**PRESENTACION DEL INFORME TECNICO**

**ARDILA OCHOA JACK NICOLAS**

**BARBOZA ALARCÓN RICHARD STEPHEN**

**CENTRO DE ELECTRICIDAD, ELECTRÓNICA Y TELECOMUNICACIONES**

**SERVICIO NACIONAL DE APRENDIZAJE (SENA)**

**ANÁLISIS Y DESARROLLO DE SISTEMAS DE INFORMACIÓN**

**BOGOTA D.C.**

**2020**

**TABLA DE CONTENIDO**

[1. MODELO RELACIONAL 4](#_Toc43464824)

[2. CUADRO COMPARATIVO 5](#_Toc43464825)

[3. HOSTING 6](#_Toc43464826)

[4. REQUERIMIENTOS TÉCNICOS 7](#_Toc43464827)

[4.1 REQUERIMIENTOS DE HARDWARE 7](#_Toc43464828)

[4.2 REQUERIMIENTOS DE SOFTWARE 7](#_Toc43464829)

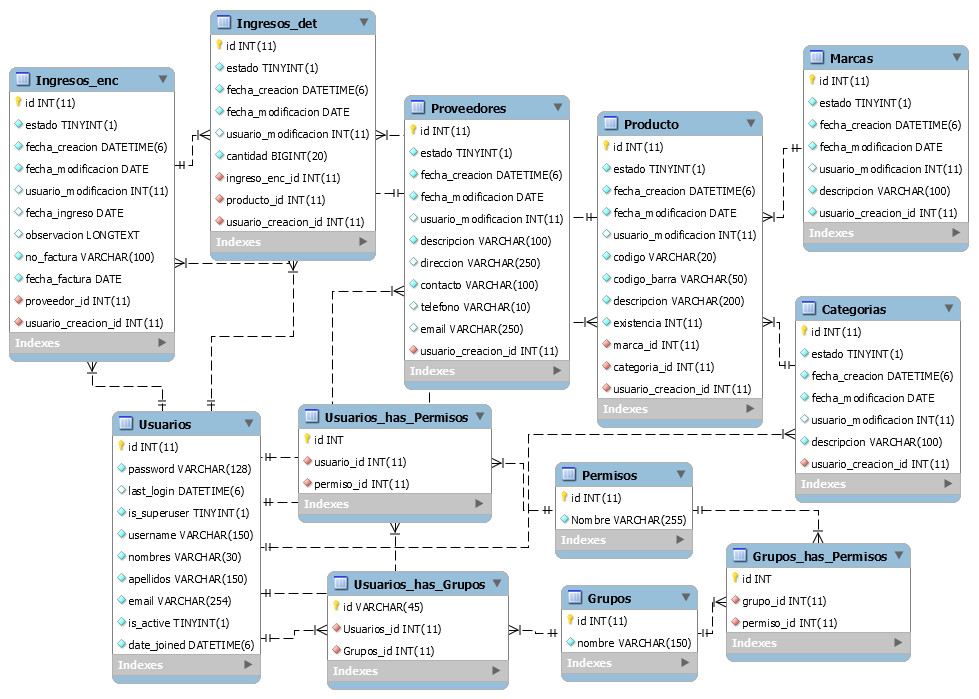
[5. PLAN DE INSTALACIÓN 8](#_Toc43464830)

[5.1 ARQUITECTURA DE SOFTWARE 8](#_Toc43464831)

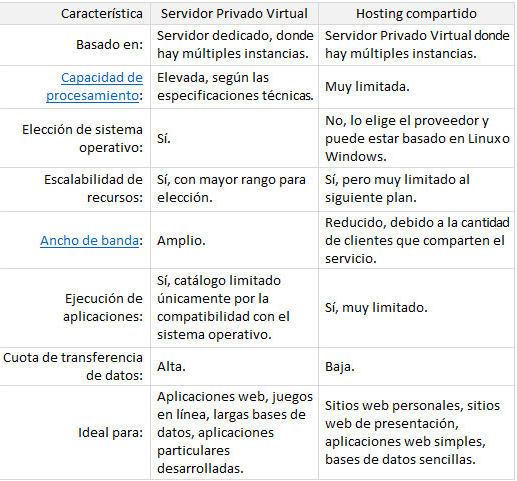
[5.2 CONFIGURACIÓN DEL SERVIDOR WEB 9](#_Toc43464832)

[6.1 ¿QUÉ ES UNA BACKUP? 10](#_Toc43464833)

# MODELO RELACIONAL



# CUADRO COMPARATIVO



# HOSTING



# REQUERIMIENTOS TÉCNICOS

## 4.1 REQUERIMIENTOS DE HARDWARE

* Equipo, teclado, mouse monitor, dispositivo móvil.
* Memoria RAM mínima de 1GB.
* Tarjeta de red LAN y/o Wireless.
* Disco duro de 500GB
* Procesador 1.4 GHz.

