APÉNDICES

A. Evaluación de Plataformas Tecnológicas

a. Scripts de instalación

```
1 #!/bin/bash
2 sudo apt-get install -y python
3 sudo apt-get install -y python-pip
4 sudo apt-get install -y python-virtualenv
5 mkdir project
6 cd project
7 sudo virtualenv env
8 source env/bin/activate
9 sudo pip install Flask==0.10.1
```

FIGURA 25 **SCRIPT DE INSTALACIÓN FLASK**
Fuente: Elaboración propia

```
1 #!/bin/bash
2 sudo apt-get install -y curl
3 \curl -L https://get.rvm.io | bash -s stable
4 source ~/.rvm/scripts/rvm
5 rvm requirements
6 rvm install ruby
7 rvm use ruby --default
8 rvm rubygems current
9 gem install sinatra
```

FIGURA 26 ***SCRIPT DE INSTALACIÓN SINATRA***
Fuente: Elaboración Propia

```
1 #!/bin/bash
2 sudo apt-get -y install oracle-java8-installer
3 apt-cache search maven
4 sudo apt-get -y install maven
5 sudo apt-get -y install tomcat7
6 mvn archetype:generate -DgroupId=net.citywallet -DartifactId=services
7  -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
8
```

FIGURA 27 ***SCRIPT DE INSTALACIÓN SPRING***
Fuente: Elaboración propia

	Spring			**Sinatra**			**Flask**	
	Líneas de código:	**7**		Líneas de código:	**9**		Líneas de código:	**9**
	Tiempo promedio instalación :	**237.63s**		Tiempo promedio instalación :	**465.05s**		Tiempo promedio instalación :	**32.72s**
	Medidas			Medidas			Medidas	
Real			**Real**			**Real**		
1	192.364	s	1	408.874	s	1	30.553	s
2	213.744	s	2	490.121	s	2	34.681	s
3	225.16	s	3	487.725	s	3	35.217	s

Cod. de Barras: 101576

4	208.316	s	4	530.256	s	4	32.807	s
5	239.52	s	5	430.569	s	5	33.628	s
6	326.475	s	6	452.689	s	6	33.628	s
7	266.77	s	7	466.875	s	7	31.848	s
8	262.638	s	8	512.645	s	8	33.436	s
9	239.042	s	9	413.865	s	9	30.421	s
10	202.299	s	10	456.887	s	10	31.028	s
CPU			**CPU**			**CPU**		
1	28.101	s	1	354.203	s	1	11.863	s
2	29.023	s	2	358.64	s	2	11.897	s
3	28.27	s	3	378.713	s	3	11.842	s
4	28.556	s	4	377.391	s	4	12.183	s
5	29.434	s	5	350.45	s	5	11.913	s
6	30.014	s	6	380.05	s	6	11.913	s
7	29.841	s	7	345.56	s	7	11.907	s
8	29.266	s	8	377.156	s	8	12.017	s
9	31.094	s	9	371.56	s	9	11.913	s
10	29.2	s	10	369.85	s	10	11.949	s
Sys			**Sys**			**Sys**		
1	27.668	s	1	6.709	s	1	3.818	s
2	25.531	s	2	6.383	s	2	3.866	s
3	25.337	s	3	6.984	s	3	3.832	s
4	26.75	s	4	7.432	s	4	3.918	s
5	28.434	s	5	6.8	s	5	3.894	s
6	31.673	s	6	6.552	s	6	3.894	s
7	29.439	s	7	6.235	s	7	3.913	s
8	29.155	s	8	7.112	s	8	3.93	s
9	34.283	s	9	7.254	s	9	3.888	s
10	28.088	s	10	6.355	s	10	3.924	s

FIGURA 28 ***RESULTADO EVALUACIÓN TIEMPO DE EJECUCIÓN DE SCRIPTS DE INSTALACIÓN***
Fuente: Elaboración propia

b. Tabla de base de datos para evaluación de frameworks

FIGURA 29 ***TABLA DE BASE DE DATOS PARA PRUEBAS DE FRAMEWORK***
Fuente: Elaboración propia

c. Resultados de benchmarks realizados por la Universidad Ahmadu Bello

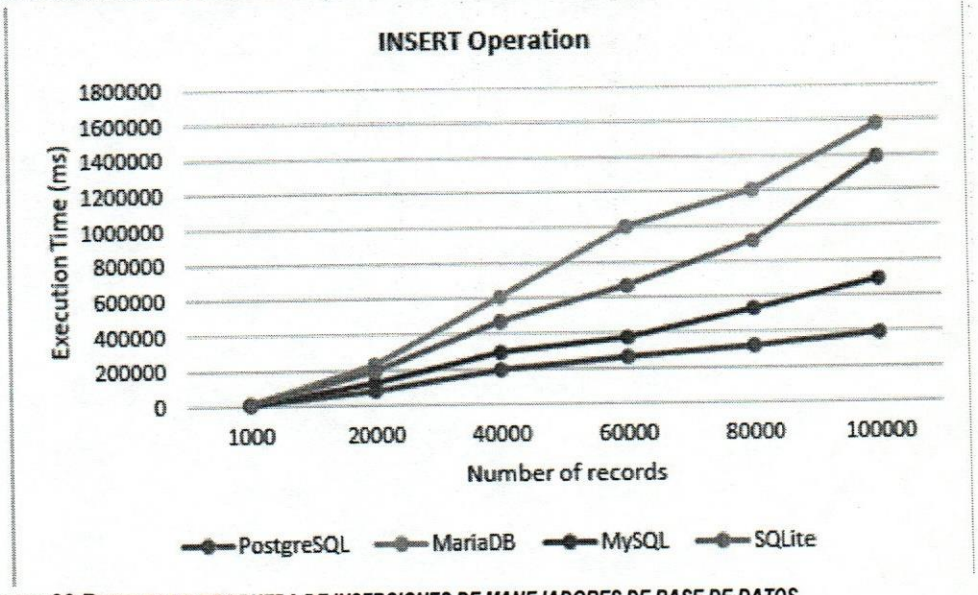

FIGURA 30 ***RESULTADO DE PRUEBA DE INSERCIONES DE MANEJADORES DE BASE DE DATOS***
Fuente: Universidad Ahmadu Bello [204]

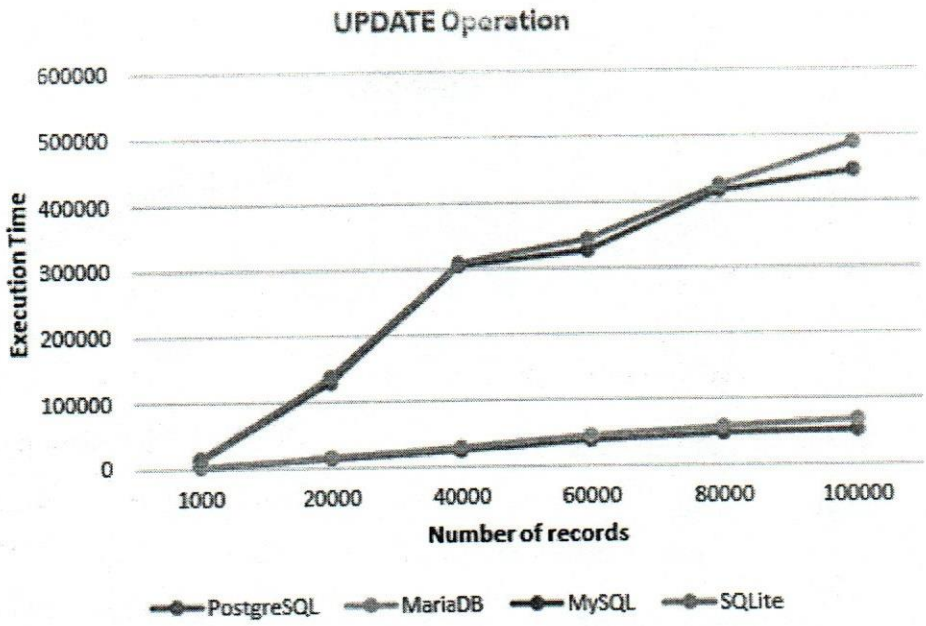

FIGURA 31 ***RESULTADO DE PRUEBA DE ACTUALIZACIÓN DE DATA DE MANEJADORES DE BASE DE DATOS***
Fuente: Universidad Ahmadu Bello [204]

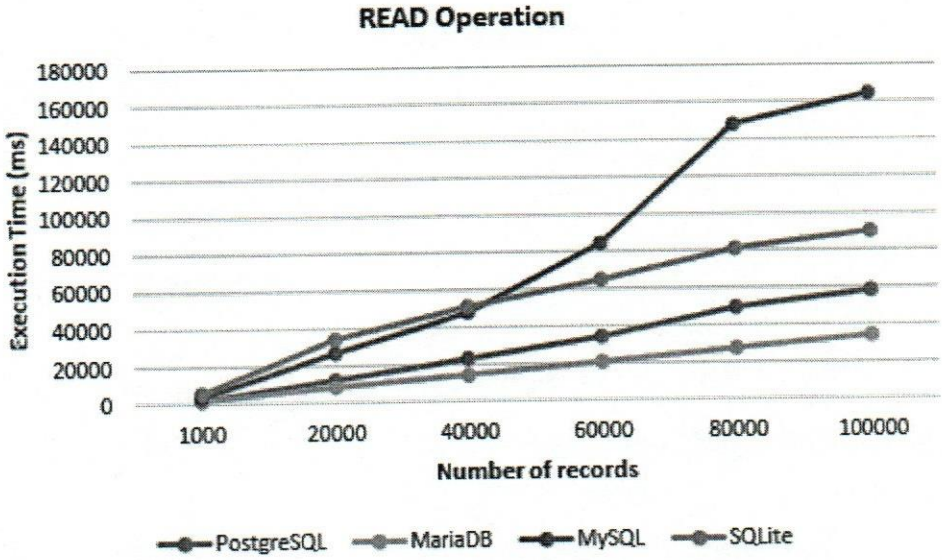

FIGURA 32 ***RESULTADOS DE PRUEBAS DE LECTURAS DE MANEJADORES DE BASE DE DATOS***
Fuente: Universidad Ahmadu Bello [204]

d. Scripts de evaluación de manejadores de base de datos

```
1 #!/bin/bash
2 mysql benchmarks < createMysql.sql &
3
4 #echo CPU: `top -b -n1 -p $$ | grep "Cpu(s)" | awk '{print $2 + $4}'`
5 cpuCounter=0
6 memCounter=0
7 cpuCount=0
8 memCount=0
9
10 while kill -0 $! > /dev/null 2>&1; do
11  cpuAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f9 -d' '`
12 if [ ! -z "$cpuAux" -a "$cpuAux" != " " ]; then
13   cpuCount=$(awk "BEGIN {print $cpuCount + $cpuAux; exit}")
14   cpuCounter=$((cpuCounter+1))
15 fi
16
17  memAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f10 -d' '`
18 if [ ! -z "$memAux" -a "$memAux" != " " ]; then
19   memCount=$(awk "BEGIN {print $memCount + $memAux; exit}")
20   memCounter=$((memCounter+1))
21 fi
22 done
23
24 cpuCount=$(awk "BEGIN {print $cpuCount / $cpuCounter; exit}")
25 memCount=$(awk "BEGIN {print $memCount / $memCounter; exit}")
26
27 echo 'cpu: ' $cpuCount
28 echo 'mem: ' $memCount
```

FIGURA 33 ***SCRIPT DE EVALUACIÓN DE CREACIÓN DE TABLAS MARIADB***
Fuente: Elaboración propia

```
1 #!/bin/bash
2 mysql benchmarks < dropMysql.sql &
3
4 #echo CPU: `top -b -n1 -p $$ | grep "Cpu(s)" | awk '{print $2 + $4}'`
5 cpuCounter=0
6 memCounter=0
7 cpuCount=0
8 memCount=0
9
10 while kill -0 $! > /dev/null 2>&1; do
11  cpuAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f9 -d' '`
12 if [ ! -z "$cpuAux" -a "$cpuAux" != " " ]; then
13   cpuCount=$(awk "BEGIN {print $cpuCount + $cpuAux; exit}")
14   cpuCounter=$((cpuCounter+1))
15 fi
16
17  memAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f10 -d' '`
18 if [ ! -z "$memAux" -a "$memAux" != " " ]; then
19   memCount=$(awk "BEGIN {print $memCount + $memAux; exit}")
20   memCounter=$((memCounter+1))
21 fi
22 done
23
24 cpuCount=$(awk "BEGIN {print $cpuCount / $cpuCounter; exit}")
25 memCount=$(awk "BEGIN {print $memCount / $memCounter; exit}")
26
27 echo 'cpu: ' $cpuCount
28 echo 'mem: ' $memCount
```

FIGURA 34 ***SCRIPT DE EVALUACIÓN DE ELIMINACIÓN DE TABLAS DE BASE DE DATOS EN MARIADB***
Fuente: Elaboración propia

```
1 #!/bin/bash
2 mysql benchmarks < insertMysql.sql &
3
4 #echo CPU: `top -b -n1 -p $$ | grep "Cpu(s)" | awk '{print $2 + $4}'`
5 cpuCounter=0
6 memCounter=0
7 cpuCount=0
8 memCount=0
9
10 while kill -0 $! > /dev/null 2>&1; do
11  cpuAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f9 -d' '`
12 if [ ! -z "$cpuAux" -a "$cpuAux" != " " ]; then
13   cpuCount=$(awk "BEGIN {print $cpuCount + $cpuAux; exit}")
14   cpuCounter=$((cpuCounter+1))
15 fi
16
17  memAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f10 -d' '`
18 if [ ! -z "$memAux" -a "$memAux" != " " ]; then
19   memCount=$(awk "BEGIN {print $memCount + $memAux; exit}")
20   memCounter=$((memCounter+1))
21 fi
22 done
23
24 cpuCount=$(awk "BEGIN {print $cpuCount / $cpuCounter; exit}")
25 memCount=$(awk "BEGIN {print $memCount / $memCounter; exit}")
26
27 echo 'cpu: ' $cpuCount
28 echo 'mem: ' $memCount
```

FIGURA 35 ***SCRIPT DE EVALUACIÓN DE INSERCIÓN DE DATA EN BASE DE DATOS EN MARIADB***
Fuente: Elaboración propia

```
1 #!/bin/bash
2 mysql benchmarks < deleteMysql.sql &
3
4 #echo CPU: `top -b -n1 -p $$ | grep "Cpu(s)" | awk '{print $2 + $4}'`
5 cpuCounter=0
6 memCounter=0
7 cpuCount=0
8 memCount=0
9
10 while kill -0 $! > /dev/null 2>&1; do
11  cpuAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f9 -d' '`
12 if [ ! -z "$cpuAux" -a "$cpuAux" != " " ]; then
13   cpuCount=$(awk "BEGIN {print $cpuCount + $cpuAux; exit}")
14   cpuCounter=$((cpuCounter+1))
15 fi
16
17  memAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f10 -d' '`
18 if [ ! -z "$memAux" -a "$memAux" != " " ]; then
19   memCount=$(awk "BEGIN {print $memCount + $memAux; exit}")
20   memCounter=$((memCounter+1))
21 fi
22 done
23
24 cpuCount=$(awk "BEGIN {print $cpuCount / $cpuCounter; exit}")
25 memCount=$(awk "BEGIN {print $memCount / $memCounter; exit}")
26
27 echo 'cpu: ' $cpuCount
28 echo 'mem: ' $memCount
```

FIGURA 36 ***SCRIPT DE EVALUACIÓN DE ELIMINACIÓN DE DATA EN BASE DE DATOS EN MARIADB***
Fuente: Elaboración propia

```
#!/bin/bash
mysql benchmarks < createMysql.sql &

#echo CPU: `top -b -n1 -p $$ | grep "Cpu(s)" | awk '{print $2 + $4}'
cpuCounter=0
memCounter=0
cpuCount=0
memCount=0

while kill -0 $! > /dev/null 2>&1; do
 cpuAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f9 -d' '`
if [ ! -z "$cpuAux" -a "$cpuAux" != " " ]; then
  cpuCount=$(awk "BEGIN {print $cpuCount + $cpuAux; exit}")
  cpuCounter=$((cpuCounter+1))
fi

 memAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f10 -d' '`
if [ ! -z "$memAux" -a "$memAux" != " " ]; then
  memCount=$(awk "BEGIN {print $memCount + $memAux; exit}")
  memCounter=$((memCounter+1))
fi
done

cpuCount=$(awk "BEGIN {print $cpuCount / $cpuCounter; exit}")
memCount=$(awk "BEGIN {print $memCount / $memCounter; exit}")

echo 'cpu: ' $cpuCount
echo 'mem: ' $memCount
```

FIGURA 37 ***SCRIPT DE EVALUACIÓN DE CREACIÓN DE TABLAS EN BASE DE DATOS MYSQL***
Fuente: Elaboración propia

```
#!/bin/bash
mysql benchmarks < dropMysql.sql &

#echo CPU: `top -b -n1 -p $$ | grep "Cpu(s)" | awk '{print $2 + $4}'
cpuCounter=0
memCounter=0
cpuCount=0
memCount=0

while kill -0 $! > /dev/null 2>&1; do
 cpuAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f9 -d' '`
if [ ! -z "$cpuAux" -a "$cpuAux" != " " ]; then
  cpuCount=$(awk "BEGIN {print $cpuCount + $cpuAux; exit}")
  cpuCounter=$((cpuCounter+1))
fi

 memAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f10 -d' '`
if [ ! -z "$memAux" -a "$memAux" != " " ]; then
  memCount=$(awk "BEGIN {print $memCount + $memAux; exit}")
  memCounter=$((memCounter+1))
fi
done

cpuCount=$(awk "BEGIN {print $cpuCount / $cpuCounter; exit}")
memCount=$(awk "BEGIN {print $memCount / $memCounter; exit}")

echo 'cpu: ' $cpuCount
echo 'mem: ' $memCount
```

FIGURA 38 ***SCRIPT DE EVALUACIÓN DE ELIMINACIÓN DE TABLAS EN BASE DE DATOS EN MYSQL***
Fuente: Elaboración propia

```
#!/bin/bash
mysql benchmarks < insertMysql.sql &

#echo CPU: `top -b -n1 -p $$ | grep "Cpu(s)" | awk '{print $2 + $4}'`
cpuCounter=0
memCounter=0
cpuCount=0
memCount=0

while kill -0 $! > /dev/null 2>&1; do
 cpuAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f9 -d' '`
if [ ! -z "$cpuAux" -a "$cpuAux" != " " ]; then
  cpuCount=$(awk "BEGIN {print $cpuCount + $cpuAux; exit}")
  cpuCounter=$((cpuCounter+1))
fi

 memAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f10 -d' '`
if [ ! -z "$memAux" -a "$memAux" != " " ]; then
  memCount=$(awk "BEGIN {print $memCount + $memAux; exit}")
  memCounter=$((memCounter+1))
fi
done

cpuCount=$(awk "BEGIN {print $cpuCount / $cpuCounter; exit}")
memCount=$(awk "BEGIN {print $memCount / $memCounter; exit}")

echo 'cpu: ' $cpuCount
echo 'mem: ' $memCount
```

FIGURA 39 ***SCRIPT DE EVALUACIÓN DE INSERCIÓN DE DATA EN BASE DE DATOS MYSQL***

Fuente: Elaboración propia

```
#!/bin/bash
mysql benchmarks < deleteMysql.sql &

#echo CPU: `top -b -n1 -p $$ | grep "Cpu(s)" | awk '{print $2 + $4}'`
cpuCounter=0
memCounter=0
cpuCount=0
memCount=0

while kill -0 $! > /dev/null 2>&1; do
 cpuAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f9 -d' '`
if [ ! -z "$cpuAux" -a "$cpuAux" != " " ]; then
  cpuCount=$(awk "BEGIN {print $cpuCount + $cpuAux; exit}")
  cpuCounter=$((cpuCounter+1))
fi

 memAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f10 -d' '`
if [ ! -z "$memAux" -a "$memAux" != " " ]; then
  memCount=$(awk "BEGIN {print $memCount + $memAux; exit}")
  memCounter=$((memCounter+1))
fi
done

cpuCount=$(awk "BEGIN {print $cpuCount / $cpuCounter; exit}")
memCount=$(awk "BEGIN {print $memCount / $memCounter; exit}")

echo 'cpu: ' $cpuCount
echo 'mem: ' $memCount
```

FIGURA 40 ***SCRIPT DE EVALUACIÓN DE ELIMINACIÓN DE DATA EN BASE DE DATOS MYSQL***

Fuente: Elaboración propia

```
#!/bin/bash
mysql benchmarks < getBalance.sql &

#echo CPU: `top -b -n1 -p $$ | grep "Cpu(s)" | awk '{print $2 + $4}'`
cpuCounter=0
memCounter=0
cpuCount=0
memCount=0

while kill -0 $! > /dev/null 2>&1; do
 cpuAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f9 -d' '`
if [ ! -z "$cpuAux" -a "$cpuAux" != " " ]; then
  cpuCount=$(awk "BEGIN {print $cpuCount + $cpuAux; exit}")
  cpuCounter=$((cpuCounter+1))
fi

 memAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f10 -d' '`
if [ ! -z "$memAux" -a "$memAux" != " " ]; then
  memCount=$(awk "BEGIN {print $memCount + $memAux; exit}")
  memCounter=$((memCounter+1))
fi
done

cpuCount=$(awk "BEGIN {print $cpuCount / $cpuCounter; exit}")
memCount=$(awk "BEGIN {print $memCount / $memCounter; exit}")

echo 'cpu: ' $cpuCount
echo 'mem: ' $memCount
```

FIGURA 41 ***SCRIPT DE EVALUACIÓN DE QUERY DE BALANCE DE BASE DE DATOS MYSQL***

Fuente: Elaboración propia

```
#!/bin/bash
psql -U santiago -d cw -a -f createPostgresql.sql &>> createOut.out &
#echo CPU: `top -b -n1 -p $$ | grep "Cpu(s)" | awk '{print $2 + $4}'`
cpuCounter=0
memCounter=0
cpuCount=0
memCount=0

while kill -0 $! > /dev/null 2>&1; do
 cpuAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f9 -d' '`
if [ ! -z "$cpuAux" -a "$cpuAux" != " " ]; then
  cpuCount=$(awk "BEGIN {print $cpuCount + $cpuAux; exit}")
  cpuCounter=$((cpuCounter+1))
fi

 memAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f10 -d' '`
if [ ! -z "$memAux" -a "$memAux" != " " ]; then
  memCount=$(awk "BEGIN {print $memCount + $memAux; exit}")
  memCounter=$((memCounter+1))
fi
done

cpuCount=$(awk "BEGIN {print $cpuCount / $cpuCounter; exit}")
memCount=$(awk "BEGIN {print $memCount / $memCounter; exit}")

echo 'cpu: ' $cpuCount
echo 'mem: ' $memCount
```

FIGURA 42 ***SCRIPT DE EVALUACIÓN DE CREACIÓN DE TABLAS EN BASE DE DATOS POSTRESQL***

Fuente: Elaboración propia

```
#!/bin/bash
psql -U santiago -d cw -a -f dropPostgresql.sql &>> createOut.out &
#echo CPU: `top -b -n1 -p $$ | grep "Cpu(s)" | awk '{print $2 + $4}'`
cpuCounter=0
memCounter=0
cpuCount=0
memCount=0

while kill -0 $! > /dev/null 2>&1; do
 cpuAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f9 -d' '`
if [ ! -z "$cpuAux" -a "$cpuAux" != " " ]; then
  cpuCount=$(awk "BEGIN {print $cpuCount + $cpuAux; exit}")
  cpuCounter=$((cpuCounter+1))
fi

 memAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f10 -d' '`
if [ ! -z "$memAux" -a "$memAux" != " " ]; then
  memCount=$(awk "BEGIN {print $memCount + $memAux; exit}")
  memCounter=$((memCounter+1))
fi

done
cpuCount=$(awk "BEGIN {print $cpuCount / $cpuCounter; exit}")
memCount=$(awk "BEGIN {print $memCount / $memCounter; exit}")

echo 'cpu: ' $cpuCount
echo 'mem: ' $memCount
```

FIGURA 43 ***SCRIPT DE ELIMINACIÓN DE TABLAS DE BASE DE DATOS POSTGRESQL***
Fuente: Elaboración propia

```
#!/bin/bash
psql -U santiago -d cw -a -f insertPostgres.sql &>> createOut.out &

#echo CPU: `top -b -n1 -p $$ | grep "Cpu(s)" | awk '{print $2 + $4}'`
cpuCounter=0
memCounter=0
cpuCount=0
memCount=0

while kill -0 $! > /dev/null 2>&1; do
 cpuAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f9 -d' '`
if [ ! -z "$cpuAux" -a "$cpuAux" != " " ]; then
  cpuCount=$(awk "BEGIN {print $cpuCount + $cpuAux; exit}")
  cpuCounter=$((cpuCounter+1))
fi

 memAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f10 -d' '`
if [ ! -z "$memAux" -a "$memAux" != " " ]; then
  memCount=$(awk "BEGIN {print $memCount + $memAux; exit}")
  memCounter=$((memCounter+1))
fi
done

cpuCount=$(awk "BEGIN {print $cpuCount / $cpuCounter; exit}")
memCount=$(awk "BEGIN {print $memCount / $memCounter; exit}")

echo 'cpu: ' $cpuCount
echo 'mem: ' $memCount
```

FIGURA 44 ***SCRIPT DE EVALUACIÓN DE INSERCIÓN DE DATA EN BASE DE DATOS POSTGRESQL***
Fuente: Elaboración propia

```
#!/bin/bash
psql -U santiago -d cw -a -f deletePostgres.sql &>> createOut.out &

#echo CPU: `top -b -n1 -p $$ | grep "Cpu(s)" | awk '{print $2 + $4}'`
cpuCounter=0
memCounter=0
cpuCount=0
memCount=0

while kill -0 $! > /dev/null 2>&1; do
 cpuAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f9 -d' '`
if [ ! -z "$cpuAux" -a "$cpuAux" != " " ]; then
  cpuCount=$(awk "BEGIN {print $cpuCount + $cpuAux; exit}")
  cpuCounter=$((cpuCounter+1))
fi

 memAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f10 -d' '`
if [ ! -z "$memAux" -a "$memAux" != " " ]; then
  memCount=$(awk "BEGIN {print $memCount + $memAux; exit}")
  memCounter=$((memCounter+1))
fi
done

cpuCount=$(awk "BEGIN {print $cpuCount / $cpuCounter; exit}")
memCount=$(awk "BEGIN {print $memCount / $memCounter; exit}")

echo 'cpu: ' $cpuCount
echo 'mem: ' $memCount
```

FIGURA 45 ***SCRIPT DE EVALUACIÓN DE ELIMINACIÓN DE DATA EN LA BASE DE DATOS POSTGRESQL***
Fuente: Elaboración propia

```
#!/bin/bash
psql -U santiago -d cw -a -f getBalance.sql &>> createOut.out &

#echo CPU: `top -b -n1 -p $$ | grep "Cpu(s)" | awk '{print $2 + $4}'`
cpuCounter=0
memCounter=0
cpuCount=0
memCount=0

while kill -0 $! > /dev/null 2>&1; do
 cpuAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f9 -d' '`
if [ ! -z "$cpuAux" -a "$cpuAux" != " " ]; then
  cpuCount=$(awk "BEGIN {print $cpuCount + $cpuAux; exit}")
  cpuCounter=$((cpuCounter+1))
fi

 memAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f10 -d' '`
if [ ! -z "$memAux" -a "$memAux" != " " ]; then
  memCount=$(awk "BEGIN {print $memCount + $memAux; exit}")
  memCounter=$((memCounter+1))
fi
done

cpuCount=$(awk "BEGIN {print $cpuCount / $cpuCounter; exit}")
memCount=$(awk "BEGIN {print $memCount / $memCounter; exit}")

echo 'cpu: ' $cpuCount
echo 'mem: ' $memCount
```

FIGURA 46 ***SCRIPT DE EVALUACIÓN DE QUERY DE BALANCE EN BASE DE DATOS POSTGRESQL***
Fuente: Elaboración propia

```
#!/bin/bash
rm cw.sqlite

cpuCounter=0
memCounter=0
cpuCount=0
memCount=0

while kill -0 $! > /dev/null 2>&1; do
 cpuAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f9 -d' '`
if [ ! -z "$cpuAux" -a "$cpuAux" != " " ]; then
  cpuCount=$(awk "BEGIN {print $cpuCount + $cpuAux; exit}")
  cpuCounter=$((cpuCounter+1))
fi

 memAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f10 -d' '`
if [ ! -z "$memAux" -a "$memAux" != " " ]; then
  memCount=$(awk "BEGIN {print $memCount + $memAux; exit}")
  memCounter=$((memCounter+1))
fi
done

cpuCount=$(awk "BEGIN {print $cpuCount / $cpuCounter; exit}")
memCount=$(awk "BEGIN {print $memCount / $memCounter; exit}")

echo 'cpu: ' $cpuCount
echo 'mem: ' $memCount
```

FIGURA 47 ***SCRIPT DE EVALUACIÓN DE ELIMINACIÓN DE DATA EN BASE DE DATOS SQLITE***

Fuente: Elaboración propia

```
#!/bin/bash
echo .read createSQLite.sql | sqlite3 cw.sqlite > /dev/null 2>&1 &
echo `top -b -n1 -p $! | grep $!`
cpuCounter=0
memCounter=0
cpúCount=0
memCount=0

while kill -0 $! > /dev/null 2>&1; do
 cpuAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f9 -d' '`
if [ ! -z "$cpuAux" -a "$cpuAux" != " " ]; then
  cpuCount=$(awk "BEGIN {print $cpuCount + $cpuAux; exit}")
  cpuCounter=$((cpuCounter+1))
fi

 memAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f10 -d' '`
if [ ! -z "$memAux" -a "$memAux" != " " ]; then
  memCount=$(awk "BEGIN {print $memCount + $memAux; exit}")
  memCounter=$((memCounter+1))
fi
done

cpuCount=$(awk "BEGIN {print $cpuCount / $cpuCounter; exit}")
memCount=$(awk "BEGIN {print $memCount / $memCounter; exit}")

echo 'cpu: ' $cpuCount
echo 'mem: ' $memCount
```

FIGURA 48 ***SCRIPT DE EVALUACIÓN DE CREACIÓN DE TABLAS EN BASE DE DATOS SQLITE***

Fuente: Elaboración propia

```
#!/bin/bash
echo .read insertSQLite.sql | sqlite3 cw.sqlite > /dev/null 2>&1 &
cpuCounter=0
memCounter=0
cpuCount=0
memCount=0

while kill -0 $! > /dev/null 2>&1; do
 cpuAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f9 -d' '`
if [ ! -z "$cpuAux" -a "$cpuAux" != " " ]; then
  cpuCount=$(awk "BEGIN {print $cpuCount + $cpuAux; exit}")
  cpuCounter=$((cpuCounter+1))
fi

 memAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f10 -d' '`
if [ ! -z "$memAux" -a "$memAux" != " " ]; then
  memCount=$(awk "BEGIN {print $memCount + $memAux; exit}")
  memCounter=$((memCounter+1))
fi
done

cpuCount=$(awk "BEGIN {print $cpuCount / $cpuCounter; exit}")
memCount=$(awk "BEGIN {print $memCount / $memCounter; exit}")

echo 'cpu: ' $cpuCount
echo 'mem: ' $memCount
```

FIGURA 49 ***SCRIPT DE EVALUACIÓN DE LA INSERCIÓN DE DATA EN BASE DE DATOS SQLITE***
Fuente: Elaboración propia

```
#!/bin/bash
echo .read deleteSQLite.sql | sqlite3 cw.sqlite > /dev/null 2>&1 &
cpuCounter=0
memCounter=0
cpuCount=0
memCount=0

while kill -0 $! > /dev/null 2>&1; do
 cpuAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f9 -d' '`
if [ ! -z "$cpuAux" -a "$cpuAux" != " " ]; then
  cpuCount=$(awk "BEGIN {print $cpuCount + $cpuAux; exit}")
  cpuCounter=$((cpuCounter+1))
fi

 memAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f10 -d' '`
if [ ! -z "$memAux" -a "$memAux" != " " ]; then
  memCount=$(awk "BEGIN {print $memCount + $memAux; exit}")
  memCounter=$((memCounter+1))
fi
done

cpuCount=$(awk "BEGIN {print $cpuCount / $cpuCounter; exit}")
memCount=$(awk "BEGIN {print $memCount / $memCounter; exit}")

echo 'cpu: ' $cpuCount
echo 'mem: ' $memCount
```

FIGURA 50 ***SCRIPT DE EVALUACIÓN DE ELIMINACIÓN DE DATA EN BASE DE DATOS SQLITE***
Fuente: Elaboración propia

```
#!/bin/bash
echo .read getBalance.sql | sqlite3 cw.sqlite
cpuCounter=0
memCounter=0
cpuCount=0
memCount=0

while kill -0 $! > /dev/null 2>&1; do
 cpuAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f9 -d' '`
if [ ! -z "$cpuAux" -a "$cpuAux" != " " ]; then
  cpuCount=$(awk "BEGIN {print $cpuCount + $cpuAux; exit}")
  cpuCounter=$((cpuCounter+1))
fi

 memAux=`top -b -n1 -p $! | grep $! | tr -s ' ' | cut -f10 -d' '`
if [ ! -z "$memAux" -a "$memAux" != " " ]; then
  memCount=$(awk "BEGIN {print $memCount + $memAux; exit}")
  memCounter=$((memCounter+1))
fi
done

cpuCount=$(awk "BEGIN {print $cpuCount / $cpuCounter; exit}")
memCount=$(awk "BEGIN {print $memCount / $memCounter; exit}")

echo 'cpu: ' $cpuCount
echo 'mem: ' $memCount
```

FIGURA 51 ***SCRIPT DE EVALUACIÓN DE QUERY DE BASE DE DATOS SQLITE***
Fuente: Elaboración propia

e. Resultados benchmarks de manejadores de base de datos

create bd

Tiempo	real	Mysql	MariaDB	Sqlite	Postgresql
	1	16.08	0.485	0.122	4.167
	2	22.17	1.154	0.095	3.02
	3	13.43	1.157	0.118	3.458
	4	14.78	0.511	0.147	3.104
	5	14.75	1.161	0.153	3.1
	6	15.47	0.433	0.093	3.09
	7	14.45	0.543	0.116	3.203
	8	13.98	1.145	0.131	3.202
	9	14.38	1.176	0.14	3.329
	10	13.3	1.126	0.143	3.214
	avg	**15.28**	**0.8891**	**0.126**	**3.29**
	devesta	**2.433**	**0.3246**	**0.02**	**0.32**
		1.41	**0.1882**	**0.011**	**0.18**
	c1	**13.87**	**0.7009**	**0.114**	**3.11**
	c2	**16.69**	**1.0773**	**0.137**	**3.47**

Cpu		Mysql	MariaDB	Sqlite	Postgresql
	1	0.20625	0.01	0	3.56923
	2	0.126923	0.002	0	4.58462
	3	0.264	0	0	3.31667
	4	0.257143	0.01289	0	5.29
	5	0	0.001	0	5.41818
	6	0.002	0	0	5.3
	7	0.01	0.1	0	4.40833
	8	0	0.001	0	1.66
	9	0.01	0.02	0	4.58462
	10	0	0.009	0	5.4
	avg	**0.088**	**0.0156**	**0**	**4.353**
	devesta	**0.109**	**0.0288**	**0**	**1.143**
		0.063	**0.0167**	**0**	**0.662**
	c1	**0.025**	**-0.0011**	**0**	**3.691**
	c2	**0.151**	**0.0323**	**0**	**5.015**

	user	Mysql	MariaDB	Sqlite	Postgresql
	1	0.036	0.022	0.112	0.12
	2	0.024	0.024	0.088	0.096
	3	0.044	0.026	0.108	0.096
	4	0.036	0.025	0.124	0.116
	5	0.02	0.023	0.144	0.108
	6	0.04	0.026	0.088	0.088
	7	0.028	0.014	0.096	0.104
	8	0.036	0.023	0.128	0.088
	9	0.032	0.023	0.124	0.108
	10	0.048	0.023	0.132	0.072
	sys	Mysql	MariaDB	Sqlite	Postgresql
	1	0.004	0.016	0.008	0.092
	2	0.024	0.014	0	0.024
	3	0.016	0.013	0.004	0.036
	4	0.016	0.013	0.016	0.016
	5	0.028	0.013	0.004	0.032
	6	0.012	0.012	0	0.052
	7	0.024	0.013	0.012	0.024
	8	0.028	0.012	0	0.036
	9	0.02	0.013	0.012	0.024
	10	0.008	0.013	0.008	0.008
	avg	**0.052**	**0.0361**	**0.121**	**0.134**
	devesta	**0.008**	**0.0011**	**0.005**	**0.02236**
		0.003	**0.0004**	**0.002**	**0.00864**
	c1	**0.049**	**0.0357**	**0.119**	**0.12536**
	c2	**0.055**	**0.0365**	**0.123**	**0.14264**

Mem		Mysql	MariaDB	Sqlite	Postgresql
	1	0.1	0.1	0	0.1
	2	0.1	0.1	0	0.1
	3	0.1	0.1	0	0.1
	4	0.1	0.1	0	0.1
	5	0.421277	0.1	0	0.1
	6	0.1	0.10098	0	0.1
	7	0.106452	0.12	0	0.1
	8	0.143478	0.01	0	0.1
	9	0.275	0.2	0	0.1
	10	0.45	0.1	0	0.1
	avg	**0.190**	**0.103**	**0**	**0.1**
	devesta	**0.133**	**0.043**	**0**	**1.38778E-17**
		0.077	**0.025**	**0**	**8.0442E-18**
	c1	**0.112**	**0.078**	**0**	**0.1**
	c2	**0.267**	**0.128**	**0**	**0.1**

FIGURA 52 ***RESULTADOS DE EVALUACIÓN DE CREACIÓN DE TABLAS EN LOS MANEJADORES DE BASE DE DATOS***
Fuente: Elaboración propia

