

# MANUAL CENTRAL SUITES FERIA

## Python: Descripción General

**Python** es un lenguaje de programación de alto nivel, interpretado y de propósito general. Fue creado por **Guido van Rossum** y lanzado por primera vez en 1991. Es conocido por su simplicidad y legibilidad, lo que facilita a los desarrolladores escribir, leer y mantener el código. Python es una excelente elección tanto para principiantes como para profesionales debido a su versatilidad y comunidad activa.

---

### Características Principales de Python

1. **Sintaxis Clara y Legible:**
    - Diseñado para ser fácil de aprender y entender, con una sintaxis similar al lenguaje humano.
  2. **Multipropósito:**
    - Adecuado para aplicaciones web, ciencia de datos, automatización, inteligencia artificial, desarrollo de software, y más.
  3. **Extensa Biblioteca Estándar:**
    - Ofrece módulos pre construidos para tareas comunes como manejo de archivos, HTTP, matemáticas avanzadas, o manipulación de datos.
  4. **Lenguaje Multiplataforma:**
    - Funciona en diferentes sistemas operativos, como Windows, macOS y Linux.
  5. **Dinámicamente Tipado:**
    - No es necesario declarar el tipo de variable; Python lo infiere automáticamente en tiempo de ejecución.
  6. **Ampliable y Flexible:**
    - Puede integrarse con otros lenguajes (como C/C++, Java) para soluciones de alto rendimiento.
- 

## Django: Descripción General

**Django** es un framework web de alto nivel escrito en Python que facilita el desarrollo rápido y limpio de aplicaciones web. Fue diseñado por el equipo de desarrolladores de la organización **Lawrence Journal-World** y lanzado por primera vez en 2005. Django sigue el principio de diseño **DRY (Don't Repeat Yourself)**, promoviendo la reutilización del código y el desarrollo ágil.

---

### Características Principales de Django

1. **Framework "Baterías Incluidas":**
  - Django ofrece soluciones integradas para casi todos los aspectos del desarrollo web: autenticación, bases de datos, formularios, enrutamiento, y más.
2. **ORM (Object-Relational Mapping):**

- Simplifica las operaciones con bases de datos, permitiendo interactuar con ellas usando modelos Python en lugar de SQL directamente.
  - 3. **Arquitectura MVC (Model-View-Controller):**
    - Organiza el proyecto en tres componentes principales: Modelos, Vistas y Plantillas.
  - 4. **Seguridad Incorporada:**
    - Incluye medidas contra ataques comunes como XSS, CSRF, inyección SQL, entre otros.
  - 5. **Escalabilidad:**
    - Diseñado para manejar aplicaciones de pequeño a gran tamaño con facilidad.
  - 6. **Versatilidad:**
    - Usado para construir blogs, sistemas de e-commerce, APIs RESTful, redes sociales, entre otros.
  - 7. **Enfoque en el Desarrollo Ágil:**
    - Reduce el tiempo de desarrollo gracias a las herramientas integradas y a una comunidad activa.
- 

## Relación entre Python y Django

Django es una extensión poderosa del lenguaje Python que proporciona las herramientas necesarias para el desarrollo web estructurado, robusto y eficiente. Mientras Python maneja la lógica general del programa, Django agrega características específicas para trabajar con aplicaciones web de manera segura y escalable.

## Implementación de Proyecto con Django 3.2

### 1. Preparación del Entorno

- **Sistema Operativo:** Ubuntu Server 20.04 (recomendado).
- **Python Version:** Python 3.9 (compatible con Django 3.2).
- **Base de Datos:** PostgreSQL o MySQL.
- **Servidor Web:** Gunicorn con Nginx (para producción).
- **Administración del entorno:** Utilizar un entorno virtual venv.
- **Instalar dependencias usando el archivo requirements.txt**

### Implementación del servidor.

En la carpeta service encontramos los servicios a usar para instalar el proyecto en línea. usando system y nginx. mas

info: <https://www.digitalocean.com/community/tutorials/how-to-set-up-django-with-postgres-nginx-and-gunicorn-on-ubuntu> Podemos apoyarnos en esta documentación el cual tiene todo lo necesario.

