

Universidad de Matanzas

Sede “Camilo Cienfuegos”



Facultad de Ciencias Técnicas

Departamento de Informática

PROPUESTA DE SISTEMA INFORMÁTICO DE GESTIÓN DE LA PROGRAMACIÓN CULTURAL

Trabajo de Diploma en opción al Título de Ingeniero Informático

Autor: Osvaldo Fernández Naranjo

Tutora: Msc. Liana Romero Lovio

Cotutor: Ing. Armando Sánchez

Matanzas, 2020

Dedicatoria

Este trabajo se lo dedico a mi familia en especial a mi madre que siempre me apoyó en todo.

Agradecimientos

Quiero agradecer a todas aquellas personas que me ayudaron para la realización de este en especial a mi familia y mis amigos, gracias. Un agradecimiento especial a Ariel del Rio que siempre me ha ayudado.

Declaración de autoría

Yo, Osvaldo Javier Fernández Naranjo, declaro que soy el único autor del trabajo “Propuesta de Sistema Informático de Gestión de la Programación Cultural” y autorizo a la Universidad de Matanzas y al Centro de Investigación y Desarrollo de Software, hacer uso del mismo en su beneficio.

Y para que así conste, firmo la presente a los _____ días del mes de junio del año 2020.

Firma del Autor

Firma del Tutor

Resumen

Las Tecnologías de la Información y de las Comunicaciones nos permiten integrar en espacios virtuales todas las actividades necesarias del día a día, lo que impulsa el avance con las estrategias de informatización y con la adopción del gobierno electrónico, esto ocurre en el mundo y Cuba no se encuentra ajena al tema. En la provincia de Matanzas se implementa esta estrategia, con el fin de facilitar los trámites de la población y de poder darle a los ciudadanos, al gobierno y a las empresas, medios para expresar inquietudes, informar y manejar datos. El objetivo principal de la presente investigación es diseñar una propuesta de aplicación web para la gestión de los procesos de programación de todas las actividades culturales para la Dirección Provincial de Cultura en la provincia de Matanzas. Actualmente este proceso se realiza de forma manual, lo que trae como consecuencia que las gestiones no sean eficientes, la información en los municipios se atrase y se pierda información, lo que genera ineficiencia en todos los renglones de la programación de las actividades culturales. La metodología que se utiliza para fundamentar el desarrollo de la investigación es Prodesoft.

Abstract

Information and Communication Technologies allow us to integrate all the necessary day-to-day activities into virtual spaces, which drives progress with computerization strategies and the adoption of electronic government, this happens in the world and Cuba does not is oblivious to the subject. In the province of Matanzas, this strategy is implemented in order to facilitate the population's procedures and to be able to give citizens, the government and companies the means to express concerns, inform and manage data. The main objective of this research is to design a web application proposal for the management of the programming processes of all cultural activities for the Provincial Directorate of Culture in the province of Matanzas. Currently this process is carried out manually, which means that the procedures are not efficient, the information in the municipalities is delayed and information is lost, which generates inefficiency in all lines of the programming of cultural activities. The methodology used to support the development of the research is Prodesoft.

Contenido

Introducción	- 1 -
Capítulo 1: Marco teórico referencial de la Investigación.	- 5 -
1.1.- Introducción.	- 5 -
1.2. - Antecedentes.	- 5 -
1.3.- Análisis de soluciones existentes.	- 6 -
1.4.- Metodología de desarrollo.	- 7 -
1.5.- Tecnologías y Herramientas.	- 8 -
1.5.1.- Herramienta del modelado.	- 8 -
1.5.2.- Lenguajes de Modelado	- 8 -
1.5.3.- Lenguajes de Programación.	- 9 -
1.5.4.- Lenguajes de Consulta.	- 11 -
1.5.5.- Gestor de Base de Datos.	- 11 -
1.5.6.- Framework.	- 12 -
1.5.7.- Servicios Web	- 13 -
1.6.- Conclusiones del Capítulo	- 14 -
Capítulo 2: Análisis y diseño de la solución propuesta.	- 15 -
2.1.- Introducción.	- 15 -
2.2.- Modelado del negocio.	- 15 -
2.2.1.- Descripción del negocio.	- 15 -
2.2.2.- Reglas del negocio.	- 16 -
2.3.- Requisitos del sistema informático.	- 17 -
2.3.1.- Requisitos No Funcionales.	- 17 -
2.3.2.- Requisitos Funcionales.	- 18 -
2.3.3.- Especificación de los requisitos funcionales.	- 20 -
2.4 Implementación de la Arquitectura de Software	- 23 -
2.4.1 Arquitectura N-Capas.	- 24 -
2.5 Patrones Utilizados.	- 25 -
2.6 Mecanismos de Diseño.	- 25 -

2.7 Diseño de la Base de Datos.....	- 26 -
2.8 Diagramas de Clases de Diseño.	- 28 -
2.9 Diagramas de Secuencia.....	- 29 -
2.10 Seguridad.....	- 30 -
2.11 Análisis de factibilidad.	- 31 -
2.11.1 Costo.....	- 31 -
2.11.2 Beneficios Tangibles e intangibles	- 32 -
2.11.3 Análisis del costo y beneficios	- 32 -
2.12 Diseño de Pruebas de Software.....	- 33 -
2.12.1 Objetivos de las Pruebas.....	- 33 -
2.12.2 Pruebas del Software.....	- 33 -
2.13 Conclusiones Parciales.....	- 44 -
Conclusiones	- 45 -
Recomendaciones	- 46 -
Bibliografía.....	- 47 -
Anexos.....	- 1 -

Índice de figuras:

Fig. 1 Diagrama del proceso del negocio	- 15 -
Fig. 2 Modelo Conceptual.	- 17 -
Fig. 3 Diagrama de Actividad del Requisito Funcional Gestionar Planificación.....	- 21 -
Fig. 4 Diagrama de Actividad del Requisito Funcional Gestionar Informes de Cumplimiento. -	22 -
Fig. 5 Diagrama de Actividad del Requisito Funcional Gestionar Encuestas.....	- 23 -
Fig. 6 Diagrama de Mecanismo de Diseño.....	- 26 -
Fig. 7 Diagrama del Modelo Lógico de Base de Datos.....	- 27 -
Fig. 8 Diagrama del Modelo Físico de Base de Datos.....	- 27 -
Fig. 9 Diagrama de Clases de Diseño del Requisito Funcional Gestionar Planificación. .	- 28 -
Fig. 10 Diagrama de Clases de Diseño del Requisito Funcional Gestionar Informe de Cumplimiento.....	- 28 -
Fig. 11 Diagrama de Clases de Diseño del Requisito Funcional Gestionar Encuestas....	- 29 -
Fig. 12 Diagrama de Secuencia del Requisito Funcional Gestionar Planificación.	- 29 -
Fig. 13 Diagrama de Secuencia del Requisito Funcional Gestionar Informes de Cumplimiento.....	- 30 -
Fig. 14 Diagrama de Secuencia del Requisito Funcional Gestionar Encuestas.....	- 30 -

Introducción

La informática por su gran desarrollo, ha venido transformando rápidamente las sociedades actuales. Las nuevas tecnologías nos permiten avanzar a pasos agigantados, ya que son aplicadas en los distintos ámbitos del quehacer cotidiano del hombre, al punto que han revolucionado la sociedad de una manera sorprendente y su desarrollo ha contribuido a la modernización y a un manejo totalmente diferente de la información.

El avance tecnológico que se ha alcanzado en los últimos tiempos, además del mayor aprovechamiento de las tecnologías de la información y las comunicaciones, ha propiciado un auge en la aparición de software de gestión en el contexto nacional. Estos cambios han alcanzado mayor auge gracias a la rápida expansión de Internet y las aplicaciones web.

Los denominados gobiernos electrónicos son el mayor avance y aporte del uso de las tecnologías de la información y las comunicaciones a los procesos gubernamentales y la relación de los mismos con sus ciudadanos. Cuba reconoce la necesidad de establecer ese gobierno electrónico en su plan de informatización de la sociedad y los avances que provoca en el desarrollo de la sociedad socialista. Como parte de la implementación del gobierno electrónico en Cuba se han desarrollado diferentes proyectos como son los portales del ciudadano donde cada provincia cuenta con un sitio web que propicia un catálogo en línea con descripciones sobre servicios y trámites.

En el portal del ciudadano matancero se explica además que el gobierno electrónico es el empleo intensivo de las tecnologías de la información y las comunicaciones para la gestión y administración pública y constituye una nueva forma de ejercicio de gobierno. (Gobierno Matancero, 2019)

En la provincia de Matanzas gracias a la voluntad del Gobierno Provincial y los Gobiernos Municipales en conjunto con el Parque Científico y Tecnológico, y el apoyo de la Universidad de Matanzas, se ejecuta el proyecto Bienestar en el que participan un grupo de empresas pertenecientes al sector de la informática y las telecomunicaciones como XETID (Empresa de Telecomunicaciones e Información para la Defensa), Joven Club, ETECSA (Empresa de Telecomunicaciones de Cuba S.A).

El proyecto de gobierno electrónico Bienestar tiene como objetivo implementar en la provincia de Matanzas de manera organizada, controlada y escalonada plataformas de integración de servicios y gestión de trámites a la población. Bienestar tiene el propósito de ofrecer por parte del gobierno y las instituciones un servicio de mayor calidad a la población matancera y perfeccionar los procesos de retroalimentación entre autoridades y sociedad.

Entre las instituciones que se suman a la iniciativa de gobierno electrónico se encuentra la Dirección Provincial de Cultura de Matanzas. En esta institución se conciben los procesos de gestión de la programación de actividades culturales. Estos procesos se realizan manualmente generando atrasos significativos en la creación de la programación cultural y los informes de cumplimiento de las mismas, la información de las instituciones no llega de forma uniforme a la entidad, limitando el alcance de los diferentes organismos que requieren de dicha información para agilizar sus servicios. Además, para la gestión de la programación, la comunicación entre las instituciones actualmente es escasa o inexistente en muchas entidades. La información que se maneja por la Dirección de Cultura sobre estos procesos no está centralizada ni tiene una estructura correctamente definida haciendo difícil su acceso y la manipulación de la misma, por lo que existe la gran necesidad de mejorar estos procesos y específicamente tener un mejor control, además, brindaría información útil para tomar decisiones y mejoraría los procesos de trabajo, operaciones y administración

Por ello se identificó como problema científico: ¿Cómo facilitar la gestión de los trámites necesarios para la programación cultural en la Dirección Provincial de Cultura de Matanzas?

El objeto de estudio de la investigación es el proceso de gestión de la programación cultural en la Dirección Provincial de Cultura de Matanzas.

Como guía a tener en cuenta para llevar a cabo la investigación se formularon las siguientes preguntas científicas:

1. ¿Qué fundamentos teóricos sustentan la gestión de procesos de la programación cultural de Matanzas?
2. ¿Qué metodología utilizar para el diseño de un sistema informático para la gestión de la programación cultural de Matanzas?
3. ¿Qué pruebas se deben diseñar para validar el sistema informático para la gestión de la programación cultural de Matanzas?

El objetivo general de la investigación es diseñar una aplicación web que mejore la eficiencia en el proceso de gestión de la programación en la Dirección Provincial de Cultura de Matanzas y el campo de acción es el diseño de un sistema informático para la gestión de la programación cultural en la Dirección Provincial de Cultura.

Los objetivos específicos son los siguientes:

1. Realizar un estudio de los fundamentos teóricos-metodológicos del proceso de gestión de la programación cultural de la Dirección Provincial de Cultura de Matanzas.

2. Diseñar un sistema informático para la gestión de la programación cultural, mediante el uso de una metodología de desarrollo de software.
3. Realizar diseños de casos de pruebas para la validación del sistema propuesto.

Para dar cumplimiento al objetivo se han empleado un grupo de métodos teóricos y empíricos de la investigación científica, dentro de los cuales se encuentran:

Del nivel teórico:

1. Análisis y síntesis: Se utilizó para realizar un análisis y una revisión de forma general de toda la bibliografía e información relacionada con el tema, y posteriormente extraer los aspectos fundamentales que sirvieron de apoyo para el desarrollo de la investigación.
2. Lógico-histórico: Se realizó un análisis detallado de los antecedentes, causas y condiciones históricas en que surgió y se desarrolló el objeto y fue utilizado durante la revisión de la literatura.
3. Hipotético-deductivo: Para el análisis y formulación de la hipótesis.

Del nivel empírico:

4. Entrevistas: Fue un método que se utilizó con bastante frecuencia para levantar los requisitos y dar los primeros pasos en el entendimiento del problema.
5. Revisión documental: Se analizaron todos los documentos que fueron facilitados para la investigación.

Con esta investigación se espera como resultado tener un diseño de una aplicación que contribuya de una forma sencilla y confiable a solucionar el problema científico ¿Cómo facilitar la gestión de los trámites necesarios para la programación cultural en la Dirección Provincial de Cultura de Matanzas?, ahorrando tiempo, medios y recursos a la Dirección Provincial de Cultura de Matanzas. Facilitará todos los modelos necesarios de acuerdo con la metodología de desarrollo empleada para la posterior implementación de la solución propuesta.

Estructura del trabajo de diploma:

El presente trabajo está compuesto por una Introducción, dos Capítulos, Conclusiones, Recomendaciones, Referencias Bibliográficas, que, apoyadas en 14 figuras constituyen la totalidad del documento presentado como trabajo de diploma y Anexos. A continuación, se presenta una síntesis de cada uno de los capítulos.

Capítulo 1: “Marco teórico-referencial de la investigación”.

En este capítulo se presenta una exposición detallada de los referentes teóricos que argumentan la propuesta y permiten un acercamiento al objeto de estudio. Además, se comenta sobre el estado del arte, las tendencias y tecnologías actuales que serán usadas para el desarrollo de la herramienta informática propuesta.

Capítulo 2: “Análisis y Diseño de la solución propuesta”.

En este capítulo se argumenta la solución propuesta al problema y se describe la modelación del software a través de la metodología Prodesoft. Se realiza un estudio tanto de factibilidad, como de los beneficios tangibles e intangibles para la realización de la herramienta informática. Se realizan los diseños de las pruebas propuestas a realizar de acuerdo a los requisitos funcionales.

Capítulo 1: Marco teórico referencial de la Investigación.

1.1.- Introducción.

En este capítulo se muestran los temas relacionados con el proceso de gestión de programación cultural en la Dirección Provincial de Cultura de Matanzas. También se muestra de una forma más detallada la definición del objeto de estudio y los antecedentes de la investigación, además de describirse las tecnologías y herramientas que fueron utilizadas en la elaboración del sistema.

1.2. - Antecedentes.

