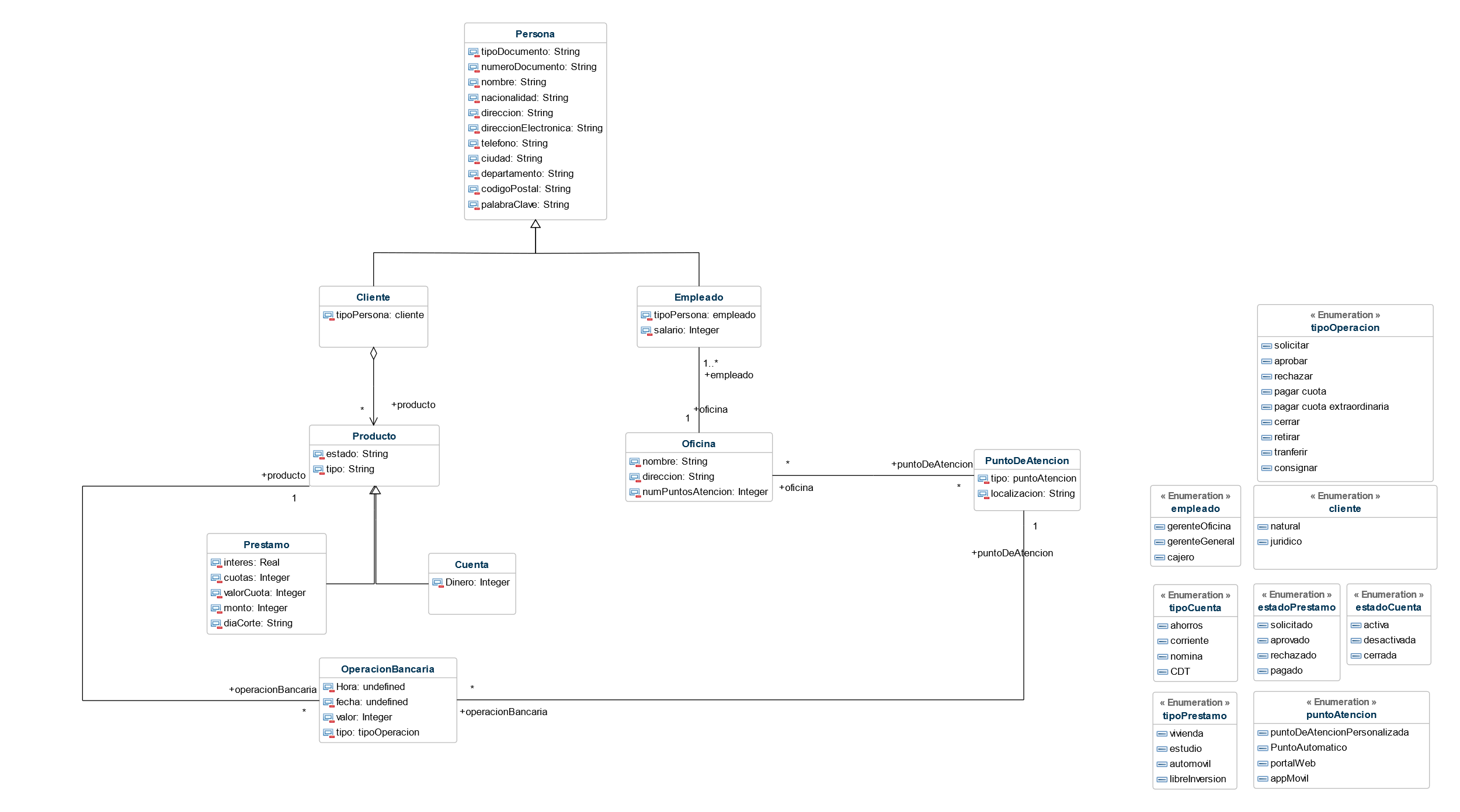
**Proyecto 1 SISTRANS**

**Martin Vasquez – 202113314**

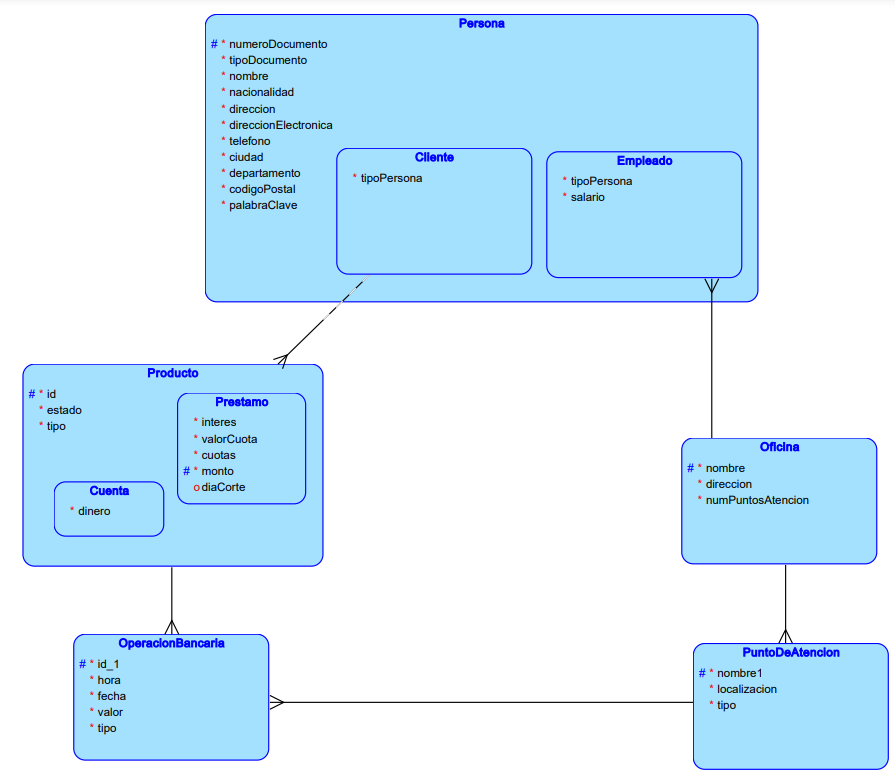
**Martin del Gordo – 202024239**

**Laura Valentina Lara - 201912967**

**Diagrama de dominio:**

**\*se actualiza el diagrama para tener en cuenta las cardinalidades mencionadas en la corrección.**

**Diagrama E-R:**

****

**\*se acualiza el modelo E-R para arreglar las direcciones de la obligatoriedad.**

**Modelo relacional:**

**Tabla Personas**

Se separa de sus hijos en 3 tablas por la relación de herencia.

**Relaciones:**

**Llave candidata: |id|**

**Primos:**

**Forma normal 1**: Cumple, ya que todos los atributos que conforman la tabla son simples. Es decir, cada columna no se puede descomponer en más de 1 elemento.

**Forma normal 2**: Cumple, ya que solo hay u atributo primo (id) y no existen dependencias parciales dependencias.

**Forma normal 3**: Cumple, ya que ningún atributo no primo determina a otro no primo. En otras palabras, no hay dependencias entre atributos no primos.

**Forma normal BC**: Cumple, ya que la tabla se encuentra en forma normal 3 y la llave de esta tabla es simple, por lo que no hay dependencias parciales de esta.



**Tabla Clientes**

**Relaciones:**

* **ND>TD,NO,NA,DR,DE,TL,PC,TC**
* **DR-> CP, CD, DP**

**Llaves candidatas: |ND,DR|-> TD,NO,NA,DR,DE,TL,PC,TC, CP, CD, DP**

**Primos: ND,DR**

**No primos: TD,NO,NA,DR,DE,TL,PC,TC, CP, CD, DP**

**Forma normal 1**: Cumple, ya que todos los atributos que conforman la tabla son simples. Es decir, cada columna no se puede descomponer en más de 1 elemento.

