

Universidad de las Ciencias Informáticas

Facultad 1



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

Título:

Sistema para la elaboración de cuestionarios en las investigaciones sociales

Autor: Pedro Farnot Zequeira

Tutores:

Ing. Estela Odelsa Martín Coronel

Ing. Alejandro Hernández Rodríguez

La Habana, 2021

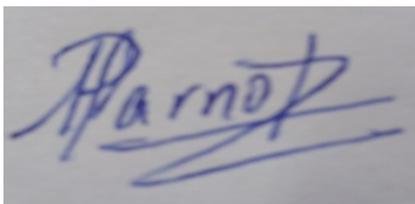
Año 63 de la Revolución

Í ECLARACIÓN DE AUTORÍA:

Declaramos ser autor(es) de la presente tesis y reconocemos a la CESPO, Centro de Estudios Sociopolíticos y de Opinión, los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los 9 días del mes diciembre del año 2021.

Pedro Farnot Zequeira.



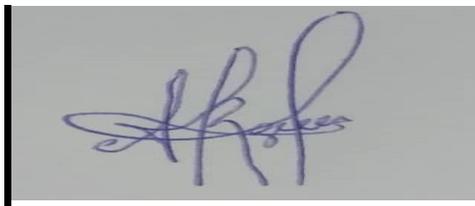
Firma de Autor

Ing. Estela Odelsa Martín Coronel.



Firma de Tutor

Ing. Alejandro Hernández Rodríguez.



Firma de Tutor

Agradecimientos:

En primer lugar me gustaría agradecer a mi mamá, a mi papá, a mi abuela, que son los verdaderos artífices de toda obra que yo pueda realizar, por ser la base de mis ideas, de mis actos, por ser ejemplo, por darme la vida, por darme cuidado. Sin dudas no puedo dejar de destacar a mi tío Félix, todo lo que soy y tengo se ve reflejado en él. Pero no menos importante son mis otros tíos y ellos lo saben. Debo agradecer también a mi hermana y mis primos, que juntos han hecho que pueda confiar en mí y me han dado la seguridad de un hogar en cada uno de ellos.

Con el mismo sentimiento agradezco a mis amigos por siempre estar cuando los he necesitado, por llorar conmigo, por reír a mi lado.

A mis compañeros del grupo, decirles que gracias por los estudios, gracias por la constancia, gracias por las fiestas, gracias por todo.

Al amor también le dedicaré su espacio en este agradecimiento, gracias Cleysi por ayudarme en todo el transcurso de la universidad. Pero muchas gracias Adarys por aparecer en el momento en que necesitaba la tranquilidad, seguridad y pasión que has logrado brindarme.

Dedicatoria:

A ti, MADRE, va dedicada esta tesis, a ti por hacerme nacer, a ti por escoger a mi padre, a ti por darme la familia que tengo.

Esto no es más que aquella promesa que te hice de ser un ingeniero en ciencias. Este trabajo es por ti, para que donde estés, te encuentres orgullosa de tu hijo, que piensa en ti cada día, que el día que nos volvamos a encontrar, pueda abrazarte y decirte: he cumplido MAMÁ...

Pensamiento:



“Si estás dispuesto a trabajar duro y a cumplir con tus responsabilidades, podrás salir adelante. No importa de dónde vengas, como luzcas o a quién ames.”

Barack Obama.

Resumen

Para los investigadores del Centro de Estudios Sociopolíticos y de Opinión (CESPO) se torna compleja la elaboración de los cuestionarios (técnica de captura de información) debido al volumen de indicadores, preguntas y respuestas que se utilizan en la confección de estos para ser aplicados en investigaciones sociales, lo que provoca una demora en el momento de diseñar los cuestionarios y a su vez un atraso en la culminación de las investigaciones, que, en dicho centro, se requiere menos tiempo que en otros que realizan investigaciones similares en el país, por su nivel de importancia para el Comité Central del Partido Comunista de Cuba (CCPCC). El presente trabajo de diploma tiene como objetivo desarrollar un Sistema para la elaboración de cuestionarios en investigaciones sociales que facilite a los investigadores la elaboración de los mismos. La reciente investigación se desarrolló teniendo en cuenta el proceso de desarrollo híbrido SCRUM-RUP, y las tecnologías que contiene el marco de trabajo Quasar (Express.js, Vue.js), además se utilizó la librería DOCX para el desarrollo de los cuestionarios. Como resultado se obtuvo un sistema capaz de elaborar los cuestionarios, posibilitando a los investigadores, facilidad en el proceso de crear los cuestionarios. La investigación finaliza con las pruebas funcionales del software, las cuales demostraron la correcta implementación de todas sus funcionalidades.

Palabras claves: investigación social, pregunta, respuesta, cuestionarios, sistema.

Índice

Introducción.....	10
Capítulo I: Fundamentación teórica del desarrollo del sistema para la elaboración de cuestionarios en investigaciones sociales.....	14
1.1. Introducción.....	14
1.2. Conceptos asociados al sistema para la elaboración de cuestionarios.....	14
1.3. Estudio de sistemas homólogos.....	15
1.3.1 Lime Survey.....	15
1.3.2 RotatorSurvey.....	16
1.3.3 Moodle.....	16
1.3.4 Banco de preguntas para la generación de encuestas.....	17
1.3.5 Análisis realizado a los sistemas informáticos para la elaboración de cuestionarios y sondeos.....	17
1.4. Tendencias en el desarrollo de aplicaciones web.....	18
1.5. Metodología de desarrollo de software.....	19
1.6. Herramientas y tecnologías.....	20
1.6.1. Lenguajes de programación utilizados.....	20
HTML.....	20
CSS.....	20
JavaScript.....	21
PostgreSQL v10.0.0.....	21
1.7. Lenguaje de modelado.....	21
UML (Lenguaje de Modelado Unificado).....	21
BPMN (Notación de Modelado de Procesos de Negocio).....	22
1.8. Herramientas de Modelado.....	22
Herramienta CASE.....	22
1.8.1. Visual Paradigmv10.0.....	22
1.9. Entorno de Desarrollo Integrado.....	23
1.12.1. Visual Studio Codev1.56.2.....	23

1.10.	Entorno de ejecución.....	23
	Node.js v14.16.1.....	23
1.11.	Framework.....	23
1.12.	Express.jsv6.14.12.....	23
	Vue.jsv3.0.1.....	23
	Quasarv2.0.0.....	24
1.13.	Servidor web.....	24
	Nginxv1.20.1.....	24
1.14.	Navegador web.....	24
	Firefoxv89.0.....	24
1.15.	Herramientas para pruebas de software.....	24
	JMeterv3.0.....	25
	Postmanv8.1.....	25
1.16.	Conclusiones Parciales.....	25
Capítulo II: Descripción del desarrollo del sistema para la elaboración de cuestionarios en las investigaciones sociales.....		26
2.1	Introducción.....	26
2.2	Propuesta de Solución.....	26
2.3	Modelado del proceso de negocio.....	26
2.4	Levantamiento de requisitos.....	28
2.5	Requisitos funcionales.....	28
2.6	Requisitos no funcionales.....	29
2.6.1	Patrón modelo vista controlador (MVC).....	31
2.7	Diagrama de clases del diseño.....	32
2.8	Diagrama de secuencia.....	33
2.9	Patrones de diseño.....	34
2.10	Diagrama de caso de uso.....	36
2.11	Diagrama de componentes.....	37

2.12	Modelo de datos.....	38
2.13	Conclusiones Parciales.....	38
Capítulo III: Implementación y prueba del Desarrollo del sistema para la elaboración de cuestionarios en investigaciones sociales.....		40
3.1	Introducción.....	40
3.2	Implementación.....	40
3.3	Diagrama de despliegue.....	40
3.4	Estándares de codificación.....	41
3.4.1	Convenciones de codificación.....	42
3.5	Pruebas al sistema para la elaboración de cuestionarios en investigaciones sociales.....	43
3.5.1	Pruebas Unitarias.....	43
3.5.2	Pruebas de integración.....	50
3.5.3	Prueba de aceptación.....	50
3.6	Conclusiones Parciales.....	50
Conclusiones generales.....		51
Recomendaciones:.....		52

Introducción

Una de las bases principales de la civilización actual es la investigación. Ella ha mejorado el bienestar de los hombres; los ha liberado de la esclavitud, del trabajo pesado y ha hecho su vida más sana, más bella y más rica en espiritualidad (Houssay, 1960).

Una investigación tiene su origen en la curiosidad inherente al género humano y en la necesidad esencial de obtener información para resolver las necesidades y preguntas que se le plantean al hombre. La investigación científica es la fuente de la ciencia, la cual se tiene que desarrollar según un método científico claramente estructurado y dirigido a profundizar y ampliar nuestro conocimiento de la realidad (Pujol, 2008).

En las investigaciones de corte social, los investigadores se encuentran ante la necesidad de conocer las escalas de categorías que utilizan las personas que constituyen la población objeto de interés. Lo ideal es recurrir a sistemas de indagación de información que respeten tales categorías, de tal forma que los datos obedezcan más a la naturalidad de los entornos cotidianos que a la artificialidad de los términos académicos (Durán y Ocaña, 2000).

No es hasta finales de la década de los años 70 y principios de los 80 del pasado siglo XX que se avanza en el empleo de las Tecnologías de la Información y la Comunicación (TIC) en las investigaciones sociales, primero como herramienta instrumental para el procesamiento, codificación y recuperación de la información y posteriormente para el análisis de datos. En la actualidad, a pesar de la resistencia al cambio, la instrumentación de estas tecnologías en las diferentes etapas o fases de las investigaciones cualitativas son un hecho, están presentes en la elaboración del problema de investigación, búsqueda y selección de la información para la construcción del marco teórico referencial, procesos de muestreo, recogida, procesamiento y análisis de la información; así como en la divulgación de los resultados. Sin lugar a dudas, la incorporación de las TIC a las investigaciones sociales revoluciona las formas del quehacer investigativo, crea novedosos espacios colaborativos y abre nuevos caminos que necesitan reformular aspectos teóricos y metodológicos (Rosabal y Vidal, 2018).

En el escenario cubano, dinamizado por importantes procesos sociopolíticos relacionados con la actualización del modelo económico y social y la renovación del texto constitucional, se advierten importantes necesidades cognitivas valorativas para la dirección del Partido Comunista de Cuba (PCC) en las singulares condiciones de transición socialista, lo que ha favorecido el desarrollo de investigaciones en esta área del conocimiento (Fuentes-Puentes, 2019).

El Centro de Estudios Sociopolíticos y de Opinión, en lo adelante CESPO adjunto al Comité Central del Partido Comunista de Cuba (PCC) acumula una amplia experiencia en el desarrollo de las investigaciones sociopolíticas que perfecciona cada día. En la realización de estas investigaciones se utilizan varias técnicas para la captura de información como son los cuestionarios (una de las más utilizadas).

El ejercicio de construcción de un cuestionario implica también un proceso de seleccionar qué es lo que se va a preguntar del conjunto de preguntas y temas posibles a investigar. Este ejercicio de recorte de la realidad implica no sólo una actividad técnica, sino también una toma de posición ideológica (consciente o

inconsciente) por parte del investigador, el que no sólo obtendrá información sobre su objeto de estudio, sino que “producirá” dicha información a partir de su propio punto de vista, contribuyendo a la construcción de una imagen social sobre sus sujetos y tema de investigación (Cerón, 2006).

Los cuestionarios basan su indagación en la validez de la información verbal de percepciones, sentimientos, actitudes o conductas que transmite el encuestado, información que, en muchos casos, es difícil de contrastar y traducir a un sistema de medida o a una puntuación. Es esta característica lo que hace tan complejo establecer los criterios de calidad de este tipo de instrumentos (Arribas, 2004).

Los cuestionarios aplicados por el CESPO acumulan un gran número de sujetos, que dan respuesta a la mayor cantidad de variables; sus dimensiones e indicadores del tema estudiado. Sus resultados se complementan con otros resultados que se obtienen en el resto de las técnicas cualitativas que se aplican en el transcurso de las investigaciones, estrategia que se planifica desde la elaboración del diseño.

La calidad de las investigaciones en el CESPO es cada vez mayor, con ello aumenta la demanda de utilizar herramientas avanzadas que aporten más información sobre el estudio realizado y rapidez en la elaboración de las técnicas. Actualmente el CESPO no cuenta con una herramienta centralizada de fácil acceso y filtrado que permita agrupar las preguntas y respuestas por temáticas y clasificaciones para la elaboración de los cuestionarios, lo cual aumenta el tiempo de trabajo en el diseño metodológico de las técnicas. Además, los cuestionarios se confeccionan manualmente, disminuyendo así la capacidad de los investigadores para la elaboración de estos. Las políticas de confidencialidad de la información inherente al centro provocan que los nuevos especialistas que se incorporen a la entidad no tengan acceso a las experiencias de investigaciones de años precedentes lo que limitaría el trabajo de los mismo.

A partir de estos elementos, se plantea el siguiente **problema de investigación**:

¿Cómo automatizar la elaboración de los cuestionarios que se aplican en las investigaciones sociales?

Para solucionar el problema planteado se declara como **objeto de estudio**: el proceso de elaboración de cuestionarios en investigaciones. El **campo de acción** se centra en la elaboración de cuestionarios de investigaciones sociales. Teniendo en cuenta el problema de investigación se define como **objetivo general**: Automatizar la elaboración de cuestionarios a través de un sistema informático que facilite el trabajo en las investigaciones sociales.

El objetivo general planteado será desglosado en los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación a partir del estado del arte referente al proceso informático para elaborar cuestionarios y sondeos de investigaciones.
- Identificar los aspectos funcionales y no funcionales que debe tener el sistema.
- Realizar las actividades y artefactos que tributen al diseño de la propuesta de solución, de acuerdo al proceso de desarrollo utilizado.
- Realizar las actividades y artefactos que tributen a la implementación de la propuesta de solución, de acuerdo al proceso de desarrollo utilizado.

- Realizar pruebas a las funcionalidades implementadas.

Idea a defender:

El desarrollo del sistema informático para la elaboración de cuestionarios en las investigaciones sociales facilitará el trabajo en las investigaciones del CESPO.

Los **métodos científicos y técnicos de investigación** utilizados en la elaboración de este trabajo fueron los siguientes:

- **Analítico - Sintético:** Para la realización de la investigación se analizaron y confrontaron los criterios de distintos autores sobre los conceptos obtenidos acerca de los hechos que se relacionan con la elaboración de los cuestionarios, partiendo de la descomposición del objeto de estudio en cada una de sus partes para estudiarlas en forma individual.
- **Histórico - Lógico:** Este método permite obtener una breve reseña sobre la evolución, desarrollo y funcionamiento de los sistemas informáticos para la elaboración de cuestionarios, investigar las leyes generales que los norman y establecer la relación actual con el desarrollo de las tecnologías informáticas, así como confrontar los elementos extraídos de las fuentes consultadas para establecer los fundamentos teóricos de la investigación.
- **Modelación:** Permite crear la abstracción con el objetivo de explicar la forma en la que los sistemas informáticos para la elaboración de cuestionarios gestionan las preguntas y respuestas. Se elaboraron representaciones explícitas de la arquitectura y la integración del sistema con otras plataformas.