Gobierno Electrónico es el término utilizado para referirse a los procesos y estructuras creadas para la oferta electrónica de los servicios gubernamentales. Consiste en el uso de las tecnologías de la información y el conocimiento en los procesos internos de gobierno y en la entrega de los productos y servicios del Estado tanto a los ciudadanos como a la industria. Muchas de las tecnologías involucradas y sus implementaciones son las mismas o similares a aquellas correspondientes al sector privado del comercio electrónico, mientras que otras son específicas o únicas en relación a las necesidades del gobierno.

La sociedad cubana desde hace algunos años viene avanzando en el tema del Gobierno Electrónico, al respecto el presidente de los Consejos de Estado y de Ministros de la República de Cuba Miguel Díaz-Canel Bermúdez comentó en la clausura de la primera Conferencia Nacional de la Unión de Informáticos de Cuba (UIC) “El proceso de informatización de la sociedad es una prioridad del Estado cubano”. (Antón, 2018)

Es política del país lograr un diálogo transparente entre el gobierno y los ciudadanos. Una manera en la que se potencia es mediante la implementación del gobierno electrónico, que en el mundo ha funcionado como vía de información, facilitación de trámites y comunicación para ambos. Oficialmente se dio por terminada esta primera fase de desarrollo del gobierno electrónico en Cuba. Ya se avanza entonces a la segunda, la de interacción, que permitiría el intercambio entre el gobierno y ciudadanos. (Milanés, 2019)

En nuestro país ya varias provincias avanzan en el tema de gobierno electrónico. Pinar del Río con el primer portal digital de Cuba dirigido a la ciudadanía, consolida su presencia en la web; en Santa Clara se realizan estudios para intensificar hasta qué punto es posible la realización de un programa nacional de Gobierno Electrónico; Granma fue presentada por la empresa Desoft con el programa del Sistema de Gobierno Electrónico que permite simplificar los procesos de soporte institucional y aumentar la participación ciudadana; y en Centro

Habana los ciudadanos pueden compartir comentarios, noticias, quejas o denuncias sobre la vida en el municipio.

Para el avance de la informatización de la sociedad cubana, funciona en la provincia de Matanzas el primer Parque Científico Tecnológico del país. La Universidad de Matanzas acoge la sede del Parque, el cual gestiona el flujo del conocimiento y actúa como puente entre la investigación y el mercado en la transferencia, comercialización de productos y actividad económica sostenible de creación. Frutos de esa integración resulta la Plataforma Bienestar que conecta a los gobiernos locales en la provincia de Matanzas a una sala situacional para gestionar la solución de planteamientos de la población. (Radio Rebelde, 2018)

En el proyecto Bienestar se informatizan un grupo de procesos internos del gobierno, así como los trámites de este y las entidades gubernamentales de cara a la población. Actualmente se realizan al mismo tiempo varias investigaciones con el objetivo de la oferta de gobierno electrónico a varias entidades de la provincia Matanzas. La Dirección Provincial de Cultura se suma a esta iniciativa de gobierno electrónico y al proyecto Bienestar mediante la solución propuesta en este trabajo.

En el caso del gobierno de Matanzas no existe en la actualidad ningún sistema informático que regule y controle el proceso de gestión de la programación en la Dirección Provincial de Cultura, por lo que la informatización de este proceso se percibe como una herramienta más en el cumplimiento de uno de los objetivos de los lineamientos para una descentralización futura de la gestión del gobierno y las instituciones, viéndose reflejada la necesidad de este trabajo en el propósito actual de que las personas puedan realizar sus trámites a través de la web y de propiciar la interacción de los representantes del Poder Popular con la población mediante las tecnologías informáticas.

La Dirección Provincial de Cultura de Matanzas posee en el programa Bienestar como parte del gobierno electrónico un buzón de quejas y sugerencias el cual permite a los ciudadanos interactuar y exponer sus sugerencias a la entidad como única forma de interacción de gobierno electrónico.

1.3.- Análisis de soluciones existentes.

En el mundo, aunque existen múltiples aplicaciones móviles para gestionar la programación cultural, ninguna se posiciona desde el punto de vista de las instituciones que desarrollan y realizan el análisis metodológico de las mismas. Algunas de estas aplicaciones son: Vibra desarrollada en Cataluña para tener toda la oferta cultural en el móvil de forma geo localizada y que tiene como novedad recoger más de 1000 experiencias de diferentes actividades

culturales (Xarxa punt TIC, 2020).Eventbrite es otra app relacionada a la programación de eventos culturales y ofrece una amplia gama de opciones para encontrar el evento de interés para el usuario.

En Cuba hasta la fecha solo se dispone de una aplicación móvil mediante la cual los usuarios pueden acceder a la cartelera de la cultura cubana y donde se encuentran manifestaciones artísticas de todo el país (Cubarte, 2020), sin embargo esta plataforma no ofrece el medio para el análisis metodológico y la gestión de la programación de eventos culturales por la Dirección Provincial de Cultura, además no está incorporada a la iniciativa de gobierno electrónico y no ofrece solución a la problemática planteada.

1.4.- Metodología de desarrollo.

Las Metodologías de Desarrollo de Software son un grupo de procesos cuya finalidad es desarrollar productos o soluciones para un cliente o un mercado en particular, teniendo en cuenta factores como los costes, la planificación, la calidad y las dificultades asociadas. Estas se clasifican en ágiles y pesadas. (OBS Business School, sf)

El desarrollo ágil de software envuelve un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Así el trabajo es realizado mediante la colaboración de equipos inmersos en un proceso compartido de toma de decisiones a corto plazo. (OBS Business School, sf)

El desarrollo de software se enmarca en la rama de la Ingeniería de Software. Prodesoft (Proceso de Desarrollo y Gestión de Proyectos de Software) es una metodología ágil que tiene como objetivo la producción eficiente de un producto de software que satisfaga los requisitos de un especialista funcional con una planificación y una estimación de recursos predecibles. El ciclo de vida de Prodesoft está compuesto por 5 fases ([Anexo1](#)): inicio, modelación, construcción, explotación experimental y despliegue”: (Xetid, 2012)

Durante la fase de Inicio se logra una visión preliminar de la problemática a resolver y se definen los recursos relevantes para la ejecución del proyecto. Es decir, se describen los objetivos y el alcance del proyecto, se identifican los involucrados y ejecutores (entidades involucradas), se estima de manera general las actividades a realizar durante todo el ciclo de desarrollo del proyecto, se establece la estrategia a seguir para realizar la modelación del negocio y la captura de requisitos y de ser necesario se estiman los recursos materiales que deberán ser adquiridos.

En la fase de Modelación se capturan las partes esenciales del sistema, donde se identifican los procesos de negocio fundamentales y se aceptan los requisitos funcionales, obteniéndose la línea base de la arquitectura y una estrategia de construcción de la aplicación aprobada por los implicados en el proyecto. El hito fundamental de esta fase es la liberación de la arquitectura de sistema, datos y despliegue.

En la fase de Construcción se aclaran los requisitos restantes y se completa el desarrollo del sistema sobre una base estable de la arquitectura. Las fases anteriores sólo dieron una arquitectura básica que es aquí refinada de manera incremental conforme se construye el producto. En esta fase todas las características, componentes, y requisitos deben ser integrados, implementados, y probados en su totalidad, obteniendo una versión liberada del producto.

Durante la fase de Explotación Experimental se convierte la versión liberada del producto en una solución estable, donde se eliminan los errores que surgen durante las pruebas y se obtiene una certificación funcional y de seguridad del producto. En la fase de Despliegue se instala y configura el sistema para un ambiente de producción real, se capacita al personal que usará la aplicación y se continúa dando soporte durante la explotación del sistema, culminando de ser preciso con transferencias tecnológicas

1.5.- Tecnologías y Herramientas.

1.5.1.- Herramienta del modelado.

En este trabajo para el modelado se empleó Visual Paradigm. Es una herramienta de software diseñada para que los equipos de desarrollo de software modelen el sistema de información empresarial y administren los procesos de desarrollo. Visual Paradigm es compatible con los lenguajes y estándares clave de modelado de la industria, como el lenguaje de modelado unificado (UML), SoaML, BPMN, XMI, etc. Ofrece un completo conjunto de herramientas que las compañías necesitan para la captura de requisitos, análisis de procesos, diseño de sistemas y diseño de bases de datos. (Visua Paradigm International LTD., sf)

1.5.2.- Lenguajes de Modelado

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado para visualizar, especificar, construir y documentar sistemas de software. No es un lenguaje de programación, pero existen herramientas que se pueden usar para generar código en diversos lenguajes usando los diagramas UML. UML es una combinación de varias notaciones orientadas a objetos: diseño orientado a objetos, técnica de modelado de objetos e ingeniería de software

orientada a objetos y representa buenas prácticas para la construcción y documentación de diferentes aspectos del modelado de sistemas de software y de negocios. Motivo por el cual se utilizó como lenguaje en la modelación del sistema, dígame diagrama de actividad, diagrama de requisitos y negocio, modelo conceptual, modelo de datos y especificación de requisitos. (Pressman, 2014) Se utilizará UML para el modelado de las clases de la solución propuesta.

BPMN: *Business Process Model and Notation* (BPMN) es una notación gráfica que describe la lógica de los pasos de un proceso de Negocio. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades. Una de sus ventajas es que al ser adoptado por la mayoría de las suites BPMN y modeladores de procesos de negocio le proporciona a quienes lo utilizan un lenguaje común para representar los distintos flujos de trabajo (Nextech, 2016) . Esta notación se usará en Process Maker y Visual Paradigm para la modelación del negocio.

Lenguaje de modelado de sistemas (SysML) SysML es un lenguaje de modelado de arquitectura de uso general para aplicaciones de ingeniería de sistemas.

SysML admite la especificación, análisis, diseño, verificación y validación de una amplia gama de sistemas y sistemas de sistemas. Estos sistemas pueden incluir hardware, software, información, procesos, personal e instalaciones. Es un dialecto de UML 2 y se define como un perfil UML 2. (Un perfil UML es un dialecto UML que personaliza el idioma a través de tres mecanismos: estereotipos, valores etiquetados y restricciones). (OMG System Modeling Language, sf)

Se utilizará para la modelación de requisitos funcionales del proyecto.

1.5.3.- Lenguajes de Programación.

PHP (*Personal Home Page*). Es un acrónimo recursivo que significa *Hypertext Pre-processor* (inicialmente *PHP Tools*, o, *Personal Home Page Tools*). Es un lenguaje de programación que se conoce como lenguaje de script o interpretado. Significa que el código que se escribe no se compila, sino que es interpretado por un núcleo o compilador en tiempo de ejecución. Diseñado originalmente para la creación de Páginas web dinámicas con acceso a información almacenada en una base de datos. Además, el código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Es usado principalmente en interpretación del lado del

servidor (*server-side scripting*) (Php, sf) Se utilizará como el lenguaje de programación para el modelado de la solución propuesta.

PhpStorm: es un IDE (entorno de desarrollo integrado) de PHP compatible con PHP 5.3 / 5.4 / 5.5 / 5.6 / 7.0 / 7.1 / 7.2, proporciona prevención de errores sobre la marcha, refactorización de código y autocompletado, depuración de configuración cero y un editor extendido de HTML, CSS y JavaScript. (Jetbrains, sf)

El IDE se utilizará como editor de código para la implementación de plugins.

Las Hojas de Estilo en Cascada (CSS, por sus siglas en inglés de *Cascading Style Sheets*), es un lenguaje de marcado que se emplea para dar formato a un sitio web CSS3 y sirve para cambiar desde las medidas para los márgenes hasta las especificaciones para las imágenes y el texto, todo esto mediante módulos. Entre sus ventajas se encuentra el ser gratuito, tener todos los estilos en una sola página y generar sitios más rápidos, además de brindarnos la posibilidad de añadir animaciones sin recurrir a otros lenguajes como JavaScript. Se puso de manifiesto en los estilos de las interfaces en *Process Maker*. (TechTerms, 2018) Se utilizará CSS para dar formato a las vistas de la solución propuesta.

Java Script es un lenguaje de programación que te permite realizar actividades complejas en una página web como mostrar actualizaciones de contenido en el momento, interactuar con mapas, animaciones gráficas 2D/3D etc. Es la tercera capa del pastel de los estándares en las tecnologías para la web, dos de las cuales son (HTML y CSS). Además, permite crear contenido nuevo y dinámico, controlar archivos de multimedia y crear imágenes animadas. La utilización de dicho lenguaje está presente en las validaciones de los componentes de las interfaces del sistema. (Mozilla, sf) Se empleará como lenguaje de programación de la validación de la solución propuesta.

Hyper Text Markup Language (HTML) es un lenguaje usado para estructurar y presentar el contenido para la web. Se trata de un sistema para formatear el *layout* de nuestras páginas, así como hacer algunos ajustes a su aspecto. Una de las más importantes novedades está relacionada con la inserción de multimedia en los sitios web, que ahora contarán con etiquetas HTML especiales para poder ser incluidos. Por otro lado, algunos aspectos de diseño también son incluidos en el lenguaje, así como también algunos detalles de navegación. Además, se reduce la dependencia de *plug-ins*. (Mozilla, sf)

En este trabajo se empleará HTML5 para la modelación de las vistas de la solución propuesta.

JQuery: es una librería de código abierto escrita en JavaScript, te permite amaestrar el poder de JavaScript para cumplir una multitud de cosas increíbles en tu página web. Con JQuery, puedes añadir, borrar o modificar elementos HTML en tu página, cambiar el estilo de elementos en tu página modificando el CSS asociado, animar elementos en tu página además de enviar y recibir datos de un servidor vía AJAX (*asynchronous* JavaScript and XML) de modo que tu página no tenga que cargarse de nuevo tras enviar un formulario. También se disfruta de mejor compatibilidad entre navegadores. Permite mediante el código JavaScript manejar de forma amigable las validaciones del sistema. En resumen, es una librería de JavaScript rápida, pequeña y con muchas funciones. (Quora, 2017) Se utilizará para dar estilo a las vistas de la solución propuesta.

1.5.4.- Lenguajes de Consulta.

El Lenguaje Estructurado de Consultas (SQL, por sus siglas en inglés de *Strutured Query Language*) es un lenguaje estándar e interactivo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. (QuickStart, 2018)

Se utilizará para las consultas a la base de datos.

DQL (Doctrine Query Language): es el lenguaje de consulta propio de Doctrine, el cual proporciona potentes capacidades de consulta sobre el modelo de objetos, imaginando todos los objetos dispersos en algún almacenamiento (como un objeto de base). (Doctrine, sf) Se utilizará en las consultas de bases de datos.

1.5.5.- Gestor de Base de Datos.

