Revista Internacional de Investigación e Innovación Tecnológica

Página principal: www.riiit.com.mx

Diseño e implementación de un sistema informático de apoyo para la generación de horarios de docentes en Instituciones de Educación Superior

Design and implementation of an informatic support system for the generation of teachers schedules in Higher Education Institutions

Ramos-Romero, P., Mendoza-Rodríguez, L.A, Vivanco-Benavides, L.E.

Av. 16 de septiembre # 54, C.P. 55700, Col. Cabecera municipal, Coacalco de Berriozábal, Estado de México.
pris.ram.rom@gmail.com; chiva-luis23@live.com

Innovación tecnológica: Agilizar procesos administrativos por medio de sistemas informáticos en Instituciones de Educación Superior.

Área de aplicación industrial: Administración académica, Administración de personal.

Recibido: 18 junio 2018.
Aceptado: 31 Octubre 2018.

Abstract

This article deals with the development and implementation of a computer support system for the generation of teacher schedules of the Technological Studies of Coacalco (TESCo), emphasizing the methodology of Rational Unified Process (RUP) and Unified Modeling Language (UML), which complement and help address software development, as well as being considered as important methodologies for the development of software projects due to their approach of adaptability and the importance that these give to the quality of the final system. The implementation of computer security is also very important, where the primary resource that represents information must be protected, considering the important items of computer security: integrity, confidentiality and availability.

Key words: Code, Encryption, Informatic security, RUP methodology, Software test.

Resumen

En el presente artículo se aborda el desarrollo y la implementación de un sistema informático de apoyo para la generación de horarios de docentes del Tecnológico de Estudios Superiores de Coacalco (TESCo), haciendo énfasis en la metodología de Proceso Racional Unificado (RUP) y en el lenguaje unificado de modelado (UML), los cuales se complementan y ayudan a abordar el desarrollo de software, además de ser considerados como metodologías importantes en la elaboración de proyectos de software por su enfoque a la adaptabilidad y la importancia que estas le dan a la calidad del sistema final. También es de suma importancia la implementación de la seguridad informática en donde se debe proteger el recurso primordial que es la información, considerando los rubros importantes de la seguridad informática: la integridad, confidencialidad y disponibilidad.

Palabras clave: Cifrado, Encriptación, Metodología RUP, Pruebas de software, Seguridad informática.

Introducción

Si bien las instituciones educativas a nivel superior deben contar con sistemas informáticos eficaces y eficientes para registrar, procesar, almacenar, recuperar y presentar información sobre sus operaciones y actividades, debe tenerse en cuenta que el hacerlo conllevará una gran responsabilidad con la información con la que se trabaje, ya que esta es esencial y un recurso clave para la posteridad. Desarrollar este tipo de sistemas para instituciones de educación superior ofrecerá múltiples posibilidades, permitiendo acceder a los datos relevantes de manera frecuente y oportuna. Un sistema informático adecuado, ofrece una importante y notable satisfacción en los usuarios que lo operan, debido a la facilidad de uso y su acceso constante, por lo tanto al diseñarlo como apoyo para la generación de horarios de los docentes, puede lograrse el reducir tiempo en esta actividad, y que los horarios estén preparados antes de los tiempos estipulados, además, ayudará a minimizar errores pues es bien sabido que llega a existir en diversos casos la redundancia de la información sobre los profesores u otra información que se maneja. Por otra parte, se debe considerar la implementación de la seguridad informática en el sistema de información que es un pilar fundamental en el desarrollo de sistemas de información.

Justificación

La importancia de implementar un sistema informático que genere horarios de los profesores del Tecnológico de Estudios Superiores de Coacalco debe garantizar que al hacerlo sea de manera apropiada, pues hasta la fecha ningún sistema previamente desarrollado apoya a la correcta realización de este proceso; debe tenerse en cuenta que lo anterior ayudará en prevenir la redundancia en la información almacenada, es decir, el no repetir datos como el nombre de materias y sus ID, salones y profesores que impartirán asignaturas. Al lograr complementar esta actividad el tiempo en la generación de horarios será menor, pues estarán preparados para los periodos de reinscripción y no se afectará la impartición de clases a los alumnos.

1. Seguridad informática

La seguridad informática es la disciplina que, con base a las políticas y normas internas y externas de una institución, se encarga en proteger la integridad y la privacidad de la información que se encuentra almacenada en un sistema informático contra cualquier tipo de amenazas, minimizando los riesgos a los que está expuesto. Se ocupa en diseñar las normas, métodos y técnicas destinados al diseño de un sistema de información y sistemas informáticos. En el sistema de información se debe conocer los elementos que componen el sistema, los peligros que lo afectan, accidentales o provocados, medidas que deberían adoptarse para conocer, prevenir, impedir y reducir los riesgos potenciales [1].

Es posible hacer cumplir la seguridad, bajo la jurisdicción del administrador de la base de datos, pues bien, se puede asegurar que el único medio de acceso a esta sea con la clave de cifrado correcta y por lo tanto se debe definir las reglas o restricciones de seguridad, como ejemplo de ellas:

- Tipos de usuarios, son aquellos usuarios dados de alta en el sistema, aquí estarán involucrados dos, el "administrador" que es aquel que tiene todos los privilegios dentro del sistema informático y un segundo usuario "jefe de carrera", que serán todos aquellos usuarios con la facultad de acceder al sistema dependiendo a la carrera de su interés y realizar todas las operaciones a excepción del borrado de datos, ya que este es un permiso único para el administrador y ayuda a tener un control sobre la información que ya se tiene almacenada.
- Elección de contraseñas para usuarios y control de accesos, ya que estas son las herramientas más utilizadas para restringir el acceso a los sistemas informáticos y se debe saber que solo son efectivas cuando se seleccionan con cuidado, las contraseñas se aplican en todos los logueos que son necesarios para ingresar al sistema, si un usuario que no precisamente es el "administrador" quiere modificar o eliminar un dato en el sistema informático, el sistema volverá a hacer un logueo pidiendo contraseña y usuario para verificar que únicamente el administrador sea el que puede realizar esta acción, de lo contrario cualquier otro usuario tendrá el permiso denegado para esta operación.

Muchos sistemas utilizan contraseñas para hacer que los usuarios solo puedan acceder a los ficheros relacionados con su trabajo [2].

1.1 Cifrado de información

"El cifrado es una operación criptográfica reversible que transforma datos significativos sin proteger, conocidos como texto sin formato, en datos ilegibles, cifrados, conocido como texto cifrado, utilizando una clave llamada clave de cifrado" [3]. En la figura 1 se muestra el rendimiento esperado de una clave cifrada.

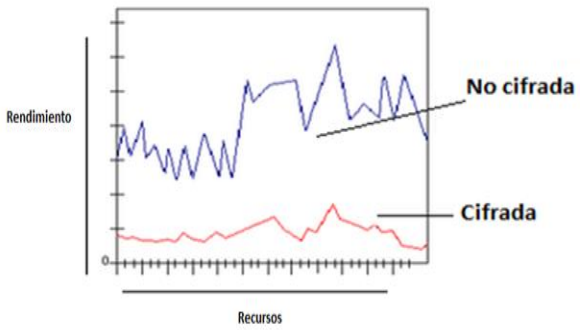

Figura 1. Rendimiento esperado en las bases de datos cifradas. Fuente: (Murillo, 2018).

Constituye uno de los métodos de protección más fiable, ya que consiste en la transformación de los datos, de forma que una persona que no deba tener acceso a ellos, es

decir, no sea capaz de entenderlos, se realiza a través de elementos lógicos o físicos [4].
Un cifrado simétrico también conocido como "Shared Key" o "Shared Secret" es aquel que utiliza la misma clave para cifrar y descifrar cierta información. Es importante que la clave sea difícil de averiguar [3]. En la figura 2 se muestra el ejemplo del cifrado simétrico.

