

# Aorist: Sistema de información orientado a la publicidad

## Documentación del proyecto

Óscar Bermúdez Garrido      Luis Castro Martín      Francisco David Charte Luque  
José Carlos Entrena Jiménez      José Ramón Trillo Vilchez

## Índice

<b>1. Descripción</b>	<b>2</b>
1.1. Áreas funcionales . . . . .	2
<b>2. Requisitos</b>	<b>3</b>
2.1. Requisitos funcionales . . . . .	3
2.2. Requisitos de datos . . . . .	4
2.3. Restricciones semánticas . . . . .	5
<b>3. Diseño de la aplicación</b>	<b>6</b>
3.1. Diagramas de flujos de datos y esquemas externos . . . . .	6
3.2. Diagrama conceptual . . . . .	12
3.3. Operaciones de datos . . . . .	13
3.4. Diseño lógico . . . . .	14
<b>4. Implementación</b>	<b>16</b>
4.1. Tecnologías utilizadas . . . . .	16
4.2. Guía de instalación . . . . .	16
4.3. Arquitectura de la implementación . . . . .	17
4.4. Uso de Aorist . . . . .	18

## 1. Descripción

Aorist (Ad Oriented Information System) es un sistema de gestión dirigido a empresas de publicidad, principalmente orientado a la publicidad online. El sistema modela las empresas anunciantes y los medios de forma que se puedan relacionar mediante los anuncios de las primeras y los espacios publicitarios de los segundos. Además, se facilita la segmentación del mercado y la asociación entre anunciantes y medios con audiencias afines.

### 1.1. Áreas funcionales

- Gestión de anunciantes: Permite dar de alta empresas que quieran promocionar sus productos, relacionarlas con tipos de audiencia y la creación de contratos con dichas empresas.
- Gestión de medios: Permite dar de alta y gestionar los medios donde se visualizarán los anuncios, por ejemplo: sitios web, aplicaciones móviles, revistas. Cada medio puede ir relacionado con ciertos tipos de audiencia y se pueden crear contratos asociados.
- Anuncios y espacios: Incluye la creación de distintos tipos de espacios que después se podrán ocupar mediante anuncios asociados a los anunciantes.
- Transacciones económicas: Permite tramitar los cobros a las empresas anunciantes, y los pagos a los medios.
- Audiencias: Administra los distintos perfiles de visitantes que se ofertan a los anunciantes y en función de los cuáles se distribuyen los anuncios a los medios.

## **2. Requisitos**

### **2.1. Requisitos funcionales**

#### **2.1.1. Funcionalidad “Gestión de anunciantes”**

Desarrollados en el documento asociado: *Requisitos de Gestión de Anunciantes*.

**RF1: Dar de alta una empresa anunciante**

**RF2: Dar de baja una empresa anunciante**

**RF3: Realizar contrato**

**RF4: Mostrar los datos de una empresa anunciante**

#### **2.1.2. Funcionalidad “Gestión de medios”**

Desarrollados en el documento asociado: *Requisitos de Gestión de Medios*.

**RF5: Dar de alta un nuevo medio**

**RF6: Dar de baja un medio**

**RF7: Mostrar los datos de un medio**

**RF8: Realizar contrato con un medio**

#### **2.1.3. Funcionalidad “Anuncios y espacios”**

Desarrollados en el documento asociado: *Requisitos de Anuncios*.

**RF9: Dar de alta nuevo espacio publicitario**

**RF10: Eliminar espacio publicitario**

**RF11: Dar de alta nuevo anuncio**

**RF12: Obtener anuncio**

**RF13: Modificar anuncio**

**RF14: Asignar anuncio a espacio publicitario**

**RF15: Mostrar anuncios puestos en marcha**

**RF16: Eliminar anuncio**

#### **2.1.4. Funcionalidad “Transacciones económicas”**

Desarrollados en el documento asociado: *Requisitos de Transacciones Económicas*.

**RF17: Hacer un pago a medio**

**RF18: Hacer un cobro a anunciante**

**RF19: Obtener balance**

**RF20: Obtener transacciones ordenadas por importe**

**RF21: Gasto por anunciante y medio**

**RF22: Gasto por contrato**

#### **2.1.5. Funcionalidad “Audiencia”**

Desarrollados en el documento asociado: *Requisitos de Audiencia*.

**RF23: Insertar un nuevo tipo de audiencia**

**RF24: Asociar un medio frecuentado por una audiencia**

**RF25: Asociar una audiencia preferida por un anunciante**

**RF26: Desasociar un medio frecuentado por una audiencia**

**RF27: Desasociar una audiencia preferida por un anunciante**

**RF28: Mostrar información sobre las audiencias disponibles**

#### **2.2. Requisitos de datos**

Desarrollados en *Requisitos de Gestión de Anunciantes*:

**RD1: Datos almacenados de empresa anunciante**

**RD2: Datos requeridos para registrar una empresa anunciante**

**RD3: Datos almacenados de un contrato con un anunciante**

Desarrollados en *Requisitos de Gestión de Medios*:

**RD4: Datos almacenados de Medio**

**RD5: Datos necesarios para almacenar un Medio**

**RD6: Datos almacenados de un contrato con un medio**

Desarrollados en *Requisitos de Anuncios*:

**RD7: Datos almacenados para Espacio publicitario**

**RD8: Datos almacenados para Anuncio**

**RD9: Datos requeridos para crear Espacio publicitario**

**RD10: Datos requeridos para crear Anuncio**

Desarrollados en *Requisitos de Transacciones Económicas*:

**RD11: Datos requeridos para pago a medio**

**RD12: Datos requeridos para cobro a anunciante**

**RD13: Datos almacenados de pago**

**RD14: Datos almacenados de cobro**

Desarrollados en *Requisitos de Audiencia*:

**RD15: Datos almacenados de Audiencia**

## **2.3. Restricciones semánticas**

Desarrolladas en *Requisitos de Gestión de Anunciantes*:

**RS1: Contratos**

**RS2: Fecha contratos**

Desarrolladas en *Requisitos de Gestión de Medios*:

**RS3: Contratos medio**

**RS4: Fecha contratos medio**

Desarrolladas en *Requisitos de Anuncios*:

**RS5: Espacios y anuncios**

**RS6: Anuncios y empresas**

**RS7: Anuncios y espacios incompatibles**

Desarrolladas en *Requisitos de Transacciones Económicas*

**RS8: Pagos**

**RS9: Cobros**

**RS10: Fecha**

Desarrolladas en *Requisitos de Audiencia*:

