

Universidad de las Ciencias Informáticas

Facultad 1



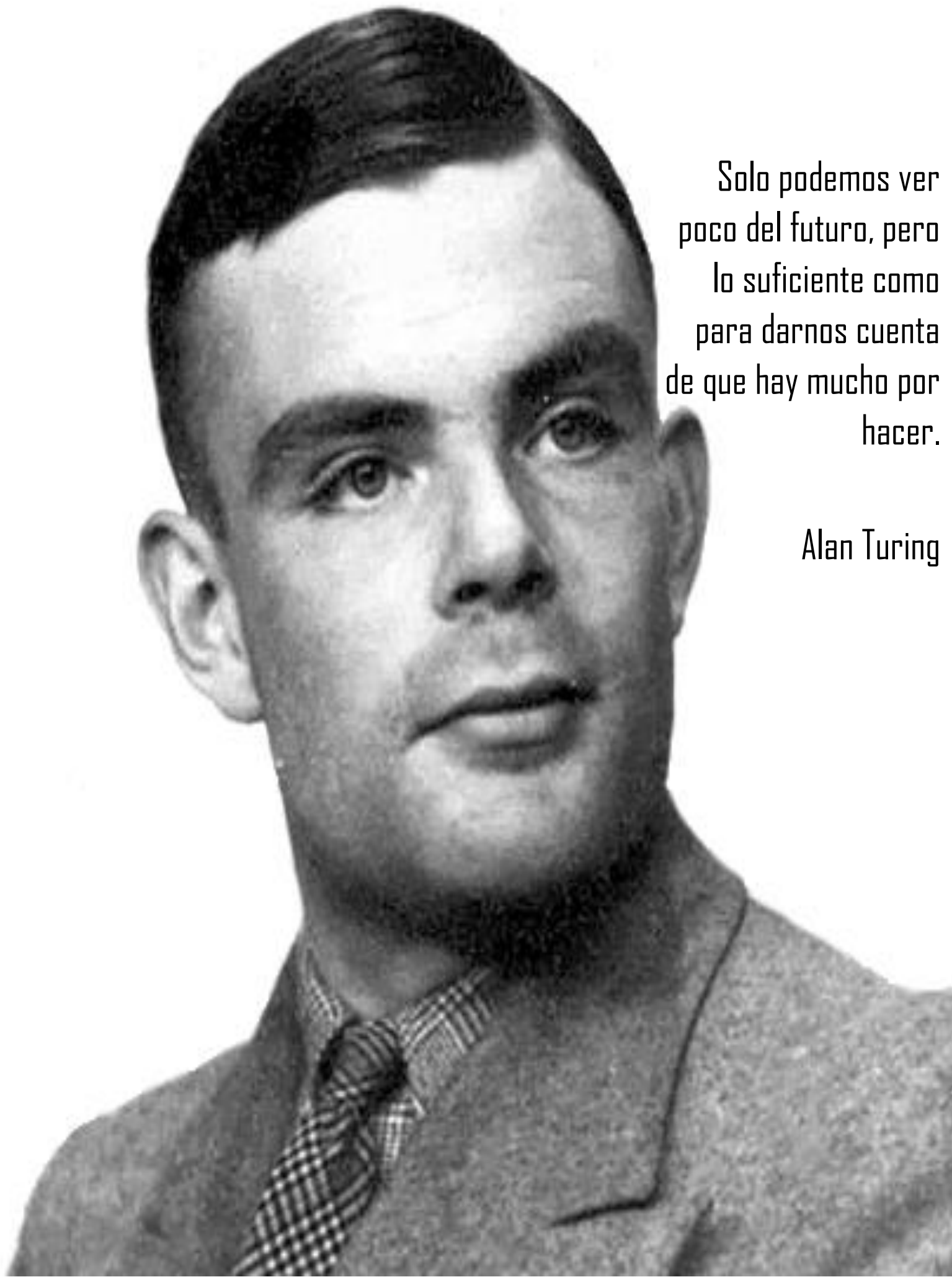
Título: *Sistema de registro y control de pasajes en territorio nacional por Viazul.*

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

Autor: *Javier Leyva Tamayo*

Tutor: *MsC. Radel Calzada Pando*

La Habana, 2019.



Solo podemos ver
poco del futuro, pero
lo suficiente como
para darnos cuenta
de que hay mucho por
hacer.

Alan Turing

Declaración de autoría

Declaro por este medio que yo Javier Leyva Tamayo, con carnet de identidad 94010829860, soy el autor principal del trabajo titulado “**Sistema de registro y control de pasajes en territorio nacional por Viazul**” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Declaro que todo lo anteriormente expuesto se ajusta a la verdad, y asumo la responsabilidad moral y jurídica que se derive de este juramento profesional.

Y para que así conste, firmo la presente declaración de autoría en La Habana a los ____ días del mes de _____ del año 2019.

Autor:

Javier Leyva Tamayo

Tutor:

MSc. Radel Calzada Pando

Resumen

Con el objetivo de dar solución a los problemas actuales referentes al registro y control de los pasajes emitidos vía voucher¹ por la empresa Viazul en la Dirección General de Economía de la Universidad de las Ciencias Informáticas (UCI) se realiza la siguiente investigación. Una vez analizadas las características del proceso y en pos de la gestión de la información que el mismo genera se plantea la necesidad de desarrollar una solución informática a dicho proceso. El estudio y análisis del estado del arte permitieron identificar funcionalidades y tecnología para la comprensión y diseño de la solución propuesta. El desarrollo estuvo guiado por la metodología XP y se seleccionó como principales tecnologías el marco de trabajo Symfony, PHP como lenguaje de programación y el entorno integrado de desarrollo PhpStorm. La estrategia de prueba aplicada permitió verificar el cumplimiento de los objetivos trazados y evaluar la calidad del sistema.

Palabras claves: gestión, sistema, tecnologías, voucher.

¹ En inglés, voucher hace referencia a un 'pedazo de papel impreso que da derecho a un descuento, o se puede intercambiar por mercancía o servicios'... Tanto la grafía hispanizada váucher, como la del inglés voucher son válidas y correctas.

Índice

| | |
|---|-----|
| Declaración de autoría | i |
| Resumen | ii |
| Índice | iii |
| Introducción | 1 |
| Capítulo 1. Fundamentación teórica | 8 |
| 1.1 Introducción | 8 |
| 1.2 Principales conceptos | 8 |
| 1.2.1 Sistema de gestión | 8 |
| 1.3 Estado del arte para el Sistema de Registro y Control de Pasajes | 9 |
| 1.3.1 Internacionales | 9 |
| 1.3.2 Nacionales | 10 |
| 1.4 Metodología | 11 |
| 1.4.1 Metodologías Ágiles | 11 |
| 1.5 Herramientas y Tecnologías | 13 |
| 1.5.1 Tecnologías de programación del lado del servidor(back-end) | 14 |
| 1.5.2 Lenguaje de modelado | 16 |
| 1.5.3 Framework de Interfaz (front-end) | 16 |
| 1.5.4 Framework de desarrollo | 18 |
| 1.5.5 Servidor web | 20 |
| 1.5.6 Herramientas CASE | 22 |
| 1.5.7 IDE de desarrollo | 23 |
| 1.5.8 Gestores de base de datos | 23 |
| 1.6 Conclusiones parciales | 25 |
| Capítulo 2 Solución propuesta | 26 |
| 2.1 Introducción | 26 |

| | | |
|-------------------------------------|--|----|
| 2.2 | Descripción general de la propuesta de solución | 26 |
| 2.3 | Descripción del sistema | 27 |
| 2.4 | Descripción de proceso de negocio | 27 |
| 2.4.1 | Actores y trabajadores que intervienen en los procesos del negocio | 30 |
| 2.5 | Requisitos del software | 30 |
| 2.5.1 | Requisitos Funcionales (RF) | 31 |
| 2.5.2 | Requisitos no Funcionales (RNF) | 32 |
| 2.6 | Historias de usuarios | 33 |
| 2.7 | Planificación | 42 |
| 2.7.1 | Plan de entrega | 42 |
| 2.7.2 | Plan de iteraciones | 44 |
| 2.8 | Diseño | 44 |
| 2.8.1 | Descripción de la arquitectura | 45 |
| 2.8.2 | Patrones de diseño | 46 |
| 2.8.3 | Patrones Generales de Software para Asignar Responsabilidades. | 47 |
| 2.9 | Modelo conceptual | 47 |
| 2.10 | Modelo de base de datos | 48 |
| 2.11 | Conclusiones parciales | 49 |
| Capítulo 3 Implementación y pruebas | | 50 |
| 3.1 | Introducción | 50 |
| 3.2 | Estándares de codificación | 50 |
| 3.2.1 | Estructura | 50 |
| 3.2.2 | Convención de Nombres | 53 |
| 3.3 | Diagrama de Despliegue | 53 |
| 3.3.1 | Descripción del Diagrama de despliegue. | 54 |
| 3.4 | Pruebas | 54 |

| | |
|---|----|
| 3.4.1 Pruebas de funcionalidad o aceptación | 55 |
| 3.4.2 Pruebas de Seguridad | 58 |
| 3.5 Conclusiones parciales | 60 |
| Conclusiones Generales | 61 |
| Recomendaciones | 62 |
| Referencias Bibliográficas | 63 |

Introducción

Las Tecnologías de la Información y las Comunicaciones (TIC) están presentes en gran parte de las actividades humanas: en el ocio, en la educación, en la comunicación, en la forma de relacionarnos con los demás y en el mundo de los negocios. El creciente desarrollo de las TIC impone llevar a cabo la transformación y el perfeccionamiento en la organización de la información, de forma eficiente, con calidad y en correspondencia con los detalles del entorno. El uso de estas tecnologías en el mundo moderno ha incrementado considerablemente la capacidad de almacenamiento, gestión, procesamiento y distribución de datos e información de todo tipo, lo que posibilita su aplicación en diferentes esferas de la sociedad tales como la educación, el transporte y salud.

“Las TIC aportan con fuerza a la reducción de costos de las compañías, ahorrando no sólo en dinero, sino que también en horas hombre y recursos energéticos. Así, las tecnologías de la información se han convertido en un elemento trascendental en nuestra forma de trabajar, y es imposible pensar el trabajo de un profesional o el desempeño de una compañía sin ellas” (Soto, 2013). De acuerdo al informe “Talent Mobility 2020, The next generation of international assignments” de la consultora PricewaterhouseCoopers (PwC), tal es la importancia que hoy las TIC tienen en la sociedad y en el mercado, que quien(es) no sepa(n) subirse a la “ola tecnológica”, no podrán sobrevivir en el futuro entorno internacional. (PricewaterhouseCoopers , 2013).

La Informatización de la Sociedad se define en Cuba como el proceso de utilización ordenada y masiva de las nuevas tecnologías de la Informática y las Comunicaciones para satisfacer las necesidades de la información y el conocimiento de la sociedad. Este proceso tiene entre sus objetivos lograr una cultura digital como una de las características imprescindibles del hombre nuevo lo que facilitaría a nuestra sociedad acercarse hacia el objetivo de un desarrollo sostenible.

Los medios de transporte han alcanzado a lo largo del tiempo una gran importancia en el desarrollo de la sociedad. Estos median y facilitan el desplazamiento de bienes o individuos a través de regiones distantes lo que trae consigo un incremento sustancial en la eficiencia de los procesos que requieren del mismo. El creciente desarrollo económico y social ha incrementado exponencialmente la demanda de medios de transporte que suplan sus necesidades. Viajar es una necesidad imperante en la sociedad humana, ya

sea por trabajo, ocio u otra situación el hombre siempre necesitará suplir esta necesidad y por ende dependerá en su mayoría de una entidad que brinde este servicio. Dichas entidades tienen la necesidad de gestionar sus medios de transporte, así como su forma de atender a los clientes, la emisión de pasajes o boletos, las reservaciones o el pago directo de viajes.

En Cuba, la gestión del transporte y sus respectivas reservaciones están teniendo un gran auge, por ello, se fomenta la expansión de proyectos que agilicen y hagan más eficientes los trámites requeridos, facilitando a su vez el acceso a la información y garantizando su calidad. Por tal razón varias entidades han dedicado tiempo y recursos para impulsar la informatización de los procesos asociados a la gestión del transporte; una de ellas es la Universidad de las Ciencias Informáticas (UCI).

En la UCI, en correspondencia con el actual desarrollo y puesta en marcha de proyectos en el ámbito de la informática y las telecomunicaciones, se están desarrollando varios proyectos que pretenden informatizar todos los procesos. Entre estos no solo relacionados con la actividad educacional dentro de la universidad, sino en el entorno de la sociedad, el cual será el punto de partida para la informatización cubana.

Las empresas, instituciones y universidades cubanas, realizan diferentes tramites de pago. Los Procesos referentes a estos trámites se efectúan tradicionalmente de manera manual pues de alguna forma siempre se han visto dificultados debido a la situación de bloqueo que tiene nuestro país, lo cual le imposibilita adquirir algunas herramientas para realizar eficientemente estos procesos.

El pago se realiza cuando se adquiere o se participa en algún servicio en el cual se vea implicado el sistema financiero de cualquier persona o entidad implicada. Uno de los factores que influirá será la forma de pago que se vaya a liquidar el servicio que se ha dado. El pago puede ser realizado al contado o aplazado (Este último término no se realiza en la UCI). La realización del pago al contado puede hacerse de diversas formas; entre las más habituales se encuentran: la entrega de dinero en efectivo, en cuyo caso se desarrolla en el Caja de la UCI, los pagos de dietas o las transferencias bancarias que son desarrolladas al igual que el cheque a pagos a empresas.

En las instituciones de Educación Superior existen diferentes procesos de pago que como ya se han venido explicando no están informatizados. Los principales son el pago de dietas a estudiantes,

profesores y trabajadores y el pago de la bonificación del 50% del pasaje a estudiantes. Otro proceso de pago importante lo constituye el pago de factura a organismos externos a las universidades, que prestan algún servicio o la universidad realiza alguna compra, un ejemplo de ellos son los pagos que se realizan en divisa mediante un documento de pago por anticipo para la adquisición de pasaje en territorio nacional.

La Dirección General de Economía, área de la UCI, se subdivide en diferentes áreas de trabajo que de una forma u otra laboran de forma conjunta, compuesta por: Dirección de Contabilidad y Finanzas, Dirección de Planificación, Dirección de Almacenes, el Grupo de Gestión Energética y el Departamento de Información y Estadística. De todas estas áreas la dirección de Contabilidad y Finanzas es quien se encarga de registrar todos los hechos económicos que se llevan a cabo en el centro. Esta área ejecuta y controla el pago en divisa de los pasajes que se autorizan por parte del Rector mediante voucher a la Empresa Viazul por el concepto: Pasaje en Territorio Nacional².

Actualmente estos procesos de pago no están informatizados en la Dirección de Contabilidad y Finanzas. Esta lo que hace es registrar el gasto asociado al área de la UCI que fue solicitado, pero no crea un expediente con nombre, destino del viaje, así como las fechas de ida y regreso del viajero. Esto implica entre otras dificultades que: no se tiene acceso a la cantidad de veces que una persona puede hacer uso de un voucher en divisa, tampoco la cantidad de veces que un área del centro ha ejecutado una solicitud, así como el estado en que está el voucher que fue entregado y no se tiene control del gasto del voucher sobre la forma de pago de cheque por el cual se deposita en la Empresa de Viazul del anticipo para ser gastado. Además de los problemas inherentes a llevar procesos complejos de forma manual como son:

- Información duplicada.
- Pérdida de información al realizarse los documentos manualmente.
- Pérdida de tiempo en la búsqueda.
- Requiere de un seguimiento continuo.

