

TEMA:

DOMINIO DEL UNIVERSO: BINGO

OBJETIVO:

Poner en practica los conocimientos obtenidos de como trabaja una base de datos en este caso hacia un software de un bingo

Estudiante:
Macias Pico Josselyn Stefany

Fuente: Propia

Índice

- [Dominio del Universo](#)
- [Modelo Conceptual](#)
- [PostgreSQL: Crear Tablas](#)
- [PostgreSQL: Insertar Datos](#)
- [PostgreSQL: Consultas](#)
- [Conclusiones](#)

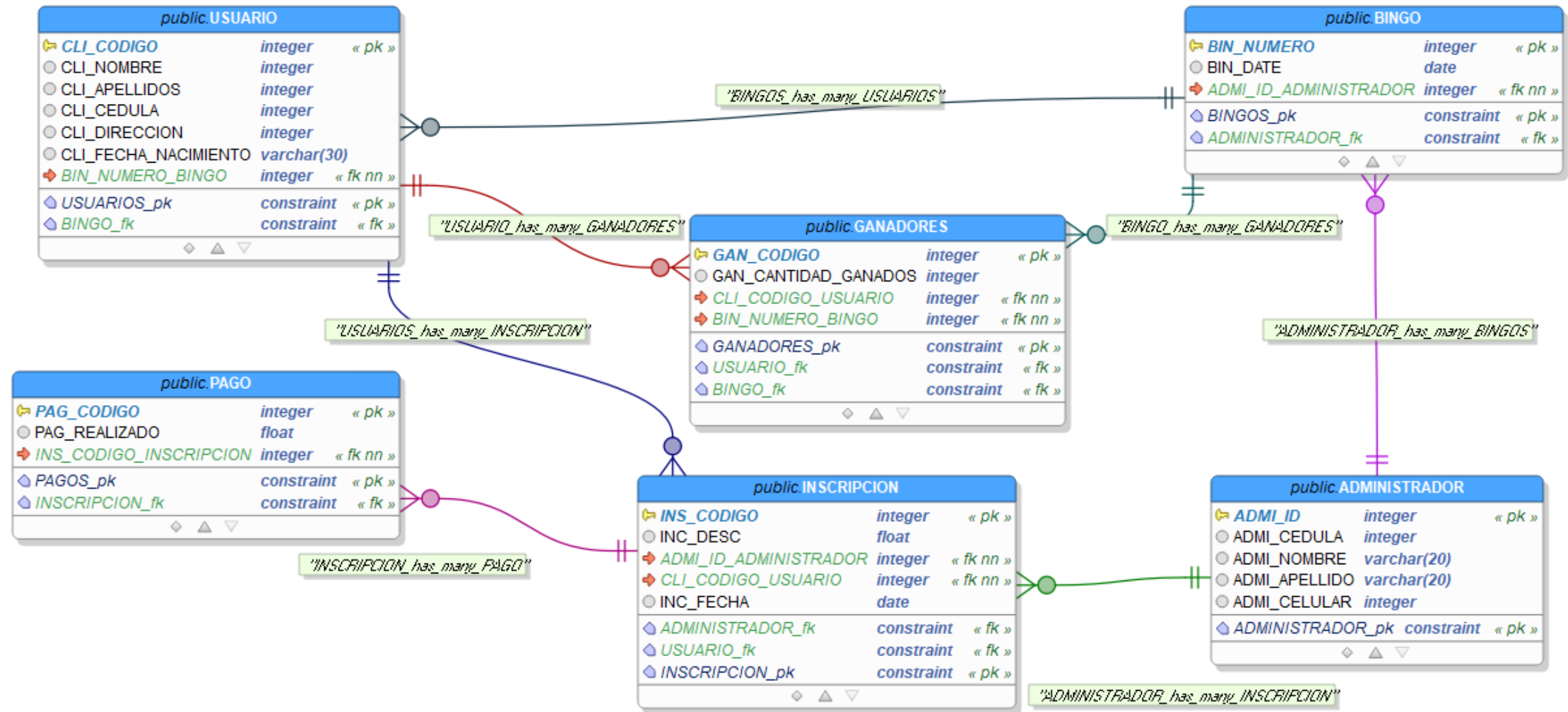
Dominio del universo

La cadena de Bingos online ha solicitado un sistema informático para Bingos el cual se ha nombrado “BINGOMAX” utilizado para la gestión de su negocio este modelo de datos debe contener lo siguiente:

La cadena cuenta con usuarios de los cuales se debe obtener el Código, los nombres, apellidos, cedula, dirección, fecha de nacimiento, fecha de inscripción y fecha de pago. También cuenta con un administrador del cual se debe conocer el nombre, celular, cedula y el usuario a el cual administra. El administrador debe generar un descuento de 10% a las personas inscritas 10 días antes del 10 de cada mes en el que comienza el bingo teniendo en cuenta que la inscripción cuesta \$40. A continuación, se muestran los primeros BINGOS realizados:

NUMERO DE BINGO	FECHA DEL BINGO	PERSONAS INSCRITAS	ADMINISTRADOR
001	10/07/2020	40 personas	Stefany Pico Mendoza
002	10/08/2020	35 personas	Pedro Moreira Cevallos
003	10/09/2020	37 personas	Tony Anchundia Murillo
004	10/10/2020	50 personas	Lucia Delgado Mantuano
005	10/11/2020	54 personas	Jacqueline Alcívar Mendoza
006	10/12/2020	40 personas	Eduardo Zambrano Cedeño
007	10/01/2021	50 personas	Yulezti Álvarez Rodríguez

Modelo Conceptual



PostgreSQL: Crear Tablas

```
/*CREAR TABLA DE ADMINISTRADOR*/  
CREATE TABLE ADMINISTRADOR(  
    ADMI_ID INTEGER PRIMARY KEY,  
    ADMI_CEDULA INTEGER,  
    ADMI_NOMBRE VARCHAR (20),  
    ADMI_APELLIDO VARCHAR(20),  
    ADMI_CELULAR INTEGER  
);
```

```
/*CREAR TABLA DE USUARIO*/  
CREATE TABLE USUARIO(  
    CLI_CODIGO INT PRIMARY KEY,  
    CLI_NOMBRE VARCHAR(20),  
    CLI_APELLIDO VARCHAR(20),  
    CLI_CEDULA INTEGER,  
    CLI_DIRECCION VARCHAR(30),  
    CLI_FECHA_NACIMIENTO DATE,  
    BIN_NUMERO INTEGER NULL,  
    CONSTRAINT BIN_NUMERO foreign key (BIN_NUMERO)  
    REFERENCES BINGO(BIN_NUMERO) ON UPDATE CASCADE  
);
```

```
/*CREAR TABLA DE BINGO*/  
CREATE TABLE BINGO(  
    BIN_NUMERO INTEGER PRIMARY KEY,  
    BIN_DATE DATE,  
    ADMI_ID INTEGER NULL,  
    CONSTRAINT ADMI_ID foreign key (ADMI_ID)  
    REFERENCES ADMINISTRADOR(ADMI_ID) ON UPDATE CASCADE  
);
```

```
/*CREAR TABLA DE USUARIO*/  
CREATE TABLE USUARIO(  
    CLI_CODIGO INT PRIMARY KEY,  
    CLI_NOMBRE VARCHAR(20),  
    CLI_APELLIDO VARCHAR(20),  
    CLI_CEDULA INTEGER,  
    CLI_DIRECCION VARCHAR(30),  
    CLI_FECHA_NACIMIENTO DATE,  
    BIN_NUMERO INTEGER NULL,  
    CONSTRAINT BIN_NUMERO foreign key (BIN_NUMERO)  
    REFERENCES BINGO(BIN_NUMERO) ON UPDATE CASCADE  
);
```

PostgreSQL: Crear Tablas

```
/*CREAR TABLA DE INSCRIPCION*/
CREATE TABLE INSCRIPCION(
INC_CODIGO INTEGER PRIMARY KEY,
INC_DESCUENTO FLOAT,
INC_FECHA DATE,
ADMI_ID INTEGER NULL,
CLI_CODIGO INT NULL,
CONSTRAINT ADMI_ID foreign key (ADMI_ID)
REFERENCES ADMINISTRADOR(ADMI_ID) ON UPDATE CASCADE,
CONSTRAINT CLI_CODIGO foreign key (CLI_CODIGO )
REFERENCES USUARIO(CLI_CODIGO ) ON UPDATE CASCADE
);
```

```
/*CREAR TABLA DE PAGO*/
CREATE TABLE PAGO(
PAG_CODIGO INTEGER PRIMARY KEY,
PAG_REALIZADO FLOAT,
INC_CODIGO INTEGER NULL,
CONSTRAINT INC_CODIGO foreign key (INC_CODIGO )
REFERENCES INSCRIPCION(INC_CODIGO ) ON UPDATE CASCADE
);
```

```
/*CREAR TABLA DE GANADORES*/
CREATE TABLE GANADORES(
GAN_CODIGO INTEGER PRIMARY KEY,
GAN_CANTIDAD_GANADOS INTEGER ,
BIN_NUMERO INTEGER NULL,
CLI_CODIGO INTEGER NULL,
CONSTRAINT BIN_NUMERO foreign key (BIN_NUMERO)
REFERENCES BINGO(BIN_NUMERO) ON UPDATE CASCADE,
CONSTRAINT CLI_CODIGO foreign key (CLI_CODIGO)
REFERENCES USUARIO(CLI_CODIGO) ON UPDATE CASCADE
);
```


