

Universidad de Granada
Diseño y desarrollo de sistemas de información

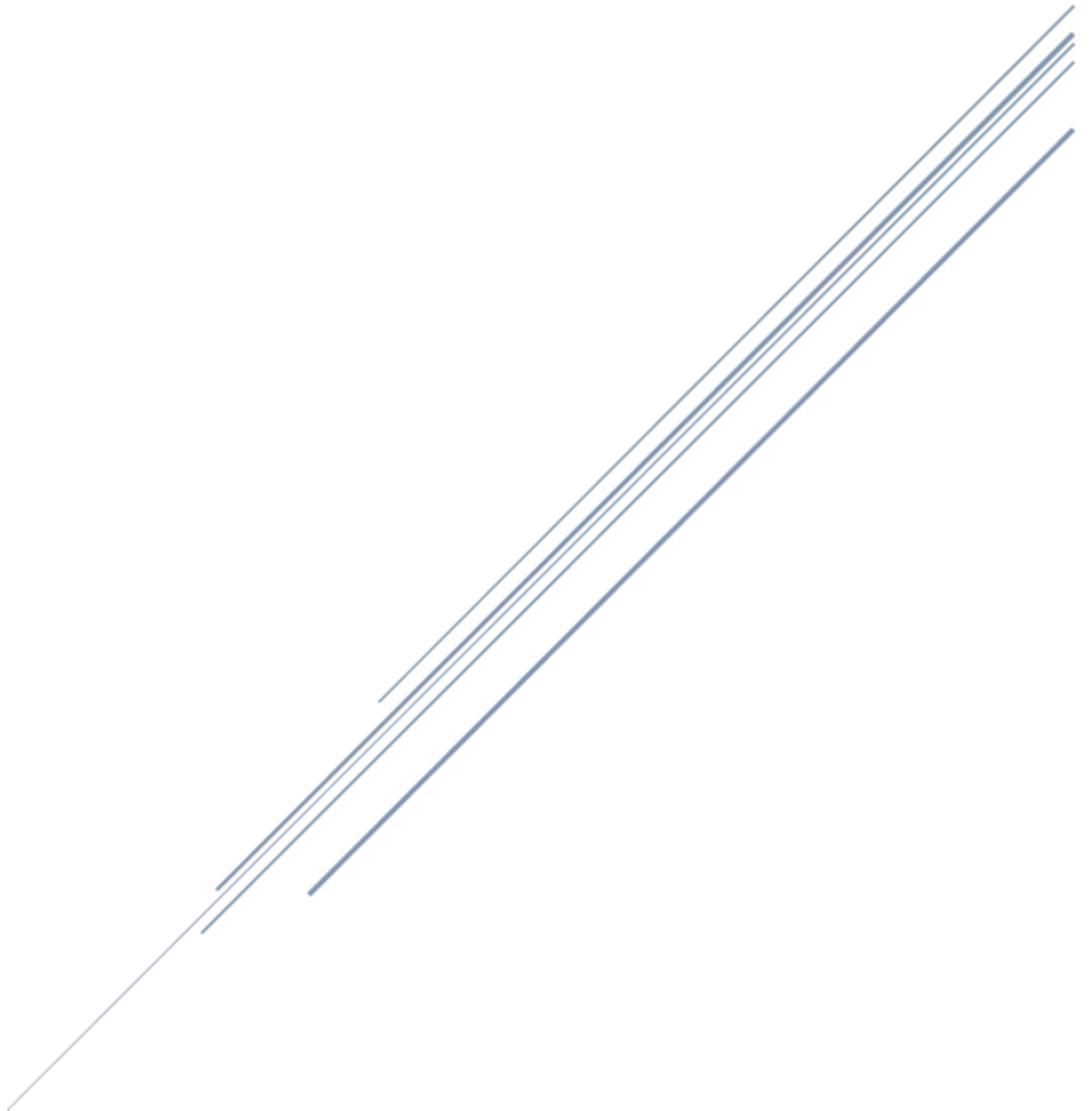
GRUPO 4 - HARDWARE EQUISDE

Ignacio Barragán Lozano

Francisco José Cotán López

Juan Manuel Rubio Rodríguez

Ignacio Vellido Expósito



Índice

Descripción del sistema	2
Análisis de requisitos	3
Producción	3
Marketing	5
Finanzas	7
Recursos humanos	10
Diseño funcional	14
Diagrama de caja negra	14
Diagrama de flujo de datos 0 (armazón)	15
Producción	15
Marketing	16
Finanzas	17
Recursos humanos	18
Diagrama de flujo de datos 1	19
Producción	19
Marketing	20
Finanzas	21
Recursos humanos	22
Diagrama de flujo de datos 2	23
Producción	23
Marketing	24
Finanzas	25
Recursos humanos	26
Esquema funcional	27
Esquemas externos	28
Producción	28
Marketing	29
Finanzas	30
Recursos humanos	31
Paso a tablas	32
Implementación	38
Creación de tablas	38
Insertado de tuplas	43
Listado de tuplas	45
Diseño de cursores	46
Diseño de disparadores	51

Descripción del sistema

Realizaremos un sistema de información para una empresa de construcción de componentes hardware, que facilite la gestión de la empresa. Entre su gama de productos se encuentran procesadores, tarjetas gráficas, de memoria, etc.

El mercado hacia el que se orienta la empresa es el de la venta de sus productos de forma individual, pero no por ello niega la fabricación de componentes específicos para otras marcas. Esta empresa depende de la fabricación de piezas de otras empresas externas. Por tanto, se debe poder gestionar la compra de estas a proveedores externos.

La empresa se divide en las áreas: Marketing, Finanzas, Producción y Recursos Humanos.

En el área de marketing se pretende hacer la gestión de las distintas campañas publicitarias que pueda llegar a tener la empresa, registrando tanto el tipo de campaña como el público objetivo al que se orienta. Además de esto, se pretende almacenar las comparaciones entre nuestros productos y los productos de los competidores.

En el área de finanzas, queremos llevar la contabilidad al día y registrar todos los ingresos y los gastos de la empresa y poder consultar en cualquier momento tanto cuándo como porqué se ha realizado cada uno.

En el área de producción debemos mantener una base de datos con todos los productos que están en fabricación, así como aquellos que hemos dejado de producir. También es necesario contabilizar el número de productos producidos y distribuidos.

En el área de recursos humanos se gestionará la información relativa al personal, entre lo que se encuentra el alta y baja de empleados. Además, se podrá administrar la agrupación de personal en departamentos, lo que permite una organización de grupos de trabajo en la empresa.

Análisis de requisitos

Producción – Ignacio Barragán Lozano

Requisitos de datos

RD1-P. Datos para dar de alta un producto:

- Nombre (cadena de hasta 30 caracteres no vacía)
- Familia (cadena de hasta 20 caracteres no vacía)
- Modelo (cadena de hasta 20 caracteres no vacía)
- Piezas usadas en su fabricación (vector de enteros mayores que 0)
- Especificaciones (cadena de hasta 100 caracteres no vacía)
- Precio (float mayor que 0)

RD2-P. Datos almacenados de un producto:

- Identificador (entero mayor que 0)
- Nombre (cadena de hasta 30 caracteres no vacía)
- Familia (cadena de hasta 20 caracteres no vacía)
- Modelo (cadena de hasta 20 caracteres no vacía)
- Piezas usadas en su fabricación (vector de enteros mayores que 0)
- Especificaciones (cadena de hasta 100 caracteres no vacía)
- Precio (float mayor que 0)
- Stock (entero mayor o igual que 0)

RD3-P. Identificador devuelto al dar de alta un producto:

- Identificador (entero mayor que 0)

RD4-P. Identificador de un producto a señalar como obsoleto:

- Identificador (entero mayor que 0)

RD5-P. Producto fabricado hoy:

- Identificador (entero mayor que 0)
- Fabricados (entero mayor que 0)

RD6-P. Producto enviado al distribuidor:

- Identificador (entero mayor que 0)
- Enviados (entero mayor que 0)

RD7-P. Identificador de un producto para obtener sus datos:

- Identificador (entero mayor que 0)

RD8-P. Datos solicitados de un producto:

Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

- Nombre (cadena de hasta 30 caracteres no vacía)
- Familia (cadena de hasta 20 caracteres no vacía)
- Modelo (cadena de hasta 20 caracteres no vacía)
- Piezas usadas en su fabricación (vector de enteros mayores que 0)
- Especificaciones (cadena de hasta 100 caracteres no vacía)
- Precio (float mayor que 0)
- Stock (entero mayor o igual que 0)

Requisitos funcionales

RF1-P. Dar de alta producto: Introduce un producto en el sistema y pone su estado “En fabricación”. El sistema pone su stock a 0, le asigna un identificador y lo devuelve como salida.

Entrada: RD1-P

Manejo: RD2-P

Salida: RD3-P

RF2-P. Poner producto como obsoleto: Cambia el estado de un producto a “Obsoleto”.

Entrada: RD4-P

Manejo: RD2-P

RF3-P. Productos fabricados hoy: Admite como entrada un identificador de producto y un número natural, y aumenta el stock del producto con ese mismo número.

Entrada: RD5-P

Manejo: RD2-P

RF4-P. Envía producto al distribuidor: Acepta como entrada un identificador de producto y un número natural y disminuye su stock en ese número.

Entrada: RD6-P

Manejo: RD2-P

RF5-P. Consulta datos de un producto: Devuelve los datos de un producto a partir de su identificador.

Entrada: RD7-P

Manejo: RD2-P

Salida: RD8-P

Restricciones semánticas

RS1-P. Para consultar o modificar datos de un producto a partir de su identificador, el producto debe estar registrado en el sistema.

Marketing – Francisco José Cotán López

Requisitos de datos

RD1-M. Datos para dar de alta una campaña publicitaria

- Nombre (cadena de hasta 50 caracteres).
- Tipo (cadena de hasta 50 caracteres)
- Público objetivo (cadena de hasta 50 caracteres)

RD2-M. Datos que se almacenan de una campaña publicitaria

- Nombre (cadena de hasta 50 caracteres)
- Tipo (cadena de hasta 50 caracteres)
- Público objetivo (cadena de hasta 50 caracteres)
- Identificador (entero mayor o igual que 0)

RD3-M. Identificador devuelto al dar de alta una campaña publicitaria

- Identificador (entero mayor o igual que 0)

RD4-M. Identificador de entrada para dar de baja a una campaña publicitaria

- Identificador (entero mayor o igual que 0)

RD5-M. Datos para modificar una campaña publicitaria

- Nombre (cadena de hasta 50 caracteres).
- Tipo (cadena de hasta 50 caracteres)
- Público objetivo (cadena de hasta 50 caracteres)
- Identificador (entero mayor o igual que 0)

RD6-M. Identificador de entrada para consultar una campaña publicitaria

- Identificador (entero mayor o igual que 0)

RD7-M. Datos como salida a la consulta de una campaña publicitaria

- Nombre (cadena de hasta 50 caracteres)
- Tipo (cadena de hasta 50 caracteres)
- Público objetivo (cadena de hasta 50 caracteres)

RD8-M. Datos para crear la comparación de un producto con uno competidor

- Nombre (cadena no mayor a 50 caracteres)

Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

- Identificador producto (entero mayor que 0)
- Precio (float)
- Rendimiento (float)
- Informe (cadena no mayor a 2000 caracteres)

RD9-M. Datos que se almacenan en la comparación de un producto propio con uno competidor

- Nombre (cadena no mayor a 50 caracteres)
- Identificador producto (entero mayor o igual que 0)
- Informe (cadena no mayor a 2000 caracteres)
- Precio (float)
- Rendimiento (float)
- Identificador (entero mayor o igual que 0)

RD10-M. Identificador devuelto al crear el análisis de un competidor

- Identificador (entero mayor o igual que 0)

Requisitos funcionales

RF1-M. Crear campaña publicitaria: Introduce en el sistema los datos de una campaña, le asigna un identificador, el cual lo devuelve como salida.

Entrada: RD1

Manejo: RD2

Salida: RD3

RF2-M. Dar de baja a una campaña publicitaria: Elimina una campaña publicitaria a partir de su identificador.

Entrada: RD4

Manejo: RD2

Salida:

RF3-M. Modificar una campaña publicitaria: Modifica los datos de una campaña adjuntando su identificador.

Entrada: RD5

Manejo: RD2

Salida:

RF4-M. Consulta de una campaña publicitaria: Adjuntando su identificador, devuelve datos de una campaña publicitaria.

Entrada: RD6

Manejo: RD2

Salida: RD7

RF5-M. Crear comparación con producto competidor: Introduce en el sistema información la comparación hecha entre un producto nuestro y un producto de una empresa competidora. Se devuelve un identificador del análisis.

Entrada: RD8

Manejo: RD9
Salida: RD10

Restricciones semánticas

RS1-M. No pueden crearse más de 3 campañas publicitarias en el mismo mes.

RS2-M. Rendimiento en la creación de la comparación estará comprendido entre $[-1;1]$, o, en otras palabras, el porcentaje entre la comparación del rendimiento entre las dos piezas, siendo negativo si el suyo es más eficiente y positivo si viceversa.

