

Universidad de las Ciencias Informáticas

Facultad 1



Título:

**Propuesta para la Administración del Sistema Transportación Nacional de
la Universidad de las Ciencias Informáticas**

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

Autor:

Dayron Alejandro Mederos Ramos

Tutores:

Jany Coto García

Vladimir Campos Kindelán

La Habana. Septiembre 2020

DECLARACIÓN DE AUTORÍA

Yo, Dayron Alejandro Mederos Ramos, declaro ser el único autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Dayron Alejandro Mederos Ramos

Firma del Autor

Vladimir Campos Kindelan

Jani Coto García

Firma del Tutor

Firma del Tutor

AGRADECIMIENTOS

Agradezco de forma incondicional a todos los que apoyaron positivamente el desarrollo de este Trabajo de Diploma con especial énfasis a mis seres más queridos que han vivido junto a mí.

DEDICATORIA

Dedico este Trabajo de Diploma en especial a mi seno familiar por dotarme de tanto optimismo, seguridad y confianza, a mis compañeros de estudio y profesores que en todo momento han sido una guía para lograr mis éxitos y siempre han estado dispuestos para ayudar y a todos aquellos que de una forma u otra han influido positivamente en el progreso de mi carrera.

RESUMEN

Por la importancia que representa para Cuba, la Universidad de las Ciencias Informáticas (UCI) cuenta con una Dirección de Transportación Nacional, encargada de trasladar desde y hacia sus municipios de residencia a estudiantes y trabajadores de la UCI. En el presente trabajo se tiene como objetivo proponer un sistema para la administración de la transportación nacional de la UCI. Para ello, se realizó un estudio de sistemas de administración de transporte existentes en Cuba y el mundo, así como su disponibilidad para emplearse en dicha universidad. Se exponen las herramientas tales como PostgreSQL para la gestión de la base de datos, PHPStorm como entorno de desarrollo de la aplicación, Pencil para el diseño de prototipo de interfaz, Visual Paradigm para modelar el sistema propuesto y tecnologías como el framework de PHP, Symfony, como lenguaje de programación y Bootstrap para el desarrollo del front-end. Se hace un análisis detallado de cómo se lleva a cabo actualmente el proceso de negocio, que permitió el levantamiento de requisitos para el diseño de una propuesta de solución que responda a las necesidades de un sistema de administración de la Dirección de Transportación Nacional de la UCI.

PALABRAS CLAVE: sistema de administración, transportación, propuesta de solución.

ABSTRACT

Due to importance it represents for Cuba, the University of Informatics Sciences (UCI) has a Directorate of National Transportation, in charge of transferring students and teachers to and from their municipalities of residence. The objective of this work is to propose a system for the administration of the UCI National Transportation. For this, a study of existing transportation management systems in Cuba and the world was carried out, as well as their availability to be used in said university. It's exposed tools such as PostgreSQL for database management, PHPStorm as the application development environment, Pencil for the interface prototype design, Visual Paradigm to model the proposed system and technologies such as the PHP framework, Symfony, as a programming language and Bootstrap for front-end development. These are used for the construction of the module; a detailed analysis is made of how the business process is currently carried out, that allowed the obtaining of requirements for the design of a solution proposal that responds to the needs of the Directorate of UCI National Transportation.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN	10
CAPÍTULO 1: ESTUDIO DE SISTEMAS DE GESTIÓN DE TRANSPORTE Y HERRAMIENTAS A UTILIZAR EN LA SOLUCIÓN.....	13
1.1. DEFINICIÓN DEL MARCO CONCEPTUAL DE LA INVESTIGACIÓN.....	13
1.2. SISTEMAS HOMÓLOGOS.....	13
1.3. TECNOLOGÍAS Y HERRAMIENTAS	15
1.4. METODOLOGÍA DE DESARROLLO ÁGIL (SCRUM).....	17
CONCLUSIONES PARCIALES.....	18
CAPÍTULO 2: DESCRIPCIÓN DEL MÓDULO DE TRANSPORTACIÓN NACIONAL.....	19
2.1 MODELADO DEL PROCESO DE NEGOCIO.....	19
2.1.1 DESCRIPCIÓN DEL FLUJO DEL PROCESO DE TRANSPORTACIÓN	19
2.1.2 REGLA DEL NEGOCIO DEL PROCESO DE TRANSPORTACIÓN.....	20
2.2 REQUISITOS DEL SOFTWARE.....	21
2.2.1 REQUISITOS FUNCIONALES.....	21
2.2.2 REQUISITOS NO FUNCIONALES	22
2.2.3 DESCRIPCIÓN DE REQUISITOS	23
2.3 PROPUESTA DE SOLUCIÓN.....	29
2.4 DESCRIPCIÓN DE LA ARQUITECTURA	30
2.4.1 PATRÓN DE ARQUITECTURA	30
2.5 PATRONES DE DISEÑO.....	31
2.5.1 PATRONES DE SOFTWARE PARA LA ASIGNACIÓN GENERAL DE RESPONSABILIDAD.....	32
2.5.2 PATRONES GOF.....	33
2.6 PATRONES DE DISEÑO DE BASE DE DATOS.....	34
2.7 DIAGRAMA DE DESPLIEGUE.....	35
CONCLUSIONES PARCIALES	37
CONCLUSIONES GENERALES	38
REFERENCIAS BIBLIOGRÁFICAS	39

ÍNDICE DE ILUSTRACIONES

FIGURA 1 DIAGRAMA DE PROCESO.....	20
FIGURA 2 MODELO - VISTA – CONTROLADOR.....	31
FIGURA 3 PATRÓN EXPERTO	32
FIGURA 4 PATRÓN CONTROLADOR.....	33
FIGURA 5 MODELO DE DATOS	35
FIGURA 6 DIAGRAMA DE DESPLIEGUE	36

ÍNDICE DE TABLAS

TABLA 1 REQUISITOS FUNCIONALES.....	22
TABLA 2 RF: GESTIONAR TIPO DE VIAJE (MODIFICAR TIPO DE VIAJE).....	25
TABLA 3 RF: GESTIONAR ENTIDAD (MODIFICAR ENTIDAD).....	26
TABLA 4 RF: GESTIONAR TIPO DE TRANSPORTE (REGISTRAR TIPO DE TRANSPORTE).....	29

INTRODUCCIÓN

A lo largo de la historia de la humanidad el transporte ha significado un medio indispensable para contribuir a la integración de las naciones. Su utilidad en las actividades cotidianas ha provocado que diseñemos un sin número de vehículos dirigidos a diferentes tareas. En este sentido, las compañías transportistas, enfrentaban grandes problemas con la gestión de sus negocios, de manera que requerían, que el almacenamiento de la información que manejaban, fuera más eficiente y segura a posibles pérdidas.

En Cuba, la gestión de transporte está teniendo un gran auge en estos tiempos, por ello se solicita fomentar la expansión de proyectos que agilicen y haga más eficientes los trámites requeridos. Por tal razón un grupo de entidades se dedican a impulsar la informatización de procesos asociados a la gestión del transporte, una de ellas es la Universidad de las Ciencias Informáticas (UCI), fundada por iniciativa del Comandante en Jefe Fidel Castro, la cual tiene a su cargo la formación de ingenieros en ciencias informáticas (1).

La UCI como único centro educativo de su tipo en Cuba acoge en su residencia más de 500 trabajadores y cercas de 3000 estudiantes de todo el territorio nacional durante el periodo escolar, por lo que ella gestiona la transportación de este personal desde y hacia sus provincias de origen, mediante un sistema de gestión desarrollado por la Dirección de Informatización (DIN).

Esta dirección de producción conduce el proceso de informatización en la Universidad en coordinación con sus diferentes instancias. Define las políticas, lineamientos, procedimientos y tecnologías que rigen los servicios de informatización de la Universidad, obteniendo como resultados servicios y soluciones informáticas de alto valor agregado que faciliten la ejecución de los procesos de la organización (2). Entre sus proyectos se encuentra el Sistema de Transportación Nacional integrado Sistema de Gestión Ciudadana (CAMPUS) de la UCI.

El Sistema de Transportación Nacional lleva en explotación 12 años, pero actualmente sus funcionalidades no cubren las necesidades de los residentes universitarios. Debido a esto y atendiendo a las continuas solicitudes del Director de Transportación Nacional, el módulo ha tenido que ser modificado en varias ocasiones, solucionando temporalmente los problemas que surgen, pero acarreando consecuencias como fallas en el código.

