## 1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

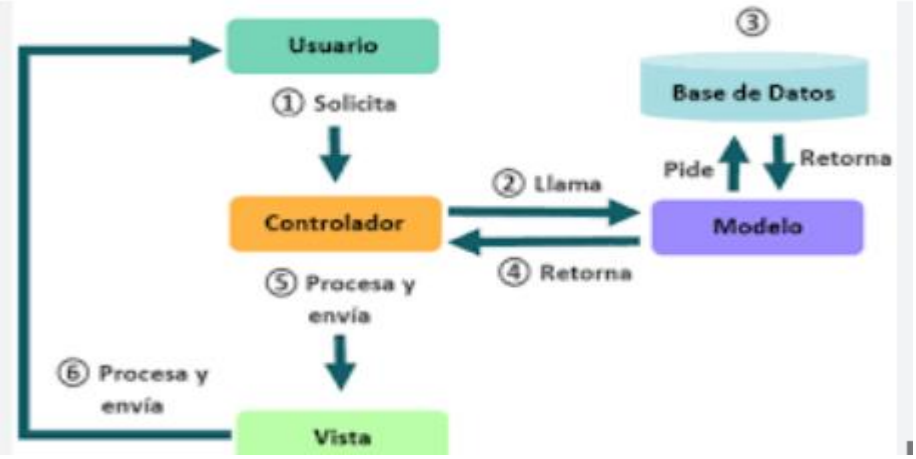
- **Denominación del Programa de Formación:** Técnico en Programación de Software
- **Código del Programa de Formación**: 233104 v1
- **Nombre del Proyecto:** Desarrollo de aplicaciones de software para el sector empresarial
- **Fase del Proyecto**: Fase 3 - Ejecución del Proyecto (Ejecución)
- **Actividad de Proyecto**: Desarrollar el software a partir del diseño y la arquitectura propuesta.
- **Competencia**: 220501097 - implementar la solución de software de acuerdo con los requisitos de operación y modelos de referencia
- **Duración de la Guía:** Total 200 horas, trabajo directo 160 horas y 40 horas de trabajo autionomo

## 2. PRESENTACIÓN

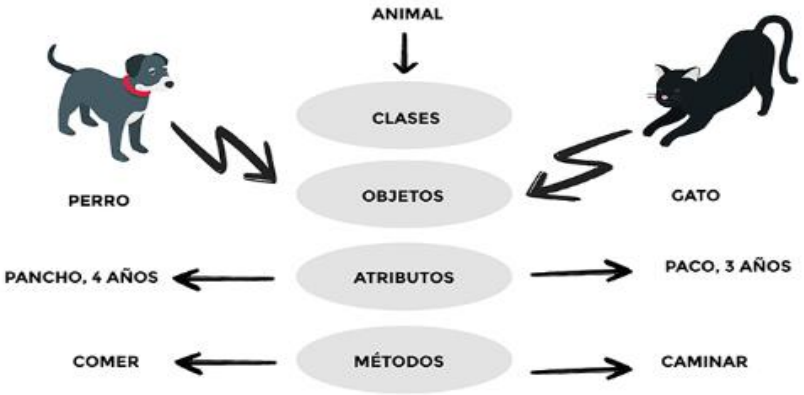
Estimado aprendiz, el objetivo de la siguiente guía es la Implementación de **Back-End,** para el logro de este objetivo se trabajara, el patrón de diseño de software **MVC** (Modelo Vista Controlador), utilizando **POO** (Programación Orientada a Objetos,el cual es un modelo de programación basado en el **concepto de clases y objetos.**

Conexion a la base de datos mysql, login y finalmente crear registros, consultas, actualizaciones, borrado en donde se utilizara bootstrap, html5 para el diseño web, en la cual será desarrollado de manera definitiva su proyecto formativo.

