

Tecnologías de Desarrollo Software  
Jesús García Molina  
Ingeniería Informática  
Universidad de Murcia



*Plataforma de Financiación Colectiva*

Francisco José Herrera Zapata  
<franciscojose.herrera@um.es>  
Juan José Andreu Blázquez  
<juanjose.andreu@um.es>  
12 de Junio de 2015

# Índice de contenidos

[Diagrama de clases del dominio](#)

[Diagramas de secuencia](#)

[Estructura de la aplicación](#)

[Decisiones de diseño](#)

[Patrones de diseño utilizados](#)

[Componentes utilizados](#)

[Tests implementados](#)

[Test Unitarios - RecompensaTest](#)

[Test de Integración - ControladorTest](#)

[Manual de Usuario](#)

[Acceso a la aplicación Apóyanos](#)

[Ventana Inicio/Presentación](#)

[Menú](#)

[Ventana Votación](#)

[Ventana Financiación](#)

[Ventana Información Proyecto en Votación](#)

[Ventana Información Proyecto en Financiación](#)

[Ventana Nuevo Proyecto](#)

[Ventana Apoyos](#)

[Ventana Notificaciones](#)

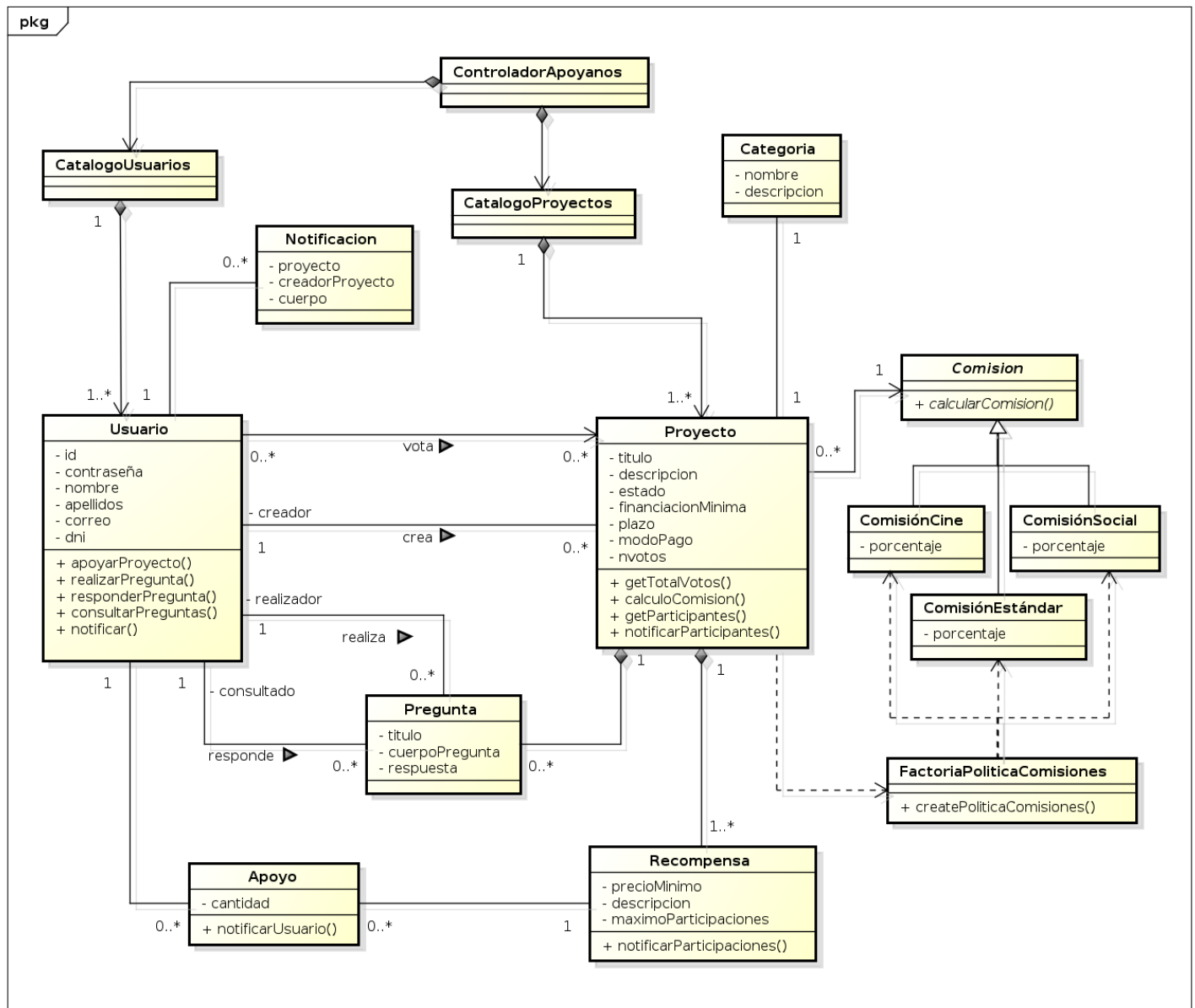
[Ventana Preguntas](#)

[Observaciones finales](#)