**Forma normal 2**: No cumple, hay dependencias parciales de uno de los atributos primos.

Se decide romper la tabla en 2: Cliente y Dirección. La primera guardara los siguientes atributos: **ND, TD,NO,NA,DR,DE,TL,PC,TC**; la segunda: **DR, CP, CD, DP.**

**Forma normal 3:** Cumple, pues no existen dependencias o relaciones entre los atributos no primos.

**Forma normal BC:** Cumple, pues no existe ninguna dependencia trivial entre los atributos primos.

**Para la tabla cliente:**

**Forma normal 1**: Cumple, ya que todos los atributos que conforman la tabla son simples. Es decir, cada columna no se puede descomponer en más de 1 elemento.

**Forma normal 2**: Cumple debido a que solo existe un atributo primo, por lo que la llave no puede determinar parcialmente a otro atributo.

**Forma normal 3**: Cumple, ya que ningún atributo no primo determina a otro no primo. En otras palabras, no hay dependencias entre atributos no primos.

**Forma normal BC**: Cumple, ya que la tabla se encuentra en forma normal 3 y la llave de esta tabla es simple, por lo que no hay dependencias parciales de esta.

****

**Para la tabla dirección:**

**Forma normal 1**: Cumple, ya que todos los atributos que conforman la tabla son simples. Es decir, cada columna no se puede descomponer en más de 1 elemento.

**Forma normal 2**: Cumple debido a que solo existe un atributo primo, por lo que la llave no puede determinar parcialmente a otro atributo.

**Forma normal 3**: No cumple, pues existen dependencias entre los atributos no primos.

Se decide romper la tabla en 2: direcciones y códigos postales. La primera con los siguientes atributos: **ID, DR, CP** y la segunda con **CP,CU,DP**

**Forma normal BC:** Cumple, ya que la tabla se encuentra en forma normal 3 y la llave de esta tabla es simple, por lo que no hay dependencias parciales de esta.

**Para la tabla direcciones:**

**Forma normal 1**: Cumple, ya que todos los atributos que conforman la tabla son simples. Es decir, cada columna no se puede descomponer en más de 1 elemento.

**Forma normal 2**: Cumple debido a que solo existe un atributo primo, por lo que la llave no puede determinar parcialmente a otro atributo.

**Forma normal 3**: Cumple, ya que ningún atributo no primo determina a otro no primo. En otras palabras, no hay dependencias entre atributos no primos.

**Forma normal BC**: Cumple, ya que la tabla se encuentra en forma normal 3 y la llave de esta tabla es simple, por lo que no hay dependencias parciales de esta.



**para la tabla códigos postales:**

**Forma normal 1**: Cumple, ya que todos los atributos que conforman la tabla son simples. Es decir, cada columna no se puede descomponer en más de 1 elemento.

**Forma normal 2**: Cumple debido a que solo existe un atributo primo, por lo que la llave no puede determinar parcialmente a otro atributo.

**Forma normal 3**: No cumple, pues existen dependencias entre los atributos no primos.

Se decide romper la tabla en 2: códigos postales y ciudades. La primera con los siguientes atributos: **CP,CU** y la segunda con **CU,DP**

**Forma normal BC**: Cumple, ya que la tabla se encuentra en forma normal 3 y la llave de esta tabla es simple, por lo que no hay dependencias parciales de esta.

**Para la tabla códigos postales**

**Forma normal 1**: Cumple, ya que todos los atributos que conforman la tabla son simples. Es decir, cada columna no se puede descomponer en más de 1 elemento.

**Forma normal 2**: Cumple debido a que solo existe un atributo primo, por lo que la llave no puede determinar parcialmente a otro atributo.

**Forma normal 3**: Cumple, ya que ningún atributo no primo determina a otro no primo. En otras palabras, no hay dependencias entre atributos no primos.

**Forma normal BC**: Cumple, ya que la tabla se encuentra en forma normal 3 y la llave de esta tabla es simple, por lo que no hay dependencias parciales de esta.



**Para la tabla ciudades**

**Forma normal 1**: Cumple, ya que todos los atributos que conforman la tabla son simples. Es decir, cada columna no se puede descomponer en más de 1 elemento.

