



Facultad de Ciencias y Tecnologías Computacionales

Módulo para la gestión del dashboard para el Sistema de Gestión para el Ingreso a la Educación Superior.

trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor(es): Nayalis González Munibes

Tutor(es): Roisbel Portales López

Co-tutor: Ernesto Gutierrez Ramos

Consultante: Sandy Guerra Fernández

La Habana, junio de 2022

Año 53 de la Revolución

Declaración de Auditoría

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor del presente trabajo de diploma con título “Módulo para la gestión de componentes dashboard para el Sistema para la Gestión del Ingreso a la Educación Superior (SIGIES)” y autorizo 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 _____ .

Firma de autor

Nayalis González Munibes

Firma del Tutor

Ing. Roisbel Portales López

Firma del Cotutor

Ing. Ernesto Gutierrez Ramos

Firma del Consultante

Ing. Sandy Guerra Fernández

Datos de Contacto

DATOS DE CONTACTO

Roisbel Portales López: Graduado de Ingeniero en Ciencias Informáticas desde junio del 2016, en la Universidad de las Ciencias Informáticas (UCI). Actualmente ocupa la plaza de Especialista A en Ciencias Informáticas en el Centro de Tecnologías para la Formación (FORTES) perteneciente a la UCI, desempeñando el rol de jefe de proyecto en el proyecto Desarrollo de herramientas informáticas para el Ministerio de Educación Superior. En los inicios de su carrera laboral comenzó desempeñando el rol de Desarrollador de aplicaciones y a finales de 2018 cambia de rol por el antes mencionado. Correo electrónico: roisbel@uci.cu. Perfil en redes profesionales: LinkedIn Roisbel Portales López.

Ernesto Gutierrez Ramos: Graduado de Ingeniero en Ciencias Informáticas desde junio del 2015, en la Universidad de las Ciencias Informáticas (UCI). Actualmente ocupa la plaza de Especialista A en Ciencias Informáticas en el Centro de Tecnologías para la Formación (FORTES) perteneciente a la UCI, desempeñando el rol de arquitecto de software en el proyecto Desarrollo de herramientas informáticas para el Ministerio de Educación Superior. En los inicios de su carrera laboral comenzó desempeñando el rol de Desarrollador de aplicaciones y a finales de 2021 cambia de rol por el antes mencionado. Correo electrónico: egutierrez@uci.cu.

Sandy Guerra Fernández: Graduado de Ingeniero en Ciencias Informáticas desde junio del 2012, en la Universidad de las Ciencias Informáticas (UCI). Actualmente ocupa la plaza de Especialista A en Ciencias Informáticas en el Centro de Tecnologías para la Formación (FORTES) perteneciente a la UCI, desempeñando el de administrador de la Calidad en el proyecto Desarrollo de Sistema de Gestión Académica XAUCE AKADEMOS para el MINED. En los inicios de su carrera laboral desempeñó distintos roles como: Administrador de la Calidad, Administrador de la Configuración, Analista y Jefe de proyecto Correo electrónico: sguerra@uci.cu.

Declaración de Auditoría

Agradecimientos

El primer agradecimiento y más importante va dirigido a mis padres que siempre estuvieron presentes impulsándome por el buen camino en estos años de universidad, confiando en mí y apoyándome incondicionalmente logrando que me forjara profesionalmente. Espero que se sientan muy orgullosos.

A mi hermana Nélide que la quiero con la vida y por a verme regalado dos sobrinos hermosos que son Cristian y Estefany.

A toda mi Familia por preocuparse siempre por mí.

A los amigos que han perdurado todo este tiempo, a los que me regaló la UCI y a los compañeros de grupo que siempre estuvieron presentes y de una forma y otra compartimos estos años de universidad.

A mis tutores por su apoyo incondicional guiándome durante el desarrollo de la investigación.

A todos los profesores que de una forma u otra siempre tuve su apoyo en la contribución a mi formación como profesional.

A la UCI por todos esos momentos inolvidables que me regaló y por permitirme graduarme en esta universidad siendo una mejor persona.

Resumen

Los Indicadores claves de desempeño son realmente importantes en las empresas en el logro de los resultados deseados, su comprensión y mejora, así como, en la medición del desempeño del negocio. Por tal motivo el Sistema para la Gestión del Ingreso a la Educación Superior (SIGIES) necesita un espacio donde se visualicen los indicadores claves que intervienen en dicho sistema.

Por tal motivo en la presente investigación se realiza una propuesta de solución basada en la inteligencia del negocio teniendo como objetivo fundamental desarrollar un módulo para gestionar componentes que permitan mostrar información de los procesos realizados en el SIGIES de forma concisa, apoyada en los indicadores claves de desempeño (KPI's) con el marco de trabajo Symfony en su versión, usando tecnologías libres. Para guiar el desarrollo de este producto se seleccionó la metodología AUP adaptado a la Universidad de las Ciencias Informáticas en las etapas de análisis, diseño, implementación y validación. Como resultado se obtuvo un módulo que permite visualizar de forma automática los indicadores claves de desempeño que intervienen en dicho sistema. El desarrollo fue basado sobre políticas de software libre empleando las tecnologías Apache, PostgreSQL, PHP, XMPP, JavaScript.

Palabras claves: módulos, componentes, sistema, indicadores claves de desempeño, negocio.

Abstract

ABSTRACT

The key performance indicators are really important in companies in achieving the desired results, their understanding and improvement, as well as in measuring business performance. For this reason, the Higher Education Admission Management System (SIGIES) needs a space where the key indicators involved in said system are displayed.

For this reason, in the present investigation, a solution proposal is made based on business intelligence, with the fundamental objective of developing a module to manage components that allow information to be displayed on the processes carried out in the SIGIES in a concise way, supported by the key indicators of performance (KPI's) with the Symphony framework in its version, using free technologies. To guide the development of this product, the AUP methodology adapted to the University of Informatics Sciences was selected in the stages of analysis, design, implementation and validation. As a result, a module was obtained that allows the automatic visualization of the key performance indicators that intervene in said system. The development was based on free software policies using Apache, PostgreSQL, PHP, XMPP, JavaScript technologies.

Keywords: *modules, components, system, key performance indicators, business.*

INTRODUCCIÓN.....	5
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
INTRODUCCIÓN.....	6
1.1 Estado del arte.....	6
1.1.1. Inteligencia de negocio.....	6
1.1.2. Indicadores claves de desarrollo o medidor de desempeño (KPI's).....	8
1.1.3. Dashboard.....	9
1.1.4. Sistemas de gestión.....	11
1.1.5. Sistemas en tiempo real.....	12

Abstract

1.2	Sistemas similares estudiados.....	12
1.2.1	Score.....	12
1.2.2	Sisense.....	13
1.2.3	Tableau.....	13
1.2.4	Power Bi.....	14
1.2.5	Acunetix.....	15
1.3	Análisis de las posibles tecnologías, técnicas, herramientas y lenguajes a utilizar.	16
1.3.1	Metodología de desarrollo de software.....	16
1.3.2	Marco de trabajo.....	17
1.3.3	Selección el framework para la manipulación de PHP.....	18
1.3.4	Lenguajes de programación.....	19
1.3.5	Lenguaje de modelado.....	20
1.3.6	Sistemas Gestores de Base de datos.....	20
1.3.7	Servidor web.....	21
1.4	Conclusiones Parciales.....	23
CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL DASHBOARD PARA SIGIES.....		24
INTRODUCCIÓN.....		24
2.1	Descripción de la propuesta de solución.....	24
2.2	Actores del sistema.....	25
2.3	Requisitos del sistema.....	26
2.3.1	Requisitos Funcionales.....	26
2.3.2	Requisitos no funcionales.....	30
2.4	Historia de Usuario.....	31
2.5	Arquitectura del sistema.....	37
2.5.1	Patrón arquitectónico Modelo Vista Controlador.....	37
2.5.2	Patrones de diseño del sistema.....	39
2.6	Diagrama de clases.....	42
2.7	Diagrama de componentes.....	43
	44
2.8	Diagrama de Despliegue.....	44
2.9	Conclusiones Parciales.....	45
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS A LA SOLUCIÓN PROPUESTA.....		46
INTRODUCCIÓN.....		46
3.1	Implementación.....	46

Abstract

3.1.1 Estándares de codificación.....	46
3.2 Pruebas de software.....	49
3.2.1 Prueba de Aceptación.....	50
3.2.2 Prueba de Integración.....	52
3.2.3 Resultado de las pruebas de Caja Negra.....	54
3.3 Conclusiones Parciales.....	55
CONCLUSIONES GENERALES.....	57
RECOMENDACIONES.....	58
BIBLIOGRAFÍA.....	59
ANEXOS.....	62

Índice

de Tablas

ÍNDICE DE TABLAS Y

Tabla 1: Fases de la variación de AUP para la UCI.....	24
Tabla 2: Actores del sistema.....	32
Tabla 3: Requisitos Funcionales del sistema.....	33
Tabla 4: Requisitos no funcionales.....	37
Tabla 5: Historia de Usuario “Mostrar total de estudiantes que alcanzaron carreras por provincia.”.....	38
Tabla 6: Historia de Usuario “Mostrar total de plazas otorgadas en el plan de plazas de una comisión.”.....	40
Tabla 7: Historia de Usuario “Mostrar datos del componente dashboard”.....	41
Tabla 8: Historia de Usuario “Listar componentes del dashboard.”.....	42
Tabla 9: Historia de Usuario “Exportar componente del dashboard.”.....	44

Índice

de Figuras

ÍNDICE DE FIGURAS

Figura 1: Patrón Arquitectónico Modelo Vista Controlador (45).....	46
Figura 2: Diagrama que representa el uso del Patrón Experto en el sistema.....	49
Figura 3: Diagrama que representa el uso del Patrón Creador en el sistema.....	50
Figura 4: Diagrama que representa el uso del Patrón Creador en el sistema.....	50
Figura 5: Diagrama de clases del Diseño.....	51
Figura6: Diagrama de Despliegue.....	52
Tabla 7: Caso de prueba de aceptación para la HU "Editar componentes del dashboard"....	56
Tabla 8: Caso de prueba de aceptación para la HU "Exportar Componentes del dashboard".	57
Figura 9: Caso de prueba de aceptación para la HU " Listar componente del dashboard"..	57
Figura 10: Integración de los gráficos de barras en el Dashboard del SIGIES. Fuente: (elaboración propia).....	58

Introducción

Introducción

Las empresas usualmente generan Bases de datos con toda la información relevante de sus procesos, clientes, finanzas, entre otros. Estos registros han alcanzado grandes niveles de automatización y eficacia gracias a las herramientas que están disponibles para la gestión empresarial, como un Almacén de Datos[CITATION byB20 \l 3082]. El uso de esta herramienta para las empresas es importante porque las ayuda en su crecimiento y desarrollo de información; una de las herramientas que puede solucionar esta problemática ejemplo de ellos son los dashboard.

Un dashboard es una herramienta de gestión de la información que sirve para monitorizar, analizar y mostrar de manera visual los [indicadores clave de desempeño \(KPI\)](#), métricas y datos fundamentales para hacer un seguimiento del estado de una empresa, un departamento, una campaña o un proceso específico.[CITATION Dan21 \l 3082]

El Sistema de Gestión para el Ingreso a la Educación Superior (SIGIES) se encarga de gestionar la información del proceso de Ingreso a la Educación Superior utilizando tecnologías libres. Dicho sistema gestiona toda la información correspondiente a la organización y realización de los exámenes de ingreso a este nivel educacional. Además, permite la gestión de varios subprocesos que son claves para su desarrollo, como, por ejemplo: Organización, Exámenes, Asignación y Otorgamiento comprendidos en el proceso de Ingreso a la Educación Superior. A pesar de las ventajas que brinda SIGIES, no cuenta con un espacio donde se pueda visualizar los indicadores claves que intervienen en la consecución de los objetivos del proceso de ingreso. Estas son métricas que ayudan a analizar el rendimiento de una determinada acción o estrategia, ejemplo de estas son:

- ❖ El total de plazas iniciales en el plan de plazas de todo el país.
- ❖ El total de plazas otorgadas en el plan de plazas de todo el país y en el plan de plazas de una comisión.

Introducción

- ❖ El total de plazas extras en el plan de plazas de todo el país y en el plan de plazas de una comisión.
- ❖ El total de plazas sin cubrir en el plan de plazas de todo el país y en el plan de plazas de una comisión.
- ❖ El total de solicitudes por fuentes de ingreso del país y por fuente de ingreso de una comisión.
- ❖ El total de examinados por fuentes de ingreso del país y por fuentes de ingreso de una comisión.
- ❖ El total de examinados por exámenes de una comisión y por exámenes de todo el país.
- ❖ El total de examinados que cumplen por comisión y los que no cumplen.
- ❖ El total de estudiantes que alcanzaron carreras por provincia y los que no alcanzaron.
- ❖ La cantidad de estudiantes que incrementaron su nota, los que la disminuyeron y los que se mantuvieron.
- ❖ El total de estudiantes que solicitaron recalificación por provincia.

Actualmente, cada Comisión de Ingreso Provincial (CIP) y la Dirección de Ingreso y Ubicación Laboral (DIUL), para consultar información del proceso de Ingreso a la Educación Superior, referente a estos indicadores claves, deben acceder a cada uno de los reportes relacionados con los datos que necesiten analizar. Debido a que existen más de 160 reportes resulta engorroso realizar una evaluación sobre en qué momento se encuentra actualmente el proceso, así como llevar un análisis estadístico de los datos. A su vez esto trae como consecuencia pérdida de tiempo en la gestión y control de dicho proceso.

Teniendo en cuenta la descripción anterior se deriva el **problema de investigación** de la siguiente manera:

Introducción

¿Cómo mostrar información de indicadores de los procesos gestionados en el SIGIES de forma concisa para una mejor visualización de los datos?

El **objeto de estudio** se enmarca en la inteligencia de negocios en los sistemas de gestión mientras que el **campo de acción** se encuentra en los componentes de los dashboard. Entonces se plantea como **objetivo general**: Desarrollar un módulo para gestionar el dashboard que permita mostrar información de los procesos realizados en el SIGIES de forma concisa. Para cumplir con el objetivo general se plantean los siguientes **objetivos específicos**:

- ❖ Elaborar el marco teórico de la investigación para obtener los fundamentos sobre los dashboard, así como las tecnologías empleadas.
- ❖ Realizar el análisis, diseño e implementación de los componentes del dashboard del SIGIES, para mostrar información de manera concisa de los procesos realizados en el sistema.
- ❖ Aplicar pruebas de software a la solución propuesta para verificar la calidad de los componentes del dashboard del SIGIES.

Para dar cumplimiento a objetivos específicos propuestos se utilizan los métodos científicos:

Métodos Teóricos

Histórico lógico se utilizó para estudiar la evolución histórica del desarrollo de los dashboard y comprobar teóricamente su evolución de información, así como las herramientas, los lenguajes y las metodologías empleadas.

Analítico-sintético se utilizó para descomponer en partes el problema y analizar desde diferentes puntos todos los conceptos asociados a él y poder llegar a una conclusión práctica.

Introducción

Métodos Empíricos

Entrevista se empleó para lograr entender todo el proceso de negocio y realizar el levantamiento de requisitos del sistema que se necesitan para el desarrollo de la aplicación.

Observación se tuvo en cuenta a la hora de analizar las características de la propuesta de solución y de los sistemas similares, con el objetivo de lograr una correcta implementación de los componentes del dashboard.

Estructura Capítular.

La presente tesis está compuesta por tres capítulos que cubren cierta información sobre Fundamentos y referentes Teóricos-Metodológicos sobre el objetivo de estudio, Análisis y Diseño, Implementación del sistema y Validación de la solución, también las conclusiones finales, recomendaciones, referencias bibliográficas y los anexos. Para una mejor comprensión de la investigación, se decidió definir una estructura capítular que aporte cierto grado de organización y facilite el estudio del documento. El presente trabajo de diploma se encuentra estructurado en tres capítulos los cuales son descritos a continuación:

CAPITULO 1: Fundamentación teórica

Se aborda una serie de conceptos donde argumentamos sobre el objeto de estudio, el campo de acción y definiciones, metodologías y herramientas empleadas para la elaboración de la solución propuesta de la Metodología a utilizar.

