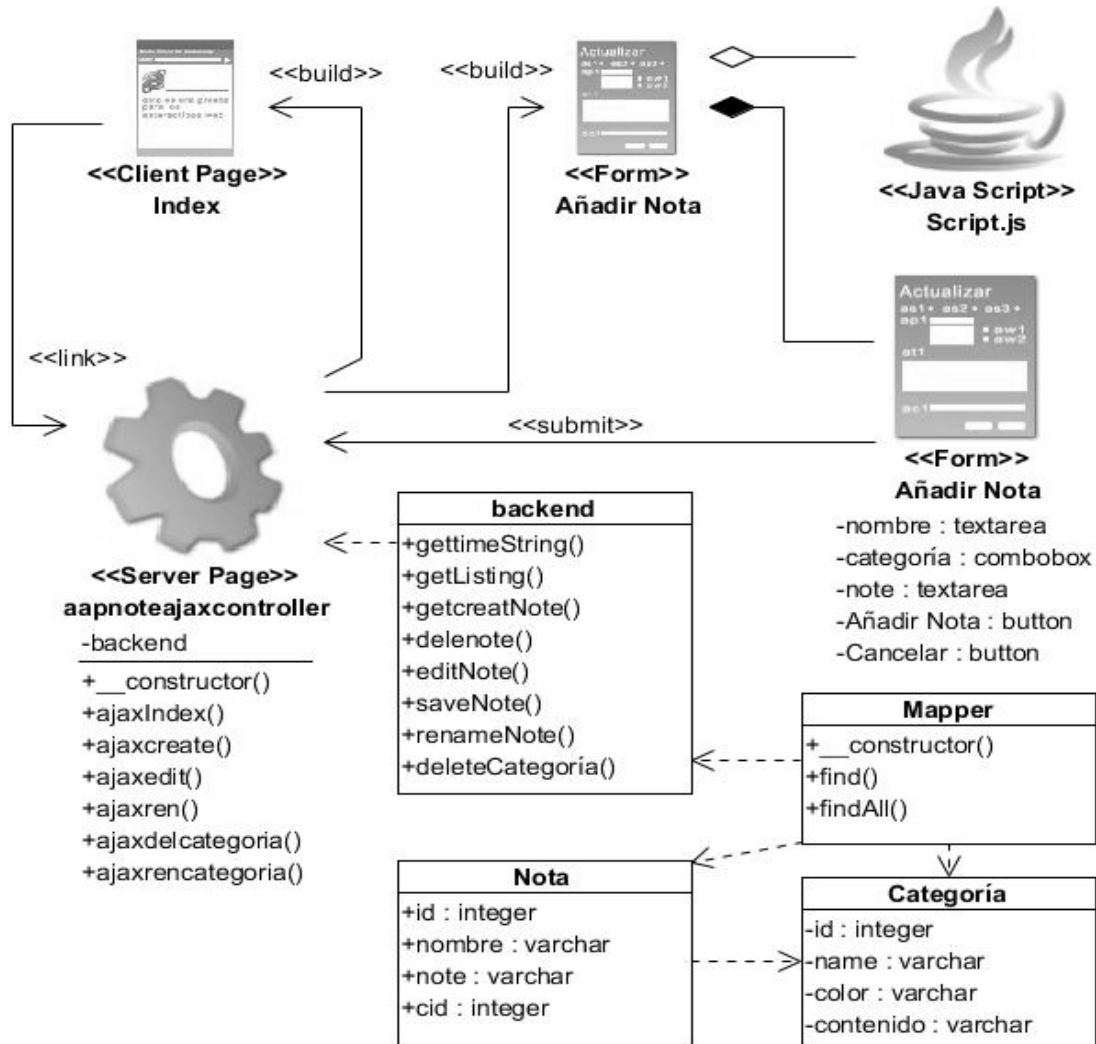


Figura 6. Diagrama de clases del diseño de la propuesta de solución

(Fuente: elaboración propia)

El diagrama de clases del diseño de la Figura 6 está formado por las siguientes clases:

- Clases de interfaz: definen todas las abstracciones necesarias para la interacción entre los usuarios y la aplicación. Están formadas por las vistas *Index* y *Adicionar Nota*.
- Clase controladora: permiten encapsular las funcionalidades necesarias para interactuar con las clases interfaces, las clases entidades y de acceso a los datos. Agrupa la lógica del negocio ya que responde a las peticiones HTTP provenientes del cliente e invoca peticiones a la base de datos. En este caso es la clase *aapnoteajaxcontroller*.
- Clase de acceso a datos: permite la comunicación entre la clase controladora del negocio de la aplicación y la base de datos. En el diagrama están representadas por las clases *backend*.
- Clases persistentes: representan el almacenamiento de datos que persistirá más allá de la ejecución del sistema. Está representada por la clases de entidad *Nota* y *Categoría*.

### 2.3.2 Patrones de diseño de software

Un patrón de diseño es una comunicación entre clases y objetos que se relacionan entre sí y se adaptan para resolver un problema de diseño general en un contexto particular. Es una herramienta de apoyo en la búsqueda de soluciones a problemas comunes durante el desarrollo del software. Permite reutilizar con éxito los diseños y arquitecturas, ayudando a elegir entre diseños alternativos (Ramos, 2015). En el diseño de la propuesta de solución se utilizaron los siguientes patrones:

#### GRASP

Los patrones GRASP (*General Responsibility Assignment Software Patterns*, Patrones Generales de Software para Asignación de Responsabilidades) describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Estos patrones fomentan un conjunto de buenas prácticas para el diseño de software. Entre estos patrones se hallan: Experto, Creador, Controlador, Bajo acoplamiento y Alta cohesión (Ramos, 2015). A continuación se describen los patrones utilizados en el diseño de la propuesta de solución:

**Experto:** Es el patrón que define bajo qué principio se debe asignar responsabilidades a los objetos. Cada clase, contiene toda la información necesaria para realizar la labor que tiene encomendada (Ramos, 2015). Este patrón se evidencia en las clases *Nota* y *Categoría* (Figura 7) ya que son las que contienen toda la información sobre las notas y las categorías respectivamente.

**Creador:** Es el patrón que define quién debe ser el responsable de crear una nueva instancia de la clase. Permite la visibilidad entre las clases creada y la clase creador (Ramos, 2015). Este patrón se evidencia en la clase *Backend* (Figura 7) la cual tiene la responsabilidad de crear una nota.

**Bajo Acoplamiento:** Es el patrón que define como mantener bajas dependencias. Permite asignar una responsabilidad de manera que el acoplamiento sea bajo. Asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas (Ramos, 2015). Este patrón se evidencia en la relación entre el controlador *appnoteajaxcontroller* y la clase *backend* (Figura 7).

**Alta cohesión:** Es el patrón que asegura la necesidad de mantener el sistema con un bajo nivel de complejidad, es decir, una clase tiene una responsabilidad moderada en un área funcional y colabora con otras clases para llevar a cabo las tareas. Este patrón se evidencia en la clase *backend* (ver Figura 7).

#### Inyección de Dependencia

La Inyección de Dependencia (DI, *Dependency Injection*) es un patrón de diseño de software usado en la Programación Orientada a Objetos, que trata de solucionar las necesidades de creación de los objetos de una manera práctica, útil, escalable y con una alta versatilidad del código. Suministra

objetos a una clase, en lugar de ser la clase la que cree dichos objetos. Demuestra que los objetos nunca deben construir aquellos otros objetos que necesitan para funcionar (Alvarez, y otros, 2015).

Este patrón se evidencia en la clase `appnoteajaxcontroller` (Figura 7) en la cual se inyectan las dependencias necesarias para su funcionamiento. Las mismas han sido registradas con anterioridad en el contenedor de `owncloud`.

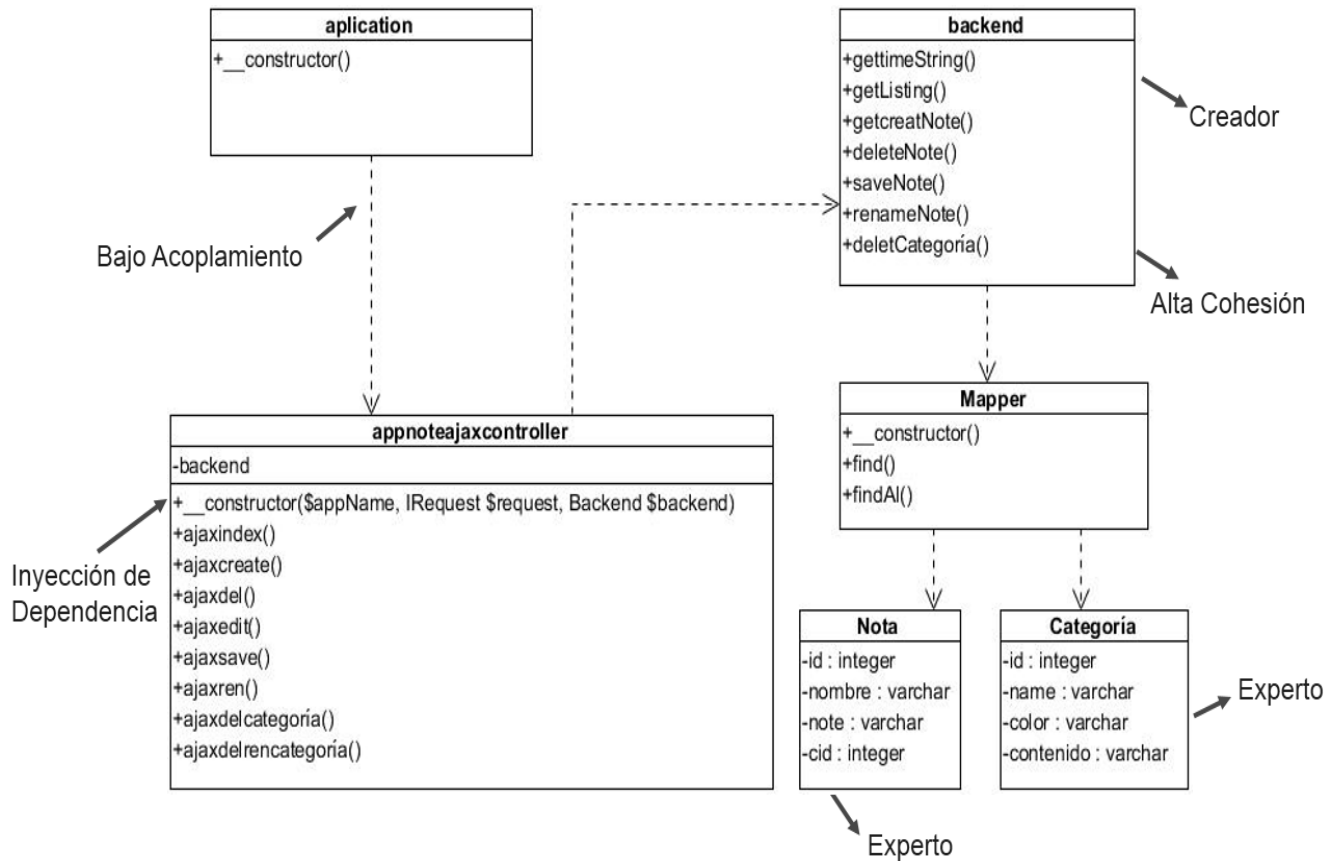


Figura 7. Aplicaciones de los patrones GRASP e Inyección de dependencias

(Fuente: elaboración propia)

### 2.3.3 Modelo de datos

Un modelo de datos muestra la estructura lógica de la base de datos resultante del negocio a informatizar, incluidas las relaciones y limitaciones que determinan cómo se almacenan los datos y se accede a ellos. La mayoría de los modelos de datos se pueden representar por medio de un diagrama entidad-relación (Lucidchart, 2018). En la Figura 8 se muestra el modelo de datos de la propuesta de solución. En él se evidencia las entidades de persistencia de datos `appnote` y `appnote_category`, sus atributos y relaciones.

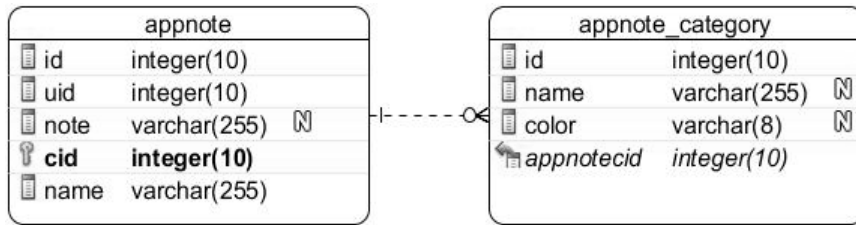


Figura 8. Modelo de datos

(Fuente: elaboración propia)

### 2.3.4 Diseño arquitectónico

El diseño arquitectónico es una actividad que permite estructurar de forma general un software. Su objetivo es tener una visión clara del software a construir. En él se identifican los componentes en los que se divide una aplicación y se establece un marco de control y comunicación entre ellos. Una de las buenas prácticas existentes en el diseño arquitectónico de sistemas informáticos es la utilización de patrones. Estos ofrecen soluciones a problemas de arquitectura de software, adaptabilidad a requisitos cambiantes, rendimiento, modularidad y acoplamiento. Especifican responsabilidades y una serie de recomendaciones para organizar los distintos elementos del sistema (MAJEDA, 2018). En la propuesta de solución se utiliza el patrón MVC (*Model – View – Controller*, Modelo – Vista – Controlador) descrito a continuación.

