

# INFORME MES SEPTIEMBRE

## SISTEMAS 2016

*Detalle de Tablas del sistema de remuneración, controlador de  
framework para conexión a la bd.*

# TABLA DE CONTENIDO

## Contenido

Información de Tecnología de Bases de Datos	1
MYSQL	1
phpmyadmin	1
Mysql workbench	2
postgresql	3
Variables y Secuencias	4
Tipos numéricos en PostgreSQL	4
Tipos de datos monetarios (moneda) en PostgreSQL	4
Tipos de datos carácter en PostgreSQL	5
Tipos de datos binarios en PostgreSQL	5
Tipos de datos Fecha/Hora en PostgreSQL	5
Tipos de datos geométricos en PostgreSQL	6
Tipos de datos de direcciones de red en PostgreSQL	6
Secuencia	6
Tablas del Sistema de Remuneración	7
Previred	7
UF	7
Indicadores Previsionales PreviRed Septiembre 2016	10
Empleado	10
Impuesto	11
Liquidación	11
Controlador	12
Framework CODEIGNITER	12
Conexión php ajax	12
Ejemplo consulta de comprobación simple	13
Crear Depósito convenido Con js	13
Envío:	13
Recepción:	13
Recuperación de datos previsionales desde el modelo de CODEIGNITER con Active Record	14

# BASE DE DATOS

## Información de Tecnología de Bases de Datos

### MYSQL



Es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporation y está considerada como la base datos open source más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web. Existen varias formas de gestionar mysql. Una es a través del panel de phpmyadmin web, el cual se ingresa por una url en el servidor y accede con un usuario y contraseña con rol en la bd. Y la otra puede ser a través de un programa gestor instalable como workbench, el cual entrega un entorno visual, configurable y adaptable por ventanas y entrega todas las prestaciones para editar a clics o por consultas sql.

### PHPMYADMIN

Es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 72 idiomas.

Bienvenido a phpMyAdmin

Idioma - Language  
Español - Spanish

Iniciar sesión

Usuario:

Contraseña:

Continuar

Las cookies deben estar activadas.

phpMyAdmin

borquez

Crear tabla

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
categoria	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	latin1_swedish_ci	16.0 KB	-
clientes	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	latin1_swedish_ci	16.0 KB	-
documentos	Examinar Estructura Buscar Insertar Vaciar Eliminar	15	InnoDB	latin1_swedish_ci	16.0 KB	-
log_productos	Examinar Estructura Buscar Insertar Vaciar Eliminar	5	InnoDB	latin1_swedish_ci	16.0 KB	-
medidas	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	latin1_swedish_ci	16.0 KB	-
pagos	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	latin1_swedish_ci	16.0 KB	-
principal_documentos	Examinar Estructura Buscar Insertar Vaciar Eliminar	6	InnoDB	latin1_swedish_ci	16.0 KB	-
producto	Examinar Estructura Buscar Insertar Vaciar Eliminar	283	InnoDB	latin1_swedish_ci	64.0 KB	-
sector	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	latin1_swedish_ci	16.0 KB	-
usuario	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	latin1_swedish_ci	16.0 KB	-
10 tablas	Número de filas	329	InnoDB	latin1_swedish_ci	208.0 KB	0 B

