

Facultad de Ciencias Exactas, Físicas y Naturales - UNC



GRUPO SOFTDUINO

Ingeniería de Software

SISTEMA DE CONTROL DE DOSIFICACION

INTEGRANTES: DETKE, Ramiro – CASTRO, Pedro – D'Andrea, Franco

PROFESORES: MICELI, Martin – NONINO Julián

TABLA DE CONTENIDOS

NOTA DE ENTREGA	3
MANEJO DE LAS CONFIGURACIONES	3
REQUERIMIENTOS.....	3
ARQUITECTURA.....	3
DISEÑO E IMPLEMENTACION.....	4
PRUEBAS UNITARIAS Y DEL SISTEMA.....	8
DATOS HISTORICOS	8
INFORMACION ADICIONAL.....	8

NOTA DE ENTREGA

El documento de nota de entrega (Release Note) está disponible en el siguiente link:

MANEJO DE LAS CONFIGURACIONES

El documento del plan de manejo de las configuraciones está disponible en el siguiente link:

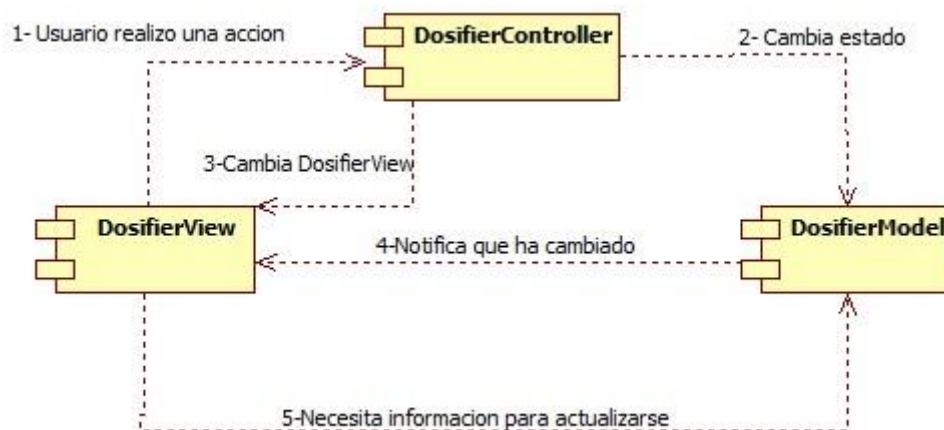
<https://github.com/castrosoft/IngSoft-2016-SoftDuino/blob/master/docs/CMPlan%20DosifierModel.md>