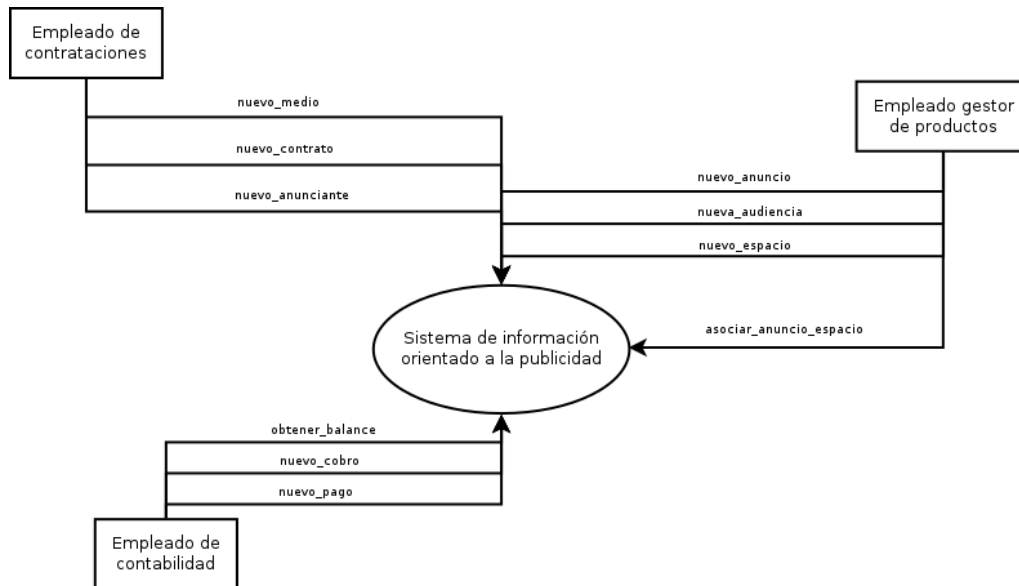
**RS11: Medios y anunciantes**

### 3. Diseño de la aplicación

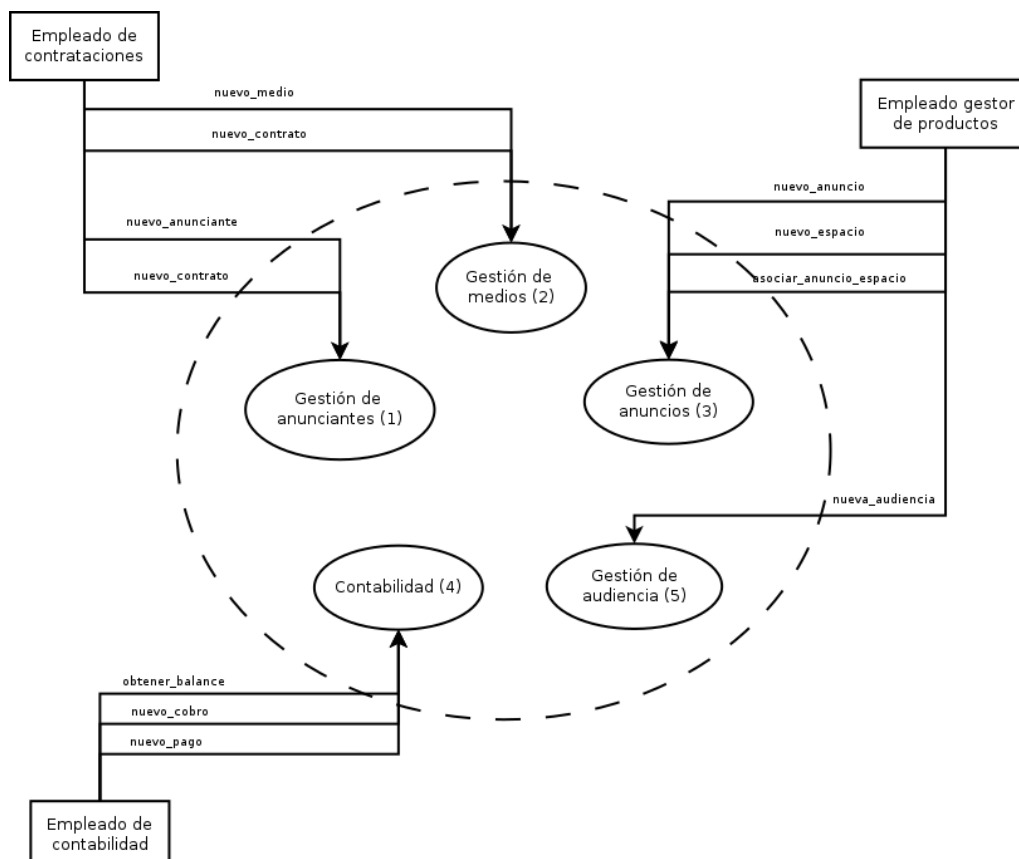
#### 3.1. Diagramas de flujos de datos y esquemas externos

##### 3.1.1. Armazón

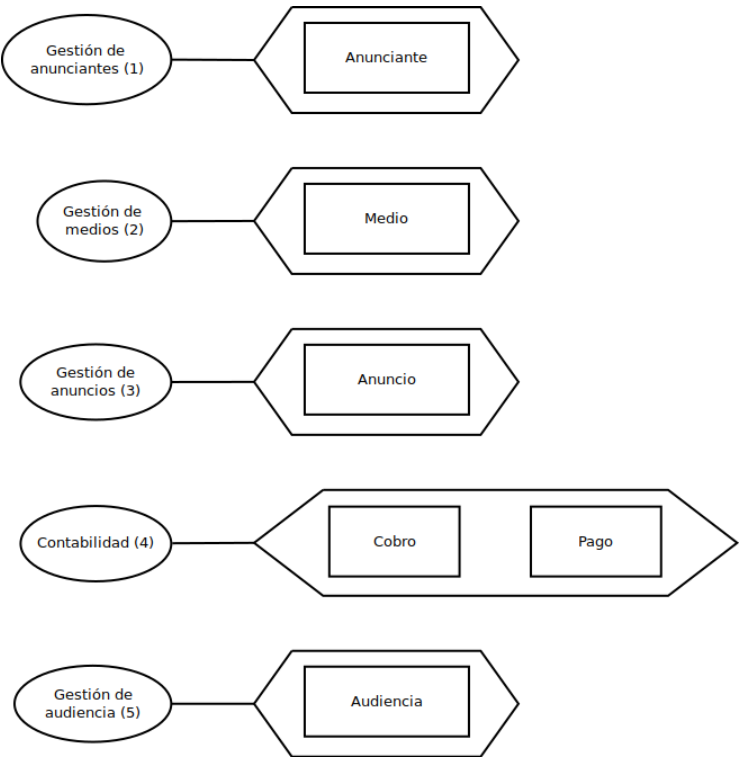
##### 3.1.1.1. Diagrama de caja negra



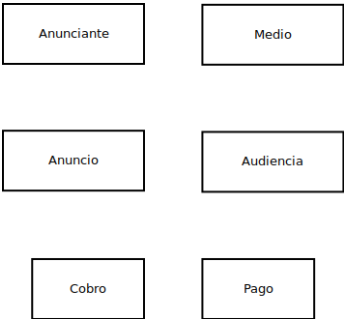
##### 3.1.1.2. Diagrama armazón F



3.1.1.3. Esquemas externos para el diagrama almacén F

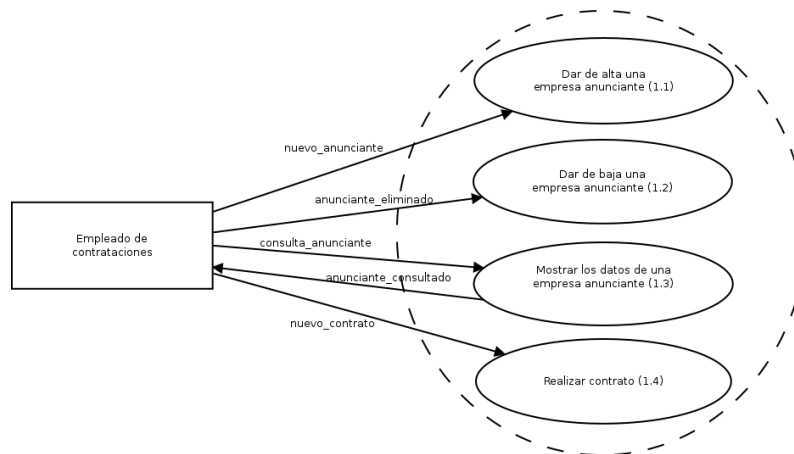


3.1.1.4. Diagrama almacén D

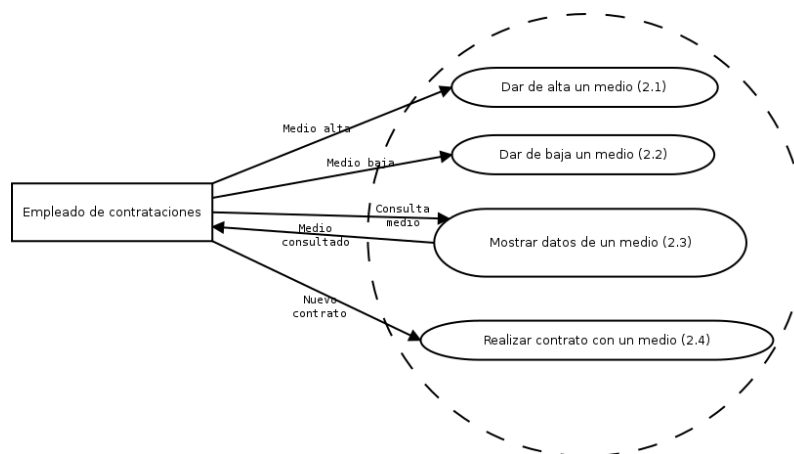


3.1.2. Primer refinamiento

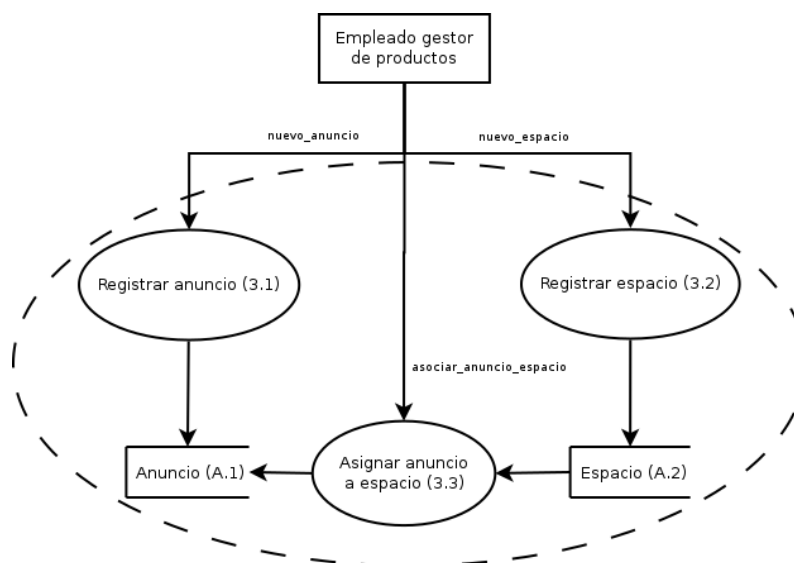
3.1.2.1. Refinamiento parcial del proceso 1