Figura 2. Cifrado simétrico. Fuente: (Bauer, 2001, pág. 250).

A los datos originales sin cifrado se les conoce como "texto plano", en este proyecto serán todos aquellos que se ingresan en el sistema informático y que se visualizan claros para cualquier usuario, una vez que este es sometido a un algoritmo de cifrado se le conoce como "texto cifrado", para este caso el texto cifrado será la información que se visualiza en la base de datos pues al ser enviado para almacenarse el algoritmo de cifrado actúa y cifra la información insertada a la base de datos, logrando así que nadie pueda reconocer campos, alterándolos y haciendo irreconocible la información, esto ayuda a cualquier ataque ajeno y protegiendo la información, los detalles de cifrado son públicos, o al menos no están ocultos especialmente; pero la clave de cifrado se mantiene en secreto. El texto cifrado que debe ser ininteligible para cualquiera que no posea la "clave de cifrado" es lo que se guarda en la base de datos o se transmite por la línea de comunicación [2].

1.2 Implementación del cifrado

En base al cifrado por sustitución se realizó la encriptación de la información, este cifrado es aquel que sustituye cada letra o grupo de letras por otra letra o grupo de letras distinta/s para cifrar el texto en claro.

Los primeros y antiguos métodos de cifrado se basaban en este principio, y hasta la fecha son de gran utilidad, pues muchos de estos cifrados que actualmente se utilizan partieron de este.

Como ejemplo más común tenemos el cifrado del César que es aquel en donde cada letra del texto en claro se sustituye por la letra que hay a "k" posiciones detrás de ella en el alfabeto.

Matemáticamente hablando [4]:

- **Ordinal de una letra:** Es la posición que ocupa la letra en el alfabeto que usamos. Se denota como: ORD (Letra) [ORD (A) = 0].
- **Carácter de un número:** Es la letra que ocupa la posición del número que tenemos. Se denota como: CAR (x) [donde "x" tiene que ser mayor o igual a cero, y menor o igual al número de letras del alfabeto usado].

Dónde:
Cifrado: $C(x) = x + k \pmod T$
Descifrado: $D(x) = x - k \pmod T$
(Siendo "x" el ORD (Letra a cifrar), "k" es el número de posiciones a desplazar y "T" es el total de letras en el alfabeto a usar). Se puede observar que la clave de este cifrado es el número de posiciones a desplazar las letras (k) [5].

Para realizar el cifrado de la información en el sistema informático, se genera una matriz de 9x5 en IDE Netbeans 8.1. Esta matriz se compone por un arreglo bidimensional. Para realizarla se declaró una cadena donde se almacena los datos y dependiendo de la

posición es la letra, número o carácter correspondiente, como se observa en la figura 3.

```
private String encri
{
   String cad=cadena.
   String matriz[][]=
   matriz[0][0]="A";
   matriz[0][1]="B";
   matriz[0][2]="C";
   matriz[0][3]="D";
   matriz[0][4]="E";
   matriz[1][0]="F";
   matriz[1][1]="G";
   matriz[1][2]="H";
   matriz[1][3]="I";
   matriz[1][4]="J";
   matriz[2][0]="K";
   matriz[2][1]="L";
   matriz[2][2]="M";
   matriz[2][3]="N";
   matriz[2][4]="Ñ";
   matriz[3][0]="O";
   matriz[3][1]="P";
   matriz[3][2]="Q";
   matriz[3][3]="R";
   matriz[3][4]="S";
   matriz[4][0]="T";
   matriz[4][1]="U";
```

Figura 3. Creación de matriz. Fuente: Elaboración propia.

Al haber aplicado el cifrado por sustitución en la información se puede visualizar que los datos se vuelven irreconocibles para cualquier usuario, esto como se muestra en la figura 4, donde se observa un panorama general de la base de datos.

id_profesor	nombre_profesor	apellido_paterno	apellido_materno	telefono	calle	correo
52	333002133072	330023110421	332213330451	6161525354606162	60	3330023400
53	1441211330	51003300	331342043334	615252535362545461	70	1441211330
54	100421133104	42002304110034	2200334013230451	5261616061525261	31330434130304400434	10042113310
60	3304502330	31302130	0200042100	5361536060637064	313304341303042340043 4	3130213074
62	14410023	120433230023030451	0204333023	616152525361	120433300434	1441002374

Figura 4. Base de datos cifrada. Fuente: Elaboración propia.

El reforzar la seguridad en las bases de datos desde un inicio ayudará a hacer cumplir el principio de la integridad refiriendo a la completitud, exactitud y coherencia del conjunto de datos, además de proteger la confidencialidad y disponibilidad en la misma.

Comprendiendo que la confidencialidad se trata del aspecto más importante de seguridad de base de datos, y que se alcanza a través de la encriptación o los datos en tránsito; por otra parte, la integridad busca garantizar que solo el personal autorizado podrá acceder a la

información y, por último, pero no menos importante la disponibilidad que hace referencia a que la información esté lista para usarse cuando se requiera [5].

2. Metodología RUP

La construcción de software es un proceso fundamental ya que en el, se integran diversas técnicas, como lo es la codificación, validación y las pruebas. Sin embargo, existen etapas en el desarrollo de software esenciales como lo es la planeación, análisis de requerimientos, el diseño y las pruebas. Es por ello, que el definir una metodología ayudará a asumir las diversas actividades hasta llegar a un producto final.

Por lo tanto, al elegir la metodología RUP para la elaboración de este proyecto se logrará satisfacer las necesidades de los usuarios en un límite de tiempo propuesto y con presupuesto previsible, además de que es una metodología de desarrollo iterativo, enfocada hacia los diagramas de casos de uso, en la figura 5 se muestran las fases de la metodología RUP.

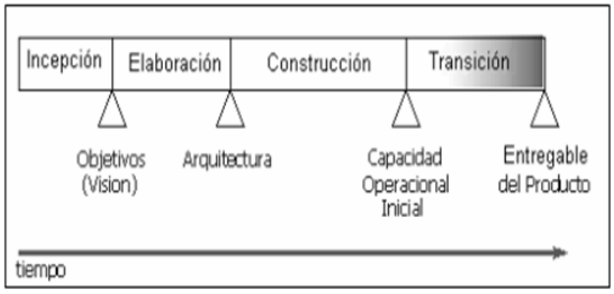

Figura 5. Fases de la Metodología RUP. Fuente: (Administración de Proyectos, 2010, p.26).

Como se aprecia en la figura 6, se hace énfasis en el caso de uso general del proyecto definiendo los módulos y procesos del sistema de información desarrollado.

