UNIVERSIDAD NACIONAL DEL ALTIPLANO

ESCUELA DE POSGRADO

MAESTRÍA EN INFORMÁTICA

TESIS

AUTOMATIZACIÓN DE DESARROLLO DE APLICACIONES WEB, BASADO EN TECNOLOGÍA DE FRAMEWORKS Y MACROS

PRESENTADA POR:

AMILKAR SUCASACA PACORI

PARA OPTAR EL GRADO ACADÉMICO DE:

MAGISTER SCIENTIAE EN INFORMÁTICA

PUNO, PERÚ

2023

NOMBRE DEL TRABAJO

AUTOMATIZACIÓN DE DESARROLLO DE APLICACIONES WEB, BASADO EN TECNOLOGÍA DE FRAMEWORKS Y MACROS

AUTOR

AMILKAR SUCASACA PACORI

RECUENTO DE PALABRAS

32544 Words

RECUENTO DE CARACTERES

178893 Characters

RECUENTO DE PÁGINAS

156 Pages

TAMAÑO DEL ARCHIVO

3.6MB

FECHA DE ENTREGA

Dec 27, 2023 11:11 AM GMT-5

FECHA DEL INFORME

Dec 27, 2023 11:15 AM GMT-5

6% de similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base

- 6% Base de datos de Internet
- 1% Base de datos de publicaciones
- Base de datos de Crossref
- Base de datos de contenido publicado de Crossr
- 4% Base de datos de trabajos entregados

Excluir del Reporte de Similitud

- Material bibliográfico
- Material citado

José P. Tito Lipa
Ing. Estadístico e Informático D.Sc.
CIP. 159645

Ing. Freddy Heric Villasante Saravia
DR EN ESTADÍSTICA E INFORMÁTICA
CIP 47405

UNIVERSIDAD NACIONAL DEL ALTIPLANO

ESCUELA DE POSGRADO

MAESTRÍA EN INFORMÁTICA

TESIS

AUTOMATIZACIÓN DE DESARROLLO DE APLICACIONES WEB, BASADO EN TECNOLOGÍA DE FRAMEWORKS Y MACROS

PRESENTADA POR:

AMILKAR SUCASACA PACORI

PARA OPTAR EL GRADO ACADÉMICO DE:

MAGISTER SCIENTIAE EN INFORMÁTICA

APROBADO POR EL JURADO SIGUIENTE:

PRESIDENTE

.....
Dr. JUAN CARLOS VARGAS JUAREZ

PRIMER MIEMBRO

.....
M.Sc. LEONID ALEMÁN GONZALES

SEGUNDO MIEMBRO

.....
M.Sc. ELQUI YEYE PARI CONDORI

ASESOR DE TESIS

.....
Dr. FREDY HERIC VILLASANTE SARAVIA

Puno, 18 de mayo de 2023

ÁREA : Informática
TEMA : Automatización de desarrollo de aplicaciones web, basado en tecnología de frameworks y macros
LÍNEA : Análisis y diseño de sistema de información

DEDICATORIA

Dedico a la memoria de mi amada madre Orfelina, quien iluminó mi vida con su amor y su presencia inolvidable. Aunque ya no estés físicamente conmigo, tu espíritu vive en mi corazón y tu influencia perdura en cada fibra de mi ser.

Dedico de gratitud a mi querida hermana Flor Ofelia, quien ha sido mi apoyo incondicional en cada paso de mi vida. Su presencia constante y su aliento han sido la fuerza que me impulsa a enfrentar los desafíos diarios con determinación y valentía. Agradezco su amor incondicional y su apoyo inquebrantable. ¡Eres un tesoro invaluable en mi vida y te estaré eternamente agradecido!

Dedico a mis amados hermanos, Harry y Walter, quienes han sido pilares fundamentales en mi vida. A través de los altibajos y las alegrías compartidas, nuestra conexión fraternal se ha fortalecido y ha dejado una huella imborrable en mi corazón

Dedico esta cariñosa dedicatoria a mi tía Hilda y mis primos Jorge, Yannet, Mijalich, Albert, quienes son parte esencial de mi familia y han dejado una huella indeleble en mi vida. A través de los años, hemos compartido risas, aventuras y momentos inolvidables que han fortalecido nuestros lazos.

A Justo y Maria padres amorosos de Margoth su amor y dedicación han dejado una huella imborrable en su vida. A través de su crianza, le brindaron el amor, los valores y la guía necesaria para convertirse en la persona maravillosa que fue. Les agradezco por haberme acogido en su hogar y haberme permitido ser parte de su vida.

Con todo mi cariño, quiero dedicar estas palabras a Mary, John, Ruben, Noe y Ronald, quienes han sido pilares fundamentales en mi vida. A través de su apoyo incondicional, hemos logrado fortalecer nuestro vínculo a pesar de la distancia.

AGRADECIMIENTOS

Quiero expresar mi más profundo agradecimiento a mi alma mater de enseñanza y aprendizaje. A lo largo de mi trayectoria académica, la Universidad Nacional del Altiplano ha sido mi hogar intelectual, brindándome las herramientas y oportunidades necesarias para crecer y alcanzar mis metas.

Quiero expresar mi más sincero agradecimiento por su invaluable apoyo como mi asesor de proyecto Dr. Fredy Heric Villasante Saravia. Su dedicación, conocimientos y guía han sido fundamentales para el éxito y desarrollo de este proyecto. Desde el principio, usted ha demostrado un compromiso excepcional al brindarme su tiempo y atención. Su experiencia y sabiduría han iluminado mi camino, permitiéndome tomar decisiones informadas y encontrar soluciones efectivas a lo largo del proceso.

Estimados docentes de la Maestría en Informática y de la Facultad de Ingeniería Estadística e Informática, deseo expresar mi más profundo agradecimiento por su dedicación y compromiso como mis docentes. Su labor ha sido fundamental en mi formación académica y profesional, y ha dejado una huella perdurable en mi vida. A lo largo de la Maestría, ustedes han compartido con nosotros su vasto conocimiento y experiencia en el campo de la informática. Han sido guías excepcionales, impartiendo clases magistrales, desafiándonos intelectualmente y fomentando un ambiente de aprendizaje estimulante.

Finalmente, agradezco a mis amigos Alcides, Wilbert Quintanilla, Wilbert Jallo, Joel, Betsy, Jenrry y Oswaldo, por brindarme su apoyo y su tiempo para la culminación de mi investigación.

ÍNDICE GENERAL

CAPÍTULO II

PLANTEAMIENTO DEL PROBLEMA

CAPÍTULO III

MATERIALES Y MÉTODOS

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

RESUMEN

La presente investigación frente a la creciente demanda de desarrollo de aplicaciones web ha impulsado la necesidad de crear herramientas que aceleren el desarrollo y mejoren la eficiencia en su creación. El proyecto se centró en desarrollar una macro en Excel que agiliza la creación de aplicaciones web utilizando el Framework Codeigniter y una base de datos MySQL. Se estructuró el proceso de desarrollo de aplicaciones basado en proyectos, módulos y formularios, que se diseñan en el entorno de Microsoft Excel. Cada formulario se asigna a una "transacción" y, al completarse, se genera el código fuente utilizando Codeigniter permitiendo la aceleración de la implementación del software. El proyecto fue validado mediante una prueba estadística de t para una muestra ($p<0.01$), comparando el tiempo de desarrollo con y sin la macro y el Framework utilizado, demostrando que la macro acelera significativamente el proceso de desarrollo de aplicaciones web. En resumen, este proyecto destacó la importancia de la implementación de un módulo de diseño en Microsoft Excel, un generador de código fuente que transforma el diseño en una hoja de cálculo y la utilización de librerías para generar informes en formato PDF. Estos componentes son fundamentales para acelerar el diseño y desarrollo de aplicaciones web, lo que resulta en una mayor eficiencia en el proceso de creación de aplicaciones web.

Palabras clave: Control, Generador de código, ribbon, objeto, renderizado, transacción.

ABSTRACT

The present research, in response to the increasing demand for web application development, has driven the necessity to create tools that expedite the development process and enhance efficiency in its creation. The project focused on developing a macro in Excel that streamlines the creation of web applications using the Codeigniter Framework and a MySQL database. The application development process was structured around projects, modules, and forms, which were designed within the Microsoft Excel environment. Each form was assigned to a "transaction," and upon completion, the source code was generated using Codeigniter, allowing for the acceleration of software implementation. The project was validated through a statistical t-test on a sample ($p<0.01$), comparing development time with and without the macro and the utilized Framework, demonstrating that the macro significantly accelerates the web application development process. In summary, this project underscored the importance of implementing a design module in Microsoft Excel, a source code generator that transforms the design into a spreadsheet, and the use of libraries to generate reports in PDF format. These components are fundamental in expediting the design and development of web applications, resulting in increased efficiency in the web application creation process.

Keywords: Control, code generator, rendering, ribbon, transaction, object.

INTRODUCCIÓN

En la actualidad el proceso de creación de una página web por parte de usuario, se hace muy dificultoso, es por eso que nos planteamos en determinar que si es posible implementar una macro en Visual Basic para Aplicaciones el cual permita desarrollar un software haciendo uso de Microsoft Excel, el cual permita generar código fuente para un entorno Web de manera ágil; esto debido a que el proceso de desarrollo de las aplicaciones web en la universidad se hace dificultoso, ya que no se cuenta con la cantidad de desarrolladores necesario, debido a la existencia de diferentes requerimientos de las oficinas.

Por lo indicado, se ha procedido a utilizar la herramienta de Microsoft Excel, así como las herramientas de Modelo – Vista – Controlador del Framework de Codeigniter, esto con el fin de poder implementar un generador de código a partir de un diseño en una hoja de Microsoft Exel, y esta pueda transformar el diseño elaborado a código fuente en formatos de PHP , HTML, CSS, JavaScript, JQuery y JSON. Para el presente proyecto se establecieron tres objetivos, siendo el primero la de elaborar de un módulo para el diseño de formularios en Microsoft Excel mediante Visual Basic para Aplicaciones el cual pueda ser almacenado en una base de datos de MySql; el segundo es la de desarrollar un generador de código en Visual Basic para Aplicaciones, el cual permita generar código fuente a partir del formulario diseñado en Microsoft Excel que sea capaz de traducir lo diseñado y traducirlo al Framework Codeigniter; y el tercero es la de implementar un módulo en Excel el cual permita diseñar reportes en formato PDF y en pantalla el cual pueda ser usado por la aplicación; por lo que para realizar el cumplimiento de estos se realizaron diferentes actividades para el logro de estos objetivos.

La presente investigación se divide en cuatro capítulos, En el capítulo I se considera la revisión de la literatura utilizado para la investigación, así como también los antecedentes; en el capítulo II se define el planteamiento del problema, justificación, objetivos e hipótesis de la investigación planteados; el capítulo III, se describen los materiales y métodos, así como la muestra, software, recursos utilizados, procedimientos desarrollados, así como los módulos desarrollados por cada objetivo planteado. En el capítulo IV se cuenta con los resultados, discusión y conclusión de la presente investigación. En la sección de Anexos se cuenta con el código fuente desarrollado, así como también se indica las métricas calculadas en relación al código fuente desarrollado.

CAPÍTULO I

REVISIÓN DE LITERATURA

1.1. Marco teórico

1.1.1. Aplicación Web

1.1.1.1. Internet

Internet en la actualidad es conocido como una gran red de redes, el cual ha permitido en la actualidad la interacción más fluida entre todos los seres humanos, es así que las instituciones privadas y públicas, así como también empresas pequeñas y grandes hacen uso de esta tecnología para promocionar algún producto o servicio, a su vez esta tecnología ha permitido el control de bienes y servicios mediante aplicaciones los cuales están conectados en una base de datos en lo que hoy se conoce como la nube, estas aplicaciones hacen uso de las tecnologías vía Web (World Wide Web), tales como HTML (HyperText Markup Language), CSS (Cascading Style Sheets), JavaScript, PHP (Hypertext Pre-Processor), entre otros, los cuales permiten el desarrollo dichas aplicaciones

1.1.1.2. HTML

El HTML de sus siglas de HyperText Markup Language (Lenguaje de marcado de hipertexto), en donde su primera versión fue publicada en 1991 por el británico Timohty John Berners-Lee, también es fundador de la W3C(World Wide Web Consortium), también es creador del protocolo HTTP (HyperText Transfer Protocol), creador de la URL(Uniform Resource Locator) y creador del primer navegador Web (*¿Qué Es y Para*

Que Sirve HTML y CSS? - Azul School, n.d.). El lenguaje HTML nos permite desarrollar páginas Web mediante marcadores, los cuales indican la forma en la que se van a mostrar en un navegador Web, el texto, imágenes, tablas entre otras opciones que vea de desarrollador, así como también se podrán incluir estilos y funciones. (Tabarés, 2021), en su artículo "HTML5 and the evolution of HTML; tracing the origins of digital platforms" habla como el HTML evoluciono a HTML5 y así de esta forma contribuye en los factores técnicos, económicos y sociales, para así generar un estándar de vida en todo el mundo.

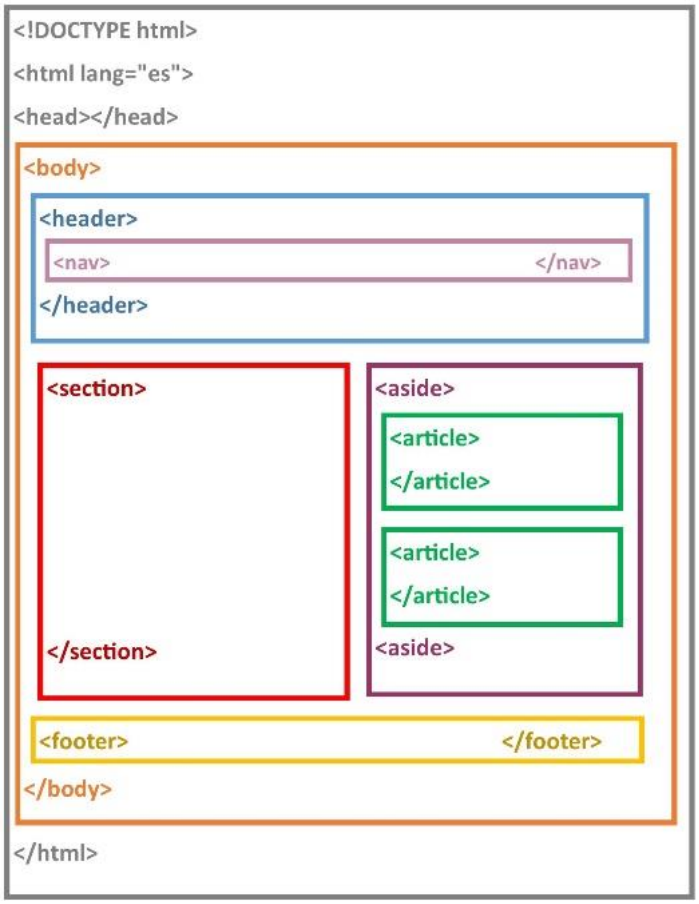


Figura 1. Estructura de un documento HTML

Fuente: https://grafismodigital.wordpress.com/2016/01/07/html5-estructura-basica-de-una-pagina-y-etiquetas/

1.1.1.3. CSS

El CSS (siglas en inglés de Cascading Style Sheets), en el lenguaje español es conocida como "hojas de estilo en cascada", el cual define el diseño de

un documento creado en lenguaje HTML en donde se le asigna un estilo a cada marcador del documento HTML.

1.1.1.4. JavaScript

El lenguaje de programación JavaScript fue desarrollado originalmente por Brendan Eich de Netscape con el nombre de Mocha, posteriormente a LiveScript y para finalmente como JavaScript (marca registrada de Oracle Corporation). Este es un lenguaje de secuencia de comandos, orientado a objetos, la utilización de JavaScript en documentos HTML permite que estas páginas Web, no sean estáticas, por lo contrario, estas sean mucho más dinámicas e interactivas con el usuario, JavaScript permite la manipulación de entornos gráficos 2D/3D, así como el manejo de multimedia, normalmente JavaScript es ejecutado del lado del cliente.

1.1.1.5. JQuery

JQuery es una librería de JavaScript, el cual simplifica el desarrollo de código JavaScript y a su vez permite más interactividad con nuestra página Web. Existen en la actualidad muchos plugin basados en esta librería que realizan tareas concretas y facilitan el desarrollo y la interactividad con nuestras aplicaciones. Esta librería debe ser incluido dentro de nuestro código HTML mediante el marcado <SCRIPT>, en el cual se debe de indicar la ubicación del archivo JQUERY.JS ya sea en las versiones más reciente o estable.

1.1.1.6. PHP

Del acrónimo Hypertext Preprocessor, es un lenguaje de código abierto, que es utilizado en el desarrollo de páginas Web, las cuales son creadas en base a este lenguaje, así como también se puede incluir en el código HTML, lo que permite que una página Web pueda ser más dinámica, PHP con lenguaje de programación que permite al desarrollador interactuar con base de datos, del lado del servidor, esto permite que el usuario pueda realizar consultas a datos mediante una interfaz HTML, el cual realizará la consulta a códigos en lenguaje PHP y retornando la información al cliente.

1.1.1.7. Bootstrap

Es un kit de herramientas de código abierto de tipo Front-End, el cual contiene una amplia gama componentes prediseñados de JavaScript. A su vez Bootstrap hace uso de la librería JQuery, CSS.

1.1.1.8. AJAX

JavaScript asíncrono y XML (Asynchronous JavaScript and XML), es un conjunto de técnicas de desarrollo Web que permiten que las aplicaciones Web funcionen de forma asíncrona, procesando cualquier solicitud al servidor en segundo plano, lo que permite una mayor interacción con el usuario.

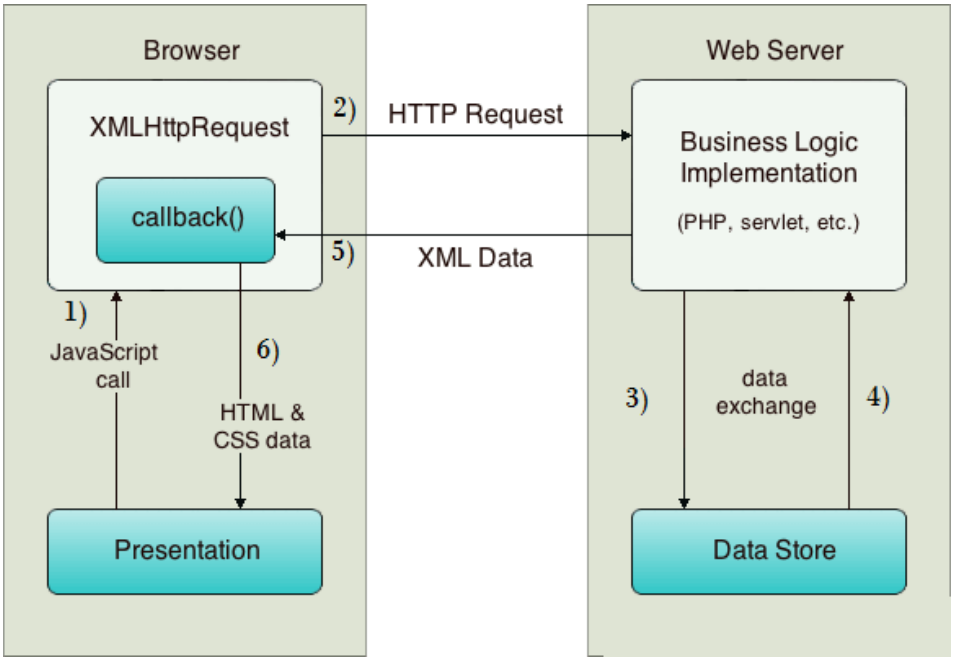


Figura 2. Funcionamiento de Ajax

Fuente: (*How Ajax Works - Javatpoint*, n.d.)

1.1.2. Framework

Un Framework, es una arquitectura tecnológica que está definido módulos que realizan acciones concretas los cuales son utilizados en desarrollo de aplicaciones web (Modelo, Vista y Controlador), que puede servir de base para la organización y desarrollo de software. Un lenguaje de programación puede disponer de diferentes Framework con los que se pueden implementar y desarrollar aplicaciones de manera más ágil. Los sistemas como .NET incorporan un Framework en sus

distribuciones más básicas, pero otros Framework basadados en PHP disponen de diversas empresas creadoras de esta herramientas o usuarios que realizan su publican en la internet, los cuales pueden ser usados opcionalmente. Encontraremos los Framework que se programan por el lado del cliente al lenguaje JavaScript el cual permite una programación de manera rápida y compatible con todos los ordenadores ya sean en plataformas Windows, Linux, MacOS entre otros. Mientras que los Framework para el desarrollo del lado del servidor se pueden encontrar los lenguajes de programación como PHP, .NET, Rubi, entre otros.

1.1.2.1. Modelo Vista Controlador (MVC)

Es un estilo de arquitectura de software que permite separar los datos de una aplicación en tres componentes, siendo la interfaz de usuario (Vista), el control a la base de datos (Modelo) y el control del flujo de la información (Controlador). Se considera como una arquitectura que ha demostrado estabilidad en diferentes aplicaciones de desarrollo Web y sobre todo la multitud de lenguajes y plataformas de que se vienen desarrollando.

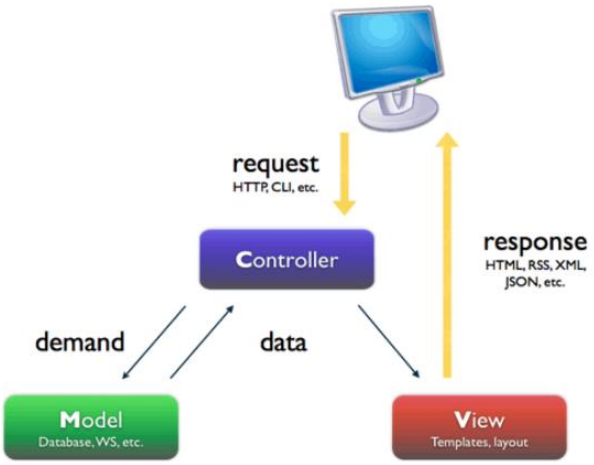


Figura 3. Funcionamiento de MVC
Fuente: https://stackoverflow.com/questions/5966905/what-is-the-right-mvc-diagram-for-a-Web-application

El modelo con la arquitectura de Modelo – Vista – Controlador (MVC) dispone de componentes útiles como son los de seguridad, creación y validación formularios, accesos rápido a base de datos, direccionamiento a procedimientos. El modelo inicialmente se implementó haciendo uso de

PHP, pero se puede hacer uso o realizar su implementación en distintos lenguajes de programación, utilizando este mismo concepto de desarrollo (Pop y Altar, 2014)

a) **Modelo (Model)**. Contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
b) **Vista (View)**. Manejador de la Interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.
c) **Controlador (Controller)**, Es el intermediario entre el Modelo y la Vista, es el encargado del controlar el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

1.1.2.2. Framework Codeigniter

Codeigniter es un Framework para aplicaciones Web de código abierto para crear y desarrollar aplicaciones o sitios Web dinámicos mediante la utilización del lenguaje PHP. Tiene como objetivo agilizar los procesos de desarrollo de aplicaciones web a los desarrolladores y que estos puedan diseñar proyectos de manera más ágil que desarrollando toda la estructura desde cero, para lo cual Codeigniter brinda una serie de bibliotecas para tareas comunes, así como una interfaz simple y una estructura lógica para acceder esas bibliotecas. Codeigniter es más rápido que muchos otros entornos (*Welcome to CodeIgniter4 — CodeIgniter 4.2.10 Documentation*, n.d.). Ahora mismo la versión más actual del Framework se puede descargar desde Github o desde la misma página de Codeigniter (Moutaouakkil y Mbarki, 2021).

- En 2014, Codeigniter fue adquirido por el Instituto de Tecnología de Columbia Británica y luego se anunció oficialmente como un proyecto mantenido por la comunidad.
- En 2019, la Fundación Codeigniter se formó para proporcionar un grupo de gestión perpetuo separado de cualquier otra entidad para ayudar a garantizar el futuro del marco.

a) Estructura de Codeigniter

La estructura de manejo de archivos de Codeigniter en su versión 3.0 para la administración del Modelo Vista Controlador este sujeto a carpetas según se muestra en la siguiente imagen.

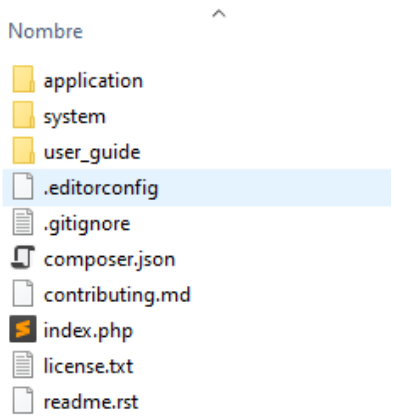


Figura 4. Estructura de archivos del Framework de Codeigniter

En donde las carpetas:

- **Application**: En esta carpeta se encuentra todo lo referente al control para el diseño de nuestras aplicaciones Web
- **System**: Es el manejador del Framework, dentro de esta carpeta se encontrará el core (núcleo) del Framework Codeigniter, así como manejadores de base de datos, fuentes, librerías ya incluidas, helpers e idioma.
- **User_guide**: Guia de uso para el desarrollador, como se indica aquí se encuentra la forma en que se puede utilizar este Framework Codeigniter

b) Modelo

En los Framework las operaciones de manejo de la base de datos se deben de efectuar dentro del modelo, para que de esa forma se puedan reutilizarse fácilmente en el desarrollo de las aplicaciones Web. Los modelos (models) son el lugar donde recupera, inserta y actualiza información. Para esto se debe de ubicar en la sub carpeta

application/models/

En la cual se deberá de generar un archivo con el nombre que elija, se sugiere que inicie con una Mayúscula y con la terminal "_model" y haciendo una extensión en la clase CI_Model como se describe en la siguiente imagen.

```php
<?php
class News_model extends CI_Model {

        public function __construct()
        {
                $this->load->database()
        }
}
```

Figura 5. Estructura de una clase model extendido de la clase CI_Model de Codeigniter

c) Vista

Es la representación visual de los datos creados como diseño de página Web y todo lo que tenga que ver con el diseño gráfico va en este componente del Framework. El controlador no se preocupará del cómo se mostrarán la información que se le proporcionará, esta acción será realizada por la vista.

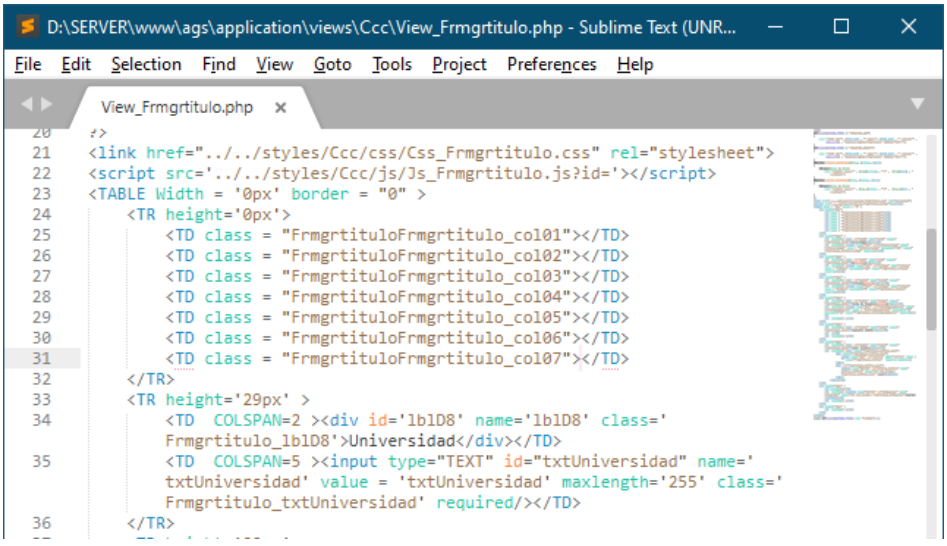


Figura 6. Codificación en HTML de una vista del Framework de Codeigniter

d) Controlador

El controlados es la capa que sirve de intermediario entre las vistas y los modelos del Framework, el controlador va ser el encargado de procesar las solicitudes que el usuario realice mediante la interfaz que se le mostrará en el navegador Web. Sin embargo, su responsabilidad no es manipular directamente la base de datos, ni mostrar ningún tipo de salida, a lo contrario sino el controlador va enlazar los modelos y las vistas en el desarrollo de nuestras aplicaciones.

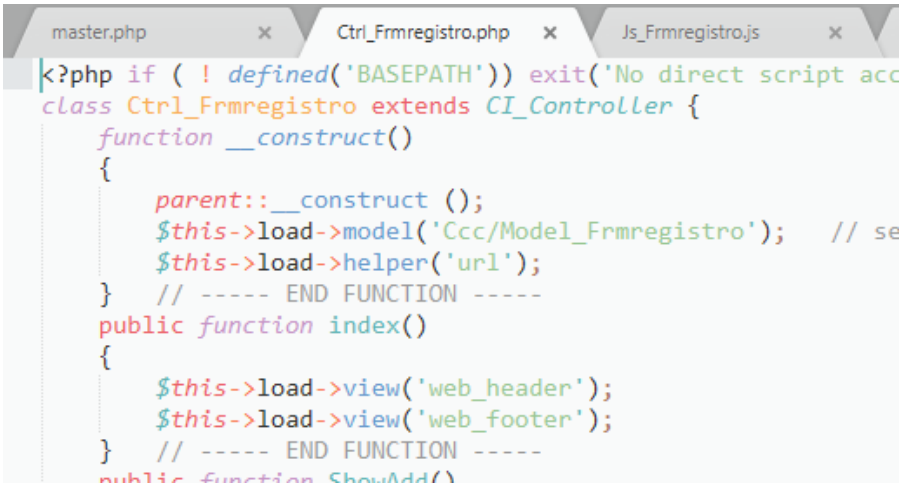


Figura 7. Clase de control extendido del controlador del Framework de Codeigniter

e) Bibliotecas de Código

Codeigniter, es uno de los Framework PHP más accesibles, el cual contiene bibliotecas de código para realizar diversas funciones útiles. Las librerías o bibliotecas de código en Codeigniter son clases preparadas para realizar tareas típicas en el desarrollo de nuestras aplicaciones Web, estas clases son códigos conocidos como programación orientada a objetos (POO). Algunas de las tareas de estas librerías son para trabajar con bases de datos, FTP, upload de archivos, sesiones, calendario, entre otras tareas.

1.1.3. Base de datos

Se conoce como una base de datos una serie de datos los cuales están organizados y relacionados entre sí, los cuales son recolectados y almacenados electrónicamente en un sistema de computadora, conocido como un servidor de base de datos, dichos datos están estructurados en tablas, registros y campos, los cuales pueden ser

controladas o administradas por un Sistema Gestión de Base de Datos (SGBD) o del inglés DataBase Management System (DBMS). En conjunto, los datos y el SGBD. La mayoría de las bases de datos utilizan el lenguaje estructurado de consultas (SQL) para realizar las consultas a la base de datos y poder ser procesadas por el lado del cliente.

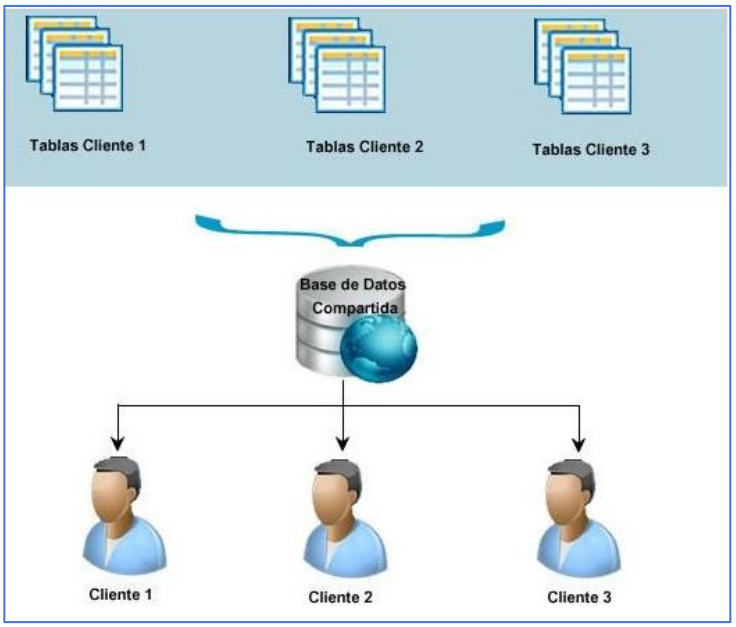


Figura 8. Estructura de una base de datos

Fuente: https://sites.google.com/site/mariluz20188/2-4-estructura-de-la-base-de-datos

a) ODBC

Del inglés Open Database Connectivity en español Conectividad de base de datos abierta ODBC) es una API de estándar abierto el cual nos permite conexiones a bases de datos, esto mediante el uso de instrucciones de ODBC, el cual le permite acceder o conectarse a diferentes base de datos, tales como Microsoft Access, Excel, FoxPro, dBase, Interbase, MySql, Sql Server, entre otros, para esto es necesario contar con un módulo de controlador independiente el cual le permitirá realizar esta tarea, por lo que para este proyecto se está haciendo uso del controlador de MySql ODBC en su versión 5.4, el cual tiene la siguiente característica para realizar la conexión al servidor de base de datos.

Conexión a una base de datos:

La cadena de conexión con instrucciones de ODBC se realiza de la siguiente forma para una conexión a MySQL o MariaDB:

DRIVER={MySQL ODBC 5.3 unicode Driver};

SERVER=$SERVER;

DATABASE=$DB;

UID=$USER;

PWD=$PWD;

PORT=3306;

Donde:

Driver, determinar el módulo de controlador con el cual se realizará la conexión a la base de datos,

Server, indicada mediante una dirección IP, nombre o dominio de conexión, donde se encuentra ubicado el servidor de base de datos

Database, se indica el nombre de la base de datos de la cual deseamos realizar tareas, el cual se encuentra ubicada en el servidor.

UID, se especifica el nombre del usuario el cual desea realizar la conexión a la base de datos antes indicada, para esto el usuario deberá de tener los permisos respectivos para realizar la conexión, caso contrario el servidor de base de datos lo rechazará.

PWD, aquí deberá de especificar la contraseña del usuario, para poder acceder a la base de datos.

PORT, determina mediante qué puerto se debe de realizar la conexión al servidor de base de datos, por lo general en MySQL o MariaDB se realiza mediante el puerto 3306, puerto TCP/IP por defecto, pudiéndose cambiar según se indique en la configuración del servidor.

b) Lenguaje Estructurado de Consultas (SQL)

SQL es definido como lenguaje estándar para realizar consultas, definir y/o manipular información de una base de datos relacional; en donde esta se crea como un conjunto tablas relacionadas mediante valores, en donde se puede realizar consultas para recuperar información que pueden derivarse de una tabla o combinación de tablas, la cuales se encuentran relacionadas mediante un valor específico (*Lenguaje de Consulta Estructurada (SQL) - Documentación de IBM*, n.d.).

1.1.4. Programación con macros para Visual Basic para Aplicaciones

a) Microsoft Excel

Es un programa que fue desarrollado por la compañía Microsoft. Este software permite al usuario realizar tareas contables, financieras, estadísticas esto debido a las funciones que son incorporadas a este software por la compañía, esto para que al usuario se la haga más fácil realizar todo tipo de tareas cotidianas, y agilizar el proceso de producción, a su vez Microsoft incorporó Visual Basic para Aplicaciones fue lanzado 1991, siendo esta su primera versión, con la intención de simplificar la programación para lo cual implemento un ambiente desarrollo amigable para programadores, por lo que para el presente proyecto haremos uso de este beneficio que se cuenta, para desarrollar aplicaciones Web utilizando módulos que nos permitan compilar nuestros formularios y llevarlos un servidor Web. Microsoft Excel por su formato de trabajo en base a celdas nos va permitir en este proyecto la elaboración de formularios ya hechos por los usuarios y así de esta forma poder compilarlos a un formato HTML (basados en tablas), PHP, JavaScript y CSS, haciendo uso de las tecnologías MVC (Módulo, Vista y Controlador) con Framework adecuado para este fin.

b) Ribbon

Un ribbon es un espacio de nombres en la barra de herramientas de Microsoft Office en donde este puede contener componentes, controles, colecciones, clases entre otros que son compatibles, los cuales permiten personalizar el interfaz de opciones de las aplicaciones de Microsoft Office, para este caso de Microsoft Excel.

Figura 9. Cinta de opciones del sistema, utilizando el Ribbon de Microsoft Office

c) Visual Basic para aplicaciones

Microsoft Visual Basic para Aplicaciones (VBA) es una versión del lenguaje de programación Visual Basic, para entornos de Microsoft Office, CorelDraw, Autocad entre otros, el cual le permite adicionar complementos, módulos, formulas y otras acciones que son ejecutados por los programas antes mencionados, los cuales no se encuentran dentro de sus funciones propias de la aplicación. A su vez Visual Basic para Aplicaciones (VBA) permite utilizar las llamadas a las funciones API de Microsoft Windows, con tipos de datos de 32 bits, el cual nos permitirá el manejo de punteros, controladores y funciones avanzadas de Kernel de Windows, para asi tener un mayor control del sistema y dar mayores opciones para el desarrollador de aplicaciones, para el presente proyecto se pretende utilizar estos beneficios en Microsoft Excel tanto para entornos de 32 bits y 64 bits, teniendo como referencia que para entornos de 64 bits las llamadas a la API de Windows se deberá de realizar agregando a las declaraciones las palabras clave *longptr, longlong y ptrsafe* para que el código pueda ser ejecutados en los entornos de 64 bits.

