

Introducción

La documentación técnica es muy importante en el desarrollo de software. Es como una carta de navegación para tu equipo. Documentar tu proceso sirve como referencia explicando las razones del desarrollo, como opera y cómo utilizarlo. Los equipos de software se refieren a este proceso de documentación técnica cuando hablan de requerimientos, notas en la release o diferentes aspectos en el desarrollo del producto. Utilizan documentos para detallar el código, APIs base de datos, y realizar un seguimiento en el proceso de desarrollo de software. Externamente, la documentación se plasma en manuales, guías de usuario para administradores de sistemas, ayuda a los equipos y demás usos.

La documentación técnica de un proyecto de desarrollo de software no debe omitir la información de la base de datos. Es importante conocer con qué tipo de base de datos estamos trabajando, servidores, diagrama de información, estructura de tablas, así como los tipos de datos utilizados, etc.

Para un programador, la **documentación técnica es un deber**. La tarea no se basa en hacerlo o no, sino en cómo y qué herramientas sirven para hacer el proceso más eficiente. El hecho de documentar el proceso sirve para conocer todos los aspectos de una aplicación, cosa que beneficia al equipo y mejora la calidad del producto de software. Con la documentación técnica, tienes toda la información necesaria para el desarrollo y el mantenimiento correcto de la aplicación. Así como una mejor transferencia de conocimiento entre developers. Incluso el mejor software, puede ser inútil si los desarrolladores son incapaces de entenderlo.

Una buena documentación técnica con las mejores herramientas hace que la información sea más accesible, ofreciendo un gran número de “entry points”, ayuda a los nuevos desarrolladores a aprender más rápido, simplifica el producto y ayuda en la documentación de costes. Además, genera confianza.

Por lo tanto, la documentación técnica sirve para ayudar a los nuevos miembros del equipo a adaptarse más rápido a los hábitos de trabajo

DIRECCIÓN GENERAL DE INNOVACIÓN - UNIDAD DE INFORMÁTICA

Guía para la Documentación técnica de un proyecto de software

de la organización. Comparte información del funcionamiento del producto y el porqué de cada requerimiento. Hace que la curva de aprendizaje de los desarrolladores sea más suave. Y lo hace señalando aquellos puntos de la aplicación en los que ha de centrarse el desarrollador para saber más del contexto de aquella aplicación en la que están trabajando.

Por lo anterior, a continuación, se especifican los puntos necesarios que debe cumplir la documentación técnica de cada proyecto elaborado en la Unidad de Informática.

I.- Definiciones

1) Definición general del proyecto de software

“Buenas Prácticas” es un sistema orientado a la administración de la Dirección General de Innovación del Estado de México, ayudando a el manejo de documentos de dependencias y de sus distintos organismos centrales o auxiliares.

El principal objetivo es ayudar a el manejo y archivo de documentos internos que le corresponden a cada dependencia, facilitando el manejo y consulta de la información ayudando a generar datos históricos de cada uno de los archivos que se van generando dependiendo de cada actualización de los órganos internos.

Esta orientando a personal administrativo dentro de la Dirección General de Innovación del Estado de México, además que personas publicas pueden consultar la información limitando las partes del sistema administrativas a personal interno o que va dirigir la administración y actualización de los datos.

2) Especificación de requerimientos del proyecto

Requerimientos técnicos:

- Buena estructura del sistema
- Diseño adaptativo
- Gestión por parte de los usuarios
- Actualizaciones constantes
- Bases de datos estables (MYSQL)
- Sistema estable basado en PHP

Alcances:

- Las estrategias de desarrollo este trabajo, estarán aplicadas para generar un gran alcance a corto y largo plazo.
- Este proyecto se enfocará directamente al segmento administrativo de información interna de dependencias y organizaciones administrativas del Estado de México.

DIRECCIÓN GENERAL DE INNOVACIÓN - UNIDAD DE INFORMÁTICA
Guía para la Documentación técnica de un proyecto de software

- Se busca resolver las necesidades de la administración de documentos; ofreciendo un sistema estable y sencillo de utilizar para ahorrar tiempos.