CAPITULO 2: Análisis y diseño de la propuesta de solución

En este capítulo se mostrará información de las herramientas propuestas. A continuación, se realizará la descripción de todos los procesos que componen el objeto de estudio mediante los cuales se elaboran, enumeran y describen los requisitos tanto funcionales como no funcionales que serán necesarios para realizar la propuesta de solución, y seguido de esto, su modelado resultante del análisis y diseño conteniendo descripción de la arquitectura. Luego realiza el diseño de los mecanismos para el almacenamiento, procesamiento y

Introducción

transmisión de los datos, y la presentación del diseño del tratamiento de los errores y el diseño del despliegue para darle solución a los objetivos propuestos.

CAPITULO 3: Propuesta de solución

El análisis de la propuesta del sistema mostrada en el capítulo anterior, y ayuda al programador a entender el sistema y poder ser capaz de desarrollar su construcción y validación. Además de su implementación, se realizan y describen un conjunto de pruebas de software a la aplicación, en este caso se utilizarán las pruebas funcionales, más específicamente el nivel de pruebas de aceptación y pruebas de integración con el objetivo de cumplir con los requisitos planteados por el cliente.

Introducción

Capítulo 1: Fundamentación teórica

Capítulo 1: Fundamentación teórica.

Introducción

En el presente capítulo se aborda aspectos relacionados con los fundamentos y referentes teóricos-metodológicos donde son argumentados sobre el objeto de estudio, el campo de acción y, definiciones, metodología y herramientas empleadas para la elaboración de la solución propuesta.

1.1 Estado del arte.

1.1.1. Inteligencia de negocio.

La inteligencia de Negocio es el conjunto de procesos, aplicaciones y tecnologías que facilitan la obtención rápida y sencilla de datos provenientes de los sistemas de gestión empresarial para su análisis e interpretación, de manera que puedan ser aprovechados para la toma de decisiones y se conviertan en conocimientos para los responsables del negocio. [CITATION Jul21 \l 3082]

Las herramientas para un negocio inteligente por lo general muestran la información en forma de cuadros de mando o “dashboards” y reportes específicos que se pueden crear a partir de los datos que se obtienen del ERP¹ que la empresa utiliza para su gestión, de tal forma que la información es presentada al usuario de manera ágil y accesible para que pueda realizarse el análisis e interpretación correspondiente. [CITATION Jul21 \l 3082]

Beneficios de Inteligencia de Negocio (Business Intelligence):

BI proporciona muchos beneficios a las empresas que lo utilizan. Puede eliminar muchas de las conjeturas dentro de una organización, mejorar la comunicación entre los departamentos

¹ERP: Software ERP, se refiere a *Enterprise Resource Planning*, que significa “sistema de planificación de recursos empresariales”. Estos programas se hacen cargo de distintas operaciones internas de una empresa, desde producción a distribución o incluso recursos humanos.

Capítulo 1: Fundamentación teórica

mientras se coordinan las actividades y permitir que las empresas respondan rápidamente a los cambios en las condiciones financieras, las preferencias de los clientes y las operaciones de la cadena de suministro. BI mejora el rendimiento general de la empresa que lo utiliza. Las empresas han reconocido la importancia de la inteligencia empresarial para las masas. Algunos de ellos se enumeran a continuación:[CITATION Dur \l 3082]

- ❖ Con herramientas superiores de BI, ahora los empleados también pueden convertir fácilmente su conocimiento comercial a través de la inteligencia analítica para resolver muchos problemas comerciales, como aumentar las tasas de respuesta de correo directo, teléfono, correo electrónico y campañas de marketing enviadas por Internet.
- ❖ Con BI, las empresas pueden identificar a sus clientes más rentables y las razones subyacentes de la lealtad de esos clientes, así como identificar futuros clientes con un potencial comparable, si no mayor.
- ❖ Analice los datos del flujo de clics para mejorar las estrategias de comercio electrónico.
- ❖ Detecte rápidamente los problemas informados por la garantía para minimizar el impacto de las deficiencias en el diseño del producto.
- ❖ Descubrir actividades delictivas de lavado de dinero.
- ❖ Analizar el crecimiento potencial de la rentabilidad de los clientes y reducir la exposición al riesgo a través de una calificación crediticia financiera más precisa de sus clientes.
- ❖ Determinar qué combinaciones de productos y líneas de servicio es probable que compren los clientes y cuándo.

Capítulo 1: Fundamentación teórica

- ❖ Analizar ensayos clínicos de fármacos experimentales.
- ❖ Establecer tarifas más rentables para las primas de seguros.
- ❖ Reducir el tiempo de inactividad de los equipos mediante la aplicación de mantenimiento predictivo.
- ❖ Determinar con análisis de deserción y abandono por qué los clientes se van a la competencia y/o se convierten en clientes.
- ❖ Detectar y evitar comportamientos fraudulentos, como picos de uso cuando se roban tarjetas de crédito o telefónicas.
- ❖ Identificar nuevos compuestos farmacológicos moleculares prometedores.

1.1.2. Indicadores claves de desarrollo o medidor de desempeño (KPI's).

¿Qué es KPI, qué significa y cuál es su traducción?

El término KPI², hace referencia a una serie de métricas que se utilizan para sintetizar la información sobre la eficacia y productividad de las acciones que se lleven a cabo en un negocio con el fin de poder tomar decisiones y determinar aquellas que han sido más efectivas a la hora de cumplir con los objetivos marcados en un proceso o proyecto concreto.

[CITATION Bla17 \l 3082]

Los KPI también son conocidos como indicadores de calidad o indicadores clave de negocio que pueden ser utilizados y aplicables en cualquier área de negocio y sector productivo,

²KPI's: Indicador Clave de Desempeño o Medidor de Desempeño.

Capítulo 1: Fundamentación teórica

aunque son utilizados de una forma muy habitual en el marketing online. “El objetivo último de un KPI es ayudar a tomar mejores decisiones respecto al estado actual de un proceso, proyecto, estrategia o campaña y de esta forma, poder definir una línea de acción futura.”[CITATION Bla17 \l 3082]

Algunas de sus ventajas serían las siguientes: [CITATION Bla17 \l 3082]

- ❖ Los KPI's en gran medida gracias a la mejora de las nuevas tecnologías de la información (Big Data: bases de datos, sistemas de automatización y computarización de datos, etc.), ofrecen frente a los canales tradicionales en los que la cuantificación de determinadas cuestiones es tediosa y complicada, una medición mucho más sencilla, objetiva y fiable.
- ❖ Actualmente existen multitud de herramientas analíticas especializadas que permiten medir todo tipo de KPI's en cualquier sector o actividad y conocer una mayor cantidad de datos objetivos sobre aquello que se quiere cuantificar. Normalmente este tipo de herramientas están muy enfocadas para su uso en un campo de acción concreto por lo que siempre utilizarán los KPI's más oportunos y adecuados en cada caso lo cual facilita enormemente el trabajo de:
 - ❖ Elección de KPI's.
 - ❖ Localización de datos objetivos, medibles y relevantes para aquello que se quiere medir.
 - ❖ Obtención de informes sintéticos sobre aquello que se quiere determinar.

1.1.3. Dashboard.

Capítulo 1: Fundamentación teórica

Un dashboard o un Cuadro de mando es una herramienta utilizada para la gestión de la información y para inteligencia empresarial, que organiza, almacena datos desde varias fuentes y los muestra gráficamente de una forma en que resulta fácil entenderlos y analizarlos.[CITATION Mei22 \l 3082]

Características de un dashboard:

- ❖ Ser flexible y escalable: este debe ofrecer la posibilidad de agregar datos extra de forma práctica y efectiva, además de mostrar un mayor volumen de movimientos e informaciones, si el usuario así lo amerita. [CITATION SEO22 \l 3082]
- ❖ Incluir recursos gráficos: Un dashboard debe ser una herramienta que permita observar datos generales y amplios de forma cualitativa y simple. [CITATION SEO22 \l 3082]
- ❖ Contar con un diseño coherente e intuitivo: un dashboard correctamente diseñado divide sus funcionalidades en diferentes secciones o apartados y cuenta con una sólida arquitectura y distribución de la información para facilitar la interacción con esta y la rápida comprensión de la misma. [CITATION SEO22 \l 3082]
- ❖ Ser accionable: Un dashboard funcional y útil es un tablero accionable, es decir, permite desarrollar funciones y acciones dentro del mismo y no solo visualizar información. Un ejemplo de estas acciones es el acceso de ciertos renglones de información para visualizar datos con mayor detalle y amplitud. También, debe contar con ventanas y funciones para comparar información y realizar cálculos o determinar valores estadísticos.[CITATION SEO22 \l 3082]

Algunas de las ventajas de emplear un dashboard son:

- ❖ Ayuda para la toma de decisiones.

Capítulo 1: Fundamentación teórica

- ❖ Identificar oportunidades de negocio.
- ❖ Fácil de compartir.
- ❖ Uso comercial y personal. [CITATION MarcadorDePosición1 \l 3082]

Algunas de las desventajas de un Dashboard:

- ❖ Los cuadros de mando pueden resultar abrumadores, sobre todo si intentan abarcar demasiada información. Puede ser difícil para los usuarios saber dónde buscar y qué es lo más importante.
- ❖ Los cuadros de mando suelen ser estáticos, lo que significa que no se actualizan en tiempo real a medida que cambian los datos. Esto puede hacer que sean menos útiles para el seguimiento de las tendencias que evolucionan rápidamente.
- ❖ Los cuadros de mando pueden ser difíciles de personalizar, lo que significa que los usuarios no pueden adaptarlos a sus necesidades específicas. Por estas razones, los cuadros de mando deben utilizarse con cuidado y solo cuando ofrezcan la mejor manera de visualizar los datos deseados.
- ❖ Los cuadros de mando también pueden ser engañosos si no se utilizan correctamente. Es fácil seleccionar los datos (sesgo de los datos) que apoyan un argumento concreto e ignorar los que no lo hacen. Esto también se llama sesgo de confirmación. Por ello, los cuadros de mando deben utilizarse con cuidado y sólo como parte de un enfoque analítico más amplio. [CITATION 21en \l 3082]

Tipos de dashboard:

Existen diferentes tipos de dashboards, donde cada uno de ellos cumple la función de atender determinada escala del negocio con sus atribuciones y lo que se debe acompañar. A continuación, se describe cada uno de los tipos existentes:[CITATION Pir21 \l 3082]

Operacionales:

Capítulo 1: Fundamentación teórica

- ❖ Se utiliza para mostrar las métricas que necesitan ser monitoreadas por los analistas para que puedan hacer optimizaciones en sus actividades y ganar más agilidad en la corrección de rutas de negocio.
- ❖ Sin cruzar datos de diferentes áreas, el analista tendría que realizar diferentes operaciones, dedicando más tiempo a sus actividades, lo que compromete su productividad.

Financiero:

- ❖ Son aquellos que incluyen información importante sobre el sector financiero de la empresa, lo que permite al gerente y su equipo evaluar cómo se encuentra la salud económica del negocio.

Táctico:

- ❖ Es un panel dirigido al equipo de gestión. La información evidenciada por este recurso ayuda a la toma de decisiones, con el objetivo de alcanzar los objetivos en el mediano plazo. Los datos disponibles en este panel deben estar controlados exclusivamente por profesionales que ocupen cargos directivos.

Estratégico:

- ❖ Es el dashboard que usa la directiva, ya que ayuda a tomar decisiones y establecer metas a largo plazo. Sin embargo, su información debe estar disponible para todos los empleados, ya que facilita el compromiso del equipo.
- ❖ Los datos financieros, los resultados de las metas establecidas, el desempeño de las áreas y la empresa en su conjunto se muestran en este modelo de dashboard.

Capítulo 1: Fundamentación teórica

1.1.4. Sistemas de gestión

En Cuba el sistema de mayor implementación inicial fue el Sistema de Gestión de la Calidad, debido a que legitima las capacidades de las empresas para satisfacer los estándares requeridos de comercialización y el cumplimiento de los marcos legales y reglamentarios aplicables. En la actualidad esos argumentos no son suficientes, porque las circunstancias son otras: el compromiso social se ha incrementado, se han consensuado metas ambientales a diferentes escalas y se controlan los indicadores económicos, lo cual induce la integración de diferentes sistemas de gestión.[CITATION 21ju \l 3082]

1.1.5. Sistemas en tiempo real

Un sistema de tiempo real puede definirse como un sistema que debe satisfacer restricciones explícitas en el tiempo de respuesta o arriesgarse a severas consecuencias, incluida la falla. Por lo tanto, un sistema de tiempo real es un sistema que responde a un estímulo externo dentro de un tiempo especificado. Su eficiencia no solo depende de la exactitud de los resultados de cómputo, sino también del momento en que los entrega. La predictibilidad es su característica principal. A diferencia de los sistemas tradicionales, que tienden a distribuir en forma equitativa los recursos disponibles entre las diferentes tareas a ejecutar, los sistemas de tiempo real deben asegurar la distribución de recursos de tal forma que se cumplan los requerimientos de tiempo.[CITATION Bla17 \l 3082]

Los sistemas de tiempo real pueden dividirse en dos tipos diferentes, en función de su severidad en el tratamiento de los errores que puedan presentarse:[CITATION Bla17 \l 3082]

- ❖ Sistemas de tiempo real blandos (Soft): pueden tolerar un exceso en el tiempo de respuesta, con una penalización por el incumplimiento del plazo.

Capítulo 1: Fundamentación teórica

- ❖ Sistemas de tiempo real duros (Hard): la respuesta fuera de término no tiene valor alguno, y produce la falla del sistema.

1.2 Sistemas similares estudiados

En la actualidad hay varias empresas que hacen uso de los Dashboard con similares herramientas, las cuales nos permiten tener información actualizada y necesaria para lograr uno o varios objetivos y que la información sea fácil de interpretar.

1.2.1 Score

En la actualidad hay varias empresas que hacen uso de los Dashboard con similares herramientas, las cuales nos permiten tener información actualizada y necesaria para lograr uno o varios objetivos y que la información sea fácil de interpretar. [CITATION Dre20 \l 3082]

Características: [CITATION Dre20 \l 3082]

- ❖ Permite gestionar proyectos.
- ❖ Administrar cada detalle del proyecto, incluido el tiempo dedicado y facturado, tareas planificadas, reuniones, facturas, gastos, comentarios y archivos, todo en una página.
- ❖ Segrega y asigna trabajos facturables y no facturables.
- ❖ Permite crear presupuestos de proyectos y compararlos con los reales.

1.2.2 Sisense

Sisense no es una herramienta de análisis tradicional. Está diseñada para manejar todo tipo de datos, desde pequeños conjuntos de datos hasta big data. También es escalable: permite a las empresas acumular más datos a medida que avanzan sin preocuparse de que los servidores se colapsen o la memoria del ordenador sea lenta. Su tecnología en chip utiliza el procesamiento en paralelo y una base de datos en columna en memoria, por lo que puede procesar terabytes de datos en un solo servidor y seguir dando servicio a múltiples usuarios.

Capítulo 1: Fundamentación teórica

Por tanto, las empresas no necesitarán gastar recursos en hardware o servidores adicionales..[CITATION Com21 \l 3082]

Características:[CITATION Fol22 \l 3082]

- ❖ Sisense proporciona una variedad de diferentes características de BI, como la capacidad de conectarse con cualquier fuente de datos, herramientas de bajo código o sin código, analítica aumentada y funcionalidad de autoservicio.
- ❖ Queremos destacar especialmente que también está diseñado para el acceso desde cualquier dispositivo móvil. Los usuarios pueden analizar los datos, acceder a los cuadros de mando, editarlos o incluso ejecutar consultas mientras están en movimiento.
- ❖ Sisense ofrece una funcionalidad de análisis integrada que permite a las organizaciones ofrecer inteligencia empresarial a sus clientes. Puede integrarse en cualquier software o aplicación, desplegarse en la nube o en las instalaciones y con una arquitectura de inquilino único o multiinquilino.