- LongPtr. VBA incluye el alias de tipo variable LongPtr. El tipo actual de datos que resuelve LongPtr depende de la versión de Office que se esté ejecutando; LongPtr resuelve Long en versiones de 32-bit de Office y LongPtr resuelve LongLong en versiones de 64-bit de Office. Use LongPtr para punteros y controladores.
- LongLong. El tipo de datos LongLong es un entero de 64 bits firmado que solo está disponible en versiones de 64 bits de Office. Use LongLong para integrales de 64 bits. Las funciones de conversión deben usarse para asignar explícitamente LongLong (incluido LongPtr en plataformas de 64 bits) a tipos enteros más pequeños. No se permiten las conversiones implícitas de LongLong a integrales menores.
- PtrSafe. La palabra clave PtrSafe confirma que una instrucción Declarar es segura para ejecutar en versiones de 64 bits de Office.

Todas las instrucciones **DECLARE** deben incluir la palabra clave PtrSafe cuando este se va ejecutar en versiones de 64 bits (VBA7). Es importante comprender que simplemente agregar la palabra clave PtrSafe para una instrucción DECLARE solo significa que la instrucción DECLARE tiene como destino explícito el extracto de 64 bits (*Información General Sobre Visual Basic for Applications de 64 Bits | Microsoft Learn*, n.d.).

Ejemplo de instrucción de VBA heredada no modificada Declarar:

Win32 – Visual Basic para Aplicaciones (VBA6)
Declare Function GetActiveWindow Lib "user32" () As Long
Win64 – Visual Basic para Aplicaciones (VBA7)
Declare PtrSafe Function GetActiveWindow Lib "user32" () As Long

Definición de llamadas de código en 32 bits o 64 bits

Para que un código pueda ser ejecutado o compilado en sistemas de 32 bits o 64 bits, las declaraciones se deberán de hacer de la siguiente forma:

Código para la declaración
#if Vba7 then ‘Determina si se utilizando una plataforma más actualizada que soporta VBA7 ‘Aqui el código para la ejecución VBA7 (Windows de 64 bits) #if Win64 then 'Aqui las declaraciones para funciones en entornos de 64 bits #else 'Aqui las declaraciones para funciones en entornos de 32 bits #end if #else ‘Aqui las declaraciones para plataformas que soportan solo VBA6 o inferiores #end if Ejemplo de uso #if Vba7 then Declare PtrSafe Sub... #else Declare Sub...

```
#endff
```

1.1.5. Compilador

Un compilador es mecanismo el que interpreta la comunicación entre el ordenador y una persona, en donde se establece el envío de datos y la recepción del mensaje de tipo textual, en donde la persona pueda interpretar lo indicado, podemos ejemplificar esto como el envío de información al ordenador por teclado y este devuelve un resultado mediante un mensaje sobre las acciones realizadas por el ordenador, para lo cual para que el ordenador realice la tarea indicada es necesario de contar con un traductor para la interacción entre hombre – maquina (*JAVA A TOPE: TRADUCTORES Y COMPILADORES CON LEX/YACC, JFLEX/CUP Y JAVACC. EDICIÓN ELECTRÓNICA*, 2005).

Definiremos como un traductor a un programa el cual permite traducir desde un texto o programa escrito en lenguaje natural, a un código máquina, en donde este código lo entenderá el ordenador, devolviendo información en mensaje natural, el cual será entendido por una persona.

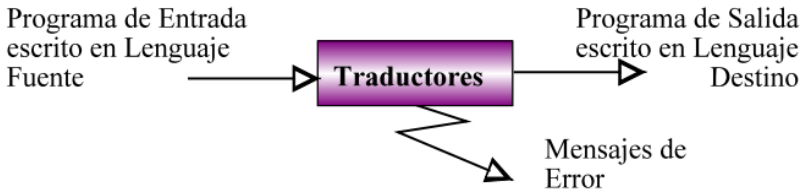


Figura 10. Esquema preliminar de un traductor
Fuente: https://leo-yac.wixsite.com/lenguajes-formales/estructura-de-un-traductor

1.1.6. Generador de Código fuente

El generador de código viene a ser una herramienta o sistema el cual nos permite realizar la automatiza y la generación de código fuente a partir de un diseño y el cual genera código a un lenguaje de programación con las especificaciones previas durante el diseño. En lugar que el desarrollado genere manualmente el código necesario para un sistema determinado el generador de código le permitirá ahorrar tiempo y esfuerzo para el desarrollo de un sistema, esto debido a que el generador de código producirá automáticamente partes del código o incluso el código completo de un módulo específico (*¿Qué Es Un Generador de Código? - Definición de Techopedia - Desarrollo 2023*, n.d.).

Las principales características de un Generador de Código son:

- Lenguaje. Se trata del lenguaje en el cual se ha de generar el código puede ser libre o de propietario.
- Portabilidad. Debe ser posible ejecutarlo en diferentes plataformas.
- Generación del esqueleto. Si solamente genera un esqueleto, será necesario implementar el resto del código mediante programación.
- Modificación. Debe de permitir acceder directamente al código para poder optimizarlo y completar acciones adicionales a la generación.
- Pantallas e informes. Esta característica permitirá el desarrollo de la interfaz de la aplicación.

(*Generador de Código - GlosarioIT: Glosario Informático*, n.d.)

1.1.7. Modelo para el desarrollo rápido de aplicaciones

Scrum es un proceso trabajo ágil el cual es utilizado para el desarrollo de proyectos de software. Este método de desarrollo, el cual se basa los principios de transparencia, inspección y adaptación, el cual promueve las buenas prácticas de trabajar colaborativamente con equipos de desarrollo, y a su vez promueve una colaboración estrecha entre los miembros del equipo de desarrollo y los interesados.

En este método de desarrollo, el trabajo el trabajo lo divide en ciclos, los cuales son llamados "Sprints", que pueden durar de entre de una a cuatro semanas. En donde cada sprint es una iteración que se enfoca en entregar incrementos en la funcionalidad del producto. Los principales componentes del modelo Scrum son los siguientes:

a) Planificación, se realiza en el primer día, el cual consta de dos (2) partes que son: Selección de requisitos y Planificación de la iteración, en el punto primero, el cliente presenta la lista de requisitos priorizado del proyecto, del cual, el equipo de trabajo selecciona los requisitos que cuenta con más relevancia y priorización. En el segundo, el equipo se encargará de realizar una lista de tareas, las cuales son necesarias para el desarrollo de los

requisitos, así mismo los miembros del equipo se asignarán tareas o se auto organizan para el desarrollo, e incluso lo pueden realizar en parejas.

b) Ejecución, durante la ejecución el equipo de trabajo realiza trabajos de sincronización, mediante un Scrum Taskboard, en donde se realiza la inspección del trabajo, para poder realizar las adaptaciones que son necesarias, el cual permite cumplir con los objetivos establecidos en los requisitos.

c) Inspección, esto se realizó en el último día de las iteraciones, la cual consta de dos (2) partes, que son: Revisión y Retrospectiva, en la revisión o demostración el equipo realiza la presentación de los requisitos completados del producto, en donde el cliente puede realizar algunos cambios lo cuales pueden ser necesarios. En la retrospectiva se, el equipo de trabajo analiza su modo de trabajo realizado y que problemas podrían impedir un progreso adecuado(*Qué Es SCRUM – Proyectos Ágiles*, n.d.)

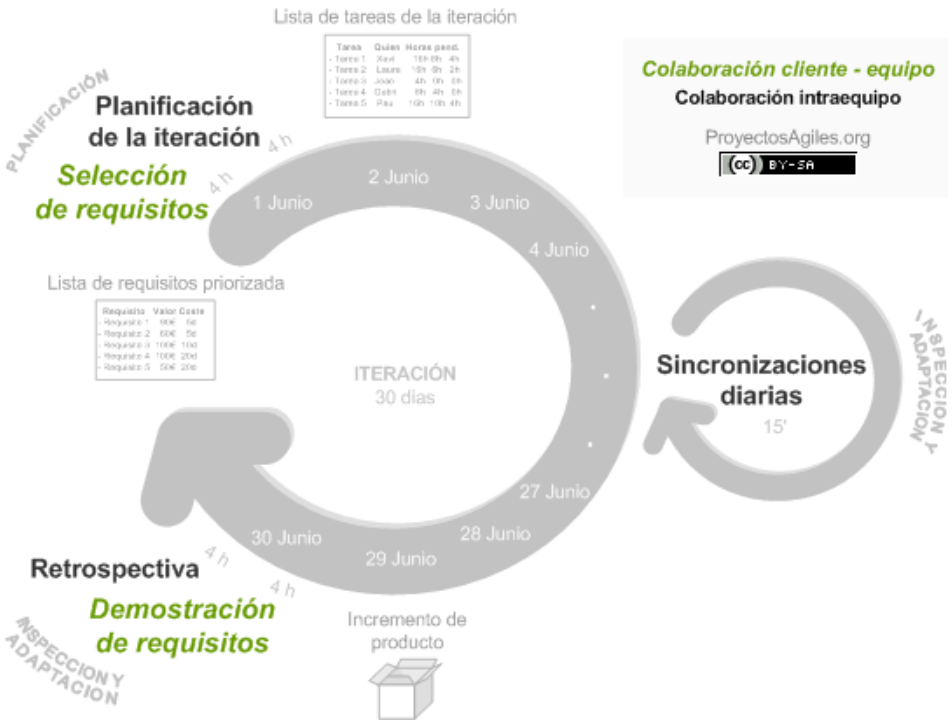


Figura 11. Proceso de desarrollo, por el método Scrum

Fuente: https://proyectosagiles.org/que-es-scrum/

1.1.8. Métricas de software

Durante la construcción y desarrollo de una aplicación, se hace necesario medir el cumplimiento de ciertos parámetros, así como en una construcción se define la

altura y ancho de alguna estructura, así como sus sub componentes del mismo, los cuales pueden ser medidos en metros o centímetros. En el desarrollo de software también es necesario de poder obtener dichas medidas, para poder determinar si lo procedimientos desarrollados son óptimos o si estos podrían ocasionar algún tipo de contratiempo durante el proceso mismo de desarrollo, para lo cual es necesario identificar los tipos de métricas a ser utilizado para determinar el impacto que puede ocasionar en desarrollo en sí.

1.1.8.1. Métrica de Ciclo de Tiempo

Esta métrica de Ciclo de Tiempo del inglés Cycle Time (CT), esta métrica nos indicará el tiempo que un proceso tarda en entregar una tarea desde el momento en que se inicia con el procesado, hasta el momento en que dicha tarea cumple con los procesos definidos en este. Por tanto, una de las medidas que podremos obtener es la velocidad del procesamiento de una tarea específica, así como también podremos obtener el rendimiento o estado de un proceso.

La métrica de Ciclo de Tiempo nos medirá la eficiencia del proceso de desarrollo de un software, el cual nos permitirá obtener indicadores de capacidad, así como también nos permitirá encontrar los cuellos de botella o sobre carga de procedimiento en un punto específico, por lo que esto favocerá y permitirá una mayor velocidad y fluidez.

Un Ciclo de Tiempo corto o reducido es un indicador de un proceso más eficiente a lo contrario que un Ciclo de Tiempo con tiempo elevados determinaría procedimientos ineficientes, los cuales generarían retrasos en la entrega (*Cycle Time: Qué Es, Cómo Medirlo y Mejorarlo En Desarrollo de Software*, n.d.)

1.2. Antecedentes

"Using Excel based war simulator to support tactical wargaming", en este artículo se presenta como hacer uso de un software informático y así de esa forma beneficiarnos, como un simulador de batallas, además este esta implementado en Microsoft Excel, el cual apoya en el entrenamiento táctico, como parte de un juego de guerra manual básico a nivel de unidad. Este simulador permite resolver los resultados de las batallas de forma

neutral, esto en contraposición con opiniones potencialmente sesgadas de los participantes participan en el juego. (Hongisto y Saastamoinen, 2023).

Aplicabilidad de los criterios del ensayo EXCEL a una cohorte del mundo real de intervenciones coronarias percutáneas del tronco principal izquierdo sin protección. El estudio Evaluación de XIENCE versus EXCEL comparó la intervención coronaria percutánea (PCI) con stents de fármacos de segunda generación y la cirugía de bypass en la enfermedad del tronco principal izquierdo desprotegido (ULMD). De 246 pacientes sometidos a PCI-ULMD entre enero de 2018 y diciembre de 2021, el 39% se clasificó como "similar a EXCEL". Estos pacientes mostraron menores eventos adversos cardiovasculares y cerebrales a un año (7% vs 17%) y a largo plazo (19% vs 37%) en comparación con el grupo "no similar a EXCEL", a pesar de tener perfiles cardiovasculares iniciales comparables. La diferencia persistente en resultados cuestiona la aplicabilidad generalizada de los resultados del ensayo EXCEL en la práctica clínica habitual para pacientes de PCI-ULMD contemporáneos. (Castaldi *et al.*, 2023)

“EMMTE: An Excel VBA tool for source apportionment of nitrate based on the stable isotope mixing model”, en este artículo se ha desarrollado una herramienta de modelo de mezcla de miembros finales en Excel (EMMTE), el cual se utiliza para identificar las fuentes de nitrato en el campo de la geoquímica, para lo cual se ha integrado con una hoja de cálculo de Excel, utilizando la simulación de Monte Carlo y la desviación relativa de restricciones, para lo cual se incluyeron en el algoritmo el que permite resolver la ecuación de balance de masa. (Cao *et al.*, 2023)

“Simulating SQL injection vulnerability exploitation using Q-learning reinforcement learning agents “, en este artículo realizaron la simulación de la explotación de vulnerabilidades de inyección SQL utilizando agentes de aprendizaje por refuerzo de Q-learning, en la consideraron la simplificación de la dinámica en ataques por técnicas de inyección a SQL, al presentarse este problema como un desafío en seguridad, para lo cual estos implementaron agentes de aprendizaje de refuerzo, los cuales fueron tuvieron la tarea de aprender alguna política efectiva el cual afectar la inyección en una consulta SQL (Erdődi *et al*., 2021).

Transformación del modelo Struts al modelo Codeigniter. Los frameworks están siendo cada vez más empleados en el desarrollo de sistemas de información, ya que permiten la reutilización del diseño para un dominio completo, lo que resulta en una disminución de

costos y una mejora en la calidad del software. En el enfoque MDA, el uso de modelos, metamodelos y transformación de modelos es fundamental. Este proyecto de investigación busca identificar una forma de convertir modelos de struts a través del lenguaje de transformación QVT-O y técnicas de ingeniería basadas en modelos hacia modelos de Codeigniter. (Moutaouakkil y Mbarki, 2021)

"GeoBalance: An Excel VBA program for mass balance calculation in geosciences", implementaron una aplicación en Visual Basic para Aplicaciones, el cual es un programa para el cálculo de balance de masa denominado GeoBalance, el cual se encuentra integrado con una hoja de Excel, con propiedades habilitadas para el manejo de macros. El algoritmo que trabajaron está basando en resolver problemas de mínimos cuadrados usando pseudo-inversa de matriz de descomposición de valores singulares (SVD), los usuarios pueden determinar el método de solución, que pueden ser restringido como no restringidos. (Li *et al*., 2020).

El artículo "Generando un metamodelo PHP usando Xtext Framework". PHP ha emergido como el idioma predominante en el desarrollo de aplicaciones web. En el enfoque MDA, la representación del código fuente mediante modelos que se ajustan a un metamodelo es esencial. Este proyecto de investigación se propone descubrir un método para obtener un metamodelo y un analizador del lenguaje PHP, empleando el marco Xtext y prácticas de ingeniería apoyadas en modelos. (Moutaouakkil & Mbarki, 2021)

El uso de Genie 2000 y Visual Basic para Aplicaciones en el entorno de Microsoft Excel, La radiación γ los cuales son emitidas por materiales de construcción, esto se calcula a partir de los valores de índices de actividad y se expresa como valores de índice de concentración de actividad. Este artículo describe como aprovechar la flexibilidad de Genie 2000 en un libro de Microsoft Excel, el cual es utilizado para programar aplicaciones, en donde se hace uso de bibliotecas de acceso a datos nucleares de Canberra, así como también el uso de herramientas de procedimientos por lotes. (Suárez-Navarro *et al.*, 2018).

El artículo "Análisis y aplicación práctica de frameworks PHP en el desarrollo de sistemas de información web". El texto examina y contrasta varios marcos de programación PHP a través de diferentes criterios, seleccionando, para un análisis más exhaustivo. Se detalla la estructura y características fundamentales de estos marcos, incluyendo su sistema de enrutamiento y plantillas. Se lleva a cabo una evaluación de rendimiento utilizando una

sección específica del sistema de reserva de boletos para comparar la eficacia de los marcos mediante la herramienta ab.exe de Apache Benchmark. Los resultados se utilizan para ofrecer recomendaciones a desarrolladores que buscan seleccionar un marco apropiado para proyectos web reales. Además, se planea expandir las capacidades del sistema utilizando servicios web basados en estándares y protocolos abiertos. (Prokofyeva y Boltunova, 2017).

"Descubrimiento automatizado de ataques de inyección de código JavaScript en aplicaciones web PHP" del inglés "Automated Discovery of JavaScript Code Injection Attacks in PHP Web Applications", en el artículo se analiza deficiencias de soluciones defensivas, en los problemas de rendimiento sobre todo en los ataques por inyección mediante JavaScript como pueden ser los ataques de tipo Cross-Site-Scripting (XSS), en donde en su artículo propuso sistema de detección automatizada, el cual realiza un análisis de los diferentes ubicaciones posibles en un sitio Web, para poder identificar vulnerabilidades mediante inyección por JavaScript. (Gupta y Gupta, 2016).

En el artículo "An improved solution of local window parameters setting for local singularity analysis based on Excel VBA batch processing technology ", dieron una solución mejorada de configuración de parámetros de ventana local para un Análisis local de singularidad basado en la tecnología de procesamiento por lotes Excel VBA desarrollado y se implementó con ArcGis 10 para separar las anomalías del fondo de acuerdo con un umbral supuesto es esencial tanto en la geoquímica de exploración como en la geoquímica ambiental, el fractal se utilizó originalmente para caracterizar la autosimilitud de los objetos geométricos en diferentes escalas (Zhang *et al*., 2016).

En el artículo Estrategia Guiada por Modelos para incluir Aspectos de Seguridad en Sistemas Empotrados Basados en Servicios (WebA Model-Driven Strategy for Including Security Aspects in Web Services-Based Embedded Services), en donde explican que en los nuevos sistemas distribuidos modernos, la seguridad tienen un papel importante preponderante, por lo que se debe poner especial atención en la seguridad, sobre en todo en los inicios del desarrollo. (Gallino *et al*., 2014).

En la investigación BEARKIMPE-2, en donde se elabora una aplicación utilizando Visual Basic para Aplicaciones en donde se caracteriza al hierro granular en estudios de tratabilidad, en donde se elabora un algoritmo cinético adecuado, el permitió investigar la velocidad de reacción de un contaminante con hierro granular (GI), para lo cual era

primordial la optimización de la barrera reactiva permeable, esto en términos de su reactividad. El algoritmo cinético de hierro, permite determinar la constante de velocidad de superficie, así como los parámetros de sorción. Dicho código fue escrito dentro del entorno de Microsoft Excel, haciendo uso de VBA. (Firdous y Devlin, 2014)

Se desarrolló un programa extensible de CorelDraw haciendo uso VBA para la elaboración geológica (CGDK: An extensible CorelDraw VBA program for geological drafting), en la cual se implementaron herramientas completas para dibujar mapas geológicos, perfiles geológicos, Secciones, y columnas de Stratigraphic, haciendo uso CorelDraw y Visual Basic para Aplicaciones, a su vez se creó su setup para que pueda ser instalado o desinstalado en equipos de cómputo (Qiu *et al*., 2013).

El artículo "Teaching Excel VBA as a problem solving tool for chemical engineering core courses " propusieron y enseñaron que Excel (VBA) como herramienta de resolución de problemas para cursos básicos de ingeniería química está enfatizado de cómo enseñar VBA en Excel y utilizarlo en la ingeniería química, el cual explica que el uso de software debe estar basado en el "enfoque a herramientas", que un profesional va usar durante su labor y mas no en el "enfoque informático" ya que este se centra en sintaxis del lenguaje de programación sin tener conexión a los problemas de ingenierías relevantes (Wong y Barford, 2010).

El artículo "Implementación de un modelo de árbol de decisiones basado en COM con VBA en ArcGIS" (Implementation of a COM-based decision-tree model with VBA in ArcGIS), la macro desarrollada CART VBA integra todos los componentes. El código se escribió en VBA y se implementó como una plantilla del cual se puede derivar cualquier proyecto de ArcGIS. También el modelo CART es fácil de aplicar y las funciones son accesibles por los usuarios que no tienen conocimiento experto de modelado y programación, la macro VBA descrita que implementa el modelo CART es una herramienta útil para la evaluación de la contaminación por metales pesados la macro VBA llena un vacío importante en la funcionalidad de ArcGIS, dado que los modelos de árbol de decisión no pertenecen a la norma (Cheng *et al*., 2010).

"GeoPlot: programa Excel VBA para trazar datos geoquímicos", Microsdr Excel es ampliamente utilizado por geoquímicos por su capacidad de almacenamiento, aunque principalmente ofrece funciones elementales de organización y representación gráfica. Los macros previos de Excel, como Sidder, Christie, Langmuir y Marshall, no logran

satisfacen algunas necesidades actuales. A pesar de que PetroPlot puede generar ciertos gráficos, no dispone de las herramientas necesarias para los gráficos triangulares y los diagramas de discriminación. En contraparte, GeoPlot, un reciente programa desarrollado en VBA, se vincula con Excel y provee funcionalidades como trazados X – Y, diagramas triangulares, de araña y de discriminación. GeoPlot incluye funciones avanzadas como el cálculo de fórmulas, la norma CIPW y la posibilidad de incorporar nuevos valores y diagramas. (Zhou & Li, 2006)

En la investigación "Solución de las ecuaciones diferenciales ordinarias de orden superior usando Matlab/Simulink". El uso de matemáticas y software en diversas áreas de la vida es fundamental para resolver problemas. En este estudio, se emplea MATLAB/Simulink en la computación científica para simular la solución de ecuaciones diferenciales de orden superior. Se analizan ecuaciones de primer y segundo orden homogéneas, además de ecuaciones de tercer y cuarto orden, tanto homogéneas como no homogéneas con coeficientes constantes. Se realizan simulaciones del crecimiento poblacional de Perú y de un modelo de Quarter-Car utilizando datos históricos y la herramienta Simulink. Los resultados del Modelo de Malthus revelan un alto coeficiente de determinación de 0.9913, lo que sugiere que aproximadamente el 99.13% de la variabilidad en los datos es explicada por el modelo.(Santander Mamani, 2023).

"Extracción automática de metadatos para la administración del Repositorio Institucional de la Universidad Nacional del Altiplano Puno", La investigación se enfoca en mejorar la obtención de información clave y la publicación de documentos científicos en Repositorios Institucionales. Se utiliza el software "E-MeRI" en un conjunto de 1518 documentos de investigación para agilizar procesos que normalmente son largos. Se desarrolla un sistema empleando capas de programación y se realiza una prueba estadística para comparar datos, en donde se crea un algoritmo basado en procesamiento de lenguaje natural para extraer datos automáticamente, mostrando una gran eficacia comparada con otras herramientas. (Herrera Urtiaga, 2022).

"Análisis y solución de vulnerabilidad de seguridad en aplicaciones web y métodos de protección anti robot y HTTP request". La investigación se centra en mejorar la gestión y supervisión para evitar intrusiones no autorizadas de crackers o nuevos programadores en los servidores de datos. Propone introducir un sistema web que administre los privilegios de seguridad sin limitarse a un solo servidor, con el fin de registrar, validar,

evaluar y publicar información sobre las aplicaciones en ese servidor. Este proyecto, de enfoque descriptivo, detalla las debilidades de seguridad y sugiere estrategias de protección más adaptables, destacando la adaptabilidad dinámica durante el ciclo de vida del software y las aplicaciones en la nube en lugar de la previsibilidad estática. Se llevó a cabo una auditoría en ciertas páginas web seleccionadas, lo que permitió evaluar la seguridad y el rendimiento en las solicitudes POST y GET, considerando la eficiencia en la transferencia de datos. (Canahuire Chambi, 2020).

Framework para el desarrollo escalable de sistemas de información en la nube orientado a MYPES. El objetivo de esta investigación es crear un marco de trabajo escalable para sistemas de información en la nube. Se emplearon modelos de programación, gestión de infraestructura en la nube, modelos de calidad de software y pruebas de carga para evaluar el desempeño del nuevo marco. Los resultados indican una mejora del 34% en el rendimiento de un sistema implementado con este marco en comparación con uno basado en la estructura tradicional cliente-servidor.(Calderon Vilca, 2020).

Arquitectura pervasiva con tecnologías WebRTC híbridas para el desarrollo de un framework modelo vista controlador de tiempo real. Los Frameworks, en particular, imponen reglas estrictas en la forma en que se llaman las funciones y se procesan los lenguajes como PHP o ASP.NET. Por eso, en la investigación propone una nueva aproximación utilizando tecnologías WebRTC en un esquema pervasivo para mejorar la arquitectura Modelo Vista Controlador (MVC). Este enfoque, llamado Pervasive Model View Controller (P.M.V.C.), busca ofrecer una versión mejorada con una jerarquía más clara en la abstracción de clases en comparación con otros Frameworks. También busca mejorar la inyección de dependencias (Dependency Injection) para permitir una mayor flexibilidad en la utilización de objetos para la conexión a bases de datos, modelos y vistas finales para la generación de resultados HTML. Además, se integrará soporte para la comunicación en tiempo real de eventos. (Laura Murillo, 2019).

CAPÍTULO II

PLANTEAMIENTO DEL PROBLEMA

2.1. Identificación del problema

En la actualidad la implementación de software basado en tecnologías Web, implica tiempo y costo para la construcción y desarrollo de estas aplicaciones. Por lo que, si una entidad desearía desarrollar su propia aplicación a medida, este tendría que tener un alto conocimiento en programación en los diferentes entornos existentes, así como también conocimientos intermedios en bases de datos, lo cual le conllevaría a contratar un personal adecuado y calificado para el desarrollo de dicha aplicación, en donde se le tendría que explicar y enseñar todos los procedimientos para el desarrollo de dicha aplicación, en muchos casos llevándose varias entrevistas, pruebas y consultas por parte del desarrollador.

2.2. Enunciado del problema

¿Es posible desarrollar una macro en Microsoft Excel el cual permita elaborar un formulario en una hoja de cálculo y que permita compilarlo para un entorno Web de manera ágil?

2.3. Justificación

El desarrollo de aplicaciones para internet ha mejorado con la aparición de las tecnologías MVC (Modelo, Vista y Controlador) utilizadas por los Framework existentes hasta el momento, lo que ayuda en gran parte al desarrollo de las aplicaciones vía Web, pero esto no hace suficiente, en la actualidad en una sociedad, donde la tecnología ha predominado y agilizado los procesos, las Instituciones no apartados de esta tecnología es que solicitan el desarrollo de aplicaciones más rápidas, que a un inicio solo les permita registrar

información básica y posterior a eso realizar reportes personalizados. En base a aplicaciones ya desarrolladas es que se observó que en la mayoría de aplicaciones siguen el patrón de registrar datos en base a las siguientes opciones de agregar, eliminar, modificar, grillas y la opción de reportes.

Así como también se observó que el uso Microsoft Excel, es usado como una herramienta para el manejo de información, así como también para dar soluciones en las diferentes áreas de trabajo, a su vez el uso de Visual Basic para Aplicaciones (VBA), ha ayudado a mejorar las tareas, y el proceso de cálculo, el cual permite la optimización de procesos tanto en áreas industriales, educación entre otros, por lo que en el área de desarrollo de sistema se ha visto que la utilización de esta herramienta ha contribuido a los usuarios a optimizar procesos que normalmente se desarrollan de manera manual. En el área de educación esta herramienta puede ser descargado de manera libre, con una licencia por educación, el cual la empresa de Microsoft ha puesto a disposición dicha herramienta, en donde el docente o estudiante podrá descargarlo de manera gratuita.

2.4. Objetivos

2.4.1. Objetivo general

Desarrollar una macro el cual permita generar de código en lenguaje PHP, a partir del uso de una hoja calculo y utilizando la arquitectura Modelo – Vista – Controlador.

2.4.2. Objetivos específicos

a) Elaborar un módulo para el diseño de formularios en una hoja de cálculo y esta pueda ser almacenado en una base de datos.
b) Desarrollar un generador de código en Visual Basic para Aplicaciones el cual permita generar código fuente a partir del formulario diseñado en la hoja de cálculo almacenada en la base de datos y que interprete a la arquitectura de Modelo – Vista - Controlador.
c) Implementar un módulo el cual permita diseñar reportes en formato HTML y PDF

2.5. Hipótesis

2.5.1. Hipótesis general

La aplicación desarrollada, acelera el proceso de desarrollo de aplicaciones basado en tecnologías Web mediante la arquitectura Modelo – Vista - Controlador.

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1. Lugar de estudio

Ámbito o lugar de estudio

Departamento: Puno

Provincia: Puno

Distrito: Puno

Entidad: Universidad Nacional del Altiplano

Figura 12. Vista aérea de la Universidad Nacional del Altiplano

3.2. Población

Según la Ley Universitaria N° 30220, se crea la Superintendencia Nacional de Educación Superior Universitaria, el cual es el encargado de verificar las Condiciones Básicas de Calidad en todas las Universidades del Perú, por lo que en el año 2015 publica "El Modelo de Licenciamiento y su Implementación en el Sistema Universitario Peruano", donde todas las universidades nacionales y privadas, están obligadas a la presentación de su documentación para su licenciamiento institucional, por lo que, estos deberían de cumplir las ocho condiciones básicas de calidad, en donde, en su "Condición I – Existencia de objetivos académicos, grados y títulos a otorgar y planes de estudio correspondientes", tenemos el componente "I.4 Sistemas de información", las universidades deben de presentar los indicadores correspondiente, para que así de esta manera la universidad tenga la autorización de funcionamiento, Por lo que, todas las universidades nacionales o públicas deben de contar con sistemas información para atender las necesidades de su población.

3.3. Muestra

Para nuestros propósitos de muestra, hemos seleccionado la Universidad Nacional del Altiplano. Esta elección se debe a que, en el distrito de Puno, Provincia de Puno y Departamento de Puno, la Universidad Nacional del Altiplano es la única institución universitaria ubicada en el mismo distrito que cuenta con la debida autorización de funcionamiento otorgada por la Superintendencia Nacional de Educación Superior Universitaria (Sunedu). Como resultado, la universidad ha proporcionado la información requerida de acuerdo con el modelo de licenciamiento establecido en 2015. Además, la institución cuenta con un equipo de desarrollo de sistemas web capacitado.

3.4. Método de investigación

Para la desarrollo e implementación de la investigación se realizado mediante un enfoque de tipo cuantitativo, dado que este tipo de investigación permite realizar una investigación con un método estructurado, mediante la utilización de datos, los cuales nos permitirán comprobar la hipótesis planteada, esto en base de la medición y análisis de la información obtenida, los cuales nos indicarán los comportamientos de los datos.

3.4.1. Métricas para el análisis la obtención de información

Para la obtención de los datos de medición se utilizarán métricas de líneas de código, tiempo, porcentaje de uso de CPU y GPU.

a) Métricas de líneas de código, para realizar el análisis de líneas de código de la macro desarrollada, se procederá a exportar los módulos implementados en un formato de tipo BAS el cual es un formato de tipo texto ASCII, en donde se aplicará otra macro, el cual nos permitirá recuperar todos estos módulos y este devolverá información de procedimientos generados, tipo de ámbito, así como también el número de líneas de código, excluyendo líneas de comentarios.

b) Métricas de Carga de Memoria, está métrica nos permitirá obtener información de cuanta memoria está utilizando la macro desarrollada, durante el proceso de compilación, para lo cual se hará uso de la función API GlobalMemoryStatusEx, el cual será integrado durante el proceso de compilación por la macro, por lo que esta función nos entregará información de carga de memoria en un punto indicado, para el cálculo de la carga total de memoria que es utilizada por la macros, se procederá a obtener información al inicio y fin de cada procedimiento de la compilación.

c) Métricas de Tiempo, para la obtención de las métricas de tiempo en cada proceso durante la compilación del formulario se ha de implementar adicionalmente un procedimiento, el cual permita la obtención del tiempo que le toma al generador de código en realizar dicha tarea, para lo cual se debe de utilizar la llamada a la API de Windows GetTikCount, dicha función nos devolverá información en milisegundos, en donde para obtener el tiempo de procesamiento se realizará la captura inicial y la captura final del tiempo, realizando la diferencia se nos entregará el tiempo que le tomo la compilación

d) Métricas de uso de CPU, como se entiendo el manejo de CPU va estar sujeto a la cantidad de programas ejecutándose, dado que Windows ejecuta diferentes programas en memoria, que hacen uso de CPU, por lo que para poder obtener información del uso de CPU que hace uso nuestra macro desarrollado en VBA, se utilizará la función API de Windows GetSystemTimes, el cual no entregará el porcentaje de uso del CPU, en donde para calcular la cantidad de CPU utilizada durante el proceso de compilación

se obtendrá dos capturas, siendo estas al inicio y al final de la compilación de un formulario

e) Métricas de uso de GPU, como es bien sabido que los ordenadores hacen uso de GPU para poder mostrar información de lo que se está procesando y de dar a conocer al usuario, por tanto, es importante verificar si la utilización de estos recursos no serán limitantes, para lo cual se utilizarán la función GetSystemTimes, el cual nos devolverá el porcentaje de utilización de los recursos de GPU, para lo cual se realizarán capturas al inicio de cada proceso de compilación y al final de compilación.

3.5. Descripción detallada de métodos por objetivos específicos

3.5.1. Software y recursos

Para el desarrollo e implementación, se ha utilizado lo siguiente:

a) Software propietario
   - Microsoft Windows 10
   - Microsoft Office
     - Microsoft Word
     - Microsoft Excel
     - Microsoft PowerPoint

b) Software libre
   - Google Chrome (Versión 111.0.5563.111)
   - Xamp (v8.2.0), el contiene Apache, PHP y MySql
   - MysqlFront (v5.3)
   - HeidiSQL (v11.3.0.6295 x64)
   - SublimeText (v4.0)
   - TcpPDF
   - CodeIgniter (1.0.0)

c) El hardware utilizado durante las compilaciones es
   - Laptop Core I7 7th Generación
   - Procesador: Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz
   - RAM instalada: 12.0 GB (11.9 GB usable

- Tipo sistema: Sistema operativo de 64 bits, procesador basado en x64
- Pantalla Intel(R) HD Graphics 620 (incorporado)
  - Memoria Total Aprox.: 6209MB
  - Memoria de Pantalla (VRAM) 128MB
  - Resolución 1366 x 768 (60Hz)
- Video externo
  - NVIDIA GeForece 940MX
  - Tipo de Chip: GeForce 940MX
  - Tipo de DAC: Integrated RAMDAC
  - Memoria Total aprox: 10147MB
  - Memoria de pantalla (VRAM) 4065MB
  - Memoria compartida: 6081MB

3.5.2. Procedimientos de desarrollo

a) Diagramas de Casos de uso

Para el diseño de la macro, el cual permita desarrollar aplicaciones Web, se plantea el siguiente diagrama UML de Casos de Uso.