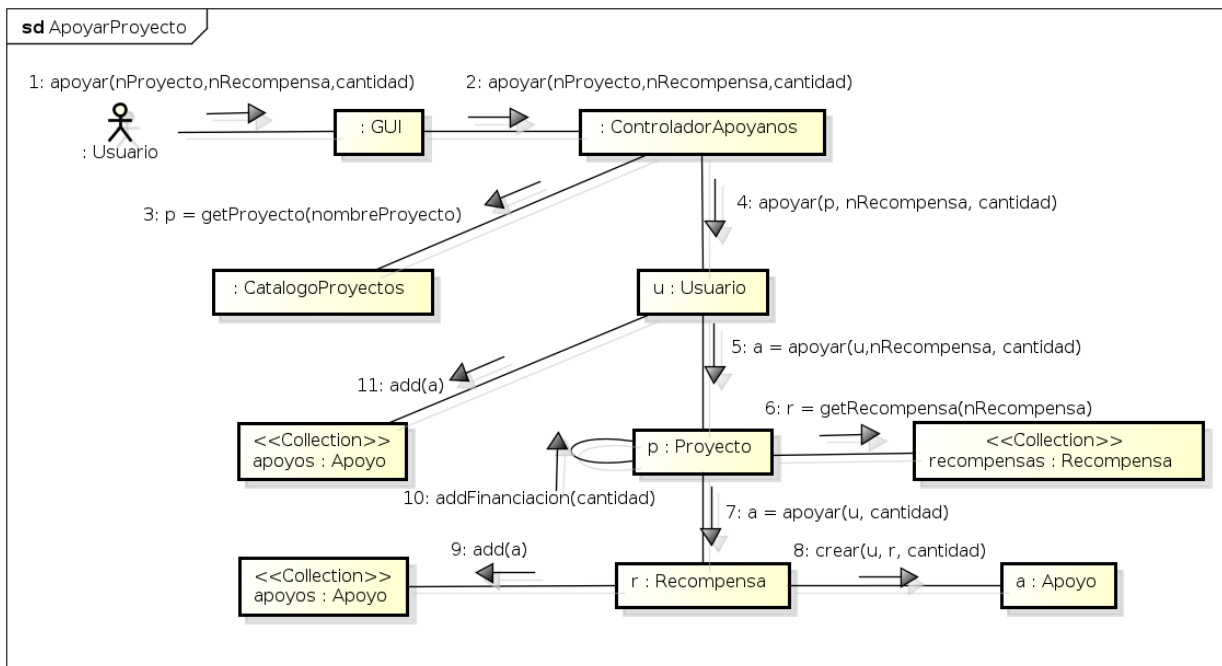
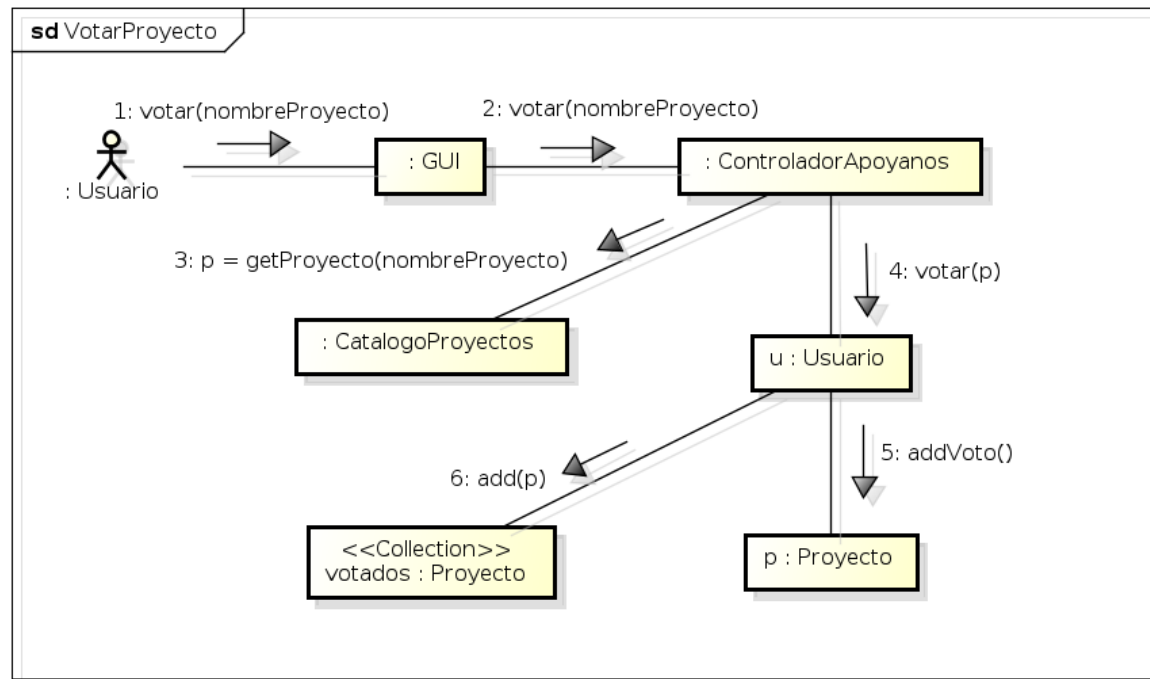
[Estimación del tiempo](#)

[Apreciaciones personales de la práctica](#)

# Diagrama de clases del dominio



## Diagramas de secuencia



## Estructura de la aplicación

La aplicación de crowdfunding o financiación colectiva Apóyanos ha sido diseñada utilizando una arquitectura de modelo-vista-controlador, de forma que se ha separado en diversos paquetes independientes el modelo, la vista y el controlador. El modelo sigue la estructura vista en la sección [diagrama de clases del dominio](#). El controlador es el punto de entrada que permite a la vista llevar a cabo operaciones sobre el modelo, y garantiza que se satisfagan ciertas restricciones. La vista realiza las operaciones sobre el modelo a través del controlador, y no interactúa directamente con él. De este modo, el modelo es independiente del controlador y de la vista, y podría funcionar con distintas implementaciones de vista haciendo solicitudes al controlador.

## Decisiones de diseño

Durante el desarrollo de la plataforma se han tomado diversas decisiones de diseño para resolver algunas cuestiones y retos a los que nos hemos tenido que enfrentar.

- Con el objetivo de facilitar la creación de un proyecto desde la vista, y sin acoplar ésta al modelo, se ha creado una clase *RecompensaVista* que solamente tiene aquellos atributos que el modelo necesita recibir de una recompensa para poder crearse adecuadamente. El controlador recibe una colección de objetos *RecompensaVista* en la llamada para crear un proyecto, y son los parámetros incluidos en ella los que le pasa al proyecto.
- Como regla de negocio adicional, hemos decidido que un proyecto que consigue toda su financiación, en vez de ser cerrado y completado, continúa con la campaña de recogida de fondos hasta que se vence su plazo, de forma similar a como funcionan muchas plataformas actuales de recogida de fondos.
- Al implementar la persistencia, nos enfrentamos al problema de que se pueden producir bucles infinitos de recuperación de entidades si hay referencias cruzadas, o bucles en las referencias. Para solucionar este problema, se ha utilizado un pool de entidades recuperadas en cada DAO. De este modo un objeto que está siendo recuperado, se introduce en el pool de objetos recuperados antes de intentar recuperar aquellas entidades que contiene, de forma que cualquier intento subsecuente de recuperar ese objeto, encuentra al objeto en el pool y lo devuelve. Además la factoría *H2FactoriaDAO* crea una única instancia de cada tipo de DAO y devuelve siempre la misma, para que el acceso al pool de objetos se realice siempre sobre el mismo pool.
- Para gestionar la creación de la comisión adecuada para un proyecto, se ha creado una factoría que encapsula la lógica de la creación de la comisión y favorece la extensibilidad de cara al futuro, si deciden añadirse nuevos tipos de comisiones.

## Patrones de diseño utilizados

En el modelo de la aplicación se han utilizado varios de los patrones de diseño vistos en la teoría de la asignatura.

- En la implementación de los diferentes tipos de comisión se ha utilizado un patrón **estrategia**. Se ha creado una clase abstracta *PoliticaComisiones* y se han creado las tres comisiones especificadas en el enunciado de la práctica. El proyecto utiliza la clase abstracta, y se abstrae de las comisiones específicas implementadas. Además, como se ha indicado en el apartado [anterior](#), se ha creado una factoría que es la encargada de crear la comisión concreta que se aplicará a un proyecto. De este modo, cuando haya que implementar una nueva política, solamente será necesario añadir la nueva clase y modificar la factoría de creación de comisiones.
- Tanto el controlador, como los catálogos utilizados en nuestra aplicación se han desarrollado siguiendo un patrón **singleton**. De este modo nos aseguramos que sólo hay una instancia en ejecución de este tipo de clase.
- En el diseño de la persistencia, se ha utilizado un patrón **factoría abstracta**. Tenemos una clase abstracta *FactoriaDAO*, implementada como singleton, que se encarga de definir los **métodos factoría** que devuelven un DAO para cada objeto con persistencia. Dichos DAOs son a su vez interfaces.  
Puesto que hemos implementado la persistencia usando la biblioteca H2 proporcionada, sólo se ha implementado una factoría concreta para H2, que implementa los métodos de la factoría abstracta, y crea los DAOs específicos de H2 también implementados.
- Además de lo anterior, cada uno de los DAOs implementados sigue el patrón **adaptador**, ya que es una interfaz de acceso a la persistencia, a través de la cual se almacenan los objetos en la base de datos, pero sin conocer las funcionalidades específicas de cada API concreta.
- Por otro lado, tanto los componentes utilizados como los manejadores de eventos de ratón y botones de la vista, utilizan el patrón **observer** en los listeners de eventos.