### 3.1.2.2. Refinamiento parcial del proceso 2

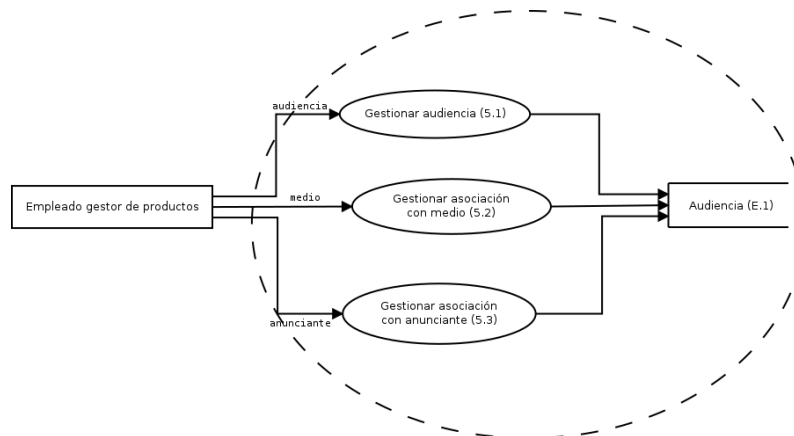


### 3.1.2.3. Refinamiento parcial del proceso 3

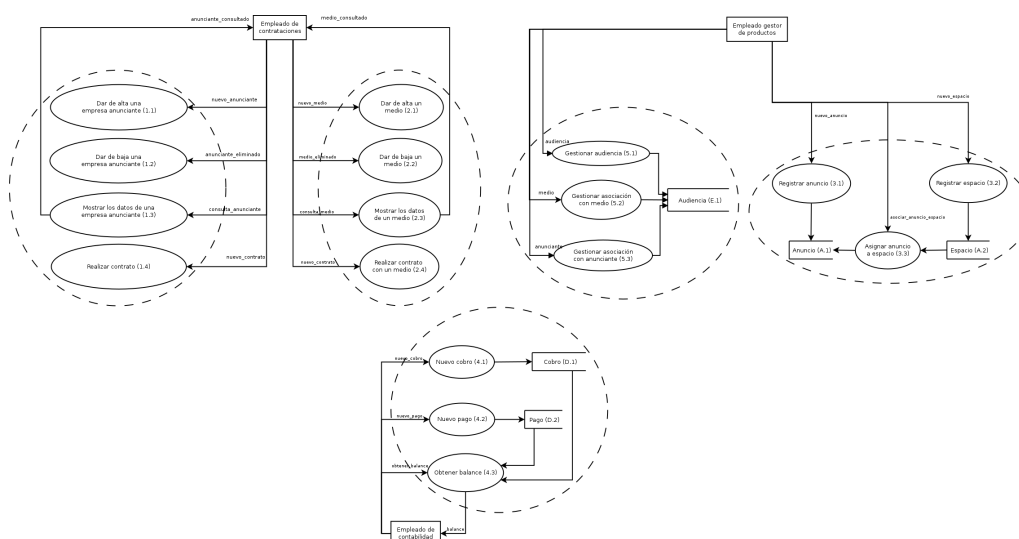


### 3.1.2.4. Refinamiento parcial del proceso 5

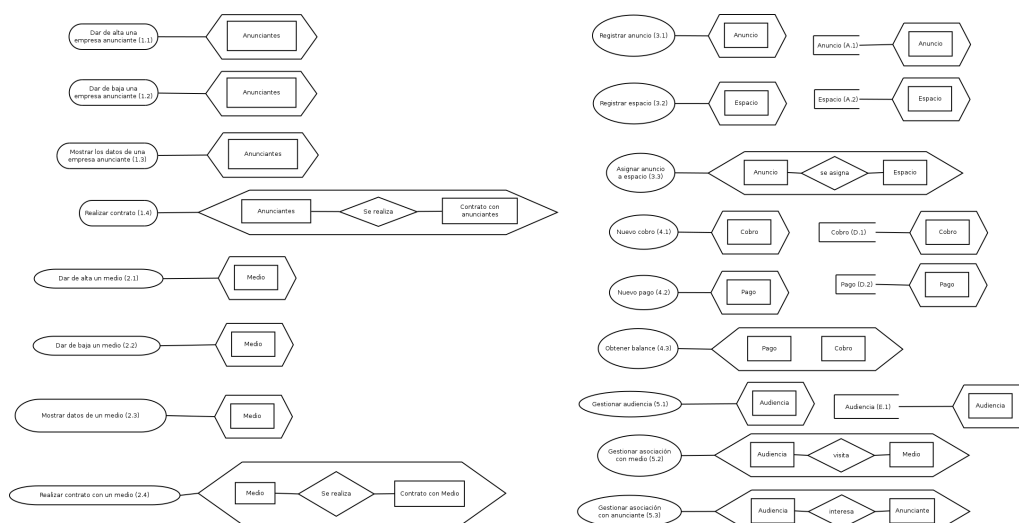




### 3.1.2.5. Plano de refinamiento F de primer nivel

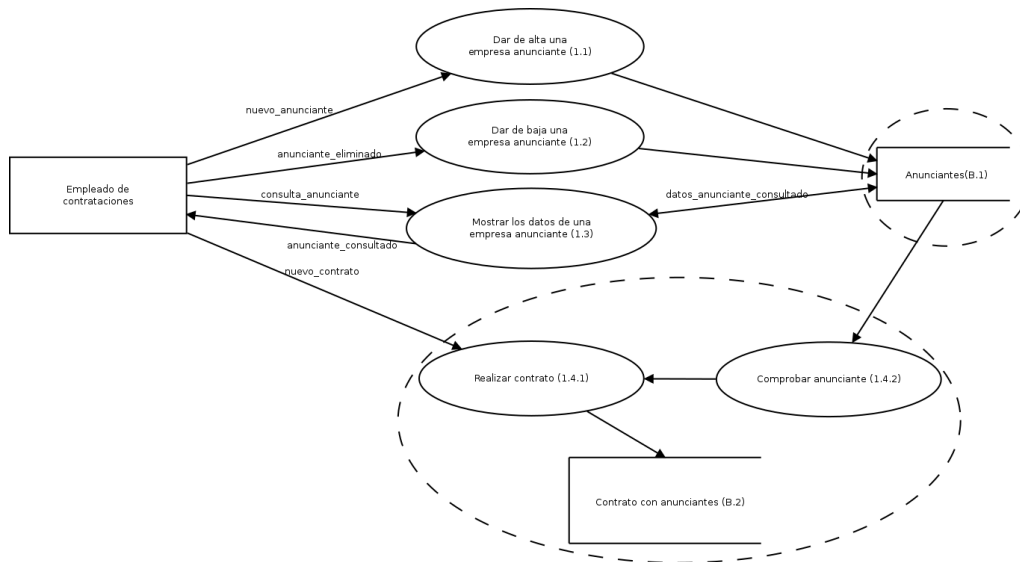


### 3.1.2.6. Esquemas externos del plano de refinamiento F

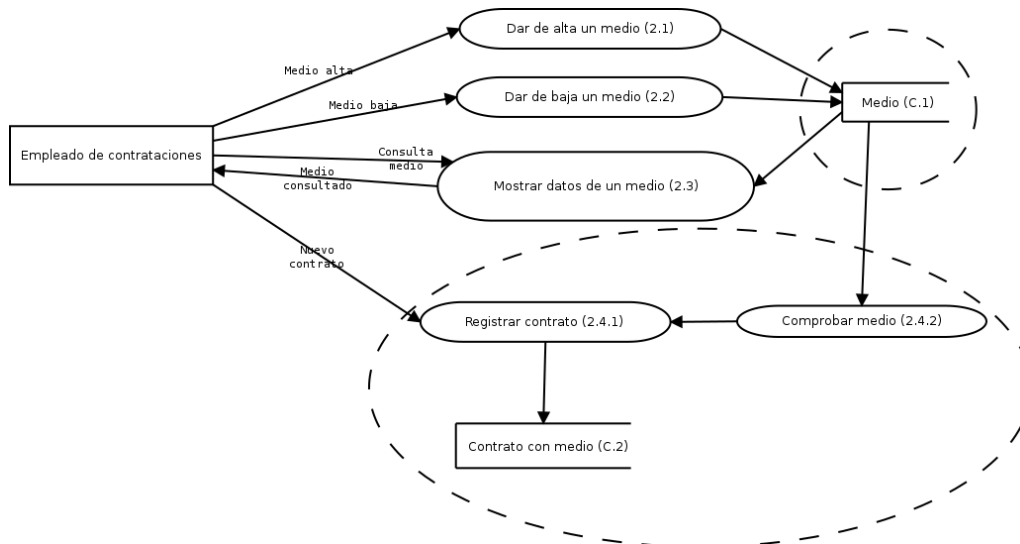