Prueba											
drop bd											
	Tiempo	real	Mysql	MariaDB	Sqlite	Postgresql	Cpu		Mysql	MariaDB	Sqlite	Postgresql
		1	6.476	0.067	0.011	0.623		1	0.388235	0	0	26.6
		2	4.707	0.071	0.012	0.47		2	0	0.2	0	32.9
		3	4.036	0.1	0.008	0.404		3	0.44	0.1	0	11.572
		4	5.174	0.089	0.004	0.507		4	0.01	0.01	0	53
		5	4.402	0.072	0.01	0.456		5	0.1	0.129	0	29.74
		6	11.39	0.0822	0.011	0.467		6	0.01	0	0	23.2
		7	8.86	0.069	0.012	0.456		7	0	0.01	0	16.55
		8	4.408	0.067	0.011	0.467		8	0.2	0.01	0	59.6
		9	4.514	0.066	0.011	0.49		9	0.01	0.1	0	26.4
		10	4.382	0.064	0.01	0.449		10	0.02	0.2	0	26.45
		avg	**5.835**	**0.0747**	**0.01**	**0.48**		**avg**	**0.1178**	**0.076**	**0**	**30.60**
		devesta	**2.31**	**0.011**	**0.002**	**0.05**		**devesta**	**0.1601**	**0.077**	**0**	**14.19**
			1.339	**0.007**	**0.001**	**0.03**			**0.0928**	**0.045**	**0**	**8.23**
		c1	**4.496**	**0.068**	**0.009**	**0.45**		**c1**	**0.0250**	**0.031**	**0**	**22.38**
		c2	**7.174**	**0.081**	**0.011**	**0.51**		**c2**	**0.2106**	**0.121**	**0**	**38.83**
		user	Mysql	MariaDB	Sqlite	Postgresql	Mem		Mysql	MariaDB	Sqlite	Postgresql
		1	0.016	0.01	0.004	0.08		1	0.1	0.1	0	0.1
		2	0.004	0.012	0.004	0.08		2	0.1	0	0	0.1
		3	0.004	0.012	0.004	0.036		3	0.1	0.1	0	0.1
		4	0.008	0.013	0	0.028		4	0	0.1	0	0.1
		5	0.008	0.014	0.004	0.068		5	0	0.1	0	0.1
		6	0.02	0.014	0.004	0.08		6	0	0	0	0.1
		7	0.012	0.011	0.008	0.08		7	0.01	0.002	0	0.1
		8	0.024	0.011	0.008	0.072		8	0.001	0.01	0	0.1
		9	0.008	0.01	0.004	0.084		9	0	0.002	0	0.1
		10	0.012	0.011	0.004	0.064		10	0	0	0	0.1
								avg	**0.0311**	**0.041**	**0**	**0.1**
								devesta	**0.045**	**0.048**	**0**	**1.38778E-17**
		sys	Mysql	MariaDB	Sqlite	Postgresql			**0.026**	**0.028**	**0**	**8.0442E-18**
		1	0.004	0.006	0.004	0.028		**c1**	**0.005**	**0.014**	**0**	**0.1**
		2	0.016	0.006	0.004	0.036		**c2**	**0.057**	**0.069**	**0**	**0.1**
		3	0.02	0.005	0	0.012						
		4	0.004	0.007	0	0.024						
		5	0.016	0.006	0	0.024						
		6	0.004	0.007	0.008	0.28						
		7	0.004	0.006	0	0.024						
		8	0	0.006	0.004	0.024						
		9	0.008	0.006	0.004	0.016						
		10	0.008	0.006	0	0.004						
		avg	**0.02**	**0.0179**	**0.007**	**0.1144**						
		devesta	**0.006**	**0.0005**	**0.003**	**0.07805**						
			0.002	**0.0002**	**0.001**	**0.03018**						
		c1	**0.018**	**0.0177**	**0.006**	**0.08422**						
		c2	**0.022**	**0.0181**	**0.008**	**0.14458**						

FIGURA 53 ***RESULTADOS DE EVALUACIÓN DE ELIMINACIÓN DE TABLAS EN MANEJADORES DE BASE DE DATOS***

Fuente: Elaboración propia

Prueba inserts											
	Tiempo	real	Mysql	MariaDB	Sqlite	Postgresql	Cpu		Mysql	MariaDB	Sqlite	Postgresql
		1	236.3	4.761	390.4	64.543		1	0.205	0.121	1.66	1.76
		2	229.1	4.873	411.4	60.833		2	0.325	0.22	8.844	1.74
		3	230	4.853	375.2	63.823		3	0.311	0.3	1.841	1.54
		4	227.1	4.269	411.3	63.171		4	0.213	0	0	1.53
		5	230.4	4.822	780.2	61.578		5	0.210	0	1.666	1.51
		6	227.2	4.303	387.3	63.424		6	0.521	0	1.95	2.03
		7	232.8	4.343	380.6	62.432		7	0.436	0.386	1.558	1.90
		8	310.1	4.292	397.4	61.842		8	0.162	0.707	1.746	1.68
		9	240.6	4.803	399.4	65.344		9	0.3	0.469	1.837	1.67
		10	242	4.806	385	62.829		10	0.452	0.253	1.926	0
		avg	**240.6**	**4.61**	**431.8**	**62.98**		**avg**	**0.31**	**0.25**	**2.303**	**1.537**
		devesta	**23.72**	**0.26**	**116.7**	**1.31**		**devesta**	**0.12**	**0.22**	**2.247**	**0.536**
			13.75	**0.15**	**67.64**	**0.76**			**0.07**	**0.13**	**1.302**	**0.310**
		c1	**226.8**	**4.46**	**364.2**	**62.22**		**c1**	**0.25**	**0.12**	**1.001**	**1.226**
		c2	**254.3**	**4.76**	**499.5**	**63.74**		**c2**	**0.38**	**0.37**	**3.605**	**1.847**
		user	Mysql	MariaDB	Sqlite	Postgresql	Mem	real	Mysql	MariaDB	Sqlite	Postgresql
		1	2.912	0.305	5.556	1.22		1	0.1	4.673	0.081	0.1
		2	2.7	0.322	4.96	1.176		2	0.248	4.045	0.060	0.1
		3	2.868	0.312	6.216	1.104		3	0.1	0.582	0.082	0.1
		4	2.832	0.299	5.792	1.172		4	0.1	4.289	0.089	0.1
		5	2.812	0.308	8.72	1.064		5	0.1	5.236	0.081	0.1
		6	2.752	0.305	6.124	1.416		6	0.27	3.544	0.081	0.1
		7	3.024	0.323	6.476	1.436		7	0.23	1.2	0.100	0.1
		8	2.084	0.301	5.5	1.264		8	0.1	1.23	0.082	0.1
		9	3.668	0.334	6.416	1.252		9	0.1	1.5	0.082	1.521
		10	3.488	0.326	6.288	1.384		10	0.284	1.3344	0.082	1.583
								avg	**0.16**	**2.763**	**0.082**	**0.390**
								devesta	**0.08**	**1.659**	**0.009**	**0.581**
		sys	Mysql	MariaDB	Sqlite	Postgresql			**0.05**	**0.962**	**0.005**	**0.337**
		1	2.824	0.503	9.952	0.02		**c1**	**0.12**	**1.802**	**0.077**	**0.054**
		2	3.776	0.535	8.608	1.28		**c2**	**0.21**	**3.725**	**0.087**	**0.727**
		3	3.66	0.522	10.29	1.264						
		4	3.504	0.501	9.86	1.144						
		5	3.668	0.515	13.62	1.3						
		6	3.624	0.512	11.37	1.212						
		7	3.908	0.529	10.49	1.572						
		8	3.648	0.5	9.96	1.392						
		9	4.2	0.057	11	1.544						
		10	4.164	0.561	10.98	1.332						
		avg	**6.612**	**0.79**	**16.82**	**2.45**						
		devesta	**0.365**	**0.14**	**1.243**	**0.42**						
			0.141	**0.05**	**0.48**	**0.16**						
		c1	**6.471**	**0.73**	**16.34**	**2.29**						
		c2	**6.753**	**0.84**	**17.3**	**2.62**						

FIGURA 54 ***RESULTADOS DE EVALUACIÓN DE INSERCIÓN DE DATA EN MANEJADORES DE BASE DE DATOS***

Fuente: Elaboración propia

Prueba deletes	Tiempos	real	Mysql	MariaDB	Sqlite	Postgresql
		1	0.964	0.983	0.328	1.874
		2	0.635	0.973	0.323	1.878
		3	0.635	0.355	0.368	1.882
		4	0.326	0.993	0.319	3.824
		5	0.623	0.331	0.326	1.87
		6	0.644	1.002	0.326	1.865
		7	0.325	0.991	0.325	1.878
		8	0.642	0.33	0.384	1.866
		9	0.592	0.33	0.496	1.881
		10	0.331	0.981	0.336	1.867
		avg	**0.572**	**0.73**	**0.353**	**2.07**
		devesta	**0.188**	**0.32**	**0.052**	**0.59**
			0.109	**0.18**	**0.03**	**0.34**
		c1	**0.463**	**0.54**	**0.323**	**1.73**
		c2	**0.681**	**0.91**	**0.383**	**2.41**

Cpu	real	Mysql	MariaDB	Sqlite	Postgresql
	1	0	0	0	5.53
	2	0	0	6.6	6.62
	3	0	0	0	6.65
	4	0	0	0	3.32
	5	0	0	0	6.63
	6	0	0	0	8.19
	7	0	0	2.65	4.43
	8	0	0	1.99	0.00
	9	0	0	2.64	0.00
	10	0	0	1.99	0.00
	avg	**0**	**0**	**1.587**	**4.14**
	devesta	**0**	**0**	**2.007**	**2.99**
		0	**0**	**1.163**	**1.73**
	c1	**0**	**0**	**0.424**	**2.41**
	c2	**0**	**0**	**2.75**	**5.87**

	user	Mysql	MariaDB	Sqlite	Postgresql
	1	0.2	0.029	0.02	0.4
	2	0.016	0.026	0.016	0.08
	3	0.012	0.012	0.028	0.064
	4	0.008	0.027	0.028	0.096
	5	0.004	0.013	0.008	0.072
	6	0.016	0.032	0.044	0.088
	7	0.012	0.029	0.036	0.052
	8	0	0.012	0.04	0.056
	9	0.012	0.012	0.032	0.052
	10	0.016	0.028	0.036	0.036

	sys	Mysql	MariaDB	Sqlite	Postgresql
	1	0.016	0.035	0.012	0.02
	2	0.08	0.03	0.016	0.016
	3	0.016	0.013	0.028	0.008
	4	0.004	0.034	0.012	0.088
	5	0	0.014	0.02	0.016
	6	0.016	0.038	0.012	0.02
	7	0.008	0.036	0.02	0.008
	8	0.012	0.014	0.008	0.036
	9	0.012	0.013	0.028	0.016
	10	0.008	0.034	0.02	0.016
	avg	**0.047**	**0.048**	**0.046**	**0.124**
	devesta	**0.022**	**0.010**	**0.006**	**0.022**
		0.008	**0.004**	**0.003**	**0.009**
	c1	**0.038**	**0.044**	**0.044**	**0.115**
	c2	**0.055**	**0.052**	**0.049**	**0.133**

		Mysql	MariaDB	Sqlite	Postgresql
Mem	1	0	0	0	0.1
	2	0.1	0	0	0.1
	3	0.1	0	0	0.1
	4	0	0	0	0.11
	5	0	0.2	0	0.1
	6	0.1	0.2	0	0.1
	7	0	0.2	0	0.1
	8	0.1	0	0	0
	9	0	0	0	0
	10	0	0.2	0	0
	avg	**0.04**	**0.08**	**0**	**0.07**
	devesta	**0.05**	**0.10**	**0**	**0.05**
		0.03	**0.06**	**0**	**0.03**
	c1	**0.01**	**0.02**	**0**	**0.04**
	c2	**0.07**	**0.14**	**0**	**0.10**

FIGURA 55 ***RESULTADOS DE EVALUACIÓN DE ELIMINACIÓN DE DATA EN MANEJADORES DE BASE DE DATOS***
Fuente: Elaboración propia

Prueba												
Consulta	tiempos	real	Mysql	MariaDB	Sqlite	Postgresql		Cpu	Mysql	MariaDB	Sqlite	Postgresql	
		1	0.325	0.328	0.03	0.313			1	0	0	0	0
		2	0.324	0.323	0.139	0.327			2	0	0	0	0
		3	0.32	0.322	0.045	0.322			3	0	0	0	0
		4	0.323	0.326	0.041	0.32			4	0	0	0	0
		5	0.322	0.328	0.04	0.315			5	0	0	0	0
		6	0.324	0.323	0.029	0.325			6	0	0	0	0
		7	0.322	0.323	0.028	0.326			7	0	0	0	0
		8	0.324	0.328	0.04	0.325			8	0	0	0	0
		9	0.325	0.352	0.018	0.323			9	0	0	0	0
		10	0.318	0.327	0.032	0.326			10	0	0	0	0
		avg	**0.323**	**0.328**	**0.044**	**0.322**			**avg**	**0**	**0**	**0**	**0**
		devesta	**0.002**	**0.008**	**0.033**	**0.005**			**devesta**	**0**	**0**	**0**	**0**
			0.001	**0.005**	**0.019**	**0.003**				**0**	**0**	**0**	**0**
		c1	**0.321**	**0.323**	**0.025**	**0.320**			**c1**	**0**	**0**	**0**	**0**
		c2	**0.324**	**0.333**	**0.063**	**0.325**			**c2**	**0**	**0**	**0**	**0**
		user	Mysql	MariaDB	Sqlite	Postgresql		Mem	Mysql	MariaDB	Sqlite	Postgresql	
		1	0.02	0.013	0.024	0.036			1	0	0	0	0
		2	0.012	0.011	0.016	0.044			2	0	0	0	0
		3	0.02	0.011	0.028	0.064			3	0	0	0	0
		4	0.016	0.012	0.032	0.04			4	0	0	0	0
		5	0.012	0.012	0.032	0.048			5	0	0	0	0
		6	0.024	0.012	0.016	0.052			6	0	0	0	0
		7	0.008	0.011	0.02	0.1			7	0	0	0	0
		8	0.016	0.012	0.02	0.084			8	0	0	0	0
		9	0.012	0.014	0.012	0.096			9	0	0	0	0
		10	0.012	0.011	0.024	0.028			10	0	0	0	0
									avg	**0**	**0**	**0**	**0**
									devesta	**0**	**0**	**0**	**0**
		sys	Mysql	MariaDB	Sqlite	Postgresql				**0**	**0**	**0**	**0**
		1	0.004	0.014	0	0.016			**c1**	**0**	**0**	**0**	**0**
		2	0.012	0.012	0.012	0.02			**c2**	**0**	**0**	**0**	**0**
		3	0	0.012	0.08	0							
		4	0.004	0.013	0	0.02							
		5	0.008	0.013	0.004	0.004							
		6	0.008	0.013	0.004	0.02							
		7	0.012	0.012	0.004	0.008							
		8	0.008	0.014	0.012	0.024							
		9	0.016	0.017	0.005	0.02							
		10	0.012	0.013	0.004	0.028							
		avg	**0.024**	**0.025**	**0.035**	**0.075**							
		devesta	**0.005**	**0.001**	**0.023**	**0.009**							
			0.002	**0.001**	**0.009**	**0.003**							
		c1	**0.022**	**0.025**	**0.026**	**0.072**							
		c2	**0.025**	**0.026**	**0.044**	**0.079**							

FIGURA 56 ***RESULTADOS DE EVALUACIÓN DE CONSULTA DE INFORMACIÓN EN MANEJADORES DE BASE DE DATOS***
Fuente: Elaboración propia

f. Script de evaluación de consumo de CPU y memoria de tomcat

```
1 #!/bin/bash
2
3 cpuCounter=0
4 memCounter=0
5 cpuCount=0
6 memCount=0
7
8 while kill -0 $! > /dev/null 2>&1; do
9  cpuAux=`top -b -n1 -p $! | grep 'tomcat' | tr -s ' ' | cut -f9 -d' '`
10 if [ ! -z "$cpuAux" -a "$cpuAux" != " " ]; then
11   cpuCount=$(awk "BEGIN {print $cpuCount + $cpuAux; exit}")
12   cpuCounter=$((cpuCounter+1))
13 fi
14
15  memAux=`top -b -n1 -p $! | grep 'tomcat' | tr -s ' ' | cut -f10 -d' '`
16 if [ ! -z "$memAux" -a "$memAux" != " " ]; then
17   memCount=$(awk "BEGIN {print $memCount + $memAux; exit}")
18   memCounter=$((memCounter+1))
19 fi
20 done
21
22 cpuCount=$(awk "BEGIN {print $cpuCount / $cpuCounter; exit}")
23 memCount=$(awk "BEGIN {print $memCount / $memCounter; exit}")
24
25 echo 'cpu: ' $cpuCount
26 echo 'mem: ' $memCount
```

FIGURA 57 ***SCRIPT DE CONSUMO DE CPU Y MEMORIA DE TOMCAT***
Fuente: Elaboración propia

g. Resultado evaluación de pasarela de pagos

	Instapago						
peticion	tiempo	exito	peticion	tiempo	exito		
1	10.01	1	51	10.63	1	Disp =	0.98
2	6.38	1	52	10.81	1	Avg tiempo=	9.20
3	10.35	1	53	6.71	1		
4	11.03	1	54	7.31	1		
5	12.77	1	55	10.85	1		
6	11.63	0	56	10.16	1		
7	9.60	1	57	6.65	1		
8	11.12	1	58	6.33	1		
9	7.92	1	59	11.29	1		
10	9.83	1	60	8.59	1		
11	8.91	1	61	9.53	0		
12	10.39	1	62	12.79	1		
13	7.50	1	63	11.41	1		
14	9.41	1	64	7.54	1		
15	9.31	1	65	8.34	1		
16	5.57	1	66	6.52	1		
17	11.69	1	67	8.22	1		
18	9.99	1	68	6.94	1		
19	6.94	1	69	10.76	1		
20	6.89	1	70	10.40	1		
21	6.89	1	71	5.54	1		
22	9.42	1	72	10.03	1		
23	7.01	1	73	12.69	1		
24	11.98	1	74	7.79	1		
25	10.99	1	75	10.11	1		
26	10.21	1	76	9.54	1		
27	5.61	1	77	12.41	1		
28	12.43	1	78	12.35	1		
29	8.36	1	79	7.21	1		
30	10.66	1	80	11.86	1		
31	11.72	1	81	5.67	1		
32	5.48	1	82	10.08	1		
33	11.08	1	83	10.65	1		
34	11.68	1	84	7.53	1		
35	5.12	1	85	7.54	1		
36	5.89	1	86	7.54	1		
37	12.41	1	87	9.51	1		
38	6.58	1	88	9.08	1		
39	10.09	1	89	8.60	1		
40	10.31	1	90	9.38	1		
41	8.82	1	91	6.58	1		
42	9.32	1	92	10.29	1		
43	8.56	1	93	5.30	1		
44	6.04	1	94	8.13	1		
45	6.15	1	95	11.15	1		
46	12.42	1	96	11.26	1		
47	12.90	1	97	9.97	1		
48	5.25	1	98	11.99	1		
49	10.74	1	99	8.06	1		
50	11.13	1	100	7.97	1		

FIGURA 58 ***RESULTADOS BENCHMARKS INSTAPAGO***
Fuente: Elaboración propia

	Neopago						
peticion	tiempo	exito	peticion	tiempo	exito		
1	13.28	1	51	26.41	1	Disp =	0.8
2	20.94	1	52	17.14	1	Avg tiempo=	20.51
3	18.60	1	53	27.57	1		
4	15.41	1	54	18.75	1		
5	15.67	1	55	27.36	1		
6	16.01	1	56	26.23	1		
7	15.91	0	57	16.17	1		
8	12.94	0	58	17.94	1		
9	19.56	1	59	15.83	1		
10	19.48	1	60	20.88	1		
11	22.04	1	61	27.34	1		
12	20.01	1	62	27.55	1		
13	18.58	1	63	24.31	1		
14	29.93	1	64	20.37	1		
15	21.81	1	65	11.92	1		
16	17.88	0	66	31.64	1		
17	12.24	0	67	20.32	0		
18	18.34	1	68	11.03	0		
19	14.44	1	69	14.03	0		
20	29.91	1	70	19.65	1		
21	19.64	1	71	11.21	1		
22	28.21	0	72	11.47	1		
23	21.16	0	73	20.27	1		
24	17.88	0	74	24.03	1		
25	22.97	0	75	31.15	1		
26	14.63	0	76	22.35	1		
27	23.60	0	77	28.16	1		
28	23.01	1	78	19.83	1		
29	30.78	1	79	27.57	1		
30	12.12	1	80	15.16	1		
31	23.05	1	81	24.47	0		
32	14.31	1	82	17.93	1		
33	18.54	1	83	24.84	1		
34	24.65	1	84	17.85	1		
35	21.75	1	85	24.56	1		
36	19.58	1	86	25.52	1		
37	29.14	1	87	22.12	1		
38	21.33	0	88	16.28	1		
39	18.72	0	89	11.83	0		
40	23.88	0	90	26.76	1		
41	26.04	1	91	18.84	1		
42	13.09	1	92	17.35	1		
43	19.74	1	93	24.32	1		
44	14.30	1	94	22.50	1		
45	18.14	0	95	28.13	1		
46	19.19	1	96	21.65	1		
47	12.67	1	97	13.01	1		
48	20.84	1	98	28.52	0		
49	26.62	1	99	18.19	1		
50	19.99	1	100	20.00	1		

FIGURA 59 ***RESULTADOS BENCHMARK NEOPAGO***
Fuente: Elaboración propia

	Mercadopago						
peticion	tiempo	exito	peticion	tiempo	exito		
1	8.92	1	51	12.50	1	Disp =	0.99
2	6.63	1	52	9.60	1	Avg tiempo=	9.04
3	10.31	1	53	7.16	1		
4	9.25	1	54	11.77	1		
5	13.10	0	55	10.12	1		
6	9.47	1	56	5.34	1		
7	7.56	1	57	11.76	1		
8	7.23	1	58	7.34	1		
9	9.65	1	59	8.28	1		
10	13.06	1	60	11.05	1		
11	12.71	1	61	11.94	1		
12	13.02	1	62	11.08	1		
13	12.17	1	63	8.31	1		
14	10.07	1	64	6.46	1		
15	10.98	1	65	5.33	1		
16	10.84	1	66	5.39	1		
17	11.19	1	67	7.73	1		
18	12.04	1	68	7.85	1		
19	7.23	1	69	9.35	1		
20	10.19	1	70	10.07	1		
21	11.64	1	71	8.02	1		
22	9.23	1	72	8.74	1		
23	11.14	1	73	8.97	1		
24	12.97	1	74	9.80	1		
25	8.92	1	75	5.44	1		
26	8.85	1	76	11.70	1		
27	13.57	1	77	6.51	1		
28	10.88	1	78	8.35	1		
29	5.36	1	79	9.35	1		
30	5.40	1	80	5.05	1		
31	9.50	1	81	7.24	1		
32	5.51	1	82	5.11	1		
33	10.56	1	83	10.27	1		
34	13.53	1	84	11.70	1		
35	8.21	1	85	6.46	1		
36	13.51	1	86	10.20	1		
37	5.56	1	87	5.97	1		
38	6.25	1	88	9.14	1		
39	12.86	1	89	10.77	1		
40	7.56	1	90	6.36	1		
41	5.93	1	91	5.27	1		
42	12.94	1	92	9.24	1		
43	9.23	1	93	6.71	1		
44	5.34	1	94	10.34	1		
45	9.44	1	95	10.89	1		
46	7.68	1	96	9.96	1		
47	8.43	1	97	10.91	1		
48	9.33	1	98	7.40	1		
49	6.01	1	99	6.31	1		
50	4.42	1	100	7.74	1		

FIGURA 60 ***RESULTADO BENCHMARKS MERCADOPAGO***
Fuente: Elaboración propia

B. Modelo lógico de Base de datos

a. Versión 1.0

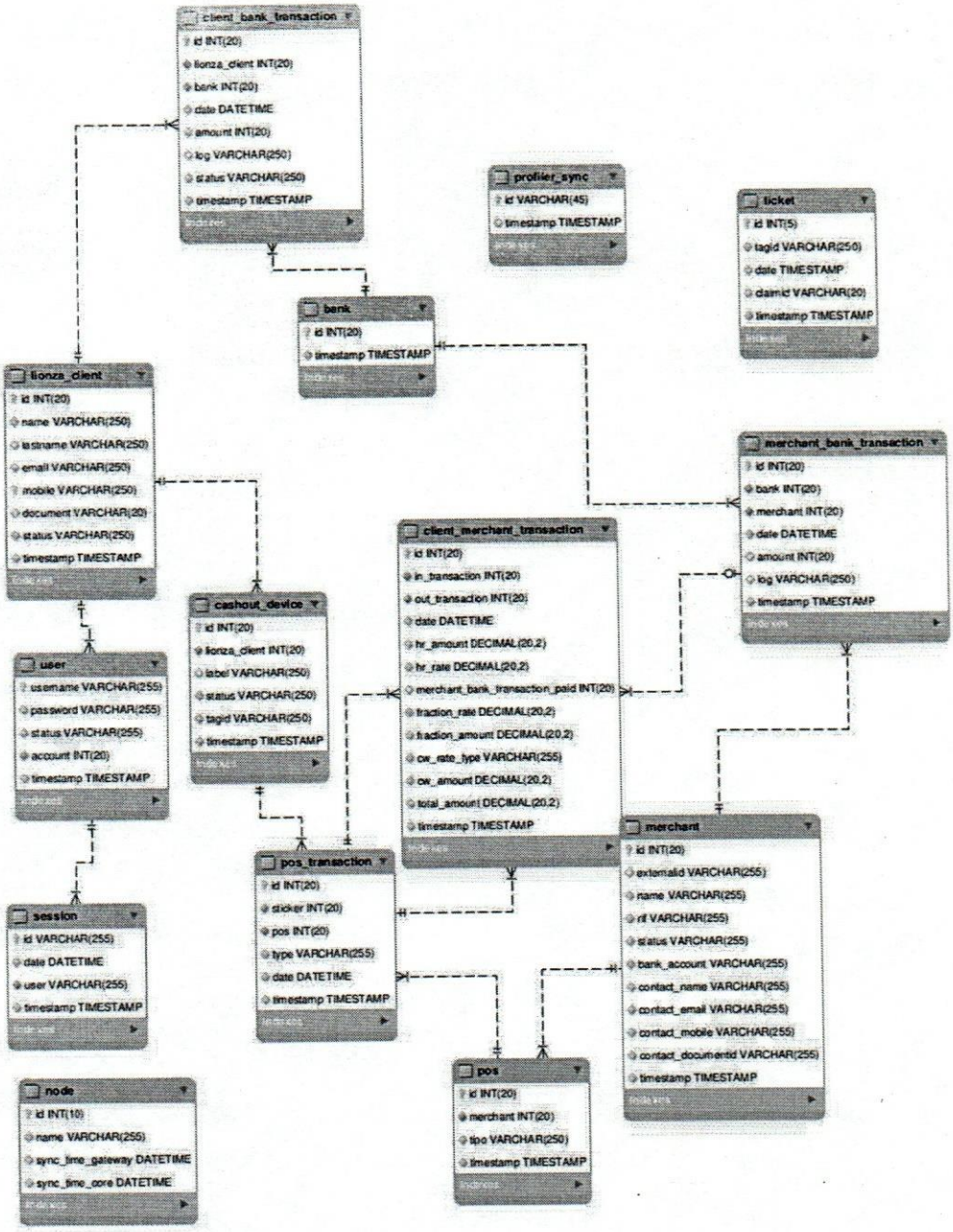

FIGURA 61 ***MODELO LÓGICO BD v1.0***
Fuente: Elaboración propia

b. Version 1.1

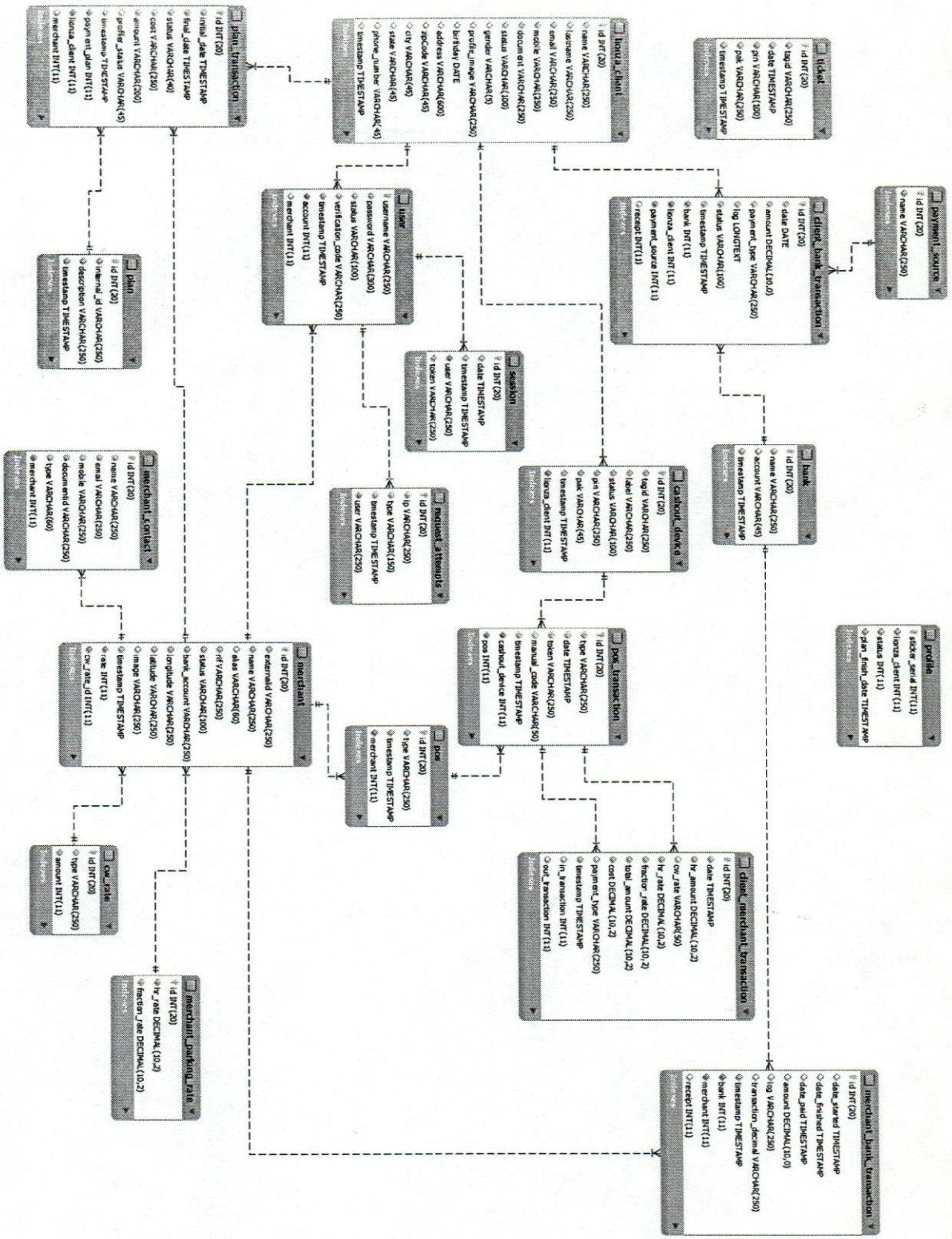

FIGURA 62 ***MODELO LÓGICO DE BASE DE DATOS VERSIÓN 1.1***
Fuente: Elaboración propia

c. Version 1.2

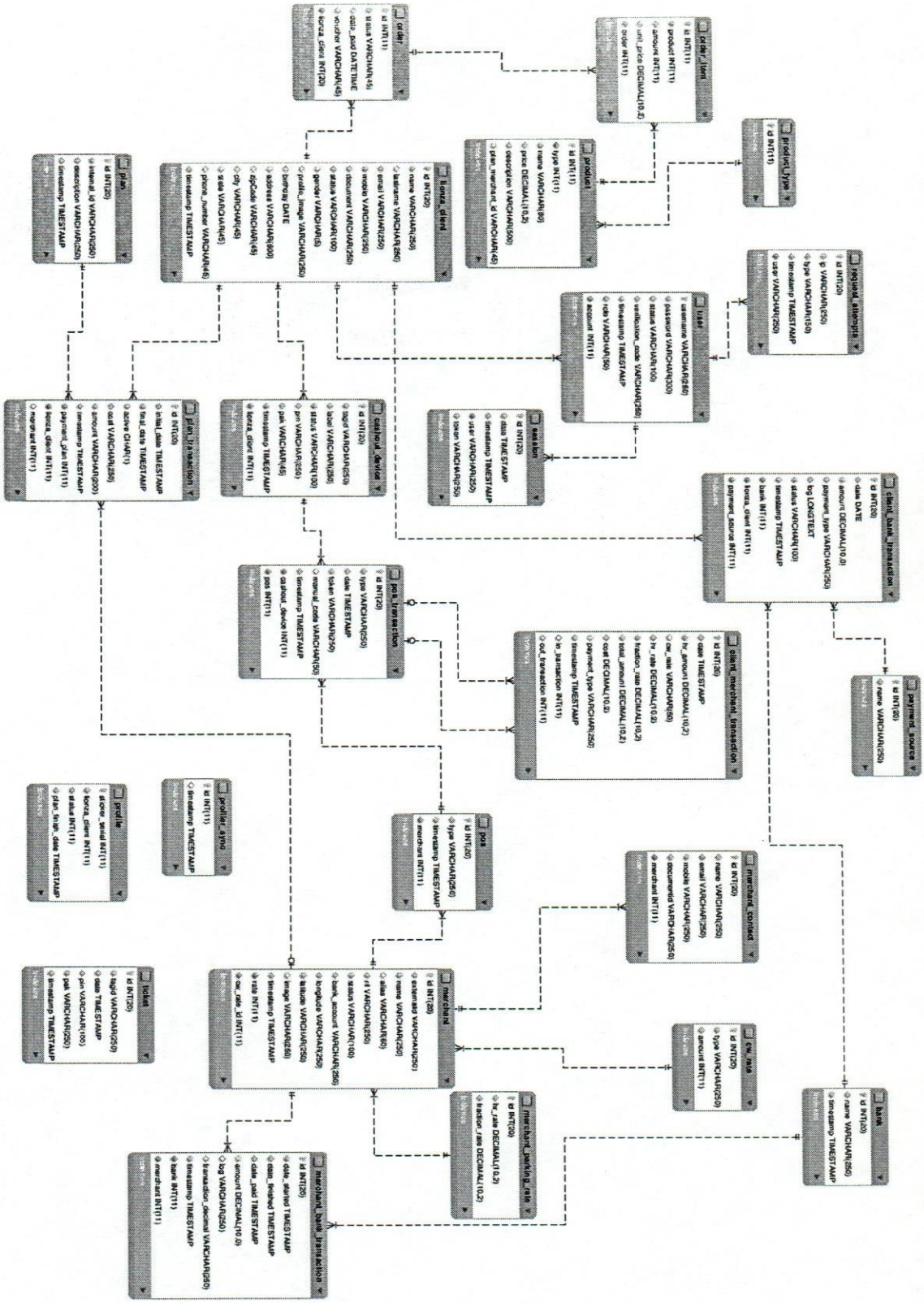

FIGURA 63 ***MODELO LÓGICO DE BASE DE DATOS VERSIÓN 1.2***
Fuente: Elaboración propia