PostgreSQL: Insertar datos

```

/*INSERTAR DATOS DE ADMINISTRADOR*/
INSERT INTO ADMINISTRADOR (ADMI_ID, ADMI_CEDULA, ADMI_NOMBRE, ADMI_APELLIDO, ADMI_CELULAR) VALUES
(001,1315698526, 'STEFANY' , ' PICO MENDOZA ', 0988682451);
INSERT INTO ADMINISTRADOR (ADMI_ID, ADMI_CEDULA, ADMI_NOMBRE, ADMI_APELLIDO, ADMI_CELULAR) VALUES
(002,1315698526 , 'JACQUELINE', 'ALCIVAR MENDOZA', 0988682451);
INSERT INTO ADMINISTRADOR (ADMI_ID, ADMI_CEDULA, ADMI_NOMBRE, ADMI_APELLIDO, ADMI_CELULAR) VALUES
(003,1379354120 , 'LUCIA', 'DELGADO MANTUANO', 0991843520);
INSERT INTO ADMINISTRADOR (ADMI_ID, ADMI_CEDULA, ADMI_NOMBRE, ADMI_APELLIDO, ADMI_CELULAR) VALUES
(004,1387565254, 'PEDRO', 'MOREIRA CEVALLOS', 0943118520);
INSERT INTO ADMINISTRADOR (ADMI_ID, ADMI_CEDULA, ADMI_NOMBRE, ADMI_APELLIDO, ADMI_CELULAR) VALUES
(005,1325547893 , 'EDUARDO', 'ZAMBRANO CEDEÑO', 0978463200);
INSERT INTO ADMINISTRADOR (ADMI_ID, ADMI_CEDULA, ADMI_NOMBRE, ADMI_APELLIDO, ADMI_CELULAR) VALUES
(006,1376008610 , 'TONY', 'ANCHUNDIA MURILLO', 0984521884);
INSERT INTO ADMINISTRADOR (ADMI_ID, ADMI_CEDULA, ADMI_NOMBRE, ADMI_APELLIDO, ADMI_CELULAR) VALUES
(007,1318761054 , 'YULETZI', 'ALVAREZ RODRIGUEZ', 0987325656);

```

Data Output		Explain	Messages	History	
	admi_id integer	admi_cedula integer	admi_nombre character varying	admi_apellido character varying(20)	admi_celular integer
1	1	1315698526	STEFANY	PICO MENDOZA	988682451
2	2	1315698526	JACQUELINE	ALCIVAR MENDOZA	988682451
3	3	1379354120	LUCIA	DELGADO MANTUANO	991843520
4	4	1387565254	PEDRO	MOREIRA CEVALLOS	943118520
5	5	1325547893	EDUARDO	ZAMBRANO CEDEÑO	978463200
6	6	1376008610	TONY	ANCHUNDIA MURILLO	984521884
7	7	1318761054	YULETZI	ALVAREZ RODRIGUEZ	987325656

Link: [Código](#)

PostgreSQL: Insertar datos

```
/*INSERTAR DATOS DE BINGO*/
insert into BINGO (BIN_NUMERO, BIN_DATE, ADMI_ID) VALUES
(0001, '10/07/2020', 001);
insert into BINGO (BIN_NUMERO, BIN_DATE, ADMI_ID) VALUES
(0002, '10/08/2020', 004);
insert into BINGO (BIN_NUMERO, BIN_DATE, ADMI_ID) VALUES
(0003, '10/09/2020', 006);
insert into BINGO (BIN_NUMERO, BIN_DATE, ADMI_ID) VALUES
(0004, '10/10/2020', 003);
insert into BINGO (BIN_NUMERO, BIN_DATE, ADMI_ID) VALUES
(0005, '10/11/2020', 002);
insert into BINGO (BIN_NUMERO, BIN_DATE, ADMI_ID) VALUES
(0006, '10/12/2020', 005);
insert into BINGO (BIN_NUMERO, BIN_DATE, ADMI_ID) VALUES
(0007, '10/01/2021', 007);
```

Data Output	Explain	Messages	History
	bin_numero integer	bin_date date	admi_id integer
1	1	2020-07-10	1
2	2	2020-08-10	4
3	3	2020-09-10	6
4	4	2020-10-10	3
5	5	2020-11-10	2
6	6	2020-12-10	5
7	7	2021-01-10	7

PostgreSQL: Insertar datos

```
/*INSERTAR DATOS DE USUARIO*/
INSERT INTO usuario (CLI_CODIGO, CLI_NOMBRE, CLI_APELLIDO, CLI_CEDULA, CLI_DIRECCION, CLI_FECHA_NACIMIENTO, BIN_NUMERO) VALUES
(1, 'JOSSELYN', 'MACIAS', 1350696397, 'ALTAMIRA FRENTE A LA ESCUELA', '2000-08-20', 1);
INSERT INTO USUARIO (CLI_CODIGO, CLI_NOMBRE, CLI_APELLIDO, CLI_CEDULA, CLI_DIRECCION, CLI_FECHA_NACIMIENTO, BIN_NUMERO) VALUES
(002, 'BRYAN', 'ARTEAGA CEDEÑO', 1379954265, 'CUBA', '17/09/1997', 0003);
INSERT INTO USUARIO (CLI_CODIGO, CLI_NOMBRE, CLI_APELLIDO, CLI_CEDULA, CLI_DIRECCION, CLI_FECHA_NACIMIENTO, BIN_NUMERO) VALUES
(003, 'VICTORIA', 'CEDEÑO MACIAS', 1302587998, 'ALTAGRACIA', '01/12/1998', 0001);
INSERT INTO USUARIO (CLI_CODIGO, CLI_NOMBRE, CLI_APELLIDO, CLI_CEDULA, CLI_DIRECCION, CLI_FECHA_NACIMIENTO, BIN_NUMERO) VALUES
(004, 'MANUEL', 'BAUTISTA ENCINA', 1348762554, 'VIA JARAMIJO', '17/10/2000', 0002);
INSERT INTO USUARIO (CLI_CODIGO, CLI_NOMBRE, CLI_APELLIDO, CLI_CEDULA, CLI_DIRECCION, CLI_FECHA_NACIMIENTO, BIN_NUMERO) VALUES
(005, 'GENARO', 'LOPEZ GENARO', 1356877230, 'VILLAMARINA', '15/07/1995', 0003);
INSERT INTO USUARIO (CLI_CODIGO, CLI_NOMBRE, CLI_APELLIDO, CLI_CEDULA, CLI_DIRECCION, CLI_FECHA_NACIMIENTO, BIN_NUMERO) VALUES
(006, 'ROCIO', 'ALONZO IBARRA', 1378625120, 'AV 4 DE NOVIEMBRE', '26/08/1999', 0005);
INSERT INTO USUARIO (CLI_CODIGO, CLI_NOMBRE, CLI_APELLIDO, CLI_CEDULA, CLI_DIRECCION, CLI_FECHA_NACIMIENTO, BIN_NUMERO) VALUES
(007, 'ROSARIO', 'MARCIAL SALGADO', 1397521200, 'FRENTE AL PARQUE DEL CENTRO', '20/01/2000', 0007);
```