² Cuenta que está asociada a una partida de gasto definida en el presupuesto de la Universidad que lleva por nombre: Pasaje en territorio nacional.

- Al estar en formato manual se necesita un local para almacenar los registros de documentos por un tiempo.
- Se dificulta la realización de reportes solicitados por la dirección.

Por todo lo anterior planteado, se llega a la conclusión de que es necesario el desarrollo de un sistema que informatice los procesos referentes al pago en divisa de los pasajes emitidos mediante voucher a la empresa Viazul por el concepto: Pasaje en territorio nacional. Con el fin de solucionar la problemática planteada se ha formulado el siguiente **problema de investigación**: ¿Cómo contribuir a la gestión de la información de los procesos referentes al pago en divisa de los pasajes emitidos mediante voucher a la empresa Viazul por el concepto Pasaje en Territorio Nacional, para mejorar su registro y control en la Dirección de Contabilidad y Finanzas?

Para dar respuesta a ello se centra el **objeto de estudio** en los sistemas de gestión de la información de procesos referentes al pago de pasajes; enmarcando el **campo de acción** de la investigación en los procesos referentes al pago en divisa de los pasajes emitidos mediante voucher a la empresa Viazul por el concepto: Pasaje en Territorio Nacional. Para guiar la búsqueda de la solución al problema planteado se define como **objetivo general**: desarrollar un sistema informático que agilice la gestión de la información de los procesos referentes al registro y control de pasaje en territorio nacional por Viazul en la Dirección de Contabilidad y Finanzas.

Del objetivo general planteado se definen los siguientes objetivos específicos:

1. Diagnosticar el estado actual de la gestión de entrega de voucher en divisa por Viazul.
2. Fundamentar la selección de la metodología, herramientas y tecnologías a utilizar en el desarrollo del sistema de gestión de la información.
3. Realizar el análisis de las funcionalidades contables para el pago de pasajes en divisa por voucher de Viazul.
4. Diseñar los elementos que se ajustan a las necesidades del proceso de gestión de entrega de voucher en divisa en Viazul.

5. Implementar el sistema para la gestión de la información referente al pago de pasajes en divisa por voucher de Viazul.
6. Realizar pruebas al sistema para la gestión de la información referente al pago de pasajes en divisa por voucher de Viazul.

Se formula la siguiente **idea a defender**:

El desarrollo de un sistema informático para la gestión de la información contable de los procesos referentes al registro y control de pasaje en territorio nacional por Viazul agilizará su procesamiento y la protección en la Dirección de Contabilidad y Finanzas.

Dentro de las **tareas de la investigación** que se propone para dar solución a los objetivos se encuentran:

1. Revisión de documentos de las áreas de la Dirección de Contabilidad y Finanzas de la Dirección General de Economía.
2. Entrevista a directivos y trabajadores.
3. Caracterización de las metodologías, herramientas y tecnologías a utilizar en el desarrollo del sistema.
4. Búsqueda de elementos contables necesarios para el control de los voucher por divisa en Viazul.
5. Aplicación de las pruebas al sistema.

Métodos teóricos:

- **Analítico-sintético:** el empleo de este método se evidencia en el análisis, identificación y selección de los principales conceptos y definiciones relacionadas con el tema de investigación, lo que permitió elaborar una propuesta de solución enfocada en la situación planteada.
- **Histórico-Lógico:** este método se utilizó para el estudio de las practicas actuales de otras aplicaciones, así como las características de la metodología de desarrollo, lenguajes de programación y framework en el desarrollo del sistema.

Métodos empíricos:

- **Entrevista en profundidad:** para identificar las funcionalidades a implementar y establecer criterios en vista del desarrollo del sistema se entrevistó al MCs. Radel Calzada Pando, trabajador encargado de los procesos referentes al pago en divisa de los pasajes emitidos mediante voucher a la empresa Viazul por el concepto: Pasaje en territorio nacional; para así tener una mejor visión de lo que necesita el cliente.

Composición del documento:

La presente investigación consta de tres capítulos en los que se abordan diferentes temáticas los cuales están desglosados de la siguiente forma:

Capítulo 1 Fundamentación teórica.

Se exponen los conceptos y fundamentos generales que sirven de soporte teórico a la propuesta de solución que se plantea y un estudio del estado actual de los sistemas informáticos para la gestión de la reservación y pago de pasaje. Se analizan las herramientas y lenguajes de programación definidos para el desarrollo del Sistema de registro y control de pasaje en territorio nacional por Viazul, así como la metodología a emplear en el desarrollo del módulo.

Capítulo 2 Solución propuesta.

Se detalla el flujo actual de los procesos involucrados en el campo de acción y se identifican las funcionalidades y características del sistema. Además, se exponen los patrones de diseño y de arquitectura que utiliza el marco de trabajo seleccionado y se elaboran los artefactos correspondientes al análisis y el diseño del software según la metodología de desarrollo de software definida para guiar el desarrollo de la solución de la presente investigación.

Capítulo 3 Implementación y pruebas.

Se describen los elementos necesarios para la implementación del módulo y se modela la vista de despliegue del sistema. A su vez se confeccionan los casos de prueba a realizar al software para

determinar si cumple con las funcionalidades identificadas, se describe el análisis de los resultados alcanzados en las pruebas y se mencionan las acciones realizadas para resolverlas.

Capítulo 1. Fundamentación teórica

1.1 Introducción

En este capítulo se exponen los conceptos y fundamentos generales que sirven de soporte teórico a la propuesta de solución que se plantea y un estudio del estado actual de los sistemas informáticos para la gestión de la reservación y pago de pasaje. Se analizan las herramientas y lenguajes de programación definidos para el desarrollo del Sistema de registro y control de pasaje en territorio nacional por Viazul, así como la metodología a emplear en el desarrollo del módulo.

1.2 Principales conceptos

Un sistema es un conjunto de partes o elementos organizados y relacionados que interactúan entre sí para lograr un objetivo. Los sistemas reciben entrada de datos y proveen salida de información. Un grupo de elementos no constituye un sistema si no hay una relación e interacción que dé la idea de un todo con un propósito (Natividad Seguencia, Ing. Rosa, 2011).

1.2.1 Sistema de gestión

Un sistema de gestión ayuda a lograr los objetivos de una organización mediante una serie de estrategias que incluyen la optimización de procesos y el enfoque centrado en la gestión de la información (Rodríguez Ferrer, Orquidia; Díaz Bestard, Rayner, 2013).

Se define sistema de gestión como un “sistema para establecer la política y los objetivos y para lograr dichos objetivos”. De igual forma se presenta al sistema como el “conjunto de elementos mutuamente relacionados o que interactúan entre sí”.

El desarrollo de un sistema de gestión proporciona ventajas como:

- Control de las actividades de la organización.
- Disponibilidad de información para los usuarios en tiempo real.
- Elimina la barrera de la distancia trabajando con un mismo sistema en puntos distantes.

- Disminuye errores, tiempo y recursos (Rosa, 2011).

1.3 Estado del arte para el Sistema de Registro y Control de Pasajes

En la actualidad existen diversos sistemas para el registro y control de pasajes. En esta sección se describirán varios de estos sistemas para el registro y control de pasajes tanto nacionales como internacionales, lo que permitirá obtener información referente a dichos sistemas para arribar a conclusiones y así guiar la propuesta de solución, hacia el cumplimiento de los objetivos propuestos.

1.3.1 Internacionales

En el mundo podemos encontrar varios sistemas encargados de gestionar el registro y control de pasajes dada la utilidad de los mismos. Estos sistemas en su mayoría son desarrollados con un alto nivel de personalización debido a las características específicas de cada institución que vaya a hacer uso del mismo. Estos sistemas si bien no son desarrollados para controlar los procesos de registro y control de pasajes de una determinada institución permiten realizar una correcta gestión de los mismos, ejemplo de ellos son:

- **Sibus:** es el producto insignia de la empresa del mismo nombre la cual cuenta con las herramientas necesarias para una óptima administración de la venta de pasajes, manteniendo un real control del efectivo y las ventas de cada oficina, así como también de las tarifas, agilizando de esta forma el proceso de venta; también cuenta con un componente de Reportera, Gráficos y Estadística de la empresa el cual permite revisar el estado de las ventas, generar informes estadísticos de las oficinas, compara tramos, rutas o periodos de venta. No realiza pago por concepto de Vouchers y es un sistema privativo. (SIBUS, 2018)
- **ControlPass:** es una solución integral de control de pasajeros específicamente diseñada para empresas de transporte que además permite conocer en tiempo real la posición y frecuencia de la flota. Del lado de la empresa contratante del servicio de transporte ControlPass cuenta con un módulo de supervisión que permite consultar el histórico de pasos por parada de los diferentes autobuses que conforman las rutas, así como el control de personas que usan el servicio (sólo las personas autorizadas acceden), control de periodos de validez y rutas, posibilita de cobrar fácilmente por viajes. No realiza pago por concepto de Vouchers y es un sistema privativo. (ControlPass, 2018).

1.3.2 Nacionales

En Cuba solo existe un sistema que permite llevar a cabo el registro y control de pasajes en territorio nacional.

- **Viazul:** es un sistema de reservas en línea de Viazul que garantiza la seguridad de las reservaciones y las transacciones efectuadas. El sitio realiza las operaciones a través de una pasarela de pagos, registra los datos personales de sus clientes y los datos de las reservaciones efectuadas a través del sitio, permite cancelar las reservaciones y obtener el reembolso del pago realizado, registra el valor de las reservaciones. No cuenta con un módulo que genere reportes a partir de los datos almacenados y no realiza pago por concepto de Vouchers. (Viazul, 2018).

Después de un análisis con el encargado de la gestión del proceso de registro y control de pasaje en territorio nacional vía voucher por Viazul concluimos que el Sistema a utilizar deberá cumplir con los siguientes parámetros:

- Licencia pública.
- Formato Web.
- Utilización de voucher o un elemento similar.
- Registro de uso de voucher.

| Sistema | Licencia | Web | Utilizar voucher o similares | Registro de uso de voucher |
|-------------|-----------|-----|------------------------------|----------------------------|
| Sibus | Privativa | si | no | no |
| ControlPASS | Privativa | si | no | non |
| Viazul | Privativa | si | no | no |

Tabla 1. Parámetros para el sistema.

En resumen, ninguno de estos sistemas consta de todos los requisitos necesarios para gestionar de forma correcta el proceso de registro y control de pasaje realizado vía voucher por contrato con la empresa

Viazul. Estos no cuentan con las funcionalidades y no fueron pensados para un nivel de personalización tan específico como el que requiere el proceso y son en su mayoría privativos.

1.4 Metodología

La rama de la metodología, dentro de la Ingeniería de Software, se encarga de estructurar, planificar y controlar el proceso de desarrollo del software a través de un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software.

Entre las metodologías de desarrollo del software existen las llamadas “tradicionales”, las cuales ayudan a los profesionales a documentar y realizar las tareas de desarrollo, pero tienen cierta resistencia a los cambios. Son dirigidas por la documentación que se genera en cada una de las actividades desarrolladas. Se aplican fundamentalmente a proyectos grandes, donde el cliente interactúa con el equipo de desarrollo mediante reuniones, debiendo existir un contrato prefijado entre éstos, que especifique el alcance del proyecto. Estas metodologías prácticamente se encuentran obsoletas; otras más actuales han sido adoptadas para guiar el proceso de desarrollo, denominadas “metodologías modernas”, dentro de las cuales figuran las “metodologías ágiles” (Torres, 2012).

1.4.1 Metodologías Ágiles

El modelado ágil (MA) es una metodología basada en la práctica para modelar y documentar con eficacia los sistemas basados en software. En pocas palabras, es un conjunto de valores, principios y prácticas para hacer modelos de software aplicables de manera eficaz y ligera a un proyecto de desarrollo de software. Los modelos ágiles son más eficaces que los tradicionales porque son sólo buenos, sin pretender ser perfectos (S. Pressman, 2010).

Sobre la base de los aspectos antes expuestos de las metodologías tradicionales, se evidencia que estas no se ajustan a las particularidades del sistema a desarrollar, ya que el equipo de trabajo es pequeño, lo conforma una sola persona, no se cuenta con una definición detallada y exacta de todos los requisitos que debe cumplir, lo que implica una alta probabilidad de ocurrencia de cambios de forma gradual, a medida que se vayan obteniendo las pequeñas versiones. Por esto es más conveniente el uso de una de las metodologías ágiles, diseñadas para pequeños equipos de desarrollo y preparadas para enfrentar cambios durante la creación del software.

Algunas de las metodologías ágiles existentes

- **Programación Extrema (en inglés abreviado XP):** A fin de lograr la comunicación eficaz entre los ingenieros de software y otros participantes (por ejemplo, para establecer las características y funciones requeridas para el software), XP pone el énfasis en la colaboración estrecha pero informal (verbal) entre los clientes y los desarrolladores, en el establecimiento de metáforas para comunicar conceptos importantes, en la retroalimentación continua y en evitar la documentación voluminosa como medio de comunicación. Para alcanzar la simplicidad, XP restringe a los desarrolladores para que diseñen sólo para las necesidades inmediatas, en lugar de considerar las del futuro. El objetivo es crear un diseño sencillo que se implemente con facilidad en forma de código. Si hay que mejorar el diseño, se rediseñará en un momento posterior. La retroalimentación se obtiene de tres fuentes: el software implementado, el cliente y otros miembros del equipo de software. Al diseñar e implementar una estrategia de pruebas eficaz, el software (por medio de los resultados de las pruebas) da retroalimentación al equipo ágil (S. Pressman, 2010).
- **Scrum:** Scrum... es un método de desarrollo ágil de software... Los principios Scrum son congruentes con el manifiesto ágil y se utilizan para guiar actividades de desarrollo dentro de un proceso de análisis que incorpora las siguientes actividades estructurales: requisitos, análisis, diseño, evolución y entrega. Dentro de cada actividad estructural, las tareas del trabajo ocurren con un patrón del proceso llamado sprint. El trabajo realizado dentro de un sprint (el número de éstos que requiere cada actividad estructural variará en función de la complejidad y tamaño del producto) se adapta al problema en cuestión y se define —y con frecuencia se modifica— en tiempo real por parte del equipo Scrum. Scrum acentúa el uso de un conjunto de patrones de proceso del software que han demostrado ser eficaces para proyectos con plazos de entrega muy apretados, requisitos cambiantes y negocios críticos (S. Pressman, 2010).
- **Proceso Unificado Abierto (en inglés abreviado *OpenUp*):** En cierto modo, el proceso unificado es un intento por obtener los mejores rasgos y características de los modelos tradicionales del proceso del software, pero en forma que implemente muchos de los mejores principios del desarrollo ágil de software. El proceso unificado reconoce la importancia de la comunicación con el cliente y los métodos directos para describir su punto de vista respecto de un sistema (el caso de uso). Hace énfasis en la importancia de la arquitectura del software y “ayuda a que el arquitecto se centre en las metas correctas, tales como que sea comprensible, permita cambios futuros y la

reutilización”. Sugiere un flujo del proceso iterativo e incremental, lo que da la sensación evolutiva que resulta esencial en el desarrollo moderno del software (S. Pressman, 2010).

Consideraciones sobre las metodologías ágiles

A partir del análisis de las metodologías de desarrollo de software expuestas anteriormente, y teniendo en cuenta las particularidades de la solución a desarrollar en la presente investigación, se elige la metodología XP para realizar el sistema de registro y control de pasaje en territorio nacional por Viazul.

