



DENTAPLUS

Manual Técnico.



20 DE MAYO DE 2024

PIXEL CODE

8va calle y 14 av. Zona 3 Quetzaltenango.

Índice

1.	Arquitectura del Sistema	2
1.2	Descripción general.	2
1.3	Tecnologías usadas.....	2
	Django.	2
	SQL.	2
	Python.	2
	HTML.	3
	CSS.....	3
	JS.	3
2.	Instalación y Configuración.	4
2.1	Requisitos del sistema.	4
2.2	Procesos de instalación.	4
3.	Estructuras del Código Fuente.....	6
3.1	Estructura de Directorios.....	6
3.2	Módulos Principal.....	6
	Pacientes.	6
	Inventario.....	7
	Ventas.	7
	Recursos Humanos.....	7
	Financiero.....	7
	Bussines Intelligence.	7
	Citas.....	7
4.	Mantenimiento y Gestión del Sistema.	11
5.	Recursos Adicionales.	11
6.	Contacto y Soporte Técnico.....	12
	Contacto 1.	12
	Contacto 2.	¡Error! Marcador no definido.
	Contacto 3.	¡Error! Marcador no definido.

1. Arquitectura del Sistema

1.2 Descripción general.

Dental Plus es un tipo de software, el cual tiene una arquitectura compleja, cuenta con un control de pacientes, para maximizar y mejorar el trato que se les da a las personas, con este apartado se controla la información básica de los clientes, desde el nombre hasta el correo electrónico y su próxima cita. Inventario es una vista a todo aquello que la clínica tiene en físico y sirve para los tratamientos dentales. El control y uso de Inventario es muy necesario, nos ayudara a llevar un control total acerca del material a usar y utilizado, para evitar la pérdida o el consumo total del material. En el módulo ventas tenemos los servicios que la empresa ofrece y se mezcla con varios módulos que ayudara al desarrollo del tratamiento. Recursos humanos nos ayuda al control del personal desde el punto de vista tanto administrativo y laboral. Financiero principalmente tiene como tarea manejar las finanzas de la clínica. Muestra desde la información hasta los desgloses y ganancias. En el bussines intelligence se encuentra una proyección de lo que han sido los tratamientos, y la bonificación que le genera a cada asistente. El modulo citas ayuda a llevar un control más específico en cuanto al horario y citas que los pacientes agenden para una próxima visita a la clínica.

1.3 Tecnologías usadas.

Django.

Django es un marco web Python de alto nivel que fomenta el desarrollo rápido y un diseño limpio y pragmático. Construido por desarrolladores experimentados, se encarga de gran parte de la molestia del desarrollo web, por lo que puede concentrarse al escribir su aplicación sin necesidad de reinventar la rueda. Es gratis y de código abierto.

Django fue diseñado para ayudar a los desarrolladores a llevar las aplicaciones desde el concepto hasta su finalización lo más rápido posible. Se toma en serio la seguridad y ayuda a los desarrolladores a evitar muchos errores de seguridad comunes. Algunos de los sitios más concurridos en la web aprovechan la capacidad de Django para escalar de manera rápida y flexible.¹

SQL.

El lenguaje de consulta estructurada (SQL) es un lenguaje de programación para almacenar y procesar información en una base de datos relacional. Una base de datos relacional almacena información en forma de tabla, con filas y columnas que representan diferentes atributos de datos y las diversas relaciones entre los valores de datos. Puede usar las instrucciones SQL para almacenar, actualizar, eliminar, buscar y recuperar información de la base de datos. También puede usar SQL para mantener y optimizar el rendimiento de la base de datos.

Python.

Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML). Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes. El software Python se puede descargar gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo.

HTML.

HTML es el lenguaje con el que se define el contenido de las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web, como imágenes, listas, vídeos, etc.

El HTML se creó en un principio con objetivos divulgativos de información con texto y algunas imágenes. No se pensó que llegara a ser utilizado para crear área de ocio y consulta con carácter multimedia (lo que es actualmente la web), de modo que, el HTML se creó sin dar respuesta a todos los posibles usos que se le iba a dar y a todos los colectivos de gente que lo utilizarían en un futuro. Sin embargo, pese a esta deficiente planificación, si que se han ido incorporando modificaciones con el tiempo, estos son los estándares del HTML.

CSS.

CSS son las siglas en inglés para «hojas de estilo en cascada» (Cascading Style Sheets).