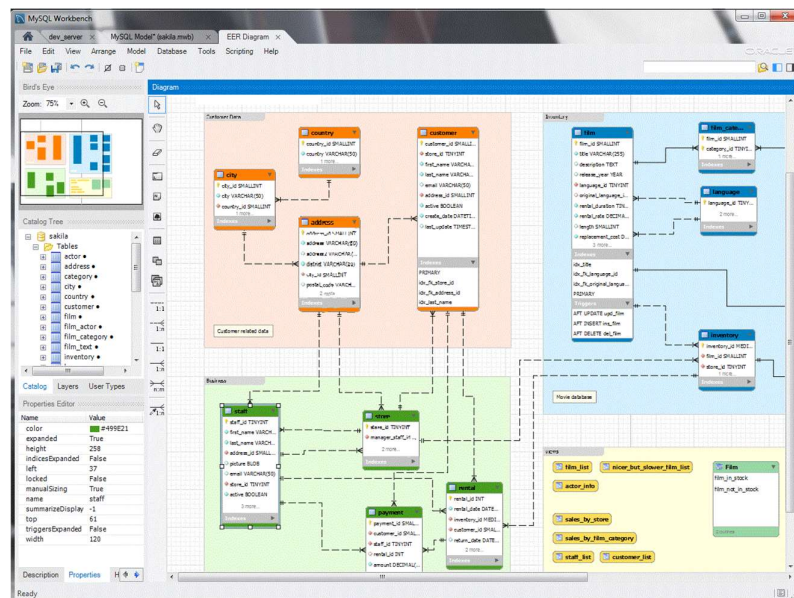
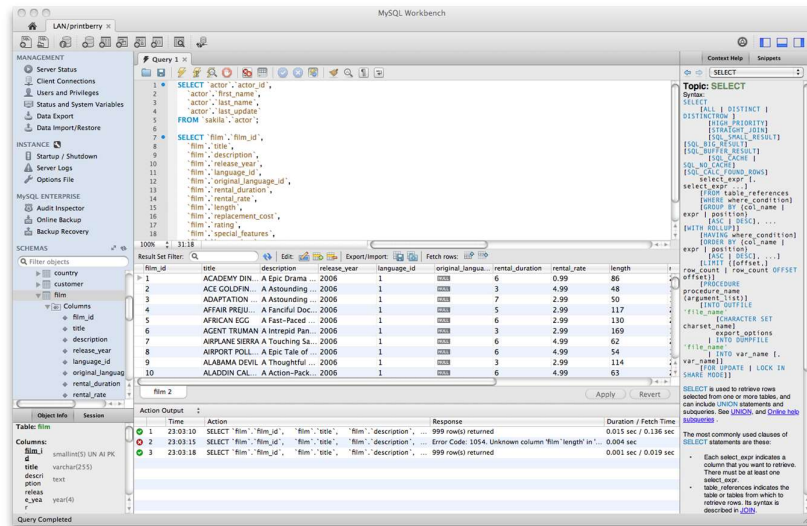
Crear nueva tabla en la base de datos borquez

Nombre:  Número de columnas:

# BASE DE DATOS

## MYSQL WORKBENCH

Es una herramienta visual de diseño de bases de datos que integra desarrollo de software, Administración de bases de datos, diseño de bases de datos, creación y mantenimiento para el sistema de base de datos MySQL. Es el sucesor de DB Designer 4 de fabFORCE.net, y reemplaza el anterior conjunto de software, MySQL GUI Tools Bundle. Permite el diseño conceptual de MER en diferentes formas normales.



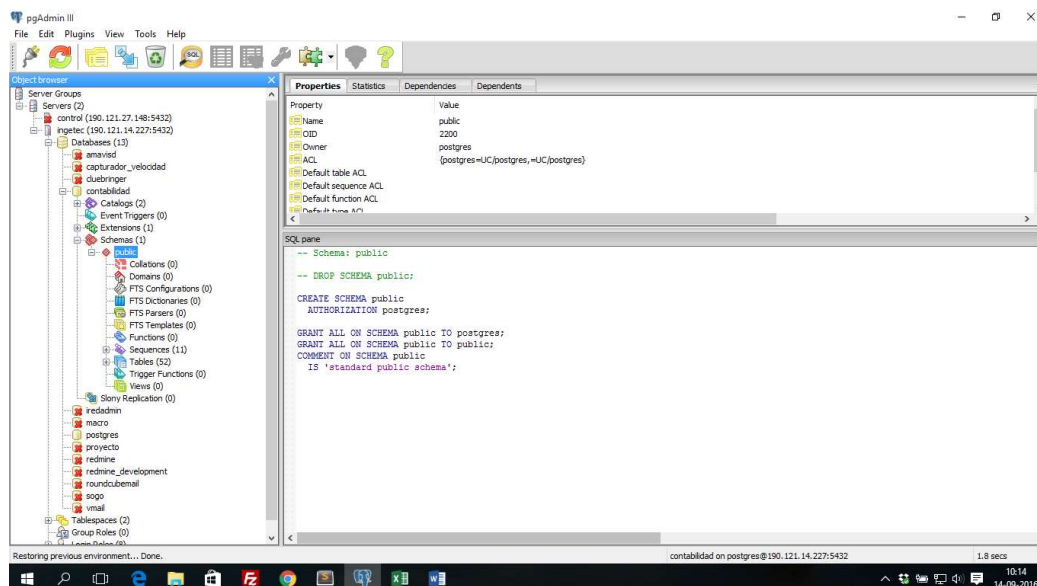
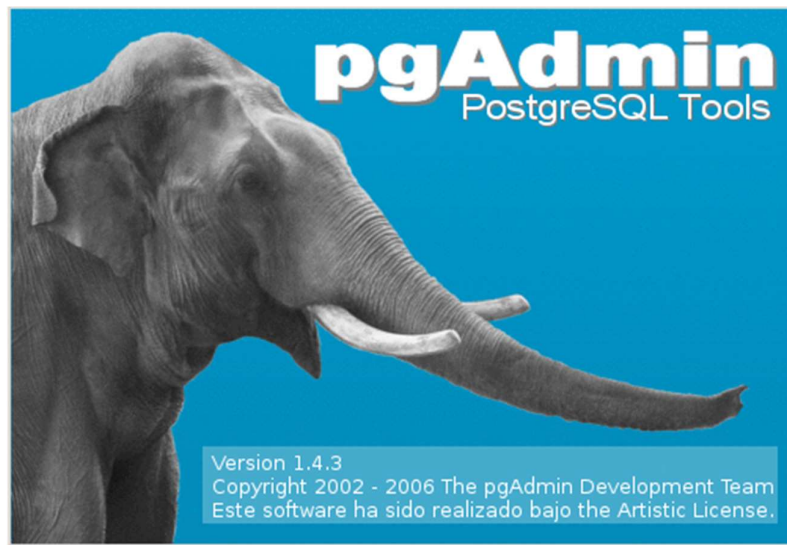
# BASE DE DATOS



