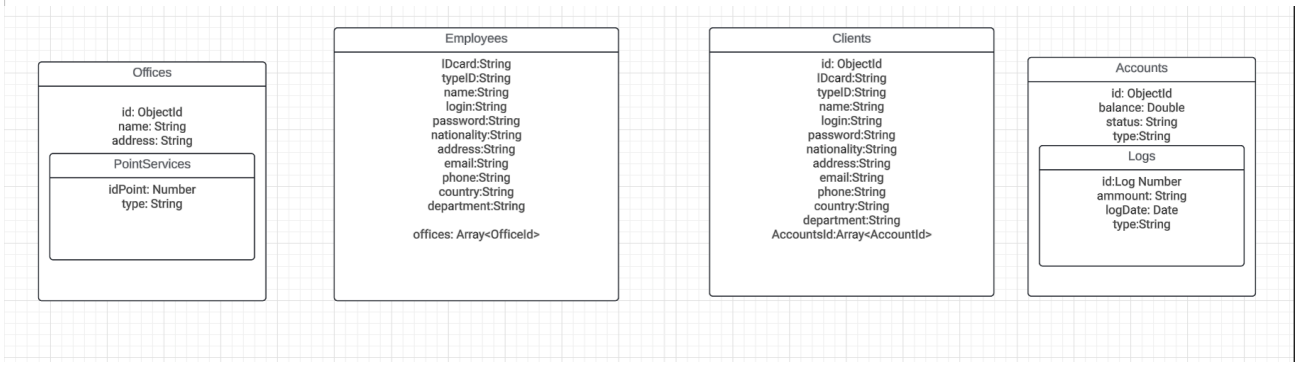
1. Análisis y modelo conceptual
a. Proponga un modelo conceptual en UML o E/R



2.

I. Offices: La colección de oficinas, donde cada archivo contiene:
   Id: ObjectId
   Name: String
   Address:String
   PointServices(Embedded){idPoint:Number, type:String}

II. Employees: La colección de empleados, donde cada archivo contiene:
IDcard:String
typeId.String
   Name: String
   Login:String
   Password:String
   Nationality:String
   Address:String
   Email:String
   Email:String
   Phone:String
   Country:String

Department:String
Offices: Array<Officeid>

III. Clients: La colección de clientes, donde cada archivo contiene:
id:objectId
IDcard: String
typeID:String
name:String
login:String
password:String
nationality:String
address:String
email:String
phone:String
country:String
department:String
AccountsID:array<Accountid>
IV.Accounts: La colección de cuentas donde cada documento contiene:
id:ObjectId
balance: double
status:string
type:string
Logs (Embedder){id: LogNumber, ammount:string, logDate:date, type:string}

B.Cuantificación:
I.Offices: 4 cada una con entre 2 y 5 puntos de servicio
II Employees: 3
III Clients: 7
IV Accounts: 2 cada uno con 15 logs aproximadamente

a. Operaciones de lectura y escritura para cada entidad y cuantificación de operaciones

Entidades	Operaciones	Información necesitada	Tipo	Tasa
Employees	Agregar un empleado	Información del empleado correspondiente	Escritura	1400/semana
Clients	Agregar un cliente	Información del cliente correspondiente	Escritura	1400/semana
Office	Agregar una oficina	Información de la oficina correspondiente	Escritura	1/mes
Office	Agregar un punto de atención a una oficina	Información del punto de atención correspondiente	Escritura	1/mes
Accounts	Agregar una cuenta	Información de la cuenta correspondiente	Escritura	500/dia
Accounts	Añadir un movimiento de una cuenta	Balance actual de la cuenta	Escritura	500/dia
Office	Eliminar un punto de atención	Id punto de atención a eliminar y oficina de la cual se va a eliminar	Borrado	1/mes
Accounts	Cambiar estado de cuenta	Estado actual de la cuenta y estado al que se quiere llevar	Actualizar	500/dia
Accounts	Consultar las cuentas	Informción de todas las cuentas registradas	Lectura	5000/dia
Accounts	Consultar el estado bancario de una cuenta	Id cuenta a buscar	Lectura	5000/dia
Employees	Agregar un empleado	Información del empleado correspondiente	Escritura	N/A
Clients	Agregar un cliente	Información del cliente correspondiente	Escritura	N/A
Office	Agregar una oficina	Información de la oficina correspondiente	Escritura	1/mes
Office	Agregar un punto de atención a una oficina	Información del punto de atención correspondiente	Escritura	1/mes
Accounts	Agregar una cuenta	Información de la cuenta correspondiente	Escritura	500/dia
Accounts	Añadir un movimiento de una cuenta	Balance actual de la cuenta	Escritura	500/dia
Office	Eliminar un punto de atención	Id punto de atención a eliminar y oficina de la cual se va a eliminar	Borrado	1/mes
Accounts	Cambiar estado de cuenta	Estado actual de la cuenta y estado al que se quiere llevar	Actualizar	500/dia
Accounts	Consultar las cuentas	Informción de todas las cuentas registradas	Lectura	5000/dia

Accounts	Consultar el estado bancario de una cuenta	Id cuenta a buscar	Lectura	5000/dia

2. Describan las entidades de datos y las relaciones entre ellas que corresponden al modelo conceptual UML propuesto.
    a. La lista de entidades con la descripción de cada una de ellas.