Un Gestor de Bases de Datos (GBD) es un sistema formado por un conjunto de datos y un paquete de software para la gestión del mismo, de modo que se controla el almacenamiento de datos redundantes, los datos resultan independientes de los programas que los usan, se almacenan las relaciones entre los datos junto con éstos y se puede acceder a los datos de diversas formas

Un Sistema Gestor de Bases de Datos (SGBD) o DGBA (*Data Base Management System*) es un conjunto de programas no visibles que administran y gestionan la información que contiene una base de datos. Se le conoce también como gestor de datos y, a través de él, se maneja todo acceso a la base de datos con el objetivo de servir de interfaz entre ésta, el usuario y las aplicaciones. (Guru99, 2020)

PostgreSQL puede ser ejecutado en los principales sistemas operativos y este soporta textos, imágenes, sonidos y video. Una de sus principales características en la actualidad es

su portabilidad y también el ser gratuito y libre. PostgreSQL es altamente extensible y se ejecuta en los principales sistemas operativos (Postgresql, sf). Se utilizará para el sistema de gestión de base de datos de la solución propuesta.

pgAdmin es la herramienta oficial para administrar nuestras bases de datos en PostgreSQL. Nos permite, desde hacer búsquedas SQL, hasta desarrollar toda nuestra base de datos de forma muy fácil e intuitiva; directamente desde la interfaz gráfica. Con pgAdmin crear una nueva base de datos es muy sencillo. Tanto principiantes como expertos se sienten cómodos con el sistema. Puedes crear respaldos, restaurar la base de datos o ejecutar tareas de mantenimiento de forma muy sencilla desde pgAdmin. (pgAdmin, 2016)

Se empleará para administrar la base de datos de la solución propuesta.

1.5.6.- Framework.

ProcessMaker es un Sistema de Gestión de Procesos de Negocio (BPM). Esta aplicación es una solución de software con base en flujos de trabajo; de código abierto. ProcessMaker ayuda a las organizaciones de todos los tamaños para diseñar fácilmente, automatizar e implementar los procesos del negocio. Permite a los usuarios de negocio crear formas y mapas de flujos de trabajo completamente funcionales. Facilita la coordinación del flujo de trabajo entre los usuarios, departamentos y organizaciones. ProcessMaker es ligero, extremadamente eficiente, e implica los gastos generales más bajos de cualquier BPM en la industria. Los usuarios del negocio pueden completar fácilmente procesos basados en formularios mediante notificaciones automatizadas y una interfaz basada en la web. Cualquier usuario puede automatizar un proceso con flujo de trabajo que tienen opciones de “arrastrar y soltar” y con configuraciones apuntar y hacer clic. Usado en la implementación de la propuesta de solución de la situación problemática. (Software Advice) Se utilizará como framework para la modelación de la solución propuesta.

Zend Framework es un framework de PHP disponible como Software Libre y Código Abierto que está totalmente orientado a objetos, y por ello usa muchos conceptos de la orientación a objetos como son herencia e interfaces, permitiendo al desarrollador implementar sus propias variantes en los componentes sin tener que modificar el código base. Se usa para la construcción de proyectos plenamente funcionales, pero hay ocasiones en que solo se necesita cierto componente. Es bastante desacoplado, esto significa que permite el uso de cualquiera de sus librerías en vez de usarlo totalmente, permite usar cualquiera de sus componentes para realizar diferentes tipos de funcionalidades que se puedan necesitar

en una aplicación. (Zend Framework, sf) Se utilizará como framework para la modelación de la solución propuesta.

Doctrine es un ORM (Object Relation Mapper), una técnica de programación que nos permite convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, es decir, las tablas de nuestra base de datos pasan a ser clases y los registros objetos que podemos manejar con facilidad. (Doctrine, sf) Se empleará como ORM de la base de datos de la solución propuesta.

Utilizar un ORM tiene una serie de ventajas que nos facilitan enormemente tareas comunes y de mantenimiento: Reutilización, encapsulación, portabilidad, seguridad y mantenimiento del código.

ExtJs es una librería. Cada objeto contiene lo típico: propiedades, métodos y eventos. Basa toda su funcionalidad en JavaScript a través de librerías. Así, en tiempo de ejecución carga y crea todos los objetos HTML a través del uso intenso de DOM. Los datos son obtenidos con AJAX a través de XML. Una de las grandes ventajas de utilizar ExtJS es que permite crear aplicaciones complejas utilizando componentes predefinidos, así como un manejador de diseños similar al que provee Java Swing. Existe un balance entre Cliente–Servidor. La carga de procesamiento se distribuye, permitiendo que el servidor al tener menor carga, pueda manejar más clientes al mismo tiempo. (Tutorials Point, sf) Se utilizará para la validación de la solución propuesta.

Bootstrap es un framework desarrollado y liberado por Twitter que tiene como objetivo facilitar el diseño web. Sus webs, son adaptables a cualquier dispositivo y tamaño de pantalla y siempre se van a ver bien. Además, permite que la carga sea rápida y que la navegación sea fluida e intuitiva. Permite utilizar elementos de HTML y CSS. Se integra bien con las librerías de JavaScript y permite modelar por columnas que son muy configurables y fáciles. (Developers Toptal, 2016) Se utilizará para la validación de la solución propuesta.

1.5.7.- Servicios Web

XML es un lenguaje que se utiliza para decir algo acerca de otro (metalenguaje) aunque no se considere un lenguaje en sí mismo sino un sistema que permite definir lenguajes de acuerdo a las necesidades, el cual tiene como principal función la organización y el etiquetado de documentos. Aparece como un estándar que estructura el intercambio de información entre las diferentes plataformas. Entre sus ventajas está el ser extensible, o sea, poder añadir nuevas etiquetas tras el diseño del documento. Otra ventaja es permitir relacionar aplicaciones de diferentes lenguajes y plataformas, aunque esto puede ser visto como un

obstáculo para la búsqueda de compatibilidad. Empleado para el tratamiento de errores en la vista del plugins. (How-to Geek, 2018) Se utilizará para la validación de la solución propuesta.

WSDL es un lenguaje de comprensión de servicios web. WSDL brinda un formato estándar que permite saber qué información puede gestionar un servicio sin la necesidad de un documento de procesamiento de texto (.txt) lo que elimina la ambigüedad y facilita que esta información pueda ser leída por humanos y por máquinas. (W3C Note, 2002) Se utilizará para la validación de la solución propuesta.

1.6.- Conclusiones del Capítulo

En este capítulo fue analizado el objeto de estudio, los antecedentes del gobierno electrónico en el mundo y en Cuba, así como las herramientas, tecnologías y metodología de desarrollo propuesta. El desarrollo del capítulo dio a conocer las bases teóricas sobre las cuales se sustenta la propuesta de esta investigación. Se analizó como se efectúa el proceso de gestión de la programación cultural por lo que se puede afirmar que no hay evidencia en Cuba de la existencia de un sistema que responda a las necesidades detectadas en la investigación. Es una prioridad del país la implementación de sistemas que soporten la estrategia de gobierno electrónico.

Capítulo 2: Análisis y diseño de la solución propuesta.

2.1.- Introducción.

En este capítulo se muestra el diseño de la solución propuesta, que consta del modelo de negocio desglosado en modelo conceptual, reglas de negocio y requisitos funcionales. Además, se evidencia la arquitectura a usar, patrones y mecanismos de diseño, diseño de la base de datos, diagramas de clases y de secuencias, seguridad y análisis de factibilidad. También se diseñan las pruebas que se les realizarán al software para comprobar el correcto funcionamiento del mismo y de detectar la mayor cantidad de errores posibles en la implementación de los requisitos funcionales, además de asegurar que estos errores encontrados sean corregidos

2.2.- Modelado del negocio.

2.2.1.- Descripción del negocio.

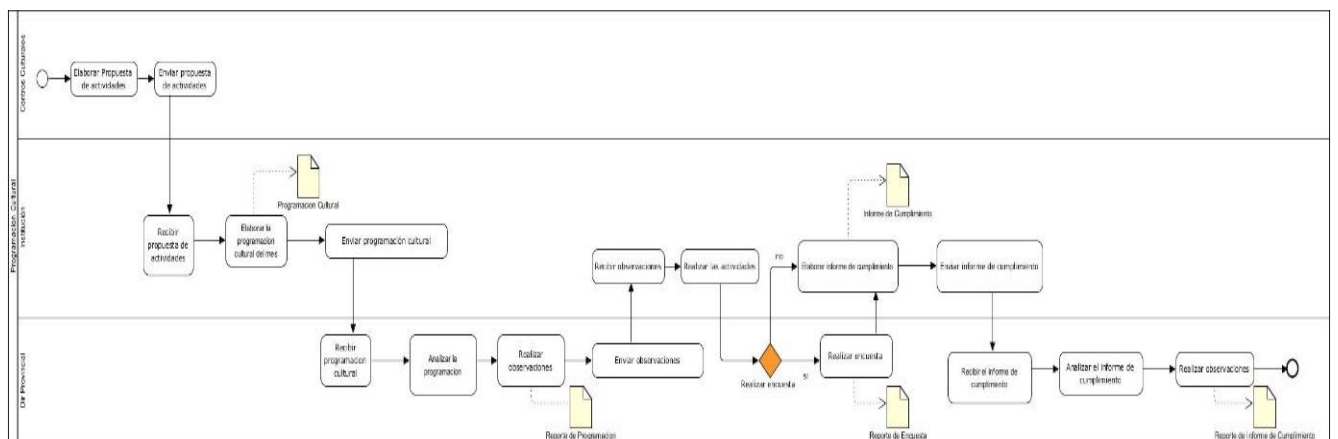


FIG. 1 DIAGRAMA DEL PROCESO DEL NEGOCIO

El proceso de Gestión de la Programación Cultural comienza cuando las instituciones culturales elaboran la programación cultural del mes, esta programación no debe sufrir cambios una vez elaborada y se envía a la Dirección Provincial de Cultura para el análisis metodológico de la misma. Cultura Provincial realiza encuestas sobre las actividades realizadas a medida que se vayan cumpliendo. Las instituciones deben elaborar también al final de cada mes, una vez realizadas las actividades, un informe de cumplimiento y enviarlo a la Dirección Provincial de Cultura donde se evalúa el cumplimiento de las actividades y el desarrollo de los programas especiales y se realizan las observaciones pertinentes a las instituciones sobre la elaboración de este documento.

2.2.2.- Reglas del negocio.

Es la declaración que rige el funcionamiento de algún aspecto del negocio. Una política que debe cumplirse y una condición a satisfacer. Son definidas y revisadas por el grupo de proyectos y por los usuarios y clientes.

En esta investigación se aprecian las siguientes reglas:

- R1. Las instituciones son las encargadas de realizar su programación provincial.
- R2. La programación cultural no puede sufrir variaciones una vez realizada.
- R3. La Dirección Provincial de Cultura realiza las observaciones a la programación cultural.
- R4. Las instituciones realizan el informe de cumplimiento de las actividades.
- R5. La Dirección Provincial de Cultura realiza las observaciones pertinentes al informe de cumplimiento.
- R6. Los usuarios solo pueden realizar las tareas asignadas.

2.2.4.- Modelo conceptual.

El modelado conceptual es una técnica de análisis de requisitos y de diseños de bases de datos. Un modelo conceptual es una representación de un sistema, hecho de la composición de conceptos que se utilizan para ayudar a las personas a conocer, comprender o simular un tema que representa el modelo. El modelo conceptual ayuda a identificar problemas en los requisitos antes de comenzar el desarrollo, evitando gastos innecesarios. (TechTerms, 2018)

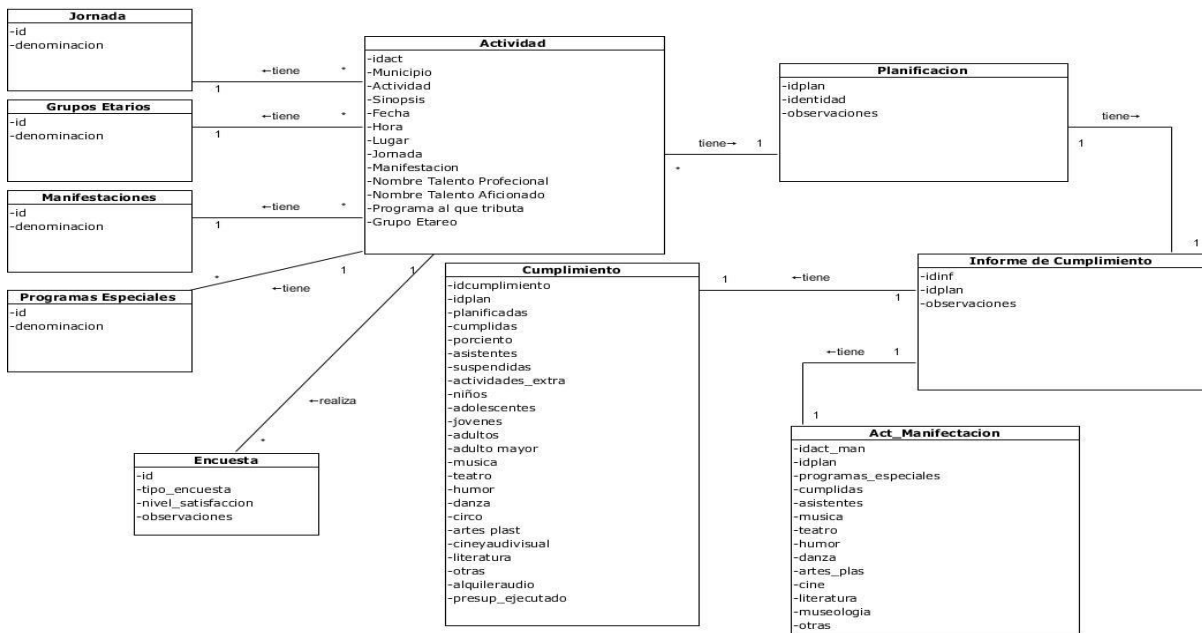


FIG. 2 MODELO CONCEPTUAL.

2.3.- Requisitos del sistema informático.

2.3.1.- Requisitos No Funcionales.

Los Requisitos No Funcionales establecidos para el sistema informático son los siguientes:

- **Usabilidad (USB)** El sistema debe ser fácil de utilizar para los usuarios que tengan niveles básicos de computación o hayan trabajado con la Web. Debe tener una opción de ayuda sobre las principales funcionalidades que brinda el sistema y sus íconos respectivos, para un mejor entendimiento. Las operaciones de la aplicación a informatizar serán lo más parecidas posible a los procesos que se realizan actualmente, para así lograr el menor tiempo en cuanto a la comprensión y adaptación del sistema.
- **Confiabilidad (CON)** Deben establecerse los mecanismos necesarios para el restablecimiento del sistema ante fallos de comunicación u otros, los tiempos mínimos para ello no deben exceder las 6hrs. Deben montarse sistemas de respaldo eléctrico en los locales de los servidores para mantener la vitalidad de los servicios.
- **Rendimiento (REN)** La aplicación debe estar concebida para el consumo mínimo de recursos. Un total de 350-400 usuarios conectados de forma simultánea al servidor central en cualquier momento. Los tiempos de respuesta y velocidad de procesamiento de la información serán rápidos, no mayores de 5 segundos para las actualizaciones y 20 para las recuperaciones.