## POSTGRESQL

Es un Sistema de gestión de bases de datos relacional orientado a objetos y libre, publicado bajo la licencia PostgreSQL1, similar a la BSD o la MIT.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre o apoyada por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group). Esta base de datos se puede gestionar con diferentes versiones de PGADMIN.



# BASE DE DATOS

## Variables y Secuencias

### TIPOS NUMÉRICOS EN POSTGRESQL

Nombre	Tamaño	Descripción	Rango
smallint	2 bytes	Entero de rango pequeño	De -32768 a +32767
integer	4 bytes	Selección habitual para tipos enteros	De -2147483648 a +2147483647
bigint	8 bytes	Entero de rango largo	De -9223372036854775808 a 9223372036854775807
decimal	variable	Precisión especificada por el usuario, exacto	Sin límite
numeric	variable	Precisión especificada por el usuario, exacto	Sin límite
real	4 bytes	Variable/precisión, inexacto	6 dígitos decimales de precisión
double precision	8 bytes	Variable/precisión, inexacto	15 dígitos decimales de precisión
serial	4 bytes	Autoincremento simple	De 1 a 2147483647
bigserial	8 bytes	Autoincremento largo	De 1 a 9223372036854775807

### TIPOS DE DATOS MONETARIOS (MONEDA) EN POSTGRESQL

Nombre	Tamaño	Descripción	Rango
money	4 bytes	Moneda	De -21474836.48 a +21474836.47

# BASE DE DATOS

## TIPOS DE DATOS CARÁCTER EN POSTGRESQL

Nombre	Descripción
character varying( <i>n</i> ), varchar( <i>n</i> )	De longitud variable, con límite
character( <i>n</i> ), char( <i>n</i> )	De longitud fija
text	De longitud variable, ilimitado

## TIPOS DE DATOS BINARIOS EN POSTGRESQL

Nombre	Tamaño	Descripción
bytea	4 bytes además de la cadena binaria actual	Cadena binaria de longitud variable

## TIPOS DE DATOS FECHA/HORA EN POSTGRESQL

Nombre	Tamaño	Descripción	Valor bajo	Valor alto	Resolución
timestamp [ ( <i>p</i> ) ] [ sin zona horaria ]	8 bytes	Fecha y hora	4713 BC	5874897 AD	1 microsegundo / 14 dígitos
timestamp [ ( <i>p</i> ) ] con zona horaria	8 bytes	Fecha y hora con zona horaria	4713 BC	5874897 AD	1 microsegundos / 14 dígitos
interval [ ( <i>p</i> ) ]	12 bytes	Intervalo de hora	-178000000 años	178000000 años	1 microsegundo
date	4 bytes	Sólo fecha	4713 BC	32767 AD	1 día
time [ ( <i>p</i> ) ] [ sin zona horaria ]	8 bytes	Sólo hora del día	00:00:00.00	23:59:59.99	1 microsegundo
time [ ( <i>p</i> ) ] con zona horaria	12 bytes	Horas del día con zona horaria	00:00:00.00+12	23:59:59.99- 12	1 microsegundo

# BASE DE DATOS

## TIPOS DE DATOS GEOMÉTRICOS EN POSTGRESQL

Nombre	Tamaño	Representación	Descripción
point	16 bytes	Punto del plano	(x,y)
line	32 bytes	Línea infinita en el plano	((x1,y1),(x2,y2))
lseg	32 bytes	Segmento de línea en el plano	((x1,y1),(x2,y2))
box	32 bytes	Rectángulo en el plano	((x1,y1),(x2,y2))
path	16+16n bytes	Trazado geométrico cerrado en el plano	((x1,y1),...)
path	16+16n bytes	Trazado geométrico abierto en el plano	[(x1,y1),...]
polygon	40+16n bytes	Plígono (similar a trazado cerrado)	((x1,y1),...)
circle	24 bytes	Círculo	<(x,y),r> (centro y radio)