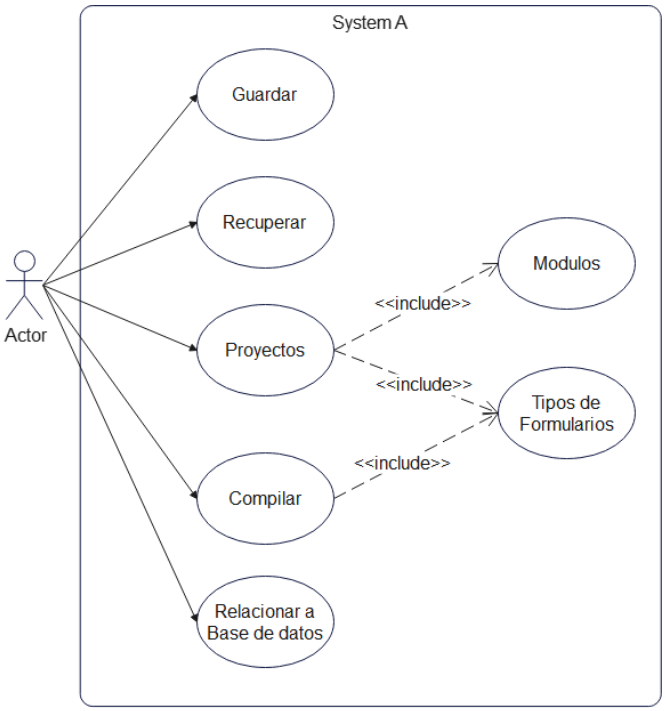


Figura 13. Diagramas de Casos de usos del Proceso de Diseño

b) Diagrama de Secuencia

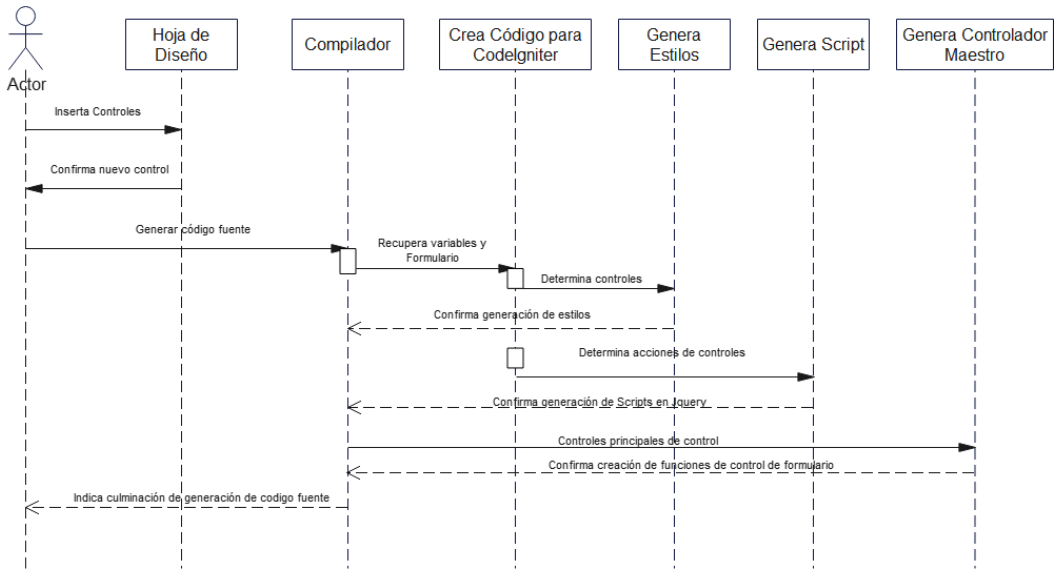


Figura 14. Diagrama de secuencia del proceso de diseño y compilación de formularios

c) Proceso de Compilación

La compilación se realiza invocando al Procedimiento de compilación en la barra de opciones del, seguido a esto la macro realizará las tareas de generación de código fuente en formato HTML y PHP compatible con el Framework de Codeigniter, para lo cual se iniciará con la verificación de que si el diseño actual es del tipo de formulario, caso contrario este recuperará el formulario, posterior a esto se iniciara con la creación de código fuente en los directorios previamente configurado correspondiente a Codeigniter generándose el código necesario en cada carpeta siendo estas las de Controller, View y Model respectivamente a su vez se genera el código CSS y JS relacionados a cada componente, en la carpeta raíz, completando los el código con la generación con el controlador maestro y las transacciones necesarias a continuación se muestra el procedimiento:

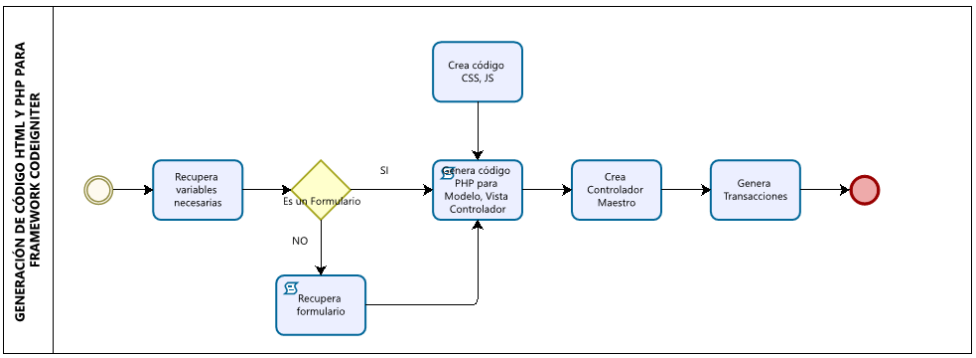


Figura 15. Secuencia del Proceso de Compilación de un Formulario

3.5.3. Módulo por objetivos

Objetivo 1: Elaborar un módulo para el diseño de formularios en Microsoft Excel mediante Visual Basic para Aplicaciones el cual pueda ser almacenado en una base de datos de MySql

Estructura de proyectos

Los proyectos a elaborar serán establecidos mediante proyectos, módulos, grupos, formularios, vistas, tablas; para la cual se tendrá la siguiente estructura

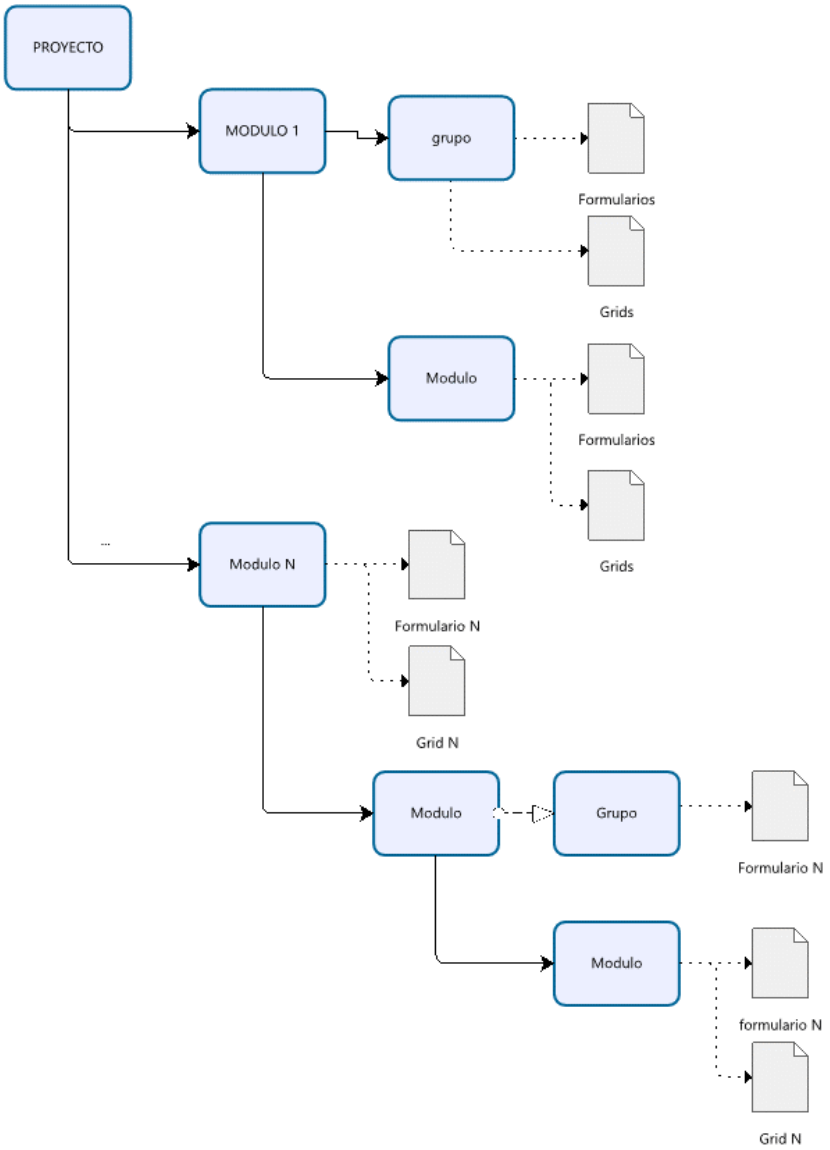


Figura 16. Estructura de un Proyecto para elaboración Web

Nota. Esta figura representa la forma de administración de un proyecto para la Web

- **Proyecto**: El proyecto viene a ser la descripción de la aplicación Web que se va a desarrollar, en el cual se va indicar:
  - Código del proyecto
  - Nombre del proyecto
  - Base de datos
  - Ruta de compilación
  - Correo electrónico
- **Módulo**: Tienen las siguientes características de cada módulo o submódulo
  - Se indica el Módulo padre
  - Denominación del módulo
  - Código de transacción
- **Grupo**: se inicia con un grupo General para todos los módulos creados
  - Denominación del grupo
- **Formularios, Vistas, Tablas**: Todos tendrán las siguientes características de acuerdo al tipo de formulario que se hará uso, siendo las características generales los siguientes:
  - Código de transacción
  - Descripción del formulario
  - Ancho en columnas
  - Alto en filas
  - Tipo
  - Parámetros de entrada
  - Consulta SQL
  - Campos
  - Acciones después de insertar
  - Acciones después de actualizar
  - Acciones después de eliminar
  - Máximo de registros admitidos para registro
  - Formulario de origen (en caso de tablas)
  - Tabla de origen en la base de datos
  - Tipo de renderizado
  - Si se muestra en la barra de menús

- **Controles**: en forma general cada control contendrá las características que corresponde a cada control, a continuación, se muestra los datos que serán almacenados en la base de datos
  - Celda
  - Nombre de la celda
  - Posición X en la hoja
  - Posición Y en la hoja
  - Ancho del control
  - Alto del control
  - Tipo de control
  - Tipo de recuperación de información (LOAD_TYPE)
  - Consulta SQL para listado (LOAD_LIST)
  - Descripción de la columna y anchos (LOAD_COLUMNS)
  - Columnas para la búsqueda (LOAD_COLUMN_FIND)
  - Título del formulario modal a mostrar (LOAD_FORM_TITLE)
  - Alto del formulario modal a mostrar (LOAD_FORM_HEIGHT)
  - Valores de retorno del formulario a otros controles (LOAD_FORM_VALUES)
  - Valor requerido
  - Texto o valor por defecto en la tabla
  - Alineación Horizontal
  - Alineación Vertical
  - Tabla relacionada al formulario (MySQL table)
  - Campo asignado al control en la tabla (MySQL field_name)
  - Tipo de dato del control en la tabla (MySQL field_type)
  - Ancho de daos del control en la tabla (MySQL field_width)
  - El control es un campo primario (MySQL field_PK)
  - El control es visible
  - El control esta activo (Enabled)
  - Texto del control
  - Tipo de fuente de la celda (Font Name)
  - Tamaño de la fuente (Font Size)
  - Fuente en negrita (Font Bold)

- Fuente inclinada (Font Italic)
- Fuente subrayado (Font Underline)
- Color de fondo de la celda
- Color de la fuente de la celda

Hoja de diseño de formularios

Para el diseño de los formularios se, agrego una hoja Excel en el cual contendrá información de la forma y controles que se necesitan para elaborar una Web, haciendo uso de la arquitectura MVC con Codeigniter, a continuación, se muestra la estructura de la hoja de diseño

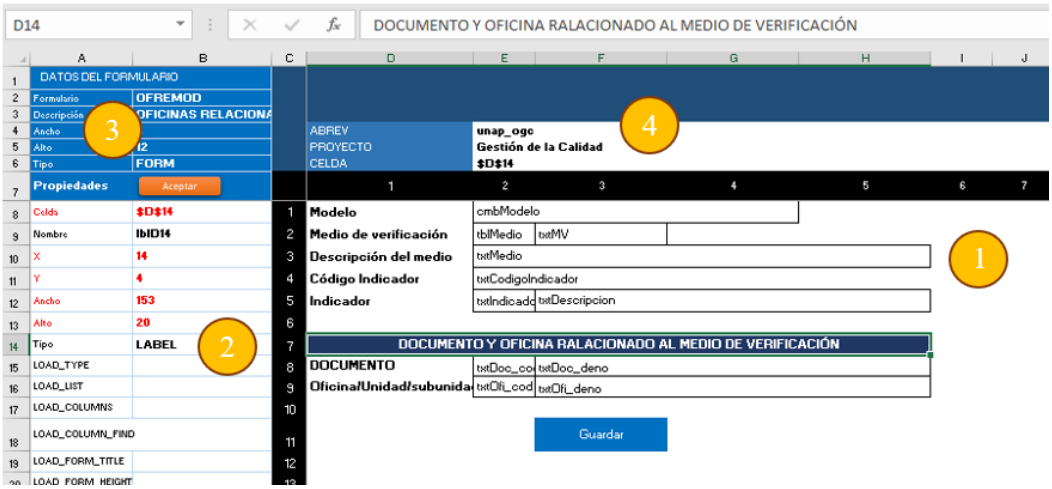


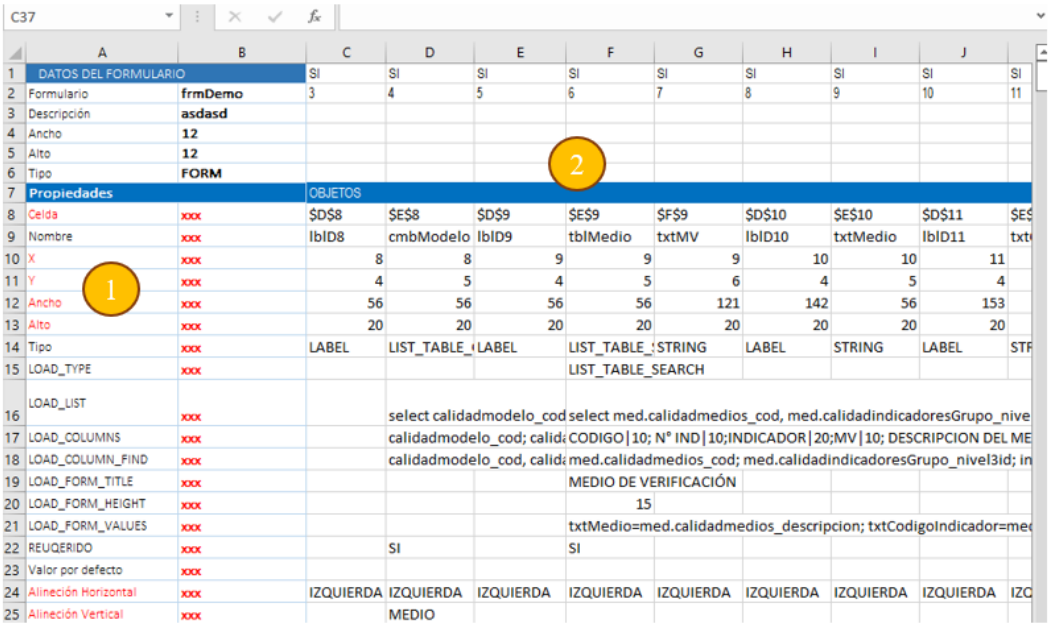
Figura 17. Hoja de diseño de Formularios

Nota: (1) Área de diseño: en se indicarán los textos a contener, así como también los controles que serán incrustados, así como la forma y estilo que llevara cada celda al momento de ser compilado. (2) Área de propiedades de la celda, aquí se indicará las características de la celda, los cuales serán tomados por el generador de código y determinar las acciones que realizarán al momento de ser compilado para Web, haciendo uso de la arquitectura MVC y la estructura dada por Codeigniter. (3) Área donde se indica la descripción del formulario, así como el ancho, alto y el tipo de formulario el cual se está diseñando. (4) Área donde se indica el proyecto y posición de la celda de trabajo

Hoja de registro de objetos de la hoja de diseño

Se ha de crear una hoja Excel adicional, esto para el almacenamiento temporal de las propiedades de los controles que se están incrustando en la hoja de diseño, esta

hoja esta fija en las filas las propiedades de cada control incrustado, y en las columnas se indicará el control que se introdujo en la hoja de diseños.



	A	B	C	D	E	F	G	H	I	J
1	DATOS DEL FORMULARIO		SI	SI	SI	SI	SI	SI	SI	SI
2	Formulario	frmDemo	3	4	5	6	7	8	9	10
3	Descripción	asdasd								
4	Ancho	12								
5	Alto	12								
6	Tipo	FORM								
7	Propiedades		OBJETOS							
8	Celda	xxx	D8	E8	D9	E9	F9	D10	E10	D11
9	Nombre	xxx	lblD8	cmbModelo	lblD9	tblMedio	txtMV	lblD10	txtMedio	lblD11
10	X	xxx	8	8	9	9	9	10	10	11
11	Y	xxx	4	5	4	5	6	4	5	4
12	Ancho	xxx	56	56	56	56	121	142	56	153
13	Alto	xxx	20	20	20	20	20	20	20	20
14	Tipo	xxx	LABEL	LIST_TABLE_	LABEL	LIST_TABLE_	STRING	LABEL	STRING	LABEL
15	LOAD_TYPE	xxx				LIST_TABLE_SEARCH				
16	LOAD_LIST	xxx		select calidadmodelo_cod		select med.calidadmedios_cod, med.calidadindicadoresGrupo_nive				
17	LOAD_COLUMNS	xxx		calidadmodelo_cod; calid		CODIGO|10; N° IND|10;INDICADOR|20;MV|10; DESCRIPCION DEL ME				
18	LOAD_COLUMN_FIND	xxx		calidadmodelo_cod, calid		med.calidadmedios_cod; med.calidadindicadoresGrupo_nivel3id; in				
19	LOAD_FORM_TITLE	xxx				MEDIO DE VERIFICACIÓN				
20	LOAD_FORM_HEIGHT	xxx				15				
21	LOAD_FORM_VALUES	xxx				txtMedio=med.calidadmedios_descripcion; txtCodigoIndicador=me				
22	REUQERIDO	xxx		SI		SI				
23	Valor por defecto	xxx								
24	Alineción Horizontal	xxx	IZQUIERDA	IZQUIERDA	IZQUIERDA	IZQUIERDA	IZQUIERDA	IZQUIERDA	IZQUIERDA	IZQUIERDA
25	Alineción Vertical	xxx		MEDIO						

Figura 18. Hoja de registro de controles

Nota. (1) Propiedades de los controles insertados en la hoja de diseño. (2) Controles insertados en la hoja de diseño.

Hoja de configuración

En esta hoja se determina las propiedades de los controles que se están insertando en la hoja de diseño, por lo que se tendrá dos tipos de propiedades, siendo estas los siguientes:

a) Controles de tipo de diseño

- **TIPO VISTA (VIEW)**

  Como se indica estos controles tienen la característica de mostrar información en pantalla, y no acepta ingreso de datos por teclado, siendo estos los siguientes:

  BTN_CANCEL: Indica que el control insertado en la hoja de diseño, que será un botón del tipo cancelar.

  BTN_LINK: Indica que este control redireccionará a un link que se haya determinado en sus propiedades

BTN_SAVE: Se indicará que este botón realizará una acción de tipo "SUBMIT" en el formulario, lo cual el sistema reconocerá que se esta enviando información del formulario al control maestro y así esta pueda registrar toda la data contenida en los controles de entrada, y poder registrarlo en una tabla de la base de datos.

LABEL: Mostrará en pantalla un texto simple, según el formato indicado en la hoja de diseño, no tiene alguna acción

GRIDBST: Este control permitirá mostrar uno o más formularios del tipo GRID en una vista.

IMAGEN: Indicará que imagen se debe de mostrar en la vista que se esta diseñando.

- **TIPOS DE ENTRADA (INPUT)**

DATE: como se indica, este indicará que la celda permitirá el ingreso de fechas al momento de realizarse la compilación del formulario, por lo que este admitirá el formato día, mes y año (dd/mm/yyyy).

DNI: Indicará al sistema que la entrada que se está realizando es un tipo de dato de Documento Nacional de Identidad, el cual permitirá el ingreso de datos de ocho (8) dígitos numéricos.

EMAIL: Indicará que la celda permitirá el ingreso en formato de correo electrónico

HORA: Indica que el control se compilará teniendo en cuenta el formato de hora

LABEL_TEXT: Este control permitirá mostrar contenido de texto no editable, pero a su vez el sistema podrá enviar mediante el submit el valor de este para poder ser almacenado en la tabla de datos.

NUMBER_INT: Indica que el único valor permitido para el ingreso de datos será del tipo entero.

STRING: Indica que el control permitirá el ingresado de datos numéricos, alfanuméricos o caracteres especiales, los cuales serán enviamos por el método submit, para ser registrado en la tabla de datos

TELEFONO: Se indica que el tipo de datos que se podrá ingresar es de tipo de número de teléfono, teniendo la característica de aceptar nueve (9) dígitos que son los de un teléfono celular.

FILE: Esto permitirá al diseñador indicar que el control permitirá la subida de archivos al servidor, por lo que se requerirá espacio disponible en el servidor e indicar la carpeta en la cual se deberá de subir los archivos, así como también las características de la subida de datos.

- **TIPO DE LISTA (LIST)**
  LISTA SIMPLE DEL TIPO SELECT

  LIST_SINGLE_COMBO: Esta lista de tipo simple tiene la característica de mostrar como un elemento de tipo <SELECT> de la nomenclatura HTML, en este control se tiene la siguiente sintaxis:

  LOAD_LIST = Código1|Descripción1; Código2|Descripción2; …..; CódigoN|DescripciónN

  LIST_TABLE_COMBO: Esta lista nos permitirá mostrar datos de una consulta SQL en un elemento HTML del tipo <SELECT>, para el uso de este control se indicar según la siguiente sintaxis:

  LOAD_LIST = SELECT Id, Descripcion FROM tabla
  LOAD_LIST = SELECT Tabla1.Id, tabla2.Descripcion FROM tabla1, tabla2 WHERE tabla1.id = tabla2.id
  La consulta SQL se debe de indicar en una sola línea continua.

  LISTA DE BÚSQUEDA MODAL

  UBIGEO: Este control módulo nos permitirá realizar búsquedas de código de UBIGEO en donde se nos mostrará un formulario modal de

búsqueda utilizando la arquitectura Bootstrap. Para el funcionamiento de esta propiedad de considerar en su base de datos la tabla UBIGEO, el cual debe de tener los siguientes campos

- ubigeo_cod
- ubigeodep_cod
- ubigeo_proc_cod
- ubigeo_distcod
- ubigeo_dep
- ubigep_prov
- ubigeo_dist

LIST_TABLE_SEARCH: Este control permitirá realizar búsquedas según una consulta SQL establecido en las propiedades en donde nos mostrará un formulario modal según la arquitectura de Bootstrap, para el funcionamiento de este control se debe especificar las propiedades con las siguientes sintaxis:

Modo 1

LOAD_LIST=	Select campo1, campo2, … campoN FROM tabla
LOAD_COLUMNS=	Campo1Deno|ancho1; Campo2Deno|ancho2; … ; CampoNDeno|AnchoN
LOAD_COLUMN_FIND	campo1; campo2; …. campoN
LOAD_FORM_TITLE	Título del formulario modal a mostrar
LOAD_FORM_HEIGHT	Altura del formulario en filas
LOAD_FORM_VALUES	Control1=Campo1; Control2=Campo2; …; ControlN=CampoN

Modo 2

LOAD_LIST=	Select Tabla1.campo1, Tabla2.campo2, … TablaN.campoN FROM tabla1, tabla2 WHERE tabl1.id = tabla2.id
LOAD_COLUMNS=	Campo1Deno|ancho1; Campo2Deno|ancho2; … ; CampoNDeno|AnchoN
LOAD_COLUMN_FIND	Tabla1.campo1; Tabla2.campo2; …. TablaN.campoN
LOAD_FORM_TITLE	Título del formulario modal a mostrar
LOAD_FORM_HEIGHT	Altura del formulario en filas
LOAD_FORM_VALUES	Control1=Campo1; Control2=Campo2; …; ControlN=CampoN

BASE DE DATOS

Se define la siguiente base de datos para el almacenamiento de la información de los proyectos, módulos, formularios y controles que serán utilizados por el sistema, para realizar la compilación del diseño en Excel a HTML y PHP utilizando la arquitectura del Framework de Codeigniter

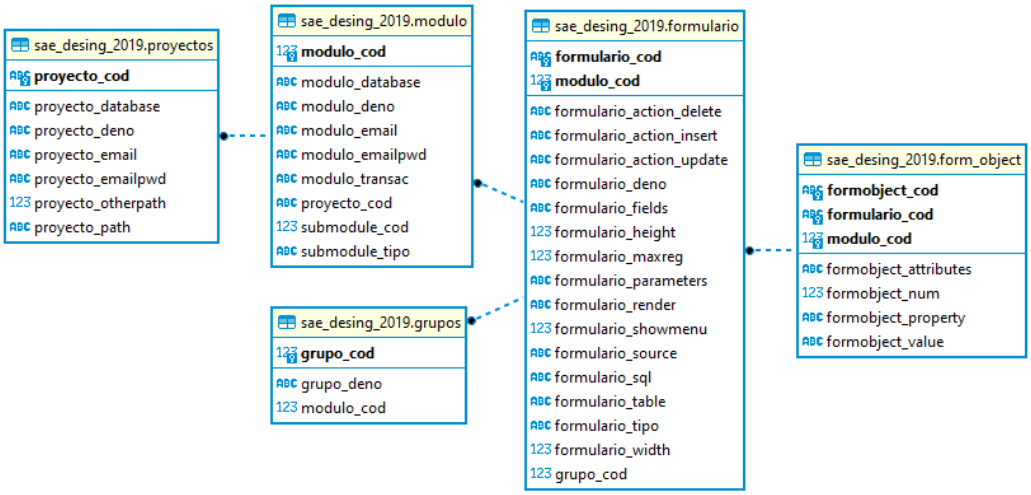


Figura 19. Diagrama ER de la Base de Datos

- **Registro de controles en la base de datos**

  Los controles insertados en la hoja de diseño se almacenarán de la siguiente forma en la tabla FORM_OBJECT

  formulario_cod: en este campo se indicará el código del Formulario

  formobject_cod: Se indicará el nombre del control, en el caso de hacer referencia a las propiedades de la columna o fila se indicará como @COLS y @ROWS

  formobject_value: Se indicará la fila y columna indicada en Excel haciendo uso del símbolo dolar ($), así como indicar en letras la columna y un valor numérico a la fila. En el caso de que en el campo formobject_cod sea @COLS o @ROWS, se debe de indicar COLUMNAS y FILAS respectivamente

  formobject_property: este campo estará dedicado para almacenar información del estilo y formatos que se han definido en la hoja de diseño, los cuales serán identificados por CELL, FONT o valores numéricos para el caso de COLUMNAS y FILAS, a continuación, se indica los valores que serán almacenados por cada registro:

  COLUMNAS = anchos de columnas definido en Excel separados un punto y coma (;)

  FILAS = altura de la fila definido en Excel separados un punto y coma (;)

  CELDAS = Se recoge la información de todas las propiedades de la celda almacenándolo según los prefijos que se indican

  CELL_MERGE**:** Valor de método MergeArea.Adress de Excel

  CELL_VALUE**:** Valor de la celda

  CELL_FORMAT**:** Valor de método NumberFormat de Excel

  CELL_VERTICAL**:** Valor de método VerticalAlignment de Excel

  CELL_HORIZONTAL**:** Valor de método HorizontalAlignment de Excel

CELL_BACKCOLOR\: Valor de método Interior.color de Excel

CELL_BORDER_TOP\: Valor de método Borders(xlEdgeTop).LineStyle de Excel

CELL_BORDER_RIGHT\: Valor de método Borders(xlEdgeRight).LineStyle de Excel

CELL_BORDER_BOTTOM\: Valor de método Borders(xlEdgeBottom).LineStyle de Excel

CELL_BORDER_LEFT\: Valor de método Borders(xlEdgeLeft).LineStyle de Excel

CELL_BORDER_COLOR\: Valor de método .Borders(xlEdgeLeft).color de Excel

FONT_NAME\: Valor de método Font.Name de Excel

FONT_SIZE\: Valor de método Font.Size de Excel

FONT_BOLD\: Valor de método Font.Bold de Excel

FONT_ITALIC\: Valor de método Font.Italic de Excel

FONT_UNDERLINE\: Valor de método Font.Underline de Excel

FONT_FORECOLOR\: Valor de método Font.color de Excel

formobject_attributes: Este campo administrará la información de las propiedades fijadas a cada control, para posteriormente el generador de código identifique la acción a realizar al momento de pasarlo a HTML y PHP según la arquitectura del Framework de CodeIgniter, almacenando la información en el orden de:

- Celda
- Nombre
- X
- Y
- Ancho

- Alto
- Tipo
- LOAD_TYPE
- LOAD_LIST
- LOAD_COLUMNS
- LOAD_COLUMN_FIND
- LOAD_FORM_TITLE
- LOAD_FORM_HEIGHT
- LOAD_FORM_VALUES
- REQUERIDO
- Valor por defecto
- Alineación Horizontal
- Alineación Vertical
- MySQL table
- MySQL field_name
- MySQL field_type
- MySQL field_width
- MySQL field_PK
- Visible
- Enabled
- Texto
- Font Name
- Font Size
- Font Bold
- Font Italic
- Font Underline
- Color de fondo
- Color de Fuente

Utilizando el separador de [@ATTR@], dado que toda la información se almacena en el mismo campo, en forma lineal.

Formobject_num: correlativo de los controles que se están registrando en la tabla.

El procedimiento de almacenamiento se va realizar ejecutando la acción que indica en los anexos.

Objetivo 2: Desarrollar un generador de código en Visual Basic para Aplicaciones, el cual permita generar código fuente a partir del formulario diseñado en Microsoft Excel que sea compatible con el Framework Codeigniter

Configuración del Framework de Codeigniter

Como ya se había indicado, el Framework de Codeigniter cuenta con dos directorios importantes, siendo "system" el CORE o núcleo para el funcionamiento de este Framework así como también el directorio de "application", en donde se determinaran las configuraciones así como también se ubicaran la administración del MVC. Para el correcto funcionamiento durante la ejecución posterior a la compilación de los diseños generados, se deberá de realizar la configuración de conexión a la base de datos, dicho archivo se ubicará en:

application\config\database.php

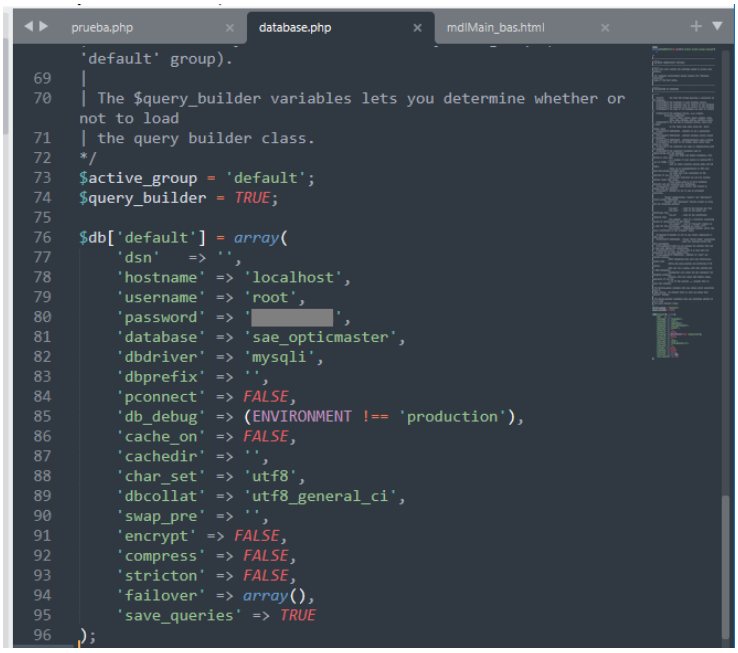


Figura 20. Configuración del Archivo database.php de Codeigniter

Nota. Se deberá de realizar la modificación en valores del hostname, username, password y database

Compilación de formulario

Para realizar la generación de código del formulario, primeramente, se debe tener guardado el proyecto, posterior a esto el generador de código seguirá los siguientes pasos:

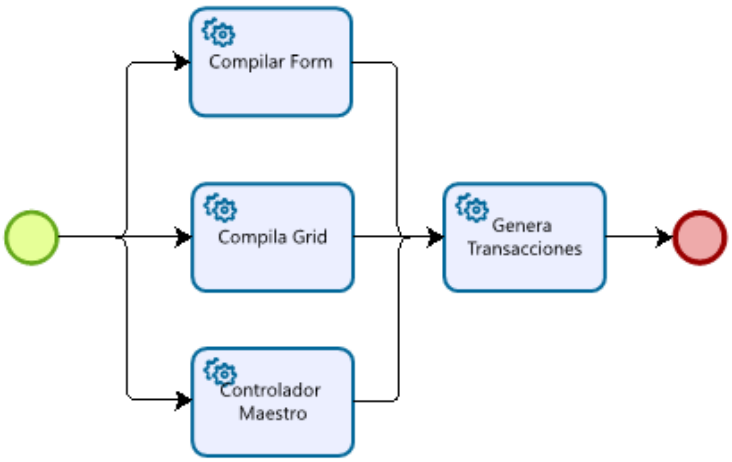


Figura 21. Secuencia de Compilación de un Formulario

El generador de código hará uso de lo especificado en la Hoja de diseño, y almacenado en la base de datos respectivamente, para lo cual en esta sección se hará uso de la arquitectura establecida en HTML, PHP, Framework de Codeigniter y a su vez se verá el modo de administración de los formularios establecidos en el sistema.

1. RECOPILACIÓN Y FIJADO DE PARÁMETROS DEL FORMULARIO

El generador de código recupera información acerca del formulario, el cual establece algunos criterios para el funcionamiento de este, siendo el procedimiento a ejecutar VAR_FORM del código fuente ubicado en las macros de Excel, en donde se recupera información acerca de:

- Formulario : Se especifica el código del formulario
- Descripción : Descripción del formulario
- Ancho : Ancho en columnas del formulario en la hoja de diseño
- Alto : Alto en filas del formulario en la hoja de diseño
- Tipo : Tipo de formulario (Form, View, Grid)
- Parameters : Parametros de entrada para el formulario

- SQL : Consulta SQL (Grids o Views)
- Fields : Campos generados por la consulta SQL (Grid, Views)
- Insert : Acciones posterior al guardado en un registro
- Update : Acciones posterior a la actualización de datos
- Delete : Acciones posterior a la eliminación de registros
- MAXREG : Maximo de registros permitidos para almacenamiento
- SOURCE : Formulario de origen (Grids)
- TABLA : Tabla de uso en la base de datos
- RENDER : Modo de renderizado (Bootrstrap, Table), por defecto TABLE
- SHOW MENU: Si el Form Gird, View se muestra en el menú lateral izquierdo

A su vez se establece los nombres del Controlador, Vista y Modelo, que hará uso el formulario, así como también el generador de código especificará las rutas de compilación según lo establecido en la configuración del proyecto, por lo que se establecerá los prefijos siguientes:

- **Ctrl_(Nombre del formulario).php**, el cual especifica el nombre del controlador que hará uso el formulario, según lo establecido en la arquitectura del Framework de Codeigniter, para lo cual se hará uso de código PHP para la programación de las acciones especificadas en la hoja de diseño
- **Model_(Nombre del formulario).php**, especifica el nombre del modelo que administra las conexiones, consultas que se realizarán a la base de datos, los cuales correspondan al Formulario y a su vez a las consultas que realizarán los Grid que hacen referencia como origen al formulario en cuestión, para cual se hará uso de código de programación de PHP y el uso del Lenguaje Estructura de Consultas (SQL), así como el uso de conexiones a la base de datos.
- **View_(Nombre del formulario).php**, Como se indica, este permitirá la visualización de la información del diseño del formulario, previamente establecido en la hoja de diseño de Excel. Para este caso se hará uso principalmente código HTML para la visualización de información, así como el uso de la arquitectura de Bootstrap, a su vez se hará uso de programación

en PHP, el cual permitirá una mejor administración de la visualización de información.