### Github proyecto.

<https://github.com/OMMDIGITAL202/HOTELESSUITEFERIA>

## Modelo de base de datos y estructura.

### 1. Booking

- **Propósito:** Representa una reserva, almacenando detalles como el título, dirección y enlace de la reserva.
  - **Campos:**
    - start: Hora o inicio de la reserva.
    - title: Título de la reserva.
    - link: Enlace relacionado con la reserva.
    - address: Dirección de la reserva.
    - distance: Distancia o ubicación relevante.
    - description: Descripción detallada.
    - img: URL de la imagen asociada.
    - updated y created: Fechas de última actualización y creación.
  - **Uso:**
    - Añadir registros con nuevos datos de reservas.
    - Visualizar todas las reservas en el panel admin.
  - **Cómo Usar en Admin:**
    - **Crear:** Proporciona los campos requeridos en el formulario de admin.
    - **Editar:** Accede a un registro y modifica la información deseada.
    - **Eliminar:** Selecciona la reserva desde el listado y elimina.
- 

### 2. Complement

- **Propósito:** Datos complementarios relacionados con búsquedas y ocupación.
  - **Campos:**
    - total\_search: Número total de búsquedas.
    - occupancy: Capacidad u ocupación de la búsqueda.
    - start: Hora o inicio.
    - date\_from, date\_to: Intervalo de fechas.
    - updated y created: Tiempos de actualización y creación.
  - **Uso:**
    - Estadísticas sobre ocupación y fechas relevantes.
  - **Cómo Usar en Admin:**
    - Igual a **Booking**.
- 

### 3. AvailableBooking

- **Propósito:** Define la disponibilidad de reservas en rangos de fecha.
- **Campos:**
  - Incluye referencias a Booking y Complement.
  - date\_from, date\_to: Fechas disponibles.
  - price: Precio de la reserva.

- active: Indica si está activo.
    - Otros: total\_search, occupancy, start, etc.
  - **Uso:**
    - Relaciona disponibilidad con reservas.
    - Proporciona precios y datos específicos por fechas.
  - **Cómo Usar en Admin:**
    - Sigue el mismo proceso de gestión en admin para relaciones.
- 

#### 4. ProcessActive

- **Propósito:** Define procesos activos con opciones como tipo de búsqueda y posiciones.
  - **Campos:**
    - date\_end: Fecha de finalización.
    - occupancy, start: Datos de ocupación y comienzo.
    - position: Información en formato JSON.
    - type\_proces: Tipo de proceso (Ciudad o Nombre).
  - **Uso:**
    - Configuración de procesos de búsqueda activos.
  - **Cómo Usar en Admin:**
    - JSON necesita formato válido al editar campos.
- 

#### 5. GeneralSearch

- **Propósito:** Datos generales para una búsqueda.
  - **Campos:**
    - url: URL base para la búsqueda.
    - city\_and\_country: Ciudad y país objetivo.
    - time\_sleep\_minutes: Tiempo de espera en minutos.
    - type\_search: Tipo de búsqueda.
    - proces\_active: Relación con procesos activos.
  - **Uso:**
    - Información básica sobre el proceso de búsqueda.
  - **Cómo Usar en Admin:**
    - Añadir configuraciones de búsqueda básicas.
- 

#### 6. AvailSuitesFeria

- **Propósito:** Fechas específicas de disponibilidad de suites.
- **Campos:**
  - date\_avail: Fecha disponible.
- **Uso:**
  - Registro de disponibilidad específica.
- **Cómo Usar en Admin:**

- Gestor simple para fechas.

---

## 7. CantAvailSuitesFeria

- **Propósito:** Cantidades relacionadas con AvailSuitesFeria.
- **Campos:**
  - type\_avail: Tipo de disponibilidad.
  - avail: Número de habitaciones disponibles.
  - Relación con AvailSuitesFeria.
- **Uso:**
  - Gestión detallada de capacidades por fecha.
- **Cómo Usar en Admin:**
  - Relacionar correctamente con AvailSuitesFeria.