### 3.1.3. Segundo refinamiento

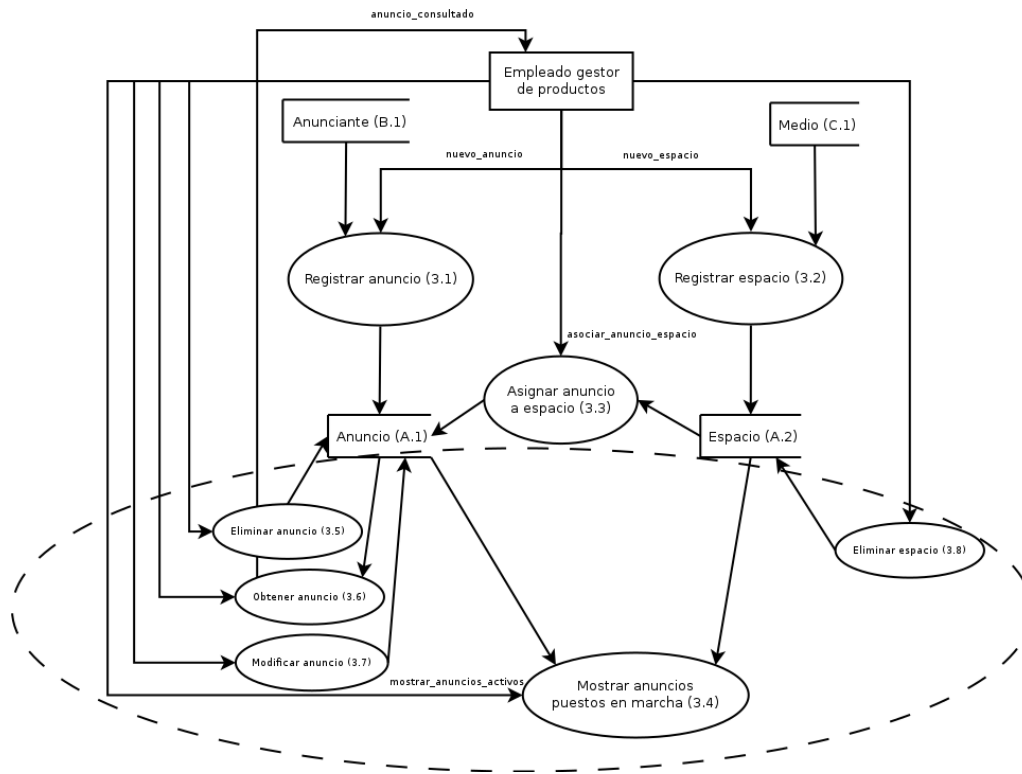
#### 3.1.3.1. Refinamiento parcial del proceso 1



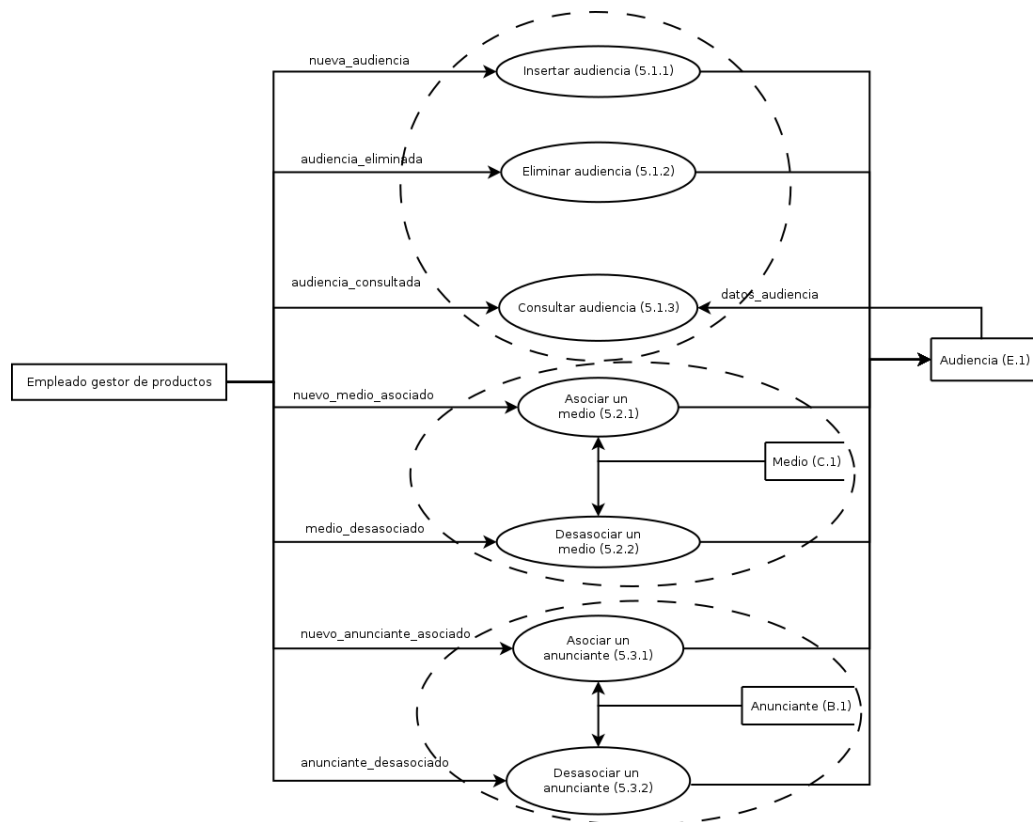
### 3.1.3.2. Refinamiento parcial del proceso 2



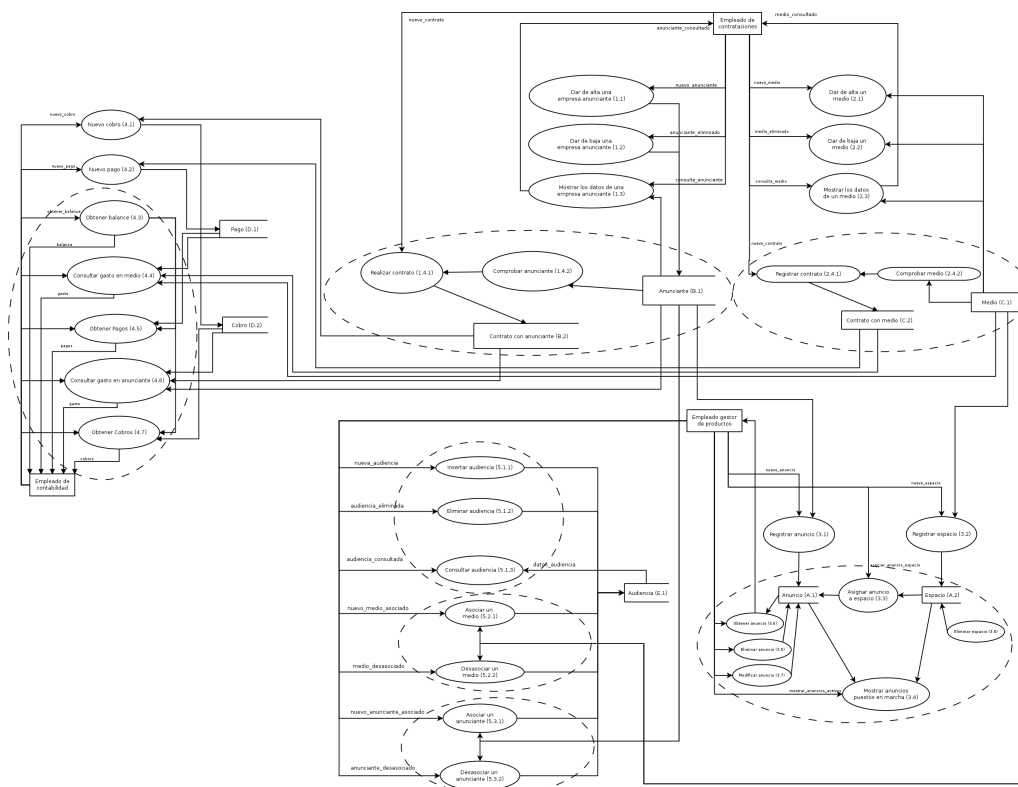
### 3.1.3.3. Refinamiento parcial del proceso 3



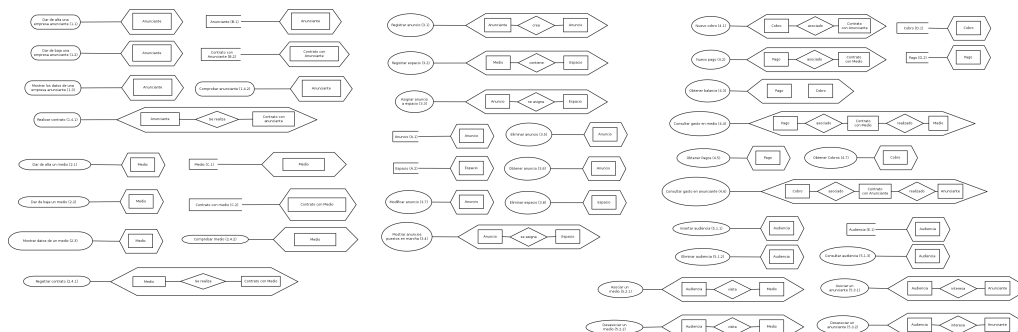
#### 3.1.3.4. Refinamiento parcial del proceso 5