## Componentes utilizados

En la plataforma apóyanos se han utilizado tres componentes diferentes:

1. Componente **Luz**: Este componente se desarrolló en clase y se nos ha proporcionado una implementación de él. En nuestro proyecto lo utilizamos en la vista tanto para mostrar un diálogo de selección de archivo y llamar al componente de financiación externa, como para simular el avance del tiempo y la comprobación de la finalización de los proyectos.
2. Componente **Financiación externa**: Este componente ha sido desarrollado por nosotros a partir del código proporcionado para cargar financiación desde fuentes externas a proyectos de nuestra plataforma. A la hora de integrar el componente, se ha decidido establecer al controlador como listener de los eventos producidos por éste, de forma que el controlador recibe el evento que contiene toda la financiación obtenida por medios externos para cada proyecto, y se encarga de añadirla a los proyectos afectados. A su vez, el listener del componente luz que lo activa, se encarga de crear este componente, y establecer al controlador como listener de éste.
3. Por último, también hemos utilizado un componente **JCalendar** que se utiliza para la elección de la fecha de finalización de la campaña de crowdfunding de un proyecto. Este componente hace muy sencillo mostrar un pequeño calendario al usuario de la aplicación, dónde él seleccione de forma visual la fecha elegida de finalización de campaña.

# Tests implementados

## Test Unitarios - RecompensaTest

Los test unitarios o pruebas unitarias sirven para comprobar el correcto funcionamiento de un módulo de código. De esta forma se asegura que cada uno de estos módulos o bloques funcionen de la manera esperada, la correcta, por separado.

El test que hemos preparado ha sido para las recompensas y sus características.

Lo primero que hemos necesitado es crear una serie de objetos que van vinculados a estas: Una *recompensa* pertenece a un *proyecto* y un *proyecto* ha de ser creado antes por un *usuario* existente.

Hemos creado:

- 2 usuarios: *usuario1* y *usuario2*.
- Un *proyecto* creado por el *usuario1*.
- Y 2 recompensas *R1* y *R2* que responden a los dos tipos de recompensas que tenemos: con límite y sin límite pero que tratamos de la misma forma ya que el valor de aquellas que no tienen límite es 0. El límite es el número máximo de participantes que puede tener una recompensa.

El primer test de creación que hemos preparado es precisamente ver que los valores después de crear las recompensas que nos devuelve el método *getMaximoParticipantes()* corresponde a lo esperado.

*R1* la hemos creado con 1 máximo de participante y efectivamente el valor devuelto por *R1.getMaximoParticipantes()* es también 1.

Por su parte, el valor *R2.getMaximoParticipantes()* es 0 que también es el valor esperado ya que como hemos mencionado anteriormente aquellas recompensas sin límite las trataremos con valor 0.

Un segundo test que se ha preparado es de apoyo de una recompensa donde el método *Recompensa.apoyar(usuario, cantidadApoyo, "")* necesita un usuario correcto, una cantidad de apoyo que ha de ser igual al valor de la recompensa y una cadena vacía que hemos dejado para posibles usos futuros que sería donde el mecenas podría introducir un texto pero actualmente no implementado.

Como se puede apreciar el primer test y el tercero no se cumplen ya que la cantidad introducida no coincide con el valor de la recompensa y se captura la excepción mientras que el segundo sí.



## Test de Integración - ControladorTest

Una vez hechos los test unitarios sabemos que todos esos módulos funcionan correctamente. ¿Esto quiere decir que nuestra aplicación funciona?

Para eso utilizamos los test de integración que prueban la funcionalidad de la aplicación en su conjunto, incluyendo aquellos componentes externos a la propia aplicación: controlador, modelo y persistencia. Todo ello de manera global para así sí poder asegurar que nuestra aplicación funciona correctamente.

En nuestra aplicación la clase que se encarga de toda la funcionalidad de la aplicación es el Controlador, que a su vez es el encargado de relacionar las clases del **modelo**.

Hemos preparado 10 test que se encargan prácticamente de la total funcionalidad de la aplicación:

- *test10RegistrarUsuario()*: Este test nos asegura cuatro cuestiones:
  - ningún usuario está registrado si antes no se ha dado de alta.
  - un usuario es dado de alta introduciendo los datos correctos.
  - un usuario está registrado si está dado de alta.
  - un usuario registrado no puede volver a ser dado de alta.
- *test20Login()*: Este test nos asegura 5 cuestiones relacionadas con el logueo de usuarios:
  - Un usuario sin registrar no se puede loguear.
  - Un usuario registrado no se puede loguear con una contraseña incorrecta.
  - Un usuario registrado con su contraseña correcta sí puede loguearse.
  - Si hay un usuario activo éste puede hacer logOut.
  - Si no hay un usuario activo no puede haber logOut.
- *test30CrearProyecto()*: Este test donde aseguramos cuestiones relacionadas con la creación de un proyecto:
  - No se pueden crear proyectos sin recompensas.
  - No se pueden crear proyectos con categorías erróneas o falsas.
  - No se pueden crear proyectos sin fecha.
  - No se pueden crear proyectos con fecha anterior o igual a la actual.
- *test40VotarProyecto()*: Test referente al acto de votar un proyecto:
  - Un proyecto recién creado no tiene votos y su número de votos es cero y su estado es de *votación*.
  - Un proyecto que recibe un voto es un proyecto que ha sido votado.

- Un proyecto votado por un usuario no puede volver a ser votado por el mismo usuario.
- *test50VotosAProyecto()*: Votos que recibe un proyecto:
  - Un proyecto que recibe 50 votos de 50 usuarios distintos es un proyecto que cambia de estado y pasa a estar en *financiación*.
- *test60ApoyarProyecto()*: Apoyos a un proyecto:
  - Un proyecto que acaba de pasar al estado *financiación* no tiene apoyos.
  - Un apoyo a un proyecto por medio de una recompensa con un valor distinto al de la recompensa no es un apoyo válido y por tanto el proyecto continúa sin apoyos.
  - Un apoyo con el mismo valor al de la recompensa seleccionada sí contabiliza.
  - Si una recompensa con un número limitado de apoyos llega a ese número no puede recibir más apoyos.
- *test60FinanciarProyectoCompleto()*: Completar financiación:
  - Un proyecto en financiación recibe tantos apoyos como sean válidos según las recompensas.
  - Un proyecto que por medio de los apoyos que iguala o supera el valor del mismo pasa a estar *financiado*.
- *test70Notificaciones()*: Funcionamiento de las notificaciones:
  - Un proyecto financiado notifica a su creador de su estado.
  - Un proyecto *financiado* notifica a sus mecenas (usuarios que lo han apoyado por medio de una recompensa) de que ha sido financiado en su totalidad.
- *test80HacerPregunta()*: Control de preguntas y respuestas:
  - Un emisor es un usuario que quiere hacer una pregunta a un receptor.
  - Un receptor es un usuario que ha creado un proyecto y que recibe una o más preguntas.
- *test90ComprobarPlazoFinalizacion()*: Finalización de un proyecto:
  - Un proyecto es completado con éxito cuando su estado es *financiado* y se cumple fecha de fin del proyecto.