La metodología XP se ajusta a las características del sistema a implementar, ya que la metodología en sus principios básicos plantea la programación en equipos pequeños con pocos roles y el equipo de desarrollo está conformado por una sola persona; también plantea q los miembros del equipo pueden intercambiar responsabilidades en un momento determinado y en el este caso el desarrollador como único miembro del equipo asumiría todos los roles. Otro argumento es que existe comunicación directa con el cliente, condicionando a que conteste rápido y correctamente a cualquier pregunta, de forma tal que no se atrase la toma de decisiones. Es además una metodología abierta a los cambios y genera poca documentación lo que hace la entrega del software menos complicada y más satisfactoria.

1.5 Herramientas y Tecnologías

En la actualidad existe una gran revolución de los medios informáticos, donde los avances en las nuevas tecnologías tienen el propósito de ofrecer soluciones adaptables a las exigencias del usuario final. La integración de las tecnologías de desarrollo con las herramientas, ha surgido como una alternativa que permite elaborar aplicaciones web para la gestión de la información. Además, permiten resolver problemas actuales de una organización, facilitando así, la interacción de los usuarios con los sectores involucrados en la problemática.

Después de un análisis desarrollado acerca de las tecnologías a utilizar se determinó realizar comparaciones entre varias tecnologías, dado que en este caso particular el cliente posee abundantes conocimientos sobre informática y está familiarizado con determinadas tecnologías por lo que después de realizada dicha comparación se llega a un consenso de las que serán utilizadas.

La decisión se basó en que son herramientas y tecnologías fáciles de utilizar, tanto por parte del programador como del usuario, se cuenta con las últimas versiones oficiales de las mismas y estas están

entre las más utilizadas hoy en el mundo del desarrollo de software, además de que cumplen con las políticas de migración hacia software libre que mantiene la UCI en estos momentos.

1.5.1 Tecnologías de programación del lado del servidor(back-end)

En el contexto del desarrollo de aplicaciones están implicadas las actividades realizadas del lado del servidor; es decir, las tareas de base de datos y los servidores de aplicaciones que el usuario no puede visualizar en el explorador de Internet. Los lenguajes usados comúnmente son PHP, Java, Ruby, .NET, Python, entre otros, los cuales son los encargados de interactuar con la base de datos (Modelo de procesos para el desarrollo del frontend en aplicaciones web, 2016).

- **JSP:** En su versión 2.1, es una tecnología web del lado del servidor, que se usa habitualmente para crear documentos XHTML (por sus siglas en inglés eXtensible HyperText Markup Language, Lenguaje de Marcado de Hipertexto Extensible) y XML (por sus siglas en inglés eXtensible Markup Language, Lenguaje de marcas extensible) dinámicos. Con JSP se pueden crear aplicaciones web que se ejecuten en varios servidores web, además es compatible con variadas plataformas, o sea, se puede ejecutar sobre varios sistemas operativos como: Linux y Windows. Esta tecnología presenta similitudes con otras como: PHP y ASP (por sus siglas en inglés *Active Server Pages, Página de Servidor Activo*), pues admite que su código sea embebido dentro del HTML (por sus siglas en inglés *HyperText Markup Language, Lenguaje de marcas de hipertexto*) con el propósito de crear información dinámicamente, basándose en instrucciones o acceso a bases de datos. Esta tecnología hace uso del lenguaje de programación Java, este es un lenguaje multiplataforma, el cual posee capacidad multihilo y tiene integrado el protocolo TCP/IP (por sus siglas en inglés *Transmission Control Protocol/Internet Protocol, Protocolo de Control de Transmisión/Protocolo de Internet*), lo que lo hace un lenguaje ideal para internet (Martinez, 2013).

Ventajas que proporciona:

- ✓ Facilita la conexión a bases de datos.
- ✓ Permite crear aplicaciones web que se ejecuten en varios servidores web.
- ✓ Permite integrar los registros de órdenes con clases del lenguaje de programación Java.
- ✓ Es libre y multiplataforma.
- ✓ Tiene el código bien estructurado.
- ✓ Se puede integrar con los módulos de Java.

Desventajas:

- ✓ Requiere de una memoria más amplia debido a que está basado en Java y consume gran cantidad de recursos.
- **PHP:** en su versión 7.3 es una tecnología del lado del servidor, de código abierto y diseñado en sus inicios para el desarrollo de páginas web dinámicas. Una de sus principales características es la capacidad de soportar gran cantidad de bases de datos. También permite la integración con varias bibliotecas externas permitiéndole al programador analizar código XML y generar documentos en diferentes formatos. Este lenguaje es rápido, libre, orientado a objetos y multiplataforma, pues permite ser utilizado sobre diferentes sistemas operativos como Linux y Windows. Posee además una amplia librería de funciones y cuenta con una extensa documentación, la cual permite un rápido aprendizaje (Doyle, 2012).

Ventajas que proporciona:

- ✓ Permite la integración con disímiles tipos de servidores de bases de datos tales como: MySQL, PostgreSQL y Oracle.
- ✓ Permite emplear técnicas de programación orientada a objetos.
- ✓ Es libre y multiplataforma.

Desventajas:

- ✓ Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto, puede ser más ineficiente a medida que las solicitudes aumenten de número.

Se determinó como tecnología de programación PHP versión 7.3, el cual está orientado al desarrollo web. Como el número de usuarios del sistema será bajo y a su vez por las características del mismo la cantidad de solicitudes al servidor también lo será no se explotará la principal desventaja del lenguaje mencionada anteriormente. Permitirá obtener una gran velocidad y que el mismo no requiera de muchos recursos de hardware y además se integra perfectamente con muchos servidores. Es libre y está disponible bajo la licencia GPL (por sus siglas en inglés *General Public License, Licencia Pública General*). Es multiplataforma por lo que no tendrá ningún inconveniente al usarlo en cualquier computadora de la Universidad. Se caracteriza por la simplicidad de su código y por la amplia documentación que existe. Además, se requiere utilizar este lenguaje de programación debido a que es necesario desarrollar una

aplicación web para acceder al sistema a través de un servidor web por diversos departamentos de la Universidad y resulta muy engorroso tener que instalar en todas las estaciones de trabajo la aplicación para que sea accedida por los usuarios.

1.5.2 Lenguaje de modelado

El lenguaje de modelado usado en el diseño y construcción de la solución que se quiere llevar a cabo en la presente investigación es el Lenguaje de Modelado Unificado. Plantea que es un lenguaje estándar para escribir diseños de software, puede usarse para visualizar, especificar, construir y documentar los artefactos de un sistema de software intensivo (PRESSMAN, 2010).

Unified Model Language (UML): es el lenguaje gráfico más conocido y utilizado en la actualidad para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. Este lenguaje es semejante al de la vida real, claro y uniforme para el diseño orientado a objetos, ya que permite la fuerte integración entre herramientas, procesos y dominios. UML se especializa en el modelado de elementos conceptuales como son: procesos de negocio y funciones de los sistemas. También este lenguaje permite escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables (Ferreira dos Santos, 2012).

1.5.3 Framework de Interfaz (front-end)

Dentro del contexto del desarrollo de aplicaciones web, implica el uso de las tecnologías con las que interactúa directamente el usuario. Normalmente estas tecnologías son desarrolladas en los lenguajes de HTML, CSS y Javascript; también se usan las herramientas de diseño gráfico como Photoshop o Fireworks. El objetivo es desarrollar la interfaz gráfica de usuario (GUI), buscando una experiencia de uso bien valorada por el usuario final, siendo en algunos casos necesario hacer investigación, estudios y pruebas para llegar a este fin. Además, dentro del desarrollo de las aplicaciones web es posible desarrollar el front-end de la aplicación sin contar con una aplicación back-end que interactúe con la base de datos (Modelo de procesos para el desarrollo del frontend en aplicaciones web, 2016). Ejemplos de esto son.

- **ExtJS:** En su versión 4.0, es un framework que logra trabajar con otras librerías JavaScript empleando adaptadores. Este framework ha demostrado ser muy eficaz y competente en la realización de interfaces complejas ya que tiene embebido la mayoría de los controles de los

formularios web incluyendo celdas para mostrar datos y elementos semejantes a la programación desktop como: formularios, paneles, barras de herramientas, árboles y menús (Groner, 2013).

Ventajas que proporciona:

- ✓ Relación entre cliente-servidor balanceada: Se distribuye la carga de procesamiento permitiendo que el servidor pueda atender más clientes al mismo tiempo.
- ✓ Permite realizar complejos módulos en una página web.

Desventajas que presenta:

- ✓ Para desarrollar con ExtJS necesitamos toda una plataforma pues dependemos del paquete ExtJS para obtener los resultados deseados.
- **Bootstrap:** En su versión 3.0.1 es un entorno de trabajo definido con artefactos o módulos de software concreto, desarrollado por Twitter. Esta tecnología simplifica el proceso de creación de la interfaz de usuario combinando HTML, JavaScript y CSS (por sus siglas en inglés *Check Cascading Style Sheets*, Hojas de Estilo en Cascada). Además, se adapta a los distintos navegadores con numerosos componentes webs como: botones, etiquetas, alertas, entre muchos otros definidos en la web. Bootstrap fue programado para dar soporte a CSS 3 y HTML 5. Bootstrap está diseñado para todos los niveles: diseñador, desarrollador y principiante. Este framework se utiliza para hacer más fácil y rápida su implementación. Bootstrap está definido por módulos que son reutilizables e independientes en la página web y es una gran comunidad abierta y seguida por millones de personas (Gonzalez, 2013).

Ventajas que proporciona:

- ✓ Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como tablets y móviles a distintas escalas y resoluciones.
- ✓ Se integra perfectamente con las principales librerías Javascript, por ejemplo, JQuery.
- ✓ Ofrece un diseño sólido usando LESS y estándares como CSS3/HTML5.
- ✓ Es un framework ligero que se integra de forma limpia en nuestro proyecto actual.

- ✓ Funciona con todos los navegadores, incluido Internet Explorer usando HTML Shim para que reconozca los tags HTML5.
- ✓ Dispone de distintos layout predefinidos con estructuras fijas a 940 píxeles de distintas columnas o diseños fluidos.

Como framework de interfaz se determinó hacer uso de Bootstrap versión 3.0.1 por ser eficaz y competente en la realización de interfaces web, además de ser recomendable para sitios web, se adapta a los distintos navegadores de la Universidad. Los diseños creados con Bootstrap son simples, limpios e intuitivos, permitiéndole ganar al sistema en agilidad a la hora de cargar y al adaptarse a otros dispositivos. Se requiere utilizar el framework debido a que brinda componentes para realizar el diseño de las vistas del sistema de forma organizada y según los requisitos del sistema.

1.5.4 Framework de desarrollo

Los frameworks proveen una implementación del andamiaje para el desarrollo completo de una aplicación, facilitando la reutilización de componentes presentes en la estructura. Proporcionan una serie de puntos donde se pueden acoplar funcionalidades adicionales. No son un patrón arquitectural pero sí una colección de patrones de diseño y clases trabajando en conjunto, que tienen como fin resolver un problema específico. Por esta razón, se cuenta con una gran variedad de frameworks en distintos lenguajes de programación ya sea en PHP, Java, Ruby, Python, Javascript, entre otros (S. Pressman, 2010).

- **Symfony:** es un framework diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón de arquitectura de software Modelo Vista Controlador (MVC). Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Symfony puede ser completamente personalizado para cumplir con los requisitos de las empresas que disponen de sus propias políticas y reglas para la gestión de proyectos y la programación de aplicaciones.

Ventajas

- ✓ Tiene su propia forma de trabajo, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones,
- ✓ Sigue la mayoría de mejores prácticas y patrones de diseño para la web,

- ✓ El completo sistema de log permite a los administradores acceder hasta el último detalle de las actividades que realiza la aplicación,
- ✓ Fácil de instalar y configurar en la mayoría de plataformas de programación.
- ✓ Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.

Desventajas

- ✓ Gran parte de la velocidad de Symfony se debe a un uso extensivo del caché por lo que cuando estás desarrollando tiende a ser algo tedioso tener que estar limpiando el caché de vez en cuando,
- ✓ Se necesita un VPS (por sus siglas en inglés *Virtual Private Server*, Servidor Virtual Privado) para poder publicar aplicaciones en la web porque es necesario poder descargar e instalar cosas en un servidor para que symfony funcione apropiadamente (Zaninotto, y otros, 2010).
- **Zend Framework.** es un framework para el desarrollo de aplicaciones y servicios web con PHP. Brinda soluciones para construir sitios web modernos, robustos y seguros. Además, implementa el patrón MVC, es orientado a objetos y sus componentes tienen un bajo acoplamiento por lo que se puede usar en forma independiente.

Ventajas que propone:

- ✓ Zend Framework permite reutilizar el código desarrollado.
- ✓ El mantenimiento de las aplicaciones es muy sencillo.

Desventajas:

- ✓ Zend Framework necesita de un servidor dedicado para poder sacar el máximo provecho a las funciones con las que esta cuenta, porque como se menciona anteriormente, se configura mediante la línea de comandos, entonces cualquier cambio que se necesitara hacer se realizaría a través de la línea de comandos del servidor dedicado, en el caso de un servidor

compartido, por ejemplo, se tendría que bajar, configurar en localhost, modificar, subir, configurar en servidor.

- ✓ Requiere de una larga curva de aprendizaje.
- ✓ La comunidad de este lenguaje es muy pobre, por lo que podría ser difícil encontrar algún buen manual para realizar un proceso o aclarar una duda (Poyo, 2010).

Entre los framework para PHP se escogió Symfony versión 3.4 debido a que posee una amplia documentación tanto en español como en inglés, tiene una gran variedad de plugins, que permiten agilizar el trabajo, reduce el tiempo de desarrollo pues proporciona varias herramientas y clases que pueden ser reutilizadas por los desarrolladores. A diferencia de Zend Framework que necesita de una gran configuración antes de poder comenzar a trabajar, la estructura de archivos no está definida por sí misma, cualquier desarrollador puede modificarla, esto indica que no sería compatible con otra aplicación desarrollada con Zend Framework. Se requiere utilizar el framework debido a que ofrece características que permiten obtener aplicaciones web robustas y seguras.

1.5.5 Servidor web

- **Apache.** En su versión 2.4.23, es un servidor web de código abierto, altamente configurable y modular. Utiliza Perl, PHP y otros lenguajes scripts. Su función principal es analizar cualquier archivo solicitado por un navegador y mostrar resultados correctos de acuerdo con el código del archivo. Permite configurar los informes de errores, presenta visualización de códigos en numerosos niveles y la capacidad de determinar qué nivel del navegador puede aceptar el contenido. Es uno de los primeros servidores en soportar host basados en direcciones IP (por sus siglas en inglés *Internet Protocol*, Protocolo de Internet) y hosts virtuales. Tiene un elaborado índice de directorios, un directorio de alias, informe de errores HTTP HTML (por sus siglas en inglés *Hypertext Transfer Protocol*, Protocolo de Transferencia de Hipertexto) configurable, gestión de recursos para procesos hijos, reescritura de las URL (por sus siglas en inglés *Uniform Resource Locator*, Localizador de Recurso Uniforme), comprobación de ortografía de las URL y manuales online (Naramore, 2005).

Ventajas que proporciona:

- ✓ Esta incluye formatos de configuración no estándar.
- ✓ Es multiplataforma.
- ✓ Permite elegir el servidor web Apache.
- ✓ Soporta múltiples plataformas por lo que genera mayor usabilidad, dando la opción de utilizar diferentes sistemas operativos sin ningún problema.