1.2.3 Tableau.

Tableau es una herramienta de análisis de datos y de inteligencia de negocios. Esta herramienta de análisis ayuda a visualizar los datos y contribuye a la resolución de problemas a través de la manera en que usamos estos datos.[CITATION SPn21 \l 3082]

3

Algunas de las características de esta plataforma son:[CITATION SPn21 \l 3082]:

- ❖ Sacar el máximo provecho a tus datos con cálculos, estadísticas y análisis avanzados.

3 **CRM:** La gestión de relaciones con los clientes (CRM se utiliza para aumentar las ventas y mejorar la eficiencia empresarial. Con la ayuda de los sistemas CRM, puede recopilar y administrar datos de clientes, hacer planes de ventas, mantener contacto con sus clientes y crear facturas.

Capítulo 1: Fundamentación teórica

- ❖ Encontrar fácilmente los datos, señalar datos relevantes, habilitar búsquedas u organizar proyectos.
- ❖ Los datos son seguros y confiables.
- ❖ La información se puede ver con claridad debido al modelado de datos automático.
- ❖ Es posible acceder a los datos en instalaciones físicas o en la nube.
- ❖ Posibilita acceder a diferentes datos y combinarlos sin escribir código.
- ❖ Conectar a los datos en tiempo real, hacer extracciones, o ambos, de acuerdo a tus necesidades.
- ❖ Ayuda a descubrir y compartir información de manera más rápida con el propósito de generar cambios.

1.2.4 Power Bi.

Power BI es una herramienta para la extracción y presentación de la información que nos asegura una buena manipulación y visualización de los datos para mejorar el entendimiento de los mismos.[CITATION XMS \l 3082]

Características de Power BI: [CITATION XMS \l 3082]

- ❖ El registro es gratuito.
- ❖ Rápido y fácil acceso a los datos con exposición de métricas claves.
- ❖ Los paneles de datos pueden ser revisados en tiempo real.

Capítulo 1: Fundamentación teórica

- ❖ Power Bi es capaz de responder preguntas como “Ventas Chile 2019”, con una visualización interactiva con gráficos e informes.
- ❖ Es totalmente personalizable para las necesidades de su empresa y equipo de trabajo.
- ❖ Capacidad de tomar decisiones basadas en los datos desde cualquier sitio del mundo, disponible para aplicaciones Windows, iOS y Android.

1.2.5 Acunetix.

Es una herramienta para la auditoria de aplicaciones web de forma automática. En la versión 4 de la herramienta se añadieron módulos para la detención de vulnerabilidades Blind SQL Injection y Bind XPath Injection.[CITATION Can18 \l 3082]

Ofrece su producto como complemento a las auditorías internas de seguridad que se realicen sobre el código. Realiza escaneos automatizados con posibilidad de ejecución tanto en interfaz gráfica como en línea de comandos. Solo está disponible para plataformas Windows. [CITATION Can18 \l 3082]

Características[CITATION Can18 \l 3082]:

- ❖ Analizador JavaScript, permitiendo auditar Ajax y aplicaciones con Web 2.0.
- ❖ Utiliza las más técnicas de SQL Injection y Cross Site Scripting.
- ❖ Utilización de tecnología “AcuSensor”.
- ❖ Analiza websites incluyendo contenido Flash, SOAP, AJAX.
- ❖ Realiza un escaneo de puertos contra el servidor web y busca vulnerabilidades en sus servicios.
- ❖ Detecta el lenguaje de programación de la aplicación.
- ❖ Proporciona extensos informes del estado de seguridad.

Capítulo 1: Fundamentación teórica

El estudio realizado sobre los sistemas homólogos permitió identificar cuáles son los componentes más usados en el desarrollo de dashboard para aplicaciones web. Dentro de estos componentes, los más utilizados son:

- ❖ Las cajas de información.
- ❖ Gráfica de barra invertida.
- ❖ Los gráficos de pastel o circulares.
- ❖ Grafica de líneas.
- ❖ Los gráficos de barra.
- ❖ Los gráficos de puntos.
- ❖ Diagramas de dispersión.
- ❖ Gráfica de embudo.
- ❖ Mapas.
- ❖ Gráfica de bala.
- ❖ Gráfica de tabla.
- ❖ Gráfica de gauge o medidor radial.
- ❖ Gráfica de burbuja.
- ❖ Mapa de calor.
- ❖ Gráfica de radar o araña.
- ❖ Diagrama de caja o bigote.
- ❖ Gráfico de cascada.
- ❖ Gráfica de área, entre otros.

Teniendo en cuenta los componentes encontrados y la información que es necesaria mostrar en el dashboard de SIGIES se deciden utilizar: las cajas de información, los gráficos de pastel y los gráficos de barra.

Capítulo 1: Fundamentación teórica

1.3 Análisis de las posibles tecnologías, técnicas, herramientas y lenguajes a utilizar.

1.3.1 Metodología de desarrollo de software.

Las metodologías de desarrollo de software son un conjunto de técnicas y métodos organizativos que se aplican para diseñar soluciones de software informático. El objetivo de las distintas metodologías es el de intentar organizar los equipos de trabajo para que estos desarrollen las funciones de un programa de la mejor manera posible. [CITATION San20 \l 3082]

AUP es una versión simplificada del Proceso Racional Unificado (RUP, por sus iniciales en inglés). Para guiar el desarrollo de la solución se aplicó la metodología AUP (Proceso Ágil Unificado) en su versión UCI. Esta versión cuenta con tres fases:

Tabla 1 Fases de la variación de AUP para la UCI.

Fases AUP	Fases Variación AUP-UCI	Objetivos de las fases (Variación AUP-UCI)
Inicio	Inicio	<ul style="list-style-type: none"> ❖ Planeación del proyecto. ❖ En esta fase se realizan un estudio inicial de la organización cliente. ❖ Alcance del proyecto. ❖ Realizar estimaciones de tiempo, esfuerzo y costo. ❖ Decidir si se ejecuta o no el proyecto.
Elaboración	Ejecución	<ul style="list-style-type: none"> ❖ Ejecutan las actividades requeridas para desarrollar el software.
Construcción		<ul style="list-style-type: none"> ❖ Ajuste de los planes del proyecto considerando los requisitos y la arquitectura.
Transición		<ul style="list-style-type: none"> ❖ Modelado del negocio. ❖ Obtención de requisitos. ❖ Elaboración de la arquitectura y el diseño. ❖ Implementan y se libera el producto.

Capítulo 1: Fundamentación teórica

Cierre	Cierre	<ul style="list-style-type: none">❖ Análisis de los resultados del proyecto como la ejecución.❖ Cierre del proyecto.
--------	--------	---

1.3.2 Marco de trabajo.

Del inglés framework. Entorno de trabajo o marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar. Varios autores y experto en desarrollo agile, indican que las propuestas que utilizan no se pueden denominar metodologías sino frameworks.[CITATION Gan17 \l 3082]

Bootstrap 3.0

Bootstrap es un framework CSS desarrollado por Twitter en 2010, para estandarizar las herramientas de la compañía. Bootstrap es un framework CSS utilizado en aplicaciones front-end, es decir, en la pantalla de interfaz con el usuario, para desarrollar aplicaciones que se adaptan a cualquier dispositivo.[CITATION Gue20 \l 1033]El framework Bootstrap combina los lenguajes de programación CSS y JavaScript para estilizar los elementos de una página HTML. Por eso, ofrece muchas más funcionalidades que, simplemente, cambiar la forma y el color de botones y enlaces. Bootstrap destaca por las siguientes características:[CITATION Jes22 \l 3082]

- ❖ Desarrollo de sitios web *Responsive*.
- ❖ Permite construir sitios web especialmente adaptada para dispositivos móviles, Smartphone, Tablets y desktop.
- ❖ La cantidad de componentes que ofrece para mejorar la comunicación con el usuario.
- ❖ Barra de navegación que permite múltiples configuraciones.

Symfony 2.17

Capítulo 1: Fundamentación teórica

“Es un framework PHP de tipo full stack construido con varios componentes independientes creados por el propio proyecto Symfony”, es decir, por sí mismo. A su vez, está planteado para trabajar lo que a nivel de desarrollo se conoce como backend. Entre las características más ventajosas de trabajar con la marca y el framework Symfony destacamos que:

- ❖ tiene un código de software libre,
- ❖ ofrece documentación libre, con libros y tutoriales.
- ❖ permite a los desarrolladores acceder a una gran variedad de proyectos para crear aplicaciones web complejas, sitios webs básicos y llevar a cabo su mantenimiento.
- ❖ los componentes que emplean han sido ya testados y se utilizan en proyectos inmensos[CITATION Stu21 \l 3082].

1.3.3 Selección el framework para la manipulación de PHP

Symfony es un entorno de trabajo estandarizado (framework PHP) que se utiliza para el desarrollo de aplicaciones web y es de los más utilizados en el entorno de desarrolladores de apps. En otras palabras, es una herramienta para desarrolladores para crear aplicaciones en PHP.[CITATION Stu21 \l 3082] Se escogió la versión 2.17 porque es en la que está desarrollado el SIGIES.

1.3.4 Lenguajes de programación.

Se escogió el framework Symfony para el desarrollo de la aplicación web haciendo necesario el modelado para su análisis y diseño, y, una selección de los lenguajes de programación PHP 5, HTML 5 (Lenguaje de Marcado de Hipertexto), CSS 3 (Hojas de estilo en cascada), JavaScript (Lenguaje interpretado), tales lenguajes permitirán el desarrollo del diseño web.

HTML5 es un Lenguaje de Marcado de Hipertexto (HyperText Markup Language) que funciona con las bases del HTML de siempre, pero que cuenta con etiquetas adicionales que

Capítulo 1: Fundamentación teórica

permiten crear sitios webs más modernos y más compatibles con las tecnologías digitales actuales.[CITATION etq39 \l 3082]

PHP5 son las siglas en inglés del acrónimo Hypertext Pre-Processor, es decir, pre-procesador de hipertexto. Es un lenguaje de programación de propósito general que se ejecuta en el lado del servidor. Es un lenguaje interpretado. Tiene múltiples formas de utilizarse, ya que puede utilizarse con scripts, de forma estructurada o programación en objetos. Está creado con la licencia de software libre PHPv3_01, que es una licencia Open Source.[CITATION Sol19 \l 3082]

CSS 3 CSS (siglas en inglés de Cascading Style Sheets), en español "Hojas de estilo en cascada", un lenguaje de marcas enfocado a definir, crear y mejorar la presentación de un documento basado en HTML. Para muchos diseñadores gráficos CSS significó la puerta de entrada al mundo de la web y junto con otras tecnologías como JavaScript, CSS se ha ido imponiendo como uno de los pilares imprescindibles de la web de hoy en día. Durante su evolución el lenguaje de etiquetas HTML ha ido permitiendo cierto grado de personalización. Desde poner en negrita un texto, cambiar su color o alinear una fotografía se iba haciendo casi de manera artesanal añadiendo línea a línea comentarios que ayudaban a su personalización.[CITATION Rob19 \l 3082]

JavaScript JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. [CITATION Etq37 \l 3082]

jQuery 1.8 se trata de una biblioteca de JavaScript que se puede agregar a proyectos de codificación. Básicamente, esta herramienta permite a los desarrolladores web conectar

Capítulo 1: Fundamentación teórica

recursos JavaScript rutinarios a una página web, y así ellos puedan dedicar más tiempo a centrarse en funciones más complejas que son exclusivas del sitio web. JQuery es de código abierto, lo que significa que cualquier persona puede contribuir o modificarlo, conquistando una gran comunidad de usuarios, además es un solo archivo JS que contiene todos los componentes DOM⁴, eventos, efectos y AJAX⁵. [CITATION Hos21 \l 3082]

1.3.5 Lenguaje de modelado.

UML el Lenguaje de Modelado Unificado UML es un lenguaje de modelados estandarizado que consta de un conjunto integrado de diagramas, desarrollados para ayudar a los desarrolladores de sistemas y software a especificar, visualizar, construir y documentar los artefactos de los sistemas de software, así como para el modelado de negocio y otros sistemas que no son de software. [CITATION Rut22 \l 3082]

1.3.6 Sistemas Gestores de Base de datos.

Un sistema de gestión de bases de datos (SGBD) es un software utilizado para gestionar, almacenar y recuperar bases de datos. Proporciona una interfaz que permite a los usuarios leer, crear, borrar y actualizar datos. [CITATION Bor19 \l 3082]

PostgreSQL: Es un sistema de gestión de bases de datos objeto -relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarles a otras bases de datos comerciales, utiliza un modelo cliente/servidor para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. [CITATION Bac17 \l 3082]

⁴**DOM:** El modelo de objeto de documento es una interfaz de programación para los documentos HTML y XML.

⁵ **AJAX:** JavaScript Asíncrono + XML (AJAX), es una técnica de desarrollo web para crear aplicaciones web asíncronas.

Capítulo 1: Fundamentación teórica

Es un servidor de bases de datos relacional libre. Sus principales características son: llaves ajenas o llaves foráneas, disparadores, vistas, integridad transaccional, acceso concurrente multiversión (no se bloquean las tablas, ni siquiera las filas, cuando un proceso escribe), capacidad de albergar programas en el servidor en varios lenguajes. Herencia de tablas, tipos de datos y operaciones geométricas. [CITATION Bac17 \l 3082]

A continuación, se muestran algunas de sus ventajas: [CITATION Bac17 \l 3082]

- ❖ Es altamente escalable en cuanto a cantidad de datos y usuarios.
- ❖ Es multiplataforma.
- ❖ Ofrece una documentación bien organizada, pública y libre.
- ❖ Presenta conectores de datos foráneos que permiten añadir y consultar fuentes de datos externas desde PostgreSQL.

La versión a utilizar será la 9.4.5.

PgAdmin III: Constituye la más completa y popular herramienta de código abierto para la administración de base de datos PostgreSQL. Hace más sencilla la administración y está diseñado para satisfacer las necesidades de todos los usuarios, teniendo en cuenta la realización de consultas SQL desde la más sencilla, hasta el desarrollo de BD de alta complejidad. [CITATION sma11 \l 3082]

1.3.7 Servidor web

Un servidor web (server) es un ordenador de gran potencia que se encarga de “prestar el servicio” de transmitir la información pedida por sus clientes (otros ordenadores, dispositivos móviles, impresoras, personas, etc.) [CITATION Iva19 \l 1033]

Los servidores web (web server) son un componente de los servidores que tienen como principal función almacenar, en web hosting, todos los archivos propios de una página web (imágenes, textos, videos, etc.) y transmitirlos a los usuarios a través de los navegadores mediante el protocolo HTTP (Hipertext Transfer Protocol).[CITATION Iva19 \l 1033]

Capítulo 1: Fundamentación teórica

Luego de haber seleccionado a PHP como lenguaje de programación, y a Symfony como framework de desarrollo, se decide utilizar el servidor Web Nginx para permitir el tratamiento de las peticiones de los clientes. Además, el mismo está condicionado por el producto SIGIES.

Nginx: es un servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico. Es software libre y de código abierto, licenciado bajo la Licencia BSD⁶ simplificada. La principal ventaja de Nginx como servidor web es que consume muchos menos recursos al servir contenido estático, y esto convierte a Nginx en una excelente opción para funcionar como proxy inverso o como balanceador de carga para otros servidores como Apache, optimizando la entrega de contenidos. Lo que hace que Nginx sea diferente a otros servidores web es su arquitectura, que permite responder a millones de peticiones por segundo aprovechando al máximo los núcleos o hilos de ejecución del servidor con una configuración muy simple. [CITATION Ari19 \l 3082]

Otras características que ofrece el servidor Nginx son: [CITATION Ari19 \l 3082]

- ❖ Capaz de manejar más de 10.000 conexiones simultáneas con un uso bajo de memoria.
- ❖ Balanceo de carga, distribuye la carga entre los servidores que formen parte de la estructura, redirigiendo cada vez la petición hacia aquella máquina que tenga una menor carga.
- ❖ Alta tolerancia a fallos.
- ❖ Compatible con el nuevo estándar de direcciones IPv6.