Finanzas – Juan Manuel Rubio Rodríguez

Requisitos de datos

RD1-F. Datos para crear un ingreso:

- Nombre (cadena de 30 caracteres no vacía)
- Apellidos (cadena de 30 caracteres no vacía)
- Teléfono (cadena de 13 caracteres)
- Correo electrónico (cadena de 40 caracteres no vacía)
- NIF (cadena de caracteres no vacía)
- Concepto (cadena de 100 caracteres)
- Importe (float mayor que cero no vacío)

RD2-F. Datos almacenados de un ingreso:

- Nombre (cadena de 30 caracteres no vacía)
- Apellidos (cadena de 30 caracteres no vacía)
- Teléfono (cadena de 13 caracteres)
- Correo electrónico (cadena de 40 caracteres no vacía)
- NIF (cadena de caracteres no vacía)
- Concepto (cadena de 100 caracteres)
- Importe (float mayor que cero)
- Fecha (date)
- Identificador (entero)

RD3-F. Datos de salida para un ingreso:

- Identificador (entero)

RD4-F. Datos para consultar un ingreso:

- Identificador (entero)

RD5-F. Datos que se devuelven al consultar un ingreso:

Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

- 100

Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

- Identificador (entero)

RD11-F. Datos que se devuelven al consultar un pago:

- Nombre (cadena de 30 caracteres no vacía)
- Apellidos (cadena de 30 caracteres no vacía)
- Teléfono (cadena de 13 caracteres)
- Correo electrónico (cadena de 40 caracteres no vacía)
- NIF (cadena de caracteres no vacía)
- Concepto (cadena de 100 caracteres)
- Fecha (date)
- Importe (float menor que cero)
- Identificador (entero)

Requisitos funcionales

RF1-F. Dar de alta un ingreso: Introduce en el sistema los datos del ingreso, se le asigna un identificador y lo devuelve.

Entrada: RD1-F

Manejo: RD2-F

Salida: RD3-F

RF2-F. Consultar un ingreso: Muestra los datos de un ingreso a partir de su identificador.

Entrada: RD4-F

Manejo: RD2-F

Salida: RD5-F

RF3-F. Dar de alta un pago: Introduce en el sistema los datos del pago a realizar, se le asigna un identificador y lo devuelve.

Entrada: RD6-F

Manejo: RD7-F

Salida: RD8-F

RF4-F. Cancelar pago: Cancela una orden de pago ya efectuada a partir del identificador de pago.

Entrada: RD9-F

Manejo: RD7-F

Salida:

RF5-F. Consultar pago: Muestra los datos de un pago a partir del identificador de pago.

Entrada: RD10-F

Manejo: RD7-F

Salida: RD11-F

Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

Restricciones semánticas

RS1-F. No existirán dos identificadores de pagos iguales.

RS2-F. No existirán dos identificadores de ingresos iguales.

Recursos humanos – Ignacio Vellido Expósito

Requisitos de datos

RD1-H: Datos para dar de alta de trabajador:

- Nombre (cadena de hasta 30 caracteres)
- DNI (cadena de 9 caracteres no vacía)
- Teléfono (cadena de 13 caracteres no vacía)
- Dirección (cadena de hasta 50 caracteres)
- Sueldo (entero)
- Estado (carácter) (alta laboral, baja, ex-empleado)

RD2-H: Datos almacenados de un empleado:

- Nombre (cadena de hasta 30 caracteres)
- DNI (cadena de 9 caracteres no vacía)
- Teléfono (cadena de 13 caracteres no vacía)
- Dirección (cadena de hasta 50 caracteres)
- Sueldo (entero)
- Estado (carácter) (alta laboral, baja, ex-empleado)
- Identificador empleado (entero)

RD3-H: Identificador de empleado devuelto tras el alta (entero)

Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

RD4-H: Datos necesarios para modificación de trabajador:

- Nombre (cadena de hasta 30 caracteres)
- DNI (cadena de 9 caracteres no vacía)
- Teléfono (cadena de 13 caracteres no vacía)
- Dirección (cadena de hasta 50 caracteres)
- Sueldo (entero)
- Estado (carácter) (alta laboral, baja, ex-empleado)
- Identificador empleado (entero)

RD5-H: Datos para dar de baja un empleado:

- Identificador empleado (entero)

RD6-H: Datos de entrada a la consulta de un trabajador:

- Identificador empleado (entero)

RD7-H: Datos devueltos en la consulta de un empleado:

- Nombre (cadena de hasta 30 caracteres)
- DNI (cadena de 9 caracteres no vacía)
- Teléfono (cadena de 13 caracteres no vacía)
- Dirección (cadena de hasta 50 caracteres)
- Sueldo (entero)
- Estado (carácter) (alta laboral, baja, ex-empleado)
- Identificador empleado (entero)

RD8-H: Datos para dar de alta un departamento:

- Área (cadena de hasta 30 caracteres)
- Localización (cadena de hasta 50 caracteres)

RD9-H: Datos almacenados de un departamento:

- Área (cadena de hasta 30 caracteres)
- Localización (cadena de hasta 50 caracteres)
- Identificador departamento (entero)

RD10-H: Identificador de departamento devuelto tras el alta (entero)

RD11-H: Datos necesarios para modificación de un departamento:

- Área (cadena de hasta 30 caracteres)
- Localización (cadena de hasta 50 caracteres)
- Identificador departamento (entero)

RD12-H: Datos para eliminar un departamento:

- Identificador departamento (entero)

RD13-H: Datos de entrada a la consulta de un departamento:

- Identificador empleado (entero)

RD14-H: Datos devueltos en la consulta de un departamento:

- Área (cadena de hasta 30 caracteres)
- Localización (cadena de hasta 50 caracteres)
- Identificador departamento (entero)

Requisitos funcionales

RF1-H. Dar de alta un trabajador: El gestor de recursos humanos puede realizar la operación introduciendo los datos básicos del empleado, a lo que el sistema responderá almacenándolos junto a un identificador único y devolviendo este identificador:

Entrada: RD1-H

Manejo: RD2-H

Salida: RD3-H

RF2-H. Modificar datos del trabajador: Indicando el identificador y los datos a cambiar, el sistema realizará las modificaciones pertinentes.

Entrada: RD4-H

Manejo: RD2-H

Salida:

RF3-H. Eliminar un empleado: A partir del identificador, el sistema elimina la información relativa al trabajador.

Entrada: RD5-H

Manejo: RD2-H

Salida:

RF4-H. Consultar datos empleado: Utilizando el identificador como entrada, el sistema devuelve toda la información almacenada del empleado

Entrada: RD6-H

Manejo: RD2-H

Salida: RD7-H

RF5-H. Crear departamento: El gestor agrupa empleados en un nuevo departamento, introduciendo la información principal, y el sistema actuará almacenando dichos datos y devolviendo el identificador generado:

Entrada: RD8-H

Manejo: RD9-H

Salida: RD10-H

RF6-H. Eliminar departamento: Recibiendo como entrada el identificador, se borra toda la información almacenada, dejando a sus trabajadores sin departamento:

Entrada: RD11-H

Manejo: RD9-H

Salida:

RF7-H. Modificar datos de un departamento: El gestor modifica la lista de miembros, o la información de un departamento, en base a la entrada del identificador y los datos a cambiar, y el sistema se alterará de la forma correspondiente:

Entrada: RD12-H

Manejo: RD9-H

Salida:

RF8-H. Consultar información de departamento: A partir del identificador se muestra toda la información del departamento:

Entrada: RD13-H

Manejo: RD9-H

Salida: RD14-H

Restricciones semánticas

RS1-H. No existirán dos identificadores de empleados iguales

RS2-H. No existirán dos identificadores de departamento iguales

Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

RS3-H. Un empleado no podrá pertenecer a dos departamentos al mismo tiempo

Diseño funcional

Diagrama de caja negra

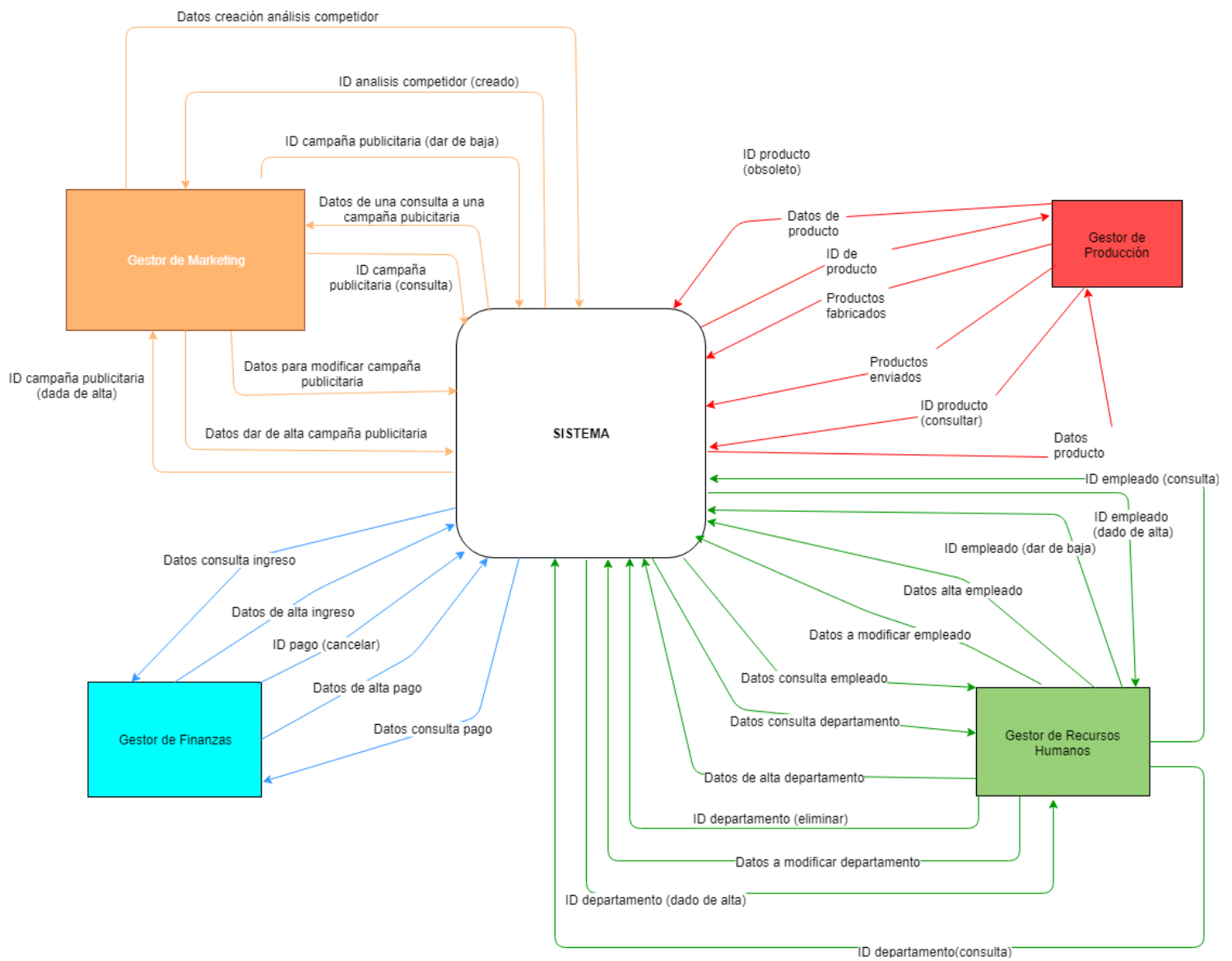


Diagrama de flujo de nivel 0

Diagrama de flujo de nivel 0 - Producción



Diagrama de flujo de nivel 0 - Marketing

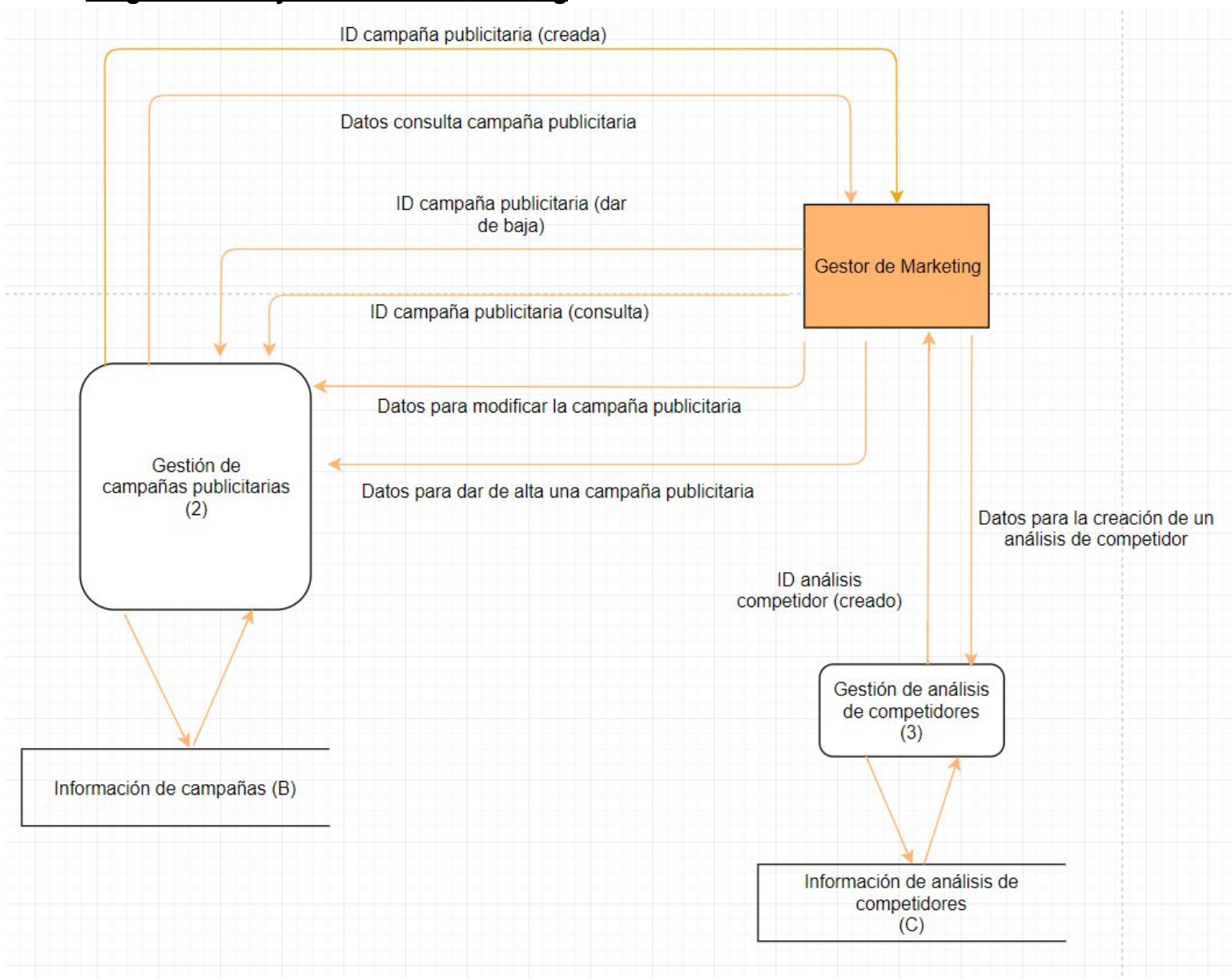
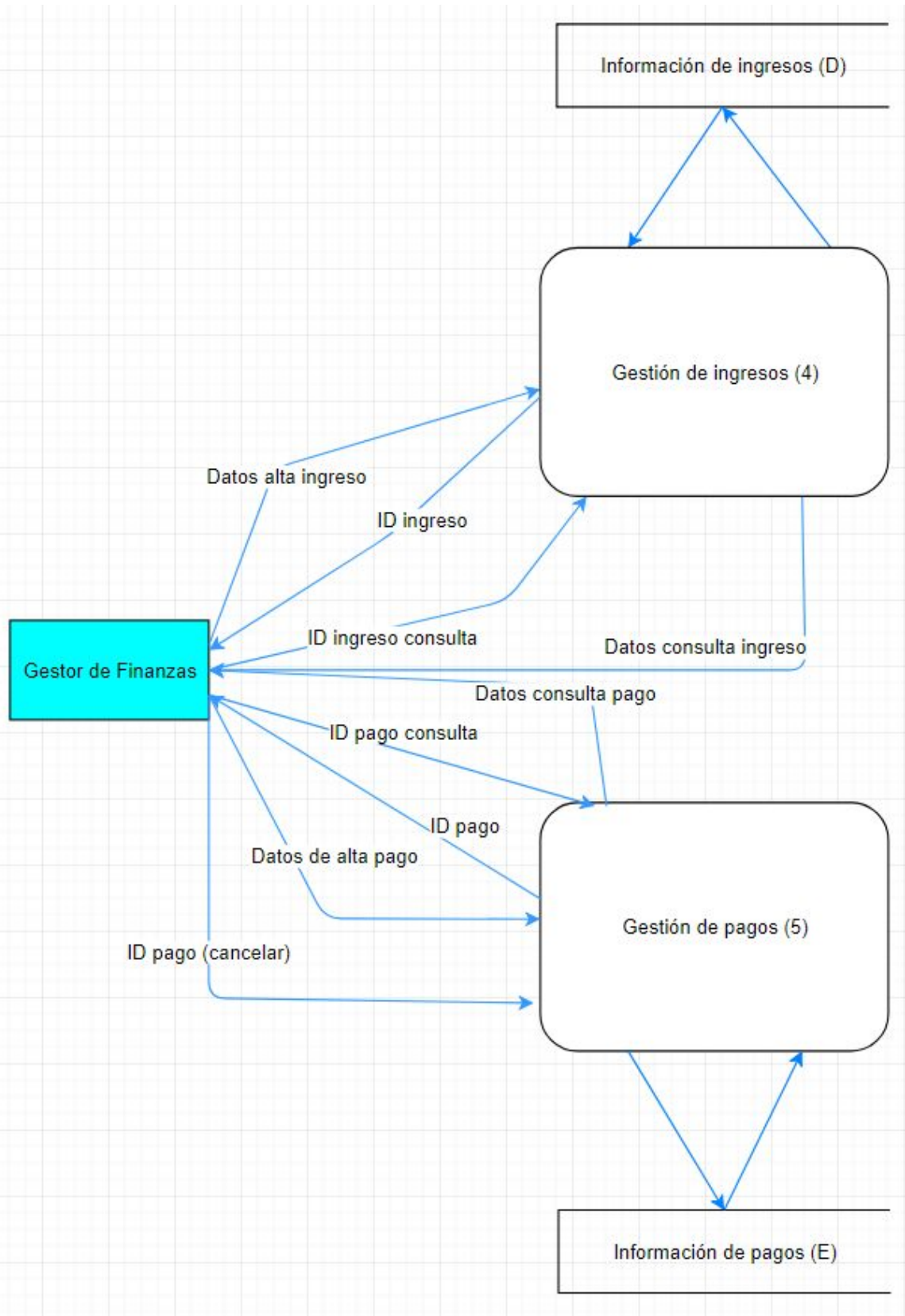
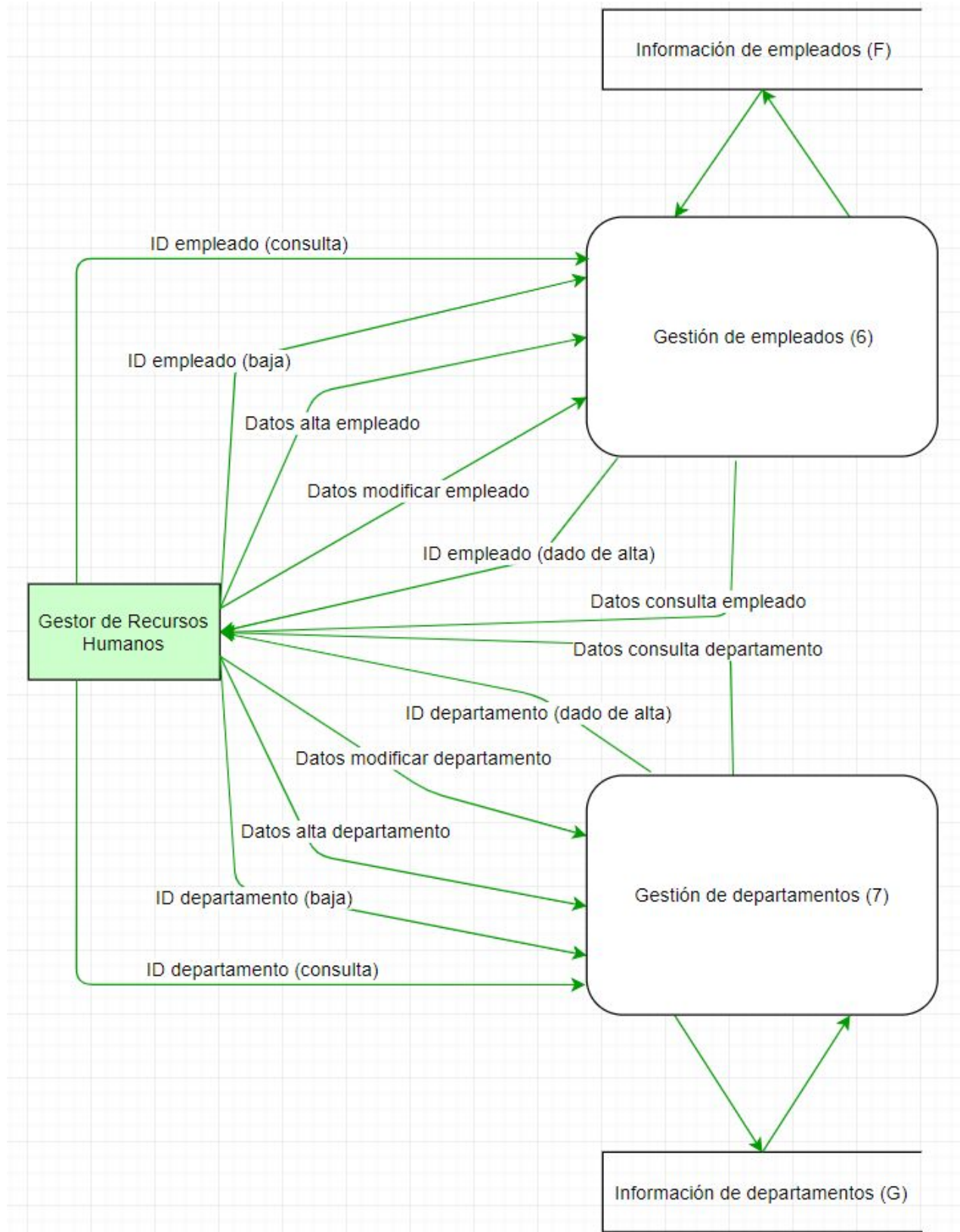


Diagrama de flujo de nivel 0 - Finanzas



Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

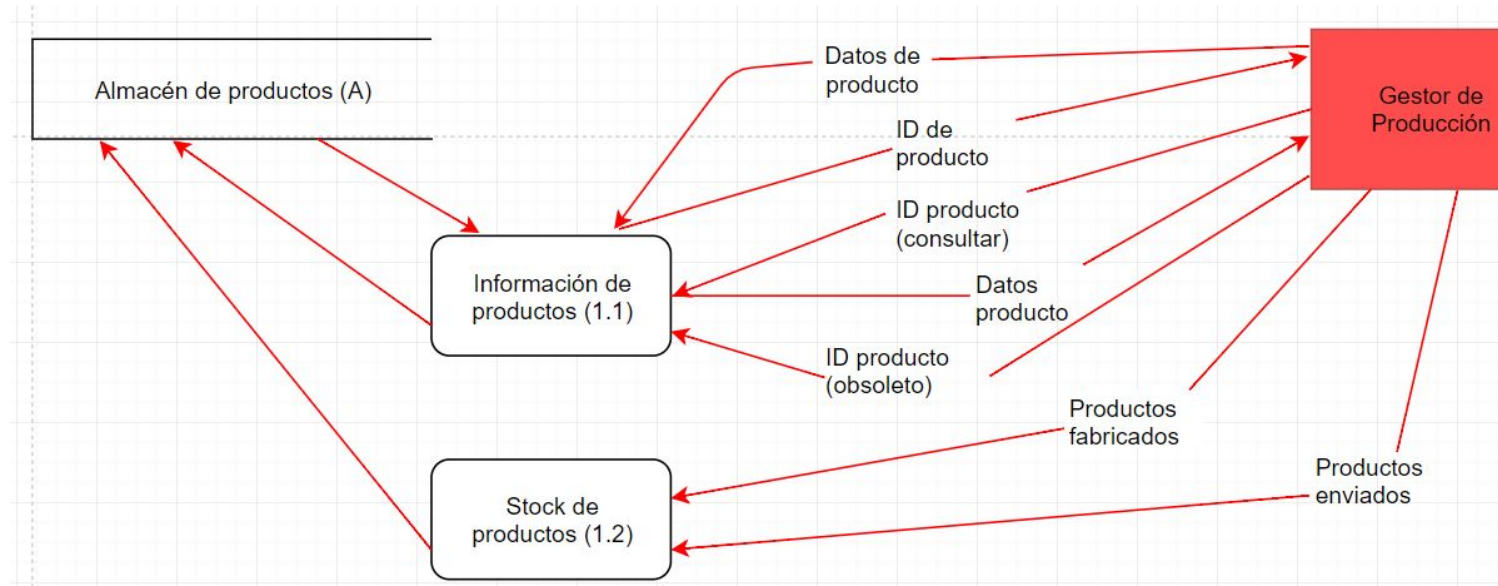
Diagrama de flujo de nivel 0 – Recursos humanos



Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

Diagrama de flujo de nivel 1

Diagrama de flujo de nivel 1 - Producción



Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

Diagrama de flujo de nivel 1 - Marketing

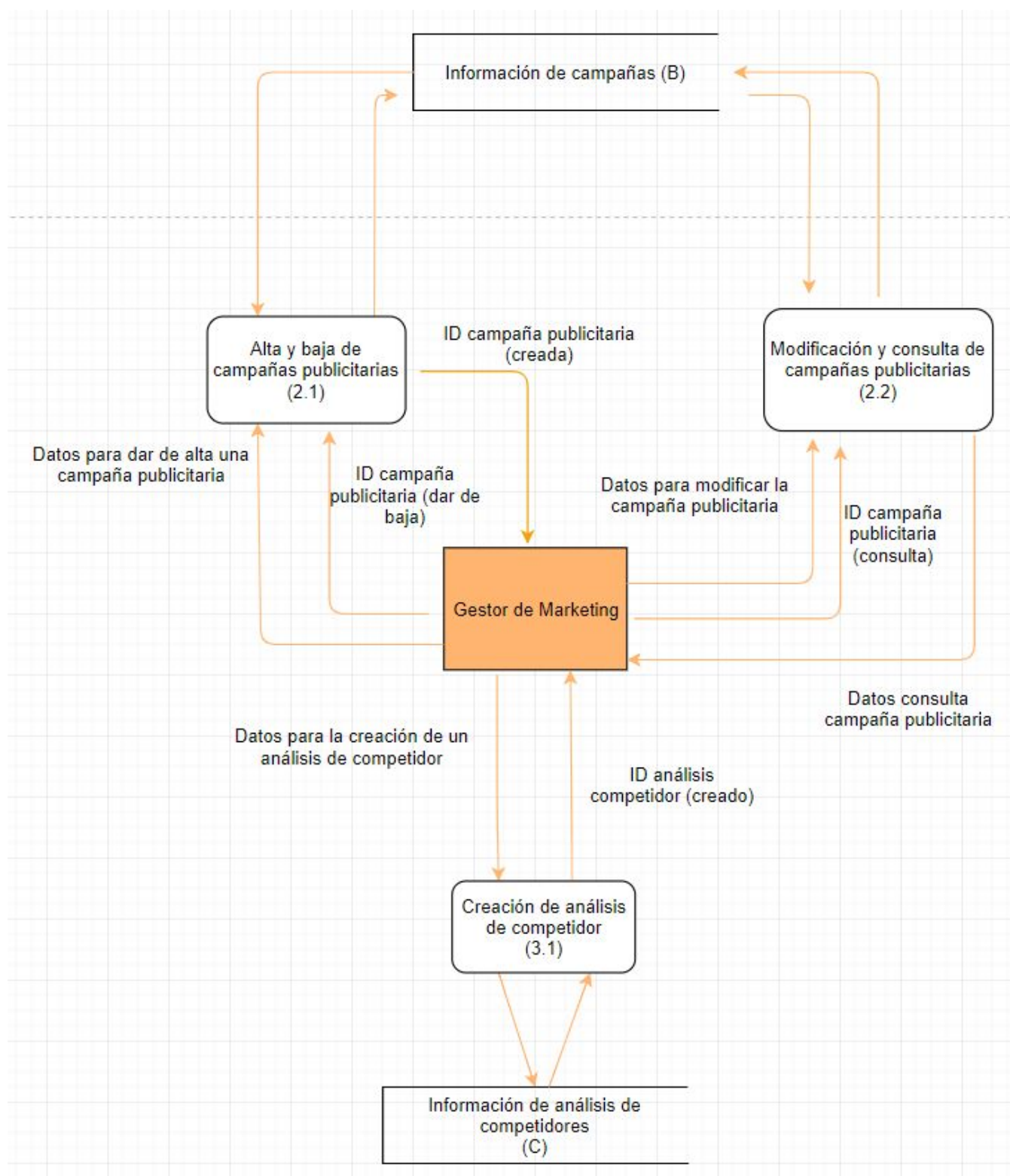


Diagrama de flujo de nivel 1 - Finanzas

Ignacio Barragán Lozano
 Francisco José Cotán López
 Juan Manuel Rubio Rodríguez
 Ignacio Vellido Expósito