Unido a lo anteriormente planteado, el sistema presenta nuevos requerimientos que no han sido resueltos, ejemplo de ellos: la función "Ordenar reservaciones" duplica o triplica las reservaciones de las personas, esto conlleva a cubrir varios asientos por la misma persona y a su vez emite listado con errores. Resulta necesario incorporar un administrador local que no sea usuario del dominio, en caso de que el Servicio Central de Autenticación (CAS-UCI) no esté funcionando y los administradores no

puedan trabajar en el sistema. También se tiene que los trabajadores que no pertenecen a la UCI pero que utilizan el servicio de transporte no pueden reservar. Hoy estas reservaciones las hace el Administrador de la aplicación mediante un documento que le envía la entidad beneficiada. Con el fin de solucionar la problemática planteada se ha formulado el siguiente **problema de investigación**: ¿Cómo mejorar el proceso de administración de la Transportación Nacional de la Universidad de las Ciencias Informáticas?

Para dar respuesta a ello se centra el **objeto de estudio** en la gestión administrativa de los sistemas de transportación; enmarcando el **campo de acción** de la investigación en el proceso de administración de la transportación nacional de la Universidad de Las Ciencias Informáticas. Para guiar la búsqueda de la solución se define como **objetivo general** proponer un sistema para la administración de la transportación nacional de la UCI.

Teniendo en cuenta además como objetivos específicos:

- Analizar los principios teóricos de la investigación enfocados en los sistemas de administración del transporte.
- Realizar análisis y diseño del módulo para la administración del transporte nacional en la UCI.
- Obtener una propuesta para la administración del transporte.

Las tareas generales de la investigación se dirigen a:

- Caracterización del proceso de administración de la transportación nacional en la Universidad de las Ciencias Informáticas, que permita dar un mejor entendimiento y orientación al tema de la investigación.
- Estudio de sistemas similares existentes a nivel nacional e internacional asociados a los sistemas de administración del transporte, que permita reutilizar las ventajas y detectar posibles errores.
- Análisis de las tecnologías y herramientas a emplear para la realización de una propuesta de solución de software.
- Identificación de los requerimientos generados según el proceso de desarrollo aplicado para la realización de una propuesta de solución de un producto de software.

Los métodos empleados durante la investigación fueron (3):

- **Métodos teóricos:**
 - Analítico-Sintético: se utilizó para el proceso de análisis y estudio de la bibliografía empleada, arribando así a conclusiones que contribuyeron a elaborar la propuesta de solución, además para el estudio y análisis de las herramientas a utilizar.
 - Dialectico-Materialista: como base general para toda la investigación.

- **Métodos empíricos:**

- Entrevista: para fundamentar y precisar el problema a resolver. Fue utilizada para establecer las necesidades del cliente, identificar los procesos de transporte que se llevan a cabo actualmente en la universidad. Las entrevistas fueron realizadas al jefe de la dirección de transportación nacional.
- Observación: posibilita verificar información obtenida a través de las entrevistas realizadas. Este método fue empleado para comprobar la manera que se gestiona el actual sistema de transportación nacional.

La investigación descrita en este trabajo está compuesta por una introducción, dos capítulos, conclusiones, referencias bibliográficas, bibliografías, la organización del mismo se describe a continuación:

Capítulo 1: Estudio de Sistemas para la Administración del Transporte y Herramientas a utilizar en la Solución.

Contiene el marco conceptual de la investigación y el estudio del actual sistema informático para la gestión de transportación nacional. De igual forma muestra el proceso de desarrollo que se utiliza y los diferentes lenguajes, tecnologías y herramientas a utilizar.

Capítulo 2: Descripción del Sistema para la Administración de la Transportación Nacional.

Se detalla el flujo actual de los procesos involucrados en el campo de acción, se identifican los requisitos del sistema y se describen las especificaciones con sus prototipos de interfaz de usuario. Además, se exponen los patrones de diseño y arquitectura que utiliza el marco de trabajo seleccionado, se muestra el diseño de la base de datos del sistema y se realiza la propuesta de solución.

CAPÍTULO 1: ESTUDIO DE SISTEMAS DE GESTIÓN DE TRANSPORTE Y HERRAMIENTAS A UTILIZAR EN LA SOLUCIÓN

En este capítulo se describen aspectos sobre otros sistemas para la administración de transporte existentes en Cuba y en el mundo. Se realiza un estudio de las principales tendencias de desarrollo sobre las que se apoyará la propuesta de solución. Se abordan las herramientas seleccionadas para el análisis y diseño de la aplicación, de forma tal que se cumplan todos los objetivos trazados.

1.1. Definición del marco conceptual de la investigación

Desde los años 70 se ha elevado la actividad comercial en el sector del transporte, en aquel momento se crearon herramientas que gestionaban los pedidos de transporte. Ya para el nuevo siglo, el desarrollo de estas herramientas permitía hacer análisis de cadena de suministro con la posibilidad de tomar de decisiones estratégicas para el mejoramiento y optimización de las transportaciones (4).

Sergio Lujan Mora define que una aplicación web es la cual el usuario por medio de un navegador realiza peticiones a una aplicación remota accesible a través de internet (o a través de la intranet) y recibe una respuesta que se muestra en su propio navegador (5).

Jarel Remick describe aplicación web, como una aplicación que utiliza la red y las tecnologías de los navegadores web con el fin de lograr una o más tareas en una red, por lo general a través de un navegador (6).

Se asume que una aplicación web son soluciones informáticas que el usuario puede utilizar accediendo a un servidor web, utilizando la red y las tecnologías de los navegadores web.

Un sistema de gestión es una metodología que ayuda a visualizar y administrar mejor una organización, empresas, áreas o procesos, por lo tanto, logra mejores resultados a través de acciones y toma de decisiones basadas en datos y hechos (7).

Un sistema de gestión se explica como el conjunto de elementos interrelacionado de una empresa u organización por lo cual se administra de forma planificada las operaciones de las mismas, en la búsqueda de satisfacción del cliente (8).

Partiendo de las definiciones anteriores se asume que un sistema de gestión es una herramienta que te permitirá optimizar recursos, reducir costes y mejorar la productividad en una empresa u organización.

1.2. Sistemas homólogos

Su fácil manejo y comodidad para los clientes hacen que estos sistemas sean muy usados en el mundo, sin importar la estructura organizacional (pequeña o grande) de las empresas propietarias. Ellos

permiten que los administradores mantengan un control total sobre las reservaciones, disponibilidad, capacidad y otras actividades para los que fueron diseñados.

JR SOFTWARE Transporte de Pasajero

El sistema argentino JR SOFTWARE se encarga de la administración de flotas de camiones, autobuses, aviones, barcos, así como la programación de las salidas de autobuses desde los diferentes puntos de ventas. Basa su desarrollo en la arquitectura cliente–servidor. Es un sistema propietario cuyo pago de licencia es costoso.

Entre las principales funcionalidades que brinda está:

- La reservación de asiento del autobús.
- Configuración de viajes y salidas por autobús y rutas.
- Exportación de la información en formato PDF.
- Permite configurar el sistema como sucursal o casa central, estableciendo los parámetros para su funcionamiento en la red local (9).

Se determina que no puede ser adaptado, debido a que su negocio no se corresponde con el llevado a cabo por la Dirección de Transportación Nacional. Además, tiene como desventaja que es un sistema propietario por lo que no permite modificaciones en el código. No cumple con las características establecidas para la integración con CAMPUS.

Radixx Air

Pertenece a una aerolínea estadounidense encargada de gestionar las reservaciones de vuelos.

Entre las características de este sistema se encuentran:

- Proveer avisos mediante una pantalla gestionada por un administrador que contiene temas de importancia para mantener el personal informado con los cambios de última hora que puedan ocurrir en las reservaciones.
- Los avisos pueden ser creados o modificados fácilmente y se le puede asignar una fecha de vencimiento de los mismos.
- Es un sistema configurable, que permita a la aerolínea el establecimiento de un plan de vuelo, la asignación de inventario de asiento para los vuelos, la creación de tarifa y las reglas de tarificación.
- Guarda el historial de las reservaciones incluyendo todos los cambios que se le efectuaron a la misma desde su creación en la base de datos, para una verificación fácil y una recuperación de los registros de nombre de los pasajeros.

- Es un sistema propietario, donde hay que pagar elevados costos para poder obtener su licencia (10).

Al culminar el análisis se determina que no fue adaptado, debido a que no se corresponde con el negocio llevado a cabo por la transportación nacional UCI.