La versión a utilizar será la 2.7

⁶ **BSD:** Berkely Software Distribution (Distribución de Software Berkely).

Capítulo 1: Fundamentación teórica

1.3.8 Herramientas del modelo.

Descripción de las herramientas del modelado a utilizar para cumplimiento con lo planteado en el objeto de estudio.

Visual Paradigm para UML v8.0 Visual Paradigm es una herramienta de gestión y diseño potente, multiplataforma y fácil de usar para sistemas de TI. ¡Visual Paradigm proporciona a los desarrolladores de software la plataforma de desarrollo de vanguardia para crear aplicaciones de calidad más rápido, mejor y más barato! Facilita una excelente interoperabilidad con otras herramientas CASE y la mayoría de los IDE líderes, lo que supera todo su proceso de desarrollo Model-Code-Implementación en esta solución integral. [CITATION Lib22 \l 3082]

Es una herramienta de código abierto.

1.4 Conclusiones Parciales

A partir del estudio realizado en el capítulo anterior, la investigación de las características, ventajas y desventajas de los dashboard, así como su utilización e importancia permitieron determinar un conjunto de tecnologías y herramientas para cumplir con el objeto de estudio planteado. También se escogió la metodología AUP-UCI

Luego de finalizar el capítulo se llegó a la conclusión de que:

- ❖ Es necesario implementar un Dashboard para SIGIES que permita gestionar la información del proceso de Ingreso a la Educación Superior utilizando KPI's.
- ❖ Se desarrolló un breve resumen de los conceptos relacionado con Dashboard, Indicador Clave de Desempeño, la Inteligencia de Negocio, Sistemas de Gestión, Sistemas en Tiempo Real y Sistemas Similares Estudiado.

Capítulo 1: Fundamentación teórica

- ❖ Se escogió la metodología de desarrollo de software AUP-UCI la cual se divide en tres fases Inicio, Ejecución y Cierre.
- ❖ Y por último se escogieron las tecnologías a utilizar para cumplir con lo planteado en el objetivo de estudio para la solución propuesta a un dashboard.

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

Capítulo 2: Análisis y diseño del Dashboard para SIGIES.

Introducción

En este capítulo se mostrará información de las herramientas propuestas. A continuación, se realizará la descripción de todos los procesos que componen el objeto de estudio mediante los cuales se elaboran, enumeran y describen los requisitos tanto funcionales como no funcionales que serán necesarios para realizar la propuesta de solución, y seguido de esto, su modelado resultante del análisis y diseño conteniendo descripción de la arquitectura. Luego realiza el diseño de los mecanismos para el almacenamiento, procesamiento y transmisión de los datos, y la presentación del diseño del tratamiento de los errores y el diseño del despliegue para darle solución a los objetivos propuestos.

2.1 Descripción de la propuesta de solución.

La propuesta de solución consiste en crear un módulo que permitirá la gestión de los componentes que conformarán el dashboard para lograr una mejor visualización de los datos en el proyecto SIGIES en tiempo real y facilitando la toma de decisiones. Dicha solución tendrá en cuenta el sistema de autenticación basado en roles (Administrador, Usuarios DIUL, Usuarios CIP) con el que cuenta actualmente SIGIES. El administrador es el encargado de asignarle los roles a los usuarios gestionados que interactúan con el sistema.

Por otra parte, el Usuario DIUL será el encargado de representar a los usuarios que provienen de la Dirección de Ingreso y Ubicación Laboral y que tienen acceso al SIGIES con algunos de estos roles: ROLES_AUDITOR_DIUL, ROLE_ESTADISTICO_DIUL, FUNCIONARIO_DIUL, ROLE_REPORS_UPDATE. El Usuario CIP será el encargado de representar a los usuarios que provienen de las Comisiones de Ingreso Provinciales y que tienen acceso al SIGIES con algunos de estos roles: ROLE_PRESIDENTE_CIP, ROLE_SECRETARIO_EJECUTIVO_CIP, ROLE_ESPECIALISTA_CIP, ROLE_AUDITOR_CIP, ROLE_ESTADISTICO_CIP, ROLE_SECRETARIO_DOCENTE_CIP,

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

ROLE_GESTOR_SOLICITUDES, ROLE_GESTOR_ESTUDIANTES y también el Usuario administrador será el encargado de representar al administrador del SIGIES con el rol: ROL.ADMIN_DIUL. Para ello debe mostrar el total de plazas iniciales, otorgadas, extras y sin cubrir en el plan de plazas de todo el país, el total de plazas otorgadas, extras y sin cubrir en el plan de plazas de una comisión y el total de solicitudes y de examinados por fuentes de ingreso del país y por fuente de ingreso de una comisión. Además, debe de ser capaz de mostrar el total de examinados por fuente de ingreso de una comisión y por exámenes de todo el país, el total de examinados que cumplen por comisión y los que no cumplen y el total de estudiantes que alcanzaron carreras por provincia y los que no alcanzaron. Por otra parte, el usuario puede configurar el dashboard para visualizar la cantidad de estudiantes que incrementaron su nota, los que la disminuyeron y los que se mantuvieron, la cantidad de estudiantes que cumplían que no cumplen y los que no cumplían que cumplen y el total de estudiantes que solicitaron recalificación por provincia. Toda la información antes mencionada se mostrará en la página de Inicio de SIGIES en la región derecha del menú lateral.

Dicho módulo se utilizará con el principio de permitirles a los responsables tener claridad de la situación actual, ya que los elementos seleccionados a monitorear son propios del proceso de ingreso a la Educación Superior.

2.2 Actores del sistema.

Tabla 2 Actores del sistema.

Usuario	Rol	Descripción
Usuarios de la DIUL	<ul style="list-style-type: none"> ROLE_AUDITOR_DIUL ROLE_ESTADISTICO_DIUL FUNCIONARIO_DIUL ROLE_REPORS_UPDATE	Representan a los usuarios que provienen de la Dirección de Ingreso y Ubicación

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

		Laboral y que tienen acceso al SIGIES.
Usuarios de la CIP	<ul style="list-style-type: none"> ✚ ROLE_PRESIDENTE_CIP ✚ ROLE_SECRETARIO_EJECUTIVO_CIP ✚ ROLE_ESPECIALISTA_CIP ✚ ROLE_AUDITOR_CIP ✚ ROLE_ESTADISTICO_CIP ✚ ROLE_SECRETARIO_DOCENTE_CIP ✚ ROLE_GESTOR_SOLICITUDES ✚ ROLE_GESTOR_ESTUDIANTES 	Representan a los usuarios que provienen de las Comisiones de Ingreso Provinciales y que tienen acceso al SIGIES.
Usuario administrador	✚ ROL.ADMIN_DIUL	Representa al administrador del SIGIES.

2.3 Requisitos del sistema

Los requisitos son, por definición "una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo" y "expresan las necesidades y limitaciones impuestas a un producto de software que contribuyen a la solución de algún problema en el mundo real". [CITATION Alo16 \l 3082]

2.3.1 Requisitos Funcionales

Los requerimientos funcionales son las descripciones explícitas que especifican lo que debe hacer el sistema. [CITATION Law18 \l 3082]

Tabla 3 Requisitos Funcionales del sistema.

No	Nombre	Descripción	Prioridad	Complejidad
RF1	Mostrar total de plazas iniciales en el plan de plazas de todo el país	El sistema debe permitir mostrar el total de plazas en el plan de plazas de todo el	Baja	Baja

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

		país		
RF2	Mostrar total de plazas otorgadas en el plan de plazas de todo el país.	El sistema debe permitir mostrar el total de plazas otorgadas en el plan de plazas de todo el país.	Baja	Baja
RF3	Mostrar total de plazas extras en el plan de plazas de todo el país.	El sistema debe permitir mostrar el total de plazas extras en el plan de plazas de todo el país.	Baja	Baja
RF4	Mostrar total de plazas sin cubrir en el plan de plazas de todo el país.	El sistema debe permitir mostrar el total de plazas sin cubrir en el plan de plazas de todo el país.	Baja	Baja
RF5	Mostrar total de plazas otorgadas en el plan de plazas de una comisión.	El sistema debe permitir mostrar el total de plazas otorgadas en el plan de plazas de una comisión.	Alta	Media
RF6	Mostrar total de plazas extras en el plan de plazas de una comisión.	El sistema debe permitir mostrar el total de plazas extras en el plan de plazas de una comisión.	Alta	Alta
RF7	Mostrar total de plazas sin cubrir en el plan de plazas de una comisión.	El sistema debe permitir mostrar el total de plazas sin cubrir en el plan de plazas de una comisión.	Alta	Media
RF8	Mostrar total de solicitudes por fuente de ingreso de una comisión.	El sistema debe permitir mostrar el total de solicitudes por fuentes de ingreso de una comisión.	Alta	Media
RF9	Mostrar total de	El sistema debe permitir	Alta	Media

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

	solicitudes por fuente de ingreso del país.	mostrar el total de solicitudes por fuentes de ingreso del país.		
RF10	Mostrar total de examinados por fuentes de ingreso del país.	El sistema debe permitir mostrar el total de examinados por fuentes de ingreso del país.	Alta	Media
RF11	Mostrar total de examinados por fuentes de ingreso de una comisión.	El sistema debe permitir mostrar el total de examinados por fuentes de ingreso de una comisión.	Alta	Media
RF12	Mostrar total de examinados por exámenes de todo el país.	El sistema debe permitir mostrar el total de examinados por exámenes de todo el país.	Alta	Media
RF13	Mostrar total de examinados por exámenes de una comisión.	El sistema debe permitir mostrar el total de examinados por exámenes de una comisión.	Alta	Media
RF14	Mostrar total de examinados que cumplen por comisión.	El sistema debe permitir mostrar el total de examinados que cumplen por provincia.	Alta	Media
RF15	Mostrar total de examinados que no cumplen por comisión.	El sistema debe permitir mostrar el total de examinados que no cumplen por provincia.	Alta	Media
RF16	Mostrar total de estudiantes que	El sistema debe permitir mostrar el total de estudiantes	Alta	Media

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

	alcanzaron carreras por provincia.	que alcanzaron carreras por provincia.		
RF17	Mostrar total de estudiantes que no alcanzaron carreras por provincia.	El sistema debe permitir mostrar el total de estudiantes que no alcanzaron carreras por provincia.	Alta	Media
RF18	Mostrar la cantidad de estudiantes que incrementaron su nota.	El sistema debe permitir mostrar la cantidad de estudiantes que incrementaron su nota.	Alta	Baja
RF19	Mostrar la cantidad de estudiantes que disminuyeron su nota.	El sistema debe permitir mostrar la cantidad de estudiantes que disminuyeron su nota.	Alta	Baja
RF20	Mostrar la cantidad de estudiantes que mantienen su nota.	El sistema debe permitir mostrar la cantidad de estudiantes que mantienen su nota.	Alta	Baja
RF21	Mostrar la cantidad de estudiantes que cumplían que no cumplen.	El sistema debe permitir mostrar la cantidad de estudiantes que cumplían que no cumplen.	Alta	Baja
RF22	Mostrar la cantidad de estudiantes que no cumplían que cumplen.	El sistema debe permitir mostrar la cantidad de estudiantes que no cumplían que cumplen.	Alta	Baja
RF23	Mostrar total de estudiantes que solicitaron recalificación	El sistema debe permitir mostrar la cantidad de estudiantes que solicitaron	Baja	Baja

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

	por provincia.	recalificación por provincia.		
RF24	Mostrar la cantidad de estudiantes con probable sexo no correspondiente en una provincia.	El sistema debe permitir mostrar la cantidad de estudiantes con probable sexo no correspondiente en una provincia de un procesamiento en una convocatoria.	Baja	Baja
RF25	Listar componentes del dashboard.	El sistema debe permitir listar los tipos de componentes y sus títulos para que el usuario seleccione el que desea ver.	Baja	Baja
RF26	Mostrar datos del componente del dashboard.	El sistema debe permitir mostrar al usuario los datos de cada componente.	Baja	Baja
RF27	Editar componentes del dashboard.	El sistema debe permitir seleccionar los usuarios y las comisiones que van a visualizar los componentes.	Baja	Media
RF28	Exportar componente del dashboard.	El sistema debe permitir al usuario exportar los datos de los componentes.	Baja	Media

2.3.2 Requisitos no funcionales.

Requerimientos de calidad, que representan restricciones o las cualidades que el sistema debe tener tales como: precisión, usabilidad, seguridad, rendimiento, confiabilidad, performance entre otras. Estos requisitos poseen una naturaleza abstracta e intangible en comparación con los RF y esto hace que sean más difíciles de especificar o documentar formalmente.[CITATION Req31 \l 3082]

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

Tabla 4 Requisitos no funcionales.

No. RnF	Tipo de RnF	Descripción
RnF. 1	Usabilidad.	<p>1.1 El módulo a desarrollar debe mostrar los textos que aparezcan en la interfaz del sistema en idioma español.</p> <p>1.2 El módulo puede ser utilizado por personas que tengan un conocimiento básico o con experiencia en la utilización de las computadoras.</p>
RnF. 2	Software.	<p>2.1 El módulo debe de correr en sistema operativo Windows y sistemas operativos Linux.</p> <p>2.2 El módulo debe de integrarse y consumir información del SIGIES.</p>
RnF. 3	Seguridad.	<p>3.1 Garantizar el acceso a las funcionalidades definidas para los usuarios de acuerdo a los roles que posean.</p>

2.4 Historia de Usuario.

Una historia de usuario es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final. Su propósito es articular cómo proporcionará una función de software valor al cliente. Las historias utilizan un lenguaje no técnico para ofrecer contexto al equipo de desarrollo y sus esfuerzos. Después de leer una historia de usuario, el equipo sabe por qué está compilando lo que está compilando y qué valor crea. Además, son uno de los componentes centrales de un programa ágil. Ayudan a proporcionar un marco centrado en el usuario para el trabajo diario, lo que impulsa la colaboración y la creatividad y mejora el producto en general. [CITATION His \ 3082]

Tabla 5 Historia de Usuario “Mostrar total de estudiantes que alcanzaron carreras por provincia.”

Historia de Usuario	
Número 1	Nombre de la Historia de Usuario: Mostrar total de estudiantes que alcanzaron carreras por provincia.
Cantidad de modificaciones a la Historia de Usuario: Ninguna.	
Usuario: Administrador, Usuarios DIUL, Usuarios CIP	Iteración asignada: 1

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

Prioridad en negocio: Alta

Riesgo en desarrollo: Media

Descripción: El sistema debe permitir mostrar el total de estudiantes que alcanzaron carreras por provincia. Para ello el actor involucrado debe llenar el siguiente formulario que le facilita el sistema. Dicho formulario cuenta con los siguientes campos: **“Nombre”**, **“Provincia”**, **“Carrera”** y **“Datos”**. Donde el campo estudiantes se refiere a los nombres de los estudiantes, este campo solo admite letras (string), el campo provincia se refiere a la provincia de cada estudiante, este campo solo admite letras (string), el campo carrera es para señalar q esos estudiantes tienen carrera y el campo datos donde se van a mostrar el total de estudiantes que alcanzaron carreras por provincia. Una vez llenados todos los campos el actor debe hacer clic en el botón **“Mostrar”** y automáticamente el sistema muestra el total de plazas otorgadas en el campo datos.