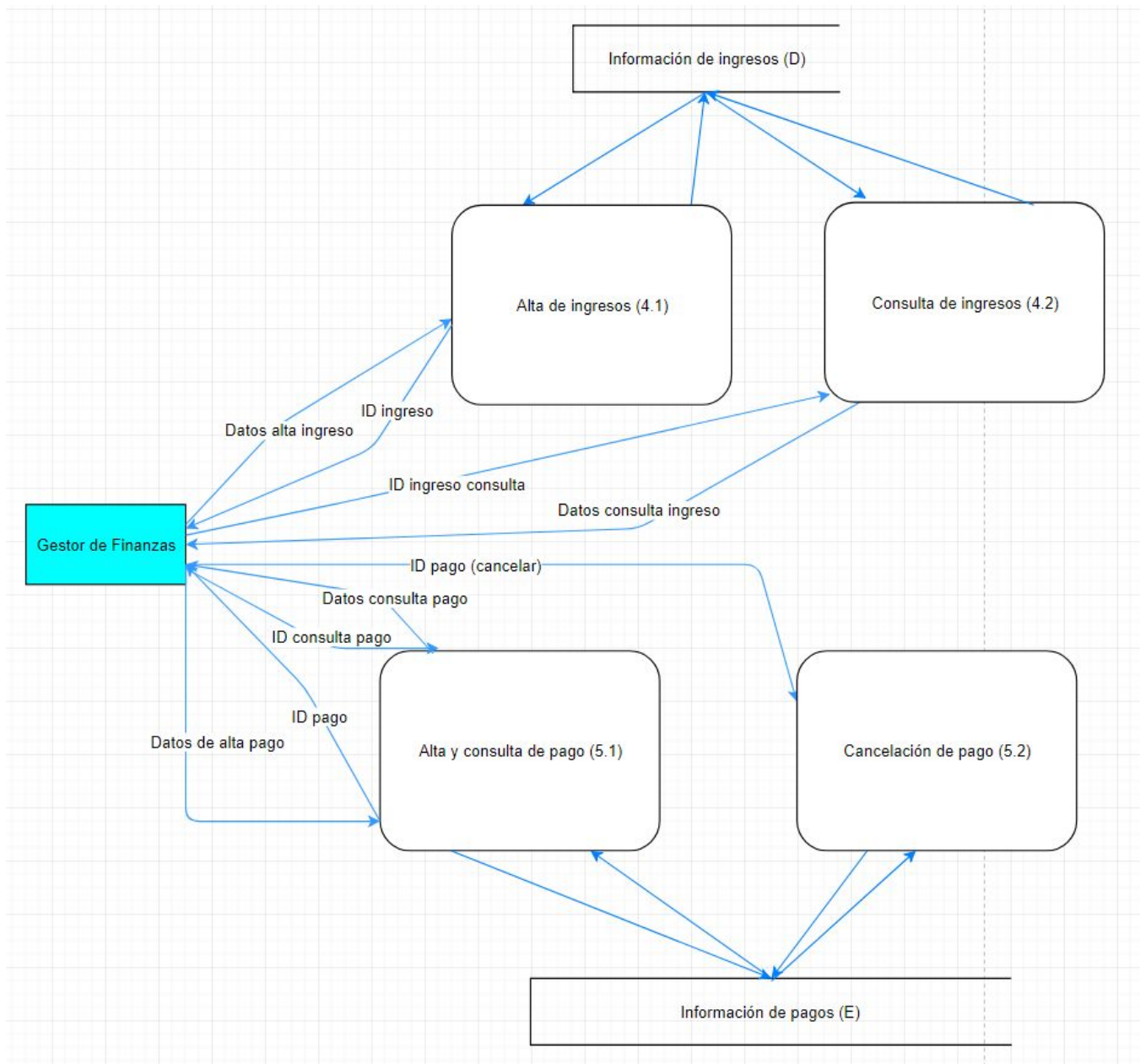


Diagrama de flujo de nivel 1 – Recursos humanos

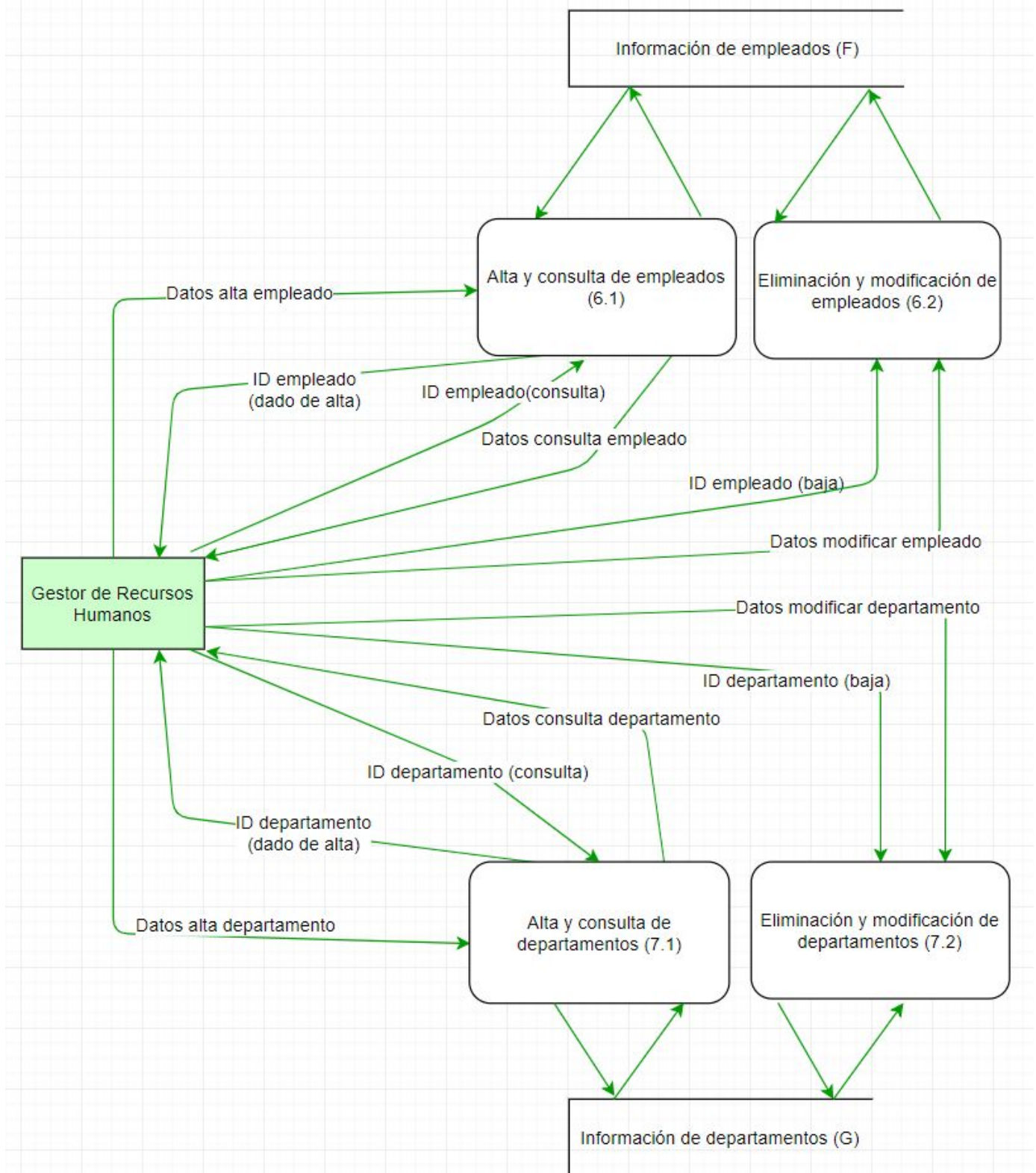
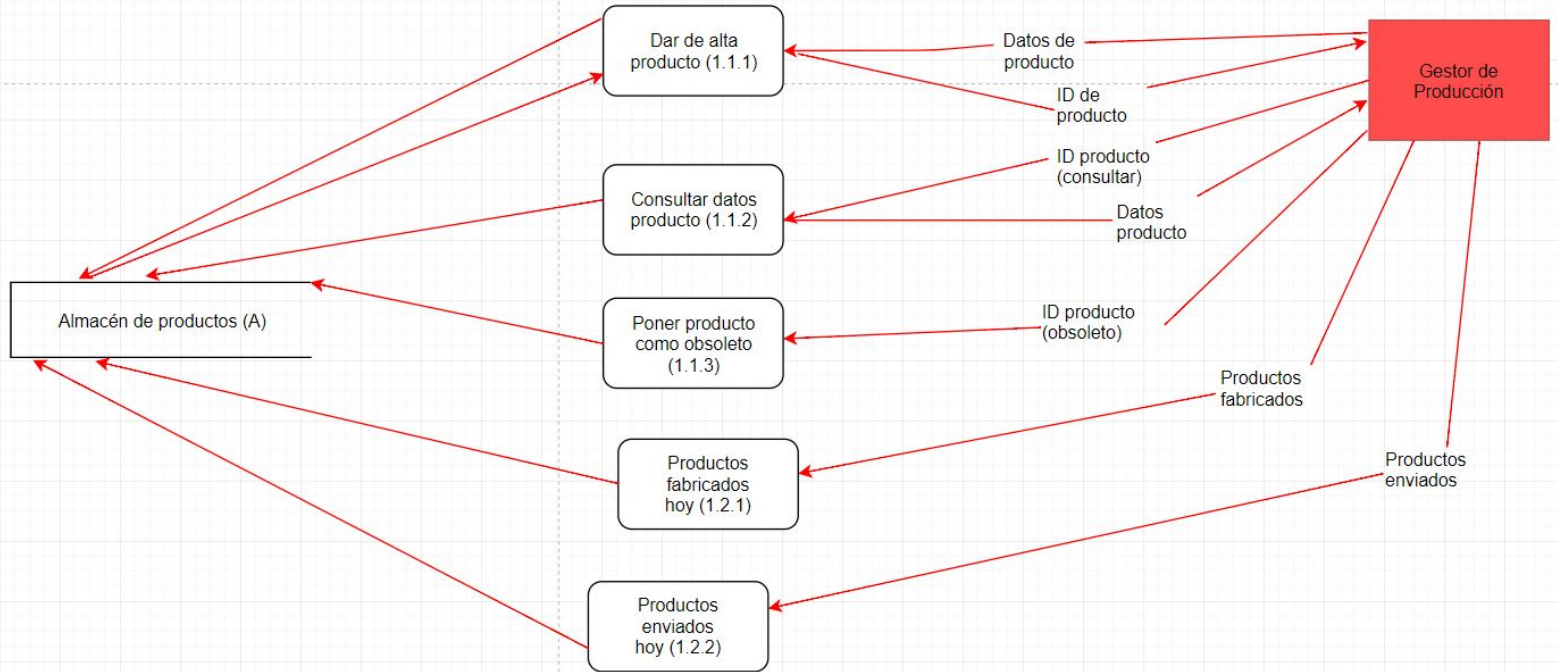