PostgreSQL: Insertar datos

Data Output	Explain	Messages	History				
	cli_codigo integer	cli_nombre character varying(20)	cli_apellido character varying(20)	cli_cedula integer	cli_direccion character varying(30)	cli_fecha_nacimie date	bin_numero integer
1	1	JOSSELYN	MACIAS	1350696397	ALTAMIRA FRENTE A LA ESCUELA	2000-08-20	1
2	2	BRYAN	ARTEAGA CEDEÑO	1379954265	CUBA	1997-09-17	3
3	3	VICTORIA	CEDEÑO MACIAS	1302587998	ALTAGRACIA	1998-12-01	1
4	4	MANUEL	BAUTISTA ENCINA	1348762554	VIA JARAMIJO	2000-10-17	2
5	5	GENARO	LOPEZ GENARO	1356877230	VILLAMARINA	1995-07-15	3
6	6	ROCIO	ALONZO IBARRA	1378625120	AV 4 DE NOVIEMBRE	1999-08-26	5
7	7	ROSARIO	MARCIAL SALGADO	1397521200	FRENTE AL PARQUE DEL CENTRO	2000-01-20	7
8	8	JUAN	PEREZ ZAMORA	1316587423	DIAGONAL A LA FERRETERIA	2001-05-30	4
9	9	LUIS	SOSA INTRIAGO	1379121215	CALLE 38 AV 104	1992-01-01	2
10	10	PEDRO	RUIZ ZAMORA	1358798524	AV 108 CALLE 112 ESQUINA	2000-11-20	4
11	11	LEONEL	MONTOYA CEDEÑO	1310259785	LOS ESTEROS FRENTE AL TIA	1997-02-17	6
12	12	EDUARDO	MOSQUERA RODRIGUEZ	1368751221	VIA SAN MATEO	1989-06-21	1
13	13	PATRICIO	GUITIERREZ RIVERA	1305498575	CIUDADELA UNIVERSITARIA	1990-12-08	7
14	14	MARIO	FREIRE CISNEROS	1305487921	EL PALMAR FRENTE AL TERMINAR	1994-06-16	5
15	15	MILTON	LUCAS CHECA	1316468852	ALTAMIRA	2000-04-28	2
16	16	MARIA	PIGUAVE HOLGUIN	1395246321	BARRIO SAN JOSE	1999-01-10	7
17	17	CECILIA	MOLINA MACIAS	1315568651	CALLE 53 AVENIDA 19	1996-08-25	6
18	18	BETTY	RODRIGUEZ LARCO	1345995532	BARRIO CENTRO	2002-09-07	5

PostgreSQL: Insertar datos

```
/*INSERTAR DATOS DE INSCRIPCION*/
INSERT INTO INSCRIPCION (INC_CODIGO, INC_DESCUENTO, INC_FECHA, ADMI_ID, CLI_CODIGO ) VALUES
(0001,0.00 , '07/10/2020',004, 008);
INSERT INTO INSCRIPCION (INC_CODIGO, INC_DESCUENTO ,INC_FECHA, ADMI_ID, CLI_CODIGO ) VALUES
(0002,0.00, '19/09/2020',004, 010);
INSERT INTO INSCRIPCION (INC_CODIGO , INC_DESCUENTO, INC_FECHA,ADMI_ID, CLI_CODIGO ) VALUES
(0003 ,0.00,'03/07/2020', 001, 001);
INSERT INTO INSCRIPCION (INC_CODIGO , INC_DESCUENTO,INC_FECHA, ADMI_ID, CLI_CODIGO ) VALUES
(0004 ,0.00,'30/06/2020', 001, 003);
INSERT INTO INSCRIPCION (INC_CODIGO , INC_DESCUENTO,INC_FECHA, ADMI_ID, CLI_CODIGO ) VALUES
(0005,0.00,'28/06/2020', 001, 012);
INSERT INTO INSCRIPCION (INC_CODIGO , INC_DESCUENTO,INC_FECHA, ADMI_ID, CLI_CODIGO ) VALUES
(0006 ,0.00,'20/07/2020', 002, 004);
INSERT INTO INSCRIPCION (INC_CODIGO , INC_DESCUENTO,INC_FECHA, ADMI_ID, CLI_CODIGO ) VALUES
(0007 ,0.00,'08/08/2020', 002, 009);
INSERT INTO INSCRIPCION (INC_CODIGO , INC_DESCUENTO,INC_FECHA, ADMI_ID, CLI_CODIGO ) VALUES
(0008 ,0.00,'07/08/2020', 002, 015);
```