Técnicas empíricas:

- **Entrevistas:** Se realizaron encuentros o entrevistas no formales a los especialistas del CESPO para obtener información acerca del proceso de elaboración de cuestionarios.
- **Observación:** Posibilitó obtener conocimiento acerca del proceso de elaboración de cuestionarios en investigaciones sociales actualmente y en particular en el CESPO.

Capítulo 1: Fundamentación teórica.

En este capítulo se hace un análisis de los principales conceptos relacionados con el dominio del objeto de estudio de la investigación y se realiza un estudio del estado del arte de herramientas informática para la creación del sistema para la elaboración de cuestionarios en investigaciones sociales. Además, se seleccionan las herramientas, tecnologías y proceso de desarrollo que guiará el ciclo de vida del producto.

Capítulo 2: Análisis y diseño.

En el capítulo se describe la propuesta de solución al problema de la investigación. Abarca lo referente al modelo de dominio. Se realiza el levantamiento de los requisitos funcionales y no funcionales, los diagramas de clases del diseño por medio de los cuales se representan la estructura estática del sistema y

los patrones de arquitectura y diseño utilizados.

Capítulo 3: Implementación y pruebas a la solución propuesta.

En el capítulo se aborda lo referente a la implementación de la solución propuesta. Se diseñan y aplican las pruebas para comprobar el correcto funcionamiento de la aplicación.

Capítulo I: Fundamentación teórica del desarrollo del sistema para la elaboración de cuestionarios en investigaciones sociales

1.1. Introducción

En este capítulo se presentan los conceptos fundamentales que permitan elaborar el marco teórico referencial para la presente investigación. Se analizan los fundamentos y principios de varios autores relacionados con el diseño, elaboración y validación de cuestionarios, asociado a investigaciones sociales. Además de la revisión de las herramientas, tecnologías y metodologías necesarias para resolver el problema de investigación.

1.2. Conceptos asociados al sistema para la elaboración de cuestionarios

El cuestionario es uno de los instrumentos que se utiliza para la obtención de la información en las investigaciones, por lo que el diseño y correcta confección es esencial para alcanzar los objetivos propuestos. En el cuestionario se formulan una o más preguntas que permiten medir una o más variables, debe tener un diseño metodológico estructurado de acuerdo a la finalidad de la encuesta, a fin de captar la atención de los encuestados y obtener resultados muy cercanos a la realidad.

Para el desarrollo de la presente investigación se exponen los siguientes conceptos del negocio para un mejor entendimiento de los procesos:

- **Encuesta:**

La encuesta es un método de investigación empleado en estudios cuantitativos y cualitativos, que estudia las características generales del grupo de involucrados en ella. Además, analiza las interacciones y comunicaciones entre las personas o entre las instituciones que conforman una población, independientemente de la cantidad de sujetos que presenten características similares, es decir, estudia la diversidad y no la frecuencia. Comprende, a su vez, indagaciones que emplean la entrevista o el cuestionario como técnica e instrumento, respectivamente, para la recogida de información (Huarcaya, 2020).

- **Cuestionario:**

Es el instrumento estandarizado que empleamos para la recogida de datos durante el trabajo de campo de algunas investigaciones cuantitativas, fundamentalmente, las que se llevan a cabo con metodologías de encuestas. Se podría decir que es la herramienta que permite al científico social plantear un conjunto de preguntas para recoger información estructurada sobre una muestra de personas, empleando el tratamiento cuantitativo y agregado de las respuestas para describir a la población a la que pertenecen y/o contrastar estadísticamente algunas relaciones entre medidas de su interés (Meneses, 2016).

- **Pregunta:**

Interrogación que se hace para que alguien responda lo que sabe de un asunto u otra cosa (RAE, 2021).

- **Respuesta:**

Dar solución a una pregunta, duda o dificultad (RAE, 2021).

- **Ítem:**

El ítem constituye la unidad básica de información de un instrumento de evaluación, y generalmente consta de una pregunta y de una respuesta cerrada o abierta, con el objetivo de obtener una conducta observable y claramente interpretable de la persona que se le aplique el instrumento (cuestionarios, pruebas, diagnósticos) en el que se encuentra (De Las Heras, 2019).

- **Fiabilidad:**

Es el grado en que un instrumento mide con precisión, sin error. Indica la condición del instrumento de ser fiable, es decir, de ser capaz de ofrecer en su empleo repetido resultados veraces y constantes en condiciones similares de medición (Arribas, 2004).

- **Validez:**

Es el grado en que un instrumento de medida mide aquello que realmente pretende medir o sirve para el propósito para el que ha sido construido (M. Arribas, 2004).

- **Motivos de consulta:**

El motivo de consulta surge de la historia de la problemática, la duración e intensidad de la misma, las ideas que tiene sobre el problema, el impacto que ha tenido en la vida de la sociedad, las expectativas asociadas, los intentos de solución y los tratamientos previos (Quesada,2004).

1.3. Estudio de sistemas homólogos

A continuación, se analizan algunos sistemas informáticos que guardan relación con la propuesta de solución que se implementa.

1.3.1 Lime Survey

Lime Survey (anteriormente **PHPSurveyor**) es una aplicación de software libre para desarrollar, publicar y coleccionar preguntas y respuestas de encuestas en línea que luego de ser creadas, pueden ser publicadas (mostrada como preguntas individuales, agrupadas, o todas en una sola página).Permiten establecer condiciones para la publicación de los cuestionarios, también consiguen variar el aspecto de las encuestas a través de un sistema de plantillas, y puede realizar análisis estadísticos básicos de los resultados(« *LimeSurvey* » [21 de junio 2021]).

Tecnologías:

- PHP
- MySQL
- PostgreSQL
- MSSQL

Licencia que utiliza:

- GNU General Public License.

1.3.2 RotatorSurvey

RotatorSurvey es una plataforma integrada de herramientas amigables de software dirigida a organizaciones de Investigación de Mercados, Investigación Social y Opinión Pública que realizan procesamiento de encuestas offline y online, encuestas en papel (PAPI), entrevistas telefónicas (CATI), entrevistas móviles (CAPI), así como en el análisis multidimensional de los datos (OLAP).

Es una poderosa aplicación de escritorio que permite diseñar, procesar y analizar encuesta aplicando cualquier método de captura de información, incluso cuestionarios en línea auto aplicados. RotatorSurvey cuenta con un producto base llamado "Rotator Modelador de Estudios" y varios módulos adicionales, los cuales pueden ser agregados para extender su funcionalidad. La suite básica se compone de "Rotator Modelador de Estudios" y de "Rotator Analizador OLAP de Estudios" (RotatorSurvey [21 de junio de 2021]).

Tecnologías:

- No se puede definir puesto que no existe documentación al respecto, además, no se puede acceder al código porque solo se tiene un archivo binario de la misma.

Licencia que utiliza:

- Starter Edition.

1.3.3 Moodle

Moodle es un *Control Management System* o Sistema de Control y Administración de Contenidos, pero especializado en educación a distancia. Moodle es una plataforma que aprovechando las potencialidades de la red, permite la creación de sistemas de enseñanzas completas en líneas. En él se puede montar desde un simple curso hasta una universidad virtual al 100% (EcuRed, 2021). Moodle está compuesta por varios módulos entre los que se encuentra el módulo de Cuestionario, en el que:

- Se pueden definir una base de datos de preguntas que podrán ser reutilizadas en diferentes cuestionarios, las preguntas pueden ser almacenadas en categorías de fácil acceso, y estas categorías pueden ser "publicadas" para hacerlas accesibles desde cualquier curso del sitio. Los cuestionarios se califican automáticamente. Puede elegirse un límite de tiempo, a partir del cual no estarán disponibles.
- Se puede definir cuántas veces pueden ser resueltos los cuestionarios. Si se mostrarán o no las respuestas correctas y los comentarios.
- Las preguntas pueden crearse en HTML y con imágenes, además, pueden importarse desde archivos de texto externos.
- Las preguntas pueden tener diferentes métricas y tipos de captura: tipo test con diferentes opciones, preguntas abiertas o aquellas que permiten al alumno comprobar si su respuesta es correcta.

Tecnologías:

- PHP 5.6. o superior (7.1 a partir de la versión 3.7).
- MySQL 5.5. o superior.
- Apache 2.X o superior.

Licencia que utiliza:

- CreativeCommons

1.3.4 Banco de preguntas para la generación de encuestas

En el Centro de Estudio Sociopolítico y de Opinión, existe un precedente de una aplicación cuyo objetivo fundamental era almacenar en una base de datos, todas las preguntas incluidas en las encuestas aplicadas a la población en diferentes investigaciones desarrolladas por el Centro.

Con el Banco de Preguntas para la Generación de Encuestas se pretendía insertar y filtrar preguntas a partir de un código establecido según alguna de las esferas: política, económica, social o cultural; modificar e incorporar preguntas a una encuesta, lo que contribuiría a agilizar el trabajo de los investigadores en la confección de las encuestas para una determinada investigación.

Esta aplicación no alcanzó los objetivos esperados, porque carecía de una documentación efectiva, se extravió la clave de acceso que permitía a los usuarios autorizados acceder a la interfaz de usuario principal. No cumplía con los parámetros de funcionalidad y usabilidad que requieren los softwares para su correcto funcionamiento. A esto podemos agregar, que solo se cuenta con un archivo binario de este banco de preguntas, esto imposibilita la visualización del código fuente para su reutilización y mantenimiento.

1.3.5 Análisis realizado a los sistemas informáticos para la elaboración de cuestionarios y sondeos

El estudio de los sistemas antes descritos muestra que no es viable el uso de alguna de estas herramientas en el desarrollo de la presente investigación, puesto que presentan indistintamente alguno de los siguientes inconvenientes:

- No se ajustan a las características necesarias para las investigaciones sociales.
- Pueden ser difíciles de usar para personas poco familiarizadas con las TIC.
- En algunos casos pueden cobrar por la adquisición y renovación de las licencias de uso de los softwares.
- Permiten acceder gratuitamente solo a una parte del software, porque presentan módulos de gran importancia por los que hay que pagar para su uso.
- No brindan la posibilidad de acceder a experiencias de diseños de cuestionarios anteriores.
- Carecen de documentación fiable para su utilización.
- Algunas por su condición de software de escritorio su acceso se limita al ordenador donde están instaladas y al sistema operativo que utilice dicho ordenador.

- Presentan una navegabilidad compleja para la elaboración de un cuestionario.

Aunque se detectaron estos percances, estas plataformas aportaron un grupo de características que se tomarán como referencia de buenas prácticas en el desarrollo de la solución propuesta con el objetivo de mejorarla tarea de elaboración de cuestionarios como son:

- El reconocimiento de patrones de diseños para la gestión de preguntas y respuestas de los cuestionarios.
- El uso de una base de datos donde se encontrará almacenada toda la información de las operaciones que se ejecutan.
- Permiten obtener comunicación con otras aplicaciones. Esto resulta necesario al momento de exportar los cuestionarios, por ejemplo, a Office o cualquier otro programa.
- Se identificaron requisitos funcionales y no funcionales, así como la necesidad de la accesibilidad y compatibilidad desde cualquier navegador web, independiente del sistema operativo utilizado.

1.4. Tendencias en el desarrollo de aplicaciones web

Podemos entender como desarrollo web a las acciones de diseñar, construir y mantener plataformas (aplicaciones, sitios, portales) web para entidades, organizaciones o empresas. Este proceso está compuesto por dos capas, la capa de presentación o lado del cliente (Frontend) y la capa de acceso a dato (Backend).

El *frontend* trabaja la interfaz visual, y permite que el usuario pueda interactuar con la aplicación o sistema. Está orientado al lenguaje de marcas y al lenguaje de programación web de ejecución en equipos clientes(Ibarra et al. 2021).

Se denomina *backend* a la capa de acceso a los datos de un software que no es accesible para el usuario final. Además, esta capa contiene toda la lógica de la aplicación que maneja los datos. Cabe señalar que los datos de una aplicación se encuentran almacenados en una base de datos dentro de un servidor(Ibarra et al. 2021).

En el contexto del desarrollo web, el *Frontend* lo conforman todas aquellas tecnologías que se ejecutan del lado del cliente, o sea, que corren en el navegador web, generalizándose en tres lenguajes HTML, CSS y JavaScript. En el *backend*, compuesto por las tecnologías que utilizan los servidores encontramos lenguajes de gran uso actualmente como son: Java, C#, PHP y JavaScript; además, es el Backend el que interactúa con las bases de datos, que entre los gestores más conocidos podemos mencionar Oracle, MySQL, PostgreSQL, MongoDB, entre otros.

La globalización y la constante evolución de los sistemas han permitido que la accesibilidad a las diversas herramientas informáticas que utilizamos en nuestras labores pueda llevarse a cabo tan solo con una computadora, generando así el desarrollo de numerosos lenguajes con múltiples opciones para cada usuario y de acuerdo a sus preferencias; así mismo se han creado varios framework (herramientas de trabajo) que facilitan y reducen el tiempo en la ejecución de las tareas. Entre los *framework* más conocidos se hallan Angular, Symfony, Node JS, Vue JS, Quásar, entre muchos otros de gran uso en nuestros días. Otra tendencia

es el uso de complementos y bibliotecas de JavaScript, JQuery (que aporta facilidades para efectos, galerías de fotos dinámicas, calendarios, entre otras prestaciones) y Bootstrap que permite adaptar el diseño a varios dispositivos y se le asocia con estilos o plantillas predefinidas fáciles de configurar (Neftalí, 2019).

En el desarrollo de la presente investigación la tendencia utilizada son las SPAs (*Single Page Application*) aplicaciones de una sola página por sus siglas en inglés, en las cuales está todo el contenido en una sola página, es decir, carga tan solo un archivo HTML y todo se produce dentro de este único archivo. De esta manera se puede ofrecer una experiencia más fluida, más rápida, es decir, todo el código HTML, JavaScript y CSS se carga una sola vez, puedes navegar entre los diferentes apartados de la página y el contenido ya estará precargado de antemano, es por esto que es tan rápido. En algunos casos también lo que se hace es cargar estos contenidos dinámicamente si se requieren, pero no tiene que cargar toda la página de nuevo, tan solo los nuevos contenidos («¿Qué es una web SPA? - *Single Page Application*» [20 de noviembre de 2021]).