**MVC** (Modelo Vista Controlador), intervienen tres componentes distintos, desempeñando funciones especificas, los cuales se pueden definir de la manera más sencilla: los datos (**modelo**), se encarga de gestionar la información y los datos de la aplicación, administrando todos los accesos a dicha información, tanto a ficheros, como a clases o bases de datos, la interfaz de usuario (**vista**), es el componente encargado de generar la información necesaria y apropiadao en el que el usuario pueda interactuar **(**botones, textos, ventanas, llamadas a la acción, estilos, etc) y la lógica de control (**controlador),** el cual gestiona los eventos y acciones del usuario, encontrándose de intermediario entre el modelo y la vista, como el componente que decide qué acciones se van a realizar a continuación de acuerdo a las peticiones que se realicen de la información y asi mismo le indica a la vista cómo debe actuar y si ha de actualizar la información que se muestra en pantalla, navegar a otras ventanas, etc.




**POO** (Programación Orientada a Objetos**), Permite** en el desarrollo de cualquier sistema de información ver la realidad en que se mueve el mundo ya que identifica las entidades que se convierten en un **objeto** con propiedades (**datos**) y comportamientos (**funcionabilidades**) que existen en donde su objetivo esta en las interaciones entre estos objetos con la finalidad de crear un objetivo común del sistema de gestión que se este creando.




## 3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

Estas actividades nos lleva a tener los conocimientos, habilidades y destrezas para desarrollar back-end, en donde se puede empezar contruyendo aplicaciones básicas (CRUD).

El backend, es el responsable de la lógica de la aplicación del lado del servidor y de la integración del trabajo que realizan los desarrolladores de front-end.



- Ambiente Requerido

- Materiales
- Descripción de la(s) Actividad(es)

**3.1. Actividades Reflexión Inicial N° 1.**

- **Ambiente Requerido**. Ambiente de formación
- **Materiales**. Video beam, videos, marcadores, lápices, esferos, hojas, computador.
- **Tiempo estimado de trabajo.** 10 horas
- **Descripción de la Actividad.** ***Conceptos de Programación básica en PHP***.En la carpeta llamada **Ova_Programacion_Básica**, que se encuentras en el material del aprendiz de la guía N°7. Se debe abrir la página web Index.html, navega esta página y realiza las actividades propuestas sobre conceptos básicos de programación en PHP, al terminar imprima el pantallazo de sus resultados y guárdelo como: 1***_Programación _Basica_Apellido****,* esta actividad debe ser subida al portafolio del aprendiz, en la carpeta **evidencias** de la **fase de ejecución** correspondiente a la **guía 7.**

**3.1.2. Actividades Reflexión Inicial N° 2**

- **Ambiente Requerido**. Ambiente de formación
- **Materiales**. Video beam, videos, marcadores, lápices, esferos, hojas, computador.
- **Tiempo estimado de trabajo.** 12 horas
- **Descripción de la Actividad.** Conceptos y análisis de la importancia de la programación orientada a objetos e importancia de los requisitos para la aplicación. La cual se inicia con la lectura individual del material del aprendiz de la guía N° 7. De la carpeta POO, en donde se encuentra **7_01_Lectura_Orientacion_Objetos_PHP.pdf** y el archivo **7_2_ Lectura _ Guia_ Basica.pdf**, el cual se debe complementar con una búsqueda en internet.

Posteriormente se realiza una socialización en inglés, en la cual cada grupo de proyecto tomará la palabra y dará su punto de vista en un escrito en inglés acerca de las ventajas y desventajas de la programación orientada a objetos que fueron halladas por su equipo de trabajo.

Guarde el archivo con el nombre.**2_Actividad_Reflexión_Inicial.pdf,** esta actividad debe ser subida al portafolio del aprendiz, en la carpeta **evidencias** de la **fase de ejecución** correspondiente a la **guía 7.**

**3.1.3. Actividades Reflexión Inicial N° 3**

- **Ambiente Requerido**. Ambiente de formación
- **Materiales**. Video beam, videos, marcadores, lápices, esferos, hojas, computador.
- **Tiempo estimado de trabajo.** 12 horas
- **Descripción de la Actividad.** Conceptos y análisis de la importancia del patrón de diseño MVC, La cual se inicia con la lectura individual del material del aprendiz de la guía N° 7. El archivo **07_04_Modelo_Vista_Controlador** y en la carpeta **MVC EN PHP.**



Posteriormente se realiza una socialización en inglés, en la cual cada grupo de proyecto tomará la palabra y dará su punto de vista en un escrito en inglés acerca de como se organiza y estrucuta los

componentes de un sistema software, sus responsabilidades y las relaciones existentes entre cada uno de ellos.