Cubana de Aviación

Cubana de Aviación es un sistema cubano, se encarga de la reservación de la transportación aérea nacional e internacional de pasajeros.

Entre sus funcionalidades destacan:

- Configuración de vuelo de origen y destino hacia diferentes países para que los usuarios efectúen la reservación.
- La información que se gestiona está estructurada por módulos, posibilitando la organización de la información.
- Brinda la posibilidad al usuario de consultar los datos (11).

Después de la investigación, el sistema no se puede utilizar como solución al problema porque su negocio no corresponde con el llevado a cabo por la dirección. Además, solamente ofrece realizar reservaciones para líneas aéreas y se desarrolló sobre un software propietario.

Transportación Nacional UCI

La aplicación esta desarrollada en Symfony 2.8, se encuentra implantado en la Universidad de las Ciencias informáticas y posibilita la realización de reservaciones del transporte para las personas de la universidad en diferentes etapas del curso escolar.

Algunas de las funcionalidades que este sistema permite realizar es:

- Gestión de las reservaciones
- La definición de puntos, entidades, tipos de transportes, tipos de viajes y tipos de bloqueos
- La configuración de reglas de acceso, transportes y viajes (12).

Cabe destacar que se identificaron algunas funcionalidades y característica que debe ser utilizada en el desarrollo del Módulo como: gestionar puntos, transportes, viajes, entidades y los tipos de transportes, viajes y bloqueos.

Al culminar el estudio de este se evidencio que no puede ser adaptado, debido a problemáticas anteriormente expuestas. Aunque se tuvo en cuenta varias funcionalidades para el desarrollo del sistema para la administración de la transportación nacional de la UCI.

1.3. Tecnologías y herramientas

Symfony 5.0

Framework diseñado para desarrollar aplicaciones web basadas en el patrón Modelo Vista Controlador. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación (13).

Esta nueva versión de Symfony trae una mejor organización en su estructura y facilita la conectividad y manejo de las bases de datos. Su integración con recientes versiones de PHP y la inclusión de su nuevo componente de notificaciones le favorece ante la versión 2.8 con la que fue implementada el Sistema de Transportación Nacional UCI, la cual perderá su soporte próximamente (14).

Bootstrap 4

Kit de herramientas de código abierto para desarrollar con HTML, CSS y JS. Permite realizar prototipos de ideas o crear aplicaciones completas con variables y combinaciones Sass, sistema de cuadrícula sensible, componentes recompilados extensos y complementos potentes creados en jQuery; haciendo posible el desarrollo de sitios web receptivo y atractivo para el cliente (15).

- **HTML 5**

Hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros (16).

- **JavaScript**

Lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador (17).

El lenguaje permite implementar funciones complejas en página web, actualizar contenido, crear animaciones de gráfico 2D/3D, desplazar máquinas reproductoras de videos, entre otras.

- **CSS 3**

Es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, entre otros (18).

Crea estilo de hoja para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas (16).

PostgreSQL 12

Potente sistema de base de datos relacional de objetos de código abierto que usa y amplía el lenguaje SQL combinado con muchas características que almacenan y escalan de manera segura las cargas de trabajo de datos más complicadas (19). PostgreSQL 12 introduce la capacidad de ejecutar consulta sobre documentos JSON. Tales consultas pueden utilizar los mecanismos de indexación existentes para documentos almacenados en el formato JSONB para recuperar datos de manera eficiente. Las optimizaciones para la utilización de espacios y el rendimiento lectura-escritura y la reducción de la sobrecarga de WAL (Write-Ahead Logging) para la creación de índice lo hace sin duda un potente gestor de base de datos para el desarrollo de proyectos (20).

La selección de esta herramienta se debe a la aceptación que tiene dentro de la política de desarrollo de software de la UCI. Además, es usada por Transportación Nacional UCI, sistema anteriormente citado, para la gestión de su base de datos.

Pencil 3.0.4

Es una herramienta gratuita y de código abierto con la que se podrá diseñar prototipos web. Es de gran ayuda para diseñadores y desarrolladores, ya que permite diseñar rápida y fácilmente documentos de propuesta para clientes. proporciona varias colecciones de formas integradas para dibujar diferentes tipos de interfaz de usuario, desde plataformas de escritorio hasta móviles.

Con mejoras en la impresión y exportación a PDF, la introducción a un nuevo formato de archivo basado en zip para dar soporte a documentos grandes y el remplazo de Mozilla XULRunner por Electron para su ejecución, la condiciona como herramienta de diseño de interfaz para el desarrollo proyectos (21).

Visual Paradigm 15.1

Es una suite de desarrollo de software que proporciona todas las herramientas que se necesitan para la gestión de proyectos, desarrollo de proyecto y colaboración en equipo. Una de las características de esta herramienta es modelar diagramas que se generan en el expediente de proyecto y que es una herramienta multiplataforma (22).

1.4. Metodología de desarrollo ágil (Scrum)

La metodología Scrum se basa en la consecución de objetivos a corto plazo dentro de un proyecto. La prioridad es presentar un producto funcional al cliente lo antes posible, y que este tenga una participación constante en el desarrollo del trabajo. Además, propicia el solapamiento de tareas que

pueden ejecutarse simultáneamente (23). El uso de esta metodología permitió la obtención de la lista de especificación de requisitos. Posibilitó además la especificación de las funcionalidades del sistema, por iteraciones, con el objetivo de lograr planificación óptima del tiempo. La principal ventaja de Scrum es que no necesita la participación continua del cliente, solo su asistencia a las reuniones de cada iteración.

Conclusiones parciales

El análisis de los sistemas de gestión de transporte nacionales e internacionales, permitió valorar las principales características de sus procesos, llegando a la conclusión, que ninguno de ellos es ideal a proponer como solución en la universidad. Se lograron fundamentar las bases teóricas de la investigación, así como la selección de herramientas y tecnología idóneas para el desarrollo de una propuesta que satisfaga las necesidades de la UCI.

CAPÍTULO 2: DESCRIPCIÓN DEL MÓDULO DE TRANSPORTACIÓN NACIONAL

En el presente capítulo se define el estado actual de los procesos involucrados en el campo de acción que son los objetos de informatización de este trabajo. Se realiza el diseño y el análisis crítico de la ejecución de los mismos. A partir describe una propuesta de solución mediante la descripción de las funcionalidades que brindará la nueva propuesta, los patrones de diseño y el modelo de datos para el registro de la información a gestionar.

2.1 Modelado del proceso de negocio

Un proceso de negocio es un conjunto de tareas relacionadas lógicamente, llevadas a cabo para generar productos y servicios. Modelar los procesos de negocio es una parte esencial de cualquier proceso de desarrollo de software. Permite capturar el esquema general y los procedimientos que gobiernan el negocio. Este modelo provee una descripción de dónde se va a ajustar el sistema de software considerado dentro de la estructura organizacional y de las actividades habituales. También provee la justificación para la construcción del sistema de software al capturar las actividades manuales y los procedimientos automatizados habituales que se incorporarán en el nuevo sistema (24).

2.1.1 Descripción del flujo del proceso de transportación

El proceso comienza cuando la Rectoría envía el “Plan calendario de transportación” donde se especifican las fechas de la transportación masiva para los respectivos pases, entiéndase: pase fin de año y pase fin de curso. Con estas fechas se crea un plan de transportación, el cual es presentado al Ministerio de Transporte (MITRANS). De ser aprobado, es consultado por el MITRANS y se asignan los transportes a la Universidad, en caso contrario se cambian las fechas del plan, se confecciona uno nuevo y se presenta nuevamente.

Con los transportes asignados, la Dirección de Transporte activa los puntos de destino y registra el transporte acorde al plan aprobado. Se habilita el sistema e informa a la comunidad universitaria que ya es posible realizar la confirmación de su transportación. Dado el plazo para la reservación, se cierra el período de confirmación y posteriormente se inhabilita el sistema de reservación.

La representación gráfica de este proceso se muestra en la siguiente imagen (Figura 1).