Limitaciones:

- Este proyecto se limitará a solo dependencias y organizaciones del Estado de México, y estará sujeto a actualizaciones del mismo.
- El manejo del sistema estará basado en una o mas personas que realizaran cambios una vez las dependencias así lo soliciten.

Este sistema forma parte de distintos entornos administrativos, pero fue desarrollado desde cero y su principal objetivo es implementar un medio mas al ya existente entorno de administración que existe ya.

Requisitos Generales:

Lenguaje de programación y almacenamiento de datos (PHP - MYSQL).

Seguridad del sistema (Sesiones)

Alojamiento del sistema (Servidores internos)

Entorno de desarrollo (PHP 7.4 – MYSQL 8.0 - APACHE 2.4.38)

Requisitos Funcionales:

- Página principal del sistema Buenas Prácticas, funcionando como apartado inicial del sistema y vinculando la seguridad del sistema por sesiones y accesos por nivel (USUARIO, ADMINISTRADOR O SUPER ADMINISTRADOR).
- Registro de usuarios dependiendo los niveles de acceso que se otorgaran al sistema.

- Listado de dependencias dividido por paneles independientes correspondientes a cada dependencia.
- Lectura de datos por medio de claves privadas correspondientes a cada dependencia para relacionar sus organismos correspondientes para así mostrar sus datos correctos.
- Apartado de detalles en donde se muestran los principales aspectos de cada dependencia además de contar con botones de edición directa sin refrescar pagina y mandar a histórico, solo habilitados para administradores o super administrador.
- Gestión de datos, edición y eliminación (edición - baja lógica), directa al mandar a histórico cada apartado.

El proyecto esta sujeto a nuevas actualizaciones futuras o retrocompatibilidad con API'S

3) Procedimiento de instalación y prueba:

Herramientas utilizadas:

- Entorno de desarrollo (Docker – LAMPP, LARAGON, XAMPP)
- Sistemas operativos (Linux o Windows)
- Bases de datos (MYSQL)
- PHP (7.4)
- Composer

Planificación: Metodología Scrum, para soluciones rápidas a problemas visibles, además de la resolución de este sistema se sometió a valores de “Prueba y error” constantes durante la creación del sistema para que la creación vaya siendo constante y rápida.

De la instalación y prueba:

Requisitos no funcionales:

- Algunos estilos y fuentes (CSS de la plantilla base).

Obtención e instalación:

- 1- Instalación: La instalación se debe hacer en un entorno que soporte versiones PHP, en este caso 7.4 además de contar con el manejo de gestor de datos MYSQL.
- 2- Accesos: Los accesos se otorgan con base a el acceso del super administrador con una clave y contraseña ya establecida
- 3- Accesos internos: Los accesos se hacen por medio de datos que el administrador crea o accesos públicos a paginas públicas.
- 4- El manejo de datos solo se hace por medio de administradores.

Especificaciones de prueba y ejecución:

Entornos de desarrollo: PHP Y MYSQL (DOCKER LAMPP, XAMPP O LARAFON)

2.-Arquitectura del Sistema

- Descripción jerárquica: Los archivos se dividen por secciones, de diseño, configuración de conexiones, recursos, carpetas de archivos de almacenaje y archivos principales de función de sistema.
- Diagrama de módulos: Consiste en un diagrama donde se representan todas las partes que componen el sistema y las relaciones que existen entre estas. El objetivo de este diagrama consiste en presentar una perspectiva global de la arquitectura y los componentes del sistema, no debería contener detalles técnicos sobre los módulos o las conexiones entre estos.
- Descripción individual de los módulos:

INDEX: Este modulo consta de la pagina principal en donde se ingresa al sistema.

ADD_ADMIN: Este módulo está habilitada solo para “Super Administrador”, ya que el puede dar de alta los accesos a administradores o usuarios.