1.5. Metodología de desarrollo de software

Las metodologías de desarrollo de software se utilizan para administrar o gestionar un proyecto de manera organizada y sistemática para que resulte exitoso. Ésta queda comprendida desde la fase en que se idea el proyecto hasta que se despliega (T. R. Sánchez, 2015).

Para el desarrollo de esta investigación se decide utilizar una metodología híbrida con una base de documentación sólida que permita en un momento dado conocer y retomar lo que se hizo, sin incrementar los tiempos de desarrollo, con entregas parciales que representen versiones del software funcionando, que puedan ser adaptables a los cambios. La metodología propuesta se basa en la integración de un marco de trabajo ágil (Scrum) con un proceso de desarrollo tradicional (RUP), teniendo en cuenta sus fases y artefactos principales, lo que conlleva a definir las fases necesarias durante el proceso de desarrollo de software y una base de documentación que pueda realizarse a la par o posterior al desarrollo, que no implique sacrificar tiempos de entrega ágil, pero sí un esfuerzo mínimo de inversión simultánea para tener un proceso más completo que permita tener software ágil, actualizado y documentado (Santos-Romero, 2020).

Este proceso de desarrollo tiene cuatro fases:

Fase de Iniciación. Se comienza definiendo los requisitos que representarán el Documento de Requerimientos Funcionales para un determinado módulo entregable (Santos-Romero, 2020).

En esta fase se genera el documento de modelo de proceso de negocio y requerimientos funcionales.

Fase de elaboración. Basados en los requisitos del entregable se documenta esta fase, generando los diagramas de secuencia (Santos-Romero, 2020).

En esta fase también se genera el documento modelo de casos de uso, documento modelo de datos, documento de diseño: diagrama de clases, de secuencia y de arquitectura, lista módulos funcionales.

Fase de Construcción. En esta fase se procede ya a la codificación de cada módulo, realizando la documentación del código, en el cual se pueden seguir estándares de calidad. Así como, un plan de

pruebas del software desarrollado, de tal manera que cuando llegue el día de la reunión de revisión y retrospectiva, el o los módulos estén libres de errores técnicos y probablemente libres de errores funcionales (Santos-Romero, 2020).

Fase de Transición. En esta fase se realiza la documentación a través del manual de implementación y diagrama de despliegue. El diagrama de despliegue es la configuración de los módulos de procesamiento en tiempo de ejecución y los componentes que interactúan entre sí. Muestra qué elementos de software se implementan mediante qué elementos de hardware, ilustran el procesamiento en tiempo de ejecución para el hardware y proporcionar una vista de la topología del sistema (Santos-Romero, 2020).

La metodología seleccionada es la utilizada por los desarrolladores del CESPO. Esta, por solo presentar tres roles (Propietario del producto, Scrum Master, Equipo de desarrollo) es ideal para equipos de desarrollo pequeños donde un integrante del equipo puede recorrer cada uno de los roles, brinda una buena documentación del producto con la que el cliente puede comprender de manera sencilla lo que se pretende lograr con cada módulo del sistema puesto que este participa en cada *sprint*(iteración) como otro miembro del equipo.

1.6. Herramientas y tecnologías

La aparición y constante desarrollo de las aplicaciones web ha propiciado el surgimiento de herramientas que ayudan al diseño, construcción, configuración y su mantenimiento, usualmente con buenas prestaciones para instituciones y empresas de mediano y gran tamaño.

1.6.1. Lenguajes de programación utilizados

Los lenguajes de programación son formas que permiten a las personas comunicarse con las computadoras para desarrollar programas de software, aplicaciones, páginas webs, scripts u otros conjuntos de instrucciones para que sean ejecutadas por los ordenadores con objetivos específicos.

HTML

El acrónimo significa lenguaje de marcado de hipertexto, se emplea para el desarrollo de aplicaciones web. Describe cosas como la sección del documento, párrafos, listas, tablas, archivos multimedia, y enlaces de hipertextos. Todos estos existen dentro de partes de un documento llamadas elemento elementos, que conforman subregiones rectangulares del navegador. HTML es un lenguaje de marcado con gramática bastante simple que el navegador puede entender. El navegador utiliza HTML para formatear y mostrar documentos. HTML5 es un estándar actual que ha sido diseñado para asegurar la compatibilidad en términos sobre la publicación sobre la Web (Morrison, 2020).

El lenguaje HTML es utilizado en la propuesta de solución por ser parte fundamental en las aplicaciones web, mediante él se pueden desarrollar e integrar recursos como textos, imágenes, animaciones y otras que facilitan la construcción de la interfaz gráfica de nuestra solución.

CSS

Las Hojas de Estilo en Cascada (CSS por sus siglas en inglés) es un concepto que se utiliza en la informática

para referirse a un lenguaje empleado en el diseño. El diseño del CSS posibilita establecer una separación entre el contenido y la forma de presentación del documento. Así se puede lograr que muchos documentos HTML compartan la apariencia, utilizando una única hoja de estilo para todos, esto permite evitar la repetición de código en la estructura (J. Salazar, 2020).

Con el uso de CSS se logra la separación de del contenido en la interfaz de nuestra aplicación, permitiendo alcanzar el diseño de la interfaz de nuestro sistema

JavaScript

Consiste en un lenguaje de programación interpretado, que habitualmente se utiliza en sitios web para ejecutar acciones en el lado del cliente. JavaScript permite, en una página web, crear elementos como cuadros de diálogo, recoger información entrada por el usuario y enviarla al servidor para ser procesada. Al ser un lenguaje abierto y modificable en elementos como la sintaxis, se permite la creación de variantes y dialectos (J. Salazar, 2020).

Basado en el estándar ECMAScript [ECMASC]. Permite acceder y manipular el árbol DOM (Modelo de Objetos del Documento) de los documentos HTML, para generar contenidos de manera dinámica, realizar peticiones HTTP asíncronas, etc. Soporta estructuras de programación similares a lenguajes como C o Java, con la principal diferencia que el ámbito en el que se define una variable no es el bloque en el que ha sido definida, sino la función en la que ha sido declarada (J. C. E. Pereda, 2020).

En nuestra aplicación nos permite aportarle el movimiento y dinamismo necesario para la ejecución y alcance de los objetivos previstos para esta, tanto del lado del servidor como en la parte del cliente.

PostgreSQL v10.0.0

PostgreSQL es un sistema de gestión de bases de datos orientado a objetos y relacional (ORDBMS) que hace énfasis en la extensibilidad y conformidad con los estándares. Está liberado bajo la licencia de código abierto PostgreSQL, además, se basa en una arquitectura cliente-servidor. El programa servidor se el movimiento y entre los muchos programas cliente tenemos, pgaccess y pgsadmin (como clientes gráficos) y psql (cliente en modo texto) (J. D. Chávez, 2020).

Es un gestor de bases de datos relacional de alta disponibilidad, con capacidad de funcionar en servidores de manera estable, esto lo hace robusto satisfaciendo una de los principales requisitos necesarios en la entidad.

1.7. Lenguaje de modelado

Los lenguajes de modelado están compuestos por vistas, diagramas, elementos de modelo, los símbolos utilizados en los modelos y un conjunto de mecanismos generales o reglas que indican cómo utilizar los elementos. Las reglas son sintácticas, semánticas y pragmáticas.

UML (Lenguaje de Modelado Unificado)

El UML constituye el estándar para *Object Management Group*(OMG), el cual permite especificar, visualizar y documentar modelos de programas que puedan cumplir requisitos, e incluso permite realizar el modelo

de negocios con sistemas de negocios no informáticos (*Object Management Group Inc.*,2019).

Permite modelar componentes estáticos de un programa y el comportamiento dinámico de sus partes mientras funciona, como diagramas de clase, diagramas de secuencias y estados. Establece un patrón para los sistemas orientados a objetos, el cual por medio de un lenguaje visual permite una mejor descripción y visualización de los sistemas estructurales que tienen su base en los programas. El UML constituye una forma gráfica de representación de sistemas de software, usado en toda clase de programas, pero no constituyen método de desarrollo de software (W. D. Muentes, 2020).

Es usado en nuestra solución porque permite modelar el sistema que se implementa, contribuyendo así a reducir el problema que se está estudiando, centrándonos en un solo aspecto a la vez.

BPMN (Notación de Modelado de Procesos de Negocio)

La notación de modelo de proceso de negocio (BPMN) es un estándar desarrollado por OMG que proporciona a las organizaciones la capacidad de comprender sus procedimientos empresariales internos en una notación gráfica de una manera estándar. Ésta cuenta con un conjunto de elementos de notación para que los usuarios puedan reconocer fácilmente los tipos básicos de elementos y comprender el diagrama BPMN. Dicha notación, presenta elementos básicos y patrones que simulan situaciones que comúnmente ocurren en cualquier modelo de proceso de negocio para una mejor comprensión del asunto (J. D. Suárez, 2020).

Es utilizada en nuestra solución porque permite visualizar oportunidades de mejoras, aportando un conocimiento más profundo de las pruebas y los resultados, imprimiendo un fuerte comienzo para la optimización de procesos.

1.8. Herramientas de Modelado

Las herramientas de modelado son aplicación o software que utilizan para facilitar el trabajo con todo tipo de lenguaje de modelado.

Herramienta CASE

Las Herramientas CASE (Computer Aided Software Engineering en español Ingeniería de Software Asistida por Ordenador) son aplicaciones informáticas hechas para aumentar la productividad en el desarrollo de software reduciendo su coste en términos de tiempo y de dinero. Estas proporcionan ayuda a los analistas, diseñadores y desarrolladores en todos los aspectos del ciclo de vida de desarrollo del software (Rumbaugh, 2017).

1.8.1. Visual Paradigm10.0

Visual Paradigm es una herramienta CASE multiplataforma que ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite importar y exportar ficheros XML, así como la generación de código para Java y exportación HTML. Es una herramienta fácil de usar que soporta ingeniería directa e inversa (código a modelo, código a diagrama),

diagramas de flujo de datos y generación de código e informes (Rumbaugh, 2017).

Es usada en esta investigación porque brinda un conjunto de ayudas para el desarrollo de software en cuanto a planificación, pasando por análisis y diseño, capaz de generar código fuente de los programas y su documentación.

1.9. Entorno de Desarrollo Integrado

Un Entorno de Desarrollo Integrado (IDE) es una aplicación que de la misma forma es un entorno de programación. En el mayor de los casos los IDEs están compuestos por un editor de código, un compilador, un depurador y en ocasiones un constructor de interfaz gráfica de usuario.

1.12.1. Visual Studio Code v1.56.2

Visual Studio Code es un editor de código fuente ligero, pero de gran potencia que se ejecuta en las computadoras con sistemas operativos Windows, MacOS o Linux. Viene con soporte integrado para JavaScript, TypeScript y Node.js, y con un buen ecosistema de extensiones para otros lenguajes («Visual Studio Code - *Code Editing. Redefined*» [26 de junio de 2021]).

Es seleccionado para el desarrollo de la presente investigación porque permite configurar la interfaz a nuestro gusto para poder tener un código más visible, propicia el acceso rápido a las carpetas del proyecto y a una terminal con los detalles del problema o error. Su curva de aprendizaje es suave y existe una gran documentación al respecto.

1.10. Entorno de ejecución

Node.js v14.16.1

Node.js es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome, está diseñado para generar aplicaciones web escalables de manera óptima y agilizada. Node.js fue desarrollado por Google para su uso en el tan popular navegador de Chrome. Por otra parte, es una librería y entorno de ejecución que maneja eventos de manera asíncrona y tiene un buen rendimiento debido a que la V8 compila JavaScript en código máquina nativo en lugar de interpretarlo (Díaz y Intriago, 2018).

1.11. Framework

Un *framework* es un conjunto estandarizado de conceptos, prácticas criterios para hacer frente a diferentes dificultades en interfaces, seguridad, rendimiento, etc. además de tener como principal objetivo una estructura común, de modo que facilite a los programadores reutilizar el código en su gran mayoría (Neftalí, 2019).

1.12. Express.js v6.14.12

Express es una infraestructura de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles. Tiene una buena cantidad de métodos de programa de utilidad para desarrollo web y HTTP, es rápido y sencillo. Cabe destacar que proporciona un excelente rendimiento y se integra muy bien con node.js (Díaz y Intriago, 2018).

Vue.jsv3.0.1

Vue.js es un framework progresivo que se encuentra dividido en diferentes librerías bien acotadas que tienen una responsabilidad específica, que permite al desarrollador ir incluyendo los módulos según las necesidades del contexto en que se encuentre. Además, permite el desarrollo de componentes de UI (Interfaz de Usuario) por medio de JavaScript (Villagómez, 2018).

Quasarv2.0.0

Quasar es un entorno de desarrollo basado en Vue.js que permite a los desarrolladores crear sitios web y aplicaciones (SPA, SSR, WPA, extensiones de navegador, aplicaciones móviles y aplicaciones de escritorio). Quasar se centra en facilitar todas las herramientas para crear aplicaciones visuales y reducir el coste del desarrollo permitiendo, con un mismo código, crear diferentes tipos de aplicaciones. Incluso proporciona la estructura necesaria para que nuevas aplicaciones creadas en él funcionen. Igualmente se puede utilizar para la elaboración de partes de la aplicación, ya sean componentes, páginas, layouts o stores (Roma, 2020).

1.13. Servidor web

Un servidor web es una aplicación que responde a solicitudes provenientes de navegadores web, proporcionando recursos solicitados a través del protocolo HTTP o de forma segura a través del protocolo HTTPS (Mejía, González y España, 2018).

Nginxv1.20.1

Nginx es un servidor web de código abierto de proxy inverso, que acelera la entrega de contenido y aplicaciones, mejora la seguridad, facilita la disponibilidad y la escalabilidad para las aplicaciones web (Aguilera, Rodríguez y Gonzales, 2021).

Es seleccionado para esta investigación por su alto rendimiento, estabilidad, amplio conjunto de funciones, configuración simple y bajo consumo de recursos.

1.14. Navegador web

Un navegador web (en inglés, web browser) es un software, que permite el acceso a la Web, interpretando la información de distintos tipos de archivos y sitios web para que estos puedan ser visualizados por el usuario (Danny Murillo, 2017).

Firefoxv89.0

Firefox es un navegador web de código abierto y software libre desarrollado por la Fundación Mozilla, este se basa en el poderoso motor de búsqueda Gecko, el cual implementa estándares web siempre actualizados («*Learn About Mozilla*» [2 de julio 2021]).

Este navegador es escogido por ser un navegador libre, multiplataforma y por la amplia comunidad que tiene para su desarrollo y mantenimiento. Es rápido con aplicaciones Web complejas y presenta una efectiva protección contra el spyware y otros tipos de malware, bloqueo asegurado contra pop-up y otras formas de publicidad comunes en la web.