Guarde el archivo con el nombre.**3_Actividad_Reflexión_Inicial.pdf,** esta actividad debe ser subida al portafolio del aprendiz, en la carpeta **evidencias** de la **fase de ejecución** correspondiente a la **guía 7.**

3.1.4 **Actividad inicial de ejercicios sobre conceptos de Programación Orientada a Objetos.N° 4**

- **Ambiente Requerido**. Ambiente de formación
- **Materiales**. Video beam, videos, marcadores, lápices, esferos, hojas, computador.
- **Tiempo estimado de trabajo.** 4 horas
- **Descripción de la(s) Actividad(es). Variables en PHP.** En el material del aprendiz encontrara el Taller **07_01_Tipos_Variables_PHP y el video 07_01_Tipos_Variables_PHP.mp4,** Guarde el archivo con el nombre **4_actividad_de_variables en php.pdf** ,esta actividad debe ser subida al portafolio del aprendiz, en la carpeta **evidencias** de la **fase de ejecución** correspondiente a la **guía 7.**

### 3.2. Actividades de Apropiación del Conocimiento (Conceptualización y Teorización).

En este apartado se requiere que el aprendiz realice la lectura del material de formación y se apropie de los conceptos y conocimientos dispuestos en los diversos recursos que se encuentran en la carpeta del material del aprendiz, relacionados con el patrón de diseño MVC aplicados al desarrollo de aplicaciones web en PHP, en donde se asociara los principios de la programación orientada a objetos, junto con la conexión a la base de datos. En guías anteriores aprendió el (DDL) para definir la estructura de los datos, ahora con el desarrollo de esta guía estará en la capacidad de realizar las operaciones básicas con datos (CRUD) empleando el (DML) y un motor de Bases de Datos como MySQL y/o SQLServer.

- **Ambiente Requerido**. Ambiente de formación
- **Materiales**. Video beam, videos, marcadores, lápices, esferos, hojas, computador.
- **Tiempo estimado de trabajo.** 40 horas
- **Descripción de la Actividad.** En los mismos grupos de elaboración de su proyecto formativo llevaran acabo las siguientes actividades:

### 3.2.1. Actividades de Apropiación del Conocimiento N° 1

En la carpeta **POO**, se encuentran los archivos: **7_2_Lectura_Guia_Basica**, la cual se debe complementar con las lecturas: **7_3_Lectura_ Clases, Objetos y Encapsulamiento, 7_4_Lectura Herencia y 7_5_Lectura Polimorfismo,** visualice los videos: **07_02_Clases_Objetos,** en donde cada equipo de trabajo debe:

1. Realizar las lecturas, elaborar un cuadro en Word de ventajas y desventajas de POO
2. Socializar en el ambiente de formación la descripción de su punto de vista a acerca de clases,objetos, herencia y polimorfismo en la POO.

Guarde el archivo con el nombre.**5_Actividad_ POO_contextualización _Apellido** esta actividad debe ser subida al portafolio del aprendiz, en la carpeta **evidencias** de la **fase de ejecución** correspondiente a la **guía 7.**

### 3.2.2. Actividades de Apropiación del Conocimiento N° 2

En la carpeta de **MVC en PHP**, se encuentra los archivos: **7_6_LECTURA_ Entendiendo _ MVC _en_PHP**,se debe complementar con los archivos **7_7_Que_ Son_Patrones_de_Diseño, 7_8_ Contextualizacion _ Modelo _ Vista _ Controlador,** en donde cada equipo de trabajo debe:

1. Realizar las lecturas corresponientes de MVC por grupos de trabajo.
2. posteriormente socializar por medio de una presentación en el ambiente de aprendizaje los conceptos acerca del MVC aplicados en desarrollo web, haciendo uso de carteleras o diapositivas con los dibujos o gráficas que consideren pertinentes.