---

## 8. TemporadaByDay

- **Propósito:** Define colores y texto para diferentes días de temporada.
- **Campos:**
  - bg\_color, text\_color: Colores asociados.
  - number: Número identificativo.
  - date\_from, updated, created: Fechas relacionadas.
- **Uso:**
  - Decoración y marcado de temporadas en frontend.
- **Cómo Usar en Admin:**
  - Asegurar las opciones válidas para colores.

---

## 9. Price

- **Propósito:** Precios por fechas y ocupación.
- **Campos:**
  - date\_from: Fecha de inicio.
  - price: Valor del precio.
  - occupancy: Ocupación relacionada.
- **Uso:**
  - Información relevante para precios históricos.
- **Cómo Usar en Admin:**
  - Relacionar ocupación correctamente.

## AvailWithDate

### Descripción

El modelo AvailWithDate se utiliza para registrar información relacionada con la disponibilidad de un elemento específico en una fecha dada. Es adecuado para gestionar datos con fechas asociadas y la capacidad de actualizarlos con marcas de tiempo.

---

### Atributos del Modelo

#### Campos

1. **date\_from**
  - **Tipo:** CharField
  - **Máxima longitud:** 30 caracteres
  - **Descripción:** Indica la fecha desde la cual está disponible el elemento. Se guarda en formato de texto.
2. **avail**
  - **Tipo:** CharField
  - **Máxima longitud:** 50 caracteres
  - **Descripción:** Representa la disponibilidad del elemento asociado.
3. **updated**
  - **Tipo:** DateTimeField
  - **Valor por defecto:** null=True, blank=True
  - **Descripción:** Guarda la fecha y hora de la última modificación del registro. Este campo es opcional.
4. **created**
  - **Tipo:** DateTimeField
  - **Valor por defecto:** null=True, blank=True
  - **Descripción:** Representa la fecha y hora de la creación del registro. Este campo es opcional.

---

### Métodos

- **\_\_str\_\_()**
  - **Descripción:** Método que representa al modelo como una cadena de texto legible.
  - **Salida:** Una concatenación del valor de avail y date\_from, separada por el carácter |.
  - **Ejemplo de salida:**  
"Disponible | 2024-12-20"

## MessageByDay

### Descripción

El modelo MessageByDay se utiliza para gestionar mensajes asociados a fechas específicas y un número determinado de ocupantes. Es adecuado para aplicaciones relacionadas con comunicación o notificaciones personalizadas basadas en ocupación y tiempo.

---

### Atributos del Modelo

#### Campos

1. **date\_from**
  - **Tipo:** CharField
  - **Máxima longitud:** 30 caracteres
  - **Descripción:** Fecha a la que está asociado el mensaje, guardada como una cadena de texto.
2. **occupancy**
  - **Tipo:** IntegerField
  - **Elecciones disponibles:**
    - 2: "2 Personas"
    - 3: "3 Personas"
    - 5: "5 Personas"
  - **Valor por defecto:** 2
  - **Descripción:** Indica la cantidad de personas asociadas al mensaje.
3. **text**
  - **Tipo:** CharField
  - **Máxima longitud:** 512 caracteres
  - **Descripción:** Contenido principal del mensaje.
4. **updated**
  - **Tipo:** DateTimeField
  - **Valor por defecto:** null=True, blank=True
  - **Descripción:** Registra la fecha y hora de la última modificación del registro.
5. **created**
  - **Tipo:** DateTimeField
  - **Valor por defecto:** null=True, blank=True
  - **Descripción:** Registra la fecha y hora de la creación del registro.

---

#### Métodos

- **\_\_str\_\_()**
  - **Descripción:** Método que convierte el modelo en una cadena de texto legible.
  - **Salida:** El contenido del campo text.