Entidad	Descripción
Room	Entidad que modela las habitaciones en la aplicación, una habitación puede o no estar ocupada y tiene un precio definido, así mismo pertenece a un conjunto de habitaciones que comparten características
RoomType	Entidad que modela los tipos de habitación, los tipos de habitación son agrupaciones que definen el tipo, las características o amenidades y la capacidad de un conjunto de habitaciones que pertenecen a dicho tipo
Service	Entidad que modela los servicios de la aplicación, estos tienen un nombre y una descripción.
Customer	Entiodad que se encarga de modelar los clientes de una aplicación. Cada cliente tiene: DNI,nombre, email. Igualmente, cada cliente debe tener al menos una reserva asociada *Booking*, con el fin de guar*dar los datos de la reserva que hizo en el hotel*
Booking	Esta entidad se encarga de modelar toda la información relacionada a una reserva, por ello dentro de bookings esta: status: Representa el estado actual de la reserva, indicando si está confirmada, pendiente, cancelada. People, indica la cantidad de personas asociadas a la reserva. initial_date, Es la fecha de inicio la reserva. final_date, Es la fecha de finalización de la reserva. Companions,personas adicionales relacionadas con la reserva. ConsumeService, Son los servicios consumidos por de la reserva.

    b. Las relaciones entre entidades y su cardinalidad.

Relación A	Relación B	Cardinalidad A - B
Offices	*PointServices*	Uno a Muchos (1...*)
Offices	*Employees*	Uno a Muchos (1...*)
Clients	*Account*	Uno a Muchos (1...*)
Account	*Logs*	Uno a Muchos (1...*)

c. El análisis de selección de esquema de asociación (referenciado o embebido) para cada relación entre entidades.

Relación A	Relación B	Esquema de asociación
Offices	*PointServices*	Embebido
Offices	*Employees*	Referenciado
Clients	*Account*	Referenciado
Account	*Logs*	Embebido

i. Offices→PointServices

Criterio	Pregunta	Embeber	Referenciar
Sencillez	¿Mantener la información junta simplificaría el modelo de datos y el código?	Si	No
Pertenencia	¿La información tiene una relación "tiene", "contiene" o similar?	Sí	No
Atomicidad de la consulta	¿La aplicación consulta la información junta?	No	No
Complejidad de actualización	¿La información se actualiza junta?	No	No
Archivo	¿La información debe archivarse al mismo tiempo?	Si	SI
Cardinalidad	¿Hay una cardinalidad alta (actual o creciente) en el lado del hijo de la relación?	Si	Si
Duplicación de datos	¿La duplicación de datos sería demasiado complicada de administrar y no deseada?	No	Si
Tamaño del documento	¿El tamaño combinado de la información ocuparía demasiada memoria o ancho de banda de transferencia para la aplicación?	No	Si
Crecimiento del documento	¿La pieza incrustada crecerá sin límite?	NO	NO
Carga de trabajo	¿La información se escribe en diferentes momentos en una carga de trabajo intensiva?	NO	NO
Individualidad	Para el lado de los hijos de la relación, ¿pueden existir las piezas por sí mismas sin un padre?	NO	NO

ii. Offices→Employees

Criterio	Pregunta	Embeber	Referenciar
Sencillez	¿Mantener la información junta simplificaría el modelo de datos y el código?	No	Si
Pertenencia	¿La información tiene una relación "tiene", "contiene" o similar?	No	No
Atomicidad de la consulta	¿La aplicación consulta la información junta?	No	No
Complejidad de actualización	¿La información se actualiza junta?	No	No
Archivo	¿La información debe archivarse al mismo tiempo?	Si	No
Cardinalidad	¿Hay una cardinalidad alta (actual o creciente) en el lado del hijo de la relación?	No	Sí
Duplicación de datos	¿La duplicación de datos sería demasiado complicada de administrar y no deseada?	Si	No