Guarde el archivo con el nombre.**6_Actividad_ MVC_contextualización _Apellido** esta actividad debe ser subida al portafolio del aprendiz, en la carpeta **evidencias** de la **fase de ejecución** correspondiente a la **guía 7.**

### 3.3. Actividades de Transferencia del Conocimiento.

La transferencia de conocimiento es la oportunidad para el desarrollo de las personas y de las comunidades, desde los diversos ámbitos del saber de los sectores productivos, para nuestro caso en particular en este apartado trabajaremos una serie de ejercicios que nos llevaran a dar cumplimiento al objetivo propuesto en esta guía, relacionado a la Implementación de **Back-End.**

- **Ambiente Requerido**. Ambiente de formación
- **Materiales**. Video beam, videos, marcadores, lápices, esferos, hojas, computador, Xampp,Sublime Text 3.0 , visual studio code,
- **Tiempo estimado de trabajo.** 120 horas
- **Descripción de la Actividad.**

#### 3.3.1. Actividades de Transferencia del Conocimiento N° 1.

- **Tiempo estimado de trabajo. (10 horas)**

Crear proyecto en gestor de base de datos, en la carpeta del material del aprendiz descargar el **Taller 07_5_Programacion_ Orientada_ objetos,** Realizar el ejercicio y ejecutar el sitio web.

Guardar la actividad.**7_Actividad_Transferencia_MVC_Clases,** esta actividad debe ser subida al portafolio del aprendiz, en la carpeta **evidencias** de la **fase de ejecución** correspondiente a la **guía 7.**

### 3.3.2. Actividades de Transferencia del Conocimiento N° 2.

- **Tiempo estimado de trabajo. (20 horas)**

En la carpeta **POO**, encuentra **7_2_Lectura_Guia Basica,** siga el ejemplo completo del ejercicio completo que solicita hacer una aplicación que permita calcular el salario a pagar a los empleados de una empresa. En la empresa hay dos tipos de empleados: unos contratistas y otros de planta, a los contratitas el salario depende del valor de la hora.

Formula: Salario = ValorHora * TotalHoras
El salario de los de planta depende del sueldo básico mensual.
Formula = SalarioBasico + HorasExtras – deducciones

Se debe elaborar el diagrama de clases , código de clase empleado y contratista. Código implementación empleado planta y contratista y resultados implementación.

Guardar la actividad:.8_**Actividad_Transferencia_PHP,** esta actividad debe ser subida al portafolio del aprendiz, en la carpeta **evidencias** de la **fase de ejecución** correspondiente a la **guía 7.**

### 3.3.3. Actividades de Transferencia del Conocimiento N° 3. (35 horas)

**Tiempo estimado de trabajo. (40 horas)**

Construir una conexión a base de datos con PDO y realizar consultas CRUD SQL a partir de un diagrama de clases.

- Basado en el material de apoyo disponible (Videos, documentos y páginas web) y el ejemplo 07_03_Diagrama_ Clases_Código_ PHP.pdf, se debe construir una base de datos del proyecto asignado y realizar las respectivas clases en PHP.

- Se tendrán que diseñar los formularios básicos para la CRUD del proyecto asignado (usando Bootstrap).

- Posteriormente se debe crear la clase de conexión a base de datos usando PDO y las clases de los modelos, correspondientes al sistema a implantar, donde estarán los métodos para crear, consultar, leer y actualizar la información a la que tenga lugar.

- Esta actividad es de carácter grupal según como este conformado el proyecto y se debe tener en cuenta que todo el código programado debe estar con (métodos, clases, atributos) y adicionalmente debe estar documentado en su totalidad.



Guardar la actividad.9_**Actividad_Transferencia_Proyecto,** esta actividad debe ser subida al portafolio del aprendiz, en la carpeta **evidencias** de la **fase de ejecución** correspondiente a la **guía 7.**

### 3.3.4. Actividades de Transferencia del Conocimiento N° 4.

- **Tiempo estimado de trabajo. (40 horas)**

Teniendo en cuenta los ejemplos desarrollados anteriormente, y el ejemplo de implementación de una CRUD con MVC y PHP, que se encuentra dentro del material de apoyo en en la carpeta **MVC en PHP** , archivo **7_11_CRUD_MVC_PHP.zip** y la guía de montaje y ejecución de una aplicación CRUD en php utilizando mvc, que se encuentra en el archivo **7_12_Montaje_Ejecucion_MVC_PHP.pdf** y el archivo **7_13_CRUD _MVC _ PHP.pdf**, realice las operaciones **CRUD Propias** de su proyecto de formación.

