

## Actividad (40% nota asignatura)

### Parte 1: Creación del esquema de una base de datos orientada a columnas

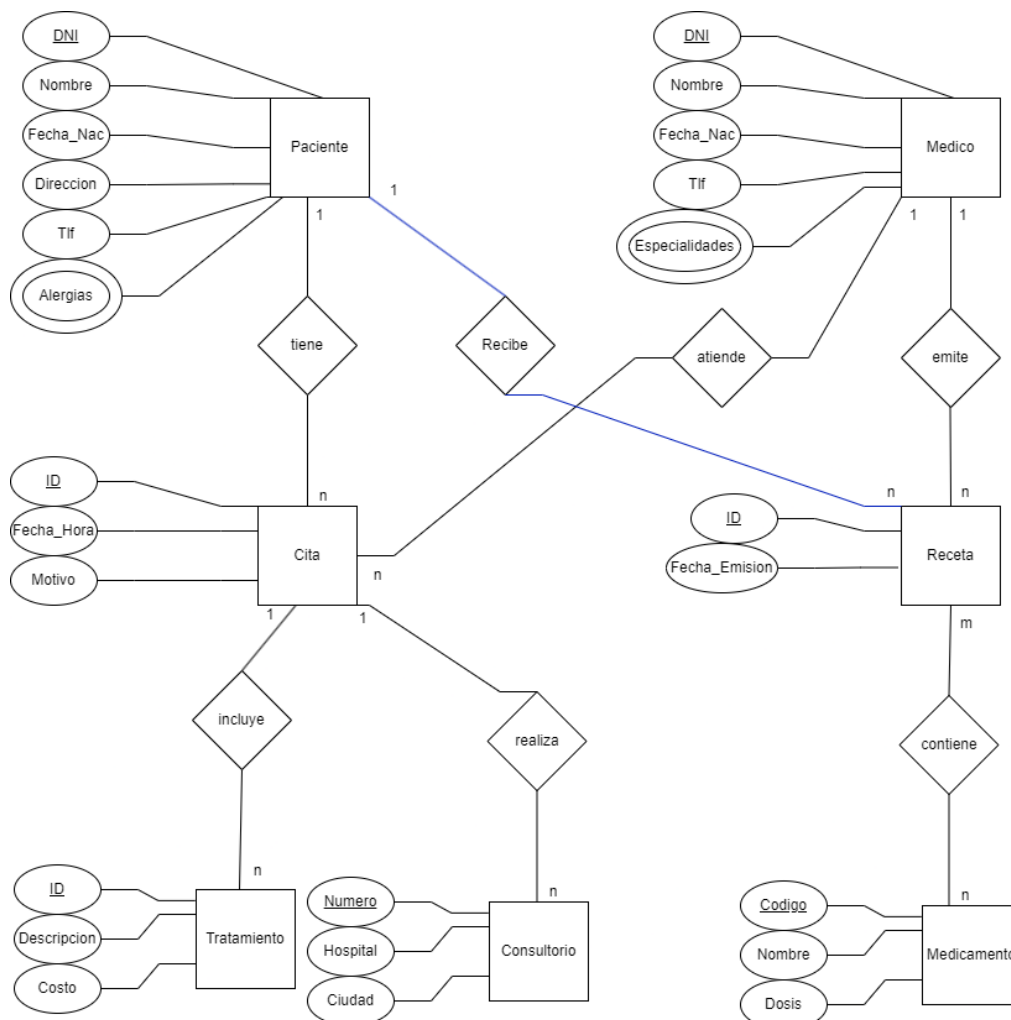
#### Descripción

Se pide realizar el esquema de una base de datos orientada a columnas que satisfaga una serie de consultas que se desean realizar sobre el modelo conceptual. La base de datos que usaremos como referencia será Apache Cassandra. Se pide una tabla para cada consulta en la que se debe especificar qué columnas serán clave primaria (partition key y clustering key). Las columnas deberán ser nombradas con el convenio siguiente: NombreEntidad\_NombreAtributo. Es decir, si queremos crear una columna asociada a un atributo "DNI" de una entidad "Usuario", la columna resultante sería "Usuario\_DNI". **En caso de no cumplir dicha norma se procederá a una penalización por cada incumplimiento, llegando a una penalización máxima de 1,5 puntos a restar de la calificación final obtenida.**

Las tablas deberán ser nombradas como Tabla1, Tabla2, Tabla3... siendo el número el que corresponda a la consulta.