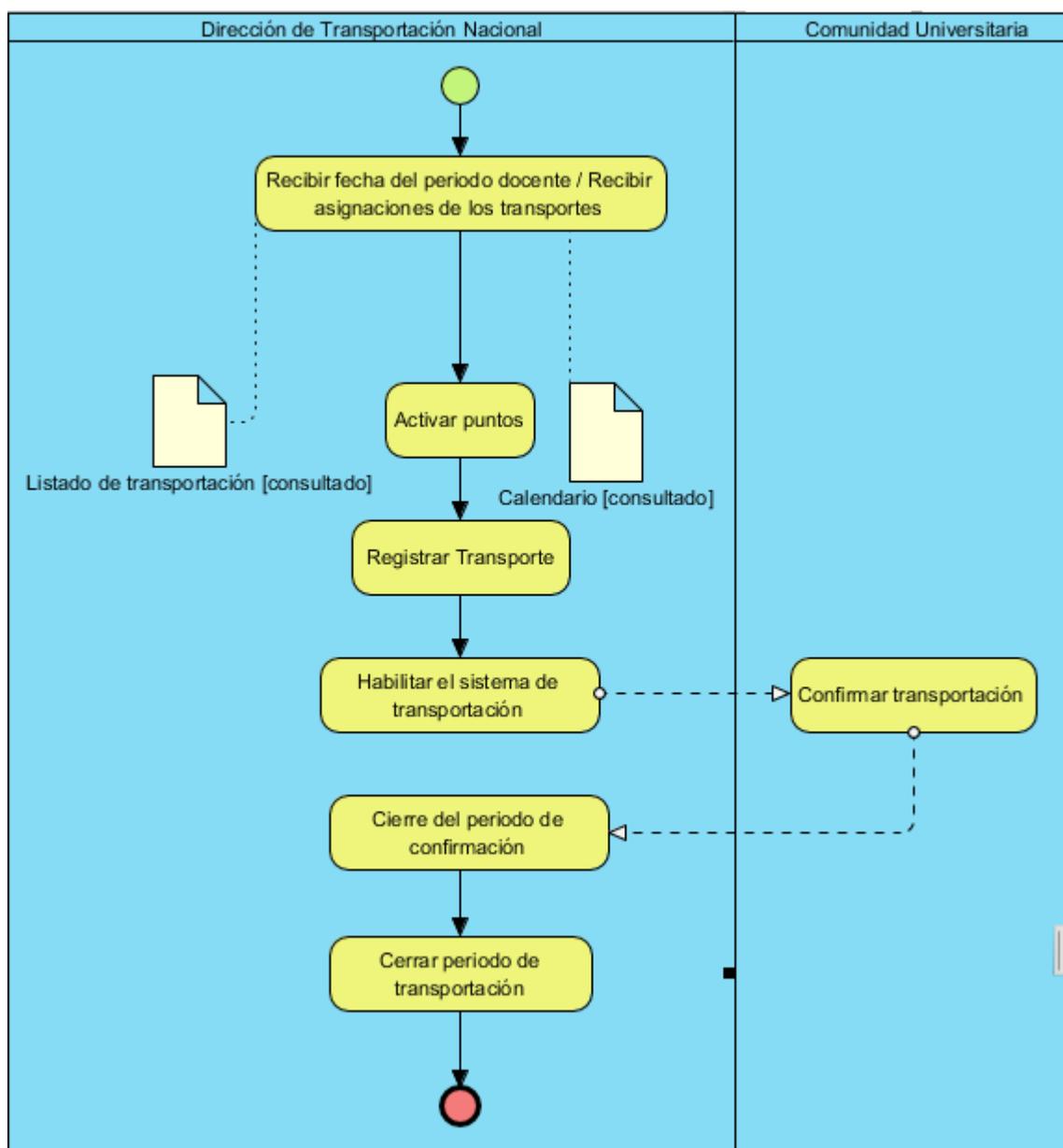


Figura 1 Diagrama de Proceso

2.1.2 Regla del negocio del proceso de transportación

Se han distinguido las siguientes reglas para lograr garantizar y cumplir las restricciones existentes en el negocio:

1. La Rectoría es la responsable de determinar las fechas del calendario de transportación.
2. El especialista de transporte es el único que puede elaborar el plan de transportación y luego darlo a conocer a la Universidad.

3. El Administrador del sistema es el único que puede configurar los puntos, registrar los transportes y habilitar y deshabilitar la transportación.
4. Todas las personas de la Universidad pueden obtener información sobre las salidas, tipo de transportes y personas que viajarán.
5. Solamente se podrá confirmar y cancelar la transportación en las fechas establecidas.
6. El especialista de transporte es el único autorizado a realizar la distribución de los viajeros en los medios de transporte asignados (ómnibus, tren, barco).

2.2 Requisitos del software

Un requisito de software es una característica que se debe exhibir en el software desarrollado o adaptado para solucionar un problema particular. A menudo los requisitos de sistemas de software se clasifican en funcionales y no funcionales (25).

Los requisitos funcionales de un sistema describen lo que el sistema debe hacer. Definen los servicios que debe proporcionar el sistema, la forma en que este debe reaccionar a ciertas entradas y cómo se debe comportar en situaciones particulares. En algunos casos, los requisitos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer.

Los requisitos no funcionales son aquellos que no se refieren directamente a las funciones específicas que proporciona un sistema, sino a las propiedades emergentes de este como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema.

2.2.1 Requisitos funcionales

No.	Nombre	Prioridad	Complejidad
RF1	Gestionar tipo de transporte	Media	Media
RF2	Gestionar tipo de viaje	Media	Media
RF3	Gestionar tipo bloqueo	Media	Media
RF4	Gestionar entidades	Media	Media
RF5	Gestionar puntos	Media	Media
RF6	Gestionar tramitador tercerizado	Media	Media
RF7	Exportar a Excel	Media	Alta
RF8	Exportar a PDF	Media	Media
RF9	Gestionar transportación	Media	Media

CAPÍTULO 2: DESCRIPCIÓN DEL MÓDULO DE TRANSPORTACIÓN NACIONAL

RF10	Gestionar viaje	Media	Media
RF11	Asociar reglas al viaje	Alta	Baja
RF12	Gestionar salida	Media	Media
RF13	Enviar correo	Alta	Alta
RF14	Gestionar transporte	Media	Media
RF15	Exportar identificador múltiple	Media	Alta
RF16	Exportar identificador	Media	Media
RF17	Configurar transporte	Alta	Media
RF18	Listar personas	Baja	Baja
RF19	Gestionar autorizo	Media	Media
RF20	Asociar persona	Alta	Media
RF21	Gestionar reservación	Media	Media
RF22	Actualizar transportación múltiple	Media	Media
RF23	Listar asociados	Baja	Baja
RF24	Listar distribuciones	Baja	Baja
RF25	Distribuir provincia	Baja	Alta
RF26	Distribución múltiple por provincia	Baja	Media
RF27	Ver detalle de persona bloqueada	Baja	Baja
RF28	Desactivar bloqueo múltiple	Baja	Baja

Tabla 1 Requisitos Funcionales

2.2.2 Requisitos no funcionales

Los requisitos no funcionales, como su nombre sugieren, son aquellos requisitos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este como la usabilidad, el tiempo de respuesta y la capacidad de almacenamiento (26).

Usabilidad

1. Sólo se mostrarán a los usuarios aquellas acciones o informaciones del menú lateral a las que por su responsabilidad o rol dentro del negocio necesiten acceder.
2. El sistema provee una interfaz visual amigable, sencilla e intuitiva, que garantiza la facilidad para el trabajo de los usuarios. Para ello presenta una iconografía que se ajusta en cada caso con las funciones que representan, los botones tienen descripciones acordes a las funcionalidades que realizan y en los campos de textos editables por el usuario existen etiquetas flotantes que describen la acción que se debe realizar en cada caso.

3. El sistema debe presentar un menú lateral y una barra de íconos flotantes que permitan el acceso rápido a la información por parte de los usuarios.
4. Las vistas del sistema deben indicar en cada momento la acción que se está realizando, así como los íconos deben estar representados por una imagen acorde a la acción que se realiza mediante el mismo.
5. Para el uso del sistema se requiere una PC cliente con los siguientes sistemas operativos: Windows o GNU/Linux.

Confiabilidad

6. El tratamiento de las excepciones permitirá un seguimiento hasta guardar información acerca del lugar dónde se produjo el error y de los parámetros de configuración del sistema que lo provocaron. Cuando ocurre una excepción el sistema mostrará un mensaje explicativo del error ocurrido y permanecerá en el mismo estado sin realizar ninguna otra operación.
7. El servidor de aplicaciones y de base de datos deberá mantener una seguridad mediante firewall para proteger el código y la información.

Restricciones del diseño

8. El software deberá ser desarrollado en su totalidad con tecnologías y componentes de código abierto.

Interfaz

9. Existirá un componente interno para la validación correcta de las entradas/salida de datos de cada una de las interfaces de la aplicación.
10. El sistema deberá implementar una solución arquitectónica para permitir la integración segura entre componentes internos.
11. La comunicación entre el servidor de aplicaciones y la base de datos se lleva a través del protocolo TCP/IP.