SECURITY: Este modulo consta con la seguridad del sistema dividida por sesiones

DBCONFIG: Contiene los archivos de conexión a base de datos (PDO Y MYSQL)

CSS: Contiene los archivos necesarios para el diseño y manejo del sistema.

EXCEL: Carpeta que contiene los archivos que el sistema va generando en el formulario

PDF: Carpeta que contiene los documentos que el sistema va generando con la administración.

VISIO: Carpeta que contiene los documentos que el sistema va generando con la administración.

RESOURCES: Carpeta que contiene archivos necesarios para la administración y el correcto funcionamiento del sistema.

SISTEMA: Contiene los principales archivos para la gestión y los archivos más complejos (Requiere de DBCONFIG).

Dependencias externas: El sistema requiere de dependencias externas tales como

- BOOTSTRAP
- COMPOSER
- PHPOFFICE
- SYMFONY
- JAVASCRIPT (LIBRERIAS) V2.3 O POSTERIOR
- SPREADHSEETREADER
- MYSQL (PDO O MYSQLI)

3.-Diseño del modelo de base de datos.

- Datos de entrada.

tbl_ciditest
id_reg (INT PRIMARY KEY 10)
cod_ant (VARCHAR 20)
cod_nue (VARCHAR 20)
cod_sim (VARCHAR 15)
uniadm (VARCHAR 300)
tipocambio (VARCHAR 200)

tbl_dependencias
id_reg (INT PRIMARY KEY 10)
cvedep (VARCHAR 15)
nomdep (VARCHAR 20)
vigente (VARCHAR 15)

tbl_testuadm3
id (INT PRIMARY KEY 10)
cveuni (VARCHAR 15)
nombre (VARCHAR 200)
cvepadre (VARCHAR 15)
estatus (INT 10)
órgano (VARCHAR 10)

DIRECCIÓN GENERAL DE INNOVACIÓN - UNIDAD DE INFORMÁTICA
Guía para la Documentación técnica de un proyecto de software

Modelo relacional: Se aplica directamente desde el código y se relaciona:

tbl_dependencias con **tbl_testuadm3** para obtener las dependencias y sus organismos correspondientes en base a:

cvedep=cvepadre

tbl_organigrama, se relaciona con **tbl_testuadm** para guardar datos:

tbl_organigrama.cveuni = tbl_testuadm.cveuni

- Datos internos.

login
id (INT PRIMARY KEY 10)
nombre (VARCHAR 200)
clave (VARCHAR 200)
password (VARCHAR 200)
usertype (VARCHAR 200)