Básicamente, es un lenguaje que maneja el diseño y presentación de las páginas web, es decir, cómo lucen cuando un usuario las visita. Funciona junto con el lenguaje HTML que se encarga del contenido básico de los sitios. Se les denomina hojas de estilo «en cascada» porque puedes tener varias y una de ellas con las propiedades heredadas (o «en cascada») de otras.

JS.

JavaScript es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas. Desde actualizar fuentes de redes sociales a mostrar animaciones y mapas interactivos, las funciones de JavaScript pueden mejorar la experiencia del usuario de un sitio web. Como lenguaje de scripting del lado del servidor, se trata de una de las principales tecnologías de la World Wide Web. Por ejemplo, al navegar por Internet, en cualquier momento en el que vea un carrusel de imágenes, un menú desplegable “click-to-show” (clic para mostrar), o cambien de manera dinámica los elementos de color en una página web, estará viendo los efectos de JavaScript.

2. Instalación y Configuración.

2.1 Requisitos del sistema.

- Sistema Operativo: Se puede ejecutar en sistemas operativos compatibles con Python, como Linux, macOS o Windows.
- Python: Django y muchas de las herramientas relacionadas con el desarrollo web en Python requieren Python como requisito previo. Se recomienda utilizar Python 3.x, ya que Django 3.x es compatible con versiones de Python 3.
- Gestor de Base de Datos: Debes tener un servidor de bases de datos compatible instalado en tu sistema. PostgreSQL, MySQL o SQLite son opciones comunes para aplicaciones Django. Es necesario tener instaladas las bibliotecas de Python para interactuar con el sistema de gestión de bases de datos elegido.
- Django: Debes instalar Django, el framework web de Python. Puedes instalarlo mediante pip, el gestor de paquetes de Python
- Navegador Web: Para interactuar con la interfaz de usuario del software, necesitarás un navegador web moderno compatible con HTML, CSS y JavaScript.
- Herramientas de Desarrollo: Un editor de código o un entorno de desarrollo integrado (IDE) como Visual Studio Code, PyCharm o Sublime Text puede facilitar el desarrollo y la edición de código.
- Dependencias de Frontend: Si estás utilizando Bootstrap u otras bibliotecas de frontend, asegúrate de tener acceso a los archivos CSS y JavaScript necesarios, ya sea descargándolos directamente o utilizando un CDN.
- Herramientas de Control de Versiones (opcional pero recomendado): Utilizar un sistema de control de versiones como Git puede ser útil para el seguimiento de cambios en el código fuente y la colaboración en equipo.
- Servidor Web (opcional para desarrollo, necesario para producción): Aunque no es necesario para el desarrollo local, para implementar la aplicación en un entorno de producción, necesitarás un servidor web como Apache o Nginx para servir la aplicación Django.
- Configuración del Entorno Virtual (recomendado): Se recomienda configurar un entorno virtual de Python para el desarrollo, lo que te permite aislar las dependencias del proyecto y evitar conflictos con otros proyectos. Puedes usar herramientas como virtualenv o conda para esto.

2.2 Procesos de instalación.

Instalación de Python:

1. **Verifica si Python está instalado:** En la mayoría de las distribuciones de Linux, Python ya está instalado. Puedes verificar la versión instalada ejecutando el siguiente comando en la terminal:

`python --version`

2. **Instalación de Python 3:** Si necesitas instalar Python 3, puedes hacerlo utilizando el gestor de paquetes de tu distribución. Por ejemplo, en Ubuntu puedes usar:

```
sudo apt-get update sudo apt-get install python3
```

Configuración de un Entorno Virtual (Opcional pero Recomendado):

1. **Instalación de virtualenv:** Puedes instalar la herramienta **virtualenv** utilizando pip, el gestor de paquetes de Python:

```
pip install virtualenv
```

2. **Crear un entorno virtual:** Navega hasta el directorio de tu proyecto y crea un entorno virtual:

```
bash
```

```
cd /ruta/a/tu/proyecto virtualenv venv
```

3. **Activar el entorno virtual:** Activa el entorno virtual antes de instalar las dependencias de tu proyecto:

```
bash
```

```
source venv/bin/activate
```

Instalación de Django y otras Dependencias de Python:

1. **Instalación de Django:** Utiliza pip para instalar Django dentro de tu entorno virtual:

```
pip install django
```

2. **Instalación de otras dependencias:** Instala cualquier otra dependencia de Python necesaria para tu proyecto utilizando pip.