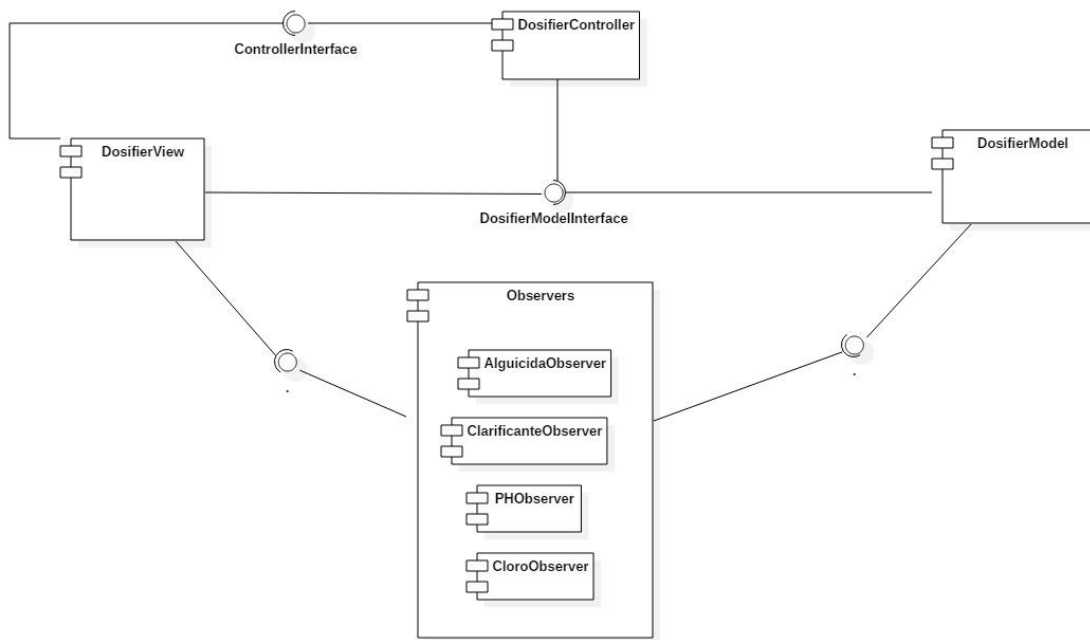
REQUERIMIENTOS

El documento de Requerimientos (SRC) está disponible en el siguiente link:

<https://github.com/castrosoft/IngSoft-2016-SoftDuino/blob/master/docs/Requerimientos.md>

ARQUITECTURA





Dado que hay múltiples formas de ver e interactuar con los datos, se usa el patrón de arquitectura MVC (Model View Controller). Permite la relativa independencia de vistas, modelos y controladores lo cual facilita la comunicación del sistema dosificador con la aplicación de control.

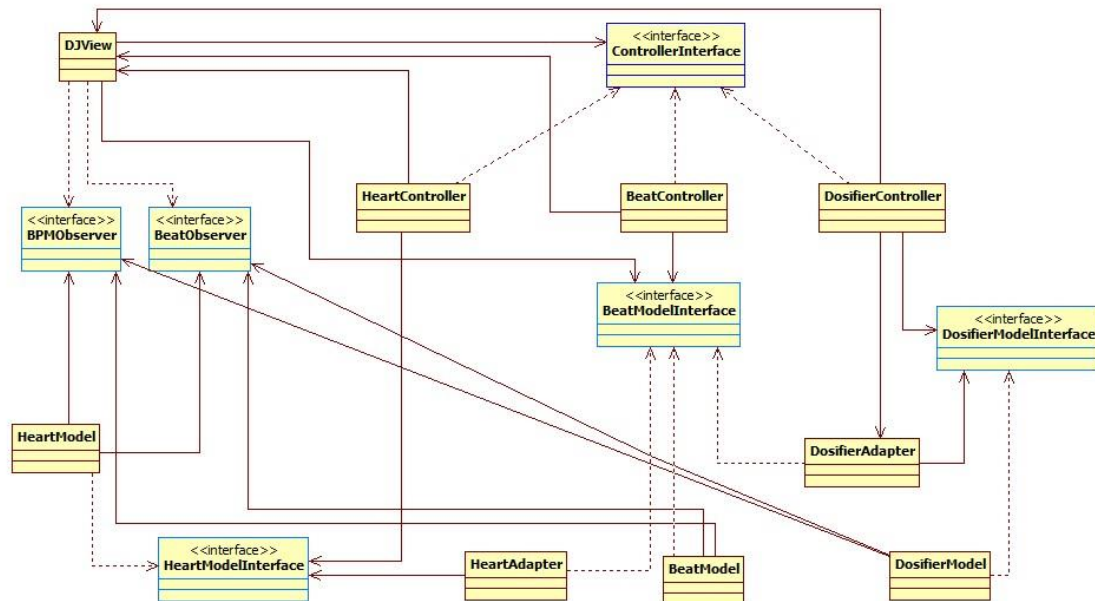
DISEÑO E IMPLEMENTACION

Se describen dos situaciones:

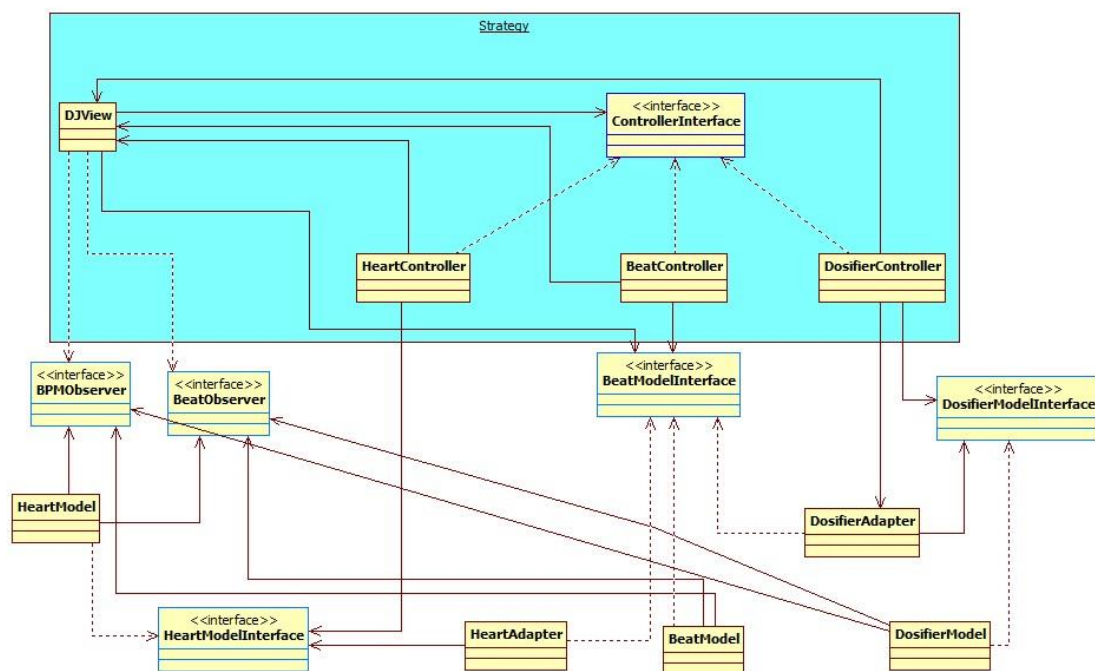
- Cuando se agrega nuestro modelo con las vistas ya existentes y
- Cuando creamos una vista propia para nuestro modelo.

Para integrar nuestro modelo con la vista ya existente se reutilizo los patrones de diseño Strategy y Observer ya implementados en el código base.

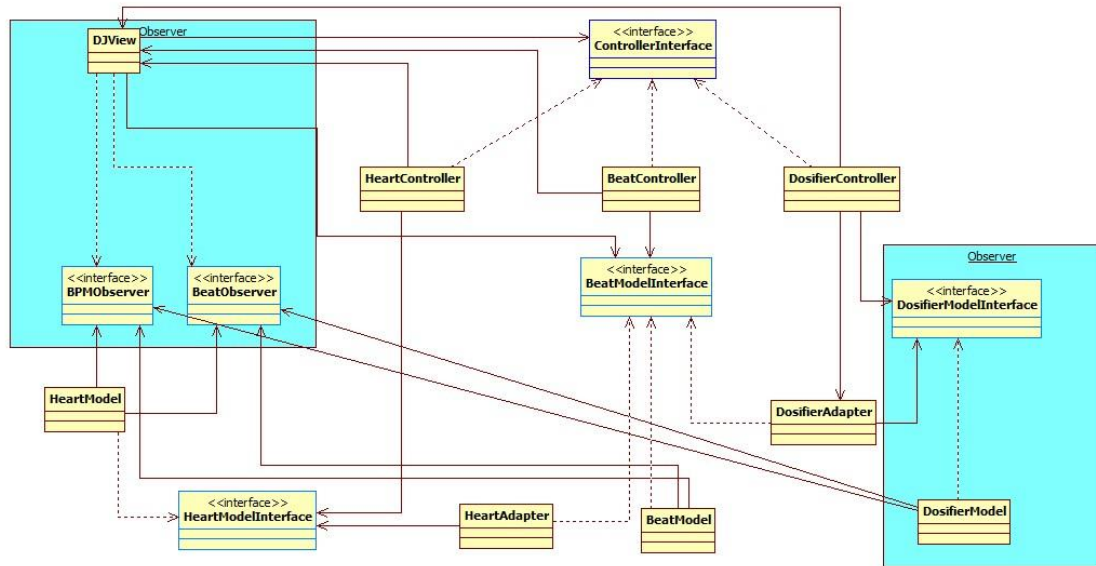
A continuación se muestra como quedo el diagrama de clases con la integración ya realizada.