## Manual de Usuario

La aplicación **Apóyanos** es una aplicación de escritorio que permite a aquellas personas que quieren poner en marcha un proyecto darle repercusión al mismo para así conseguir financiación para poder llevarlo a cabo. Esta financiación correrá a cargo de unos mecenas que a cambio pueden obtener una recompensa. Los proyectos una vez dados de alta tendrán que ser valorados por el resto de usuarios de la aplicación obteniendo votos. De esta forma podrán pasar a la siguiente fase, la de financiación propiamente dicha, donde ya los mecenas irán sufragando el coste del proyecto a cambio de las mencionadas recompensas.

En síntesis, Apóyanos es una aplicación de crowdfunding, tu aplicación de crowdfunding.

### Acceso a la aplicación Apóyanos

Una vez arrancada la aplicación lo primero que nos encontramos es la ventana de acceso que solicita un *Login* y un *Password*.

La imagen muestra una ventana de acceso con el título "Acceso". En el centro, hay dos campos de entrada: uno etiquetado "Login:" y otro etiquetado "Password:". Debajo de estos campos, hay tres botones: "Login", "Registro" y "Salir". La ventana tiene una barra de título con tres botones de control (rojo, amarillo, gris) a la izquierda.

Figura: Ventana de Acceso.

En caso de no estar registrado se debe pulsar el botón *Registro* y así proceder al alta de un *usuario*. Recordar que un *usuario* puede tener **dos roles** al mismo tiempo dentro de la aplicación: por una parte como creador de un proyecto y por otra como persona que apoya un proyecto ya sea simplemente dando su voto o como mecenas colaborando con una o varias recompensas.

El procedimiento de *Registro* solicita una serie de datos personales de carácter obligatorio debiendo aceptar la casilla de “*Aceptar y ver términos de uso*”.



The image shows a registration window titled "Acceso". It contains the following fields and controls:

- Nombre : [text input]
- Apellidos : [text input]
- DNI : [text input]
- Email : [text input]
- Usuario : [text input]
- Password : [text input] Otra vez: [text input]
- ☐ Aceptar y ver términos de uso
- Volver [button]
- Registrar [button]

Figura: Ventana de registro.

## Ventana Inicio/Presentación

Una vez que el *usuario* está registrado y ha completado el acceso con su *Login* y *Password* la aplicación accede a la Ventana de Presentación de la aplicación (*VentanaIntroApoyanos*) donde se puede ver el logo de la aplicación en grande, los botones de acceso a los componentes y el menú de la aplicación, presente en todas las ventanas.



Figura: Ventana de Presentación de la aplicación.

## Menú

El *menú* permite desplazarse casi la práctica totalidad de las ventanas de la aplicación.

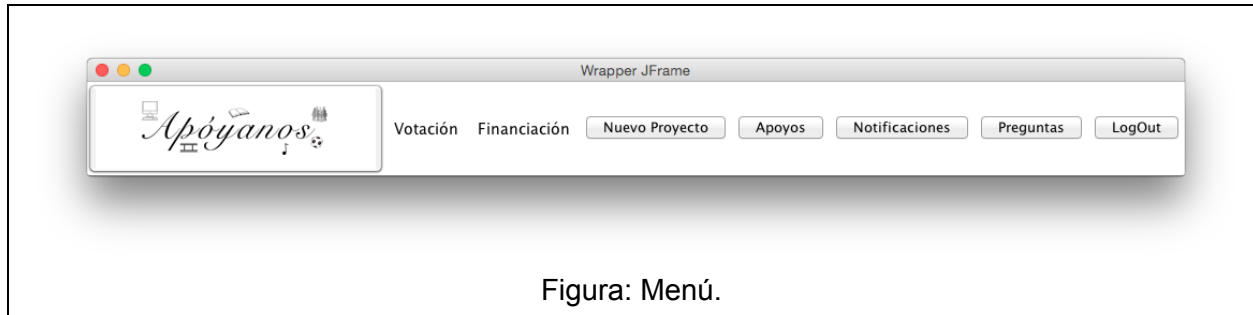


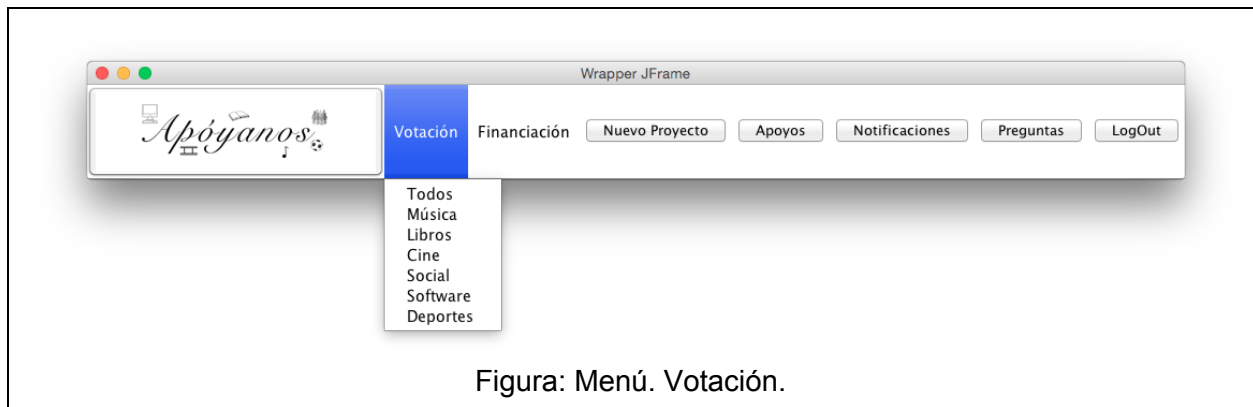
Figura: Menú.

La funcionalidad del *menú*, de derecha a izquierda es la siguiente:

- **LogOut:** Desactiva el usuario activo llevándolo a la ventana de acceso, en caso de que él mismo u otro usuario deseen volver a entrar o pudiendo salir y cerrar la aplicación desde ahí.
- **Preguntas:** Al pulsar este botón la aplicación muestra una ventana donde podremos formular preguntas y ver las preguntas formuladas cuando como usuario estamos en el rol de mecenas o persona interesada en un proyecto, o ver y responder las preguntas que pueden hacer cuando el usuario activo es creador de un proyecto.
- **Notificaciones:** Lleva a una ventana que muestra un listado con las notificaciones que el sistema le ha hecho llegar al usuario activo.
- **Apoyos:** Lleva a una ventana que muestra los apoyos a proyectos que ha realizado el usuario activo.
- **Nuevo Proyecto:** Lleva a una ventana donde el usuario activo podrá crear un proyecto en la aplicación.
- **Financiación:** Lleva a una ventana con un listado de proyectos en fase de Financiación. Se podrán ver todos los proyectos en esta fase o segmentados por categorías: Música, Libros, Cine, Social, Software y Deportes.



