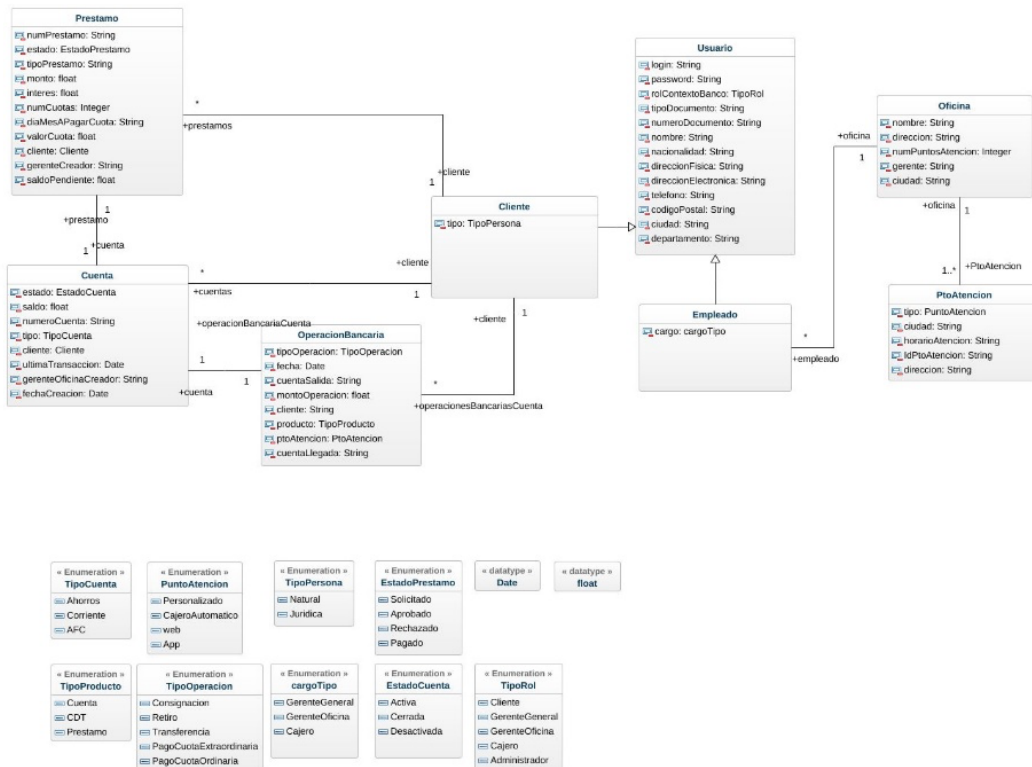
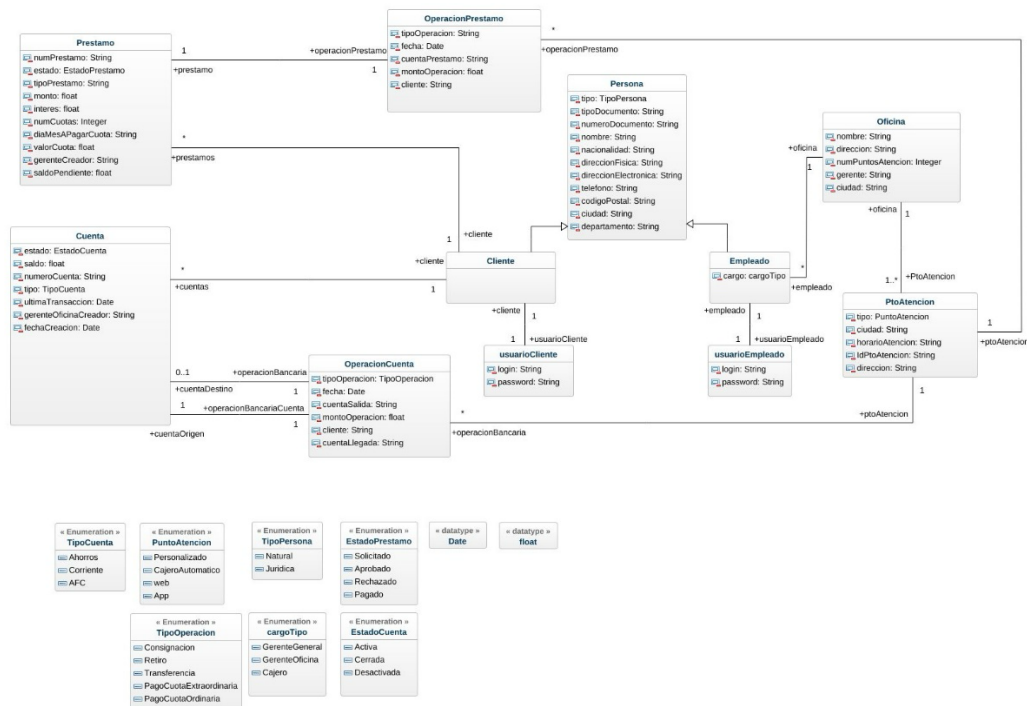