#### Patrón arquitectónico MVC

MVC permite separar el sistema en 3 capas, el Modelo es la capa de acceso a la base de datos, contiene toda la información sobre los datos, cómo acceder a estos y validarlos, así como su comportamiento y relaciones que existen entre ellos. La Vista es la capa lógica del negocio incluye la lógica que accede al modelo y la delega en la plantilla apropiada. El Controlador dependiendo de la acción solicitada por el usuario es el que pide al modelo la información necesaria e invoca a la plantilla (de la vista) que corresponda para que la información sea presentada (MAJEDA, 2018). En la Figura 9 se evidencia la estructura de separación en capas propuesta en MVC.



Figura 9. Modelo Vista Controlador

(Fuente: maestrodelaWeb.com)

En la Figura 10 se evidencia la utilización del patrón arquitectónico MVC en la propuesta de solución. En ella se presenta la estructura de la aplicación web para la gestión de notas en Nova 360 en las siguientes capas:

- Modelo: contiene las clases entidades *category.php* y *note.php*
- Controlador: contiene los controladores encargados de acceder al modelo e invocar la plantilla correspondiente. Presentados controladores *appnoteajaxcontroller.php* y *appnoteajaxcontroller.php*
- Vista: contiene dos plantillas *admin.php* y *main.php*

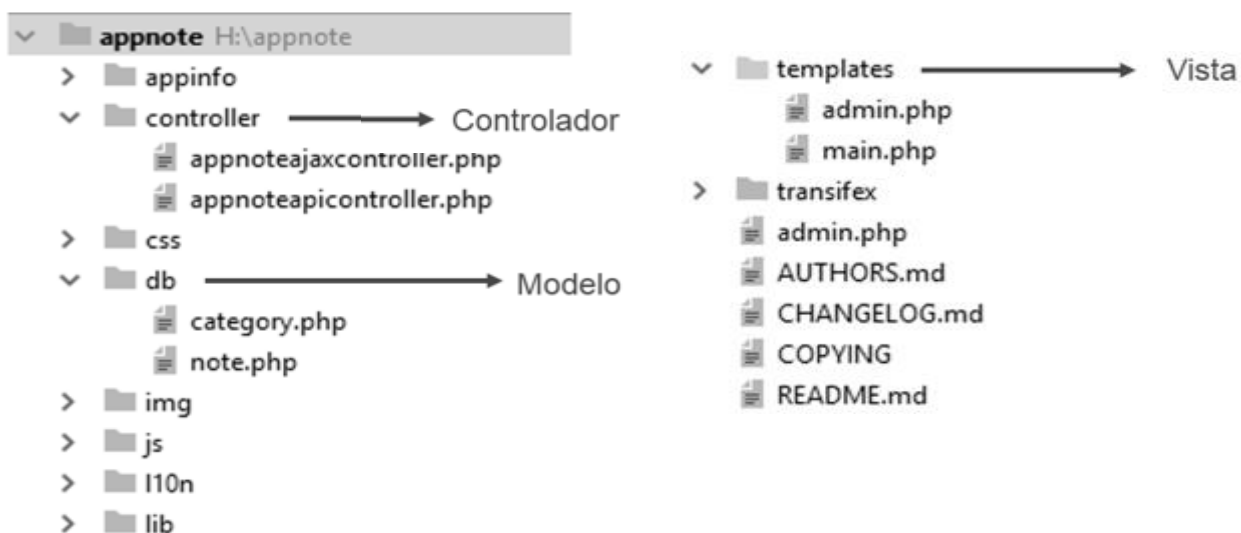


Figura 10. Arquitectura de la Solución

(Fuente: elaboración propia)

## 2.4 Conclusiones del capítulo

En el análisis y diseño de la aplicación web para la gestión de notas en Nova 360 se identificaron 9 requisitos funcionales y 5 no funcionales cumpliendo con las necesidades del cliente. Se elaboró el diagrama de clases del diseño que permitió modelar las diferentes clases, sus atributos, tipo de datos y relaciones. Se emplearon los patrones de diseño de software GRASP e Inyección de dependencias para proporcionarle una mayor calidad al software desarrollado. Se representó la estructura de la base de datos mediante el diagrama de entidad-relación compuesto por las entidades `appnote` y `appnote_category`. La utilización del patrón arquitectónico MVC garantizó la organización y comunicación entre los diferentes componentes de la propuesta de solución.

## **Capítulo 3: Implementación y evaluación de la aplicación web para la gestión de notas en Nova 360**

### **Introducción**

El capítulo se enfoca en la construcción de la aplicación web a partir de los resultados del análisis y diseño. En él se presenta el diagrama de despliegue, se definen los estándares de codificación a utilizar en la implementación de la solución. Además, se realizan pruebas de software con el objetivo de descubrir y corregir errores. Se evalúa el índice de satisfacción grupal de los potenciales usuarios hacia la propuesta de solución a través de la técnica de ladov.

### **3.1 Implementación**

El objetivo de esta disciplina es construir el sistema, a partir de los resultados del Análisis y Diseño (Rodríguez, 2015). En la implementación de la aplicación web para la gestión de notas en Nova 360 se utilizaron los estándares de codificación que se describen a continuación.

#### **3.1.1 Estándares de codificación**

Estándares de codificación para PHP utilizados en el Manual de Desarrollo de *Owncloud* basado en los estándares de codificación PEAR (*Extension Application Repository*, Repositorio de Aplicaciones de Extensión) (Marcos, 2017).

#### **Identación**

- Se utilizaron 4 espacios sin caracteres de tabulación.

#### **Tabuladores o Espacios**

- Las estructuras de control deben tener un espacio entre el *keyword* (palabra clave) de la estructura y el signo de apertura de paréntesis para distinguir entre las llamadas de las funciones.
- Las funciones son llamadas sin espacios entre el nombre de la función, el signo de paréntesis y el primer parámetro; espacios entre cada coma por parámetro y sin espacios entre el último paréntesis, el signo de paréntesis cerrado y el signo de punto y coma (;).

#### **Tamaño máximo de línea**

- Se limitaron todas las líneas a un máximo de 80 caracteres.

#### **Estándares de Nomenclatura**

- Se utilizó *Upper Camel Case* (la primera letra de cada palabra es mayúscula) para el nombre de las clases.

- Se utilizó *Lower Case* (todas las letras con minúscula) para los nombres de los métodos en los controladores y *Lower Camel Case* (la letra inicial de la primera palabra con minúscula y el resto de las palabras comienzan con mayúscula) para el nombre de los métodos de las clases.

### Otras Consideraciones

- Se rodearon siempre los siguientes operadores binarios con un espacio en cada lado.
- Se utilizaron espacios alrededor de los operadores aritméticos.

A continuación en la Figura 11 se presenta un fragmento de código del controlador AppnoteAjaxController donde se evidencia el uso de los estándares de codificación utilizados.

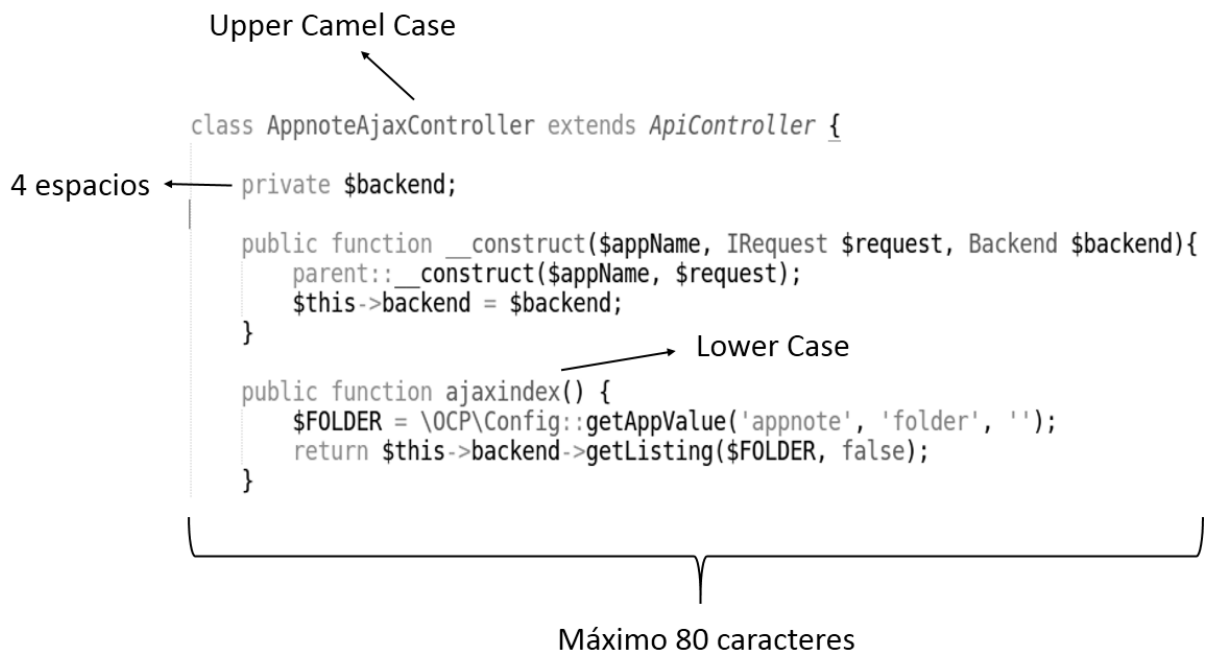


Figura 11. Estándar de codificación

(Fuente: elaboración propia)

En el epígrafe 3.1.2 se evidencia el resultado de la implementación de la aplicación web para la gestión de notas en Nova 360 mediante la interfaz gráfica de usuario. En su implementación se tuvo en cuenta las reglas de oro definidas por Pressman 2010 para el diseño de interfaces gráficas.

### 3.1.2 Ejemplo de la interfaz gráfica de usuario de la aplicación web para la gestión de notas

La interfaz gráfica de usuario es un conjunto de formas y métodos que posibilitan a los usuarios la interacción con un sistema, empleando para esto gráficos e imágenes. Su función principal consiste en facilitar un entorno sencillo que permita la comunicación con el ordenador (Albornoz, 2014). Para el diseño de las interfaces gráficas de usuario de la propuesta de solución se tuvieron en cuenta las siguientes reglas.



### Reglas de Oro (Pressman, 2010)

- Dar control al usuario: Definir los modos de interacción de forma que el usuario no realice acciones innecesarias o indeseables, proporcionar una interfaz flexible, incluir opciones de interrumpir y deshacer la interacción del usuario, ocultar al usuario ocasional los elementos técnicos internos, diseñar interacción directa con los objetos que aparecen en pantalla.
- Reducir la carga en memoria del usuario: Reducir la demanda de memoria a corto plazo, definir valores por defecto que tengan significado, definir accesos directos intuitivos, el formato visual de la interfaz debe basarse en la metáfora tomada de la realidad, desglosar la información de manera progresiva.
- Lograr que la interfaz sea consistente: Permitir que el usuario incluya la tarea actual en el contexto que tenga algún significado.



Figura 12. Interfaz gráfica de usuario para gestionar notas

(Fuente: elaboración propia)

La interfaz de la Figura 12 le permite al usuario gestionar sus notas. En ella se puede adicionar una nota con categoría, color. Se muestra el listado de las notas del usuario organizada por su categoría, además brinda la opción de eliminarlas y buscarlas por su contenido.

En el epígrafe 3.1.3 se presenta el diagrama de despliegue de la aplicación para la gestión de notas en Nova 360. En él se detalla las características del entorno real de ejecución de la aplicación.

### 3.1.3 Diagrama de despliegue

El diagrama de despliegue describe la distribución física de un sistema informático en términos de cómo se distribuyen las funcionalidades entre los nodos que integran el sistema. Muestra las



relaciones físicas entre los componentes de hardware y software. Permite visualizar la distribución de los componentes de software en los nodos físicos (Zurdo, y otros, 2014). A continuación en la Figura 13 se presenta el diagrama de despliegue elaborado para la propuesta de solución.