1.15. Herramientas para pruebas de software

Las herramientas para pruebas de software ayudan a gestionar el proceso de pruebas, algunas sirven para gestionar pruebas, resultados de pruebas, datos requisitos, incidencias, defectos, etc., además, permiten elaborar informes y monitorizar la ejecución de pruebas.

JMeter3.0

Es una herramienta de código abierto desarrollada en Java que permite realizar pruebas de rendimiento y pruebas funcionales. Originalmente se diseñó para pruebas de aplicaciones web, pero desde entonces ha aumentado su funcionalidad a otro tipo de pruebas (EcuRed, 2021). Sus funcionalidades se pueden resumir en tres:

- Diseñar un testplan, esto es generar un archivo.jmx.
- Ejecutar un testplan.
- Ver de distintas formas los resultados de la ejecución de un testplan.

Postmanv8.1

Postman es una herramienta para la realización de pruebas de servicios web. Ofrece una interfaz de usuario para realizar peticiones hacia una API, evita la molestia de escribir íntegramente el código al diseccionar la funcionalidad de una API.

Se utiliza Postman en esta investigación para realizar peticiones HTTP a la API REST creada y de esta manera validar el aseguramiento del servicio web y el acceso no autorizado al mismo.

1.16. Conclusiones Parciales

- El estudio realizado permitió identificar que la forma de confección de cuestionarios que se aplican en el CESPO no está acorde con la tendencia del uso de tecnologías y métodos con los que se cuenta en la actualidad. Se pudo identificar que se hace necesaria la creación de una aplicación web que responda a las exigencias de tiempo que poseen las investigaciones desarrolladas en el Centro.
- En este capítulo se realizó un análisis del estudio del estado del arte referente a la elaboración de cuestionarios, lo cual permitió identificar los elementos fundamentales para el desarrollo de la presente investigación.
- Se realizó un estudio de un proceso de desarrollo de software híbrido, para la modelación y la implementación del componente el cual permitió definir las etapas en las que se desglosa su proceso de desarrollo.

Capítulo II: Descripción del desarrollo del sistema para la elaboración de cuestionarios en las investigaciones sociales

2.1 Introducción

En el presente capítulo se describen las principales características que debe presentar la solución propuesta. Esta solución es basada en la definición de los requisitos y la arquitectura. Se presentan los principales artefactos propuestos en la fase de modelación en la metodología de desarrollo a aplicar (en nuestro caso proceso de desarrollo) y se fundamentan los patrones de diseño empleados en su elaboración.

2.2 Propuesta de Solución

El sistema para la elaboración de cuestionarios en las investigaciones sociales (o banco de preguntas) permitirá al CESPO contar con una herramienta centralizada de fácil acceso y filtrado que agrupe las preguntas y respuestas que se utilizan en investigaciones por temáticas y clasificaciones para la elaboración de los cuestionarios, lo cual disminuirá el tiempo de trabajo y aumentará la eficiencia de los investigadores para la elaboración de estos. Además, reducirá la posibilidad de inserción de errores en los cuestionarios puesto que ya no se desarrollarán de forma manual y brindará a los nuevos especialistas que se incorporen a la entidad acceso a las experiencias de investigaciones de años anteriores.

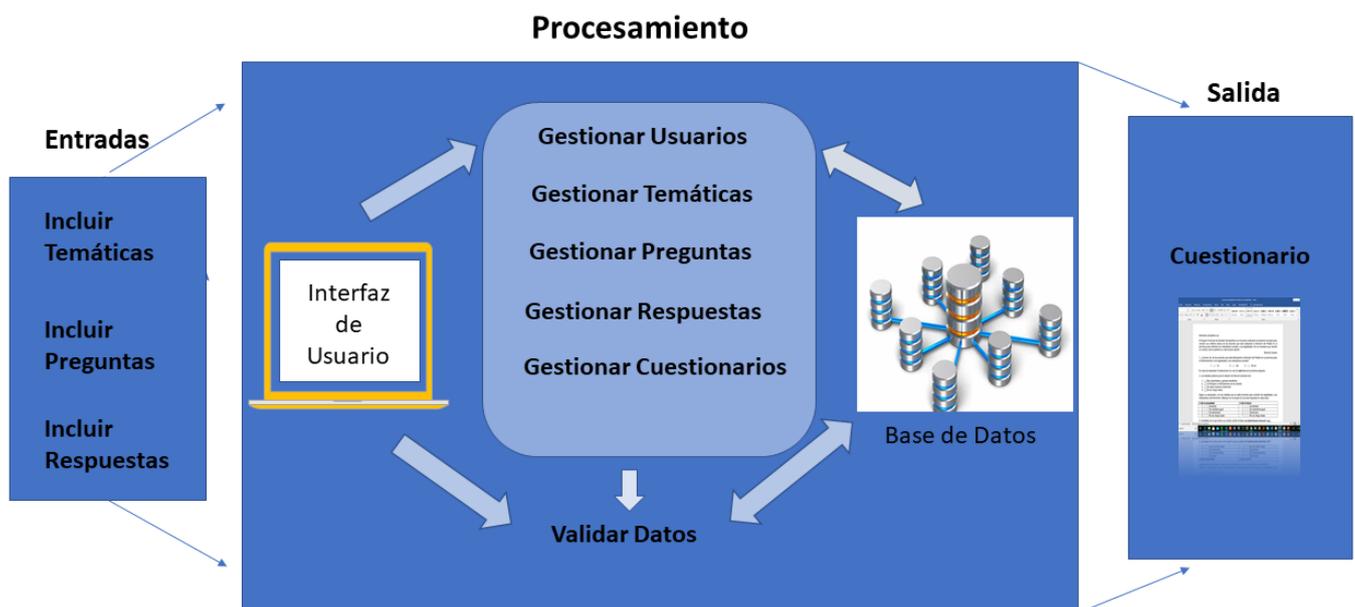


Figura2.2-1 Representación visual del sistema

2.3 Modelado del proceso de negocio

El modelado del proceso de negocio proporciona una notación gráfica estándar que es fácilmente legible y entendible por parte de todos los involucrados e interesados del negocio, ayuda a mejorar la calidad de los procesos de negocio, permite reducir costos en tiempo y recursos, y ayuda identificar las tareas o

procesos que se pueden automatizar en la organización (Gartner Group, 2011).

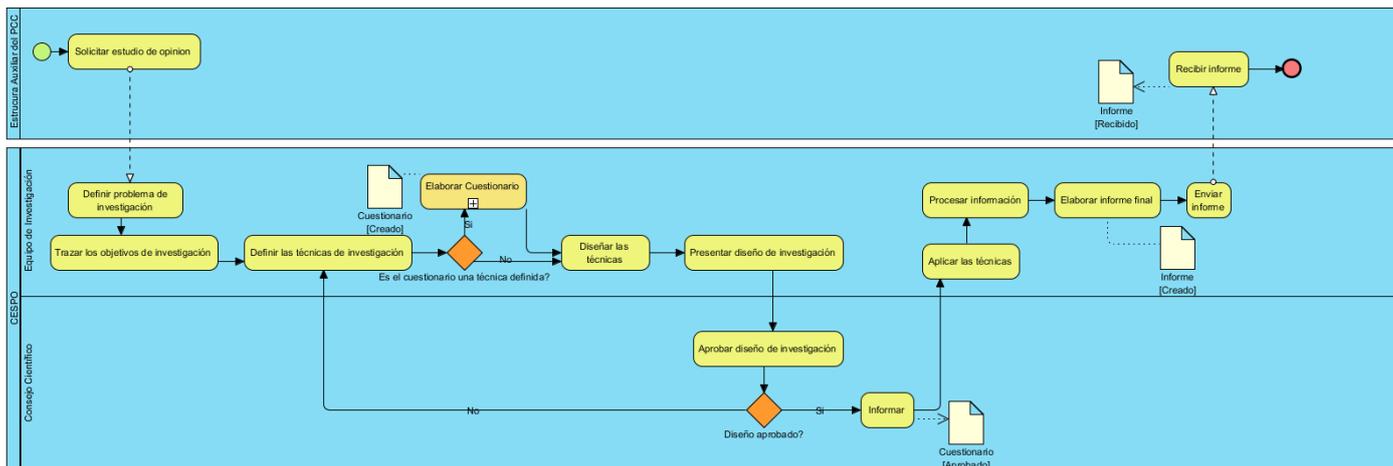


Figura 2.3-1 Diagrama del proceso de negocio de estudio de opinión

Descripción del diagrama de proceso de negocio de estudio de opinión

1. El proceso inicia cuando el cliente (Estructura auxiliar del PCC - secretariado) solicita el estudio de opinión de algún sector de interés.
2. El equipo de investigación del CESPO recibe la solicitud y define el problema científico.
3. El equipo de investigación traza los objetivos que se persiguen en la investigación donde se validan los indicadores a seguir.
4. El equipo de investigación define las técnicas que se utilizarán para el estudio de opinión (investigación).
5. Si una de las técnicas es el cuestionario se desata un subproceso para la elaboración del cuestionario (proceso que se va a automatizar).
 - 5.1 Se definen los indicadores que se abarcarán en el cuestionario.
 - 5.2 Se realiza el diseño del cuestionario.
 - 5.3 Se decide el o los investigadores encargados de elaborar el cuestionario.
 - 5.4 Se elabora el cuestionario.
6. Si no, se diseñan las técnicas.
7. Se presentan las técnicas al consejo científico para su aprobación.
8. Si el consejo científico no aprueba las técnicas diseñadas entonces se vuelven a definir las técnicas (punto 4).
9. Si el consejo científico aprueba las técnicas diseñadas entonces informa al equipo de investigación.

10. El equipo de investigación aplica las técnicas.

11. El equipo de investigación procesa la información que arroja el resultado de la aplicación de las técnicas.

12. El equipo de investigación elabora el informe final y lo envía al cliente.

13. El cliente recibe el informe final.

2.4 Levantamiento de requisitos

Para un sistema los requisitos son descripciones detalladas de las funciones, servicios y las restricciones operacionales de software. El documento de requisitos del sistema define con exactitud lo que se implementará. A menudo, los requisitos de software se clasifican como requisitos funcionales o requisitos no funcionales (Sommerville, 2011). Para poder cumplir con la implementación de la propuesta se llevó a cabo formalmente el levantamiento de requisitos. Para ello los requisitos funcionales y no funcionales quedaron expresamente descritos en el documento, tras haber realizado varias entrevistas al cliente y haber consultado el criterio de expertos respectivamente.

2.5 Requisitos funcionales

Los requisitos funcionales para un sistema refieren lo que el sistema debe hacer, ellos dependen del tipo de software que se esté desarrollando, de los usuarios esperados del software y del enfoque general que adopta la organización cuando estos se escriben. Estos varían desde requisitos generales que cubren lo que tiene que hacer el sistema, hasta requisitos muy específicos que reflejan maneras locales de trabajar o los sistemas existentes de una organización (Sommerville, 2011). A continuación, se muestra un listado de los requisitos identificados y que el sistema deberá cumplir.

Tabla 2.5-1 Especificación de requisitos funciones del sistema

ID	Nombre	Descripción	Prioridad	Complejidad
RF_1	Adicionar pregunta	Se guarda una pregunta.	Alta	Media
RF_2	Modificar pregunta	Se modifican las características de la pregunta.	Alta	Media
RF_3	Eliminar pregunta	Se elimina la pregunta de la base de datos.	Alta	Media
RF_4	Listar pregunta	Se muestran las preguntas almacenadas en la base de datos.	Alta	Media
RF_5	Adicionar respuesta	Se guarda una respuesta.	Alta	Media
RF_6	Modificar	Se modifican las características de la respuesta.	Alta	Media

	respuesta			
RF_7	Eliminar respuesta	Se elimina la respuesta de la base de datos.	Alta	Media
RF_8	Listar respuesta	Se muestran las respuestas almacenadas en la base de datos.	Alta	Media
RF_9	Crear cuestionario	Se insertan preguntas y respuestas a los ítems y se visualiza como va quedando estructurado el cuestionario.	Alta	Alta
RF_10	Descargar cuestionario	Se exporta el cuestionario a un archivo Word en el escritorio.	Alta	Alta
RF_11	Guardar cuestionario	Se guardan las preguntas y respuestas del cuestionario en la base de datos.	Alta	Alta
RF_12	Registrar usuario	Se crea un nuevo usuario.	Alta	Media
RF_13	Autenticar usuario	Se autentica un usuario en el sistema e inicia sesión.	Alta	Alta
RF_14	Cerrar usuario	Sierra sesión un usuario.	Alta	Media
RF_15	Visualizar consejos	Se visualizan consejos para la confección de los cuestionarios.	Media	Baja
RF_16	Visualizar ayuda	Se visualizan consejos para el uso correcto de la aplicación.	Media	Baja
RF_17	Adicionar temática	Se guarda una temática.	Alta	Media
RF_18	Modificar temática	Se modifican las características de la temática.	Alta	Media
RF_19	Eliminar temática	Se elimina la temática de la base de datos.	Alta	Media
RF_20	Listar temática	Se muestran las temáticas almacenadas en la base de datos.	Alta	Media
RF_21	Adicionar rol	Se guarda un rol.	Alta	Media
RF_22	Modificar rol	Se modifican las características del rol.	Alta	Media

RF_23	Eliminar rol	Se elimina el rol de la base de datos.	Alta	Media
RF_24	Listar rol	Se muestran los roles almacenados en la base de datos.	Alta	Media
RF_25	Adicionar tipo de respuesta	Se guarda un tipo de respuesta.	Alta	Media
RF_26	Modificar tipo de respuesta	Se modifican las características del tipo de respuesta.	Alta	Media
RF_27	Eliminar tipo de respuesta	Se elimina el tipo de respuesta de la base de datos.	Alta	Media
RF_28	Listar tipos de respuestas	Se muestran los tipos de respuestas almacenados en la base de datos.	Alta	Media

2.6 Requisitos no funcionales

Los requisitos no funcionales son limitaciones sobre servicios o funciones que ofrece el sistema, incluyen restricciones tanto de temporización y del proceso de desarrollo, como impuestas por los estándares. Estos se suelen aplicar al sistema como un todo, más que a características o a servicios individuales. Pueden relacionarse con propiedades emergentes, como fiabilidad, tiempo de respuesta y uso de almacenamiento. De forma alternativa, pueden definir restricciones sobre la implementación del sistema, como las capacidades de los dispositivos o las representaciones de datos usados en las interfaces con otros sistemas. Los requisitos no funcionales, como el rendimiento, la seguridad o la disponibilidad, especifican o restringen por lo general características del sistema como un todo (Sommerville, 2011). A continuación, se precisan los requisitos no funcionales de la propuesta de solución:

Tabla 2.6-1 Especificación de requisitos no funciones del sistema

ID	Categoría	Descripción
RNF_1	Usabilidad	El sistema debe ser fácil de utilizar para los usuarios que tengan niveles básicos de computación o hayan trabajado con la Web.
RNF_2	Usabilidad	Debe tener una opción de ayuda sobre las principales funcionalidades que brinda el sistema y sus iconos respectivos, para un mejor entendimiento.
RNF_3	Usabilidad	Las operaciones de la aplicación a informatizar serán lo más parecidas posible a los procesos que se realizan actualmente en el CESPO, para así lograr el menor tiempo en cuanto a la comprensión y adaptación del sistema.
RNF_4	Rendimiento	La aplicación debe estar concebida para el consumo mínimo de recursos.
RNF_5	Rendimiento	La aplicación debe estar concebida para que se conecten 100 usuarios como

		máximo, simultáneamente conectados.
RNF_6	Soporte	Se necesita un servidor de bases de datos que soporte grandes volúmenes de datos.
RNF_7	Requerimiento de Ayuda y Documentación	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.
RNF_8	Interfaz	La interfaz de la aplicación a desarrollar debe ser sencilla para reducir el tiempo de capacitación de los usuarios.
RNF_9	Interfaz	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.
RNF_10	Políticos Culturales	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.
RNF_11	Software	El servidor debe tener como requerimientos mínimos de software sistemas operativos Linux o Windows 7 o superior.
RNF_12	Software	El servidor debe tener como requerimientos mínimos de software Servidor Web Nginx v1.20.1
RNF_13	Software	El servidor debe tener como requerimientos mínimos de software PostgreSQL8.3.9 como Sistema Gestor de Base de Datos.
RNF_14	Software	Para las PCs clientes se recomienda Mozilla Firefox 70 o superior, recomendado 89.
RNF_15	Software	Para las PCs clientes soporte para formato Word.
RNF_16	Hardware	Para las PC clientes Procesador: 3.0 GHz o superior.
RNF_17	Hardware	Para las PC clientes RAM: 2Gb o superior.
RNF_18	Hardware	Para los servidores Procesador: Doble Micro a 3.0 GHz o superior.
RNF_19	Hardware	Para los servidores RAM: 4GB o superior.
RNF_20	Hardware	Para los servidores Disco duro: 300 GB o superior.
RNF_21	Hardware	Para los servidores UPS: 1.