**Forma normal 2**: Cumple debido a que solo existe un atributo primo, por lo que la llave no puede determinar parcialmente a otro atributo.

**Forma normal 3**: Cumple, ya que ningún atributo no primo determina a otro no primo. En otras palabras, no hay dependencias entre atributos no primos.

**Forma normal BC**: Cumple, ya que la tabla se encuentra en forma normal 3 y la llave de esta tabla es simple, por lo que no hay dependencias parciales de esta.



**Tabla empleado**

**Forma normal 1**: Cumple, ya que todos los atributos que conforman la tabla son simples. Es decir, cada columna no se puede descomponer en más de 1 elemento.

**Forma normal 2**: Cumple debido a que solo existe un atributo primo, por lo que la llave no puede determinar parcialmente a otro atributo.

**Forma normal 3**: Cumple, ya que ningún atributo no primo determina a otro no primo. En otras palabras, no hay dependencias entre atributos no primos.

**Forma normal BC**: Cumple, ya que la tabla se encuentra en forma normal 3 y la llave de esta tabla es simple, por lo que no hay dependencias parciales de esta.



**Tabla productos**

como se tiene una relacion de herencia en esta tabla únicamente se guarda el ID del producto y el cliente. Se hacen otras 2 tablas para representar la relación de herencia.

**Relaciones**

* **ID -> CL**

**llave candidata:**

* **|ID|->CL**

**Primos: ID**

**No primos: CL**

**Forma normal 1**: Cumple, ya que todos los atributos que conforman la tabla son simples. Es decir, cada columna no se puede descomponer en más de 1 elemento.

**Forma normal 2**: Cumple debido a que solo existe un atributo primo, por lo que la llave no puede determinar parcialmente a otro atributo.

**Forma normal 3**: Cumple, ya que ningún atributo no primo determina a otro no primo. En otras palabras, no hay dependencias entre atributos no primos.

**Forma normal BC**: Cumple, ya que la tabla se encuentra en forma normal 3 y la llave de esta tabla es simple, por lo que no hay dependencias parciales de esta.



**Tabla Cuentas**

**Relaciones:**

* **Id -> ES, TI, DI**

**Llave candidata: |ID|-> ES, TI, DI**

**Primos: ID**

**No primos: ES, TI, DI**

**Forma normal 1**: Cumple, ya que todos los atributos que conforman la tabla son simples. Es decir, cada columna no se puede descomponer en más de 1 elemento.

**Forma normal 2**: Cumple debido a que solo existe un atributo primo, por lo que la llave no puede determinar parcialmente a otro atributo.

**Forma normal 3**: Cumple, ya que ningún atributo no primo determina a otro no primo. En otras palabras, no hay dependencias entre atributos no primos.

**Forma normal BC**: Cumple, ya que la tabla se encuentra en forma normal 3 y la llave de esta tabla es simple, por lo que no hay dependencias parciales de esta.



**Tabla Prestamos**

**Relaciones:**

* **ID->ES, TI, IN, CU, VC, MO, DC, SP**

**Llave candidata: |id|-> ES, TI, IN, CU, VC, MO, DC, SP**

**Primos: ID**

**No primos: ES, TI, IN, CU, VC, MO, DC, SP**

**Forma normal 1**: Cumple, ya que todos los atributos que conforman la tabla son simples. Es decir, cada columna no se puede descomponer en más de 1 elemento.

**Forma normal 2**: Cumple debido a que solo existe un atributo primo, por lo que la llave no puede determinar parcialmente a otro atributo.

**Forma normal 3**: Cumple, ya que ningún atributo no primo determina a otro no primo. En otras palabras, no hay dependencias entre atributos no primos.

**Forma normal BC**: Cumple, ya que la tabla se encuentra en forma normal 3 y la llave de esta tabla es simple, por lo que no hay dependencias parciales de esta.

**Tabla Operación bancaria**

**Relaciones:**

* **ID->HO, FC, IO, PA, VA, TP, IE**

**Llave candidata: |ID|-> HO, FC, IO, PA, VA, TP, IE**

**Primos: ID**

**No primos: HO, FC, IO, PA, VA, TP, IE**

**Forma normal 1**: Cumple, ya que todos los atributos que conforman la tabla son simples. Es decir, cada columna no se puede descomponer en más de 1 elemento.