Guardar la actividad.10_**Actividad**_ **CRUD_PHP_MVC_Proyecto_ Final_PHP,** esta actividad debe ser subida al portafolio del aprendiz, en la carpeta **evidencias** de la **fase de ejecución** correspondiente a la **guía 7.**

### 3.3.5. Actividades de Transferencia del Conocimiento N° 5.

- **Tiempo estimado de trabajo. (27 horas)**

Esta actividad trata de recopilar en una carpeta las evidencias de los diferentes reportes a saber:

1. Cree una carpeta con e nombre 07_11_CARPETA_Evidencias_Reportes_Proyecto y en ella guarde las siguientes evidencias:

2. Soportes de reportes generados. Ej: Archivo PDF/EXCEL con el resultado del reporte emitido por su sistema de información. Cada reporte debe ser guardado con el nombre 07_11_REPORTE_"nombre_del reporte" Ej: 07_11_REPORTE_Productos_Por_Agotarse.

3. Luego de terminar de implementar cada uno de los reportes solicitados y como copia de seguridad, guarde una copia del proyecto completo y 100 % funcional, que incluya el SCRIPT de la Base de Datos en un archivo en una carpeta con el nombre 07_11_BACKUP_Reportes_Proyecto

4. Por último, comprima la carpeta inicial en formato .ZIP o .RAR y realice la carga del mismo en la carpeta EVIDENCIAS_DE_APRENDIZAJE" de su portafolio de evidencias.

No olvide: Consolidar las evidencias de las actividades, realice la carga de las mismas en Territorium en el enlace de envío correspondiente, dentro del portafolio de evidencias. Susténtela cuando su instructor le indique.

## 4. ACTIVIDADES DE EVALUACIÓN

| Evidencias de Aprendizaje | Criterios de Evaluación | Técnicas e Instrumentos de Evaluación |
|---|---|---|
| **1. Evidencias de Conocimiento:**<br>• Características de la programación orientada a objetos.<br>• Responder sobre que es el patrón de diseño MVC, el esquema de ficheros y la interacción entre cada una de las capas que lo componen<br>**2. Evidencias de Desempeño:**<br>• Implementa los conceptos de la programación orientada a objetos y la conexión a base de datos mediante PDO.<br>• Presentar sustentación de ejercicios propuestos en la guía de aprendizaje correspondiente a MVC.<br>**3. Evidencias de Producto:**<br>• Aplica a su proyecto de formación cada uno de los conceptos de forma pertinente y de acuerdo a los estándares de programación.<br>• Desarrollar ejercicios propuestos en la guía, en los cuales aplique los conceptos de MVC de acuerdo a las situaciones planteadas | • Comprende de forma clara las ventas y desventajas de la programación orientada a objetos, así como sus conceptos básicos.<br>• Identifica los principios teóricos de la implementación de MVC en aplicaciones web<br>• Aplica de forma correcta los principios de programación orientada a objetos y conexión a base de datos mediante PDO.<br>• Reconoce las diferencias entre aplicaciones que implementan patrones de diseño con las que no lo hacen.<br>Aplica de forma correcta los principios de programación orientada a objetos y conexión a base de datos mediante PDO.<br>• Elabora aplicaciones web haciendo uso del patrón de diseño MVC | • Formulación de Preguntas.<br>• Instrumento Cuestionario.<br>• Taller de programación.<br>• Incorporación al proyecto formativo<br>• Actividad de Pre Saberes<br>• Cuestionario sobre MVC<br>• Lista de Chequeo<br>• Entregable CRUD con MVC aplicada a proyecto de formación |