Desventajas:

- ✓ No dispone de un entorno integrado con una sofisticada interfaz de usuario, asistente y ayuda en línea (ARRENDONDO, 2012).
- **IIS:** En su versión 7.5, es un servidor web capaz de brindar servicios web, los cuales transforman a una PC en un servidor web tanto para intranet como para Internet, permitiendo de esta forma publicar páginas web. Este servidor proporciona gran seguridad, capacidad de administración y escalabilidad para aplicaciones web sobre todas las versiones de *Windows Server 2003*. Además, proporciona las herramientas y funciones necesarias para administrarlo de forma sencilla (GAVILAN, 2011).

Ventajas que proporciona:

- ✓ Proporciona un ambiente confiable para lograr una mejor seguridad, ya que incluye la verificación del cambio de las aplicaciones.
- ✓ Proporciona una alta capacidad de administración, ya que incluyen cambios tecnológicos y de procesamiento de solicitudes.

Desventajas:

- ✓ Solo se puede utilizar en sistemas operativos de Windows.

Se determinó como servidor web Apache versión 2.4.23 porque no requiere la utilización de muchos recursos y permite además que muchos lenguajes puedan ser utilizados del lado del servidor. Posee además la capacidad de permitir la protección por contraseñas de las páginas de un gran número de

usuarios, la visualización del código HTML en varios niveles, registra los errores en varios formatos y tiene soporte para *host* virtuales. Se requiere utilizar el servidor web Apache debido a que la información que maneja el proceso de gestión de la información del combustible es accedida por diversos departamentos de la Universidad y resulta muy engorroso tener que instalar en todas las estaciones de trabajo la aplicación para que sea accedida por los usuarios. Además, se prevé en el futuro que se incremente el número de usuarios conectados concurrentemente, por lo que se requiere la implantación de una aplicación cliente-servidor.

1.5.6 Herramientas CASE

- **Visual Paradigm.** Es una herramienta CASE que utiliza el lenguaje de modelado estándar UML, permite la generación de códigos e ingeniería inversa. Esta herramienta cumple con las políticas de migración a software libre en Cuba, ya que es una herramienta multiplataforma que se puede utilizar tanto en Linux como en Windows. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código para entornos integrados de desarrollo tales como: NetBeans, Eclipse, Oracle JDeveloper y JBuilder (Howard, 2012).

Ventajas que proporciona:

- ✓ Tiene una interfaz muy intuitiva y es de fácil aprendizaje para los desarrolladores.
 - ✓ Permite la generación automática de diagramas a partir de descripciones de casos de usos.
 - ✓ Permite hacer descripción de los casos de usos dando una gran variedad de plantillas predeterminadas permitiendo personalizarlas.
 - ✓ Combina las funcionalidades de todas las ediciones en una amplia plataforma de modelado visual.
- **Rational Rose Enterprise.** Proporciona un lenguaje común de modelado y facilita la creación de software de calidad rápidamente. Es una herramienta de desarrollo que utiliza UML para realizar el desarrollo de software de aplicaciones, modelado de datos, servicios de diseño web, modelado de negocios y el modelado basado en componentes. Pero su principal desventaja es que solo puede ser utilizado en Sistemas operativos de Windows, además de ser una herramienta privativa (Marcisack, 2011).

Como herramienta CASE fue seleccionada Visual Paradigm versión 8.0, pues propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código para entornos integrados de desarrollo tales como: NetBeans, Eclipse, Oracle JDeveloper y JBuilder. Esta herramienta a diferencia de *Rational Rose* es libre y se destaca por ser multiplataforma. Se requiere utilizar la herramienta CASE debido a que la información que maneja el proceso de gestión de la información del combustible es extensa y se hace necesario modelar todos los procesos que se realizan para realizar la distribución y consumo del combustible en la UCI.

1.5.7 IDE de desarrollo

- **Netbeans:** en su versión 8.2, es un IDE bajo licencia GPL y de código abierto. Esta herramienta tiene la finalidad de permitirle a los desarrolladores crear diferentes sistemas y proyectos orientados sobre todo a la creación de soluciones en lenguaje Java, ya sea que se encuentren en Java SE (Edición Estándar) o Java EE (Edición Empresarial), además de soportar otros lenguajes tales como PHP y JavaScript (Jácome, y otros, 2009).
- **JetBrains PhpStorm:** es un Entorno Integrado de Desarrollo (IDE, por sus siglas en inglés) de código abierto y multiplataforma creado por la compañía JetBrains, es un poderoso y completo editor del lenguaje de programación PHP centrado en la productividad del desarrollador. Contiene un editor de código inteligente que analiza sintaxis, configuración de formato del código extendido, comprobación de errores en tiempo real sobre el proceso de implementación, plegado del código en bloques y autocompletado. Brinda soporte a los lenguajes CSS, JavaScript y HTML. Dispone de los sistemas de control de versiones más populares como Git, Subversion, Mercurial y Perforce. Ofrece un formato de código acorde a los estándares de codificación de Symfony con la integración de un módulo para este framework (JetBrains, 2019). Para el desarrollo de la propuesta de solución se utiliza JetBrains PhpStorm en su versión 2018.2.3.

1.5.8 Gestores de base de datos

- **PostgreSQL.** En su versión 9.1, es uno de los motores de base de datos relacionales más potentes que existen actualmente. Permite ejecutar consultas SQL, las cuales posibilitan actualizar, insertar, eliminar y realizar reportes sobre los datos almacenados en ficheros o bases de datos. Ofrece la posibilidad de ejecutar y trabajar varios procesos al mismo tiempo sobre la

misma tabla sin ser dañada, donde cada usuario obtiene una versión de lo último que ha hecho evitando la pérdida de información. Tiene su propio lenguaje PL/PgSQL, pero también se pueden usar lenguajes como C, C++, Gambas, Java PL, Java Web, Perl, Php, Python (NOVELLA, 2012).

Ventajas que proporciona:

- ✓ Mejor soporte que los proveedores comerciales: tiene una importante comunidad de profesionales.
- ✓ El código fuente está disponible para todos de manera gratuita.
- ✓ Multiplataforma: PostgreSQL está disponible en varias plataformas como son Linux y Windows.
- **MySQL.:** es un software libre, licenciado bajo la licencia Dual. Es un sistema de administración de bases de datos para bases de datos relacionales. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Fue escrito en C y C++ y se destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más empleados como PHP, Perl y Java, y su adaptación a distintos sistemas operativos. Es un sistema gestor de bases de datos muy rápido, fiable y fácil de usar. MySQL presenta como desventaja, que, una vez eliminada una tabla, no se eliminan automáticamente los privilegios concebidos sobre la misma. Debe usarse explícitamente un comando REVOKE para eliminar los privilegios de una tabla, acción que resulta tediosa para los usuarios con permiso de administración para la base de datos (Avitia, 2019).

Características fundamentales de MySQL:

- ✓ Está desarrollado en C/C++.
- ✓ Se distribuyen ejecutables para cerca de diecinueve plataformas diferentes.
- ✓ La API se encuentra disponible en C, C++, Eiffel, Java, Perl, PHP, Python, Ruby y TCL.
- ✓ Está optimizado para equipos de múltiples procesadores.
- ✓ Es muy destacable su velocidad de respuesta.
- ✓ Se puede utilizar como cliente-servidor o incrustado en aplicaciones.
- ✓ Cuenta con un rico conjunto de tipos de datos.
- ✓ Soporta múltiples métodos de almacenamiento de las tablas, con prestaciones y rendimiento diferentes para poder optimizar el SGBD a cada caso concreto.

- ✓ Su administración se basa en usuarios y privilegios.
- ✓ Se tiene constancia de casos en los que maneja cincuenta millones de registros, sesenta mil tablas y cinco millones de columnas.
- ✓ Los mensajes de error pueden estar en español y hacer ordenaciones correctas con palabras acentuadas o con la letra “ñ”.
- ✓ Es altamente confiable en cuanto a estabilidad se refiere.

Para el desarrollo de la propuesta de solución se selecciona MySQL en su versión 5.7 por su velocidad de procesamiento en operaciones y garantizar un mayor rendimiento cuando se consultan las bases de datos, su bajo consumo ante servidores con bajas prestaciones y su facilidad de uso en la administración.

1.6 Conclusiones parciales

El análisis de los referentes teóricos acerca del proceso de gestión de pasajes en territorio nacional por Viazul, 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 demostró la necesidad de desarrollar una aplicación web ya que hoy se carece de una que muestre un expediente de viaje por cada persona que haga uso de Viazul en el centro. Además, se definió para la elaboración de la propuesta de solución el empleo de la metodología de software XP y herramientas libres.

Capítulo 2 Solución propuesta

2.1 Introducción

En el presente capítulo se describe el flujo actual de los procesos involucrados en el pago de pasaje en divisa mediante voucher para ser utilizado en la empresa de ómnibus: Viazul. Se identifican las funcionalidades y características que tendrá la propuesta de solución, además, se exponen los patrones de diseño y de arquitectura que serán utilizados en el trabajo. Los artefactos correspondientes al análisis y el diseño del software según la metodología de desarrollo de software definida para guiar el desarrollo de la solución se verán también en el presente capítulo.

2.2 Descripción general de la propuesta de solución

El pago de los voucher por Viazul en moneda libremente convertible cubana CUC, se realiza en la Universidad de las Ciencias Informáticas (UCI) de forma no informatizada, lo que trae consigo deficiencias como son:

- ✓ Se desconoce el estado en que se encuentra un pago de pasaje por anticipo.
- ✓ Demoras en la emisión de reportes, no es posible asociar los gastos de este servicio a un área específica de la universidad.
- ✓ No se tiene control de la cantidad de los datos del viajero ya que no se tiene un registro de los mismos que guarde su nombre, destino, fecha, gasto asociado y la frecuencia de solicitud ni el motivo por el cual es solicitado el servicio.
- ✓ Se evidencia deficiencias es el tema de la seguridad de la información ya que esta es almacenada en un Excel al que puede tener acceso personal no autorizado.

Por todos estos motivos la propuesta de solución plantea la informatización del proceso de utilización de voucher por Viazul para la compra de pasaje en territorio nacional en CUC. Permitiendo a los jefes de áreas de la Universidad y específicamente a la Dirección General de Economías (DGE), así como la Dirección de Contabilidad y Finanzas (DCF) quienes operan este proceso contable, tener un mejor manejo y control del gasto de voucher en CUC.

2.3 Descripción del sistema

El sistema propuesto será una herramienta útil para la gestión de la información referente al registro y control de pasajes en territorio nacional en CUC mediante la empresa Viazul a través de voucher. Este sistema permitirá:

- ✓ Crear un expediente con los datos del viajero.
- ✓ Acceder a la cantidad de veces que una persona hace uso de voucher en divisa.
- ✓ Acceder a la cantidad de veces que un área ha ejecutado una solicitud.
- ✓ Acceder al estado del voucher.
- ✓ Controlar el gasto del voucher sobre la forma de pago de cheque por el cual se deposita en la Empresa de Viazul del anticipo para ser utilizado.
- ✓ Generar reportes de informaciones sobre el uso de un voucher con todos los datos asociados a este: Factura, Cheque y consumo.

2.4 Descripción de proceso de negocio

Un proceso de negocio (Business Process) no es más que un conjunto de actividades que se realizan en coordinación en entorno organizacional y técnico. Estas actividades, en su conjunto, ayudan a alcanzar un determinado objetivo de negocio. Cada proceso de negocio es realizado por una única organización, pero puede interactuar con procesos de otras organizaciones (*HERRAMIENTAS PARA LA GESTIÓN DE PROCESOS DE NEGOCIO Y SU RELACIÓN CON EL CICLO DE VIDA DE LOS PROCESOS DE NEGOCIO, 2014*).

La (Business Process Management) BPM es una forma de mejorar la visibilidad que se tiene del negocio y de dotar a las organizaciones con habilidades y herramientas que les permitan responder rápidamente a los cambios. Los beneficios que obtienen las organizaciones al implementar la BPM son: mejor entendimiento y visibilidad de los procesos, mejora en el manejo de las excepciones y los errores, ahorro de tiempo y reducción de costos por el incremento en la eficiencia de las operaciones, mejora en el desempeño de los empleados, y posibilidad de mejora de los procesos con base en evidencias obtenidas por el monitoreo de la ejecución de los mismos (*HERRAMIENTAS PARA LA GESTIÓN DE PROCESOS DE NEGOCIO Y SU RELACIÓN CON EL CICLO DE VIDA DE LOS PROCESOS DE NEGOCIO, 2014*).

El siguiente diagrama describe el proceso general del voucher:

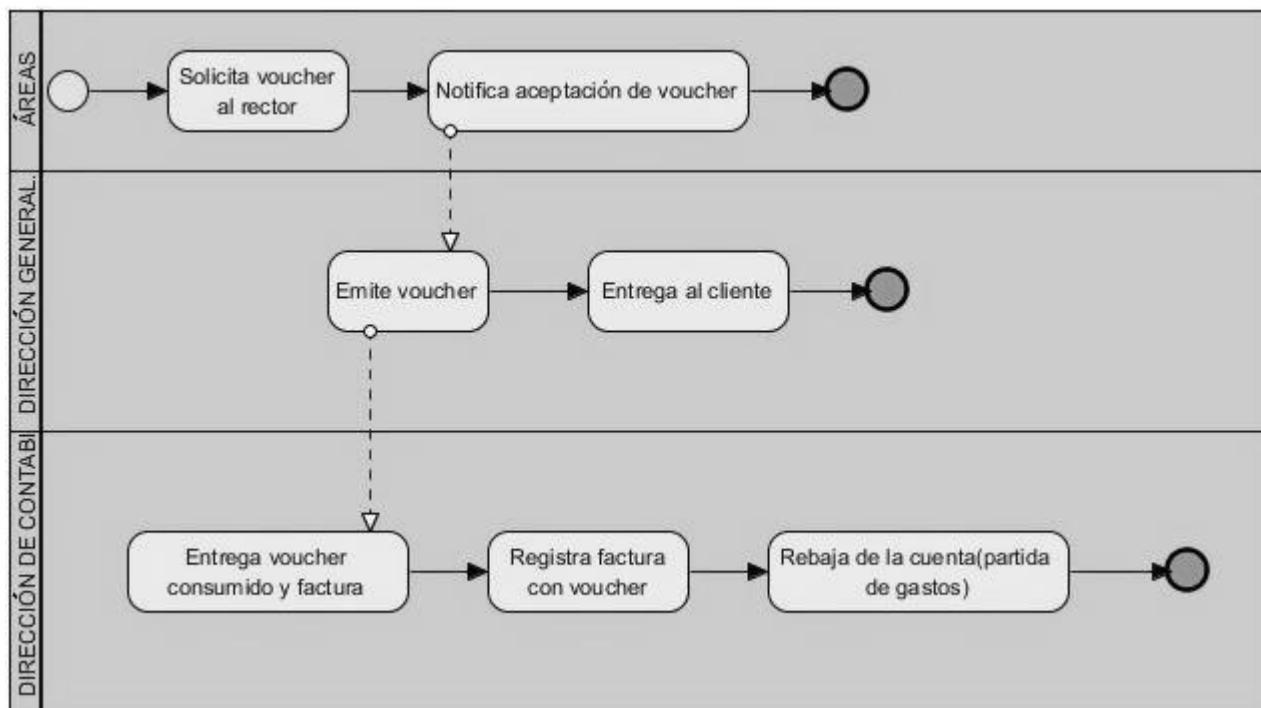


Figura 1. Diagrama del proceso general del voucher.