- Soporte (SOP) Se necesita un servidor de bases de datos que soporte grandes volúmenes de datos. Debe elaborarse un paquete de instalación que abarque verificación de componentes ya instalados y la instalación de los nuevos.

- Requerimiento de Ayuda y Documentación (ADO) Se propone que el sistema cuente con una ayuda general en la página principal, que guiará al usuario de cómo trabajar en el sistema, también estará disponible en cada una de las interfaces, de esta forma los usuarios tendrán conocimiento de las funcionalidades del mismo y hacer un mejor uso de estas.

- Interfaz (INU) La interfaz de la aplicación a desarrollar debe ser sencilla para reducir el tiempo de capacitación de los usuarios. Además, por el uso diario y constante que tendrá el software, la interfaz debe ser agradable, que favorezca el estado de ánimo del cliente y que combine correctamente los colores, tipo de letra, tamaño y que los iconos estén en correspondencia con lo que representan.

- Portabilidad (POR) El sistema será multiplataforma (Linux y Windows fundamentalmente).

- Políticos Culturales (CUL) El producto no debe contener palabras en otros idiomas y debe respetar los términos empleados por los especialistas en el tema de la esfera que se automatiza.

- Seguridad (SEG) Los usuarios debe autenticarse antes de entrar al sistema. Debe garantizar el acceso controlado a la información. Se presentarán las interfaces para cada usuario dependiendo del nivel de acceso a la información. La información que se maneje en el sistema estará protegida de acceso no autorizado y divulgación, a partir de los diferentes roles de los usuarios que empleen el sistema. La información existente en el sistema será protegida contra actos ilícitos, de igual manera el origen y fuente de los datos.

2.3.2.- Requisitos Funcionales.

Los Requisitos Funcionales (RF) registrados para este sistema informático son los siguientes:

- 1 Gestionar jornada
 - 1.1 Adicionar jornada
 - 1.2 Modificar jornada
 - 1.3 Eliminar jornada

- 2 Gestionar programas especiales
 - 2.1 Adicionar programas especiales
 - 2.2 Modificar programas especiales
 - 2.3 Eliminar programas especiales
- 3 Gestionar manifestaciones
 - 3.1 Adicionar manifestaciones
 - 3.2 Modificar manifestaciones
 - 3.3 Eliminar manifestaciones
- 4 Gestionar grupo etario
 - 4.1 Adicionar grupo
 - 4.2 Modificar grupo
 - 4.3 Eliminar grupo
 - 4.4 Asociar personas a grupo
- 5 Gestionar planificación
 - 5.1 Adicionar actividad
 - 5.2 Modificar actividad
 - 5.3 Buscar actividad
 - 5.4 Eliminar actividad
- 6 Gestionar informe de cumplimiento
 - 6.1 Adicionar informe de cumplimiento
 - 6.2 Modificar informe de cumplimiento
 - 6.3 Buscar informe de cumplimiento
- 7 Gestionar Encuesta

7.1 Adicionar encuesta

7.2 Buscar encuesta

Reportes y Recuperaciones

8. Listado de las actividades de la programación cultural

9. Listado del informe de cumplimiento

10. Listado de encuestas

Alertas y notificaciones

Notificar al recibir la programación.

Notificar al recibir el informe

2.3.3.- Especificación de los requisitos funcionales.

A continuación, se mostrarán los diagramas de actividad de algunos de los principales requisitos funcionales.

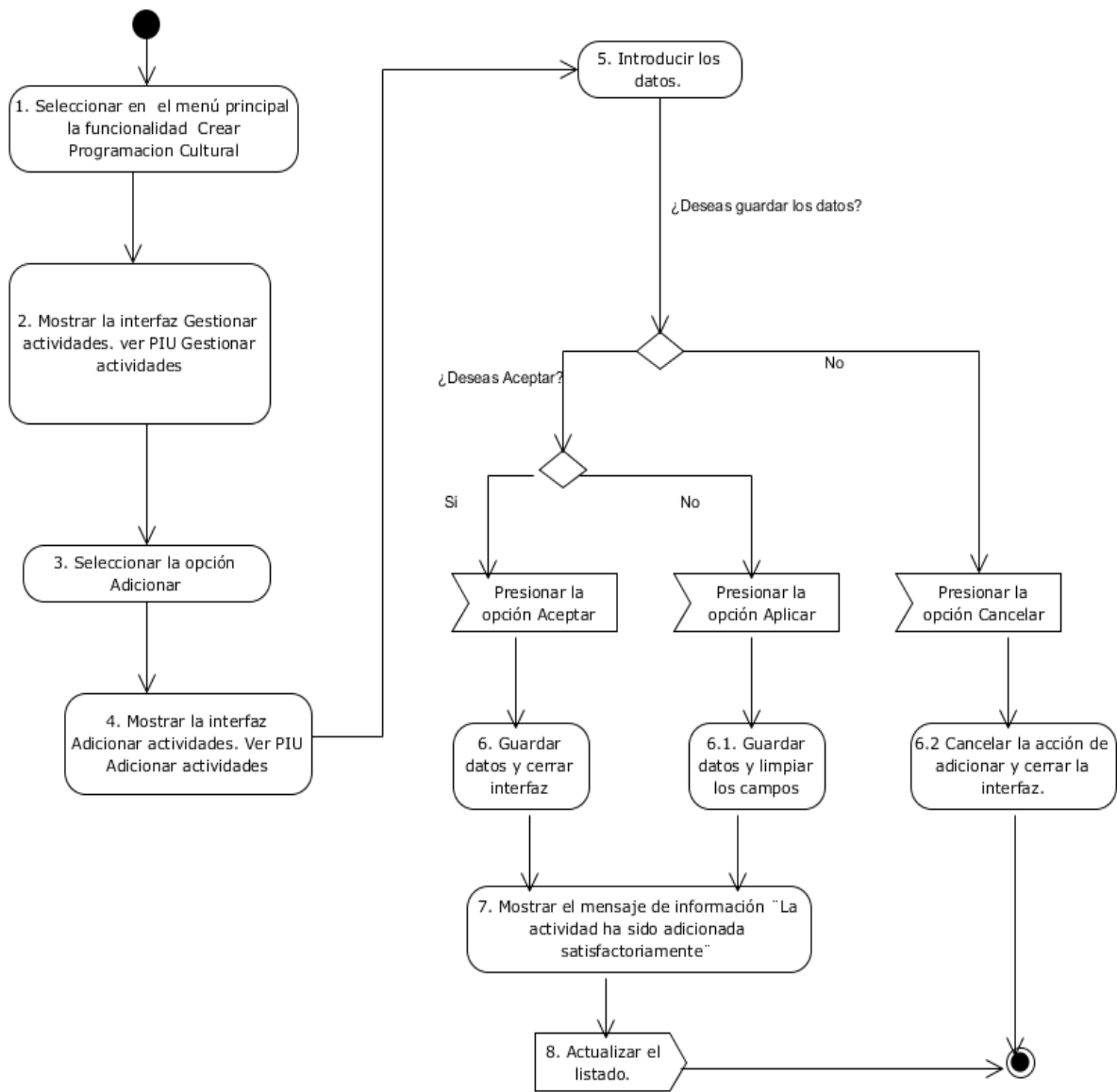


FIG. 3 DIAGRAMA DE ACTIVIDAD DEL REQUISITO FUNCIONAL GESTIONAR PLANIFICACIÓN.

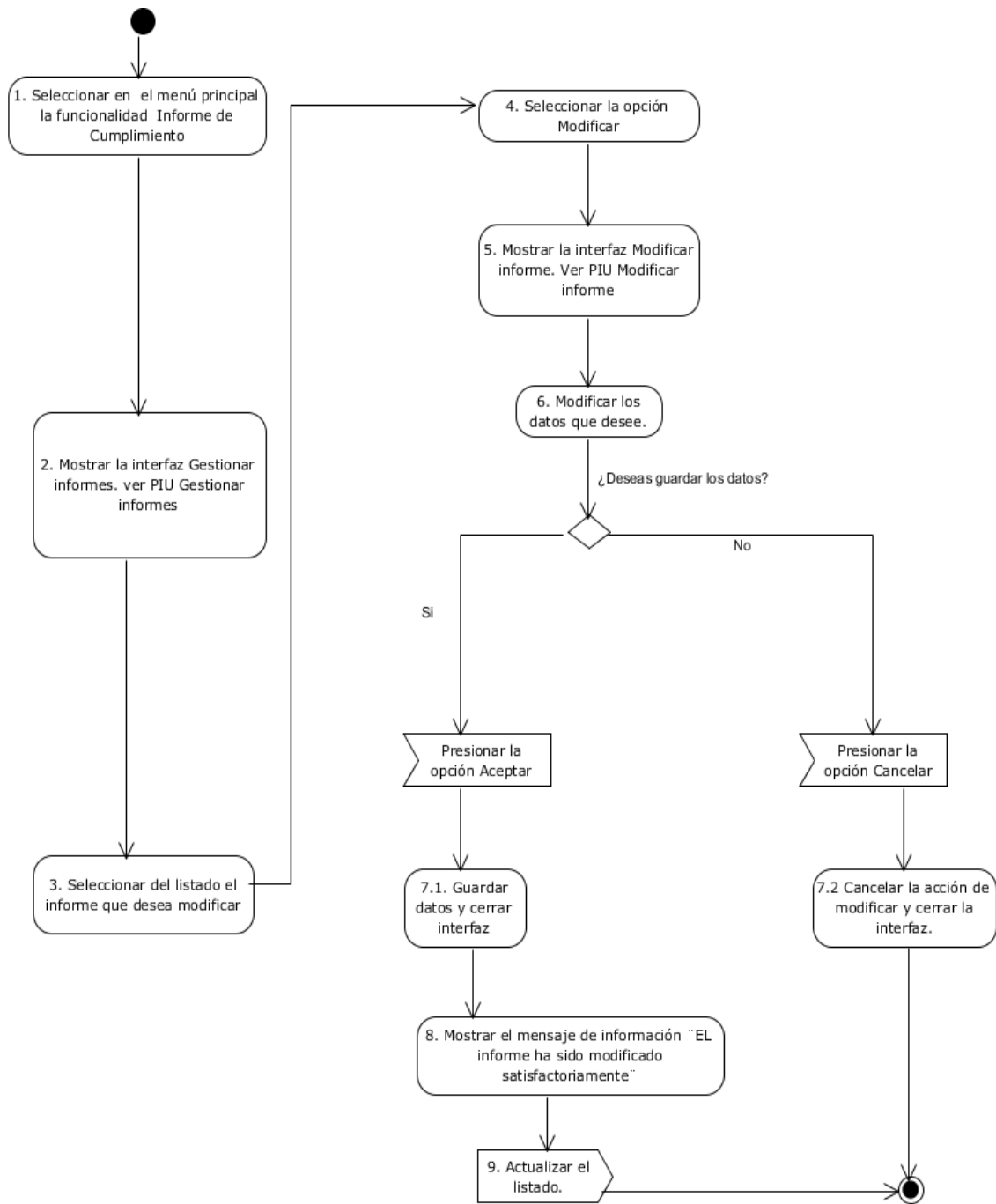


FIG. 4 DIAGRAMA DE ACTIVIDAD DEL REQUISITO FUNCIONAL GESTIONAR INFORMES DE CUMPLIMIENTO.

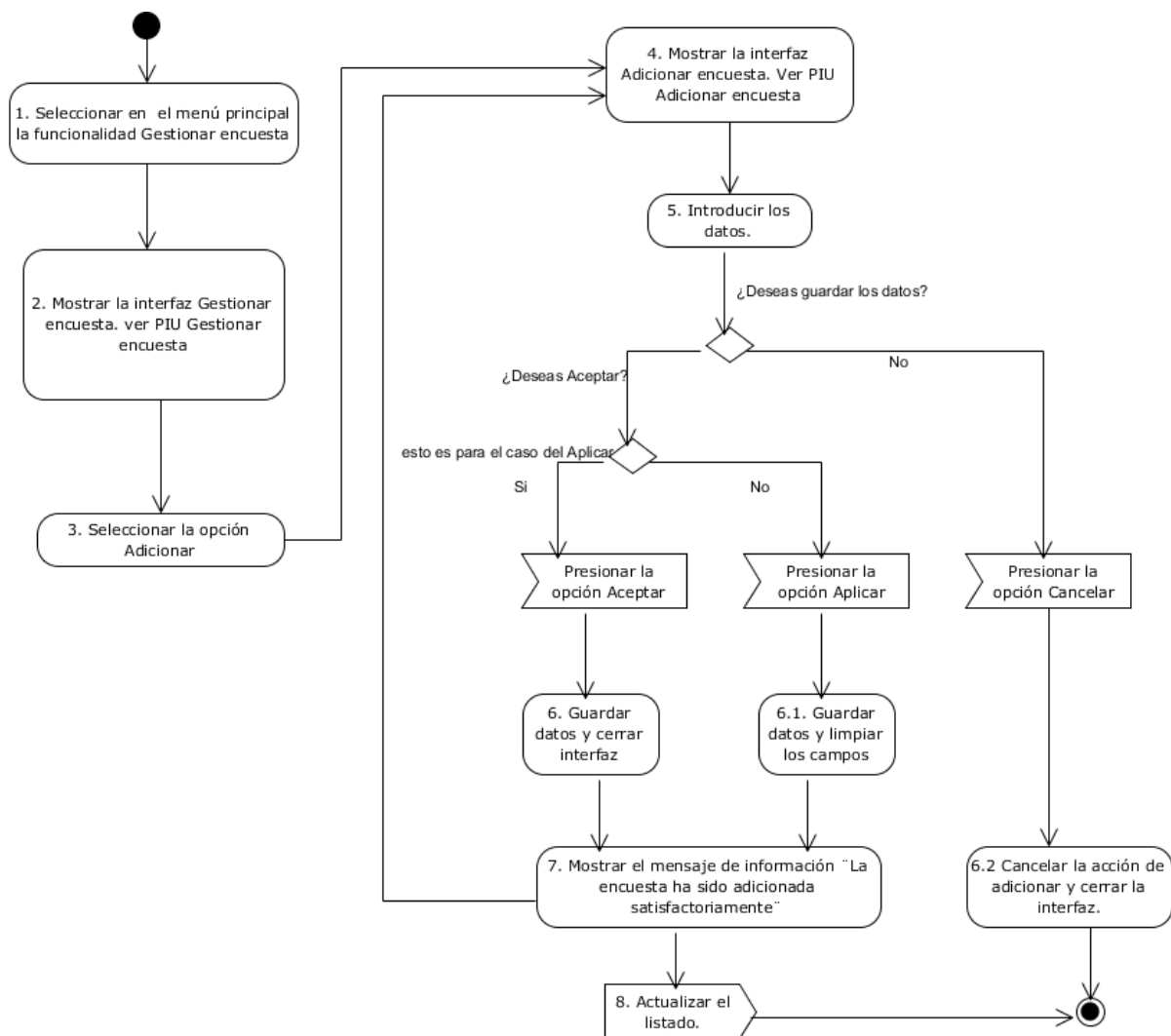


FIG. 5 DIAGRAMA DE ACTIVIDAD DEL REQUISITO FUNCIONAL GESTIONAR ENCUESTAS.