## TIPOS DE DATOS DE DIRECCIONES DE RED EN POSTGRESQL

Nombre	Tamaño	Descripción
cidr	12 ó 24 bytes	Redes IPv4 ó IPv6
inet	12 ó 24 bytes	Hosts y redes IPv4 ó IPv6
macaddr	6 bytes	Dirección MAC

## SECUENCIA

En mysql se le puede dar la propiedad de autoincrementable a un campo de clave única, y en postgres se crea una secuencia la que se le asigna a un campo.

```
-- Sequence: increment_componente
-- DROP SEQUENCE increment_componente;

CREATE SEQUENCE increment_componente
  INCREMENT 1
  MINVALUE 1
  MAXVALUE 9223372036854775807
  START 15
  CACHE 1;
ALTER TABLE increment_componente
  OWNER TO postgres;

ADD COLUMN id numeric;
ALTER COLUMN id SET DEFAULT nextval('in_lcompra'::regclass);
```



## Tablas del Sistema de Remuneración

PREVIRED

UF

```
CREATE TABLE uf_remuneracion
(
    uf1 real,
    mes text,
    year integer
);
WITH (
    OIDS=FALSE
);
ALTER TABLE uf_remuneracion
    OWNER TO postgres;
```

- UTM

```
CREATE TABLE utm_remuneracion
(
    valor real,
    mes text,
    year integer
);
WITH (
    OIDS=FALSE
);
ALTER TABLE utm_remuneracion
    OWNER TO postgres;
```

- RENTA TOPES IMPONIBLE

```
CREATE TABLE rti_remuneracion
(
    rti1 integer,
    rti2 integer,
    rti3 integer,
    mes text,
    year integer
);
WITH (
    OIDS=FALSE
);
ALTER TABLE rti_remuneracion
    OWNER TO postgres;
```

# BASE DE DATOS

- RENTAS MINIMAS IMPONIBLES

```
CREATE TABLE rmi_remuneracion
(
    rmi1 integer,
    rmi2 integer,
    rmi3 integer,
    rmi4 integer,
    mes text,
    year integer
)
WITH (
    OIDS=FALSE
);
ALTER TABLE rmi_remuneracion
    OWNER TO postgres;
```

- AHORRO PREVISIONAL VOLUNTARIO

```
CREATE TABLE apv_remuneracion
(
    apv1 integer,
    apv2 integer,
    mes text,
    year integer
)
WITH (
    OIDS=FALSE
);
ALTER TABLE apv_remuneracion
    OWNER TO postgres;
```

- DEPOSITO CONVENIDO

```
CREATE TABLE dc_remuneracion
(
    dc1 integer,
    mes text,
    year integer
)
WITH (
    OIDS=FALSE
);
ALTER TABLE dc_remuneracion
    OWNER TO postgres;
```

# BASE DE DATOS

- SEGURO DE CESANTIA (AFC)

```
CREATE TABLE sc_remuneracion
(
    sce1 real,
    sce2 real,
    sce3 real,
    sct1 real,
    sct2 real,
    sct3 real,
    mes text,
    year integer
);
WITH (
    OIDS=FALSE
);
ALTER TABLE sc_remuneracion
    OWNER TO postgres;
```

- TASA COTIZACION OBLIGATORIA

```
CREATE TABLE afp_remuneracion
(
    afp1 text,
    tafpd1 text,
    sisd1 text,
    tafp11 text,
    mes text,
    year integer
);
WITH (
    OIDS=FALSE
);
ALTER TABLE afp_remuneracion
    OWNER TO postgres;
```

- ASIGNACION FAMILIAR

```
CREATE TABLE af_remuneracion
(
    monto1 integer,
    monto2 integer,
    monto3 integer,
    monto4 integer,
    mes text,
    year integer
);
WITH (
    OIDS=FALSE
);
ALTER TABLE af_remuneracion
    OWNER TO postgres;
```

# BASE DE DATOS

## INDICADORES PREVISIONALES PREVIED SEPTIEMBRE 2016

VALOR UF	
al 30 de Septiembre 2016	\$ 26.224,30
al 31 de Agosto 2016	\$ 26.209,10

VALOR	UTM	UTA
Septiembre 2016	\$ 45.999	\$ 551.988

RENTAS TOPES IMPONIBLES	
Para afiliados a una AFP (74,3 UF):	\$ 1.948.465
Para afiliados al IPS (ex INP) (60 UF):	\$ 1.572.546
Para Seguro de Cesantía (111,4 UF):	\$ 2.921.387