- **Css_(Nombre del formulario).css**, como se indica este archivo controlará los estilos que se hayan establecido en la hoja de diseño de Excel, así como otros estilos necesarios para la visualización de información de los formularios
- **Js_(Nombre del formulario).js**, este archivo está destinado al manejo de las scripts con jQuery, el cual administrará las acciones de los controles que se hayan insertado, así como también algunas acciones de control que el sistema requiere, así el uso de Ajax, manejo de archivos, el control de información entre otros.

```
(General)                                                    VAR_FORM

End Sub
Sub VAR_FORM()
    InitSheet
    If Trim(MODULE.codigo) = 0 Then VAR_MODULE

    FORMULARIO.codigo = hProyect.Range("CONFIG_FORM_NAME")
    FORMULARIO.CodeSource_Name = Get_CodeSourceName(FORMULARIO.codigo)

    FORMULARIO.Denominacion = hProyect.Range("CONFIG_FORM_DENO")
    FORMULARIO.Width = hProyect.Range("CONFIG_FORM_WIDTH")
    FORMULARIO.Height = hProyect.Range("CONFIG_FORM_HEIGHT")

    FORMULARIO.TypeForm = hProyect.Range("CONFIG_FORM_TYPE")
    FORMULARIO.Parameters = hProyect.Range("CONFIG_FORM_PARAMETERS")
    FORMULARIO.sql = hProyect.Range("CONFIG_FORM_SQL")
    FORMULARIO.Fields = hProyect.Range("CONFIG_FORM_FIELDS")
    FORMULARIO.Post_Insert = hProyect.Range("CONFIG_FORM_INSERT")
    FORMULARIO.Post_Update = hProyect.Range("CONFIG_FORM_UPDATE")
    FORMULARIO.Post_Delete = hProyect.Range("CONFIG_FORM_DELETE")
    FORMULARIO.MAXREG = hProyect.Range("CONFIG_FORM_MAXREG")
    FORMULARIO.Source = hProyect.Range("CONFIG_FORM_SOURCE")
    FORMULARIO.Table = hProyect.Range("CONFIG_FORM_TABLE")
    FORMULARIO.Form_Render = hProyect.Range("CONFIG_FORM_RENDER")


    FORMULARIO.Name_Controller = "Ctrl_" & FORMULARIO.CodeSource_Name
    FORMULARIO.Name_Model = "Model_" & FORMULARIO.CodeSource_Name
    FORMULARIO.Name_View = "View_" & FORMULARIO.CodeSource_Name
    FORMULARIO.Name_Css = "Css_" & FORMULARIO.CodeSource_Name
    FORMULARIO.Name_Script = "Js_" & FORMULARIO.CodeSource_Name

    'Indicando nombres de los ruta y archivos del formulario
    FORMULARIO.File_Controller = MODULE.Path_Controller & "\" & FORMULARIO.Name_Controller & ".php"
    FORMULARIO.File_Model = MODULE.Path_Model & "\" & FORMULARIO.Name_Model & ".php"
    FORMULARIO.File_View = MODULE.Path_View & "\" & FORMULARIO.Name_View & ".php"
    FORMULARIO.File_Css = MODULE.Path_Css & "\" & FORMULARIO.Name_Css & ".css"
    FORMULARIO.File_Script = MODULE.Path_Script & "\" & FORMULARIO.Name_Script & ".js"
    'Indicando Area de trabajo del formulario
    FORMULARIO.cIni = hMain.Range("PROPIEDADES").Column + 3
    FORMULARIO.cEnd = FORMULARIO.cIni + hProyect.Range("CONFIG_FORM_WIDTH") - 1

    FORMULARIO.fIni = hMain.Range("PROPIEDADES").Row + 1
    FORMULARIO.fEnd = FORMULARIO.fIni + hProyect.Range("CONFIG_FORM_HEIGHT") - 1

    FORMULARIO.Form_Cols = hProyect.Range("CONFIG_FORM_WIDTH")
    FORMULARIO.Form_Rows = hProyect.Range("CONFIG_FORM_HEIGHT")


End Sub
```

Figura 22. Secuencia de Recuperación de Información y Fijación de Parámetros del Formulario

2. CREACIÓN DE LA PLANTILLA DE VISUALIZACIÓN DEL FORMULARIO (VIEW)

La visualización del diseño de formulario se realiza en base a lo que se especifique en el diseño en la hoja Excel, por lo que será necesario recuperar el diseño del formulario en la hoja de diseño de Excel, posterior a esto se procede a realizar la compilación del diseño de la hoja Excel, mediante los procedimientos de compilación relacionados a la creación de la plantilla de visualización para lo cual se ejecuta el procedimiento VIEW_CREATE_FORM el cual realiza el procedimiento:

- Se recupera información sobre el formulario y rutas de almacenamiento
- El generador de código procederá de generar la plantilla de celdas en filas y columnas iniciando con la ejecución del procedimiento *BootStrapInit*, así como también reconocerá los controles ingresados en la hoja de diseño de Excel y los insertará en su ubicación respectiva identificándolo con el nombre del control, para tal efecto se ejecuta el procedimiento de *Createplantillabootstrap* en la cual el generador de código diseñará el formulario en código HTML el cual se almacenará en memoria con el nombre *PLANTILLA_CELLS*, una vez finalizado la creación de la plantilla se procede a ejecutar el procedimiento *BootStrapEnd* el cual finaliza el diseño de la plantilla.
- Ya completado el diseño de la plantilla el generador de código procede a identificar los controles y agregar los comandos necesarios en código HTML, para lo cual se ejecutará el procedimiento de *Command_Replace*, el cual tiene las siguientes instrucciones:
  - LABEL: Command_Label(mModeRender)
  - LABEL_TEXT: Command_Label_Text(mModeRender)
  - STRING: Command_String(mModeRender)
  - NUMBER_INT: Command_Number_Int(mModeRender)
  - EMAIL: Command_Email(mModeRender)
  - TELEFONO: Command_Telefono(mModeRender)
  - DNI: Command_DNI(mModeRender)
  - DATE: Command_Date(mModeRender)
  - IMAGE: Command_Image(mModeRender)
  - BTN_LINK: Command_Btn_Link(mModeRender)
  - BTN_SAVE: Command_Btn_Save(mModeRender)

- BTN_CANCEL: Command_Btn_Cancel(mModeRender)
- FILE: Command_File(mModeRender)
- UBIGEO: Command_Ubigeo(mModeRender)
- LIST_SINGLE_COMBO: Command_Single_Combo(obj, mModeRender)
- LIST_TABLE_COMBO: Command_List_Table_Combo(obj, mModeRender)
- LIST_TABLE_SEARCH: Command_List_Table_Search(obj, mModeRender)

Se recomienda ver el código fuente relacionado a cada comando en la sección de Anexos.

- Ya finalizado con la asignación de los acciones en cada control ingresado en la hoja de diseño, este procede a ser almacenado en el archivo *View_(Nombre del formulario).php;* en la ruta especificada a su vez el generador de código agregará código necesario según lo determinado en los parámetros.

3. CREACIÓN DEL CONTROLADOR DEL FORMULARIO (CONTROLLER)

El generador de código va generar un archivo compatible con la arquitectura del Framework de Codeigniter, para lo cual se creará un archivo PHP en la ruta que se especifica en la sección de recuperación de datos del formulario, para lo cual se implementará una Clase extendida de la Clase CI_Controller, en la cual se incluirá su constructor __construct(), al mismo tiempo se incluirá dentro del constructor el model relacionado al formulario que se está compilando, a su vez se incluirá el helper de tipo URL.

database.php | Ctrl_Ofremod.php | mdlMain_bas.html

```php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Ctrl_Ofremod extends CI_Controller {
    function __construct()
    {
        parent::__construct ();
        $this->load->model('Ogccfg/Model_Ofremod');   // se recupera el modelo para manejar los datos
        $this->load->helper('url');
    }   // ----- END FUNCTION -----
    public function index()
    {
        $this->load->view('web_header');
        $this->load->view('web_footer');
    }   // ----- END FUNCTION -----
```

Figura 23. Clase de Formulario Extendida de la clase CI_Controller de Codeigniter

Creado archivo controlador del formulario, se incluirán algunas funcionalidades para el funcionamiento del formulario que se ha creado en la hoja de diseño de Excel, siendo los principales las siguientes funciones

- Controller_Body_ShoAdd: en esta función, el generador de código incluirá información de la denominación del formulario, así como también se incluirán funciones de visualización de encabezado del formulario, VIEW del formulario y finalizando, con el pie de página del formulario.
- Controller_Body_Edit: Para eso de generador de código indicará como se debe de realizar la modificación de información del formulario
- Conroller_Body_ShowGrid: como se indica en esta sección, el generador de código recupera información de formularios de tipo grid, que están relacionados al formulario, para así crear una funciona individual por cada grid, indicándolo con su nombre del formulario de tipo grid.

Se puede observar el código fuente relacionado para observar cómo se crea la función relacionada a cada acción.

4. CREACIÓN DEL MODELO DE FORMULARIO (MODEL)

Para la conexión a la base de datos y el manejo de la información, la aplicación genera una clase extendida de la clase CI_Model del Framework de Codeigniter, el cual permitirá el vínculo entre la base de datos y el formulario, esta clase tendrá el nombre con el prefijo “Model_” y el nombre del formulario.

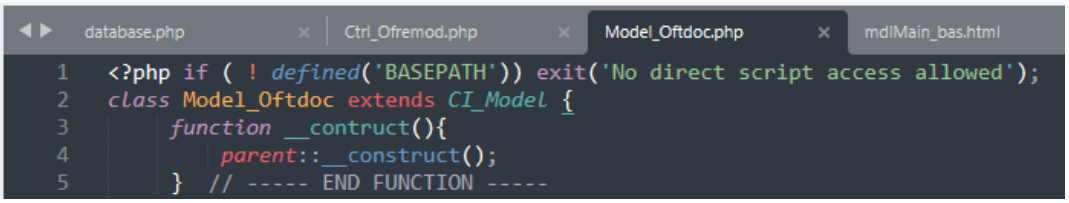


Figura 24. Clase Model del Formulario una Extensión de la clase CI_Model del Framework de Codeigniter

El generador de código generará las funciones siguientes

- **Model_Query**: Esta función está destinado a realizar consulta SQL, el cual devolverá una serie de registros resultado de la consulta efectuada
- **Model_GetInfoRow**: Devuelve información de una fila de la consulta realizada.

- **Model_Insert**: como se indica esta función tienen al objetivo de registrar en la tabla de la base de datos la información proporcionada en el formulario, en esta función, el generador de código indicará la tabla a cual se hará el registro de la información, como también se indica los campos a las cuales se ha indicado el ingreso de información mediante el formulario, vinculando cada información a los campos indicados en las propiedades de cada control indicado en la hoja de diseño de Excel, para posteriormente realizar el registro en la data.
- **Model_update:** esta función tiene como objetivo actualizar información de los registros mediante consulta Sql
- **Model_delete:** la función de este modelo es de eliminar de manera definitiva de la tabla de la base de datos especificada, para lo cual se realizará mediante una consulta sql.
- **Model_ListCombo**: El modelo List_combo es utilizado por el generador de código para dar información a los controles en el formulario, los cuales son del tipo LIST_SINGLE_COMBO y LIST_TABLE_COMBO

  $datos[Control]["LIST"] = Lista simple;

  $datos[Control]["TYPE"] = Tipo de control;

  $datos[Control]["CODIGO"] = Campo código ;

  $datos[Control]["DENO"] = Campo descripción a visualizar en lista

  $datos[Control]["LIST"]=Lista de registros realizada a la tabla especificada
- **Model_Table_Search**: Esta función está destinado a la utilización en controles que tienen la característica de mostrar información en un formulario modal de búsqueda, para lo cual en este se le da las propiedades de búsqueda en la hoja de diseño del control.

  $DATA[Control][Ítem Búsqueda] = Consulta SQL de búsqueda

  Donde "Ítem búsqueda" hace referencia a los criterios de búsqueda especificada en la hoja de diseño, en donde este debe contener los campos en la consulta sql que se está realizando.

  En esta función se incluirán información de todos los controles que tengan esta propiedad, el cual será convocado a esta función mediante consulta

del tipo AJAX, dado que desde el mismo formulario modal se podrá especificar el modo de búsqueda.

- **Model_ShowGrid**: Esta función está destinada a la creación de funciones por cada grid que se haya creado en el formulario de tipo grid, para lo cual las funciones se crearán con el Nombre del formulario seguido de postfijo de "_GetGridRow", el cual será invocado desde el controlador para ser mostrado en el formulario de tipo grid.

5. CREACIÓN DE LOS ESTILOS DE VISUALIZACIÓN DEL FORMULARIO

Como parte importante de la visualización de los formularios, el generador de código tiene en consideración el manejo de los CSS o más conocidos como los estilos en el formato de archivos HTML, así como también el uso de la arquitectura de Bootstrap, es que se implementa una sección de creación de estilos según lo determinado en la hoja de diseño de Excel, del cual se obtiene las características indicadas en las celdas de Excel, así como también se verifica el tipo de control insertado, para lo cual se ha generado una función denominado "GetStyleForm", el cual devuelve los estilos incluidos en la celda, para ser guardado en la ruta especificada, en donde el formulario hará referencia a este para la visualización del mismo.

```
.Ofremod_tblMedio_box0 {    border-style: none;     width:80px; height:25px;}
.Ofremod_tblMedio_box1 {    float:left; width:55px; height:25px;}
.Ofremod_tblMedio_box2 {    float:left; width:25px; height:25px;    border-radius: 0px 5px 5px 0px; padding: 0;
.Ofremod_Modal_tblMedio_col1{ width:80px; }
.Ofremod_Modal_tblMedio_col2{ width:80px; }
.Ofremod_Modal_tblMedio_col3{ width:160px; }
.Ofremod_Modal_tblMedio_col4{ width:80px; }
.Ofremod_Modal_tblMedio_col5{ width:240px; }
.Ofremod_Modal_tblMedio_col6{ width:40px; }
.Ofremod_Modal_tblMedio_col7{ width:40px; }
.Ofremod_Modal_tblMedio_search { width:720px; }
.Ofremod_Modal_tblMedio_table { width:720px; height:384px; margin:none; }
.Ofremod_Modal_tblMedio_table thead { width:720px; }
.Ofremod_Modal_tblMedio_table tbody { width:720px; height:360px; }
.Ofremod_Modal_tblMedio_Form {width:760px; height:434px; }
.Ofremod_tblMedio {
    padding-left: 5px;
    padding-right: 5px;
    width: 55px;
    height: 25px;
    font-family: Calibri;
    font-size: 13px;
    font-weight: normal;
    font-style: normal;
    text-overflow: ellipsis;
    white-space: nowrap;
    overflow:hidden;
    display:table-cell;
    vertical-align: middle;
    border: 1px solid;
    border-color: #000000;
    border-radius: 3px;
}
```

Figura 25. Compilación de Estilos de un Formulario

6. CREACIÓN DE LOS SCRIPTS DE JQUERY PARA LAS ACCIONES DEL FORMULARIO

Los scripts en el diseño de páginas Web son una herramienta importante, dado que nos permite realizar control de acciones que se ejecutan del lado de cliente, por lo que mejora el manejo de los formularios para la mejora administración, para esto el generador de código, genera algunos scripts dependiendo del tipo de control y propiedades del formulario, para lo cual se establecen los scripts siguientes:

- **Script_ButtonSave**: Este script está determinado para que el formulario pueda enviar información al controlador para que pueda ser almacenado en la tabla de la base de datos

```
$(document).ready(function() //Inicianlizando JQuery pcontrol de Formulario
{
    $("#btnSave").click(function()
    {
        var msg = "";
    //Verificando si existen datos requeridos
        if($('#cmbModelo_sel').hasClass('border-danger') == true ) $('#cmbModelo_sel').removeClass('border-danger
        if($('#tblMedio').hasClass('border-danger') == true ) $('#tblMedio').removeClass('border-danger');
        if($('#txtDoc_cod').hasClass('border-danger') == true ) $('#txtDoc_cod').removeClass('border-danger');
    //indicando valores faltantes
        if($('#cmbModelo').val() == '' || $('#cmbModelo').val() == '-1') { msg = 'Error'; $('#cmbModelo_sel').adc
        if($('#tblMedio').val() == '' ) { msg = 'Error'; $('#tblMedio').addClass('border-danger'); }
        if($('#txtDoc_cod').val() == '' ) { msg = 'Error'; $('#txtDoc_cod').addClass('border-danger'); }
        if(msg == 'Error')
        {
            $('#Ofremod_Modal_Message_title').html('Error en valores requeridos');
            $('#Ofremod_Modal_Message_msg').html('Debe de completar los valores requeridos');
            $('#Ofremod_Modal_Message').modal();
        }
        else
            $('#Ofremod').submit();
    });
```

Figura 26. Ejemplo de Compilación en JQuery de Control de tipo BTN_SAVE

- **Script_JQuery_List:** El script está destinado para el control de los Controles de tipo *LIST_SINGLE_COMBO*, en donde se verifica si este control en el momento de la visualización contiene o no información para mostrar, en el caso de que se encuentre información, el script controlará que se realice una seleccione este se asigne como valor seleccionado.

```
//OBJECT: lblE10
//Verifica si el objeto no contiene a nada se inicializa con el primer valor
if( $("#lblE10").val() == "" )  $("#lblE10").val( $("#lblE10_sel").val()  );
//Control de cambios del select option
$("#lblE10_sel").change(function()
{
    $("#lblE10").val( $('option:selected', this).val() );
});
```

Figura 27. Ejemplo de compilación de control del tipo LIST_SINGLE_COMBO

- **Script_JQuery_TABLE_SEARCH:** este script está destinado para los controles de tipo lista con visualización de formulario modal, para realizar

sus búsquedas de información, siendo estos lo de tipo *INPUT_LIST_MODAL*, según lo indicado en la hoja de configuración. A su vez en este script se genera acciones relacionadas a la asignación de valores a otros controles al momento de seleccionar una opción de la lista mostrada.


```
    });
    //**************************************************************************************
    //** lblCalidadTipo_deno MODAL
    //**************************************************************************************
    //Control de de visualización del objeto modal: lblCalidadTipo_deno
    $("#lblCalidadTipo_deno_btn").click(function()
    {
        $("#Oftdoc_Modal_lblCalidadTipo_deno_Select").prop("selectedIndex", 0);
        $("#Oftdoc_Modal_lblCalidadTipo_deno_txt").val("");
        Oftdoc_Query("Oftdoc_Modal_lblCalidadTipo_deno");
        $("#Oftdoc_Modal_lblCalidadTipo_deno").modal();
    });
    //Control de busqueda SELECT: lblCalidadTipo_deno
    $("#Oftdoc_Modal_lblCalidadTipo_deno_Select").change(function()
    {
        Oftdoc_Query("Oftdoc_Modal_lblCalidadTipo_deno");
    });
    //Control de busqueda TEXT: lblCalidadTipo_deno
    $("#Oftdoc_Modal_lblCalidadTipo_deno_txt").keydown(function()
    {
        Oftdoc_Query("Oftdoc_Modal_lblCalidadTipo_deno");
    });
    //Control de busqueda TEXT: lblCalidadTipo_deno
    $("#Oftdoc_Modal_lblCalidadTipo_deno_txt").change(function()
    {
        Oftdoc_Query("Oftdoc_Modal_lblCalidadTipo_deno");
    });
    //Control de objetos de tipo List Search: lblCalidadTipo_deno
    $("#Oftdoc_Modal_lblCalidadTipo_deno_btnSearch").click(function()
    {
        Oftdoc_Query("Oftdoc_Modal_lblCalidadTipo_deno");
    });
    //Control de doble clic en la tabla
    $("#Oftdoc_Modal_lblCalidadTipo_deno").on("click", "tbody tr", function(event) {
        var tbl = $("#Oftdoc_Modal_lblCalidadTipo_deno_table tr");
        if (tbl.hasClass("Sistema_Table_Selected") == true ) tbl.removeClass("Sistema_Table_Selected");
        var currentRow=$(this).closest("tr");
        currentRow.addClass("Sistema_Table_Selected");
        $("#Oftdoc_Modal_lblCalidadTipo_deno_row_col1").val(currentRow.find("td:eq(0)").text());
        $("#Oftdoc_Modal_lblCalidadTipo_deno_row_col2").val(currentRow.find("td:eq(1)").text());
    });
    //Control del boton aceptar en el formulario modal
    $("#Oftdoc_Modal_lblCalidadTipo_deno_btnAceptar").click(function()
        {
            $("#lblCalidadTipo_deno").val( $("#Oftdoc_Modal_lblCalidadTipo_deno_row_col1").val() );
            $("#lblDenoOficina").html( $("#Oftdoc_Modal_lblCalidadTipo_deno_row_col2").val() );
            $("#Oftdoc_Modal_lblCalidadTipo_deno").modal("hide");
    });
function Oftdoc_Query(objModal)
{
    var info = $("#" + objModal +"_txt").val();
    var tipo = $("#" + objModal +"_Select").prop("selectedIndex");
    var met = "e3808814fcaf7f8c40c71c4ef012f3ad";
    $.ajax
    ({
            url: "../../Ogccfg_master/Lista",
            type: "Post",
            dataType: "json",
            data: {query: objModal, type: tipo, value: info, method: met },
            beforeSend: function(){
                //Precarga
            },
            success: function(response)
            {
                var cad,i ;
                cad="<tbody>";
                for(i = 0; i< response.length; i++)
                {
                    var item = response[i];
                    c=1
                    cad = cad + "<tr>";
                    $.each(item, function(key, value)
                    {
                        cad = cad + "<td class='" + objModal + "_col" + c + "'>" + value + "</td>";
                        c++;
                    });
                    cad = cad + "</tr>";
                }
                cad=cad + "</tbody>"
                $("#" + objModal + "_table").children("tbody").replaceWith(cad);
            }
    });
}
```

Figura 28. Ejemplo de compilación en JQuery de un control del tipo TABLE_SEARCH

Cabe precisar que los Scripts se generarán para cada control, para esto se hace uso la arquitectura de JQuery

database.php | Ctrl_Ofremod.php | Model_Oftdoc.php | Css_Ofremod.css | Js_Ofremod.js | mdlMain_bas.htm

```
$(document).ready(function() //Inicianlizando JQuery pcontrol de Formulario
{
    $("#btnSave").click(function()
    {
        var msg = "";
    //Verificando si existen datos requeridos
        if($('#cmbModelo_sel').hasClass('border-danger') == true ) $('#cmbModelo_sel').removeClass('border-dange
        if($('#tblMedio').hasClass('border-danger') == true ) $('#tblMedio').removeClass('border-danger');
        if($('#txtDoc_cod').hasClass('border-danger') == true ) $('#txtDoc_cod').removeClass('border-danger');
    //indicando valores faltantes
        if($('#cmbModelo').val() == '' || $('#cmbModelo').val() == '-1') { msg = 'Error'; $('#cmbModelo_sel').ad
        if($('#tblMedio').val() == '' ) { msg = 'Error'; $('#tblMedio').addClass('border-danger'); }
        if($('#txtDoc_cod').val() == '' ) { msg = 'Error'; $('#txtDoc_cod').addClass('border-danger'); }
        if(msg == 'Error')
        {
            $('#Ofremod_Modal_Message_title').html('Error en valores requeridos');
            $('#Ofremod_Modal_Message_msg').html('Debe de completar los valores requeridos');
            $('#Ofremod_Modal_Message').modal();
        }
        else
            $('#Ofremod').submit();
    });
    //OBJECT: cmbModelo
    //Verifica si el objeto no contiene a nada se inicializa con el primer valor
    if( $("#cmbModelo").val() == "" )   $("#cmbModelo").val( $("#cmbModelo_sel").val()  );
    //Control de cambios del select option
    $("#cmbModelo_sel").change(function()
    {
        $("#cmbModelo").val( $('option:selected', this).val() );
    });
```

Figura 29. Script en JQuery Generado para el Formulario

7. CREACIÓN DE FORMULARIOS DE TIPO GRID RELACIONADO AL FORMULARIO

El generador de código identifica los formularios de tipo grid que se encuentran vinculados a un formulario, para así realizar, por lo que se ejecuta el procedimiento *Compile_GridView_Init* , el cual está encargado de generar el código relación a mostrar información en tablas, por lo que este procedimiento procede a recuperar información de parámetros de la consulta así como también los campos que se van a mostrar en cada columna del grid, para así generar código HTML y PHP, en la cual se hace uso de las etiquetas relacionadas a <TABLE>, así mismo en agrega código en PHP, para mostrar lo registros entregados por el model, el cual fue requerido por el controlador.

```php
<table class="sistema_table">
    <thead>
        <tr>
            <th width="24px"> </th>
            <th width="91.44px">C&Oacute;DIGO</th>
            <th width="246.88px">MODELO</th>
            <th width="49.12px">I/E</th>
            <th width="275.44px">DESCRIPCI&Oacute;N</th>
            <th width="48px">MV</th>
            <th width="243.44px">MEDIO DE VERIFICACION</th>
            <th width="244.56px">OFICINA</th>
            <th width="256px">DOCUMENTO</th>
            <th width="24px"><img src="<?=base_url();?>images/btn/btn_add.png" class="sistema_icons"></th>
        </tr>
    </thead>
    <tbody>
    <?php
        $iRow = 0;
        if($REGISTROS != false)
            foreach ($REGISTROS as $row)
            { ?>
                <tr>
                    <td width="24px">
                        <img src="<?=base_url();?>images/btn/btn_edit.png" class="sistema_icons">
                    </td>
                    <td width="91.44px">
                        <?php echo $row->calidadrelacionmodelo_cod; ?>
                    </td>
                    <td width="246.88px">
                        <?php echo $row->calidadmodelo_deno; ?>
                    </td>
                    <td width="49.12px">
                        <?php echo $row->calidadindicadoresgrupo_nivel3id; ?>
                    </td>
                    <td width="275.44px">
```

Figura 30. Código Compilado de Formulario de tipo grid para el Framework de Codeigniter

8. CREACIÓN DEL CONTROLADOR DE MÓDULOS DEL PROYECTO

En controlador de módulo del proyecto es una clase extendida de la clase CI_Controller del Framework de Codeigniter, cuya finalidad es la de administrar la ejecución de los formularios pertenecientes a dicho módulo, en donde el código de invocación a cada formulario es codificado mediante la encriptación de tipo MD5, así como también este controlador ejecutara procedimiento de registro de información, consulta de listas utilizadas por los controles de tipo búsqueda en tabla, así como también permitirá la subida de archivos. Esta clase contiene los siguientes procedimientos

```php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Ogccfg_master extends CI_Controller {
    function __construct()
    {
        parent::__construct ();
        $this->load->helper('url');
    }       // ----- END FUNCTION -----
```

Figura 31. Código de Controlador de Módulo de Formularios del Proyecto

Index: Para el caso en que se invoque solamente mediante el código del módulo, este solo mostrará información de encabezado y pie de página del formulario, más no alguna información.

```
8       public function index()
9       {
10          $this->load->view('web_header');
11          $this->load->view('web_footer');
12      }   // ----- END FUNCTION -----
```

Figura 32. Procedimiento index del Controlador de un Módulo del Proyecto

GetModule: El procedimiento es el encargado de indicar los códigos de los módulos que será ejecutado según la petición realizada en su parámetro. En donde el parámetro de entrada será una codificación en MD5, el cual será comparado con los códigos de los formularios disponibles del dicho módulo

```
13      public function GetModule($eject)
14      {
15          $modulo = '-1';
16          if($eject == md5('Ofremod')) $modulo = 'Model_Ofremod';
17          if($eject == md5('Oftdoc')) $modulo = 'Model_Oftdoc';
18          if($eject == md5('Ogctipo')) $modulo = 'Model_Ogctipo';
19          return $modulo;
20      }   // ----- END FUNCTION -----
```

Figura 33. Procedimiento de Obtención de Formulario a Ejecutar

Upload: Este procedimiento permitirá la subida de archivos a la carpeta uploads ubicado en el lado del servidor, en el caso de exista algún error en la subida de datos, este devolverá un error.

```php
function upload()
{
    //upload file
    $dir = 'uploads/';
    $dirpath = $dir . $diruser ;
    if(!file_exists($dirpath )) mkdir( $dirpath, 0777 );
    $config['upload_path'] = $dirpath . "/";
    $config['allowed_types'] = '*';
    $config['max_filename'] = '255';
    $config['encrypt_name'] = false;
    $config['max_size'] = 1024 * 10; //1 MB
    $config['overwrite'] = false;
    $config['max_filename_increment'] = 'none'; //1 MB
    if (isset($_FILES['file']['name'])) {
        if (0 < $_FILES['file']['error']) {
            echo 'Error durnte la subida del archivo: ' . $
                _FILES['file']['error'];
        } else {
            if (file_exists($dir . $_FILES['file']['name'])) {
                echo 'El nombre de archivo ya existe : ' . $
                    _FILES['file']['name'];
            } else {
                $this->load->library('upload', $config);
                if (!$this->upload->do_upload('file')) {
                    echo 'Error al subir: El archivo ya existe
                        en el servidor' ;//.
                    $this->upload->display_errors();
                } else {
                    echo 'OK:' . $_FILES['file']['name'];
                }
            }
        }
    } else {
        echo 'Error:Sin conexión';
    }
}  // ----- END FUNCTION -----
```

Figura 34. Código del Procedimiento de Subida de Archivos

Register: este procedimiento está encargado de realizar el registro de información llenada en los formularios pertenecientes a dicho módulo del proyecto, para el cual el formulario en la visualización se indicará el código mediante una encriptación de tipo MD5, y será enviado a este procedimiento mediante el método de POST.

```php
public function register()
{
    $eject = $this->input->post('method');
    $modulo = $this->GetModule($eject);
    $this->load->model('Ogccfg/' . $modulo);    // se recupera
    el modelo para manejar los datos
    $this->$modulo->Insert();
}   // ----- END FUNCTION -----
```

Figura 35. Procedimiento de Registro de Información de Formularios Pertenecientes al Módulo

Lista: Este procedimiento está destinado al manejo de consultas realizadas desde los formularios modales, pertenecientes a los controles del tipo *LIST_TABLE_SEARCH o UBIGEO*, para lo cual el generador de código genera este procedimiento con los parámetros de entrada de la consulta realizada, el tipo de consulta que está relacionado a los campos indicados en el formulario modal, y el código del formulario, para lo cual realizada la consulta SQL invocando al modelo

del Framework de Codeigniter relacionado al formulario, devolviendo como resultado los registros en formato de *json_encode*

```
60      public function Lista()
61      {
62          $modeQuery = $this->input->post('query');
63          $modetype = $this->input->post('type');
64          $method = $this->input->post('method');
65          $modulo = $this->GetModule($method);
66          $modeSearch =utf8_decode( strtoupper( $this->input->post('
                value') ));
67          if($modeQuery != '' && $modetype != '' && $method != '' && $
                modulo != '-1')
68          {
69              $modeSearch = str_replace("  ", " ", $modeSearch);
70              $modeSearch = str_replace("%", " ", $modeSearch);
71              $modeSearch = str_replace("'", " ", $modeSearch);
72              $modeSearch = str_replace("#", " ", $modeSearch);
73              $modeSearch = str_replace("-", " ", $modeSearch);
74              $modeSearch = str_replace(" ", "%", $modeSearch);
75              $this->load->model('Ogccfg/' . $modulo);   // se
                recupera el modelo para manejar los datos
76              $data = $this->$modulo->TableSearch_Sql();
77              $sql = $data[$modeQuery][$modetype];
78              $sql = str_replace("[@SEARCH]", $modeSearch, $sql);
79              $r = $this->$modulo->Query($sql);
80              echo json_encode($r);
81          }
82          else
83          {
84              redirect('/');
85          }
```

Figura 36. Procedimiento de Consulta a Listas en Tablas de Formularios

9. GENERACIÓN DE TRANSACCIONES DE EJECUCIÓN DE FORMULARIOS.

Las transacciones hacen referencia a la ejecución de los formularios mediante el código indicado en el momento de diseño, así como también determina la visualización del menú lateral izquierdo al momento de visualizar la página Web ya compilada, por lo que el generador de código verifica la existencia de la tabla de *sae_transaction* en la base de datos, caso no existiese este se crea ejecutando el procedimiento *GenerateTableTransaction*, siendo la estructura de la tabla de la siguiente forma:

Nombre	Tipo	N...	Por defecto	Extras	C..
Indices (1)					
Clave primaria (PK)	saetransaction_cod			unique	
Campos (11)					
saetransaction_cod	int(11)	No	<auto_increment>		
modulo_cod	int(11)	Sí	<NULL>		
transaction_cod	varchar(25)	Sí	-		
transaction_deno	varchar(255)	Sí	-		
transaction_tipo	varchar(15)	Sí	-		
submodulo_cod	int(11)	Sí	0		
transaction_modo	int(1)	Sí	0		
transaction_class	varchar(80)	Sí			
transaction_php	varchar(255)	Sí			
transaction_md5	varchar(100)	Sí			
transaction_function	varchar(100)	Sí			

Figura 37. Estructura de la tabla sae_transaction

Una vez creada la tabla de transacciones, el generador de código procede a genera la lista de las transacciones ejecutando el procedimiento de *GenerateList* en donde se registrará la información relacionada a los formularios que serán utilizados por el controlador maestro.

saetransactio...	m	transaction_cod	transaction_deno	trans...	s.	t..	transaction_c...	transaction_php	transaction_md5	transaction_function
414	28	OGCCFG	CONFIGURACIÓN	-	0	0			6cba4ea0ea9bd19	
415	-1	GRDOREMDOC	RELACIÓN DE MEDIOS A DOCU	GRID	28	1	Ctrl_Ofremod	application/controllers/Ogccfg/Ctrl_Ofremod.php	ea337b1d48d6fe9(	Show_Grdoremdoc
416	-1	GRDTDOCS	DOCUMENTOS DE OFICINAS	GRID	28	1	Ctrl_Oftdoc	application/controllers/Ogccfg/Ctrl_Oftdoc.php	ad67d46b9418f6af	Show_Grdtdocs
417	-1	grdTipoModelo	TIPOS DE MODELOS	GRID	28	1	Ctrl_Ogctipo	application/controllers/Ogccfg/Ctrl_Ogctipo.php	7d4c57ee3ed2971	Show_Grdtipomodelo
418	-1	OFREMOD	OFICINAS RELACIONADOS MOI	FORM	28	1	Ctrl_Ofremod	application/controllers/Ogccfg/Ctrl_Ofremod.php	b2bfe2fbbb845a9a	ShowAdd
419	-1	OFTDOC	DOCUMENTOS DE OFICINA	FORM	28	1	Ctrl_Oftdoc	application/controllers/Ogccfg/Ctrl_Oftdoc.php	e3808814fcaf7f8c4	ShowAdd

Figura 38. Registro de Transacciones Disponibles para el Proyecto

10. CONTROLADOR MAESTRO

El controlador maestro será el encargado de realizar las invocaciones a los formularios solicitados por el lado del cliente, para lo cual se ha configurado el Framework de Codeigniter para ejecute como controlador principal el archivo *master.php*, siendo este lo primero que se ejecutará al momento de abrir la página Web, por lo que se ha creado una clase *master* el cual es la extensión de la clase CI_Controller de Codeigniter, para lo cual se ha creado el modelo *Model_master* para el manejo y acceso a la base de datos, siendo uno de los principales la ejecución del procedimiento de *Load_transaction*, así como también el procedimietno de e*ject* el cual será el encargado de mostrar los formularios solicitados código de transacción indicada al momento de mostrarse la página Web

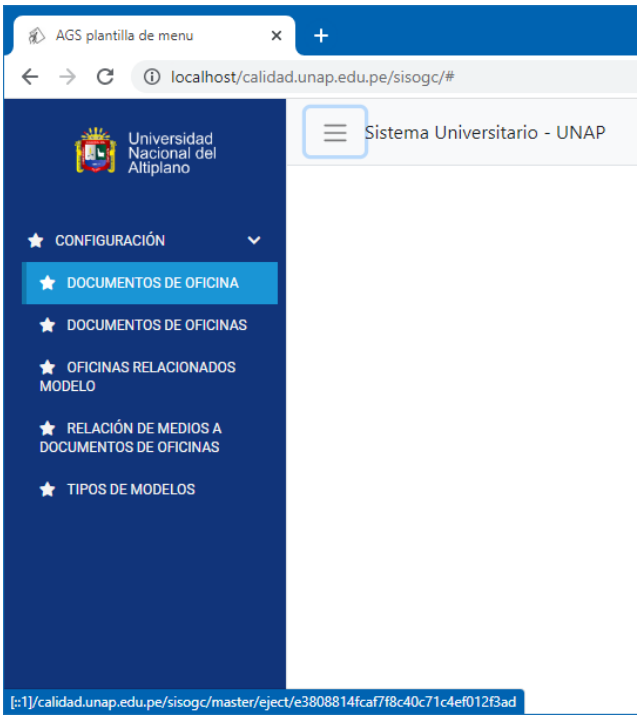


Figura 39. Visualización de la Página Web

Como se muestra en la imagen anterior, se observa que se muestra la lista de transacciones como un menú desplegable vertical, y al momento de seleccionar una opción se muestra en la parte inferior la dirección, invocando al controlador *master* solicitando que se ejecute el procedimiento *eject* con parámetro de entrada en código cifrado con MD5 el formulario que se desea visualizar.

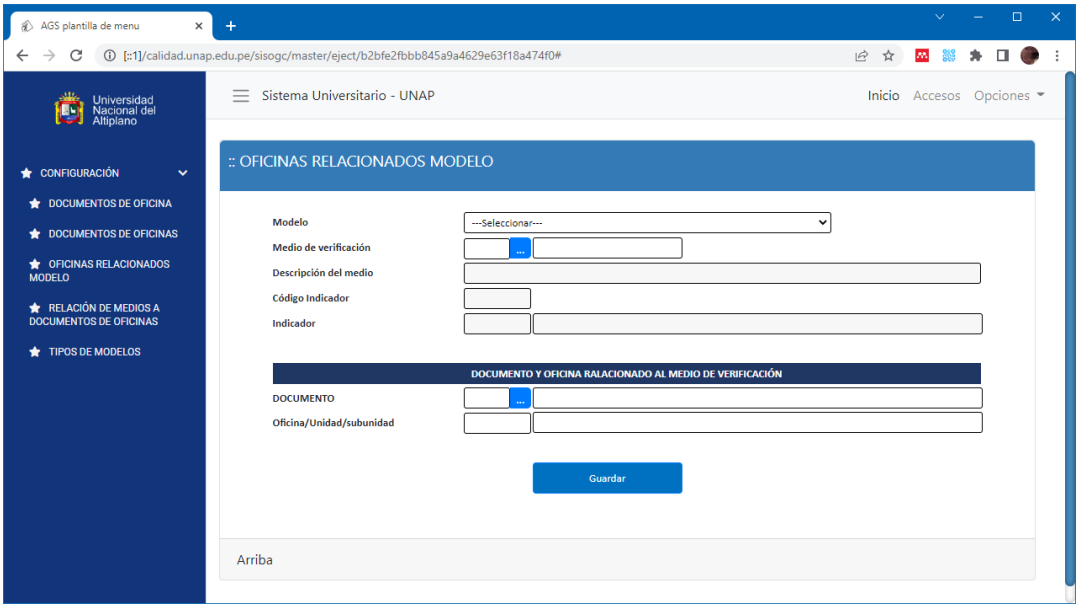


Figura 40. Imagen de Ejecución de Formulario Mediante el Procedimiento Eject del Controlador Master

Objetivo 3: Implementar un módulo en Excel el cual permita diseñar reportes en formato PDF y en pantalla el cual pueda ser usado por la aplicación

Para que el generador de código pueda generar archivos en formato PDF será necesario indicar al Framework de Codeigniter la utilización de la librería de TCPDF, en la carpeta de *application\libraries* a su vez se debe de generar el archivo Pdf.php, el cual será la extensión de la clase TCPDF, en donde se podrá realizar la configuración de del encabezado del formato PDF en el procedimiento de Header, posterior a esto se debe de generar código PHP, en donde se recupere la librería PDF, donde se indique el tamaño de papel encabezados, pie de página. Seguido a este se debe agregar una página nueva, posterior a esto se debe de crear una variable invocando a la visualización de un formulario, pero sin visualización del formulario, para posteriormente ejecutar el procedimiento de *writeHTML* de la librería de TCPDF, finalizando con la escritura del archivo PDF ejecutando el procedimiento de Output(nombre_archivo, 'I')

Objetivo General: Desarrollar una macro el cual permita generar de código en lenguaje PHP, a partir del uso de una hoja calculo y utilizando la arquitectura Modelo – Vista – Controlador.