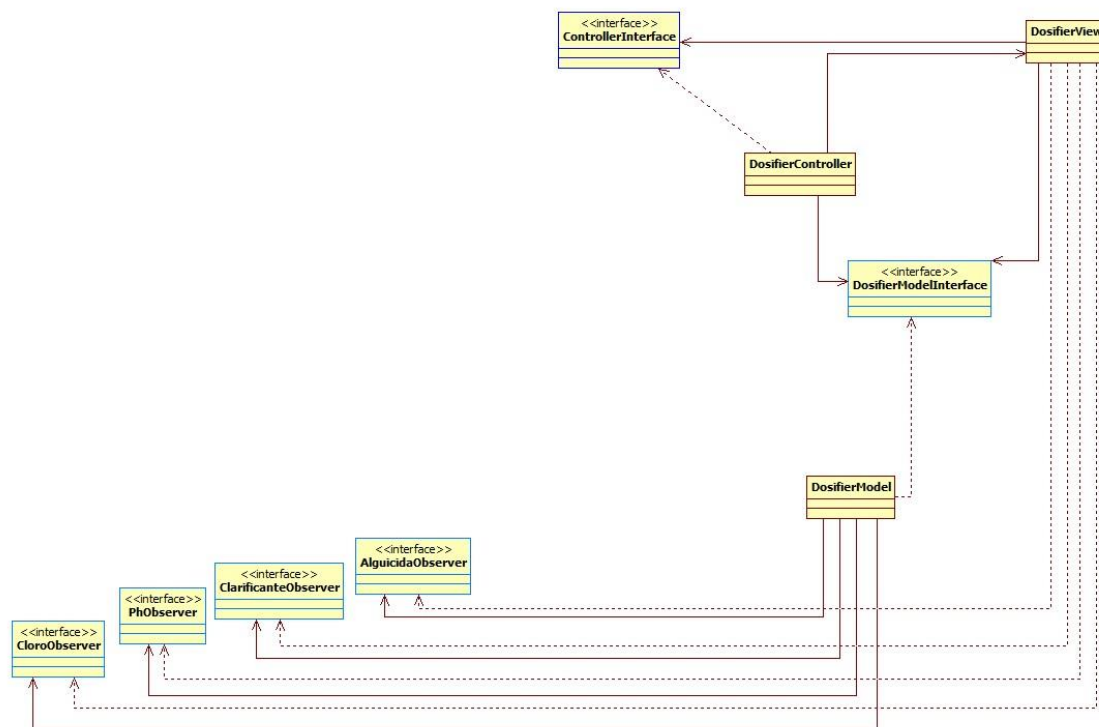
Se evidencia el patrón Strategy con el agregado de un nuevo controlador (DosifierController) para nuestro modelo.



Se evidencia el patrón Observer donde el nuevo modelo (DosifierModel) actúa como sujeto (es observado).



Se aprecia como se usó un patrón de arquitectura MVC porque contiene un modelo que interactúa con la vista a través del controlador



Se implementó el patrón Observer para que nuestra vista obtenga los valores de las variables de nuestro modelo.

