

Modelo: ServiciosMedicosEspacios

Tabla: tbc_servicios_medicos_espacios

Código:

```
# pylint: disable=too-few-public-methods, invalid-name, non-ascii-name
"""Modelo ORM para representar la asignación de espacios físicos a servicios médicos."""

import enum
import uuid
import datetime
from sqlalchemy import Column, String, ForeignKey, Enum, DateTime
from sqlalchemy.orm import relationship
from config.db import Base

class EstatusAprobacionEnum(str, enum.Enum):
    Pendiente = 'Pendiente'
    Aprobado = 'Aprobado'
    Rechazado = 'Rechazado'

class EstatusEnum(str, enum.Enum):
    Activo = 'Activo'
    Inactivo = 'Inactivo'

class ServiciosMedicosEspacios(Base):
    __tablename__ = 'tbc_servicios_medicos_espacios'

    id = Column(
        String(36),
        primary_key=True,
        index=True,
        default=lambda: str(uuid.uuid4())
    )
    fk_servicio = Column(
        String(36),
        ForeignKey("tbc_servicios_medicos.id", ondelete="CASCADE"),
        nullable=False,
        index=True
    )
    fk_espacio = Column(
        String(36),
        ForeignKey("tbc_espacios.id"),
        nullable=False,
        index=True
    )
    observaciones = Column(String(255), nullable=True)
    estatus_aprobacion = Column(
        Enum(EstatusAprobacionEnum),
        nullable=False,
```

```

        default=EstatusAprobacionEnum.Pendiente
    )
    estatus = Column(
        Enum(EstatusEnum),
        nullable=False,
        default=EstatusEnum.Activo
    )
    fecha_registro = Column(
        DateTime,
        default=datetime.datetime.utcnow
    )
    fecha_inicio = Column(DateTime, nullable=True)
    fecha_termino = Column(DateTime, nullable=True)
    fecha_ultima_actualizacion = Column(
        DateTime,
        nullable=True,
        onupdate=datetime.datetime.utcnow
    )

    servicio = relationship("ServiceM", back_populates="espacios")
    espacio = relationship("Espacio", back_populates="servicios_medicos_espacios")

```