Diagrama de flujo de nivel 2

Diagrama de flujo de nivel 2 - Producción



Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

Diagrama de flujo de nivel 2 - Marketing

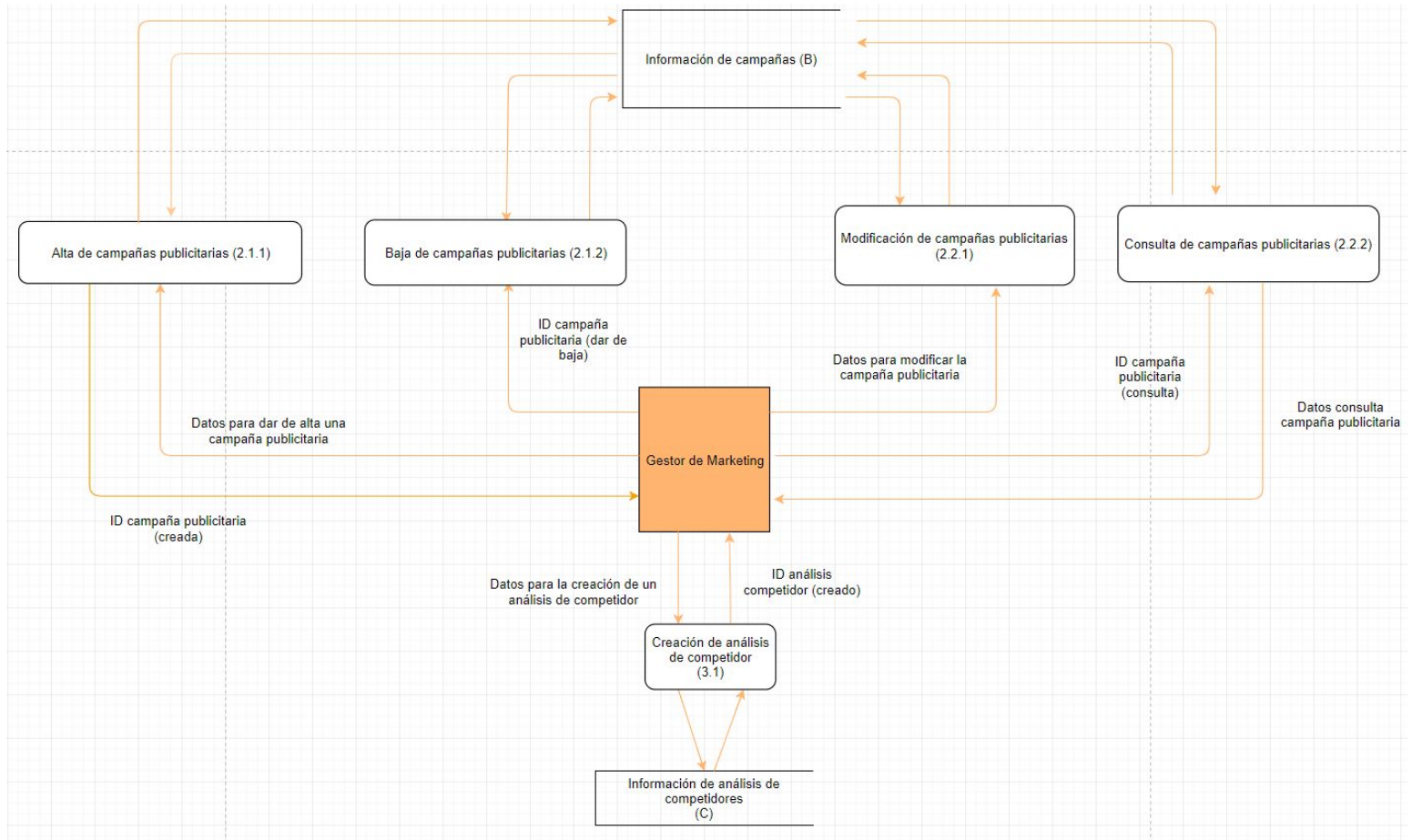
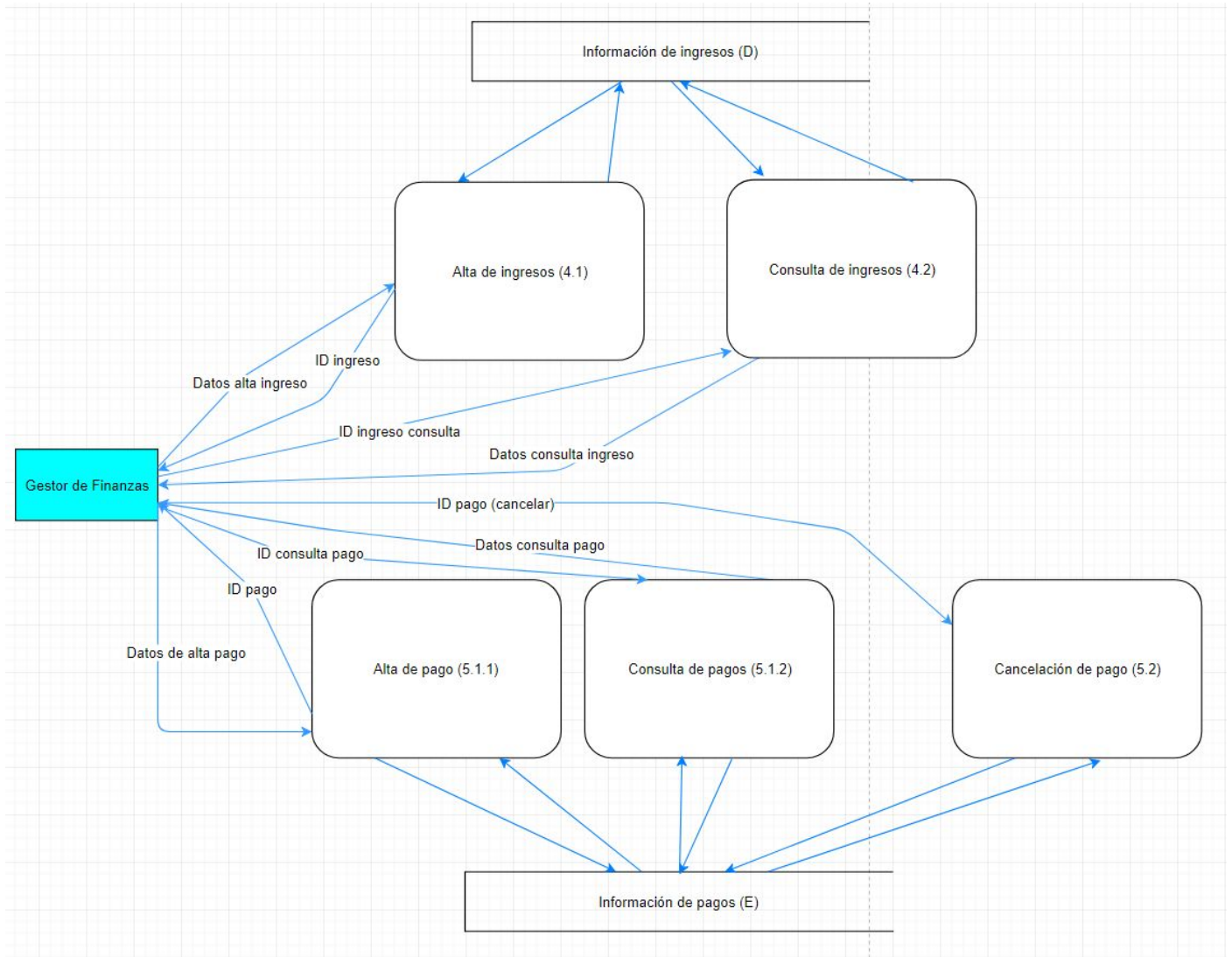
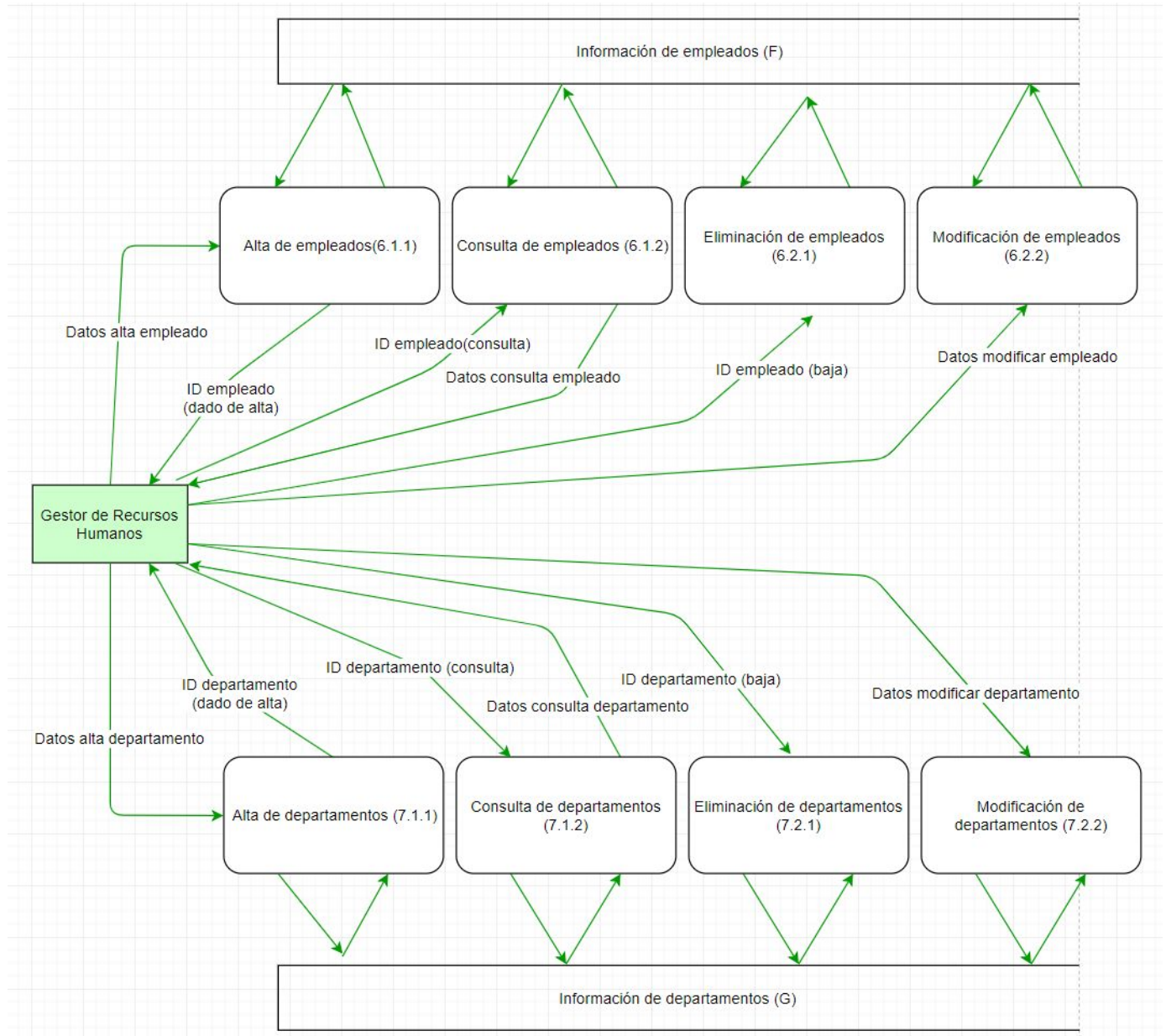