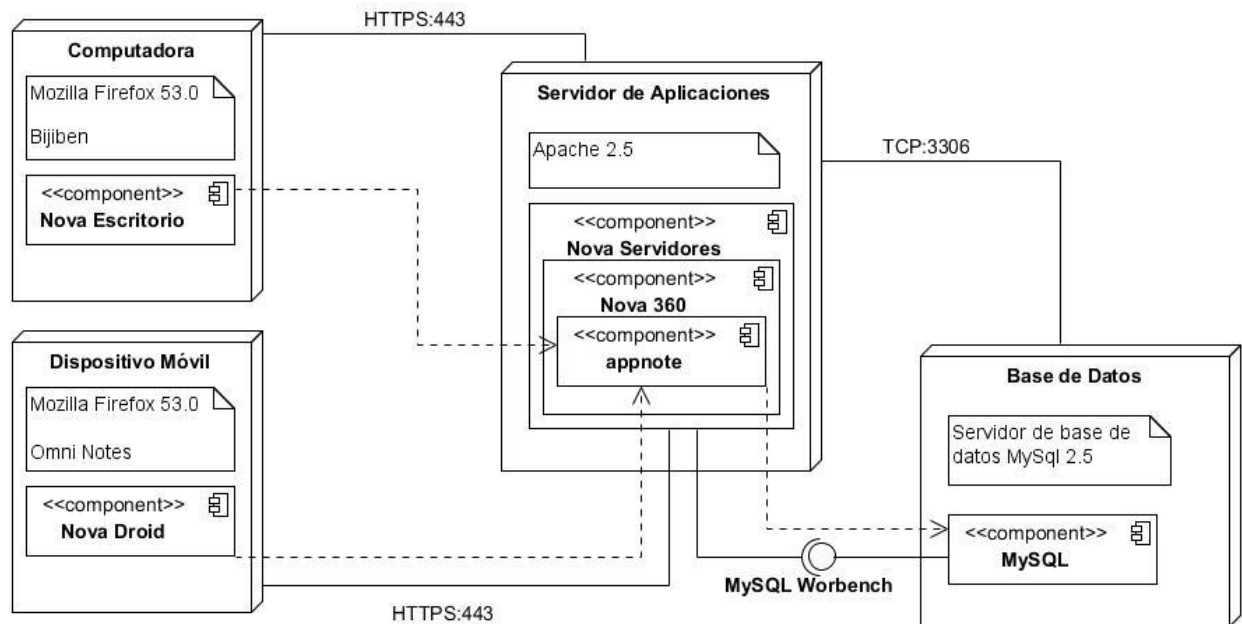


Figura 13. Diagrama de Despliegue  
(Fuente: elaboración propia)

#### Descripción de los nodos:

- Computadora: desde la computadora se podrá acceder a la aplicación a través de un navegador web o la aplicación para la gestión de notas Bijiben.
- Dispositivo Móvil: desde el dispositivo móvil se podrá acceder a la aplicación a través de un navegador web o la aplicación para la gestión de notas Omni Notes.
- Servidor de Aplicaciones: proporciona los servicios de la aplicación y gestiona las funciones de la lógica del negocio y acceso a los datos necesarios, el servidor de la aplicación será Apache en su versión 2.5.
- Base de Datos: servidor de base de datos MySQL en su versión 2.5.

#### Descripción de los componentes:

- Nova Escritorio: variante de la distribución cubana de GNU/Linux Nova.
- Nova Droid: personalización del sistema operativo Android.
- Nova Servidores: variante de la distribución cubana de GNU/Linux para servidores.
- Nova 360: personalización de Owncloud.
- appnote: aplicación de notas para Nova 360.

- MySQL: servidor de base de datos en su versión 2.5.

#### **Descripción de los protocolos:**

- HTTPS (*Hypertext Transfer Protocol*, Protocolo de Transferencia de Hipertexto): Permite la transferencia segura de datos.
- TCP (*Transmission Control Protocol*, Protocolo de Transmisión de Control): Permite enviar conexiones entre sí facilitando el flujo de datos.

### **3.2 Pruebas de software para la evaluación de la aplicación de gestión de notas**

Las pruebas de software son la ejecución del código usando combinaciones de entradas, en un determinado estado, para revelar defectos. Su objetivo fundamental es verificar los requisitos del sistema. Consiste en la verificación dinámica del comportamiento de un programa en un conjunto finito de casos de prueba, adecuadamente seleccionado de los posibles escenarios del sistema, para asegurarse que arroja el resultado definido en la especificación de requisitos (Pérez, y otros, 2014). En los subepígrafos siguientes se describen los tipos de pruebas de software, métodos y técnicas aplicados para la evaluación de la aplicación desarrollada.

#### **3.2.1 Tipos de pruebas de software**

##### **Pruebas unitarias**

Las pruebas unitarias, también llamadas pruebas de componentes, se encargan de probar, individualmente, subprogramas, subrutinas o procedimientos en un programa. El propósito de este tipo de pruebas es comparar la función de una unidad con cierta especificación funcional que define la unidad. Centran su actividad en verificar la funcionalidad y la estructura (lógica interna) de cada elemento individualmente, una vez que ha sido codificado (Cabrera, 2014).

##### **Pruebas funcionales**

Las pruebas funcionales, también llamadas pruebas de comportamiento se centran en obtener conjuntos de condiciones de entrada que ejerciten completamente los requisitos funcionales de un programa. La validación se alcanza cuando el software funciona de tal manera que satisface las expectativas razonables del cliente (Cabrera, 2014).

##### **Pruebas de integración**

Las pruebas de integración se realizan para integrar los diferentes componentes de un sistema informático. El objetivo es tomar estos componentes a los que se aplicó las pruebas unitarias, para comprobar que funcionan correctamente, e integrarlos a un software base para construir la estructura del programa. Se centra principalmente en probar la comunicación entre los diferentes elementos del software (Pressman, 2010).

### **3.2.2 Métodos de prueba de software**

**Caja Negra:** se centran en obtener conjuntos de condiciones de entrada que ejerciten completamente los requisitos funcionales de un programa. Tiene como objetivo validar que las salidas sean las esperadas (Cabrera, 2014).

**Caja Blanca:** tiene como objetivo analizar el objeto de prueba, ejecutando el mismo. Se llevan a cabo por medio de revisiones y herramientas. Permite disminuir en un gran porcentaje el número de errores existentes en los sistemas (Cabrera, 2014).

### **3.2.3 Técnicas de prueba de software**

**Partición de equivalencia:** divide en subconjuntos (clases de equivalencia) los valores de los parámetros del sistema (o módulo) al asumir que todos los elementos en el mismo subconjunto resultan en un comportamiento similar (Cabrera, 2014).

**Camino básico:** permite obtener una medida de la complejidad lógica de un diseño procedimental y que use esta medida como guía para definir un conjunto básico de rutas de ejecución. Los casos de pruebas derivados para ejercitar el conjunto básico deben garantizar que se ejecuta cada instrucción por lo menos una vez durante la prueba (Pressman, 2010).

**Prueba basada en subprocesos:** integra el conjunto de clases requerido para responder a una entrada o evento del sistema. Cada subproceso se integra y prueba individualmente (Pressman, 2010).

## **3.3 Aplicación de las pruebas de software**

En el epígrafe se presenta los productos de trabajo desarrollados en las disciplinas de Pruebas internas y Pruebas de aceptación propuestas por la metodología de desarrollo AUP-UCI. Contiene la aplicación y resultados de las pruebas de software definidas anteriormente y realizadas mediante los métodos y técnicas descritas en los subepígrafes 3.2.2 y 3.2.3.

### **3.3.1 Pruebas internas**

En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas (Rodríguez, 2015). En la disciplina se aplicaron pruebas unitarias, funcionales, funcionales para el Api Rest y de integración, descritas a continuación.

#### **Pruebas unitarias**

Las pruebas unitarias se utilizaron usando la técnica del camino básico del método de prueba de caja blanca. En esta técnica se utilizó la métrica de software complejidad ciclomática que proporciona una medida cuantitativa de la complejidad lógica de un programa o procedimiento. Define el número de

caminos independientes en el conjunto básico de un programa o procedimiento y proporciona un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez (Pressman, 2010). En la Figura 14 se presenta el método createNote del RF1 Adicionar Nota.

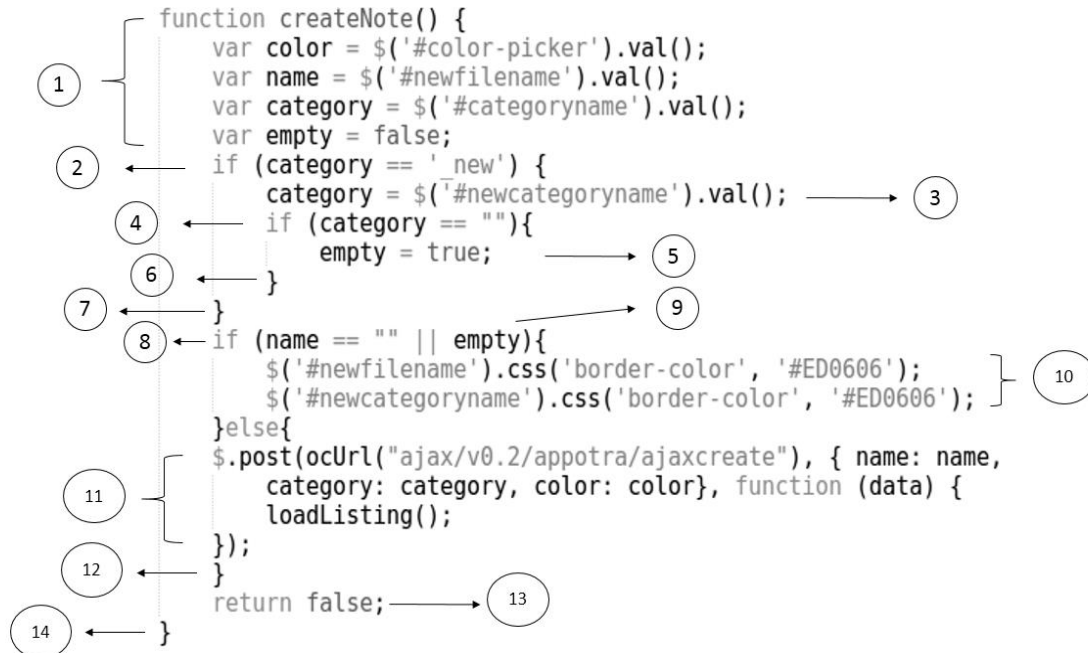


Figura 14. Método createNote del RF1 Adicionar Nota

(Fuente: elaboración propia)

### 1. Dibujar el grafo de flujo de la funcionalidad o procedimiento a analizar

En la Figura 14 se utilizó la notación de grafo de flujo para representar el código del método a probar.

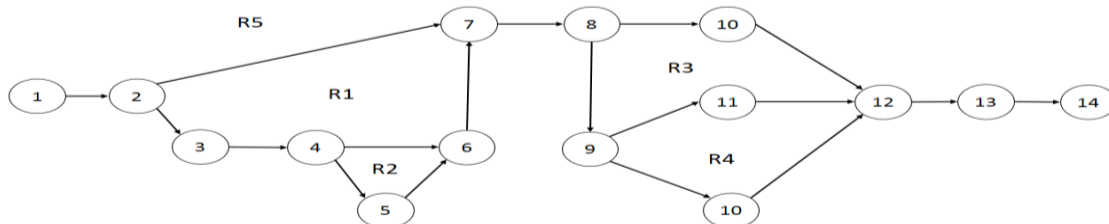


Figura 15. Grafo de flujo

(Fuente: elaboración propia)

### 2. Determinar la complejidad ciclomática del grafo

La complejidad ciclomática del grafo  $V(G)$  se puede calcular de las tres formas siguientes:

$$V(G) = A - N + 2 \quad \text{Donde: } A \text{ es el número de aristas del grafo de flujo y } N \text{ es el número de nodos}$$

$V(G) = 18 - 15 + 2$  del grafo.

$V(G) = 5$

$V(G) = P + 1$

Donde: P es el número de nodos predicados (nodos con más de una arista de salida) contenidos en el grafo.

$V(G) = 4 + 1$

$V(G) = 5$

$V(G) = R$

Donde: R son las regiones, áreas delimitadas por nodos y aristas en el grafo.

$V(G) = 5$

### 3. Determinar los caminos linealmente independientes

- Camino básico 1: 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14
- Camino básico 2: 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14
- Camino básico 3: 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 13, 14
- Camino básico 4: 1, 2, 3, 4, 6, 7, 8, 10, 12, 13, 14
- Camino básico 5: 1, 2, 7, 8, 10, 12, 13, 14

### 4. Definir los casos de prueba para comprobar la ejecución de cada camino

En el diseño de los casos de prueba se debe especificar los siguientes elementos:

- Descripción: contiene una descripción sobre las restricciones de los datos de entrada que debe tener el caso de prueba.
- Condición de ejecución: se especifican los parámetros que deben poseer el caso de prueba para que se cumpla una condición deseada como respuesta del funcionamiento del método.
- Entrada: se muestran los parámetros de entradas del método.
- Resultados esperados: se explica el resultado esperado de la ejecución del método.

La Tabla 9 muestra el diseño de caso de prueba para el camino 5 del conjunto de caminos básicos linealmente independiente correspondiente a la funcionalidad createNote.

Tabla 7. Caso de prueba para el camino 5 del método createNote

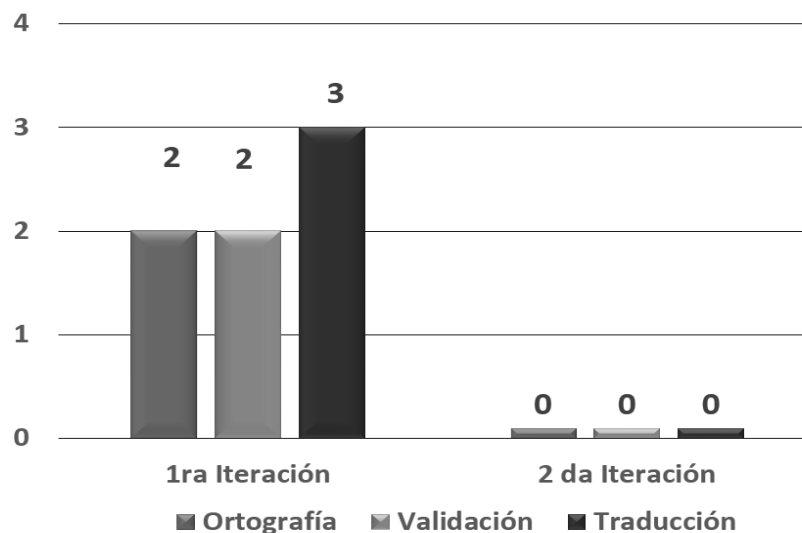
(Fuente: elaboración propia)

| <b>Diseño de caso de prueba para el camino 5</b> |  |
|--|--|
| <b>Descripción</b>                               | Método para crear una nota   |
| <b>Condición de ejecución</b>                    | El usuario selecciona la opción de adicionar una nueva nota a una categoría existente. |
| <b>Entradas</b>                                  | color = #000000<br>name = " "<br>category = Importante                                 |
| <b>Resultado esperado</b>                        | No añade la nota y señala el campo vacío de color rojo.                                |

Como resultados de las pruebas unitarias se comprobó que el flujo de trabajo de las funcionalidades de la aplicación para la gestión de notas en Nova 360 es correcto, ya que cada sentencia del código fuente es ejecutada al menos una vez durante la ejecución de la prueba.