d. Version 1.3

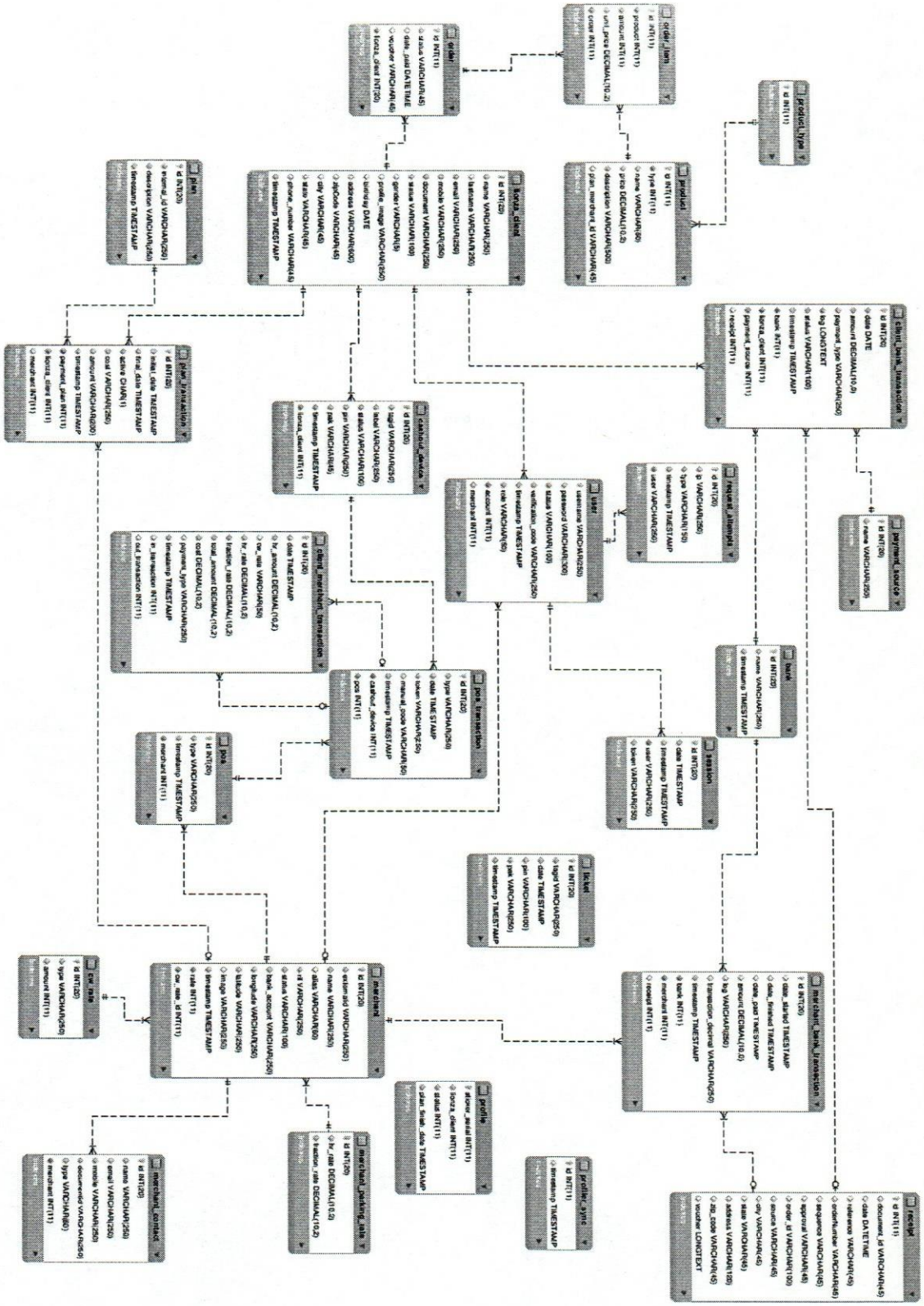

FIGURA 64 ***MODELO LÓGICO DE BASE DE DATOS VERSIÓN 1.3***

Fuente: Elaboración propia

e. Version 1.4

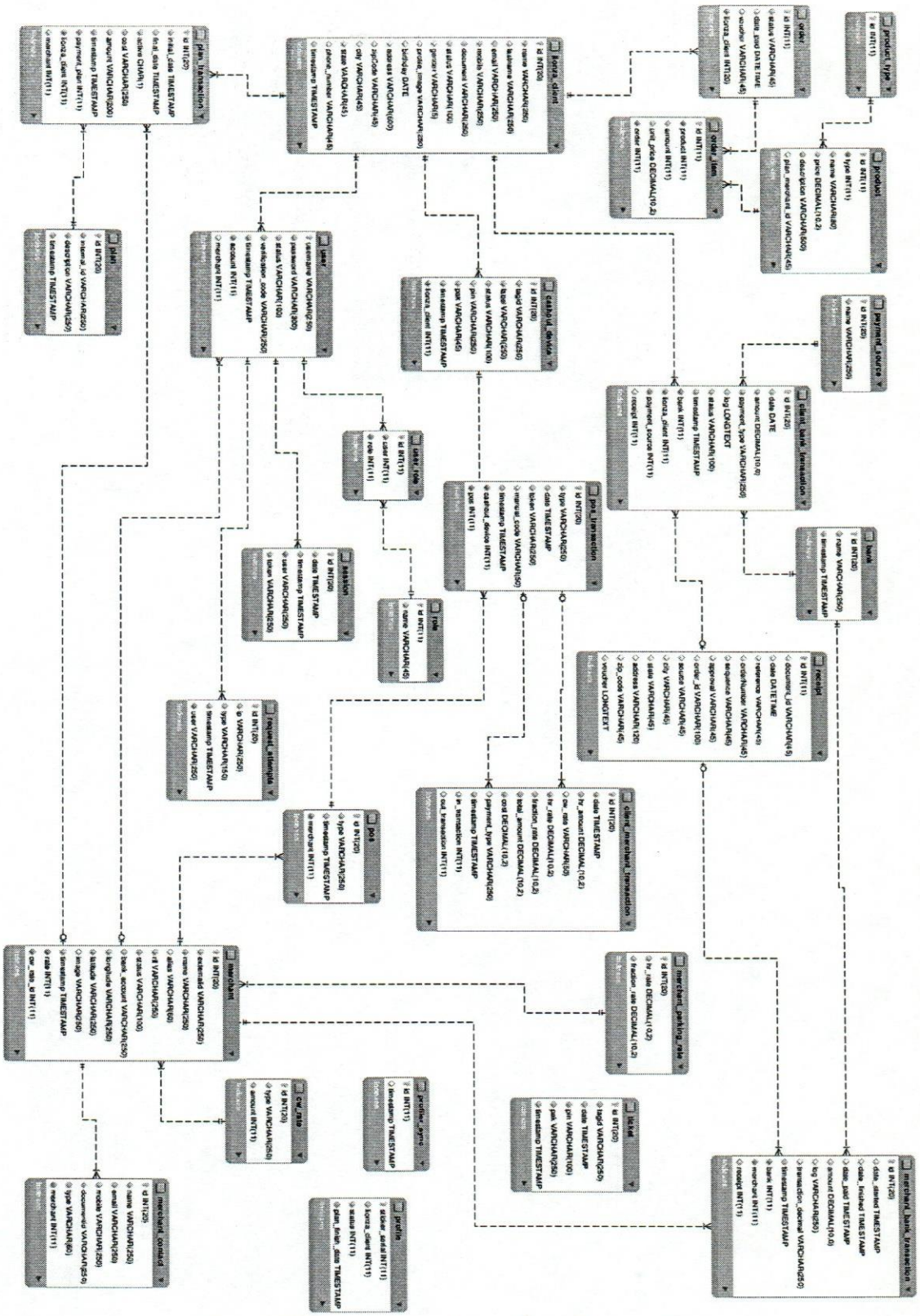

FIGURA 65 ***MODELO LÓGICO DE BASE DE DATOS VERSIÓN 1.4***

Fuente: Elaboración propia

f. Versión 1.5

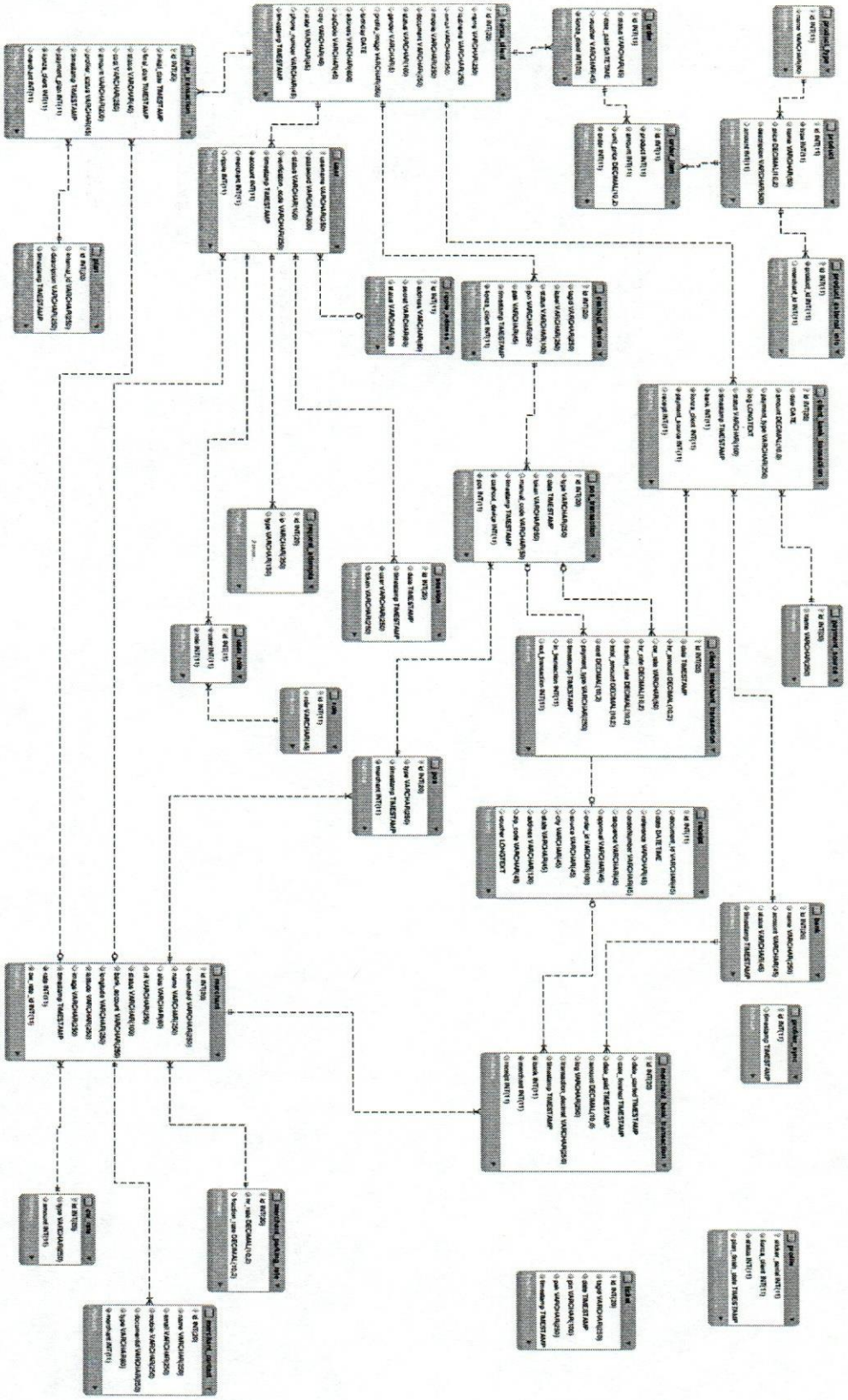

FIGURA 66 ***MODELO LÓGICO DE BASE DE DATOS DE CITYWALLET VERSIÓN 1.5***
Fuente: Elaboración propia

g. Modelo Lógico Base de Datos Gateway

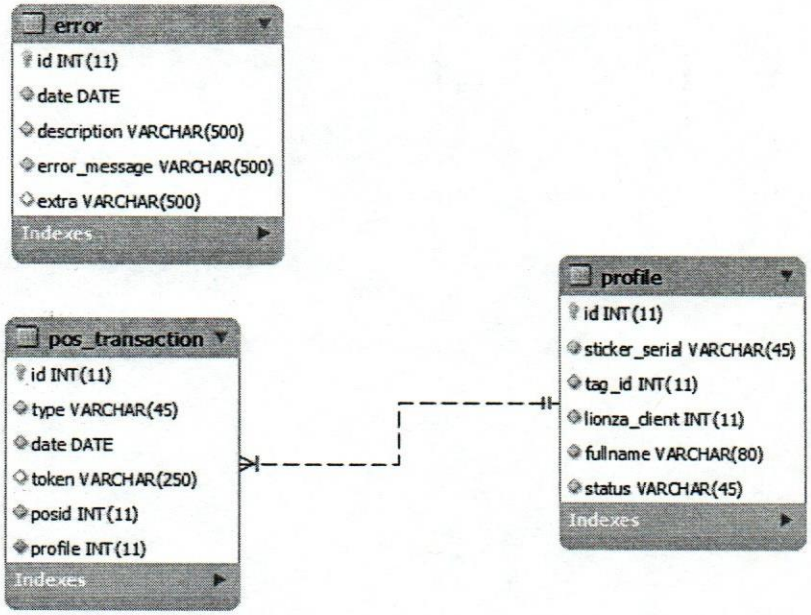

FIGURA 67 ***MODELO LÓGICO DE BASE DE DATOS DE GATEWAY***
Fuente: Elaboración propia

MODELO LÓGICO DE BASE DE DATOS CENTRAL GATEWAY

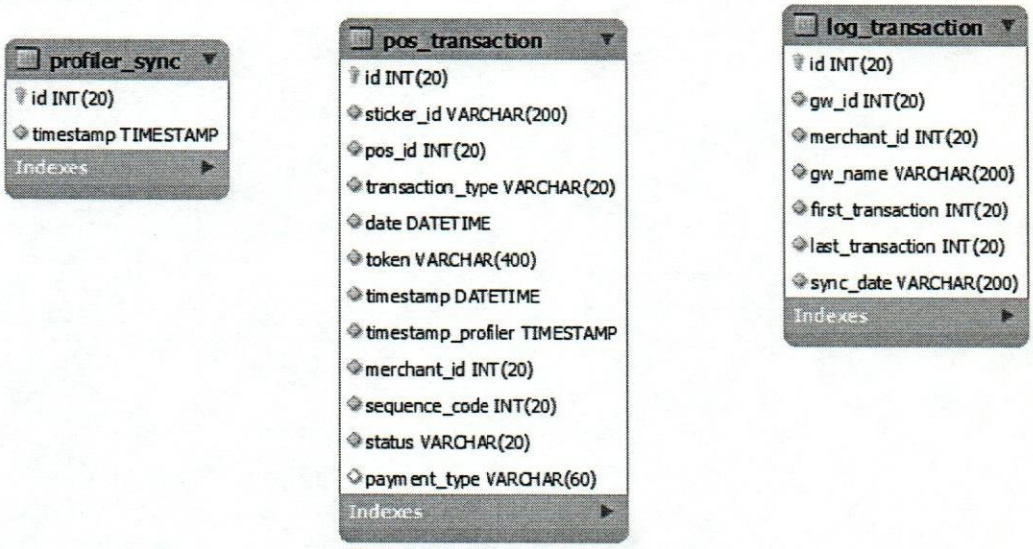

FIGURA 68 ***MODELO LÓGICO DE BASE DE DATOS DE CENTRAL GATEWAY***
Fuente: Elaboración propia

h. Modelo lógico de base de datos Ripple

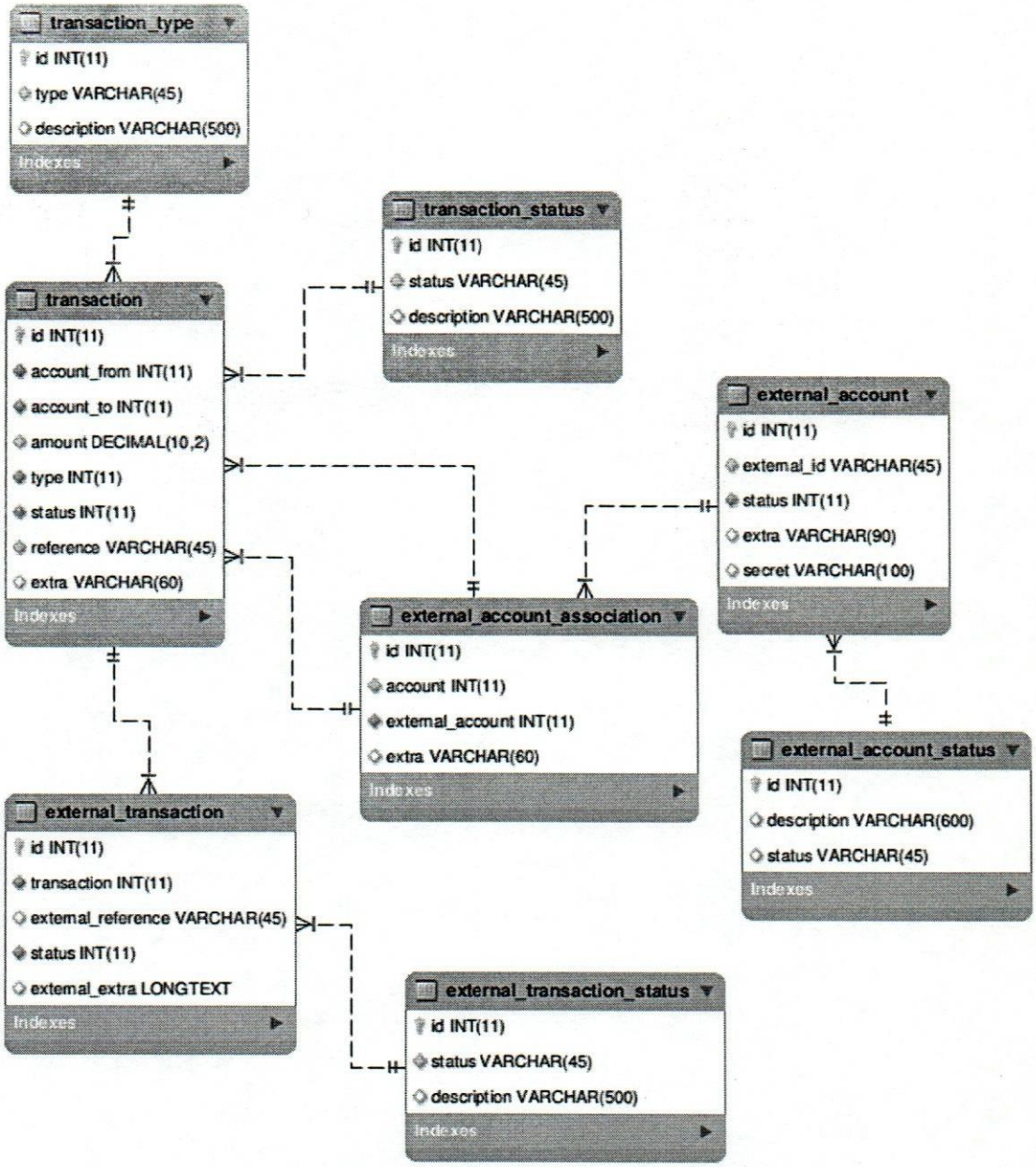

FIGURA 69 ***MODELO LÓGICO DE BASE DE DATOS DE RIPPLE***
Fuente: elaboración propia

C. Sprint Backlogs

a. Sprint 1

		Días									
Historia	**Tarea**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
	Especificación de los requerimientos no funcionales del sistema	1	0	0	0	0	0	0	0	0	0
	Búsqueda de lenguajes a evaluar	1	0	0	0	0	0	0	0	0	

	Comparación de lenguajes de programación con los requerimientos no funcionales.	5	0	0	0	0	0	0	0	0	0
	Selección de frameworks a evaluar	1	0	0	0	0	0	0	0	0	0
	Evaluación de frameworks con respecto a los requerimientos no funcionales	6	6	0	0	0	0	0	0	0	0
PBL-02	Selección de IDE a evaluar	1	1	0	0	0	0	0	0	0	0
	Evaluación de IDE del proyecto	3	3	2	0	0	0	0	0	0	0
	Selección de manejadores de base de datos a utilizar	1	1	1	0	0	0	0	0	0	0
	Evaluación de manejadores de base de datos	10	10	10	5	0	0	0	0	0	0
	Diseño de la arquitectura de los servidores	16	16	16	16	13	5	0	0	0	0
	Diseño de la arquitectura de los POS	9	9	9	9	9	9	6	0	0	0
	Diseño de la arquitectura lógica	6	6	6	6	6	6	6	4	0	0
PBL-06	Diseño del modelo lógico de base de datos	20	20	20	20	20	20	20	20	16	8
	Total	80	72	64	56	48	40	32	24	16	8

FIGURA 70 ***SPRINT BACKLOG, SPRINT 1***
Fuente: Elaboración propia

b. Sprint 2

		Días									
Historia	**Tarea**	1	2	3	4	5	6	7	8	9	10
PBL-06	Cambios en modelo de base de datos	2	0	0	0	0	0	0	0	0	0
PBL-28	Especificación de estándares API	4	0	0	0	0	0	0	0	0	0
	Estándares Documentación API	4	2	0	0	0	0	0	0	0	0
PBL-07	Diagrama de secuencia creación de perfil	4	4	0	0	0	0	0	0	0	0
	Creación de base de datos	1	1	0	0	0	0	0	0	0	0
	Inserts de prueba en base de datos	1	1	0	0	0	0	0	0	0	0
	Implementación Creación de perfil	8	8	8	0	0	0	0	0	0	0
	Diagrama de Secuencia actualizar perfil	4	4	4	4	0	0	0	0	0	0
	Implementación Actualizar perfil	8	8	8	8	4	0	0	0	0	0
	Especificación endpoint Crear perfil en API	2	2	2	2	2	0	0	0	0	0
	Especificación endpoint Actualizar perfil en API	2	2	2	2	2	0	0	0	0	0

	Web service Crear perfil	4	4	4	4	4	4	0	0	0	0
	Web Service Actualizar perfil	4	4	4	4	4	4	0	0	0	0
PBL-08	Diseño de proceso de recuperación de clave	4	4	4	4	4	4	4	0	0	0
	Diagrama de secuencia de recuperación de clave	4	4	4	4	4	4	4	0	0	0
	Implementación recuperación de clave	8	8	8	8	8	8	8	8	0	0
	Especificación endpoint recuperación de clave en API	4	4	4	4	4	4	4	4	4	0
	Web service recuperación de clave	4	4	4	4	4	4	4	4	4	0
PBL-05	Diseño del proceso de pago entre puntos de servicio y CityWallet	8	8	8	8	8	8	8	8	8	8
	Total	**80**	**72**	**64**	**56**	**48**	**40**	**32**	**24**	**16**	**8**

FIGURA 71 ***SPRINT BACKLOG, SPRINT 2***
Fuente: Elaboración propia

c. Sprint 3

		Días									
Historia	**Tarea**	1	2	3	4	5	6	7	8	9	10
PBL-07	Pruebas operaciones de perfil	12	4	0	0	0	0	0	0	0	0
	Corrección de error de operaciones perfil	8	8	4	0	0	0	0	0	0	0
PBL-08	Pruebas recuperación de claves	4	4	4	0	0	0	0	0	0	0
	Corrección de recuperación de clave	4	4	4	4	0	0	0	0	0	0
PBL-19	Diagrama de secuencia de registro de transacción	8	8	8	8	4	0	0	0	0	0
	Implementación registro de transacción	12	12	12	12	12	8	0	0	0	0
PBL-20	Diagrama de secuencia sincronización de transacciones	4	4	4	4	4	4	4	0	0	0
	Implementación sincronización de transacciones	12	12	12	12	12	12	12	8	0	0
PBL-21	Diagrama de secuencia sincronización de perfiles	4	4	4	4	4	4	4	4	4	0
	Implementación de sincronización de perfiles	12	12	12	12	12	12	12	12	12	8
	Total	**80**	**72**	**64**	**56**	**48**	**40**	**32**	**24**	**16**	**8**

FIGURA 72 ***SPRINT BACKLOG, SPRINT 3***
Fuente: Elaboración Propia

d. Sprint 4

		Días									
Historia	**Tarea**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PBL-22	Diagrama de secuencia cálculo de transacciones	2	0	0	0	0	0	0	0	0	0
	Implementación cálculo de transacciones y de balance de usuario	6	0	0	0	0	0	0	0	0	0
	Pruebas de transacciones y balance de usuario	2	2	0	0	0	0	0	0	0	0
	Correcciones de transacciones y balance de usuario	4	4	0	0	0	0	0	0	0	0
PBL-10	Diagrama de secuencia consulta de consumos	2	2	0	0	0	0	0	0	0	0
	Implementación consulta de consumos	6	6	6	0	0	0	0	0	0	0
	Pruebas consulta de consumos	2	2	2	0	0	0	0	0	0	0
	Correcciones consulta de consumos	4	4	4	4	0	0	0	0	0	0
PBL-19	Pruebas registro de transacciones	4	4	4	4	0	0	0	0	0	0
	Correcciones registros de transacciones	8	8	8	8	8	0	0	0	0	0
PBL-20	Pruebas sincronización de transacciones	3	3	3	3	3	3	0	0	0	0
	Correcciones sincronización de transacciones	8	8	8	8	8	8	3	0	0	0
PBL-21	Pruebas sincronización de perfiles	3	3	3	3	3	3	3	0	0	0
	Correcciones sincronizaciones de perfiles	6	6	6	6	6	6	6	4	0	0
PBL-04	Diseño de proceso de asociación de cashout devices	16	16	16	16	16	16	16	16	12	4
PBL-09	Diagrama de secuencia asociación cashout devices	4	4	4	4	4	4	4	4	4	4
	Total	80	72	64	56	48	40	32	24	16	8

FIGURA 73 ***SPRINT BACKLOG, SPRINT 4***
Fuente: Elaboración propia

e. Sprint 5

		Días									
Historia	**Tarea**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PBL-09	Implementación asociación cashout	8	0	0	0	0	0	0	0	0	0

	device										
	Diagrama de secuencia bloqueo cashout device	4	4	0	0	0	0	0	0	0	0
	Implementación bloqueo cashout device	6	6	2	0	0	0	0	0	0	0
	Diagrama secuencia consulta de cashout devices	4	4	4	0	0	0	0	0	0	0
	Implementación consulta cashout devices	6	6	6	4	0	0	0	0	0	0
	Pruebas asociación cashout device	4	4	4	4	0	0	0	0	0	0
	Pruebas bloqueo cashout device	4	4	4	4	4	0	0	0	0	0
	Pruebas consultas cashout devices	4	4	4	4	4	0	0	0	0	0
	Correcciones módulo cashout devices	8	8	8	8	8	8	0	0	0	0
PBL-12	Diseño API y sql reporte transacciones diarias en un comercio	8	8	8	8	8	8	8	0	0	0
PBL-13	Diseño API y sql reporte usuarios sin transacciones	8	8	8	8	8	8	8	8	0	0
PBL-15	Diseño API y sql reporte de usuarios y sus datos	8	8	8	8	8	8	8	8	8	0
PBL-16	Diseño API y sql reporte de transferencias y pagos de los usuarios y los comercios	8	8	8	8	8	8	8	8	8	8
	Total	**80**	**72**	**64**	**56**	**48**	**40**	**32**	**24**	**16**	**8**

FIGURA 74 ***SPRINT BACKLOG, SPRINT 5***
Fuente: Elaboración propia

f. Sprint 6

		Días									
Historia	**Tarea**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PBL-12	Implementación reporte transacciones diarias en un comercio	4	0	0	0	0	0	0	0	0	0
	Pruebas de reporte transacciones diarias en un comercio	2	0	0	0	0	0	0	0	0	0
	corrección de reporte de transacciones diarias en un comercio	4	2	0	0	0	0	0	0	0	0
PBL-13	Implementación reporte usuarios sin transacciones	4	4	0	0	0	0	0	0	0	0
	Pruebas de reporte usuarios sin	2	2	0	0	0	0	0	0	0	0

	transacciones										
	Corrección de reporte usuarios sin transacciones	4	4	4	0	0	0	0	0	0	0
PBL-15	Implementación reporte de usuarios y sus datos	4	4	4	0	0	0	0	0	0	0
	Pruebas reporte de usuarios y sus datos	2	2	2	2	0	0	0	0	0	0
	Corrección reporte de usuarios y sus datos	4	4	4	4	0	0	0	0	0	0
PBL-16	Implementación reporte de transferencias y pagos de los usuarios y los comercios	4	4	4	4	2	0	0	0	0	0
	Pruebas reporte de transferencias y pagos de los usuarios y los comercios	2	2	2	2	2	0	0	0	0	0
	Corrección reporte de las transferencias y pagos de los usuarios y los comercios	4	4	4	4	4	0	0	0	0	0
PBL-14	Diseño API y sql reporte calcomanías sin reclamar	8	8	8	8	8	8	0	0	0	0
	Implementación reporte calcomanías sin reclamar	4	4	4	4	4	4	4	0	0	0
	Pruebas reporte calcomanías sin reclamar	2	2	2	2	2	2	2	0	0	0
	Correcciones reporte calcomanías sin reclamar	4	4	4	4	4	4	4	2	0	0
PBL-02	Investigación de alternativas de proveedores de pagos	8	8	8	8	8	8	8	8	2	0
	Evaluación de implementación alternativas de proveedores de pagos	8	8	8	8	8	8	8	8	8	2
	Evaluación costos proveedores de pagos	2	2	2	2	2	2	2	2	2	2
	Diagrama de proceso de depósitos de dinero dentro del sistema	4	4	4	4	4	4	4	4	4	4
	Diagrama de secuencia pagos dentro del sistema	4	4	4	4	4	4	4	4	4	4
	Total	**84**	**76**	**68**	**60**	**52**	**44**	**36**	**28**	**20**	**12**

FIGURA 75 ***SPRINT BACKLOG, SPRINT 6***
Fuente: Elaboración propia

g. Sprint 7

		Días									
Historia	**Tarea**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PBL-11	Diseño integración con Instapago	6	0	0	0	0	0	0	0	0	0
	Diseño API instapago	2	0	0	0	0	0	0	0	0	0
	Instalación certificado SSL	4	4	0	0	0	0	0	0	0	0
	Cambios en ER para instapago	8	8	4	0	0	0	0	0	0	0
	Implementación integración instapago	16	16	16	12	4	0	0	0	0	0
	Pruebas integración instapago	8	8	8	8	8	4	0	0	0	0
	Correcciones integración instapago	6	6	6	6	6	6	2	0	0	0
PBL-12	Cambios en reportes por cambios en BD y pruebas	6	6	6	6	6	6	6	0	0	0
PBL-13	Cambios en reportes por cambios en BD y pruebas	6	6	6	6	6	6	6	6	0	0
PBL-14	Cambios en reportes por cambios en BD y pruebas	6	6	6	6	6	6	6	6	4	0
PBL-15	Cambios en reportes por cambios en BD y pruebas	6	6	6	6	6	6	6	6	6	2
PBL-16	Cambios en reportes por cambios en BD y pruebas	6	6	6	6	6	6	6	6	6	6
	Total	**80**	**72**	**64**	**56**	**48**	**40**	**32**	**24**	**16**	**8**

FIGURA 76 ***SPRINT BACKLOG, SPRINT 7***
Fuente: Elaboración propia

h. Sprint 8

		Días									
Historia	**Tarea**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PBL-17	Diseño API de módulo de reportes de comercio	8	0	0	0	0	0	0	0	0	0
	Integración reporte de comercio con módulo de comercio	2	2	0	0	0	0	0	0	0	0
	Pruebas de reporte de transacciones diaria de comercio	2	2	0	0	0	0	0	0	0	0
	Correcciones de reporte de transacciones diarias de comercio	2	2	0	0	0	0	0	0	0	0
PBL-18	SQL de reporte de pagos de CityWallet al comercio	2	2	0	0	0	0	0	0	0	0
	Implementación reporte de pagos de	4	4	4	0	0	0	0	0	0	0

	CityWallet al comercio										
	Pruebas de reporte de pagos de CityWallet	1	1	1	0	0	0	0	0	0	0
	corrección de reporte de pagos de CityWallet	2	2	2	0	0	0	0	0	0	0
PBL-24	Diagrama de secuencias proceso de verificación de montos	8	8	8	7	0	0	0	0	0	0
	Implementación de verificación de montos	8	8	8	8	7	0	0	0	0	0
	Pruebas verificador de montos	4	4	4	4	4	3	0	0	0	0
	Correcciones verificador de montos	2	2	2	2	2	2	0	0	0	0
PBL-25	Diagrama de secuencias de notificación de estado de servicios	8	8	8	8	8	8	5	0	0	0
	Listado de servicios a monitorear	1	1	1	1	1	1	1	0	0	0
	Implementación de notificación de estado de servicios	8	8	8	8	8	8	8	6	0	0
	Pruebas notificación de estado de servicios	6	6	6	6	6	6	6	6	4	0
	Correcciones de notificación de estado de servicios	4	4	4	4	4	4	4	4	4	0
PBL-26	Diagrama de secuencia de pernocta	8	8	8	8	8	8	8	8	8	8
	Total	**80**	**72**	**64**	**56**	**48**	**40**	**32**	**24**	**16**	**8**

FIGURA 77 ***SPRINT BACKLOG, SPRINT 8***
Fuente: Elaboración propia

i. Sprint 9

		Días									
Historia	**Tarea**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PBL-26	Implementación proceso de pernocta	8	0	0	0	0	0	0	0	0	0
	Pruebas de proceso de pernocta	6	6	0	0	0	0	0	0	0	0
	Correcciones de procesos de pernocta	4	4	2	0	0	0	0	0	0	0
PBL-03	Investigación Ripple	12	12	12	6	0	0	0	0	0	0
	Investigación blockchain	16	16	16	16	14	6	0	0	0	0
	Diagrama de proceso respaldo de moneda digital con Ripple	16	16	16	16	16	16	14	6	0	0
PBL-23	Diagrama de secuencia respaldo en moneda digital	8	8	8	8	8	8	8	8	6	0

	Integración Ripple	8	8	8	8	8	8	8	8	8	6
	Inicio Implementación respaldo en moneda digital	2	2	2	2	2	2	2	2	2	2
	Total	**80**	**72**	**64**	**56**	**48**	**40**	**32**	**24**	**16**	**8**

FIGURA 78 ***SPRINT BACKLOG, SPRINT 9***
Fuente: Elaboración propia

j. Sprint 10

		Días									
Historia	**Tarea**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PBL-23	Cambios en BD Ripple	8	0	0	0	0	0	0	0	0	0
	Implementación integración con Ripple	48	48	40	32	24	16	8	0	0	0
	Pruebas integración Ripple	8	8	8	8	8	8	8	8	0	0
	corrección de integración Ripple	16	16	16	16	16	16	16	16	16	8
	Total	**80**	**72**	**64**	**56**	**48**	**40**	**32**	**24**	**16**	**8**

FIGURA 79 ***SPRINT BACKLOG, SPRINT 10***
Fuente: Elaboración propia

k. Sprint 11

		Días									
Historia	**Tarea**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PBL-03	Cambios en diagrama de proceso de Ripple	40	32	24	16	8	0	0	0	0	0
PBL-23	BD de Ripple	16	16	16	16	16	16	8	0	0	0
	Cambio en diagrama de secuencia métodos de ripple	16	16	16	16	16	16	16	16	8	0
	API de Ripple	8	8	8	8	8	8	8	8	8	8
	Total	**80**	**72**	**64**	**56**	**48**	**40**	**32**	**24**	**16**	**8**

FIGURA 80 ***SPRINT BACKLOG, SPRINT 11***
Fuente: Elaboración propia

l. Sprint 12

		Días									
Historia	**Tarea**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PBL-23	Migración de código de implementación de ripple	16	8	0	0	0	0	0	0	0	0
	Implementación de cambios nuevos	24	24	24	16	8	0	0	0	0	0
	Pruebas de proceso de respaldo	24	24	24	24	24	24	16	8	0	0

	Correcciones de proceso de respaldo	16	16	16	16	16	16	16	16	16	8
	Total	80	72	64	56	48	40	32	24	16	8

FIGURA 81 ***SPRINT BACKLOG, SPRINT 12***
Fuente: Elaboración propia

m. Sprint 13

		Días									
Historia	**Tarea**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
	Revisión de objetivos y corrección de errores	24	16	8	0	0	0	0	0	0	0
	Total	24	16	8	0	0	0	0	0	0	0

FIGURA 82 ***SPRINT BACKLOG, SPRINT 13***
Fuente: Elaboración propia

D. API

a. Tabla de Errores

Código	Error	Json
999	Internal Error	{"code": 999, "msg": "Internal Error"}
0	Ok	{"code": 0, "msg": "Ok"}
1	Missing input parameter	{"code": 1, "msg": "Missing input parameter"}
2	Invalid input parameter	{"code": 2, "msg": "Invalid input parameter"}
3	Invalid Account id	{"code": 3, "msg": "Invalid account id"}
4	Invalid Sticker id	{"code": 4, "msg": "Invalid Sticker id"}
5	Invalid POS id	{"code": 5, "msg": "Invalid POS id"}
6	User already exists	{"code": 6, "msg": "User already exists"}
7	Invalid user	{"code": 7, "msg": "Invalid user"}
8	Invalid password	{"code": 8, "msg": "Invalid password"}
9	Invalid Account Status to Login	{"code": 9, "msg": "Invalid Account Status to Login"}
10	Invalid activation token	{"code": 10, "msg": "Invalid activation token"}
11	User not allowed	{"code": 11, "msg": "User not allowed"}
12	Invalid sticker status	{"code": 12, "msg": "Invalid sticker status"}
13	Non existing user	{"code": 13, "msg": "Non existing user"}
15	Password mismatch	{"code": 15, "msg": "Password mismatch"}
16	Password length less than 5	{"code": 16, "msg": "Password length less than 5"}
17	Invalid date range	{"code": 17, "msg": "Invalid date range"}
18	**Invalid plan**	{"code": 18, "msg": "Invalid plan"}
100	Insufficient balance	{"code": 100, "msg": "Insufficient balance"}
101	Invalid payment sequence	{"code": 101, "msg": "Invalid payment sequence"}
102	Invalid sticker	{"code": 102, "msg": "Invalid sticker"}
103	Invalid POS	{"code": 103, "msg": "Invalid POS"}
104	Payment rejected by bank	{"code": 104, "msg": "Payment rejected by bank"}
105	Error connecting with InstaPago	{"code": 105, "msg": "Error connecting with InstaPago"}
106	Invalid data sent to InstaPago	{"code": 106, "msg": "Invalid data sent to InstaPago"}
107	Error processing data by InstaPago	{"code": 107, "msg": "Error processing data by InstaPago"}

200	Invalid Session	{"code": 200, "msg": "Invalid Session"}
201	No connection with instapago	{"code": 201, "msg": "No connection with instapago"}
202	Invalid data sent to instapago	{"code": 202, "msg": "Invalid data sent to instapago"}
203	Payment was rejected by the bank	{"code": 203, "msg": "Payment was rejected by bank"}
997	Unable to send email	{"code": 997, "msg": "Unable to send email"}
999	Internal Error	{"code":999, "msg":"Internal Error"}

FIGURA 83 ***TABLA DE ERRORES DE LOS API***
Fuente: Elaboración propia

b. CityWallet

b.a. Login

Request:

Method: POST
URL: /services/v2/login
Authorization: none
Header: ip – String – not null – requesting user IP address

Input:

- **uid** - String – not null – User id, email
- **pwd** – String - not null – user password

Output:

```
{
        "response": {
                "sessionToken": "generated token"
        }
        "error": {
                "code": 0,
                "msg": "OK",
        }
}
```

FIGURA 84 ***MENSAJE DE ÉXITO LOGIN***
FUENTE***: ELABORACIÓN PROPIA***

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**
- **If Missing required input parameters, return 1 code**
- **If Invalid user, return 7 code**
- **If Invalid password, return 8 code**
- **If Invalid account status, return 9 code**
- **If non existing user, return 13 error code**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 1,
```

