## Comentarios generales:

Un proyecto muy justo, al límite de suspender. Si no corregís las cosas que están mal y faltan, es probable que suspendáis en la entrega de la interfaz gráfica.

## Diseño:

Habéis puesto muchos métodos en donde no son. Este es un error grave de diseño. Si hacéis esto en el examen vais a suspender. Por ejemplo, habéis puesto en usuario enviarAFinaciar(). Habéis confundido actor usuario con objeto usuario. Ese método debería estar en proyecto.

No hay encapsulación en los objetos. Hacéis cosas como Sistema.getInstancia().getProyectos().add(p). El añadir un proyecto al sistema debería ser un método de sistema.

## Junit:

Los test no son exhaustivos. En colectivo, solo se prueba un método, por ejemplo. Lo mismo para proyectos.

Los test dan errores.

El validador no prueba toda la funcionalidad de la aplicación.

## Gestión Global Aplicación:

La carga de distritos es incorrecta, pues tiene una ruta absoluta para windows. La aplicación solo funciona en windows. Debería funcionar en cualquier SO.

La serialización es incorrecta. No se guardan los atributos estáticos. Hay que guardarlos individualmente. Esto significa que la implementación de salvado no funcionará correctamente. Basta con actualizar el atributo instancia de Sistema. El guardado de datos de sistema debería ser transparente y hacerse dentro de sistema (en el constructor, o al cerrar sesión). Ahora se hace fuera. Es mal diseño.

El método LogIn debería ser logIn. Es mala notación empezar métodos con mayúscula. Esto en general. Deberíais guardar en sistema el usuario que ha iniciado sesión(USUARIO O LISTA DE USUARIOS?). Es mal diseño poner un atributo de usuario a true para indicar esto.

La caducidad de proyectos está a medio implementar. No se llama nunca a comprobarCaducidad.

La fijación del nº de apoyos para financiar un proyecto es incorrecta. Debería ser un atributo global. Es el mismo para todos los proyectos. Os lo dije en el feedback de diseño. No lo habéis tenido en cuenta.

La carga de los distritos no es transparente. Hay que invocar el método leerDistritos. Debería ser automática, por ejemplo, en el constructor.

## Gestión de Ciudadanos:

No comprobáis que el login no esté repetido al registrar un usuario. /\*a que se refiere\*/

Bloquear usuario aceptar etc. Debería estar en usuario. Lo mismo para proyecto. Es mal diseño. Parece que estos métodos están repetidos en sistema y en usuario (que reciben el usuario a modificar, lo que no tiene sentido). Parece que os gusta programar más de la cuenta.

Tiene que ser posible sacar los proyectos que ha apoyado un usuario a título individual o como representante de colectivo. Deberíais tener atributos en usuario para guardar esa información.

## Gestión de Proyectos:

El cambio de estado de proyecto no es automático. Por ejemplo, hay que enviar a financiar para que el proyecto pase a financiable.

La creación de proyectos está en Usuario. Es muy mal diseño. Debería estar en sistema. Si no sabéis por qué, vais a suspender el examen de la asignatura. Se debería verificar el tamaño de los campos al crear proyecto. No guardáis en usuario los proyectos creados.

Las imágenes de los proyectos de infraestructura se deben copiar a alguna carpeta de la aplicación. Se debe borrar si se rechaza el proyecto.

Parece que al cambiar el estado del proyecto no notificáis. Hay que notificar a los suscriptores.

Al aceptar proyecto no comprobáis que sea el administrador, lo mismo para usuarios.

Falta el estado rechazado en proyecto. Al rechazar hay que indicar motivo (por parte del administrador).

No es posible suscribirse a un proyecto. Las notificaciones no están implementadas. Esto si no lo arregláis para la siguiente entrega suspenderéis el proyecto. Lista de suscriptores y meterlos al suscribierte

El apoyo a proyectos como representante de colectivo está mal. Se debe poder especificar el colectivo que apoya el proyecto. En vuestro caso son todos para los que es representante el usuario.

El cálculo de los apoyos recibidos es incorrecto. Si un usuario está bloqueado, no contaría su apoyo. Su estado puede cambiar de bloqueado a desbloqueado. No lo tenéis en cuenta. Sería mucho más fácil si tuvieséis un método que calcula los votos recibidos. Los colectivos pueden incluir subcolectivos. No lo habéis tenido en cuenta al votar. Si no corregís esto en la siguiente entrega el proyecto no será apto.

Os falta un estado en proyecto que es VOTOS ALCANZADOS que indique que el proyecto se puede enviar a financiar. (FINANCIABLE)

La solicitud de financiación es incorrecta. Debéis guardar la clave que devuelve submitRrequest para seguir el estado del proyecto.

En caso de error por no poderse comunicar con el sistema CCGG habría que reintentar la funcionalidad, hasta un número máximo de veces. No es posible consultar el estado de la financiación.

## Gestión de Colectivos:

Al añadir usuario a colectivo no comprobáis si ya pertenece a él de forma directa o indirecta (a través de un subcolectivo).

Al crear subcolectivo no actualizáis colectivo padre.

## Gestión de Notificaciones:

Hay clase notificación, pero las notificaciones no están hechas.

Preguntar casos en los que usan notificaciones.

# Respuestas

* Solo un usuario loggeado.
* Apoyar proyecto a proyecto.
* Recomienda usar función apoyos.
* Ruta imagen donde se ha copiado la que se guarda.
* Notificaciones en usuario, con estado de leídas o no.
* Cada vez que cambia de estado, notificación

Recorrer subcolectivos bien.

Recorrer proyectos

suscribirte y apoyar son cosas distintas

SI ESTAS SUSCRITO NOTIFICACIONES