### Pruebas funcionales

Se realizaron pruebas funcionales, a través del método de caja negra y la técnica de partición de equivalencia. El diseño de caso de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada. Se retomaron los casos de pruebas diseñados en el epígrafe 2.2.5 Validación de requisitos de software, que contiene clases de equivalencia válidas y no válidas, para cada campo de entrada del sistema. En la Figura 16 se presenta un gráfico con los resultados de las pruebas funcionales.



*Figura 16. Resultados de las pruebas funcionales  
(Fuente: elaboración propia)*

Las pruebas funcionales se realizaron en dos iteraciones, en la primera se encontraron 7 no conformidades, de ellas 2 de validación, 2 de ortografía y 3 de traducción, en la segunda iteración no se encontraron no conformidades.

### Pruebas funcionales para el Api Rest

Para probar el funcionamiento correcto del Api Rest se utilizó Insomnia Rest y Rest Cliente ambos clientes Rest que te permiten mostrar el funcionamiento de la Api.

Insomnia es un cliente Rest que te permite enviar peticiones y estudiar sus respuestas. Permite autenticación básica con HTTP. Reutiliza valores de la Api. A través de esta herramienta se visualiza el código del estado, el cuerpo y la cabecera (Burgos, 2017). A continuación en la Figura 17 se presenta un ejemplo de la prueba para el Api Rest con Insomnia Rest.

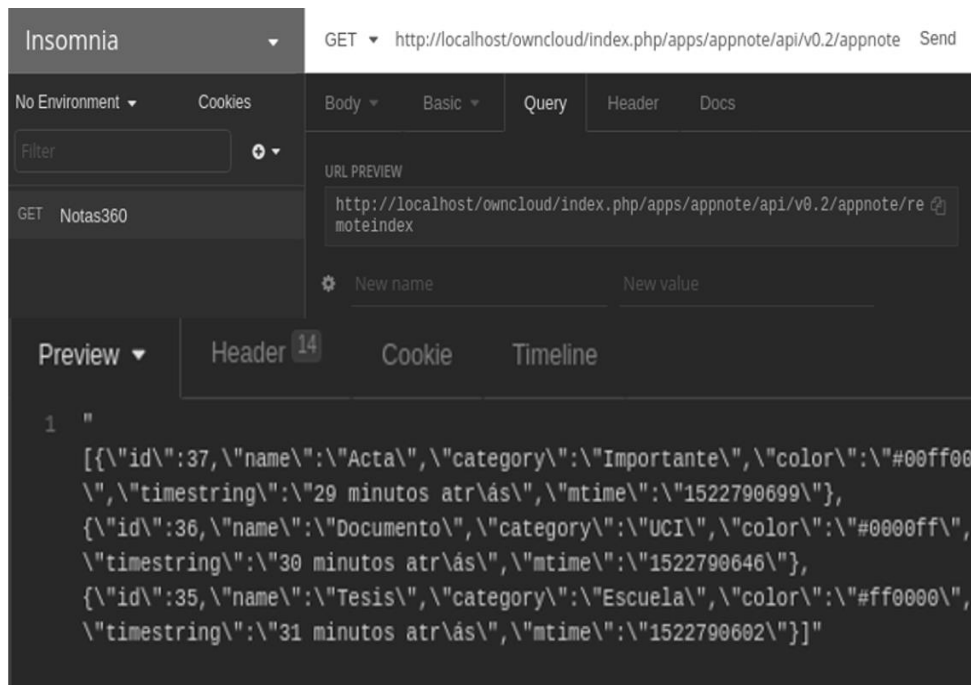


Figura 17. Prueba funcional con el Insomnia Rest

(Fuente: elaboración propia)

El resultado mostrado en la figura 17 evidencia la respuesta de Insomnia Rest después de que el usuario haya introducido una solicitud GET para consultar las notas.

Rest Client es un plugin para Firefox para realizar peticiones Rest contra un Api bien definida. Permite la autenticación básica con HTTP. Soporta gran variedad de métodos HTTP como, GET, POST, PUT, DELETE (Gómez, 2015). A continuación en la Figura 18 se presenta un ejemplo de la prueba para el Api Rest con Rest Client.



Figura 18. Prueba funcional con el Rest Client

(Fuente: elaboración propia)



El resultado mostrado en la figura 18 evidencia la respuesta de Rest Client después de que el usuario haya introducido una solicitud GET para consultar las notas y como respuesta obtiene una lista de estas.

### Pruebas de integración

Para la prueba de integración se realizó el diseño de caso de prueba mostrado en la Figura 19 para comprobar la integración de la aplicación web de gestión de notas en Nova 360.

### Descripción general

Permite instalar la aplicación web para la gestión de notas en Nova 360

### Condiciones de ejecución

El código fuente de la aplicación para la gestión de notas debe estar en la carpeta apps de Nova 360

| Escenario   | Descripción  | Respuesta del sistema  | Flujo central   |
|---|--|--|---|
| EC 1.1 Instalar la aplicación de gestión de notas                                 | Instala la aplicación de gestión de notas en Nova 360              | Instala satisfactoriamente la aplicación de gestión de notas y la visualiza en el menú "Ajustes" de Nova 360 | <ol style="list-style-type: none"> <li>1. Seleccionar del menú "admin" de Nova 360 la opción "Ajustes"</li> <li>2. Seleccionar el menú "Aplicaciones" de la sección Administración</li> <li>3. En la vista "Gestión de Apps", seleccionar la opción Mostrar appd inhabilitadas</li> <li>4. Seleccionar la aplicación de gestión de notas y seleccionar la opción "Activar"</li> </ol> |
| EC 1.2 Instalar la aplicación de gestión de notas con errores en el código fuente | La aplicación permite verificar que no haya campos obligatorios    | Nova 360 no muestra la aplicación  | <ol style="list-style-type: none"> <li>1. Seleccionar del menú "admin" de Nova 360 la opción "Ajustes"</li> <li>2. Seleccionar el menú "Aplicaciones" de la sección Administración</li> <li>3. En la vista "Gestión de Apps", seleccionar la opción Mostrar appd inhabilitadas</li> <li>4. Seleccionar la aplicación de gestión de notas y seleccionar la opción "Activar"</li> </ol> |
| EC 1.3 Visualizar el contenido de la aplicación de gestión de notas en Nova 360   | Nova 360 muestra el contenido de la aplicación de gestión de notas | Muestra satisfactoriamente el contenido de la aplicación de gestión de notas                                 | <ol style="list-style-type: none"> <li>1. Seleccionar del menú "Ajustes" de Nova 360 la aplicación de gestión de notas</li> </ol>   |

*Figura 19. Diseño de caso de prueba para la Prueba de integración*

*(Fuente: elaboración propia)*

Para instalar la aplicación web de gestión de notas en Nova 360, se debe incluir en la carpeta apps. Posteriormente se debe habilitar la aplicación para su correcto funcionamiento. En la Figura 19 se evidencia el caso de prueba diseñado para comprobar la integración de la aplicación a su software base. Está compuesto por 3 escenarios donde se describen las respuestas de los sistemas ante cada escenario.



Después de haber realizado las operaciones necesarias se compró que la aplicación web para la gestión de notas en Nova 360 se integró de manera satisfactoria.

### **3.3.2 Pruebas de aceptación**

Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido (Rodríguez, 2015).

El cliente realizó pruebas funcionales en las cuales no se detectaron no conformidades y emitió un acta de aceptación de productos de trabajo (Anexo 12) en total conformidad con la aplicación web desarrollada.

### **3.4 Evaluación de la satisfacción de los potenciales usuarios**

La satisfacción del cliente dentro del proceso de desarrollo de software se ha convertido en un punto central en la validación de cualquier investigación científica, esta información es útil para conocer las debilidades de la propuesta y profundizar en sus fortalezas. Para medir la satisfacción del cliente se usó la **técnica de ladov**, ya que es un instrumento que ayuda a conocer el grado de satisfacción de los potenciales usuarios (Calzada, 2013).

ladov es una técnica efectiva para el estudio del nivel de satisfacción de los participantes a través de la consulta a un panel de expertos. Este método calcula el Índice de Satisfacción Grupal (ISG) se implementa mediante un cuestionario (Anexo 13) en el cual se le incluyen tres preguntas cerradas que se intercalan dentro de un cuestionario de cinco preguntas cuya relación el encuestado desconoce (Montiel, y otros, 2016).

Estas tres preguntas se relacionan a través del "Cuadro Lógico de ladov" el cual permite ubicar a cada encuestado, según el cuadro lógico en una escala de satisfacción, para luego calcular el ISG. La escala de satisfacción la cual toma valores de 1 a 6 es la siguiente 1-Clara satisfacción, 2-Más satisfecho que insatisfecho, 3-No definida, 4-Más insatisfecho que satisfecho, 5-Clara insatisfacción y 6-Contradictoria (Montiel, y otros, 2016).

*Tabla 8. Cuadro Lógico de ladov*

*(Fuente: elaboración propia)*

|  |  |              |           |
|--|--|--------------|-----------|
| 5. ¿Qué opina usted acerca de los beneficios que trae consigo la centralización de la gestión de notas entre las diferentes aplicaciones existentes en las | 2. Actualmente las aplicaciones para la gestión de notas que poseen las diferentes variantes de Nova son: Bijiben en Nova y Omni Notes en Nova Droid. ¿Considera, usted que estas aplicaciones satisfacen sus necesidades? |              |           |
|  | <b>No</b>  | <b>No sé</b> | <b>Sí</b> |

|  |   |              |           |           |              |           |           |              |           |
|--|---|--------------|-----------|-----------|--------------|-----------|-----------|--------------|-----------|
| diferentes variantes de Nova para dicho propósito? | 3. ¿Considera usted que es necesario contar con una aplicación informática que permita de forma centralizada la gestión de notas para las diferentes variantes de Nova? |              |           |           |              |           |           |              |           |
|  | <b>Sí</b>   | <b>No sé</b> | <b>No</b> | <b>Sí</b> | <b>No sé</b> | <b>No</b> | <b>Sí</b> | <b>No sé</b> | <b>No</b> |
| Me gusta mucho                                     | 1   | 2            | 6         | 2         | 2            | 6         | 6         | 6            | 6         |
| Me gusta más de lo que me disgusta                 | 2   | 2            | 3         | 2         | 3            | 3         | 6         | 3            | 6         |
| Me da lo mismo                                     | 3   | 3            | 3         | 3         | 3            | 3         | 3         | 3            | 3         |
| Me disgusta más de lo que me gusta                 | 6   | 3            | 6         | 3         | 4            | 4         | 3         | 3            | 4         |
| No me gusta nada                                   | 6   | 6            | 6         | 6         | 4            | 4         | 6         | 6            | 5         |
| No sé decir  | 2   | 3            | 6         | 3         | 3            | 3         | 6         | 6            | 4         |

El número de la interrelación de las tres preguntas indica la posición en la escala de satisfacción siguiente: clara satisfacción (A), más satisfecho que insatisfecho (B), no definida (C), más insatisfecho que satisfecho (D), clara insatisfacción (E) y contradictoria (C).

A partir de la cantidad de respuestas por categoría es posible calcular el Índice de Satisfacción Grupal (ISG) siguiendo la siguiente fórmula:

$$ISG = \frac{A(+1) + B(+0.5) + C(0) + D(-0.5) + E(-1)}{N}$$

Donde N es la cantidad total de respuestas

El valor del ISG permite identificar las siguientes categorías grupales:

- Máxima insatisfacción: desde -1 hasta -0.49
- Más insatisfecho que satisfecho: desde -0.5 hasta -0.1
- No definido y contradictorio: 0
- Más satisfecho que insatisfecho: desde 0.1 hasta 0.49
- Máximo de satisfacción: desde 0.5 hasta 1

El índice general arroja valores entre + 1 y - 1. Los valores que se encuentran comprendidos entre -1 y - 0.5 indican insatisfacción; los comprendidos entre - 0.49 y + 0.49 evidencian contradicción y los que están entre 0.5 y 1 indican que existe satisfacción.

### Resultados obtenidos

En la Tabla 12 se presentan los resultados obtenidos de la aplicación de la encuesta a los especialistas del centro de CESOL.

Tabla 9. Resultados de la escala de satisfacción

(Fuente: elaboración propia)

| <b>Categorías grupales de satisfacción</b> | <b>N = 10</b> | <b>Escala</b> |
|--|---------------|---------------|
| Clara satisfacción                         | 7             | A             |
| Más satisfecho que insatisfecho            | 3             | B             |
| No definido                                | 0             | C             |
| Más insatisfecho que satisfecho            | 0             | D             |
| Máxima insatisfacción                      | 0             | E             |
| Contradictorio                             | 0             | F             |

Cálculo del Índice de Satisfacción Grupal

$$ISG = A (+1) + B (+0.5) / N$$

$$ISG = (7(+1) + 3(+0.5)) / 10$$

$$ISG = 0.85$$

Interpretación del resultado del ISG

El valor obtenido del ISG fue 0.85 lo que indica máxima satisfacción de los usuarios con respecto a la aplicación web para la gestión de notas en Nova 360. Por lo que se puede afirmar que se cumplió con el objetivo de la investigación. Se pudo reafirmar gracias a las respuestas abiertas brindadas por los encuestados los beneficios que traerá consigo la aplicación web para la gestión de notas en Nova 360.