DOCUMENTO INFORME

1. DIAGRAMA UML

El UML tuvo varios cambios. EL primer cambio que hicimos es agregar una nueva clase al modelo OperacionPrestamo para representar las operaciones sobre los préstamos. También quitamos la relación que tenía préstamo con cuenta pues préstamo está relacionado con cliente y cliente tiene una cuenta asociada, entonces así podríamos determinar la cuenta del préstamo. El mayor cambio que hicimos es el cambio de la clase usuario, cliente y empleado. Ahora, dividimos la clase usuario en varias tablas. Ahora tenemos la clase persona, de la que heredan las clases empleado y cliente y estas se relacionan con usuarioCliente y usuarioEmpleado. De esta forma todo estaría normalizado también.

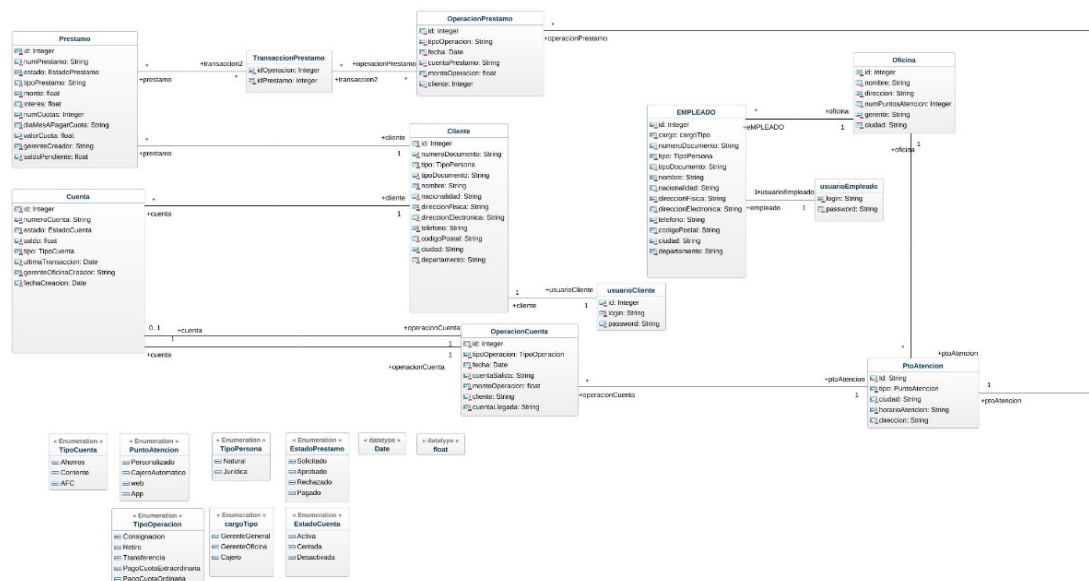




2. UML ENTREGA 2

2. DIAGRAMA UML BASE DE DATOS

Este UML representa es representativo de la base de datos, es mas para representar como hicimos las relaciones entre las clases, siguiendo el ejemplo del proyecto guiado parranderos. Al final se nos dificultó representar la herencia, por lo que decidimos eliminar la clase persona y tener una clase para clientes y otra para empleados.