RENTAS MÍNIMAS IMPONIBLES	
Trab. Dependientes e Independientes:	\$ 257.500
Menores de 18 y Mayores de 65:	\$ 192.230
Trabajadores de Casa Particular:	\$ 257.500
Para fines no remuneracionales:	\$ 166.103

AHORRO PREVISIONAL VOLUNTARIO (APV)	
Tope Mensual (50 UF):	\$ 1.311.215
Tope Anual (600 UF):	\$ 15.734.580

DEPÓSITO CONVENIDO	
Tope Anual (900 UF):	\$ 23.601.870

SEGURO DE CESANTÍA (AFC)		
TIPO CONTRATO (*)	FINANCIAMIENTO	
	EMPLEADOR	TRABAJADOR
Contrato Plazo Indefinido	2,4% R.I.	0,6% R.I.
Contrato Plazo Fijo	3,0% R.I.	-
Contrato Plazo Indefinido 11 años o más	0,8% R.I.	-

(\*\*)

AFP	TASA COTIZACIÓN OBLIGATORIO AFP		
	TASA AFP TRABAJADORES		
	DEPENDIENTES		INDEPENDIENTES
	TASA AFP	SIS (1) (2)	TASA AFP (3)
Capital	11,44%	1,41%	12,85%
Cuprum	11,48%	1,41%	12,89%
Habitat	11,27%	1,41%	12,68%
PlanVital	10,41%	1,41%	11,82%
ProVida	11,54%	1,41%	12,95%
Modelo	10,77%	1,41%	12,18%

1) Esta cotización es de cargo del Empleador.

2) SIS no aplica para trabajador dependiente pensionado.

3) Esta tasa incluye el SIS, que es de cargo del trabajador.

TRAMO	ASIGNACIÓN FAMILIAR	
	MONTO	REQUISITO DE RENTA
A	\$ 10.577	Renta < ó = \$ 270.196
B	\$ 6.491	Renta > \$ 270.196 <= \$ 394.651
C	\$ 2.052	Renta > \$ 394.651 <= \$ 615.521
D	-	Renta > \$ 615.521

COTIZACIÓN PARA TRABAJOS PESADOS (*)			
CALIFICACIÓN		FINANCIAMIENTO	
PUESTO DE TRABAJO		EMPLEADOR	TRABAJADOR
Trabajo pesado		4%	2% R.I.
Trabajo menos pesado		2%	1% R.I.

(\*) Ver listador en Superintendencia de Pensiones, Haz Click aquí

## EMPLEADO

CREATE TABLE empleado\_remuneracion

```
(
  rut character varying(25) NOT NULL,
  sr text,
  apellido_paterno character varying(25),
  apellido_materno character varying(30),
  nombres character varying(50),
  fecha_nacimiento date,
  edad integer,
  nacionalidad character varying(25),
  estado_civil character varying(20),
  sexo character varying(10),
  empresa character varying(25),
  estado text,
  CONSTRAINT empleado_remuneracion_pkey PRIMARY KEY (rut)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE empleado_remuneracion
OWNER TO postgres;
```

Edit Data - ingetec (190.121.14.227:5432) - contabilidad - empleado\_remuneracion