#### Modelo conceptual

Los atributos subrayados son clave primaria de la entidad. Los atributos con elipses representan atributos de conjunto.



## Consultas a satisfacer

1. Obtener toda la información de un paciente en base a su nombre.
2. Obtener según el DNI de un médico todas las citas que atiende.
3. Obtener según el DNI de un paciente, todos los tratamientos que tiene incluidos en las citas que tuvo.
4. Obtener cuantas citas tiene un paciente.
5. Obtener todas las asociaciones entre recetas y medicamentos a través de la fecha de la receta.
6. Obtener los pacientes que tienen una alergia en concreto

**Nota: Puede haber entidades y relaciones que no participen en ninguna de las consultas.**

Se requiere en cada tabla que satisfaga la consulta que se implemente, como mínimo, las columnas necesarias para satisfacer la consulta (tanto clave primaria como columnas no clave a consultar). Además, se deben incluir las columnas necesarias para garantizar la unicidad de la tabla.

Al seleccionar las claves primarias será necesario diferenciar entre partition key (PK) y clustering key (CK). Se valorará positivamente una elección justificada y correcta de estas claves. Se deberá justificar en cada tabla la elección de las claves primarias con un texto, así como la de cualquier columna que no haya sido explícitamente pedida en la consulta pero que se considere necesaria por cualquier motivo. **En el caso de no incluirse explicación o en el caso de ser muy escueta solo exponiendo de forma escrita lo que se diseña sin justificarse, se calificará dicha consulta con 0 puntos.**

**En el caso de detectarse un mismo diseño y una misma justificación o muy similar en dos estudiantes, se procederá a la calificación de 0 en la actividad además de tomarse posteriormente las sanciones pertinentes, incluyendo el suspenso de la convocatoria.**

## Parte 2: Creación de tablas (10%)

Cree una base de datos Cassandra que implemente las tablas diseñadas en la Parte 1. El nombre del keyspace debe ser su nombre y su primer apellido en minúsculas. Por ejemplo, si usted se llama Juan Méndez, el nombre del keyspace será "juanmendez". Las columnas de las tablas serán todas de tipo text excepto las que se encuentren asociadas a los siguientes atributos del modelo conceptual:

- Paciente
  - DNI: Int
  - Fecha\_Nac: date
  - Alergias: set<text>
- Paciente
  - DNI: Int
  - Fecha\_Nac: date
  - Especialidades: set<text>
- Cita
  - ID: Int
  - Fecha\_Hora: date
- Receta
  - ID: Int
  - Fecha\_Emision: date

- Tratamiento
  - ID: Int
  - Costo: float
- Consultorio
  - nro: Int
- Tipo\_Boleto
  - Numero: Int
- Medicamento
  - Codigo: Int

\*Recuerde que cuando un atributo conjunto tenga que ser clave primaria de una tabla, este deberá implementarse como una columna de tipo individual.

Cada table deberá ser nombrada como Tabla#, siendo # el número de la consulta de la Parte 1 que satisface. En el caso de nombrar a las tablas de otra forma, se descontará la mitad de lo obtenido en este apartado.

### Parte 3: Programa Python de gestión de datos (40%)

Se pide realizar un programa que tenga las siguientes funciones o código implementado:

#### Clases por entidad o relación (5%):

Por cada entidad o relación n:m se deberá tener una clase de Python tal y como se explicó en la Práctica 2. Se recomienda usar estas clases para guardar la información en el resto de la práctica.

#### Creación de métodos destinados a consultar la información de tablas soporte (5%)

Se deberán crear tablas soporte para consultar información de las siguientes entidades a través de su clave primaria:

- La entidad Paciente
- La entidad Médico

En el fichero cql a entregar (Parte 2) deberá constar la creación de estas tablas. El nombre de estas tablas será SoportePaciente y SoporteMedico.