El proceso comienza cuando un área de la UCI solicita al rector un voucher para un trabajador o un estudiante con el objetivo de que este viaje a provincia por motivo de salud, trabajo o evento. Posteriormente de hecha la solicitud al rector, este notifica al responsable del área la aceptación para el uso del voucher (que puede ser de ida y regreso o solo una de las dos opciones).

La persona que solicitó el voucher presenta en la DGE la aprobación del rector y esta se encarga de emitir el voucher que no es más que un modelo con el cual el cliente se presenta en la agencia de viajes Viazul y adquiere un pasaje con el cual viajará a provincia.

Luego que el cliente utilice el voucher, a su regreso entregará en un término de no más de 72 horas, el comprobante de pasaje, que no es más que una copia del pasaje comprado en la agencia que es con el cual aborda a los ómnibus. Luego la DGE entregará en la DCF el comprobante de pasajes, copia del

voucher emitido y la factura que hace referencia al voucher utilizado. Todo este proceso sucede al finalizar cada mes.

La DCF, en el departamento de economía, registra la factura con el voucher y rebaja del saldo existente en la cuenta. Esta cuenta está asociada a una partida de gasto definida en el presupuesto de la Universidad que lleva por nombre: Pasaje en Territorio Nacional en CUC, donde a inicios de cada año se le planifica un presupuesto para luego ser ejecutado durante todo el año, de esta forma concluye el proceso de emisión del voucher.

Llenar Datos del voucher: el trabajador de la DGE autorizado mediante resolución rectoral del centro para gestionar el uso de los voucher y que se encuentra autorizado, se encarga de incorporar los datos correspondientes a un voucher.

El siguiente diagrama describe el proceso de llenado de los datos de un voucher:

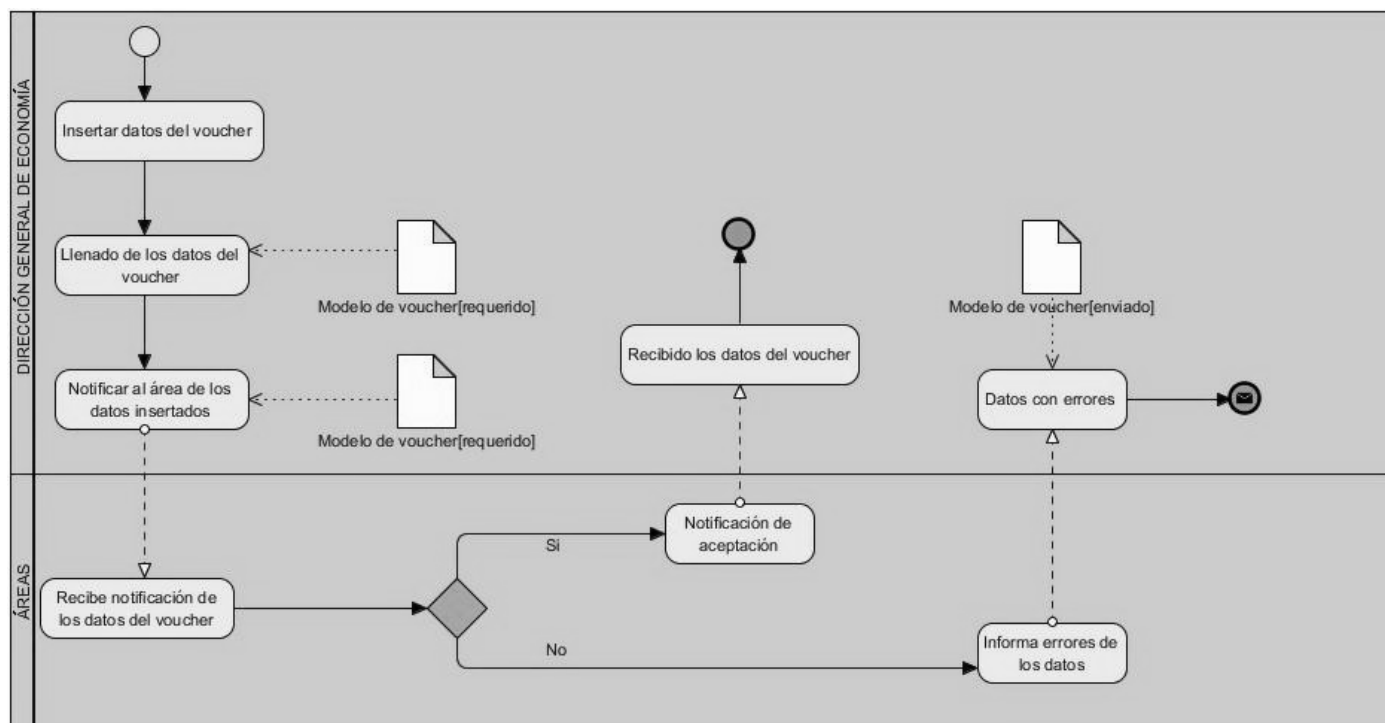


Figura 2. Diagrama de llenado de los datos del voucher.

Notificar al área los datos del voucher: Una vez llenado los datos del voucher, se le notifica al área que puede informar a la persona que utilizará el voucher, que debe presentarse en la DGE para recoger el voucher y presentarse con 72 horas antes de su viaje en la empresa Viazul para que compra el pasaje.

2.4.1 Actores y trabajadores que intervienen en los procesos del negocio

Estudiante, profesor o trabajador: Persona que solicita el servicio. Este inicia todos los procesos.

Jefe de Área: Persona encargada de presentar ante el rector la solicitud del servicio, para ser consumido por un miembro del área de la cual él es jefe. Esta se encarga de la gestión de autorizo presentando las causas.

Rector Persona que interviene en el proceso autorizando el pago del pasaje mediante el uso de voucher.

Especialista de la DGE: Persona que interviene en el proceso de emitir el modelo de voucher.

Especialistas de la DCF: Personas que intervienen en el proceso, encargándose de gestionar mediante el sistema contable Assets, el estado de las cuentas destinadas al consumo de este servicio y el manejo de los vouchers.

2.5 Requisitos del software

Los requisitos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requisitos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema...El proceso de descubrir, analizar, documentar, y verificar estos servicios y restricciones se denomina ingeniería de requisitos...El término requisito no se utiliza de una forma constante en la industria de software. En algunos casos, un requisito es simplemente una declaración abstracta de alto nivel de un servicio q debe proporcionar el sistema o una restricción de éste. En el otro extremo, es una definición detallada y formal de una función del sistema (Sommerville, 2011). Los funcionales declaran los servicios que debe brindar el sistema, de la manera que éste debe reaccionar y funcionar ante una entrada o situación en particular y los no funcionales son las restricciones de los servicios o funciones ofrecidas por el sistema (Capó Marrero, y otros, 2014).

2.5.1 Requisitos Funcionales (RF)

Los requisitos funcionales (RF) que describen las funcionalidades que el sistema debe cumplir son:

| No. RF | Nombre del RF |
|--------|---|
| RF-1 | Gestionar usuario. <ul style="list-style-type: none">• Insertar Usuario• Modificar Usuario• Eliminar Usuario• Listar Usuario |
| RF-2 | Autenticar Usuario. |
| RF-3 | Gestionar Cuenta. <ul style="list-style-type: none">• Insertar Cuenta• Modificar Cuenta• Eliminar Cuenta• Listar Cuenta |
| RF-4 | Gestionar voucher. <ul style="list-style-type: none">• Insertar voucher• Modificar voucher• Eliminar voucher• Listar voucher |
| RF-5 | Buscar voucher. |
| RF-6 | Gestionar Cheque. <ul style="list-style-type: none">• Insertar Cheque• Modificar Cheque• Eliminar Cheque• Listar Cheque. |
| RF-7 | Gestionar Factura. <ul style="list-style-type: none">• Insertar Factura• Modificar Factura |

| | |
|-------|--|
| | <ul style="list-style-type: none"> • Eliminar Factura • Listar Factura |
| RF-8 | Gestionar Reintegro. <ul style="list-style-type: none"> • Insertar Reintegro • Modificar Reintegro • Eliminar Reintegro • Listar Reintegro |
| RF-9 | Registrar consumo del voucher. |
| RF-10 | Generar reportes de estado. |
| RF-11 | Exportar reporte a pdf. |
| RF-12 | Exportar reporte a Excel. |

Tabla 2. Requisitos funcionales.

2.5.2 Requisitos no Funcionales (RNF)

| No. RNF | De tipo | Nombre del RNF |
|---------|---------------|---|
| RNF-1 | Usabilidad | El sistema debe ser dinámico y de fácil navegación, pues se realizará con campos y botones fáciles y sugerentes para los usuarios. |
| RNF-2 | Confiabilidad | El sistema validará la entrada de datos para que no se inserten datos erróneos. |
| RNF-3 | Seguridad | Se garantizará la integridad y confidencialidad de la información mediante mecanismos de control de acceso no autorizados utilizando: usuario, contraseña y definiendo niveles de acceso para cada usuario, de manera que cada usuario pueda tener disponible solamente las opciones relacionadas con su actividad y tenga datos de acceso propios. |
| RF-4 | | El sistema debe permitir que cuando se borre cualquier información pueda existir una opción de advertencia antes realizar la acción. |

| | | |
|-------|--|---|
| RNF-5 | Apariencia o interfaz externa | Se debe garantizar una correcta organización de la información para permitir una adecuada interpretación. |
| RNF-6 | Requisitos legales, derecho de autor y otros | La mayoría de las herramientas de desarrollo deben ser libres y las que no, sus licencias deben estar avaladas. Además, debe poseer documentación legal de uso como declaración de autoría. |

Tabla 3. Requisitos no funcionales.

2.6 Historias de usuarios

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas (Extreme Programming Installed, 2012).

Estas se usan para estimar el tiempo y plan de lanzamiento, también dirigen la creación de las pruebas de aceptación. Se caracterizan por ser independientes una de otras, negociables, valoradas por los clientes o usuarios, estimables, pequeñas y verificables.

A continuación, se definen las HU que tienen mayor importancia en la solución propuesta, el resto se pueden encontrar en los anexos.

Historias de Usuarios del Sistema

| Historia de Usuario | |
|--|--|
| Número: HU_1 | Nombre Historia de Usuario: Gestionar Usuario |
| Modificación de Historia de Usuario Número: 1 | |
| Usuario: Javier Leyva Tamayo | Iteración Asignada: 1 |
| Prioridad en Negocio: Alta | Puntos Estimados: 2 días. |
| Riesgo en Desarrollo: Baja | Puntos Reales: 2 días. |
| <p>Descripción: La HU inicia cuando el usuario selecciona en el Menú Principal la opción Usuarios. El sistema muestra un listado con los usuarios registrados previamente además de la opción de insertar un nuevo usuario. Cada tupla contendrá las opciones de ser modificada o eliminada, finalizando así la HU.</p> | |
| <p>Observaciones:</p> <ul style="list-style-type: none"> ✓ El usuario debe tener permiso de acceso al sistema. ✓ El usuario debe tener rol de administrador. ✓ Si el usuario no tiene acceso en el sistema debe aparecer un mensaje de error: "Usted no tiene permiso de acceso al sistema". | |

Tabla 4. HU_1 Gestionar Cuentas de usuario.

| Historia de Usuario | |
|--|---|
| Número: HU_2 | Nombre Historia de Usuario: Autenticar Usuario |
| Modificación de Historia de Usuario Número: 1 | |
| Usuario: Javier Leyva Tamayo | Iteración Asignada: 1 |

| | |
|---|----------------------------------|
| Prioridad en Negocio: Media | Puntos Estimados: 2 días. |
| Riesgo en Desarrollo: Baja | Puntos Reales: 2 días. |
| <p>Descripción: La HU inicia cuando el usuario se autentica en el sistema, para posteriormente realizar determinadas operaciones en él. Para que el usuario pueda autenticarse en el sistema tiene que entrar los siguientes datos:</p> <ul style="list-style-type: none"> ✓ Usuario. ✓ Contraseña. <p>Luego selecciona la opción: “Entrar”, finalizando así la HU.</p> | |
| <p>Observaciones:</p> <ul style="list-style-type: none"> ✓ El usuario debe tener permiso de acceso al sistema. ✓ Si el usuario no tiene acceso en el sistema debe aparecer un mensaje de error: “Usted no tiene permiso de acceso al sistema”. | |

Tabla 5. HU_2 Autenticar usuario.

| Historia de Usuario | |
|--|--|
| Número: HU_3 | Nombre Historia de Usuario: Gestionar Cuentas |
| Modificación de Historia de Usuario Número: 1 | |
| Usuario: Javier Leyva Tamayo | Iteración Asignada: 1 |
| Prioridad en Negocio: Muy Alta | Puntos Estimados: 2 días. |
| Riesgo en Desarrollo: Media | Puntos Reales: 2 días. |

| |
|--|
| <p>Descripción: La HU inicia cuando el usuario selecciona la opción “Cuentas” en el menú principal, luego el sistema permitirá Insertar, Modificar o Eliminar, en cualquiera de los casos el usuario debe seleccionar que operación realizar dando clic en el botón correspondiente a la operación.</p> |
| <p>Observaciones:</p> <ul style="list-style-type: none"> ✓ Se debe estar autenticado en el sistema. ✓ Todos los campos son obligatorios. |

Tabla 6. HU_3 Gestionar Cuentas.

| Historia de Usuario | |
|---|--|
| Número: HU_4 | Nombre Historia de Usuario: Gestionar voucher |
| Modificación de Historia de Usuario Número: 1 | |
| Usuario: Javier Leyva Tamayo | Iteración Asignada: 1 |
| Prioridad en Negocio: Muy Alta | Puntos Estimados: 2 días. |
| Riesgo en Desarrollo: Media | Puntos Reales: 2 días. |
| <p>Descripción: La HU inicia cuando el usuario selecciona la opción “voucher” en el menú principal, luego el sistema muestra un listado con los voucher registrados previamente además de la opción de insertar un nuevo voucher. Cada tupla contendrá las opciones de ser modificada o eliminada, en cualquiera de los casos el usuario debe seleccionar que operación realizar dando clic en el botón correspondiente a la operación, finalizando así la HU.</p> | |
| <p>Observaciones:</p> <ul style="list-style-type: none"> ✓ Se debe estar autenticado en el sistema. | |

Tabla 7. HU_4 Gestionar voucher.

| Historia de Usuario | |
|---|---|
| Número: HU_5 | Nombre Historia de Usuario: Buscar voucher |
| Modificación de Historia de Usuario Número: 1 | |
| Usuario: Javier Leyva Tamayo | Iteración Asignada: 2 |
| Prioridad en Negocio: Muy Alta | Puntos Estimados: 2 días. |
| Riesgo en Desarrollo: Media | Puntos Reales: 2 días. |
| <p>Descripción: La HU inicia cuando el usuario selecciona la opción “voucher” en el menú principal, luego “aparecerá un campo para introducir el nombre de la persona que consumió el vouchers y posteriormente el sistema mostrara todos los datos del vouchers correspondiente, finalizando así la HU.</p> | |
| <p>Observaciones:</p> <ul style="list-style-type: none"> ✓ Todos los campos son obligatorios. ✓ Se debe estar autenticado en el sistema. | |

Tabla 8. HU_5 Buscar vouchers.