rut	sr	apellido_paterno	apellido_materno	nombres	fecha_nacimiento	edad	nacionalidad	estado_civil	sexo	empresa	estado
PK character varying(25)	text	character varying(25)	character varying(30)	character varying(50)	date	integer	character varying(25)	character varying(20)	character varying(10)	character varying(25)	text
1 16.004.422-2	Sr.	Moche	Soto	Rene Ernesto	1981-07-29	35	Chilena	Soltero	Hombre	76.106.452-1	activo
2 16.507.327-K	Sr.	Bravo	Saldivia	Carlos Anibal	1978-07-14	38	Chilena	Casado	Hombre	76.396.038-2	activo
3 16.967.030-5	Sr.	Marin	Poblete	Christian Rodrigo	1979-12-31	36	Chilena	Soltero	Hombre	76.106.452-1	activo
4 16.058.518-1	Sr.	Irribarria	Almaraz	Victor Manuel	1981-09-10	34	Chilena	Soltero	Hombre	76.106.452-1	activo
5 16.591.955-8	Sr.	Arcois	Sanchez	Rodrigo Andres	1983-10-10	33	Chilena	Soltero	Hombre	76.396.038-2	activo
6 16.117.690-7	Sr.	Gonzalez	Gonzalez	Christopher William	1985-10-24	31	Chilena	Soltero	Hombre	76.106.452-1	activo
7 16.217.079-4	Sr.	Vidal	Sobrero	Christian Marcos	1971-05-10	45	Chilena	Casado	Hombre	76.396.038-2	activo
8 16.497.263-1	Sr.	Espinoza	Rojas	Cesar Antonio	1987-05-07	29	Chilena	Soltero	Hombre	76.396.038-2	activo
9 16.596.411-K	Sr.	Morgado	Piña	Javier Ramon	1987-05-27	29	Chilena	Casado	Hombre	76.106.452-1	activo
10 16.782.039-6	Sr.	Olivares	Saldivia	Marcela Raquel	1987-08-24	28	Chilena	Soltero	Mujer	76.106.452-1	activo
11 17.128.042-8	Sr.	Sandoval	Poblete	Eusebio Mauricio	1988-09-08	27	Chilena	Soltero	Hombre	76.396.038-2	activo
12 17.346.704-4	Sr.	San Martin	Moreno	Juan Pablo	1990-03-27	26	Chilena	Soltero	Hombre	76.106.452-1	activo
13 17.742.322-K	Sr.	Acum	Jaramillo	Dario	1991-03-01	25	Chilena	Soltero	Hombre	76.106.452-1	activo
14 17.749.164-0	Sr.	Gonzalez	Diaz	Diego Valentin	1991-02-19	25	Chilena	Soltero	Hombre	76.396.038-2	activo
15 17.749.422-4	Sr.	Reyes	Hernandez	Sara Ester	0119-03-27	25	Chilena	Soltero	Mujer	76.396.038-2	activo
16 16.155.262-K	Sr.	Ortiz	Sanhueza	Makarena Estephani	1990-11-12	25	Chilena	Soltero	Hombre	76.106.452-1	activo
17 16.452.054-0	Sr.	Medina	Soto	Jennifer Alejandra	1988-06-30	27	Chilena	Soltero	Mujer	76.396.038-2	activo
18 16.856.026-2	Sr.	Jaque	Aguilera	Alexis Eduardo	1995-02-04	21	Chilena	Soltero	Hombre	76.396.038-2	activo



# BASE DE DATOS

IMPUESTO  
CREATE TABLE impuesto\_remuneracion

```
(
  desde text,
  hasta text,
  factor text,
  descuento text,
  mes text,
  year text,
  tramo text
);
WITH (
  OIDS=FALSE
);
ALTER TABLE impuesto_remuneracion
  OWNER TO postgres;
```

	desde text	hasta text	factor text	descuento text	mes text	year text	tramo text
1	619744.51	1377210	0.04	24789.78	Agosto	2016	tramo 1
2	1377210.01	2295350	0.08	79878.18	Agosto	2016	tramo 2
3	2295350.01	3213490	0.135	206122.43	Agosto	2016	tramo 3
4	617274.01	1371720	0.04	24690.96	Julio	2016	tramo 1
5	1371720.01	2286200	0.08	79559.76	Julio	2016	tramo 2
6	2286200.01	3200680	0.135	205300.76	Julio	2016	tramo 3
7	620986.51	1379970	0.04	24839.46	Septiembre	2016	tramo 1
8	1379970.01	2299950	0.08	80038.26	Septiembre	2016	tramo 2
9	2299950.01	3219930	0.135	206535.5	Septiembre	2016	tramo 3
10	606892.51	1348650	0.04	24275.70	Enero	2016	tramo 1
11	1348650.01	2247750	0.08	78221.70	Enero	2016	tramo 2
12	2247750.01	3146850	0.135	201847.95	Enero	2016	tramo 3
13	606892.51	1348650	0.04	24275.70	Febrero	2016	tramo 1
14	1348650.01	2247750	0.08	78221.70	Febrero	2016	tramo 2
15	2247750.01	3146850	0.135	201847.95	Febrero	2016	tramo 3
16	609930.01	1355400	0.04	24397.20	Marzo	2016	tramo 1
17	1355400.01	2259000	0.08	78630.20	Marzo	2016	tramo 2
18	2259000.01	3162600	0.135	202858.20	Marzo	2016	tramo 3
19	611766.01	1359480	0.04	24470.64	Abril	2016	tramo 1
20	1359480.01	2265800	0.08	78849.84	Abril	2016	tramo 2
21	2265800.01	3172120	0.135	203468.84	Abril	2016	tramo 3
22	614209.51	1364910	0.04	24568.38	Mayo	2016	tramo 1
23	1364910.01	2274850	0.08	79164.78	Mayo	2016	tramo 2
24	2274850.01	3184790	0.135	204281.53	Mayo	2016	tramo 3
25	616045.51	1368990	0.04	24641.82	Junio	2016	tramo 1
26	1368990.01	2281650	0.08	79401.42	Junio	2016	tramo 2
27	2281650.01	3194310	0.135	204892.17	Junio	2016	tramo 3