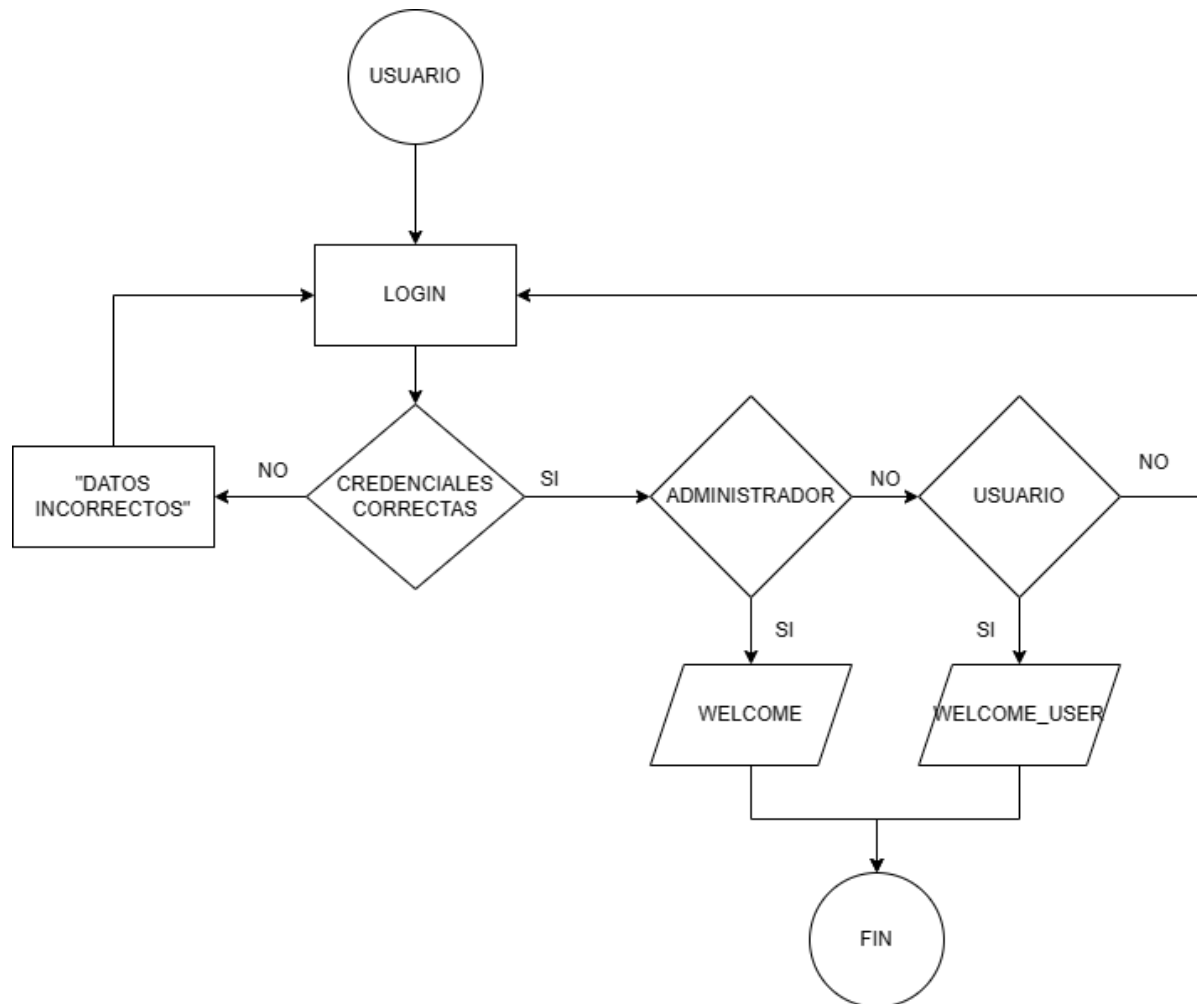
- Datos de salida.

tbl_organigrama
id_organigrama (INT PRIMARY KEY 10)
fecha (DATE)
oficio (VARCHAR 20)
cvedep (VARCHAR 15)
cveuni (VARCHAR 300)
secretaria (VARCHAR 200)
organismo (VARCHAR 200)
url_pdf (VARCHAR 200)
url_visio (VARCHAR 200)
vigente (VARCHAR 200)

historico
id_organigrama (INT)
fecha (DATE)
oficio (VARCHAR 20)
cvedep (VARCHAR 15)
cveuni (VARCHAR 300)
secretaria (VARCHAR 200)
organismo (VARCHAR 200)
url_pdf (VARCHAR 200)
url_visio (VARCHAR 200)
vigente (VARCHAR 200)
id_historico (INT PRIMARY KEY 10)