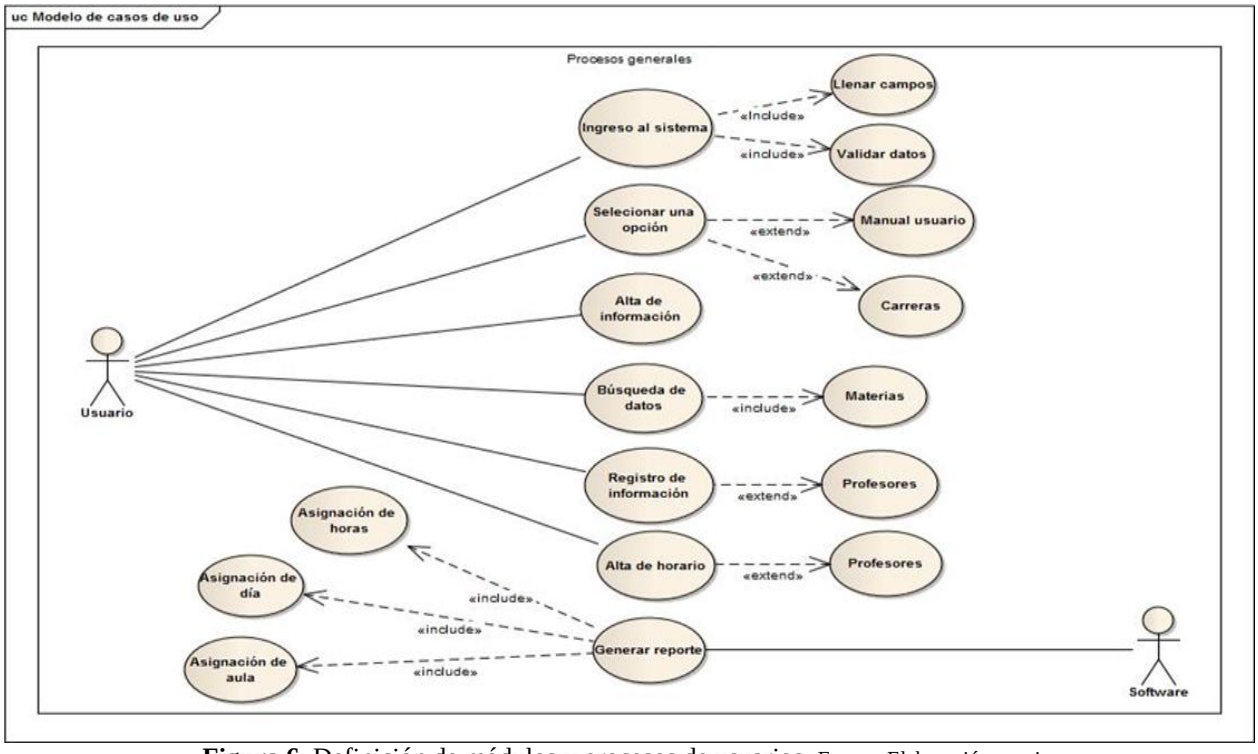

Figura 6. Definición de módulos y procesos de usuarios. Fuente: Elaboración propia.

En este caso de uso, se contemplan todos los procesos con los que cuenta el sistema informático, desde el ingreso al sistema, búsqueda de datos, alta o baja de docentes y asignaturas hasta la generación final del horario.

Al ser una metodología por su estructura dinámica, se muestran inmersas cuatro fases:

- Fase de inicio/incepción o concepción, donde se prepara y evalúa el desarrollo a seguir, planteando objetivos, así como analizando posibles riesgos al alcanzar esos objetivos.
- Fase elaboración/diseño, se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollan en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis de dominio del problema, se diseña una solución preliminar.
- Fase de desarrollo/implementación o construcción, se complementa la funcionalidad del sistema, clasificando requerimientos pendientes de acuerdo a las evaluaciones por los usuarios, se realizan mejoras.
- Fase de cierre/transición, se asegura que el software esté disponible para usuarios finales, ajustando errores y defectos encontrados, capacitar a usuarios y proveer el soporte técnico necesario, el producto debe cumplir con las especificaciones entregadas por el personal involucrado [6].

2.1 Modelo de Negocios

El modelo de negocios, es aquel modelado que ayuda a evaluar los aspectos de la viabilidad en la implementación y el desarrollo del sistema de información incluyendo la descripción y las perspectivas, es decir, cómo se va a hacer, quién va a ser el público objetivo. Esta técnica consta de las siguientes tareas [7]:

- Definir los procesos.
- Analizar los procesos actuales.

En la figura 7, se muestra el modelo de negocio con todos los procesos que conforman el sistema de información.

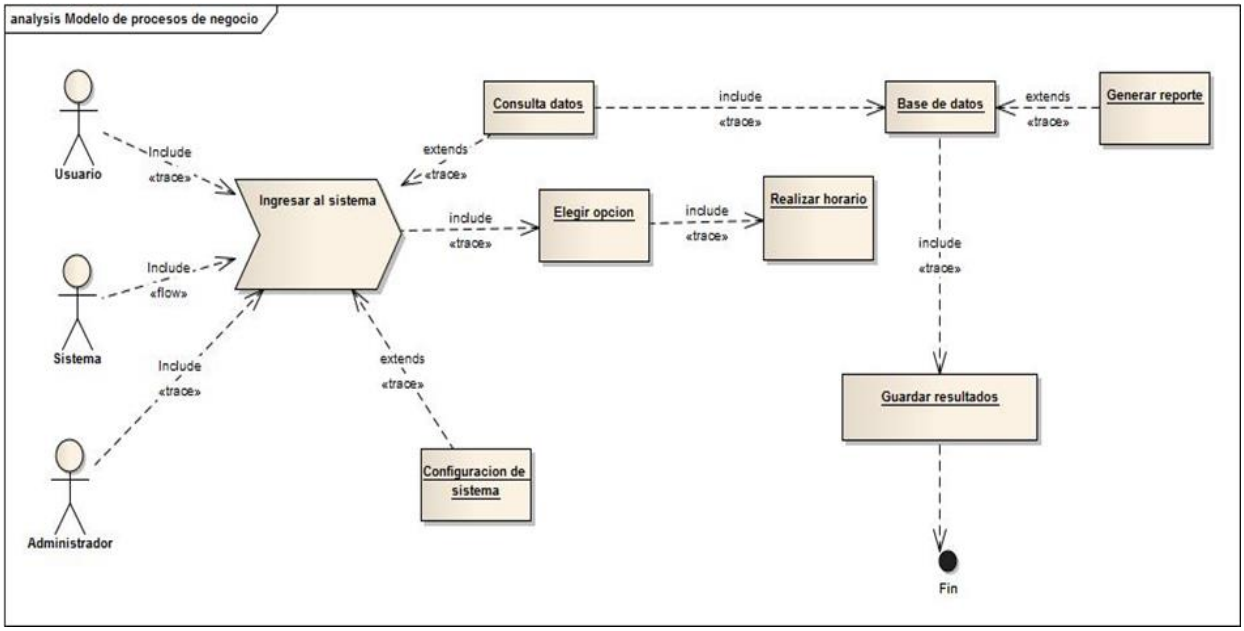

Figura 7. Modelo de Negocios del sistema informático. Fuente: Elaboración propia.

Al realizar el modelo de negocios, se cubren todos los aspectos básicos de un negocio, desde los segmentos de clientes hasta incluso los socios claves y la estructura de costos [8].

2.2 Implementación de los Casos de uso en la metodología RUP

Ya que la metodología RUP, se enfoca hacia los diagramas de casos de uso, es importante saber que son un tipo de requerimientos utilizados para especificar funcionalidad, es decir, los intercambios entre el sistema informático y las personas que interactúan con él, ayudando a capturar requerimientos, definiendo arquitectura, estableciendo pautas para el diseño y las pruebas funcionales.

El primer caso de uso es aquel que define los procesos con los que cuenta el sistema de información, contemplando todos los módulos y los usuarios que interactuarán con cada uno de ellos.

Además de este caso de uso, se contemplan algunos otros como son, el ingreso al sistema por parte de los usuarios, la selección de carreras para dar de alta a los profesores o las asignaturas de cada carrera, y como proceso fundamental del sistema informático la generación y asignación de horarios a los docentes, en la figura 8 Se muestra la representación general del sistema informático.