Prototipo:

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

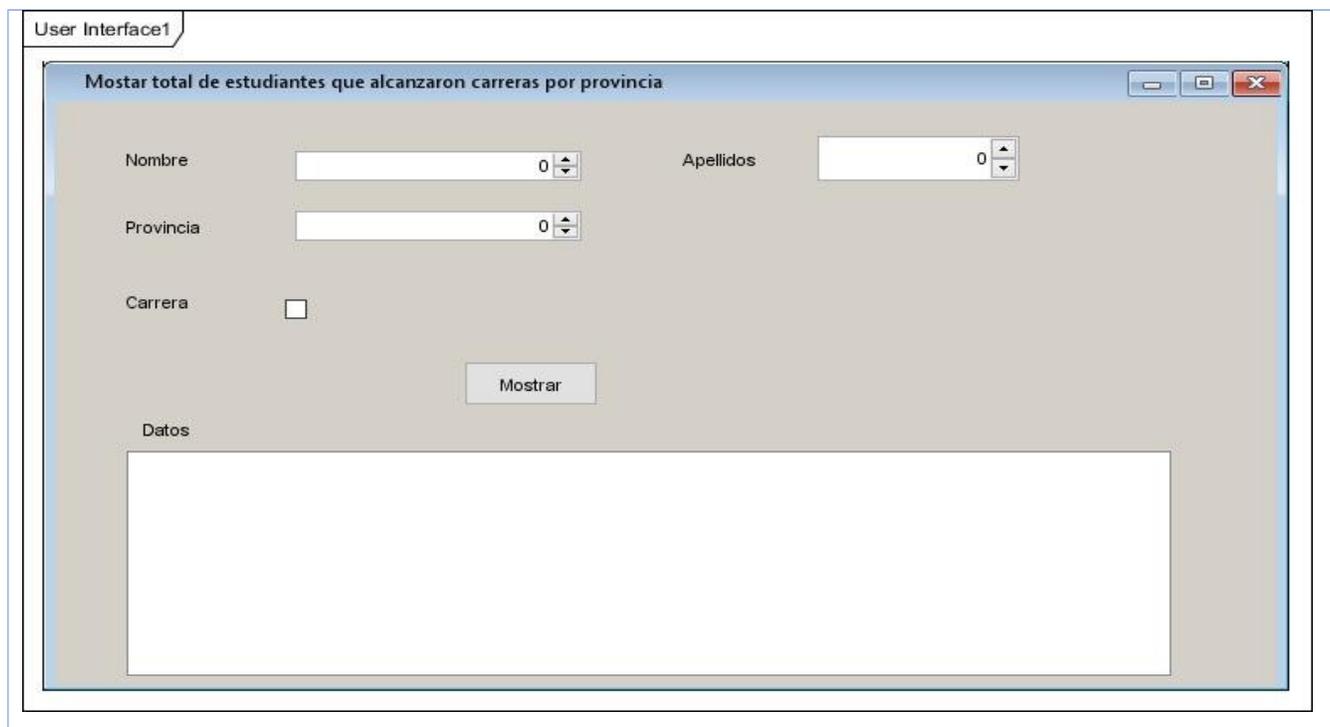


Tabla 6 Historia de Usuario “Mostrar total de plazas otorgadas en el plan de plazas de una comisión.”

Historia de Usuario	
Número 2	Nombre de la Historia de Usuario: Mostrar total de plazas otorgadas en el plan de plazas de una comisión.
Cantidad de modificaciones a la Historia de Usuario: Ninguna.	
Usuario: Administrador, Usuarios DIUL, Usuarios CIP	Iteración asignada: 1
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
<p>Descripción: El sistema debe permitir mostrar el total de plazas otorgadas en el plan de plazas de una comisión. Para ello el actor involucrado debe llenar el siguiente formulario que le facilita el sistema. Dicho formulario cuenta con los siguientes campos: “Código “, “Plazas”, “Cant. Plazas” y “Datos”. Donde el campo código se refiere al orden en que van a ir ubicadas las plazas que se otorgarán, este campo solo admite números (int), el campo</p>	

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

plazas se refiere a las plazas que se van a otorgar, este campo solo admite letras (*string*), el campo cant. Plazas se refiere a la cantidad por cada plaza q se otorgarán y el campo datos donde se van a mostrar las plazas. Luego de llenar todos los campos el actor debe hacer clic en el botón “**Mostrar**” y automáticamente el sistema muestra el total de plazas otorgadas en el campo datos.

Prototipo:

Tabla 7 Historia de Usuario “Mostrar datos del componente dashboard”.

Historia de Usuario	
Número 3	Nombre de la Historia de Usuario: Mostrar datos del componente dashboard
Cantidad de modificaciones a la Historia de Usuario: Ninguna.	
Usuario: Administrador, Usuarios DIUL,	Iteración asignada: 1

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

Usuarios CIP	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Descripción: El sistema debe permitir a los usuarios mostrar los datos de cada componente.	

Prototipo:

User Interface3

Mostrar datos del componente dashboard

Componente

Descripción

Mostrar

Datos

Tabla 8 Historia de Usuario “Listar componentes del dashboard.”

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

Historia de Usuario	
Número 4	Nombre de la Historia de Usuario: Listar componentes del dashboard.
Cantidad de modificaciones a la Historia de Usuario: Ninguna.	
Usuario: Administrador, Usuarios DIUL, Usuarios CIP	Iteración asignada: 1
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
<p>Descripción: El sistema debe permitir listar los tipos de componentes y sus títulos para que el usuario seleccione el que desea ver. Para ello el actor involucrado debe llenar el siguiente formulario que le facilita el sistema. Dicho formulario cuenta con los siguientes campos: “Tipos “, “Título” y “Datos”. Donde el campo tipos se refiere a los tipos de componentes del sistema, y el campo título se refiere a los títulos de los componentes, el campo datos es para cuando los usuarios seleccionen los tipos y títulos de los componentes que desee ver y los muestre en una liste. Luego de llenar todos los campos el actor debe hacer clic en el botón “Listar” y automáticamente el sistema muestra el listado de componentes del dashboard.</p>	
Prototipo:	

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

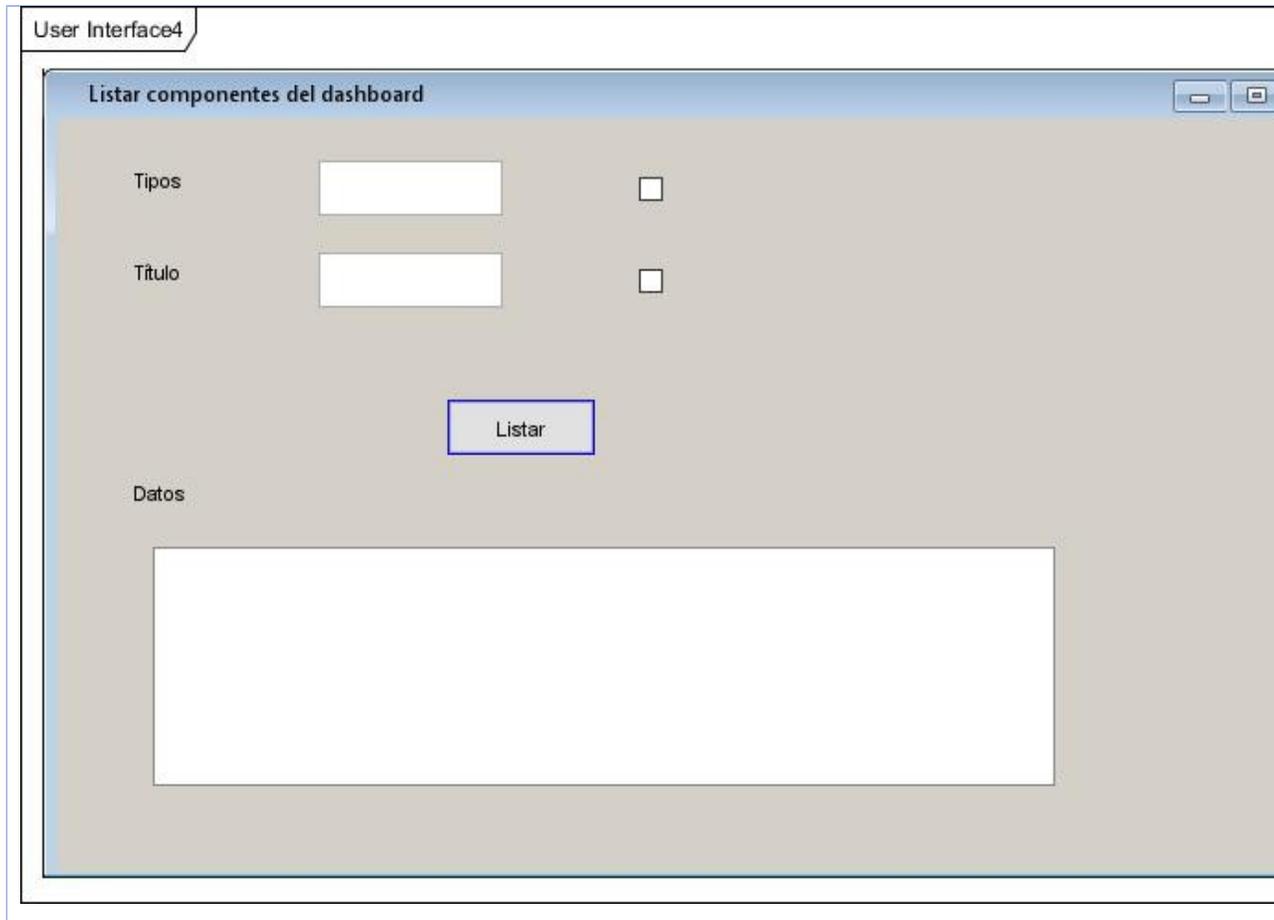


Tabla 9 Historia de Usuario “Exportar componente del dashboard.”

Historia de Usuario	
Número 5	Nombre de la Historia de Usuario: Exportar componente del dashboard.
Cantidad de modificaciones a la Historia de Usuario: Ninguna.	
Usuario: Administrador, Usuarios DIUL, Usuarios CIP	Iteración asignada: 1
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

Descripción: El sistema debe permitir al usuario exportar los datos de los componentes.

Prototipo:

2.5 Arquitectura del sistema.

Un patrón arquitectónico es un paquete de decisiones de diseño que se encuentra repetidamente en la práctica, tiene propiedades bien definidas que pueden ser reutilizadas y describe una clase de arquitecturas. [CITATION Los33 \l 3082]

2.5.1 Patrón arquitectónico Modelo Vista Controlador.

El patrón MVC es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo. A partir del uso de frameworks basados en el patrón MVC se puede lograr una mejor organización del trabajo y mayor especialización de los desarrolladores y diseñadores. [CITATION Etq34 \l 3082]

- ❖ Modelo (datos): Es el objeto que representa los datos del programa. Maneja los datos del y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre Modelos y sus Vistas, y notificar a las Vistas cuando cambiar el Modelo. [CITATION Etq34 \l 3082]
- ❖ Vistas (UI): El trabajo de la vista es decidir qué verá el usuario en su pantalla y cómo. [CITATION Etq34 \l 3082]
- ❖ Controlador (Cerebro): La responsabilidad del controlador es extraer, modificar y proporcionar datos al usuario. Esencialmente, el controlador es el enlace entre la Vista y el Modelo. [CITATION Etq34 \l 3082]

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

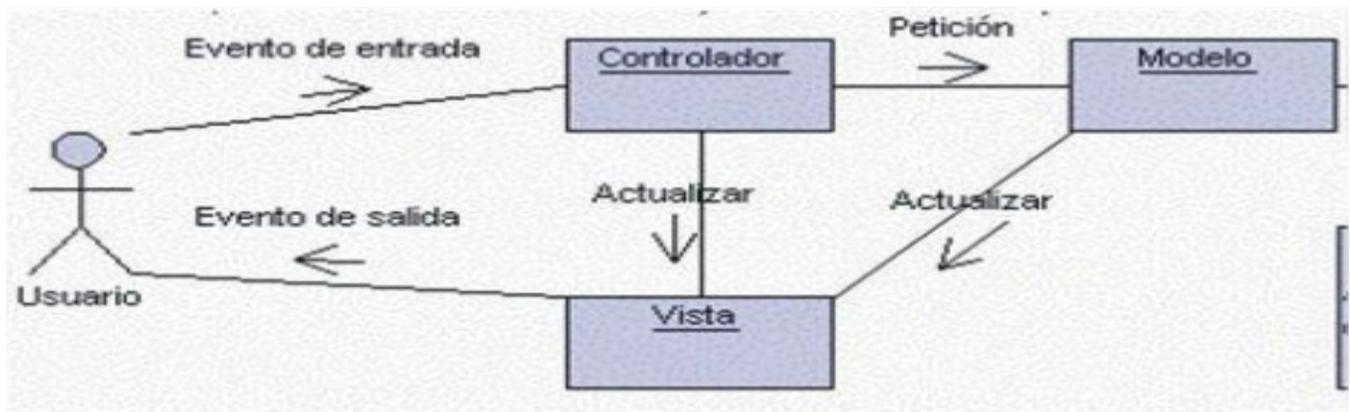


Figura 1: Patrón Arquitectónico Modelo Vista Controlador. [CITATION Etq34 \l 3082]

Paso a paso cómo sería el flujo de trabajo característico en un esquema MVC.

- ❖ El usuario realiza una solicitud a nuestro sitio web. Generalmente estará desencadenada por acceder a una página de nuestro sitio. Esa solicitud le llega al controlador.[CITATION Etq35 \l 3082]
- ❖ El controlador comunica tanto con modelos como con vistas. A los modelos les solicita datos o les manda realizar actualizaciones de los datos. A las vistas les solicita la salida correspondiente, una vez se hayan realizado las operaciones pertinentes según la lógica del negocio.[CITATION Etq35 \l 3082]
- ❖ Para producir la salida, en ocasiones las vistas pueden solicitar más información a los modelos. En ocasiones, el controlador será el responsable de solicitar todos los datos a los modelos y de enviarlos a las vistas, haciendo de puente entre unos y otros. Sería corriente tanto una cosa como la otra, todo depende de nuestra implementación; por eso esa flecha la hemos coloreado de otro color.[CITATION Etq35 \l 3082]

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

- ❖ Las vistas envían al usuario la salida. Aunque en ocasiones esa salida puede ir de vuelta al controlador y sería éste el que hace el envío al cliente, por eso he puesto la flecha en otro color.[CITATION Etq35 \l 3082]

2.5.2 Patrones de diseño del sistema.

Los patrones de diseño (GRASP⁷) o design patterns, son una solución general, reutilizable y aplicable a diferentes problemas de [diseño de software](#). Se trata de plantillas que identifican problemas en el sistema y proporcionan soluciones apropiadas a problemas generales a los que se han enfrentado los desarrolladores durante un largo periodo de tiempo, a través de prueba y error.[CITATION Etq36 \l 3082]

- ❖ **Experto en información:** Este patrón implica que cada objeto es responsable de almacenar su propia información, no la de los demás conceptos.[CITATION Etq36 \l 3082]
- ❖ **Creador:** Este patrón implica que un objeto debe responsabilizarse de crear otros: Si contiene o agrega varios objetos del tipo de los creados. Si se encarga de registrar objetos del tipo de los creados. Si utiliza mucho los objetos creados. O si contiene los datos para crear los del tipo creado.[CITATION Etq36 \l 3082]
- ❖ **Alta cohesión:** La alta cohesión la tiene un objeto si todo lo que hace está bien delimitado dentro del mismo objeto. Además, para tener alta cohesión, debe de ser un objeto cuanto más pequeño mejor. Este concepto está relacionado inversamente con el acoplamiento, es decir, si aumenta la cohesión, suele disminuir el acoplamiento, y viceversa.[CITATION Etq36 \l 3082]

⁷ **GRASP:** son patrones generales de software para asignar responsabilidades. Son guías, buenas ideas y buenos motivos para definir a qué clases asignarle ciertas responsabilidades.