- **Ejemplo de salida:**  
"Recordatorio: Confirmar asistencia"

## EventByDay

### Descripción

El modelo EventByDay se utiliza para registrar eventos asociados a fechas específicas y una cantidad específica de ocupantes. Este modelo es ideal para aplicaciones relacionadas con la planificación y organización de eventos según la ocupación.

---

## Atributos del Modelo

### Campos

(Sus atributos son idénticos a MessageByDay, pero destinados a eventos en lugar de mensajes).

1. **date\_from**
  - **Tipo:** CharField
  - **Descripción:** Fecha asociada al evento.
2. **occupancy**
  - **Tipo:** IntegerField
  - **Elecciones disponibles:**
    - 2: "2 Personas"
    - 3: "3 Personas"
    - 5: "5 Personas"
  - **Valor por defecto:** 2
3. **text**
  - **Tipo:** CharField
  - **Descripción:** Contenido o descripción del evento.
4. **updated**
  - **Tipo:** DateTimeField
5. **created**
  - **Tipo:** DateTimeField

---

### Métodos

- **\_\_str\_\_()**
  - **Salida:** Contenido del campo text.
  - **Ejemplo de salida:**  
"Evento especial para grupos grandes"

## CopyPriceWithDay

### Descripción

Este modelo relaciona precios con un objeto de reserva disponible (AvailableBooking), manejando precios para días específicos.

---

### Atributos del Modelo

#### Campos

1. **price**
    - **Tipo:** CharField
    - **Máxima longitud:** 30 caracteres
    - **Descripción:** Precio asociado a una reserva disponible.
  2. **created**
    - **Tipo:** DateField
    - **Valores permitidos:** null=True, blank=True
    - **Descripción:** Fecha de creación del precio.
  3. **avail\_booking**
    - **Tipo:** ForeignKey
    - **Relacionado con:** AvailableBooking
    - **Acción al eliminar:** CASCADE
    - **Descripción:** Enlace con el objeto de reserva disponible.
- 

#### Métodos

- **\_\_str\_\_()**
  - **Descripción:** Devuelve el precio en forma de cadena.
  - **Salida:** 120

## PriceWithNameHotel

### Descripción

Modelo utilizado para gestionar precios y detalles de hoteles en un rango de fechas, con información adicional como dirección, descripción, imágenes y ocupación.

---

### Atributos del Modelo

## Campos

1. **start**
  - **Tipo:** CharField
  - **Máxima longitud:** 20 caracteres
  - **Descripción:** Punto de inicio de referencia para el hotel.
2. **title**
  - **Tipo:** CharField
  - **Máxima longitud:** 512 caracteres
  - **Descripción:** Nombre del hotel.
3. **link**
  - **Tipo:** CharField
  - **Máxima longitud:** 3000 caracteres
  - **Descripción:** Enlace al detalle del hotel.
4. **address, distance, description, img, etc.**
  - **Tipo:** CharField
  - **Máxima longitud:** Variada según el atributo.
  - **Descripción:** Información relevante del hotel (dirección, distancia, descripción e imágenes).
5. **date\_from, date\_to**
  - **Tipo:** CharField
  - **Máxima longitud:** 30 caracteres
  - **Descripción:** Fechas de inicio y fin.
6. **price**
  - **Tipo:** CharField
  - **Máxima longitud:** 30 caracteres
  - **Descripción:** Precio asociado.
7. **occupancy**
  - **Tipo:** IntegerField
  - **Valor por defecto:** 2
  - **Descripción:** Cantidad de personas asociadas al precio.

## CopyPriceWithNameFromDay

### Descripción

Modelo para relacionar precios con información más detallada del hotel (PriceWithNameHotel).

---

### Atributos del Modelo

1. **price**
2. **created**
3. **avail**
  - Relación con el modelo PriceWithNameHotel.