## 4.2 REQUERIMIENTOS DE SOFTWARE

* Sistema operativo: Windows NT/98/Me/2000/2003/XP/Vista/7/8/10.
* Privilegios de administrador.
* Conexión internet local y móvil.
* Navegador: Chrome, Firefox, Internet Explorer, Opera, entre otros.

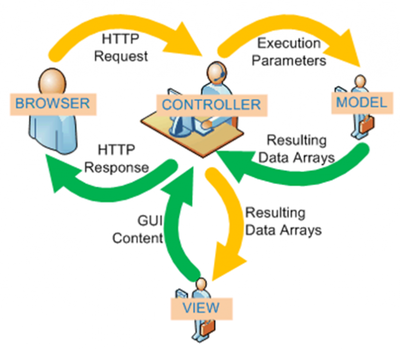
# PLAN DE INSTALACIÓN

## ARQUITECTURA DE SOFTWARE

Modelo-vista-controlador (MVC) es un patrón de arquitectura de software, que separa los datos y principalmente lo que es la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones.

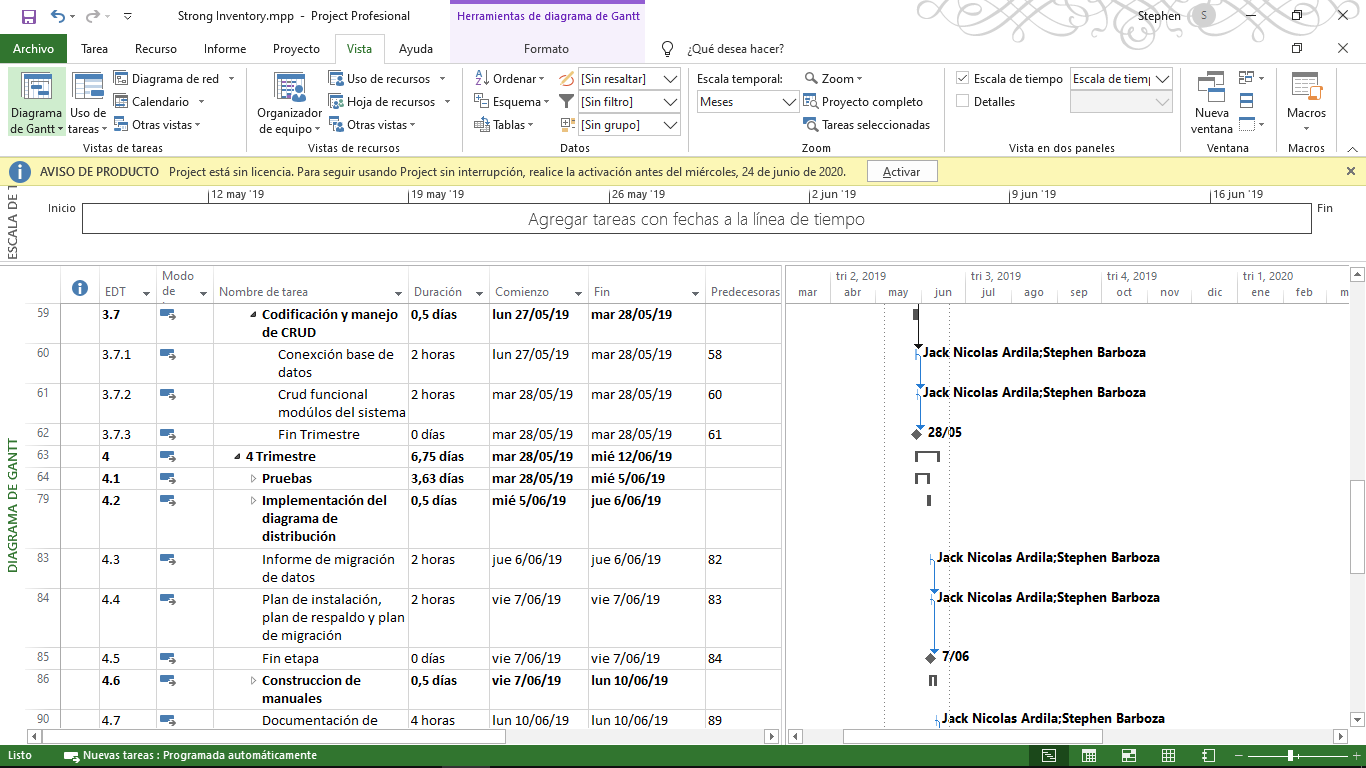
* El **Modelo**: Es la representación de la información con la cual el sistema opera, por lo tanto, gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio).
* El **Controlador**: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información.
* La **Vista**: Presenta el 'modelo' (información y *lógica de negocio*) en un formato adecuado para interactuar (usualmente la interfaz de usuario), por tanto, requiere de dicho 'modelo' la información que debe representar como salida.