```
                "msg": "Missing input parameter",
            }
    }
```

FIGURA 85 ***MENSAJE DE ERROR LOGIN***
FUENTE: ***ELABORACIÓN PROPIA***

b.b. Logout

Request:

Method: GET
URL: /services/v2/logout
Authorization: sessionToken:"token"

Input:

- **none**

Output:

```
{
        "response": {
        }
        "error": {
                "code": 0,
                "msg": "OK",
        }
}
```

FIGURA 86 ***MENSAJE DE ÉXITO LOGOUT***
FUENTE: ***ELABORACIÓN PROPIA***

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 999,
                "msg": "Internal Error",
        }
}
```

FIGURA 87 ***MENSAJE DE ERROR LOGOUT***
FUENTE: ***ELABORACIÓN PROPIA***

b.c. GetUser

Request:

Method: Get
URL: /services/v2/getuser
Authorization: sessionToken:"token"

Input:

- none

Output:


```
{
        "response": {
                "id": "1",
                "email": "john@doe.com",
                "name": "John",
                "lastName": "mobile",
                "icon":"img",
                "mobile":"04152554555",
                "documentId":"198765432",
                "dateOfBirth:"dd/MM/yyyy",
                "pins": [
                        {
                            "pin": "AAAA"
                        },
                        {
                            "pin": "AAAB"
                        }
                ]
                "balance": "200",
                "status": "NO_STICKER"
        }
        "error": {
                "code": 0,
                "msg": "OK",
        }
}
```


FIGURA 88 ***MENSAJE DE ÉXITO GETUSER***

FUENTE***: ELABORACIÓN PROPIA***

Considerations:

- **If user has no stickers, status is NO_STICKERS**
- **If user has not verified his account, status is EMAIL_NOT_VERIFIED**
- **If user has no visits, list should be EMPTY**

Error Output:

If success, return 0 code

- **If Generic error, return 999 code**
- **If Missing required input parameters, return 1 code**
- **If Invalid session, return 200 error code**
- **If Invalid user, return 7 code**
- **If Invalid password, return 8 code**
- **If Invalid account status, return 9 code**
- **If non existing user, return 13 error code**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 1,
                "msg": "Missing input parameter",
        }
}
```

FIGURA 89 ***MENSAJE DE ERROR GETUSER***
FUENTE: ***ELABORACIÓN PROPIA***

b.d. CreateAccount

Request:

Method: POST
URL: /services/v2/createaccount
Authorization: none
Header: ip – String – not null – requesting user IP address

Input (for natural person) :

- **name** String – not null – users name
- dateOfBirth – nullable – format: dd/MM/yyyy, users birthday
- **lastName** String – not null – users lastname
- **email** String – not null – users email
- **mobile** String – not null – users mobile
- **documentId** String – not null – users document id
- gender String – nullable – users gender
- pwd String – not null -users password
- cpwd String – not null – confirm password
- icon String – nullable – users profile image
- ***Input (for legal person):***
- **name** String – not null – legal persons name
- **documentId** String – not null – legal id
- address – String – not null – legal person address
- mobile – String -not null – legal person mobile number

- phoneNumber – String –not null – legal person phone number
- email – String – not null – legal persons email
- pwd String – not null -users password
- cpwd String – not null – confirm password
- icon String – nullable – users profile image

Output:

```
{
        "response": {
                "id": "1"
        }
        "error": {
                "code": 0,
                "msg": "OK",
        }
}
```

FIGURA 90 *MENSAJE DE ÉXITO CREATEACCOUNT*
FUENTE: *ELABORACIÓN PROPIA*

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**
- **If Missing required input parameters, return 1 code**
- **If user already exists, return 6 error code**
- **If Invalid email format, return 7 error code**
- **If invalid document input parameter, return 2 code and invalid parameter**
- **If password length is less than 5, return 16 error code**
- **If password mismatch, return 15 error code**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 1,
                "msg": "Missing input parameter",
        }
}
```

FIGURA 91 *MENSAJE DE ERROR CREATEACCOUNT*
FUENTE: *ELABORACIÓN PROPIA*

b.e. ForgotPassword

Request:

Method: POST
URL: /services/v2/forgotpassword
Authorization: none
Header: ip – String – not null – requesting user IP address

Input:

- **uid** String – not null – users id, email

Output:

```
{
        "response": {
        }
        "error": {
                "code": 0,
                "msg": "OK",
        }
}
```

FIGURA 92 ***MENSAJE DE ÉXITO FORGOTPASSWORD***
FUENTE: ***ELABORACIÓN PROPIA***

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**
- **If Missing required input parameters, return 1 code**
- **If Invalid input parameter, return 2 code**
- **If Invalid user, return 7 code**
- **If non existing user, return 13 code**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 1,
                "msg": "Missing input parameter",
        }
}
```

FIGURA 93 ***MENSAJE DE ERROR FORGOTPASSWORD***
FUENTE: ***ELABORACIÓN PROPIA***

b.f. ValidateCreateAccountToken

Request:

Method: Get
URL: /services/v2/validatecreateaccounttoken
Authorization: none

Input:

- **token** String – not null – users token, sent by email

Output:

```
{
        "response": {
        }
        "error": {
                "code": 0,
                "msg": "OK",
        }
}
```

FIGURA 94 ***MENSAJE DE ÉXITO VALIDATECREATEACCOUNTTOKEN***
FUENTE: ***ELABORACIÓN PROPIA***

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**
- **If Missing required input parameters, return 1 code**
- **If Invalid input parameter, return 2 code**
- **If Invalid user, return 7 code**
- **If non existing user, return 13 code**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 1,
                "msg": "Missing input parameter",
        }
}
```

FIGURA 95 ***MENSAJE DE ERROR VALIDATECREATEACCOUNTTOKEN***
FUENTE: ***ELABORACIÓN PROPIA***

b.g. ResendCreateEmail

Request:

Method: POST

URL: /services/v2/resendcreateemail
Authorization: none

Input:

- **email** String – not null – users email

Output:

```
{
        "response": {
        }
        "error": {
                "code": 0,
                "msg": "OK",
        }
}
```

FIGURA 96 ***MENSAJE DE ÉXITO RESENDCREATEEMAIL***
FUENTE: ***ELABORACIÓN PROPIA***

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**
- **If Missing required input parameters, return 1 code**
- **If Invalid input parameter, return 2 code**
- **If Invalid user, return 7 code**
- **If non existing user, return 13 code**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 1,
                "msg": "Missing input parameter",
        }
}
```

FIGURA 97 ***MENSAJE DE ERROR RESENDCREATEEMAIL***
FUENTE: ***ELABORACIÓN PROPIA***

b.h. AddSticker

Request:

Method: POST
URL: /services/v2/addsticker
Authorization: sessionToken: "token"

Header: **ip** – String – not null – requesting user IP address

Input:

- **pin** String – not null – stickers pin
- **pak** String – not null – pak id

Output:

```
{
        "response": {
      "sticker":
        }
        "error": {
                "code": 0,
                "msg": "OK",
        }
}
```

FIGURA 98 ***MENSAJE DE ÉXITO ADDSTICKER***
FUENTE: ***ELABORACIÓN PROPIA***

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**
- **If Missing required input parameters, return 1 code**
- **If Invalid input parameter, return 2 code**
- **If invalid ticket parameter, return 10 error code**
- **If invalid session, return 200**
- **If Invalid user, return 7 code**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 1,
                "msg": "Missing input parameter",
        }
}
```

FIGURA 99 ***MENSAJE DE ERROR ADDSTICKER***
FUENTE: ***ELABORACIÓN PROPIA***

b.i. GetVisits

Request:

Method: Get

URL: /services/v2/getvisits
Authorization: sessionToken:"token"

Input:

- skip – String – not null – list skip
- limit -String- not null -list limit

Output:


```
{
        "response": {
      "visits":[
            {
              "id":"1",
              "merchantId":"4",
              "merchantIcon": "serialized",
              "merchantName": "Merchant",
              "in_id":"1",
               "out_id":"2",
              "in_date":"dd/MM",
              "out_date":"dd/MM",
               "in_hour":"HH:mm",
              "out_hour":"HH:mm",
              "cost":"200.00"
            }
            {
             "id":"2",
              "merchantId":"4",
              "merchantIcon": "serialized",
              "merchantName": "Merchant",
              "in_id":"3",
               "out_id":"4",
              "in_date":"dd/MM",
              "out_date":"dd/MM",
              "in_hour":"HH:mm",
              "out_hour":"HH:mm",
              "cost":"200.00"
            }
        }
        "error": {
                "code": 0,
                "msg": "OK",
        }
}
```


FIGURA 100 ***MENSAJE DE ÉXITO GETVISITS***
FUENTE***: ELABORACIÓN PROPIA***

Considerations:

- If user has no visits, list should be EMPTY

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**
- **If Invalid session, return 200 error code**
- **If Invalid user, return 7 code**
- **If Invalid account status, return 9 code**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 1,
                "msg": "Missing input parameter",
        }
}
```

FIGURA 101 ***MENSAJE DE ERROR GETVISITS***
FUENTE: ***ELABORACIÓN PROPIA***

b.j. SetPassword

Request:

Method: POST
URL: /services/v2/setpassword
Authorization: sessionToken:"token"
Header: ip – String – not null – requesting user IP address

Input:

- **pwd –** String – not null – old password
- **cpwd** – String – not null – confirm password
- **npwd** -String- not null – new password

Output:

```
{
        "response": {
    "id":"1"
        }
        "error": {
                "code": 0,
                "msg": "OK",
        }
}
```

FIGURA 102 ***MENSAJE DE ÉXITO SETPASSWORD***
FUENTE: ***ELABORACIÓN PROPIA***

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**
- **If Invalid session, return 200 error code**
- **If Invalid user, return 7 code**
- **If Invalid account status, return 9 code**
- **If Invalid input parameter, return 2 code**
- **If invalid password, return 8 code**
- **If password < 5, return 16 code**
- **If password confirmation mismatch, return 11 code**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 1,
                "msg": "Missing input parameter",
        }
}
```

FIGURA 103 ***MENSAJE DE ERROR SETPASSWORD***
FUENTE: ***ELABORACIÓN PROPIA***

b.k. GetStickers

Request:

Method: Get
URL: /services/v2/getstickers
Authorization: sessionToken:"token"
Header: ip – String – not null – requesting user IP address

Input:

- *none*

Output:

```
{
        "response": {
      "stickers":[
                {
                "id":"4",
                "pin":"AAAA",
                "client":1234,
                "status":"ACTIVE",
```

```
                "tagId":"58ADA8664"
                },
                {
                "id":"4",
                "pin":"AAAA",
                "client":1234,
                "status":"ACTIVE",
                "tagId":"58ADA8664"
                }
            ]
        }
        "error": {
                "code": 0,
                "msg": "OK",
        }
    }
```

FIGURA 104 ***MENSAJE DE ÉXITO GETSTICKERS***
FUENTE: ***ELABORACIÓN PROPIA***

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**
- **If Invalid session, return 200 error code**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 999,
                "msg": "Internal Error",
        }
}
```

FIGURA 105 ***MENSAJE DE ERROR GETSTICKERS***
FUENTE: ***ELABORACIÓN PROPIA***

b.I. UpdateAccount

Request:

Method: POST
URL: /services/v2/createaccount
Authorization: sessionToken="session"
Header: ip – String – not null – requesting user IP address

Input (for natural person) :

- **name** String – not null – users name
- dateOfBirth – nullable – format: dd/MM/yyyy, users birthday
- **lastName** String – not null – users lastname
- **mobile** String – not null – users mobile
- **documentId** String – not null – users document id
- gender String – nullable – users gender
- icon String – nullable – users profile image
- ***Input (for legal person):***
- **name** String – not null – legal persons name
- address – String – not null – legal person address
- mobile – String -not null – legal person mobile number
- phoneNumber – String –not null – legal person phone number
- icon String – nullable – users profile image

Output:

```
{
        "response": {
                "id": "1"
        }
        "error": {
                "code": 0,
                "msg": "OK",
        }
}
```

FIGURA 106 ***MENSAJE DE ÉXITO UPDATEACCOUNT***
FUENTE: ***ELABORACIÓN PROPIA***

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**
- **If Missing required input parameters, return 1 code**
- **If user already doesnt exist, return 13 error code**
- **If invalid document input parameter, return 2 code and invalid parameter**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 1,
                "msg": "Missing input parameter",
        }
```

```
}
```

FIGURA 107 *MENSAJE DE ERROR UPDATEACCOUNT*
FUENTE: *ELABORACIÓN PROPIA*

b.m. GetVisit

Request:

Method: Get
URL: /services/v2/getvisit
Authorization: sessionToken:"token"

Input:

- **visitId** – String – not null – plans id

Output:

```
{
        "response": {
            "id":"2",
            "merchantName":"merchant",
            "merchantIcon":"merchant",
            "in_date":"dd/MM/yyyy",
            "out_date":"dd/MM/yyyy",
            "in_hour":"08:00am",
            "out_hour":"09:30pm",
          "duration":"13.30h",
          "hr_rate":"13.44",
          "cw_rate":"49.99",
          "total_amount":"145.50"
        }
        "error": {
              "code": 0,
              "msg": "OK",
        }
}
```

FIGURA 108 *MENSAJE DE ÉXITO GETVISIT*
FUENTE: *ELABORACIÓN PROPIA*

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**
- **If Missing required input parameters, return 1 code**

Example:

```
{
        "response": {
```

```
        }
        "error": {
                "code": 1,
                "msg": "Missing input parameter",
        }
}
```

FIGURA 109 *MENSAJE DE ERROR GETVISIT*
FUENTE: *ELABORACIÓN PROPIA*

b.n. CreatePaymentInstapago

Request:

Method: POST
URL: /services/v2/instapago/createpayment
Authorization: sessionToken:"token"

Input:

- **amount –** String – not null – plans id
- **cardHolderName –** String – not null – name thats on the card
- **cardHolderId –** String – not null – document id of the cardholder, must be sent in a "V-20000000" format where V can be: J,G,E or P
- **cardNumber –** String – not null – number on the card
- **expirationDate –** String – not null – format: MM/yyyy
- **cvc –** String – not null – cards cvc
- **address –** String – nullable – users address
- **city –** String – nullable – users city
- **zipCode –** String – nullable – users zipcode
- **state –** String – nullable – users state

Output:

```
{
        "response": {
         "voucher":"ENCODED HTML WITH VOUCHER"
        }
        "error": {
                "code": 0,
                "msg": "OK",
        }
}
```

FIGURA 110 *MENSAJE DE ÉXITO CREATEPAYMENTINSTAPAGO*
FUENTE: *ELABORACIÓN PROPIA*

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**
- **If Missing required input parameters, return 1 code**

- If Invalid Input parameter, return 2 code
- If no connection with instapago, return 201 code
- If invalid data sent to instapago, return 202 code
- If payment rejected, return 203 code

Example:

```
{
        "response": {
        }
        "error": {
                "code": 1,
                "msg": "Missing input parameter",
        }
}
```

FIGURA 111 ***MENSAJE DE ERROR CREATEPAYMENTINSTAPAGO***
FUENTE: ***ELABORACIÓN PROPIA***

b.o. GetPaymentSliderInfo

Request:

Method: Get
URL: /services/v2/getpaymentsliderinfo
Authorization: sessionToken:"token"

Input:

- none

Output:

```
{
        "response": {
         "minAmount":"250",
         "maxAmount":"10000",
          "step":"50"
        },
        "error": {
                "code": 0,
                "msg": "OK"
        }
}
```

FIGURA 112 ***MENSAJE DE ÉXITO GETPAYMENTSLIDERINFO***
FUENTE: ***ELABORACIÓN PROPIA***

Error Output:

- If success, return 0 code
- If Generic error, return 999 code

Example:

```
{
        "response": {
        }
        "error": {
                "code": 999,
                "msg": "Internal Error"
        }
}
```

FIGURA 113 *MENSAJE DE ERROR GETPAYMENTSLIDERINFO*
FUENTE: *ELABORACIÓN PROPIA*

b.p. CreateCashInNotification

Request:

Method: POST
URL: /services/v2/createcashinnotification
Authorization: sessionToken:"token"
Header:ip - String - not null

Input:

- **-reference** - String - not null
- **-amount** - String - not null
- **-bankFrom-** String - not null
- **-bankTo** - String - not null
- **-type -** String - not null

Output:

```
{
        "response": {

        },
        "error": {
                "code": 0,
                "msg": "OK"
        }
}
```

FIGURA 114 *MENSAJE DE ÉXITO CREATECASHINNOTIFICATION*
FUENTE: *ELABORACIÓN PROPIA*

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**
- **If invalid session return 200 code**
- **If invalid input parameter return 2 code**

- If missing input parameter return 1 code

Example:

```
{
        "response": {
        }
        "error": {
                "code": 999,
                "msg": "Internal Error"
        }
}
```

FIGURA 115 ***MENSAJE DE ERROR CREATECASHINNOTIFICATION***
FUENTE: ***ELABORACIÓN PROPIA***

b.q. GetCashIns

Request:

Method: GET
URL: /services/v2/getcashins
Authorization: sessionToken:"token"

Input:

- **-skip** - String- not null
- **-limit** - String - not null

Output:

```
{
        "response": {
    cashIns:[
      {
        "id":"1",
        "type":"transfer",
        "status":"NOT_PROCESSED",
        "amount":"5000",
        "date":"13/04/2016"
      },
      {
        "id":"2",
        "type":"credit card",
        "status":"VALID",
        "amount":"4000",
        "date":"20/05/2016"
          }
      ]
          },
        "error": {
```

```
                "code": 0,
                "msg": "OK"
            }
    }
```

FIGURA 116 *MENSAJE DE ÉXITO GETCASHINS*
FUENTE: *ELABORACIÓN PROPIA*

Considerations

- **Status values are: VALID, NOT_PROCESSED**
- **Type values are: transfer, deposit or credit card**

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**
- **If invalid session return 200 code**
- **If invalid input parameter return 2 code**
- **If missing input parameter return 1 code**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 999,
                "msg": "Internal Error"
        }
}
```

FIGURA 117 *MENSAJE DE ERROR GETCASHINS*
FUENTE: *ELABORACIÓN PROPIA*

b.r. PosTransaction

Request:

Method: POST
URL: /services/v2/postransaction
Authorization: "posAuth:"token"

Input:

- **nfc- not null – sticker serial**
- **posId – not null – pos's id**
- **type – not null – transaction type**

Output:

```
{
        "response": {
        },
```

```
        "error": {
                "code": 0,
                "msg": "OK"
        }
}
```

FIGURA 118 *MENSAJE DE ÉXITO POSTRANSACTION*
FUENTE: *ELABORACIÓN PROPIA*

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**
- **If invalid session return 200 code**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 999,
                "msg": "Internal Error"
        }
}
```

FIGURA 119 *MENSAJE DE ERROR POSTRANSACTION*
FUENTE: *ELABORACIÓN PROPIA*

b.s. UpdateSticker

Request:

Method: POST
URL: /services/v2/updatesticker
Authorization: sessionToken:"token"

Input:

- **id** -not null – sticker id
- **status** – not null – status to be set

Output:

```
{
        "response": {
        }
        "error": {
                "code": 0,
                "msg": "OK"
        }
}
```

FIGURA 120 *MENSAJE DE ÉXITO UPDATESTICKER*
FUENTE: *ELABORACIÓN PROPIA*

Error Output:

- If success, return 0 code
- If Generic error, return 999 code
- If invalid session return 200 code
- If missing input parameter, return 1 code
- If invalid input parameter, return 2 code

Example:

```
{
        "response": {
        }
        "error": {
                "code": 999,
                "msg": "Internal Error"
        }
}
```

FIGURA 121 *MENSAJE DE ERROR UPDATESTICKER*
FUENTE: *ELABORACIÓN PROPIA*

b.t. GetActiveBanks

Request:

Method: GET
URL: /services/v2/getactivebanks
Authorization: sessionToken:"token"

Input:

Output:

```
{
        "response": {
    banks:[
      {
        "id":"1",
        "bank":"Banesco",
        "documentId":".",
        "accountType":"Corriente",
        "email":"recargas@citywallet.co",
        "account":"0123-1564-213135315133",
        "owner":"CityWallet Group C.A."
      },
      {
        "id":"2",
        "bank":"BanCaribe",
```

```
            "documentId":"J-1234568-9",
            "accountType":"Corriente",
            "email":"recargas@citywallet.co",
            "account":"0123-1564-213135315133",
            "owner":"CityWallet Group C.A."
              }
            ]
              },
              "error": {
                        "code": 0,
                        "msg": "OK"
              }
      }
```


FIGURA 122 ***MENSAJE DE ÉXITO GETACTIVEBANKS***
FUENTE: ***ELABORACIÓN PROPIA***

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**
- **If invalid session return 200 code**

Example:


```
{
            "response": {
            }
            "error": {
                        "code": 999,
                        "msg": "Internal Error"
            }
}
```


FIGURA 123 ***MENSAJE DE ERROR GETACTIVEBANKS***
FUENTE: ***ELABORACIÓN PROPIA***

c. Módulo Admin

c.a. Login

Request:

Method: Get
URL: /services/v2/useradmin/login
Authorization: none
Header: ip – String – not null – requesting user SessionToken

Input:

- **uid** - String – not null – User id, email
- **pwd** – String - not null – user password

Output:

```
{
  "response":{
 "sessionToken":"11111"
}
  "error":{"code":0,"msg":"Ok"}
}
```

FIGURA 124 *EJEMPLO DE RESPUESTA DE ÉXITO LOGIN EN ADMIN*
Fuente: Elaboración propia

Error Output:

- **If success,return 0 error code.**
- **If Generic Error, return 999 error code**
- **If Missing required parameters, return 001 error code**
- **If Invalid user, return 007 error code**
- **If Invalid password,return 008**
- **If Invalid account status, return 009 error code**
- **If non existing user, return 013 error code**
- **If Invalid account status, return 009 error code**
- **Invalid password: 008**
- **Crear sesión en el servidor.**
- **Response the jsession Id on the variable sessionToken.**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 1,
                "msg": "Missing input parameter",
        }
}
```

FIGURA 125 *EJEMPLO MENSAJE DE FALLA LOGIN DE ADMIN*
Fuente: Elaboración propia

c.b. Logout

Request:

Method: Get
URL: /services/v2/useradmin/logout

Input:

none

Output:

```
{

  "response":{},
  "error":{"code":0,"msg":"Ok"}

}
```

FIGURA 126 *EJEMPLO DE RESPUESTA DE ÉXITO DE LOGOUT ADMIN*
Fuente: Elaboración propia

Error Output:

- **If generic error, returns error code 999.**
- **Must remove current user session.**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 999,
                "msg": "Internal Error",
        }
}
```

FIGURA 127 *EJEMPLO DE RESPUESTA DE FALLA LOGOUT ADMIN*
Fuente: Elaboración propia

c.c. GetUsersWithoutSticker

Request:

Method: Get
URL: /services/v2/useradmin/getuserswithoutsticker

Input:

- **sessionToken** - String - not null
- **limit** - String - not null
- **skip** - String - not null

Output:

```
{
 "response":{
  "users":[
    {
     "name" : "Carlos",
     "lastName" : "Unda",
     "mobile" : "04142021493",
     "email" : "cunda@citywallet.co"
```

```
        }
    ]
  },
  "error":{
    "code": 0,
    "msg": "OK"
  }
}
```

FIGURA 128 *EJEMPLO DE RESPUESTA DE ÉXITO OBTENER USUARIOS SIN CALCOMANÍAS*
Fuente: Elaboración propia

Error Output:

- **If success, return 0 error code**
- **If Generic Error, return 999 error code**
- **If invalid input parameter, return 002 error code**
- **If missing a required input parameter, return 001 code**
- **If invalid session, return 200 error code**
- **skip – defines where in the list to start loading the next set of records when user scrolls. For example, if skip is set to 10, then load records 11, 12, 13...**
- **limits defines how many records to show (load)**
- **returns the list of users without sticker.**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 001,
                "msg": "Missing input parameter",
        }
}
```

FIGURA 129 *EJEMPLO DE RESPUESTA DE FALLO DE OBTENER USUARIOS SIN CALCOMANÍAS*
Fuente: Elaboración propia

c.d. GetUsersWithoutTransactions

Request:

Method: Get
URL: /services/v2/useradmin/getuserswithouttransactions

Input:

- **sessionToken** - String - not null
- **limit** - String - not null
- **skip** - String - not null

Output:

```
{
 "response":{
  "users":[
    {
      "name" : "Carlos",
      "lastName" : "Unda",
      "mobile" : "04142021493",
      "email" : "cunda@citywallet.co"
    }
  ]
 },
 "error":{
  "code": 0,
  "msg": "Ok"
 }
}
```


FIGURA 130 *EJEMPLO DE RESPUESTA DE ÉXITO USUARIOS SIN TRANSACCIONES*
Fuente: Elaboración propia

Error Output:

- **If success, return 0 error code**
- **If Generic Error, return 999 error code**
- **If invalid input parameter, return 002 error code**
- **If missing a required input parameter, return 001 code**
- **If invalid session, return 200 error code**
- **skip – defines where in the list to start loading the next set of records when user scrolls. For example, if skip is set to 10, then load records 11, 12, 13...**
- **limits defines how many records to show (load)**
- **returns the list of users without sticker.**

Example:


```
{
        "response": {
        }
        "error": {
                "code": 001,
                "msg": "Missing input parameter",
        }
}
```


FIGURA 131 *EJEMPLO DE MENSAJE DE FALLO USUARIOS SIN TRANSACCIONES*
Fuente: Elaboración propia

c.e. GetUnclaimedStickers

Request:

Method: Get

URL: /services/v2/useradmin/getunclaimedstickers

Input:

- **sessionToken** - String - not null
- **limit** - String - not null
- **skip** - String - not null

Output:


```
{
 "response":{
  "stickers":[
   {
    "id":1,
    "name": "04121ASDSDF54",
    "pin": "AAA",
    "pak": "1234"
   },
   {
    "id":1,
    "name": "04121ASDSDF54",
    "pin": "AAA",
    "pak": "1234"
   }
  ]
 },
  "error":{
   "code": 0,
   "msg": "OK"
  }
 }
```


FIGURA 132 ***EJEMPLO RESPUESTA ÉXITO CALCOMANÍAS SIN RECLAMAR***
Fuente: Elaboración propia

Error Output:

- **If success, return 0 error code**
- **If Generic Error, return 999 error code**
- **If invalid input parameter, return 002 error code**
- **If missing a required input parameter, return 001 code**
- **If invalid session, return 200 error code**
- **skip – defines where in the list to start loading the next set of records when user scrolls. For example, if skip is set to 10, then load records 11, 12, 13...**
- **limits defines how many records to show (load)**
- **returns the list of unclaimed stickers.**

Example:


```
{
        "response": {
```

```
                }
                "error": {
                        "code": 001,
                        "msg": "Missing input parameter",
                }
        }
```

c.f. GetAllUsers

Request:

Method: Get

URL: /services/v2/useradmin/getallusers

Input:

- **sessionToken** - String - not null
- **ip** - String - not null

Output:

```
{
 "response": {
        "users": [
        {
        "uid": 1,
        "name": "Carlos",
        "last_name": "Unda",
   "document": "V-20800824",
   "email": "cunda@citywallet.co",
        "role": "role name",
        "status": "active"
        },
        {
        "uid": 2,
        "name": "María",
        "last_name": "Unda",
        "document": "V-20800824",
        "email": "cunda@citywallet.co",
        "role": "role name",
        "status": "active"
        }
 },
 "error": {
  "code": 0,
        "msg": "Ok"
 }
}
```

FIGURA 133 ***EJEMPLO MENSAJE DE ÉXITO LISTA DE USUARIO***

Fuente: Elaboración propia

Error Output:

- **If success, return 0 error code**

- If Generic Error, return 999 error code
- **If invalid input parameter, return 002 error code**
- **If missing a required input parameter, return 001 code**
- **If invalid session, return 200 error code**
- **returns the last transaction of each user**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 001,
                "msg": "Missing input parameter",
        }
}
```

FIGURA 134 ***EJEMPLO DE MENSAJE DE FALLO LISTA DE USUARIO***
Fuente: Elaboración propia

c.g. GetTransactionsByMerchant

Request:

Method: Get

URL: /services/v2/useradmin/gettransactionsbymerchant

Input:

- **sessionToken** - String - not null
- **merchant_name** – String – not null
- **star_date** – Date – not null
- **end_date** – Date –not null
- **limit** – String – not null
- **skip** – String – not null

Output:

```
{
 "response":{
  "transactions":[
    {
     "name" : "Carlos",
     "lastName" : "Unda",
     "mobile" : "04142021493",
     "email" : "cunda@citywallet.co",
     "in_date" : "2015/12/14 18:14:22",
     "out_date" : "2015/12/14 19:14:22",
     "amount" : "13.44"
    }
  ]
 },
```

```
"error":{
 "code": 0,
 "msg": "Ok"
}
}
```

FIGURA 135 *MENSAJE DE ÉXITO OBTENER TRANSACCIONES POR COMERCIO*
Fuente: Elaboración propia

Error Output:

- **If success, return 0 error code**
- **If Generic Error, return 999 error code**
- **If invalid input parameter, return 002 error code**
- **If missing a required input parameter, return 001 code**
- **If invalid session, return 200 error code**
- **If invalid date range, return 15 error code**
- **skip – defines where in the list to start loading the next set of records when user scrolls. For example, if skip is set to 10, then load records 11, 12, 13...**
- **limits defines how many records to show (load)**
- **returns the list of transactions that occurred in a merchant between start_date and end_date parameters.**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 001,
                "msg": "Missing input parameter",
        }
}
```

FIGURA 136 *MENSAJE DE FALLO OBTENER TRANSACCIONES POR COMERCIO*
Fuente: Elaboración propia

c.h. GetMerchantsList

Request:

Method: Get

URL: /services/v2/useradmin/getmerchantslist

Input:

- **sessionToken** - String - not null
- **ip** – String – not null

Output:

```
{
"response": {
 "merchants": [
  {
   "id": 1,
   "merchant_name": "Torre Xerox"
  },
  {
   "id": 2,
   "merchant_name": "CC La Lagunita"
  }
 ]
},
"error": {
 "code": 0,
 "msg": "OK"
}
}
```


FIGURA 137 ***EJEMPLO DE MENSAJE DE ÉXITO OBTENER LISTA DE COMERCIOS***
Fuente: Elaboración propia

Error Output:

- **If success, return 0 error code**
- **If Generic Error, return 999 error code**
- **If invalid input parameter, return 002 error code**
- **If missing a required input parameter, return 001 code**
- **If invalid session, return 200 error code**
- **returns the list of all merchants**

Example:


```
{
        "response": {
        }
        "error": {
                "code": 001,
                "msg": "Missing input parameter",
        }
}
```


FIGURA 138 ***EJEMPLO MENSAJE DE FALLO OBTENER TODOS LOS COMERCIOS***
Fuente: Elaboración propia

c.i. AddTicket

Request:

Method: Get
URL: /services/v2/useradmin/addticket

Input:

- **sessionToken** - String - not null
- **tag_id** – String – not null
- **claim_id** – String – not null
- **date** – String – not null

Output:

```
{
 "response":{
  "tickets" : [
   {
    "claimid" : "aaaa",
    "date" : "2015/12/14 19:14:22"
   }
  ]
 },
 "error":{
  "code": 0,
  "msg": "OK"
 }
}
```

FIGURA 139 ***EJEMPLO MENSAJE DE ÉXITO AGREGAR CALCOMANÍA***
Fuente: Elaboración propia

Error Output:

- **If success, return 0 error code**
- **If Generic Error, return 999 error code**
- **If invalid input parameter, return 002 error code**
- **If missing a required input parameter, return 001 code**
- **If invalid session, return 200 error code**
- **Creates a new ticket**
- **Returns the list of unclaimed stickers**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 001,
                "msg": "Missing input parameter",
        }
}
```

FIGURA 140 ***EJEMPLO MENSAJE DE FALLO AGREGAR CALCOMANÍA***

Fuente: Elaboración propia

c.j. UpdateSticker

Request:

Method: Get
URL: /services/v2/useradmin/updatesticker

Input:

- **sessionToken** - String - not null
- **stickerId** – String – not null
- **status** – String – not null

Output:

```
{
 "response":{
  "tickets" : [
   {
    "tagid" : "aa",
    "claimid" : "aaaa",
    "date" : "2015/12/14 19:14:22"
   }
  ]
 },
 "error":{
  "code": 0,
  "msg": "OK"
 }
}
```

FIGURA 141 ***EJEMPLO MENSAJE DE ÉXITO ACTUALIZAR CALCOMANÍA***
Fuente: Elaboración propia

Error Output:

- **If success, return 0 error code**
- **If Generic Error, return 999 error code**
- **If invalid input parameter, return 002 error code**
- **If missing a required input parameter, return 001 code**
- **If invalid session, return 200 error code**
- **Updates a ticket's status**
- **Returns the users that owns the sticker**

Example:

```
{
        "response": {
```

```
                }
                "error": {
                        "code": 001,
                        "msg": "Missing input parameter",
                }
        }
```

FIGURA 142 ***EJEMPLO MENSAJE DE FALLO ACTUALIZAR CALCOMANÍA***
Fuente: Elaboración propia

c.k. AddMerchant

Request:

Method: Get
URL: /services/v2/useradmin/addmerchant

Input:

- **sessionToken** - String - not null
- **ip** – String – not null
- **merchantName** – String – not null
- **alias** – String – nullable
- **documentId** – String – not null
- **address** – String – not null
- **telephone** – String – not null
- **email** – String – nullable
- **contactName** – String – nullable

Output:

```
{
 "response":{
 },
 "error":{
  "code": 0,
  "msg": "OK"
 }
}
```

FIGURA 143 ***EJEMPLO MENSAJE DE ÉXITO AGREGAR COMERCIO***
Fuente: Elaboración propia

Error Output:

- **If success, return 0 error code**
- **If Generic Error, return 999 error code**
- **If invalid input parameter, return 002 error code**
- **If missing a required input parameter, return 001 code**
- **If invalid session, return 200 error code**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 001,
                "msg": "Missing input parameter",
        }
}
```

FIGURA 144 *EJEMPLO MENSAJE DE FALLO AGREGAR COMERCIO*
Fuente: Elaboración propia

c.l. EditMerchant

Request:

Method: Get
URL: /services/v2/useradmin/editmerchant

Input:

- **merchantId** - String – not null
- **merchantName** - String – nullable
- **alias** – String – not null
- **documentId** - String – not null
- **address** - String – not null
- **telephone** - String – not null
- **email** - String – nullable
- **contactDocumentId**– String – not null
- **contactEmail** - String – not null
- **contactName** - String – not null
- **contactType** - String – not null
- **contactMobile** - String – not null
- **ip** – String – not null

Output:

```
{
 "response":{
 },
 "error":{
  "code": 0,
  "msg": "OK"
 }
```

```
}
```

FIGURA 145 *EJEMPLO MENSAJE DE ÉXITO EDITAR COMERCIO*
Fuente: Elaboración propia

Error Output:

- **If success, return 0 error code**
- **If Generic Error, return 999 error code**
- **If invalid input parameter, return 002 error code**
- **If missing a required input parameter, return 001 code**
- **If invalid session, return 200 error code**
- **returns the whole data from a user given his email as a parameter**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 001,
                "msg": "Missing input parameter",
        }
}
```

FIGURA 146 *EJEMPLO MENSAJE DE FALLO EDITAR COMERCIO*
Fuente: Elaboración propia

c.m. UpdateUserStatus

Request:

Method: Get
URL: /services/v2/useradmin/updateuser

Input:

- **sessionToken** - String - not null
- **id** – String – not null
- **status** – String – not null

Output:

```
{
 "response":{
   "users":[
     {
       "name" : "Carlos",
```

```
            "lastName" : "Unda",
            "mobile" : "04142021493",
            "email" : "cunda@citywallet.co",
            "status" : "active"
          }
      ]
    },
    "error":{
      "code": 0,
      "msg": "OK"
    }
  }
```


FIGURA 147 ***EJEMPLO MENSAJE DE ÉXITO ACTUALIZAR ESTADO DE USUARIO***
Fuente: Elaboración propia

Error Output:

- **If success, return 0 error code**
- **If Generic Error, return 999 error code**
- **If invalid input parameter, return 002 error code**
- **If missing a required input parameter, return 001 code**
- **If invalid session, return 200 error code**
- **Updates a user's status**
- **Returns the updated user**

Example:


```
{
        "response": {
        }
        "error": {
                "code": 001,
                "msg": "Missing input parameter",
        }
}
```


FIGURA 148 ***EJEMPLO MENSAJE DE FALLO ACTUALIZAR ESTADO DE USUARIO***
Fuente: Elaboración propia

c.n. GetTransactionsByDateRange

Request:

Method: Get
URL: /services/v2/useradmin/gettransactionsbydaterange

Input:

- **sessionToken** - String - not null

- **startDate** - String - nullable
- **endDate** – String – nullable
- **merchantName** – String – nullable
- **limit** - String - not null
- **skip** - String - not null

Output:


```
{
 "response":{
  "transactions":[
    {
     "uid" : 1,
     "pin" : "Cc22",
     "name" : "Carlos",
     "last_name" : "Unda",
     "merchant_id" : 1,
     "merchant_name" : "Xerox",
     "in_date" : "2015/12/14",
     "out_date" : "2015/12/14",
     "in_hour" : "18:14:22",
     "out_hour" : "19:14:22",
     "total_amount" : "14.784",
     "merchant_amount" : "13.44",
     "cw_amount" : "1.344"
    },
    {
     "uid" : 1,
     "pin" : "Cc22",
     "name" : "Carlos",
     "last_name" : "Unda",
     "merchant_id" : 1,
     "merchant_name" : "Xerox",
     "in_date" : "2015/12/15 18:14:22",
     "out_date" : "2015/12/15 19:14:22",
     "in_hour" : "18:14:22",
     "out_hour" : "19:14:22",
     "total_amount" : "14.784",
     "merchant_amount" : "13.44",
     "cw_amount" : "1.344"
    }
  ]
 },
 "error":{
  "code": 0,
  "msg": "OK"
 }
}
```


FIGURA 149 ***EJEMPLO MENSAJE DE ÉXITO TRANSACCIONES EN RANGO DE FECHA***
Fuente: Elaboración propia

Error Output:

- If success, return 0 error code
- If Generic Error, return 999 error code
- If invalid input parameter, return 002 error code
- If missing a required input parameter, return 001 code
- If invalid session, return 200 error code
- Date parameter format: dd/mm/yyyy
- skip – defines where in the list to start loading the next set of records when user scrolls. For example, if skip is set to 10, then load records 11, 12, 13...
- limits defines how many records to show (load)
- returns the list of transactions done in the date range used as parameter.