RNF_22	Hardware	Para los servidores Tarjeta de Red: 1 Gb.
RNF_23	Hardware	Del lado del servidor de Base de datos Procesador: Doble Micro a 3.0 GHz o superior.
RNF_24	Hardware	Del lado del servidor de Base de datos RAM: 4GB o superior.
RNF_25	Hardware	Del lado del servidor de Base de datos Disco duro: 1 TB o superior.
RNF_26	Hardware	Del lado del servidor de Base de datos UPS: 1.
RNF_27	Hardware	Del lado del servidor de Base de datos Tarjeta de Red: 1 Gb.

Arquitectura del sistema

La arquitectura de software conforma el esqueleto de cualquier sistema, y es la principal responsable de los atributos de calidad del sistema (parte de los requerimientos no funcionales del mismo) la misma identifica los elementos más importantes de un sistema, así como sus relaciones, es decir, da una visión global del sistema (Pressman, 2010).

2.6.1 Patrón modelo vista controlador (MVC)

El patrón de arquitectura de software MVC separa los datos de la aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos, permitiendo mayor independencia, mantenimiento y reutilización (Coba y Fernando, 2016).

Este patrón arquitectónico está compuesto por tres componentes:

- **Modelo:** es la clase que representa las tablas de la base de datos para manejar los datos y controlar todas sus transformaciones. Además, se especifican las relaciones entre clases y las migraciones que expresan cambios en la base de datos. Un ejemplo de modelo es la clase pregunta que permite gestionar la información de la tabla de la base de datos pregunta.
- **Vista:** es el objeto que maneja la presentación visual de los datos representados por el Modelo, creando una interfaz visual donde se muestran los datos a los usuarios. Un ejemplo de vista es la clase pregunta del controlador pregunta, que muestra un listado de preguntas y un formulario para adicionar las preguntas en caso de ser deseado.
- **Controlador:** proporciona acciones de respuesta cuando ocurra una interacción del usuario con la Vista y responde manipulando los datos representados del Modelo. Un ejemplo de controlador es la clase preguntaSimpleController que permite gestionar todos los datos correspondientes a las preguntas del sistema.

Modelo - Vista - Controlador

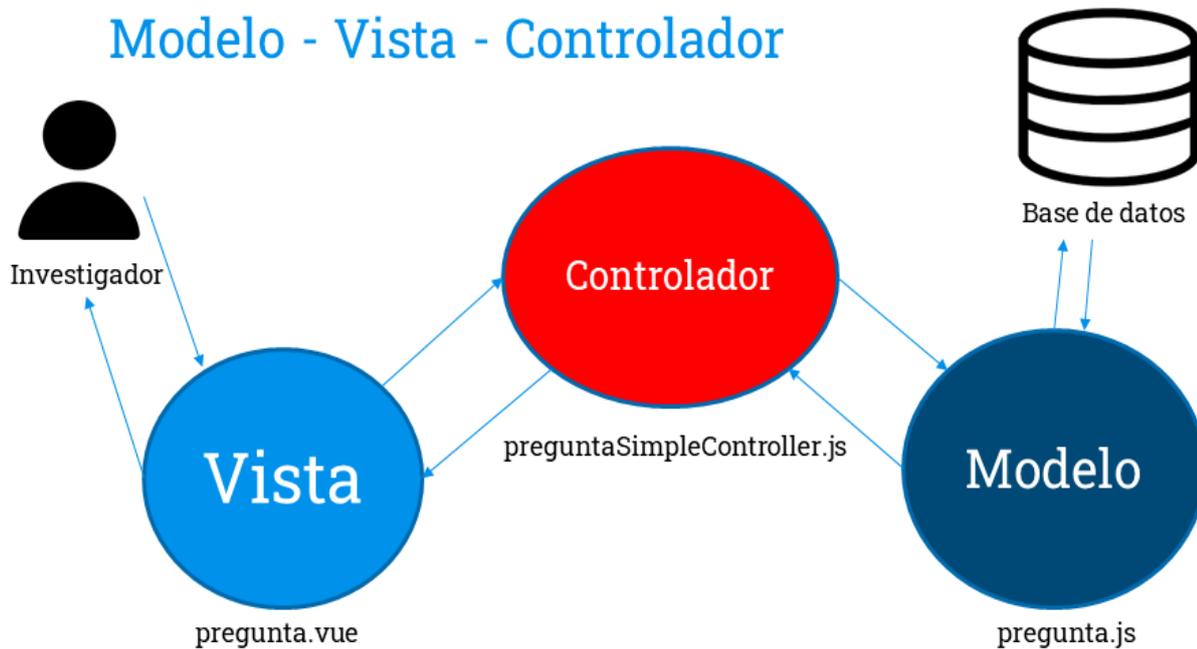


Figura 2.6-1 Arquitectura modelo vista controlador (MVC) aplicada a la solución.

El uso de la arquitectura Modelo Vista Controlador aterrizado en la solución se puede evidenciar si un usuario autenticado con el rol de investigador desea agregar una pregunta, cuando los ficheros **pregunta** con extensión vue y js se encargan de construir la vista, luego el controlador de dicha vista se evidencia con la clase **preguntaSimpleController** que se encuentra en el archivo con el mismo nombre con una extensión js, encargado de recoger los eventos y gestionar las preguntas comunicándose con las clases que se encuentran en el modelo **pregunta**, alguna de otras clases encontradas en el modelo son: respuesta y cuestionario, ubicadas en los archivos con los mismos nombres con extensión js los mismos gestionan el trabajo con la información que se encuentra almacenada en base de datos.

2.7 Diagrama de clases del diseño

Los diagramas de clases del diseño visualizan las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso o de agregación, ya que una clase es una descripción de conjunto de objetos que comparten los mismos atributos, operaciones, métodos, relaciones y semántica; mostrando un conjunto de elementos que son estáticos, como las clases y tipos junto con sus contenidos y relaciones (EcuRed, 2021). A continuación, la figuras 2.7-1 muestra el diseño de clases a implementar para adicionar preguntas.

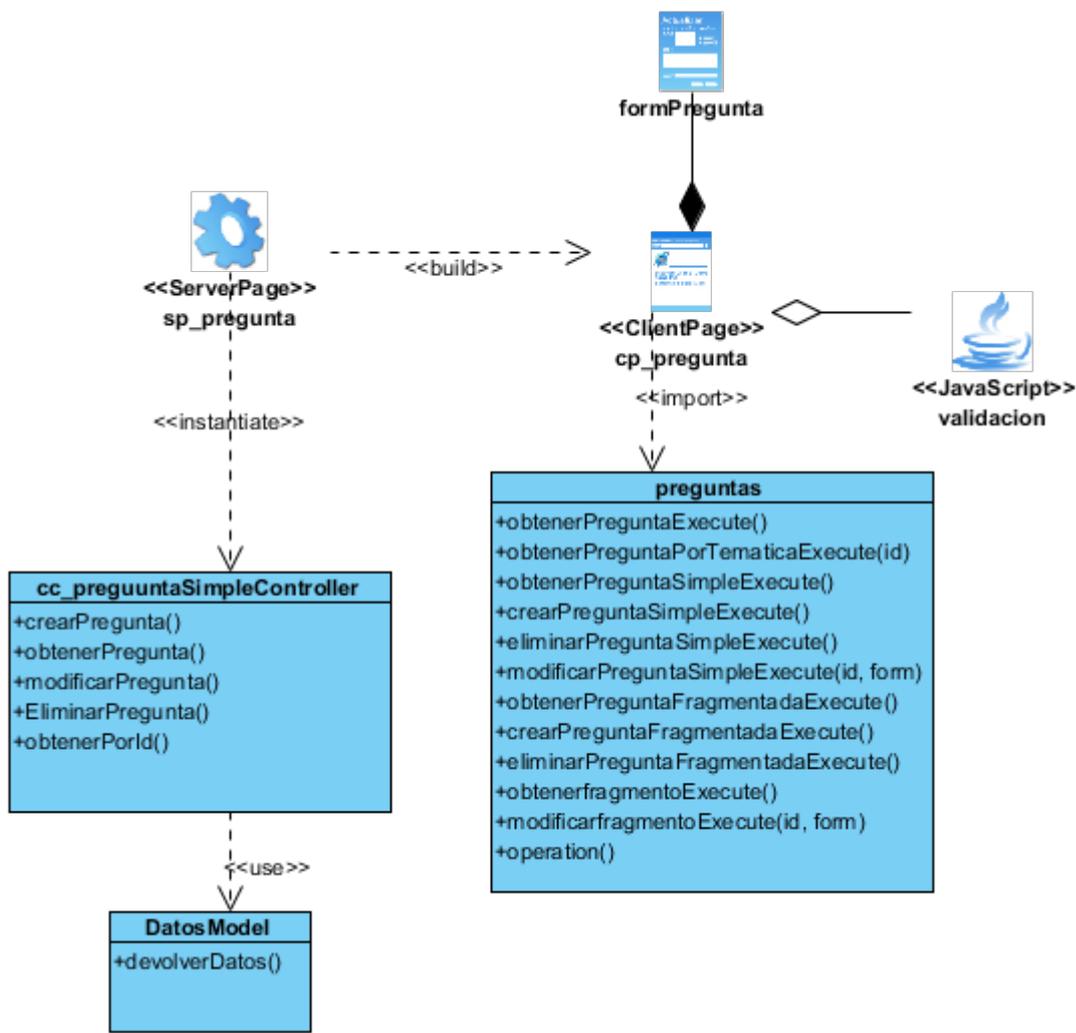


Figura 2.7-1 Diagrama de clases del diseño con estereotipos web para adicionar preguntas.

Descripción del diagrama de clases del diseño para adicionar pregunta

El diagrama está compuesto por una página cliente llamada pregunta que contiene un formulario que permite luego de insertar los datos de la pregunta (temática, pregunta) y validarlos con el uso de una clase llamada pregunta con extensión js, que se encarga de realizar las peticiones al controlador **preguntaSimpleController** ubicada en el servidor y este a su vez posibilita la gestión y manejo del modelo que es el encargado de devolver los datos que se encuentran almacenados en la base de datos.

2.8 Diagrama de secuencia

Un diagrama de secuencia es un tipo de diagrama de interacción porque describe cómo y en qué orden un grupo de objetos funcionan en conjunto. Tanto los desarrolladores de software como los profesionales de negocios usan estos diagramas para comprender los requisitos de un sistema nuevo y documentar un proceso existente. A los diagramas de secuencia en ocasiones se los conoce como diagramas de eventos o escenarios de eventos (Santos-Romero, 2020).

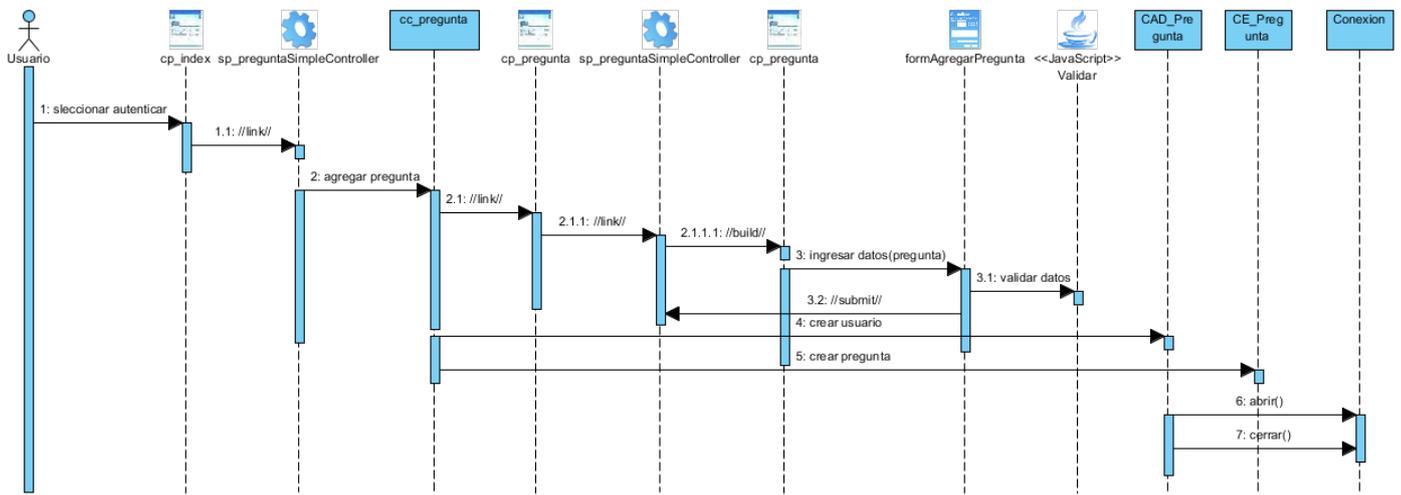


Figura 2.8-1 Diagrama de secuencia para adicionar preguntas.