PostgreSQL: Insertar datos

Data Output		Explain	Messages	History		
	inc_codigo integer	inc_desci double pr	inc_fecha date	admi_ intege	cli_coc intege	
1	1	0	2020-10-07	4	8	
2	2	0	2020-09-19	4	10	
3	3	0	2020-07-03	1	1	
4	4	0	2020-06-30	1	3	
5	5	0	2020-06-28	1	12	
6	6	0	2020-07-20	2	4	
7	7	0	2020-08-08	2	9	
8	8	0	2020-08-07	2	15	
9	9	0	2020-09-09	3	2	
10	10	0	2020-09-07	3	5	
11	11	0	2020-11-08	5	6	
12	12	0	2020-11-05	5	14	
13	13	0	2020-11-09	5	18	
14	14	0	2020-11-24	6	11	
15	15	0	2020-12-08	6	17	
16	16	0	2021-01-07	7	7	
17	17	0	2020-12-28	7	13	
18	18	0	2021-01-05	1	16	
19	19	0	2020-07-25	2	5	
20	20	0	2020-06-30	5	3	
21	21	0	2020-11-24	3	10	
22	22	0	2020-08-10	4	9	
23	23	0	2020-09-24	7	7	
24	24	0	2020-11-08	1	9	
25	25	0	2021-01-07	6	7	
26	26	0	2020-12-23	2	15	
27	27	0	2021-01-06	7	5	

Link: [Código](#)

PostgreSQL: Insertar datos

```

/*INSERTAR DATOS DE PAGO*/
INSERT INTO PAGO (PAG_CODIGO, PAG_REALIZADO, INC_CODIGO) VALUES
(0001, 40.00, 0001);
INSERT INTO PAGO (PAG_CODIGO, PAG_REALIZADO, INC_CODIGO) VALUES
(0002, 35.00, 0002);
INSERT INTO PAGO (PAG_CODIGO, PAG_REALIZADO, INC_CODIGO) VALUES
(0003, 40.00, 0003);
INSERT INTO PAGO (PAG_CODIGO, PAG_REALIZADO, INC_CODIGO) VALUES
(0004, 36.00, 0004);
INSERT INTO PAGO (PAG_CODIGO, PAG_REALIZADO, INC_CODIGO) VALUES
(0005, 15.00, 0005);
INSERT INTO PAGO (PAG_CODIGO, PAG_REALIZADO, INC_CODIGO) VALUES
(0006, 25.00, 0006);
INSERT INTO PAGO (PAG_CODIGO, PAG_REALIZADO, INC_CODIGO) VALUES
(0007, 00.00, 0007);
INSERT INTO PAGO (PAG_CODIGO, PAG_REALIZADO, INC_CODIGO) VALUES
(0008, 10.00, 0008);
INSERT INTO PAGO (PAG_CODIGO, PAG_REALIZADO, INC_CODIGO) VALUES
(0009, 35.00, 0009);
INSERT INTO PAGO (PAG_CODIGO, PAG_REALIZADO, INC_CODIGO) VALUES
(0010, 27.00, 0010);
INSERT INTO PAGO (PAG_CODIGO, PAG_REALIZADO, INC_CODIGO) VALUES
(0011, 00.00, 0011);

```

Data Output	Explain	Messages	History
	pag_codigo integer	pag_realizado integer	inc_codigo integer
1	1	40	1
2	2	35	2
3	3	40	3
4	4	36	4
5	5	15	5
6	6	25	6
7	7	0	7
8	8	10	8
9	9	35	9
10	10	27	10
11	11	0	11
12	12	0	12
13	13	20	13
14	14	25	14
15	15	10	15
16	16	0	16
17	17	40	17
18	18	40	18
19	19	30	19
20	20	0	20
21	21	10	21
22	22	15	22
23	23	40	23
24	24	40	24
25	25	0	25
26	26	0	26
27	27	25	27

PostgreSQL: Insertar datos

```
/*INSERTAR DATOS DE GANADORES*/
INSERT INTO GANADORES (GAN_CODIGO, GAN_CANTIDAD_GANADOS, BIN_NUMERO, CLI_CODIGO)
VALUES (001, 3 ,001 , 005 );
INSERT INTO GANADORES (GAN_CODIGO, GAN_CANTIDAD_GANADOS, BIN_NUMERO, CLI_CODIGO)
VALUES (002, 0 ,004 , 010 );
INSERT INTO GANADORES (GAN_CODIGO, GAN_CANTIDAD_GANADOS, BIN_NUMERO, CLI_CODIGO)
VALUES (003, 2 ,006 , 016 );
INSERT INTO GANADORES (GAN_CODIGO, GAN_CANTIDAD_GANADOS, BIN_NUMERO, CLI_CODIGO)
VALUES (004, 8 ,005 , 014 );
INSERT INTO GANADORES (GAN_CODIGO, GAN_CANTIDAD_GANADOS, BIN_NUMERO, CLI_CODIGO)
VALUES (005, 0 ,007 , 015 );
INSERT INTO GANADORES (GAN_CODIGO, GAN_CANTIDAD_GANADOS, BIN_NUMERO, CLI_CODIGO)
VALUES (006, 3 , 002, 009 );
INSERT INTO GANADORES (GAN_CODIGO, GAN_CANTIDAD_GANADOS, BIN_NUMERO, CLI_CODIGO)
VALUES (007, 2 ,002 , 012 );
INSERT INTO GANADORES (GAN_CODIGO, GAN_CANTIDAD_GANADOS, BIN_NUMERO, CLI_CODIGO)
VALUES (008, 0 , 003, 001);
INSERT INTO GANADORES (GAN_CODIGO, GAN_CANTIDAD_GANADOS, BIN_NUMERO, CLI_CODIGO)
VALUES (009, 7 ,005 , 002 );
```