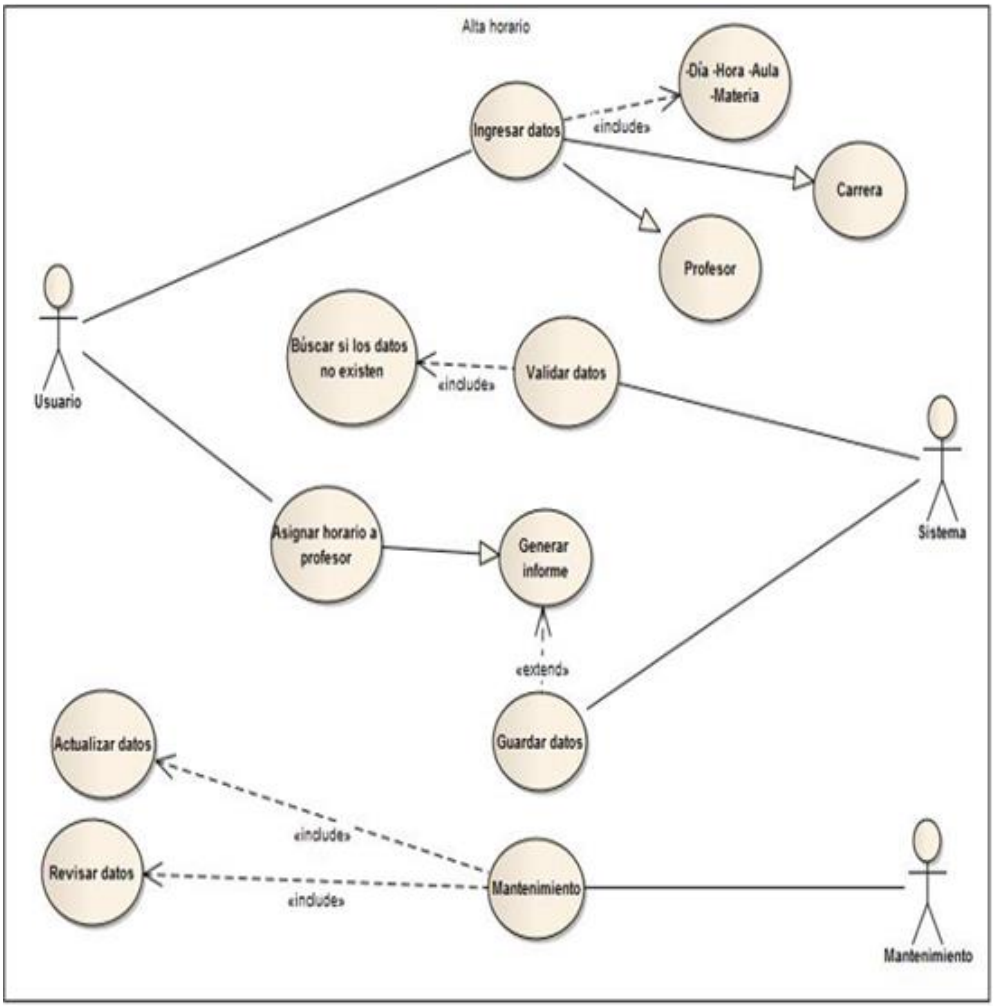

Figura 8. Caso de uso general del sistema informático. Fuente: elaboración propia.

2.3 Modelo de Dominio

El modelo de dominio es aquel que puede ser tomado como el punto de partida para el diseño del sistema informático, en donde todos los usuarios y clientes deberían de reconocer todos los conceptos, se puede desarrollar una terminología común al razonar sobre los casos de uso y así disminuir malentendidos entre los clientes y los desarrolladores [8].

Para el diseño y la implementación de este sistema, se muestra el modelo de dominio del software a realizar, donde se observan los diferentes atributos de cada tabla, así como sus funciones que dependen a su vez de tablas interrelacionadas que se comunican entre sí. Este modelo proporciona al cliente una perspectiva conceptual donde se contemplan tres rubros importantes como lo son:

- Objetos de dominio o clases conceptuales, aquí se encuentran todos aquellos procesos en el proyecto que serán indispensables para la correcta ejecución del sistema informático, estos tienen una identidad y son distinguibles.
- Asociaciones entre clases, es aquella relación que existe entre un proceso y otro, ya que todos dependen entre sí y cuando no existe un proceso correcto puede crear conflictos en los que dependan de él, una clase describe un grupo de objetos con las mismas propiedades comportamientos y relaciones posibles.
- Atributos de las clases conceptuales, son todas aquellas características que conforman a un proceso [9]. En la figura 9 se muestra el diagrama de dominio del sistema informático.

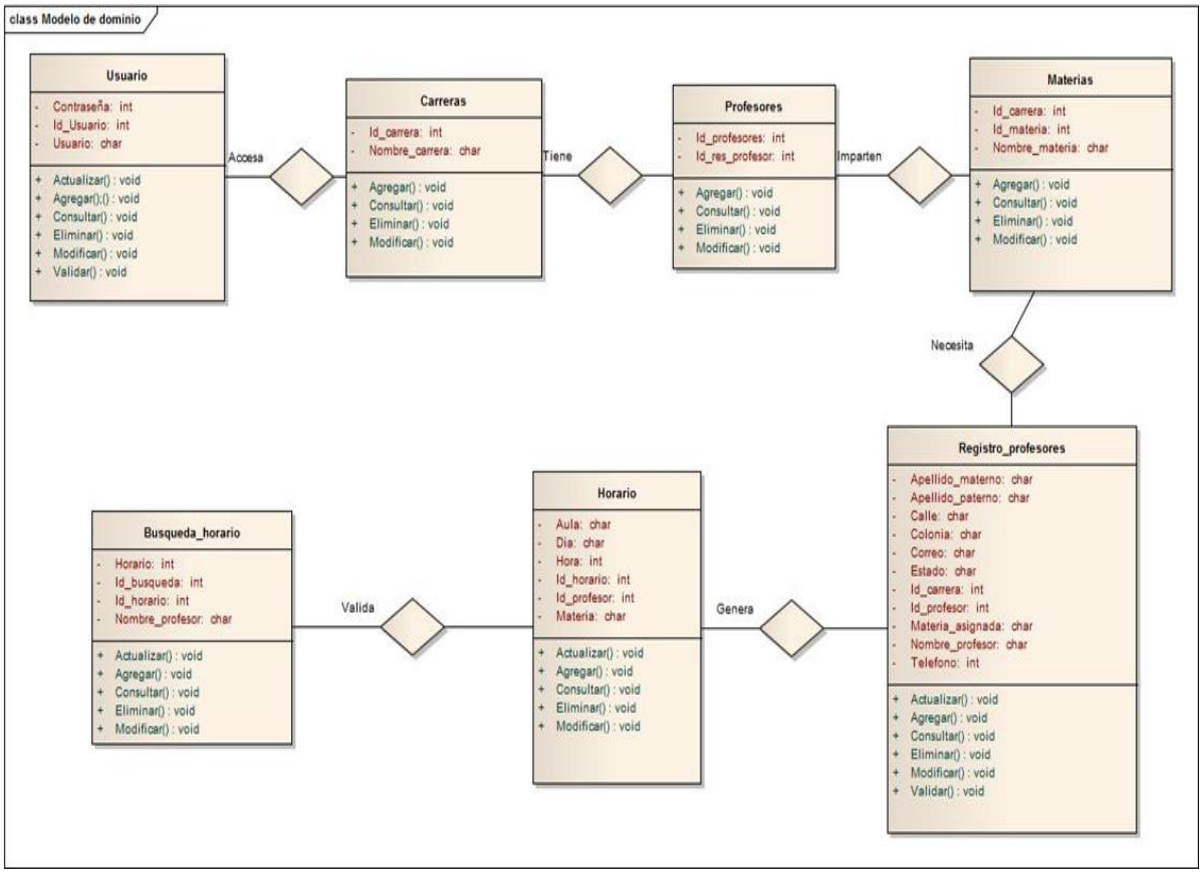

Figura 9. Diagrama de dominio. Fuente: Elaboración propia.