LIQUIDACIÓN  
CREATE TABLE liquidacion\_remuneracion

```
(
  sbase integer DEFAULT 0,
  imsegres real DEFAULT 0,
  imsis real DEFAULT 0,
  colacion integer DEFAULT 0,
  dntrabajados integer DEFAULT 0,
  tipo_sueldo character varying(25),
  gratificacion real DEFAULT 0,
  aguinaldo real DEFAULT 0,
  comision real DEFAULT 0,
  mret real DEFAULT 0,
  movilizacion integer DEFAULT 0,
  dxlic integer DEFAULT 0,
  dias_mensuales integer DEFAULT 0,
  horas_semanales integer DEFAULT 0,
  dias_semanales integer DEFAULT 0,
  atrasos integer DEFAULT 0,
  h_ext5 real DEFAULT 0,
  h_ext10 real DEFAULT 0,
  monto real DEFAULT 0,
  tramo character varying(10),
  fam integer DEFAULT 0,
  mat integer DEFAULT 0,
  inva integer DEFAULT 0,
  empleado character varying(25),
  mes text,
  year text,
  empresa text
);
WITH (
  OIDS=FALSE
);
ALTER TABLE liquidacion_remuneracion
  OWNER TO postgres;
```

	sbase integer	imsegres real	imsis real	colacion integer	dntrabajados integer	tipo_sueldo character varying(25)	gratificacion real	aguinaldo real	comision real	movilizacion integer	dxlic integer	dias_mensuales integer	horas_semanales integer	dias_semanales integer	atrasos integer	h_ext5 real	h_ext10 real	monto real	tramo character varying(10)	fam integer	mat integer	inva integer	empleado character varying(25)	mes year	empresa text
1	600000	0	0	0	0	Normal	151927	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17.780.146-0	Julio 2016	76.106-E
2	200000	0	0	0	0	Normal	76000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18.452.054-0	Mayo 2016	76.106-E
3	300000	0	0	0	0	Normal	76000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18.782.039-0	Junio 2016	76.106-E
4	300000	0	0	0	0	Normal	76000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18.139.140-0	Mayo 2016	76.106-E
5	300000	0	0	0	0	Normal	89956	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18.056.550-2	Mayo 2016	76.106-E
6	447000	0	0	0	0	Normal	89956	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18.987.201-0	Mayo 2016	76.106-E
7	500000	0	0	0	0	Normal	89956	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16.117.880-7	Mayo 2016	76.106-E
8	400000	0	0	0	0	Normal	89956	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17.046.704-4	Mayo 2016	76.106-E
9	200000	0	0	0	18	Normal	76000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17.120.942-0	Mayo 2016	76.106-E
10	146000	0	0	0	0	Normal	151927	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15.561.955-0	Julio 2016	76.106-E
11	146000	0	0	0	0	Normal	89956	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15.561.955-0	Junio 2016	76.106-E
12	190000	0	0	0	20	Normal	87900	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17762022-0	Mayo 2016	76.106-E
13	300000	0	0	0	0	Normal	76000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17.120.942-0	Junio 2016	76.106-E
14	300000	0	0	0	0	Normal	87900	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17762022-0	Junio 2016	76.106-E
15	300000	0	0	0	22	Normal	76000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16.596.841-90	Junio 2016	76.106-E
16	647000	0	0	0	0	Normal	151927	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15.987.201-0	Junio 2016	76.106-E
17	520000	0	0	0	15	Normal	89956	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16.556.550-2	Junio 2016	76.106-E
18	400000	0	0	0	0	Normal	89956	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16.497.260-1	Junio 2016	76.106-E
19	500000	0	0	0	0	Normal	151927	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16.117.880-7	Julio 2016	76.106-E
20	447000	0	0	0	0	Normal	89956	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15.987.201-0	Junio 2016	76.106-E
21	500000	0	0	0	0	Normal	89956	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16.117.880-7	Junio 2016	76.106-E
22	400000	0	0	0	0	Normal	89956	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17.046.704-4	Junio 2016	76.106-E
23	146000	0	0	0	0	Normal	89956	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15.987.201-0	Junio 2016	76.106-E
24	460000	0	0	0	0	Normal	89956	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18.452.054-0	Junio 2016	76.106-E
25	300000	0	0	0	0	Normal	151927	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18.056.550-2	Julio 2016	76.106-E
26	400000	0	0	0	0	Normal	89956	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16.497.260-1	Mayo 2016	76.106-E
27	400000	0	0	0	0	Normal	151927	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16.596.841-90	Julio 2016	76.106-E
28	200000	0	0	0	0	Normal	76000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16.596.841-90	Julio 2016	76.106-E
29	400000	0	0	0	0	Normal	100000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17.046.704-4	Julio 2016	76.106-E
30	146000	0	0	0	0	Normal	151927	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15.987.201-0	Julio 2016	76.106-E
31	400000	0	0	0	12	Normal	100000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16.497.260-1	Julio 2016	76.106-E
32	200000	0	0	0	0	Normal	76000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17.120.942-0	Julio 2016	76.106-E
33	500000	0	0	0	0	Normal	151927	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16.497.260-1	Sept 2016	76.106-E
34	300000	0	0	0	0	Normal	87900	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17762022-0	Julio 2016	76.106-E
35	400000	0	0	0	0	Normal	100000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18.452.054-0	Julio 2016	76.106-E
36	146000	0	0	0	0	Normal	89956	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16.217.279-4	Junio 2016	76.106-E
37	146000	0	0	0	0	Normal	151927	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16.217.279-4	Julio 2016	76.106-E
38	146000	0	0	0	0	Normal	151927	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16.217.279-4	Agosto 2016	76.106-E
39	200000	0	0	0	0	Normal	76000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16.782.839-0	Agosto 2016	76.106-E
40	300000	0	0	0	12	Normal	76000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16.782.839-0	Julio 2016	76.106-E
41	300000	0	0	0	0	Normal	76000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16.596.841-90	Sept 2016	76.106-E