#### 3.1.3.5. Plano de refinamiento F de segundo nivel

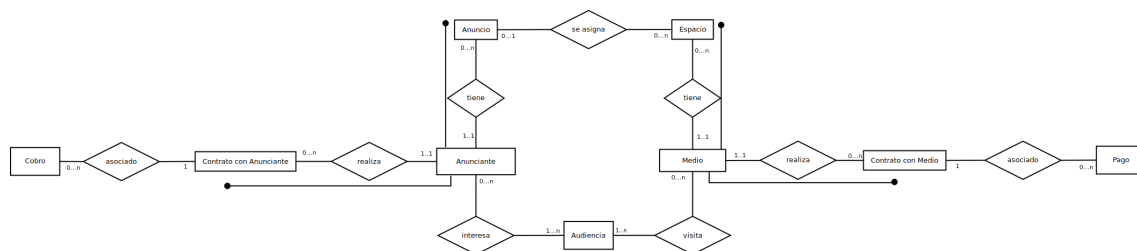


### 3.1.3.6. Esquemas externos del plano de refinamiento F

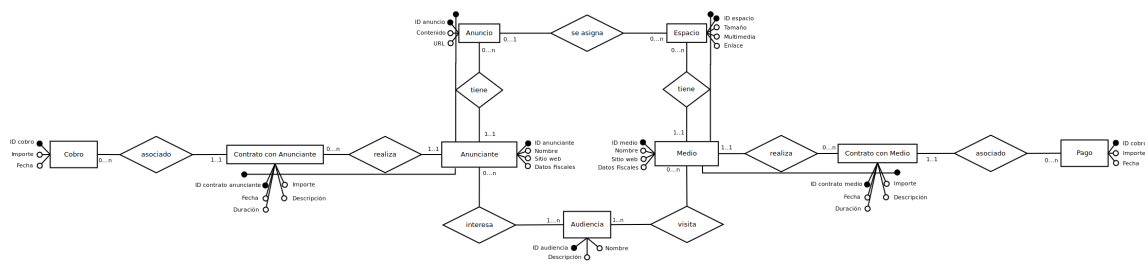


## 3.2. Diagrama conceptual

### 3.2.1. Unión de los esquemas externos



### 3.2.2. Diagrama resultante



### 3.3. Operaciones de datos

#### 3.3.1. Descripción de las operaciones

##### 3.3.1.1. O1: Alta de un medio

Insertar un nuevo medio a partir de su identificador, nombre, sitio web y datos fiscales.

##### 3.3.1.2. O2: Baja de un anunciante

Eliminar un anunciante dado su identificador.

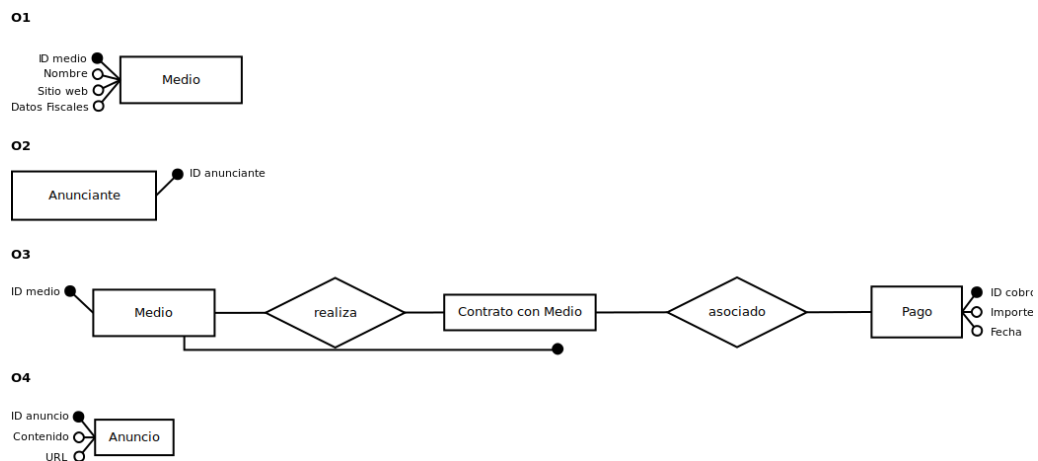
##### 3.3.1.3. O3: Obtención de pagos

Buscar los pagos asociados a contratos de medio a partir del identificador del medio.

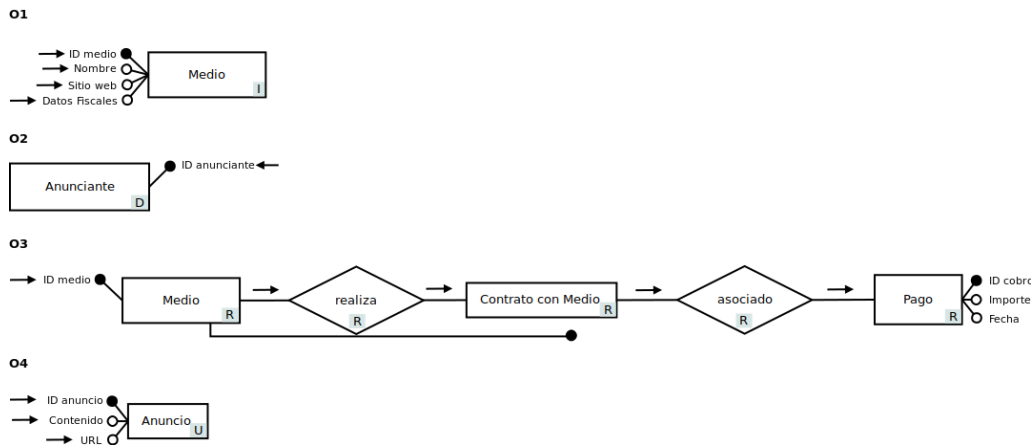
##### 3.3.1.4. O4: Modificación de anuncio

Actualizar el contenido y URL de un anuncio, dado su identificador.

#### 3.3.2. Esquemas de operación



### 3.3.3. Esquemas de navegación



### 3.4. Diseño lógico

Anunciante(ID anunciante, Nombre, Sitio web, Datos fiscales)

Medio(ID medio, Nombre, Sitio web, Datos fiscales)

Anuncio(ID anuncio, ID anunciante, Contenido, URL)

FK: ID anunciante → Anunciante.ID anunciante

Espacio(ID espacio, ID medio, Multimedia, Tamaño, Enlace)

FK: ID medio → Anunciante.ID medio

Audiencia(ID audiencia, Nombre, Descripción)

Contrato\_anunciante(ID contrato anunciante, ID anunciante, Fecha, Importe, Descripción, Duración)

FK: ID anunciante → Anunciante.ID anunciante

Cobro(ID cobro, ID contrato anunciante, ID anunciante, Importe, Fecha) (Fusión de tablas)

FK: ID contrato anunciante → Contrato\_anunciante.ID contrato anunciante FK: ID anunciante → Anunciante.ID anunciante

Contrato\_medio(ID contrato medio, ID medio, Fecha, Duración, Importe, Descripción)

FK: ID medio → Anunciante.ID medio

Pago(ID pago, ID contrato medio, ID medio, Importe, Fecha) (Fusión de tablas)

FK: ID contrato medio → Contrato\_medio.ID contrato medio

FK: ID medio → Anunciante.ID medio

Se\_asigna(ID espacio, ID medio, ID anuncio, ID anunciante)

FK: ID espacio → Espacio.ID espacio

FK: ID medio → Anunciante.ID medio

FK: ID anuncio → Anuncio.ID anuncio

FK: ID anunciante → Anunciante.ID anunciante

Interesa(ID anunciante, ID audiencia)

FK: ID anunciante → Anunciante.ID anunciante

FK: ID audiencia → Audiencia.ID audiencia

Visita(ID medio, ID audiencia)

FK: ID medio → Medio.ID medio

FK: ID audiencia → Audiencia.ID audiencia

En el archivo `tablas.sql` asociado se especifican las sentencias DDL que se pueden utilizar para crear una base de datos siguiendo este diseño físico en el SGBD Oracle.

### 3.4.1. Disparadores PL/SQL

Hemos creado tres disparadores PL/SQL en nuestra implementación (se encuentran en el archivo `triggers.sql`), que detallamos a continuación:

- Disparador para anuncios (preserva **RS7**): Este disparador se activará al modificar un anuncio, de forma que si añadimos una URL a un anuncio asociado a un espacio que no las permite, la asociación se elimine.
- Disparador para cobros (preserva **RS9**): En el momento en el que intentamos insertar un nuevo cobro, antes de insertarlo comprobamos que el importe conjunto con los demás cobros del mismo contrato no superen el importe especificado en el contrato. De ser así, el cobro no será insertado y se lanzará una excepción.
- Disparador para pagos (preserva **RS8**): Idéntico al anterior, solo que se lanzará cuando intentemos insertar un pago.

## 4. Implementación

### 4.1. Tecnologías utilizadas

En el desarrollo de Aorist hemos hecho uso de diversas tecnologías, principalmente el framework Rails sobre el lenguaje Ruby, los sistemas gestores de bases de datos SQLite y Oracle, y varias gemas (librerías de Ruby) que facilitan el uso de Rails y la gestión de los datos, entre las que destacan:

- *activerecord*, que sincroniza los objetos del modelo con la base de datos
- *sqlite3*, *ruby-oci8* y *oracle\_enhanced* para la comunicación con las bases de datos
- *haml*, *sass-rails*, *pure-css-rails* y *font-awesome-sass* para la creación de las vistas

Todas las librerías usadas se detallan en el archivo `Gemfile` del código.