3. Descripción de Operaciones en el sistema informático

La descripción de las operaciones permite conocer de manera completa aquellas partes o procedimientos que conforman el sistema informático, ayudando a abordar la problemática de naturaleza real y permitiendo así encontrar una solución fácil y oportuna [10].

Dentro de las operaciones a realizar por el sistema informático se encuentran:

3.1 Operación de autenticación, en este proceso se verifica que un usuario que busca acceder al sistema sea aquel que tiene los permisos para acceder a esos recursos. Debe existir una condición previa, que en este caso será el escribir el usuario y la contraseña, donde se valida que la información proporcionada sea correcta y si es así, se permite el acceso al sistema. Lo anterior se muestra en la figura 10.

Figura 10. Operación autenticación. Fuente: Elaboración propia.

3.2 Operación búsqueda de asignaturas, es aquí donde una vez que se ha permitido el acceso a un usuario pueda hacer la búsqueda de las asignaturas. Esta búsqueda se realiza a través del código identificador de cada asignatura ya que, aunque en algunas carreras existan materias similares deben estar diferenciadas por su ID, esto las hará independientes entre ellas, el código de cada materia es ingresado a la barra de búsqueda y este puede estar constituido por letras y números. Se recomienda contar con una tira de materias que contengan los códigos para evitar errores y así colocar correctamente el ID de la asignatura, realizando con éxito la búsqueda. A continuación, se representa esta operación en la figura 11.

Figura 11. Operación búsqueda de materia. Fuente: Elaboración propia.

3.3 Operación alta de docentes, es aquella en donde se permite ingresar a cada uno de los docentes para ser almacenados en la base de datos. Esta operación se puede hacer de manera manual, es decir, un usuario llena todos los campos solicitados con la información, o bien puede llenarse de manera automática con la ayuda de un archivo Excel xsl o xslx. En este archivo estarán contenidos todos los campos necesarios para realizar la correcta alta de los docentes y una vez que la información fue verificada se procede a almacenarla en la base de datos, como se observa en la siguiente figura 12.

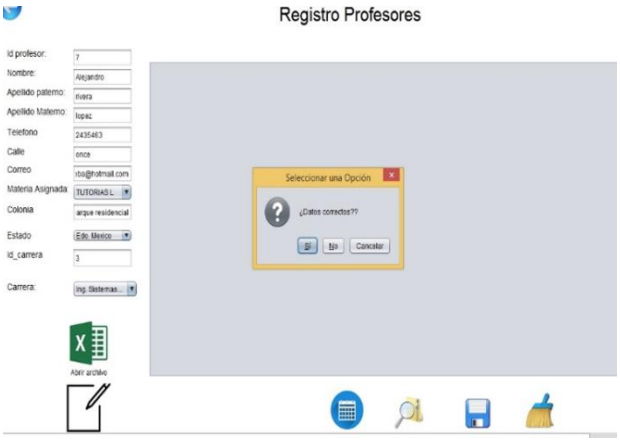

Figura 12. Operación autenticación. Fuente: Elaboración propia.

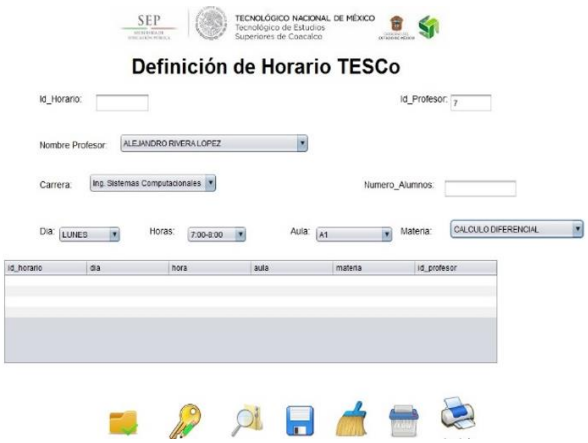

Figura 13. Operación autenticación. Fuente: Elaboración propia.

3.4 Operación definición de horarios a docentes, en ella se procede a asignarle cada horario a los maestros que fueron dados de alta en el proceso anterior, seleccionando al docente, a la carrera asociada con las asignaturas específicamente de cada carrera, así como el día, la hora y el aula. Este proceso es el más importante, pues es aquí donde se verifica que no exista la redundancia en la información, ya que un docente no puede tener mismos valores como pueden ser, día, hora, aula, entre otros. La sentencia implicada en este proceso busca reducir al mínimo la redundancia de los datos pues al encontrar valores repetidos tomando como punto de partida los campos mencionados evitando la duplicidad en los datos, haciendo más fácil y rápido este proceso dentro de la Institución educativa, en la figura 13 se muestra el proceso de la definición de horarios a docentes.

3.5 Operación generación de reportes de docentes en Excel, una vez definido el horario de cada docente y corroborando que la información es correcta, se puede generar un reporte en formato xsl o xslx, esto no es más que un documento en Excel con el membretado de la Institución, el nombre de cada docente, las asignaturas que impartirá y sus horarios, como puede verse en la figura 14.

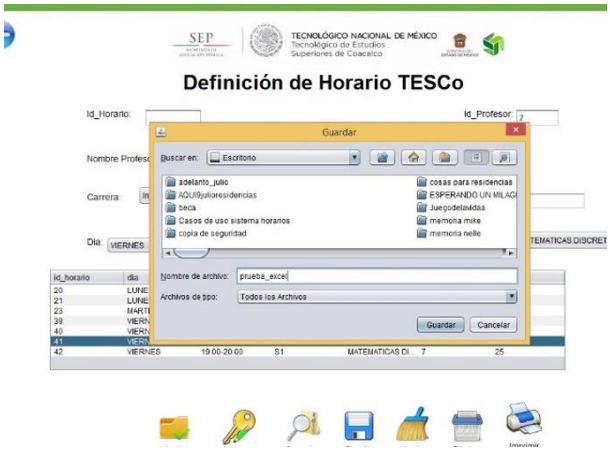

Figura 14. Operación generación de reportes de docentes en Excel. Fuente: Elaboración propia.

4. Pruebas de software

Las pruebas de software consisten en la verificación del comportamiento de un sistema informático en un finito de casos de prueba

contra comportamientos esperados y se realizan con el propósito de encontrar fallos de implementación, calidad o usabilidad. Dentro de los objetivos que se plantean en estas pruebas, se encuentran los siguientes [11]:

- Detectar defectos en el software.
- Verificar la integración adecuada de los componentes.
- Verificar que todos los requisitos se han implementado correctamente.
- Identificar los defectos que se han encontrado antes de que el proyecto sea entregado al cliente.

4.1 Principios de las Pruebas de software

Los principios de las pruebas de software pueden ser considerados como las siguientes [11]:

- La prueba puede ser usada para mostrar la presencia de errores, pero nunca su ausencia.
- Definir los resultados esperados en la realización de las pruebas.
- El número de errores descubiertos es proporcional al número de errores no descubiertos.

En el proyecto desarrollado se generaron las pruebas de software durante las etapas del desarrollo del sistema informático, por ejemplo, en la fase de inicio se especifican las pruebas de validación que consisten en comprobar que el software desarrollado satisface las expectativas razonables del cliente, como son el comportamiento y el rendimiento especificados durante el análisis.

Durante la fase de diseño, se especifican las pruebas de integración, donde se captura cada módulo del sistema y a su vez, se comprueba que globalmente funciona. En esta fase también se especifican las pruebas de unidad de cada módulo.

