



PROYECTO #2



Bryan España - 21550
Sebastián Estrada - 21405
Javier Prado - 21486

INTRODUCCIÓN



Las clínicas aún mantienen registros manuales en formatos físicos, lo que puede ser tedioso para médicos y pacientes y resultar en la pérdida de información importante. Además, el médico a menudo debe preguntar detalles de la historia clínica durante cada visita.

TECNOLOGIAS A UTILIZAR



Flask

FLASK

Flask es un framework para aplicaciones web para Python, utilizado para construir aplicaciones ligeras y medianas. Flask proporciona una variedad de características útiles, como rutas, sesiones, autenticación y plantillas HTML.



AMAZON WEB SERVICES

Es una plataforma de servicios en la nube que permite a los usuarios acceder y utilizar una amplia gama de servicios de computación en la nube, almacenamiento de datos, análisis de datos y más

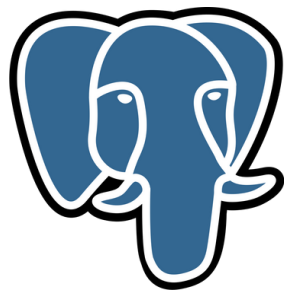




Amazon
RDS

AMAZON RELATIONAL DATABASE SERVICE

Amazon RDS facilita la configuración, el funcionamiento y el escalado de bases de datos relacionales en la nube, sin tener que preocuparse por la infraestructura subyacente.

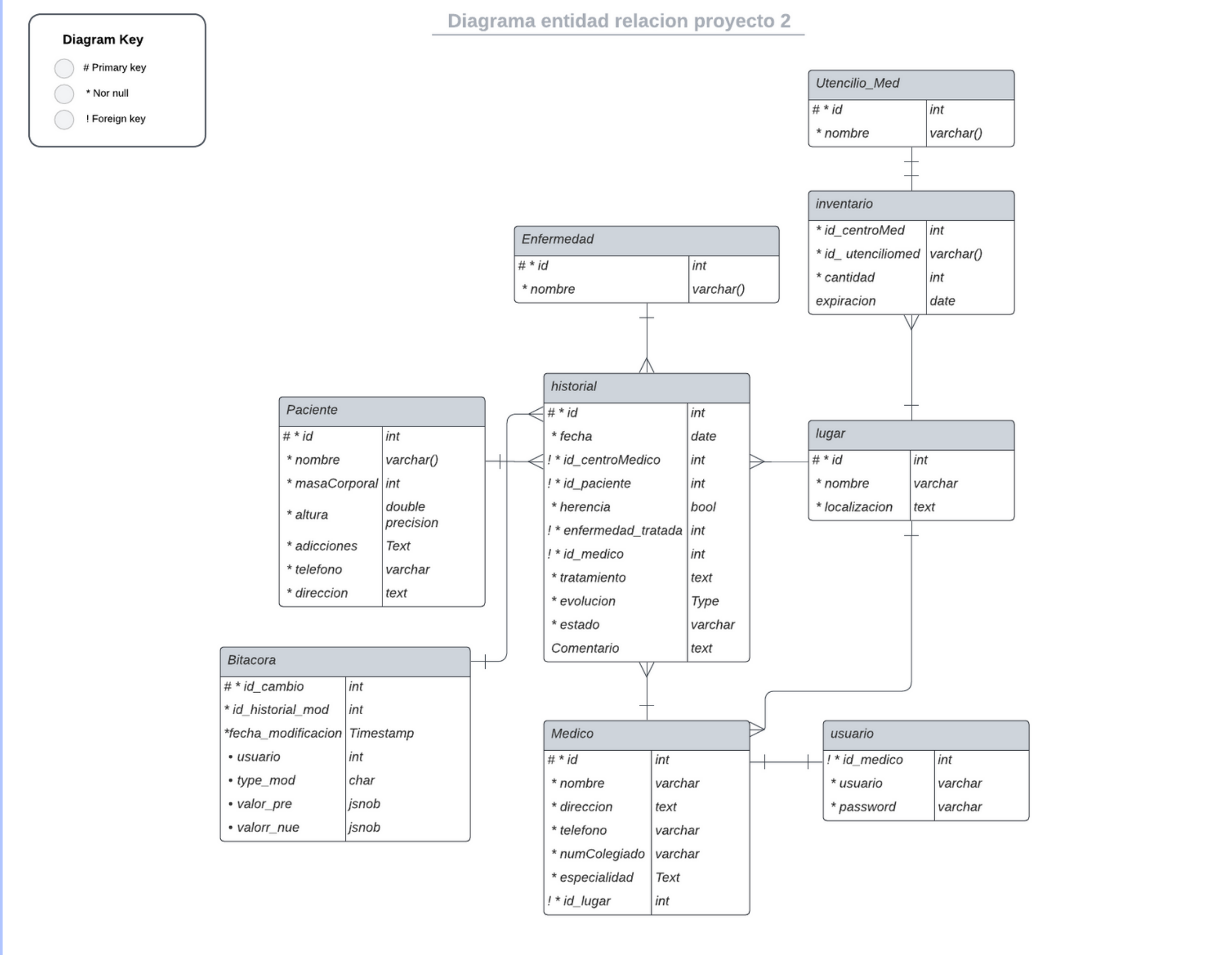


POSTGRESQL

es un sistema de gestión de bases de datos relacional de código abierto (open source) y gratuito que se utiliza para almacenar y gestionar grandes cantidades de datos de manera eficiente y segura.




Diagrama Entidad-Relacion



Funcion de Trigger

```
CREATE OR REPLACE FUNCTION historial_trigger() RETURNS TRIGGER AS $$
DECLARE
    jsonb_dataNew JSONB;
    jsonb_dataOld JSONB;
BEGIN
    IF (TG_OP = 'INSERT') THEN-----INSERT
        jsonb_dataNew = row_to_json(NEW.*);
        INSERT INTO bitacora (fecha_modificacion, id_historial_mod, type_mod, valor_nue)
        VALUES (CURRENT_TIMESTAMP, NEW.id_historial, 'I', jsonb_dataNew);
        RETURN NEW;
    ELSIF (TG_OP = 'UPDATE') THEN -----UPDATE
        jsonb_dataNew = row_to_json(NEW.*);
        jsonb_dataOld = row_to_json(OLD.*);
        INSERT INTO bitacora (fecha_modificacion, id_historial_mod, type_mod, valor_pre, valor_nue)
        VALUES (CURRENT_TIMESTAMP, OLD.id_historial, 'U', jsonb_dataOld, jsonb_dataNew);
        RETURN NEW;
    ELSIF (TG_OP = 'DELETE') THEN -----
DELETE
        jsonb_dataOld = row_to_json(OLD.*);
        INSERT INTO bitacora (fecha_modificacion, id_historial_mod, type_mod, valor_pre)
        VALUES (CURRENT_TIMESTAMP, OLD.id_historial, 'D', jsonb_dataOld);
        RETURN OLD;
    END IF;
END;
$$ LANGUAGE plpgsql;
```


Trigger



```
CREATE TRIGGER historial_changes  
AFTER INSERT OR UPDATE OR DELETE  
ON historial  
FOR EACH ROW  
EXECUTE PROCEDURE historial_trigger();
```