**Forma normal 2**: Cumple debido a que solo existe un atributo primo, por lo que la llave no puede determinar parcialmente a otro atributo.

**Forma normal 3**: Cumple, ya que ningún atributo no primo determina a otro no primo. En otras palabras, no hay dependencias entre atributos no primos.

**Forma normal BC**: Cumple, ya que la tabla se encuentra en forma normal 3 y la llave de esta tabla es simple, por lo que no hay dependencias parciales de esta.



**Tabla Oficina**

**Relaciones:**

* **NO -> DR, PA**

**Llave candidata: |NO| -> DR, PA**

**Primos: NO**

**No primos: DR,PA**

**Forma normal 1**: Cumple, ya que todos los atributos que conforman la tabla son simples. Es decir, cada columna no se puede descomponer en más de 1 elemento.

**Forma normal 2**: Cumple debido a que solo existe un atributo primo, por lo que la llave no puede determinar parcialmente a otro atributo.

**Forma normal 3**: Cumple, ya que ningún atributo no primo determina a otro no primo. En otras palabras, no hay dependencias entre atributos no primos.

**Forma normal BC**: Cumple, ya que la tabla se encuentra en forma normal 3 y la llave de esta tabla es simple, por lo que no hay dependencias parciales de esta.



**Tabla punto de atención**

**Relaciones:**

* **ID -> TP, LC, OF**

**Llave candidata: |ID| -> TP, LC, OF**

**Primos: ID**

**No primos: TP,LC,OF**

**Forma normal 1**: Cumple, ya que todos los atributos que conforman la tabla son simples. Es decir, cada columna no se puede descomponer en más de 1 elemento.

**Forma normal 2**: Cumple debido a que solo existe un atributo primo, por lo que la llave no puede determinar parcialmente a otro atributo.

**Forma normal 3**: Cumple, ya que ningún atributo no primo determina a otro no primo. En otras palabras, no hay dependencias entre atributos no primos.

**Forma normal BC**: Cumple, ya que la tabla se encuentra en forma normal 3 y la llave de esta tabla es simple, por lo que no hay dependencias parciales de esta.



**Escenarios de prueba:**

En las tablas de las imágenes y en el excel adjunto se encuentran al lado de cada tupla el requerimiento funcional que cada tupla cumple en el escenario de prueba. De ser verde la tupla significa que la tupla es correcta de ser roja sígnica que la tupla seria rechazada por el sistema.

**Escenario 1 – RF1**

El gerente general crea los clientes con numero de documento 1234, 1235. El cliente 1234 también es un empleado. Comete un error con el nombre y la dirección del 1235, el sistema rechaza, el corrige.

También crea al 1236 un nuevo gerente de oficina.

**Escenario 2 – RF2**

el administrador intenta crear 2 oficinas, la calle 127 y la calle 100. Se equivoca en la dirección de la calle 100 y es exitoso en la calle 127.

Crea los respectivos puntos de atención de la oficina de la calle 127 de manera exitosa, un punto automático y un punto personalizado con los id 3 y 4 respectivamente.

**Escenario 3 – RF3**

Un gerente de oficina crea la cuenta de ahorros #2, asociada al cliente 1234 de manera exitosa.

**Escenario 4 – RF4**

Se regitstra la operación de apertura anteriormente mencionada. Acto seguido el cliente 1234 consigna 2’100.000, se registra esta operación. Se hace una transferencia entre cunetas 1 y 4 por un valor de 100.000. se cierra la cuenta #3.

**Escenario 5 – RF5**

Se aprueba el préstamo 1 de tipo Estudio para el cliente 1234. Paga su primera cuota, se registran las operaciones.

**Escenario 6 -RF6**

El cliente 1234 termina de pagar su anterior préstamo de estudio pagando una cuota extraordinaria, el préstamo se cierra. Se registran las operaciones.