Diagrama de flujo de nivel 2 - Finanzas

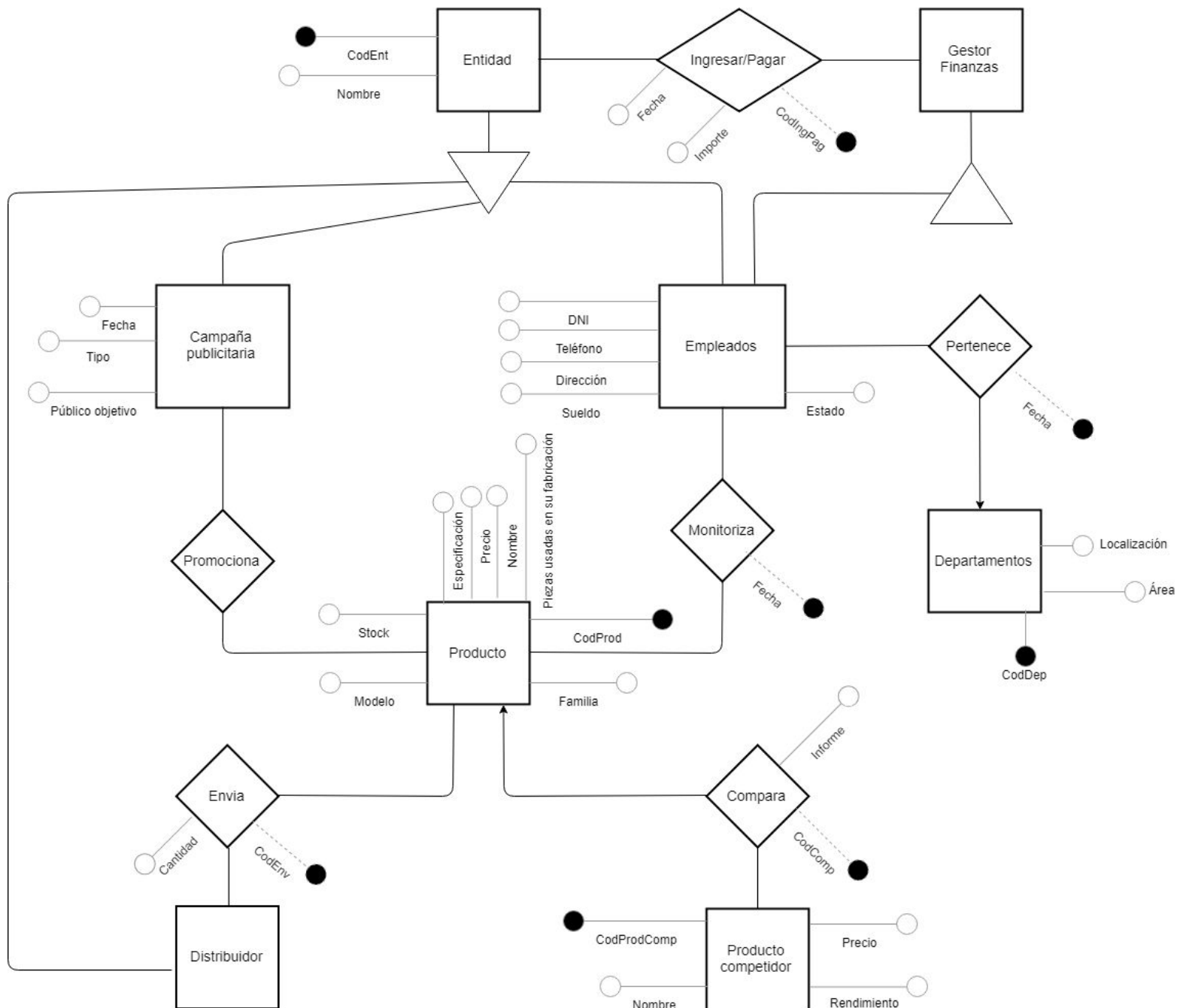


Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

Diagrama de flujo de nivel 2 – Recursos humanos



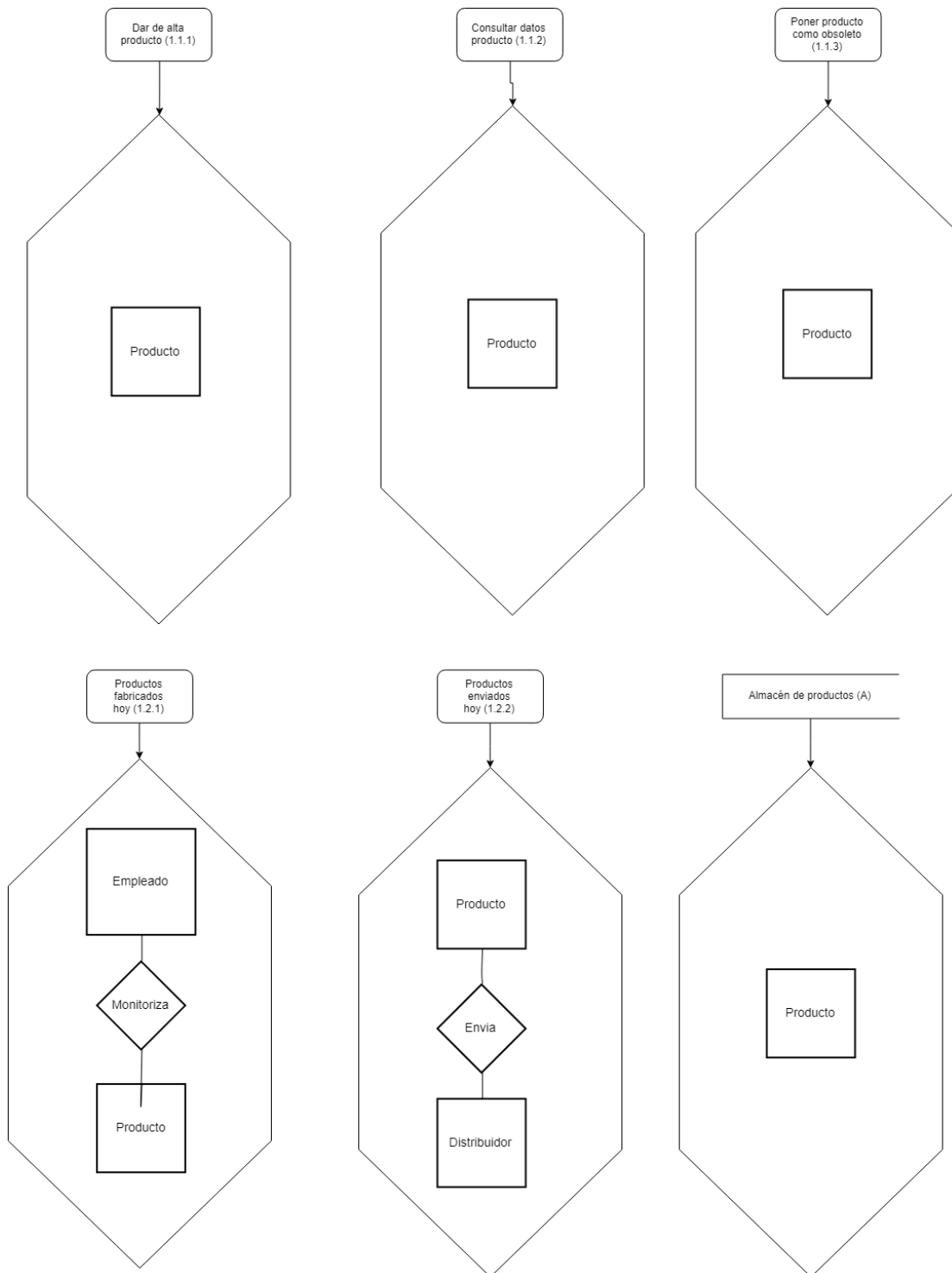
Esquema funcional



Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

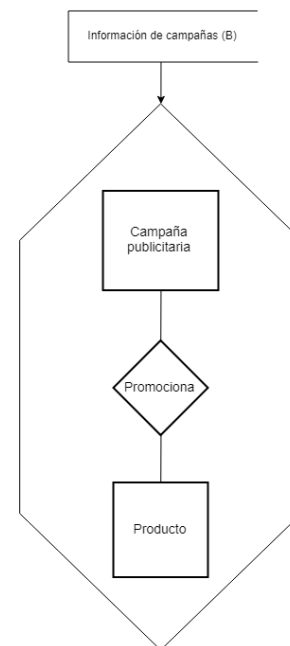
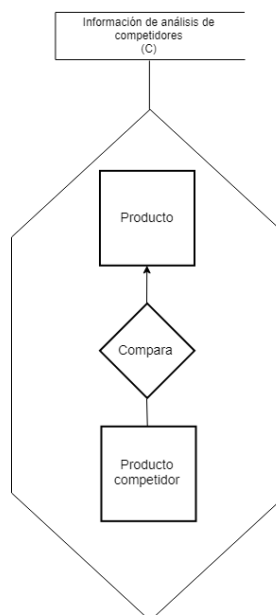
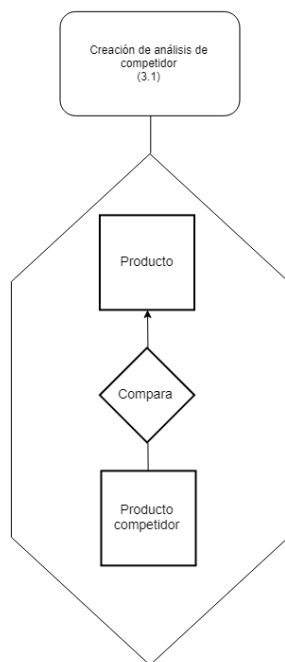
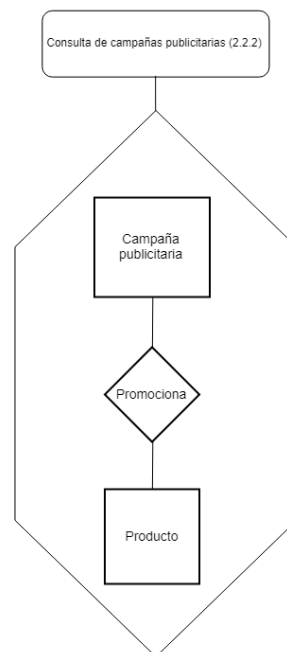
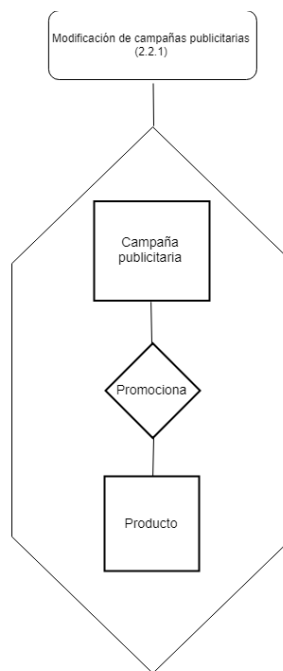
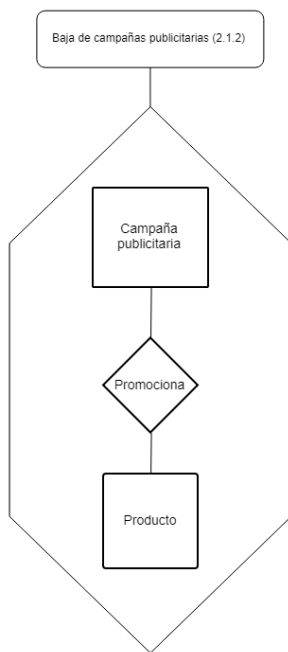
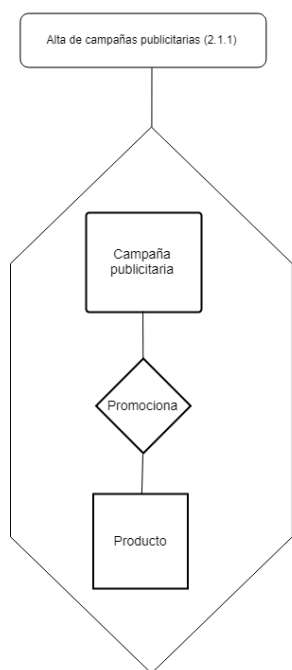
Esquemas externos – Producción

Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

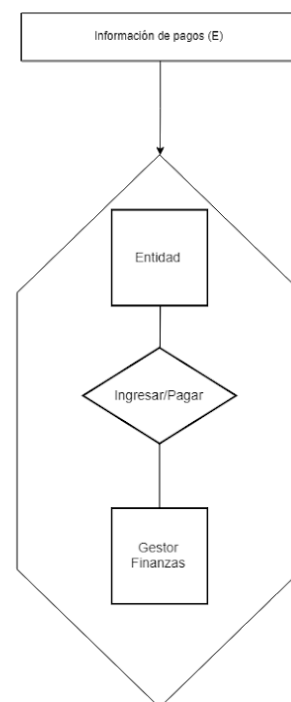
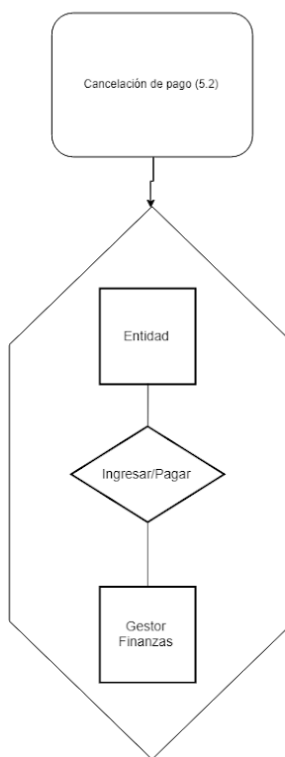
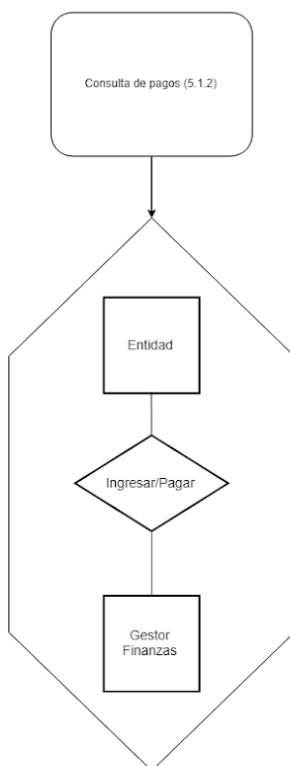
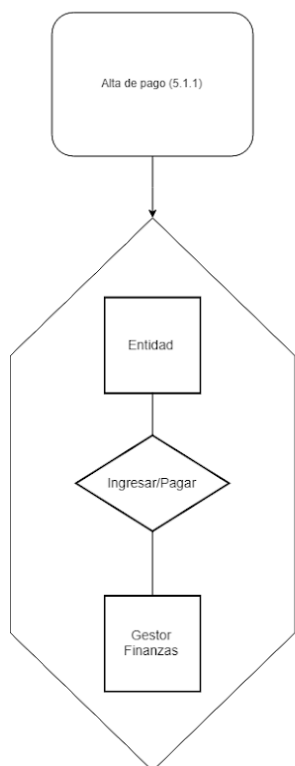
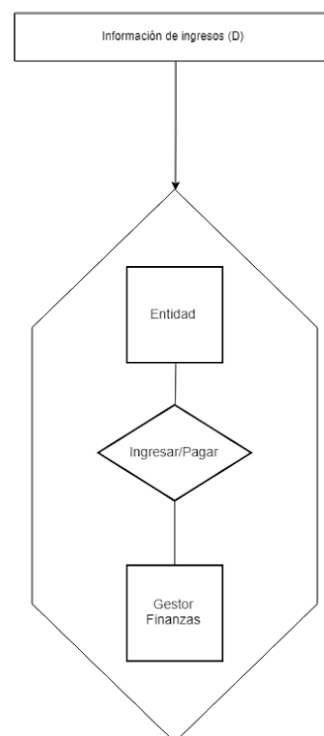
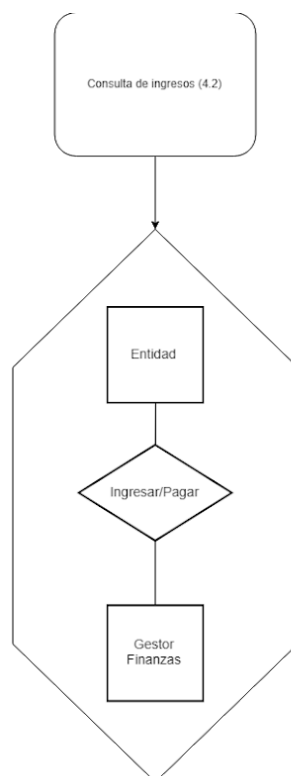
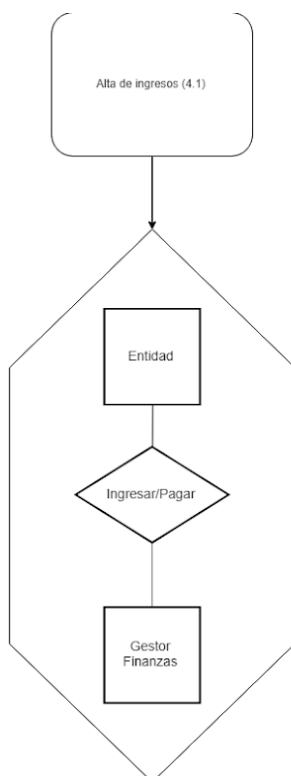