- **Votación:** Lleva a ventana con un listado de proyectos en fase de Votación. Se podrán ver todos los proyectos en esta fase o segmentados por categorías: Música, Libros, Cine, Social, Software y Deportes.

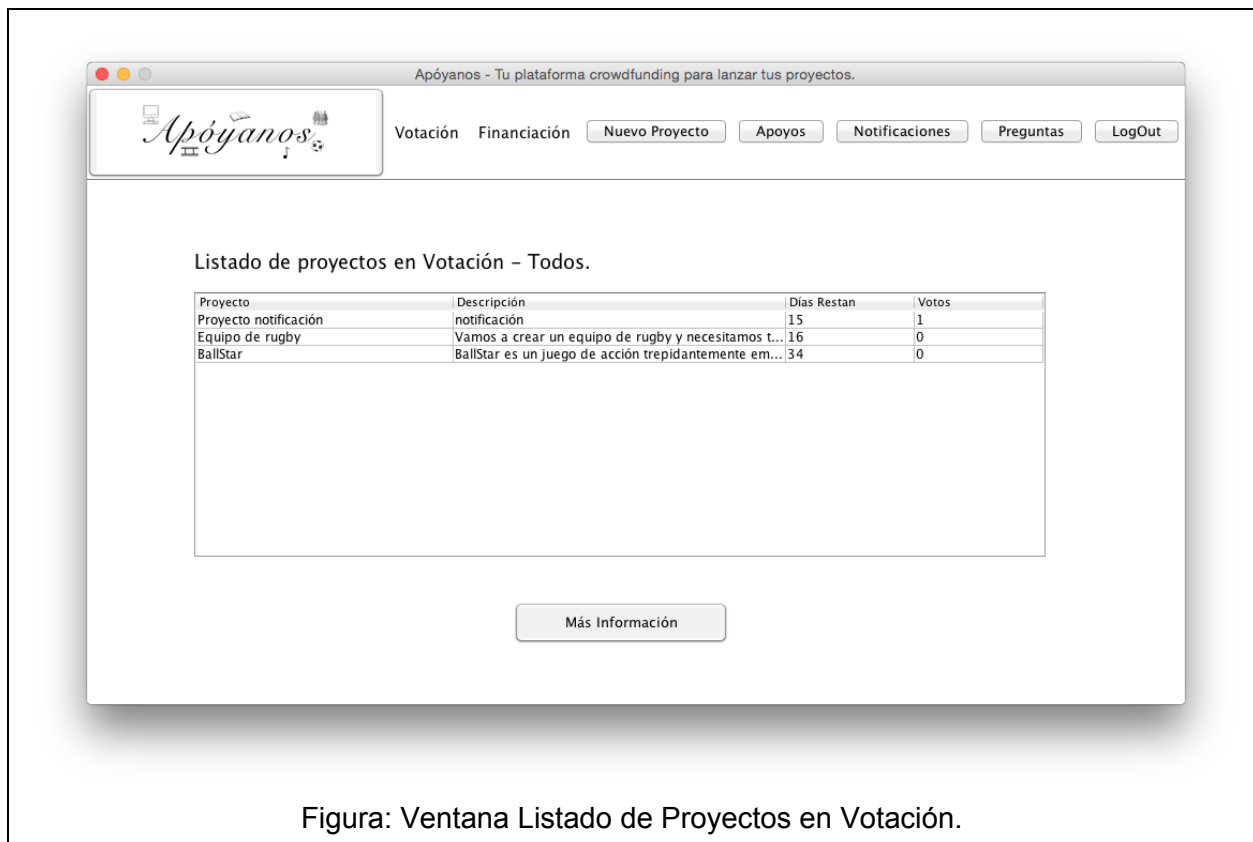


- **Logo de la aplicación Apóyanos:** Es un botón que lleva a la ventana de inicio/presentación de la aplicación.

## Ventana Votación

Desde la *Ventana Votación* se verá un listado ordenado decrecientemente por el número de votos, primero los que más votos tengan. Este listado contendrá a todos los proyectos o sólo aquellos que sean de la categoría seleccionada.

Seleccionando un *proyecto* y pulsando el botón “*Más Info*” la aplicación muestra información más detallada del mismo. (*Para saber más ir a Ventana Información Proyecto en Votación*)\*





## Ventana Financiación

Desde la *Ventana Financiación* se verá un listado ordenado el número de días para que estos finalicen. Este listado contendrá a todos los proyectos a sólo aquellos que sean de la categoría seleccionada.

Seleccionando un *proyecto* y pulsando el botón “*Más Información*” la aplicación muestra información más detallada del mismo. (*Para saber más ir a Ventana Información Proyecto en Financiación*)\*



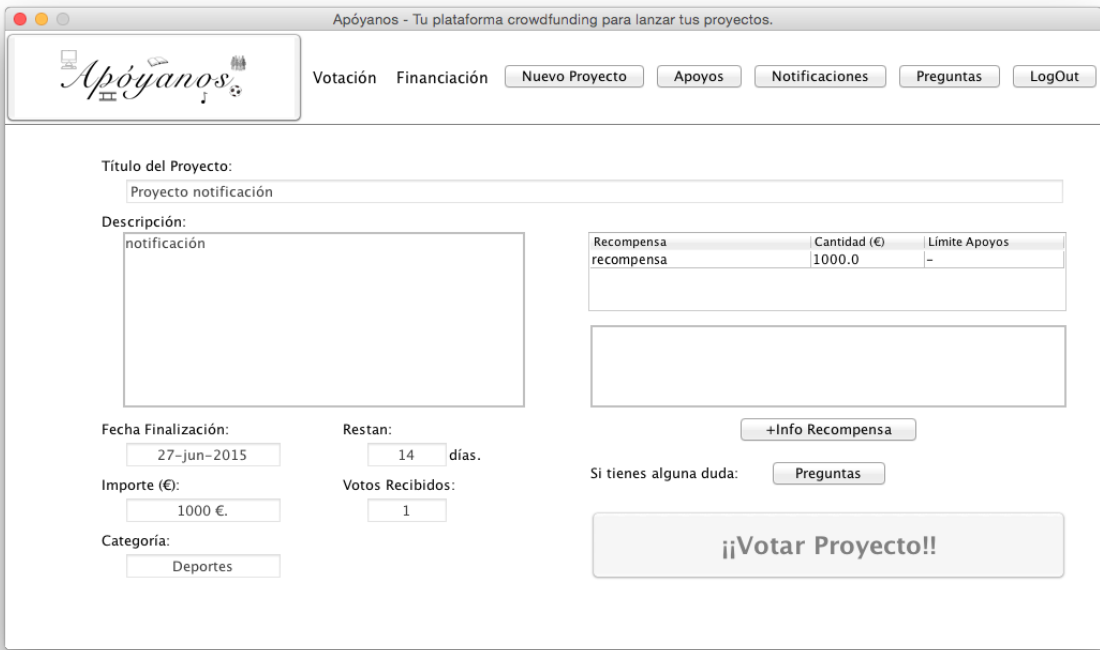
## Ventana Información Proyecto en Votación

Esta ventana muestra un formulario con toda la información relativa al *proyecto*: Nombre, descripción, fecha límite para alcanzar el proyecto, cantidad de dinero a recaudar, creador del proyecto, categoría, número de votos recibidos y recompensas. En las recompensas seleccionando una y pulsando “+Info Recompensas” verá la descripción de la misma.