Según los objetivos específicos se ha diseñados en cada módulo, el cual nos permite realizar un diseño de formularios en una hoja de cálculo de Microsoft Excel, esto ha permitiendo la inclusión de controles, haciendo uso de cada celda de la misma hoja Excel y asignándole propiedades las mismas que contienen información necesaria, como son el uso de los estilos, conexión a una tabla en la base de datos así como a sus campos indicando el tipo de campo, ancho entre otros, esto para que el generador de código pueda utilizar al momento de la elaboración del código fuente y genere código de los Modelos (conexión a base de datos), vistas, y Controlador del formulario y el controlador maestro, lo cual permitió el diseño más ágil de un formulario, finalizado el diseño y la asignación de las propiedades de los mismos se procedió con la elaboración del generador de código el cual permitió pasar del diseño al obtener un código fuente en formato PHP y compatible con el Framework de Codeigniter utilizando las arquitectura Modelo, Vista y Controlador (MVC), para lo cual se crearon módulos y procedimiento que permitan realizar dicha tarea de conversión de diseño a código fuente, esto incluía la generación de

código de estilos (CSS), código de acción mediante JQuery y código PHP para los controles diseñados en el presente trabajo.

Para la verificación, si el diseño de esta aplicación mediante la utilización de macros con Visual Basic para Aplicaciones (VBA), se plantea realizar la prueba estadística de t para una muestra, el cual nos permitirá evaluar la opinión de expertos en desarrollo de aplicaciones web, en relación del tiempo promedio que les demora en desarrollar aplicaciones, y esto será comparado con el tiempo promedio en que se desarrolla aplicación con nuestra macro, así como también se utilizaran métricas para verificar el tiempo que le lleva al generador de código en generar el código fuente así como también la utilización de los recursos del computador

3.5.4. Métricas de validación de software

3.5.4.1. Métricas de Líneas de Código

Se realizaron el análisis de métricas de líneas de código de los módulos desarrollados para dar solución a los objetivos planteados, siendo los resultados obtenidos lo siguiente:

Se aprecia que se han desarrollado veintitrés (23) módulos principales, para la generación de código fuente del diseño del formulario a código PHP, en donde se crearon un total de 249 procedimientos entre los (23) módulos, de los cuales 158 son procedimiento de tipo FUNCTION y 91 son de tipo SUB, en donde se ha generado un total de 4890 líneas de código necesarios para realizar las tareas planteadas en los objetivos, de los cuales se observa que el promedio de líneas de código por procedimiento es de 25, siendo este no mayor a 50 líneas de código por cada procedimiento, esto hace que el desarrollo pueda tener una mayor visión de lo que se está trabajando en el procedimiento. Para mayor información de las métricas realizadas a las líneas de código por cada módulo se le recomienda revisar el anexo de métricas realizadas.

Tabla 1

Líneas de código fuente por módulos y procedimientos

Ítem	Módulos	Procedimientos				
		Function	Sub	Total	Número de Líneas	Promedio de Líneas por Procedimiento
1	mdlAnalizeCode.bas	1	2	3	81	27
2	mdlCommandos.bas	17	1	18	190	11
3	mdlCompact.bas	0	1	1	38	38
4	mdlCompile.bas	0	1	1	75	75
5	mdlCompile_Controller.bas	9	0	9	161	18
6	mdlCompile_css.bas	7	4	11	245	22
7	mdlCompile_gridView.bas	0	8	8	226	28
8	mdlCompile_MasterController.bas	19	1	20	539	27
9	mdlCompile_Model.bas	14	0	14	445	32
10	mdlCompile_root.bas	0	5	5	156	31
11	mdlCompile_Script.bas	12	1	13	433	33
12	mdlCompile_Transac.bas	0	3	3	89	30
13	mdlCompile_ViewForm.bas	3	12	15	270	18
14	mdlDataBase.bas	14	9	23	416	18
15	mdlFormulario.bas	0	1	1	37	37
16	mdlIni.bas	3	0	3	18	6
17	mdlMain.bas	1	8	9	92	10
18	mdlMD5.bas	11	5	16	285	18
19	mdlObj.bas	24	5	29	499	17
20	mdlParse.bas	1	0	1	41	41
21	mdlRibbon.bas	0	17	17	131	8
22	mdlSound.bas	0	7	7	48	7
23	mdlUtil.bas	22	0	22	375	17
TOTAL		158	91	249	4890	569
PROMEDIO						25

3.5.4.2. Métricas de tiempo

Se realizaron los cálculos del tiempo de procesamiento durante la compilación de formularios, para determinar la velocidad del proceso de compilación de los mismos, para lo cual se implementaron formularios, cargados con controles, algunos de los casos solo se agregaron cantidades mínimas para el proceso de compilación, de los cuales se obtuvieron los siguientes resultados en segundos:

Como se observa en la Tabla N° 2, los tiempos de generación de código fuente de un formulario son cortos, teniendo un total de 5.98 segundos, para generar el código fuente de formularios diseñados, esto hace que el proceso de implementación de un software vía web se agilice.

Tabla 2

Métrica de tiempos de compilación de formulario y grids en segundos

Formulario	**Compilación del formulario**
OFUPDFILE	2.720
OFREMOD	1.824
OFTDOC	1.024
OGCTIPO	0.416
TOTAL	5.984

Cada formulario generado, cuenta con sub componente, los cuales se detallan en la Tabla N° 3, en donde se especifican el tiempo de procesado por cada sub componente de un formulario, en los cuales se puede mostrar los tiempo de generación de código fuente de los componentes de Framework que son Modelo (M) que tiene un tiempo de 1.696 segundos, Vista (V) con un tiempo de 1.216 segundos y Controlador (C) con un tiempo de 0.192 segundos, a su vez muestra los tiempo de carga de las variables (Var), tiempo de generación de los estilos de una página web

(CSS) con un tiempo de 0.352 segundos, Tiempo de creación de los script (JS) con un tiempo de 2.016 segundos, tiempo de generación de código de formularios tipo Grid relacionados al formulario principal con un tiempo de 0.160 segundos, Creación del Controlador Maestro del Formulario (Ctrl-M) con un tiempo de 0.192 segundos y el tiempo de creación de las transacciones utilizadas para generar las llamadas al formulario (Transac) con un tiempo de 0.160 segundos.

Tabla 3

Métrica de tiempos de compilación de sub componentes del formulario en segundos

Formulario	Var	M	V	C	Css	Js	Grid	Ctrl-M	Transac
OFUPDFILE	0.000	0.800	0.576	0.064	0.128	0.992	0.032	0.064	0.064
OFREMOD	0.000	0.480	0.416	0.032	0.128	0.672	0.032	0.032	0.032
OFTDOC	0.000	0.288	0.160	0.096	0.064	0.256	0.064	0.064	0.032
OGCTIPO	0.000	0.128	0.064	0.000	0.032	0.096	0.032	0.032	0.032
Total	0.000	1.696	1.216	0.192	0.352	2.016	0.160	0.192	0.160

3.5.4.3. Métricas de carga de Memoria

Como se ha indicado, el proceso de verificación de carga de memoria se realizó durante la generación de código de cuatro formularios, obteniéndose que el porcentaje utilizado en la carga de memoria es nulo, como se muestra en la Tabla N° 4, donde se puede observar que la carga de memoria es de 0% del total de la memoria.

Tabla 4

Carga de memoria realizada durante la compilación

Formulario	Carga de Memoria %
OFUPDFILE	0
OFREMOD	0
OFTDOC	0
OGCTIPO	0

3.5.4.4. Métricas de uso de CPU

Para realizar el control de uso de CPU utilizado al realizar la compilación se realizó la captura del uso del CPU al inicio de la compilación y a su vez se volvió a capturar el uso de CPU al concluir la generación de código fuente, realizando la diferencia se obtuvo la cantidad que el generador de código utiliza del CPU, teniendo los resultados del uso en porcentaje en la Tabla N° 5, en donde se observa que el porcentaje de uso de CPU tiene una variación de 0.000013% a 0.000120% del total de CPU del ordenador, teniendo un acumulado de 0.000215% de uso de CPU de cuatro formularios generados.

Tabla 5

Métrica de uso de CPU en el momento de la compilación de cada formulario

Formulario	CPU %
OFUPDFILE	0.000120
OFREMOD	0.000046
OFTDOC	0.000036
OGCTIPO	0.000013
TOTAL	0.000215

A su vez en la Tabla N° 6, se obtuvo el porcentaje de uso de CPU por cada sub componente que el generador de código realiza para generar el código PHP compatible con el Framework de Codeigniter, en donde se observa que el porcentaje uso de CPU está comprendido entre 0.000002% a 0.000042% para lo cual se muestra los siguientes resultados.

Tabla 6

Métricas de uso de CPU en porcentaje, por cada sub componente del Framework

Formulario	Variables	Modelo	Vista	Controlador
OFUPDFILE	0.000002	0.000027	0.000042	0.000004
OFREMOD	0.000000	0.000013	0.000002	0.000000
OFTDOC	0.000002	0.000019	0.000000	0.000002
OGCTIPO	0.000002	0.000004	0.000006	0.000002
TOTAL	0.000006	0.000063	0.000050	0.000008

En la Tabla N° 7, se muestra información del porcentaje de uso de CPU en la generación de código durante la creación de las funciones auxiliares para el Framework de Codeigniter, en donde se observa que el uso de CPU oscila entre 0.000002% a 0.000044%, considerándose este valor en 0% de uso de CPU.

Tabla 7

Métricas de uso de CPU en Porcentaje, por cada sub componente y funciones auxiliares al Framework

Formulario	Css	Js	Grid	Controlador Maestro	Transacciones
OFUPDFILE	0.000006	0.000044	0.000002	0.000006	0.000004
OFREMOD	0.000015	0.000023	0.000004	0.000002	0.000006
OFTDOC	0.000006	0.000006	0.000010	0.000004	0.000004
OGCTIPO	0.000002	0.000002	0.000002	0.000000	0.000010
TOTAL	0.000029	0.000075	0.000018	0.000012	0.000024

3.5.4.5. Métricas de uso de GPU

Para el control de uso del GPU, se entiende que al utilizar el software Microsoft Excel a su vez de realizar el uso de CPU del ordenador, también utiliza recursos de GPU para mostrar la información en pantalla, así como las acciones que realiza el generador de código desarrollado para generar el código necesario para el Framework. Para esto se procedió a realizar con el mismo proceso de captura inicial del uso del GPU antes de la

compilación y al finalizar la compilación de un formulario, los resultados obtenidos se muestran en la tabla siguiente en porcentaje de uso.

Tabla 8

Métrica de uso de GPU en el proceso de compilación de formularios

Formulario	GPU %
OFUPDFILE	0.00012
OFREMOD	0.00006
OFTDOC	0.00003
OGCTIPO	0.00001
Total	0.00022

Así como también se procedió a realizar los cálculos de uso del GPU, en la Tabla N° 9 se muestra en cada sub procedimiento utilizado por el generador de código para generar el código necesario para el uso del Framework de Codeigniter, viéndose que el porcentaje de uso de GPU es menor a 0.0001% viéndose que para generar el código fuente, no se realiza uso considerable del GPU del ordenador.

Tabla 9

Métrica de uso de GPU en porcentaje, por cada sub componente del Framework

Formulario	Variables	Modelo	Vista	Controlador
OFUPDFILE	0.00000	0.00008	0.00005	0.00005
OFREMOD	0.00000	0.00002	0.00001	0.00001
OFTDOC	0.00000	0.00000	0.00000	0.00001
OGCTIPO	0.00000	0.00001	0.00000	0.00001
Total	0.00000	0.00011	0.00006	0.00008

Se procedió a realizar el cálculo de uso de GPU en el proceso de generación código fuente en relación a los sub componentes y funciones auxiliares, en la Tabla N° 10 se observa que en la generación del componente CSS es un porcentaje de 0.00013% acumulado durante la generación de cuatro formularios, en la generación del script se tiene un

porcentaje de 0.00021%, en el componente Grid se tiene 0.00022%, en el controlador maestro se tiene un 0.00023% de uso de GPU y se tiene un 0.00022% en la generación de las transacciones de acceso a cada formulario.

Tabla 10

Métricas de uso de GPU en porcentaje, por cada sub componente y funciones auxiliares al Framework

Formulario	Css	Js	Grid	Controlador Maestro	Transacciones
OFUPDFILE	0.00008	0.00012	0.00012	0.00012	0.00012
OFREMOD	0.00004	0.00007	0.00007	0.00007	0.00006
OFTDOC	0.00000	0.00002	0.00003	0.00003	0.00003
OGCTIPO	0.00001	0.00000	0.00000	0.00001	0.00001
TOTAL	0.00013	0.00021	0.00022	0.00023	0.00022

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

Se ha realizado una verificación de sistemas con las que cuenta la Universidad Nacional del Altiplano y se ha encontrado, que se encuentra en la aplicación de grupo de sistemas basados en Web, los cuales fueron desarrollados por una consultora. Por lo que, para la universidad es necesario que se implementen en la mayoría de oficinas sistemas Web que puedan reportar información, para lo cual el desarrollo de este aplicativo que permite realizar las tareas para formularios de tipo de registro, aceleraría estos procedimientos.

4.1. Exposición de resultados

Según lo establecido em los objetivos a continuación se muestran los resultados obtenidos para el cumplimiento de estos.

4.1.1. Elaboración del módulo para el diseño de formularios

Luego de haber hecho un análisis del funcionamiento de Microsoft Excel, este permitía realizar, modificaciones en su cinta de opciones, por lo que, para una mayor flexibilidad, se decidió agregar una nueva cinta de opciones, para el sistema, para así poder facilitar al usuario el manejo y creación de los mismos.

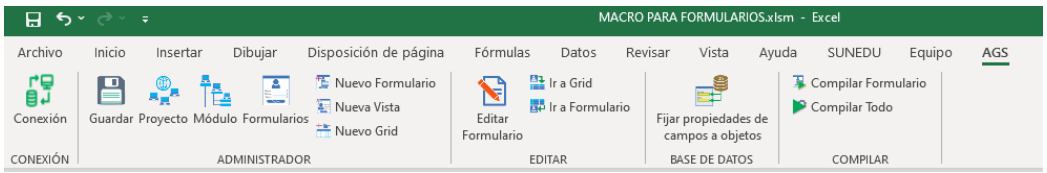


Figura 41. Cinta de opciones generada con Custom UI Editor for Microsoft Office

Para la implementación de esta cinta de opciones se utilizó "*Custom UI editor for Microsoft Office"* el cual permitió editar su archivo XML y agregar las opciones correspondientes, esto facilito la edición del diseño de un formulario.

Para el manejo de proyectos, se elaboró un formulario, el cual permite administrarlos y seleccionar uno para la elaboración de diseños de formularios.

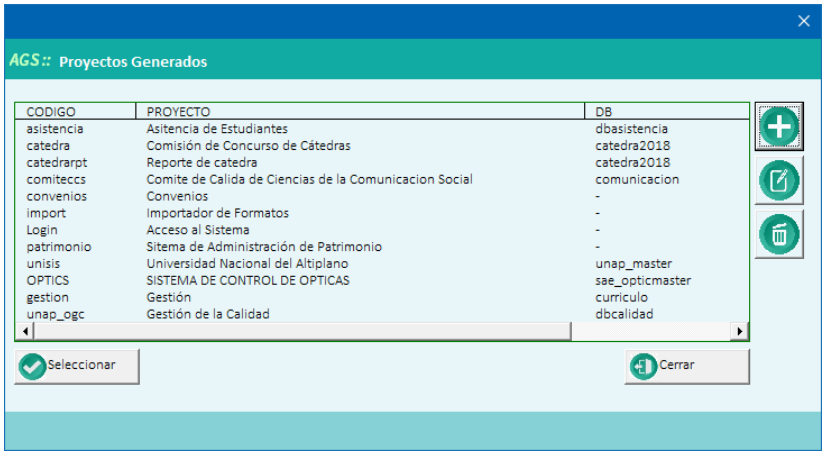


Figura 42. Formulario de administración de Proyectos

Una vez que se ha seleccionado el proyecto, en la cinta de opciones se puede seleccionar o agregar un módulo o sub módulo en la cual se va a trabajar.

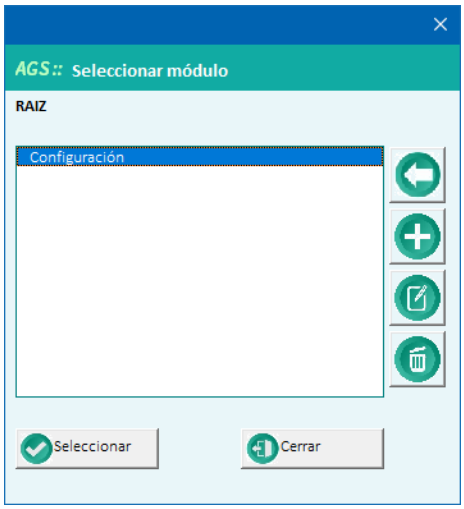


Figura 43. Formulario de administración de módulos

Ya seleccionado el módulo, también se cuenta con un administrador de formularios.

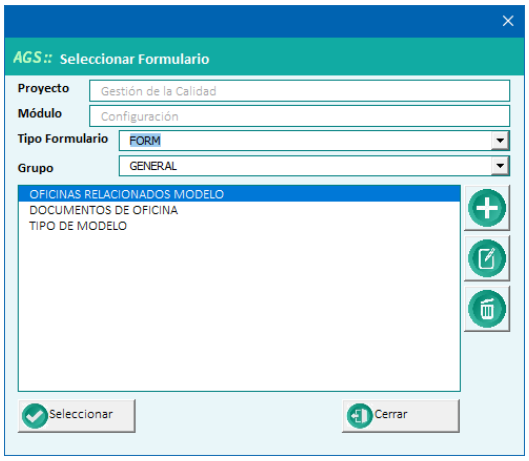


Figura 44. Administrador de formularios

En este administrador de formularios se puede seleccionar un formulario ya generado o crear uno nuevo, dependiendo del tipo de formulario de la cual se desea.

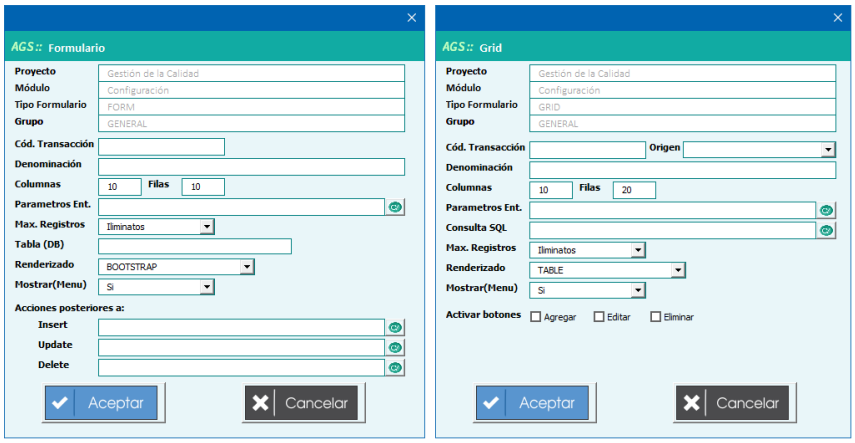


Figura 45. Formulario de tipo Form y Grid

Para el diseño de formularios se creó una hoja Excel, para que el usuario pueda diseñar el formulario según sus requerimientos, en el área de diseño

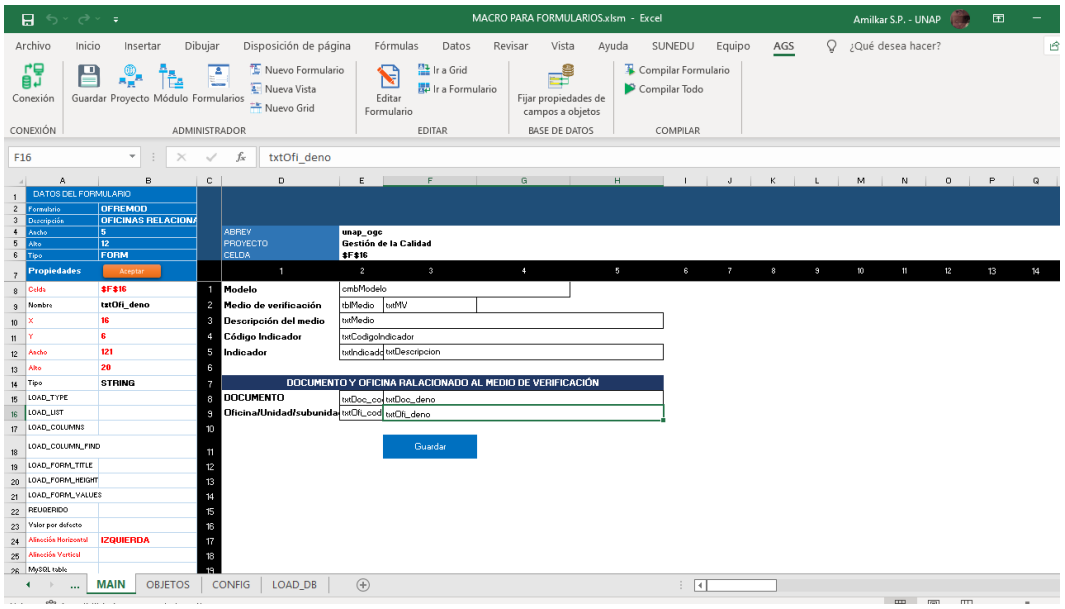


Figura 46. Hoja Excel de para diseño de formularios

Como se pudo ver, que en artículos de investigación es frecuente el uso de formularios y la creación de botones dentro de la hoja Excel, como en artículo "GeoBalance: un programa Excel VBA para el cálculo del balance de masa en geociencias" (Li et al., 2020b) en donde hacen uso de las macros en Exel para realizar cálculos en el balance de masa en geociencias, en dende se observó que hace uso de hojas de cálculo del Excel habilitando las macros para el uso de VBA, en la cual se hacen uso de algoritmos que permite realizar dichos cálculos de manera más rápida y así optimizaron el proceso de cálculo de las masas, en cambio en el artículo "CGDK: un programa CorelDRAW con VBA extensible para dibujo geológico" (Qiu et al., 2013), en este facilita la elaboración de dibujos, vinculando los datos de Excel, y diagramándolo en CorelDraw, por lo que en el diseño que se propone es la elaboración de un formulario en una hoja Excel, para después poder generarlo una vista en HTML con funcionalidades diferentes. Como se observó en área de diseño, la creación de formularios, grids los cuales fueron convertidos en código para páginas Web.

4.1.2. Desarrollo del generador de código

El desarrollo del generador de código es parte importante del proyecto, dado que este es el medio en el cual, se podrá observar los resultados del diseño que se realiza en la hoja Excel, en un navegador de Internet, para esto se ha identificado la estructura del Framework de Codeigniter, así como también se verifico la estructura

necesaria de una HTML, CSS, JSON y JavaScript así como también JQuery, para lo cual se ha vinculado los controles insertados en la hoja Excel al momento de realizar la compilación con los nombres de dichos objetos, creando así los diferentes procedimientos y estilos necesarios para visualización del formulario a continuación se muestra los resultados de la compilación de en cada una de sus fases:

a) Se inicia con la creación de la vista, para lo cual se verifica los controles insertados y se procede con la generación del código en formato HTML, CSS y Js, los cuales son necesarios para que el navegador de internet pueda mostrar información.

```
44      }
45  ?>
46  <link href="../../styles/Ogcccfg/css/Css_Ofremod.css" rel="stylesheet">
47  <script src='../../styles/Ogcccfg/js/Js_Ofremod.js?l=<?php echo rand(100,10000); ?>'></script>
48  <INPUT TYPE="HIDDEN" name="method" id="method" value="b2bfe2fbbb845a9a4629e63f18a474f0"  />
49  <TABLE Width = '0px' border = "0" >
50      <TR height='0px'>
51          <TD class = "Ofremod_col01"></TD>
52          <TD class = "Ofremod_col02"></TD>
53          <TD class = "Ofremod_col03"></TD>
54          <TD class = "Ofremod_col04"></TD>
55          <TD class = "Ofremod_col05"></TD>
56      </TR>
57      <TR height='30px' >
58          <TD  ><div id='lblD8' name='lblD8' class='Ofremod_lblD8'>Modelo</div></TD>
59          <TD  COLSPAN=3 ><SELECT id='cmbModelo_sel' name='cmbModelo_sel' class='Ofremod_cmbModelo' required>
            <?php Select_ListTable($cmbModelo["LIST"], $cmbModelo["CODIGO"], $cmbModelo["DENO"]); ?></SELECT><INPUT
             TYPE='HIDDEN' id='cmbModelo' name='cmbModelo' value='-1'></TD>
60          <TD  ></TD>
61      </TR>
62      <TR height='30px' >
63          <TD  ><div id='lblD9' name='lblD9' class='Ofremod_lblD9'>Medio de verificaci&oacute;n</div></TD>
64          <TD  ><div class='Ofremod_tblMedio_box0'>
65                  <div class='Ofremod_tblMedio_box1'>
66                      <input type='text' id='tblMedio' name='tblMedio' value = '<?php echo $tblMedio; ?>'
                        maxlength='11' class='Ofremod_tblMedio' required ReadOnly/>
67                  </div>
68                  <div id='Ofremod_tblMedio_box2'>
69                      <button type='button' id='tblMedio_btn' name='tblMedio_btn' class='btn btn-primary
```

Figura 47. Código en HTML generado

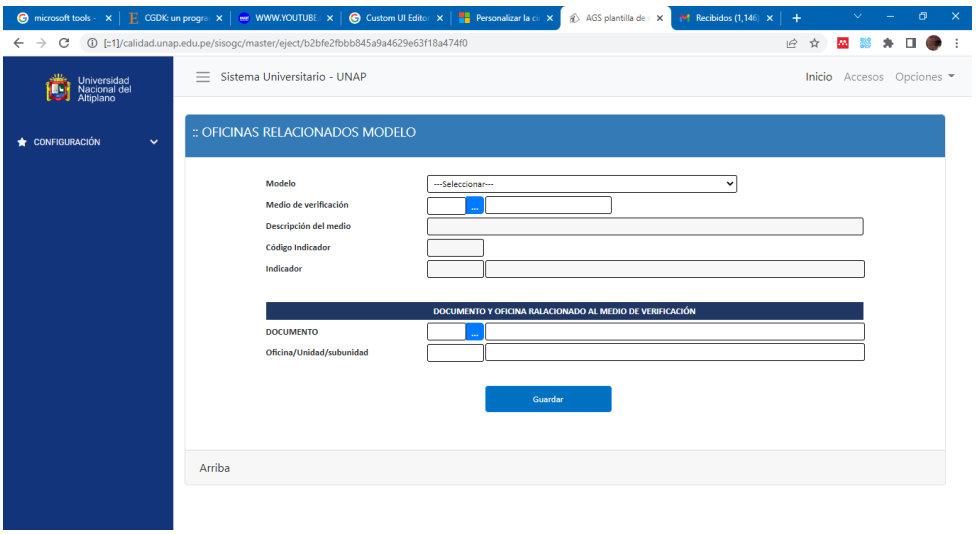


Figura 48. Vista de resultados en navegador de internet

b) Generando código para el Framework de Codeigniter

El generador de código genera código PHP, según la arquitectura de Modelo – Vista - Controlador utilizada por el Framework de Codeigniter, para lo cual sistema realiza la compilación de lo siguiente

- Controlador del Formulario: El cual se encuentra ubicado en la carpeta de controladores, dicho controlador tiene las funciones propias establecidas al memento del diseño, por lo que, al realizar la compilación se genera una clase extendida del controlador del Framework de Codeigniter

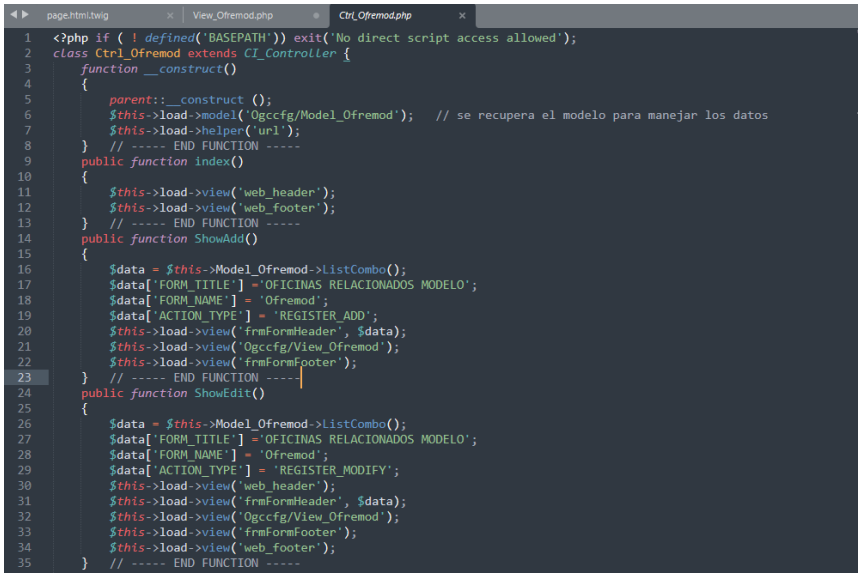


Figura 49. Controlador del formulario compilado

- Modelo, como se había indicado el modelo nos permite realizar la conexión con la base de datos, para así de esta manera registrar información, para lo cual el generador de código genera código necesario para realizar dicho vínculo.

```php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Model_Ofremod extends CI_Model {
     function __contruct(){
         parent::__construct();
     }  // ----- END FUNCTION -----
    //Realizar consulta en base a las llamadas
    public function Query($QUERY)
    {
        $status = true;
        try
        {
            $query = $this->db->query($QUERY);
            if($query->num_rows() == 0 )
                $status = false;
            else
            {
                $rows = $query->result();
                return $rows;
            }
        }
        catch (Exception $e) {
            $status = false;
        }
        return $status;
    }   // ----- END FUNCTION -----
    //Recupera informacion de un soslo Registro de la base de datos y si no existen mas devuelve FALSE
    public function GetInfoRow($QUERY)
    {
        $status = true;
        try
        {
            $query = $this->db->query($QUERY);
            if($query->num_rows() == 0 )
                $status = false;
            else
```

Figura 50. Código generado del Modelo para acceso a base de datos

Como podemos ver, el proceso de compilado, una vez finalizado el diseño sin algún tipo de error al momento de diseño no le lleva realizar dicha tarea mucho tiempo, por lo que en el diseño y la compilación hace que usuario pueda realizar modificaciones constantes hasta llegar a un producto terminado.

c) Reportes en formato PDF, para el diseño de reportes en formato el sistema necesitará de la librería TCPDF, dicha librería lo podemos descargar de https://tcpdf.org/ e instalarlo en la carpeta de librerías del Framework de Codeigniter, para realizar los reportes es necesario que el área usuaria haga uso del diseñador de formularios, haciendo uso de los formularios de tipo vista o de tipo grid, para poder generar los reportes correspondientes.

page.html.twig | View_Ofremod.php | Model_Ofremod.php | Catedra.php

```
137         {
138
139                 //if( !isset($_SESSION['usuario_cod']) )  redirect("/");
140
141         $this->load->library('Pdf');
142         ob_start();
143         ini_set('display_errors',0);
144         ini_set('log_errors',1);
145         ob_end_clean();
146
147         //session_start();
148         $apenom = ""; //$_SESSION['usuario_infodata'];
149
150         $pdf = new Pdf('P', 'mm', 'A4', true, 'UTF-8', false);
151         $pdf->active_borrador(true);
152         $pdf->SetCreator(PDF_CREATOR);
153         $pdf->SetAuthor('Comisión de Concurso de Cátedras 2019');
154         $pdf->SetTitle('Curriculum Vitae - ' . $apenom);
155         $pdf->SetSubject('CONCURSO PÚBLICO DE DOCENTES EN LA MODALIDAD DE CONTRATO 2019');
156         $pdf->SetKeywords('');
157
158 // datos por defecto de cabecera, se pueden modificar en el archivo tcpdf_config_alt.php de libraries/config
159         $pdf->SetHeaderData(PDF_HEADER_LOGO, PDF_HEADER_LOGO_WIDTH, 'UNIVERSIDAD NACIONAL DEL ALTIPLANO', "
                COMISION DE CONCURSO DE CATEDRAS\nCONCURSO PÚBLICO DE DOCENTES EN LA MODALIDAD DE CONTRATO
                2019", array(0, 0, 0), array(0, 0, 0));
160         //$pdf->SetHeaderData("../../images/borrador.png" , 50, "");
161         $pdf->setFooterData($tc = array(0, 0, 0), $lc = array(0, 0, 0));
162
163 // datos por defecto de cabecera, se pueden modificar en el archivo tcpdf_config.php de libraries/config
164         $pdf->setHeaderFont(Array(PDF_FONT_NAME_MAIN, '', PDF_FONT_SIZE_MAIN));
165         $pdf->setFooterFont(Array(PDF_FONT_NAME_DATA, '', PDF_FONT_SIZE_DATA));
166
167 // se pueden modificar en el archivo tcpdf_config.php de libraries/config
168         $pdf->SetDefaultMonospacedFont(PDF_FONT_MONOSPACED);
169
```

Figura 51. Generación de un Reporte en Formato PDF

4.1.3. Validación del software

Según las métricas desarrolladas, realizaron las pruebas necesarias en relación a:

- Líneas código, se observó que la cantidad de líneas de código en los diferentes módulos desarrollados, no superan en gran parte a las 50 líneas de código, por lo que su mantenimiento no se hace complejo (Vea Tabla 1).
- Tiempo de Procesamiento, según la Tabla 2, el tiempo de procesamiento de compilación de un formulario, es un máximo de 2.72 segundo y un mínimo de 0.42 segundos, por lo que a la macro no llevaría mucho tiempo de generar el código fuente necesario para que se genere código HTML y PHP.
- Carga de Memoria, como se observa en la Tabla 4, la macro no realiza carga de memoria innecesaria para realizar la compilación de un formulario y llevarlo a código fuente den HTML y PHP necesario para que el Framework de Codeigniter funcione.
- Uso de recurso de CPU, como se puede observar en la Tabla 5, la macro desarrollada hace uso de CPU en un 0.000120 porciento, por lo que, no le lleva muchas complicaciones al CPU para poder procesar la información durante la compilación del mismo.
- Uso de recurso de GPU, en la Tabla 8, la macro implementada hace uso del 0.0001 porciento del uso del GPU del ordenador,

Según los cálculos realizados en tiempo de procesamiento de generación de código fuente de un formulario, es de un máximo de 2.72 segundos en relación del total que es de 5.984 segundo, por lo que a la macro no llevaría mucho tiempo de generar el código fuente necesario para que se genere código HTML y PHP, así mismo la carga de memoria en relación al total de 0% por lo que al generar el código fuente no se realiza una carga sustancial en la memoria del ordenador, con respecto al uso de CPU del ordenador, se observa que se tiene un 0.000215% del total del CPU del ordenador, siendo esto una cantidad prácticamente de cero. Con relación al uso del GPU o video del ordenador se tiene que el sistema hace un uso del 0.000022% de la memoria de video.

Tabla 11

Cuadro comparativo de procesamiento durante la compilación

Formulario	Tiempo Compilación del formulario (Segundos)	Carga de Memoria %	Uso de CPU %	Uso de GPU %2
OFUPDFILE	2.720	0	0.00012	0.00012
OFREMOD	1.824	0	0.000046	0.00006
OFTDOC	1.024	0	0.000036	0.00003
OGCTIPO	0.416	0	0.000013	0.00001
TOTAL	5.984	0	0.000215	0.00022

4.1.4. Prueba estadística de validación

Para realizar la validación de la aplicación se utilizó la prueba t se, el cual nos permitirá realizar la comparación de medias de dos muestras obtenidas mediante, la primera fue obtenida a partir de una encuesta realizada a expertos y la otra utilizando este generador de código, esto nos permitirá determinar si existe una diferencia significativa entre ellas.

En relación a la encuesta a los expertos, se ha procedido a realizar una encuesta sobre el tiempo (horas de trabajo) en el cual le toma al experto (desarrollador web) en implementar una aplicación. Se le pregunto sobre cuánto tiempo le toma en desarrollar una aplicación web, con un registro de datos personales, Registro de Grados Académicos (permita subida de archivos), y un listado de los mismo, con su respectivo menú lateral izquierdo de acceso (En horas de trabajo).

Se determinó que el promedio de desarrollo de una aplicación web según las características indicadas en la encuesta es de 40.1 horas de trabajo, por lo que un programador, trabajando ocho (8) horas diarias culminaría en cinco días la aplicación web.

Tabla 12

Resultados de la pregunta sobre el tiempo de desarrollo de unas aplicaciones web.

Experto	Tiempo en horas
1	75
2	72
3	4
4	24
5	24
6	12
7	10
8	60
9	80
Promedio en horas	40.1
Días de trabajo (8 horas)	5.01

Se realizaron pruebas de desarrollo de algunos proyectos con la aplicación elaborada con el presente trabajo, se obtuvieron los siguientes resultados en minutos, los cuales fueron convertidor en horas.

De los proyectos generados con el sistema se puede observar que el promedio que llevaría desarrollar aplicaciones es de 0.43 horas, que es equivalente a 25.67 minutos.

Tabla 13

Tiempo de desarrollo de aplicación utilizando la macro del proyecto

Proyecto	Tiempo en Minutos	Tiempo en Horas
1	30	0.50
2	22	0.37
3	32	0.53
4	28	0.47
5	16	0.27
6	26	0.43
Promedio	25.67	0.43

Para demostrar los resultados utilizaremos la prueba estadística *t de muestra* y compararemos los resultados, para poder validad nuestra hipótesis del presente trabajo.

Se verificó que la aplicación efectivamente acelera el proceso de desarrollo de aplicaciones. Los resultados obtenidos son altamente significativos ($p < 0.01$), lo que respalda la hipótesis alterna y valida la aplicación desarrollada mediante el uso de macros con Visual Basic para Aplicaciones. Estos resultados demuestran claramente los beneficios y la eficacia de la aplicación en términos de mejorar la eficiencia y la productividad en el desarrollo de aplicaciones.