Esquemas externos – Marketing

Ignacio Barragán Lozano
 Francisco José Cotán López
 Juan Manuel Rubio Rodríguez
 Ignacio Vellido Expósito

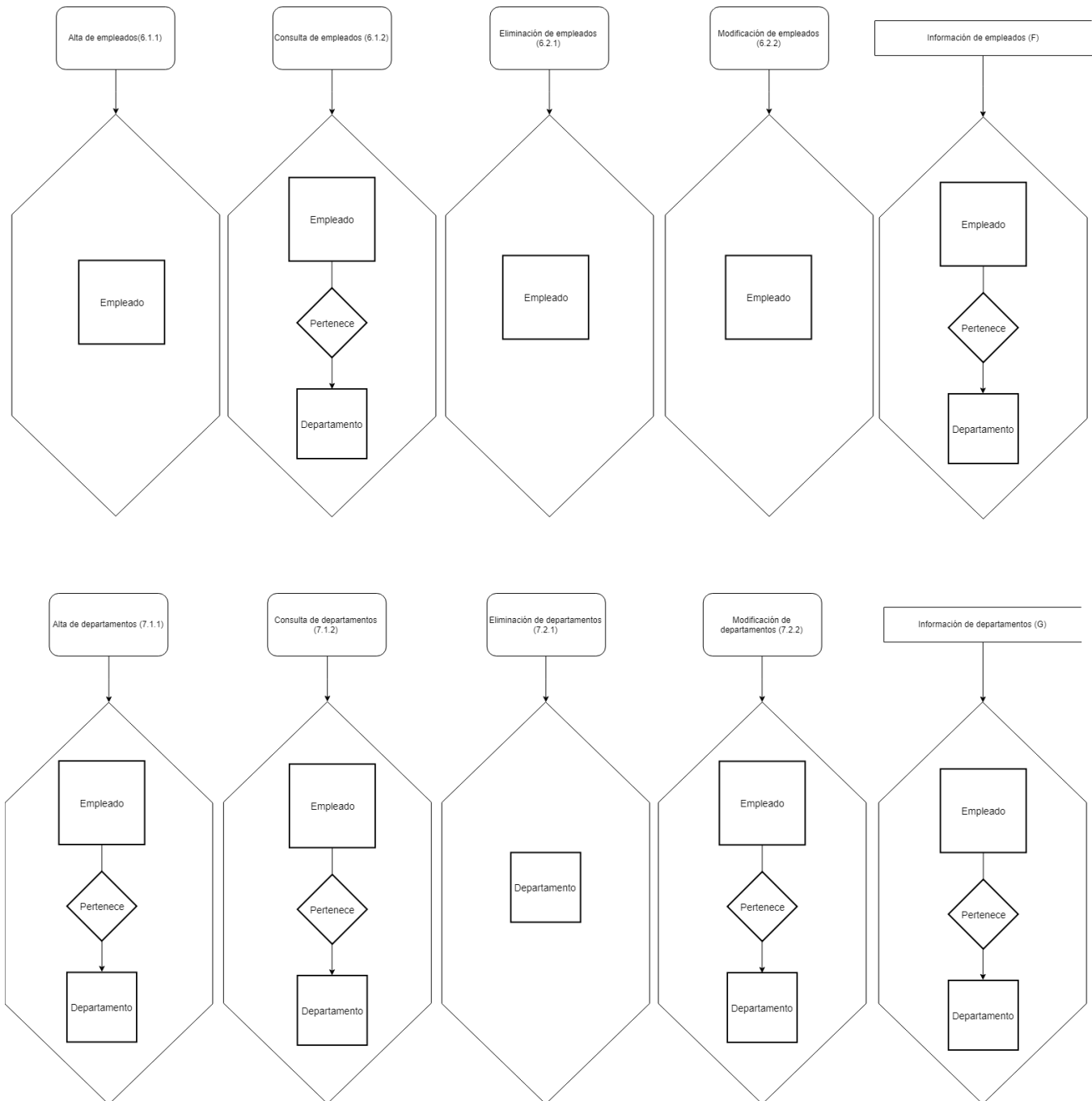


Esquemas externos – Finanzas



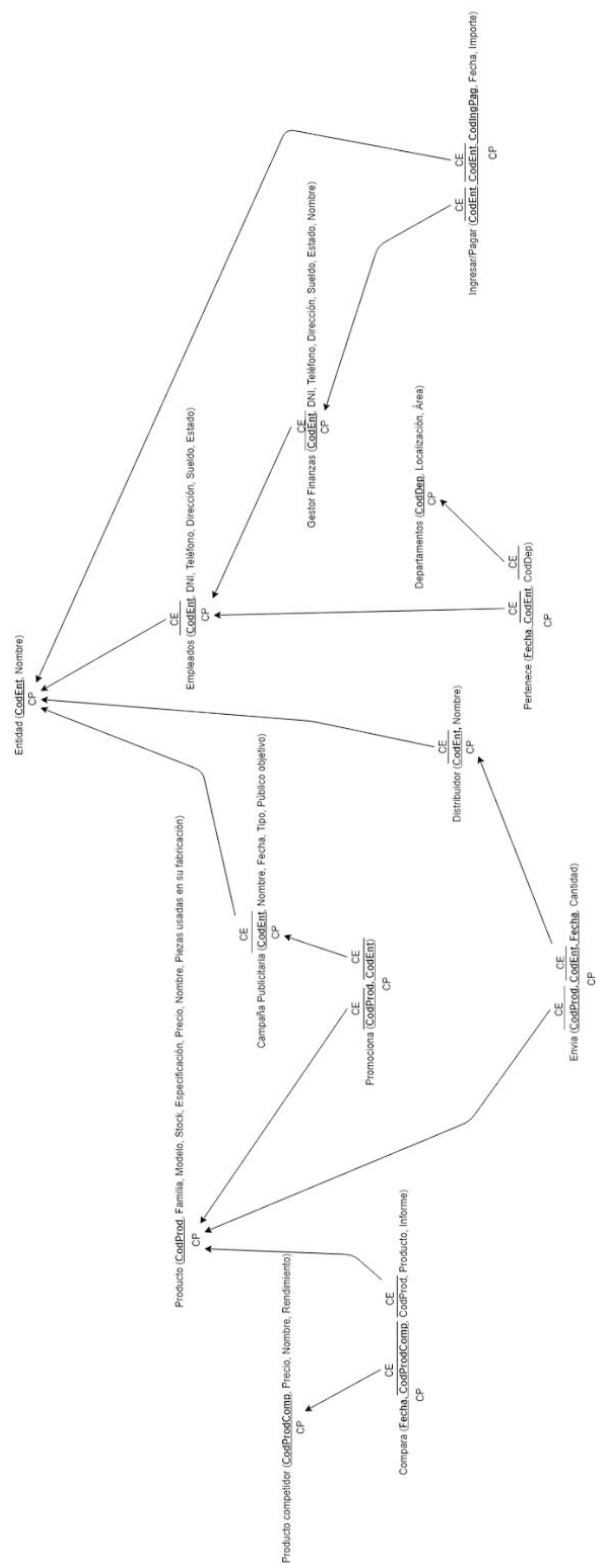
Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

Esquemas externos – Recursos humanos



Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

Paso a tablas



Implementación – Creación de tablas

Comunes

```
1 CREATE TABLE Entidad (  
2     CodEnt     INT PRIMARY KEY,  
3     Nombre     VARCHAR(30)  
4 );
```

Producción

```
1 CREATE TABLE Producto (  
2     CodProd     INT PRIMARY KEY,  
3     Nombre      VARCHAR(20) NOT NULL,  
4     Familia     VARCHAR(20) NOT NULL,  
5     Modelo      VARCHAR(20) NOT NULL,  
6     Especificaciones VARCHAR(60),  
7     Piezas      VARCHAR(50),  
8     Stock       INT,  
9     Precio      INT  
10 );
```

```
1 CREATE TABLE Distribuidor (  
2     CodEnt       INT NOT NULL,  
3     Nombre       VARCHAR(20),  
4  
5     PRIMARY KEY (CodEnt),  
6     CONSTRAINT CodEnt_fk  
7     FOREIGN KEY (CodEnt)  
8     REFERENCES Entidad(CodEnt)  
9     ON DELETE CASCADE  
10 );
```

```

1  CREATE TABLE Envia (
2      CodProd INT NOT NULL,
3      CodEnt INT NOT NULL,
4      Fecha VARCHAR(10) NOT NULL,
5      Cantidad INT NOT NULL,
6
7      PRIMARY KEY (CodProd, CodEnt, Fecha),
8      CONSTRAINT CodProd_fk
9          FOREIGN KEY (CodProd)
10         REFERENCES Producto(CodProd)
11         ON DELETE CASCADE,
12     CONSTRAINT CodEnt_Envia_fk
13         FOREIGN KEY (CodEnt)
14         REFERENCES Distribuidor(CodEnt)
15         ON DELETE CASCADE
16 );

```

Marketing

```

1  CREATE TABLE CampaniaPublicitaria (
2      CodEnt INT,
3      Tipo VARCHAR(50) NOT NULL,
4      PublicoObjetivo VARCHAR(50) NOT NULL,
5      primary key (CodEnt),
6      foreign key (CodEnt) references Entidad(CodEnt) on delete cascade
7  );

```

```
1 CREATE TABLE Compara (  
2     CodProdComp INT NOT NULL,  
3     CodProd INT NOT NULL,  
4     CodComp INT NOT NULL,  
5     Informe VARCHAR(100),  
6     PRIMARY KEY (CodProdComp, CodProd, CodComp),  
7     foreign key (CodProdComp) references ProductoCompetidor(CodProdComp),  
8     foreign key (CodProd) references Producto(CodProd)  
9 );
```

```
1 CREATE TABLE ProductoCompetidor (  
2     CodProdComp INT PRIMARY KEY,  
3     Precio REAL NOT NULL,  
4     Nombre VARCHAR(50) NOT NULL,  
5     Rendimiento REAL  
6 );
```

```
1 CREATE TABLE Promociona (  
2     CodEnt INT NOT NULL,  
3     CodProd INT NOT NULL,  
4     foreign key (CodEnt) references CampanaPublicitaria(CodEnt),  
5     foreign key (CodProd) references Producto(CodProd)  
6 );
```


Finanzas

```
1 CREATE TABLE Gestor(  
2     CodEnt          INT          NOT NULL,  
3     Nombre          VARCHAR(30),  
4     DNI             VARCHAR(9)   NOT NULL,  
5     TELEFONO        VARCHAR(13)  NOT NULL,  
6     Direccion       VARCHAR(50),  
7     Sueldo          INT,  
8     Estado          CHAR(1),  
9  
10    PRIMARY KEY (CodEnt),  
11    FOREIGN KEY (CodEnt)  
12    REFERENCES Empleados(CodEnt)  
13  
14 );
```

```
1 CREATE TABLE Ingresarpagar (  
2     CodGest          INT          NOT NULL,  
3     CodEnt          INT          NOT NULL,  
4     CodIngPag        INT          NOT NULL,  
5     Fecha           VARCHAR(10)  NOT NULL,  
6     Importe         FLOAT,  
7  
8     PRIMARY KEY (CodEnt, CodIngPag),  
9     FOREIGN KEY (CodEnt)  
10    REFERENCES Entidad(CodEnt),  
11    FOREIGN KEY (CodGest)  
12    REFERENCES Gestor(CodEnt)  
13 );
```

Recursos humanos

```
1 CREATE TABLE Empleados (  
2     CodEnt     INT          NOT NULL,  
3     DNI        VARCHAR(9)   NOT NULL,  
4     Telefono   VARCHAR(13)  NOT NULL,  
5     Direccion  VARCHAR(50),  
6     Sueldo     INT,  
7     Estado     CHAR(1),  
8  
9     PRIMARY KEY (CodEnt),  
10    CONSTRAINT CodEnt_fk  
11        FOREIGN KEY (CodEnt)  
12            REFERENCES Entidad(CodEnt)  
13            ON DELETE CASCADE  
14 );  
  
1 CREATE TABLE Departamentos (  
2     CodDep      INT          PRIMARY KEY,  
3     Localizacion VARCHAR(50),  
4     Area        VARCHAR(30)  
5 );  
  
1 CREATE TABLE Pertenece (  
2     CodEnt INT          NOT NULL,  
3     CodDep INT          NOT NULL,  
4     Fecha  VARCHAR(10)  NOT NULL,  
5  
6     PRIMARY KEY (CodEnt, Fecha),  
7     CONSTRAINT CodEnt_Pertenece_fk  
8         FOREIGN KEY (CodEnt)  
9             REFERENCES Empleados(CodEnt)  
10            ON DELETE CASCADE,  
11    CONSTRAINT CodDep_fk  
12        FOREIGN KEY (CodDep)  
13            REFERENCES Departamentos(CodDep)  
14            ON DELETE CASCADE  
15 );
```

Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

Implementación – Insertado de tuplas

Comunes

```
1  INSERT INTO Entidad VALUES ('0','A'), ('1','B'), ('2','C'), ('3','D'),  
2  ('4','E'), ('5','F'), ('6','G'), ('7','H'), ('8','I'), ('9','J');
```