2.4 Implementación de la Arquitectura de Software

La arquitectura de un sistema describe sus componentes principales, sus relaciones (estructuras) y cómo interactúan entre sí. La arquitectura y el diseño del software incluyen varios factores que, como la estrategia comercial, los atributos de calidad, la dinámica humana, el diseño y el entorno de TI, contribuyen a la realización de un mejor producto. Podemos segregar la arquitectura y el diseño de software en dos fases distintas: arquitectura de software y diseño de software. En Arquitectura, las decisiones no funcionales son emitidas y separadas por los requisitos funcionales. En diseño, se cumplen los requisitos funcionales. (Tutorialspoint, 2020)

La arquitectura utilizada por la empresa XETID es Cliente/Servidor en la que se reparten las tareas entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Las aplicaciones Clientes realizan peticiones a una o varias aplicaciones Servidores, que deben encontrarse en ejecución para atender dichas demandas. El modelo Cliente/Servidor permite diversificar el trabajo que realiza cada aplicación, de forma que los Clientes no se sobrecarguen. En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. Tanto el Cliente como el Servidor son entidades abstractas que pueden residir en la misma máquina o en máquinas diferentes.

2.4.1 Arquitectura N-Capas.

La arquitectura N-Capas es un concepto de arquitectura cliente-servidor en ingeniería de software donde las funciones de presentación, procesamiento y gestión de datos están separadas lógicamente y físicamente. Estas funciones se ejecutan en una máquina separada o en clústeres separados para que cada uno pueda proporcionar los servicios a su máxima capacidad ya que no se comparten recursos. Esta separación facilita la administración de cada uno por separado, ya que trabajar en uno no afecta a los demás, aislando los problemas que puedan ocurrir.

La arquitectura de N niveles generalmente divide una aplicación en tres niveles: el nivel de presentación, el nivel lógico y el nivel de datos. Es la separación física de las diferentes partes de la aplicación en oposición a la separación generalmente conceptual o lógica de los elementos en el marco del modelo-vista-controlador (MVC). Otra diferencia del marco MVC es que las capas de n niveles están conectadas linealmente, lo que significa que toda comunicación debe pasar por la capa intermedia, que es el nivel lógico. En MVC, no hay una capa intermedia real porque la interacción es triangular; la capa de control tiene acceso tanto a la vista como a las capas del modelo y el modelo también accede a la vista; el controlador también crea un modelo basado en los requisitos y lo empuja a la vista. Sin embargo, no son mutuamente excluyentes, ya que el marco MVC se puede usar junto con la arquitectura de n niveles, siendo el nivel n la arquitectura general utilizada y MVC como marco para el nivel de presentación. (Technopedia, 2020)

Las ventajas de la arquitectura de n niveles incluyen:

- Escalable: escale niveles separados sin tocar otros niveles

- Gestión individual: evita los efectos en cascada; aísla el mantenimiento
- Flexible: se expande de cualquier manera según los requisitos
- Seguro: cada nivel se puede proteger por separado y de diferentes maneras

2.5 Patrones Utilizados.

Singleton: El patrón Singleton garantiza que una clase tenga solo una instancia y proporciona un punto de acceso global a esa instancia. Es de tipo creacional. Lleva el nombre del conjunto singleton, que se define como un conjunto que contiene un elemento. (Source Making, sf)

Abstract Factory (Fábrica Abstracta): Permite trabajar con objetos de distintas familias de manera que estas no se mezclen entre sí y haciendo transparente el tipo de familia concreta que se esté usando. El problema a solucionar por este patrón es el de crear diferentes familias de objetos, como por ejemplo la creación de interfaces gráficas de distintos tipos (ventana, menú, botón, etc.). (Source Making, sf)

Facade(Fachada): Provee de una interfaz unificada simple para acceder a una interfaz o grupo de interfaces de un subsistema.

Comando: Este patrón permite solicitar una operación a un objeto sin conocer realmente el contenido de esta operación, ni el receptor real de la misma. Para ello se encapsula la petición como un objeto, con lo que además se facilita la parametrización de los métodos.

Modelo Vista Controlador: Es un patrón de diseño arquitectónico de software, que sirve para clasificar la información, la lógica del sistema y la interfaz que se le presenta al usuario. En este tipo de arquitectura existe un sistema central o controlador que gestiona las entradas y la salida del sistema, uno o varios modelos que se encargan de buscar los datos e información necesaria y una interfaz que muestra los resultados al usuario final. Es muy usado en el desarrollo web porque al tener que interactuar varios lenguajes para crear un sitio es muy fácil generar confusión entre cada componente si estos no son separados de la forma adecuada. Este patrón permite modificar cada uno de sus componentes si necesidad de afectar a los demás.

2.6 Mecanismos de Diseño.

El mecanismo de diseño resume cómo construir el diseño de las clases que se pueden implementar en el software, se visualizan las relaciones entre ellas y se muestra gráficamente la interacción de los objetos para comunicarse entre sí.

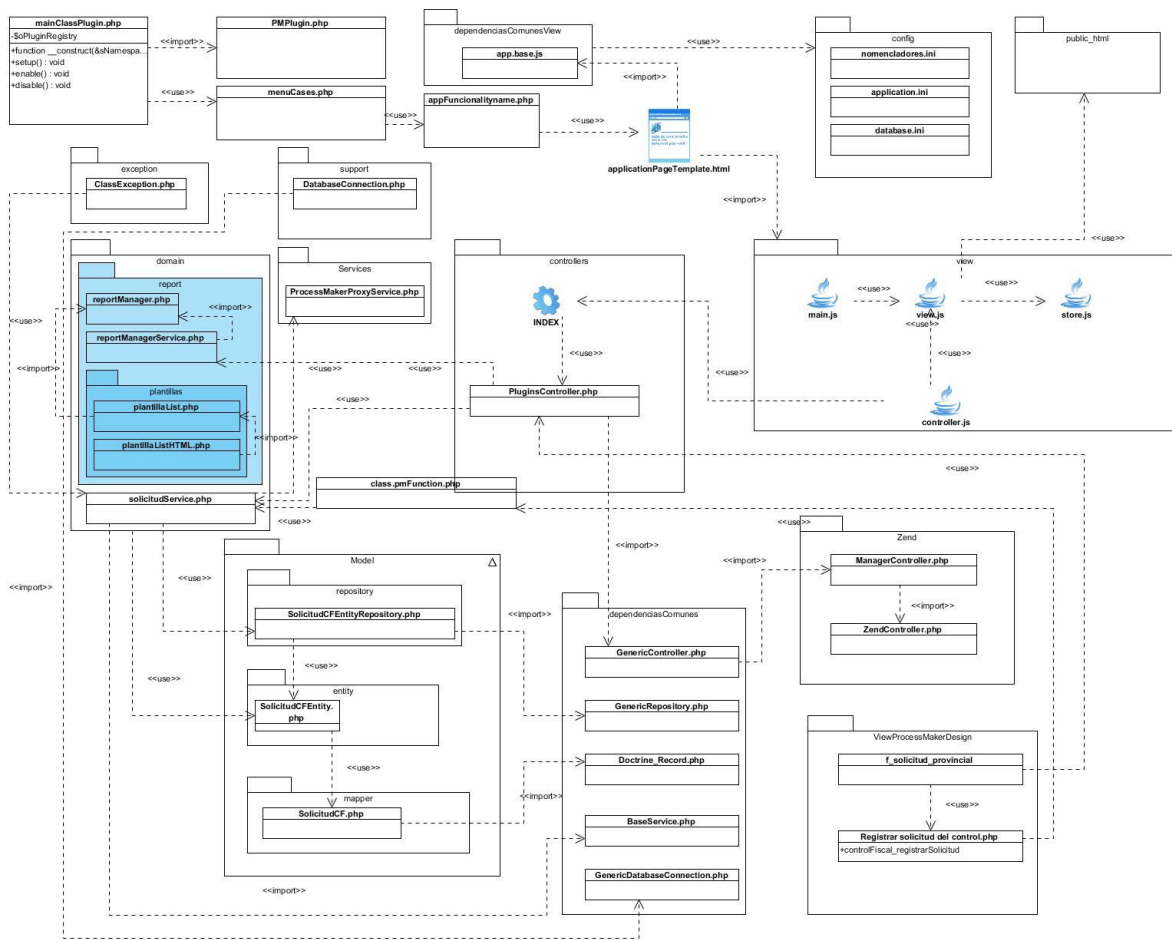


FIG. 6 DIAGRAMA DE MECANISMO DE DISEÑO

2.7 Diseño de la Base de Datos.

El diseño de base de datos es un proceso cuyo objetivo de definir la estructura adecuada para nuestro sistema de información. Una base de datos correctamente diseñada proporciona acceso a información actualizada y precisa, nos permite ahorrar espacio, ayuda a que se preserve la precisión e integridad de los datos y que no se pierda información, y agiliza el acceso y el procesamiento de los datos (Microsoft, 2016)

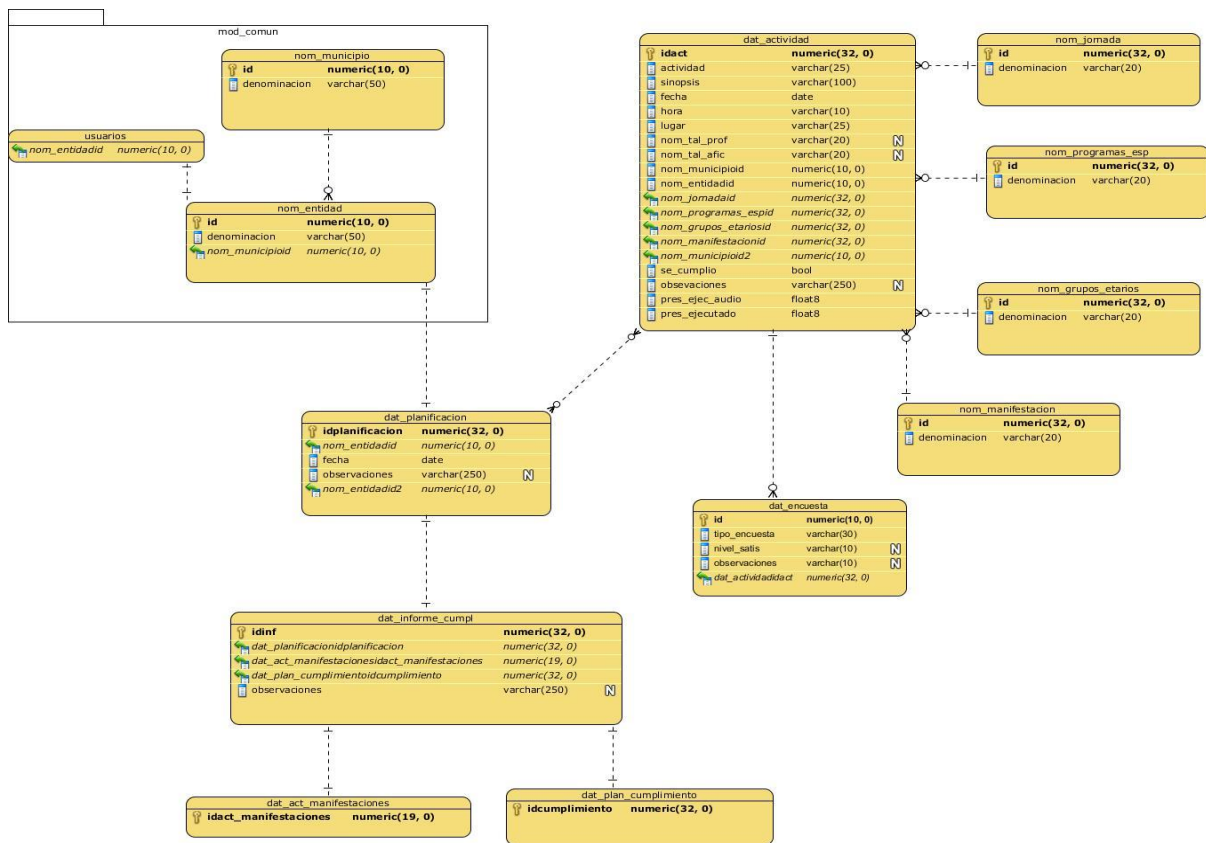


FIG. 7 DIAGRAMA DEL MODELO LÓGICO DE BASE DE DATOS.

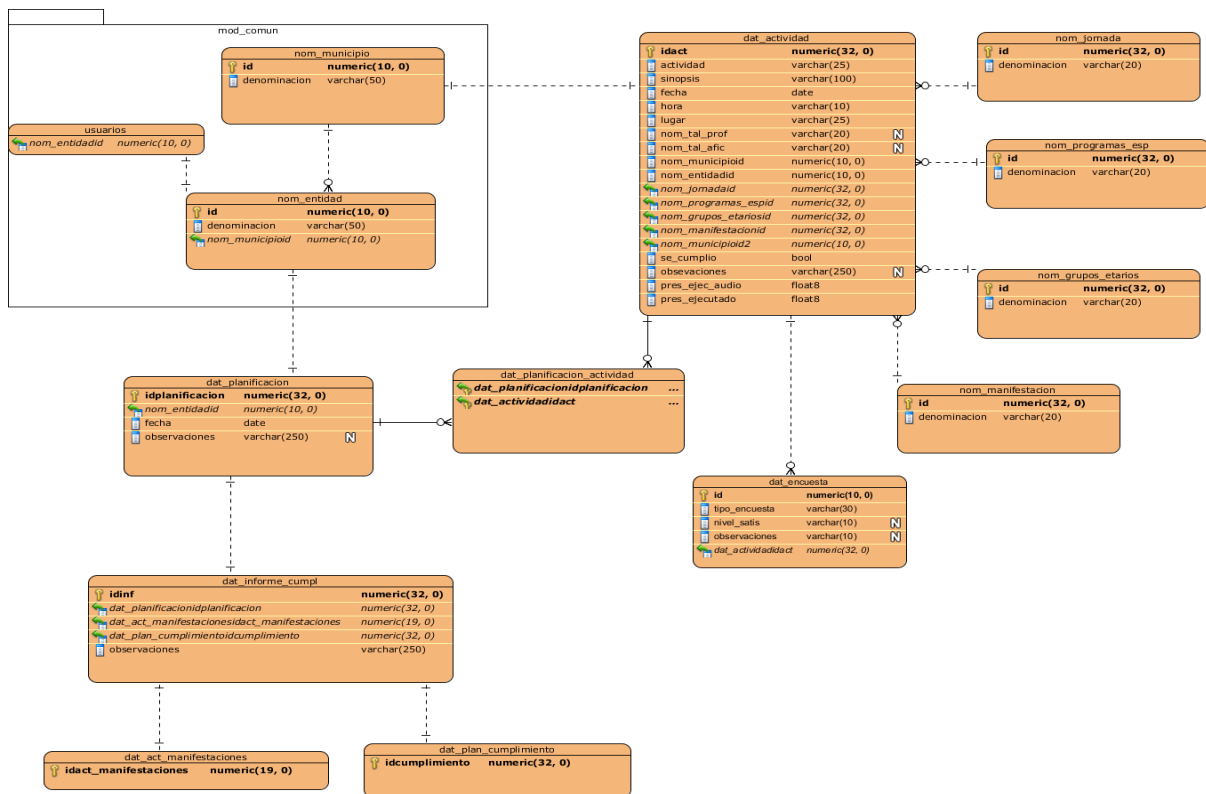


FIG. 8 DIAGRAMA DEL MODELO FÍSICO DE BASE DE DATOS.