Descripción del diagrama de secuencia para adicionar pregunta

El proceso inicia cuando un usuario con rol de cliente, investigador, administrador, o super usuario desea adicionar una pregunta a la base de datos. Este debe autenticarse antes de registrar la nueva pregunta, luego la clase controladora de preguntas enviara a la vista las preguntas existentes en el sistema y las posibles temáticas con las que se puede adicionar una pregunta, construyéndose así la pagina cliente pregunta, donde el usuario ingresara los datos de la nueva pregunta que desea adicionar y estos serán validados por una clase pregunta con extensión js. Luego de enviar los datos del controlador de la pagina cliente al controlador este establecerá conexión con la calase entidad pregunta que abrirá la conexión con la base de datos e insertara la nueva pregunta en ella para luego cerrar la conexión con la base de datos y así quedar guardada la nueva pregunta en el sistema.

2.9 Patrones de diseño

Un patrón de diseño provee un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular. Tienden a ser independientes de los lenguajes y paradigmas de programación y su aplicación no afecta necesariamente al sistema completo, pero si a un subsistema o parte del mismo (Ramírez, 2020). A continuación, se muestran los patrones de diseño empleados en la implementación de la propuesta de solución:

Patrones generales de software para asignar responsabilidades (GRASP)

Experto: Se encarga de asignar la responsabilidad al experto en la información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Brinda la posibilidad de conservar el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide, lo que provee un bajo nivel de acoplamiento. Promueve clases sencillas y cohesivas que son más fáciles de mantener y comprender (Pressma, 2010). Solo la clase cuestionarios es encargada de gestionar los cuestionarios por lo cual en ella se evidencia este patrón.

Creador: Consiste en asignar a un Objeto la responsabilidad de crear otro Objeto. Un objeto es responsable de crear una nueva instancia de alguna clase si: agrega o contiene objetos de ella, registra las instancias de

sus objetos o tiene los datos de inicialización que serán enviados a ella cuando el objeto sea creado (Pressma, 2010). La utilización de este patrón se evidencia desde la clase preguntas.

```
import { api } from "src/boot/axios";
...
export async function crearPreguntaSimpleExecute(form) {
  await api.post('/preguntaSimple', form)
  let resp = await obtenerPreguntaSimpleExecute()
  return resp
}
```

Figura 2.9-1 Patrón creador en la clase preguntas

Controlador: El patrón ofrece una guía para tomar decisiones sobre los eventos de entrada, asignando la responsabilidad del manejo de mensajes de los eventos del sistema a una clase controladora, ya que los elementos de interfaz y sus controladores de eventos, no deben ser responsables de controlar los eventos del sistema (Pressma, 2010). La utilización de este patrón se evidencia en la utilización de la clase controladora **preguntaSimpleController**.

Bajo acoplamiento: El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases. Este patrón da soporte a una mínima dependencia y a un aumento de la reutilización; una clase con bajo acoplamiento no depende de "muchas otras" clases para realizar sus tareas, permitiendo que se pueda reutilizar con mayor facilidad y flexibilidad (Pressma, 2010). Un ejemplo es la librería docx.ts la cual solo se relaciona con la clase controladora **documentoController**, evitando así que una modificación en alguna de ellas conlleve a una gran repercusión en las restantes.

Alta cohesión: asignar una responsabilidad de modo que la unión se mantenga a gran escala. Asignar a las clases responsabilidades que trabajen sobre una misma área de aplicación y que no tengan mucha complejidad. Mejoran la claridad y facilidad con que se entiende el diseño. Este patrón arquitectónico permite agrupar de forma clara las vistas, la clase controladora y las clases modelos, estas últimas donde se implementa la lógica de negocio. Un ejemplo es la librería axios.ts que ayuda a no sobrecargar innecesariamente al código de la clase controladora correspondiente.

Patrones Gang of Four (GOF)

Los patrones GOF (*GandofFour*, Banda de los Cuatro) son patrones de diseño que definen una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular (Ramírez, 2020). Estos patrones se dividen en tres categorías: los creacionales, los estructurales y los de comportamiento, a continuación, se muestran los que se implementaron en el sistema:

Patrones de creación

- *Singleton*: garantiza que una clase solo tenga una instancia y proporcionar un punto de acceso global a ella.

Patrones de estructura

- *Fachada*: brinda una interfaz que abstrae completamente al usuario de la complejidad de los procesos.

2.10 Diagrama de caso de uso

Este documento modela el comportamiento del sistema, mostrando un conjunto de casos de uso, actores y sus relaciones. Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso (Santos-Romero, 2020).

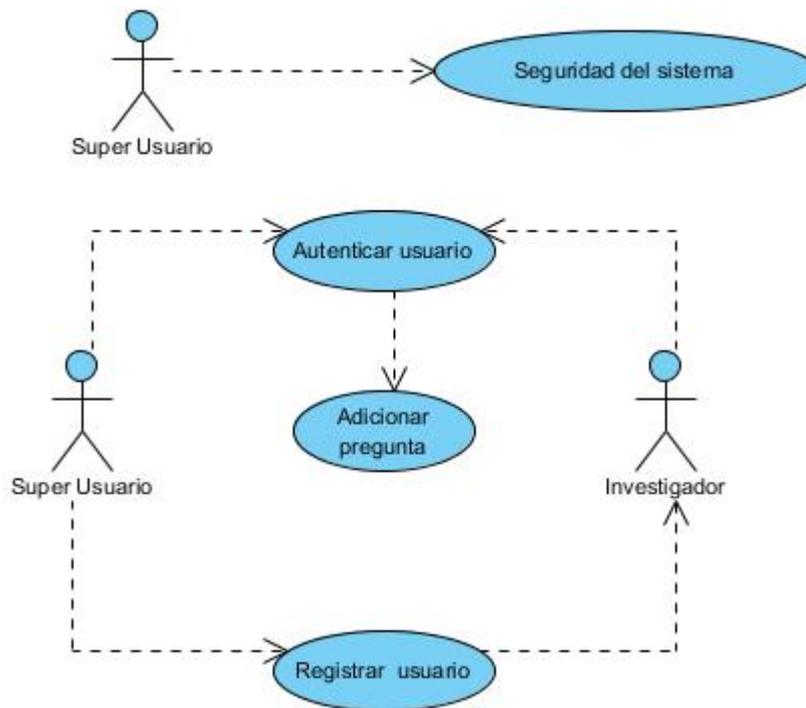


Figura 2.10-1 Diagrama de caso de uso adicionar pregunta

Descripción del diagrama de caso de uso adicionar pregunta

Descripción textual	Permite adicionar preguntas con su clasificación por temática a la base de datos.
Actores	Súper usuario e investigador.
Precondición	Se ha autenticado correctamente el usuario.
Flujo de eventos	
Flujo básico "Adicionar Pregunta"	
	<ol style="list-style-type: none"> 1. El investigador selecciona la pestaña preguntas. 2. El Sistema muestra una ventana para ingresar los datos de la pregunta de elección simple y una tabla con las preguntas existentes en la base de datos. Además, el sistema muestra las opciones de modificar pregunta de elección simple, adicionar pregunta de elección múltiple, modificar pregunta de selección múltiple, adicionar opción de respuesta, modificar opción de respuesta. 3. El investigador llena los campos de la pregunta y su temática. 4. El Sistema verifica si los campos no están vacíos y envía la pregunta al servidor.

Pos-condiciones

1. El **Sistema** muestra el mensaje “Se copiaron en la base de datos la pregunta”.

Flujos alternos

Flujo alternativo “Adicionar Pregunta”

1. El **Sistema** muestra el mensaje “La pregunta no fue guardada correctamente en la base de datos”.

2.11 Diagrama de componentes

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y permite visualizar las dependencias que existen entre estos. En estos se ilustran las piezas del software, controladores embebidos, etc. que conformarán un sistema (Pressma, 2010).

A continuación, se muestran los diagramas de componentes para la solución propuesta del sistema para la elaboración de cuestionarios en investigaciones sociales.

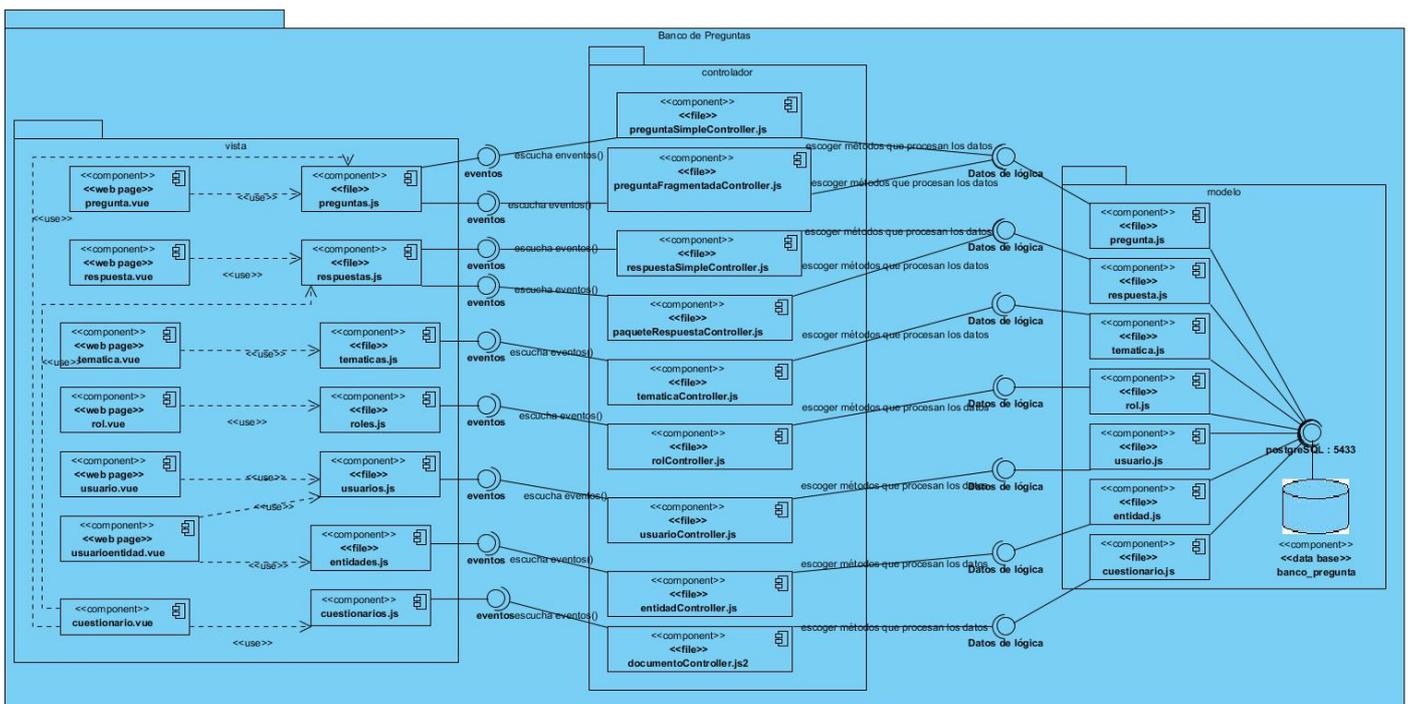


Figura 2.11-1 Diagrama de diseño de componentes interno

Descripción del diagrama de componentes del sistema

Se listan todos los componentes definidos en el sistema para la elaboración de cuestionarios en investigaciones sociales y de cada uno de ellos se especificará el nombre del componente, una descripción basada en el objetivo y principales características del componente, los servicios que provee ese componente con sus detalles.

2.12 Modelo de datos

Un modelo de datos es un lenguaje orientado a describir una Base de Datos. Típicamente un modelo de

datos permite describir las estructuras de datos que representan a los elementos del dominio, el tipo de dato de los atributos y la forma en que se relacionan (Pressma, 2010).

A continuación, se muestra el modelo de datos de la solución:

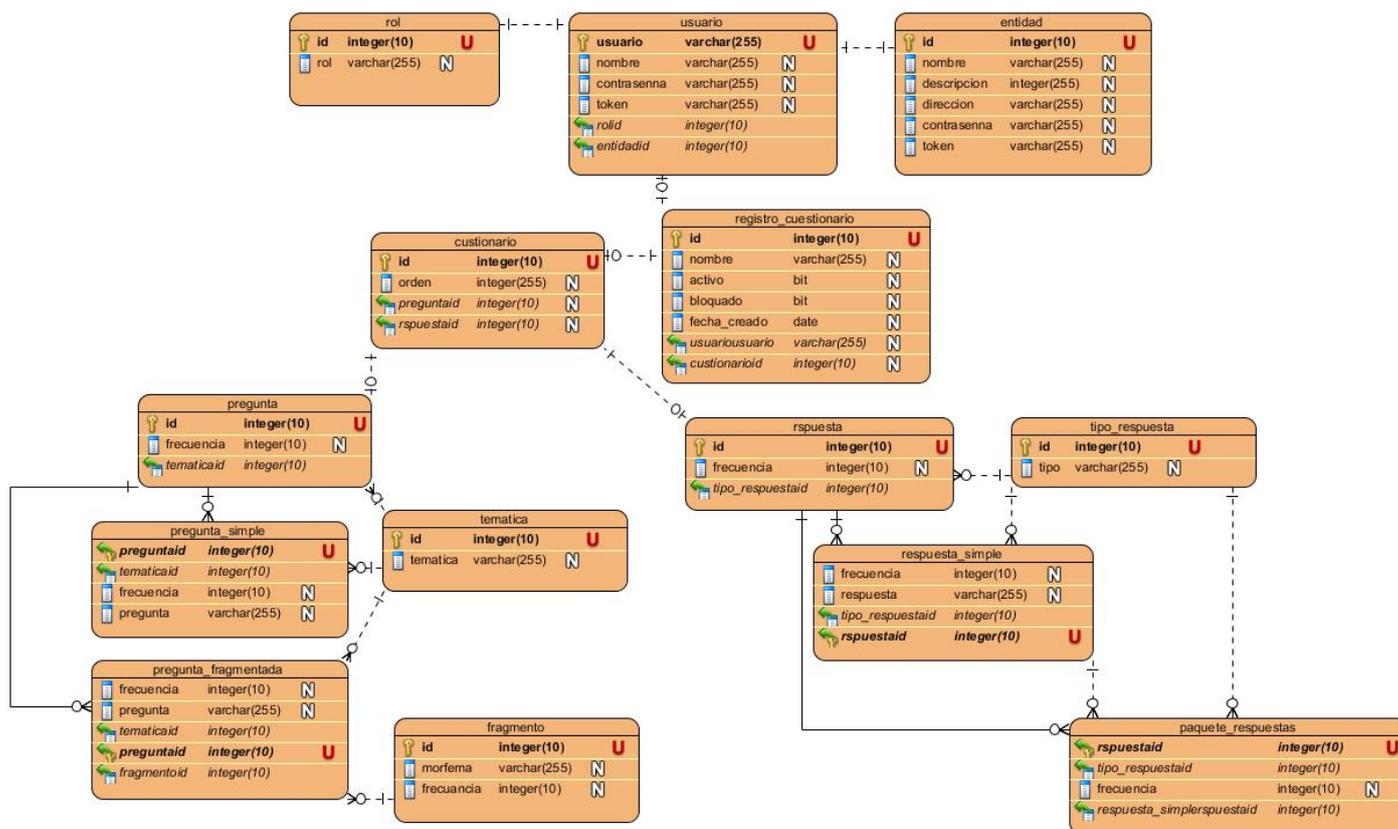


Figura 2.12-1 Modelo de datos del sistema

Descripción del modelo de datos del sistema

El modelo de datos de la solución propuesta está compuesto por dos esquemas, el del negocio que presenta 14 tablas y un esquema cuestionarios, que se encarga de conservar los cuestionarios que se van creando con las preguntas y respuestas organizadas con su formato para cada ítem. En el modelo se consideran pregunta, pregunta_sipmle, pregunta_fragmentada, respuesta, respuesta_simple y paquete_respuesta como tablas críticas del sistema atendiendo a que la mismas, son las que más crecen y a su vez en las que más acciones de UPDATE, INSERT, DELETE y SELECT se realizan.