### 4.2. Guía de instalación

Esta es una guía para instalar Aorist en un sistema operativo con kernel Linux, creada y seguida desde Ubuntu 14.04.

#### 4.2.1. Dependencias previas

Necesitaremos paquetes para la gestión de la base de datos y un intérprete de JavaScript. Para la base de datos, instalamos *sqlite3*, mientras que como intérprete de JavaScript usaremos *nodejs*. Podemos instalarlos con los comandos:

```
sudo apt-get install sqlite3
sudo apt-get install nodejs
```

#### 4.2.2. Instalación de Ruby

En caso de que no tengamos Ruby instalado en nuestro sistema, es necesario realizar esta instalación. Lo haremos instalando previamente RVM (*Ruby Version Manager*), que nos permitirá usar Ruby con mayor comodidad. Para instalar RVM y Ruby ejecutaremos los siguientes comandos:

```
gpg --keyserver hkp://keys.gnupg.net --recv-keys\
  409B6B1796C275462A1703113804BB82D39DC0E3
\curl -sSL https://get.rvm.io | bash -s stable --ruby
```

En este punto ya tendremos instalado RVM con Ruby. Seguidamente, ejecutamos el comando `source /home/usuario/.rvm/scripts/rvm` para poder usar rvm. Cambiaremos de versión de Ruby a la 2.2.1 con el comando `rvm use 2.2.1`.

#### 4.2.3. Instalación de las gemas y migración de la base de datos

Ya tenemos instalado Ruby. Ahora tenemos que instalar las gemas necesarias. Para empezar, instalamos *bundler* con `gem install bundler`, lo que nos permitirá instalar el resto de gemas.

Una vez instalado, en la carpeta de la aplicación ejecutamos `bundle --without=test` para instalar todas las gemas necesarias para Aorist.

Instaladas las gemas, hemos de migrar la base de datos. Para ello, ejecutamos `rake db:migrate`. Una vez hecho esto, la aplicación está lista para usarse. Solo queda lanzar el servidor con el comando `rails server`.



### 4.3. Arquitectura de la implementación

Aorist es una aplicación web implementada con Ruby on Rails siguiendo un patrón Modelo-Vista-Controlador. En esta sección, detallaremos cómo gestiona Rails cada parte, y qué hemos implementado nosotros en cada una de ellas.

#### 4.3.1. Modelo

El modelo es la realización del diseño conceptual en forma de clases que se comunican con la base de datos. Cada modelo corresponde al menos a dos archivos: un archivo de código Ruby en el directorio `app/models` y un archivo de migración que Rails utiliza para ejecutar el código SQL necesario al llevar el modelo a la base de datos, de forma que no es necesario escribir sentencias de definición de datos en SQL.

En el código del modelo especificamos sus relaciones con otros, mediante distintas asociaciones como `has_many` (uno a muchos, con el uno en el modelo actual), `belongs_to` (uno a muchos, con el uno en el otro modelo), `has_one` (uno a uno), etc. Esto creará automáticamente métodos para operar con los atributos relacionados de otro modelo que podremos usar en el controlador (por ejemplo, obtener todos los elementos de una entidad débil desde un elemento de la entidad fuerte). Dentro de este archivo podemos definir también validaciones, que son pequeñas comprobaciones que se hacen a la hora de insertar datos, como por ejemplo la presencia obligatoria de un dato o exigir que un valor numérico esté en un rango determinado.

Hemos generado un modelo para cada entidad en nuestro diagrama conceptual, y en ocasiones, en las relaciones muchos a muchos añadimos un modelo adicional representando la asociación. Una vez creados los modelos, se ejecutan las migraciones para que se modifique el esquema de la base de datos, con la creación de nuevas tablas y asociaciones. Dicha migración se realiza con el comando de consola `rake db:migrate`. Las migraciones se asocian a un sello temporal, de forma que el esquema de la base de datos se puede modificar progresivamente y cada vez que se ejecutan migraciones solo se realizan aquellas que no estaban aplicadas.

A continuación mostramos la lista completa de clases utilizadas para el modelo:

- `Anunciante`
- `AnuncianteContrato`
- `Cobro`
- `Anuncio`
- `AnuncioAllocation` (representa la asignación de un anuncio a un espacio)
- `Medio`
- `MedioContrato`
- `Pago`
- `Espacio`
- `Audience`

#### 4.3.2. Controladores

Los controladores de la aplicación gestionan los datos que proporcionan los modelos y los preparan para mostrarlos en las vistas, a la vez que tramitan las peticiones que envían los usuarios. Se encuentran en el directorio `app/controllers`.

La arquitectura de un controlador en Rails es sencilla: cada controlador es una clase, que contiene métodos que corresponden a acciones. La mayoría de las acciones tienen una vista asociada, aunque otras pueden redirigir a otra acción. Además, las acciones permiten organizar la forma en que se responde a las rutas pedidas por el navegador del usuario. Por ejemplo, si tenemos definida la ruta GET `'contabilidad/show'`, al acceder a dicha dirección Rails ejecutará el método `show` del controlador `ContabilidadController`.

Todas las entidades del diseño conceptual tienen un controlador asociado, que responde a rutas siguiendo la convención REST (Representational State Transfer). Por ejemplo:

```
get    '/anunciantes'      => AnunciantesController#index
get    '/anunciantes/new' => AnunciantesController#new
```

```

post    '/anunciantes'      => AnunciantesController#create
get     '/anunciantes/:id' => AnunciantesController#show
delete  '/anunciantes/:id' => AnunciantesController#destroy

```

Asimismo, algunos controladores adicionales añaden funcionalidad a la aplicación, en concreto `HomeController` proporciona una página de inicio y `ContabilidadController` da acceso a la contabilidad. Las rutas se especifican y se configuran con más detalle en el archivo `config/routes.rb`. La lista completa de controladores es la siguiente:

- `ApplicationController`
- `HomeController`
- `AnunciantesController`
- `AnuncianteContratosController`
- `CobrosController`
- `AnunciosController`
- `MediosController`
- `MedioContratosController`
- `PagosController`
- `EspaciosController`
- `AudiencesController`
- `ContabilidadController`
- `ErrorsController`

#### 4.3.3. Vistas

Las vistas de la aplicación son archivos de código en lenguaje Haml (HTML Abstraction Markup Language), que residen bajo el directorio `app/views`, con un nombre que referencia la acción del controlador a la que están asociadas. Haml permite referenciar a variables que haya definido dicha acción, y ejecutar métodos auxiliares definidos en el controlador como `helper_method`.

Todas las vistas tienen como plantilla principal al archivo `application.haml` del directorio `layouts`, y algunas utilizan `with_side.haml` para mostrar una barra lateral. Además, el estilo de las páginas viene definido en Sass (lenguaje que compila a CSS) en el archivo `home.sass` del directorio `app/assets/stylesheets`.

#### 4.3.4. Otros archivos

En el directorio `config` se encuentran diversos archivos de configuración de la aplicación, en particular `routes.rb` define las rutas aceptadas, `database.yml` especifica parámetros para la conexión a las bases de datos y los archivos bajo el directorio `environments` configuran la aplicación según el entorno en que se ejecute (desarrollo, test o producción).

El directorio `db` contiene los archivos de migración (bajo `migrate`), el esquema de la base de datos en `schema.rb` y, en el caso del SGBD SQLite, el archivo que contiene la base de datos.

Los directorios `public` y `test` contienen código auxiliar para páginas de error y tests unitarios, respectivamente, que es creado automáticamente por Rails sin intervención del desarrollador.

### 4.4. Uso de Aorist

La interfaz de Aorist se jerarquiza en cuatro partes: Gestión de anunciantes, Gestión de medios, Gestión de audiencias y Contabilidad. En la pantalla de inicio de la aplicación podremos acceder a cada una de ellas, además de hacer una gestión básica de los datos almacenados en la base de datos; podemos eliminar todo lo que hay almacenado, dejando la base de datos limpia, y podemos crear unos datos de ejemplo. Esta opción ha sido creada en vistas a una exposición de la aplicación, para no tener que introducir manualmente datos necesarios para visualizar el funcionamiento de la misma. En la Figura 1 podemos ver cómo es la pantalla de inicio de Aorist.