Hardware

12. Para la ejecución del sistema se requiere que la PC cliente tenga los siguientes componentes de hardware: Celeron o superior, 1024 MB RAM como mínimo.
13. La comunicación entre el cliente y el servidor de aplicaciones se lleva a través del protocolo HTTPS.

2.2.3 Descripción de requisitos

A continuación, se presentará como ejemplo las descripciones de los requisitos “Modificar tipo de viaje”, “Modificar entidad” y “Registrar tipo de transporte”, el resto de las descripciones se encuentran en el expediente de proyecto.

Número: 2	Nombre del requisito: Modificar tipo de viaje.
Programador: Dayron Alejandro Mederos Ramos	Iteración Asignada: 1ra
Prioridad: Baja	Tiempo Estimado: 1
Riesgo en Desarrollo:	Tiempo Real: 1
<p>Descripción: El usuario podrá modificar los campos que desee del tipo de viaje seleccionado:</p> <p>Nombre:</p> <ul style="list-style-type: none"> • <i>Campo requerido.</i> • <i>No admite varios espacios antes de palabras.</i> • <i>Entre letras, números y espacio o guion bajo entre las palabras.</i> • <i>Cantidad de caracteres por palabra es 30.</i> • <i>Cantidad de caracteres máxima es 250 y mínima es 2.</i> <p>Descripción:</p> <ul style="list-style-type: none"> • <i>Admite cualquier tipo de carácter.</i> • <i>Cantidad de caracteres por palabra es 30.</i> <p><i>Cantidad de caracteres máxima es 250 y mínima es 2.</i></p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> • El usuario debe estar autenticado en el sistema. • Debe existir elementos creados en el sistema. • Si el usuario modifica la información correctamente el sistema muestra el mensaje <i>“La acción ha sido realizada satisfactoriamente.”</i> y adiciona el nuevo tipo de viaje a la Lista de tipos de viaje. • Si ocurre un error durante la operación se muestra el mensaje: <i>“Ha ocurrido un error cuando intentaba realizar la acción”.</i> • En caso que el valor de un campo único exista se muestra un mensaje de error: <i>“No fue posible ejecutar la operación porque el elemento ya existe.”</i> • En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo <i>“Campo requerido”</i> encima del campo que debe ser llenado obligatoriamente. • Si el usuario introduce en los campos más caracteres de los permitidos para una palabra o deja espacios innecesarios el sistema muestra el mensaje de error <i>“Verifique la cantidad de palabras o espacios vacíos”.</i> 	

- Si el usuario introduce más caracteres de los permitidos en los campos, el sistema muestra el mensaje *“Por favor, no escribas más de <cantidad> caracteres”*.
- Si el usuario introduce menos de los caracteres permitidos para un campo, el sistema muestra el mensaje *“Por favor, no escribas menos de <cantidad> caracteres”*.
- Si el usuario desea cancelar la operación el sistema muestra el mensaje de advertencia *“¿Está seguro que desea cancelar la operación? Si el usuario selecciona la opción “Si” el sistema regresa a la interfaz que lista los tipos de viaje creados. Si el usuario selecciona la opción “No” el sistema se mantiene en la vista actual.*

Tabla 2
RF:

Prototipo elemental de interfaz gráfica de usuario:

Gestionar tipo de viaje (Modificar tipo de viaje)

Número: 4 **Nombre del requisito:** Modificar entidad.

Programador: Dayron Alejandro Mederos Ramos **Iteración Asignada:** 1ra

Prioridad: Baja	Tiempo Estimado: 1
Riesgo en Desarrollo:	Tiempo Real: 1
<p>Descripción: El usuario podrá modificar los campos que desee de la entidad seleccionada:</p> <p>Nombre:</p> <ul style="list-style-type: none"> • <i>Campo requerido.</i> • <i>No admite varios espacios antes de palabras.</i> • <i>Entre letras, números y espacio o guion bajo entre las palabras.</i> • <i>Cantidad de caracteres por palabra es 30.</i> • <i>Cantidad de caracteres máxima es 250 y mínima es 2.</i> <p>Descripción:</p> <ul style="list-style-type: none"> • <i>Admite cualquier tipo de carácter.</i> • <i>Cantidad de caracteres por palabra es 30.</i> <p><i>Cantidad de caracteres máxima es 250 y mínima es 2.</i></p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> • El usuario debe estar autenticado en el sistema. • Debe existir elementos creados en el sistema. • Si el usuario modifica la información correctamente el sistema muestra el mensaje <i>“La acción ha sido realizada satisfactoriamente.”</i> y adiciona la nueva entidad a la Lista de entidades. • Si ocurre un error durante la operación se muestra el mensaje: <i>“Ha ocurrido un error cuando intentaba realizar la acción”</i>. • En caso que el valor de un campo único exista se muestra un mensaje de error: <i>“No fue posible ejecutar la operación porque el elemento ya existe.”</i> • En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo <i>“Campo requerido”</i> encima del campo que debe ser llenado obligatoriamente. • Si el usuario introduce en los campos más caracteres de los permitidos para una palabra o deja espacios innecesarios el sistema muestra el mensaje de error <i>“Verifique la cantidad de palabras o espacios vacíos”</i>. • Si el usuario introduce más caracteres de los permitidos en los campos, el sistema muestra el mensaje <i>“Por favor, no escribas más de <cantidad> caracteres”</i>. 	

Tabla 3 RF: Gestionar entidad (Modificar entidad)

- Si el usuario introduce menos de los caracteres permitidos para un campo, el sistema muestra el mensaje *“Por favor, no escribas menos de <cantidad> caracteres”*.
- Si el usuario desea cancelar la operación el sistema muestra el mensaje de advertencia *“¿Está seguro que desea cancelar la operación? Si el usuario selecciona la opción “Si” el sistema regresa a la interfaz que lista las entidades creadas. Si el usuario selecciona la opción “No” el sistema se mantiene en la vista actual.*

Prototipo elemental de interfaz gráfica de usuario:

Número: 1 **Nombre del requisito:** Registrar tipo de transporte.

Programador: Dayron Alejandro Mederos Ramos **Iteración Asignada:** 1ra

Prioridad: Alta **Tiempo Estimado:** 1

Riesgo en Desarrollo: **Tiempo Real:** 2

Descripción:

El usuario podrá registrar un tipo de transporte seleccionando en el área de íconos flotantes la opción Registrar. Para ello debe llenar los siguientes campos:

Nombre:

- *Campo requerido.*
- *No admite varios espacios antes de palabras.*
- *Entre letras, números y espacio o guion bajo entre las palabras.*
- *Cantidad de caracteres por palabra es 30.*
- *Cantidad de caracteres máxima es 250 y mínima es 2.*

Descripción:

- *Admite cualquier tipo de carácter.*
- *Cantidad de caracteres por palabra es 30.*
- *Cantidad de caracteres máxima es 250 y mínima es 2.*

Observaciones:

- El usuario debe estar autenticado en el sistema.
- Si no existen tipos de transporte creados el listado se mostrará vacío.
- Si el usuario introduce la información correcta el sistema muestra el mensaje *“La acción ha sido realizada satisfactoriamente.”* y adiciona el nuevo tipo de transporte a la Lista de tipos de transportes.
- Si ocurre un error durante la operación se muestra el mensaje: *“Ha ocurrido un error cuando intentaba realizar la acción”.*
- En caso que el valor de un campo único exista se muestra un mensaje de error: *“No fue posible ejecutar la operación porque el elemento ya existe.”*
- En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo *“Campo requerido”* encima del campo que debe ser llenado obligatoriamente.
- Si el usuario introduce en los campos más caracteres de los permitidos para una palabra o deja espacios innecesarios el sistema muestra el mensaje de error *“Verifique la cantidad de palabras o espacios vacíos”.*
- Si el usuario introduce más caracteres de los permitidos en los campos, el sistema muestra el mensaje *“Por favor, no escribas más de <cantidad> caracteres”.*
- Si el usuario introduce menos de los caracteres permitidos para un campo, el sistema muestra el mensaje *“Por favor, no escribas menos de <cantidad> caracteres”.*
- Si el usuario desea cancelar la operación el sistema muestra el mensaje de advertencia *“¿Está seguro que desea cancelar la operación?”* Si el usuario selecciona la opción “Si” el sistema regresa a la interfaz que lista los tipos de

transporte creados. Si el usuario selecciona la opción “No” el sistema se mantiene en la vista actual.