2.13 Conclusiones Parciales

La obtención de los elementos y artefactos de diseño y arquitectura, permitió proyectar una visión más acabada de cuáles son las prestaciones con las que se debe contar, cómo debe funcionar el componente y su integración con los restantes del sistema, las clases que lo componen y los métodos y propiedades que deben de implementarse:

1. Los requisitos funcionales y no funcionales identificados permitieron definir las funciones que deben realizarse y las restricciones a cumplir por el sistema a implementar.
2. La utilización del patrón arquitectónico modelo-vista-controlador (MVC) permitió estructurar los componentes del módulo, de esta manera, se obtuvo una clara separación entre interfaz, lógica de negocio y datos, lo cual sirvió de soporte a la reutilización de los componentes reduciendo el costo

y tiempo de desarrollo.

3. Los diagramas de clases del diseño permitieron establecer la base estructural de la implementación de la propuesta de solución.
4. El modelo de datos permite describir la estructura de la base de datos con el fin de garantizar una correcta gestión de los mismos.

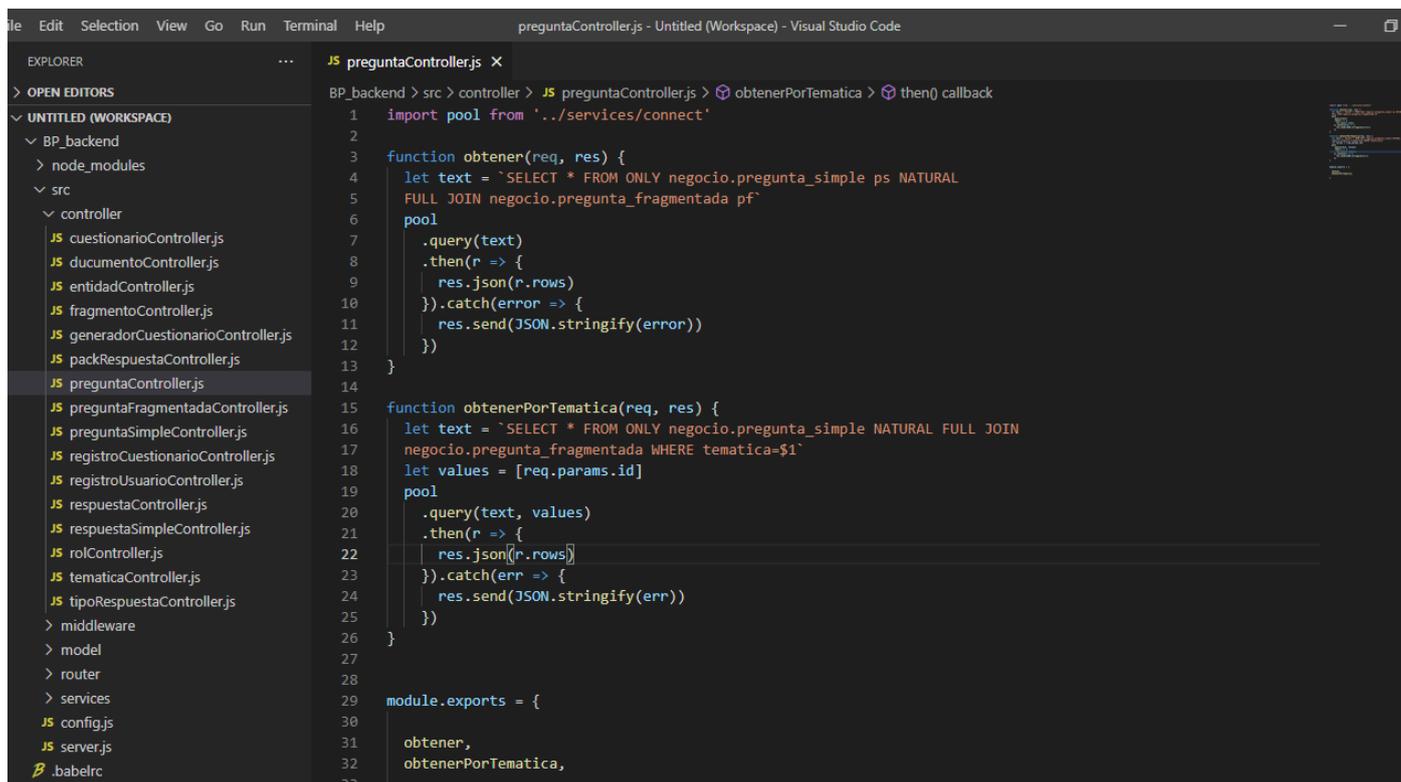
Capítulo III: Implementación y prueba del Desarrollo del sistema para la elaboración de cuestionarios en investigaciones sociales.

3.1 Introducción

En el presente capítulo se abordan los aspectos fundamentales de las fases de Construcción y Transición, dentro de la cual se destaca el proceso de implementación y prueba de software. La implementación del componente de análisis comprende el desarrollo e integración de artefactos que deberán ser sometidos a diferentes pruebas de calidad de software para garantizar que estos funcionen correctamente.

3.2 Implementación

En este proceso se procede a la codificación de cada módulo, con el objetivo principal desarrollar la arquitectura y el sistema como un todo, planificar qué subsistemas deben ser implementados y en qué orden deben ser integrados, notificar si se encuentran errores de diseño e integrar el sistema siguiendo el plan (Fernández y Cadelli, 2014). A continuación, la figura 3.2-1 muestra una parte de la implementación del sistema para la elaboración de cuestionarios en investigaciones sociales:



```
file Edit Selection View Go Run Terminal Help      preguntaController.js - Untitled (Workspace) - Visual Studio Code

EXPLORER
... JS preguntaController.js X
> OPEN EDITORS
UNTITLED (WORKSPACE)
  BP_backend
  node_modules
  src
  controller
  JS cuestionarioController.js
  JS documentoController.js
  JS entidadController.js
  JS fragmentoController.js
  JS generadorCuestionarioController.js
  JS packRespuestaController.js
  JS preguntaController.js
  JS preguntaFragmentadaController.js
  JS preguntaSimpleController.js
  JS registroCuestionarioController.js
  JS registroUsuarioController.js
  JS respuestaController.js
  JS respuestaSimpleController.js
  JS rolController.js
  JS tematicaController.js
  JS tipoRespuestaController.js
  middleware
  model
  router
  services
  JS config.js
  JS server.js
  .babelrc

BP_backend > src > controller > JS preguntaController.js > obtenerPorTematica > then() callback
1  import pool from '../services/connect'
2
3  function obtener(req, res) {
4    let text = `SELECT * FROM ONLY negocio.pregunta_simple ps NATURAL
5    FULL JOIN negocio.pregunta_fragmentada pf`
6    pool
7      .query(text)
8      .then(r => {
9        res.json(r.rows)
10     })
11     .catch(error => {
12       res.send(JSON.stringify(error))
13     })
14   }
15
16   function obtenerPorTematica(req, res) {
17     let text = `SELECT * FROM ONLY negocio.pregunta_simple NATURAL FULL JOIN
18     negocio.pregunta_fragmentada WHERE tematica=$1`
19     let values = [req.params.id]
20     pool
21       .query(text, values)
22       .then(r => {
23         res.json(r.rows)
24       })
25       .catch(err => {
26         res.send(JSON.stringify(err))
27       })
28   }
29
30   module.exports = {
31     obtener,
32     obtenerPorTematica,
33   }
```

Figura 3.2-1 Ejemplo de implementación del sistema

3.3 Diagrama de despliegue

Un diagrama de despliegue muestra las configuraciones de los nodos de procesamientos en tiempo de ejecución, los componentes que existen en ellos, cómo y dónde se desplegará el sistema. Estos nodos forman la topología de hardware sobre el que se ejecuta el sistema. Este diagrama se preocupa principalmente de la entrega, distribución e instalación de los fragmentos que conforman el sistema físico («[PDF] Diagrama de despliegue - Free Download PDF» [27 de octubre del 2021]).

En la figura 3.3-1 que se muestra a continuación, se observan los componentes de hardware y sus relaciones para realizar el despliegue del sistema. Este modelo de despliegue está compuesto por tres nodos que representan al menos una computadora del usuario, esta se conecta a un servidor de aplicaciones (mediante el protocolo de hipertexto HTTP o HTTPS) y éste establece una conexión con el servidor de base de datos (mediante TCP/IP):

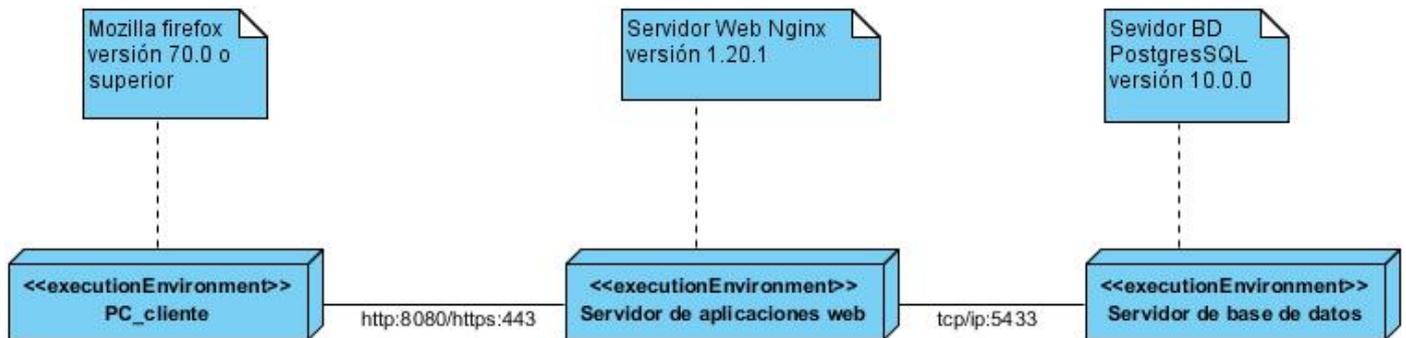


Figura 3.3-1 Diagrama de despliegue BP

3.4 Estándares de codificación

Los estándares de codificación son establecidos por el equipo de desarrollo, abarcan todos los aspectos relacionados con la generación de código en un proyecto para propiciar que todos los desarrolladores lo puedan entender en el menor tiempo y que el código en consecuencia sea mantenerle («[PDF] Carmen. Estándares de codificación. Manuel Arias Calleja - *Free Download PDF*» [27 de octubre de 2021]).

En la figura 3.4-1 se muestran algunos ejemplos de estándares de codificación utilizados en la implementación de la solución como son:

- Formato de los archivos .js,.tsó .vue.
- Convenciones de nombre.
- Evitar líneas de más de 100 caracteres.
- El tamaño de los métodos no exceda las 20 líneas de código, de ser así se dividirá en dos métodos.
- Estilo de código.

```

function obtenerPreguntaSimplePorId(req, res) {

  let text = "SELECT * FROM negocio.pregunta_simple WHERE id=$1"
  let values = [req.body.id]
  pool
    .query(text, values)
    .then(r => {
      res.json(r.rows)
    }).catch(err => {
      res.send(JSON.stringify(err))
    })
}
}

```

Figura 3.4-1 Ejemplo de uso de estándares de codificación.

3.4.1 Convenciones de codificación

Las convenciones de codificación son un grupo de normas para un lenguaje de programación determinado, estas recomiendan estilos de programación, métodos para mantener el aspecto del código fuente y buenas prácticas. Las convenciones incluyen las declaraciones de espacios en blanco, las llaves de apertura y cerrado, la organización de archivos y los comentarios («Convención de nombres: desde el CamelCase hasta el kebab-case | by Adrián Alonso | Medium» [27 de octubre de 2021]).

Para la implementación de la propuesta de solución se utilizaron los estándares de codificación utilizados por el equipo de desarrollo del CESPO, con el propósito de estandarizar las nomenclaturas en el desarrollo del sistema y mantener las buenas prácticas. A continuación, se muestran algunas de las convenciones empleadas, como se muestra en las figuras 3.4-2 y 3.4-3.

- Emplear el signo de // para comentarios simples.
- **CamelCase:** Se utilizó para escribir los nombres de las clases, siguiendo una estructura de frases compuestas donde la primera letra de cada palabra se escribirá en mayúscula, excepto la primera palabra y todas las palabras estarán juntas.
- Las variables deben comenzar con una letra minúscula y las variables globales con mayúscula.
- Se deben evitar nombres de una sola letra. Debemos usar nombres descriptivos que ayuden a entender el código. Nunca debemos usar nombres que se puedan confundir con palabras reservadas del lenguaje como break, const, else, new, entre otras.

```

function obtenerPorTematica(req, res) {
  // la variable text contiene la consulta a la base de datos para acceder a las preguntas
  // con el id recibido por parámetro
  let text = `SELECT * FROM ONLY negocio.pregunta_simple NATURAL FULL JOIN
  | negocio.pregunta_fragmentada WHERE tematica=$1`
  let values = [req.params.id]
  pool
  .query(text, values)
  .then(r => {
    console.log(r.rows[0])
    res.json(r.rows)
  }).catch(err => {
    console.log(err)
  })
}

```

Figura 3.4-2 Ejemplo de uso donde se emplean el signo de // para comentarios y donde se emplea CamelCase.

```

async crearCuestionario() {
  this.form.usuario = this.userActual.usuario;
  let tabla = await crearCuestionarioExecute(this.form);
  this.$router.push({
    name: "cuestionario",
    params: { cuestionario_name: tabla.data.msg, nombre: this.form.nombre },
  });
}

```

Figura 3.4-3 Ejemplo de uso de minúscula para las variables locales y no uso de variables de una sola letra.