Desde esta ventana un *usuario* también puede acceder a una *ventana de preguntas* para obtener más información de un *proyecto*.

El *usuario* también puede *votar* el *proyecto* pulsando el botón **¡¡Votar Proyecto!!**

Un usuario no podrá votar su propio proyecto.



The screenshot shows a web browser window titled 'Apóyanos - Tu plataforma crowdfunding para lanzar tus proyectos.' The navigation bar includes links for 'Votación', 'Financiación', 'Nuevo Proyecto', 'Apoyos', 'Notificaciones', 'Preguntas', and 'LogOut'. The main content area displays the following information:

- Título del Proyecto:** Proyecto notificación
- Descripción:** notificación
- Recompensa:** recompensa
- Cantidad (€):** 1000.0
- Límite Apoyos:** ~
- Fecha Finalización:** 27-jun-2015
- Restan:** 14 días.
- Importe (€):** 1000 €.
- Votos Recibidos:** 1
- Categoría:** Deportes

Additional features include a '+Info Recompensa' button, a 'Preguntas' button for users with doubts, and a large '¡¡Votar Proyecto!!' button.

Figura: Ventana Información Proyecto en Votación.

## Ventana Información Proyecto en Financiación

Esta ventana muestra un formulario con toda la información relativa al *proyecto*: Nombre, descripción, fecha límite para alcanzar el proyecto, cantidad de dinero a recaudar, creador del proyecto, días que restan para que finalice el proyecto, dinero recaudado, porcentaje del dinero

recaudado, categoría y recompensas. En las *recompensas* seleccionando una y pulsando “+Info Recompensas” verá la descripción de la misma.

Desde esta ventana un *usuario* también puede acceder a una *ventana de preguntas* para obtener más información de un *proyecto*.

El *usuario* desde esta ventana tiene la opción de convertirse en mecenas de un *proyecto* apoyándolo, en esta también puede votar el proyecto pulsando el botón *¡¡Votar Proyecto!!*

The screenshot shows a web browser window titled 'Apóyanos - Tu plataforma crowdfunding para lanzar tus proyectos.' The interface includes a navigation bar with links: 'Votación', 'Financiación', 'Nuevo Proyecto', 'Apoyos', 'Notificaciones', 'Preguntas', and 'Logout'. The main content area displays project information for 'A la luna'. The title is 'A la luna' and the description is 'Vamos a la luna en un cohete, nos damos un paseo y nos volvemos.' A table lists rewards: 'Luna1' (10.0, unlimited), 'Luna2' (200.0, 100), 'Luna3' (500.0, 10), and 'Astronauta' (5000.0, 1). Below the table is a '+Info Recompensa' button. Project statistics show a completion date of 17/06/2015, 4 days remaining, a total amount of 10000 €, and a current amount of 10080 € (100.8% funded). A 'Preguntas' button is available for users with questions. A large button at the bottom right says '¡¡Apoyar Proyecto!!'.

Recompensa	Cantidad (€)	Límite Apoyos
Luna1	10.0	-
Luna2	200.0	100
Luna3	500.0	10
Astronauta	5000.0	1

Fecha Finalización:	Restan:
17/06/2015	4 días.

Importe (€):	Financiado:
10000 €.	10080 €.

Categoría:	% Financiado:
Otros	100,8

Si tienes alguna duda: Preguntas

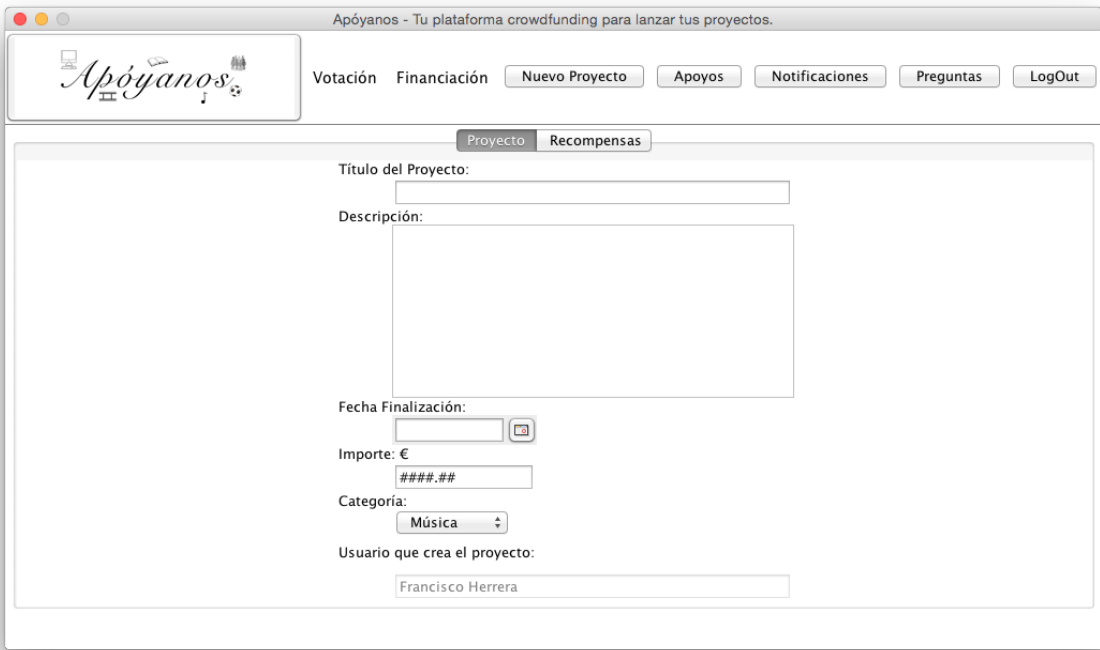
**¡¡Apoyar Proyecto!!**

## Ventana Nuevo Proyecto

En esta ventana un *usuario* creará un *proyecto* rellenando los campos que les solicita. Se divide en dos partes: *Proyecto* y *Recompensas*.

En la parte *Proyecto* el *usuario* rellenará los datos genéricos del mismo: nombre, descripción, fecha de finalización, importe y categoría.

Todos los campos de esta pestaña son obligatorios y no se podrá dar de alta un *proyecto* si no están debidamente cumplimentados.