Prototipo elemental de interfaz gráfica de usuario:

El prototipo muestra una interfaz con el título "Registrar tipo de transporte" en la parte superior izquierda y un ícono de menú en la parte superior derecha. Debajo del título, hay un campo de texto etiquetado "Nombre *" y un botón "Activo" con un ícono de interruptor. Abajo de eso, hay un campo de texto etiquetado "Descripción". En la parte inferior derecha, hay dos botones: "Aceptar" y "Cancelar".

Tabla 4 RF: Gestionar tipo de transporte (Registrar tipo de transporte)

2.3 Propuesta de solución

La propuesta de solución se basa en el desarrollo del Módulo de Transportación Nacional que formará parte del Sistema de Gestión Ciudadano de la Universidad de las Ciencias Informáticas. El módulo a desarrollar contará con nuevas funcionalidades que posibilitará el aumento de la agilidad y organización de este proceso. Estará estructurado por 4 agrupaciones funcionales para un mejor orden de la aplicación.

El grupo funcional Transportación contendrá lo relacionado a gestionar las transportaciones, viajes, salidas, transportes, bloqueos y autorizos. La agrupación de Reportes será la encargada de gestionar los diferentes tipos de reportes que genera la aplicación y la de Configuración será utilizada para gestionar diferentes datos que utiliza la aplicación para garantizar su funcionamiento, estos pueden ser: tipo de transporte, entidades, tipo de bloqueo, puntos y los tramitadores tercerizados. Por último, la agrupación Reservación gestionará las reservaciones hecha por el administrador a los usuarios UCI, a los familiares de estos y a los terceros.

Los usuarios podrán loguearse en el módulo a través del CAS-UCI o mediante un usuario registrado en el software en caso de que el servicio de autenticación no esté funcional. Los usuarios no UCI y no asociado los gestionará un tercero, asignado por el administrador, que tendrá permiso a realizar la reservación y cancelarla si estos desean.

De manera general la investigación propone un sistema capaz de gestionar proceso de una mejor manera, de cómo se realiza actualmente.

2.4 Descripción de la arquitectura

Según Pressman, la arquitectura de software de un programa o sistema de cómputo es la estructura o las estructuras del sistema, que incluyen los componentes del software, las propiedades visibles externamente de los componentes y las relaciones entre ellos. Es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de los componentes (26).

2.4.1 Patrón de arquitectura

Los patrones arquitectónicos son patrones del software, que se encargan de definir la estructura de un sistema. Estos a su vez se componen de subsistemas con sus responsabilidades, también poseen una serie de directivas para organizar los componentes del mismo sistema, con el objetivo de facilitar la tarea del diseño. Un patrón arquitectónico se enfoca en dar solución a un problema en específico y abarca solo parte de la arquitectura. Cuando un patrón de este tipo brinda una imagen general de un sistema, él no es una arquitectura como tal. Es por esto que un patrón arquitectónico es un concepto que captura elementos esenciales de una arquitectura de software (27).

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC (**Modelo-Vista-Controlador**), que está formado por tres niveles:

El **Controlador**, que contiene el código que liga la lógica de negocio con la presentación, está dividido en varios componentes que se utilizan para diversos propósitos:

- El controlador frontal es el único punto de entrada a la aplicación. Carga la configuración y determina la acción a ejecutarse.
- Las acciones contienen la lógica de la aplicación. Verifican la integridad de las peticiones y preparan los datos requeridos por la capa de presentación.
- Los objetos *request*, *response* y *session* dan acceso a los parámetros de la petición, las cabeceras de las respuestas y a los datos persistentes del usuario. Se utilizan muy a menudo en la capa del controlador.

- Los filtros son trozos de código ejecutados para cada petición, antes o después de una acción. Por ejemplo, los filtros de seguridad y validación son comúnmente utilizados en aplicaciones web (28).

La **Vista** se encarga de producir las páginas que se muestran como resultado de las acciones. La vista en Symfony está compuesta por diversas partes, estando cada una de ellas especialmente preparadas para que puedan ser fácilmente modificables por la persona que normalmente trabaja con cada aspecto del diseño de las aplicaciones (28).

El **Modelo** es responsable de acceder a la capa de almacenamiento de datos y notificar a la vista los cambios que en los datos pueda producir un agente externo si está ante un modelo activo. Este comportamiento permite un alto nivel de abstracción y una fácil portabilidad (29).

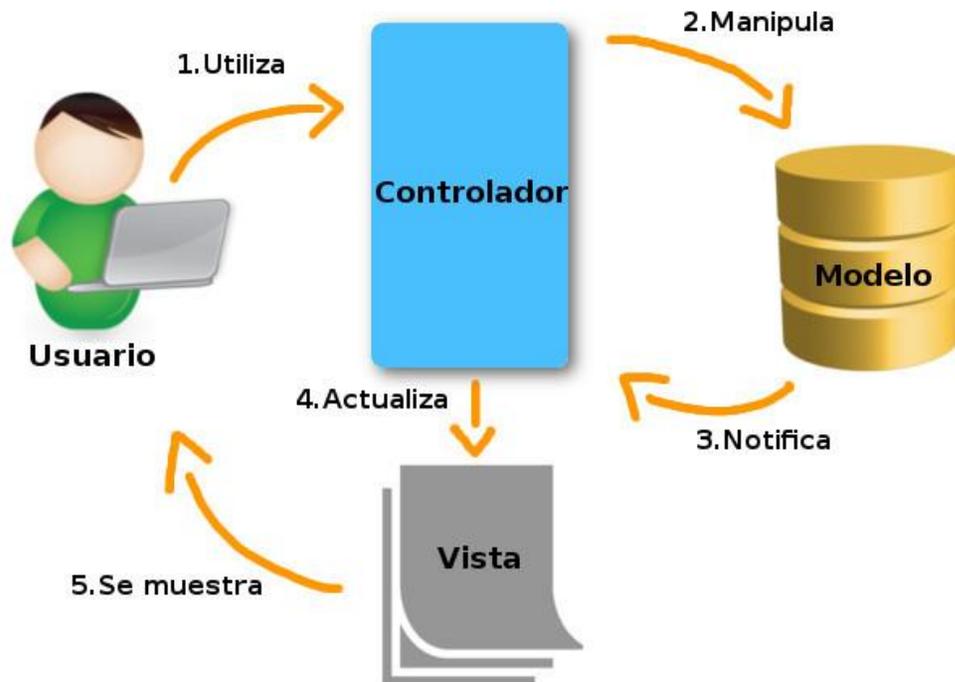


Figura 2 Modelo - Vista – Controlador

2.5 Patrones de diseño

Los patrones de diseño son la base a la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o de interfaces. Los patrones de diseño pretenden (30):

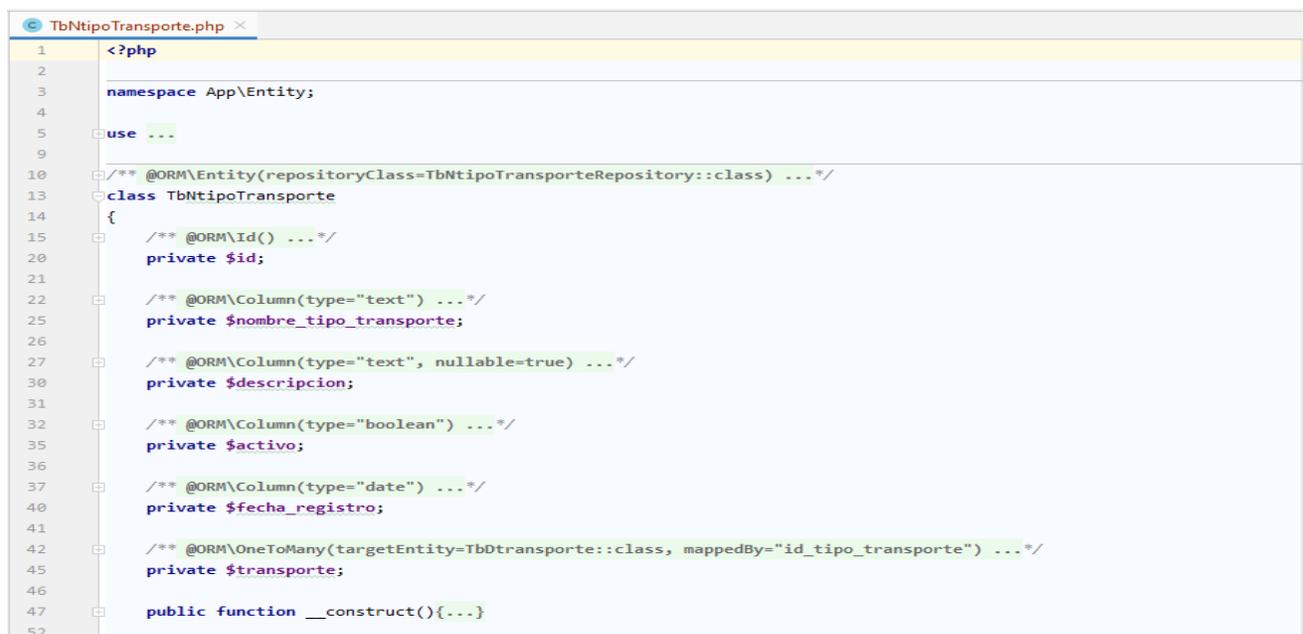
- Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Estandarizar el modo en que se realiza el diseño.