| Historia de Usuario | |
|--|---|
| Número: HU_6 | Nombre Historia de Usuario: Gestionar Cheque |
| Modificación de Historia de Usuario Número: 1 | |
| Usuario: Javier Leyva Tamayo | Iteración Asignada: 1 |
| Prioridad en Negocio: Muy Alta | Puntos Estimados: 2 días. |

| | |
|---|-------------------------------|
| Riesgo en Desarrollo: Media | Puntos Reales: 2 días. |
| <p>Descripción: La HU inicia cuando el usuario selecciona la opción “Cheque” en el menú principal, luego el sistema muestra un listado con los cheques registrados previamente además de la opción de insertar un nuevo cheque. Cada tupla contendrá las opciones de ser modificada o eliminada, en cualquiera de los casos el usuario debe seleccionar que operación realizar dando clic en el botón correspondiente a la operación, finalizando así la HU.</p> | |
| <p>Observaciones:</p> <p>✓ Se debe estar autenticado en el sistema.</p> | |

Tabla 9. HU_6 Gestionar Cheque.

| Historia de Usuario | |
|---|--|
| Número: HU_7 | Nombre Historia de Usuario: Gestionar Factura |
| Modificación de Historia de Usuario Número: 1 | |
| Usuario: Javier Leyva Tamayo | Iteración Asignada: 1 |
| Prioridad en Negocio: Muy Alta | Puntos Estimados: 2 días. |
| Riesgo en Desarrollo: Media | Puntos Reales: 2 días. |
| <p>Descripción: La HU inicia cuando el usuario selecciona la opción “Factura” en el menú principal, luego el sistema muestra un listado con las facturas registrados previamente además de la opción de insertar una nueva factura. Cada tupla contendrá las opciones de ser modificada o eliminada, en cualquiera de los casos el usuario debe seleccionar que operación realizar dando clic en el botón correspondiente a la operación, finalizando así la HU.</p> | |
| <p>Observaciones:</p> <p>✓ Se debe estar autenticado en el sistema.</p> | |

Tabla 10. HU_7 Gestionar Factura.

| Historia de Usuario | |
|--|--|
| Número: HU_8 | Nombre Historia de Usuario: Gestionar Reintegro |
| Modificación de Historia de Usuario Número: 1 | |
| Usuario: Javier Leyva Tamayo | Iteración Asignada: 1 |
| Prioridad en Negocio: Muy Alta | Puntos Estimados: 2 días. |
| Riesgo en Desarrollo: Media | Puntos Reales: 2 días. |
| <p>Descripción: La HU inicia cuando el usuario selecciona la opción “Reintegro” en el menú principal, luego el sistema muestra un listado con los reintegros registrados previamente además de la opción de insertar un nuevo reintegro. Cada tupla contendrá las opciones de ser modificada o eliminada, en cualquiera de los casos el usuario debe seleccionar que operación realizar dando clic en el botón correspondiente a la operación, finalizando así la HU.</p> | |
| <p>Observaciones:</p> <ul style="list-style-type: none"> ✓ Se debe estar autenticado en el sistema. | |

Tabla 11. HU_8 Gestionar Factura.

| Historia de Usuario | |
|--|--|
| Número: HU_9 | Nombre Historia de Usuario: Registrar consumo del voucher |
| Modificación de Historia de Usuario Número: 1 | |
| Usuario: Javier Leyva Tamayo | Iteración Asignada: 2 |
| Prioridad en Negocio: Media | Puntos Estimados: 2 días. |

| | |
|---|-------------------------------|
| Riesgo en Desarrollo: Baja | Puntos Reales: 2 días. |
| <p>Descripción: La HU inicia cuando el usuario accede a la página principal del sistema, y selecciona la opción: “voucher”. El sistema muestra los datos correspondientes a los vouchers almacenados en el sistema, un campo para que el usuario pueda introducir el nombre de la persona que consumió el vouchers, una vez seleccionado el voucher se selecciona la opción “Editar” que permitirá al usuario cambiar el estado del voucher ha consumido. El sistema seguidamente debe mostrar los datos del vouchers modificado, finalizando así la HU.</p> | |
| <p>Observaciones:</p> <ul style="list-style-type: none"> ✓ Se debe estar autenticado en el sistema. ✓ Todos los campos son obligatorios. | |

Tabla 12. HU_9 Registrar consumo del vouchers.

| Historia de Usuario | |
|--|--|
| Número: HU_10 | Nombre Historia de Usuario: Generar Reportes de estado. |
| Modificación de Historia de Usuario Número: 1 | |
| Usuario: Javier Leyva Tamayo | Iteración Asignada: 2 |
| Prioridad en Negocio: Media | Puntos Estimados: 2 días. |
| Riesgo en Desarrollo: Baja | Puntos Reales: 2 días. |
| <p>Descripción: Descripción: La HU inicia cuando el usuario accede a la página principal del sistema, y selecciona la opción: “: Generar Reportes”. El sistema muestra los elementos por los cuales se pueden realizar los reportes para que el usuario seleccione los elementos que contendrá el reporte. El sistema</p> | |

| |
|--|
| seguidamente debe mostrar los datos almacenados correspondientes a la elección hecha por el usuario, y luego el cliente da un clic en el botón “: Generar Reportes”, finalizando así la HU |
| <p>Observaciones:</p> <ul style="list-style-type: none"> ✓ Se debe estar autenticado en el sistema. ✓ Todos los campos son obligatorios. |

Tabla 13. HU_10 Generar Reportes.

| Historia de Usuario | |
|--|--|
| Número: HU_11 | Nombre Historia de Usuario: Exportar Reportes a PDF |
| Modificación de Historia de Usuario Número: 1 | |
| Usuario: Javier Leyva Tamayo | Iteración Asignada: 2 |
| Prioridad en Negocio: Media | Puntos Estimados: 2 días. |
| Riesgo en Desarrollo: Baja | Puntos Reales: 2 días. |
| <p>Descripción: La HU inicia cuando el usuario accede a la página principal del sistema, y selecciona la opción: “: Exportar Reportes a PDF”. El sistema muestra los elementos por los cuales se pueden realizar los reportes para que el usuario seleccione los elementos que contendrá el reporte. El sistema seguidamente debe mostrar los datos almacenados correspondientes a la selección hecha por el usuario, y luego el cliente da un clic en el botón “: Exportar Reportes a PDF”, finalizando así la HU.</p> | |
| <p>Observaciones:</p> <ul style="list-style-type: none"> ✓ Se debe estar autenticado en el sistema. | |

Tabla 14. HU_11 Exportar Reportes a PDF.

| Historia de Usuario | |
|---|--|
| Número: HU_12 | Nombre Historia de Usuario: Exportar Reportes a Excel |
| Modificación de Historia de Usuario Número: 1 | |
| Usuario: Javier Leyva Tamayo | Iteración Asignada: 2 |
| Prioridad en Negocio: Media | Puntos Estimados: 2 días. |
| Riesgo en Desarrollo: Baja | Puntos Reales: 2 días. |
| <p>Descripción: La HU inicia cuando el usuario accede a la página principal del sistema, y selecciona la opción: “: Exportar Reportes a Excel”. El sistema muestra los elementos por los cuales se pueden realizar los reportes para que el usuario conozca los elementos que contendrá el reporte. El sistema seguidamente debe mostrar los datos almacenados correspondientes a la selección hecha por el usuario, y luego el cliente da un clic en el botón “: Exportar Reportes a Excel”, finalizando así la HU.</p> | |
| <p>Observaciones:</p> <p>✓ Se debe estar autenticado en el sistema.</p> | |

Tabla 15. HU_12 Exportar Reportes a Excel.

2.7 Planificación

El uso de la palabra planificación (en cualquier contexto) es un anatema para algunos desarrolladores web que no planifican; sólo arrancan, con la esperanza de que surja una webapp asesina. Un enfoque más disciplinado reconoce que la planificación establece un mapa de ruta para todo el trabajo que va después. Vale la pena el esfuerzo (PRESSMAN, 2010).

2.7.1 Plan de entrega

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos

sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días (Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP), 2016).

Se propone el siguiente plan de entregas para la solución propuesta:

| Entregables | Iteración | Fin de la iteración |
|--------------------------------|-----------|---------------------|
| Gestionar Usuario | 1 | Febrero 2019 |
| Autenticar Usuario | 1 | Febrero 2019 |
| Gestionar Cuenta | 1 | Febrero 2019 |
| Gestionar vouchers | 1 | Febrero 2019 |
| Buscar vouchers | 1 | Febrero 2019 |
| Gestionar Cheque | 1 | Febrero 2019 |
| Gestionar Factura | 1 | Febrero 2019 |
| Registrar consumo del vouchers | 2 | Marzo 2019 |
| Gestionar Reintegros | 2 | Marzo 2019 |
| Generar reportes | 2 | Abril 2019 |
| Mostrar Historial de Acciones | 2 | Abril 2019 |
| Exportar reportes a PDF | 2 | Mayo 2019 |
| Exportar reportes a Excel | 2 | Mayo 2019 |

Tabla 16. Plan de Entrega.

2.7.2 Plan de iteraciones

El proceso de desarrollo de software según la metodología XP se caracteriza por ser iterativo e incremental, por lo que se realizan varias iteraciones sobre el sistema antes de su fase de producción. Para elaboración de este plan se debe tomar en cuenta las HU no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas (Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP), 2016). El sistema de gestión de pago de pasaje por vouchers en divisa se realizará en 2 iteraciones como se demuestra en la tabla.

| Iteración | Historia de Usuario | Semanas estimadas |
|-----------|--------------------------------|-------------------|
| 1 | Gestionar Usuario | 8 |
| | Autenticar Usuario | |
| | Gestionar Cuenta | |
| | Gestionar voucher | |
| | Buscar voucher | |
| | Gestionar Cheque | |
| | Gestionar Factura | |
| 2 | Registrar consumo del vouchers | 8 |
| | Gestionar Reintegros | |
| | Generar reportes | |
| | Mostrar Historial de Acciones | |
| | Exportar reportes a PDF | |
| | Exportar reportes a Exel | |

Tabla 17. Plan de Iteraciones.

2.8 Diseño

El diseño de software agrupa el conjunto de principios, conceptos y prácticas que llevan al desarrollo de un sistema o producto de alta calidad. Los principios de diseño establecen una filosofía general que guía

el trabajo de diseño que debe ejecutarse. Deben entenderse los conceptos de diseño antes de aplicar la mecánica de éste, y la práctica del diseño en sí lleva a la creación de distintas representaciones del software que sirve como guía para la actividad de construcción que siga (S. Pressman, 2010).

2.8.1 Descripción de la arquitectura

Cuando se piensa en la arquitectura de una construcción, llegan a la mente muchos atributos distintos. En el nivel más sencillo, se considera la forma general de la estructura física. Pero, en realidad, la arquitectura es mucho más que eso. Es la manera en la que los distintos componentes del edificio se integran para formar un todo cohesivo. Es la forma en la que la construcción se adapta a su ambiente y se integra a los demás edificios en la vecindad. Es el grado en el que el edificio cumple con su propósito y en el que satisface las necesidades del propietario. Es la sensación estética de la estructura —el efecto visual de la edificación— y el modo en el que se combinan texturas, colores y materiales para crear la fachada en el exterior y el “ambiente de vida” en el interior. Es los pequeños detalles: diseño de las lámparas, tipo de piso, color de las cortinas... la lista es casi interminable. Y, finalmente, es arte (S. Pressman, 2010).

Para el desarrollo del sistema de gestión de la información referente al control de pago de voucher de Viazul en divisa en la UCI, se utilizará el patrón de arquitectura de software Modelo Vista Controlador (MVC). Modelo Vista Controlador es el concepto de encapsular datos y procesamiento (modelo) aislándolo de su manipulación (controlador) y presentación (vista) que forman parte de la interfaz de usuario (MVC DESDE SMALLTALK A LA WEB, 2004).

- **Modelo (Model):** Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada (Utilización del Patrón Modelo – Vista – Controlador (MVC) en el diseño de software educativos. , 2006).
- **Vista (View):** Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador (Utilización del Patrón Modelo – Vista – Controlador (MVC) en el diseño de software educativos. , 2006).
- **Controlador (Controller):** Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio (“service requests”) para el modelo o la vista (Utilización del Patrón Modelo – Vista – Controlador (MVC) en el diseño de software educativos. , 2006).

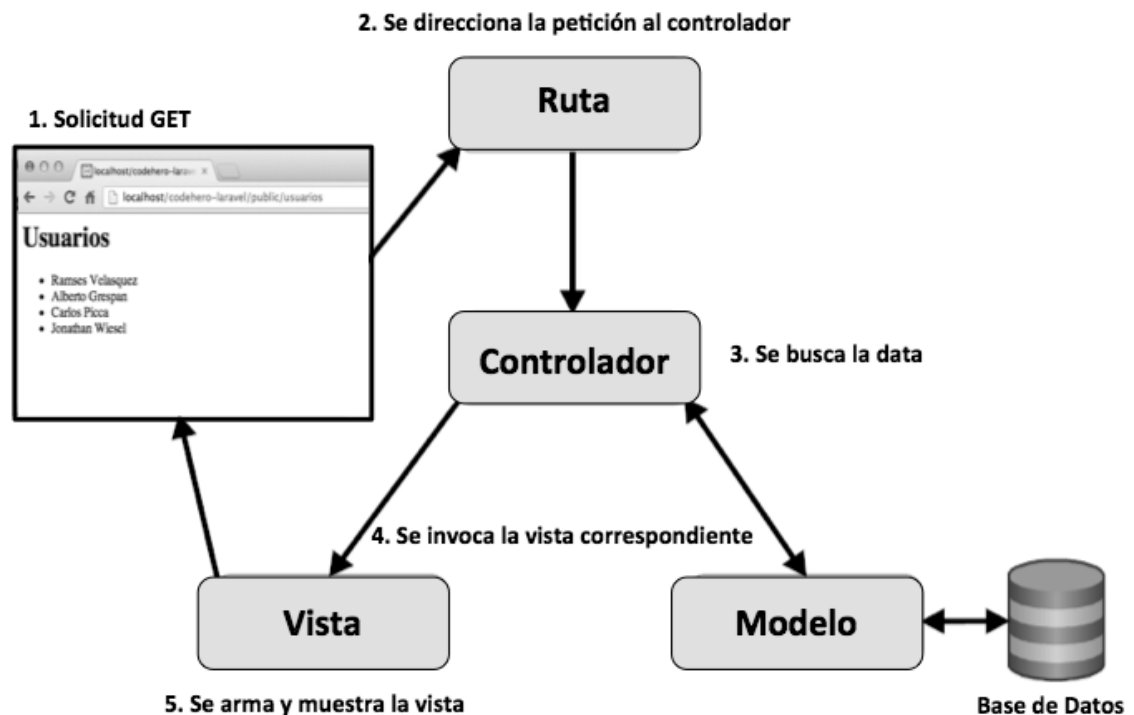


Figura 3. Modelo-Vista-Controlador (MVC).

2.8.2 Patrones de diseño

Los patrones de diseño se derivaron de las ideas introducidas por Christofer Alexander, quien sugirió que existían ciertos patrones del diseño de edificios que eran comunes e inherentemente interesantes y efectivos. El patrón es una descripción del problema y la esencia de su solución, de forma que la solución se pueda reutilizar en diferentes situaciones. El patrón no es una especificación detallada. Antes bien, puede pensarse en él como una descripción del conocimiento y experiencia acumulados. Es una solución adecuada a un problema común. Una frase del sitio web Hillside.net, que se dedica a mantener información sobre patrones, encapsula su papel en la reutilización: Los patrones y los lenguajes de patrones son formas de describir las mejores prácticas, Buenos diseños, y encapsulan la experiencia de tal forma que es posible para otros el reutilizar dicha experiencia. (Sommerville, 2011)

2.8.3 Patrones Generales de Software para Asignar Responsabilidades.

En el diseño orientado a objetos, es muy importante asignar las responsabilidades de forma correcta. Para ello se utilizan los patrones GRASP, los cuales promueven buenos principios para la realización de dicha tarea. Para el diseño de la propuesta que se desea implementar fueron empleados los siguientes:

Experto: se emplea en las clases del sistema debido a que cada una de las mismas es experta, pues contienen la información necesaria para cumplir con su responsabilidad. El uso de este patrón trae consigo que se conserve el encapsulamiento debido a que las clases se valen de su propia información para cumplir con las tareas asignadas, soportando además un bajo acoplamiento.