4.- Procesos y servicios ofrecidos por el sistema

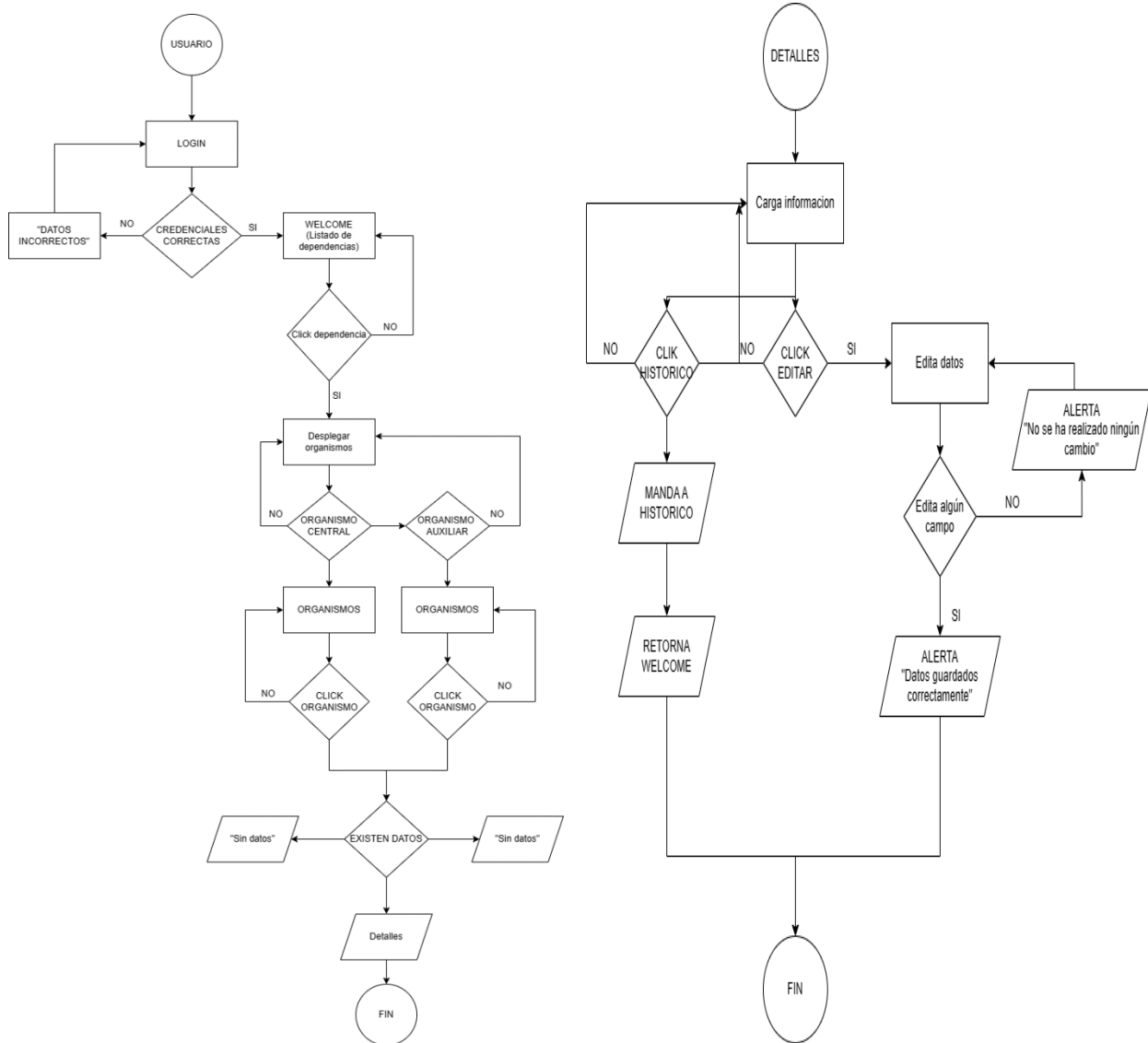
Login y acceso.



DIRECCIÓN GENERAL DE INNOVACIÓN - UNIDAD DE INFORMÁTICA
Guía para la Documentación técnica de un proyecto de software