## 4. GLOSARIO DE TÉRMINOS

- **POO:** Programación Orientada a Objetos**.**
- **Construct:** Método Constructor de una clase en PHP.
- **Destruct:** Método destructor de una clase en PHP.
- **$this:** Variable que se utiliza en una clase para poder acceder a los métodos o atributos de la clase.
- **Class:** Palabra reservada utilizada en la definición de una clase.
- **Extends:** Palabra utilizada en la definición de una clase que hereda a otra clase.
- **Tiempo real de ejecución:** Es una acción llevada de forma paralela a la interpretación es decirlínea leída, línea ejecutada.
- **Embebidos:** Es decir que pueden ser "pegados" dentro de otros lenguajes o "combinar" las funciones de uno con otro y obtener un resultado más amplio**.**
- **Librerías:** Son componentes específicos (en este caso) de un lenguaje de programación que le permiten el correcto funcionamiento e interpretación del código fuente
- **Servidor Apache: E**s el encargado de interpretar el código PHP, así como también ejecutarlo**.**
- **Servidor Mysql:** Encargado de ejecutar mysql, es decir el responsable de la base de datos y de la manipulación de losmismos.
- **Xampp:** Un paquete (que nos facilita la vida), con los componentes necesarios para la ejecución de nuestros scripts PHP, ya que contiene las librerías del lenguaje, servidor apache, y servidor MySql.
- **Editor de Código:** Es un procesador de texto que nos ayuda a visualizar de manera más directa el código fuente utilizado, resaltando las palabras claves del lenguaje.
- **Frontend:** Son todas aquellas tecnologías que corren del lado del cliente, es decir corren del lado del navegador web, generalizandose mas que nada en tres lenguajes, Html , CSS Y JavaScript
- **Backend:** El programador backend es aquel que se encuentra del lado del servidor, es decir, esta persona se encarga de lenguajes como PHP, Python, .Net, Java, etc, es aquel que se encarga de interactuar con bases de datos, verificar manejo de sesiones de usuarios, montar la página en un servidor, y desde éste "servir", todas las vistas que el FrontEnd crea, es decir, uno como backend se encarga mas que nada de la manipulación de los datos.
- **Patrón de Diseño:** "Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software." En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares.
- **Arquitectura de Software:** Conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software



- **Framework:** Podemos definir a un Framework, como un marco de trabajo, como un conjunto de código que podemos utilizar y reutilizar es un sistema personalizado, también en sistemas complejos o simples.

- **MVC:** Modelo-Vista-Controlador es un patrón de arquitectura de software que, utilizando 3, componentes (Vistas, Models y Controladores) separa la lógica de la aplicación de la lógica de la vista en una aplicación
- **ASP:** Active Server Pages (ASP) también conocido como ASP, es una tecnología de Microsoft del tipo "lado del servidor" para páginas web generadas dinámicamente

- **PROGRAMACIÓN POR CAPAS:** es un modelo de desarrollo software en el que el objetivo primordial es la separación (desacomplamiento) de las partes que componen un sistema software o también una arquitectura cliente-servidor: lógica de negocios, capa de presentación y capa de datos. De esta forma, por ejemplo, es sencillo y mantenible crear diferentes interfaces sobre un mismo sistema sin requerirse cambio alguno en la capa de datos o lógica.

## 5. REFERENTES BILBIOGRÁFICOS

El paradigma de Programación Orientada a Objetos en ASP y el patrón de arquitectura MVC:


- Documentación oficial PHP. 30 de Enero de 2018. Disponible en: **http://php.net/manual/es/index.php**

- Sintaxis básica PDO. 30 de Enero de 2018. Disponible en: https://diego.com.es/tutorial-de-pdo

- **CRUD con PDO: Conexión a Base de Datos.** Disponible en: https://www.youtube.com/watch?v=7yvGDbjQ6UY

- **CRUD MVC: Creación de Modelos [PHP 7 https://www.youtube.com/watch?v=5RNMtY19hLw**


**Videos:**


- Diagrama de Clases en PHP https://www.youtube.com/watch?v=Y2YmiaAhKHU

- Clases y objetos en php - Intro Parte 1 https://www.youtube.com/watch?v=ZO58K06tfaM

- Clases y objetos en php - Intro Parte 2 https://www.youtube.com/watch?v=sGno5LKVQ0s

- Video tutorial como genera un reporte grafico con PHP y la librería Jpgraph https://www.youtube.com/watch?v=xAtkj8r1_ec

## 6. CONTROL DEL DOCUMENTO

| | Nombre | Cargo | Dependencia | Fecha |
|---|---|---|---|---|
| **Autor (es)** | Fabiola Garcia Arismendi | Instructora | Centro de Gestión de Mercados, Logística y Tecnologías de la Información | Julio 2023 |

## 7. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

| | Nombre | Cargo | Dependencia | Fecha | Razón del Cambio |
|---|---|---|---|---|---|
| **Autor (es)** | | | | | |