### **3.5 Conclusiones del capítulo**

La informatización de la gestión de notas en Nova 360 se obtuvo mediante la aplicación web desarrollada, para ello se tuvo en cuenta 9 requisitos funcionales y 5 no funcionalidades cumpliendo con las necesidades del cliente, las restricciones del diseño y los estándares de codificación definidos. La aplicación de las pruebas unitarias comprobó que el flujo de trabajo de las funcionalidades es correcto, ya que cada sentencia del código fuente es ejecutada al menos una vez y las pruebas funcionales verificó en dos iteraciones realizadas que la aplicación desarrollada cumple con los requisitos determinados. Se evidenció la alta satisfacción del cliente a través de las pruebas de aceptación y la aplicación de la técnica de Iadov arrojando un ISG de 0.85, propiciando la evaluación satisfactoria de la aplicación web para la gestión de notas en Nova 360.

## **Conclusiones generales**

La investigación realizada cumple con los objetivos planteados mediante la aplicación para la gestión de notas en Nova 360 y se arriba a las siguientes conclusiones:

- El análisis de los referentes teóricos acerca del proceso de gestión de notas para Nova 360, así como el estudio de los 5 sistemas informáticos investigados, demostraron la necesidad de desarrollar una aplicación web que permita la gestión de notas en Nova 360.
- Se diseñó una aplicación web para la gestión de notas en Nova 360 en correspondencia con los 9 requisitos funcionales y 5 no funcionales definidos con el cliente. La utilización de los patrones de diseño de software GRASP e Inyección de dependencias y arquitectónico MVC proporcionaron una mayor calidad al software desarrollado.
- Se desarrolló una aplicación web que permite la gestión de notas en Nova 360; cumpliendo con los requisitos definidos con el cliente.
- La evaluación de la aplicación para la gestión de notas se realizó mediante la aplicación de pruebas de software y la técnica de ladov. Se logró garantizar el correcto funcionamiento de la aplicación y se demostró con un ISG de 0.85 la satisfacción del cliente hacia la solución desarrollada.

## **Recomendaciones**

- Se recomienda incluir el sistema de etiquetas de Nova 360 a la aplicación web de gestión de notas.
- Permitir el envío de notificaciones desde la aplicación de notas a la aplicación móvil de Nova 360.

## Referencias bibliográficas

**Fermín Díaz, José Amado. 2014.** Distribución de un Entorno de Modelado Utilizando Servicios Web. Tesis para optar por el Grado en Ingeniería. [En línea] 05 de 2014. [Citado el: 9 de 11 de 2017.] <http://repositorio.upct.es/bitstream/handle/10317/4375/tfg242.pdf;jsessionid=20BCF79C03757B90B7EC396261E10C66?sequence=1>.

**Abistar. 2017.** Curso práctico de Modelado de Negocios con BPMN y UML. [En línea] 2017. [Citado el: 2 de 11 de 2017.] <http://www.abiztar.com.mx/m/CursoModeladoNegociosBPMN.htm>.

**Aguirre, Fernando y Belloni, Edgardo. 2013.** “Aproximación a los Métodos Formales: una experiencia con un método de desarrollo liviano.”. [En línea] 2013. [Citado el: 13 de 02 de 2018.] [http://sedici.unlp.edu.ar/bitstream/handle/10915/27222/Documento\\_completo.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/27222/Documento_completo.pdf?sequence=1).

**Albornoz, Claudia. 2014.** Diseño de Interfaz Gráfica de Usuario. [En línea] 2014. [Citado el: 21 de 03 de 2018.] [http://sedici.unlp.edu.ar/bitstream/handle/10915/41578/Documento\\_completo.pdf](http://sedici.unlp.edu.ar/bitstream/handle/10915/41578/Documento_completo.pdf).

**Alicante, Universidad de. 2017.** Herramientas en la nube. [En línea] 2017. [Citado el: 2 de 11 de 2017.] [https://moodle2015-16.ua.es/moodle/pluginfile.php/59104/mod\\_resource/content/4/page\\_01..](https://moodle2015-16.ua.es/moodle/pluginfile.php/59104/mod_resource/content/4/page_01..)

**Alvarez, Miguel Angel y Torres, José Miguel. 2015.** Inyección de dependencias Desarrollowe.com. [En línea] 08 de 05 de 2015. [Citado el: 23 de 02 de 2018.] <https://desarrolloweb.com/articulos/patron-diseno-contenedor-dependencias.html>.

**Amoedo, Damian. 2017.** MySQL Workbench, herramienta visual para el diseño de bases de datos. [En línea] 04 de 09 de 2017. [Citado el: 07 de 11 de 2017.] <https://ubunlog.com/mysql-workbench-bases-datos/>.

**—. 2017.** QOwnNotes, sincronización con ownCloud/Nextcloud y generación de notas. [En línea] 11 de 08 de 2017. [Citado el: 07 de 11 de 2017.] [https://ubunlog.com/qownnotesnotasubuntu/?utm\\_source=feedburner&utm\\_medium=%24{feed%2C+email}&utm\\_campaign](https://ubunlog.com/qownnotesnotasubuntu/?utm_source=feedburner&utm_medium=%24{feed%2C+email}&utm_campaign).

**Andalucía, Junta de. 2017.** Validar los requisitos del sistema. [En línea] 2017. [Citado el: 8 de 02 de 2018.] <http://www.juntadeandalucia.es/servicios/madeja/contenido/libro-pautas/185>.

**Anyelguti. 2017.** Aprende Web. *Crea y diseña tus sitios en la Web*. [En línea] 2017. [Citado el: 14 de 11 de 2017.] [http://aprende-web.net/progra/ajax/ajax\\_1.php](http://aprende-web.net/progra/ajax/ajax_1.php).

**Apps, Great. 2017.** alternativeTo. *Quicknotes*. [En línea] 2017. [Citado el: 15 de 11 de 2017.] <https://alternativeto.net/software/quicknotes/>.

**apps.ownCloud. 2017.** Owncloud. [En línea] 2017. [Citado el: 15 de 11 de 2017.] <https://apps.owncloud.com/content/show.php?content=155599>.

**Arias, Jorge Adalberto, Arias Rodriguez, Jose Roberto y Ortiz Aguirre, Ana Marisol. 2012.** Sistema Informático con Georeferencia Espacial en Interfaz Web, para el control de las areas administrativa tecnología agropecuaría y forestal .Tesis para obter por el título de Ingeniero en Sistemas Informáticos. [En línea] mayo de 2012. [Citado el: 13 de 02 de 2018.] <http://ri.ues.edu.sv/1242/1/TESIS.pdf>.

**ARR. 2018.** Estudios de tipo encuesta. [En línea] 2018. [Citado el: 19 de 02 de 2018.] [http://recursos.uco.mx/tesis/estudios\\_tipo\\_encuesta.php](http://recursos.uco.mx/tesis/estudios_tipo_encuesta.php).

**Aviztar. 2017.** Curso práctico de Modelado de Negocios con BPMN y UML. [En línea] 2017. [Citado el: 2017 de 2 de 9.] <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm>.

**Axel Springer España S.A. 2016.** TICbeat. [En línea] 31 de julio de 2016. [Citado el: 23 de 11 de 2017.] <http://www.ticbeat.com/tecnologias/cuantos-tipos-de-cloud-computing-existen-y-en-que-se-diferencian/>.

**BBVA. 2016.** Nimble Payments, una solución de BBVA para startups y pequeños comercios virtuales. [En línea] 9 de 05 de 2016. [Citado el: 09 de 11 de 2017.] <https://bbvaopen4u.com/es/actualidad/metricas-para-decidir-un-cambio-en-una-aplicacion>.

**Bernal, J Chaur. 2018.** Modelo de Diseño Ingeniería del Diseño. [En línea] 2018. [Citado el: 19 de 02 de 2018.] <http://www.tdx.cat/bitstream/handle/10803/6837/05Jcb05de16.pdf>.

**Briones, Darwin Ponce. 2016.** Análisis Comparativo de los Entornos de Desarrollo Integrados (IDE): Eclipse, Netbeans y EL Desarrollo de Aplicaciones. Previa a la obtención del Título de:Ingeniero en Ciencias Computacionales. [En línea] 2016. [Citado el: 03 de 11 de 2017.] [http://repositorio.ug.edu.ec/bitstream/redug/15862/1/B\\_CISC\\_PTG\\_1185.Ponce%20Briones%20Darwin%20Kleber.pdf](http://repositorio.ug.edu.ec/bitstream/redug/15862/1/B_CISC_PTG_1185.Ponce%20Briones%20Darwin%20Kleber.pdf).

**Burgos, Suero, Luciana Estefania. 2017.** Análisis y Evaluación de la arquitectura REST Y SOAP , para el desarrollo de servicios web aplicados al ERP"Adriserp"y su versión móvil en Android. Tesis para optar por el título de Ingeniero en Sistemas. [En línea] 2017. [Citado el: 4 de 04 de 2018.] <http://repositorio.uss.edu.pe/xmlui/handle/uss/4067>.

**Cabrera González , Lianet y Pompa Torres, Enrique Roberto. 2012.** Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información. *Visual*

---

*Paradigm for UML plug-in for Model Driven Development of Information Management Systems*. [En línea] 15 de 10 de 2012. [Citado el: 2 de 11 de 2017.] [https://publicaciones.uci.cu/?journal=SC&page=article&op=viewFile&path\[\]=1032&path\[\]=581](https://publicaciones.uci.cu/?journal=SC&page=article&op=viewFile&path[]=1032&path[]=581).

**Cabrera, Fermín Moreno. 2014.** Herramienta para generación y ejecución automática de pruebas aleatorias de programas en lenguaje C. [En línea] noviembre de 2014. [Citado el: 21 de 03 de 2018.] <https://www.cs.cinvestav.mx/TesisGraduados/2014/TesisFerminMoreno.pdf>.

**Calzada, Aymeé Hernández. 2013.** Sistema de actividades para proporcionar la evaluación formativa en la enseñanza de la física. Tesis para obtener el grado de Doctora en Tecnología Educativa. [En línea] 2013. [Citado el: 27 de 03 de 2018.] <http://www.tdx.cat/bitstream/handle/10803/127225/tahc1de1.pdf>.

**Carvajal, Luís Felipe Ospina. 2016.** El Engagement como herramienta publicitaria para generar relación incondicional entre los consumidores y una marca (estudio de caso). Proyecto de grado para optar al título de Publicista. [En línea] 2016. [Citado el: 26 de 02 de 2018.] <https://red.uao.edu.co/bitstream/10614/8785/1/T06569.pdf>.

**Casta, Boris Paredes. 2015.** "Proceso Metodológico para la mejora continua de la elicitación de requerimientos de software basado en el manejo de requerimientos de CMMI dev V1.3". Tesis de Maestría en Ingeniería en Sistemas de Información. [En línea] 2015. [Citado el: 13 de 02 de 2018.] <http://www.yoamolozapatos.com/?s=zapatos+alis+vizon>.

**ceatic.ujaen.es. 2017.** Almacenamiento en la nube. [En línea] 2017. [Citado el: 6 de 12 de 2017.] <https://ceatic.ujaen.es/es/almacenamiento-en-la-nube>.

**Cevallos, Ingrid Evelyn. 2014.** "Análisis Comparativo de Respaldo y Recuperación de Base de Datos Licenciada (Oracle Utilizando RMAN). Previa a la obtención del Título de: Ingeniero en Sistemas Computacionales. [En línea] 21 de 03 de 2014. [Citado el: 09 de 11 de 2017.] <http://repositorio.cisc.ug.edu.ec/bitstream/123/239/1/Tesis.pdf>.

**Christopher. 2016.** Juntos Somos Solución. *Características y Ventajas del Lenguaje PHP*. [En línea] 19 de 12 de 2016. [Citado el: 10 de 11 de 2017.] <http://www.jlaya.com/caracteristicas-y-ventajas-del-lenguaje-php/>.

**Cibernat. 2017.** Computación en la nube . [En línea] 2017. [Citado el: 19 de 10 de 2017.] <http://cibernat.com/articulos/computacion-en-la-nube>.



---

**CloudSleuth. 2013.** Los 25 mejores proveedores de Cloud Computing ComputerWorld. [En línea] 10 de 09 de 2013. [Citado el: 27 de 02 de 2018.] <http://computerworldmexico.com.mx/Los-25-mejores-proveedores-de-Cloud-Computing/>.

**Codejobs. 2017.** Tecnologías. [En línea] 2017. [Citado el: 2 de 11 de 2017.] <https://www.codejobs.biz/es/blog/2013/12/08/que-es-bpmn>.

**Cueto, A Suárez. 2013.** T01 Concepto de Base de Datos y Relación. [En línea] 2013. [Citado el: 03 de 11 de 2017.] [https://rua.ua.es/dspace/bitstream/10045/28357/1/T01\\_concepto\\_BD\\_y\\_relacion.pdf](https://rua.ua.es/dspace/bitstream/10045/28357/1/T01_concepto_BD_y_relacion.pdf).

**Dayerling, Daniel. 2016.** Servidor Web: Que es y como funciona. [En línea] 29 de marzo de 2016. [Citado el: 14 de 11 de 2017.] <https://www.gmtech.es/que-es-servidor-web/>.