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

- ❖ **Bajo acoplamiento:** El acoplamiento entre objetos se refiere a cuánta relación tienen los objetos entre sí. Por ejemplo, si para calcular el total de una Venta, tenemos que recorrer la Línea De Venta, entonces aumentamos el acoplamiento entre Venta y Línea De Venta. Pero si no lo hacemos la alternativa es almacenar el total en Venta, y actualizarlo con cada cambio en Línea De Venta. [CITATION Etq36 \l 3082]
- ❖ **Controlador:** Es un patrón por el cual definimos objetos llamados controladores que independizan las interfaces con las acciones que haya que hacer. [CITATION Etq36 \l 3082]

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

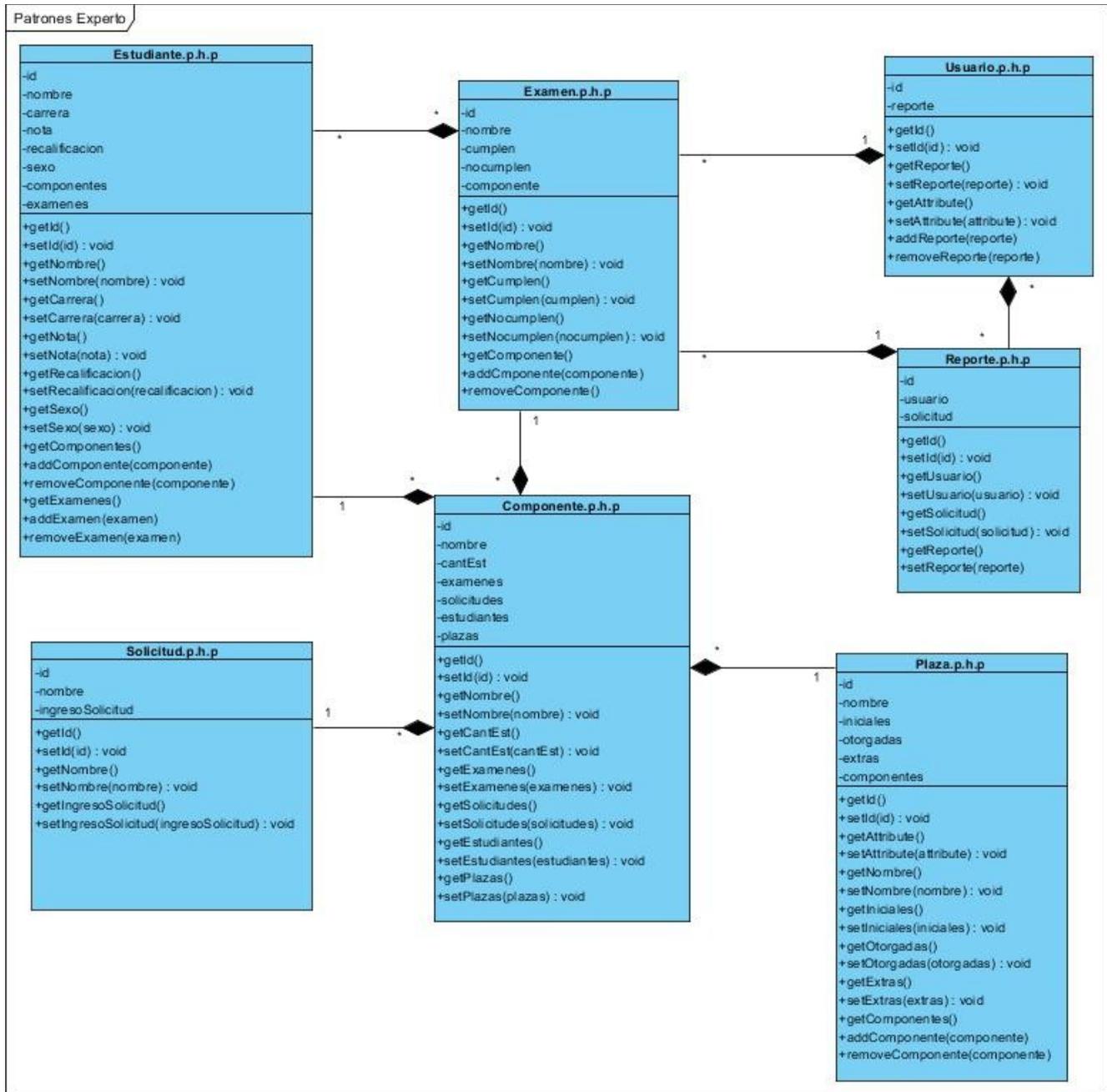


Figura 2: Diagrama que representa el uso del Patrón Experto en el módulo.

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

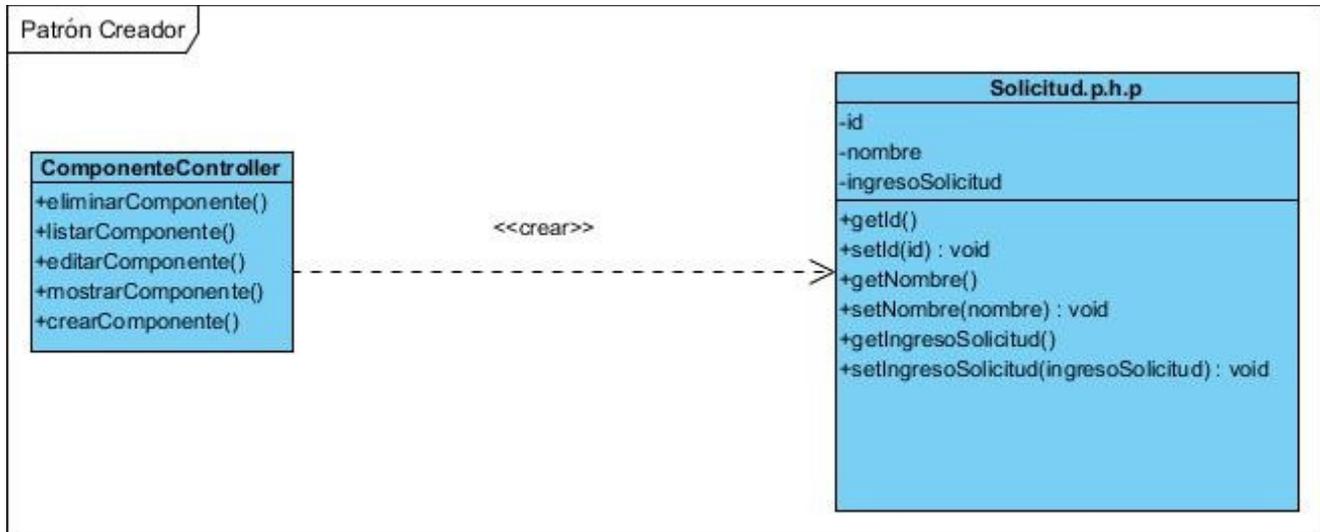


Figura 3: Diagrama que representa el uso del Patrón Creador en el módulo.

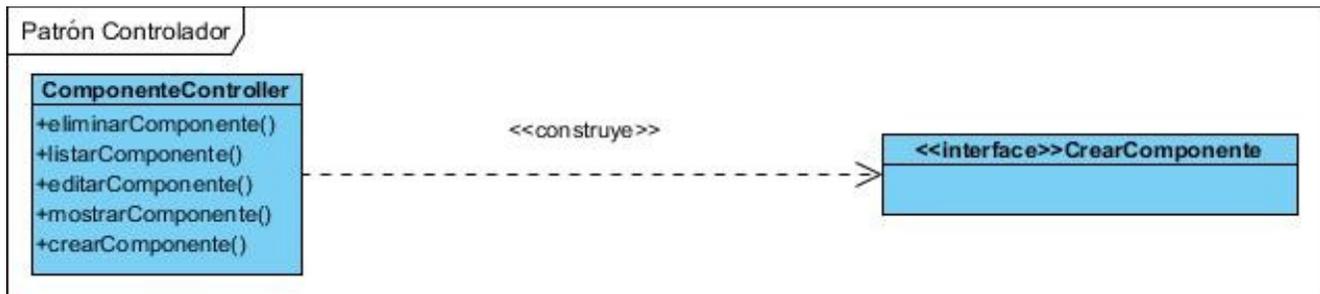


Figura 4: Diagrama que representa el uso del Patrón Creador en el módulo.

2.6 Diagrama de clases.

Los diagramas de clase del diseño son de vital importancia para describir gráficamente las clases del software. En la figura 5 se muestra el diagrama de clases del diseño.

2.7 Diagrama de componentes

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

El diagrama de componente permite visualizar la estructura de alto nivel del sistema y el comportamiento del servicio que estos componentes proporcionan y usan a través de

3082]

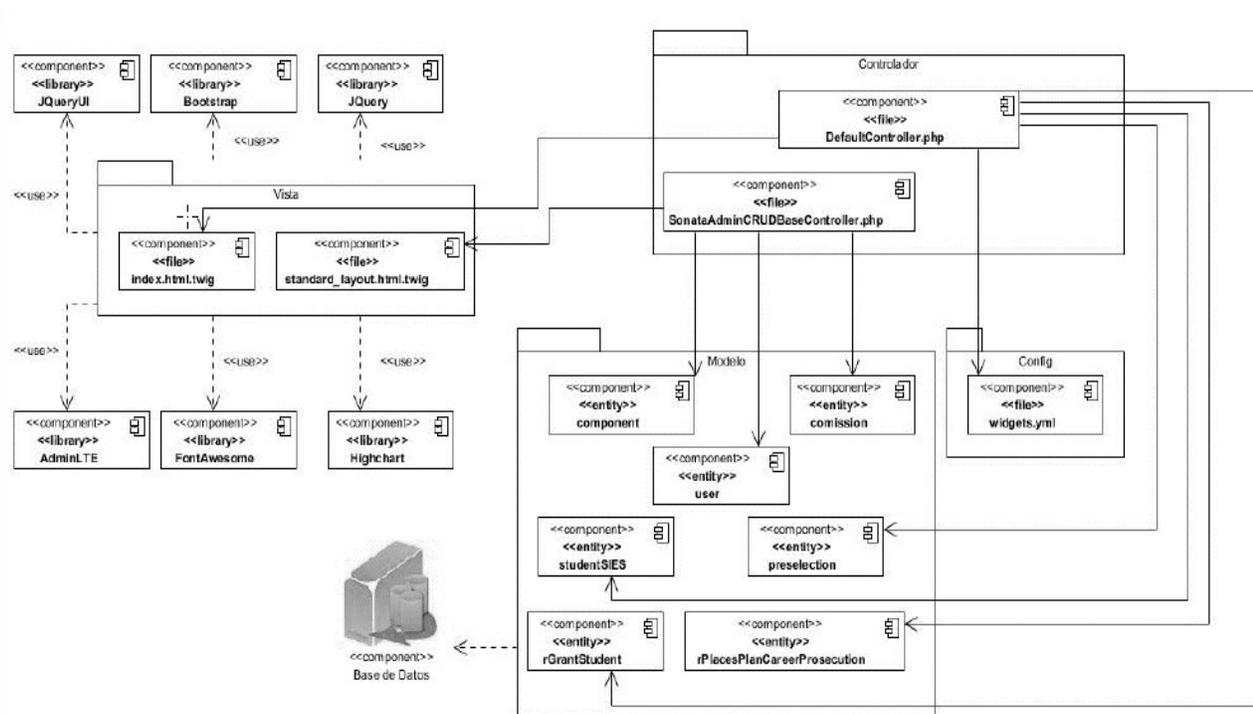
A continuación, se mencionan los estereotipos que conforman un diagrama de componente y posteriormente se muestra el diagrama de componente de la propuesta de solución:

`<<file>>`: ficheros que contienen el código fuente de la aplicación.

`<<entity>>`: entidades de la aplicación.

`<<library>>`: especifica una biblioteca de objetos estática o dinámica.

A continuación, se muestra el diagrama de componente de la propuesta de solución:



Capítulo 2: Análisis y diseño del Dashboard para SIGIES

Figura 6 Diagrama de componentes de la propuesta de solución. Fuente: Expediente del proyecto Sistema de Gestión para el ingreso a la Educación Superior.

2.8 Diagrama de Despliegue

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. [CITATION LaG22 \l 3082]

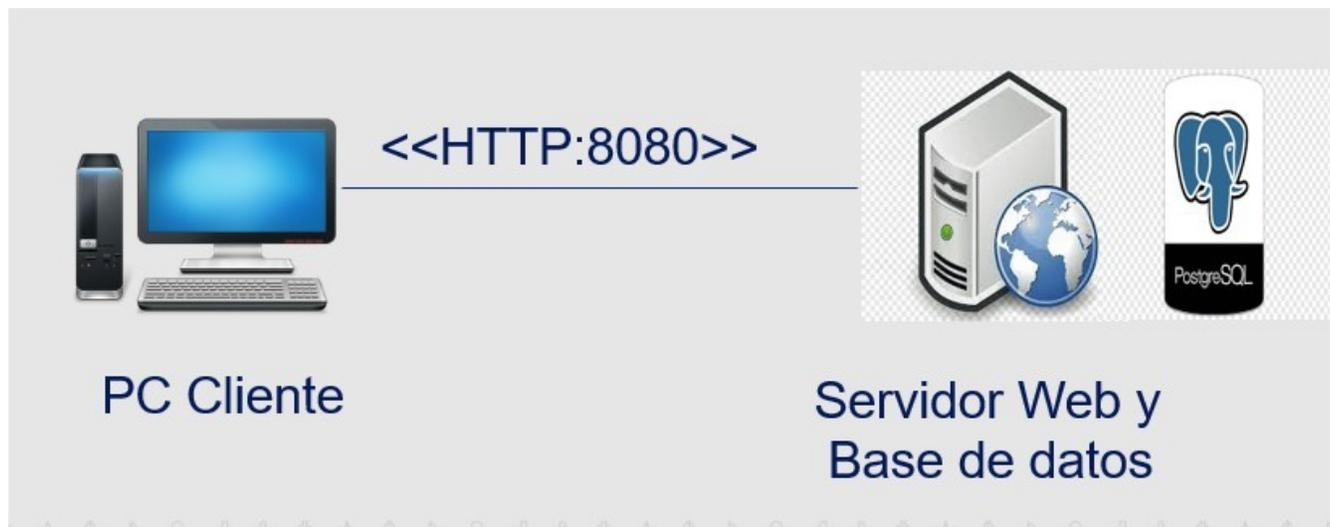


Figura 7: Diagrama de Despliegue (Elaboración propia)

Descripción de los Nodos:

- ❖ Nodo PC Cliente: Es el nodo que se encarga de instalar el sistema con el cual el usuario interactúa desde el espacio de trabajo. Esta PC debe tener instalado el SO Windows o Linux y un navegador web, Google Chrome o Mozilla Firefox como mínimo.
- ❖ Nodo Servidor de Aplicaciones: Es el nodo donde se encontrará la aplicación del sistema SIGIES empleando el paquete de software libre NGINX como servidor php, el lenguaje de programación PHP y el sistema de gestión de base de datos PostgreSQL, donde este último se encuentra almacenada toda la información de los datos del