2.8 Diagramas de Clases de Diseño.

Los diagramas de clases de diseño correspondientes a los requisitos funcionales: Gestionar Manifestaciones, Gestionar Jornadas, Gestionar Programas Especiales, Gestionar Grupos Etarios, se encuentran el documento Anexos de Evidencias.

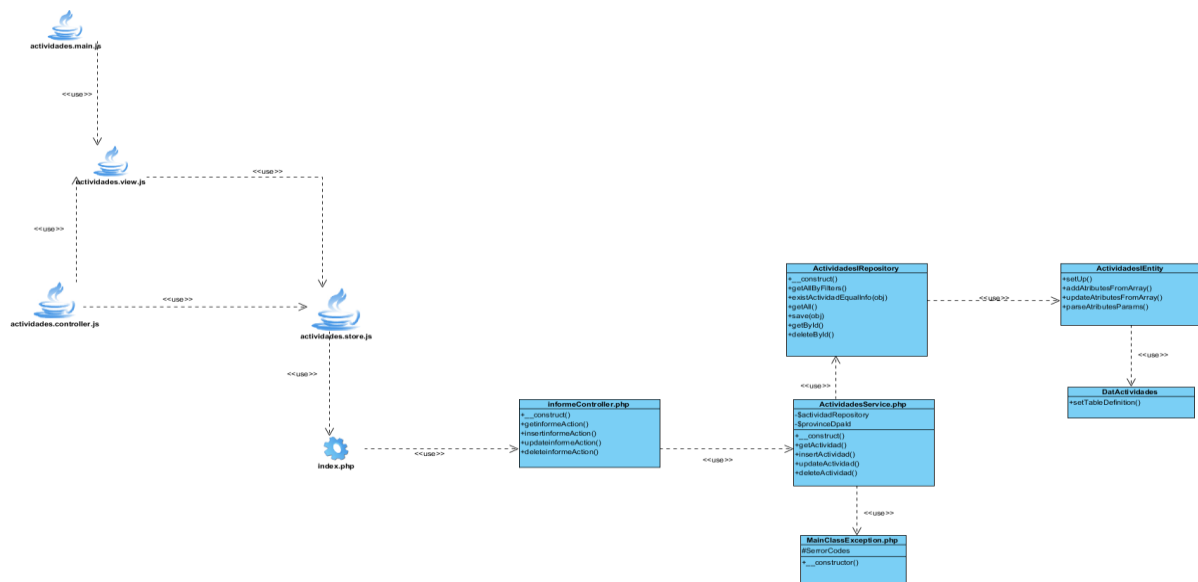


FIG. 9 DIAGRAMA DE CLASES DE DISEÑO DEL REQUISITO FUNCIONAL GESTIONAR PLANIFICACIÓN.

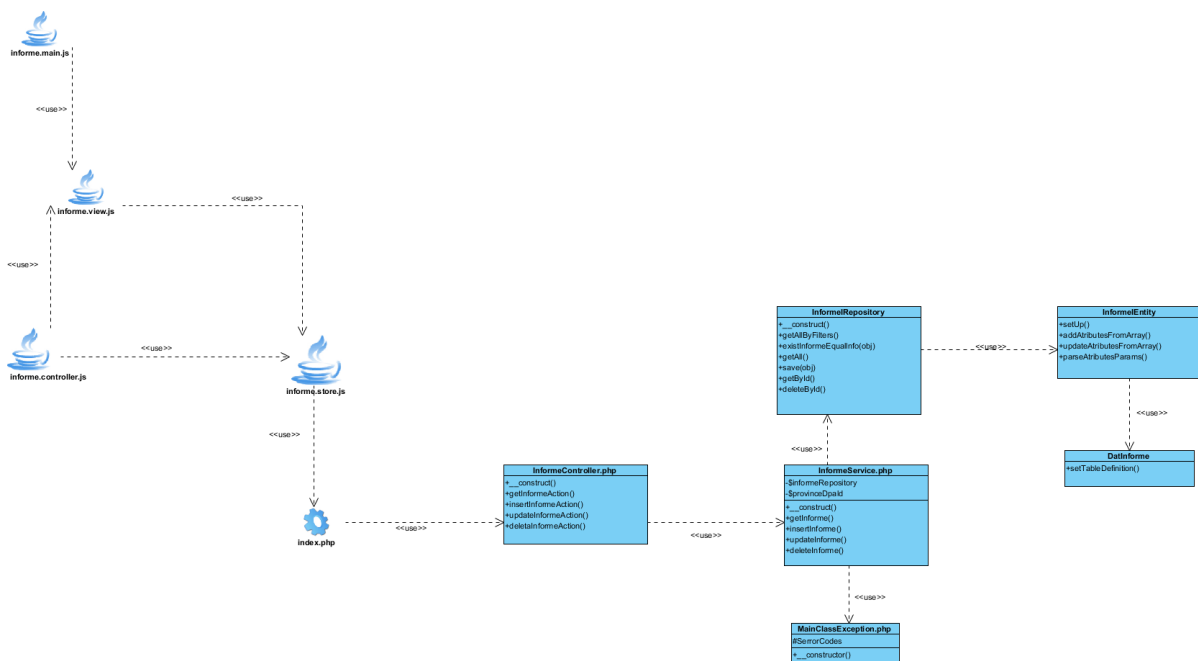


FIG. 10 DIAGRAMA DE CLASES DE DISEÑO DEL REQUISITO FUNCIONAL GESTIONAR INFORME DE CUMPLIMIENTO.

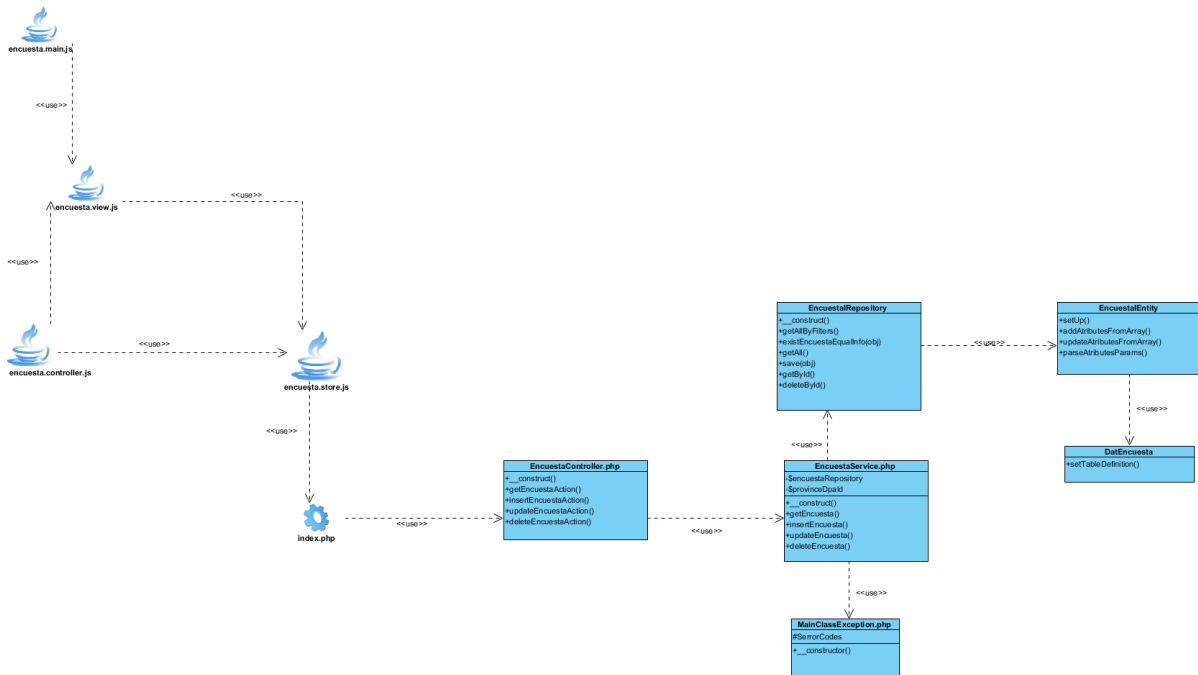


FIG. 11 DIAGRAMA DE CLASES DE DISEÑO DEL REQUISITO FUNCIONAL GESTIONAR ENCUESTAS.

2.9 Diagramas de Secuencia.

Los diagramas de secuencia correspondientes a los requisitos funcionales: Gestionar Manifestaciones, Gestionar Jornadas, Gestionar Programas Especiales, Gestionar Grupos Etarios, se encuentran en el documento Anexos de Evidencias.

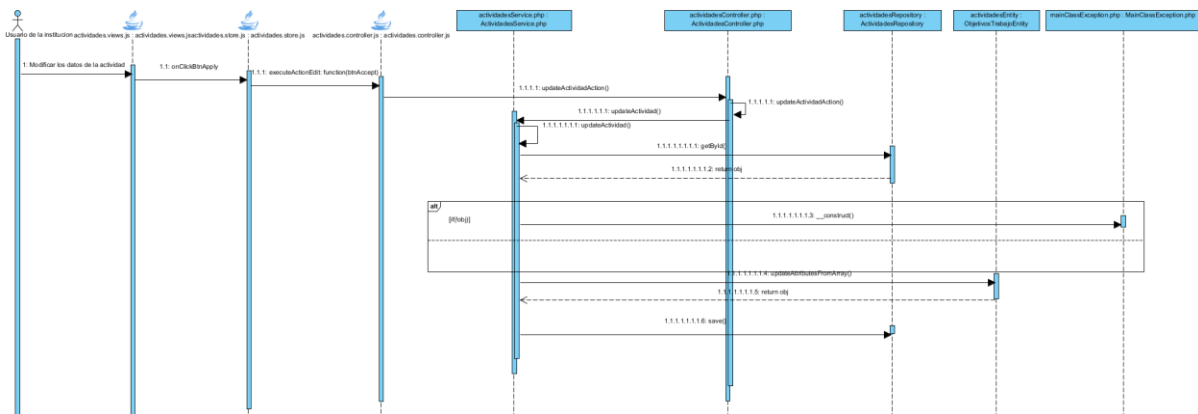


FIG. 12 DIAGRAMA DE SECUENCIA DEL REQUISITO FUNCIONAL GESTIONAR PLANIFICACIÓN.

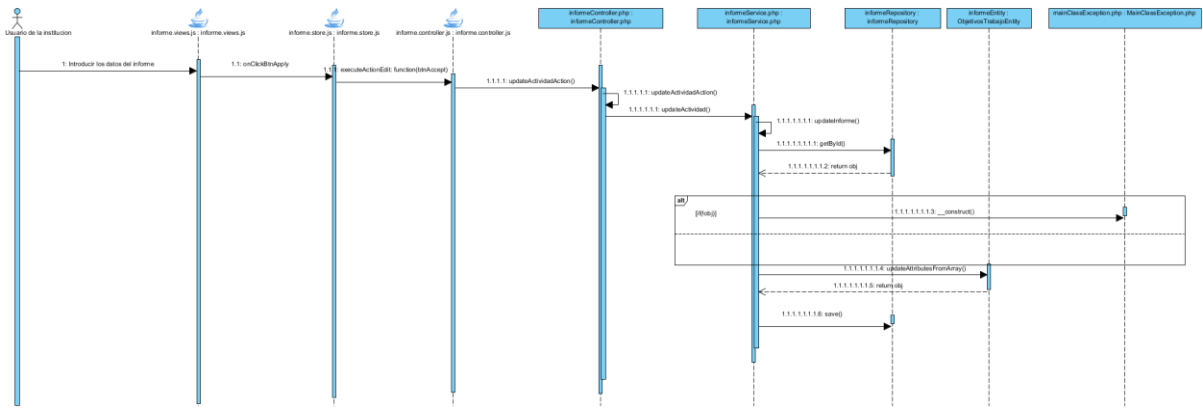


FIG. 13 DIAGRAMA DE SECUENCIA DEL REQUISITO FUNCIONAL GESTIONAR INFORMES DE CUMPLIMIENTO.

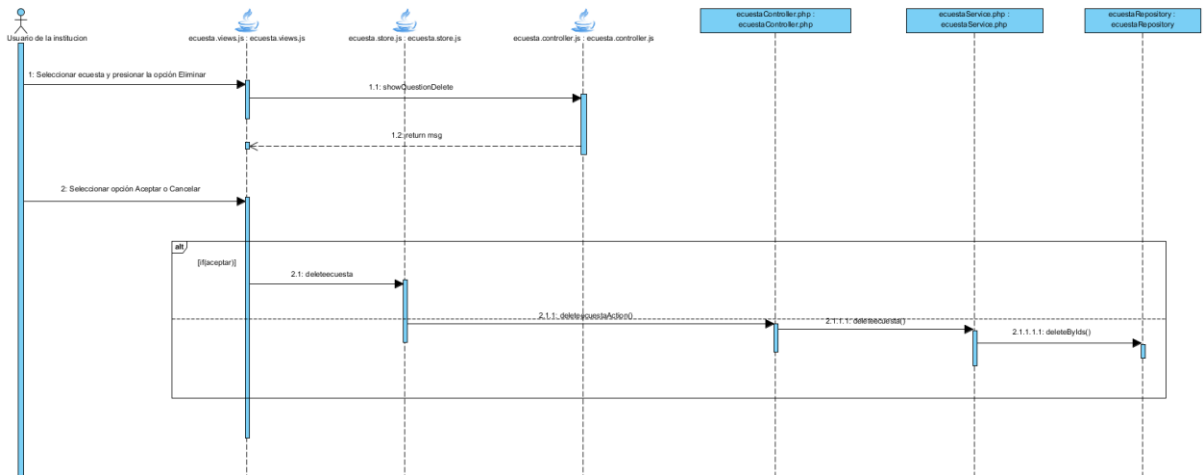


FIG. 14 DIAGRAMA DE SECUENCIA DEL REQUISITO FUNCIONAL GESTIONAR ENCUESTAS.