LOGIN -> WELCOME

ADMINISTRADOR



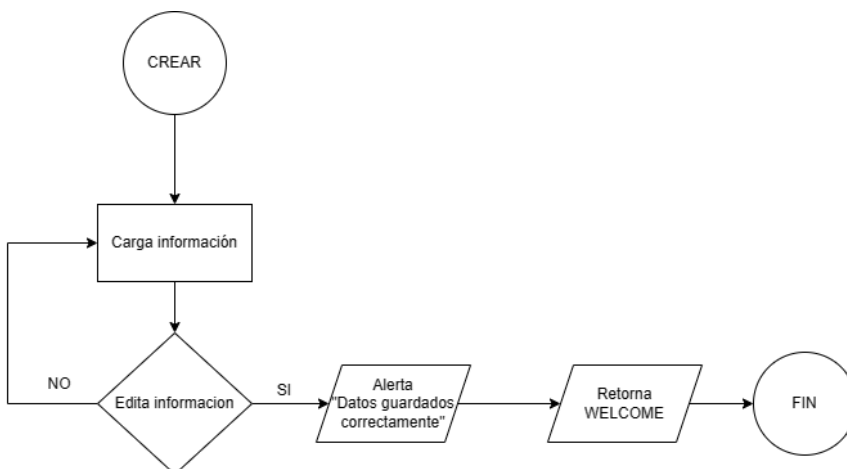
LOGIN -> WELCOME

En este apartado el administrador accede a la vista (welcome) que contiene el listado de las dependencias además de las validaciones a nivel backend para mostrar sus organismos correspondientes y si el usuario da click en detalle lo redirige a detalles con funciones de administrador.

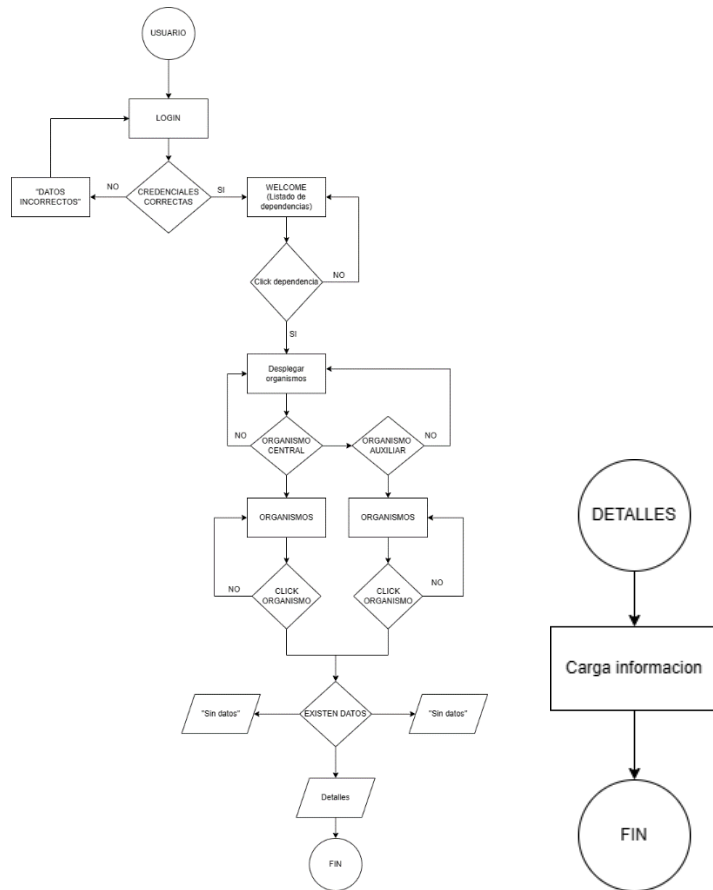
Si el administrador no encuentra ningún dato puede crear solo si hay organismos existentes rediriéndolos a la pagina de crear, validando los datos y si guarda retornando a la vista principal (welcome) para mostrar la nueva información cargada.

Si el administrador en la vista detalle da click en histórico los datos se mandan a una nueva tabla de histórico eliminando el valor de la tabla principal.

Si el administrador en la vista detalle da click en editar, los campos se habilitan en función de la validación de que valores si o no pueden activarse y si intentan guardar algún campo muestra alerta que no se ha guardado nada, caso contrario si se edita algún campo muestra alerta de datos guardados y los cambios se hacen directamente sin refrescar la pagina por medio de Ajax.