2.5.1 Patrones de software para la asignación general de responsabilidad

Los patrones de asignación general de responsabilidad (GRASP) describe los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituye un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable (31).

Los patrones GRASP se puede destacar 5 patrones principales:

- **Experto:** se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Da origen a diseños donde el objeto de software realiza las operaciones que normalmente se aplican a la cosa real que representa. Con este patrón se conserva el encapsulamiento, ya que los objetos se vale de su propia información para hacer lo que se les pide.



```
1 <?php
2
3 namespace App\Entity;
4
5 use ...
6
7
8
9
10 /** @ORM\Entity(repositoryClass=TbNtipoTransporteRepository::class) ... */
11
12
13 class TbNtipoTransporte
14 {
15     /** @ORM\Id() ... */
16     private $id;
17
18
19     /** @ORM\Column(type="text") ... */
20     private $nombre_tipo_transporte;
21
22
23     /** @ORM\Column(type="text", nullable=true) ... */
24     private $descripcion;
25
26
27     /** @ORM\Column(type="boolean") ... */
28     private $activo;
29
30
31     /** @ORM\Column(type="date") ... */
32     private $fecha_registro;
33
34
35     /** @ORM\OneToMany(targetEntity=TbDtransporte::class, mappedBy="id_tipo_transporte") ... */
36     private $transporte;
37
38
39     public function __construct(){...}
40
41
42
43
44
45
46
47
48
49
50
51
52
```

Figura 3 Patrón Experto

- **Creador:** el propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento.
- **Bajo acoplamiento:** este patrón estimula asignar una responsabilidad de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propio de un alto acoplamiento. Soporta el diseño de clases más independiente, que reducen el impacto de los cambios, y también más reutilizables, que acrecienten la oportunidad de una mayor productividad.
- **Alta cohesión:** es un patrón evaluativo que el desarrollador aplica al valorar decisiones de un diseño. Una clase de alta cohesión posee un número relativamente pequeño, con una importante funcionalidad relacionada y poco trabajo que hacer.
- **Controlador:** este patrón ofrece una guía para tomar decisiones apropiada que generalmente se aceptan.

```
TbNtipoTransporteController.php x
1  <?php
2
3  namespace App\Controller;
4
5  use ...
6
14
15  /**
16   * @Route("/tipo_transporte")
17   */
18  class TbnTipoTransporteController extends AbstractController
19  {
20      /** @Route("/", name="tb_ntipo_transporte_index", methods={"GET"}) ...*/
23      public function index(TbnTipoTransporteRepository $tbnTipoTransporteRepository, PaginatorInterface $paginator, Request $request, EntityManagerInterface $em): Response {...}
38
39      /** @Route("/registrar", name="tb_ntipo_transporte_new", methods={"GET","POST"}) ...*/
42      public function new(Request $request): Response {...}
61
62      /** @Route("/{id}", name="tb_ntipo_transporte_show", methods={"GET"}) ...*/
65      public function show(TbnTipoTransporte $tbnTipoTransporte): Response {...}
71
72      /** @Route("/{id}/editar", name="tb_ntipo_transporte_edit", methods={"GET","POST"}) ...*/
75      public function edit(Request $request, TbnTipoTransporte $tbnTipoTransporte): Response {...}
91
92      /** @Route("/{id}", name="tb_ntipo_transporte_delete", methods={"DELETE"}) ...*/
95      public function delete(Request $request, TbnTipoTransporte $tbnTipoTransporte): Response {...}
105
106  }
```

Figura 4 Patrón Controlador

2.5.2 Patrones GoF

Los patrones “Banda de los Cuatro” (Gang-of-Four) describen las formas comunes en que diferentes tipos de objetos, pueden ser organizado para trabajar uno con otros. Tratan la relación entre clases y la formación de estructura de mayor complejidad. Además, permite crear grupos de objetos para ayudar a realizar tareas complejas. Existe 3 tipos de patrones: creación, estructurales y de comportamiento. Los patrones de creación abstrae la forma en la que se crea el objeto, permitiendo tratar las clases a crear de forma genérica dejando para más tarde la decisión de que clase crear o cómo crearla (32).

- **Instancia única (Singleton):** es un patrón diseñado para restringir la creación de objetos perteneciente a una clase u objeto. Su intención consiste garantizar que una clase solo tenga una instancia y proporcionar un punto de acceso a ella. Este patrón se refleja en las clases controladora que son instancias únicas para la interacción entre competentes.
- **Mediador (Mediator):** define un objeto q coordine la comunicación entre objetos de distintas clases. Se refleja en las librerías que funciona como mediadora entre las clases controladoras y los modelos de acceso a datos.
- **Observador (Observer):** este patrón se encarga de definir dependencia entre objeto, de forma que, si alguno cambia su estado, automáticamente se notifica y se actualizan todos los objetos que dependen de él.

2.6 Patrones de diseño de base de datos

Se hace necesaria una base de datos que almacene la información que el sistema genere y necesite para funcionar correctamente. A continuación, se muestra el diagrama de BD, donde se muestra la relación de la información de los elementos con los que el sistema interactúa.

los protocolos de comunicación que se utilizan para acceder a cada uno. En la siguiente figura puede visualizarse el diagrama de despliegue definido para la solución propuesta:

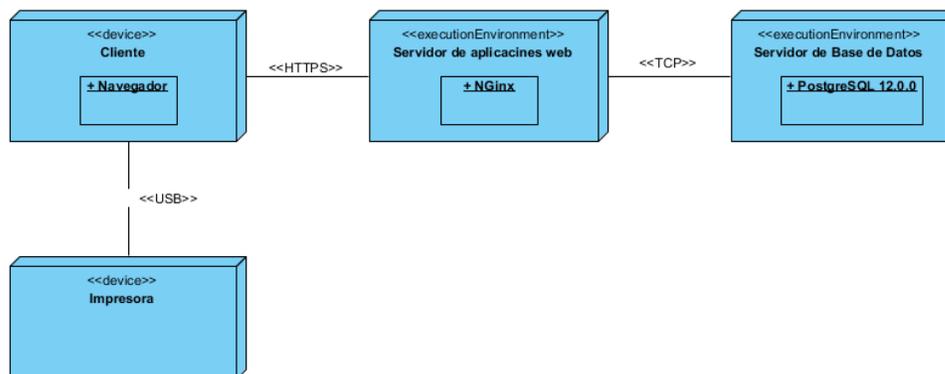


Figura 6 Diagrama de despliegue

El diagrama de despliegue representado muestra la siguiente distribución:

Cliente: dispositivo cliente capaz de conectarse al servidor de aplicaciones mediante el protocolo de comunicaciones HTTPS.

Servidor de aplicaciones web: computadora en que se encuentra el servidor web NGinx, este será el lugar en que se gestione todo el contenido de la aplicación. El mismo establecerá comunicación con los ordenadores clientes mediante protocolo HTTPS y con el servidor de base de datos por medio del protocolo TCP.

Servidor de base de datos: ordenador en que se encuentra el gestor de base de datos PostgreSQL capaz de mantener persistente la información generada y a utilizar.

HTTPS: protocolo de transferencia de hipertexto seguro, por sus siglas en inglés, *Hypertext Transfer Secure Protocol* (HTTPS), es un protocolo de red basado en HTTP por lo que está orientado a transacciones sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores y sigue el esquema petición-respuesta entre un cliente y un servidor (33).