2.10 Seguridad.

La seguridad de software es una disciplina que se encarga de proteger las aplicaciones y el software de amenazas exteriores, así como garantizar la integridad y la privacidad de la información almacenada en un sistema. La seguridad de un producto desarrollado se orienta a la búsqueda de que dicho producto continúe funcionando correctamente ante ataques maliciosos. (Wordpress, 2016)

El mecanismo de seguridad utilizado para ello es una seguridad RBAC o Control de acceso basado en roles, consiste en asignar derechos de acceso a los usuarios de su organización en función de sus roles y las tareas que realizan. Esto garantiza que los usuarios y equipos solo puedan tener acceso a los niveles que pertenezcan.

El framework que se utilizará para el acceso a los datos, en este caso Doctrine, al igual que la mayoría de los ORM, presenta su propia capa de seguridad para la protección contra los ataques más comunes, como son las inyecciones SQL.

2.11 Análisis de factibilidad.

Para el análisis del costo y beneficios del sistema en este trabajo se utilizó la metodología de Prodesoft donde desde el inicio se estima de forma empírica la duración de la implementación de cada uno de los requisitos, basado en la experiencia del programador en el trabajo con el lenguaje de programación, el entorno de desarrollo, el conocimiento sobre el tema de investigación y las técnicas de programación necesarias para resolver el problema. Para esto es necesario realizar una estimación del esfuerzo, el tiempo de desarrollo de cada requisito y la cantidad de personas que participarán para poder determinar eficazmente si resulta beneficioso el desarrollo, aunque es importante señalar que solo se habla de una estimación. Esta metodología guía un proceso basado en componentes lo que permite ahorrar gran cantidad de tiempo.

2.11.1 Costo.

El análisis de costo es el proceso de identificación de los recursos necesarios para llevar a cabo el trabajo o proyecto eficientemente. Para la estimación del costo del software se empleará la fórmula propuesta por la metodología, para lo cual primeramente se tiene en cuenta los requisitos funcionales a desarrollar, así como su prioridad, complejidad y tiempo estimado de desarrollo, lo que se detalla en la siguiente tabla.

No	Nombre de requisito	Prioridad	Complejidad	Tiempo de desarrollo (semanas)
1	Gestionar Planificaciones	Alta	Alta	3
2	Gestionar Informes de Cumplimiento	Alta	Alta	3
3	Gestionar Encuestas	Alta	Alta	3
4	Gestionar Manifestaciones	Medio	Medio	2
5	Gestionar Jornadas	Medio	Medio	2
6	Gestionar Programas Especiales	Medio	Medio	2
7	Gestionar Grupos Etarios	Medio	Medio	2

Para la estimación del costo del software se empleó la fórmula propuesta por la metodología, donde se determina:

Costo diario de un trabajador= Tarifa horaria (MT) * Trabajador * Tiempo diario. = \$18.00 * 1 trabajador * 8 horas.
= \$144.00.

Tomando en cuenta que un trabajador cubano, trabaja 20 días laborables
= \$144.00. *20 = \$2 880.00.

Costo mensual de un trabajador = \$2 880.00.

Teniendo en cuenta un tiempo estimado de aproximadamente 6 meses para el desarrollo del proyecto se determina:

Costo total= \$2 880.00.* 5 meses.

= \$ 14 400.00

El costo de desarrollo del sistema será \$ 14 400.00 aproximadamente, lo que en materia económica constituye una cifra moderada de dinero con respecto a los beneficios que se evidencian a continuación.

2.11.2 Beneficios Tangibles e intangibles

La aplicación diseñada para la Dirección Provincial de Cultura, disminuirá el gasto de materiales y medios utilizados en la actualidad en el proceso que se realiza de forma manual, eliminará los retrasos y garantizará que la planificación llegue a la empresa de manera uniforme. Además, elevará la calidad de los servicios a la población y la eficiencia de forma general. El diseño del software cumple con las expectativas especificadas en los requerimientos funcionales del cliente.

2.11.3 Análisis del costo y beneficios

Con la estimación anterior de los costos y el análisis de los beneficios tangibles e intangibles, se llega a establecer una correlación entre ambos y arribar a la conclusión de que, es factible el desarrollo e implementación posterior en las entidades del territorio matancero, de un proceso para la gestión de solicitudes de servicios y más, si logra integrar e insertar a la población en la interacción con el gobierno y empresas de la comunidad.

2.12 Diseño de Pruebas de Software

2.12.1 Objetivos de las Pruebas.

Los principales objetivos de las pruebas son encontrar el mayor número de defectos en el código para que se resuelvan y eliminarlos. Asegurar que el producto funciona tal y cómo se ha definido en los requisitos. Dar información al cliente de los defectos que no se han podido eliminar del producto final. Proporcionar al producto final un grado mayor de calidad. Para cumplir estándares: Otro motivo por el cual las pruebas son necesarias es que los estándares de la industria muchas veces exigen la existencia de fases de pruebas en el proceso de desarrollo. Una de las razones por la que es necesario probar es que eliminar un defecto de software cuesta mucho más cuanto más tarde se detecte. Si no se prueba durante el proceso de desarrollo de software, encontrar un error en producción puede hacer que el producto no sea rentable. (Tester House, 2019)

2.12.2 Pruebas del Software.

Las pruebas de software propuestas se encuentran en el expediente de proyecto en el documento Anexos de Evidencias.

Todos los tipos de pruebas de software que existen, básicamente, se pueden agrupar en dos grupos: las pruebas funcionales y las pruebas no funcionales.

Tipos de pruebas funcionales:

- Pruebas unitarias.
- Pruebas de aceptación.
- Pruebas de integración.
- Pruebas de regresión.

Las pruebas no funcionales se centran en aspectos muy importantes del comportamiento del producto pero que no están relacionados con las funciones que realiza el sistema. (Tester House, 2019)

Pruebas de Aceptación.

Las pruebas de aceptación son las últimas pruebas realizadas donde el cliente prueba el software y verifica que cumpla con sus expectativas. Estas pruebas generalmente son funcionales y se basan en los requisitos definidos por el cliente y deben hacerse antes de la salida a producción. Las pruebas de aceptación son fundamentales por lo cual deben incluirse obligatoriamente en el plan de pruebas de software. Estas pruebas se realizan una vez que ya se ha probado que cada módulo funciona bien por separado, que el software realice las

funciones esperadas y que todos los módulos se integran correctamente. El test de aceptación termina de definir el nivel de calidad de la aplicación y le permite conocer al equipo qué tan bien supo interpretar correctamente los requerimientos del usuario o Product owner. Es el cliente quien tendrá la decisión final de aprobar o no el producto, como también de solicitar modificaciones. (Los Andes Training, 2017)

Pruebas de Caja Negra.

Las pruebas de caja negra, es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o escenarios de ejecución internos en el software. En las pruebas de caja negra, nos enfocamos solamente en las entradas y salidas del sistema, sin preocuparnos en tener conocimiento de la estructura interna del programa de software. Para obtener el detalle de cuáles deben ser esas entradas y salidas, nos basamos únicamente en los requerimientos de software y especificaciones funcionales. Al estar basadas en los requerimientos de software y en las entradas y salidas de cada funcionalidad, al definir una prueba de caja negra lo principal es identificar los datos de prueba (entradas) y el resultado esperado del sistema al ingresar esos datos, bien sean los datos de salida o algún comportamiento específico. (PMO Informatica , 2017)

A continuación, se muestran algunas pruebas de caja negra realizadas sobre la propuesta de solución:

Adicionar Actividades:

Pasos	Procedimientos	Resultados Esperados
EP 1.1: Adicionar actividades correctamente.	<ol style="list-style-type: none"> 1. Seleccionar en el menú principal la funcionalidad Crear Programación Cultural 2. Mostrar la interfaz Gestionar actividades. ver PIU Gestionar actividades 3. Seleccionar la opción Adicionar 	<ol style="list-style-type: none"> 1. Cerrar la ventana Adicionar actividad y mostrar el mensaje de información "La actividad ha sido adicionada satisfactoriamente". 2. Adicionar la actividad y actualizar el listado de actividades.

	<p>4. Mostrar la interfaz Adicionar actividades. Ver PIU Adicionar actividades</p> <p>5. Introducir los datos.</p> <p>Presionar la opción Aceptar</p> <p>6. Guardar datos y cerrar interfaz</p> <p>7. Mostrar el mensaje de información "La actividad ha sido adicionada satisfactoriamente"</p> <p>8. Actualizar el listado.</p>	
<p>EP 1.2 Adicionar actividad oprimiendo el botón Aplicar.</p>	<p>1. Seleccionar en el menú principal la funcionalidad Crear Programación Cultural</p> <p>2. Mostrar la interfaz Gestionar actividades. ver PIU Gestionar actividades</p> <p>3. Seleccionar la opción Adicionar</p> <p>4. Mostrar la interfaz Adicionar actividades. Ver PIU Adicionar actividades</p> <p>5. Introducir los datos.</p> <p>Presionar la opción Aplicar</p> <p>6.1. Guardar datos y limpiar los campos</p>	<p>1. Mostrar el mensaje de información: "La actividad se ha guardado satisfactoriamente", se actualiza el listado y limpiar los campos de la interfaz, permitiendo adicionar otra disposición.</p>

	<p>7. Mostrar el mensaje de información "La actividad ha sido adicionada satisfactoriamente"</p> <p>8. Actualizar el listado.</p>	
<p>EP 1.3: Adicionar actividad incorrectamente.</p>	<p>1. Seleccionar en el menú principal la funcionalidad Crear Programación Cultural</p> <p>2. Mostrar la interfaz Gestionar actividades. ver PIU Gestionar actividades</p> <p>3. Seleccionar la opción Adicionar</p> <p>4. Mostrar la interfaz Adicionar actividades. Ver PIU Adicionar actividades</p> <p>5. Introducir los datos.</p> <p>Presionar la opción Aceptar</p> <p>6. Guardar datos y cerrar interfaz</p> <p>7. Mostrar el mensaje de información "La actividad ha sido adicionada satisfactoriamente"</p> <p>8. Actualizar el listado.</p>	<p>1. Mostrar un mensaje de error en los tooltips de los campos según corresponda.</p>

<p>EP 1.4 Oprimir botón Cancelar</p>	<p>1. Seleccionar en el menú principal la funcionalidad Crear Programación Cultural</p> <p>2. Mostrar la interfaz Gestionar actividades. ver PIU Gestionar actividades</p> <p>3. Seleccionar la opción Adicionar</p> <p>4. Mostrar la interfaz Adicionar actividades. Ver PIU Adicionar actividades</p> <p>5. Introducir los datos.</p> <p>Presionar la opción Cancelar</p> <p>6.2 Cancelar la acción de adicionar y cerrar la interfaz.</p>	<p>1. Cancelar todas las acciones, cerrar la ventana, y regresar a la interfaz principal.</p>
--------------------------------------	--	---

Modificar Actividades:

Pasos	Procedimientos	Resultados esperados
<p>EP 1.1: Modificar actividad correctamente.</p>	<p>1. Seleccionar en el menú principal la funcionalidad Crear Programación Cultural</p> <p>2. Mostrar la interfaz Gestionar actividades. ver PIU Gestionar actividades</p>	<p>1. Cerrar la ventana Modificar actividad y mostrar el mensaje de información "La actividad ha sido adicionada satisfactoriamente".</p> <p>2. Modificar la actividad y actualizar el listado de actividades.</p>

	<p>3. Seleccionar del listado la actividad que desea modificar</p> <p>4. Seleccionar la opción Modificar</p> <p>5. Mostrar la interfaz Modificar planificación. Ver PIU Modificar planificación</p> <p>6. Modificar los datos que desee.</p> <p>Presionar la opción Aceptar</p> <p>7.1. Guardar datos y cerrar interfaz</p> <p>8. Mostrar el mensaje de información "La actividad ha sido modificada satisfactoriamente"</p> <p>9. Actualizar el listado.</p>	
--	---	--

<p>EP 1.2: Modificar actividad incorrectamente.</p>	<ol style="list-style-type: none"> 1. Seleccionar en el menú principal la funcionalidad Crear Programación Cultural 2. Mostrar la interfaz Gestionar actividades. ver PIU Gestionar actividades 3. Seleccionar del listado la actividad que desea modificar 4. Seleccionar la opción Modificar 5. Mostrar la interfaz Modificar planificación. Ver PIU Modificar planificación 6. Modificar los datos que desee. Presionar la opción Aceptar 	<ol style="list-style-type: none"> 1. Mostrar un mensaje de error en los tooltips de los campos según corresponda.
<p>EP 1.3_Oprimir botón Cancelar.</p>	<ol style="list-style-type: none"> 1. Seleccionar en el menú principal la funcionalidad Crear Programación Cultural 	<ol style="list-style-type: none"> 1. Cancelar todas las acciones, se cierra la ventana, y se regresa a la interfaz principal.