Producción

```
1  INSERT INTO Producto VALUES ('0','A', '000', '0A', '', '', 6, 100),  
2  ('1','B', '000', '0B', '', '', 8, 200),  
3  ('2','C', '000', '0C', '', '', 3, 300),  
4  ('3','D', '000', '0D', '', '', 5, 400),  
5  ('4','E', '000', '0E', '', '', 9, 500),  
6  ('5','F', '111', '1A', '', '', 0, 600),  
7  ('6','G', '111', '1B', '', '', 0, 700),  
8  ('7','H', '111', '1C', '', '', 0, 800),  
9  ('8','I', '111', '1D', '', '', 0, 900),  
10 ('9','J', '111', '1E', '', '', 0, 1000);
```

```
1  INSERT INTO Distribuidor VALUES ('20','A'),  
2  ('21','B'),  
3  ('22','C'),  
4  ('23','D'),  
5  ('24','E'),  
6  ('25','F'),  
7  ('26','G'),  
8  ('27','H'),  
9  ('28','I'),  
10 ('29','J');
```

```
1  INSERT INTO Envia VALUES (0, '1', '10-11-2018', 6),  
2  (1, '2', '10-11-2018', 8),  
3  (2, '6', '10-11-2018', 3),  
4  (3, '1', '10-11-2018', 5),  
5  (4, '6', '10-11-2018', 9);
```

Marketing

```
1 INSERT INTO CampaniaPublicitaria VALUES ('10','A', 'A'),
2 ('11','B', 'B'), ('12','C', 'C'), ('13','D', 'D'),
3 ('14','E', 'E'), ('15','F', 'F'), ('16','G', 'G'),
4 ('17','H', 'H'), ('18','I', 'I'), ('19','J', 'J');
```

```
1 INSERT INTO Compara VALUES ('0','0', '0', null), ('1','1', '1', null),
2 ('2','2', '2', null), ('3','3', '3', null), ('4','4', '4', null),
3 ('5','5', '5', null), ('6','6', '6', null), ('7','7', '7', null),
4 ('8','8', '8', null), ('9','9', '9', null);
```

```
1 INSERT INTO ProductoCompetidor VALUES ('0', '1.2', 'A', null),
2 ('1', '2.2', 'B', null), ('2', '3.2', 'C', null), ('3', '4.2', 'D', null),
3 ('4', '5.2', 'E', null), ('5', '6.2', 'F', '0.5'), ('6', '7.2', 'G', '0.9'),
4 ('7', '8.2', 'H', '0.2'), ('8', '9.2', 'I', '0.3'), ('9', '10.2', 'J', '0.8');
```

```
1 INSERT INTO Promociona VALUES ('0', '0'), ('1', '1'),
2 ('2', '2'), ('3', '3'), ('4', '4'), ('5', '5'),
3 ('6', '6'), ('7', '7'), ('8', '8'), ('9', '9');
```

Finanzas

```
1 INSERT INTO Gestor VALUES ('0','A', '00000000A', '000-000-000', 'AAAA', 0, 'N'),
2 ('1','B', '11111111B', '111-111-111', 'BBBB', 1, 'N'),
3 ('2','C', '22222222C', '222-222-222', 'CCCC', 2, 'N'),
4 ('3','D', '33333333D', '333-333-333', 'DDDD', 3, 'N'),
5 ('4','E', '44444444E', '444-444-444', 'EEEE', 4, 'N'),
6 ('5','F', '55555555F', '555-555-555', 'FFFF', 5, 'N'),
7 ('6','G', '66666666G', '666-666-666', 'GGGG', 6, 'N'),
8 ('7','H', '77777777H', '777-777-777', 'HHHH', 7, 'N'),
9 ('8','I', '88888888I', '888-888-888', 'IIII', 8, 'N'),
10 ('9','J', '99999999J', '999-999-999', 'JJJJ', 9, 'N');
```

```
1 INSERT INTO Ingresarpagar VALUES ('0', '1', '1', '10-10-2010', '1000'),
2 ('0', '2', '2', '11-10-2010', '-1300'),
3 ('0', '3', '3', '12-10-2010', '1500'),
4 ('0', '4', '4', '13-10-2010', '-1010'),
5 ('0', '3', '5', '13-10-2010', '1800');
```


Recursos humanos

```
1  INSERT INTO Empleados VALUES
2  ('0', '00000000A', '000-000-000', 'AAAA', 0, 'N'),
3  ('1', '11111111B', '111-111-111', 'BBBB', 1, 'N'),
4  ('2', '22222222C', '222-222-222', 'CCCC', 2, 'N'),
5  ('3', '33333333D', '333-333-333', 'DDDD', 3, 'N'),
6  ('4', '44444444E', '444-444-444', 'EEEE', 4, 'N'),
7  ('5', '55555555F', '555-555-555', 'FFFF', 5, 'N'),
8  ('6', '66666666G', '666-666-666', 'GGGG', 6, 'N'),
9  ('7', '77777777H', '777-777-777', 'HHHH', 7, 'N'),
10 ('8', '88888888I', '888-888-888', 'IIII', 8, 'N'),
11 ('9', '99999999J', '999-999-999', 'JJJJ', 9, 'N');
```

```
1  INSERT INTO Departamentos VALUES ('0','A', 'Spain'), ('1','B', 'Spain')
2  ,('2','C', 'Spain'), ('3','D', 'Spain'), ('4','E', 'Spain'), ('5','F', 'Spain')
3  ,('6','G', 'Spain'), ('7','H', 'Spain'), ('8','I', 'Spain'), ('9','J', 'Spain');
```

```
1  INSERT INTO Pertenece VALUES
2  ('0', '1', CONVERT(CURRENT_DATE(), CHAR(10))),
3  ('1', '2', CONVERT(CURRENT_DATE(), CHAR(10))),
4  ('2', '6', CONVERT(CURRENT_DATE(), CHAR(10))),
5  ('3', '8', CONVERT(CURRENT_DATE(), CHAR(10))),
6  ('4', '8', CONVERT(CURRENT_DATE(), CHAR(10)));
```

Implementación – Listado de tablas

Producción

Marketing

Finanzas

Recursos humanos

Implementación – Diseño de cursores

Producción

```
1  /* Cursor de producción - Ignacio Barragán Lozano */
2
3  -- Listado de productos enviados a un distribuidor y su familia --
4
5  DELIMITER $$
6
7  CREATE PROCEDURE cursorEnviados (INOUT listaEnviados varchar(4000))
8  BEGIN
9
10     DECLARE NombreProducto VARCHAR(20);
11     DECLARE FamiliaProducto VARCHAR(20);
12     DECLARE fin INTEGER DEFAULT 0;
13
14     DECLARE cProductos CURSOR FOR
15         SELECT Producto.Nombre, Producto.Familia
16         FROM Producto, Envia
17         WHERE Envia.CodEnt = 1 AND Envia.CodProd = Producto.CodProd;
18
19     DECLARE CONTINUE HANDLER FOR NOT FOUND SET fin = TRUE;
20
21     OPEN cProductos;
22
23     bucle: LOOP
24         FETCH cProductos INTO NombreProducto, FamiliaProducto;
25         IF fin = 1 THEN
26             LEAVE bucle;
27         END IF;
28
29         SET listaEnviados = CONCAT(listaEnviados, NombreProducto, " de la Familia: ", FamiliaProducto, "<br\>");
30     END LOOP;
31
32
33     CLOSE cProductos;
34 END$$
35
36 DELIMITER ;
```

Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

Marketing

```
38 DELIMITER $$
39
40 CREATE PROCEDURE cursorFran (INOUT listaNombres varchar(4000))
41 BEGIN
42     declare nombreProductoCompetidor varchar(50);
43     declare nombreProducto varchar(50);
44     declare fin integer default 0;
45     declare nombres CURSOR for
46         select Producto.Nombre, ProductoCompetidor.Nombre from ProductoCompetidor, Producto, Compara
47         where Compara.CodProdComp = ProductoCompetidor.CodProdComp and Compara.CodProd = Producto.CodProd;
48
49     -- Equivalente a %found en PL/SQL --
50     DECLARE CONTINUE HANDLER FOR NOT FOUND SET fin = TRUE;
51
52     OPEN nombres;
53
54     bucle: loop
55         fetch nombres into nombreProducto, nombreProductoCompetidor;
56
57         IF fin = 1 THEN
58             LEAVE bucle;
59         END IF;
60
61         SET listaNombres =
62             CONCAT(listaNombres, "Producto: ", nombreProducto, " y Competidor: ",
63                 nombreProductoCompetidor, "<br\\>");
64     end loop;
65
66     close nombres;
67 end$$
68
69 DELIMITER ;
```

Finanzas

Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

```

1  /* Cursor de finanzas - Juan Manuel Rubio Rodriguez */
2
3  -- Listado de gestores de finanzas y su DNI --
4
5  DELIMITER $$
6
7  CREATE PROCEDURE cursorFinanzas (INOUT listaGestor varchar(4000))
8  BEGIN
9      -- Declaraciones --
10
11     DECLARE NombreGestor VARCHAR(30);
12     DECLARE DNIGestor VARCHAR(9);
13     DECLARE fin INTEGER DEFAULT 0;          -- Variable %found
14
15     DECLARE cGestor CURSOR FOR
16         SELECT Entidad.Nombre, Empleados.DNI FROM Empleados, Gestor, Entidad
17         WHERE Gestor.CodEnt = Empleados.CodEnt AND Entidad.CodEnt = Empleados.CodEnt;
18
19
20     -- Equivalente a %found en PL/SQL --
21     DECLARE CONTINUE HANDLER FOR NOT FOUND SET fin = 1;
22
23     -- Ejecución --
24
25     OPEN cGestor;
26
27     bucle: LOOP
28         FETCH cGestor INTO NombreGestor, DNIGestor;
29
30         IF fin = 1 THEN
31             LEAVE bucle;
32         END IF;
33
34         SET listaGestor = CONCAT(listaGestor, NombreGestor, " con DNI: ", DNIGestor, "<\br>");
35     END LOOP;
36
37     CLOSE cGestor;
38
39     END$$
40
41     DELIMITER ;

```

Ignacio Barragán Lozano
 Francisco José Cotán López
 Juan Manuel Rubio Rodríguez
 Ignacio Vellido Expósito

Recursos humanos

Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

```

3  -- Listado de empleados pertenecientes a un departamento y su DNI --
4  -- Se pasa una lista por argumento a la que se le añaden los empleados --
5
6  DELIMITER $$ /* Para que se ejecute el bloque al completo */
7
8  CREATE PROCEDURE cursorRH (INOUT listaEmpleados varchar(4000))
9  BEGIN
10     -- Declaraciones --
11
12     DECLARE NombreEmpleado VARCHAR(30);
13     DECLARE DNIEmpleado VARCHAR(9);
14     DECLARE fin INTEGER DEFAULT 0;          -- Variable %found
15
16     DECLARE cEmpleados CURSOR FOR
17         SELECT Entidad.Nombre, Empleados.DNI FROM Empleados, Pertenece, Entidad
18         WHERE Pertenece.CodDep = '1' AND Pertenece.CodEnt = Empleados.CodEnt
19             AND Entidad.CodEnt = Empleados.CodEnt;
20
21
22     -- Equivalente a %found en PL/SQL --
23     DECLARE CONTINUE HANDLER FOR NOT FOUND SET fin = TRUE;
24
25     -- Ejecución --
26
27     OPEN cEmpleados;
28
29     bucle: LOOP
30         FETCH cEmpleados INTO NombreEmpleado, DNIEmpleado;
31
32         IF fin = 1 THEN
33             LEAVE bucle;
34         END IF;
35
36         SET listaEmpleados = CONCAT(listaEmpleados, NombreEmpleado, " con DNI: ", DNIEmpleado, "<br\>");
37     END LOOP;
38
39
40     CLOSE cEmpleados;
41 END$$
42
43 DELIMITER ;

```

Ignacio Barragán Lozano
 Francisco José Cotán López
 Juan Manuel Rubio Rodríguez
 Ignacio Vellido Expósito

Implementación – Diseño de disparadores

Producción

```
1  /* Disparador de producción - Ignacio Barragán Lozano */
2
3  /* Modificar el stock de un producto cuando este se envía a un distribuidor */
4
5  CREATE TRIGGER modificarStock BEFORE INSERT ON Envía
6  FOR EACH ROW
7  DECLARE
8      tStock INTEGER;
9  BEGIN
10     SELECT Stock INTO tStock
11     FROM Producto
12     WHERE CodProd = :new.CodProd;
13
14     IF tStock >= :new.Cantidad THEN
15         UPDATE Producto
16         SET Stock = tStock - :new.Cantidad
17         WHERE CodProd = :new.CodProd;
18     ELSE
19         UPDATE Producto
20         SET Stock = 0
21         WHERE CodProd = :new.CodProd;
22     END IF;
23 END;
```

Marketing

```
10  -- Rebaja el precio de un producto si el de competidor es más barato --
11
12  CREATE TRIGGER rebaja BEFORE INSERT ON Compara
13  FOR EACH ROW
14      UPDATE Producto, ProductoCompetidor SET Producto.Precio = ProductoCompetidor.Precio
15      WHERE NEW.CodProd = Producto.CodProd AND NEW.CodProdComp = ProductoCompetidor.CodProdComp
16             AND Producto.Precio > ProductoCompetidor.Precio;
17
```

Finanzas

Ignacio Barragán Lozano
Francisco José Cotán López
Juan Manuel Rubio Rodríguez
Ignacio Vellido Expósito

```

1  /* Disparador de Finanzas - Juan Manuel Rubio Rodr\'edguez */
2
3  /* Insertar la fecha actual en caso de que no se introduzca una */
4
5  CREATE TRIGGER prima BEFORE INSERT ON Ingresarpagar
6  FOR EACH ROW BEGIN
7      IF NEW.Importe > 1000 THEN
8          UPDATE Empleados SET Sueldo = Sueldo + 100
9          WHERE CodEnt = NEW.CodGest;
10     END IF;
11 END
12

```

Recursos humanos

```

3  -- Aumentar salario a los empleados que entren en el departamento 1 --
4
5  CREATE TRIGGER subirSueldo BEFORE INSERT ON Pertenece
6  FOR EACH ROW
7      UPDATE Empleados SET Sueldo = Sueldo + 500
8      WHERE CodEnt = NEW.CodEnt AND NEW.CodDep = '1';

```