Al finalizar la codificación de cada proceso se realiza la prueba de unidad, donde se comprueba que cada módulo funciona correctamente, es decir, que todos los procesos se ejecutan de acuerdo a lo especificado. Dentro de las pruebas de unidad se encuentran las de caja blanca, caja negra y caja gris, en la figura 15 se muestra el diagrama del proceso del ciclo de pruebas de software.

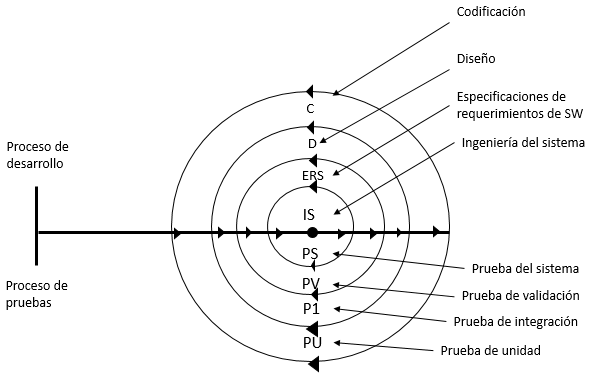

Figura 15. Proceso de desarrollo y pruebas. Fuente: (Martínez F. A., 2005, pág. 89).

En otras palabras, las pruebas de unidad validan la eficacia funcional de cada módulo del sistema informático, las de integración validan la eficiencia funcional y las de validación comprueban que todos los requerimientos han sido incorporados.

Normalmente ambas pruebas se mezclan entre validando la interfaz de software y a su vez asegurando el funcionamiento interno del software de manera correcta.

4.2 Pruebas de caja blanca

Las pruebas de caja blanca son aquellas que comprueban aquellos caminos lógicos del software, tomando trozos específicos, como lo son bucles, sentencias de bifurcación entre otros [12].

Para la implementación de las pruebas de caja blanca se utilizó la herramienta Junit que es un

framework java que permite la ejecución de clases de manera controlada para poder comprobar que los métodos del sistema informático realizan su cometido de forma correcta. También sirve como herramienta para realizar las pruebas de regresión, que se implementan cuando una parte del código ha sido modificada y es necesario comprobar que se sigue cumpliendo con todos los requisitos [13].

Las pruebas de caja blanca tipo unitarias con ayuda de la herramienta Junit versión 4.x, donde para crear la clase Test, se tiene que dar clic derecho en el Jframe que se pretende validar y posteriormente se escoge la opción Tools - Create/Update Test, aparecerá una ventana de confirmación y se dará ok. La herramienta nos creará una clase con el nombre del jframe que se escogió agregando la palabra Test, tal y como se muestra en la figura 16.

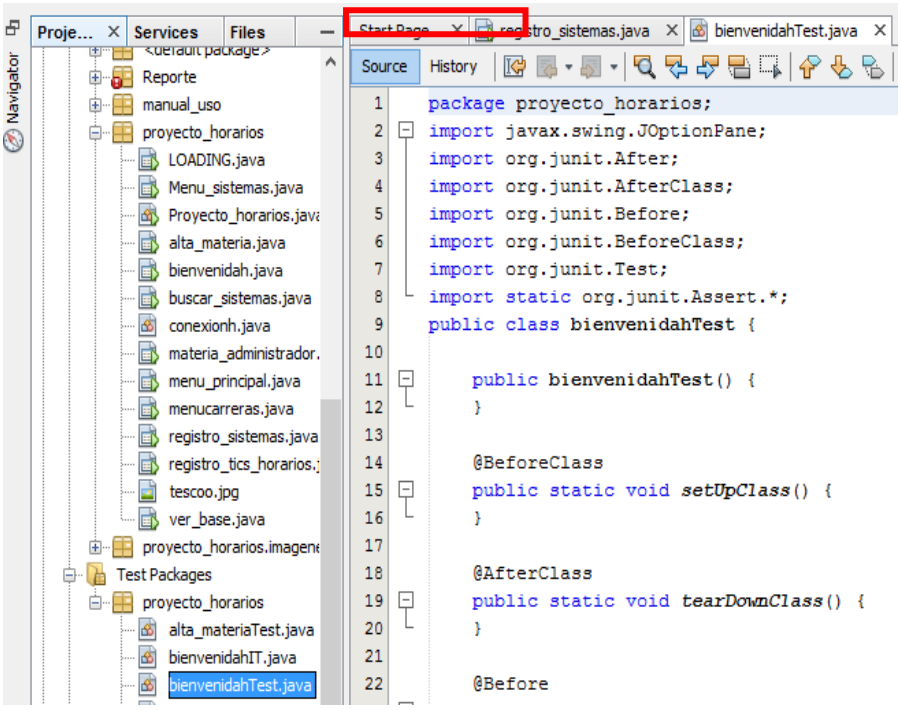

Figura 16. Creación Test. Fuente: Elaboración propia.

Creada la prueba, se observó que contiene bloques de los métodos creados en la clase original ya que es donde se hace alguna validación con la base de datos, en este caso el proceso de autenticación. El bloque creado por la herramienta tiene dos mensajes: uno en caso de que se ejecute correctamente y otro en caso de fallo. Estos dos mensajes se tienen que añadir dentro de una estructura de control que permite determinar una cierta acción a tomar si el usuario se autentica correctamente o incorrectamente y es encontrado como un usuario perteneciente al sistema informático, permite su acceso. La estructura de control en cuestión puede encontrarse en la figura 17.

```
@Test
public void testMain() {
    System.out.println("administrador");
    String[] args = null;
    bienvenidah.main(args);
    int encontrado=0;
    // TODO review the generated test code and remove the default ca
    if (encontrado == 1) {
    JOptionPane.showMessageDialog(null, "Correcto acceso","Correcto

    fail("The test case is a prototype.");
}
}
}
```

Figura 17. Estructura de control. Fuente: Elaboración propia.

Después de crear la sentencia se ejecuta la clase para conocer la forma en que trabaja internamente este proceso, tal y como se muestra en las figuras 18 y 19.

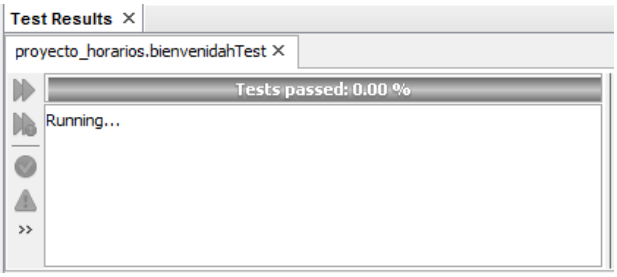

Figura 18. Cargando Test. Fuente: Elaboración propia.

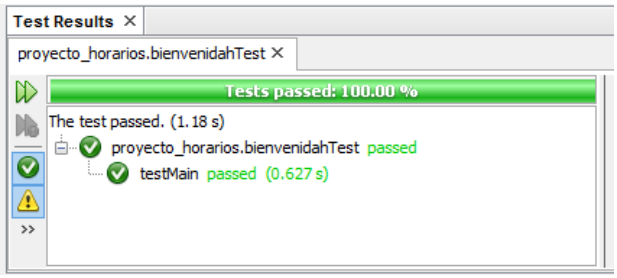

Figura 19. Prueba Satisfactoria. Fuente: Elaboración propia.

Para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera, es decir, en función de algún valor de entrada, se evalúa el valor de retorno esperado; si la clase cumple con la especificación, entonces Junit devolverá que el método de que la clase pasó exitosamente la prueba; en caso de que el valor esperado sea diferente al que regresó el método durante la ejecución, se devolverá un fallo en el método correspondiente.

Las pruebas de integración en el sistema consisten en el funcionamiento de varios componentes de un jframe. En el proceso del registro de docentes se observa en la figura 20 esta prueba donde el botón valida si en los campos existen datos y posteriormente es cifrado para enviarlos y almacenarlos en la base de datos.