Example:

```
{
        "response": {
        }
        "error": {
                "code": 999,
                "msg": "Internal Error",
        }
}
```

FIGURA 150 ***EJEMPLO MENSAJE DE FALLO OBTENER TRANSACCIONES EN RANGO DE FECHA***
Fuente: Elaboración propia

c.o. GetCWMerchantPayments

Request:

Method: Get
URL: /services/v2/useradmin/getcwmerchantpayments

Input:

- **sessionToken** - String - not null

Output:

```
{
 "response":{
  "payments":[
   {
    "id":1,
    "date": "2016-10-23",
```

```
        "amount": 500,
        "log": "pago",
        "merchant":"Torre Xerox"
      },
      {
        "id":2,
        "date": "2016-10-23",
        "amount": 500,
        "log": "pago ",
        "merchant":"Torre Xerox"
      }
    ]
  },
  "error":{
    "code": 0,
    "msg": "OK"
  }
}
```


FIGURA 151 ***EJEMPLO MENSAJE DE ÉXITO PAGOS DE CITYWALLET A COMERCIOS***
Fuente: Elaboración propia

Error Output:

- **If success, return 0 error code**
- **If Generic Error, return 999 error code**
- **If invalid input parameter, return 002 error code**
- **If missing a required input parameter, return 001 code**
- **If invalid session, return 200 error code**

Example:


```
{
        "response": {
        }
        "error": {
                "code": 999,
                "msg": "Internal Error",
        }
}
```


FIGURA 152 ***EJEMPLO MENSAJE DE FALLO OBTENER LOS PAGOS DE CITYWALLET A COMERCIOS***
Fuente: Elaboración propia

c.p. GetUsersPayments

Request:

Method: Get

URL: /services/v2/useradmin/getusersrpayments

Input:

- **sessionToken** - String - not null

Output:


```
{
 "response":{
"payments":[
    {
     "id":1,
     "name": "Santiago",
     "lastname": "Bernal",
     "email": "lq@lql",
     "mobile":"0412556686",
     "date":"2016-03-20",
     "amount":"500"
    },
    {
     "id":1,
     "name": "Santiago",
     "lastname": "Bernal",
     "email": "lq@lql",
     "mobile":"0412556686",
     "date":"2016-03-20",
     "amount":"500"
    }
   ] },
  "error":{
   "code": 0,
   "msg": "OK"
  }
 }
```


FIGURA 153 ***EJEMPLO MENSAJE DE ÉXITO OBTENER PAGOS DE USUARIOS***
Fuente: Elaboración propia

Error Output:

- **If success, return 0 error code**
- **If Generic Error, return 999 error code**
- **If invalid input parameter, return 002 error code**
- **If missing a required input parameter, return 001 code**
- **If invalid session, return 200 error code**

Example:


```
{
        "response": {
        }
        "error": {
                "code": 999,
```

```
                    "msg": "Internal Error",
                }
            }
```

FIGURA 154 *EJEMPLO MENSAJE DE FALLO OBTENER PAGOS DE USUARIOS*
Fuente: Elaboración propia

d. Módulo Admin Comercio

d.a. GetParkingMerchant

Request:

Method: Get
URL: /services/v1/merchant/getparkingmerchant
Authorization: sessionToken:"token"

Input:

- **None**

Output:

```
{
        "response": {
    "id": 1
        "name": "Xerox",
    "alias": "Xerox",
    "merchant_rif": "J-5213587",
        "merchant_icon": "serialized"
                "total_consumption": 132.25,
                "total_hours": 50.30 horas,
                "last_paid_date": "dd/MM/yyyy",
    "merchant_fraction_rate": 5.60
    "merchant_hour_rate": 11.20,
    "contacts":[
        {
        "name": "maria",
        "mobile": "04144444444"
        "email": "test@gmail.com"
        "documentId": "19555555",
        "type": "admin"
        }
    ]
        }
        "error": {
            "code": 0,
            "msg": "OK",
        }
    }
```

FIGURA 155 ***EJEMPLO MENSAJE DE ÉXITO OBTENER INFORMACIÓN DE COMERCIO***
Fuente: Elaboración propia

Error Output:

- If success, return 0 code
- If Generic error, return 999 code
- If empty session, return 200 error code
- If invalid user role, return 19 error code
- If invalid session, return 19 error code.

Example:

```
{
        "response": {
        }
        "error": {
                "code": 1,
                "msg": "Missing input parameter",
        }
    }
```

FIGURA 156 ***EJEMPLO MENSAJE DE FALLO OBTENER INFORMACIÓN DE COMERCIO***
Fuente: Elaboración propia

d.b. GetTransactions

Request:

Method: GET
URL: /services/v1/merchant/gettransactions
Authorization: sessionToken:"token"

Input:

- **startDate** – String – nullable – initial date - formato: yyyy-MM-dd
- **finishDate** -String- nullable -end date - formato: yyyy-MM-dd

Output:

```
{
        "response": {
                "transactions":[
                        {
                                "id":"1",
    "in_id": 102
    "out_id": 105
                                "in_date":"dd/MM",
                                "out_date":"dd/MM",
    "in_hour": "HH:mm",
    "out_hour": "HH:mm",
                                "cost":"200.00"
    "total_hour": "5"
                        }
                        {
```

```
                                        "id":"2",
            "in_id": 102
            "out_id": 105
                                        "in_date":"dd/MM",
                                        "out_date":"dd/MM",
            "in_hour": "HH:mm",
            "out_hour": "HH:mm",
                                        "cost":"200.00"
            "total_hour": "5"
                                        }
        }


        "error": {
                "code": 0,
                "msg": "OK",
        }
    }
```


FIGURA 157 ***EJEMPLO MENSAJE DE ÉXITO OBTENER TRANSACCIONES***
Fuente: Elaboración propia

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**
- **If empty session, return 200 error code**
- **If invalid user role, return 19 error code**
- **If invalid session, return 19 error code**
- **If invalid input parameter, return 2 error code.**

Example:


```
{
        "response": {
        }
        "error": {
                "code": 1,
                "msg": "Missing input parameter",
        }
    }
```


FIGURA 158 ***EJEMPLO MENSAJE DE FALLO OBTENER TRANSACCIONES***
Fuente: Elaboración propia

d.c. GetSummary

Request:

Method: GET
URL: /services/v1/merchant/getsummary
Authorization: sessionToken:"token"

Input:

- none

Output:


```
{
        "response": {
        "merchant_name": "Xerox",
        "date": "Tuesday 25/08/1990",
        "merchant_rif": "J-0000000000",
        "in_count": 250,
        "out_count": 35,
        "transit": 265,
        "earnings": 5500
        "summary": [
            {
                "id":1,
                                "in_date": "dd/MM",
                                "out_date": "dd/MM",
                "in_hour": "HH:mm",
                "out_hour": "HH:mm",
                                "cost": "200.00",
                "total_hour": "5.00"
                            },
                            {
                                "id":"2",
                                "in_date":"dd/MM",
                        "out_date":"dd/MM",
                "in_hour": "HH:mm",
                "out_hour": "HH:mm",
                                "cost":"200.00"
                "total_hour": "5.20"
                            }
        ],
        "x":"06:00,07:00,08:00,09:00,10:00,11:00",
        "y1":"5,8,9,50,6,8",
        "y2":"6,8,9,12,3,1"
          }
          "error": {
                "code": 0,
                "msg": "OK",
          }
    }
```


FIGURA 159 ***EJEMPLO MENSAJE DE ÉXITO OBTENER RESUMEN DE COMERCIO***

Fuente: Elaboración propia

Error Output:

- **If success, return 0 code.**
- **If Generic error, return 999 code**
- **If empty session, return 200 error code**
- **If invalid user role, return 19 error code**
- **If invalid session, return 19 error code**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 1,
                "msg": "Missing input parameter",
        }
    }
```

FIGURA 160 ***EJEMPLO MENSAJE DE FALLO OBTENER RESUMEN DE COMERCIO***
Fuente: Elaboración propia

d.d. Login

Request:

Method: POST
URL: /services/v1/merchant/login

Input:

- **uid** – string – notnull – user email
- **pwd** - string – not null – password
- **ip** - string – not null – ip

Output:

```
{
        "response": {
        "sessionToken": "token"
            }
        "error": {
                "code": 0,
                "msg": "OK",
        }
    }
```

FIGURA 161 ***EJEMPLO MENSAJE DE ÉXITO INICIO DE SESIÓN DE COMERCIO***
Fuente: Elaboración propia

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**
- **If Invalid user, return 7 code**
- **If Invalid account status, return 9 code**
- **If invalid user role, return 19 error code**

Example:

```
    {
        "response": {
        }
        "error": {
                "code": 1,
```

```
                "msg": "Missing input parameter",
            }
    }
```

FIGURA 162 ***EJEMPLO MENSAJE DE FALLO INICIO SESIÓN COMERCIO***
Fuente: Elaboración propia

d.e. Logout

Request:

Method: GET
URL: /services/v1/merchant/logout
Authorization: sessionToken: "token"

Input:

None

Output:

```
    {
        "response": { }
        "error": {
            "code": 0,
            "msg": "OK",
        }
    }
```

FIGURA 163 ***EJEMPLO MENSAJE DE ÉXITO CERRAR SESIÓN COMERCIO***
Fuente: Elaboración propia

Error Output:

- If success, return 0 code
- If Generic error, return 999 code
- If Invalid user, return 7 code
- If Invalid account status, return 9 code
- If invalid user role, return 19 error code

Example:

```
    {
        "response": {
        }
        "error": {
            "code": 1,
            "msg": "Missing input parameter",
        }
    }
```

FIGURA 164 ***EJEMPLO MENSAJE DE FALLO CERRAR SESIÓN COMERCIO***
Fuente: Elaboración propia

d.f. ForgotPassword

Request:

Method: Get
URL: /services/v1/merchant/forgotpassword
Authorization: none
Header: ip – String – not null – requesting user IP address

Input:

- **uid** String – not null – users id, email

Output:

```
{
  "response": {
  }
  "error": {
        "code": 0,
        "msg": "OK",
  }
}
```

FIGURA 165 ***EJEMPLO MENSAJE DE ÉXITO RECUPERACIÓN DE CLAVE***
Fuente: Elaboración propia

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**
- **If Missing required input parameters, return 1 code**
- **If Invalid input parameter, return 2 code**
- **If Invalid user, return 7 code**
- **If non existing user, return 13 code**

Example:

```
{
  "response": {
  }
  "error": {
        "code": 1,
        "msg": "Missing input parameter",
  }
}
```

FIGURA 166 ***EJEMPLO MENSAJE DE FALLO RECUPERACIÓN DE CLAVE***
Fuente: Elaboración propia

d.g. GetVoucher

Request:

Method: GET
URL: /services/v1/merchant/getvoucher

Authorization: sessionToken:"token"

Input:

- **visitId** – String – not null

Output:


```
{
        "response": {
                                "id":"1",
                "in_id": 102
                "out_id": 105
                                "merchantIcon": "serialized",
                                "merchantName": "Merchant",
            "pin":"AAAA",
                                "in_date":"dd/MM/YYY",
                                "out_date":"dd/MM/YYY",
                "in_hour": "HH:mm",
                "out_hour": "HH:mm",
                                "cost":"200.00"
        }
        "error": {
            "code": 0,
            "msg": "OK",
        }
    }
```


FIGURA 167 ***EJEMPLO MENSAJE DE EXITO OBTENER VOUCHER DE VISITA***

Fuente: Elaboración propia

Error Output:

- **If success, return 0 code**
- **If Generic error, return 999 code**
- **If empty session, return 200 error code**
- **If missing input parameter, return 1 code**
- **If invalid user role, return 19 error code**
- **If invalid session, return 19 error code**

Example:


```
{
        "response": {
        }
        "error": {
            "code": 1,
            "msg": "Missing input parameter",
        }
    }
```


FIGURA 168 ***EJEMPLO MENSAJE DE FALLO OBTENER VOUCHER DE VISITA***

Fuente: Elaboración propia

d.h. GetCWMerchantPayments

Request:

Method: Get

URL: /services/v2/merchant/getcwmerchantpayments

Input:

- **sessionToken** - String - not null

Output:


```
{
 "response":{
  "payments":[
   {
    "id":1,
    "date": "2016-10-23",
    "amount": 500,
    "log": "pago",
    "merchant":"Torre Xerox"
   },
   {
    "id":2,
    "date": "2016-10-23",
    "amount": 500,
    "log": "pago ",
    "merchant":"Torre Xerox"
   }
  ]
 },
 "error":{
  "code": 0,
  "msg": "OK"
 }
}
```


FIGURA 169 ***EJEMPLO MENSAJE DE ÉXITO PAGOS DE CITYWALLET A COMERCIOS***

Fuente: Elaboración propia

Error Output:

- **If success, return 0 error code**
- **If Generic Error, return 999 error code**
- **If invalid input parameter, return 002 error code**
- **If missing a required input parameter, return 001 code**
- **If invalid session, return 200 error code**

Example:


```
{
        "response": {
        }
        "error": {
```

```
                    "code": 999,
                    "msg": "Internal Error",
                }
            }
```

FIGURA 170 *EJEMPLO MENSAJE DE FALLO OBTENER LOS PAGOS DE CITYWALLET A COMERCIOS*
Fuente: Elaboración propia

e. Ripple

e.a. Asociar cuenta

Request:

Method: POST
URL: /ripple/account
Header: authToken = "token"

Input:

- **id** - String - not null – client's id

Output:

```
{
 "response":{
      "id": 5
 },
 "error":{
  "code": 0,
  "msg": "OK"
 }
}
```

FIGURA 171 *EJEMPLO MENSAJE DE ÉXITO ASOCIACIÓN DE CUENTA CITYWALLET CON RIPPLE*
Fuente: Elaboración propia

Error Output:

- **If success, return 0 error code**
- **If Generic Error, return 999 error code**
- **If invalid input parameter, return 002 error code**
- **If missing a required input parameter, return 001 code**
- **If invalid session, return 200 error code**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 999,
```

```
                    "msg": "Internal Error",
                }
            }
```

FIGURA 172 *EJEMPLO MENSAJE DE FALLO ASOCIACION DE CUENTA CITYWALLET CON RIPPLE*
Fuente: Elaboración propia

e.b. CashIn

Request:

Method: POST
URL: /ripple/transaction/{accountId}/cashin
Header: authToken = "token"

Input:

- **accountId** – String – not null
- **amount** – String – not null
- **reference** – String – not null

Output:

```
{
 "response":{
 },
 "error":{
  "code": 0,
  "msg": "OK"
 }
}
```

FIGURA 173 *EJEMPLO MENSAJE DE INGRESO DE DINERO CON RIPPLE*
Fuente: Elaboración propia

Error Output:

- **If success, return 0 error code**
- **If Generic Error, return 999 error code**
- **If invalid input parameter, return 002 error code**
- **If missing a required input parameter, return 001 code**
- **If invalid session, return 200 error code**

Example:

```
{
        "response": {
        }
        "error": {
                "code": 999,
                "msg": "Internal Error",
                "internalError": {
```

```
                    }
            }
    }
```

FIGURA 174 *EJEMPLO MENSAJE DE FALLO DE INGRESO DE DINERO EN RIPPLE*
Fuente: Elaboración propia

E. Diagramas de Secuencia

a. AddSticker

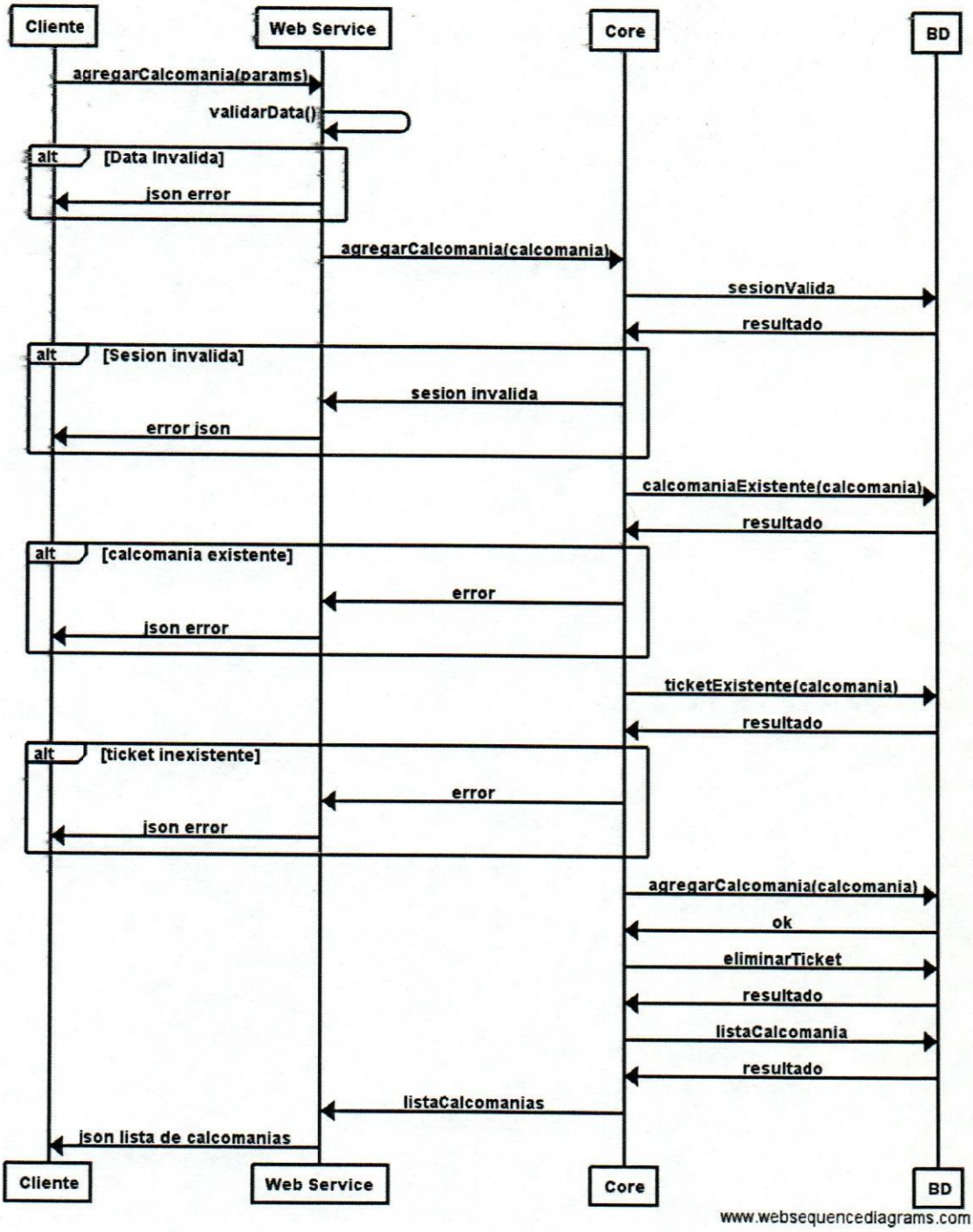

FIGURA 175 *DIAGRAMA DE SECUENCIA AGREGAR CALCOMANÍA*
Fuente: Elaboración propia

b. CreateAccount

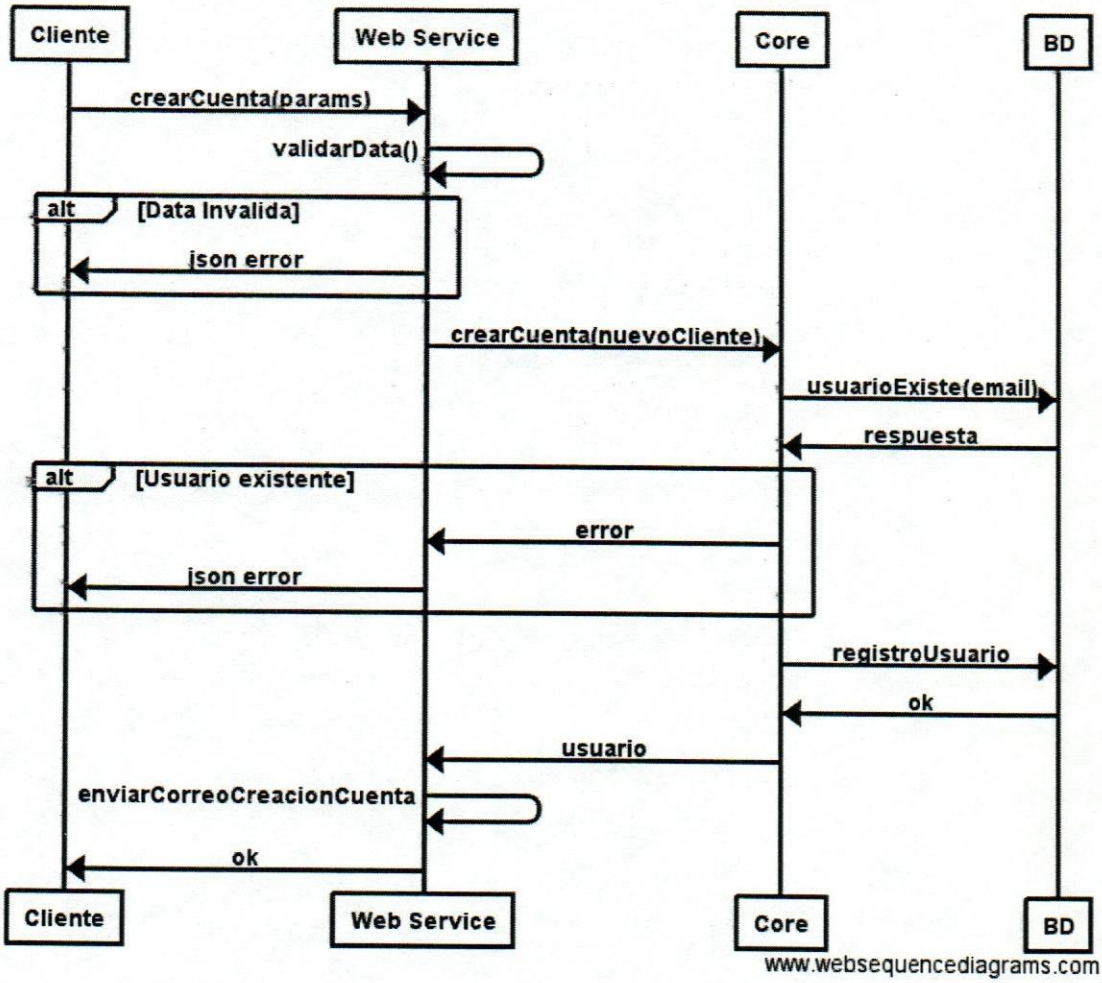

FIGURA 176 *DIAGRAMA DE SECUENCIA CREAR CUENTA*
Fuente: Elaboración propia

c. CreateCashInNotification

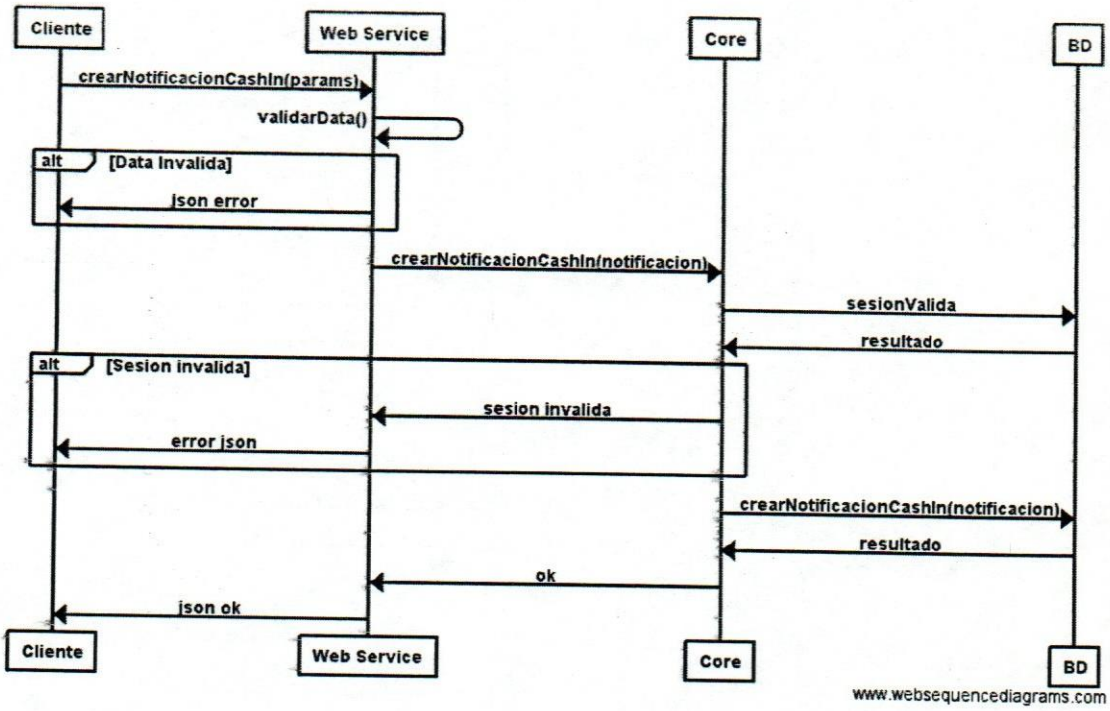

FIGURA 177 ***DIAGRAMA DE SECUENCIA CREAR NOTIFICACION DE CASHINS***
Fuente: Elaboración propia

d. CreatePaymentInstapago

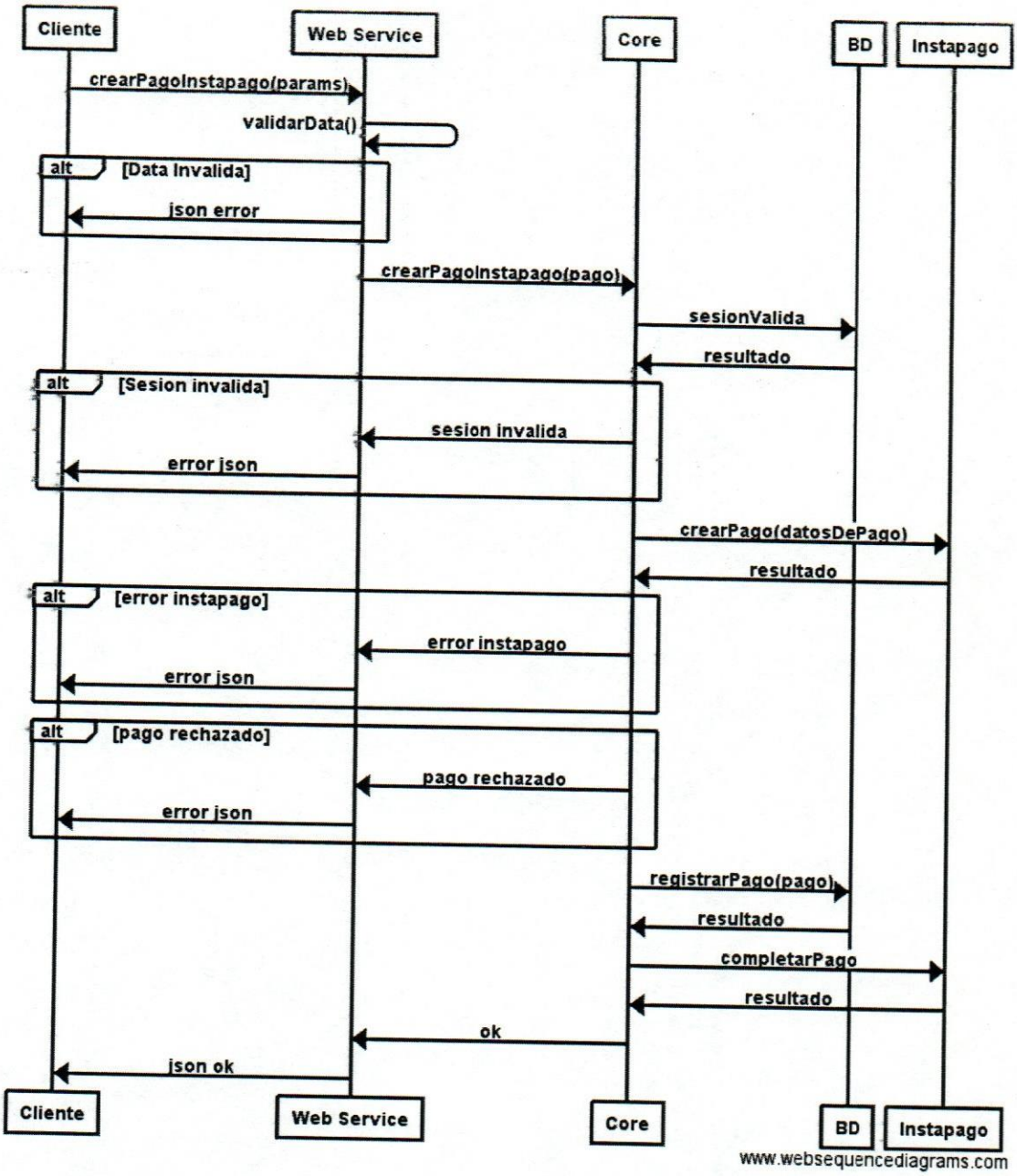

FIGURA 178 *DIAGRAMA DE SECUENCIA CREAR PAGO INSTAPAGO*
Fuente: Elaboración propia

e. ForgotPassword

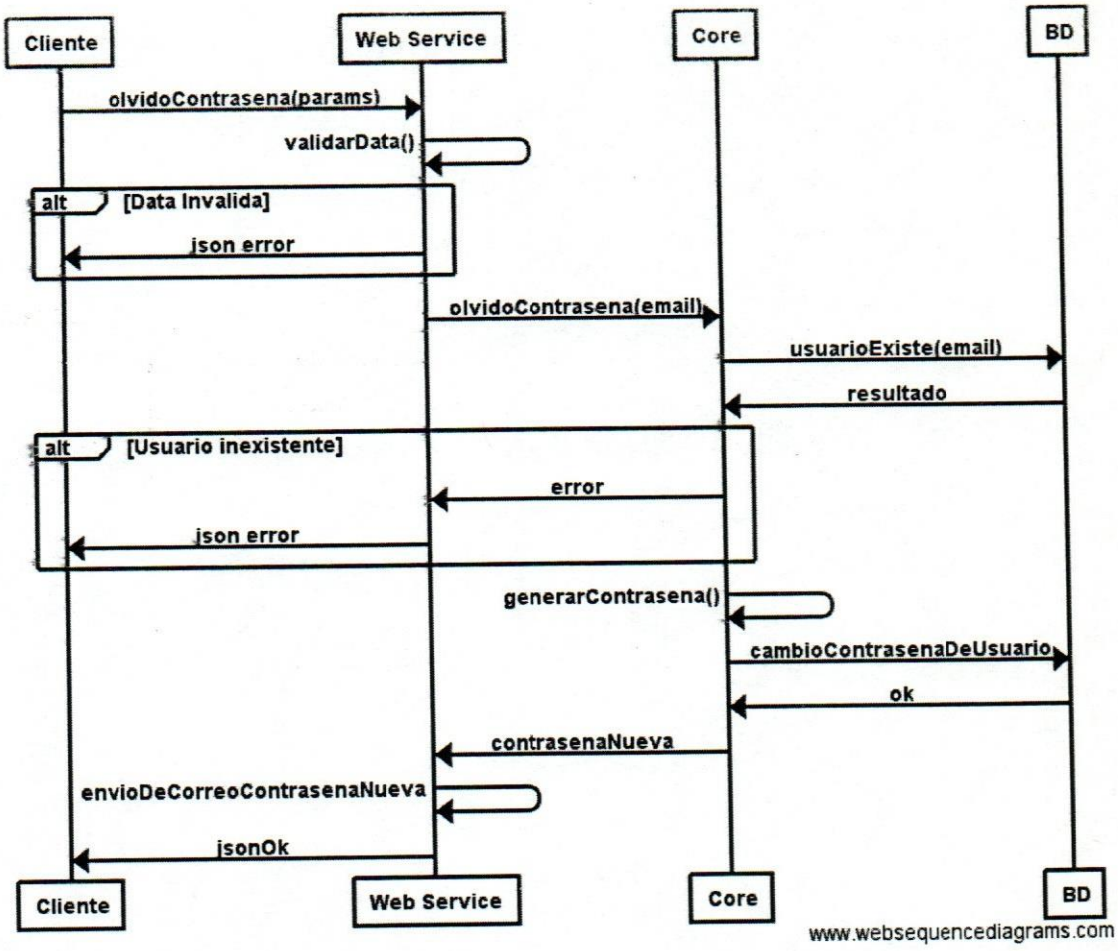

FIGURA 179 ***DIAGRAMA DE SECUENCIA OLVIDO DE CONTRASEÑA***
Fuente: Elaboración propia

f. GetCashIns

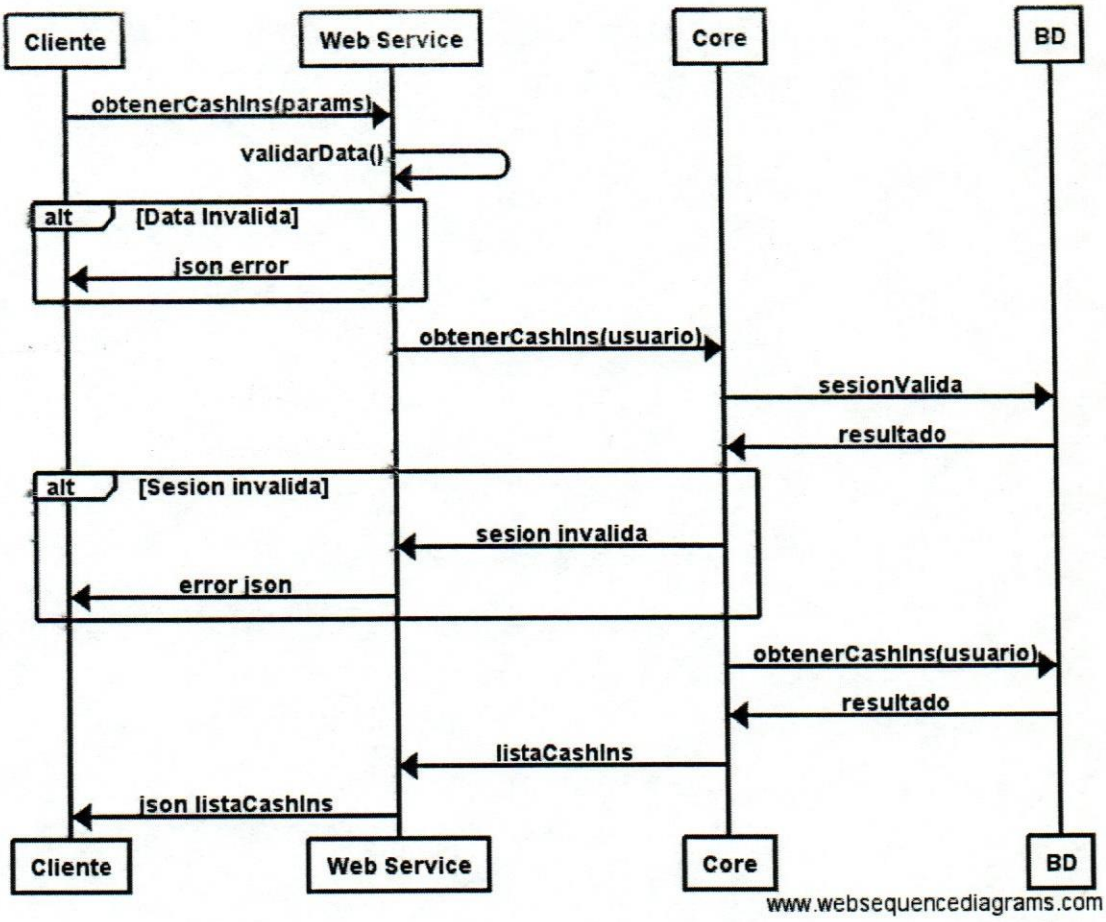

FIGURA 180 ***DIAGRAMA DE SECUENCIA OBTENER CASHINS***
Fuente: Elaboración propia