3.Normalización del model

Tabla cuentas

-1FN: Cuenta se encuentra en primera forma normal, porque no hay grupos repetidos en la tabla.

-2FN: Primero, la tabla se encuentra en 1FN y cada atributo depende completamente de la PK compuesta y no hay dependencias parciales.

-3FN: La relación ya se encuentra en 2FN y no hay dependencias transitivas.

Se encuentra en FNBC porque los únicos determinantes son las claves candidatas

Tabla préstamos

-Cumple la primera forma normal pues no se repiten valores en la tabla, por ejemplo, cada préstamo tendrá un solo estado y tipo de préstamo.

-2FN: Ya se encuentra en 1FN y también se encuentra en 2FN ya que los atributos dependen completamente de la PK numPrestamo depende completamente de la PK

-3FN: Ya se encuentra en 1Fn y 2FN. Se puede decir que también se encuentra en 3FN porque no hay relaciones transitivas entre los atributos de las tablas. Por ejemplo, numPrestamo determina estado, pero estado no determina otros atributos y si seguimos analizándolo nos damos cuenta de que no hay relaciones transitivas. Finalmente, está en forma BC porque no existen varias llaves candidatas y no hay dependencias parciales entre atributos primos.

Se encuentra en FNBC

Tabla Operación Bancaria

Esta tabla consideramos que no estaba normalizada por lo que decidimos dividirla en 2 tablas.

-1FN: La cumple ya que no existen atributos multivalor y además el dominio de todos los atributos es atómico.

-2FN: Ya cumple con la 1FN,y pero los valores no dependen de una única llave primaria pues dependían de las llaves id y tipo operación. Para solucionar esto decidimos dividir en dos tablas operación bancaria y ahora tenemos operaciones_cuentas y operaciones_prestamos. Donde en cada tabla la llave primaria es id y el resto de los atributos dependen únicamente de esta PK.

-3FN: Ya que se encuentra en 1FN y 2FN, se puede decir que las tablas nuevas operaciones_cuentas y operaciones_prestamos se encuentra en 3FN ya que no hay ninguna dependencia funcional transitiva entre los atributos, o sea los demás atributos no dependen de otros que no sean llaves la PK

Cumple con FNBC ya que no hay dependencias entre los atributos primos y tan solo existe una llave tanto para operaciones_cuentas como para operaciones_prestamos

Tabla Punto de atención

-1FN: Cumple con la 1FN ya que cada atributo no es multivalor, por ejemplo el tipo de operación no puede ser multivalor ya que no sería conciso con su Id de punto de atención, no pueden haber 2 ciudades con el mismo Id de punto de atención, ya que cada punto tiene solo un horario de atención no existe la posibilidad de que sean varios, y que un punto de atención tenga varias direcciones.

-2FN: Ya cumple con la 1FN y además el único atributo primo es IdPtoAtencion por tanto todos los demás dependen de manera directa a este.

-3FN: Cumplio con la 1FN y además la 2FN, y es evidente que no existen dependencias entre atributos no primos.

Cumple con FNBC ya que no hay dependencias entre los atributos primos y tan solo existe una llave

Tabla oficinas

1FN: La tabla se encuentra en primera forma normal, porque no hay atributos multivalor.

2FN: Está en 1FN y consideramos que está en 2FN porque los atributos dependen únicamente de la llave primaria.

3FN: Se encuentra en 2FN y en 3FN porque no hay transitivas.

Nota: Al principio decidimos modelar esto como un parranderos definiendo clases rompedoras para hacer mas fácil el modelamiento pero al presentarse un error muy raro en el spring decidimos desistir aunque nos comunicamos con varias personas no se logro solucionar este, las pruebas de lo hecho son este UML realizado junto con el link de ese repositorio donde se modelo este problema de la siguiente manera: <https://github.com/Marcosespa/PSistrans.git> Además adjunto el error persistente dicho anteriormente org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'entityManagerFactory' defined in class path resource [org.springframework.boot.autoconfigure.orm.jpa.HibernateJpaConfiguration.class]: Could not