Creador: este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos. Se pone de manifiesto en las clases controladoras del sistema, las cuales tienen dicha responsabilidad a su cuenta, además brinda soporte a un bajo acoplamiento y mejora la reutilización. Este patrón se observa en la clase `VoucherController`.

Bajo acoplamiento: el diseño del sistema consta de este patrón pues cada clase depende lo menos posible de otra, de manera tal que una de ellas solo recurre a otra en caso de que exista referencia dentro de sus atributos, lo cual permite que el sistema sea mucho más robusto y de fácil mantenimiento. Se evidencia el uso de este patrón en la relación existente entre las clases `CuentaController`, `EntityManager` y `EntityRepository`.

Alta cohesión: el sistema está diseñado de forma tal que las clases tienen responsabilidades estrechamente relacionadas y no realizan un trabajo excesivo, lo cual permite simplificar el mantenimiento, generar el bajo acoplamiento y aumentar la capacidad de reutilización de las mismas.

Los patrones del diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

2.9 Modelo conceptual

Con la construcción del modelo conceptual se logra una mejor comprensión del dominio del problema. Un modelo conceptual no es más que una representación visual de los conceptos u objetos del mundo real que son significativos para el problema o el área que se analiza, representando las clases conceptuales,

no los componentes de software. Puede verse como un modelo que comunica a los interesados, cuáles son los términos importantes y cómo se relacionan entre sí (Larman, 1999).

El modelo conceptual propuesto anteriormente describe como el fondo de anticipo es consumido al emitir un cheque que a su vez es consumido tras a la adquisición de un talonario de voucher. Cada voucher tiene asociado una factura las cuales son pagadas mediante liquidaciones.

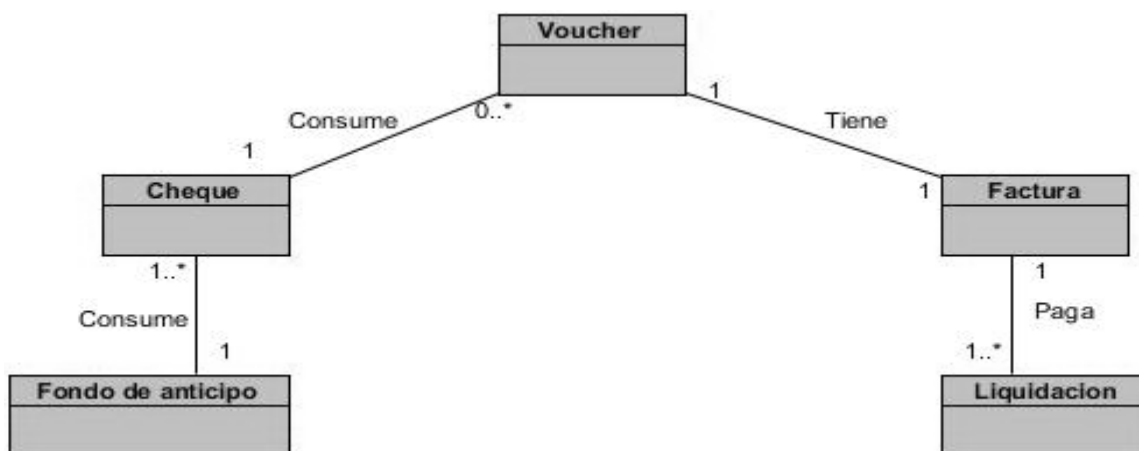


Figura 4. Modelo conceptual.

2.10 Modelo de base de datos

Para el desarrollo de cualquier aplicación es fundamental el correcto diseño de la base de datos dado que este describe el modelo lógico y físico de los datos persistentes en el sistema. A continuación, se muestra un segmento del diagrama de base de datos que genera el sistema a implementar.

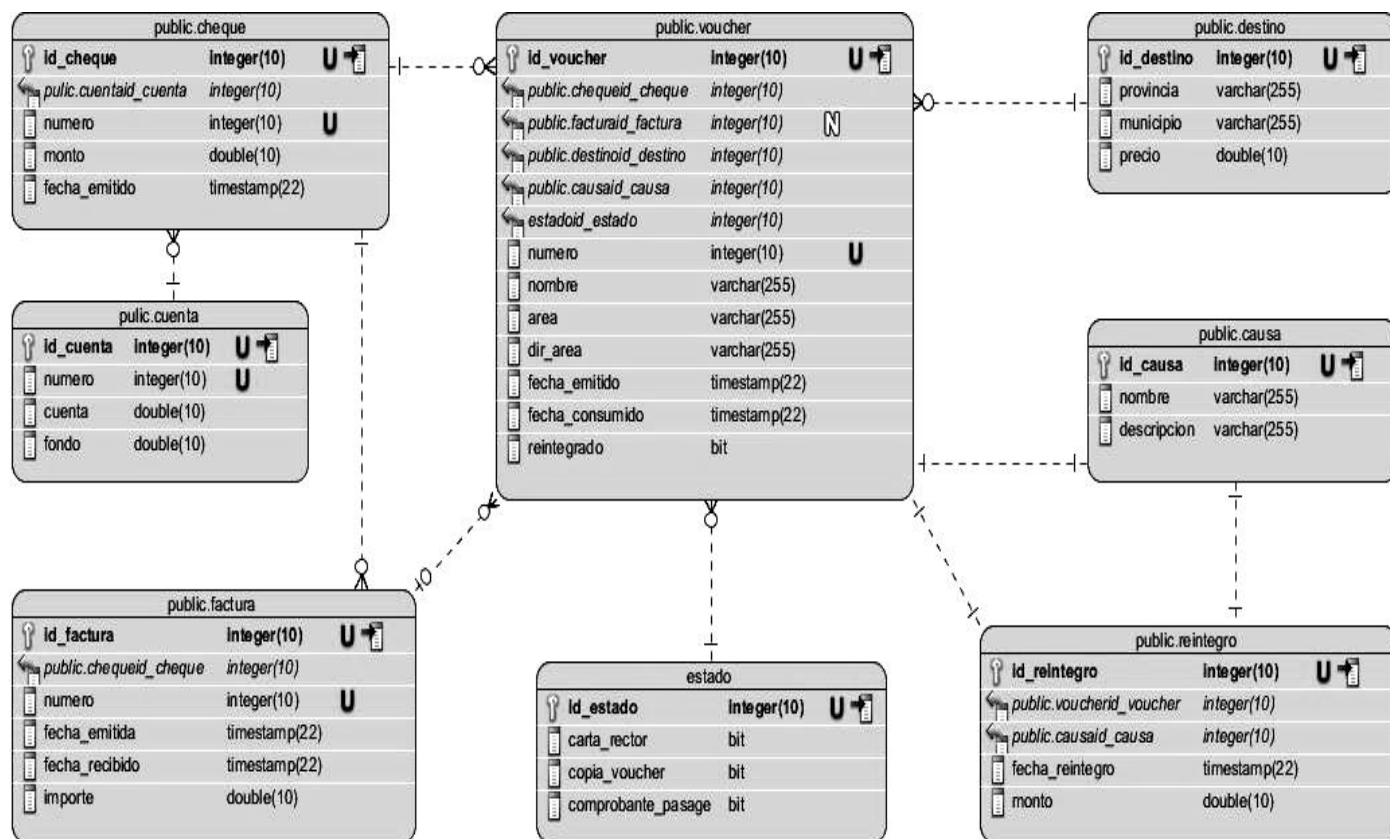


Figura 5. Diagrama de base de datos.

2.11 Conclusiones parciales

A partir de las deficiencias y necesidades identificadas en el desarrollo actual del proceso de registro y control de pasaje en territorio nacional por Viazul y después de realizar una descripción completa del comportamiento del sistema a desarrollar y una identificación de los requisitos funcionales y no funcionales con los que debería contar el sistema a partir de los cuales se incluyen un conjunto de Historias de Usuario q describen todas las interacciones que tendrán los usuarios con el mismo, así como la creación de planes de entrega e iteración propios de la metodología de desarrollo de software utilizada(XP) se logra plantear una propuesta de solución completa así como el uso la arquitectura MVC y los patrones de diseño.

Capítulo 3 Implementación y pruebas

3.1 Introducción

En el siguiente capítulo se describen los elementos necesarios para la implementación del sistema y se modela la vista de despliegue del sistema. A su vez se confeccionan los casos de prueba a realizar al software para determinar si cumple con las funcionalidades identificadas, se ejecutan, se describe el análisis de los resultados alcanzados en las pruebas y se mencionan las acciones realizadas para resolverlas.

3.2 Estándares de codificación

Es necesario establecer un criterio fijo que proporcione reglas para la creación de nombre para variables y métodos, permitiendo una mejor lectura del software, y un mejor entendimiento del código por parte de los desarrolladores. Las pautas fundamentales definidas para la implementación son las siguientes:

3.2.1 Estructura

- La apertura de llaves para clases y funciones deben estar en la próxima línea después de su declaración.

```
public function indexAction()  
{  
    $em = $this->getDoctrine()->getManager();  
  
    $causas = $em->getRepository('Causa')->findAll();  
  
    return $this->render('causa/index', array(  
        'causas' => $causas,  
    ));  
}
```

- Las aperturas de llaves en las estructuras de control se colocan en la misma línea.

```
if ($form->isSubmitted() && $form->isValid()) {  
    $em = $this->getDoctrine()->getManager();  
    $em->persist($cheque);  
    $em->flush();  
  
    return $this->redirectToRoute( route: '/{id}/show', array('id' => $cheque->getId()));  
}
```

- El cierre de llaves para clases, funciones y estructuras de control deben estar en la próxima línea después del bloque de código.

```
public function indexAction()  
{  
    $em = $this->getDoctrine()->getManager();  
  
    $cheques = $em->getRepository( className: 'Cheque' )->findAll();  
  
    return $this->render( view: 'cheque/index', array(  
        'cheques' => $cheques,  
    ));  
}
```

- La indentación se realiza con tabulación y no con 4 espacios en blanco

```
public function showAction(Cheque $cheque)  
{  
    $deleteForm = $this->createDeleteForm($cheque);  
  
    return $this->render( view: 'cheque/show', array(  
        'cheque' => $cheque,  
        'delete_form' => $deleteForm->createView(),  
    ));  
}
```

- Se deberá agregar una línea en blanco delante de la sentencia *return*.

```
public function showAction(Cheque $cheque)
{
    $deleteForm = $this->createDeleteForm($cheque);
    return $this->render( view: 'cheque/show', array(
        'cheque' => $cheque,
        'delete_form' => $deleteForm->createView(),
    ));
}
```

- Se puede agregar líneas en blanco para mejorar la legibilidad e indicar bloques de códigos relacionados.
- Debe haber una declaración por línea.

```
public function deleteAction(Request $request, Cuenta $cuenta)
{
    $form = $this->createDeleteForm($cuenta);
    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()) {
        $em = $this->getDoctrine()->getManager();
        $em->remove($cuenta);
        $em->flush();
    }

    return $this->redirectToRoute( route: '/' );
}
```

- ¿Los archivos PHP deben utilizar solo las etiquetas de apertura `<?php` omitiendo las etiquetas de cierre `?>`.
- Utilizar la codificación UTF-8.
- Se debe utilizar un espacio en blanco después de la palabra clave en las estructuras de control.

```
if ($form->isSubmitted() && $form->isValid()) {
    $em = $this->getDoctrine()->getManager();
    $em->remove($cuenta);
    $em->flush();
}
```

- No deben existir espacios en la apertura y cierre de los paréntesis.

```
if ($form->isSubmitted() && $form->isValid()) {  
    $em = $this->getDoctrine()->getManager();  
    $em->remove($cuenta);  
    $em->flush();  
}
```

- Debe usarse un espacio en blanco entre los operadores lógicos, aritméticos, de comparación y asignación.

```
if ($form->isSubmitted() && $form->isValid()) {  
    $em = $this->getDoctrine()->getManager();  
    $em->remove($cuenta);  
    $em->flush();  
}
```

3.2.2 Convención de Nombres

- Utiliza *camelCase* y no guiones bajos, para variables, funciones y nombres de métodos,
- Utiliza guiones bajos para definir opciones, argumentos y nombres de parámetros,
- Utiliza los *namespace* para todas las clases,
- Utiliza *Symfony* como el *namespace* de primer nivel,
- Añade como sufijo *Interface* a las interfaces,
- Utiliza caracteres alfanuméricos y guiones bajos para nombres de archivos.

3.3 Diagrama de Despliegue

Los diagramas de despliegue UML muestran cómo los componentes de software se despliegan físicamente en los procesadores; es decir, el diagrama de despliegue muestra el hardware y el software en el sistema, así como el middleware usado para conectar los diferentes componentes en el sistema. En esencia, los diagramas de despliegue se pueden considerar como una forma de definir y documentar el entorno objetivo (Sommerville, 2011).

En la imagen siguiente se muestra el diagrama de Despliegue correspondiente al sistema a desarrollar.



Figura 6. Diagrama de Despliegue.

3.3.1 Descripción del Diagrama de despliegue.

- **Dispositivo Cliente:** Permite a los usuarios consultar y actualizar la información que se encuentra en el Servidor Web utilizando el protocolo de comunicación HTTPS.
- **Servidor Web (Apache):** Se ubican íntegramente las capas de presentación, lógica del negocio y de acceso a datos del sistema, así como los servicios que se brindan.
- **Servidor de base de datos (PostgreSQL):** Se almacenan todos los datos que son consultados y actualizados por los usuarios del Sistema de Gestión de la Información para el control de la venta de pasajes en divisa por Viazul y garantizará el acceso a ellos a través del Servidor Web.

3.4 Pruebas

Las pruebas intentan demostrar que un programa hace lo que se intenta que haga, así como descubrir defectos en el programa antes de usarlo. Al probar el software, se ejecuta un programa con datos artificiales. Hay que verificar los resultados de la prueba que se opera para buscar errores, anomalías o información de atributos no funcionales del programa (Sommerville, 2011).

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requisitos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requisitos funcionales para un programa. Las

pruebas de caja negra intentan encontrar errores en las categorías siguientes: 1) funciones incorrectas o faltantes, 2) errores de interfaz, 3) errores en las estructuras de datos o en el acceso a bases de datos externas, 4) errores de comportamiento o rendimiento y 5) errores de inicialización y terminación (S. Pressman, 2010).

A la aplicación web no se le realizaron las pruebas de caja blanca porque la mayoría de los Requisitos Funcionales son Gestionar y los métodos no tienen un nivel de complejidad alto como para realizarle dicha prueba.