PostgreSQL: Insertar datos

Data Output	Explain	Messages	History		
	gan_codigo integer	gan_cantidad_ganados integer	bin_numero integer	cli_codigo integer	
1	1	3	1	5	
2	2	0	4	10	
3	3	2	6	16	
4	4	8	5	14	
5	5	0	7	15	
6	6	3	2	9	
7	7	2	2	12	
8	8	0	3	1	
9	9	7	5	2	
10	10	4	7	4	
11	11	3	1	7	
12	12	2	4	7	
13	13	0	2	11	
14	14	1	6	3	
15	15	5	5	6	
16	16	3	7	13	
17	17	0	3	17	
18	18	0	1	18	

Link: [Código](#)

PostgreSQL: Consultas

- ¿Cuáles son los usuarios que llevan más de dos meses inscritos?

```
SELECT CLI_NOMBRE, COUNT(CLI_NOMBRE) AS CANTIDAD
FROM USUARIO
INNER JOIN PUBLIC.INScripcion ON USUARIO.CLI_CODIGO = INSCRIPCION.CLI_CODIGO
GROUP BY CLI_NOMBRE
HAVING COUNT (*) > 2
```

	Data Output	Explain	Messages	History
	cli_nombre character varying(20)	cantidad bigint		
1	ROSARIO	3		
2	GENARO	3		
3	LUIS	3		

PostgreSQL: Consultas

- ¿Cuál es la cantidad de bingos que ha ganado cada jugador?

```
SELECT
    USUARIO.CLI_NOMBRE,
    USUARIO.CLI_APELLIDO,
    BINGO.BIN_DATE,
    GANADORES.GAN_CANTIDAD_GANADOS
FROM    PUBLIC.USUARIO
INNER JOIN PUBLIC.GANADORES ON USUARIO.CLI_CODIGO = GANADORES.CLI_CODIGO
INNER JOIN PUBLIC.BINGO ON    BINGO.BIN_NUMERO = USUARIO.BIN_NUMERO
WHERE
    GANADORES.GAN_CANTIDAD_GANADOS>0
```

PostgreSQL: Consultas

Data Output	Explain	Messages	History	
	cli_nombre character varying(20)	cli_apellido character varying(20)	bin_date date	gan_cantid integer
1	GENARO	LOPEZ GENARO	2020-09-10	3
2	MARIA	PIGUAVE HOLGUIN	2021-01-10	2
3	MARIO	FREIRE CISNEROS	2020-11-10	8
4	LUIS	SOSA INTRIAGO	2020-08-10	3
5	EDUARDO	MOSQUERA RODRIGUEZ	2020-07-10	2
6	BRYAN	ARTEAGA CEDEÑO	2020-09-10	7
7	MANUEL	BAUTISTA ENCINA	2020-08-10	4
8	ROSARIO	MARCIAL SALGADO	2021-01-10	3
9	ROSARIO	MARCIAL SALGADO	2021-01-10	2
10	VICTORIA	CEDEÑO MACIAS	2020-07-10	1
11	ROCIO	ALONZO IBARRA	2020-11-10	5
12	PATRICIO	GUITIERREZ RIVERA	2021-01-10	3

PostgreSQL: Consultas

- Mostrar la fecha del primer bingo y la fecha del ultimo bingo con el total de ingresos

```
SELECT
MIN(BIN_DATE) AS INICIO,
MAX(BIN_DATE) AS FIN,
SUM( 40.00 - INSCRIPCION.INC_DESCUENTO) AS TOTAL
FROM PUBLIC.USUARIO
INNER JOIN PUBLIC.INSCRIPCION ON USUARIO.CLI_CODIGO = INSCRIPCION.CLI_CODIGO
INNER JOIN PUBLIC.PAGO ON INSCRIPCION.INC_CODIGO = PAGO.INC_CODIGO
INNER JOIN PUBLIC.BINGO ON BINGO.BIN_NUMERO = USUARIO.BIN_NUMERO
```

	Data Output	Explain	Messages	History
	inicio date	fin date	total double prec	
1	2020-07-10	2021-01-10	1036	