Instalación del Gestor de Base de Datos:

1. **Instalación de PostgreSQL, MySQL, o SQLite:** Utiliza el gestor de paquetes de tu distribución para instalar el servidor de base de datos que prefieras.

Configuración de la Base de Datos:

1. **Configuración de la Base de Datos en Django:** Configura la conexión a tu base de datos en el archivo **settings.py** de tu proyecto Django.

Ejecución del Servidor de Desarrollo:

1. **Iniciar el servidor de desarrollo:** Ejecuta el servidor de desarrollo de Django con el siguiente comando:

```
python manage.py runserver
```

2. **Accede a tu aplicación:** Abre un navegador web y visita la dirección proporcionada por Django en la terminal (normalmente **http://localhost:8000**).

3. Estructuras del Código Fuente.

3.1 Estructura de Directorios.

Listado de rutas de carpetas para el volumen OS

El número de serie del volumen es 2099-6565

C:.

```
├──.idea
|   └──inspectionProfiles
├──.vscode
├──app
|   ├──database
|   ├──migrations
|   |   └──__pycache__
|   ├──static
|   |   ├──css
|   |   ├──img
|   |   └──js
|   ├──templates
|   |   └──layouts
|   ├──templatetags
|   |   └──__pycache__
|   └──__pycache__
└──clinica
    └──__pycache__
```

3.2 Módulos Principal

Pacientes.

Se refiere a la sección en la cual se maneja la información de los clientes, tales como ID, Nombre, Apellido, Número telefónico, Correo electrónico, Edad, Numero de seguro y última visita. Al igual que actualizar cada uno de los campos. Se trabaja de esta manera para evitar algún inconveniente en cuanto a los errores para programar las citas.

[illegible]












Table Name: <input type="text" value="dentista"/>		Schema: clinica_dental									
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression	
 idDentista	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
 nombre	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	
 apellido	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	
 numero_telefono	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	
 correo_electronico	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	
 no_colgado	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	
 estado	TINYINT(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
 uri	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
 Tipo_Especialidad_idTipo_Es...	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		








Table Name: <input type="text" value="tratamiento"/>		Schema: clinica_dental									
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression	
 idTratamiento	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
 detalle	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	
 precio	FLOAT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	
 cantidad_citas	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	
 estado	TINYINT(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
 Cita_idCita	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
 Asistente_idAsistente	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		









Table Name: <input type="text" value="alergia"/>		Schema: clinica_dental									
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression	
 idAlergia	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
 descripcion	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	

Table Name: <input type="text" value="factura"/>		Schema: clinica_dental									
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression	
 idFactura	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
 detalle	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	
 cantidad_servicios	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	
 fecha	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	
 total	FLOAT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	
 Tratamiento_idTratamiento	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

[illegible]

[illegible][illegible][illegible][illegible][illegible]

4. Mantenimiento y Gestión del Sistema.

Utilizando PostgreSQL como base de datos:

Exportar la Base de Datos: Utiliza el comando `pg_dump` para exportar la base de datos a un archivo SQL:

```
pg_dump nombre_basededatos > backup_nombre_basededatos.sql
```

Programación de Tareas de Respaldo Automático: Puedes programar tareas en tu sistema para que realicen copias de seguridad periódicas utilizando `cron` o algún servicio similar.

Procedimiento para Realizar Copias de Seguridad de Archivos del Sistema:

Copia de Archivos Manualmente:

Archivos Estáticos: Copia los archivos estáticos (HTML, CSS, JS) a una ubicación segura en tu sistema.

Archivos de Configuración: Asegúrate de tener copias de seguridad de los archivos de configuración relevantes, como `settings.py` en un proyecto Django.

5. Recursos Adicionales.

Enlaces a páginas de documentación de los lenguajes de programación utilizados.

Python

<https://docs.python.org/3/>

Django.

<https://www.djangoproject.com/>

HTML

<https://developer.mozilla.org/es/docs/Web/HTML>

CSS.

<https://devdocs.io/css/>

Js.

<https://devdocs.io/javascript/>

Bootstrap.

<https://getbootstrap.com/docs/4.1/getting-started/theming/>

6. Contacto y Soporte Técnico.

Contacto 1.

Ana Masielle Coti Rodas.

Correo: amcotir@correo.url.edu.gt

Número de teléfono: +502 4608 7147