The screenshot shows a web browser window titled "Apóyanos - Tu plataforma crowdfunding para lanzar tus proyectos." The navigation bar includes links for "Votación", "Financiación", "Nuevo Proyecto" (active), "Apoyos", "Notificaciones", "Preguntas", and "LogOut". The main content area has two tabs: "Proyecto" (active) and "Recompensas". The "Proyecto" tab contains the following form fields:


- Título del Proyecto:
- Descripción:
- Fecha Finalización:  
- Importe: €
- Categoría:
- Usuario que crea el proyecto:

Figura: Nuevo Proyecto. Pestaña Proyecto.

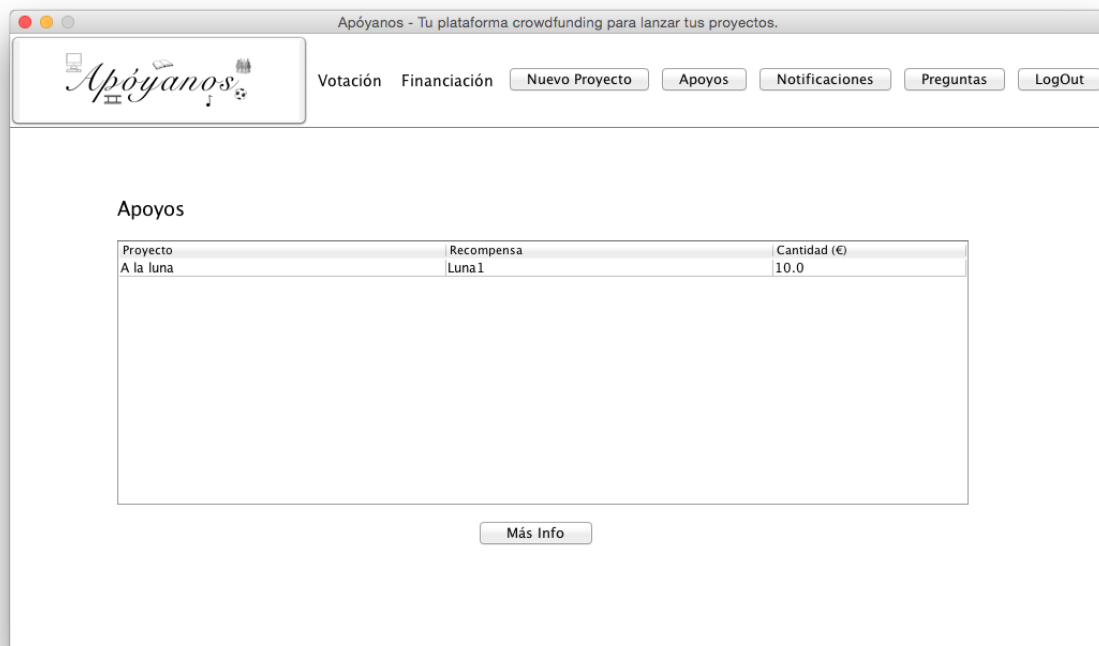
Por su parte, en *Recompensa* es el formulario donde un usuario irá incluyendo las recompensas que van asociadas al *proyecto*. Todos los campos del formulario *recompensa* son obligatorios excepto *Máximo apoyos*. Mientras no se registre el *proyecto* un *usuario* podrá ir creando *nuevas recompensas*, modificar y/o eliminar las existentes. Para registrar un proyecto, además de estar cumplimentado el formulario de la pestaña *Proyecto* deberá haber como **mínimo una *recompensa***.

The screenshot displays the 'Apóyanos' crowdfunding platform interface. The browser window title is 'Apóyanos - Tu plataforma crowdfunding para lanzar tus proyectos.' The navigation bar includes links for 'Votación', 'Financiación', 'Nuevo Proyecto', 'Apoyos', 'Notificaciones', 'Preguntas', and 'LogOut'. The main content area has two tabs: 'Proyecto' and 'Recompensas', with the latter being active. The 'Recompensas' form contains the following fields and elements:

- Recompensa:** A text input field.
- Cantidad:** A text input field followed by the label 'Euros.'
- Descripción:** A large text area.
- Máximo apoyos:** A text input field with a note '\* En blanco si no hay límite.'
- Table:** A table with three columns: 'Recompensa', 'Cantidad (€)', and 'Límite Apoyos'. It is currently empty.
- Buttons:** 'Añadir', 'Limpiar', 'Registrar Proyecto', 'Editar', and 'Eliminar'.

## Ventana Apoyos

Cuando un *usuario* entra en esta ventana verá un listado con los proyectos a los que ha apoyado junto a la *recompensa* elegida y el valor de la misma. Seleccionando un *apoyo* de la lista y pulsando el botón “*Más Info*” nos llevaría a la ventana ***Información Proyecto en Financiación*** donde se puede ver de nuevo toda la información del *proyecto*.



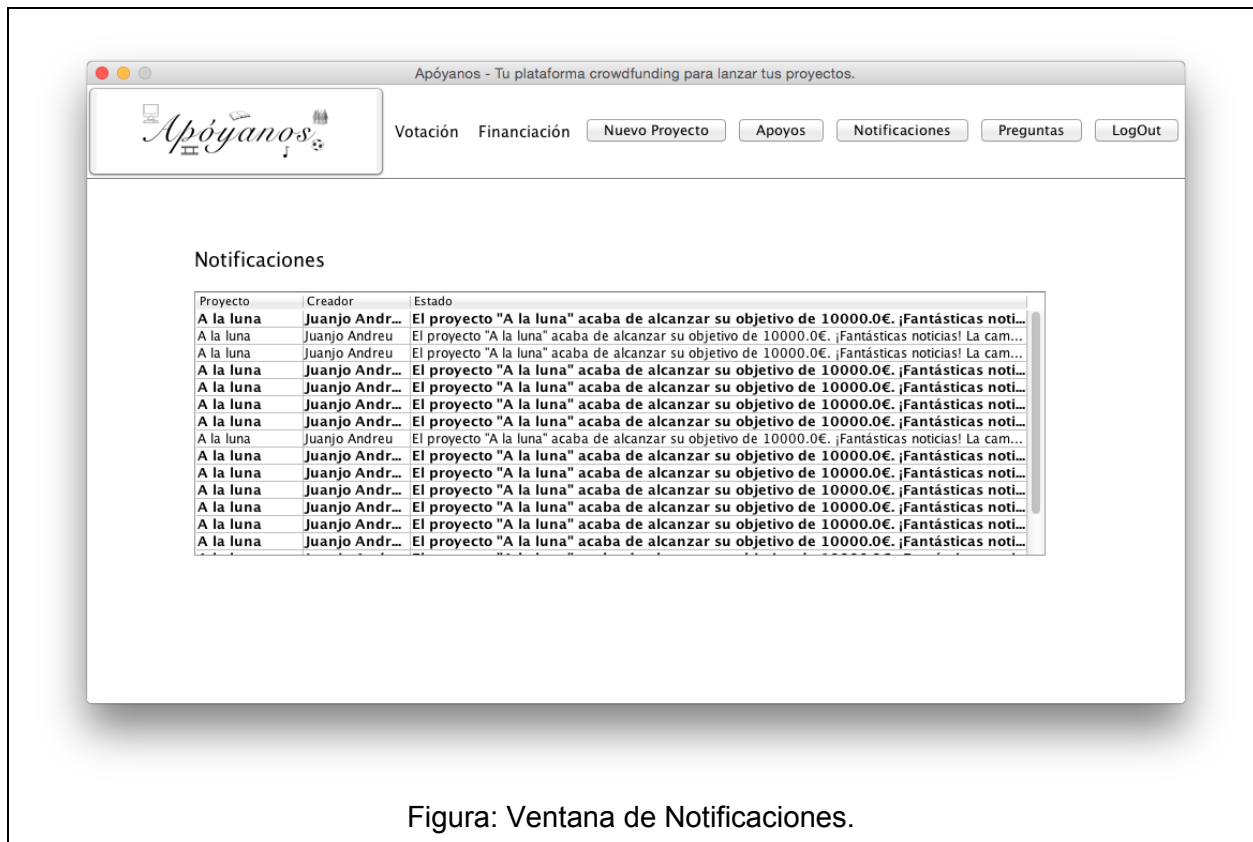
Proyecto	Recompensa	Cantidad (€)
A la luna	Luna1	10.0