El flujo que sigue el control generalmente es el siguiente:



1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón Observador para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

## CRONOGRAMA DE INSTALACIÓN



## CONFIGURACIÓN DEL SERVIDOR WEB

Toda la configuración para el funcionamiento de Apache se guarda en un archivo de texto nombrado: httpd.conf que se encuentra en la ruta *C:\Apache\conf*, lo podemos editar en cualquier editor de texto como el Bloc de notas.  
Abre el archivo httpd.conf y editar manualmente las líneas que se indican:

Todas las líneas que comienzan con el símbolo # son comentarios, explican en cada sección las distintas opciones, pero se encuentran en inglés.

La línea 52 Listen indica el puerto y dirección IP por el que el servidor va a recibir las peticiones, puedes usarla de las siguientes maneras:  
1- El servidor va recibir peticiones solo de la misma PC: Listen localhost:80  
2- Recibirá peticiones de otras máquinas en una red local: Listen 80

Esta es la configuración con los parámetros esenciales para comenzar a utilizar Apache. Guarda los cambios realizados y reinicia el servidor dando clic en el icono del área de notificación.

**6. PLAN DE RESPALDO DE DATOS**

## 6.1 ¿QUÉ ES UNA BACKUP?

Backup, respaldo, copia de seguridad o copia de reserva a una copia de los datos originales de un sistema de información o de un conjunto de software (archivos, documentos, etc.) que se almacena en un lugar seguro o una región segura de la memoria del sistema, con el fin de poder volver a disponer de su información en caso de que alguna eventualidad, accidente o desastre ocurra y ocasione su pérdida del sistema. En otras palabras, se trata de una copia por si acaso que, usualmente, se actualiza cada cierto tiempo como medida de seguridad.

Los sistemas informáticos en la actualidad pueden ser víctima de diversos tipos de siniestro: ataques remotos por parte de hackers o de virus informáticos, destrucción física de los soportes de almacenamiento, e incluso eliminación accidental por parte de un usuario autorizado. En estos casos se recurre manual o automáticamente al backup para restaurar la información perdida, minimizando así las pérdidas en materia de datos o información.

**6.2 TIPOS DE BACKUP**

**-Backup completo:** este tipo de backup hace un respaldo completo de todas las carpetas y archivos seleccionados. El respaldo abarca el 100% de los datos, por lo que suele ser el que lleva más tiempo en realizarse.

**-Backup diferencial:** contiene los archivos que han cambiado desde la última vez que se hizo el backup. Solo se incluyen los archivos nuevos y/o modificados desde el último backup.

**-Backup incremental:** se realiza un respaldo de todos los archivos que han sido modificados desde que fue ejecutado el último backup completo, diferencial o incremental. Es el método más rápido para realizar respaldos.

**–Backup espejo:** similar al backup completo, pero la diferencia es que los archivos no son comprimidos y no pueden ser protegidos usando un password.

**6.3 BACKUP STRONG INVENTORY**

Strong Inventory utilizará un Backup completo con el motor de base de datos PostgreSQL con el fin de hacer un Backup consistente y como tiene una base de datos pequeña será muy rápida, además que restaura todo.