PostgreSQL: Consultas

- Mostrar la fecha del primer bingo y la fecha del ultimo bingo con el total de ingresos

```
SELECT
MIN(BIN_DATE) AS INICIO,
MAX(BIN_DATE) AS FIN,
SUM( 40.00 - INSCRIPCION.INC_DESCUENTO) AS TOTAL
FROM PUBLIC.USUARIO
INNER JOIN PUBLIC.INSCRIPCION ON USUARIO.CLI_CODIGO = INSCRIPCION.CLI_CODIGO
INNER JOIN PUBLIC.PAGO ON INSCRIPCION.INC_CODIGO = PAGO.INC_CODIGO
INNER JOIN PUBLIC.BINGO ON BINGO.BIN_NUMERO = USUARIO.BIN_NUMERO
```

	Data Output	Explain	Messages	History
	inicio date	fin date	total double prec	
1	2020-07-10	2021-01-10	1036	

PostgreSQL: Consultas

- Mostrar la fecha del primer bingo y la fecha del ultimo bingo con el total de ingresos

```
SELECT
MIN(BIN_DATE) AS INICIO,
MAX(BIN_DATE) AS FIN,
SUM( 40.00 - INSCRIPCION.INC_DESCUENTO) AS TOTAL
FROM PUBLIC.USUARIO
INNER JOIN PUBLIC.INSCRIPCION ON USUARIO.CLI_CODIGO = INSCRIPCION.CLI_CODIGO
INNER JOIN PUBLIC.PAGO ON INSCRIPCION.INC_CODIGO = PAGO.INC_CODIGO
INNER JOIN PUBLIC.BINGO ON BINGO.BIN_NUMERO = USUARIO.BIN_NUMERO
```

	Data Output	Explain	Messages	History
	inicio date	fin date	total double prec	
1	2020-07-10	2021-01-10	1036	

PostgreSQL: Consultas

- ¿Lista de las personas que tienen pagos atrasados?

```
SELECT
  USUARIO.CLI_NOMBRE,
  USUARIO.CLI_APELLIDO,
  INSCRIPCION.INC_DESCUENTO,
  40.00 - INSCRIPCION.INC_DESCUENTO AS TOTAL,
  PAGO.PAG_REALIZADO,
  (40.00 - INSCRIPCION.INC_DESCUENTO) - PAGO.PAG_REALIZADO AS DEUDA
FROM
  PUBLIC.INSCRIPCION
  INNER JOIN PUBLIC.PAGO ON INSCRIPCION.INC_CODIGO = PAGO.INC_CODIGO
  INNER JOIN PUBLIC.USUARIO ON USUARIO.CLI_CODIGO = INSCRIPCION.CLI_CODIGO
WHERE
  ((40.00 - INSCRIPCION.INC_DESCUENTO) - PAGO.PAG_REALIZADO) > 0
GROUP BY INSCRIPCION.INC_CODIGO, PAGO.PAG_REALIZADO, USUARIO.CLI_NOMBRE, USUARIO.CLI_APELLIDO
```


PostgreSQL: Consultas

- ¿Lista de las personas que tienen pagos atrasados?

	cli_nombre character varying(20)	cli_apellido character varying(20)	inc_descuento double precision	total double precision	pag_realizado integer	deuda double precision
1	GENARO	LOPEZ GENARO	4	36	30	6
2	PEDRO	RUIZ ZAMORA	4	36	30	6
3	ROCIO	ALONZO IBARRA	0	40	0	40
4	VICTORIA	CEDEÑO MACIAS	4	36	0	36
5	LUIS	SOSA INTRIAGO	0	40	15	25
6	MILTON	LUCAS CHECA	0	40	10	30
7	CECILIA	MOLINA MACIAS	0	40	10	30
8	MANUEL	BAUTISTA ENCINA	4	36	25	11
9	EDUARDO	MOSQUERA RODRIGUEZ	4	36	15	21
10	ROSARIO	MARCIAL SALGADO	0	40	0	40
11	MARIO	FREIRE CISNEROS	0	40	0	40
12	BRYAN	ARTEAGA CEDEÑO	0	40	35	5
13	GENARO	LOPEZ GENARO	0	40	27	13
14	LEONEL	MONTOYA CEDEÑO	4	36	25	11
15	GENARO	LOPEZ GENARO	0	40	25	15
16	BETTY	RODRIGUEZ LARCO	0	40	20	20
17	MILTON	LUCAS CHECA	4	36	0	36
18	ROSARIO	MARCIAL SALGADO	0	40	0	40
19	PEDRO	RUIZ ZAMORA	4	36	10	26
20	LUIS	SOSA INTRIAGO	0	40	0	40