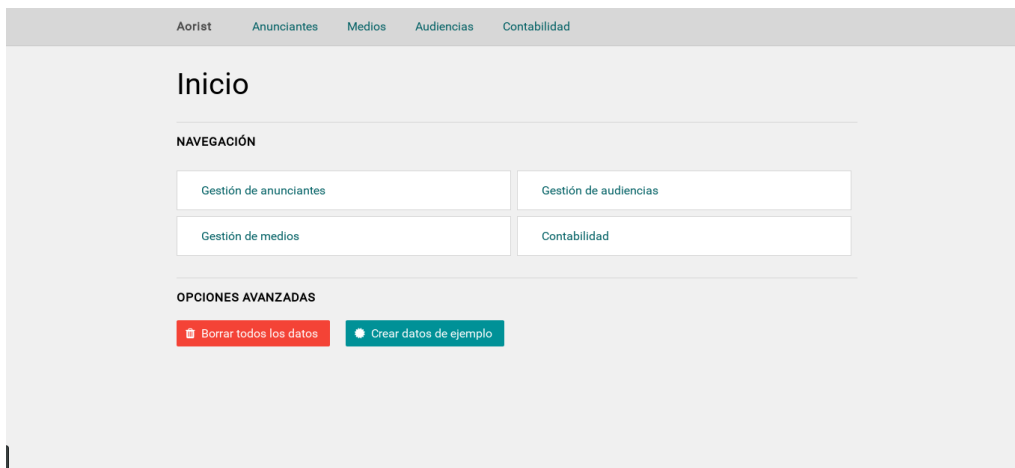


Figura 1: Inicio de Aorist

#### 4.4.1. Gestión de anunciantes

Al acceder al menú de Gestión de anunciantes, nos encontramos una pantalla con el listado de los anunciantes almacenados en el sistema, junto a la opción de crear un nuevo anunciante. Podemos verlo en la Figura 2.

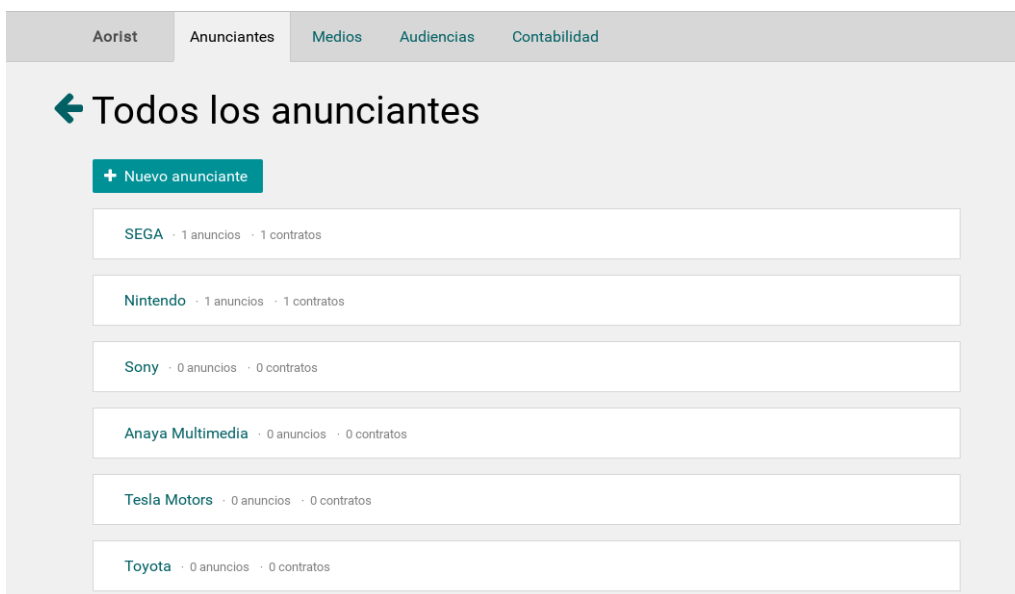


Figura 2: Lista de anunciantes

Si accedemos al menú de creación de un nuevo anunciante, nos encontramos un formulario para rellenar los datos del nuevo anunciante: el nombre, la web y los datos fiscales. Además, a la izquierda podemos ver los anunciantes ya creados en el sistema, y podemos acceder a ellos pinchando en su recuadro. Lo observamos en la Figura 3.

The screenshot shows the 'Crear un nuevo anunciante' (Create a new advertiser) form. At the top, there is a navigation bar with tabs: 'Aorist', 'Anunciantes', 'Medios', 'Audiencias', and 'Contabilidad'. The 'Anunciantes' tab is active. The form has a left sidebar with a '+ Nuevo anunciante' button and a list of existing advertisers: SEGA, Nintendo, Sony, Anaya Multimedia, Tesla Motors, and Toyota. The main area contains three input fields: 'Nombre' (Name), 'Web' (Website), and 'Datos fiscales' (Tax data). Below these fields is a green 'Crear anunciante' (Create advertiser) button.

Figura 3: Nuevo anunciante

Si hacemos click en cualquier anunciante, accedemos a su ficha. En ella vemos todos sus datos y tenemos opciones para ver sus anuncios, sus contratos y eliminar el anunciante, además de la lista de audiencias que tiene asociadas y las que no. Es posible asociarle una nueva audiencia con la opción que aparece al lado de cada una de las que no tiene asociadas.

Para acceder a los detalles de sus anuncios y contratos deberemos hacer click en la opción correspondiente. Podemos ver la ficha completa en la Figura 4.

The screenshot shows the 'SEGA' advertiser profile page. The navigation bar is the same as in Figure 3. The page title is 'SEGA'. Below the title, there are three buttons: 'Anuncios (1)' (Adverts), 'Contratos (1)' (Contracts), and 'Eliminar' (Delete). The page is divided into three main sections: 'PROPIEDADES' (Properties), 'AUDIENCIAS ASOCIADAS' (Associated Audiences), and 'ASOCIAR NUEVAS AUDIENCIAS' (Associate New Audiences). The 'PROPIEDADES' section shows 'Web: http://sega.com' and 'Datos fiscales: 536400000'. The 'AUDIENCIAS ASOCIADAS' section shows a list of associated audiences with a 'Quitar' (Remove) button next to 'Gamers'. The 'ASOCIAR NUEVAS AUDIENCIAS' section shows a list of new audiences with an 'Asociar' (Associate) button next to 'Geeks', 'Conductores' (Drivers), and 'Lectores' (Readers).

Figura 4: Datos de anunciante

Si accedemos a los anuncios de nuestro anunciante, se mostrarán dos listados: aquellos anuncios que están puestos en marcha (es decir, ya tengan un espacio asociado) y aquellos que no lo estén. Además, tendremos la opción

de crear un nuevo anuncio asociado al anunciante, especificando su contenido y su URL, en caso de tenerla. De nuevo tendremos un listado de los anuncios ya existentes a la izquierda, como teníamos en los anunciantes.

The screenshot shows the 'Nuevo anuncio' (New ad) form. At the top, there is a navigation bar with links: Aorist, Anunciantes, Medios, Audiencias, and Contabilidad. The main heading is 'Nuevo anuncio' with a back arrow icon. Below the heading, there is a sidebar on the left with a '+ Nuevo anuncio' button and two sections: 'ANUNCIOS PENDIENTES' (with a message 'Aquí no hay nada') and 'ANUNCIOS PUESTOS EN MARCHA' (with a card for 'Sonic The Hedgehog...'). The main form area is titled 'del anunciante SEGA' and contains two input fields: 'Contenido' and 'URL'. Below these fields is a 'Crear anuncio' button.

Figura 5: Nuevo anuncio

En la ficha de cada anuncio tenemos sus datos y las opciones de editar dichos datos, lo que nos lleva a un formulario similar al de creación de un nuevo anuncio, y eliminarlo completamente del sistema. Además, podemos ver un listado de los espacios que tiene asociados (en caso de tener alguno) y de aquellos que no pero es posible establecer dicha asociación, con la opción para realizarla. En estos espacios también se muestra el número de audiencias en común entre el anunciante y el medio, criterio a tener en cuenta a la hora de asociar el anuncio a algún espacio. Podemos verlo de forma detallada en la siguiente Figura.

The screenshot shows the 'Anuncio 9' (Ad 9) details page. At the top, there is a navigation bar with links: Aorist, Anunciantes, Medios, Audiencias, and Contabilidad. The main heading is 'Anuncio 9' with a back arrow icon. Below the heading, there is a sidebar on the left with a '+ Nuevo anuncio' button and two sections: 'ANUNCIOS PENDIENTES' (with a message 'Aquí no hay nada') and 'ANUNCIOS PUESTOS EN MARCHA' (with a card for 'Sonic The Hedgehog 2016 (9)'). The main form area is titled 'del anunciante SEGA' and contains two buttons: 'Editar' and 'Eliminar'. Below these buttons, there are sections: 'PROPIEDADES' (with fields for 'Contenido: Sonic The Hedgehog 2016' and 'URL:'), 'ESPACIOS ASIGNADOS' (with a card for 'Espacio 13 (Polygon)' and a 'Quitar' button), and 'ASIGNAR A UN ESPACIO' (with two cards for 'Espacio 15 (Polygon)' and 'Espacio 16 (Polygon)', each with an 'Asignar' button and a note '1 audiencia en común').

Figura 6: Datos de anuncio

Entrando ahora en los contratos de cada anunciante, nos encontramos de nuevo un listado con los contratos

creados, con la posibilidad de acceder a un menú para crear uno nuevo proporcionando sus datos. Dentro de la ficha de cada contrato nos encontramos con todos sus datos, junto a sus cobros asociados y la posibilidad de registrar uno nuevo. Para ello simplemente especificaremos la fecha y el importe del cobro. Podemos ver estos menús en la Figura.