g. GetPaymentSliderInfo

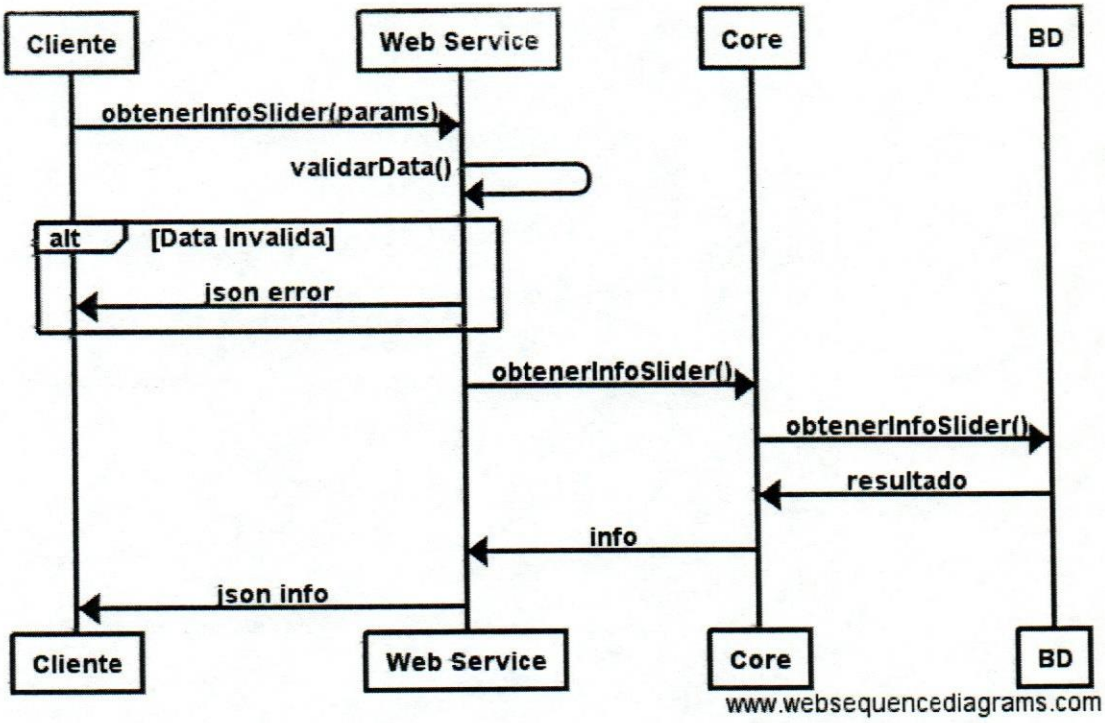

FIGURA 181 ***DIAGRAMA DE SECUENCIA OBTENER INFO DE SLIDER DE PAGO***
Fuente: Elaboración propia

h. GetStickers

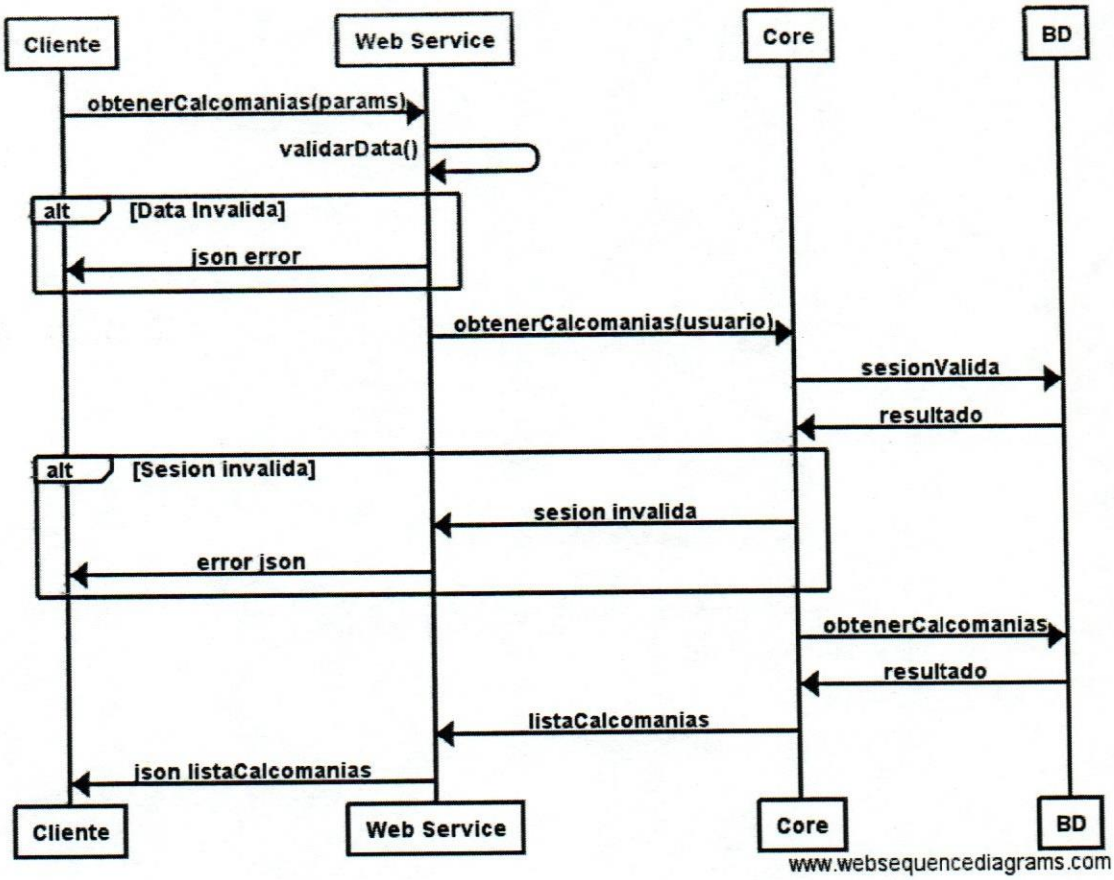

FIGURA 182 ***DIAGRAMA DE SECUENCIA OBTENER CALCOMANÍAS***
Fuente: Elaboración propia

i. GetVisit

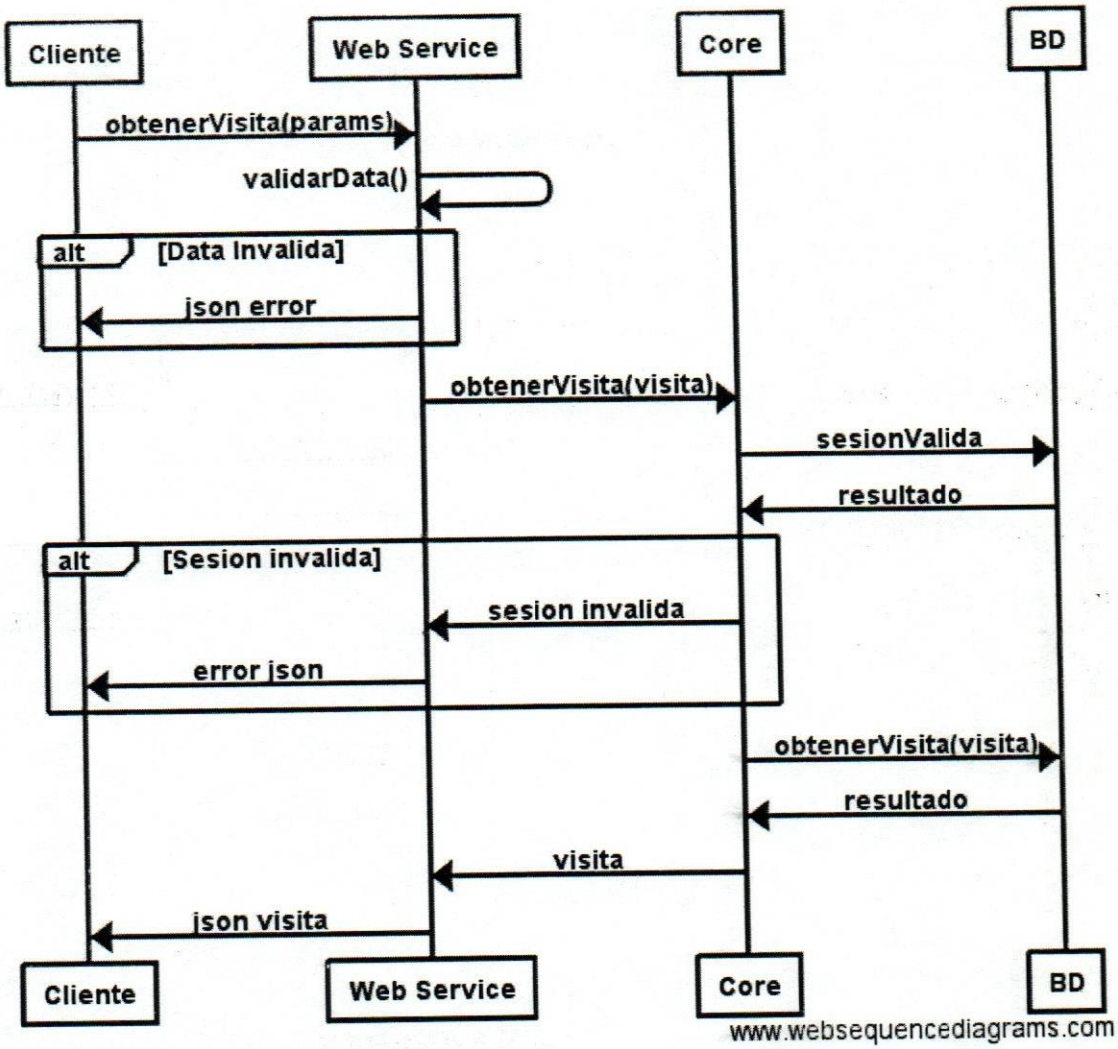

FIGURA 183 ***DIAGRAMA DE SECUENCIA OBTENER VISITA***
Fuente: Elaboración propia

j. GetVisits

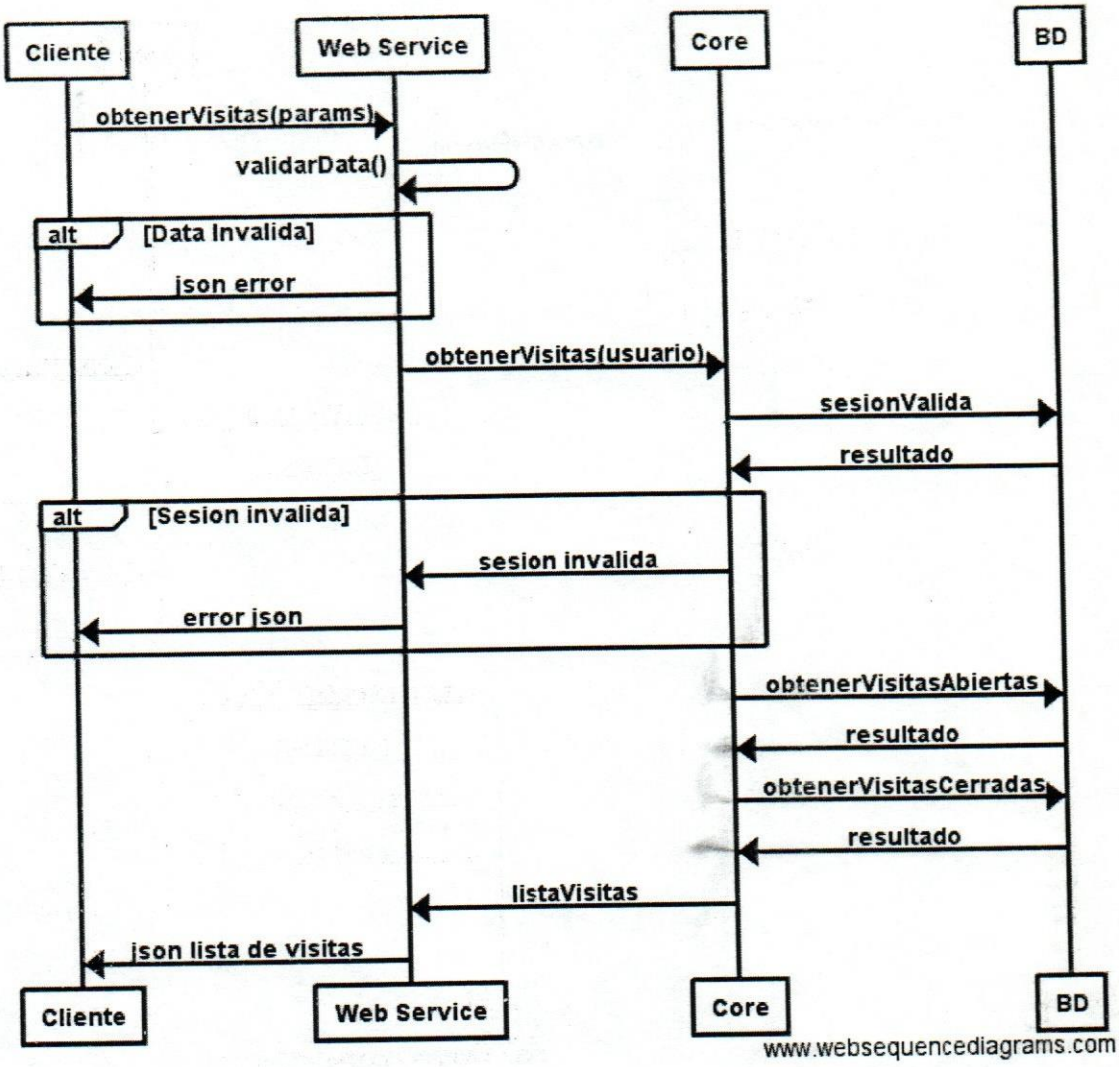

FIGURA 184 ***DIAGRAMA DE SECUENCIA OBTENER VISITAS***
Fuente: Elaboración propia

k. Login

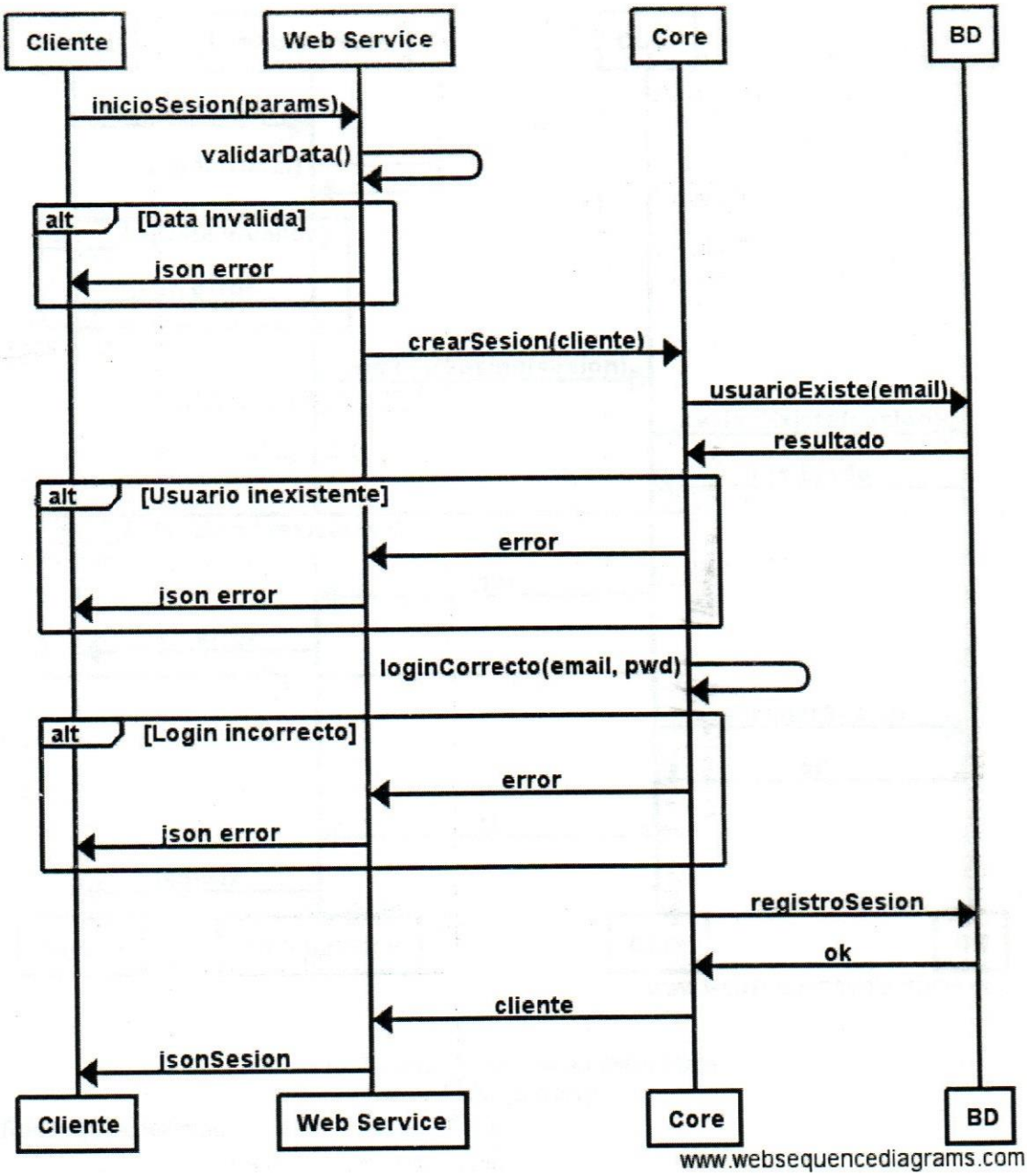

FIGURA 185 ***DIAGRAMA DE SECUENCIA INICIO DE SESIÓN***
Fuente: Elaboración propia

I. Logout

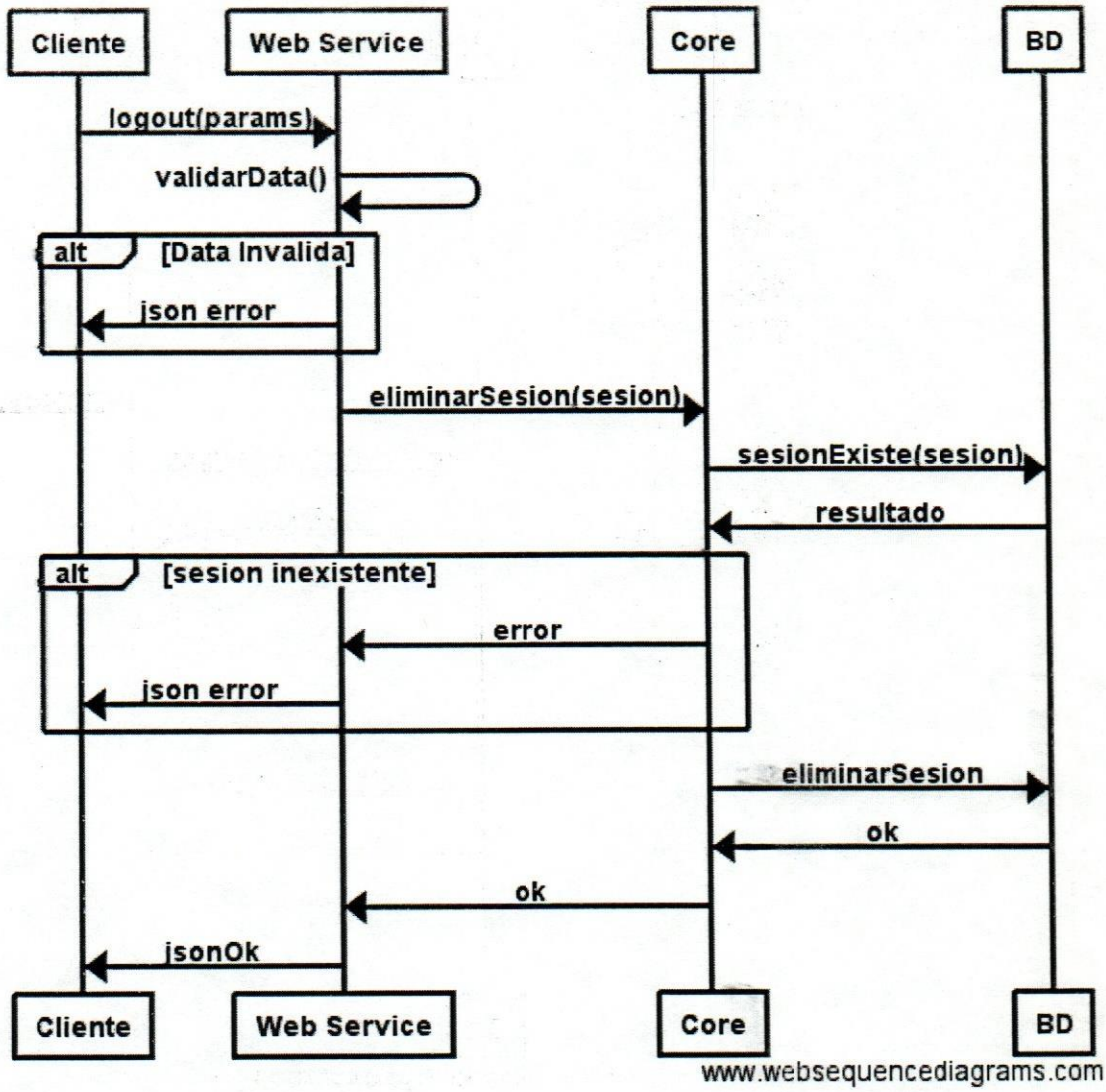

FIGURA 186 ***DIAGRAMA DE SECUENCIA CERRAR SESIÓN***
Fuente: Elaboración propia

m. ResendCreateEmail

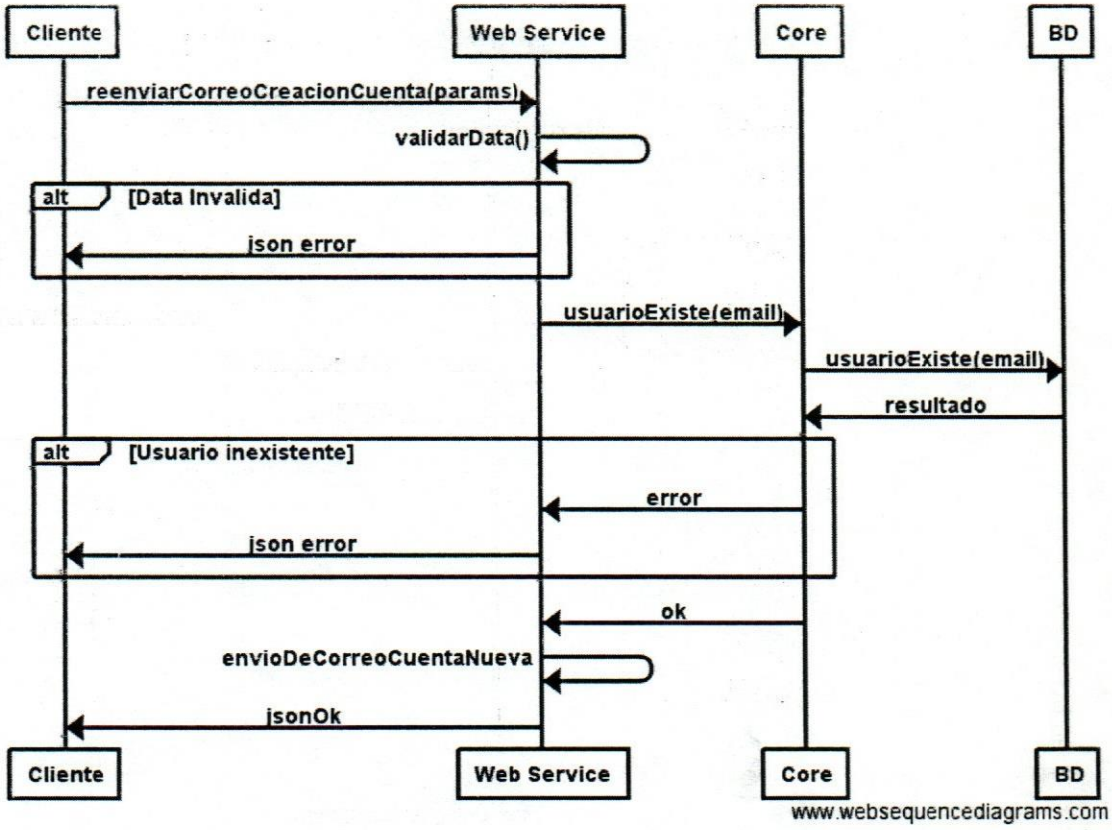

FIGURA 187 ***DIAGRAMA DE SECUENCIA REENVIAR CORREO DE CREACIÓN DE CUENTA***
Fuente: Elaboración propia

n. SetPassword

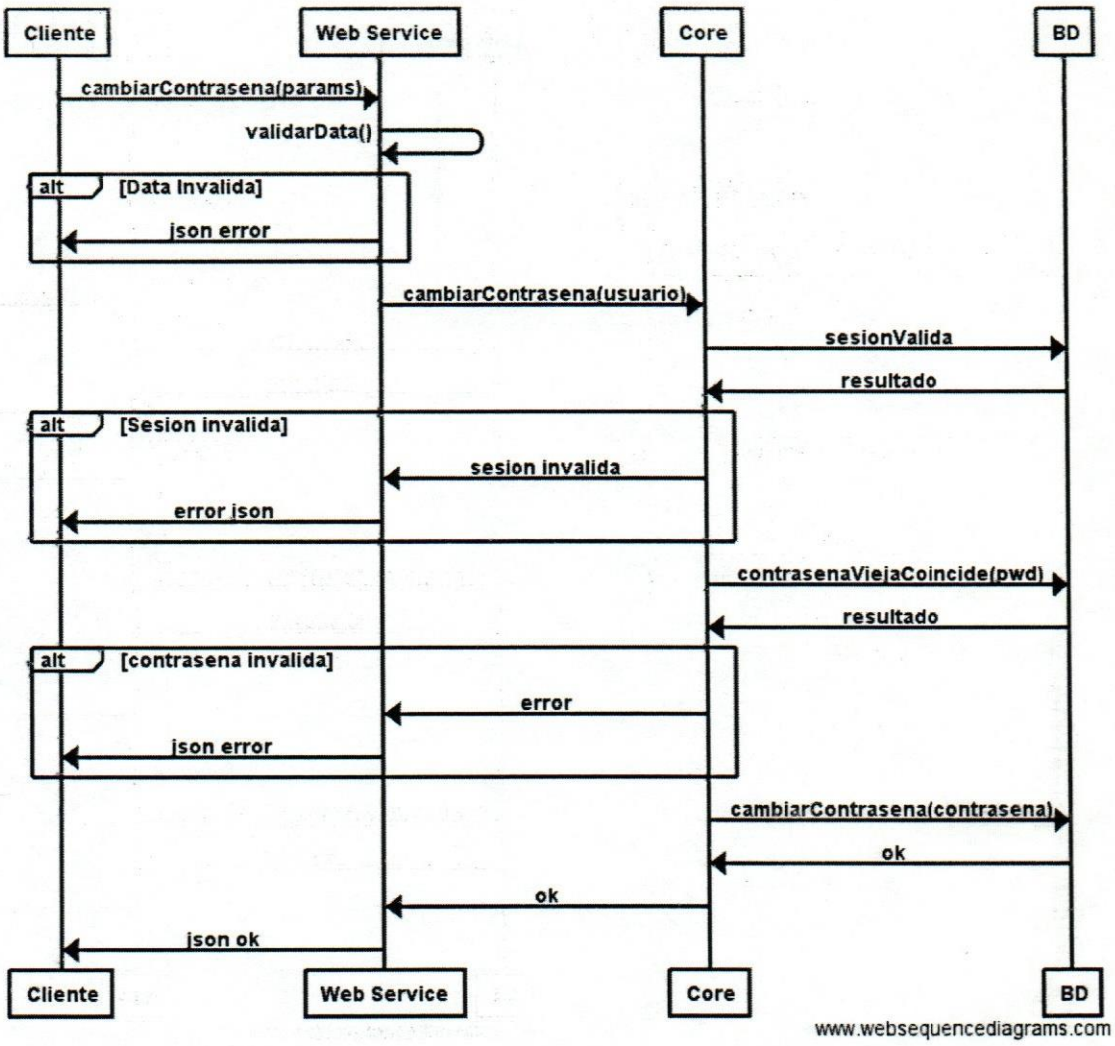

FIGURA 188 ***DIAGRAMA DE SECUENCIA CAMBIAR CONTRASEÑA***
Fuente: Elaboración propia

o. UpdateAccount

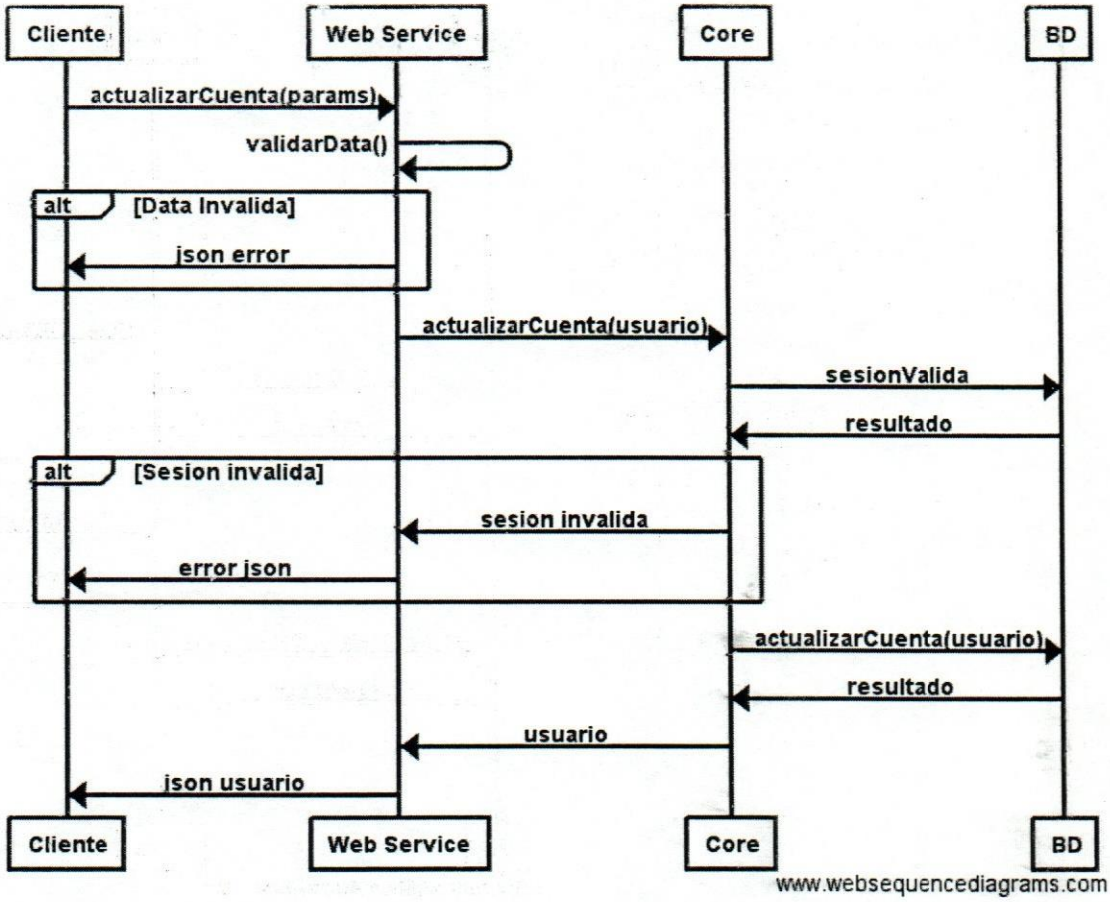

FIGURA 189 ***DIAGRAMA DE SECUENCIA ACTUALIZAR CUENTA***
Fuente: Elaboración propia

p. ValidateCreateAccountToken

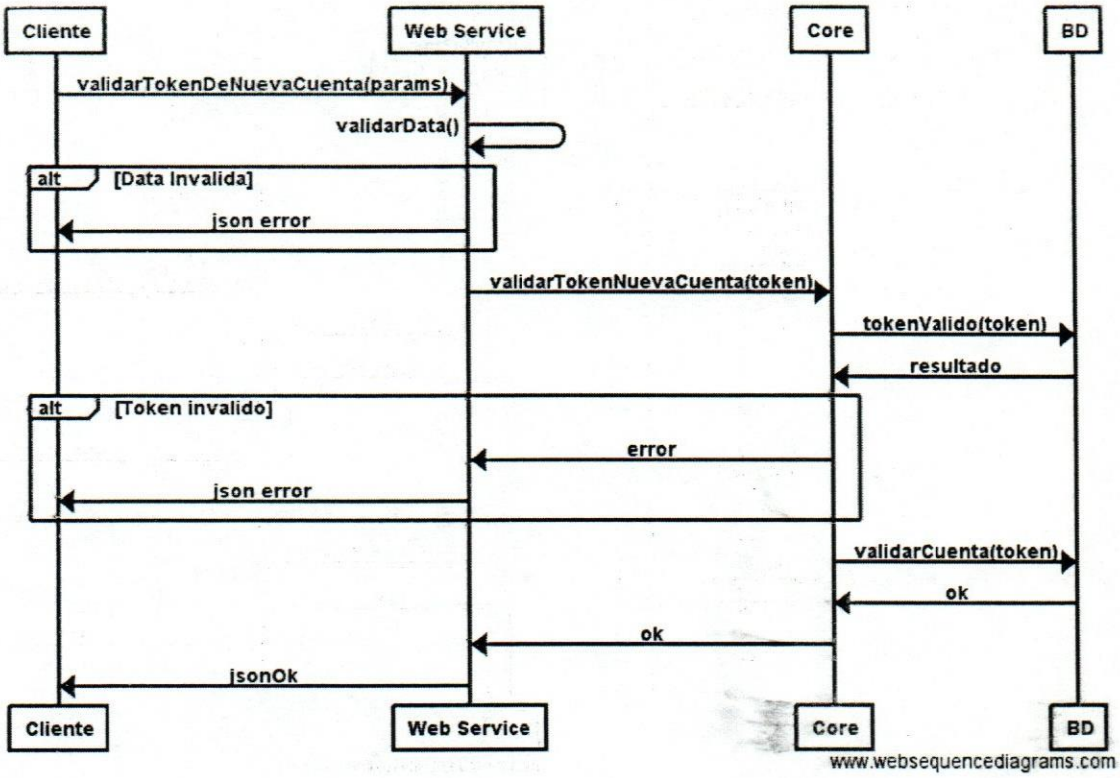

FIGURA 190 ***DIAGRAMA DE SECUENCIA VALIDAR TOKEN DE NUEVA CUENTA***
Fuente: Elaboración propia

q. GetActiveBanks

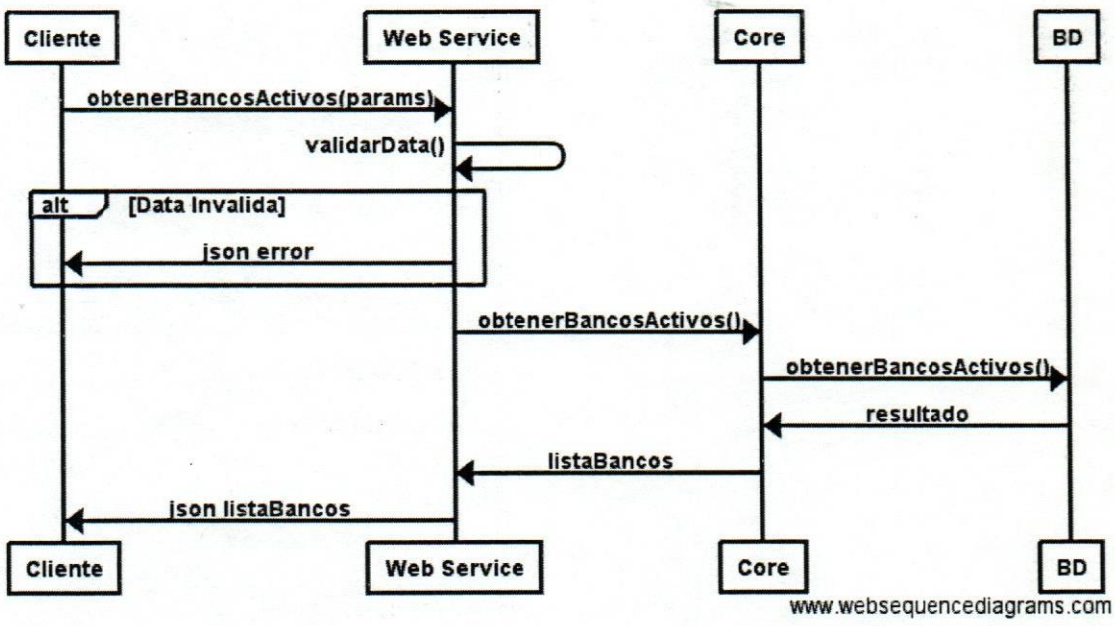

FIGURA 191 ***DIAGRAMA DE SECUENCIA OBTENER BANCOS ACTIVOS***

Fuente: Elaboración propia

r. Registro Transacciones en POS

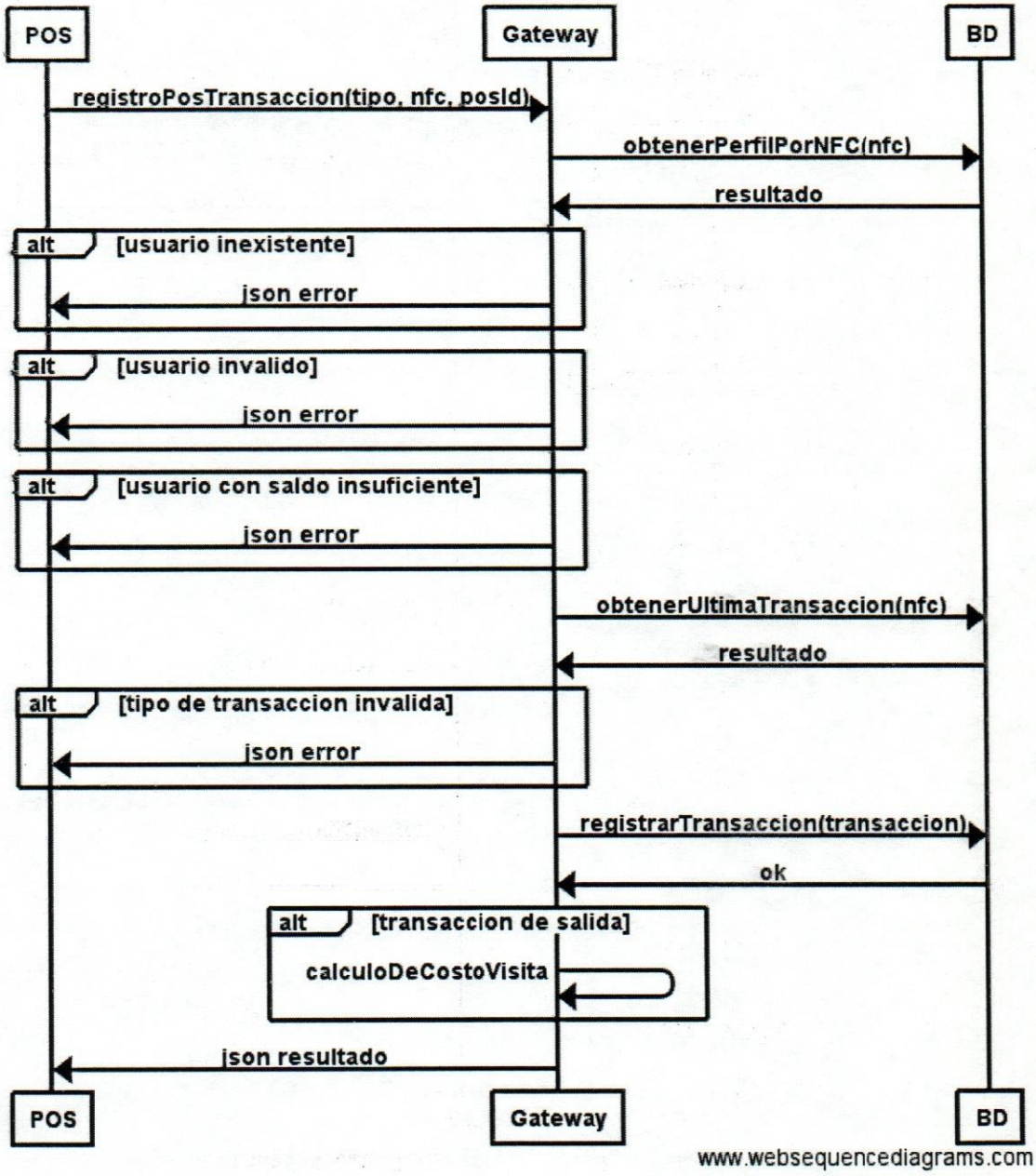

FIGURA 192 ***DIAGRAMA DE SECUENCIA REGISTRO DE TRANSACCIONES EN POS***
Fuente: Elaboración propia