TCP/IP: base de Internet y sirve para enlazar computadoras que utilizan diferentes sistemas operativos. Familia de protocolos utilizada para la conexión entre el servidor web y el servidor donde se encuentra ubicada la base de datos (33).

USB: Universal Serial Bus (USB) (bus universal en serie) es un estándar industrial desarrollado a mediados de los años 1990 que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre ordenadores y periféricos y dispositivos electrónicos (34).

Conclusiones parciales

Con el desarrollo de este capítulo se concluye la identificación de los requisitos necesarios e importantes para el cliente. El modelo físico de datos permite describir la estructura de dato de la base. Además, con la utilización de la arquitectura MVC y el uso de los patrones de diseño, se garantizará una mayor organización y se obtiene una propuesta de solución que responde a las demandas de la Dirección de Transportación Nacional de la UCI.

CONCLUSIONES GENERALES

Producto de la investigación desarrollada se arriba a las siguientes conclusiones:

- El estudio de la transportación Nacional de la UCI permitió entender mejor el sistema, destacándose los principales problemas de su funcionamiento.
- Mediante el análisis a sistemas con características similares al del problema de investigación, se identificó la necesidad de proponer un nuevo sistema que se ajustara a las necesidades actuales de la universidad. Además, permitió destacar características que fueron útiles a la propuesta de solución.
- La selección de las tecnologías y herramientas como PostgreSQL, Symfony, Bootstrap, Pencil y PHPStorm, son las indicadas a emplear en la propuesta de software, ya que estas se integran con facilidad y aún mantienen soporte por sus creadores.
- El empleo de un proceso de desarrollo de enfoque ágil, favoreció la obtención de requerimientos aplicados a la propuesta de solución, con requisitos funcionales y no funcionales debidamente identificados.
- Se logra una propuesta de software para la administración de la transportación nacional de la Universidad de las Ciencias Informáticas que cumple con sus requerimientos.

REFERENCIAS BIBLIOGRÁFICAS

1. Historia | Intranet 3.0. [online]. [Accessed 7 December 2019]. Available from: <https://intranet.uci.cu/universidad/historia>
2. Estructura | Intranet 3.0. [online]. [Accessed 8 December 2019]. Available from: <https://intranet.uci.cu/universidad/estructura>
3. *Metodologia-de-la-Investigaci3n_Sampieri.pdf* [online]. [Accessed 24 August 2020]. Available from: https://www.uv.mx/personal/cbustamante/files/2011/06/Metodologia-de-la-Investigaci%C3%83%C2%B3n_Sampieri.pdf
4. El *transportation management system* (TMS) en la logística. [online 2019]. Available from: <https://www.zonalogistica.com/el-transportation-managementsystem-tms-en-la-logistica-2/>
5. LUJÁN MORA, Sergio. *Programación de aplicaciones web: historia, principios básicos y clientes web*. Alicante : Club Universitario, 2002. ISBN 84-8454-206-8.
6. ÁNGEL COBO, PATRICIA GÓMEZ, DANIEL PÉREZ and ROCIO ROCHA. *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web*. Díaz de Santos, 2005. ISBN 84-7978-706-6.
7. ¿Qué es un sistema de gestión y para qué sirve? - Caltic Consultores. *Caltic Consultores* [online]. [Accessed 7 December 2019]. Available from: <https://www.calticconsultores.com/articulos/sistema-gestion-sirve.html>
8. CONSULTORESINTEGRA. Sistemas de Gestión | Integra. [online]. [Accessed 8 December 2019]. Available from: <https://www.consultoresdesistemasdegestion.es/sistemas-de-gestion/>
9. SOFTWARE para AGENCIAS de VIAJES y TURISMOS Reserva PASAJES en BUS. [online 2019]. Available from: www.softwarejr.com.ar
10. Omni-Channel airline merchandising and operational solutions for airlines. [online 2019]. Available from: www.radixx.com
11. C.D AVIACIÓN. Cubana de Aviacion Airline Profile | CAPA. [online]. [Accessed 7 December 2019]. Available from: <https://centreforaviation.com/data/profiles/airlines/cubana-de-aviacion-cu>
12. Campus - Transportación - Transportación. [online 2019]. Available from: <https://campus.uci.cu/trans/ppc111>
13. What is Symfony. [online]. [Accessed 7 December 2019]. Available from: <https://symfony.com/what-is-symfony>
14. Symfony, High Performance PHP Framework for Web Development. [online]. [Accessed 8 December 2019]. Available from: <https://symfony.com/> Symfony is a set of reusable PHP components and a PHP framework to build web applications, APIs, microservices and web services.
15. Conoce todo acerca de Bootstrap y sus Características. [online]. [Accessed 30 August 2020]. Available from: <https://www.nextu.com/blog/que-es-bootstrap/>

16. Learn to style HTML using CSS. *MDN Web Docs* [online]. [Accessed 8 December 2019]. Available from: <https://developer.mozilla.org/en-US/docs/Learn/CSS>
17. JORGE SÁNCHEZ. *Java Script manual de referencia*. Ciudad de Mexico D.F, 2003.
18. EGUÍLUZ PÉREZ, Javier. *Introducción a CSS*. 2009.
19. PostgreSQL: The world's most advanced open source database. [online]. [Accessed 8 December 2019]. Available from: <https://www.postgresql.org/>
20. PostgreSQL: PostgreSQL 12 Released! [online]. [Accessed 30 August 2020]. Available from: <https://www.postgresql.org/about/news/1976/>
21. Pencil V3: El renovado maquetador de interfaces gráficas – Desde Linux. [online]. [Accessed 30 August 2020]. Available from: <https://blog.desdelinux.net/pencil-v3-maquetador-de-interfaces-graficas/amp/>
22. Visual Paradigm 15.1 Released. [online 2020]. Available from: www.visual-paradigm.com/tw/aboutus/newreleasesvp151.jsp
23. Metodología SCRUM: ¿qué es y cómo aplicarlo en tu proyecto? *Gestor de proyectos online* [online]. [Accessed 16 January 2020]. Available from: <https://www.sinnaps.com/blog-gestion-proyectos/metodologia-scrum>
24. GEOFFREY SPARKS. *Introducción al modelado de sistemas de software*. 2015.
25. IAN SOMMERVILLE. *Ingeniería del software* [online]. Séptima edición. 2008. [Accessed 19 February 2020]. ISBN 978-0-321-31379-9. Available from: https://books.google.com/books/about/Ingenier%C3%ADa_del_software.html?hl=es&id=gQWd49zSut4C
26. PRESSMAN, Roger S. *Ingeniería del software: un enfoque práctico* [online]. 2013. [Accessed 25 February 2020]. ISBN 978-1-4562-1836-2. Available from: http://www.ingebook.com/ib/NPcd/IB_BooksVis?cod_primaria=1000187&codigo_libro=4272
27. GAMMA, Erich, HELM, Richard, JOHNSON, Ralph and VLISSIDES, John. *Design Patterns : Elements of Reusable Object-Oriented Software*. . P. 431.
28. Symfony 1.1, la guía definitiva. [online]. [Accessed 25 February 2020]. Available from: <https://uniwebsidad.com/libros/symfony-1-1?from=libroswebLibrary> Catalog: uniwebsidad.com
29. YENISLEIDY FERNÁNDEZ ROMERO and YANETTE DÍAZ GONZÁLEZ. *Patrón Modelo-Vista-Controlador*. [online 2019]. Vol. 11. Available from: <http://revistatelematica.cujae.edu.cu/index.php/tele>
30. Patrones de diseño – Ingeniería de software. [online]. [Accessed 29 August 2020]. Available from: <https://ittgweb.wordpress.com/2016/05/29/3-2-patrones-de-diseno/>
31. MORA, Roberto Canales. *Patrones de GRASP. Adictos al trabajo* [online]. 22 December 2003. [Accessed 30 August 2020]. Available from: <https://www.adictosaltrabajo.com/2003/12/22/grasp/Os> presentamos una introducción a los patrones de asignación de responsabilidades y su relación con el proceso unificado.

32. *UML y Patrones (Craig Larman).WWW.FREELIBROS.COM.pdf.*
33. *Stallings-william-comunicaciones-y-redes-de-computadores.pdf* [online]. [Accessed 26 February 2020]. Available from: <https://richardfong.files.wordpress.com/2011/02/stallings-william-comunicaciones-y-redes-de-computadores.pdf>
34. Boston Globe Online / Business / USB merece más soporte. [online]. [Accessed 26 February 2020]. Available from: https://simson.net/clips/1999/99.Globe.05-20.USB_deserves_more_support+.shtml