Tabla 14

Prueba t de una muestra para comparación de medias (tiempo)

N	Media	Desv.Est.	Error estándar de la media	Límite superior de 95% para μ
6	0.4278	0.0976	0.0398	0.5080

μ: media de población de Tiempo

Prueba

Hipótesis nula	H_0: $\mu = 40.01$, La aplicación no acelera el proceso de desarrollo de aplicaciones web.
Hipótesis alterna	H_1: $\mu < 40.01$, La aplicación si acelera el tiempo de desarrollo de aplicaciones web

Valor T	Valor p
-993.78	0.000

DISCUSIÓN

En el artículo de Xu Cao, "EMMTE: An Excel VBA tool for source apportionment of nitrate based on the stable isotope mixing model", realiza una efectiva utilización de la hoja de cálculo de Excel, combinando con la simulación de Monte Carlo y la desviación relativa de restricciones, para lo cual tuvieron que recurrir a la utilización de las macros y poder incluir los algoritmos necesarios, los cuales les permitieron resolver ecuaciones de balance de masa, por tanto es importante, el manejo de la programación de macros para Visual Basic para Aplicaciones, así como también conocer sobre lo que se desea resolver y utilizar las metodologías adecuadas.

Xiaoyan en su artículo "GeoBalance: An Excel VBA program for mass balance calculation in geosciences", se encontró que para el cálculo de balance de masa de su aplicación denominada GeoBalance, se centraron en resolver problemas de mínimos cuadrados usando pseudo-inversa de matriz de descomposición de valores singulares, en donde para poder aplicar estos cálculos obligatoriamente tuvieron que utilizar las macros de Visual Basic para Aplicaciones, y dar soluciones correspondientes, así también era necesario conocer algoritmos para dar solución mediante la utilización de las macros

En el artículo Daojun Zhang, se observa que la utilización de las macros, no solamente se utiliza en Microsoft Excel, sino que también es posible utilizar en otro software, en donde en el artículo utilizó ArcGis en donde desarrollaron e implementaron un procesamiento por lotes utilizando Excel con VBA, el cual les permitía separar las anomalías en la geoquímica de exploración como en la geoquímica de ambiental.

Los resultados de este estudio indican que la implementación de macros en VBA puede ser una estrategia efectiva para automatizar procesos en el desarrollo de aplicaciones en diferentes áreas, así como en el desarrollo de aplicaciones Web planteadas en esta investigación, haciendo uso de la arquitectura de los Frameworks en nuestro caso el de CodeIgniter conjuntamente con el uso de hojas de cálculo de Excel como medios de diseño de los formularios. En particular, se encontró que las macros pueden reducir significativamente el tiempo necesario para completar tareas repetitivas y minimizar la posibilidad de errores humanos.

Además, se encontró que la capacidad de Visal Basic para Aplicaciones para interactuar con otras aplicaciones de cómo son Base de Datos de MySql, ODBC, Microsoft Office (Word, PowerPoint, Outlook), Autocad, Inventor, CorelDraw, ArcGis, así como también

permite el control de Windows haciendo uso de sus API del kernel del sitema, por lo que esto puede aumentar la eficiencia y la productividad en entornos de oficina o diseño que utilizan múltiples programas.

Sin embargo, es importante tener en cuenta que la implementación de macros en VBA requiere conocimientos avanzados de programación y que la creación de macros personalizadas puede ser un proceso lento y laborioso esto en el desarrollo de sistemas más complejos. Además, la falta de experiencia en programación puede llevar a la creación de macros ineficaces o inseguras.

Por lo tanto, se recomienda que las empresas o instituciones estatales proporcionen capacitación y recursos adecuados para garantizar que los empleados tengan las habilidades necesarias para implementar macros de manera efectiva. Además, se sugiere que se realicen pruebas exhaustivas de las macros antes de su implementación para asegurar que no introduzcan errores o problemas de seguridad.

En conclusión, aunque la implementación de macros en VBA puede ser una herramienta valiosa para aumentar la eficiencia y la productividad en entornos de oficina, se deben tomar precauciones para garantizar que se utilicen de manera efectiva y segura.

CONCLUSIONES

- En este proyecto, se logró desarrollar una aplicación utilizando Visual Basic para Aplicaciones en Microsoft Excel con el objetivo de optimizar el desarrollo de sitios web. La aplicación resultante permite un diseño eficiente y simplificado de páginas web, ofreciendo a los usuarios una interfaz intuitiva. Mediante la integración de Visual Basic para Aplicaciones, Microsoft Excel y el Framework de Codeigniter, se han logrado automatizar tareas repetitivas, agilizar el proceso de diseño y garantizar la coherencia en la estructura y el aspecto visual de las páginas. Esta solución ha demostrado ser una herramienta efectiva para aumentar la productividad y la calidad del desarrollo web, al proporcionar un entorno familiar y accesible para diseñadores y desarrolladores. En resumen, la aplicación desarrollada con Visual Basic para Aplicaciones en Microsoft Excel ha sido un éxito en la optimización del desarrollo web, mejorando la eficiencia y facilitando el diseño de páginas web de alta calidad.

- La creación del generador de código o intérprete ha sido un elemento fundamental en este proyecto, permitiendo automatizar las tareas del programador y reducir considerablemente los tiempos de desarrollo de páginas web. Este generador de código es capaz de trabajar en conjunto con los formularios diseñados en la hoja de diseño en Microsoft Excel, convirtiéndolos en código compatible con el Framework de Codeigniter. Como resultado, se logra un proceso eficiente de generación de código, que incluye procedimientos de seguridad implementados a través de un sistema de transacciones controlado por el controlador maestro. Esta integración entre el diseño en Excel y el Framework de Codeigniter ha demostrado ser exitosa, brindando una solución que agiliza el desarrollo web y garantiza la integridad y seguridad del sistema. En resumen, el generador de código desarrollado ha sido una herramienta clave en la optimización del proceso de desarrollo web, permitiendo una programación automatizada y confiable a través de una interfaz intuitiva en Microsoft Excel.

- Durante el proceso de diseño de formularios, vistas y cuadrículas, el generador de código desarrollado ha permitido imprimir dichos códigos utilizando la librería TCPDF. Para lograrlo, se utiliza el procedimiento 'writeHTML' de la librería TCPDF, que facilita la inclusión de los códigos generados a partir de la hoja de

diseño implementada. Finalmente, se emplea el procedimiento 'Output' para escribir y completar el archivo PDF resultante. Esta integración con TCPDF ha demostrado ser eficiente y efectiva en la impresión de los códigos generados en los diseños de formularios, vistas y cuadrículas, ofreciendo una forma cómoda de visualizar y compartir la información. En resumen, el generador de código ha permitido la impresión automatizada de los códigos utilizando la librería TCPDF, mejorando así la capacidad de generar archivos PDF a partir del diseño realizado en el proyecto.

- Se pudo determinar que la prueba estadística de t para una muestra, era determinante la consulta a expertos sobre el tiempo en desarrollar una aplicaciones web, el cual nos indicaron el tiempo en horas de trabajo, así como también fue importante realizar pruebas de tiempo de desarrollo haciendo uso de la macro, lo cual permitió a la prueba realizar una comparación, de que si, los módulos desarrollados aceleran el proceso de creación de una aplicación web, siendo la prueba altamente significativa ($p < 0.01$), de que el uso de esta macro si acelera el proceso de desarrollo de aplicaciones web. Así mismo se realizó uso de métricas en el desarrollo de software, el cual nos proporciona indicadores de satisfacción en el proceso de desarrollo e implementación de software durante la compilación de estos, dando resultados óptimos durante el proceso de compilación.

- El uso de macros a través de Visual Basic para Aplicaciones se considera fundamental para brindar soluciones rápidas en los procesos y contribuir a la mejora continua de la organización. En comparación con los procedimientos manuales, el uso de macros permite reducir significativamente el tiempo requerido para realizar tareas. Además, la optimización de las macros proporciona una variedad de enfoques y perspectivas para abordar los desafíos planteados. Esto se evidencia en los diversos antecedentes mencionados, donde se ha demostrado la eficacia de las macros en la mejora de la productividad y eficiencia. En resumen, el uso de macros con Visual Basic para Aplicaciones ofrece soluciones rápidas, mejora continua y perspectivas innovadoras en los procesos, lo que se traduce en beneficios significativos para la organización.

RECOMENDACIONES

- Recomendamos enfocar los esfuerzos en mejorar el sistema con el objetivo de generar aplicaciones web más complejas. Aunque se ha puesto énfasis en la creación de formularios, vistas y tablas, es importante reconocer la necesidad de implementar otros comandos adicionales en la hoja de diseño para potenciar aún más las aplicaciones web. Estas mejoras desempeñarán un papel fundamental en la expansión de la funcionalidad y la versatilidad del sistema, permitiendo la incorporación de características avanzadas y optimizando la experiencia del usuario. Recomendamos priorizar la investigación y desarrollo de estos comandos adicionales que mejorarán la hoja de diseño, lo cual contribuirá a la creación de aplicaciones web más completas y sofisticadas. Esta evolución del sistema permitirá ofrecer soluciones más robustas y adaptadas a las necesidades de los usuarios. En resumen, recomendamos trabajar en la integración de comandos adicionales en la hoja de diseño para mejorar y enriquecer las aplicaciones web generadas por el sistema.

- Recomendamos enfocar los esfuerzos en lograr una integración más sólida de las opciones generadas en este proyecto con Microsoft Excel. En la configuración actual, existe una dependencia de hojas adicionales en Excel, lo cual puede generar problemas si estas hojas son eliminadas o se modifica su estructura. Para evitar estas limitaciones, es fundamental desarrollar una integración más robusta que asegure un funcionamiento estable y sin interrupciones. Para lograrlo, recomendamos explorar alternativas que reduzcan la dependencia de hojas adicionales y garanticen la compatibilidad y coherencia del sistema, incluso en caso de cambios en la estructura de Excel. El objetivo es lograr una mayor integración de las opciones generadas en este proyecto con Microsoft Excel, superando las limitaciones actuales y asegurando un funcionamiento sólido y confiable del sistema. Esta mejora permitirá una mayor flexibilidad y escalabilidad, y reducirá la vulnerabilidad ante cambios inesperados en las hojas de Excel. En resumen, recomendamos priorizar la implementación de una integración más robusta y estable para garantizar la eficacia y la confiabilidad del sistema en el largo plazo.

- Recomendamos incrementar la funcionalidad del sistema para incluir un sólido manejo de usuarios. En la configuración actual, los formularios y vistas se generan de manera general, lo que limita la capacidad de acceso personalizado de cada usuario a los elementos correspondientes. Para mejorar esta situación, es fundamental implementar un sistema de gestión de usuarios que permita asignar permisos y privilegios individuales. De esta manera, se garantizará que cada usuario tenga acceso exclusivo a los formularios, vistas y otros componentes pertinentes, según sus roles y responsabilidades específicos. Esta mejora en el manejo de usuarios fortalecerá la seguridad, personalización y eficiencia del sistema en general. En resumen, recomendamos ampliar el sistema para incluir un sólido manejo de usuarios, lo que proporcionará un acceso más preciso y controlado a los formularios, vistas y otros componentes del sistema, mejorando así la experiencia del usuario y la seguridad del sistema.

- Recomendamos ampliar la funcionalidad del sistema para permitir la generación de código no solo para aplicaciones web, sino también para otros tipos de aplicaciones como Java, Android u otros. Esto brindará a los usuarios la flexibilidad de utilizar el diseño existente en diferentes entornos de desarrollo y maximizar la reutilización del código. Al implementar esta capacidad, se fomentará la eficiencia y la versatilidad en el proceso de desarrollo de aplicaciones, permitiendo a los usuarios aprovechar al máximo el diseño realizado en el proyecto. Además, al ofrecer soporte para diferentes plataformas, se abrirán nuevas oportunidades y posibilidades de expansión. En resumen, recomendamos considerar la ampliación del sistema para permitir la generación de código en diversos tipos de aplicaciones, lo que proporcionará beneficios significativos en términos de reutilización, eficiencia y flexibilidad.

BIBLIOGRAFÍA

Calderon Vilca, E. F. (2020). Framework para el desarrollo escalable de sistemas de información en la nube orientado a MYPES. *Universidad Nacional Del Altiplano*. Recuperado de: https://repositorio.unap.edu.pe/handle/20.500.14082/18206

Canahuire Chambi, S. G. (2020). Análisis y solución de vulnerabilidad de seguridad en aplicaciones web y métodos de protección anti robot y HTTP request. *Universidad Nacional Del Altiplano*. Recuperado de: https://repositorio.unap.edu.pe/handle/20.500.14082/16118

Cao, X., He, W., Shi, Y., An, T., Wang, X., Liu, F., Zhao, Y., Zhou, P., Chen, C., & He, J. (2023). EMMTE: An Excel VBA tool for source apportionment of nitrate based on the stable isotope mixing model. *Science of The Total Environment*, *868*, 161728. https://doi.org/10.1016/J.SCITOTENV.2023.161728

Castaldi, G., Vermeersch, P., Zivelonghi, C., Benedetti, A., Buongiorno, A., Scott, B., Convens, C., Verheye, S., Ribichini, F., & Agostoni, P. (2023). Applicability of the EXCEL Trial Criteria to an All-Comers Real-World Cohort of Unprotected Left Main Percutaneous Coronary Intervention. *The American Journal of Cardiology*, *195*, 98–106. https://doi.org/10.1016/J.AMJCARD.2023.01.024

Cheng, W., Wang, K., & Zhang, X. (2010). Implementation of a COM-based decision-tree model with VBA in ArcGIS. *Expert Systems with Applications*, *37*(1), 12–17. https://doi.org/10.1016/j.eswa.2009.01.006

Cycle Time: qué es, cómo medirlo y mejorarlo en desarrollo de software. (n.d.). Retrieved April 17, 2023, Recuperado de: https://sentrio.io/blog/cycle-time-que-es/

Erdődi, L., Sommervoll, Å. Å., & Zennaro, F. M. (2021). Simulating SQL injection vulnerability exploitation using Q-learning reinforcement learning agents. *Journal of Information Security and Applications*, *61*, 102903. https://doi.org/10.1016/J.JISA.2021.102903

Firdous, R., & Devlin, J. F. (2014). BEARKIMPE-2: A VBA Excel program for characterizing granular iron in treatability studies. *Computers and Geosciences*, *63*, 54–61. https://doi.org/10.1016/j.cageo.2013.10.005

Gallino, J. P. S., De Miguel, M., Briones, J. F., & Alonso, A. (2014). Estrategia Guiada por Modelos para incluir Aspectos de Seguridad en Sistemas Empotrados Basados en Servicios Web. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, *11*(1), 86–97. https://doi.org/10.1016/J.RIAI.2013.11.006

Generador de código - GlosarioIT: Glosario Informático. (n.d.). Retrieved July 1, 2023, Recuperado de: https://www.glosarioit.com/Generador_de_código

Gupta, S., & Gupta, B. B. (2016). Automated Discovery of JavaScript Code Injection Attacks in PHP Web Applications. *Procedia Computer Science*, *78*, 82–87. https://doi.org/10.1016/J.PROCS.2016.02.014

Herrera Urtiaga, A. P. (2022). Extracción automática de metadatos para la administración del Repositorio Institucional de la Universidad Nacional del Altiplano Puno. *Universidad Nacional Del Altiplano*. Recuperado de: https://repositorio.unap.edu.pe/handle/20.500.14082/18923

Hongisto, T., & Saastamoinen, K. (2023). Using Excel based war simulator to support tactical wargaming. *Procedia Computer Science*, *225*, 912–921. https://doi.org/10.1016/J.PROCS.2023.10.078

How ajax works - javatpoint. (n.d.). Retrieved December 21, 2022, Recuperado de: https://www.javatpoint.com/how-ajax-works

Información general sobre Visual Basic for Applications de 64 bits | Microsoft Learn. (n.d.). Retrieved April 25, 2023, Recuperado de: https://learn.microsoft.com/es-es/office/vba/language/concepts/getting-started/64-bit-visual-basic-for-applications-overview

JAVA A TOPE: TRADUCTORES Y COMPILADORES CON LEX/YACC, JFLEX/CUP Y JAVACC. EDICIÓN ELECTRÓNICA. (2005).

Laura Murillo, R. P. (2019). Arquitectura pervasiva con tecnologías WebRTC híbridas para el desarrollo de un framework modelo vista controlador de tiempo real. *Universidad Nacional Del Altiplano*. Recuperado de: https://repositorio.unap.edu.pe/handle/20.500.14082/12281

Lenguaje de consulta estructurada (SQL) - Documentación de IBM. (n.d.). Retrieved

December 21, 2022, Recuperado de: https://www.ibm.com/docs/es/db2woc?topic=reference-sql

Li, X., Zhang, C., Almeev, R. R., & Holtz, F. (2020a). GeoBalance: An Excel VBA program for mass balance calculation in geosciences. *Chemie Der Erde*, *80*(2), 125629. https://doi.org/10.1016/j.chemer.2020.125629

Li, X., Zhang, C., Almeev, R. R., & Holtz, F. (2020b). GeoBalance: An Excel VBA program for mass balance calculation in geosciences. *Chemie Der Erde*, *80*(2), 125629. https://doi.org/10.1016/j.chemer.2020.125629

Moutaouakkil, A., & Mbarki, S. (2021). Transformation of Struts Model to Codeigniter Model. *Procedia Computer Science*, *184*, 767–772. https://doi.org/10.1016/J.PROCS.2021.03.095

Pop, D. P., & Altar, A. (2014). Designing an MVC Model for Rapid Web Application Development. *Procedia Engineering*, *69*, 1172–1179. https://doi.org/10.1016/J.PROENG.2014.03.106

Prokofyeva, N., & Boltunova, V. (2017). Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems. *Procedia Computer Science*, *104*, 51–56. https://doi.org/10.1016/J.PROCS.2017.01.059

Qiu, J. T., Song, W. J., Jiang, C. X., Wu, H., & Dong, R. M. (2013). CGDK: An extensible CorelDRAW VBA program for geological drafting. *Computers and Geosciences*, *51*, 34–48. https://doi.org/10.1016/j.cageo.2012.07.020

Qué es SCRUM – Proyectos Ágiles. (n.d.). Retrieved July 1, 2023, Recuperado de: https://proyectosagiles.org/que-es-scrum/

Santander Mamani, O. (2023). *Solución de las ecuaciones diferenciales ordinarias de orden superior usando Matlab/Simulink*. Recuperado de: https://repositorio.unap.edu.pe/handle/20.500.14082/20893

Suárez-Navarro, J. A., Gascó, C., Alonso, M. M., Blanco-Varela, M. T., Lanzon, M., & Puertas, F. (2018). Use of Genie 2000 and Excel VBA to correct for γ-ray interference in the determination of NORM building material activity concentrations. *Applied Radiation and Isotopes*, *142*, 1–7.

https://doi.org/10.1016/J.APRADISO.2018.09.019

Tabarés, R. (2021). HTML5 and the evolution of HTML; tracing the origins of digital platforms. *Technology in Society*, *65*, 101529. https://doi.org/10.1016/j.techsoc.2021.101529

Welcome to CodeIgniter4 — CodeIgniter 4.2.10 documentation. (n.d.). Retrieved December 21, 2022, Recuperado de: https://codeigniter.com/user_guide/intro/index.html

Wong, K. W. W., & Barford, J. P. (2010). Teaching Excel VBA as a problem solving tool for chemical engineering core courses. *Education for Chemical Engineers*, *5*(4), e72–e77. https://doi.org/10.1016/j.ece.2010.07.002

Zhang, D., Cheng, Q., Agterberg, F., & Chen, Z. (2016). An improved solution of local window parameters setting for local singularity analysis based on Excel VBA batch processing technology. *Computers & Geosciences*, *88*, 54–66. https://doi.org/10.1016/J.CAGEO.2015.12.012

Zhou, J., & Li, X. (2006). GeoPlot: An Excel VBA program for geochemical data plotting. *Computers and Geosciences*, *32*(4), 554–560. https://doi.org/10.1016/j.cageo.2005.07.005

ANEXOS

Anexo 1. Módulo de guardado de diseño de hoja en base de datos

```
Sub SaveForm()
   Dim PROY_COD As String
   Dim PROY_DENO As String
   Dim Form_cod As String
   Dim FORM_DENO As String
   Dim rsCon As ADODB.Recordset
   Dim num As Integer
   InitSheet
   VAR_PROJECT
   VAR_MODULE
   VAR_FORM
         PROY_COD = PROJECT.codigo
   PROY_DENO = PROJECT.Denominacion
   Form_cod = FORMULARIO.codigo
   FORM_DENO = FORMULARIO.Denominacion
   'Conetando de la base de datos
   mdlDataBase.ConnectDB
   ONCONN.rsActiveConnection rsCon
      sql = "DELETE FROM form_object WHERE modulo_cod = '$mod' AND formulario_cod ='$form'"
   sql = Replace(sql, "$mod", MODULE.codigo)
   sql = Replace(sql, "$form", FORMULARIO.codigo)
     ONCONN.Execute sql
   'Agregando informacion del objeto
   sql = " SELECT
modulo_cod,formulario_cod,formobject_cod,formobject_value,formobject_property,formobject_attribute
s,formobject_num " & _
         " FROM form_object " & _
         " WHERE modulo_cod = '$mod' AND formulario_cod ='$form'"
   sql = Replace(sql, "$proy", MODULE.codigo)
   sql = Replace(sql, "$form", FORMULARIO.codigo)
   rsCon.Open sql
           AddObject rsCon, "@ROWS", "FILAS", FORM_GetRow(), "", num: num = num + 1
      AddObject rsCon, "@COLS", "COLUMNAS", FORM_GetCol(), "", num: num = num + 1
           hMainFrm.Range(hMainFrm.Cells(FORMULARIO.fIni, FORMULARIO.cIni),
hMainFrm.Cells(FORMULARIO.fEnd, FORMULARIO.cEnd)) = "0"
           For y = FORMULARIO.fIni To FORMULARIO.fEnd
         For x = FORMULARIO.cIni To FORMULARIO.cEnd
            c_Address = hMain.Range(hMain.Cells(y, x), hMain.Cells(y, x)).Address
            If hMainFrm.Range(c_Address) = "0" Then
              If hMain.Range(c_Address) <> "" Then
                 c_AreaAddress = hMain.Range(c_Address).MergeArea.Address
                 hMainFrm.Range(c_AreaAddress) = "1"
                 FRM_OBJ = GetObject(c_Address)
                 If FRM_OBJ.Exist = True Then
                    AddObject rsCon, FRM_OBJ.m_Name, FRM_OBJ.m_Celda,
FORM_GetProperty(FRM_OBJ.m_Celda), FORM_GetAttribute(FRM_OBJ), num
                    num = num + 1
                 End If
              End If
            End If
         Next x
      Next y
      rsCon.Close
   ONCONN.rsDeleteConnection rsCon
   mdlDataBase.Disconect
```

```
    MessageBox "Se ha guardado satisfactoriamente", "SAE", MSG_ONLYOK
End Sub
```

Anexo 2. Módulo de obtención de Propiedades de Celda

```
Function FORM_GetProperty(Celda As String)
    Dim Cad As String
    Dim y As Integer
    InitSheet
        Cad = Cad & "CELL_MERGE\:" & hMain.Range(Celda).MergeArea.Address & mSeparador
    Cad = Cad & "CELL_VALUE\:" & hMain.Range(Celda) & mSeparador
    Cad = Cad & "CELL_FORMAT\:" & hMain.Range(Celda).NumberFormat & mSeparador
    Cad = Cad & "CELL_VERTICAL\:" & hMain.Range(Celda).VerticalAlignment & mSeparador
    Cad = Cad & "CELL_HORIZONTAL\:" & hMain.Range(Celda).HorizontalAlignment & mSeparador
    Cad = Cad & "CELL_BACKCOLOR\:" & hMain.Range(Celda).Interior.color & mSeparador
        If hMain.Range(Celda).Borders(xlEdgeTop).LineStyle <> xlNone Then
       Cad = Cad & "CELL_BORDER_TOP\:" & hMain.Range(Celda).Borders(xlEdgeTop).LineStyle &
mSeparador
    End If
    If hMain.Range(Celda).Borders(xlEdgeRight).LineStyle <> xlNone Then
       Cad = Cad & "CELL_BORDER_RIGHT\:" & hMain.Range(Celda).Borders(xlEdgeRight).LineStyle
& mSeparador
    End If
    If hMain.Range(Celda).Borders(xlEdgeBottom).LineStyle <> xlNone Then
       Cad = Cad & "CELL_BORDER_BOTTOM\:" &
hMain.Range(Celda).Borders(xlEdgeBottom).LineStyle & mSeparador
    End If
    If hMain.Range(Celda).Borders(xlEdgeLeft).LineStyle <> xlNone Then
       Cad = Cad & "CELL_BORDER_LEFT\:" & hMain.Range(Celda).Borders(xlEdgeLeft).LineStyle &
mSeparador
    End If
    If hMain.Range(Celda).Borders(xlEdgeLeft).LineStyle <> xlNone Then
       Cad = Cad & "CELL_BORDER_COLOR\:" & hMain.Range(Celda).Borders(xlEdgeLeft).color &
mSeparador
    End If
    Cad = Cad & "FONT_NAME\:" & hMain.Range(Celda).Font.Name & mSeparador
    Cad = Cad & "FONT_SIZE\:" & hMain.Range(Celda).Font.Size & mSeparador
    Cad = Cad & "FONT_BOLD\:" & hMain.Range(Celda).Font.Bold & mSeparador
    Cad = Cad & "FONT_ITALIC\:" & hMain.Range(Celda).Font.Italic & mSeparador
    Cad = Cad & "FONT_UNDERLINE\:" & hMain.Range(Celda).Font.Underline & mSeparador
    Cad = Cad & "FONT_FORECOLOR\:" & hMain.Range(Celda).Font.color & mSeparador
    FORM_GetProperty = Cad
End Function
```

Anexo 3. Módulo de obtención de atributos y propiedades del control

```
Function FORM_GetAttribute(obj As FRM_OBJECT)
    Dim Cad As String
    Dim y As Integer
    InitSheet
    Cad = ""
    For y = 8 To 40
       Cad = Cad & hObject.Cells(y, obj.m_ColProperty) & mSeparador
    Next y
    FORM_GetAttribute = Cad
End Function
```

Anexo 4. Command_Label

```
Function Command_Label(ModeRender As MODE_RENDER) As String
    Dim strObj As String
```

```
    If ModeRender = MODE_TABLE Then
      strObj = "<div id='$command_id' name='$command_name'
class='$command_css'>$command_value</div>"
    Else
      strObj = "<div id='$command_id' name='$command_name'
class='$command_css'>$command_value</div>"
    End If
    Command_Label = strObj
End Function
```

Anexo 5. Command_Label_Text

```
Function Command_Label_Text(ModeRender As MODE_RENDER) As String
    Dim strObj As String
    If ModeRender = MODE_TABLE Then
      strObj = "<input type='TEXT' id='$command_id' name='$command_name' class='$command_css'
value = ""$command_value"" ReadOnly/>"
    Else
      strObj = "<input type='TEXT' id='$command_id' name='$command_name' class='form-control
$command_css' value = ""$command_value"" ReadOnly/>"
    End If
    Command_Label_Text = strObj
End Function
```

Anexo 6. Command_String

```
Function Command_String(ModeRender As MODE_RENDER) As String
    Dim strObj As String
    If ModeRender = MODE_TABLE Then
      strObj = "<input type=""TEXT"" id=""$command_id"" name='$command_name' value =
'$command_value' maxlength='$command_textwidth' class='$command_css' $command_required/>"
    Else
      strObj = "<input type=""TEXT"" id=""$command_id"" name='$command_name' value =
'$command_value' maxlength='$command_textwidth' class='form-control $command_css'
$command_required/>"
    End If
    Command_String = strObj
End Function
```

Anexo 7. Command_Number_Int

```
Function Command_Number_Int(ModeRender As MODE_RENDER) As String
    Dim strObj As String
    If ModeRender = MODE_TABLE Then
      strObj = "<input type='NUMBER' id='$command_id' name='$command_name' value =
'$command_value' class='$command_css' $command_required/>"
    Else
      strObj = "<input type='NUMBER' id='$command_id' name='$command_name' value =
'$command_value' class='form-control $command_css' $command_required/>"
    End If
    Command_Number_Int = strObj
End Function
```

Anexo 8. Command_DNI

```
Function Command_DNI(ModeRender As MODE_RENDER) As String
    Dim strObj As String
    If ModeRender = MODE_TABLE Then
```

```
      strObj = "<input type='TEXT' id='$command_id' name='$command_name' value =
'$command_value' minlength='8' maxlength='8' class='$command_css' $commandrequired/>"
   Else
      strObj = "<input type='TEXT' id='$command_id' name='$command_name' value =
'$command_value' minlength='8' maxlength='8' class='form-control $command_css'
$commandrequired/>"
   End If
   Command_DNI = strObj
End Function
```

Anexo 9. Command_Date

```
Function Command_Date(ModeRender As MODE_RENDER) As String
   Dim strObj As String
   If ModeRender = MODE_TABLE Then
      strObj = "<input type='DATE' id='$command_id' name='$command_name' value =
'$command_value' minlength='10' maxlength='10' class='$command_css' $command_required/>"
   Else
      strObj = "<input type='DATE' id='$command_id' name='$command_name' value =
'$command_value' minlength='10' maxlength='10' class='form-control $command_css'
$command_required/>"
   End If
   Command_Date = strObj
End Function
```

Anexo 10. Command_Btn_Link

```
Function Command_Btn_Link(ModeRender As MODE_RENDER) As String
   Dim strObj As String
   If ModeRender = MODE_TABLE Then
      strObj = "<a href='$command_link' id='$command_id' name='$command_name' class='btn btn-
success $command_css'>$command_value</a>"
   Else
      strObj = "<a href='$command_link' id='$command_id' name='$command_name' class='btn btn-
success $command_css'>$command_value</a>"
   End If
End Function
```

Anexo 11. Command_Btn_Save

```
Function Command_Btn_Save(ModeRender As MODE_RENDER) As String

   Dim strObj As String

   If ModeRender = MODE_TABLE Then

      strObj = "<button type='BUTTON' id='$command_id' name='$command_name' class='btn btn-
primary $command_css'>$command_value</button>"

   Else

      strObj = "<button type='BUTTON' id='$command_id' name='$command_name' class='btn btn-
primary $command_css'>$command_value</button>"
```

```
    End If

    Command_Btn_Save = strObj

End Function
```

Anexo 12. Command_File

```
Function Command_File(ModeRender As MODE_RENDER) As String
    Dim strObj As String
    strObj = ""
    If ModeRender = MODE_TABLE Then
        strObj = strObj & "<div class='$command_css_box0" & "'>" & vbCrLf
        strObj = strObj & vbTab & vbTab & vbTab & vbTab & "<div class='$command_css_box1" & "'>"
& vbCrLf
        strObj = strObj & vbTab & vbTab & vbTab & vbTab & vbTab & "<input type='TEXT'
id='$command_id' name='$command_name' value = '' class='$command_css' $command_required
ReadOnly/>" & vbCrLf
        strObj = strObj & vbTab & vbTab & vbTab & vbTab & "</div>" & vbCrLf
        strObj = strObj & vbTab & vbTab & vbTab & vbTab & "<div id='$command_css_box2" & "'>" &
vbCrLf
        strObj = strObj & vbTab & vbTab & vbTab & vbTab & vbTab & "<button type='BUTTON'
id='btnFile_$command_id' name='btnFile_$command_name' class='btn btn-primary
$command_css_box2' >Subir</button>" & vbCrLf
        strObj = strObj & vbTab & vbTab & vbTab & vbTab & "</div>" & vbCrLf
        strObj = strObj & vbTab & vbTab & vbTab & "</div>"
    Else
        strObj = strObj & "<div class='$command_css_box0" & "'>" & vbCrLf
        strObj = strObj & vbTab & vbTab & vbTab & vbTab & "<div class='$command_css_box1" & "'>"
& vbCrLf
        strObj = strObj & vbTab & vbTab & vbTab & vbTab & vbTab & "<input type='TEXT'
id='$command_id' name='$command_name' value = '' class='$command_css' $command_required
ReadOnly/>" & vbCrLf
        strObj = strObj & vbTab & vbTab & vbTab & vbTab & "</div>" & vbCrLf
        strObj = strObj & vbTab & vbTab & vbTab & vbTab & "<div id='$command_css_box2" & "'>" &
vbCrLf
        strObj = strObj & vbTab & vbTab & vbTab & vbTab & vbTab & "<button type='BUTTON'
id='btnFile_$command_id' name='btnFile_$command_name' class='btn btn-primary
$command_css_box2' >Subir</button>" & vbCrLf
        strObj = strObj & vbTab & vbTab & vbTab & vbTab & "</div>" & vbCrLf
        strObj = strObj & vbTab & vbTab & vbTab & "</div>"
    End If
    Command_File = strObj
End Function
```

Anexo 13. Command_Single_Combo

```
Function Command_Single_Combo(obj As FRM_OBJECT,
ModeRender As MODE_RENDER) As String

    Fun = "Select_ListSingle($" & obj.m_Name & "[""LIST""], $" & obj.m_Name & "[""CODIGO""], $"
& obj.m_Name & "[""DENO""]);"
    strObj = ""
    strObj = strObj & "<SELECT id='$command_id_sel' name='$command_name_sel'
class='$command_css' $command_required>" & vbrclf
    strObj = strObj & "<?php " & Fun & " ?>"
    strObj = strObj & "</SELECT>"
```

```
   strObj = strObj & "<INPUT TYPE='HIDDEN' id='$command_id' name='$command_name' value='-
1'>" & vbrclf
      Command_Single_Combo = strObj
End Function
```

Anexo 14. Command_List_Table_Combo

```
Function Command_List_Table_Combo(obj As FRM_OBJECT,
ModeRender As MODE_RENDER) As String

      Fun = "Select_ListTable($" & obj.m_Name & "[""LIST""], $" & obj.m_Name & "[""CODIGO""],
$" & obj.m_Name & "[""DENO""]);"
   strObj = ""
   strObj = strObj & "<SELECT id='$command_id_sel' name='$command_name_sel'
class='$command_css' $command_required>" & vbrclf
   strObj = strObj & "<?php " & Fun & " ?>"
   strObj = strObj & "</SELECT>"
   strObj = strObj & "<INPUT TYPE='HIDDEN' id='$command_id' name='$command_name' value='-
1'>" & vbrclf

      Command_List_Table_Combo = strObj
End Function
```

Anexo 15. Command_List_Table_Search

```
Function Command_List_Table_Search(obj As FRM_OBJECT,
ModeRender As MODE_RENDER) As String
   strObj = ""
   strObj = strObj & "<div class='$command_css_box0" & "'>" & vbCrLf
   strObj = strObj & vbTab & vbTab & vbTab & vbTab & "<div class='$command_css_box1" & "'>" &
vbCrLf
   strObj = strObj & vbTab & vbTab & vbTab & vbTab & vbTab & "<input type='text'
id='$command_id' name='$command_name' value = '$command_value'
maxlength='$command_textwidth' class='$command_css' $command_required ReadOnly/>" & vbCrLf
   strObj = strObj & vbTab & vbTab & vbTab & vbTab & "</div>" & vbCrLf
   strObj = strObj & vbTab & vbTab & vbTab & vbTab & "<div id='$command_css_box2" & "'>" &
vbCrLf

   strObj = strObj & vbTab & vbTab & vbTab & vbTab & vbTab & "<button type='button'
id='$command_id_btn' name='$command_name_btn' class='btn btn-primary $command_css_box2" & "'
>...</button>" & vbCrLf
   strObj = strObj & vbTab & vbTab & vbTab & vbTab & "</div>" & vbCrLf
   strObj = strObj & vbTab & vbTab & vbTab & "</div>"

      Command_List_Table_Search = strObj
End Function
```

Anexo 16. Controller_Create

```
Function Controller_Create()
   Open FORMULARIO.File_Controller For Output As #1
      Print #1, "<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');"
      Print #1, "class " & FORMULARIO.Name_Controller & " extends CI_Controller {"
      Print #1, vbTab & "function __construct()"
      Print #1, vbTab & "{"
      Print #1, vbTab & vbTab & "parent::__construct ();"
```

```
    Print #1, vbTab & vbTab & "$this->load->model('" & MODULE.CodeSource_Name & "/" &
FORMULARIO.Name_Model & "');   // se recupera el modelo para manejar los datos"
    Print #1, vbTab & vbTab & "$this->load->helper('url');"
    Print #1, vbTab & "}" & END_FUNCTION
    Print #1, vbTab & "public function index()"
    Print #1, vbTab & "{"
    Print #1, vbTab & vbTab & "$this->load->view('web_header');"
    Print #1, vbTab & vbTab & "$this->load->view('web_footer');"
    Print #1, vbTab & "}" & END_FUNCTION
  Close #1
  Controller_Body
  Controller_Close
End Function
```

Anexo 17. Controller_Body_ShowAdd

```
Private Function Controller_Body_ShowAdd()
  Dim Cad As String
  Open FORMULARIO.File_Controller For Append As #1
    Print #1, vbTab & "public function ShowAdd()"
    Print #1, vbTab & "{"
    '$data = $this->Model_Frmregistro->ListCombo();
    Print #1, vbTab & vbTab & "$data = $this->" & FORMULARIO.Name_Model & "->ListCombo();"
    Print #1, vbTab & vbTab & "$data['FORM_TITLE'] ='" & FORMULARIO.Denominacion & "';"
    Print #1, vbTab & vbTab & "$data['FORM_NAME'] = '" & FORMULARIO.CodeSource_Name &
"';"
    Print #1, vbTab & vbTab & "$data['ACTION_TYPE'] = 'REGISTER_ADD';"
    'Print #1, vbTab & vbTab & "$this->load->view('web_header');"
    Print #1, vbTab & vbTab & "$this->load->view('frmFormHeader', $data);"
    Print #1, vbTab & vbTab & "$this->load->view('" & MODULE.CodeSource_Name & "/";
FORMULARIO.Name_View & "');"
    Print #1, vbTab & vbTab & "$this->load->view('frmFormFooter');"
    'Print #1, vbTab & vbTab & "$this->load->view('web_footer');"
    Print #1, vbTab & "}" & END_FUNCTION
  Close #1
End Function
```