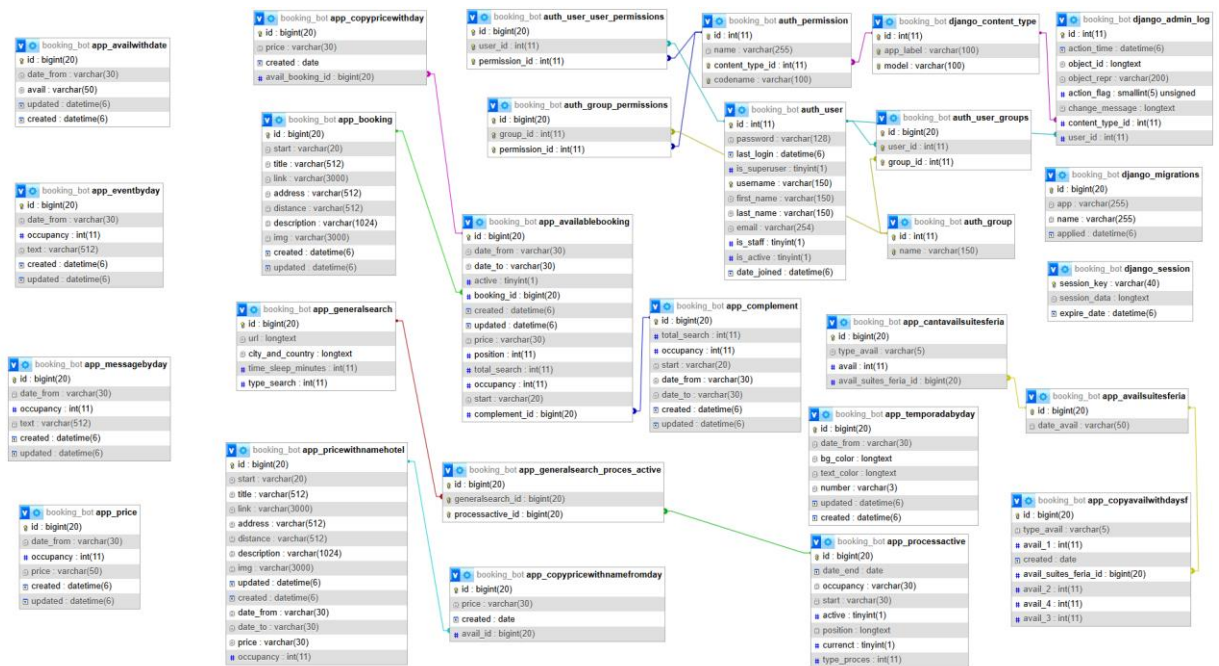
# CopyAvailWithDaySF

## Descripción

Modelo para gestionar disponibilidad específica por tipo y categorías para Suites de FERIA.

## Atributos del Modelo

1. **type\_avail**
2. **avail\_1, avail\_2, etc.**
3. **avail\_suites\_feria**



## Configuraciones search.

1. vamos al admin django.
2. buscamos la opcion General searchs.
3. agregar general search.
  - url: <https://www.booking.com/searchresults.es.html?>
  - city and country: Madrid, Comunidad de Madrid, España
  - time sleep minutes: 10 or > 0

## Configuraciones Process.

1. vamos al admin django.
2. buscamos la opcion Process actives.
3. agregamos los procesos.
  - Date end: 2024-09-15
  - Occupancy: 3
  - Start: 4
  - Position: [0, 1, 2, 3, 4, 9]
  - Active: Descheck
  - Current: Descheck
4. agregaremos 5 procesos iguales. al anterior cambiando algunos parametros.
  - Occupancy: 2
  - Start: 4
  - Position: [0, 1, 2, 3, 4, 9, 14, 19, 24]

- 
- Occupancy: 2
  - Start: 3
  - Position: [0, 1, 2, 3, 4]

- 
- Occupancy: 3
  - Start: 3
  - Position: [0, 1, 2, 3, 4]

- 
- Occupancy: 5
  - Start: 3
  - Position: [0, 1, 2, 3, 4]

- 
- Occupancy: 5

- Start: 4
- Position: [0, 1, 2, 3, 4, 9]