	<p>2. Mostrar la interfaz Gestionar actividades. ver PIU Gestionar actividades</p> <p>3. Seleccionar del listado la actividad que desea modificar</p> <p>4. Seleccionar la opción Modificar</p> <p>5. Mostrar la interfaz Modificar planificación. Ver PIU Modificar planificación</p> <p>6. Modificar los datos que desee.</p> <p>Presionar la opción Cancelar</p> <p>7.2 Cancelar la acción de modificar y cerrar la interfaz.</p>	
--	--	--

Adicionar Encuesta:

Pasos	Procedimientos	Resultados Esperados

<p>EP 1.1: Adicionar encuesta correctamente.</p>	<ol style="list-style-type: none"> 1. Seleccionar en el menú principal la funcionalidad Gestionar encuestas 2. Mostrar la interfaz Gestionar encuestas. ver PIU Gestionar encuestas 3. Seleccionar la opción Adicionar 4. Mostrar la interfaz Adicionar encuestas. Ver PIU Adicionar encuestas 5. Introducir los datos. Presionar la opción Aceptar 6. Guardar datos y cerrar interfaz 7. Mostrar el mensaje de información "La encuesta ha sido adicionada satisfactoriamente" 8. Actualizar el listado. 	<ol style="list-style-type: none"> 1. Cerrar la ventana Adicionar encuesta y mostrar el mensaje de información "La encuesta ha sido adicionada satisfactoriamente". 2. Adicionar la encuesta y actualizar el listado de encuestas.
<p>EP 1.2 Adicionar encuesta</p>	<ol style="list-style-type: none"> 1. Seleccionar en el menú principal la funcionalidad Gestionar encuestas de cumplimiento 	<ol style="list-style-type: none"> 1. Mostrar el mensaje de información: "La encuesta se ha guardado satisfactoriamente",

<p>oprimiendo el botón Aplicar.</p>	<p>2. Mostrar la interfaz Gestionar encuestas. ver PIU Gestionar encuestas</p> <p>3. Seleccionar la opción Adicionar</p> <p>4. Mostrar la interfaz Adicionar encuestas. Ver PIU Adicionar encuestas</p> <p>5. Introducir los datos.</p> <p>Presionar la opción Aplicar</p> <p>6.1. Guardar datos y limpiar los campos</p> <p>7. Mostrar el mensaje de información "La encuesta ha sido adicionada satisfactoriamente"</p> <p>8. Actualizar el listado.</p>	<p>se actualiza el listado y limpiar los campos de la interfaz, permitiendo adicionar otra disposición.</p>
<p>EP 1.3: Adicionar encuesta incorrectamente.</p>	<p>1. Seleccionar en el menú principal la funcionalidad Gestionar encuestas de cumplimiento</p>	<p>1. Mostrar un mensaje de error en los tooltips de los campos según corresponda.</p>

	<p>2. Mostrar la interfaz Gestionar encuestas. ver PIU Gestionar encuestas</p> <p>3. Seleccionar la opción Adicionar</p> <p>4. Mostrar la interfaz Adicionar encuestas. Ver PIU Adicionar encuestas</p> <p>5. Introducir los datos.</p> <p>Presionar la opción Aceptar</p>	
<p>EP 1.4 Oprimir botón Cancelar</p>	<p>1. Seleccionar en el menú principal la funcionalidad Gestionar encuestas de cumplimiento</p> <p>2. Mostrar la interfaz Gestionar encuestas. ver PIU Gestionar encuestas</p> <p>3. Seleccionar la opción Adicionar</p> <p>4. Mostrar la interfaz Adicionar encuestas. Ver PIU Adicionar encuestas</p> <p>5. Introducir los datos.</p>	<p>1. Cancelar todas las acciones, cerrar la ventana, y regresar a la interfaz principal.</p>

	<p>Presionar la opción Cancelar</p> <p>6.2 Cancelar la acción de adicionar y cerrar la interfaz.</p>	
--	--	--

2.13 Conclusiones Parciales.

En este capítulo luego de describir la propuesta de solución al problema científico de esta investigación se concluye que la modelación de todos los procesos que intervinieron durante el desarrollo del sistema permite una visión general del producto deseado. La realización del modelado del proceso permitió un mayor entendimiento del negocio, lo que facilitará la implementación de la solución propuesta. El levantamiento de los requisitos funcionales y su descripción fue fundamental para lograr un diseño más apropiado del sistema. Los diagramas de secuencia permitieron el proceso de implementación del sistema de una forma más clara. La estimación del costo del software junto con el estudio de la relación beneficio demostró la factibilidad del desarrollo del mismo.

Conclusiones

Con el desarrollo de esta investigación podemos concluir que se cumplió con los objetivos trazados a partir de:

- El estudio realizado sobre los antecedentes, el estado actual de la temática, la bibliografía y documentos relacionados con el objeto de estudio, permitió contar con los elementos necesarios para dar solución a la problemática planteada.
- Se realizó un estudio acerca de las herramientas y tecnologías a utilizar, determinando así sus eficiencias y deficiencias.
- Se realizó la estimación de costo de implementación del sistema y el estudio de factibilidad, arrojando como resultado la factibilidad de la realización del sistema informático.
- Se realizó el modelado del negocio teniendo en cuenta los pasos que describe la metodología Prodesoft para su realización.
- Las pruebas propuestas se emplearán con el fin de entregarle al cliente un producto totalmente funcional, cumpliendo con todos los requisitos demandados por el mismo y satisfaciendo sus necesidades.
- El software propuesto será una herramienta fácil de utilizar para la Dirección Provincial de Cultura y le facilitará a la población la gestión de estos servicios desde cualquier lugar que se encuentre.
- Permitirá el ahorro de tiempo, medios, y recursos para la gestión de los procesos.
- La aplicación de este trabajo tiene gran importancia en el país ya que constituye un paso más al desarrollo informático dentro del gobierno electrónico y una herramienta de gran ayuda para la realización de este proceso.

Recomendaciones

Desde el punto de vista del alcance del presente trabajo y teniendo en cuenta el tiempo para el desarrollo del mismo, se proponen las siguientes recomendaciones:

- Desplegar el sistema en la empresa Dirección Provincial de Cultura de Matanzas para validar de forma experimental la utilidad de la misma.
- Explotar al máximo las potencialidades del sistema.
- Proponer la validación de la aplicación web para la gestión de la programación cultural en la Dirección Provincial de Cultura.

Bibliografía

Antón, Susana. 2018. Cómo contribuye la UIC a la informatización de la sociedad? [En línea] 4 de Octubre de 2018. [Citado el: 2 de Diciembre de 2019.] <http://www.granma.cu/cuba/2018-10-03/como-contribuye-la-uic-a-la-informatizacion-de-la-sociedad-03-10-2018-21-10-06>.

Cubarte. 2020. Cubarte. [En línea] 2020. [Citado el: 16 de 08 de 2020.] <http://app.lapapeleta.cu/>.

Developers Toptal. 2016. [En línea] 3 de 2016. <https://www.toptal.com/front-end/what-is-bootstrap-a-short-tutorial-on-the-what-why-and-how>.

Doctrine. sf. www.doctrine-project.org. www.doctrine-project.org. [En línea] sf. [Citado el: 12 de Mayo de 2020.] <https://www.doctrine-project.org/projects/doctrine-orm/en/current/tutorials/getting-started.html>.

EcuRed contributors. 2019. Gobierno Electrónico. *EcuRed*. [En línea] 22 de julio de 2019. https://www.ecured.cu/index.php?title=Gobierno_electr%C3%B3nico&oldid=3469296.

García, Miriam. 2017. coding or not. *coding or not*. [En línea] 5 de octubre de 2017. <https://codingornot.com/mvc-modelo-vista-controlador-que-es-y-para-que-sirve>.

Gobierno Matancero. 2019. www.matanceros.gob.cu. www.matanceros.gob.cu. [En línea] 20 de 11 de 2019. [Citado el: 18 de Febrero de 2020.] <http://www.matanceros.gob.cu>.

Guru99. 2020. [En línea] 17 de 6 de 2020. <https://www.guru99.com/what-is-dbms.html>.

How-to Geek. 2018. [En línea] 5 de 7 de 2018. <https://www.howtogeek.com/357092/what-is-an-xml-file-and-how-do-i-open-one/>.

Isasi, Yunielis Moliner. 2018. Radio 26. [En línea] 25 de 5 de 2018. [Citado el: 15 de 8 de 2020.] <http://www.radio26.cu/plataforma-digital-bienestar/>.

Jetbrains. sf. www.jetbrains.com. www.jetbrains.com. [En línea] sf. [Citado el: 12 de Mayo de 2020.] <https://www.jetbrains.com/phpstorm/features/>.

Los Andes Training. 2017. [En línea] 23 de 8 de 2017. <https://losandestraining.com/2017/08/23/que-son-las-pruebas-de-aceptacion/>.

Microsoft. 2016. [En línea] 2016. [Citado el: 25 de 6 de 2020.] <https://support.microsoft.com/es-es/office/conceptos-b%C3%A1sicos-del-dise%C3%B1o-de-una-base-de-datos-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5?ui=es-es&rs=es-es&ad=es>.

Milanés, Laydis. 2019. Hacia un gobierno electrónico mas interactivo. *Cubahora*. [En línea] 2019 de abril de 2019. <https://www.cubahora.cu/ciencia-y-tecnologia/hacia-un-gobierno-electronico-con-mas-interaccion>.

Ministerio de Cultura. 2019. [En línea] 30 de octubre de 2019. <http://www.ministeriodecultura.gob.cu/que-es/#atribuciones>.

Mozilla. sf. developer.mozilla.org. *developer.mozilla.org*. [En línea] sf. [Citado el: 12 de Mayo de 2020.] <https://developer.mozilla.org/docs/Web/HTML>.

Mozilla. sf. developer.mozilla.org. *developer.mozilla.org*. [En línea] sf. [Citado el: 20 de Mayo de 2020.] <https://developer.mozilla.org/docs/Web/JavaScript>.

Nextech. 2016. Nextech. *Nextech Web Site*. [En línea] 16 de 9 de 2016. <https://nextech.pe/que-es-bpmn-y-para-que-sirve/>.

OBS Business School. sf. [En línea] sf. [Citado el: 26 de 8 de 2020.] <https://obsbusiness.school/es/blog-proyect-managment/metodologia-agile/que-son-las-metodologias-de-desarrollo-de-software>.

OMG System Modeling Language. sf. [En línea] sf. [Citado el: 1 de june de 2020.] <https://omgsysml.org/what-is-sysml>.

pgAdmin. 2016. [En línea] 20 de 8 de 2016. <https://www.pgadmin.org/faq/#1>.

Php. sf. www.php.net. *www.php.net*. [En línea] sf. [Citado el: 12 de Mayo de 2020.] <https://www.php.net/manual/preface.php>.

PMO Informatica . 2017. [En línea] 20 de 2 de 2017. <http://www.pmoinformatica.com/2017/02/pruebas-de-caja-negra-ejemplos.html>.

Postgresql. sf. www.postgresql.org. *www.postgresql.org*. [En línea] sf. [Citado el: 12 de Mayo de 2020.] <https://www.postgresql.org/about/>.

Pressman. 2014. *Software Engineering: A Practitioner's Approach*. s.l. : 8, 2014.

QuickStart. 2018. [En línea] 13 de 3 de 2018. <https://www.quickstart.com/blog/what-is-sql-server-and-how-does-it-work/>.

Quora. 2017. [En línea] 1 de 6 de 2017. <https://www.quora.com/What-exactly-is-JQuery-and-what-can-you-use-it-for>.

Radio Rebelde. 2018. Radio Rebelde. *Radio Rebelde*. [En línea] 19 de 12 de 2018. <http://www.radiorebelde.cu/noticia/funciona-matanzas-primer-parque-cientifico-tecnologico-pais-20181219/>.

Software Advice. [En línea] <https://www.softwareadvice.com/bpm/processmaker-profile/>.

Source Making. sf. Source Making. [En línea] sf. https://sourcemaking.com/design_patterns/singleton.

Technopedia. 2020. Technopedia. [En línea] 2020. <https://www.techopedia.com/definition/17185/n-tier-architecture>.

TechTerms. 2018. The Tech Terms Computer Dictionary. [En línea] 2 de 2018. <https://techterms.com/definition/css>.

Tester House. 2019. [En línea] 9 de 4 de 2019. <https://testerhouse.com/teoria-testing/objetivo-de-las-pruebas-de-software/>.

Tutorials Point. sf. *Tutorials Point*. [En línea] sf.
https://www.tutorialspoint.com/extjs/extjs_overview.htm.

Tutorialspoint. 2020. Tutorialspoint. [Online] 2020.
https://www.tutorialspoint.com/software_architecture_design/introduction.htm.

Visua Paradigm International LTD. sf. Visua Paradigm. [Online] sf. [Cited: june 1, 2020.]
<https://www.visual-paradigm.com/support/faq.jsp>.

W3C Note. 2002. [En línea] 15 de 3 de 2002. <https://www.w3.org/TR/wsd.html>.

Wordpress. 2016. [En línea] 2016. [Citado el: 25 de 8 de 2020.]
<https://www.google.com/amp/s/ittgweb.wordpress.com/2016/05/29/4-1-seguridad-de-software/amp/>.

Xarxa punt TIC. 2020. puntic.gencat.cat. [En línea] 03 de 02 de 2020. [Citado el: 15 de 08 de 2020.]
<https://punttic.gencat.cat/es/article/vibra-la-nueva-app-para-recoger-la-programacion>.

Xetid. 2012. *Proceso de Desarrollo y Gestión de Proyectos de Software*. La Habana : s.n., 2012.

Zend Framework. sf. framework.zend.com. *framework.zend.com*. [En línea] sf. [Citado el: 12 de Mayo de 2020.] <https://framework.zend.com/about>.

Anexos



Fig. 1 Etapas del ciclo de vida del proyecto

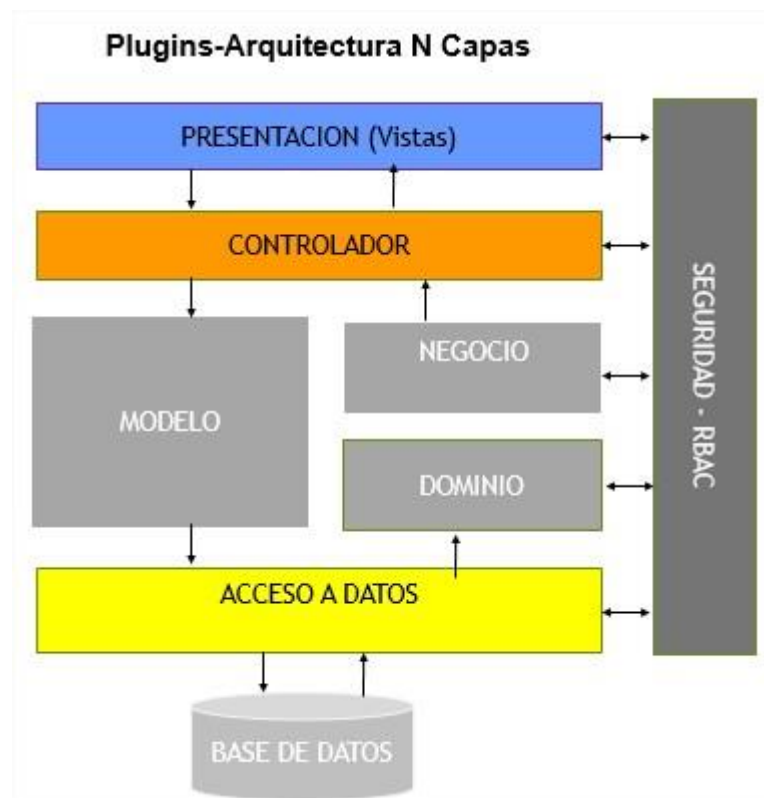


Fig. 2 Arquitectura N-Capas