PostgreSQL: Consultas

- El administrador debe generar un descuento de 10% a las personas inscritas 10 días antes del 10 de cada mes en el que comienza el bingo teniendo en cuenta que la inscripción cuesta \$40.

```
/*DESCUENTO DEL BINGO 1*/  
SELECT * FROM INSCRIPCION;  
UPDATE INSCRIPCION  
SET INC_DESCUENTO = 4.00  
WHERE ( INC_FECHA >'10/06/2020'  
AND INC_FECHA < '01/07/2020');
```

```
/*DESCUENTO DEL BINGO 4*/  
SELECT * FROM INSCRIPCION;  
UPDATE INSCRIPCION  
SET INC_DESCUENTO = 4.00  
WHERE ( INC_FECHA >'10/09/2020'  
AND INC_FECHA < '01/10/2020');
```

```
/*DESCUENTO DEL BINGO 2*/  
SELECT * FROM INSCRIPCION;  
UPDATE INSCRIPCION  
SET INC_DESCUENTO = 4.00  
WHERE ( INC_FECHA >'10/07/2020'  
AND INC_FECHA < '01/08/2020');
```

```
/*DESCUENTO DEL BINGO 5*/  
SELECT * FROM INSCRIPCION;  
UPDATE INSCRIPCION  
SET INC_DESCUENTO = 4.00  
WHERE ( INC_FECHA >'10/10/2020'  
AND INC_FECHA < '01/11/2020');
```

```
/*DESCUENTO DEL BINGO 7*/  
SELECT * FROM INSCRIPCION;  
UPDATE INSCRIPCION  
SET INC_DESCUENTO = 4.00  
WHERE ( INC_FECHA >'10/12/2020'  
AND INC_FECHA < '01/01/2021');
```

```
/*DESCUENTO DEL BINGO 3*/  
SELECT * FROM INSCRIPCION;  
UPDATE INSCRIPCION  
SET INC_DESCUENTO = 4.00  
WHERE ( INC_FECHA >'10/08/2020'  
AND INC_FECHA < '01/09/2020');
```

```
/*DESCUENTO DEL BINGO 6*/  
SELECT * FROM INSCRIPCION;  
UPDATE INSCRIPCION  
SET INC_DESCUENTO = 4.00  
WHERE ( INC_FECHA >'10/11/2020'  
AND INC_FECHA < '01/12/2020');
```

Link: [Código](#)

PostgreSQL: Consultas

	inc_codigo integer	inc_descuento double precision	inc_fecha date	admi_id integer	cli_codigo integer
1	1	0	2020-10-	4	8
2	3	0	2020-07-	1	1
3	7	0	2020-08-	2	9
4	8	0	2020-08-	2	15
5	9	0	2020-09-	3	2
6	10	0	2020-09-	3	5
7	11	0	2020-11-	5	6
8	12	0	2020-11-	5	14
9	13	0	2020-11-	5	18
10	15	0	2020-12-	6	17
11	16	0	2021-01-	7	7
12	18	0	2021-01-	1	16
13	22	0	2020-08-	4	9
14	24	0	2020-11-	1	9
15	25	0	2021-01-	6	7
16	27	0	2021-01-	7	5
17	4	4	2020-06-	1	3
18	5	4	2020-06-	1	12
19	20	4	2020-06-	5	3
20	6	4	2020-07-	2	4
21	19	4	2020-07-	2	5
22	2	4	2020-09-	4	10
23	23	4	2020-09-	7	7
24	14	4	2020-11-	6	11
25	21	4	2020-11-	3	10
26	17	4	2020-12-	7	13
27	26	4	2020-12-	2	15

Link: [Código](#)

PostgreSQL: Consultas

- **Mostrar una lista con los usuarios que tiene a cargo cada administrador en cada uno de los bingos**

```
SELECT
    BINGO.BIN_NUMERO,
    BINGO.BIN_DATE,
    ADMINISTRADOR.ADMI_NOMBRE,
    USUARIO.CLI_NOMBRE,
    USUARIO.CLI_APELLIDO,
    INSCRIPCION.INC_FECHA,
    INSCRIPCION.INC_DESCUENTO
FROM    PUBLIC.INSCRIPCION
INNER JOIN PUBLIC.ADMINISTRADOR ON ADMINISTRADOR.ADMI_ID = INSCRIPCION.ADMI_ID
INNER JOIN PUBLIC.BINGO ON ADMINISTRADOR.ADMI_ID = BINGO.ADMI_ID
INNER JOIN PUBLIC.PAGO ON INSCRIPCION.INC_CODIGO = PAGO.INC_CODIGO
INNER JOIN PUBLIC.USUARIO ON USUARIO.CLI_CODIGO = INSCRIPCION.CLI_CODIGO
WHERE ADMINISTRADOR.ADMI_NOMBRE = 'STEFANY'
```

PostgreSQL: Consultas

Data Output	Explain	Messages	History				
	bin_numero integer	bin_date date	admi_nombre character varying	cli_nombre character varying(20)	cli_apellido character varying(20)	inc_fecha date	inc_descuento double precision
1	1	2020-07-10	STEFANY	JOSSELYN	MACIAS	2020-07-03	0
2	1	2020-07-10	STEFANY	VICTORIA	CEDEÑO MACIAS	2020-06-30	4
3	1	2020-07-10	STEFANY	EDUARDO	MOSQUERA RODRIGUEZ	2020-06-28	4
4	1	2020-07-10	STEFANY	MARIA	PIGUAVE HOLGUIN	2021-01-05	0
5	1	2020-07-10	STEFANY	LUIS	SOSA INTRIAGO	2020-11-08	0

Conclusiones

- Se realizó el modelo conceptual del Sistema de Bingos tomando en cuenta cada uno de los campos utilizados, manejando el software PgModeler.
- Se mostró la creación de cada una de las tablas con su respectiva inserción de datos. Utilizando el software PostgreSQL 9.5, el cual trabaja con la interfaz del PgAdmin III
- Así mismo se pudo efectuar algunas de las consultas solicitadas por el sistema realizado

Gracias