Anexo 18. Controller_Body_Edit

```
Private Function Controller_Body_Edit()
  Dim Cad As String
  Open FORMULARIO.File_Controller For Append As #1
    Print #1, vbTab & "public function ShowEdit()"   'De acuerdo a la clave primaria y los parametros
```

```
        Print #1, vbTab & "{"
        Print #1, vbTab & vbTab & "$data = $this->" & FORMULARIO.Name_Model & "->ListCombo();"
        Print #1, vbTab & vbTab & "$data['FORM_TITLE'] ='" & FORMULARIO.Denominacion & "';"
        Print #1, vbTab & vbTab & "$data['FORM_NAME'] = '" & FORMULARIO.CodeSource_Name &
"';"
        Print #1, vbTab & vbTab & "$data['ACTION_TYPE'] = 'REGISTER_MODIFY';"
        Print #1, vbTab & vbTab & "$this->load->view('web_header');"
        Print #1, vbTab & vbTab & "$this->load->view('frmFormHeader', $data);"
        Print #1, vbTab & vbTab & "$this->load->view('" & MODULE.CodeSource_Name & "/";
FORMULARIO.Name_View & "');"
        Print #1, vbTab & vbTab & "$this->load->view('frmFormFooter');"
        Print #1, vbTab & vbTab & "$this->load->view('web_footer');"
        Print #1, vbTab & "}" & END_FUNCTION
    Close #1
End Function
```

Anexo 19. Controller_Body_ShowGrid

```
Private Function Controller_Body_ShowGrid()
    Dim Cad As String
    Dim rsCon As ADODB.Recordset
    Open FORMULARIO.File_Controller For Append As #1
        mdlDataBase.ConnectDB
    ONCONN.rsActiveConnection rsCon
    sql = "select * from formulario where modulo_cod = '$mod' and formulario_source = '$frm'"
    sql = Replace(sql, "$mod", MODULE.codigo)
    sql = Replace(sql, "$frm", FORMULARIO.codigo)
        rsCon.Open sql
        If rsCon.RecordCount > 0 Then
            While Not rsCon.EOF
                Debug.Print rsCon!formulario_cod, rsCon!formulario_deno
                Print #1, vbTab & "public function Show_" & Get_CodeSourceName(rsCon!formulario_cod)
& "()"
                Print #1, vbTab & "{"
                '$data = $this->Model_Frmregistro->ListCombo();
                Print #1, vbTab & vbTab & "$data['FORM_TITLE'] ='" & StrConv(rsCon!formulario_deno,
vbUpperCase) & "';"
                Print #1, vbTab & vbTab & "$data['FORM_NAME'] = '" &
Get_CodeSourceName(rsCon!formulario_cod) & "';"
                Print #1, vbTab & vbTab & "$data['ACTION_TYPE'] = ';" 'REGISTER_ADD';"
                'Print #1, vbTab & vbTab & "$this->load->view('web_header');"
```

```
            Print #1, vbTab & vbTab & "$this->load->view('frmFormHeader', $data);"
            Print #1, vbTab & vbTab & "$data['REGISTROS'] = $this->" & FORMULARIO.Name_Model
& "->" & Get_CodeSourceName(rsCon!formulario_cod) & "_GetGridRow();"
                    Print #1, vbTab & vbTab & "$this->load->view('" & MODULE.CodeSource_Name
& "/grd" & Get_CodeSourceName(rsCon!formulario_cod) & "', $data);"
            Print #1, vbTab & vbTab & "$this->load->view('frmFormFooter');"
            'Print #1, vbTab & vbTab & "$this->load->view('web_footer');"
            Print #1, vbTab & "}" & END_FUNCTION
            rsCon.MoveNext
          Wend
      End If
       rsCon.Close
   ONCONN.rsDeleteConnection rsCon
   mdlDataBase.Disconect
      Close #1
End Function
```

Anexo 20. Model_Query

```
Function Model_Query()
   Dim Cad As String
   Open FORMULARIO.File_Model For Append As #1
      Print #1, vbTab & "//Realizar consulta en base a las llamadas"
      Print #1, vbTab & "public function Query($QUERY)"
      Print #1, vbTab & "{"
      Print #1, vbTab & vbTab & "$status = true;"
      Print #1, vbTab & vbTab & "try"
      Print #1, vbTab & vbTab & "{"
      Print #1, vbTab & vbTab & vbTab & "$query = $this->db->query($QUERY);"
      Print #1, vbTab & vbTab & vbTab & "if($query->num_rows() == 0 )"
      Print #1, vbTab & vbTab & vbTab & vbTab & "$status = false;"
      Print #1, vbTab & vbTab & vbTab & "else"
      Print #1, vbTab & vbTab & vbTab & "{"
      Print #1, vbTab & vbTab & vbTab & vbTab & "$rows = $query->result();"
      Print #1, vbTab & vbTab & vbTab & vbTab & "return $rows;"
      Print #1, vbTab & vbTab & vbTab & "}"
      Print #1, vbTab & vbTab & "}"
      Print #1, vbTab & vbTab & "catch (Exception $e) {"
      Print #1, vbTab & vbTab & vbTab & "$status = false;"
      Print #1, vbTab & vbTab & "}"
      Print #1, vbTab & vbTab & "return $status;"
```

```
    Print #1, vbTab & "}" & END_FUNCTION
  Close #1
End Function
```

Anexo 21. Model_GetInfoRow

```
Function Model_GetInfoRow()
  Dim Cad As String
  Open FORMULARIO.File_Model For Append As #1
    Print #1, vbTab & "//Recupera informacion de un soslo Registro de la base de datos y si no existen
mas devuelve FALSE"
    Print #1, vbTab & "public function GetInfoRow($QUERY)"
    Print #1, vbTab & "{"
    Print #1, vbTab & vbTab & "$status = true;"
    Print #1, vbTab & vbTab & "try"
    Print #1, vbTab & vbTab & "{"
    Print #1, vbTab & vbTab & vbTab & "$query = $this->db->query($QUERY);"
    Print #1, vbTab & vbTab & vbTab & "if($query->num_rows() == 0 )"
    Print #1, vbTab & vbTab & vbTab & vbTab & "$status = false;"
    Print #1, vbTab & vbTab & vbTab & "else"
    Print #1, vbTab & vbTab & vbTab & "{"
    Print #1, vbTab & vbTab & vbTab & vbTab & "$row = $query->row();"
    Print #1, vbTab & vbTab & vbTab & vbTab & "return $row;"
    Print #1, vbTab & vbTab & vbTab & "}"
    Print #1, vbTab & vbTab & "}"
    Print #1, vbTab & vbTab & "catch (Exception $e) {"
    Print #1, vbTab & vbTab & vbTab & "$status = false;"
    Print #1, vbTab & vbTab & "}"
    Print #1, vbTab & vbTab & "return $status;"
    Print #1, vbTab & "}" & END_FUNCTION
  Close #1
End Function
```

Anexo 22. Model_Insert

```
Private Function Model_Insert()
  'Inserta variable DATA para la insercion de datos
  Dim Cad As String, CadPk As String
  Dim Tables As String
  Dim Parameters As String, mParams() As String, Param, mField As String, mValue As String
    Tables = FORM_GET_TABLES()
  Open FORMULARIO.File_Model For Append As #1
    Print #1, vbTab & "public function Insert()"
```

```
        Print #1, vbTab & "{"
        If RENDER_OBJECT_COUNT > 0 Then
            'Agregando las tablas para insertar
            Print #1, vbTab & vbTab & "//tablas en la base de datos"
            Print #1, vbTab & vbTab & "$datos['table'] = " & Chr(34) & Tables & Chr(34) & ";"
            Print #1, vbTab & vbTab & "//variables de los campos para actualizar"
            'Agregando datos de campos primarios
            CadPk = ""
            Cad = vbTab & vbTab & "$datos['fields'] = array("
            Dim SqlField As String, SqlValues As String, CadParam As String
            SqlField = ""
            SqlValues = ""
            Parameters = FORMULARIO.Parameters
            If Parameters <> "" Then
              'Debug.Print Parameters
              CadParam = ""
              mParams = Split(Parameters, ";")
              For Each Param In mParams
                n = InStr(1, Param, "=")
                If n > 0 Then
                    mField = Trim(Mid(Param, 1, n - 1))
                    mValue = Trim(Mid(Param, n + 1, Len(mField)))
                    'If StrConv(mValue, vbUpperCase) = "$_SESSION" Then mValue = "$param"
                    Print #1, vbTab & vbTab & "$" & mField & " = " & mValue & ";"
                    SqlField = SqlField & mField & ","
                    SqlValues = SqlValues & "'$" & mField & "',"
                    CadParam = CadParam & "'" & mField & "' => $" & mField & ","
                End If
              Next
            End If
            Cad = Cad & CadParam
            For i = 0 To RENDER_OBJECT_COUNT - 1
              FRM_OBJ = GetObject(RENDER_OBJECT(i).OBJECT_XLS)
              If ValideTypeInput(FRM_OBJ.m_Type) = True And FRM_OBJ.m_MySQL_field_name <> ""
Then
                Print #1, vbTab & vbTab & "$" & FRM_OBJ.m_Name & " = $this->input->post('" &
FRM_OBJ.m_Name & "');"
                Cad = Cad & "'" & FRM_OBJ.m_MySQL_field_name & "' => $" & FRM_OBJ.m_Name &
"," & vbCrLf & vbTab & vbTab & vbTab & vbTab & vbTab & vbTab
                SqlField = SqlField & FRM_OBJ.m_MySQL_field_name & ","
```

```
                SqlValues = SqlValues & "'$" & FRM_OBJ.m_Name & "',"
                        If (FRM_OBJ.m_MySQL_field_PK = True) Then
                    CadPk = CadPk & "'" & FRM_OBJ.m_MySQL_field_name & "' => $" &
FRM_OBJ.m_Name & "," & vbCrLf & vbTab & vbTab & vbTab & vbTab & vbTab & vbTab
                End If
              End If
            Next i
                    Print #1, vbTab & vbTab & "//datos para verificar campos unicos"
            If CadPk <> "" Then
              CadPk = vbTab & vbTab & "$datos['pk'] = array(" & CadPk
                        Print #1, Mid(CadPk, 1, Len(CadPk) - Len("," & vbCrLf & vbTab & vbTab & vbTab
& vbTab & vbTab & vbTab)) & ");"
            Else
              CadPk = vbTab & vbTab & "$datos['pk'] = " & Chr(34) & Chr(34) & ";"
              Print #1, CadPk
            End If
            Print #1, vbTab & vbTab & "//registro para insertar en los campos"
            Print #1, Mid(Cad, 1, Len(Cad) - Len("," & vbCrLf & vbTab & vbTab & vbTab & vbTab & vbTab
& vbTab)) & ");"
            If Len(SqlField) > 0 Then
              SqlField = Left(SqlField, Len(SqlField) - 1)
              SqlValues = Left(SqlValues, Len(SqlValues) - 1)
            End If
            CadPk = vbTab & vbTab & "$datos['sql'] = " & Chr(34) & "INSERT INTO " & Tables & "(" &
SqlField & ") VALUES (" & SqlValues & ")" & Chr(34) & ";"
                    Print #1, CadPk
          End If
          'Almacena datos en la base de datos
          Print #1, vbTab & vbTab & "//Insertar un registro en la base de datos y si es error devuelve false"
          Print #1, vbTab & vbTab & "$status = true;"
          Print #1, vbTab & vbTab & "try"
          Print #1, vbTab & vbTab & "{"
          Print #1, vbTab & vbTab & vbTab & "//$this->db->insert($REG[" & Chr(34) & "table" & Chr(34) &
"],$REG[" & Chr(34) & "fields" & Chr(34) & "]);"
          Print #1, vbTab & vbTab & vbTab & "$this->db->query($datos['sql']);"
          Print #1, vbTab & vbTab & vbTab & "$status = $this->db->insert_id();"
          Print #1, vbTab & vbTab & "}"
          Print #1, vbTab & vbTab & "catch (Exception $e) {"
          Print #1, vbTab & vbTab & vbTab & "$status = false;"
          Print #1, vbTab & vbTab & "}"
```

```
    Print #1, vbTab & vbTab & "return $status;"
    Print #1, vbTab & "}" & END_FUNCTION
  Close #1
  End Function
```

Anexo 23. Controller_Body_Update

```
Private Function Controller_Body_Update()
  'Update variable DATA para la insercion de datos
  Dim Cad As String, CadPk As String
  Dim Tables As String
  Dim Parameters As String, mParams() As String, Param, mField As String, mValue As String
    Tables = FORM_GET_TABLES()
  Open FORMULARIO.File_Controller For Append As #1
    Print #1, vbTab & "private function Update($param, $id)"
    Print #1, vbTab & "{"
    If RENDER_OBJECT_COUNT > 0 Then
      'Agregando las tablas para insertar
      Print #1, vbTab & vbTab & "//tablas en la base de datos"
      Print #1, vbTab & vbTab & "$datos['table'] = " & Chr(34) & Tables & Chr(34) & ";"
      Print #1, vbTab & vbTab & "//variables de los campos para actualizar"
      'Agregando datos de campos primarios
      CadPk = ""
      Cad = vbTab & vbTab & "$datos['fields'] = array("
      Dim SqlField As String, SqlValues As String, CadParam As String
      SqlField = ""
      SqlValues = ""
      Parameters = FORMULARIO.Parameters
      If Parameters <> "" Then
        'Debug.Print Parameters
        CadParam = ""
        mParams = Split(Parameters, ";")
        For Each Param In mParams
          n = InStr(1, Param, "=")
          If n > 0 Then
            mField = Trim(Mid(Param, 1, n - 1))
            mValue = Trim(Mid(Param, n + 1, Len(mField)))
            'If StrConv(mValue, vbUpperCase) = "$_SESSION" Then mValue = "$param"
            Print #1, vbTab & vbTab & "$" & mField & " = " & mValue & ";"
            SqlField = SqlField & mField & ","
            SqlValues = SqlValues & "'$" & mField & "',"
```

```
                CadParam = CadParam & "'" & mField & "' => $" & mField & ","
            End If
        Next
    End If
    Cad = Cad & CadParam
    For i = 0 To RENDER_OBJECT_COUNT - 1
        FRM_OBJ = GetObject(RENDER_OBJECT(i).OBJECT_XLS)
        If ValideTypeInput(FRM_OBJ.m_Type) = True And FRM_OBJ.m_MySQL_field_name <> "" Then
            Print #1, vbTab & vbTab & "$" & FRM_OBJ.m_Name & " = $this->input->post('" & FRM_OBJ.m_Name & "');"
            Cad = Cad & "'" & FRM_OBJ.m_MySQL_field_name & "' => $" & FRM_OBJ.m_Name & "," & vbCrLf & vbTab & vbTab & vbTab & vbTab & vbTab
            SqlField = SqlField & FRM_OBJ.m_MySQL_field_name & ","
            SqlValues = SqlValues & "'$" & FRM_OBJ.m_Name & "',"
                If (FRM_OBJ.m_MySQL_field_PK = True) Then
                CadPk = CadPk & "'" & FRM_OBJ.m_MySQL_field_name & "' => $" & FRM_OBJ.m_Name & "," & vbCrLf & vbTab & vbTab & vbTab & vbTab & vbTab & vbTab
            End If
        End If
    Next i
        Print #1, vbTab & vbTab & "//datos para verificar campos unicos"
    If CadPk <> "" Then
        CadPk = vbTab & vbTab & "$datos['pk'] = array(" & CadPk
            Print #1, Mid(CadPk, 1, Len(CadPk) - Len("," & vbCrLf & vbTab & vbTab & vbTab & vbTab & vbTab & vbTab)) & ");"
    Else
        CadPk = vbTab & vbTab & "$datos['pk'] = " & Chr(34) & Chr(34) & ";"
        Print #1, CadPk
    End If
    Print #1, vbTab & vbTab & "//registro para insertar en los campos"
    Print #1, Mid(Cad, 1, Len(Cad) - Len("," & vbCrLf & vbTab & vbTab & vbTab & vbTab & vbTab & vbTab)) & ");"
    SqlField = Left(SqlField, Len(SqlField) - 1)
    SqlValues = Left(SqlValues, Len(SqlValues) - 1)
        'CadPk = vbTab & vbTab & "$datos['sql'] = " & Chr(34) & "INSERT INTO " & Tables & "(" & SqlField & ") VALUES (" & SqlValues & ")" & Chr(34) & ";"
    'Print #1, CadPk
  End If
    Print #1, vbTab & vbTab & "$status = $this->" & FORMULARIO.Name_Model & "-
```

```
>Update($datos);"
      Print #1, vbTab & vbTab & "return $status;"
      Print #1, vbTab & "}"
   Close #1
   End Function
```

Anexo 24. Model_ListCombo

```
Private Function Model_ListCombo()
   Dim Cad As String
   Dim lst() As String
      Open FORMULARIO.File_Model For Append As #1
      Print #1, vbTab & "//Funcion para controlar las llamadas a las funciones para el formulario"
      Print #1, vbTab & "public function ListCombo()"
      Print #1, vbTab & "{"
      Print #1, vbTab & vbTab & "$datos[""INFO""] = ';"
      Print #1, vbTab & vbTab & "//Informacion autonoma de objetos del formulario"
         For i = 0 To RENDER_OBJECT_COUNT - 1
        FRM_OBJ = GetObject(RENDER_OBJECT(i).OBJECT_XLS)   'Obtiene el objeto a utilizar
        If ValideTypeInputListSingle(FRM_OBJ.m_Type) = True Or
ValideTypeInputListTable(FRM_OBJ.m_Type) = True Then
            Print #1, vbTab & vbTab & "//Objecto " & FRM_OBJ.m_Type & " -> " & FRM_OBJ.m_Name
& ", tipo:" & FRM_OBJ.m_LOAD_TYPE & " valores posible = " & FRM_OBJ.m_LOAD_LIST
            'Creando array para la lista
            Cad = "$datos['" & FRM_OBJ.m_Name & "']['LIST'] = array();"
            Cad = Replace(Cad, """, Chr(34))
            Print #1, vbTab & vbTab & Cad
            '*************************************************************************
**************
            'Si el combo es de tipo simple
                  If ValideTypeInputListSingle(FRM_OBJ.m_Type) = True Then
              FRM_OBJ.m_LOAD_LIST = "-1|Selccionar;" & FRM_OBJ.m_LOAD_LIST
              lst = Split(FRM_OBJ.m_LOAD_LIST, ";")
              n = 0
              Cad = "$datos['" & FRM_OBJ.m_Name & "']['TYPE'] = '" & FRM_OBJ.m_Type & "';"
              Cad = Replace(Cad, """, Chr(34))
              Print #1, vbTab & vbTab & Cad
                     Cad = "$datos['" & FRM_OBJ.m_Name & "']['CODIGO'] = '" & "Codigo" & "';"
              Cad = Replace(Cad, """, Chr(34))
              Print #1, vbTab & vbTab & Cad
                     Cad = "$datos['" & FRM_OBJ.m_Name & "']['DENO'] = '" & "Deno" & "';"
```

```
                Cad = Replace(Cad, """, Chr(34))
                Print #1, vbTab & vbTab & Cad
                Dim CodList() As String
                For Each lista In lst
                    CodList = Split(lista & "|||", "|")
                    Cad = "$datos['" & FRM_OBJ.m_Name & "']['LIST'][" & n & "] = array('Codigo'=>'" &
CodList(0) & "', 'Deno'=>'" & CodList(1) & "');"
                    Cad = Replace(Cad, """, Chr(34))
                    Print #1, vbTab & vbTab & Cad
                    n = n + 1
                Next
            End If
            If ValideTypeInputListTable(FRM_OBJ.m_Type) = True Then
                '*************************************************************************
***************
                'otras listas
                Dim sql As String
                Dim Fields() As String, Campos As String
                Dim Field
                sql = Trim(FRM_OBJ.m_LOAD_LIST)
                n = InStr(1, StrConv(sql, vbUpperCase), "FROM")
                Campos = Trim(Mid(sql, Len("SELECT") + 1, n - Len("SELECT") - 1))
                Campos = Replace(Campos, " ", "")
                Fields = Split(Campos, ",")
                Cad = "$datos['" & FRM_OBJ.m_Name & "']['TYPE'] = '" & FRM_OBJ.m_Type & "';"
                Cad = Replace(Cad, """, Chr(34))
                Print #1, vbTab & vbTab & Cad
                        Print #1, vbTab & vbTab & "$datos[" & Chr(34) & FRM_OBJ.m_Name &
Chr(34) & "][""CODIGO""] = " & Chr(34) & Fields(0) & Chr(34) & ";"
                Print #1, vbTab & vbTab & "$datos[" & Chr(34) & FRM_OBJ.m_Name & Chr(34) &
"][""DENO""] = " & Chr(34) & Fields(1) & Chr(34) & ";"
                        Cad = "$datos['" & FRM_OBJ.m_Name & "']['LIST'] = $this->" &
FORMULARIO.Name_Model & "->Query("
                Cad = Replace(Cad, """, Chr(34))
                Cad = Cad & Chr(34) & sql & Chr(34) & ");"
                Print #1, vbTab & vbTab & Cad
                    End If
        End If
    Next i
    Print #1, vbTab & vbTab & "return $datos;"
```

```
    Print #1, vbTab & "}" & END_FUNCTION
  Close #1
End Function
```

Anexo 25. Model_Table_Search

```
Private Function Model_Table_Search()
  Dim Cad As String
  Dim sql As SQL_SINTAX
  Dim Cadena As String
  Dim mFind As String, mColumn() As String, mCol, CadFind As String
  Open FORMULARIO.File_Model For Append As #1
    Print #1, vbTab & "public function TableSearch_Sql()"
    Print #1, vbTab & "{"
      For i = 0 To RENDER_OBJECT_COUNT - 1
        FRM_OBJ = GetObject(RENDER_OBJECT(i).OBJECT_XLS)
        If ValideTypeInputListModal(FRM_OBJ.m_Type) = True Then
          sql = mdlUtil.GET_SQL_SINTAX(FRM_OBJ.m_LOAD_LIST)
          mFind = StrConv(FRM_OBJ.m_LOAD_COLUMN_FIND, vbLowerCase)
                    Cadena = sql.S1_SELECT & " " & _
            sql.S2_FROM & " "
          If sql.S3_WHERE <> "" Then
            Cadena = Cadena & sql.S3_WHERE & " AND ([@BUSCAR@])"
          Else
            Cadena = Cadena & "WHERE ([@BUSCAR@]) "
          End If
          sql.S7_LIMIT = "LIMIT 0,100"
          Cadena = Cadena & sql.S4_GROUP_BY & " " & _
                    sql.S5_HAVING & " " & _
                    sql.S6_ORDER_BY & " " & _
                    sql.S7_LIMIT
                     Cadena = Replace(Cadena, " ", " ")
          Cad = Cadena
          CadFind = "concat(" & Replace(mFind, ";", ", ' ', ") & ") LIKE '%[@SEARCH]%' "
          Cad = Replace(Cad, "[@BUSCAR@]", CadFind)
          Print #1, vbTab & vbTab & "$DATA[""" & FRM_OBJ.m_ModalName & """][0] = """ & Cad
& """;"
          mColumn = Split(mFind, ";")
          n = 1
          For Each mCol In mColumn
            Cad = Cadena
```

```
                CadFind = mCol & " LIKE '%[@SEARCH]%' "
                Cad = Replace(Cad, "[@BUSCAR@]", CadFind)
                Print #1, vbTab & vbTab & "$DATA[""" & FRM_OBJ.m_ModalName & """][" & n & "]
= """ & Cad & """;"
                n = n + 1
            Next
                    End If
        Next i
        Print #1, vbTab & vbTab & "return $DATA;"
    Print #1, vbTab & "}" & END_FUNCTION
  Close #1
End Function
```

Anexo 26. Model_ShowGrid

```
Private Function Model_ShowGrid()
  Dim frmSource As String
  Dim rsCon As ADODB.Recordset
  mdlDataBase.ConnectDB
  ONCONN.rsActiveConnection rsCon
  VAR_PROJECT
  VAR_MODULE
  VAR_FORM
  sql = "select * from formulario where modulo_cod = '$mod' and formulario_source = '$frm'"
  sql = Replace(sql, "$mod", MODULE.codigo)
  sql = Replace(sql, "$frm", FORMULARIO.codigo)
    Open FORMULARIO.File_Model For Append As #1
  rsCon.Open sql
    If rsCon.RecordCount > 0 Then
      While Not rsCon.EOF
        tblCodigo = rsCon!formulario_cod
        tblDeno = rsCon!formulario_deno
        tblWidth = rsCon!formulario_width
        tblHeight = rsCon!formulario_height
        tblParameter = formulario_parameters
        tblSql = rsCon!formulario_sql
        tblFields = rsCon!formulario_fields
        tblMaxReg = rsCon!formulario_maxreg
        tblReder = rsCon!formulario_render
        'Debug.Print rsCon!formulario_cod, rsCon!formulario_deno
        'GridForm_Create MODULE.codigo, Get_CodeSourceName(rsCon!formulario_cod)
```

```
                    Print #1, vbTab & "// Recupera informacion de los registros para el grid " &
StrConv(Get_CodeSourceName(rsCon!formulario_cod), vbUpperCase)
            Print #1, vbTab & "public function " & Get_CodeSourceName(rsCon!formulario_cod) &
"_GetGridRow()"
            Print #1, vbTab & "{"
            Print #1, vbTab & vbTab; "$sql = """ & tblSql & """;"
            Print #1, vbTab & vbTab; "$consult = $this->Query($sql);"
            Print #1, vbTab & vbTab; "return $consult;"
            Print #1, vbTab & "}" & END_FUNCTION
                    rsCon.MoveNext
        Wend
    End If
  rsCon.Close
  Close #1
End Function
```

Anexo 27. GetStyleForm

```
Function GetStyleForm(frmObj As FRM_OBJECT, ModeRender As MODE_RENDER) As String
  Dim Style As STYLE_OBJECT_CSS
  Dim Cad As String
  Dim strAddress As String
  Dim RGN As Range
  Dim NewWidth As Single
  InitSheet
  strAddress = frmObj.m_Celda
  Set RGN = hMain.Range(strAddress)
  Style = frmObj.CSS_Style
  NewWidth = Style.Width
  '*********************************************************
  'Verificando si son comandos de tipo LIST
  If frmObj.m_Type = "UBIGEO" Then Cad = Cad & GetStyle_TextShowForm(frmObj): NewWidth =
NewWidth - Style.Height
  If frmObj.m_Type = "LIST_TABLE_SEARCH" Then Cad = Cad &
GetStyle_TextShowForm(frmObj): NewWidth = NewWidth - Style.Height
    '*********************************************************************
  'Inicio del estilo del objeto
  Cad = Cad & "." & frmObj.CSS_NAME & " {" & vbCrLf
  '*********************************************************************
  If mdloBJ.ValideTypeInput(frmObj.m_Type) = True Then
    Cad = Cad & vbTab & "padding-left: 5px;" & vbCrLf
```

```
    Cad = Cad & vbTab & "padding-right: 5px;" & vbCrLf
  End If
  'Manejo de ancho y alto del objeto
  If ModeRender = MODE_TABLE Then
    Cad = Cad & vbTab & "width: " & NewWidth & "px;" & vbCrLf
    Cad = Cad & vbTab & "height: " & Style.Height & "px;" & vbCrLf
  End If
     '-------------------------------------------
  'Manejo de texto
  Cad = Cad & vbTab & "font-family: " & Style.FontName & ";" & vbCrLf
  Cad = Cad & vbTab & "font-size: " & Style.FontSize & "px;" & vbCrLf
  Cad = Cad & vbTab & "font-weight: " & IIf(Style.FontBold = True, "bold;", "normal;") & vbCrLf
  Cad = Cad & vbTab & "font-style: " & IIf(Style.FontItalic = True, "italic;", "normal;") & vbCrLf
  If Style.FontUnderline = True Then
    Cad = Cad & vbTab & "text-decoration: underline;" & vbCrLf
  End If
  If Style.WrapText = False Then
    Cad = Cad & vbTab & "text-overflow: ellipsis;" & vbCrLf
    Cad = Cad & vbTab & "white-space: nowrap;" & vbCrLf
    Cad = Cad & vbTab & "overflow:hidden;" & vbCrLf
  End If
  '-------------------------------------------
  'Manejo de alineación y justificación
  If Style.Justify = True Then
    Cad = Cad & vbTab & "text-align: justify;" & vbCrLf
    Cad = Cad & vbTab & "text-justify: inter-word;" & vbCrLf
  Else
    If Style.Horizontal_Aligment = agsLeft Then Cad = Cad & vbTab & "text-align: left;" & vbCrLf
    If Style.Horizontal_Aligment = agsCenter Then Cad = Cad & vbTab & "text-align: center;" &
vbCrLf
    If Style.Horizontal_Aligment = agsRight Then Cad = Cad & vbTab & "text-align: right;" & vbCrLf
  End If
  If Style.Vertical_Aligment = agsTop Then Cad = Cad & ""
  If Style.Vertical_Aligment = agsMiddle Then
    Cad = Cad & vbTab & "display:table-cell;" & vbCrLf
    Cad = Cad & vbTab & "vertical-align: middle;" & vbCrLf
  End If
  If Style.Vertical_Aligment = agsBottom Then
    Cad = Cad & vbTab & "display:table-cell;" & vbCrLf
    Cad = Cad & vbTab & "vertical-align: bottom;" & vbCrLf
```

```
    End If
      If Style.Orientation <> 0 Then
      Cad = Cad & "-webkit-transform: rotate(" & Style.Orientation & "deg);" & vbCrLf
      Cad = Cad & "-moz-transform: rotate(" & Style.Orientation & "deg);" & vbCrLf
      Cad = Cad & "-o-transform: rotate(" & Style.Orientation & "deg);" & vbCrLf
      Cad = Cad & "writing-mode: lr-tb;" & vbCrLf
    End If
    '-------------------------------------------
    'Manejo de color
    If Style.Forecolor <> "#000000" Then
      Cad = Cad & vbTab & "color: " & Style.Forecolor & ";" & vbCrLf
    End If
    '-------------------------------------------
    If frmObj.m_Type = "LABEL_TEXT" Or frmObj.m_Enabled = False Then
      Cad = Cad & vbTab & "background-color:#F7F7F8;" & vbCrLf    ' SI EL OBJECTO ES UN
LABEL_TEXT O ES DISABLED
    Else
      If Style.BackColor <> "#FFFFFF" Then
          Cad = Cad & vbTab & "background-color: " & Style.BackColor & ";" & vbCrLf
      End If
    End If
    'Verificando estylo del borde
    If Style.Border = True Then
      Cad = Cad & vbTab & "border: 1px solid;" & vbCrLf
      Cad = Cad & vbTab & "border-color: " & Style.BorderTop.BorderColor & ";" & vbCrLf
      Cad = Cad & vbTab & "border-radius: 3px;" & vbCrLf
    End If
        '-------------------------------------------
    'Culmina el estilo del objeto
    Cad = Cad & "}" & vbCrLf
    GetStyleForm = Cad
End Function
```

Anexo 28. GetStyle_TextShowForm

```
Function GetStyle_TextShowForm(frmObj As FRM_OBJECT) As String
    Dim Cad As String
    Dim w As Single
    Dim h As Single
    w = frmObj.CSS_Style.Width
    h = frmObj.CSS_Style.Height
```

```
Cad = ""
'Cuadro principal
Cad = Cad & "." & frmObj.CSS_NAME & "_box0" & " {"
Cad = Cad & vbTab & "border-style: none; "
Cad = Cad & vbTab & "width:" & w & "px;"
Cad = Cad & vbTab & "height:" & h & "px;"
Cad = Cad & "}" & vbCrLf
'Cuadro de texto 1 (contenido del id)
Cad = Cad & "." & frmObj.CSS_NAME & "_box1" & " {"
  Cad = Cad & vbTab & "float:left;"
  Cad = Cad & vbTab & "width:" & (w - h) & "px;"
  Cad = Cad & vbTab & "height:" & h & "px;"
Cad = Cad & "}" & vbCrLf
'boton que despliega el formulario modal
Cad = Cad & "." & frmObj.CSS_NAME & "_box2" & " {"
  Cad = Cad & vbTab & "float:left;"
  Cad = Cad & vbTab & "width:" & h & "px;"
  Cad = Cad & vbTab & "height:" & h & "px;"
  Cad = Cad & vbTab & "border-radius: 0px 5px 5px 0px;"
  Cad = Cad & vbTab & "padding: 0;"
  Cad = Cad & vbTab & "margin: 0;"
Cad = Cad & "}" & vbCrLf
'*************************************************************************
'Agregando las propiedades del formulario modal
Dim mColumns() As String, Col
Dim mColProp() As String
Dim mWidth As Single
Dim mWidthModal As Single, C As Integer, hForm As Single, hBody As Single, hTable
   hBody = frmObj.m_LOAD_FORM_HEIGHT * cellHTML_Height
hTable = hBody + 24
hForm = hTable + 50
mColumns = Split(frmObj.m_LOAD_COLUMNS, ";")
C = 1
mWidth = 0
For Each Col In mColumns
  mColProp = Split(Col & "|10", "|")
  w = Val(mColProp(1)) * CellWidth 'Ancho de columna
  Cad = Cad & "." & frmObj.m_ModalName & "_col" & C & "{ width:" & w & "px; }" & vbCrLf
  mWidth = mWidth + w
  C = C + 1
```

```
    Next
    Cad = Cad & "." & frmObj.m_ModalName & "_search { width:" & (mWidth) & "px; }" & vbCrLf
    Cad = Cad & "." & frmObj.m_ModalName & "_table { width:" & (mWidth) & "px; height:" & hTable
& "px; margin:none; }" & vbCrLf
    Cad = Cad & "." & frmObj.m_ModalName & "_table thead { width:" & (mWidth) & "px; }" & vbCrLf
    Cad = Cad & "." & frmObj.m_ModalName & "_table tbody { width:" & (mWidth) & "px; height:" &
hBody & "px; }" & vbCrLf
        Cad = Cad & "." & frmObj.m_ModalName & "_Form {"
       Cad = Cad & "width:" & (mWidth + 40) & "px; "
       Cad = Cad & "height:" & hForm & "px; "
    Cad = Cad & "}" & vbCrLf
          GetStyle_TextShowForm = Cad
End Function
```

Anexo 29. Script_ButtonSave

```
Function Script_ButtonSave()
    Dim i As Integer
    Dim obj As FRM_OBJECT
    Dim strCSS_Fail As String
    strCSS_Fail = "border-danger"
       Open FORMULARIO.File_Script For Append As #1
      For i = 0 To RENDER_OBJECT_COUNT - 1
          FRM_OBJ = GetObject(RENDER_OBJECT(i).OBJECT_XLS)
          'Verifica si existen el objeto de BTN_SAVE
          If FRM_OBJ.m_Type = "BTN_SAVE" Then
             Print #1, vbTab & "$(""#" & FRM_OBJ.m_Name & """).click(function()"
             Print #1, vbTab & "{"
             Print #1, vbTab & vbTab & "var msg = """";"
             Print #1, vbTab & "//Verificando si existen datos requeridos"
             For j = 0 To RENDER_OBJECT_COUNT - 1
               obj = GetObject(RENDER_OBJECT(j).OBJECT_XLS)
               If ValideTypeInput(obj.m_Type) = True And obj.m_PROP_REQUERIDO = "SI" Then
                  If mdloBJ.ValideTypeInputListSingle(obj.m_Type) = True Then     'single combo
                     Cad = "if($('#$obj_sel').hasClass('" & strCSS_Fail & "') == true )
$('#$obj_sel').removeClass('" & strCSS_Fail & "');"
                  Else
                     If mdloBJ.ValideTypeInputListTable(obj.m_Type) = True Then     'Lista Table
                       Cad = "if($('#$obj_sel').hasClass('" & strCSS_Fail & "') == true )
$('#$obj_sel').removeClass('" & strCSS_Fail & "');"
                     Else
```

```
                    Cad = "if($('#$obj').hasClass('" & strCSS_Fail & "') == true )
$('#$obj').removeClass('" & strCSS_Fail & "');"
                End If
            End If
            Cad = Replace(Cad, "$obj", obj.m_Name)
            Print #1, vbTab & vbTab & Cad
        End If
    Next j
    Print #1, vbTab & "//indicando valores faltantes"
            For j = 0 To RENDER_OBJECT_COUNT - 1
        obj = GetObject(RENDER_OBJECT(j).OBJECT_XLS)
        If ValideTypeInput(obj.m_Type) = True And obj.m_PROP_REQUERIDO = "SI" Then
            If mdloBJ.ValideTypeInputListSingle(obj.m_Type) = True Then 'Single combo
                Cad = "if($('#$obj').val() == '' || $('#$obj').val() == '-1') { msg = 'Error';
$('#$obj_sel').addClass('" & strCSS_Fail & "'); }"
            Else
                If mdloBJ.ValideTypeInputListTable(obj.m_Type) = True Then 'Single combo
                    Cad = "if($('#$obj').val() == '' || $('#$obj').val() == '-1') { msg = 'Error';
$('#$obj_sel').addClass('" & strCSS_Fail & "'); }"
                Else
                    Cad = "if($('#$obj').val() == '' ) { msg = 'Error'; $('#$obj').addClass('" & strCSS_Fail
& "'); }"
                End If
            End If
            Cad = Replace(Cad, "$obj", obj.m_Name)
            Print #1, vbTab & vbTab & Cad
        End If
    Next j
            Print #1, vbTab & vbTab & "if(msg == 'Error') "
    Print #1, vbTab & vbTab & "{"
    Print #1, vbTab & vbTab & vbTab & "$('#" & FORMULARIO.CodeSource_Name &
"_Modal_Message_title').html('Error en valores requeridos');"
    Print #1, vbTab & vbTab & vbTab & "$('#" & FORMULARIO.CodeSource_Name &
"_Modal_Message_msg').html('Debe de completar los valores requeridos');"
    Print #1, vbTab & vbTab & vbTab & "$('#" & FORMULARIO.CodeSource_Name &
"_Modal_Message').modal();" 'alert('Debe de completar los valores requeridos');"
    Print #1, vbTab & vbTab & "}"
    Print #1, vbTab & vbTab & "else"
    Print #1, vbTab & vbTab & vbTab & "$('#" & FORMULARIO.CodeSource_Name &
"').submit();"
```

```
            Print #1, vbTab & "});"
        End If
    Next i
  Close #1
End Function
```