# BASE DE DATOS

## Controlador

### FRAMEWORK CODEIGNITER

```
$active_group = 'default';
$query_builder = TRUE;
$active_record = TRUE;
$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'postgres',
    'password' => ' ',
    'database' => 'contabilidad',
    'dbdriver' => 'postgre',
    'port' => '5432',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);
```

### CONEXIÓN PHP AJAX

```
<?
$conn_string = "host=localhost port=5432 dbname=contabilidad user=postgres password=" . $password . ";";
$conexion = pg_connect($conn_string);
```

# BASE DE DATOS

## Ejemplo consulta de comprobación simple

### CREAR DEPÓSITO CONVENIDO CON JS

Se incluye la conexión, se reciben los parámetros enviados con cadena json desde Ajax del js de previred, y se comprueba la existencia del depósito por el mes y por el año. Si no existe una coincidencia, se hace una inserción con los parámetros recibidos, y si existe, se realiza una actualización del valor del depósito mientras se cumpla la condición de que el mes y el año sea igual a los recibidos por parámetro.

Se envía un mensaje de confirmación al Ajax para que muestre un mensaje al usuario de la acción realizada, como creación o actualización.

Envío:

```
function crearDc(){
    var dc1 = $('#dc1').val();
    var mes1 = $('#mes1').val();
    var year = $('#year').val();
    $.ajax({
        type: 'POST',
        dataType: 'json',
        data: 'dc1='+dc1 + '&mes1='+ mes1 + '&year='+ year,
        url: 'http://remuneracion.ingetecservice.cl/remuneracion/application/js/phpprevi/crearDc.php',
        success: function(data){
            if(data == 'Ingresado'){
                $('#mensajedc').html('<p class="alert alert-success">Registro Ingresado</p>');
            }else if(data == 'Actualizado'){
                $('#mensajedc').html('<p class="alert alert-success">Registro Actualizado</p>');
            }else{
                $('#mensajedc').html('<p class="alert alert-danger">Intentalo mas Tarde.</p>');
            }
        }
    });
}
```

Recepción:

```
<?php
include('conexion.php');
session_start();
$dc1 = $_POST['dc1'];
$mes1 = $_POST['mes1'];
$year = $_POST['year'];

$comprobar = pg_num_rows(pg_query("SELECT * FROM dc_remuneracion WHERE mes='$mes1' AND year='$year'"));

if($comprobar == 0){

    pg_query("INSERT INTO dc_remuneracion (dc1,mes,year)VALUES('$dc1','$mes1','$year')");

    $registro = 'Ingresado';
    echo json_encode($registro);

}else{

    pg_query("UPDATE dc_remuneracion SET dc1='$dc1' WHERE mes='$mes1' AND year='$year'");

    $registro = 'Actualizado';
    echo json_encode($registro);
}
?>
```

# BASE DE DATOS

## RECUPERACIÓN DE DATOS PREVISIONALES DESDE EL MODELO DE CODEIGNITER CON ACTIVE RECORD

Se reciben los parámetros de mes y año en la función `getAfp()`, con el active record se le indica que el mes, tiene que ser igual a la variable recibida junto al año, desde la tabla AFP, si el result del query es mayor a 0 se devuelve el query para hacer el listar desde un `foreach()` en el controlador o en la vista desde un array.

```
function getAfp($mes,$year){
    $this->db->where('mes',$mes);
    $this->db->where('year',$year);
    $query=$this->db->get('afp_remuneracion');

    if($query->num_rows() > 0)
    {
        return $query;
    }else{
        return false;
    }
}
```