s. Sincronización de Transacciones POS – Central Gateway

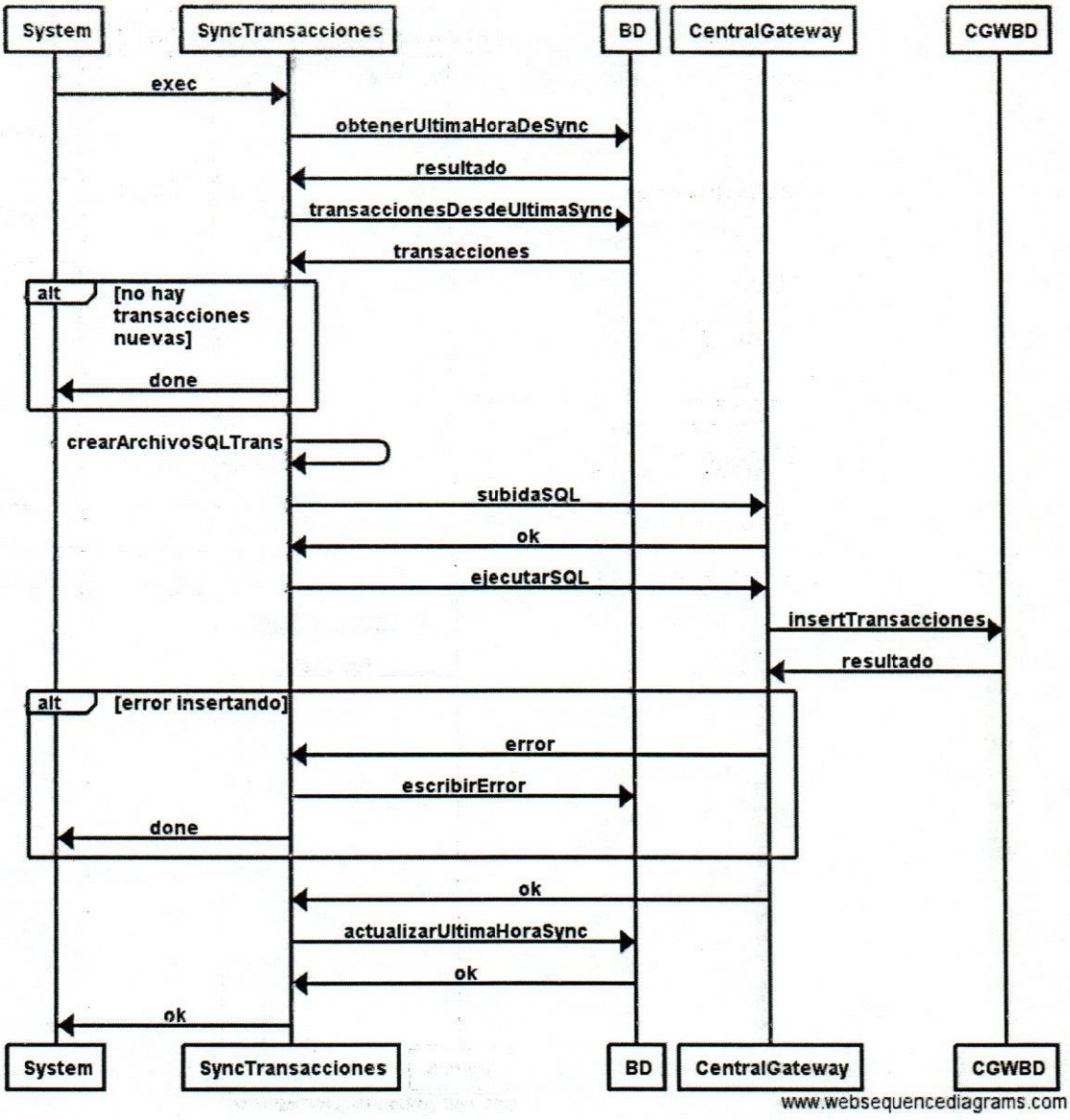

FIGURA 193 ***DIAGRAMA DE SECUENCIA SINCRONIZACIÓN DE TRANSACCIONES POS A CENTRAL GATEWAY***
Fuente: Elaboración propia

t. Sincronización de Transacciones Central Gateway – CityWallet

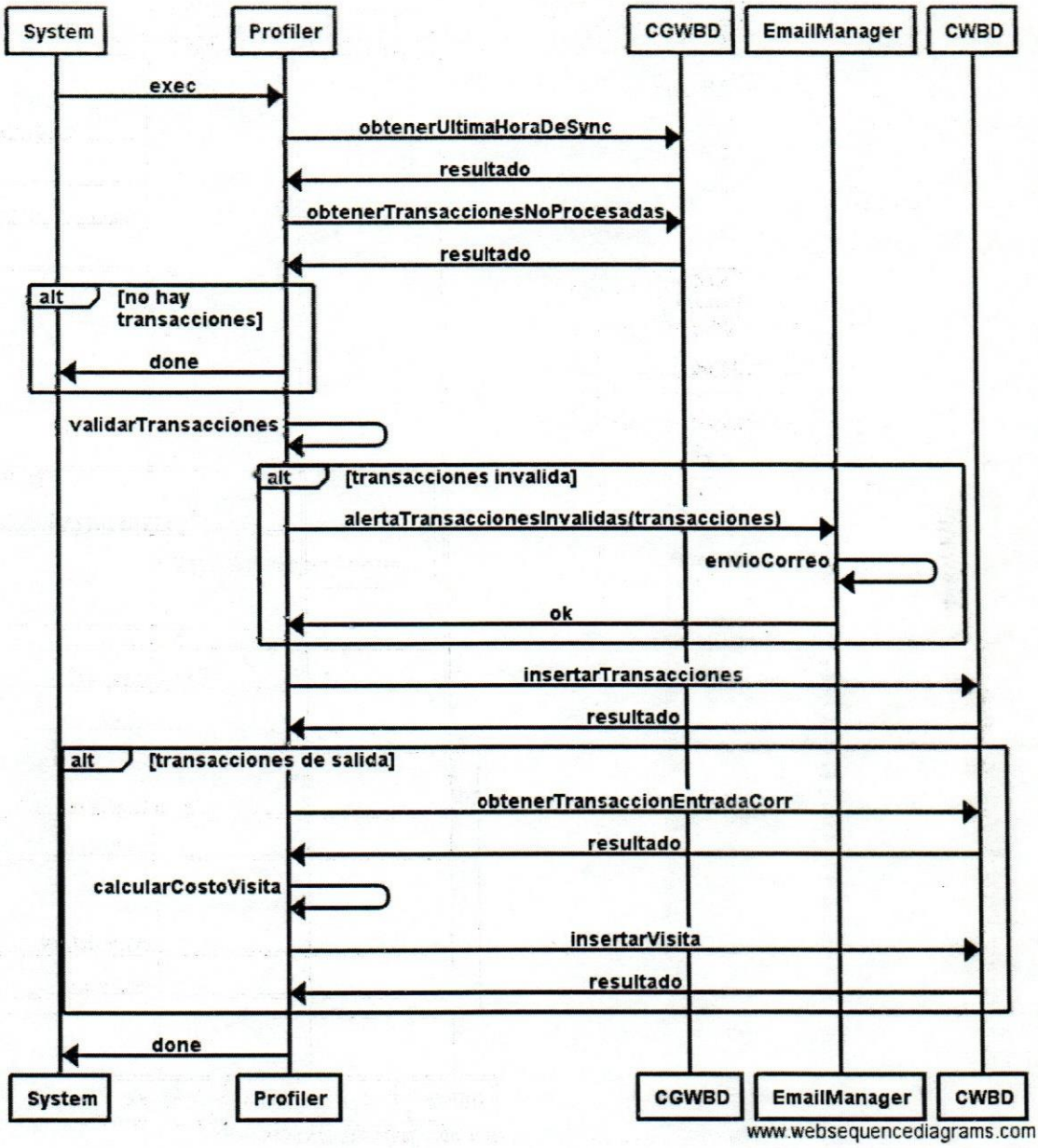

FIGURA 194 *DIAGRAMA DE SECUENCIA SINCRONIZACIÓN DE TRANSACCIONES CENTRAL GATEWAY CON CITYWALLET*
Fuente: Elaboración propia

u. Sincronización de perfiles CityWallet – Central Gateway

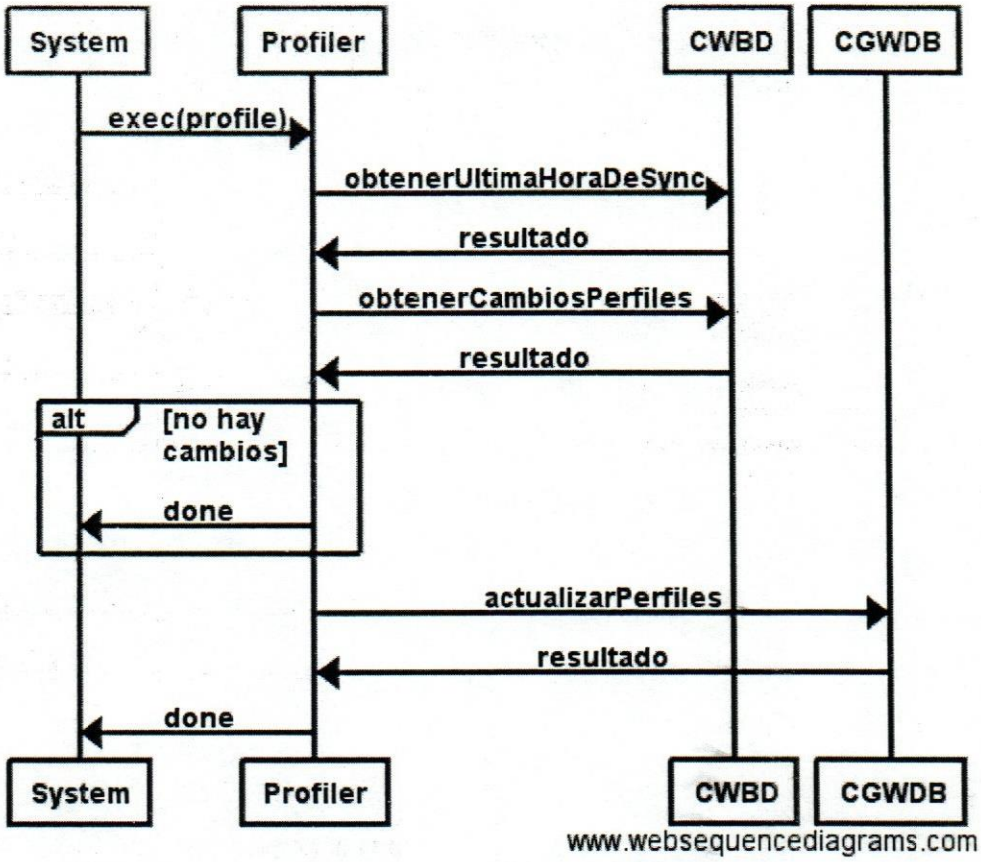

FIGURA 195 *DIAGRAMA DE SECUENCIA SINCRONIZACIÓN DE PERFILES CITYWALLET CON CENTRAL GATEWAY*
Fuente: Elaboración propia

v. Sincronización de perfiles Central Gateway – POS

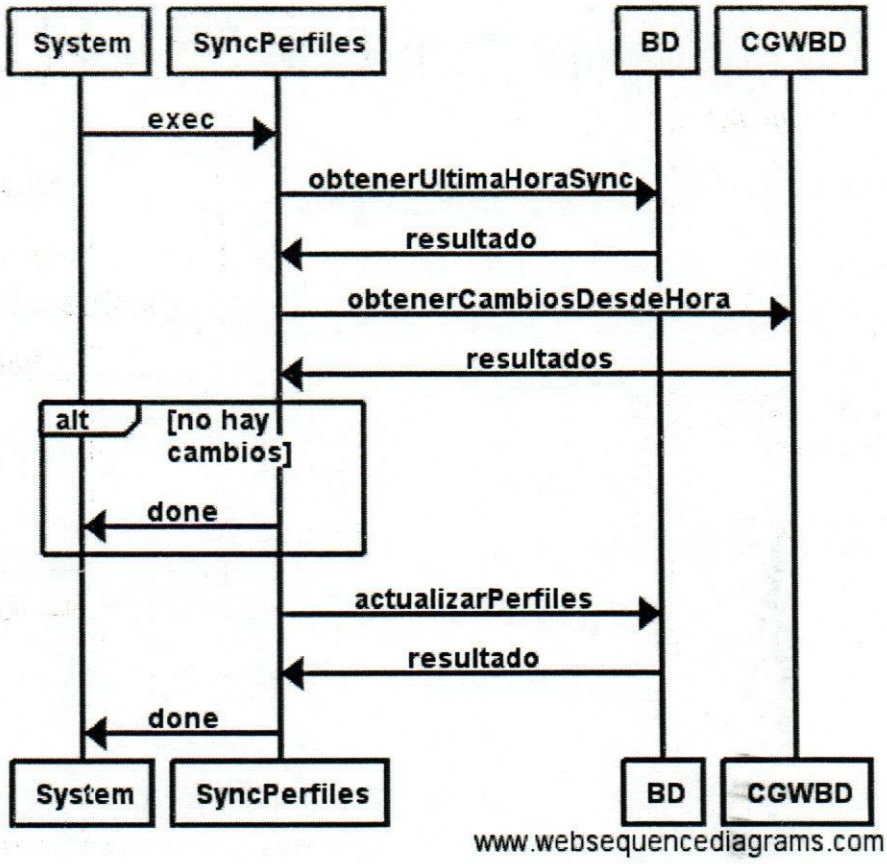

FIGURA 196 ***DIAGRAMA DE SECUENCIA SINCRONIZACIÓN DE PERFILES CENTRAL GATEWAY A POS***
Fuente: Elaboración propia

w. GetUser

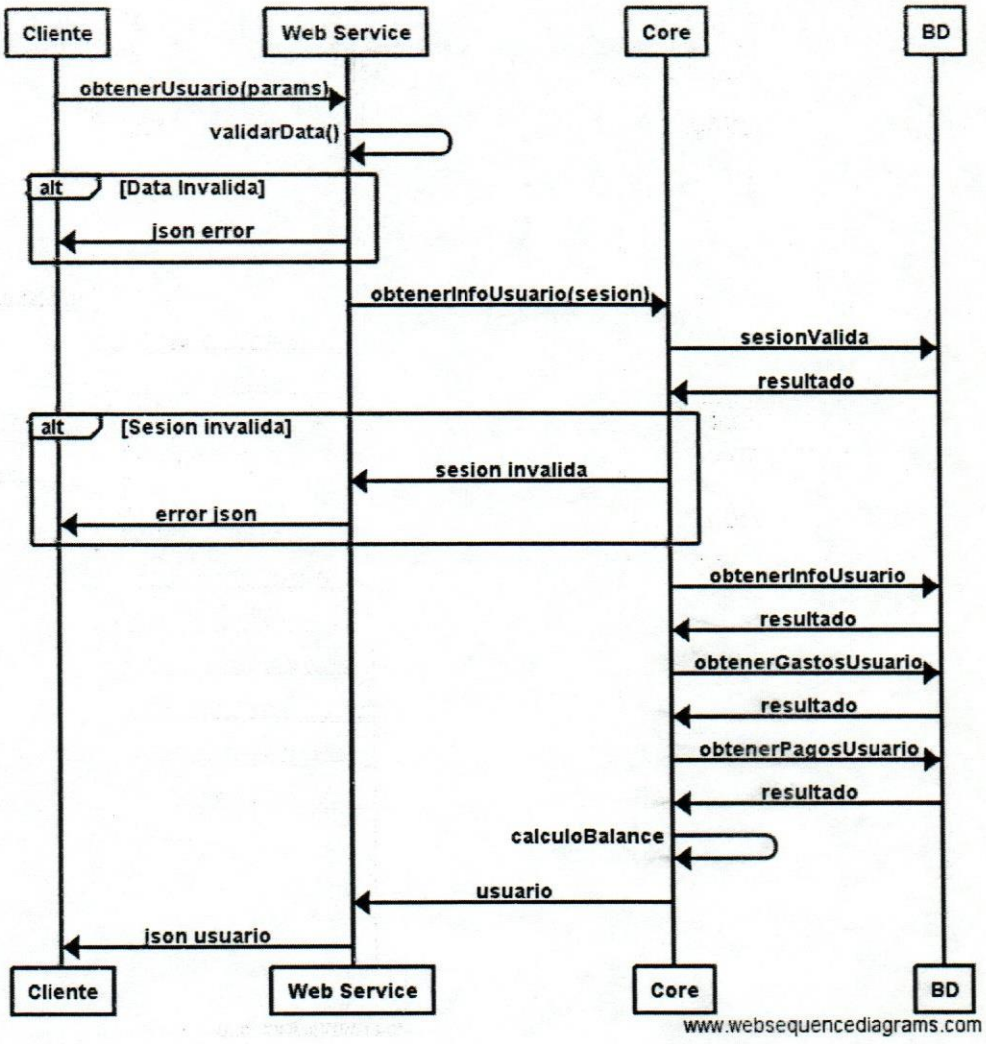

FIGURA 197 ***DIAGRAMA DE SECUENCIA OBTENER USUARIO***
Fuente: Elaboración propia

x. Creación de cuenta con Ripple

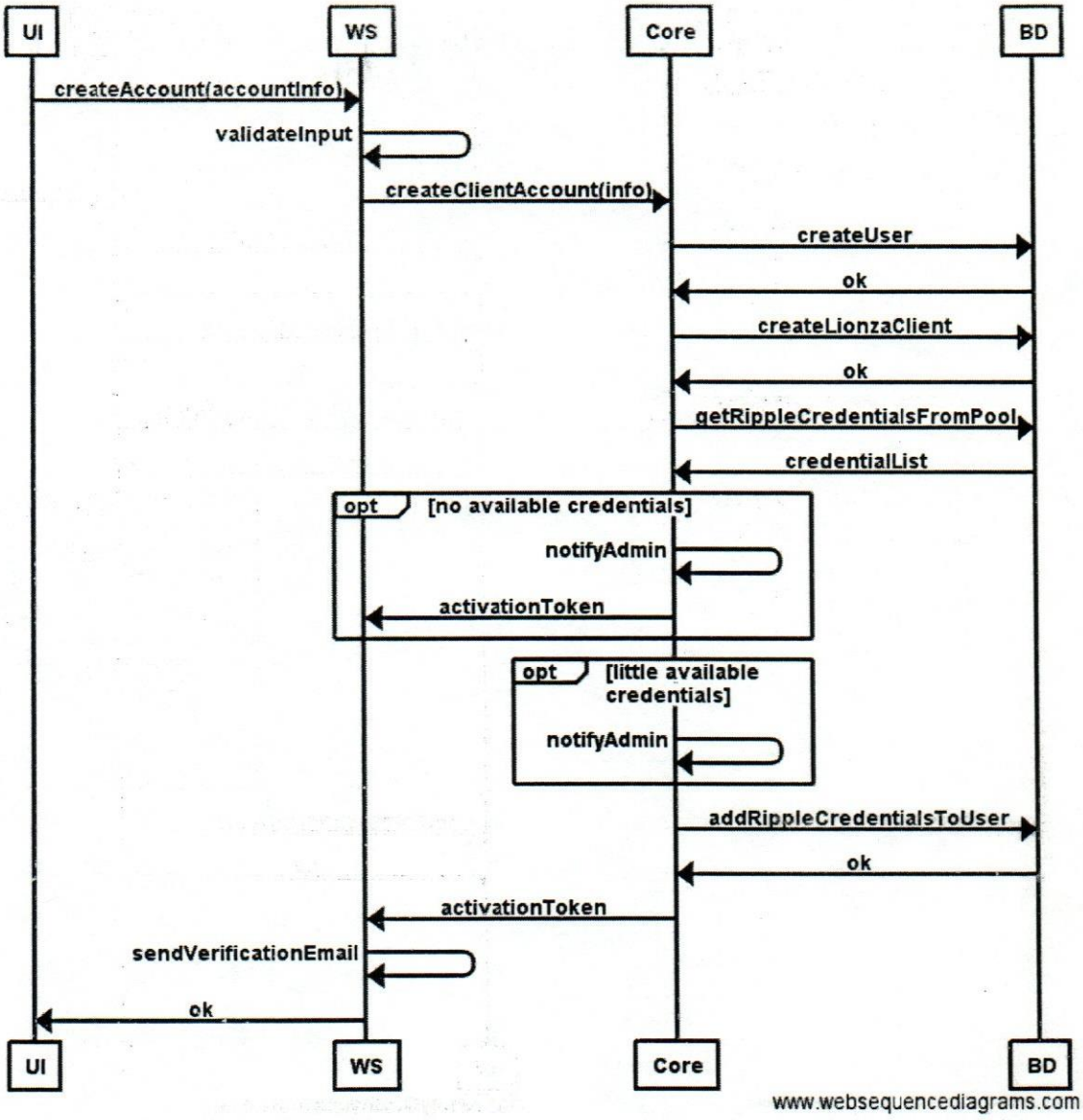

FIGURA 198 ***DIAGRAMA DE SECUENCIA ASOCIACIÓN DE CUENTA CON RIPPLE VERSIÓN 1***
Fuente: elaboración propia

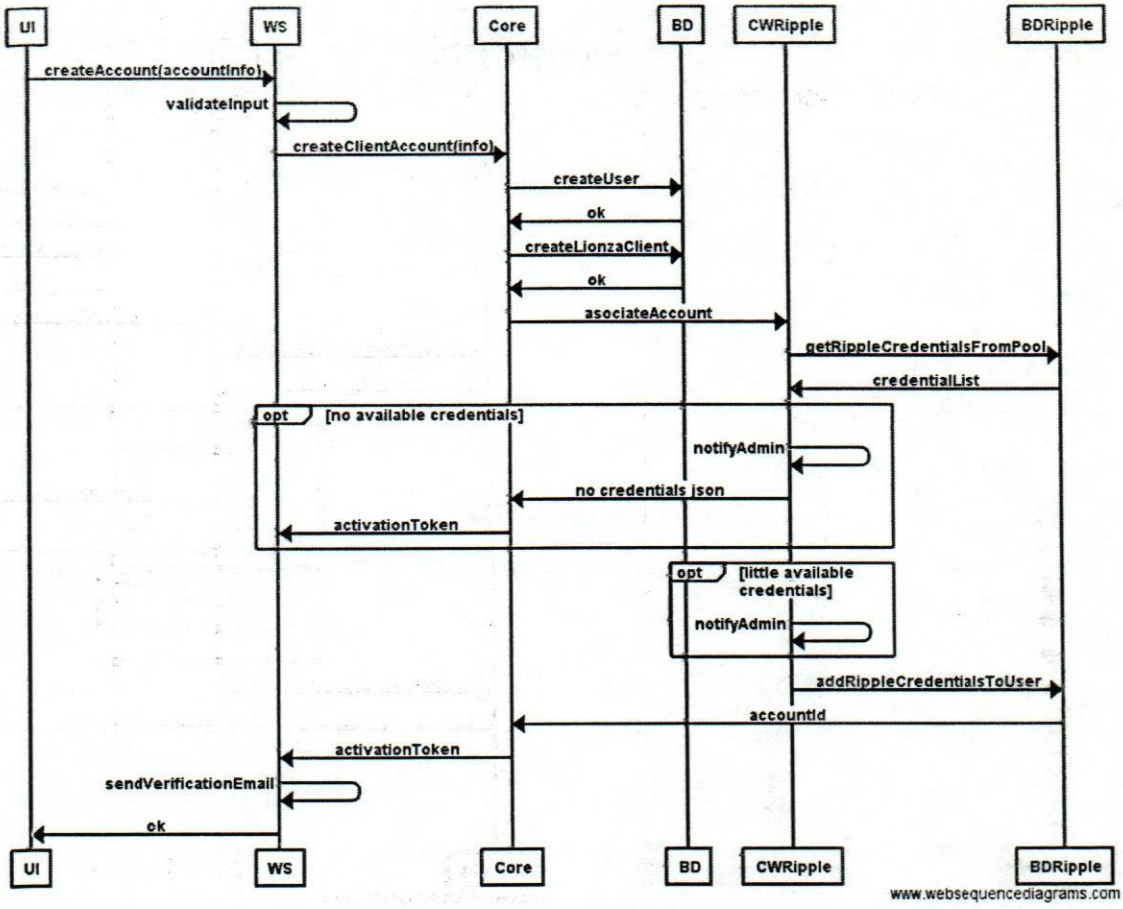

FIGURA 199 ***DIAGRAMA DE SECUENCIA DE CREACIÓN DE CUENTA CON RIPPLE VERSIÓN 2***
Fuente: elaboración propia

y. Ingreso de dinero con Ripple

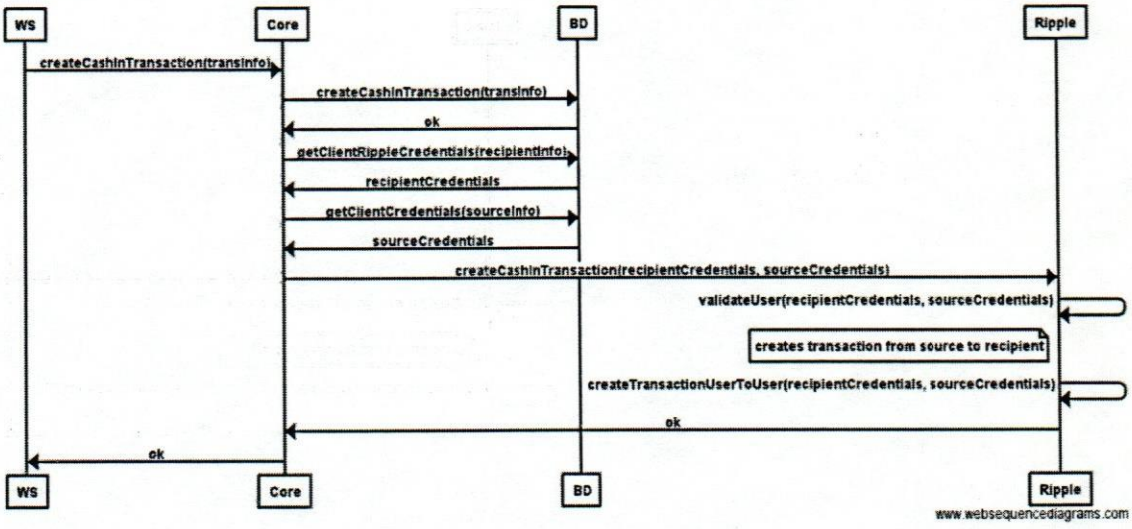

FIGURA 200 ***DIAGRAMA DE SECUENCIA INGRESO DE DINERO CON RIPPLE VERSIÓN 1***
Fuente: elaboración propia

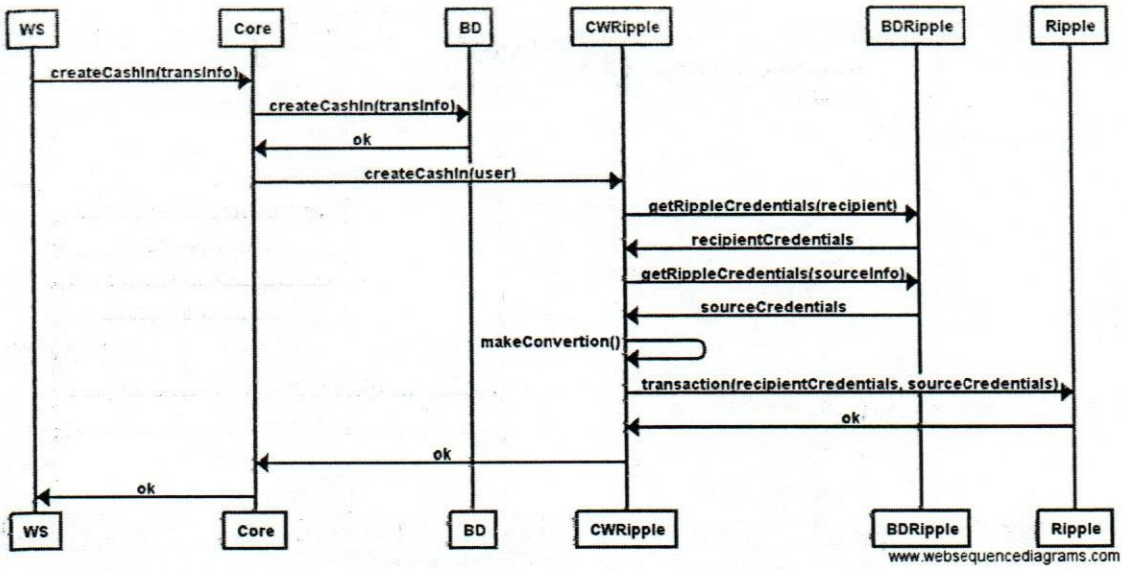

FIGURA 201 ***DIAGRAMA DE SECUENCIA INGRESO DE DINERO CON RIPPLE VERSIÓN 2***
Fuente: elaboración propia

F. Diagramas de Proceso

a. Proceso de creación de cuenta

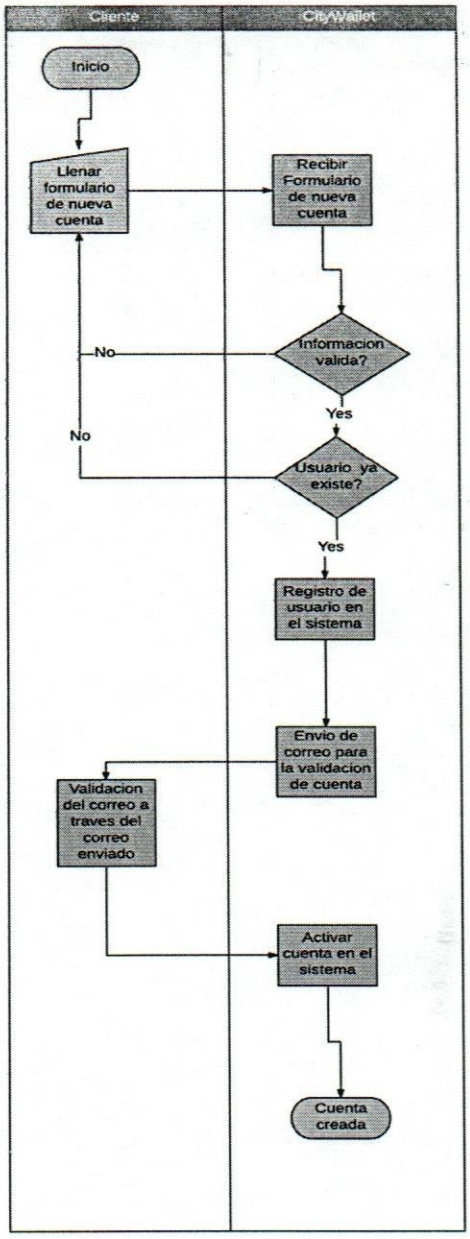

FIGURA 202 ***DIAGRAMA DE PROCESO CREACIÓN DE CUENTA***
Fuente: Elaboración propia

b. Proceso de recuperación de clave

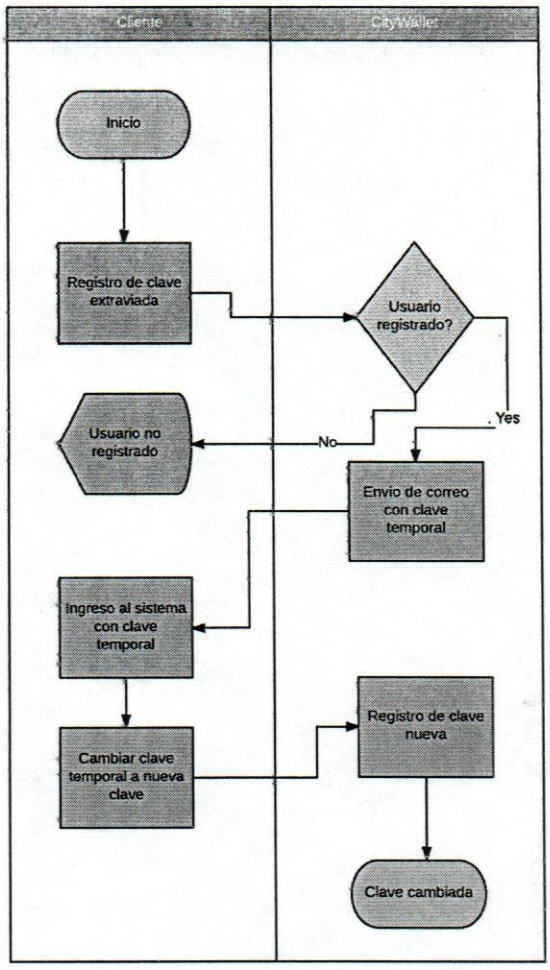

FIGURA 203 ***DIAGRAMA DE PROCESO DE RECUPERACIÓN DE CLAVE***
Fuente: Elaboración propia

c. Proceso de transacciones

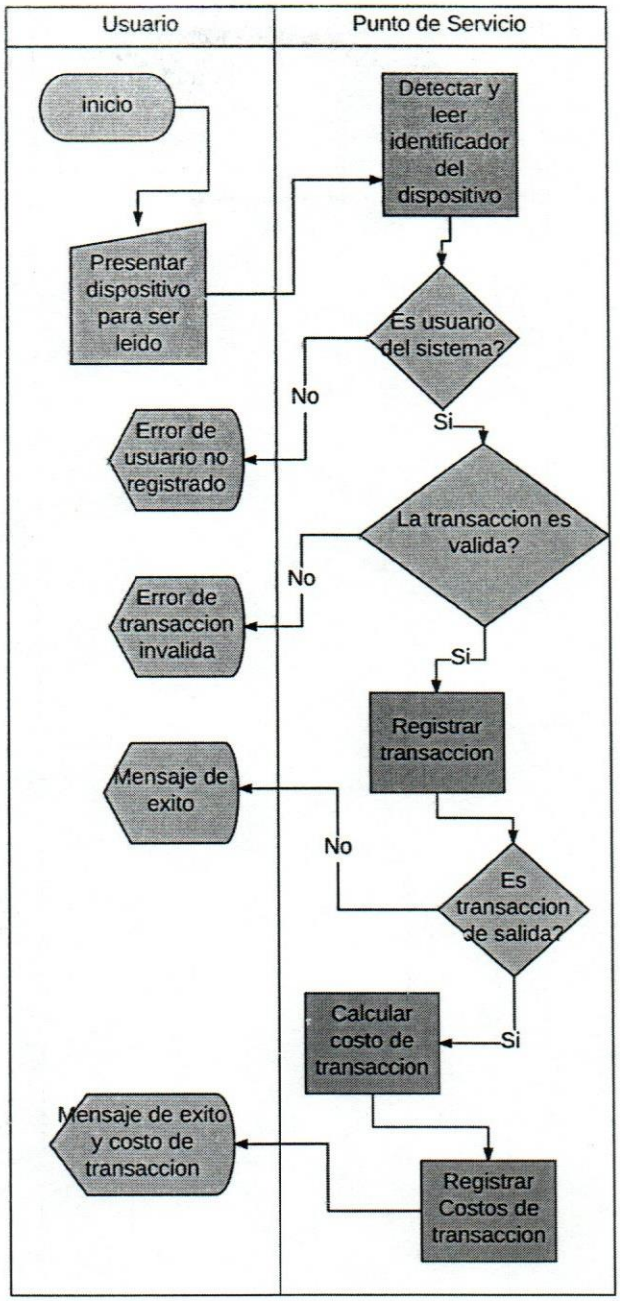

FIGURA 204 ***DIAGRAMA DE PROCESO TRANSACCIONES DE PAGO EN POS***
Fuente: Elaboración propia