**de la Cruz Londoño, Cristian Andres y Castro Guevarra, Gustavo Andres. 2015.** Requerimientos en el Desarrollo de Software para pequeñas y medianas empresas (PYMES) del Departamento de Risaralda. Tesis presentada como requisito para optar por el título de: Magíster en Ingeniería y Computación. [En línea] 2015. [Citado el: 13 de 02 de 2018.] <http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/5589/0053D278.pdf?sequence=1>.

**Definicion.de. 2017.** Definición de Modelo de Negocio. [En línea] 2017. [Citado el: 22 de 01 de 2018.] <https://definicion.de/modelo-de-negocio/>.

**Garro, Arkaitz. 2015.** Open Libra. *HTML 5*. [En línea] 27 de 07 de 2015. [Citado el: 10 de 11 de 2011.] <https://openlibra.com/es/book/html5>.

**Genbenta. 2017.** Ocho excelentes aplicaciones para tomar notas en Windows y en la Web. [En línea] 2017. [Citado el: 23 de 11 de 2017.] <https://www.genbeta.com/a-fondo/ocho-excelentes-aplicaciones-para-tomar-notas-en-windows-y-en-la-web>.

**Genbeta. 2017.** NetBeans. [En línea] 2017. [Citado el: 03 de 11 de 2017.] <https://www.genbetadev.com/herramientas/netbeans-1>.

**Gesoft. 2015.** Personalización de software. [En línea] 2015. [Citado el: 8 de 12 de 2016.] <http://www.gesoft.com.br/vista/desarrollo/personalizacion-de-software.jsp>.

**—. 2015.** Personalización de software. [En línea] 22 de 10 de 2015. [Citado el: 18 de 10 de 2017.] <http://www.gesoft.com.br/vista/desarrollo/personalizacion-de-software.jsp>.

**Gómez, Javier Combarros. 2015.** Aplicación web y Móvil usando Android y Angular JS contra Spring Web Services+Proyecto de fin Máster en Ingeniería Web. [En línea] 7 de 2015. [Citado el: 4 de 04 de 2018.] [http://oa.upm.es/43117/9/TFM\\_JAVIER\\_COMBARROS\\_GOMEZ.pdf](http://oa.upm.es/43117/9/TFM_JAVIER_COMBARROS_GOMEZ.pdf).

---

**Gómez, Ramiro A. 2017.** Lenguajes de programación. [En línea] 2017. [Citado el: 02 de 11 de 2017.] <http://www.ilustrados.com/documentos/lenguajes-programacion-020608.pdf>.

**González, Largaespada, Daniela María. 2013.** Diseño del Módulo de gestión de expedientes laborales de ingreso y egreso al Sistema de Información y Registro de Personal (SIRCOP) de la policía nacional. Tesis para la obtención del grado de Máster Informática Empresarial. [En línea] 2013. [Citado el: 28 de 03 de 2018.] <http://ribuni.uni.edu.ni/847/1/38059.pdf>.

**Guerra, César Arturo. 2017.** Obtención de Requerimientos. Técnicas y Estrategia. [En línea] 2017. [Citado el: 22 de 01 de 2018.] <https://sg.com.mx/revista/17/obtencion-requerimientos-tecnicas-y-estrategia>.

**Idrovo Tapia, Cristian Iván y Pacheco Campoverde, Lucy Gabriela. 2014.** "Desarrollo de una aplicación móvil en android de soporte para la recaída de pacientes en proceso de recuperación del hospital psiquiátrico Humberto Ugalde Camacho". Tesis para la obtención del título de Ingeniero en Sistema. [En línea] 2014. [Citado el: 08 de 02 de 2018.] <https://dspace.ups.edu.ec/bitstream/123456789/6294/1/UPS-CT002853.pdf>.

**Inc, CBS Interactive. 2017.** ownNote - Notes for ownCloud for Android. [En línea] 2017. [Citado el: 08 de 11 de 2017.] [http://descargar.cnet.com/ownNote-Notes-for-ownCloud/3000-20418\\_4-77376439.html](http://descargar.cnet.com/ownNote-Notes-for-ownCloud/3000-20418_4-77376439.html).

**Iruela, Juan. 2016.** Los gestores de bases de datos más usados. [En línea] 19 de 01 de 2016. [Citado el: 03 de 11 de 2017.] <https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/>.

**J, General. 2017.** Definición de Javascript. [En línea] 0 de 10 de 2017. [Citado el: 13 de 11 de 2017.] <http://conceptodefinicion.de/javascript/>.

**Latinoamérica, Red Gráfica. 2017.** El lenguaje de programación PHP. [En línea] 2017. [Citado el: 10 de 11 de 2017.] <http://redgrafica.com/El-lenguaje-de-programacion-PHP>.

**Leva, Maria de las Mercedes. 2013.** "Cloud Computing: Escenarios Actuales y Tendencias". [En línea] 2013. [Citado el: 17 de 10 de 2017.] [http://tesis.blanque.com.ar/tesis/Home\\_files/Tesis\\_Mercedes\\_Leva.pdf](http://tesis.blanque.com.ar/tesis/Home_files/Tesis_Mercedes_Leva.pdf).

**Leva, María de las Mercedes. 2013.** "CLOUD COMPUTING: Escenarios Actuales y Tendencias". [En línea] 2013. [Citado el: 18 de 10 de 2017.] [http://tesis.blanque.com.ar/tesis/Home\\_files/Tesis\\_Mercedes\\_Leva.pdf](http://tesis.blanque.com.ar/tesis/Home_files/Tesis_Mercedes_Leva.pdf).

---

**Lucidchart. 2018.** Qué es un modelo de base de datos . [En línea] 2018. [Citado el: 19 de 02 de 2018.] [//www.lucidchart.com/pages/es/qué-es-un-modelo-de-base-de-datos](http://www.lucidchart.com/pages/es/qué-es-un-modelo-de-base-de-datos).

**MADEJA. 2018.** Especificación de Requisitos del Sistema. Marco de Desarrollo de la Junta de Andalucía (MADEJA). [En línea] 2018. [Citado el: 26 de 01 de 2018.] <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/407>.

**MAJEDA . 2018.** Aplicación del patrón MVC en PHP Marco de Desarrollo de la Junta de Andalucía. [En línea] 2018. [Citado el: 19 de 02 de 2018.] <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/257>.

**Marcos, Bartolomé Sintés. 2017.** Guía de estilo para PHP. *Programación web en PHP* . [En línea] 1 de 11 de 2017. [Citado el: 28 de 03 de 2018.] <http://www.mclibre.org/consultar/php/otros/guia-estilo.html>.

**Martínez, Evelio Martínez. 2017.** Big Data:¿Cómo procesar volúmenes de información? [En línea] 2017. [Citado el: 18 de 10 de 2017.] <http://www.eveliux.com/mx/Big-Data-como-procesar-grandes-volumenes-de-informacion.html>.

**MaxCDN. 2017.** ¿Qué es Apache? [En línea] 2017. [Citado el: 14 de 11 de 2017.] <http://culturacion.com/que-es-apache/>.

**Mdn. 2017.** Tecnologías web. [En línea] 21 de 09 de 2017. [Citado el: 13 de 11 de 2017.] [https://developer.mozilla.org/es/docs/Learn/Getting\\_started\\_with\\_the\\_web/JavaScript\\_basics](https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/JavaScript_basics).

**Montiel, Torrado, Jordan Abdul, González Castro, Yoandry y Carbonell Tamayo, Arianna. 2016.** Análisis y Diseño de un Sistema para La Gestión de la Información de los Procedimientos del Servicio de Hemodinámica. [En línea] 2016. [Citado el: 27 de 03 de 2018.] <http://www.informaticahabana.cu/sites/default/files/ponencias/SLD09.pdf>.

**Pérez Jiménez, Surelys G y Puldón, Jaime Joan. 2012.** [En línea] 2012. [Citado el: 22 de 01 de 2018.] <http://www.redalyc.org/pdf/3604/360433568009.pdf>.

**Perez, Deyrin y Fuentes Puentes, Carlos. 2014.** Aplicación informática para la gestión de los entornos de pruebas del Laboratorio Industrial de Pruebas de Software. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. [En línea] junio de 2014. [Citado el: 21 de 03 de 2018.] [https://repositorio\\_institucional.uci.cu/jspui/bitstream/ident/9237/2/TD\\_07524\\_14.pdf](https://repositorio_institucional.uci.cu/jspui/bitstream/ident/9237/2/TD_07524_14.pdf).

**Pérez, Julian y Gardey, Ana. 2012.** Definición de nota. [En línea] 2012. [Citado el: 10 de 11 de 2011.] <https://definicion.de/nota/>.

**PMOinformática.com. 2017.** La oficina de proyectos de informática. [En línea] 2017. [Citado el: 26 de 01 de 2018.] <http://www.pmoinformatica.com/2015/05/requerimientos-no-funcionales-ejemplos.html>.

**POMinformática.com. 2017.** La oficina de proyectos de informática. [En línea] 2017. [Citado el: 26 de 01 de 2018.] <http://www.pmoinformatica.com/2015/05/requerimientos-no-funcionales-ejemplos.html>.

**Portable, Loren. 2017.** Los Portables para tu Pc. *QOwnNotes v17.06.6 Build 3071 Español Portable*. [En línea] 5 de junio de 2017. [Citado el: 07 de noviembre de 2017.] <http://www.muchosportables.com/qownnotes-v17-06-6-build-3071-espanol-portable/>.

**Pressman, R.S. 2010.** *Software engineering: a practitioner's approach. 7th ed.* New York : EUA, 2010. ISBN:978-0-07-337597.

**Pressman, Roger S. 2010.** Ingeniería del software un enfoque práctico Séptima edición. [En línea] 2010. [Citado el: 14 de 02 de 2018.] <http://cotana.informatica.edu.bo/downloads/Id-Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF>.

**Ramírez, José. 2017.** Novedades en Note Station, el Evernote de tu NAS Synology. [En línea] 2017. [Citado el: 6 de 12 de 2017.] <https://www.jmramirez.pro/articulo/note-station/>.

**Ramos, Ernesto Hitas. 2015.** mplementación de la herramienta SIRNA, orientada a la extracción interactiva de patrones a partir de imágenes digitales para NEUROLAB. [En línea] 2015. [Citado el: 23 de 02 de 2018.] [https://repositorio\\_institucional.uci.cu/jspui/bitstream/1234567897145/1/TD\\_07939\\_15.pdf](https://repositorio_institucional.uci.cu/jspui/bitstream/1234567897145/1/TD_07939_15.pdf).

**Ramos, Galo y Páez, Jaime. 2012.** *Añalisis del Método para la Clasificación del Software QSOS para la Selección de Software Aplicables a Proyectos Educativos*. s.l. : Universidad Tecnológica Equinoccial, 2012.

**Redrejo, José Luis. 2012.** Metodología GTD para el profesorado organizado. [En línea] 2012. [Citado el: 30 de 10 de 2017.] <http://recursostic.educacion.es/observatorio/web/fr/software/software-general/1078-evernote-metodologia-gtd-para-el-profesorado-organ>.

**Rodríguez, Tamara. 2015.** Metodología de desarrollo para la Actividad Productiva de la Uci. [En línea] 2015. [Citado el: 11 de 10 de 2017.] [excriba.prod.uci.cu/page/context/shared/document-details?](http://excriba.prod.uci.cu/page/context/shared/document-details?).

**Rodríguez, Tamara. 2015.** *Metodología de desarrollo para la actividad productiva de la UCI*. Universidad de las Ciencias Informáticas. La Habana. Cuba : s.n., 2015.

---

**Rosillo, Pablo. 2017.** Omni Notes, crea y gestiona tus notas en esta interesante aplicación de código abierto. [En línea] 2017. [Citado el: 21 de 11 de 2017.] <https://andro4all.com/2014/01/omni-notes-crear-gestionar-notas-aplicacion-codigo-abierto>.

**Sáez, Francisco. 2013.** FacileThings. *Nueva opción: Notas*. [En línea] 25 de julio de 2013. [Citado el: 10 de 11 de 2017.] <https://facilethings.com/blog/es/notes>.

**Sánchez, Antonio José Gallo. 2012.** Memoria de Sistemas Informáticos. [En línea] 2012. [Citado el: 18 de 10 de 2017.] <http://eprints.ucm.es/13329/1/memoriaFinal.pdf>.

**Sogeti. 2017.** En 2020 el volumen de información será de 35,2 ZB. [En línea] 2017. [Citado el: 25 de 10 de 2017.] <http://www.networkworld.es/big-data/en-2020-el-volumen-de-informacion-sera-de-352-zb>.

**Sparxsystems. 2017.** Sparxsystems. *Diagrama de Componentes UML 2*. [En línea] 2017. [Citado el: 22 de 03 de 2017.] [http://www.sparxsystems.com.ar/resources/tutorial/uml2\\_componentdiagram.html](http://www.sparxsystems.com.ar/resources/tutorial/uml2_componentdiagram.html).

**Staurdo Lucho, Romero. 2012.** Diseño e Implementación de un Sistema de Administración de Calendarios Online con Sincronización Móvil+Tesis para optar el Título de Ingeniero de las Telecomunicaciones. [En línea] 17 de 02 de 2012. [Citado el: 03 de 11 de 2017.] [http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/1504/LUCHO\\_ROMERO\\_CESAR\\_CALENDARIOS\\_ONLINE.pdf;jsessionid=140F9B](http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/1504/LUCHO_ROMERO_CESAR_CALENDARIOS_ONLINE.pdf;jsessionid=140F9B).