Capítulo 2: Análisis y diseño del Dashboard para SIGIES

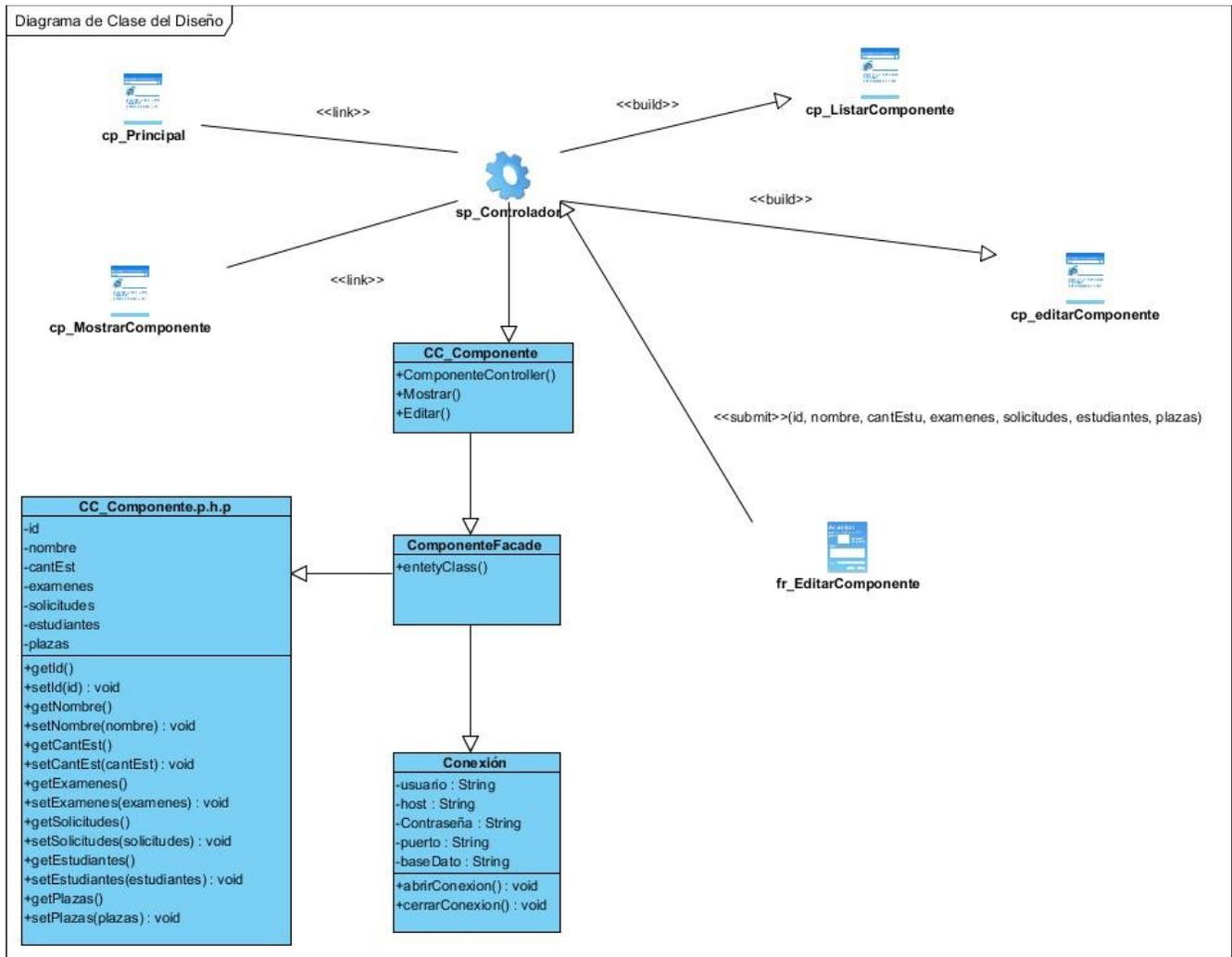
sistema. Además, se utiliza el sistema de gestión de paquetes Composer v2.4.4 y el framework Symfony v2.17 para el desarrollo de la aplicación web.

- ❖ Conexión TCP/IP: Este modelo es un protocolo para comunicación en redes que permite que un equipo pueda comunicarse dentro de una red

2.9 Conclusiones Parciales

Con el desarrollo de este capítulo se concluye que, fueron definidos los requisitos funcionales del sistema, los requisitos no funcionales y las diferentes herramientas y tecnologías empleadas para la captura y validación de los mismos para el SIGIES y así satisfacer las necesidades del cliente. Según la metodología AUP-UCI, el empleo de patrones GRASP y el diseño del diagrama de clase con estereotipo web fueron de gran ayuda para la realización de la propuesta de solución.

Conclusiones Generales



Capítulo 3: Implementación y pruebas a la solución propuesta.

Introducción

El análisis de la propuesta de solución mostrada en el capítulo anterior, ayuda al programador a entender el sistema y ser capaz de desarrollar su construcción y validación. Además de su implementación se realizan y describen un conjunto de pruebas de software a la aplicación, en este caso se utilizarán las pruebas funcionales, más específicamente el nivel de pruebas de aceptación y pruebas de integración con el objetivo de cumplir con las necesidades del cliente.

3.1 Implementación

Conclusiones Generales

La implementación no es más que una aplicación que describe con los resultados del modelo de diseño, el código final del sistema, que está determinado por el lenguaje de programación, con el objetivo de llevar a cabo la implementación de cada una de sus clases.

3.1.1 Estándares de codificación

Los estándares de codificación abarcan todos los aspectos generales del código fuente de una aplicación. Su importancia radica en la legibilidad del código para que sea entendible por todos los programadores de un proyecto como si este hubiese sido escrito por una sola persona. Cuando se añade código nuevo o se realice mantenimiento a la solución, el estándar de codificación debe sugerir como trabajar con el código ya existente. [CITATION SUÁ17 \l 3082]

En la implementación de la solución se utilizaron estándares de codificación tanto para Symfony como para PHP, los cuales son los utilizados por el proyecto SIGIES. A continuación, se definen los mismos:

Estructura Symfony:

- ❖ Se añade un solo espacio después del delimitador coma.
- ❖ Se añade una línea en blanco antes de las declaraciones *return*, a menos que el valor devuelto solo sea dentro de un grupo de declaraciones (tal como una declaración *if*).
- ❖ Se usan llaves para indicar la estructura del cuerpo de control, independientemente del número de declaraciones que contenga.
- ❖ Se define una clase por archivo.
- ❖ Se declara las propiedades de las clases antes de los métodos

Convenciones de nomenclatura:

- ❖ Se utiliza mayúsculas, sin guiones bajos, en nombre de variable, función, método o argumentos.

Conclusiones Generales

- ❖ Se usa guiones bajos para nombre de opción y nombres de parámetro.
- ❖ Se utiliza caracteres alfanuméricos y guiones bajos para los nombres de archivos.

Tamaño máximo de línea en el código PHP:

- ❖ La longitud recomendable para una línea de código es de 80 caracteres. Esto significa que los desarrolladores deberían intentar mantener cada línea de su código por debajo de los 80 caracteres, siempre que sea posible. No obstante, líneas más largas pueden ser aceptables en algunas situaciones. El tamaño máximo de cualquier línea de código PHP es de 120 caracteres.

Nombre de archivos:

- ❖ Para los archivos, sólo caracteres alfanuméricos, barras bajas (`_`) y guiones (`-`) están permitidos. Los espacios en blanco están estrictamente prohibidos.
- ❖ Cualquier archivo que contenga código PHP debe terminar con la extensión " `php` ", con la excepción de los *scripts* de la vista.

Funciones y métodos:

- ❖ Los nombres de funciones pueden contener únicamente caracteres alfanuméricos. Los guiones bajos (`_`) no están permitidos. Los números están permitidos en los nombres de función, pero no se aconseja en la mayoría de los casos.
- ❖ Los nombres de funciones deben empezar siempre con una letra minúscula. Cuando un nombre de función consiste en más de una palabra, la primera letra de cada nueva palabra debe estar en mayúsculas.

Variables:

Conclusiones Generales

- ❖ Los nombres de variables pueden contener caracteres alfanuméricos. Las barras bajas (_) no están permitidas. Los números están permitidos en los nombres de variables, pero no se aconseja en la mayoría de los casos.

Constantes:

- ❖ Las constantes pueden contener tanto caracteres alfanuméricos como barras bajas (_). Los números están permitidos. Todas las letras pertenecientes al nombre de una constante deben aparecer en mayúsculas.

Declaración de clases:

- ❖ Las Clases deben ser nombradas de acuerdo a las convenciones de nombres. La llave "{" deberá escribirse siempre en la línea debajo del nombre de la clase.

Declaración de funciones y métodos:

- ❖ Las funciones deben ser nombradas de acuerdo a las convenciones de nombrado establecido.
- ❖ Los métodos dentro de clases deben declarar siempre su visibilidad usando un modificador *private*, *protected* o *public*.

3.2 Pruebas de software

Las pruebas del software implican ejecutar una implementación del software con datos de prueba. Se examinan las salidas del software y su entorno operacional para comprobar que funciona tal y como se requiere. Las pruebas son una técnica dinámica de verificación y validación. [CITATION Cát \l 3082]

Métodos de las pruebas

- ❖ Pruebas unitarias de caja blanca: Las pruebas de caja blanca (también llamadas estructurales o de caja transparente) describen pruebas o métodos en los que

Conclusiones Generales

se conocen los detalles y el funcionamiento interno del software que se está probando. Dado que se conoce las funciones, los métodos, las clases, cómo funcionan y cómo se unen, generalmente está mejor equipado para examinar la integridad lógica del código.[CITATION Dev21 \l 3082]

- ❖ Pruebas funcionales de caja negra: Por el contrario, las pruebas de caja negra (también llamadas funcionales, de comportamiento o de caja cerrada) describen cualquier prueba o método en el que se desconocen los detalles y el funcionamiento interno del software que se está probando.[CITATION Dev21 \l 3082]

Dado que no conoce ninguno de los detalles, realmente no puede crear casos de prueba que se dirijan a escenarios de nicho específicos o que hagan hincapié en la lógica específica del sistema. Lo único que sabe es que, para una solicitud o una entrada determinada, se espera un determinado comportamiento o salida. Por lo tanto, las pruebas de caja negra prueban principalmente el comportamiento de un sistema. Las pruebas de un extremo a otro suelen ser una caja negra.[CITATION Dev21 \l 3082]

Cada una de estas técnicas cuentan con distintos niveles de pruebas que a continuación se mencionan:

- ❖ Prueba de desarrollador.
- ❖ Prueba independiente.
- ❖ Prueba de unidad.
- ❖ Prueba de integración.
- ❖ Prueba de sistema.
- ❖ Prueba de aceptación.

Es necesario realizarle pruebas al software para poder garantizarle al cliente una calidad requerida. Existen diferentes tipos de pruebas con el objetivo de detectar errores, anomalías o cualquier otra información en los resultados del programa, orientadas a comprobar su funcionalidad. verificando que el sistema cumpla con las especificaciones que pide el cliente.

Conclusiones Generales

A continuación, se muestran a través del método de caja negra, las pruebas realizadas al sistema mediante las pruebas de aceptación e integración.

3.2.1 Prueba de Aceptación

Las pruebas de aceptación suelen ser una fase del ciclo de desarrollo. El objetivo principal de las pruebas de aceptación es verificar que un producto o característica determinada se haya desarrollado de acuerdo con las especificaciones establecidas por un cliente o un interesado interno, como un gerente de producto. [CITATION Dan17 \l 3082]

Dentro de las pruebas de aceptación, también puede haber múltiples fases, como las pruebas α o las pruebas β . A medida que gran parte del mundo del desarrollo de software avanza hacia los procesos ágiles, las pruebas de aceptación del usuario se han vuelto mucho menos rígidas y más colaborativas. [CITATION Dan17 \l 3082]

A continuación, se muestran una parte de los CP correspondientes a las pruebas de aceptación realizadas a la aplicación:

Tabla 8: Caso de prueba de aceptación para la HU "Editar componentes del dashboard".

Caso de Prueba de Aceptación	
Código: R27_P1	Historia de Usuario: 6
Nombre: Editar componentes del dashboard.	
Descripción: El sistema debe permitir seleccionar los usuarios y las comisiones que van a visualizar los componentes.	
Condiciones de Ejecución: Usuario con permisos del usuario Admin_DIUL debe estar autenticado y deben existir modelos en la base de datos del sistema.	
Entrada/Pasos de ejecución:	
<ul style="list-style-type: none">❖ Clic en la opción "Editar Componentes" en el módulo Componentes que se encuentra en el menú lateral izquierdo.❖ Seleccionar el usuario y la comisión que desea editar y de este la opción de editar componentes.❖ Introducir los datos del/los campos que se desean editar (Nombre y Descripción).❖ Clic en el botón "Aceptar".	
Resultado esperado: El sistema debe modificar en la base de datos los valores introducidos en el modelo. Se muestra un mensaje de operación satisfactoria.	

Conclusiones Generales

Resultado obtenido:
Evaluación de la prueba: Satisfactoria.

Tabla 9: Caso de prueba de aceptación para la HU "Exportar Componentes del dashboard".

Caso de Prueba de Aceptación	
Código: R28_P2	Historia de Usuario: 7
Nombre: Exportar componente del dashboard.	
Descripción: El sistema debe permitir al usuario exportar los datos de los componentes.	
Condiciones de Ejecución: Usuario con permisos del usuario Admin_DIUL debe estar autenticado y deben existir modelos en la base de datos del sistema.	
Entrada/Pasos de ejecución:	
<ul style="list-style-type: none"> ❖ Clic en la opción "Exportar Componentes" en el módulo Componentes que se encuentra en el menú lateral izquierdo. ❖ Seleccionar el usuario que desea exportar. ❖ Clic en el botón "Exportar". 	
Resultado esperado: El sistema debe exportar de la base de datos los valores introducidos en el modelo. Se muestra un mensaje de operación satisfactoria.	
Resultado obtenido:	
Evaluación de la prueba: Satisfactoria.	

Figura 10: Caso de prueba de aceptación para la HU "Listar componente del dashboard".

Caso de Prueba de Aceptación	
Código: R25_P3	Historia de Usuario: 8
Nombre: Listar componente del dashboard.	
Descripción: El sistema debe permitir listar los tipos de componentes y sus títulos para que el usuario seleccione el que desea ver.	
Condiciones de Ejecución: Usuario con permisos del usuario Admin_DIUL debe estar autenticado y deben existir modelos en la base de datos del sistema.	
Entrada/Pasos de ejecución:	
<ul style="list-style-type: none"> ❖ Clic en la opción "Listar Componentes" en el módulo Componentes que se encuentra en el menú lateral izquierdo del sistema. ❖ Seleccionar los datos en los campos (títulos y tipos de componentes) que el usuario desea ver. ❖ Clic en el botón "Listar". 	
Resultado esperado: El sistema debe exportar de la base de datos los valores introducidos en el modelo. Se muestra un mensaje de operación satisfactoria.	
Resultado obtenido:	
Evaluación de la prueba: Satisfactoria.	

Conclusiones Generales

3.2.2 Prueba de Integración.

De las pruebas software existentes mencionadas con anterioridad de integración son un método de prueba centrado en examinar varios componentes juntos. El objetivo principal de las pruebas de integración es garantizar la integridad de la relación y el flujo de datos entre componentes o unidades.[CITATION Dev21 \l 3082]

A continuación, se muestra la imagen evidenciándose la integración del módulo desarrollado con el SIGIES:



Figura 11 Integración de las cajas de información en el Dashboard del SIGIES (Elaboración propia).

Conclusiones Generales

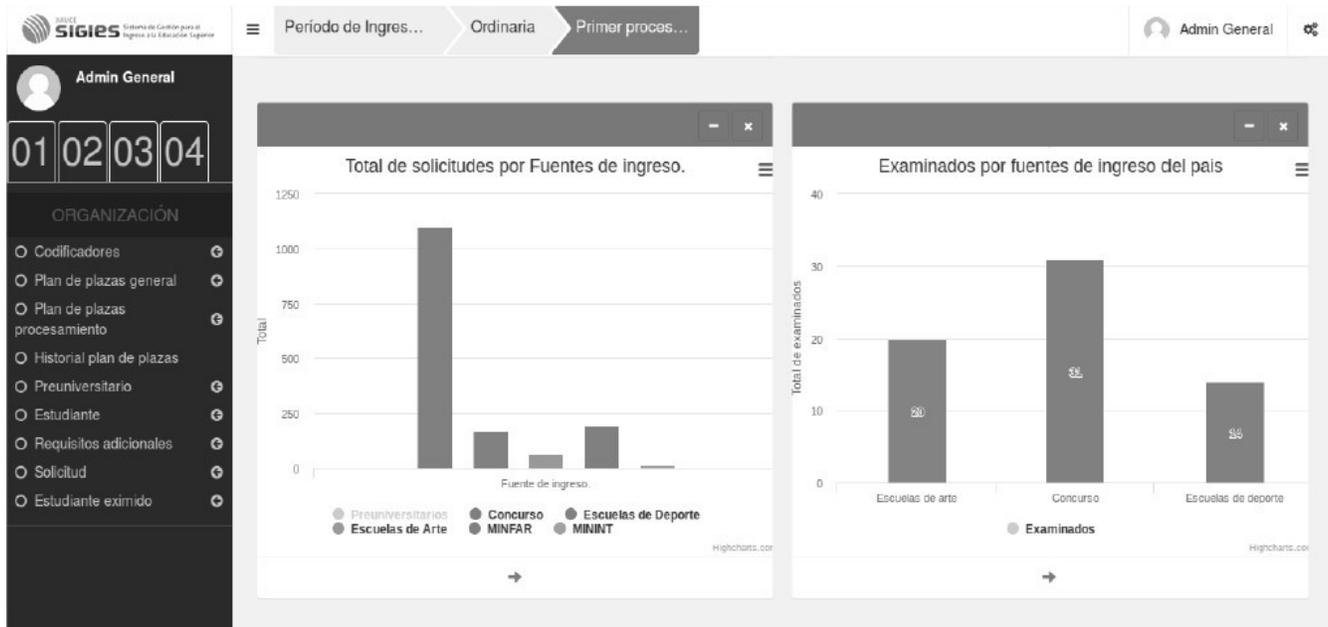


Figura 12 Integración de los gráficos de barra en el Dashboard del SIGIES (Elaboración propia).