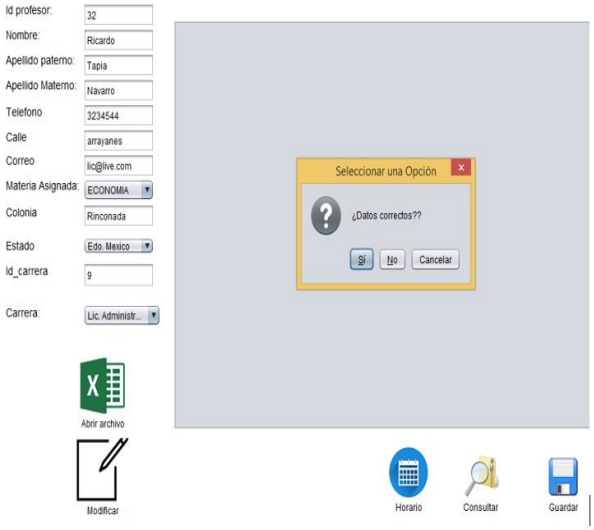

Figura 20. Pruebas de Integración. Fuente: Elaboración propia.

4.3 Pruebas de caja negra

Las pruebas de caja negra son aquellas que se llevan a cabo sobre la interfaz de software y pretenden demostrar que este funciona adecuadamente, donde las entradas se aceptan de forma adecuada y se produce una salida correcta, cabe mencionar que este tipo de pruebas pueden aplicarse al sistema sin necesidad de conocer como está construido por dentro [11].

Como ejemplo de la implementación de las pruebas de caja negra se tiene la interfaz gráfica del acceso al sistema, ya que esta es un proceso en el cual se valida una entrada pues al ingresar un usuario y una contraseña se espera que esté registrado en la base de datos. Una vez ingresados los datos correctos se hace la validación e ingresamos al sistema de manera correcta, tal y como se muestra en la figura 21.

Figura 21. Prueba de entrada. Fuente: Elaboración propia.

En la realización de las pruebas de caja negra también se valida el proceso de salida en el sistema. El producto final es un reporte que está elaborado en Excel, el cual es generado por información existente en la base de datos que fueron introducidos por el usuario. El reporte es capaz de agrupar los registros de cada profesor en diferentes hojas, con el membretado de la Institución, ID y nombre del docente, asignaturas a impartir en el horario y carrera correspondiente, generando una hoja para cada docente, tal como se visualiza en la figura 22.

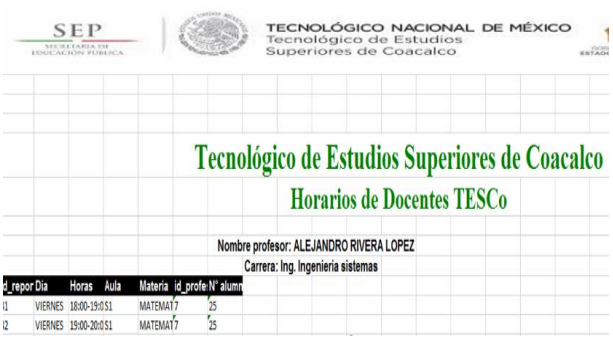

Figura 22. Prueba Salida. Fuente: Elaboración propia.

4.4 Pruebas de seguridad de software

Las pruebas de seguridad de software, son aquellos procesos que permiten verificar o revelar la calidad de la seguridad dentro de un sistema informático, ayudan a detectar vulnerabilidades y huecos de seguridad protegiendo la información evitando perdidas que pueden llegar a ser invaluables [12].

Para la realización de estas pruebas, se instaló el sistema informático en el sistema operativo Windows server 2008 R2, configurando el entorno para instalar el sistema. Se utilizó un sistema operativo Kali Linux para realizar las pruebas de auditoria a través herramienta Nmap con la cual se realizó un escaneo de servicios a el sistema operativo del servidor con parámetros específicos se obtiene la información suficiente para conocer las aplicaciones o programas que se encuentran dentro de dicho sistema operativo, como se muestra en la figura 23.

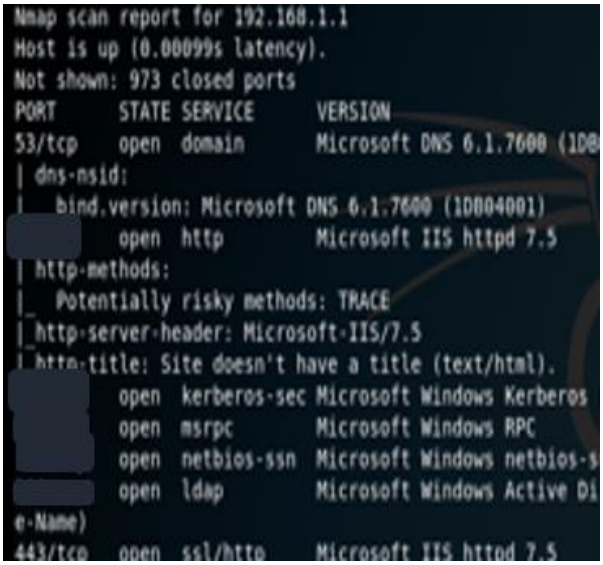

Figura 23. Escaneo Servicios. Fuente: Elaboración propia.

Terminado de escanear todos los servicios con los que cuenta el servidor, se procede a realizar una valoración de vulnerabilidades sobre el mismo para conocer cuáles son los principales riesgos que se presentan, tal y como se puede apreciar en la figura 24.

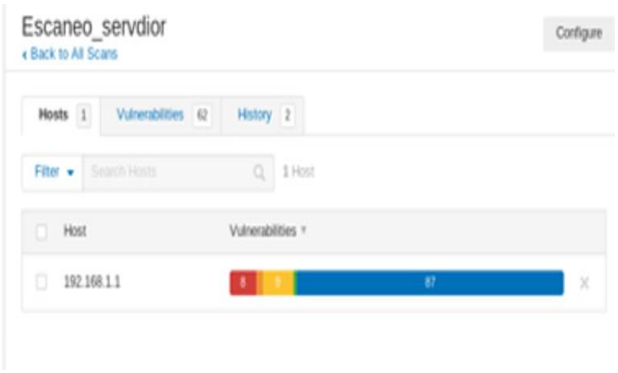

Figura 24. Escaneo Vulnerabilidades. Fuente: Elaboración propia.

La herramienta Nmap tiene como característica en este escaneo, clasificar las vulnerabilidades por colores, los cuales son:

- Rojo: Critico.
- Naranja: Alto.
- Amarillo: Medio.
- Verde: Bajo.
- Azul: Información.

En la figura 25, se pueden observar las vulnerabilidades de tipo crítico que en este caso son las que tiene que ver con los servicios HTTP, DNS, UPDATE, entre otras opciones de comunicación del sistema Windows server 2008 r2.

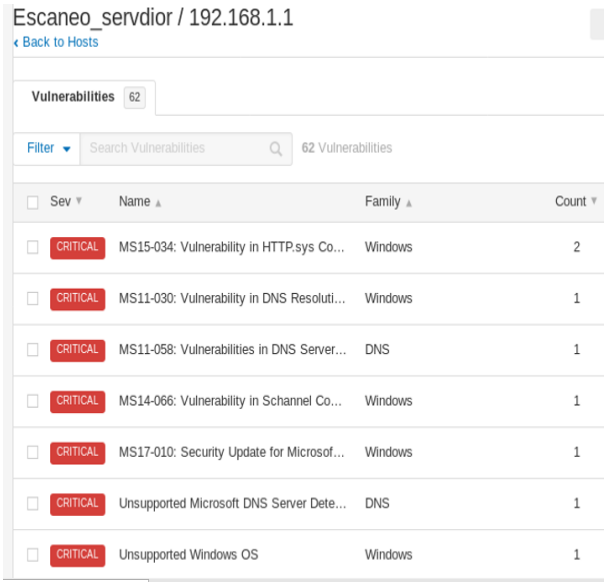

Figura 25. Vulnerabilidades críticas. Fuente: Elaboración propia.