#### Creación de métodos de inserción de datos (15%)

Debe crear funciones para insertar instancias de entidades o relaciones en las tablas que se especifiquen.

Se piden funciones para insertar instancias de las siguientes entidades. Se indican las consultas de las que se crearon las tablas donde se debe insertar la información:

- Paciente en tabla de consulta 1, 6 y soporte.
- Médico en tabla de consulta 2 y soporte.

Se piden funciones para insertar las siguientes relaciones entre instancias en todas las tablas necesarias:

- Relación Tiene.
- Relación Contiene.

Para determinar dónde insertar la información de una relación se sigue el siguiente criterio:

**La tabla deberá incluir exclusivamente columnas asociadas a atributos de las entidades relacionadas.**

Nota 1: Pueden quedar tablas sin datos a insertar si tienen como objetivo almacenar información que no se requiere en este ejercicio.

Nota 2: Se deben usar consultas a tablas soporte cuando sea posible. En el resto de los casos toda la información podrá ser insertada por el usuario no siendo necesario el mantenimiento de la integridad de los datos en estas inserciones (inconsistencias entre la información almacenada en diferentes tablas). Se permite, si el alumno lo desea, la implementación de tablas soporte adicionales a las requeridas.

#### Creación de funciones de actualización de datos (5%)

Se deberán crear funciones que realices las siguientes tareas:

- Actualizar la dirección de un paciente según su DNI en la tabla de la consulta 1. (10%)

Nota: Se deben usar consultas a tablas soporte cuando sea necesaria.

#### Creación de funciones de borrado de datos (2,5%)

Se deberán crear funciones que realices las siguientes tareas:

- Borrar la relación entre receta y medicamento en base a la fecha de la receta en la tabla de la consulta 5.

#### Creación de funciones de consulta de información general (5%)

Realice funciones en las que se pueda consultar la información de las tablas y en los que se pueda filtrar la información a través de una cláusula WHERE según los requisitos originales de las consultas. Se debe recibir del usuario los valores por los que se quiere buscar según lo especificado en la consulta. Si el enunciado de la Actividad 1 no especifica la búsqueda de información a través de un atributo en concreto, se deberá buscar por la partition key de la tabla. En cada consulta se debe mostrar la información de al menos las columnas clave y al menos dos columnas no clave en el caso de que las hubiera.

#### Creación de interfaz de interacción de usuario (2,5%)

Debe crear una interfaz de consola que le pida al usuario realizar alguna de las operaciones programadas en las funciones creadas en los distintos apartados o salir de la aplicación pulsando 0. La elección se realizará a través de un número introducido por el usuario empezando por el 1. Se debe explicar al usuario la operación que se realizará a través de un mensaje por pantalla.

#### Sobre penalizaciones con respecto a malos diseños de las tablas

En general no se penalizará que el código trabaje sobre tablas con un diseño incorrecto. Solo se penalizará en el caso de que las tablas diseñadas faciliten de manera excesiva la realización de la parte 3 evitando el uso de técnicas de manipulación de datos vista en clase.

**NOTA IMPORTANTE: Se comprobarán los códigos entregados por los alumnos usando un software anti plagio. Si se detecta que dos códigos son copiados, se procederá a la calificación de 0 suspenso de la actividad, además de las sanciones posteriores pertinentes.**

#### Entrega

Se deberán subir los siguientes archivos en la tarea de la actividad:

Parte 1: Se deberá subir al campus virtual un archivo PDF en el que se especifiquen las tablas para cada una de las consultas además de la justificación del diseño de esta. Se recomienda que se creen tablas similares a las usadas en las transparencias de clase para la entrega.

Parte 2: Archivo cql que contenga la creación de la estructura de la base de datos con la creación del keyspace y de las tablas.

Parte 3: Programa python con las funciones necesarias implementadas y listo para su funcionamiento. Es importante que la extensión del archivo entregado sea .py. Si se entrega otro formato que necesite de un software específico para ser analizado, el profesor se reserva el derecho a no valorar la entrega.

**La fecha de entrega es el 18 de noviembre de 2024 a las 23:59h.**