Figura 13 Integración de los gráficos de pastel en el Dashboard del SIGIES (Elaboración propia).

3.2.3 Resultado de las pruebas de Caja Negra

A través del método de caja negra, para validar los requisitos funcionales se realizaron 3 iteraciones, realizando en las dos primeras pruebas de regresión con el objetivo de

Conclusiones Generales

comprobar que los errores encontrados fueron corregidos. A continuación, se muestran los resultados obtenidos en cada una de las iteraciones de pruebas realizadas sistema SIGIES. Como se puede apreciar en la primera iteración fueron detectadas 5 no conformidades (NC) de las cuales fueron resueltas las 5. Luego en la segunda iteración se encontraron 3 nuevas NC siendo resueltas las mismas. Por último, se realizó una tercera iteración, la cual no arrojó nuevas NC, por lo que no fue necesario ejecutar una cuarta iteración de pruebas. Entre las no conformidades detectadas durante el proceso de pruebas se destacan las siguientes:

- ❖ Errores de ortografía.
- ❖ Errores de Interfaz.
- ❖ Errores Funcionales.
- ❖ Errores en los formatos de las páginas.

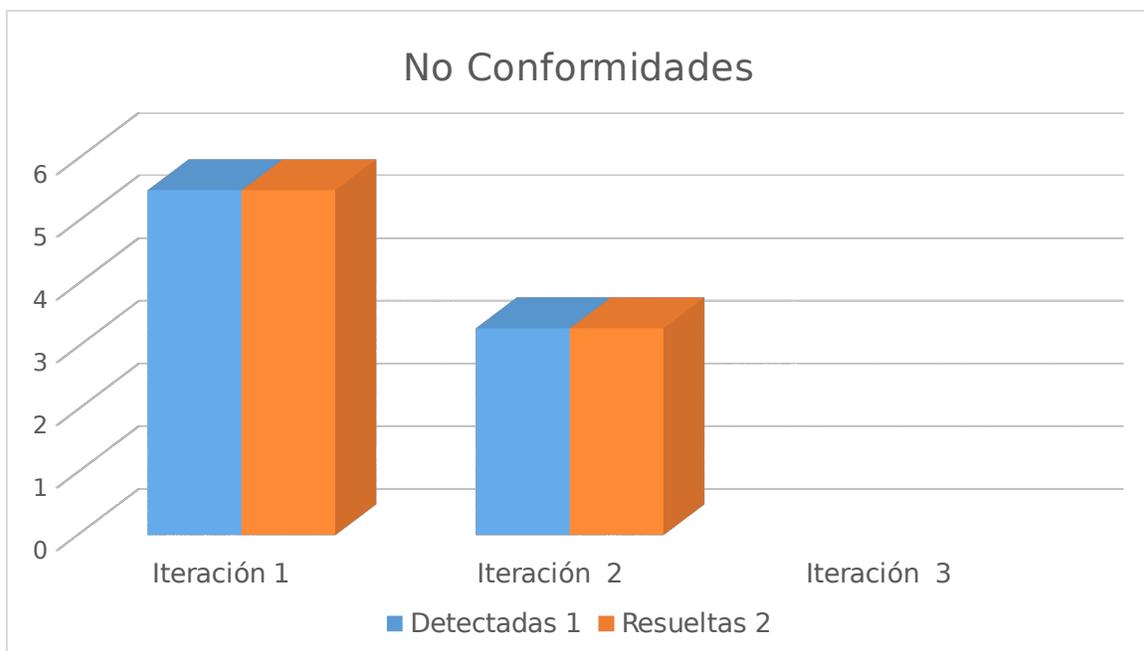


Figura 14 Resumen de la cantidad de No Conformidades detectadas por cada una de las iteraciones de prueba (Elaboración propia).

Conclusiones Generales

3.3 Conclusiones Parciales.

Después de realizado el presente capítulo se arribó a las siguientes conclusiones:

- ❖ Se especificaron los tipos de pruebas a realizar por cada uno de los niveles de pruebas, ya sean nivel de pruebas de integración y de aceptación.
- ❖ La realización de las pruebas de software permitió evaluar la calidad del módulo desarrollado y detectar no conformidades, las cuales fueron corregidas en cada iteración.
- ❖ La aplicación de las pruebas funcionales, mediante sus métodos y técnicas, permitieron evaluar las funcionalidades del software.

Conclusiones Generales

Al completar el desarrollo de la propuesta de solución se le dio cumplimiento al objetivo planteado inicialmente, obteniendo como resultado principal el módulo que permite gestionar el dashboard para SIGIES. Con el cual se concluye lo siguiente:

- ❖ El uso de un conjunto de patrones y buenas prácticas de programación, permitió diseñar e implementar diferentes clases que dan cumplimiento a los requisitos funcionales y no funcionales de la propuesta de solución.
- ❖ La utilización de la metodología AUP-UCI, facilitó la implementación de la plataforma SIGIES mediante los artefactos generados.
- ❖ Las pruebas realizadas al sistema permitieron obtener una propuesta que satisface los requisitos funcionales definidos durante la investigación.
- ❖ La propuesta de solución permitió mostrar a los usuarios CIP y DIUL información de forma concisa de los indicadores gestionados en el proceso de Ingreso a la Educación Superior y que responda a las necesidades actuales del cliente.

Conclusiones Generales

Recomendaciones

Recomendaciones

- ❖ Se recomienda para futuras versiones desarrollar otros tipos de componentes como gráficos lineales.
- ❖ Desarrollar un calendario que permita gestionar el cronograma de ingreso a la Educación Superior.

Recomendaciones

- Ana Lucía. 2020.** Platzi. [En línea] 2020. <https://platzi.com/blog/que-es-tableau/>.
- Andrea Flores. 2022.** CREHANA. [En línea] 16 de mayo de 2022. <https://www.crehana.com/blog/transformacion-digital/que-es-power-bi/>.
- Ariel.Saavedra. 2019.** servidor nginx. 6 de NOVIEMBRE de 2019.
- Bach. Ponce Córdoba Jorge Luis. 2017.** 2017.
- Betania V. 2022.** HOSTINGER TUTORIALES. [En línea] 25 de mayo de 2022. <https://www.hostinger.es/tutoriales/sgbd>.
- Blanco,Manuel Porras. 2017.** LOGICALIS . [En línea] 29 de Septiembre de 2017. [Citado el: 7 de junio de 2022.] <https://blog.es.logicalis.com/analytics/kpis-qu%C3%A9-son-para-qu%C3%A9-sirven-y-por-qu%C3%A9-y-c%C3%B3mo-utilizarlos>.
- By ReQtest. 2012.** Why is the difference between functional and Non-functional requirements important? *ReQtest*. [En línea] 5 de abril de 2012. <http://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/>.
- Canelo, Mirian Mar. 2020.** ¿Qué son los patrones de diseño de software ? [En línea] 24 de junio de 2020. <https://profile.es/blog/patrones-de-diseno-de-software/>.
- capterra. [En línea] <https://www.capterra.es/software/145716/visual-paradigm>.
- Capterra. [En línea] <https://www.capterra.es/software/86955/sisense>.
- Castro, Julio. 2021.** [En línea] 9 de Julio de 2021. [Citado el: 5 de Junio de 2022.] <https://blog.corponet.com/que-es-la-inteligencia-de-negocios>.
- Cátedra: Ingeniería del software.** Verificación y validación del software. [En línea] <https://suriweb.com.ar/wp/tecnologia/wp-content/uploads/sites/18/2019/03/Verificacion-y-Validacion-de-SW.pdf>.
- Daniel Ramos, Raúl Noriega, José Rubén Laínez, Alicia Durango. 2017.** *Curso de Ingeniería de Software: 2ª*. 2017.
- Drew, Equipo de redacción de. 2020.** [En línea] 29 de julio de 2020. [Citado el: 7 de junio de 2022.] <https://blog.wearedrew.co/scoro-el-erp-con-mas-crecimiento-para-que-sirve>.
- Ekaterina Novoseltseva. 2020.** Los 5 principales patrones de Arquitectura de Software. [En línea] 2 de junio de 2020. <https://apiumhub.com/es/tech-blog-barcelona/principales-patrones-arquitectura-software/>.
- Gandarillas, Aurelio. 2017.** Metodología . [En línea] 25 de junio de 2017. <https://metodologia.es/marco-de-trabajo/>.
- Group, by Bantu. 2020.** Blog oficial de Bantu. [En línea] 20 de abril de 2020. [Citado el: 5 de junio de 2022.] <https://www.bantugroup.com/blog/almacen-de-datos-que-es-y-como-beneficia-tu-empresa>.
- Guest. 2020.** Bootstrap: guía para principiantes de qué es, por qué y cómo usarlo. [En línea] 12 de Abril de 2020. <https://rockcontent.com/es/blog/bootstrap/>.
- 2021.** Gunka Studios. [En línea] 7 de Abril de 2021. <https://gunkastudios.com/symfony-que-es-para-que-utilizarlo/>.
- HostGator. 2021.** jQuery: aprende qué es y cómo puedes usarlo. [En línea] 12 de abril de 2021. <https://www.hostgator.mx/blog/jquery-que-es-y-como-usarlo/>.
- Ismary Guzmán Fernández. 2011.** 2011.
- Ivan de Souza. 2019.** rockcontent | blog. *¿Qué es un servidor web y para qué sirve en Internet?* [En línea] 14 de junio de 2019. <https://rockcontent.com/es/blog/que-es-un-servidor/>.
- Javier Eguiluz . 2022.** KINSTA. 2022.
- Jesuites educación, Formacion Profesional. 2022.** Desarrollo de Aplicaciones Mutiplataforma. [En línea] 26 de enero de 2022. <https://fp.uoc.fje.edu/blog/que-es-bootstrap-y-cuales-son-sus-caracteristicas/>.
- José Manuel Ortega Candel. 2018.** *Seguridad en aplicaciones Web Java*. España : Ra-Ma 2018, 2018.
- Kumar, Ajitesh. 2022.** [En línea] 28 de mayo de 2022. https://vitalflux.com/dashboard-purpose-advantages-disadvantages/#What_are_the_disadvantages_of_using_a_dashboard.
- . 2022. TuDashboard. [En línea] 28 de mayo de 2022. https://vitalflux.com/dashboard-purpose-advantages-disadvantages/#What_are_the_advantages_of_using_a_dashboard.
- Latam, Devops. 2021.** 15 métodos de prueba que todos los desarrolladores deben conocer. [En línea] 15 de abril de 2021. <https://devopslatam.com/15-metodos-de-prueba-que-todos-los-desarrolladores-deben-conocer/>.
- Luis Soto Soto. 2017.** [En línea] 2017. <https://www.monografias.com/docs114/implementacion-componentes/implementation-componentes>.
- María Ángeles Arroyo Montes. 2019.** Diagrama de despliegue. [En línea] 2019. <https://docplayer.es/92849368-Diagrama-de-despliegue.html>.

Recomendaciones

- Max Rehkopf. 2022.** Historias de usuarios con ejemplos y plantillas. [En línea] 2022. <https://www.atlassian.com/es/agile/project-management/user-stories>.
- Melissa Pardo-Bunte. 2022.** BetterBuys. [En línea] 1 de septiembre de 2022. <https://www.betterbuys.com/bi/reviews/sisense-business-intelligence/>.
- . 2022. Compara Software. [En línea] 1 de septiembre de 2022. <https://www.betterbuys.com/bi/reviews/sisense-business-intelligence/>.
- Melo, Sandra. 2022.** DataScoper. [En línea] 4 de enero de 2022. [Citado el: 7 de junio de 2022.] <https://datascope.io/es/blog/las-mejores-herramientas-para-hacer-un-dashboard-en-2022/>.
- Miguel Angel Alvarez. 2020.** Qué es MVC. [En línea] 28 de julio de 2020. <https://desarrolloweb.com/articulos/que-es-mvc.html>.
- Miranda, Elizabeth Cabalé. 2020.** s.l. : 14 de abril del 2020, 28 de julio de 2020, Vol. vol.8 . no 1.
- Mujica-Sequera, Ruth M. 2022.** Medio Oriente : Grupos Docentes 2.0, 1 de junio de 2022.
- NORLYS SUÁREZ VALDERRAMA. 2017.** Herramienta para detectar cambios en el código fuente del Sistema de Gestión para el Ingreso a la Educación Superior (SIGIES). Tesis de Licenciatura. Universidad de las Ciencias Informáticas. Facultad 4, 2017.
- Ortiz, . 2021.** cyberclick.es. [Online] 20 10 2021. <https://www.cyberclick.es/numerical-blog/que-es-un-dashboard>.
- Peláez, Alonso Toro y Luis.E. 2016.** 20, Pereira : s.n., 2016, Vol. 10.
- Pires, Raphael. 2021.** Rock Content. [En línea] 14 de octubre de 2021. [Citado el: 7 de junio de 2021.] <https://rockcontent.com/es/blog/dashboard/>.
- RANJAN, JAYANTHI. 2010.** India : s.n., 2010, Vol. 9Vol9No1.
- Robledano, Angel. 2019.** OpenWebinars. [En línea] 26 de junio de 2019. <https://openwebinars.net/blog/que-es-css/>.
- Santander Universidades. 2020.** Santander. [En línea] 21 de diciembre de 2020. <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>.
- Solano, Alex Arenols. 2019.** OpenWebinars. [En línea] 1 de enero de 2019. [Citado el: 8 de junio de 2022.] <https://openwebinars.net/blog/que-es-php/>.
- Studios, Gunka. 2021.** Gunka Studios. [En línea] 7 de abril de 2021. [Citado el: 8 de junio de 2022.] <https://gunkastudios.com/symfony-que-es-para-que-utilizarlo/>.
- Thiago Soares. 2020.** ipnet. *¿Qué características tiene un dashboard de calidad?* [En línea] 25 de septiembre de 2020. <https://es.blog.ipnet.cloud/dashboard/>.
- Visual Paradigm User's Guide. 2020.** Visual Paradigm User's Guide. [En línea] 21 de febrero de 2020. <https://www.download3k.es/Desarrollo-Software/Otros/Download-Visual-Paradigm-for-UML-Standard-Edition-for-Windows.html>.
- Workana. 2012 - 2021.** [En línea] 2012 - 2021. <https://i.workana.com/glosario/que-es-html-5/>.
- Yenisei Molina Hernández. 2019.** s.l. : Ediciones Futuro, 20 de abril de 2019, Vol. 13. No.2.
- Yeniseidy Fernández Romero, Yanette Díaz González. 2012.** El patrón modelo-vista-controlador: Arquitectura y frameworks explicados. Revista Telem@tica., enero-abril de 2012, Vol. 11.

Anexos

Anexos

Anexo 1: Entrevista realizada al jefe del proyecto SIGIES durante el levantamiento de información.

1. ¿A qué usuarios va dirigido el producto?
2. ¿Qué nivel de accesibilidad a la información según los roles tendrán los usuarios?
3. ¿Qué información es necesaria mostrar?
4. ¿Cuáles serían los componentes que usted diseñaría para mostrar esa información?