USUARIO



En este apartado el usuario accede a la vista `welcome_user` que contiene el listado de las dependencias además de las validaciones a nivel backend para mostrar sus organismos correspondientes y si el usuario da click en detalle lo redirige a detalles sin funciones.

Si el usuario intenta acceder a alguna pagina de administrador el sistema muestra una ventana de negación de acceso y muestra un botón que lo retorna a la vista principal de usuario (`welcome_user`).

5.-Documentación técnica - Especificación API

Guía para el desarrollo y documentación del sistema

Módulos del sistema:

Logueo - Principal

- Index.php
- Code.php
- Logout.php
- Security.php

Vista Super Administrator

- Add_admin.php
- Vista a todos los modulos de administrador

Vista Administrator

- Welcome.php
- Detalle_ua_org.php
- Crear_ua_org.php
- Editar.php
- Enviar.php
- Registrar.php
- Guardar.php

Vista Usuario

- Welcome_user.php
- Detalle_ua_org_user.php

DIRECCIÓN GENERAL DE INNOVACIÓN - UNIDAD DE INFORMÁTICA
Guía para la Documentación técnica de un proyecto de software

Aspectos a considerar

Componentes	Descripción
BASE DE DATOS	
Database	Base de datos donde se almacenará información de un sistema.
Table	Tablas internas de la base de datos relacionales o no relacionales.
Int	Tipo de variable entero o numérico.
Varchar	Tipo de variable, cadena de texto.
Date	Tipo de variable de fecha.
Primary key	Tipo de variable para establecer datos y relaciones.
FRONT END PHP	
HTML	Se utiliza para estructurar el contenido y dar formato a los documentos mediante un conjunto de etiquetas y atributos.
Style	Es la etiqueta que le dará estilo a la plantilla y sus distintos módulos.
Footer	Es el pie de pagina que se le asignara a la estructura html.

DIRECCIÓN GENERAL DE INNOVACIÓN - UNIDAD DE INFORMÁTICA
Guía para la Documentación técnica de un proyecto de software

Header	Es el encabezado que se le asignara a la estructura html.
Nav	Es la navegación que se le asignara a la estructura html.
Div	Consta de pequeños fragmentos de código para implementar variables o funciones de texto.
Input	Son variables de campos para insertar contenido.
Button	Son acciones que redirigen a un modulo o ejecutan alguna acción en específico.
Background-color	Color de fondo.
Color	Color de letra.
Font-family	Tipo de letra.
BACKEND PHP	
Framework	Marco de trabajo.
Bootstrap	Marco de trabajo de diseño.
Vendor	Frameworks dependientes.
.js	Librerías Javascript.

DIRECCIÓN GENERAL DE INNOVACIÓN - UNIDAD DE INFORMÁTICA
Guía para la Documentación técnica de un proyecto de software

Ajax	Marco de trabajo para Interactuar en la página web sin interrupción.
.css	Hojas de estilos.
.php	Marco de trabajo para módulos php del lado backend.

Funciones			
Nombre	Tipo	Propósito	Descripción
\$_SESSION	Variable	Asignar valor de sesión	Se basa en aplicar y almacenar las variables de logueo para guardar sesiones.
session_start();	Dato	Guardar valor de sesión	Almacena la información de sesión para asignar a la variable.
header	Cabecera de retorno	Retornar mensaje de variable	Retorna un mensaje en concreto en la pagina

DIRECCIÓN GENERAL DE INNOVACIÓN - UNIDAD DE INFORMÁTICA
Guía para la Documentación técnica de un proyecto de software

Include	Elemento	Incluir información	Permite incluir una página entera dentro de otra
Foreach	Bucle	Recorrer estructuras	Permite leer información y asignar de manera independiente
If	Estructura de control	Ejecución de condicional	Ejecutar sentencias verdaderas de manera indefinida
Else	Estructura de control	Ejecución de condicional	Ejecutar sentencias falsas de manera indefinida
Const	Elemento	Guardar valores	Guarda un valor fijo que no se puede modificar
Return	Ejecución	Termina la ejecución	Finaliza la función y retorna a algún modulo
\$POST	Array	Enviar variables	Permite enviar y recibir variables
\$FILES	Script	Almacenar información	Asigna nombre del archivo