d. Proceso de Sincronización de Transacciones

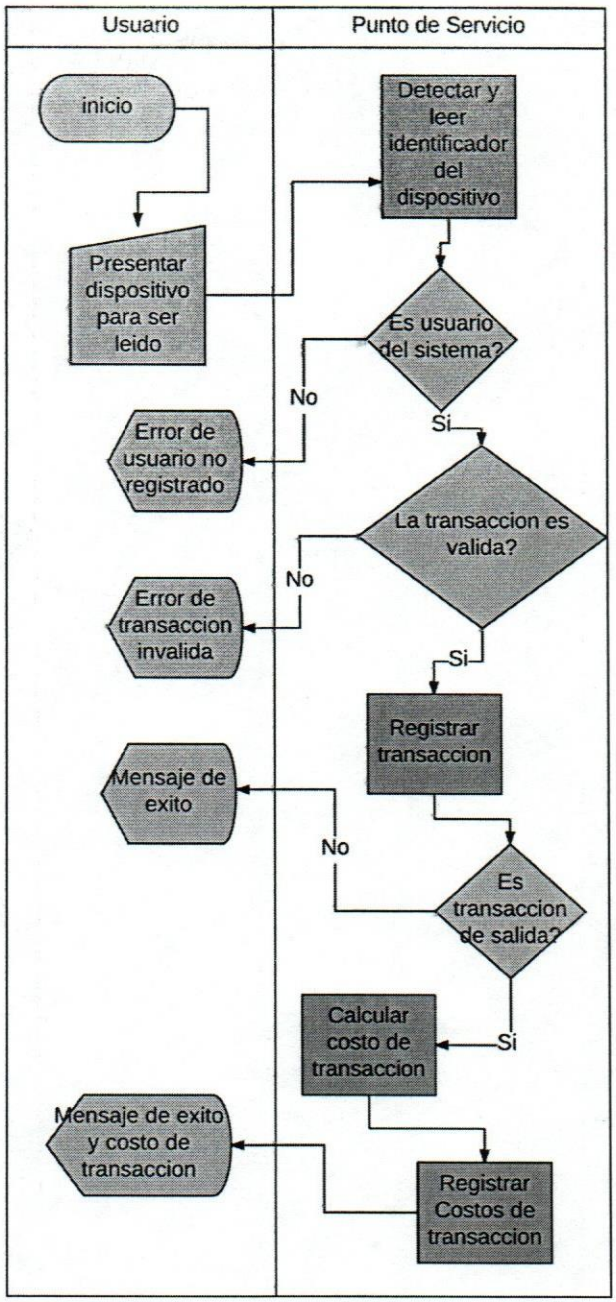

FIGURA 205 ***DIAGRAMA DE PROCESO DE SINCRONIZACIÓN DE TRANSACCIONES***
Fuente: Elaboración propia

e. Proceso de Sincronización de Perfiles CityWallet con Central Gateway

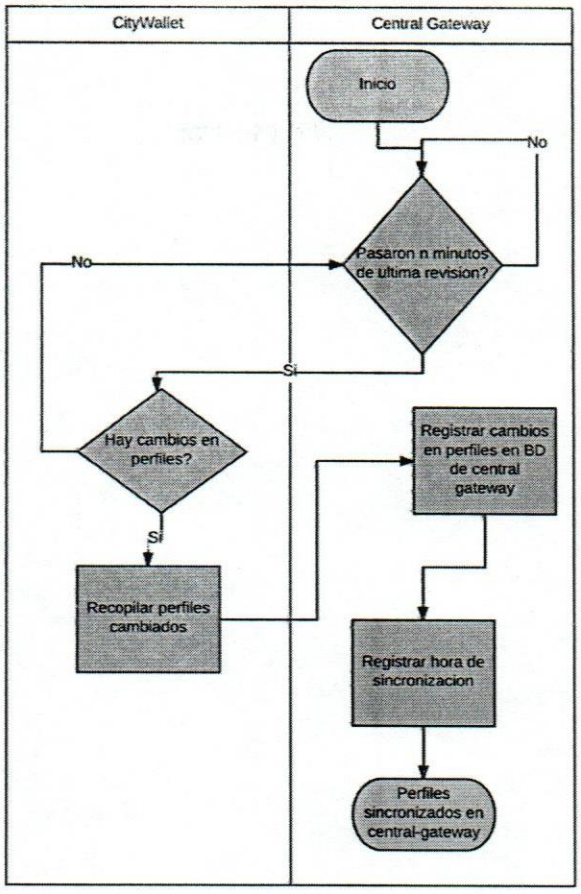

FIGURA 206 ***DIAGRAMA DE PROCESO SINCRONIZACIÓN DE PERFILES CITYWALLET Y CENTRAL GATEWAY***
Fuente: Elaboración propia

f. Proceso de Sincronización de Perfiles Pos con Central Gateway

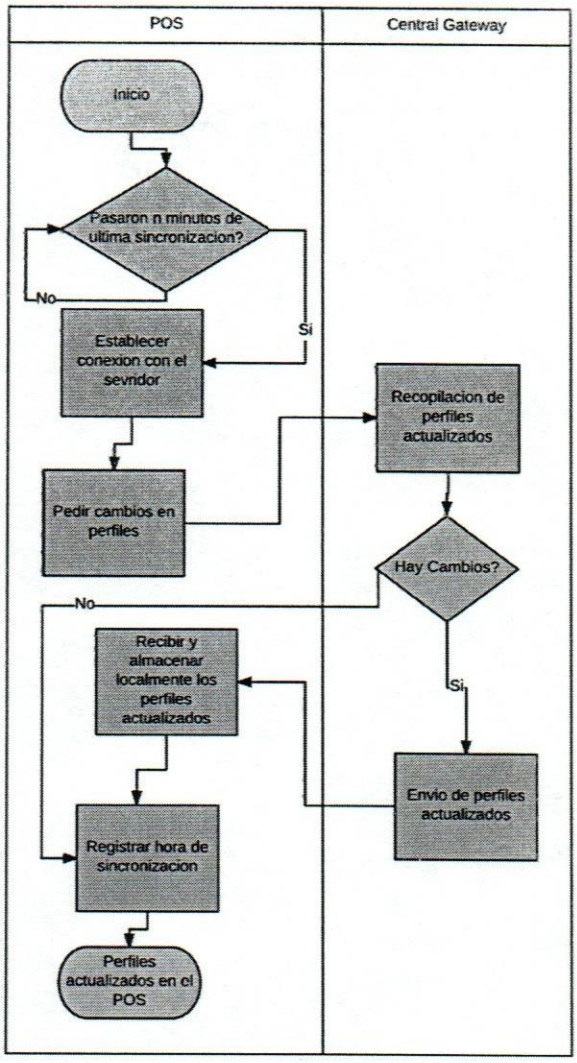

FIGURA 207 ***DIAGRAMA DE PROCESO SINCRONIZACIÓN PERFILES CENTRAL GATEWAY Y POS***
Fuente: Elaboración propia

g. Proceso de asociación de cashout devices

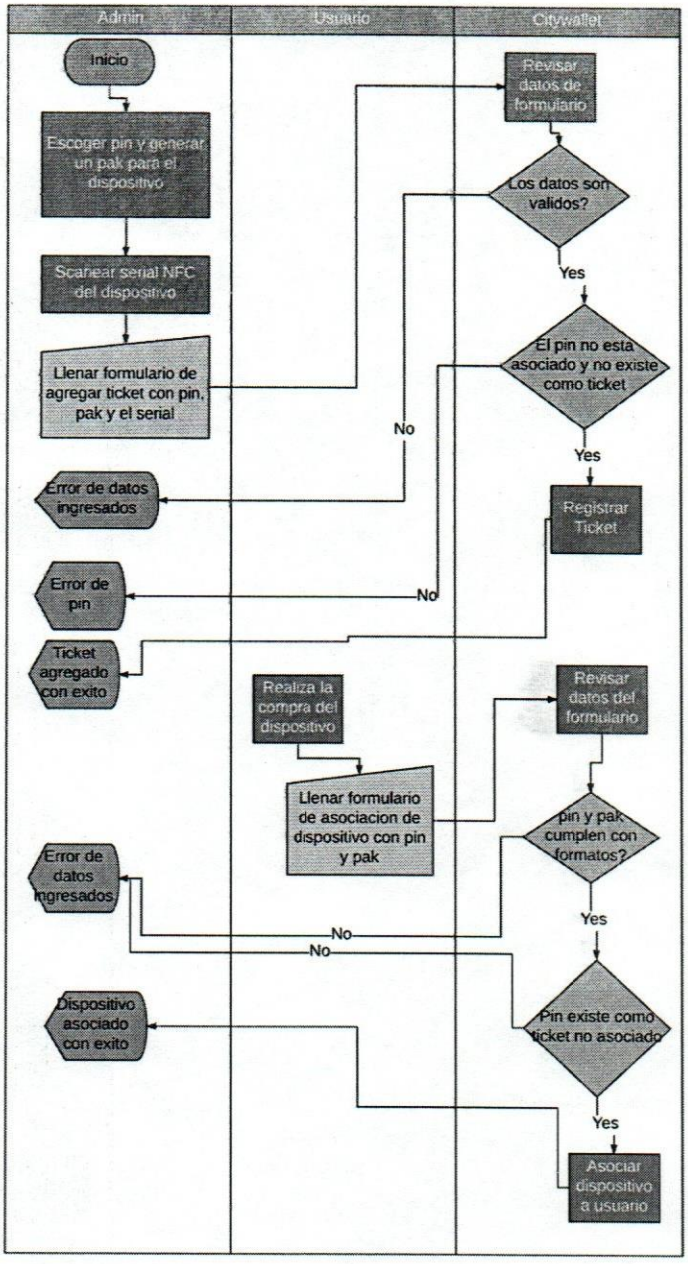

FIGURA 208 ***DIAGRAMA DE PROCESO DE ASOCIACIÓN DE CASHOUT DEVICES***
Fuente: Elaboración propia

h. Proceso de pago instapago

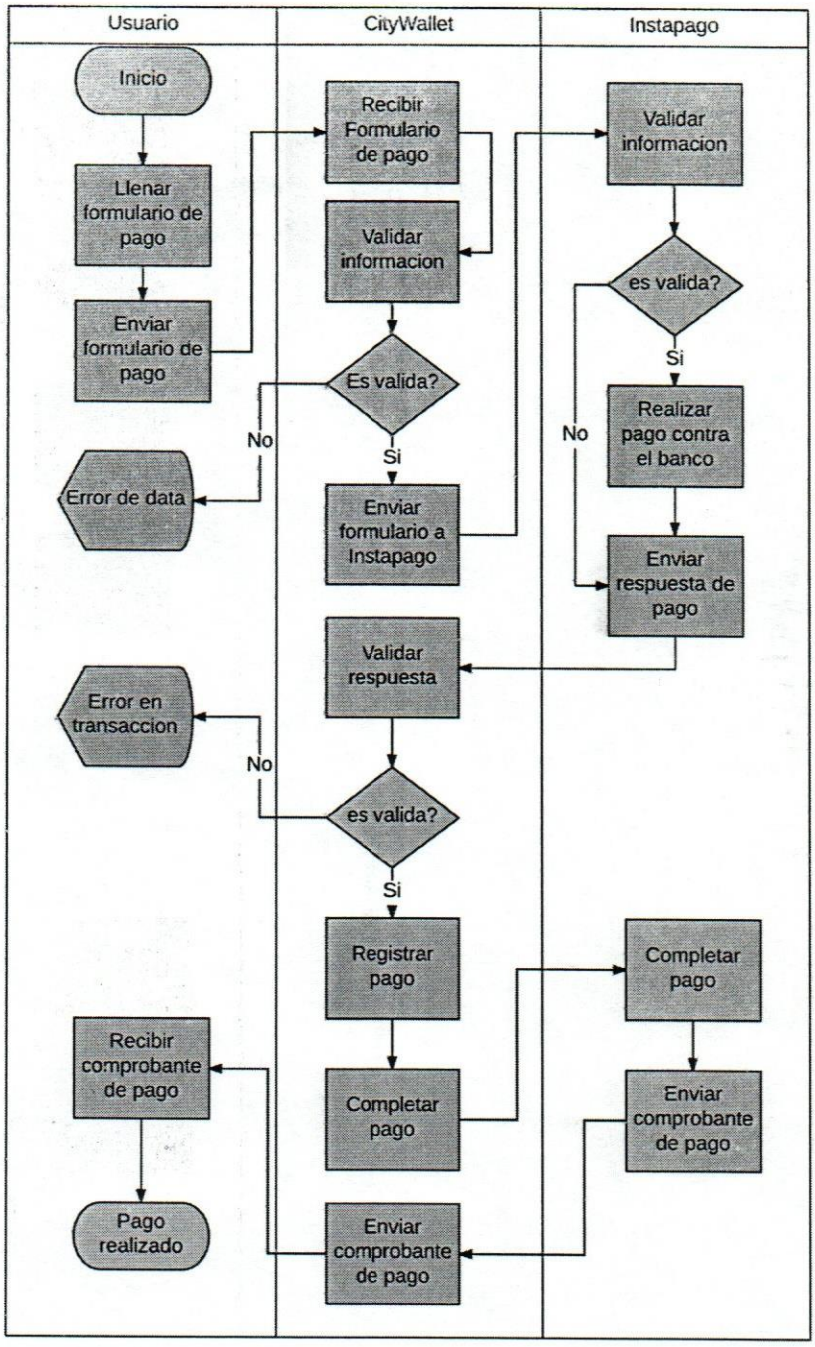

FIGURA 209 ***PROCESO DE PAGO CON INSTAPAGO***
Fuente: Elaboración propia

i. Proceso de asociación de cuentas CityWallet con Ripple

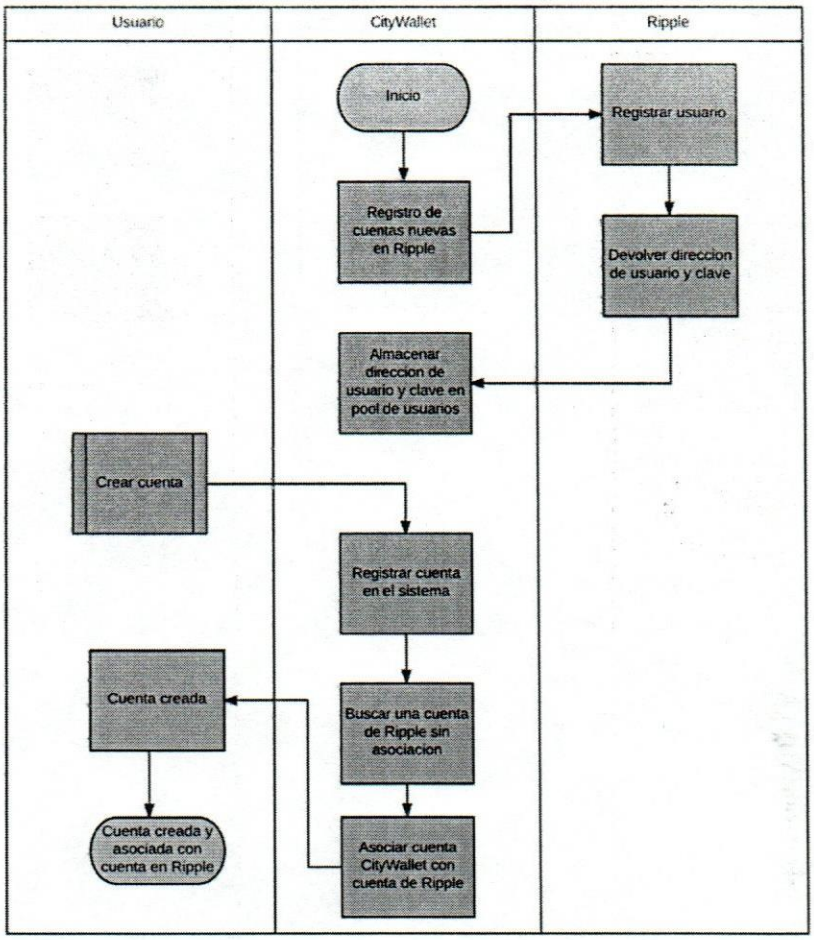

FIGURA 210 ***PROCESO DE ASOCIACIÓN DE CUENTAS CITYWALLET CON RIPPLE***
Fuente: elaboración propia

j. Proceso de creación de ingreso de dinero en CityWallet con Ripple

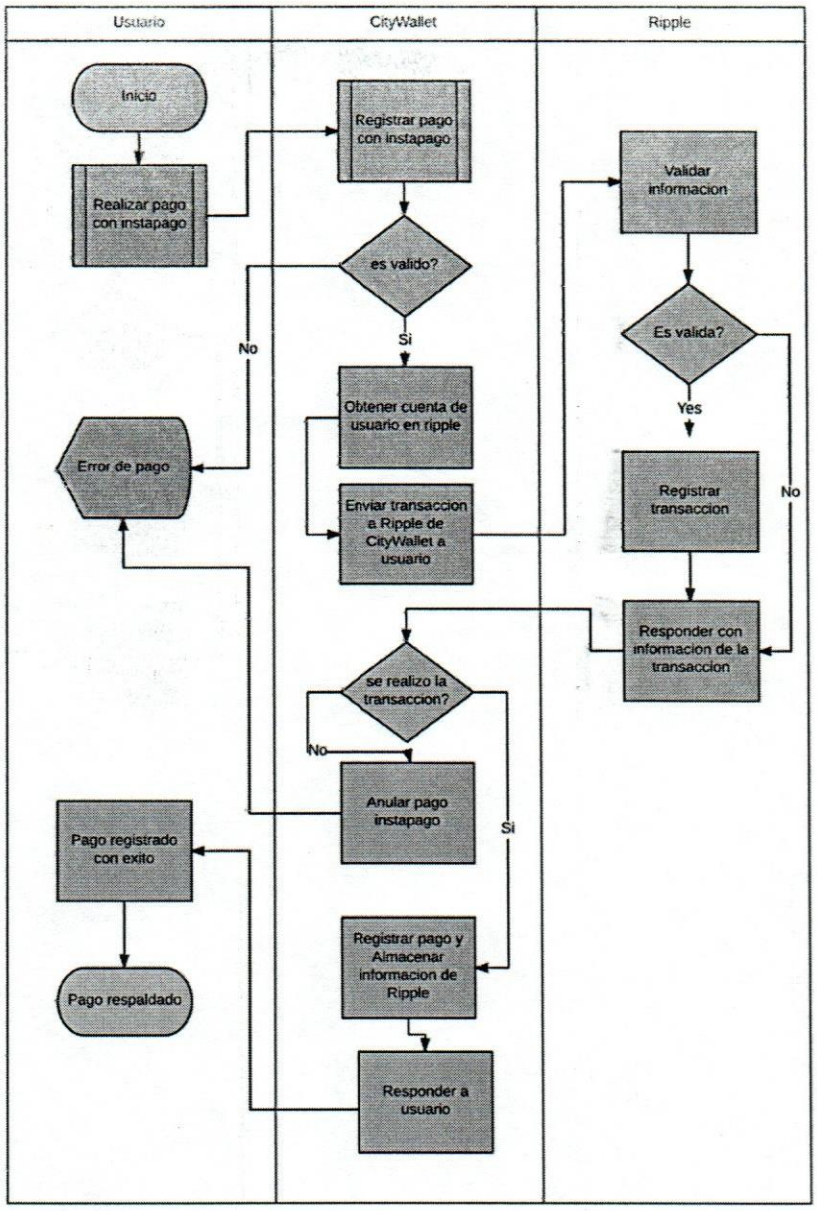

FIGURA 211 ***DIAGRAMA DE PROCESO DE ENTRADA DE DINERO EN CITYWALLET CON RIPPLE***
Fuente: Elaboración propia

G. Capturas de Pantalla

a. Modulo Administrativo

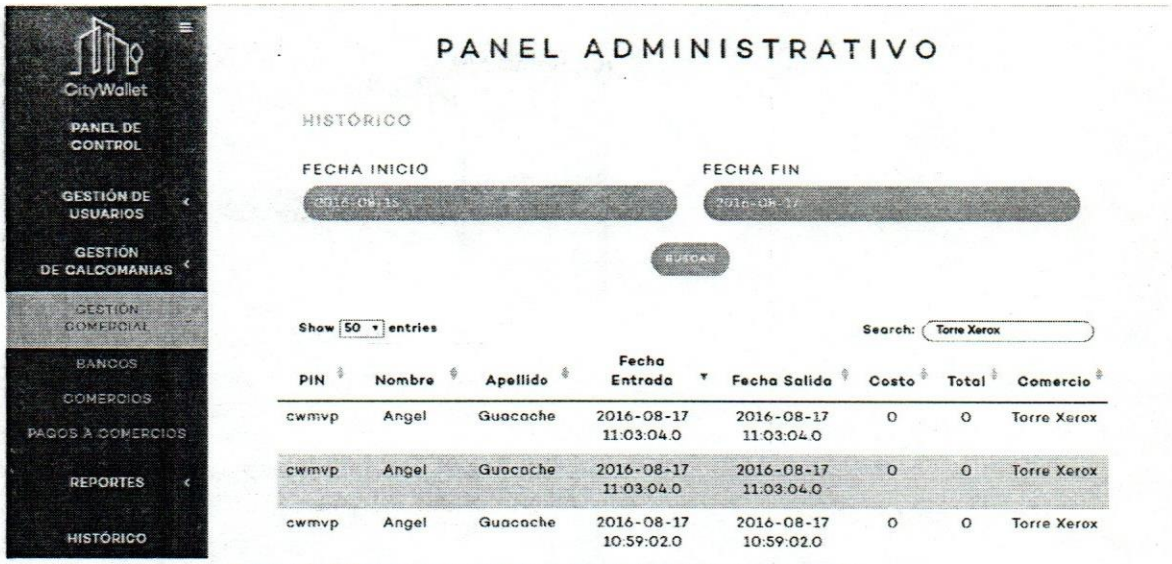

FIGURA 212 ***CAPTURA DE PANTALLA, TRANSACCIONES POR COMERCIO***
Fuente: Elaboración propia

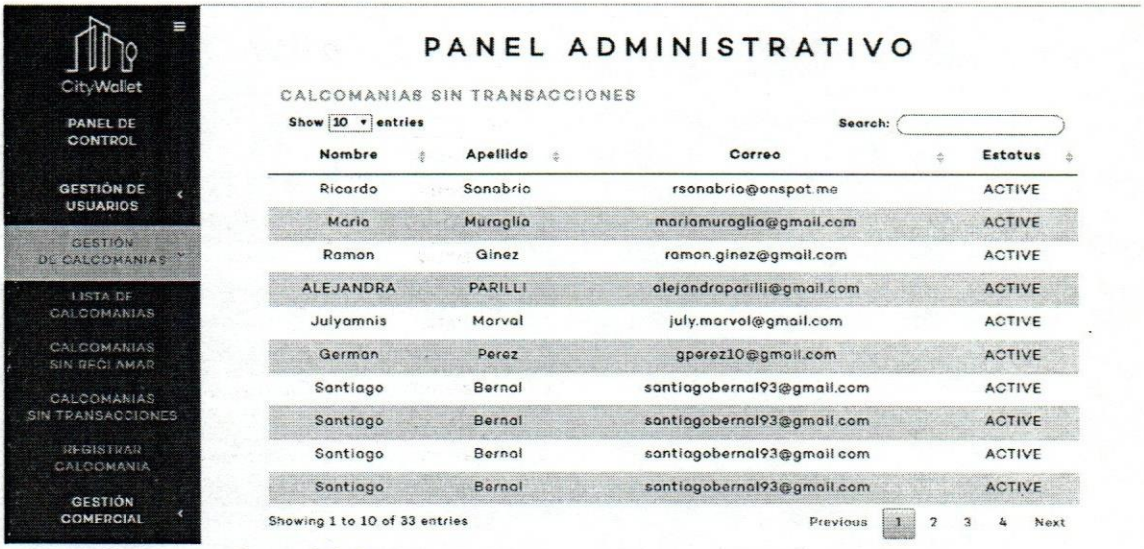

FIGURA 213 ***CAPTURA DE PANTALLA, USUARIOS SIN TRANSACCIONES***
Fuente: Elaboración propia

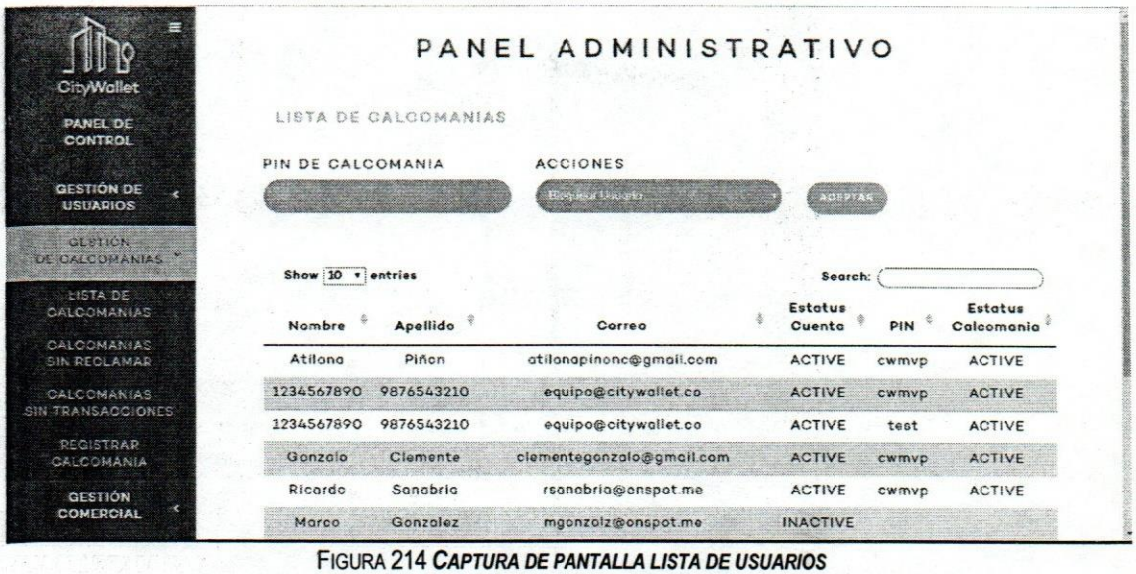

FIGURA 214 ***CAPTURA DE PANTALLA LISTA DE USUARIOS***
Fuente: Elaboración propia

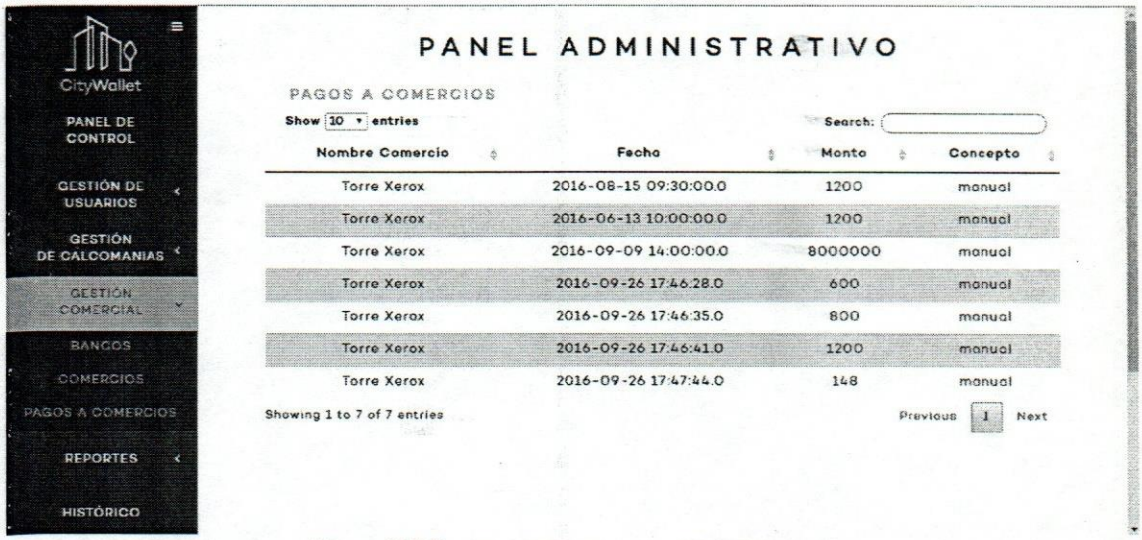

FIGURA 215 ***CAPTURA DE PANTALLA PAGOS A COMERCIOS***
Fuente: Elaboración propia

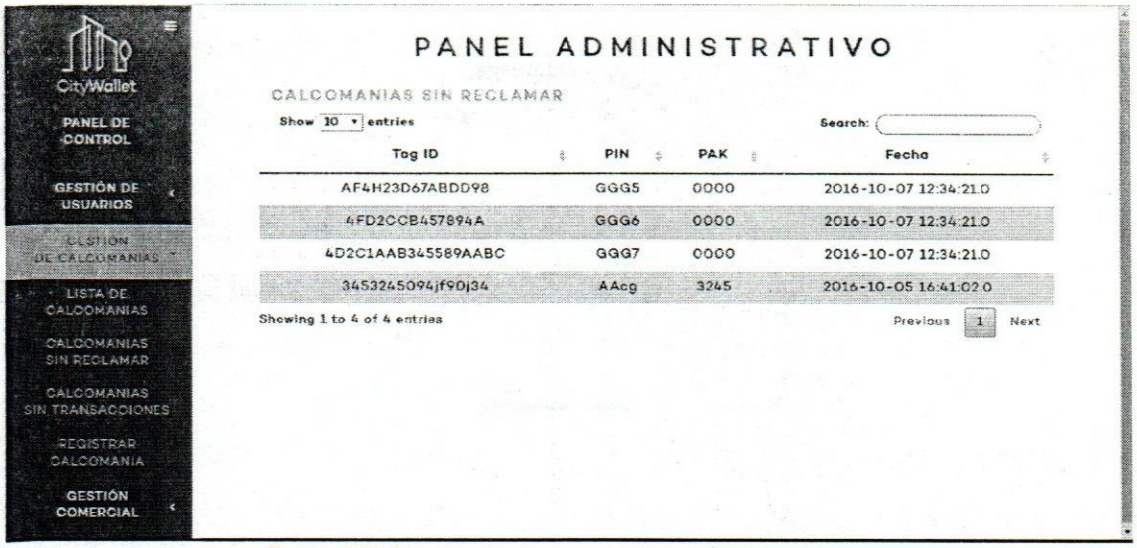

FIGURA 216 ***CAPTURA DE PANTALLA CALCOMANÍAS SIN RECLAMAR***
Fuente: Elaboración propia

b. Modulo Administrativo de Comercio

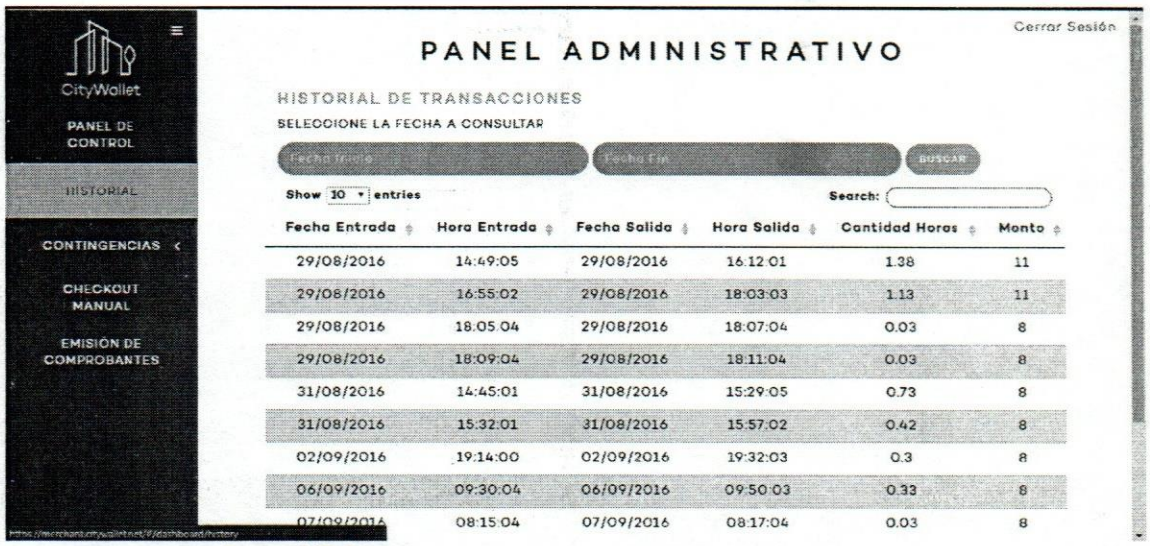

FIGURA 217 ***CAPTURA DE PANTALLA TRANSACCIONES EN UN COMERCIO***
Fuente: Elaboración propia

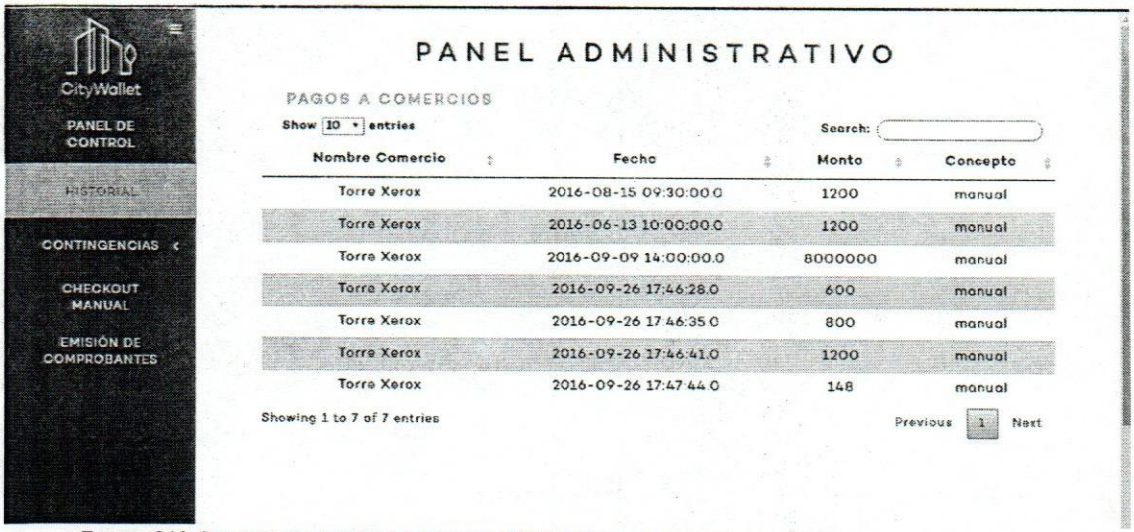

FIGURA 218 ***CAPTURA DE PANTALLA PAGOS DE CW A COMERCIOS MODULO ADMINISTRATIVO DE COMERCIOS***
Fuente: Elaboración propia

H. Correos de Alerta

root Sent on: Monday, October 10 2016, 5:27 pm

Error en pos-ucab-in

To infra

```
2016-10-08 09:36:04 Error - Error code: 7 Msj:Non Existing TagId
```

FIGURA 219 ***CORREO DE ERROR EN POS***
Fuente: Elaboración propia

```
Notification Type: RECOVERY

Service: Internet Info Speed
Host: UCABGateway
Address: 192.168.5.13
State: OK

Date/Time: Tue Oct 11 12:20:39 VET 2016

Additional Info:

External IP: 200.2.11.131, Hosted by movistar (Caracas) [1.91 km]: 8.112 ms
```

FIGURA 220 ***CORREO DE NOTIFICACIÓN DE VELOCIDAD DE INTERNET***
Fuente: Elaboración propia

```
Notification Type: RECOVERY
Host: CELLPosSalida
State: UP
Address: 192.168.5.12
Info: PING OK - Packet loss = 0%, RTA = 154.60 ms

Date/Time: Fri Oct 7 19:32:48 VET 2016
```

FIGURA 221 ***CORREO DE NOTIFICACIÓN DE ESTADO DE SERVICIOS DE POS***
Fuente: Elaboración propia

```
Notification Type: PROBLEM
Host: UCABPosEntrada
State: DOWN
Address: 192.168.5.15
Info: CRITICAL - Host Unreachable (192.168.5.15)

Date/Time: Mon Oct 10 18:49:36 VET 2016
```

FIGURA 222 ***CORREO DE NOTIFICACIÓN DE SERVICIOS CAÍDOS EN POS***
Fuente: Elaboración propia

CityWallet

Notificación de transacciones en pernocta

Las siguientes transacciones del día se encuentran abiertas en pernocta:

id: 3166, clients full name: Santiago Bernal, merchant: Torre Xerox, date: Fri Oct 07 10:30:00 VET 2016
id: 3167, clients full name: Santiago Bernal, merchant: Torre Xerox, date: Fri Oct 07 09:30:00 VET 2016
id: 3168, clients full name: Santiago Bernal, merchant: Torre Xerox, date: Fri Oct 07 11:30:00 VET 2016
id: 3169, clients full name: mannuel pachecp, merchant: Torre Xerox, date: Fri Oct 07 11:30:00 VET 2016
id: 3170, clients full name: alejandro garcia, merchant: Torre Xerox, date: Fri

FIGURA 223 ***CORREO DE NOTIFICACIÓN DE USUARIOS EN PERNOCTA***
Fuente: Elaboración propia

Notificaciones Sent on: Friday, October 7 2016, 1:54 pm

Error en monto de transaccion

To Santiago Bernal

CityWallet

Notificación de error en montos de transacciones

Se ha encontrado errores en las siguientes transacciones:

id: 177, total amount: 94.0, calculated amount: 78.0, date: Thu Dec 03 14:43:45 VET 2015
id: 180, total amount: 46.0, calculated amount: 39.0, date: Thu Dec 03 18:30:53 VET 2015
id: 181, total amount: 27.0, calculated amount: 13.0, date: Fri Dec 04 09:08:28 VET 2015
id: 182, total amount: 59.0, calculated amount: 52.0, date: Fri Dec 04 16:02:01 VET 2015

FIGURA 224 ***CORREO DE ERROR EN MONTOS DE TRANSACCIONES***
Fuente: Elaboración propia