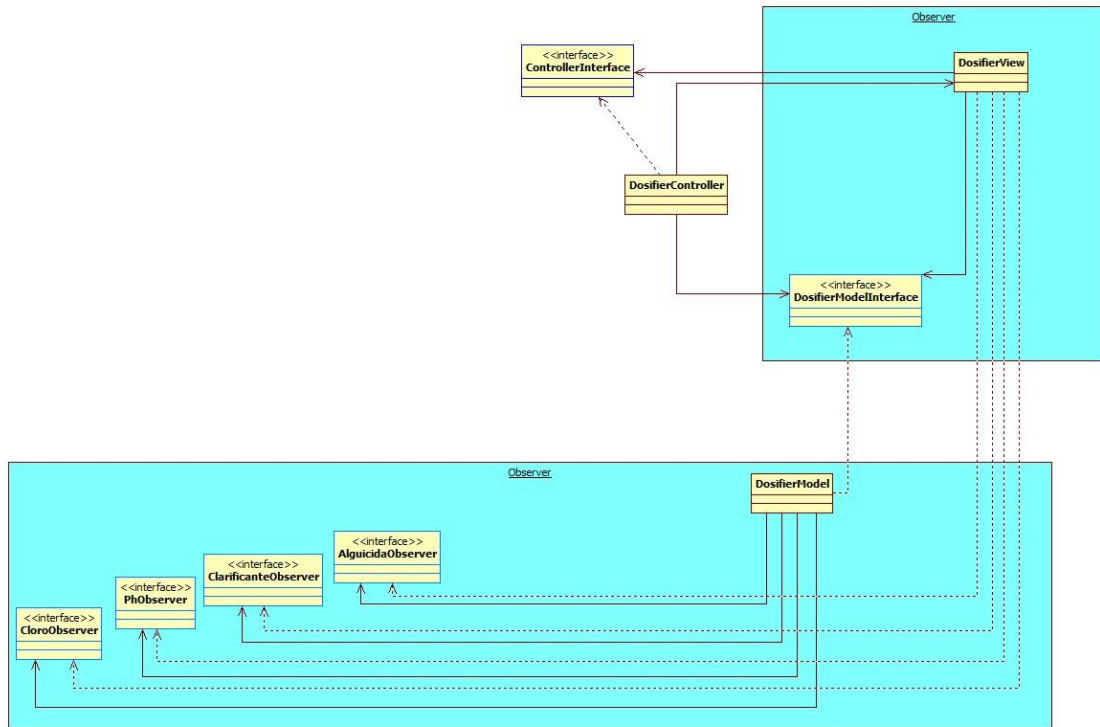
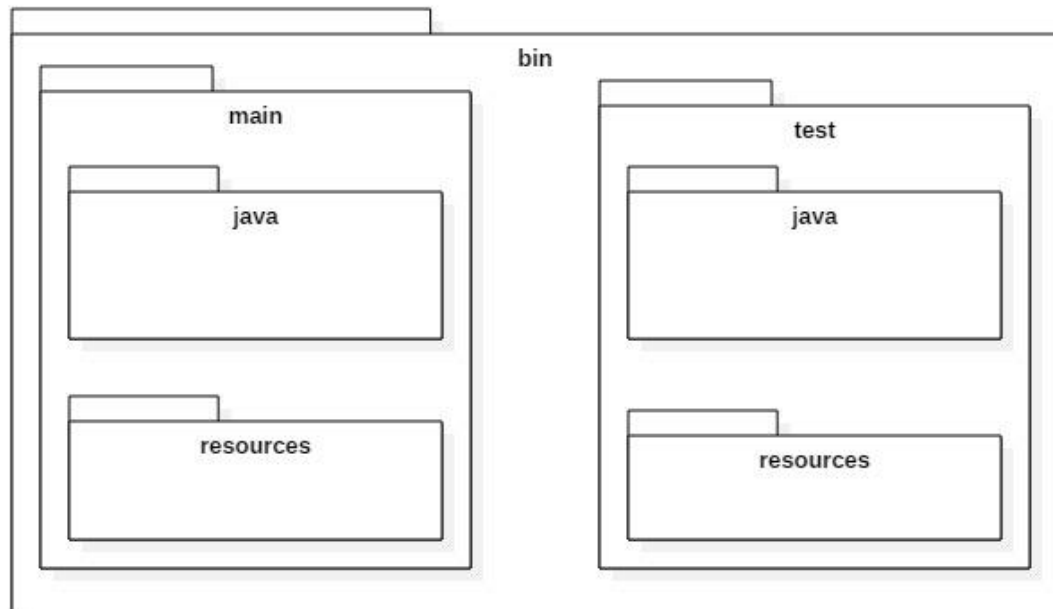


Diagrama de paquetes



PRUEBAS UNITARIAS Y DEL SISTEMA

El documento de Pruebas está disponible en el siguiente link:

<https://github.com/castrosoft/IngSoft-2016-SoftDuino/blob/master/docs/DocumentoDePruebas.md>

DATOS HISTORICOS

INFORMACION ADICIONAL