3.5 Pruebas al sistema para la elaboración de cuestionarios en investigaciones sociales

Las pruebas al software son un elemento crítico en la garantía de calidad del software y representa la revisión final de las especificaciones del diseño y de la codificación que se realizaron en etapas anteriores (Pressman, 2013), estas permiten validar (proceso que determina si el software cumple los requisitos) y verificar el software (proceso que determina si los productos de una fase satisfacen las condiciones de dicha fase).

Descripción de las pruebas utilizadas

Las pruebas aplicadas al sistema son: **pruebas unitarias, pruebas de integración y las pruebas de aceptación del cliente.**

Previamente diseñadas para el software (Díaz Asencio, 2021).

3.5.1 Pruebas Unitarias

Las pruebas de unidad o pruebas unitarias son procesos en los que se prueban unidades del programa o clases de objetos individuales. Las pruebas unitarias deben enfocarse en comprobar la funcionalidad de objetos o métodos (Sommerville, 2011). Estas pruebas pueden ser automatizadas, lo cual facilita que el programador identifique funciones que no ofrezcan una salida acorde con la lógica implementada. Al sistema le fueron aplicados los métodos de prueba de caja blanca y caja negra.

Pruebas de caja blanca

Las pruebas de caja blanca se basan en la elaboración de casos de prueba, que usan las estructuras de control para el diseño procedimental del software, logrando como resultado que disminuya en un gran porcentaje el número de errores existentes en los sistemas, una mayor calidad y confiabilidad, por lo que su estructura está fuertemente ligada al código fuente del sistema (EcuRed, 2021).

En el caso de la técnica de las pruebas de caja blanca que se realizaron al sistema para la elaboración de cuestionarios, estas pruebas están dirigidas a las funcionalidades internas del sistema. Se ejecutan probando caminos lógicos por los cuales deberá transitar el sistema una vez puesto en funcionamiento y se observan los resultados arrojados en diferentes puntos; lo que permitió constatar que los resultados son los esperados. Estas pruebas se desarrollan de forma que se asegure que la operación interna se ajusta a las especificaciones, y que todos los componentes internos se han probado de forma adecuada. Dentro de las pruebas de caja blanca se encuentra la prueba del camino básico. Este método permite que el diseñador de casos de pruebas obtenga una medida de complejidad de un diseño procedimental y que use esta medida como guía para definir un conjunto básico de rutas de ejecución. Los casos de pruebas derivados para ejercitar el conjunto básico deben garantizar que se ejecuta cada instrucción del programa por lo menos durante la prueba (Pressma, 2010).

Esta prueba se enmarca en la obtención de la complejidad lógica de un diseño. El método aplicado consiste en la representación mediante un grafo de flujo asociado de todos los posibles caminos que se puedan iniciar en una determinada porción de código y aplicarle a cada uno de estos caminos un caso de prueba (Pressma, 2010). A la representación realizada por el grafo se le calculará la complejidad ciclométrica. Los pasos a seguir para la aplicación del método son los siguientes:

- Dada una selección del código fuente se representa un grafo asociado.
- Se calcula la complejidad ciclométrica del grafo.
- Se determina un conjunto básico de caminos independientes.
- Se elaboran los casos de pruebas que obligan a la ejecución de cada camino del conjunto básico.

Tabla 3.5-1 Diseño del caso de prueba (Técnica del Camino Básico).

Prueba de caja blanca	
Código al que se aplica: functioncrearPreguntaSimple(req, res) { lettematica = req.body.tematica//1 letpregunta = req.body.ptregunta//1 if (tematica == null pregunta == null){//2 res.send("Faltan datos");//9 } else {//3	Representación en grafo de flujo:

<pre> let text = 'INSERT INTO negocio.pregunta_simple(tematica,pregunta) VALUES(\$1,\$2) RETURNING *'//4 letvalues = [tematica, pregunta]//5 pool .query(text, values)//6 .then(r => {//7 res.send({ cantAñadida: r.rowCount })//9 }).catch(error => {//8 res.send(JSON.stringify(error))//9 }) } } </pre>	<pre> graph TD 1 --> 2 1 --> 3 2 --> 9 3 --> 4 3 --> 8 4 --> 5 4 --> 8 5 --> 6 6 --> 8 6 --> 7 7 --> 9 8 --> 9 </pre>
<p>Complejidad ciclomática:</p> <p>$V(G) = \text{Cantidad de Aristas } [A] - \text{Cantidad de nodos } [N] + 2.$</p> <ul style="list-style-type: none"> $V(G) = 10 - 9 + 2$ $V(G) = 3$ 	<p>Caminos independientes:</p> <ol style="list-style-type: none"> 1-2-9 1-3-4-5-6-7-9 1-3-4-5-6-8-9

Dando continuidad al método y contando con el grafo y los caminos identificados previamente en la prueba de caja blanca, se pasa a la creación de casos de pruebas para cada uno de estos caminos. La tabla muestra el caso de prueba para el camino 1-3-4-5-6-7-9, los restantes casos de pruebas se encuentran en los anexos.

Tabla 3.5-2 Caso de prueba para el camino No 2.

Caso de prueba para el camino No 2	
Camino	1-3-4-5-6-7-9
Descripción.	<ul style="list-style-type: none"> • Captura los datos de la temática y pregunta. • Verifica que la temática y la pregunta no estén vacío. • Crea la sentencia para consultar la base de datos. • Apunta a los valores que van a ser insertados en la base de datos. • Ejecuta la consulta. • Se guarda el resultado de la consulta.

	<ul style="list-style-type: none"> • Se envían los datos correctamente en la base de datos.
Entrada.	Chequea la variable recogida por parámetros.
Resultado esperado.	Retorna los datos asociados a la pregunta guardada.

Pruebas de caja negra

Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas no son importantes las estructuras de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos (EcuRed, 2021).

Las pruebas de caja negra se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada para estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro. Las pruebas de caja negra están especialmente indicadas en aquellos módulos que van a ser interfaz con el usuario (en sentido general: teclado, pantalla, ficheros, canales de comunicaciones, etc.) (López, 2005).

Descripción del caso de prueba adicionar Pregunta

Se realizó un caso de prueba para cada requisito implementado. A continuación, se muestra el caso de prueba realizado al requisito adicionar pregunta que describe cada uno de los escenarios que pueden existir ante las posibles acciones realizadas por el usuario. De manera similar se realizaron los casos de prueba a los requisitos restantes. Para ver los requisitos restantes dirigirse a los anexos.

Condiciones de ejecución:

- El usuario debe estar autenticado.
- Debe ser un usuario con roles de investigador, administrador, súper usuario o una entidad.
- Se debe seleccionar la opción **Inicio/ Preguntas**.

Tabla 3.5-3 Diseño de caso de prueba adicionar pregunta.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
Adicionar pregunta.	Con la realización de este requisito se pretende adicionar preguntas a la base de datos para poder ser	EP 1.1: Adicionar pregunta correctamente.	1. Se muestra una interfaz con una lista de las preguntas existes en la base de datos.

			6. El sistema muestra el mensaje de información: " <i>La pregunta ha sido adicionada satisfactoriamente</i> " y se limpia el formulario de pregunta simple.
		EP 1.4: Cancelar operación.	<p>1. Se muestra una interfaz con una lista de las preguntas existentes en la base de datos.</p> <p>Se selecciona el botón adicionar pregunta simple.</p> <p>3. Se muestra la interfaz adicionar pregunta simple.</p> <p>4. Se introducen todos los datos.</p> <p>5. Se presiona el botón Cancelar.</p> <p>6. Se cancela la acción, se cierra la ventana y se limpia el formulario de pregunta simple.</p>

Descripción de variables

Tabla 3.5-4 Descripción de variables del RF Adicionar pregunta.

No	Nombre del campo	Clasificación	Puede ser nulo	Descripción
1	Temática	Select	No	Permite seleccionar una temática para asociar una pregunta a ella.
2	Pregunta	Input	No	Permite introducir una pregunta para ser adicionada a la base de datos.

Juego de datos a probar

Tabla 3.5-5 Juego de datos a probar del RF Adicionar pregunta variable 1

+	Escenario	Variable 1	Respuesta del sistema
		actividad	

EP 1.1	Adicionar pregunta correctamente.	V (seleccionada)	El sistema muestra el mensaje de información: " <i>La pregunta ha sido adicionada satisfactoriamente</i> " y se limpia el formulario de pregunta simple.
EP 1.2	Adicionar pregunta incorrectamente.	V (no seleccionada)	Al presionar Añadir no permite realizar ninguna operación.
EP 1.3	Adicionar pregunta correctamente presionando el botón Añadir .	V(seleccionada)	El sistema muestra el mensaje de información: " <i>La pregunta ha sido adicionada satisfactoriamente</i> " y se limpia el formulario de pregunta simple.
EP 1.4	Cancelar operación.	NA	El sistema cancela las acciones y cierra la ventana Adicionar pregunta.

Tabla 3.5-6 Juego de datos a probar del RF Adicionar pregunta variable 2

Id del escenario	Escenario	Variable 2	Respuesta del sistema
		actividad	
EP 1.1	Adicionar pregunta correctamente.	V (llenado)	El sistema muestra el mensaje de información: " <i>La pregunta ha sido adicionada satisfactoriamente</i> " y se limpia el formulario de pregunta simple.
EP 1.2	Adicionar pregunta incorrectamente.	V (no llenado)	Al presionar Añadir no permite realizar ninguna operación.
EP 1.3	Adicionar pregunta correctamente presionando el botón Añadir .	V(llenado)	El sistema muestra el mensaje de información: " <i>La pregunta ha sido adicionada satisfactoriamente</i> " y se limpia el formulario de pregunta simple.
EP 1.4	Cancelar operación.	NA	El sistema cancela las acciones y cierra la ventana Adicionar pregunta.

En el primer sprint de las pruebas de software de caja negra realizadas se detectaron un total de 12 no conformidades en el sistema para la elaboración de cuestionarios, de ellas 5 de funcionalidad y 7 de ortografía, las cuales fueron resueltas en su totalidad. En una segunda iteración de pruebas se encontraron 5 no conformidades que no se habían identificado en el sprint anterior, las mismas de concordancia y fueron resueltas; ya para el tercer sprint de pruebas al sistema, no se detectaron no conformidades. Finalmente se comprobó que la aplicación cumple con los requisitos planteados al inicio del desarrollo y que responde exitosamente las peticiones de los usuarios, teniendo como resultado una versión estable del sistema.

3.5.2 Pruebas de integración

Las pruebas de integración constituyen el mecanismo para comprobar el correcto ensamblaje del sistema completo. Estas pruebas se realizan al terminar las pruebas unitarias con el objetivo de verificar que los módulos o componentes que conforman un sistema funcionan correctamente una vez que han sido integrados. Para la realización de las mismas debe tenerse en consideración un conjunto de aspectos tales como, el grado de complejidad de los módulos o componentes, su importancia funcional con respecto a las especificaciones del sistema, la interrelación entre los mismos y las estadísticas de error en las pruebas unitarias de forma tal que se prueben primeramente los más críticos. Para ejecutar las mismas se seleccionan los casos de pruebas que serán diseñados teniendo en cuenta el grado de dependencia entre los componentes para realizar una funcionalidad específica (Pressman, 2010).

3.5.3 Prueba de aceptación

El objetivo de las pruebas de aceptación es validar que el sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento. Las pruebas de aceptación son definidas por el usuario del sistema y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final corresponden al usuario (Pressman, 2010).

Las pruebas de aceptación representan aquella fase del ciclo de vida de desarrollo de software en el que el equipo de desarrollo y el área usuaria de un sistema de información tienen que garantizar que el sistema desarrollado se corresponde con los requerimientos definidos (González y Mayo, 2014).

Los principales investigadores trabajaron con el software y validaron la aceptación del mismo, ya que cumple con la elaboración de los cuestionarios llevados a cabo en las diferentes investigaciones sociales que se aplican a la población del país, cumpliéndose así con los requisitos capturados en la primera etapa.

3.6 Conclusiones Parciales

- Con la realización de este capítulo, se le dio cumplimiento a la fase de Construcción y Transición del sistema propuesto. Se abordó a cerca de los diferentes aspectos con alto significado para las pruebas de software, acentuando el significado de las mismas en la realización de un proyecto.
- Luego de llevar a cabo las pruebas unitarias, integración y de aceptación, se consiguió una versión estable del producto, obteniendo como resultado que el sistema para la elaboración de cuestionarios en investigaciones sociales desarrollado, satisficel funcionamiento esperado y cumple cada uno de los requisitos funcionales.

Conclusiones generales

- El estudio de los diferentes sistemas para la elaboración de cuestionarios determinó que ninguno de estos cumple con los requisitos necesarios planteados por el usuario, pues no conservan diseños de cuestionarios, ni preguntas a fines de investigaciones sociales. Se hizo necesario el desarrollo de un sistema para la elaboración de cuestionarios en investigaciones sociales que cumpla con estas funcionalidades.
- Para el desarrollo del sistema se identificaron y aprobaron un grupo de requisitos funcionales y no funcionales con los que debe cumplir el producto final, los mismos fueron determinados en conjunto con los especialistas e investigadores del CESPO lo que permitió definir las características del producto final.
- Para la realización del diseño se definieron un conjunto artefactos especificados el proceso de desarrollo híbrido SCRUM-RUP. Estos permitieron reducir los esfuerzos en el desarrollo, eficiencia y consistencia de los diseños del producto final.
- Para elevar la calidad del producto se efectuó un conjunto de pruebas; estas se desarrollaron en función de los requisitos, como se define en la propuesta por la metodología de desarrollo. Las pruebas se aplicaron en varios sprint hasta tener un producto altamente funcional.
- En sentido general se puede afirmar que se cumplieron los objetivos trazados para el desarrollo del trabajo. Se logró implementar un Sistema para la elaboración de cuestionarios en investigaciones sociales que permite crear los cuestionarios utilizados en las investigaciones del CESPO para contribuir a mejorar el proceso de elaboración de las técnicas de captura de información que se aplican a la población.

Recomendaciones:

Al culminar la investigación y el desarrollo de la propuesta de solución, el autor recomienda:

- Continuar el desarrollo del diseño de los ítems.
- Desarrollar el manual de usuario para publicarlo en la pestaña ayuda.
- Publicar los cuestionarios de investigaciones anteriores.