**UML . 2018.** Diagramas de Clase Lenguaje de Modelado Unificado. [En línea] 2018. [Citado el: 26 de 02 de 2018.] [http://stadium.unad.edu.co/ovas/10596\\_9836/diagramas\\_de\\_clase.html](http://stadium.unad.edu.co/ovas/10596_9836/diagramas_de_clase.html).

**Vasquez, Daniel Villanueva. 2015.** Diseño de un modelo conceptual multi-dominio para recomendaciones mediante el filtrado de información semántica. Doctorado en Ciencia y Tecnología Informática Departamento de Informática. [En línea] 09 de 2015. [Citado el: 22 de 01 de 2018.] [https://e-archivo.uc3m.es/bitstream/handle/10016/22134/tesis\\_daniel\\_villanueva\\_vasquez\\_2015.pdf](https://e-archivo.uc3m.es/bitstream/handle/10016/22134/tesis_daniel_villanueva_vasquez_2015.pdf).

**visual\_paradigm.com. 2017.** Visual Paradigm. [En línea] 2017. <https://www.visual-paradigm.com/aboutus/>.

**wordpress.com. 2017.** INGTI. [En línea] 2017. [Citado el: 22 de 01 de 2018.] <https://ingti.wordpress.com/2011/09/27/definicion-de-proceso-de-negocio/>.

**Zurdo, Dayana y González Izarga, Jorge Luis. 2014.** Componente para la Digitalización de Contornos de Estructura Geológicas. Trabajo de Diploma Para Optar por el Título de Ingeniero en Ciencias Informáticas. [En línea] junio de 2014. [Citado el: 19 de 03 de 2018.] [https://repositorio\\_institucional.uci.cu/jspui/bitstream/ident/9119/2/TD\\_07756\\_14.pdf](https://repositorio_institucional.uci.cu/jspui/bitstream/ident/9119/2/TD_07756_14.pdf).

---

**Anexos****Anexo 1: Entrevista realizada a los especialistas del equipo de desarrollo del proyecto Nova 360 de CESOL**

**Objetivo:** conocer el contexto del proyecto Nova 360, las herramientas y tecnologías compatibles con el entorno de desarrollo de dicho proyecto, así como los sistemas que se integran con Nova 360 y las características que debe tener una aplicación para ser instalada en dicho software.

1. ¿Cuáles son los sistemas que se integran con Nova 360? ¿Qué características deben tener estos sistemas?
2. ¿Qué características o requisitos debe tener una aplicación para ser integrada en Nova 360?
3. ¿Cree usted que es importante el uso de una aplicación que permita la gestión de notas en Nova 360? ¿Por qué?
4. ¿Qué características debe tener esta aplicación?

## **Anexo 2: Guía de observación para verificar la gestión de notas en aplicaciones informáticas basadas en owncloud, que permiten la sincronización con clientes externos**

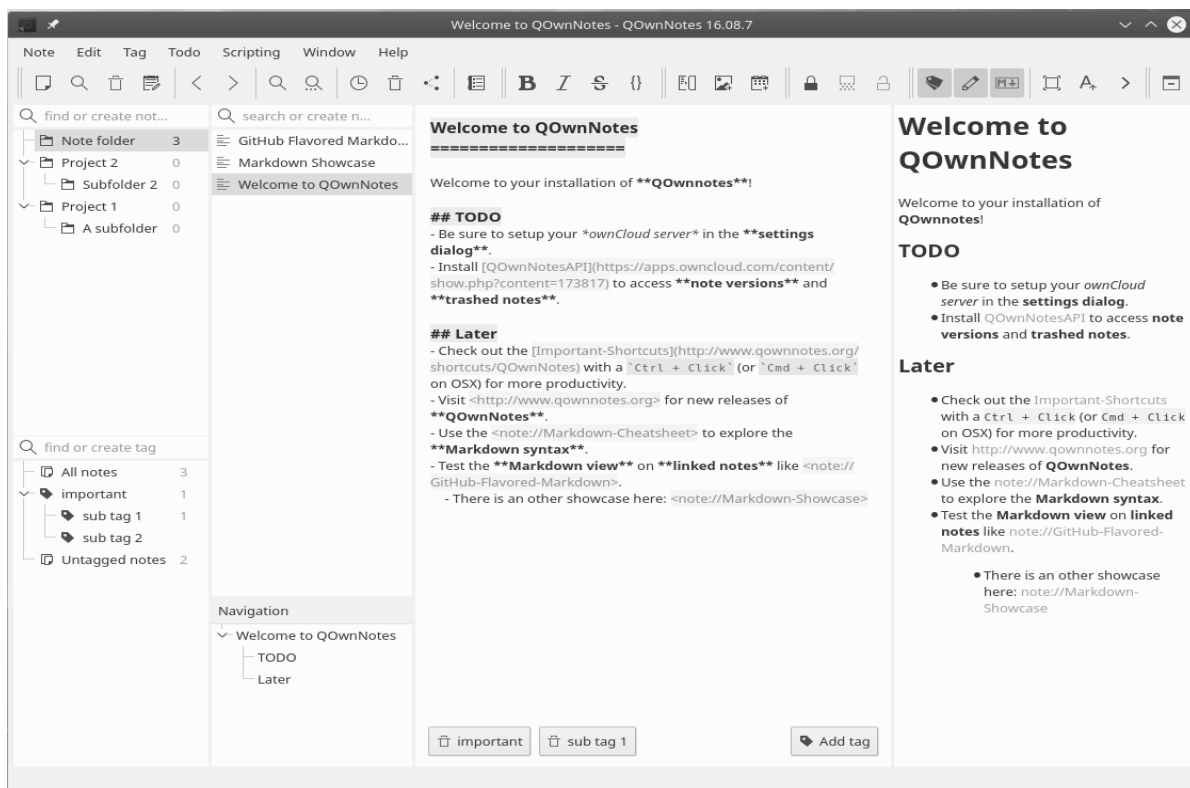
**Observadora:** Aideé Mora Urdaneta

**Lugar:** Laboratorio de práctica profesional de CESOL

**Objetivo:** Identificar los elementos fundamentales para la gestión de notas en aplicaciones informáticas basadas en *Owncloud*, que permiten la sincronización con clientes externos.

1. Datos de las aplicaciones informáticas que permiten la gestión de notas
  - Nombre de las aplicaciones
  - Lenguaje de programación de las aplicaciones
  - Compatibilidad con diferentes sistemas operativos
2. Características de las aplicaciones informáticas que permiten la gestión de notas
  - ¿Cómo son las aplicaciones informáticas que permiten la gestión de notas?
  - ¿Qué herramientas necesitan para ser instaladas?
  - ¿Cuáles son los pasos a seguir para instalar estas herramientas?
3. Impacto social de las aplicaciones informáticas que permiten la gestión de notas

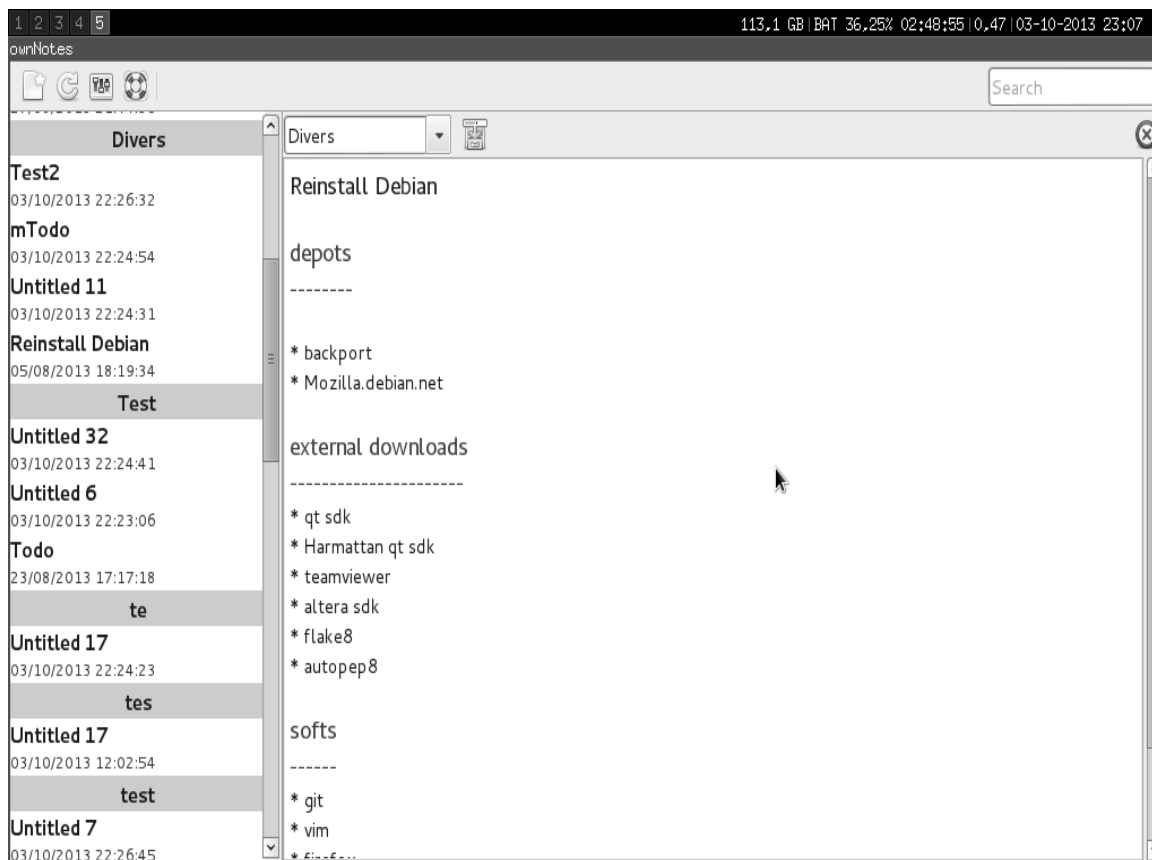
## Anexo 3: QownNotes



Interfaz gráfica de usuario de la aplicación QownNotes



## Anexo 4: OwnNote



Interfaz gráfica de usuario de la aplicación *OwnNote*

## Anexo 5: Notes

The screenshot shows the OwnCloud interface with the 'Notes' application selected in the sidebar. The main content area is divided into two columns. The left column lists several notes, with the 'Introduction' note selected. The right column displays the content of the selected note, including a title field, a content text area, and a hint about using Markdown. At the bottom, there are 'Delete' and 'Save' buttons.

**OwnCloud**

🏠 Dateien

🎵 Musik

📅 Kalender

👤 Kontakte

✉ Webmail

📝 Notes

📖 Journal

🖼 Bilder

📖 Lesezeichen

📧 RoundCubeMail

**Another note**  
December 2, 2012, 17:38

**General Notes**  
December 2, 2012, 17:39

**Introduction**  
December 2, 2012, 17:38

**Shopping list**  
December 2, 2012, 17:38

**Useful links**  
December 2, 2012, 17:38

**Introduction:**  
Notes is a simple OwnCloud plugin which can show notes.  
The notes are simple text files stored in the Folder **Notes**.  
Copyright 2012 by George Ruinelli  
--  
To format the notes, you can use the *Markdown* language.

---

**Title:**  
Introduction

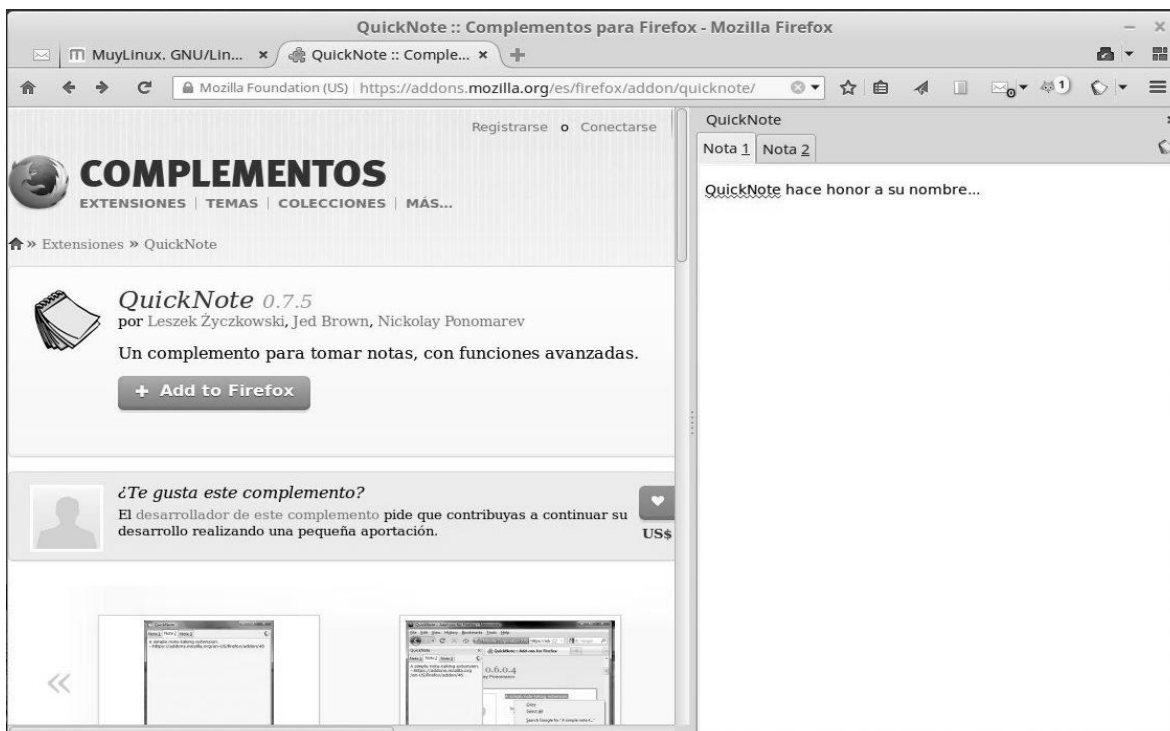
**Content:**  
\*\*Notes\*\* is a simple OwnCloud plugin which can show notes.  
The notes are simple text files stored in the Folder **\*\*Notes\*\***.  
Copyright 2012 by George Ruinelli  
--  
To format the notes, you can use the *Markdown* language.