3.4.1 Pruebas de funcionalidad o aceptación

Las pruebas de aceptación XP, también llamadas pruebas del cliente, son especificadas por el cliente y se centran en las características y funcionalidad generales del sistema que son visibles y revisables por parte del cliente. Las pruebas de aceptación se derivan de las historias de los usuarios que se han implementado como parte de la liberación del software (S. Pressman, 2010). A continuación, se describe el Caso de prueba para la HU_4 Gestionar Voucher.

| Caso de Prueba de Aceptación | |
|---|---|
| Código Caso de Prueba: SRCPTNV-1.1 | Nombre Historia de Usuario: Gestionar Voucher. |
| Nombre de la persona que realiza la prueba: Javier Leyva Tamayo | |
| Descripción de la prueba: Se crea un voucher en el cual los campos requeridos son: numero, nombre, área, director del área, causa, destino, fecha de emisión, fecha consumido y estado. | |
| Condiciones de ejecución: Luego de autenticado el usuario, este podrá acceder a una interfaz que le permita crear un voucher. | |
| Entrada/Pasos de ejecución: Una vez dentro del sistema en el menú lateral izquierdo, se escoge la opción voucher, luego selecciona la opción de adicionar y seguidamente se llenan todos los campos requeridos del voucher a crear, los campos obligatorios son: numero, nombre, área, director del área, causa, destino, fecha de emisión y estado. Se debe asegurar que ningún campo quede en blanco y la fecha no debe exceder el día en que se emite el voucher. Luego se presiona el botón Crear. | |

| |
|--|
| Resultado esperado: El sistema muestra un listado donde se muestra el voucher creado. |
| Evaluación de la prueba: Satisfactoria. |

Tabla 18. Caso de Prueba: Gestionar Voucher adicionar.

| Caso de Prueba de Aceptación | |
|---|---|
| Código Caso de Prueba: SRCPTNV -1.2 | Nombre Historia de Usuario: Gestionar voucher. |
| Nombre de la persona que realiza la prueba: Javier Leyva Tamayo | |
| Descripción de la prueba: Se edita un voucher y en el proceso se verifica que todos los campos requeridos se llenen correctamente. | |
| Condiciones de ejecución: Luego de autenticado el usuario, este podrá acceder a una interfaz que le permita editar un voucher. | |
| Entrada/Pasos de ejecución: En el menú lateral izquierdo, se escoge la opción voucher y se selecciona el voucher a editar. Seguidamente se modifican los campos deseados y se debe asegurar que el campo fecha no exceda la fecha en que se modifica. Luego se selecciona la opción aceptar. | |
| Resultado esperado: El sistema muestra el listado de Vouchers donde se incluye el editado. | |
| Evaluación de la prueba: Satisfactoria. | |

Tabla 19. Caso de Prueba: Gestionar Voucher editar.

| Caso de Prueba de Aceptación | |
|--|---|
| Código Caso de Prueba: SRCPTNV -1.3 | Nombre Historia de Usuario: Gestionar voucher. |

| |
|---|
| Nombre de la persona que realiza la prueba: Javier Leyva Tamayo |
| Descripción de la prueba: Se elimina un voucher. |
| Condiciones de ejecución: Luego de autenticado el usuario, este podrá acceder a una interfaz que le permita eliminar un voucher. |
| Entrada/Pasos de ejecución: En el menú lateral izquierdo, se escoge la opción voucher y se selecciona el voucher a eliminar, seguidamente el sistema muestra un mensaje para confirmar la eliminación del mismo. Luego se selecciona la opción sí. |
| Resultado esperado: El sistema muestra el listado de voucher sin el eliminado. |
| Evaluación de la prueba: Satisfactoria. |

Tabla 20. Caso de Prueba: Gestionar Voucher eliminar.

Luego de realizados un conjunto de casos de prueba se detectaron 10 no conformidades en la primera iteración que causaban que el sistema no se comportara de la forma esperada, de las cuales 6 eran errores ortográficos y 4 de validación. De estas no conformidades se solucionaron las 10. Luego de una segunda iteración se encontraron 2 no conformidades de las cuales se solucionaron las 2. Para la tercera iteración no se encontraron no conformidades obteniéndose un resultado satisfactorio para cada una de las combinaciones de datos por escenario. La siguiente figura muestra estos resultados en forma de gráfica:

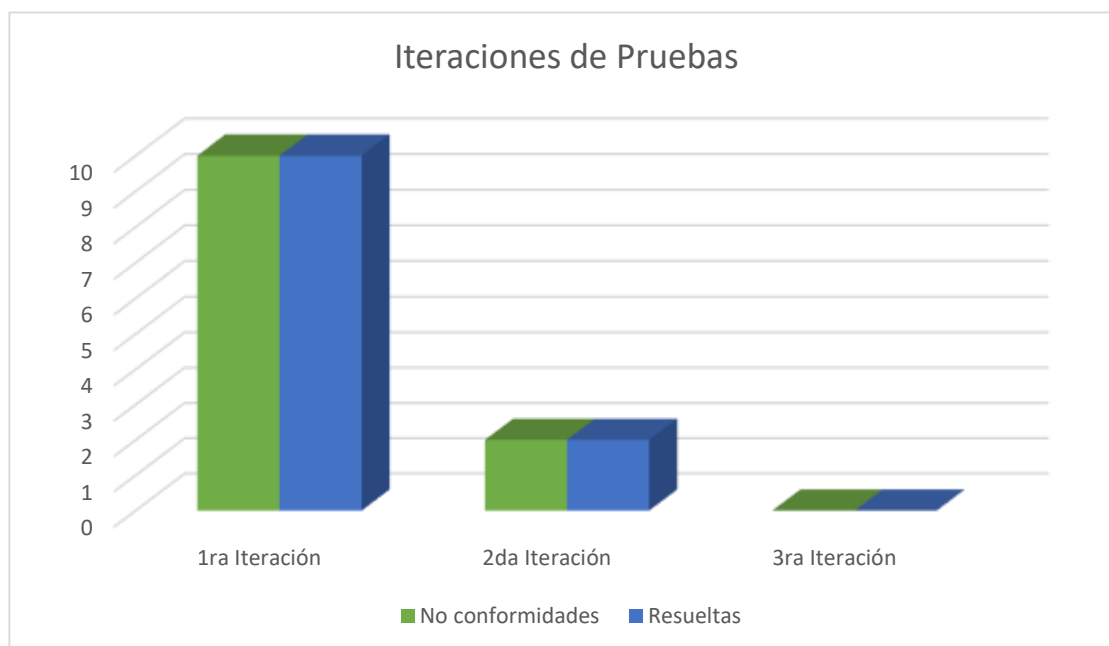


Figura 7. Resultados de pruebas de aceptación.

3.4.2 Pruebas de Seguridad

Cualquier sistema basado en computadora que gestione información sensible o cause acciones que puedan dañar (o beneficiar) de manera inadecuada a individuos es un blanco de penetración inadecuada o ilegal. La penetración abarca un amplio rango de actividades: hackers que intentan penetrar en los sistemas por deporte, empleados resentidos que intentan penetrar por venganza, individuos deshonestos que intentan penetrar para obtener ganancia personal ilícita. La prueba de seguridad intenta verificar que los mecanismos de protección que se construyen en un sistema en realidad lo protegerán de cualquier penetración impropia (S. Pressman, 2010).

Para probar la seguridad del sistema se aplicó el método de prueba basada en herramienta para el cual se seleccionó el sistema Acunetix 12.0 el cual luego de haber escaneado el sistema generó un reporte de las vulnerabilidades de seguridad más importantes clasificadas por el impacto que estas pudieran tener en cuanto a los niveles de seguridad, la clasificación se realiza en alta, media, baja e informativa. En una primera iteración se identificaron 25 vulnerabilidades entre las que se encuentran: formulario HTML sin la protección CSRF, envío de información sensible en texto plano, el método TRACE está activado, enlaces

rotos y el tipo de contraseña de entrada con autocompletado habilitado. De las vulnerabilidades detectadas se encontraron 18 medias, 5 baja y 2 informativas.

Scan details

| | |
|---------------------|--|
| Scan information | |
| Start time | 02/05/2019, 14:18:58 |
| Start url | http://10.8.121.147/portal/web/ |
| Host | 10.8.121.147 |
| Scan time | 7 minutes, 12 seconds |
| Profile | Full Scan |
| Server information | Apache/2.4.34 (Win32) OpenSSL/1.1.0i PHP/7.2.9 |
| Responsive | True |
| Server OS | Windows |
| Server technologies | PHP |

Threat level

Acunetix Threat Level 2

One or more medium-severity type vulnerabilities have been discovered by the scanner. You should investigate each of these vulnerabilities to ensure they will not escalate to more severe problems.

Alerts distribution

| | |
|--------------------|----|
| Total alerts found | 25 |
| ● High | 0 |
| ● Medium | 18 |
| ○ Low | 5 |
| ○ Informational | 2 |

Figura 8. Resultados de pruebas de seguridad por Acunetix.

3.5 Conclusiones parciales

En el presente capítulo se llega a la solución final arribando a las siguientes conclusiones:

- El empleo de los estándares de codificación permitió se obtuviera como resultado un código más limpio, legible y organizado.
- El tratamiento de las excepciones identificadas permitió se interceptaran a tiempo los posibles errores que pudieran ser presentados tanto por la aplicación como por el usuario.
- El diagrama de despliegue permitió el uso de la aplicación teniendo en cuenta los requisitos no funcionales definidos.
- La realización de las pruebas de seguridad con la herramienta Acunetix permitió se detectaran y resolvieran las fallas de seguridad que pudieran comprometer la información manejada por el sistema.

Conclusiones Generales

Con la realización del presente trabajo de diploma se obtuvo la primera versión del Sistema de Registro y Control de Pasajes en Territorio Nacional por Viazul. Con ella se estará aportando una herramienta que permitirá un mejor control y seguridad de la información que se gestiona en la Dirección de Contabilidad y Finanzas referente a la forma de pago para la adquisición de pasajes.

Los resultados obtenidos permiten al autor plantear las siguientes conclusiones:

- El análisis y estudio de sistemas similares demostró que no existe un sistema que permita gestionar la información del control de pagos de Vouchers para la confección de un expediente contable que gestione dicha información basado en tecnologías libres.
- El enfoque ágil propuesto por la metodología XP, y el uso de tecnologías y herramientas libres, permitieron realizar el diseño de un sistema que gestiona la información que genera la utilización de Vouchers en concordancia con las especificaciones del cliente.
- La arquitectura y los estándares de codificación empleados permitieron la implementación del sistema garantizando el cumplimiento de normas y reglas de implementación.
- La realización de las pruebas aceptación y seguridad permitió comprobar el correcto funcionamiento del sistema desarrollado, además demostró su validez con resultados satisfactorios.
- El desarrollo del sistema para el registro y control de los procesos de pago de los pasajes emitidos por Viazul vía voucher, permite un mejor control y seguridad de toda la información que se genera por este concepto en la DCF.

Recomendaciones

Se sugieren a continuación aspectos que se deben tener en cuenta para el futuro perfeccionamiento del sistema surgidos durante el desarrollo de la investigación:

- Implementar un componente de notificaciones que permita informar al administrador del sistema sobre la actividad de otros usuarios en el mismo.
- Implementar un componente que permita obtener la información de las personas desde el directorio activo de la UCI para garantizar la fiabilidad de la información que se registra.

Referencias Bibliográficas

Capó Marrero, Lisué y García Dorta, Daily. 2014. 2014.

Cervantes, Humberto. 2018. SG. [En línea] 7 de mayo de 2018. <https://sg.com.mx/revista/27/arquitectura-software>.

Chacón Vázquez, Elisa Lisbeth. 2013. 2013.

2018. ContaPyme. [En línea] 2018. www.contapyme.com.

ControlPass. 2018. Controldepasajeros. *ControlPass*. [En línea] ControlPass, 2018. www.controldepasajeros.com.

Doyle, Matt. 2012. 2012.

Extreme Programming Installed. Jeffries, R., Anderson, A., Hendrickson, C. 2012. s.l. : Addison-Wesley, 2012.

Ferreira dos Santos, Joao Carlos. 2012. 2012.

Gonzalez, Mario. 2013. *BOOTSTRAP*. 2013.

Groner, Loinane. 2013. *Mastering Ext JS*. s.l. : PACKT Publishing, 2013.

Guevara Benites, Alexander . 2018. DevCode. [En línea] 2018. <https://devcode.la/blog/frontend-y-backend/>.

Hernandez Delgado, Alicia. 2013. 2013.

HERRAMIENTAS PARA LA GESTIÓN DE PROCESOS DE NEGOCIO Y SU RELACIÓN CON EL CICLO DE VIDA DE LOS PROCESOS DE NEGOCIO. Galvis Lista, Ernesto A. y González Zabala, Mayda Patricia. 2014. s.l. : Ciencia e Ingeniería Neogranadina, 2014.

Howard. 2012. *Howard*. 2012.

INFORMATICA HOY. 2007. 2007.

Informaticas. 2013. 2013.

Jácome, Santiago, Tituaña Cumbal, Walter Celiano y Torres Cañizares, Edwin Jesús. 2009. 2009.

Larman, Craig. 1999. *Applying UML and patterns : an introduction to object oriented analysis and design.* s.l. : Pearson Educación, 1999.

—. **2006.** jorgesaavedra.wordpress.com. [En línea] 17 de 8 de 2006. [Citado el: 16 de 2 de 2019.]
<https://jorgesaavedra.wordpress.com/2007/05/08/patrones-grasp-patrones-de-software-para-la-asignacion-general-de-responsabilidadparte-ii/>.

Londres Romero, Pedro Luis. 2012. 2012.

Marcisack. 2011. *Marcisack.* 2011.

Martinez, R. 2013. 2013.

Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP). **Letelier, Patricio y Penadés, Carmen. 2016.** Valencia : Universidad Politécnica de Valencia, 2016.

Modelo de procesos para el desarrollo del frontend en aplicaciones web. **Valdivia Caballero, José Jesus. 2016.** Lima, Perú : Universidad de Lima, 2016.

MVC DESDE SMALLTALK A LA WEB. **ALEJANDRA GOS, MARÍA . 2004.** La Plata : s.n., 2004.

Naramore. 2005. *Naramore.* 2005.

Patron Modelo Vist Controlador - EcuRed. *EcuRed.* [En línea] [Citado el: 20 de 1 de 2019.]
https://www.ecured.cu/Patr%C3%B3n_Modelo_Vista_Controlador.

Poyo, Delgado. 2010. 2010.

PRESSMAN. 2010. *PRESSMAN.* 2010.

Pressman, Roger S. 2010. INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO. *INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO.* s.l. : MC Graw Hill, 2010, pág. 767.

PricewaterhouseCoopers . 2013. *Talent Mobility 2020, The next generation of international assignments.* 2013.

Rosa, Ing. 2011. 2011.

—. 2011. *asdasd*. 2011.

S. Pressman, Roger. 2010. *INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO Séptima edición*. Mexico : s.n., 2010.

Sánchez, Tamara Rodríguez. 2015. *Metodología de desarrollo para la Actividad productiva de la UCI*. . 2015.

SIBUS. 2018. Sibus. *Sibus*. [En línea] Sibus, 2018. www.sibus.cl.

Sommerville, Ian. 2011. *Ingeniería de Software*. s.l. : Pearson Educación, 2011.

—. 2011. *Ingeniería de Software*. s.l. : Pearson Educación, 2011.

Soto, Jaime. 2013. Chile, 2013.

Tarazona, Ivon y Gomez, Oriana. es.scribd.com. [En línea] [Citado el: 17 de febrero de 2019.]
<https://es.scribd.com/document/8962964/uml>.

Torres, Liuba Almira. 2012. 2012.

Utilización del Patrón Modelo – Vista – Controlador (MVC) en el diseño de software educativos. . **Ciudad Ricardo, Febe Ángel. 2006.** La Habana : UCI, 2006.

Viazul. 2018. Viazul. *Viazul*. [En línea] 2018. www.viazul.com.

WESKE. 2007. *WESKE*. 2007.

Zaninotto, François y Potencier, Fabien. 2010. 2010.