Anexo 30. Script_Query

```
Function Script_Query()
  Dim i As Integer
  Open FORMULARIO.File_Script For Append As #1
    Print #1, "function " & FORMULARIO.CodeSource_Name & "_Query(objModal)"
    Print #1, "{"
    Print #1, vbTab & "var info = $(""#"" + objModal +""_txt"").val();"
    Print #1, vbTab & "var tipo = $(""#"" + objModal +""_Select"").prop(""selectedIndex"");"
    Print #1, vbTab & "var met = """ & MD5(FORMULARIO.CodeSource_Name) & """;"
    Print #1, vbTab & "$.ajax"
    Print #1, vbTab & "({"
    'Print #1, vbTab & vbTab & vbTab & "url: ""<?php echo $base_url; ?>" &
FORM_PROPERTY.PROJECT_NAME & "/ajax/" & FORM_PROPERTY.FORM_NAME & ""","
    Print #1, vbTab & vbTab & vbTab & "url: ""../../" & MODULE.Name_Controller & "/Lista"", "
    Print #1, vbTab & vbTab & vbTab & "type: ""Post"","
    Print #1, vbTab & vbTab & vbTab & "dataType: ""json"","
    Print #1, vbTab & vbTab & vbTab & "data: {query: objModal, type: tipo, value: info, method: met
},"
    Print #1, vbTab & vbTab & vbTab & "beforeSend: function(){"
    Print #1, vbTab & vbTab & vbTab & vbTab & "//Precarga"
    Print #1, vbTab & vbTab & vbTab & "},"
    Print #1, vbTab & vbTab & vbTab & "success: function(response)"
    Print #1, vbTab & vbTab & vbTab & "{"
    Print #1, vbTab & vbTab & vbTab & vbTab & "var cad,i ;"
    Print #1, vbTab & vbTab & vbTab & vbTab & "cad=""<tbody>"";"
    Print #1, vbTab & vbTab & vbTab & vbTab & "for(i = 0; i< response.length; i++)"
    Print #1, vbTab & vbTab & vbTab & vbTab & "{"
    Print #1, vbTab & vbTab & vbTab & vbTab & vbTab & "var item = response[i];"
    Print #1, vbTab & vbTab & vbTab & vbTab & vbTab & "c=1"
    Print #1, vbTab & vbTab & vbTab & vbTab & vbTab & "cad = cad + ""<tr>"";"
    Print #1, vbTab & vbTab & vbTab & vbTab & vbTab & "$.each(item, function(key, value)"
    Print #1, vbTab & vbTab & vbTab & vbTab & vbTab & "{"
    Print #1, vbTab & vbTab & vbTab & vbTab & vbTab & vbTab & "cad = cad + ""<td class="" +
objModal + ""_col"" + c + """>"" + value + ""</td>"";"
```

```
        Print #1, vbTab & vbTab & vbTab & vbTab & vbTab & "c++;"
        Print #1, vbTab & vbTab & vbTab & vbTab & vbTab & "});"
        Print #1, vbTab & vbTab & vbTab & vbTab & vbTab & "cad = cad + ""</tr>"";"
        Print #1, vbTab & vbTab & vbTab & vbTab & "}"
        Print #1, vbTab & vbTab & vbTab & vbTab & "cad=cad + ""</tbody>"""
        Print #1, vbTab & vbTab & vbTab & vbTab & "$(""#"" + objModal +
""_table"").children(""tbody"").replaceWith(cad);"
        Print #1, vbTab & vbTab & vbTab & "}"
        Print #1, vbTab & "});"
        Print #1, "}"
    Close #1
End Function
```

Anexo 31. Script_AjaxUpload

```
Function Script_AjaxUpload()
    Dim i As Integer
    Open FORMULARIO.File_Script For Append As #1
        Print #1, "$('#upload').click(function()"
        Print #1, "{"
        Print #1, vbTab & "var file_data = $('#file').prop('files')[0];"
        Print #1, vbTab & "var form_data = new FormData();"
        Print #1, vbTab & "$('#msg').html(""Espere un momento mientras cargamos su archivo..."");"
        Print #1, vbTab & "$('#FileName').val = ';"
        Print #1, vbTab & "form_data.append('file', file_data);"
        Print #1, vbTab & "$.ajax({"
        Print #1, vbTab & vbTab & "url: ""../upload_file"",    // point to server-side controller method"
        Print #1, vbTab & vbTab & "dataType: 'text',       // what to expect back from the server"
        Print #1, vbTab & vbTab & "cache: false,"
        Print #1, vbTab & vbTab & "contentType: false,"
        Print #1, vbTab & vbTab & "processData: false,"
        Print #1, vbTab & vbTab & "data: form_data,"
        Print #1, vbTab & vbTab & "type: 'post',"
        Print #1, vbTab & vbTab & "success: function (response) {"
        Print #1, vbTab & vbTab & vbTab & "var cad = response;"
        Print #1, vbTab & vbTab & vbTab & "if( cad.substr(0,3) == 'OK:')"
        Print #1, vbTab & vbTab & vbTab & "{"
        Print #1, vbTab & vbTab & vbTab & vbTab & "cad = cad.substr(3, cad.length);"
        Print #1, vbTab & vbTab & vbTab & vbTab & "$('#msg').html('Su archivo se ha cargado
satisfactoriamente...<br>Archivo:' + cad); // display success response from the server"
        Print #1, vbTab & vbTab & vbTab & vbTab & "$('#FileName').val(cad); // display success response
```

```
from the server"
      Print #1, vbTab & vbTab & vbTab & "}"
      Print #1, vbTab & vbTab & vbTab & "else"
      Print #1, vbTab & vbTab & vbTab & vbTab & "$('#msg').html(response); // display success response
from the server"
      Print #1, vbTab & vbTab & "},"
      Print #1, vbTab & vbTab & "error: function (response) {"
      Print #1, vbTab & vbTab & vbTab & "$('#msg').html(response); // display error response from the
server"
      Print #1, vbTab & vbTab & "}"
      Print #1, vbTab & "});"
      Print #1, "});"
   Close #1
End Function
```

Anexo 32. Script_JQuery_TABLE_SEARCH

```
Function Script_JQuery_TABLE_SEARCH()
   Dim i As Integer
   Open FORMULARIO.File_Script For Append As #1
     For i = 0 To RENDER_OBJECT_COUNT - 1
        FRM_OBJ = GetObject(RENDER_OBJECT(i).OBJECT_XLS)
        If FRM_OBJ.m_Type = "FILE" Then
                    '*********************************************
           Print #1, vbTab & "//Control de Modal Aceptar: " & FRM_OBJ.m_Name
           Print #1, vbTab & "$(""#" & FRM_OBJ.m_ModalName & "_btnAceptar"").click(function()"
           Print #1, vbTab & "{"
           Print #1, vbTab & vbTab & "if($(""#FileName"").val() == '') "
           Print #1, vbTab & vbTab & vbTab & "alert('Error: Debe de cargar un archivo al sistema');"
           Print #1, vbTab & vbTab & "else"
           Print #1, vbTab & vbTab & "{"
           Print #1, vbTab & vbTab & vbTab & "$(""#" & FRM_OBJ.m_Name &
""").val($('#FileName').val());"
           Print #1, vbTab & vbTab & vbTab & "$(""#" & FRM_OBJ.m_ModalName &
""").modal('hide');"
           Print #1, vbTab & vbTab & "}"
           Print #1, vbTab & "});"
        End If
        If ValideTypeInputListModal(FRM_OBJ.m_Type) = True Then
           'Verifica si el objeto no contiene a nada se inicializa con el primer valor
           Print #1, vbTab &
```

```
"//*****************************************************************************"
        Print #1, vbTab & "//** " & FRM_OBJ.m_Name & " MODAL"
        Print #1, vbTab &
"//*****************************************************************************"
                Print #1, vbTab & "//Control de de visualización del objeto modal: " &
FRM_OBJ.m_Name
        Print #1, vbTab & "$(""#" & FRM_OBJ.m_Name & "_btn"").click(function()"
        Print #1, vbTab & "{"
        Print #1, vbTab & vbTab & "$(""#" & FRM_OBJ.m_ModalName &
"_Select"").prop(""selectedIndex"", 0);"
        Print #1, vbTab & vbTab & "$(""#" & FRM_OBJ.m_ModalName & "_txt"").val("""");"
        Print #1, vbTab & vbTab & FORMULARIO.CodeSource_Name & "_Query(""" &
FRM_OBJ.m_ModalName & """);"
        Print #1, vbTab & vbTab & "$(""#" & FRM_OBJ.m_ModalName & """).modal();"
        Print #1, vbTab & "});"
        '*********************************************
        Print #1, vbTab & "//Control de busqueda SELECT: " & FRM_OBJ.m_Name
        Print #1, vbTab & "$(""#" & FRM_OBJ.m_ModalName & "_Select"").change(function()"
        Print #1, vbTab & "{"
        Print #1, vbTab & vbTab & FORMULARIO.CodeSource_Name & "_Query(""" &
FRM_OBJ.m_ModalName & """);"
        Print #1, vbTab & "});"
                '*********************************************
        Print #1, vbTab & "//Control de busqueda TEXT: " & FRM_OBJ.m_Name
        Print #1, vbTab & "$(""#" & FRM_OBJ.m_ModalName & "_txt"").keydown(function()"
        Print #1, vbTab & "{"
        Print #1, vbTab & vbTab & FORMULARIO.CodeSource_Name & "_Query(""" &
FRM_OBJ.m_ModalName & """);"
        Print #1, vbTab & "});"
        '*********************************************
        Print #1, vbTab & "//Control de busqueda TEXT: " & FRM_OBJ.m_Name
        Print #1, vbTab & "$(""#" & FRM_OBJ.m_ModalName & "_txt"").change(function()"
        Print #1, vbTab & "{"
        Print #1, vbTab & vbTab & FORMULARIO.CodeSource_Name & "_Query(""" &
FRM_OBJ.m_ModalName & """);"
        Print #1, vbTab & "});"
                '*********************************************
        Print #1, vbTab & "//Control de objetos de tipo List Search: " & FRM_OBJ.m_Name
        Print #1, vbTab & "$(""#" & FRM_OBJ.m_ModalName & "_btnSearch"").click(function()"
        Print #1, vbTab & "{"
```

```
            Print #1, vbTab & vbTab & FORMULARIO.CodeSource_Name & "_Query(""" &
FRM_OBJ.m_ModalName & """);"
            Print #1, vbTab & "});"
            '*********************************************
            Dim Column() As String, Col, OBJ_RELA As FRM_OBJECT, spIndex As Integer
            Column = Split(FRM_OBJ.m_LOAD_COLUMN_FIND, ";")
                    Print #1, vbTab & "//Control de doble clic en la tabla"
            Print #1, vbTab & "$(""#" & FRM_OBJ.m_ModalName & """).on(""click"", ""tbody tr"",
function(event) {"
            'Print #1, vbTab & vbTab &
"$(this).addClass(""success"").siblings().removeClass(""success"");"
            Print #1, vbTab & vbTab & "var tbl = $(""#" & FRM_OBJ.m_ModalName & "_table tr"");"
            Print #1, vbTab & vbTab & "if (tbl.hasClass(""Sistema_Table_Selected"") == true )
tbl.removeClass(""Sistema_Table_Selected"");"
            Print #1, vbTab & vbTab & "var currentRow=$(this).closest(""tr"");"
            Print #1, vbTab & vbTab & "currentRow.addClass(""Sistema_Table_Selected"");"
            n = 1
            For Each Col In Column
              Print #1, vbTab & vbTab & "$(""#" & FRM_OBJ.m_ModalName & "_row_col" & n &
""").val(currentRow.find(""td:eq(" & (n - 1) & ")"").text());"
              n = n + 1
            Next
                    Print #1, vbTab & "});"
            Print #1, vbTab & "//Control del boton aceptar en el formulario modal"
            Print #1, vbTab & "$(""#" & FRM_OBJ.m_ModalName & "_btnAceptar"").click(function()"
            Print #1, vbTab & vbTab & "{"
            Print #1, vbTab & vbTab & vbTab & "$(""#" & FRM_OBJ.m_Name & """).val( $(""#" &
FRM_OBJ.m_ModalName & "_row_col1"").val() );"
            Dim mObj As String, value As String, Fields
                    Column = Split(FRM_OBJ.m_LOAD_FORM_VALUES, ";")
            n = 1
            For Each Col In Column
              Fields = Split(Col & "=", "=")
              mObj = Trim(Fields(0))    'Object Target
              value = Trim(Fields(1))   'Value
              OBJ_RELA = mdloBJ.GetObjectName(mObj)   'recupera propiedades del objeto
                      If OBJ_RELA.Exist = True Then
                spIndex = Split_Index(FRM_OBJ.m_LOAD_COLUMN_FIND, value)
                If spIndex >= 0 Then
                  'Debug.Print OBJ_RELA.m_Name; " "; OBJ_RELA.m_Type
```

```
                    If OBJ_RELA.m_Type = "LABEL" Then
                      Print #1, vbTab & vbTab & vbTab & "$(""#" & mObj & """).html( $(""#" &
FRM_OBJ.m_ModalName & "_row_col" & (spIndex + 1) & """).val() );"
                    Else
                      Print #1, vbTab & vbTab & vbTab & "$(""#" & mObj & """).val( $(""#" &
FRM_OBJ.m_ModalName & "_row_col" & (spIndex + 1) & """).val() );"
                    End If
                                Else
                    Debug.Print "JQUERY_TABLE_SEARCH(INDEX): " & FRM_OBJ.m_Name & " - ["
& value & "] en COLUMN_FIND no existe"
                  End If
                Else
                  Debug.Print "JQUERY_TABLE_SEARCH(OBJ): " & FRM_OBJ.m_Name & " - [" &
mObj & "] El objeto no existe"
                End If
                    Next
            Print #1, vbTab & vbTab & vbTab & "$(""#" & FRM_OBJ.m_ModalName &
""").modal(""hide"");"
            Print #1, vbTab & "});"
          End If
      Next i
    Close #1
End Function
```

Anexo 33. Script_JQuery_FILE

```
Function Script_JQuery_FILE()
    Dim i As Integer
    Open FORMULARIO.File_Script For Append As #1
      For i = 0 To RENDER_OBJECT_COUNT - 1
        FRM_OBJ = GetObject(RENDER_OBJECT(i).OBJECT_XLS)
        If FRM_OBJ.m_Type = "FILE" Then
          'Verifica si el objeto no contiene a nada se inicializa con el primer valor
          Print #1, vbTab &
"//*****************************************************************************"
          Print #1, vbTab & "//** " & FRM_OBJ.m_Name & " MODAL"
          Print #1, vbTab &
"//*****************************************************************************"
                    Print #1, vbTab & "//Control de de visualización del objeto modal: " &
FRM_OBJ.m_Name
          Print #1, vbTab & "$(""#btnFile_" & FRM_OBJ.m_Name & """).click(function()"
```

```
            Print #1, vbTab & "{"
            Print #1, vbTab & vbTab & "$(""#msg"").html('');"
            Print #1, vbTab & vbTab & "$(""#" & FRM_OBJ.m_ModalName & """).modal();"
            Print #1, vbTab & "});"
        End If
    Next i
  Close #1
End Function
```

Anexo 34. Líneas de código del Módulo mdlCommandos.bas

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
GETCODEEXIST		SUB	0	12
ANALIZECODE		SUB	0	60
EXISTFUNCTION		FUNCTION	0	9
COMMAND_REPLACE		SUB	0	43
COMMAND_LABEL		FUNCTION	0	7
COMMAND_LABEL_TEXT		FUNCTION	0	7
COMMAND_STRING		FUNCTION	0	7
COMMAND_NUMBER_INT		FUNCTION	0	7
COMMAND_EMAIL		FUNCTION	0	7
COMMAND_TELEFONO		FUNCTION	0	7
COMMAND_DNI		FUNCTION	0	7
COMMAND_DATE		FUNCTION	0	7
COMMAND_IMAGE		FUNCTION	0	7
COMMAND_BTN_LINK		FUNCTION	0	6
COMMAND_BTN_SAVE		FUNCTION	0	7

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
COMMAND_BTN_CANCEL		FUNCTION	0	7
COMMAND_FILE		FUNCTION	0	18
COMMAND_UBIGEO		FUNCTION	0	22
COMMAND_SINGLE_COMBO		FUNCTION	1	7
COMMAND_LIST_TABLE_COMBO		FUNCTION	1	7
COMMAND_LIST_TABLE_SEARCH		FUNCTION	1	10

Nota. Este módulo es el encargado de generar código PHP de vista del modelo MVC,

Anexo 35. Líneas de código del Módulo mdlCompact.bas

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
COMPACT		SUB	0	38

Anexo 36. Líneas de código del Módulo mdlCompile.bas

Procedimiento	Ámbito	Tipo	Número de parámetros	Número de líneas
COMPILE_FORM		SUB	0	75

Nota. Módulo encargado de realizar la compilación del formulario, el cual realiza a llamadas a diferentes módulos de compilación

Anexo 37. Líneas de código del Módulo mdlCompile_Controller.bas

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
CONTROLLER_CREATE		FUNCTION	0	17
CONTROLLER_BODY	PRIVATE	FUNCTION	0	4
CONTROLLER_BODY_SHOWADD	PRIVATE	FUNCTION	0	13
CONTROLLER_BODY_EDIT	PRIVATE	FUNCTION	0	15
CONTROLLER_BODY_SHOWGRID	PRIVATE	FUNCTION	0	29
CONTROLLER_BODY_SHOWADDAJAX	PRIVATE	FUNCTION	0	13
CONTROLLER_BODY_GETINFO	PRIVATE	FUNCTION	0	19
CONTROLLER_BODY_DELETE	PRIVATE	FUNCTION	0	48
CONTROLLER_CLOSE	PRIVATE	FUNCTION	0	3

Anexo 38. Líneas de código del Módulo mdlCompile_css.bas

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
CSS_CREATESTYLE		SUB	0	21
STYLE_FILES_TABLE		SUB	0	40
STYLE_ROWS		FUNCTION	0	8
STYLE_COLS		FUNCTION	0	8
GETSTYLEFORM		FUNCTION	1	90

GETSTYLE_TEXTSHOWFORM	FUNCTION	0	51
STYLE_SCROLL	SUB	0	23
GETNAMESTYLE	FUNCTION	0	1
GETROWSTYLE	FUNCTION	0	1
GETCOLSTYLE	FUNCTION	0	1

Anexo 39. Líneas de código del Módulo mdlCompile_gridView.bas

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
COMPILE_GRIDVIEW_INIT		SUB	0	30
GRIDFORM_CREATE	PRIVATE	SUB	1	58
GRIDFORM_HEADER	PRIVATE	SUB	0	27
GRIDFORM_BODY	PRIVATE	SUB	0	46
GRIDFORM_FOOTER	PRIVATE	SUB	0	3
GRIDFORM_CLOSE	PRIVATE	SUB	0	2
LOADG		SUB	0	30
GRIDFORM_ADDMODALNEW		SUB	0	30

Anexo 40. Líneas de código del Módulo mdlCompile_MasterController.bas

Procedimiento	Tipo	Número de Parámetros	Número de Líneas
CONTROLLERMASTER_CREATE	FUNCTION	0	19
CONTROLLERMASTER_UPLOADFILE	FUNCTION	0	67
CONTROLLERMASTER_REGISTER	FUNCTION	0	11
CONTROLLERMASTER_GETMODULE	FUNCTION	0	12
CONTROLLERMASTER_BODY	FUNCTION	0	23
CONTROLLERMASTER_DISPCOMP	FUNCTION	0	31
CONTROLLERMASTER_DISPLAY	FUNCTION	0	60
CONTROLLERMASTER_LISTA	FUNCTION	0	31
CONTROLLERMASTER_SHOWFORM	FUNCTION	0	10
CONTROLLERMASTER_GETGRIDVIEW	FUNCTION	0	31
CONTROLLERMASTER_EXEVIEW	FUNCTION	0	76
CONTROLLERMASTER_CLOSE	FUNCTION	0	7
CONTROLLERMASTER_SENDMAIL	SUB	0	52
MODULEMASTER_CREATE	FUNCTION	0	9
MODULEMASTER_BODY	FUNCTION	0	4
MODULEMASTER_MODULEGETSQL	FUNCTION	0	22
MODULEMASTER_MODULEGETSQL_NUMROWS	FUNCTION	0	21
MODULEMASTER_MODULEGRID	FUNCTION	0	25
MODULEMASTER_MODULEVIEWBST_PDF	FUNCTION	0	25

Procedimiento	Tipo	Número de Parámetros	Número de Líneas
MODULEMASTER_CLOSE	FUNCTION	0	3

Anexo 41. Líneas de código del Módulo mdlCompile_Model.bas

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
MODEL_CREATE		FUNCTION	0	9
MODEL_BODY		FUNCTION	0	10
MODEL_QUERY		FUNCTION	0	23
MODEL_GETINFOROW		FUNCTION	0	23
MODEL_GETINFOALL		FUNCTION	0	7
MODEL_GETINFORANGE		FUNCTION	0	7
MODEL_INSERT	PRIVATE	FUNCTION	0	130
MODEL_UPDATE		FUNCTION	0	16
MODEL_DELETE		FUNCTION	0	16
MODEL_LISTCOMBO	PRIVATE	FUNCTION	0	61
MODEL_TABLE_SEARCH	PRIVATE	FUNCTION	0	43
MODEL_SHOWGRID	PRIVATE	FUNCTION	0	35
CONTROLLER_BODY_UPDATE	PRIVATE	FUNCTION	0	62
MODEL_CLOSE		FUNCTION	0	3

Anexo 42. Líneas de código del Módulo mdlCompile_root.bas

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
ROOT_CREATE_CONTROLLER_INIT	PRIVATE	SUB	0	18
ROOT_CREATE_CONTROLLER_INDEX	PRIVATE	SUB	0	5
ROOT_CREATE_MENU_PHP	PRIVATE	SUB	0	34
ROOT_CREATE_MENU_JS	PRIVATE	SUB	0	8
ROOT_CREATE_MENU_CSS	PRIVATE	SUB	0	91

Anexo 43. Líneas de código del Módulo mdlCompile_Script.bas

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
SCRIPT_CREATE		FUNCTION	0	10
SCRIPT_HEADER		FUNCTION	0	16
SCRIPT_JQUERY_INI		FUNCTION	0	4
SCRIPT_JQUERY_LIST		FUNCTION	0	17
SCRIPT_FILE		FUNCTION	0	35
SCRIPT_BUTTONSAVE		FUNCTION	0	68
SCRIPT_QUERY		FUNCTION	0	38
SCRIPT_AJAXUPLOAD_ANTERIOR		FUNCTION	0	34
SCRIPT_JQUERY_TABLE_SEARCH		FUNCTION	0	95

SCRIPT_JQUERY_FILE	FUNCTION	0	17
SCRIPT_JQUERY_CLOSE	FUNCTION	0	3
SCRIPT_CLOSE	FUNCTION	0	2
SCRIPT_COMMANDUPLOAD	SUB	1	94

Anexo 44. Líneas de código del Módulo mdlCompile_Transac.bas

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
GENERATETRANSACTION		SUB	0	11
GENERATETABLETRANSACTION		SUB	0	16
GENERATELIST		SUB	0	62

Anexo 45. Líneas de código del Módulo mdlCompile_ViewForm.bas

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
VIEW_CREATE_FORM		SUB	0	17
VIEW_INIT		SUB	0	21
CREATEPLANTILLABOOTSTRAP		SUB	0	26
BOOTSTRAPINIT		SUB	0	10
BOOTSTRAPEND		SUB	0	3
GETCELL		FUNCTION	1	1

ROW_CREATE	SUB	2	8
ROW_END	SUB	1	8
COL_CREATE	SUB	4	24
GETCOLUMNWIDTHPIXEL	FUNCTION	0	2
GETROWHEIGHTPIXEL	FUNCTION	0	1
VIEW_FUNCTIONLIST	SUB	0	15
VIEW_MODAL	SUB	0	100
VIEW_MODAL_FAIL	SUB	0	24
VIWE_LIST_FILES	SUB	0	10

Anexo 46. Líneas de código del Módulo mdlDataBase.bas

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
INITPARAMETERDB		SUB	0	13
CONNECTDB		SUB	0	9
DISCONECT		SUB	0	1
ADDOBJECT		SUB	5	9
FORM_GETROW		FUNCTION	0	8
FORM_GETCOL		FUNCTION	0	8
FORM_GETATTRIBUTE		FUNCTION	0	8
FORM_GETPROPERTY		FUNCTION	0	31
LOADFORM		SUB	0	44

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
FORM_SETROWS		SUB	1	7
FORM_SETCOLS		SUB	1	7
FORM_OBJECT		SUB	6	61
LOADFORMSTITLESTRING		FUNCTION	0	16
LOADEMAIL		FUNCTION	0	15
LOADFORMSACTION		FUNCTION	1	17
CREATESQL		FUNCTION	0	38
GETFORMNULARIOVALUE		FUNCTION	1	16
GETFORMNULARIOTABLE		FUNCTION	0	30
CLIENTCONNECT		FUNCTION	0	15
CLIENTDISCONECT		FUNCTION	0	1
TABLE_EXISTVALUE		FUNCTION	2	17
TABLE_EXIST		FUNCTION	1	10
LOAD_LISTDB		SUB	2	35

Anexo 47. Líneas de código del Módulo mdlFormulario.bas

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
LOAD_FORMPROPERTY		SUB	0	37

Anexo 48. Líneas de código del Módulo mdlIni.bas

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
WRITEINI		FUNCTION	3	9
GETINI		FUNCTION	2	5
GETPATHCONFIG		FUNCTION	0	4

Anexo 49. Líneas de código del Módulo mdlMain.bas

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
CONNECT_DB_FORM		SUB	0	2
ONLOAD		SUB	0	17
PROYECT_SELECT		SUB	0	1
PROYECT_NEW		SUB	0	1
PROYECT_LIST		SUB	0	1
FORM_SELECT		SUB	0	1
SAVEFORM		SUB	0	49
GETSTRINGFORMACTION		FUNCTION	0	6
LOAD_INI		SUB	0	14

Anexo 50. Líneas de código del Módulo mdlMD5.bas

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
INIMD5	PRIVATE	SUB	0	62
LSHIFT	PRIVATE	FUNCTION	1	18
RSHIFT	PRIVATE	FUNCTION	1	17
ROTATELEFT	PRIVATE	FUNCTION	1	1
ADDUNSIGNED	PRIVATE	FUNCTION	1	22
F	PRIVATE	FUNCTION	2	1
G	PRIVATE	FUNCTION	2	1
H	PRIVATE	FUNCTION	2	1
I	PRIVATE	FUNCTION	2	1
FF	PRIVATE	SUB	6	3
GG	PRIVATE	SUB	6	3
HH	PRIVATE	SUB	6	3
II	PRIVATE	SUB	6	3
CONVERTTOWORDARRAY	PRIVATE	FUNCTION	0	36
WORDTOHEX	PRIVATE	FUNCTION	0	6
MD5	PUBLIC	FUNCTION	0	107

Anexo 51. Líneas de código del Módulo mdlObj.bas

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
GETPROPERTIES		SUB	0	13
GETPROPERTYROWINI		FUNCTION	0	2
GETPROPERTYROWMAX		FUNCTION	0	2
GETPOSOBJETO		FUNCTION	0	9
SETDEFAULTOBJETO		FUNCTION	1	20
SETVALUEPROPERTY		FUNCTION	2	11
GETMAXOBJETO		FUNCTION	0	8
GETCELLWIDTH		FUNCTION	0	2
GETCELLHEIGHT		FUNCTION	0	2
GETROWPROPERTY		FUNCTION	0	4
GETFULLOBJECT		SUB	0	25
GETPROPERTYROW		FUNCTION	0	4
PROPERTY_GETVALUE		FUNCTION	0	11
PROPERTY_SETVALUE		FUNCTION	2	18
PROPERTY_EXIST		FUNCTION	0	5
PROPERTY_SET		SUB	0	50
GETOBJECTNAME		FUNCTION	0	17
GETOBJECT		FUNCTION	0	66
GETOBJADDRESS		FUNCTION	1	7

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
SETSTYLE_PROPERTY		FUNCTION	1	4
SETFIELDOBJECT	PUBLIC	SUB	0	92
SETGRIDVISTA		SUB	0	18
GETOBJECTSTYLE		FUNCTION	0	47
VALIDETYPEINPUT		FUNCTION	0	8
VALIDETYPEINPUTLIST		FUNCTION	0	8
VALIDETYPEINPUTLISTSINGLE		FUNCTION	0	8
VALIDETYPEINPUTLISTTABLE		FUNCTION	0	8
VALIDETYPEINPUTLISTMODAL		FUNCTION	0	8
LIST_FORM_LOAD		FUNCTION	0	22

Anexo 52. Líneas de código del Módulo mdlParse.bas

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
ANALIZEPARSE		FUNCTION	0	41

Anexo 53. Líneas de código del Módulo mdlRibbon.bas

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
RIBBONONLOAD		SUB	0	2
GETENABLEDMACRO		SUB	1	9
REFRESHRIBBON		SUB	0	6
RIBBON_CONNECT_OPEN		SUB	0	1
RIBBON_SAVEFORM		SUB	0	1
RIBBON_PROYECTO		SUB	0	1
RIBBON_MODULO		SUB	0	6
RIBBON_FORMULARIO		SUB	0	6
RIBBON_FORM_NEW_FRM		SUB	0	13
RIBBON_FORM_NEW_VIEW		SUB	0	2
RIBBON_FORM_NEW_GRID		SUB	0	13
RIBBON_EDITFORM		SUB	0	38
RIBBON_EDITFORM_TO_GRID		SUB	0	15
RIBBON_EDITGRID_TO_FORM		SUB	0	15
RIBBON_COMPILEFORM		SUB	0	1
RIBBON_COMPILEALL		SUB	0	1
RIBBON_DATA_FIELDOBJECT		SUB	0	1

Anexo 54. Líneas de código del Módulo mdlSound.bas

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
EJECTSOUND		SUB	0	6
BUTTON_CLICK		SUB	0	7
SNDERROR		SUB	0	7
SNDACEPTAR		SUB	0	7
SNDINFORMACION		SUB	0	7
SNDPREGUNTA		SUB	0	7
SNDVIEWMSGBOX		SUB	0	7

Anexo 55. Líneas de código del Módulo mdlUtil.bas

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
GETSHOWVALUE		FUNCTION	0	4
GETROMANOS		FUNCTION	0	40
MESSAGEBOX		FUNCTION	2	26
GETMAXROWS		FUNCTION	3	8
GETEXISTVALUE		FUNCTION	4	9
GETPOSVALUE		FUNCTION	4	9
GETSTRCOLUMN		FUNCTION	0	6

Procedimiento	Ámbito	Tipo	Número de Parámetros	Número de Líneas
GETINTCOLUMN		FUNCTION	0	9
FILEEXIST		FUNCTION	0	5
DIREXIST		FUNCTION	0	9
LONGTORGB		FUNCTION	0	8
GETWIDTHAREA		FUNCTION	1	8
GETHEIGHTAREA		FUNCTION	1	8
REPLACECHARTOHTML		FUNCTION	0	14
GET_SQL_SINTAX		FUNCTION	0	60
SPLIT_INDEX		FUNCTION	1	13
CREATEDIRECTORY		FUNCTION	0	19
GET_CODESOURCENAME		FUNCTION	0	13
MESSAGELIST		FUNCTION	3	12
VIRIFY_ROOT_PATH		FUNCTION	0	9
GETADDRESS_XY		FUNCTION	0	28
GETPROPERTYSTRING		FUNCTION	0	58

Universidad Nacional del Altiplano Puno | Vicerrectorado de Investigación | Repositorio Institucional

DECLARACIÓN JURADA DE AUTENTICIDAD DE TESIS

Por el presente documento, Yo Amilkar Sucasaca Paconi,
identificado con DNI 42690094 en mi condición de egresado de:

☐ **Escuela Profesional,** ☐ **Programa de Segunda Especialidad,** ☒ **Programa de Maestría o Doctorado**

Maestria en Informatica,

informo que he elaborado el/la ☒ **Tesis** o ☐ **Trabajo de Investigación** denominada:
"AUTOMATIZACIÓN DE DESARROLLO DE APLICACIONES WEB, BASADO EN TECNOLOGÍA DE FRAMEWORKS Y MACROS"

Es un tema original.

Declaro que el presente trabajo de tesis es elaborado por mi persona y **no existe plagio/copia** de ninguna naturaleza, en especial de otro documento de investigación (tesis, revista, texto, congreso, o similar) presentado por persona natural o jurídica alguna ante instituciones académicas, profesionales, de investigación o similares, en el país o en el extranjero.

Dejo constancia que las citas de otros autores han sido debidamente identificadas en el trabajo de investigación, por lo que no asumiré como suyas las opiniones vertidas por terceros, ya sea de fuentes encontradas en medios escritos, digitales o Internet.

Asimismo, ratifico que soy plenamente consciente de todo el contenido de la tesis y asumo la responsabilidad de cualquier error u omisión en el documento, así como de las connotaciones éticas y legales involucradas.

En caso de incumplimiento de esta declaración, me someto a las disposiciones legales vigentes y a las sanciones correspondientes de igual forma me someto a las sanciones establecidas en las Directivas y otras normas internas, así como las que me alcancen del Código Civil y Normas Legales conexas por el incumplimiento del presente compromiso

Puno 27 de DICIEMBRE del 2023

FIRMA (obligatoria)

Huella

Universidad Nacional del Altiplano Puno | Vicerrectorado de Investigación | Repositorio Institucional

AUTORIZACIÓN PARA EL DEPÓSITO DE TESIS O TRABAJO DE INVESTIGACIÓN EN EL REPOSITORIO INSTITUCIONAL

Por el presente documento, Yo Amilkar Sucasaca Pacori,
identificado con DNI 42690094 en mi condición de egresado de:

☐ **Escuela Profesional,** ☐ **Programa de Segunda Especialidad,** ☒ **Programa de Maestría o Doctorado**

Maestria en Informatica,

informo que he elaborado el/la ☒ **Tesis** o ☐ **Trabajo de Investigación** denominada:

"AUTOMATIZACIÓN DE DESARROLLO DE APLICACIONES WEB, BASADO EN TECNOLOGÍA DE FRAMEWORK Y MACROS"

para la obtención de ☒**Grado,** ☐ **Título Profesional** o ☐ **Segunda Especialidad.**

Por medio del presente documento, afirmo y garantizo ser el legítimo, único y exclusivo titular de todos los derechos de propiedad intelectual sobre los documentos arriba mencionados, las obras, los contenidos, los productos y/o las creaciones en general (en adelante, los "Contenidos") que serán incluidos en el repositorio institucional de la Universidad Nacional del Altiplano de Puno.

También, doy seguridad de que los contenidos entregados se encuentran libres de toda contraseña, restricción o medida tecnológica de protección, con la finalidad de permitir que se puedan leer, descargar, reproducir, distribuir, imprimir, buscar y enlazar los textos completos, sin limitación alguna.

Autorizo a la Universidad Nacional del Altiplano de Puno a publicar los Contenidos en el Repositorio Institucional y, en consecuencia, en el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto, sobre la base de lo establecido en la Ley N° 30035, sus normas reglamentarias, modificatorias, sustitutorias y conexas, y de acuerdo con las políticas de acceso abierto que la Universidad aplique en relación con sus Repositorios Institucionales. Autorizo expresamente toda consulta y uso de los Contenidos, por parte de cualquier persona, por el tiempo de duración de los derechos patrimoniales de autor y derechos conexos, a título gratuito y a nivel mundial.

En consecuencia, la Universidad tendrá la posibilidad de divulgar y difundir los Contenidos, de manera total o parcial, sin limitación alguna y sin derecho a pago de contraprestación, remuneración ni regalía alguna a favor mío; en los medios, canales y plataformas que la Universidad y/o el Estado de la República del Perú determinen, a nivel mundial, sin restricción geográfica alguna y de manera indefinida, pudiendo crear y/o extraer los metadatos sobre los Contenidos, e incluir los Contenidos en los índices y buscadores que estimen necesarios para promover su difusión.

Autorizo que los Contenidos sean puestos a disposición del público a través de la siguiente licencia:

Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional. Para ver una copia de esta licencia, visita: https://creativecommons.org/licenses/by-nc-sa/4.0/

En señal de conformidad, suscribo el presente documento.

Puno 27 de DICIEMBRE del 2023

FIRMA (obligatoria)

Huella