**Hint:** You can use **Markdown** to format your text.

Delete

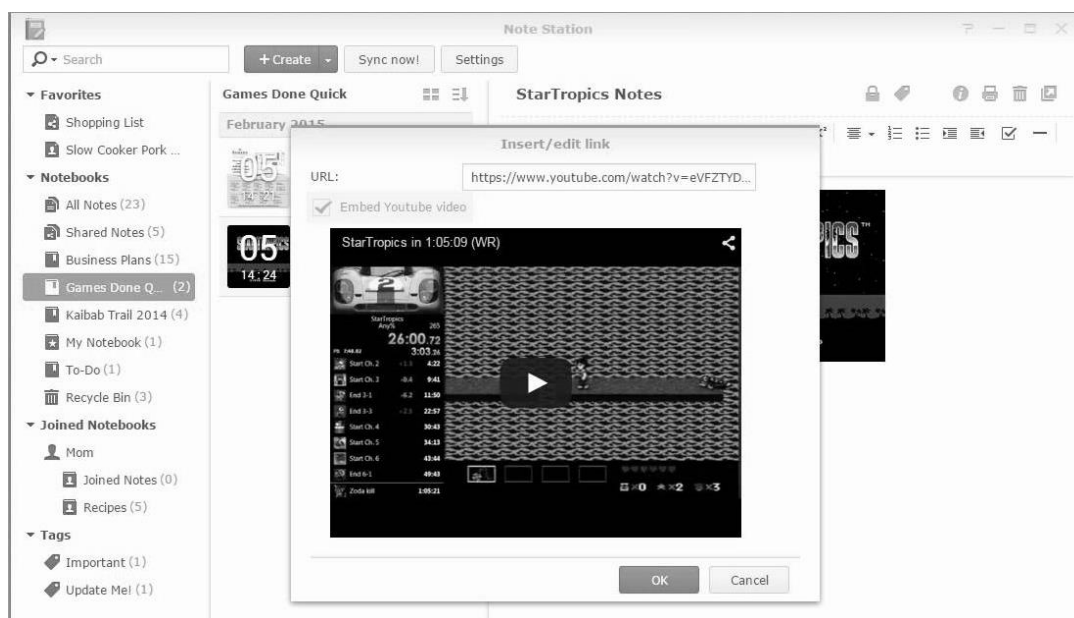
Interfaz gráfica de usuario de la aplicación *Notes*

## Anexo 6: Quicknote



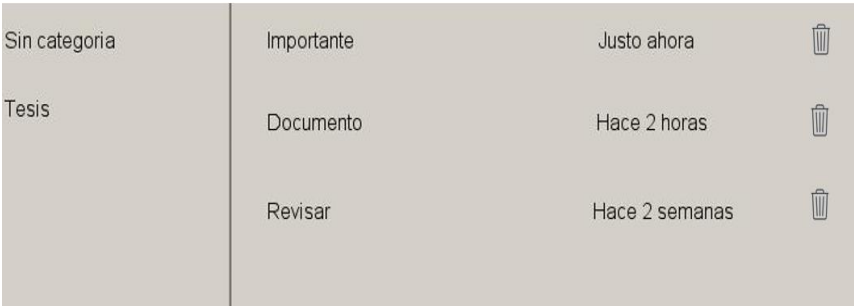
Interfaz gráfica de usuario de la aplicación *Quicknote*

## Anexo 7: Note Station

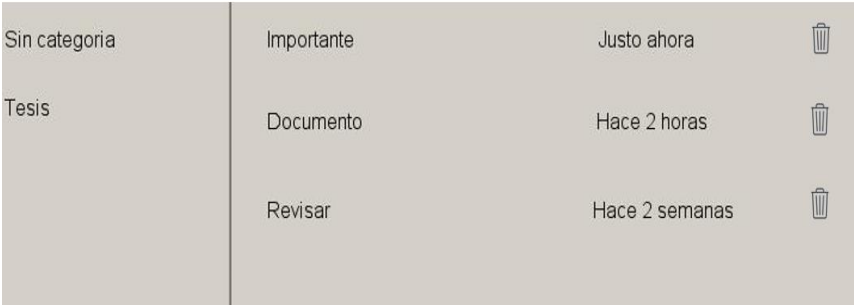


Interfaz gráfica de usuario de la aplicación *Note Station*


**Anexo 8 : Historia de usuario del RF 3. Eliminar una nota**

|   |  |  |  |
|---|--|--|--|
| <b>Número:</b> HU_3   | <b>Nombre del requisito:</b> Eliminar una nota |  |  |
| <b>Programadora:</b> Aideé Mora Urdaneta  | <b>Iteración Asignada:</b> 1                   |  |  |
| <b>Prioridad:</b> Baja  | <b>Tiempo Estimado:</b> 20 horas               |  |  |
| <b>Riesgo en Desarrollo:</b> no aplica  | <b>Tiempo Real:</b> 15 horas                   |  |  |
| <p><b>Descripción:</b> La aplicación web permite eliminar una nota.</p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la nota que desea eliminar.</li> <li>2. El usuario selecciona la opción eliminar.</li> <li>3. El usuario confirma la acción.</li> </ol>   |  |  |  |
| <b>Observaciones:</b> Las notas se eliminan correctamente.  |  |  |  |
| <b>Prototipo elemental de interfaz gráfica de usuario:</b>  |  |  |  |
|  <p>The screenshot shows a list of notes in a table-like format. The first row has 'Sin categoria' in the first column, 'Importante' in the second, 'Justo ahora' in the third, and a trash icon in the fourth. The second row has 'Tesis' in the first column, 'Documento' in the second, 'Hace 2 horas' in the third, and a trash icon in the fourth. The third row has 'Revisar' in the second column, 'Hace 2 semanas' in the third, and a trash icon in the fourth. The first column is partially obscured by a grey bar.</p> |  |  |  |

**Anexo 9 : Historia de usuario del RF 6. Buscar notas por categoría**

|   |   |  |  |
|---|---|--|--|
| <b>Número:</b> HU_6   | <b>Nombre del requisito:</b> Buscar notas por categoría |  |  |
| <b>Programadora:</b> Aideé Mora Urdaneta  | <b>Iteración Asignada:</b> 1                            |  |  |
| <b>Prioridad:</b> Media   | <b>Tiempo Estimado:</b> 20 horas                        |  |  |
| <b>Riesgo en Desarrollo:</b> no aplica  | <b>Tiempo Real:</b> 15 horas                            |  |  |
| <p><b>Descripción:</b> La aplicación web permite buscar las notas según su categoría.</p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la categoría en el menú lateral izquierdo.</li> <li>2. Se listan todas las notas de esa categoría.</li> </ol>   |   |  |  |
| <b>Observaciones:</b> Puede existir una nota sin categoría.   |   |  |  |
| <b>Prototipo elemental de interfaz gráfica de usuario:</b>  |   |  |  |
|  <p>The screenshot shows a list of notes in a table-like format. The first column lists categories: 'Sin categoría' and 'Tesis'. The second column lists note titles: 'Importante', 'Documento', and 'Revisar'. The third column shows timestamps: 'Justo ahora', 'Hace 2 horas', and 'Hace 2 semanas'. Each row has a trash icon on the right side.</p> |   |  |  |

**Anexo 10** : Historia de usuario del RF 7: Buscar notas por contenido de texto

|  |  |  |
|--|--|--|
| <b>Número:</b> HU_7  | <b>Nombre del requisito:</b> Buscar notas por contenido de texto |  |
| <b>Programadora:</b> Aideé Mora Urdaneta   | <b>Iteración Asignada:</b> 1                                     |  |
| <b>Prioridad:</b> Alta   | <b>Tiempo Estimado:</b> 20 horas                                 |  |
| <b>Riesgo en Desarrollo:</b> no aplica   | <b>Tiempo Real:</b> 15 horas                                     |  |
| <p><b>Descripción:</b> La aplicación web permite buscar notas por contenido de texto.</p> <ol style="list-style-type: none"> <li>1. El usuario escribe el contenido que desea buscar en las notas.</li> <li>2. Presiona la tecla ENTER.</li> </ol> |  |  |
| <p><b>Observaciones:</b> Para buscar el contenido de una nota se tiene en cuenta la coincidencia de minúscula con mayúscula.</p>   |  |  |
| <p><b>Prototipo elemental de interfaz gráfica de usuario:</b></p> <div style="text-align: center;">  </div>   |  |  |

---

## **Anexo 11: Encuesta realizada a los especialistas del equipo de desarrollo del proyecto Nova 360 de CESOL**

**Objetivo:** identificar la información a procesar en el negocio informatizado, la obtención de requisitos y las restricciones de diseño existentes en Nova 360.

Especialista, le invitó a responder el siguiente cuestionario con el fin de conseguir su colaboración en la presente investigación, solicitó que exprese en sus repuestas criterios verídicos que guíen a la autora del trabajo. Muchas gracias, por el tiempo brindado.

1. ¿Qué información considera usted, que debe tener una nota?
2. ¿Qué acciones le gustaría realizar sobre una nota?
3. ¿Considera usted que es necesario agrupar las notas de acuerdo a criterios definidos por el usuario?

Sí\_\_\_No\_\_\_

4. ¿Por qué?
5. ¿Considera factible algún mecanismo de búsqueda para las notas?

Sí\_\_\_No\_\_\_

6. ¿Por qué?
7. ¿Cómo le gustaría que se realizara ese proceso de búsqueda?
8. ¿Considera importante que la aplicación para la gestión de notas sincronice con clientes externos?

Sí\_\_\_No\_\_\_

9. ¿Por qué?



## Anexo 12: Acta de aceptación de productos de trabajo



### Acta de aceptación de productos de trabajo

#### ACTA DE ACEPTACIÓN DE PRODUCTOS DE TRABAJO

En cumplimiento del **Convenio de colaboración** establecido entre el **Centro de Software Libre (CESOL)** y la estudiante **Aideé Mora Urdaneta** de la Facultad 1 de la Universidad de las Ciencias Informáticas y en función de la ejecución del proyecto: **Aplicación web para la gestión de Notas en Nova 360**, se hace entrega del producto que se relaciona a continuación:

- Aplicación web para la gestión de Notas en Nova 360

La parte Cliente, luego de haber revisado el producto de trabajo relacionado anteriormente procede a firmar la aceptación de los mismos en total conformidad.

| Entrega   | Recibe   |
|---|--|
| <b>Nombre y apellidos:</b> Aideé Mora Urdaneta  | <b>Nombre y apellidos:</b> Yoandy Pérez Villazón   |
| <b>Cargo:</b> Estudiante Facultad 1   | <b>Cargo:</b> Director de CESOL  |
| <b>Firma:</b>  | <b>Firma:</b>  |
|   |                |

**Fecha:** 08/05/2018

## **Anexo 13: Encuesta a especialistas del Centro de Software Libre (CESOL) de la Universidad de las Ciencias Informáticas**

**Objetivo:** evaluar la satisfacción de los usuarios potenciales hacia la solución desarrollada.

Especialista, le invitó a responder el siguiente cuestionario con el fin de conseguir su colaboración en la presente investigación, solicitó que exprese en sus repuestas criterios verídicos que guíen a la autora del trabajo. Marque en cada pregunta con una X en una sola opción y en el caso de la 5 responda brevemente. Muchas gracias, por el tiempo brindado.

1. ¿Considera usted que es necesario el uso de aplicaciones que permitan la gestión de notas contenidas en las diferentes variantes de Nova en la universidad?

Sí \_\_\_ No \_\_\_ No sé \_\_\_

2. Actualmente las aplicaciones para la gestión de notas que poseen las diferentes variantes de Nova son: Bijiben en Nova y Omni Notes en Nova Droid. ¿Considera, usted que estas aplicaciones satisfacen sus necesidades?

Sí \_\_\_ No \_\_\_ No sé \_\_\_

3. ¿Considera usted que es necesario contar con una aplicación informática que permita de forma centralizada la gestión de notas para las diferentes variantes de Nova?

Sí \_\_\_ No \_\_\_ No sé \_\_\_

4. Luego de haber visto la aplicación web para la gestión de notas en Nova 360 refleje en qué medida le gusta la solución desarrollada.

|  |  |
|--|--|
| ___ Me gusta mucho                     | ___ Me disgusta más de lo que me gusta |
| ___ Me gusta más de lo que me disgusta | ___ No me gusta nada                   |
| ___ Me da lo mismo                     | ___ No sé decir                        |

5. ¿Qué opina usted acerca de los beneficios que trae consigo la centralización de la gestión de notas entre las diferentes aplicaciones existentes en las diferentes variantes de Nova para dicho propósito?