Más Info

Figura: Ventana Apoyos.

## Ventana Notificaciones

Cuando un *usuario* entra en esta ventana visualiza un listado informándolo del estado de los *proyectos* a los cuales ha apoyado.



## Ventana Preguntas

En esta ventana un *usuario* cuando entra dispones de dos pestañas correspondientes a los distintos roles que puede tener. Por un lado como posible mecenas que quiere informarse sobre un *proyecto* y por tanto emite o realiza preguntas o como creador de un *proyecto* que las responde.

El acceso a esta ventana es directo pulsando el botón *Preguntas* y le aparecerá un listado con las preguntas emitidas, estén o no respondidas, o desde las ventanas de información de proyecto: *Ventana Información Proyecto en Votación* y *Ventana Información Proyecto en Financiación*.

Accediendo desde estas ventanas, además del listado aparece relleno el formulario de la derecha con el nombre del *proyecto*, el *nombre del creador* del mismo y los campos *Asunto* *Pregunta* y *Respuesta* en blanco. Los campos *Asunto* y *Pregunta* son los que el *usuario* activo cumplimentará para formular una pregunta.

Si un *usuario* hace “doble-clic” sobre uno de los elementos de la lista se cumplimenta el formulario de la derecha con los datos correspondientes.





La parte *Preguntas Recibidas* su funcionalidad es igual que la anterior pero en esta ocasión el rol del *usuario* es el de creador de un *proyecto* y por tanto irá seleccionando las preguntas en **negrita** del listado para responderlas en el formulario. Si *cliquea* sobre una *pregunta* ya respondida sólo podrá leer la información de la misma.

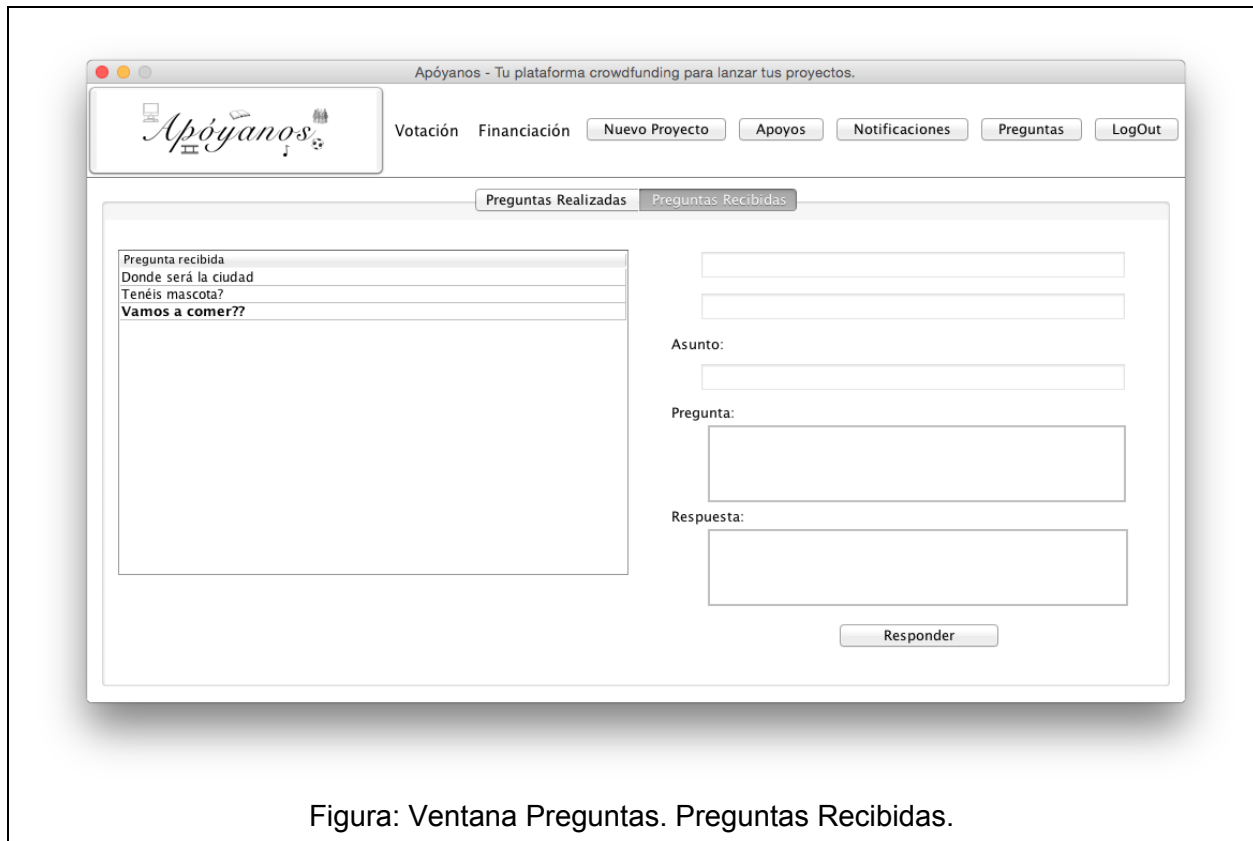


Figura: Ventana Preguntas. Preguntas Recibidas.

# Observaciones finales

## Estimación del tiempo

Trabajo realizado por persona:

3 horas x 3 días semanales: 9 horas semanales.

13 semanas x 9 horas semanales = **117 horas / persona.**

## Apreciaciones personales de la práctica

Una de las cosas interesantes de la práctica es que intenta recoger la problemática que nos encontraríamos a la hora de afrontar un proyecto completo:

- Diagramas de clases del dominio.
- Diagramas de colaboración.
- Diseño de la aplicación.
- Uso de patrones en el diseño.
- Test de pruebas, tanto unitarios como de integración.
- Diseño de la interfaz gráfica.
- Y hasta un manual de usuario.

Todas estas cuestiones recaen en un **grupo de trabajo de dos personas**, cada una con sus habilidades, donde el resultado en la práctica es un volumen de trabajo horas/persona bastante alto a nuestro juicio.