Tamaño del documento	¿El tamaño combinado de la información ocuparía demasiada memoria o ancho de banda de transferencia para la aplicación?	Si	No
Crecimiento del documento	¿La pieza incrustada crecerá sin límite?	Si	No
Carga de trabajo	¿La información se escribe en diferentes momentos en una carga de trabajo intensiva?	Si	Sí
Individualidad	Para el lado de los hijos de la relación, ¿pueden existir las piezas por sí mismas sin un padre?	Si	No

iii. Clients→Account

Criterio	Pregunta	Embeber	Referenciar
Sencillez	¿Mantener la información junta simplificaría el modelo de datos y el código?	No	Si
Pertenencia	¿La información tiene una relación "tiene", "contiene" o similar?	Si	SI
Atomicidad de la consulta	¿La aplicación consulta la información junta?	No	Si
Complejidad de actualización	¿La información se actualiza junta?	No	No
Archivo	¿La información debe archivarse al mismo tiempo?	Si	No
Cardinalidad	¿Hay una cardinalidad alta (actual o creciente) en el lado del hijo de la relación?	No	No
Duplicación de datos	¿La duplicación de datos sería demasiado complicada de administrar y no deseada?	Si	No
Tamaño del documento	¿El tamaño combinado de la información ocuparía demasiada memoria o ancho de banda de transferencia para la aplicación?	Si	No
Crecimiento del documento	¿La pieza incrustada crecerá sin límite?	SI	No
Carga de trabajo	¿La información se escribe en diferentes momentos en una carga de trabajo intensiva?	SI	Sí
Individualidad	Para el lado de los hijos de la relación, ¿pueden existir las piezas por sí mismas sin un padre?	NO	No

iv. Account→Logs

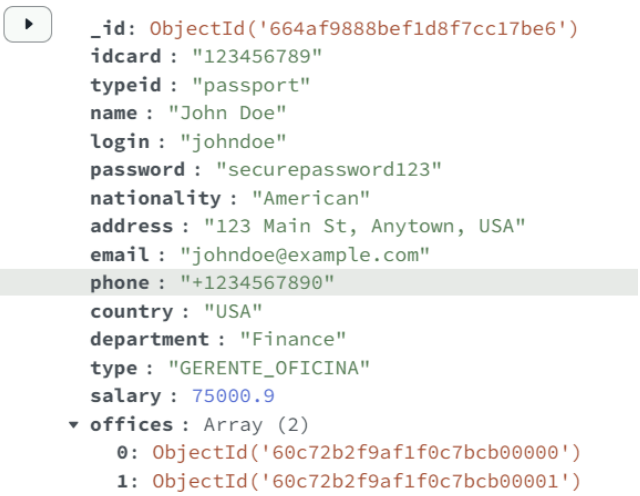
Criterio	Pregunta	Embeber	Referenciar
Sencillez	¿Mantener la información junta simplificaría el modelo de datos y el código?	Si	No
Pertenencia	¿La información tiene una relación "tiene", "contiene" o similar?	Si	NO
Atomicidad de la consulta	¿La aplicación consulta la información junta?	Si	SI
Complejidad de actualización	¿La información se actualiza junta?	SI	NO
Archivo	¿La información debe archivarse al mismo tiempo?	Si	SI
Cardinalidad	¿Hay una cardinalidad alta (actual o creciente) en el lado del hijo de la relación?	SI	SI
Duplicación de datos	¿La duplicación de datos sería demasiado complicada de administrar y no deseada?	NO	SI
Tamaño del documento	¿El tamaño combinado de la información ocuparía demasiada memoria o ancho de banda de transferencia para la aplicación?	SI	NO

Crecimiento del documento	¿La pieza incrustada crecerá sin límite?	SI	NO
Carga de trabajo	¿La información se escribe en diferentes momentos en una carga de trabajo intensiva?	NO	Sí
Individualidad	Para el lado de los hijos de la relación, ¿pueden existir las piezas por sí mismas sin un padre?	NO	SI

d. Una descripción gráfica usando JSON de cada relación entre entidades en donde presente un ejemplo de datos junto con el esquema de asociación usado (referenciado o embebido).

i. *Offices, point_services*

ii. *Employee*



i. *Client*


```
_id: ObjectId('664ae2ba8bef1d8f7cc17bd9')
IDcard : "123456789"
typeID : "CC"
name : "Maria Lopez"
login : "maria123"
password : "securepassword"
nationality : "Colombian"
address : "Calle 100 #20-30, Bogotá, Colombia"
email : "maria.lopez@example.com"
phone : "+57 310 234 5678"
country : "Colombia"
department : "Marketing"
accounts : Array (1)
  0: "6653ab49c0e6ecc1ae8a4f5b"
```


ii. *Account - Logs*


```
_id: ObjectId('6653ab49c0e6ecc1ae8a4f5b')
balance : 700
status : "Activar"
type : "ahorros"
log_accounts : Array (20)
  0: Object
    idLog : 1
    ammount : 500.1
    logdate : 2024-01-15T10:00:00.000+00:00
    typelog : "deposit"
  1: Object
    idLog : 2
    ammount : 1000.1
    logdate : 2024-01-16T11:00:00.000+00:00
    typelog : "deposit"
  2: Object
    idLog : 3
    ammount : 11
    logdate : 2024-05-26T21:59:10.991+00:00
    typelog : "Retiro"
```