The screenshot shows a web application interface with a top navigation bar containing 'Aorist', 'Anunciantes', 'Medios', 'Audiencias', and 'Contabilidad'. The main heading is '← Contrato 8'. On the left, there is a sidebar with a '+ Nuevo contrato' button and a 'Contrato 8' link. The main content area is divided into sections: 'DATOS' with fields for 'Descripción: Anunciar Smash Bros', 'Fecha de inicio: 2016-01-31', 'Fecha de fin: 2018-01-31', and 'Importe: 6280.0'; 'REALIZAR UN COBRO' with a form for 'Importe', year (2016), month (February), and day (1), and a 'Realizar cobro' button; and 'COBROS REALIZADOS' with a list of payments: '222.0 € - 2016-01-31' and '100.0 € - 2016-01-31'.

Figura 7: Datos contrato

#### 4.4.2. Gestión de medios

El menú de Gestión de medios es similar al de Gestión de anunciantes. Inicialmente, nos encontramos un listado de todos los medios y la posibilidad de crear uno nuevo, accediendo a una pantalla donde especificar su nombre, web y datos fiscales. Dentro de la ficha de cada medio encontraremos sus datos, un enlace a sus contratos y a sus espacios, sus audiencias asociadas y las no asociadas (con la posibilidad de asociarle nuevas), y una opción de eliminar el medio del sistema. Podemos ver todo esto en las siguientes Figuras.

The screenshot shows a web application interface with a top navigation bar containing 'Aorist', 'Anunciantes', 'Medios', 'Audiencias', and 'Contabilidad'. The main heading is '← Todos los medios'. On the left, there is a sidebar with a '+ Nuevo medio' button. The main content area displays a list of media entities: 'Polygon' (4 espacios, 1 contratos), 'The Verge' (0 espacios, 0 contratos), 'MotorAdictos' (0 espacios, 0 contratos), 'Fantifica' (0 espacios, 0 contratos), and 'Spotify' (0 espacios, 0 contratos).

Figura 8: Lista de medios

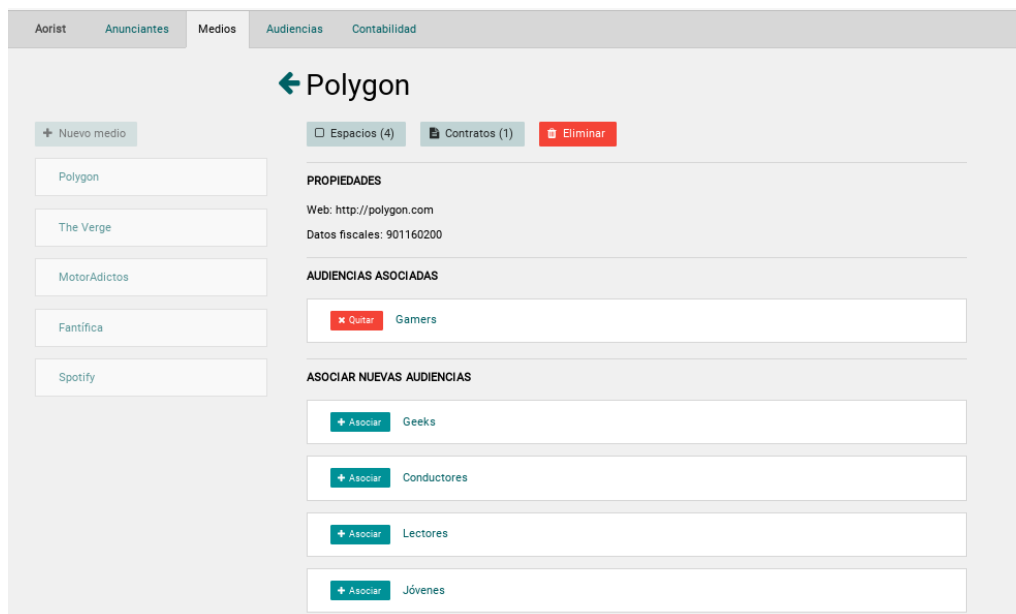


Figura 9: Datos de medio

Si nos vamos a los espacios asociados a cada medio, encontramos un listado de los ya creados. Accediendo a cualquiera de ellos podemos ver sus datos y tenemos la posibilidad de eliminarlo del sistema. Al contrario que en los anuncios, los datos de los espacios no pueden ser editados, por tanto no nos encontraremos dicha opción.



Figura 10: Espacios

Para los contratos asociados a un medio, nos encontraremos la misma funcionalidad que en los asociados a los anunciantes: tendremos un listado de los contratos, la posibilidad de crear uno nuevo, y en su ficha podremos visualizar sus datos y crear un nuevo pago asociado a dicho contrato. Podemos ver esto en la siguiente Figura.

**← Contrato 8**

[+ Nuevo contrato](#)

Contrato 8

**DATOS**

Descripción: Anunciar videojuegos  
 Fecha de inicio: 2016-01-31  
 Fecha de fin: 2018-01-31  
 Importe: 12000.0

**REALIZAR UN PAGO**

Importe  2016 February 1 [Realizar pago](#)

**PAGOS REALIZADOS**

1000.0 €	2016-01-31
----------	------------

Figura 11: Contrato con medio

#### 4.4.3. Gestión de audiencias

Al acceder al menú de Gestión de audiencias, nos encontramos un listado con todas las audiencias creadas en el sistema. Tenemos la posibilidad de crear una nueva audiencia, para la que especificaremos un nombre y una descripción detallada, que nos será útil en caso de consultas futuras. Accediendo a cada audiencia podemos ver sus datos, si bien no se pueden establecer relaciones entre las audiencias y las empresas; estas se harán en el apartado de gestión de cada una de estas empresas. Vemos en la siguiente Figura un ejemplo del listado de las audiencias.

**← Audiencias**

[+ Nueva audiencia](#)

- Gamers
- Geeks
- Conductores
- Lectores
- Jóvenes
- Niños

Figura 12: Audiencias

#### 4.4.4. Contabilidad

La sección de contabilidad nos lleva a una pantalla que recoge todos los datos relacionados con los cobros y los pagos asociados a los contratos. Podemos ver un balance total, que suma todos los cobros realizados y le resta los pagos, y se mostrará en rojo en caso de estar en negativo. A la derecha tendremos la suma de los cobros y los pagos por separado, que estarán detallados en la parte inferior de la sección. Tendremos todos los cobros y los pagos en dos columnas (una para cada tipo), donde se especificará la fecha, el contrato asociado, la empresa asociada y el importe de la transacción. Podremos acceder al contrato o la empresa desde la misma transacción, con un enlace situado en ella.



Es posible ordenar o agrupar las transacciones siguiendo distintos criterios. Inicialmente estarán ordenadas por fecha, apareciendo primero las más recientes. Podemos ordenar también por importe, donde situaremos primero aquellas transacciones con mayor cuantía. Para agruparlas, podremos hacerlo por contrato, donde mostraremos el importe total y la empresa a la que está asociada, o hacerlo directamente por empresa, viendo la cantidad invertida o recibida de cada una de ellas. Podemos ver esta pantalla resumida en la siguiente Figura.

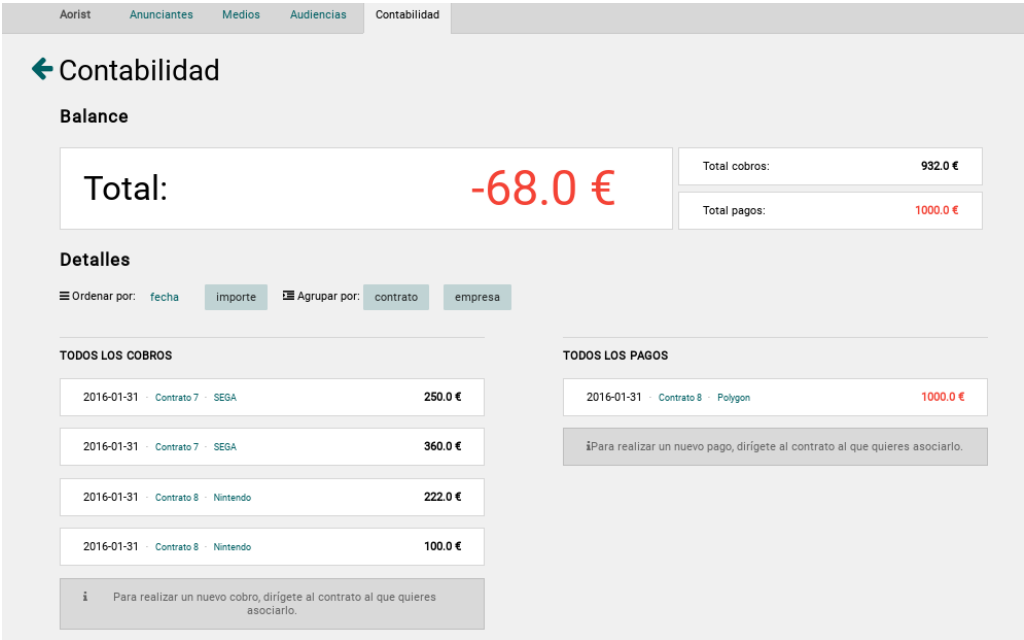


Figura 13: Contabilidad