Funciones descritas

Validación de sesión: Las sesiones comienzan por obtener las variables del formulario de inicio por medio de \$POST y primero se hace una consulta a base para verificar si el usuario existe anteriormente además de después hacer validaciones por medio de if y else a las sesiones que traen valores de acuerdo a cada tipo de usuario y dependiendo de este redirigiendo a una página páginas en concreto.

Nuevos usuarios: Los usuarios se insertan con un nuevo formulario enviando los valores por \$POST y simplemente validando si existen datos similares o aplicando la función de INSERT en la base de datos.

Listado de dependencias: Las dependencias se listan por medio de collapse de Bootstrap, y se asignan collapse independientes por medio de la consulta a base de datos por medio de PDO y foreach y al final los valores dentro de cada collapse (Collapse1, Collapse2).

Listado de organismos: Los organismos se listan dependiendo de los valores cvedep y cvepadre referenciándose, además de contar con valores de tipo de organismo que los diferencia entre auxiliares o centrales haciendo una nueva validación WHERE organismo = DESC o ODESC.

Creación de datos: Los datos se crean mandando la información por variables de welcome a crear y recibéndolos por \$POST para así saber que dependencia y organismo le corresponde además de sus claves.

Detalle de datos: Se hace una validación de datos por medio de los collapse de la página welcome y simplemente se manda el id que va recibir la información en la vista detalle.

Edición de datos: Los datos se editan por medio de Ajax en tiempo real, sin tener que actualizar la página, esto se hace por medio de Ajax y un form que contiene los botones, estos se habilitan y deshabilitan por la función PROP, y los valores se mandan por post a otra vista donde se hace la inserción de los datos por medio de UPDATE.

Cerrar sesión: Las sesiones que se guardan simplemente se mandan por \$POST a otro modulo que contiene \$session(destroy); que elimina y borra para que un nuevo usuario pueda acceder.

Se indica el propósito y breve descripción de cada método/función, con su prototipo indicando argumentos (nombre, tipo, propósito de cada uno) y respuesta (tipo, descripción).

Para llevar a cabo esta tarea, es posible utilizar una variedad de herramientas de generación de documentación automática, a partir del código en el encabezado de cada función (por ejemplo, PHPDoc, Doxygen, etc).

La documentación técnica debe pensarse como el manual del programador, y debe apuntar a aquellas personas que estarán a cargo de mantener, ampliar, o crear un proyecto derivado a partir de nuestro proyecto.

Aspectos relevantes

- Indicar claramente cómo invocar el programa (como la que haría cualquier sinopsis de una página de manual), conteniendo qué parámetros son opcionales, cuales son obligatorios, y documentar bien cuál es la utilidad de cada parámetro y cuál es el comportamiento por defecto si se omite algún parámetro opcional. Esto conforma comúnmente el manual del usuario final de la aplicación. (No aplica en Sistema de Mejores Prácticas).
- Incorporar diagramas de flujo y explicaciones a nivel método de la solución, debe explicarse la estrategia general de resolución donde se pueda apreciar cómo interactúan los módulos entre sí.

- Los tipos de datos abstractos (TDAs) deben estar adecuadamente documentados en el código, por otra parte, en el manual deben constar las limitaciones que posee la representación, cómo se representa una determinada estructura y detalle de métodos que provee el TDA para la manipulación de los datos.
- Incluir una sección de “Conclusiones y Glosario, si es necesario”, donde se deben resumir complicaciones encontradas durante el desarrollo del proyecto, políticas adoptadas para su resolución, restricciones al problema original, casos particulares y finalmente aspectos relacionados a la experiencia obtenida en base a la temática del proyecto.