5. Resultados

Al haber seguido la metodología descrita (RUP) en el punto 2 como la más conveniente para aplicarla a este proyecto, ya que es una metodología adaptable a las necesidades cambiantes de los usuarios finales, y desarrollar así un sistema de información se describen mediante la tabla 1 los procedimientos realizados que contiene los elementos que fueron desarrollados en cada etapa del Sistema de información.

Tabla 1. Procedimiento realizado siguiendo los métodos de desarrollo del sistema.

Actividad		Fecha	Instrumentos	Productos resultantes
Recopilacion de informacion relacionada.		08/03/2018 10/03/2018	• Bases de datos de los docentes de la Institucion en Excel. • Documentacion proporcionada por la institucion. • Repositorios academicos.	• Levantamiento y especificaciones de requerimientos dados por el cliente. • Revision de trabajos relacionados.
Conceptualizacion del problema.		11/03/2018 15/03/2018	• Elaboracion de documentacion mediante Microsoft office (Word 2010).	• Anteproyecto de investigacion. • Justificacion y objetivos del proyecto.
Desarrollo del sistema mediante la metodologia RUP.	Concepcion	16/03/2018 21/03/2018	• Enterprise Architect para la elaboracion de diagramas UML. • Desarrollo del sistema de informacion siguiendo la metodologia.	• Analisis de funcionalidades del sistema. • Elaboracion de casos de uso. • Elaboracion de diagrama de dominio. • Elaboracion de Modelo de negocios. • Elaboracion de Modelo Entidad-Relación.
	Desarrollo	05/04/2018 20/04/2018	• Balsamiq Studios. • Insercion de informacion, bases de datos.	• Simulacion del sistema (maquetado). • Excel para ingresar informacion y bases de datos.
	Implementacion	26/04/2018 18/06/2018	• Java con IDE Netbeans 8.1 • AppServer 6.0 • Microsoft office (Excel 2010)	• Codigo fuente del sistema. • Desarrollo de interfaces. • Desarrollo de bases de datos mediante. • Generacion de reportes Electronicos.
	Transicion	20/06/2018 08/07/2018	• Herramienta JUnit de Netbeans. • Microsoft office 2010(Word)	• Pruebas del sistema. • Elaboracion del manual de usuario.
Validacion y verificacion del sistema		10/07/2018 15/07/2018	• Ubuntu 16.0.4 • Windows server 2008 profesional • Kali Linux 2017.	• Simulaciones • Sistema terminado • Prueba UAT
Analisis de resultados y conclusiones		18/07/2018 01/08/2018	• Procesador de texto Microsoft office (Word 2010)	• Analisis de resultados • Uso del sistema de informacion dentro del area de control escolar de la Institucion. • Uso de manual de usuario para la instalacion y el empleo del sistema de informacion.

Dentro del desarrollo del sistema de información y siguiendo la metodología RUP, se desarrollaron diferentes modelos que UML provee, como por ejemplo, el modelo de negocio, que ayuda a conocer los procesos del desarrollo del sistema, los casos de uso para definir los módulos con los que cuenta el sistema y comprender el comportamiento y la funcionalidad que éste abarcará mediante la interacción con los usuarios que hacen uso de el, es decir, la relación entre los actores y los casos de uso del sistema. Para el diseño de las pantallas del sistema se utilizó la herramienta de balsamiq para mostrar al usuario final una simulación de las interfaces del sistema, con ayuda del diagrama de dominio que es utilizado en la fase de concepción y es tomado como punto de partida para el diseño del sistema Por otra parte, con la ayuda de Excel se pudo hacer la importación de bases de datos a PHP y a su vez la exportación del horario correspondiente a cada docente. El uso del IDE de Netbeans mediante el lenguaje Java para la codificación y el desarrollo del sistema de información.

Figura 26. Modelo de interfaces y navegación en el Sistema de información.

6. Trabajo a futuro

Debido a la naturaleza evolutiva del software, debe actualizarse constantemente, adaptándose a las necesidades cambiantes de su entorno.

Se sugiere realizar otro apartado en el sistema informático donde se puedan verificar los perfiles de cada docente, esto con el fin de que cada uno pueda impartir una materia que sea afín a sus conocimientos y a su experiencia laboral, ayudando a mejorar el aprendizaje del alumnado dentro de la Institución.

7. Conclusiones

La metodología RUP es un proceso iterativo que se adapta a los cambios y que pretende abarcar todas las fases en el desarrollo de software desde la concepción hasta la transición. Al realizar este proyecto de investigación se diseñó, desarrollo e implementó un sistema informático que apoyará a la generación de horarios de los docentes del Tecnológico de Estudios Superiores de Coacalco. Cabe resaltar que en todas las etapas de desarrollo se consideró la seguridad informática, logrando tener un mejor control en la información que es un pilar importante dentro de la Institución educativa mencionada. El uso de herramientas y técnicas de seguridad permite resguardar el activo principal que es la información, sin embargo, aún hace falta mucho por hacer, pues es bien sabido que la seguridad completa no existe y se debe trabajar en ella día con día.

8. Agradecimientos

Los autores del artículo agradecen al Tecnológico de Estudios Superiores de Coacalco como Institución que brindo la información necesaria para la realización del presente proyecto de investigación.

9. Referencias

[1] Urbina Baca, G. (2016). Introducción a la seguridad informática (primera ed.). (J. E. Callejas, Ed.) México, México: Grupo editorial PATRIA.

[2] Benítez, H. (10 de julio de 2018). La informática. (Anónimo) Recuperado el julio de 10 de 2018, de La informática: https://hugobenitez.es.tl/Restricci%F3n-al-acceso-f%EDsico-.-.htm.

[3] García Alfonso, Alegre Ramos María del Pilar. (2011). Seguridad informática. España: Ediciones Paraninfo.

[4] Gaussianos. (10 de julio del 2018). Recuperado el 10 de julio de 10, de Gaussianos: https://www.gaussianos.com/critpografia-cifrado-por-sustitucion/.

[5] Noriega, S. (10 de julio de 2018). Blog de seguridad informática. Obtenido de https://www.certsuperior.com/Blog/que-es-el-cifrado-y-para-que-funciona-medida-de-seguridad-primordial-para-tu-empresa.

[6] Apodaca, F. (2012). Ingeniería de Software, Metodología RUP. Recuperado el 23 de Mayo de 2018, de http://metodologiadesoftware.blogspot.com/2012/11/fases-del-modelo-rup_27.html.

[7] Debrauwer, L. (2016). UML 2.5 Iniciación, ejemplos y ejercicios corregidos. México D.F: eni.

[8] Martínez, I. C. (4 de junio de 2018). Modelo Conceptual/Modelo de dominio. Obtenido de Modelo Conceptual/Modelo de dominio: https://ldc.usb.ve/~martinez/cursos/ci3715/clase6_AJ2010.pdf.

[9] Fossati, M. (2015). Introducción a UML lenguaje para modelar objetos. Argentina: Alfaomega.

[10] Arisa, G. (2015). Manual práctico de investigación de operaciones (Cuarta ed.). Barranquilla: Universidad del norte.

[11] Arias, Á. (2015). Aprende sobre la Ingeniería de software. México: Alfaomega.

[12] Guillermo Pantaleo, L. R. (2016). Ingeniería de software. Argentina: Alfaomega.