INGENIERÍA DEL SOFTWARE II

AUTORES

JULIÁN ANDRES CASTRO RUGE. CÓDIGO: 1410013519

ANTONIO CARLOS PATERNINA OROZCO. CÓDIGO: 1221070176

JHOJAN STEVEN QUINTERO MELO. CODIGO:1420014038

DAVID HERNANDO MACIAS VILLALOBOS CODIGO: 1410013446

SEGUNDA ENTREGA DE PROYECTO DE AULA

TUTOR

DIEGO OLIVEROS

POLITÉCNICO GRANCOLOMBIANO

FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS

INGENIERÍA DEL SOFTWARE

BOGOTÁ

2017

TABLA DE CONTENIDOS

TABLA DE CONTENIDOS 2

1 Implementación 4

1.1 Funcionamiento (Mínimo 3 Funciones especializadas para el negocio (Entrega de valor para el negocio)) 4

1.2 Coherencia diseño (arquitectura y patrones) 5

1.3 Presentación 5

1.3.1 Despliegue de la solución servidor público 5

1.3.2 Interface responsive (compatibilidad móviles) 5

1.3.3 Manejo de repositorios Informes de actividad 6

2 Documentación 8

2.1 Requerimientos 8

2.1.1 Modelado del negocio (Canvas) 8

2.1.2 Objetivos SMART (general y especifico) 8

2.1.3 Documento de levantamiento de requerimientos (deseable IEE 830) 8

2.1.4 Modelado de casos de Uso + fichas. 8

2.1.5 Diagramas de clases 8

2.1.6 Diagramas de secuencia 8

2.1.7 Diagramas de navegación Mockups 8

2.1.8 Diseño BI (Reporte gráfico su solución) 8

2.2 Planeación 8

2.2.1 Perfilado del equipo de trabajo 8

2.2.2 Selección de metodología 8

2.2.3 History Map- Cronograma. 8

2.2.4 Release planning- 8

2.2.5 Estimación por puntos de función. 8

2.2.6 BurnDown chart Seguimiento de actividades (Reportes de velocidad de código). 8

2.3 Diseño 8

2.3.1 Patrones de diseño (50%) 8

2.3.2 Explicación 8

2.3.3 Diagrama UML de patrones 8

2.3.4 Arquitectura del sistema (50%) 8

2.3.5 Diagrama de infraestructura 8

2.3.6 Estilos: Diagrama de capas –diagrama componentes 8

2.3.7 De Conectores, Interfaces (protocolosTivadis) 8

2.3.8 De las Herramientas y frameworks 8

2.4 Implementación 8

2.4.1 Reporte de funcionamiento 8

2.4.2 documentación 8

2.4.3 Código fuente. 8

2.5 Pruebas 8

2.5.1 Documentación del Código fuente. 8

2.5.2 Plan de pruebas 8

2.5.3 Pruebas- unitarias 8

2.5.4 Logs de pruebas 8

SERVICIOS AL ESTUDIANTE - POLI BOOKING

# Implementación

## Funcionamiento (Mínimo 3 Funciones especializadas para el negocio (Entrega de valor para el negocio))

En esta entrega se logró tener las siguientes funciones especializadas de negocio:

1. Reserva de cubículos de estudio
2. Reserva de cubículos de video
3. Reserva de canchas de fútbol
4. Reserva de canchas de tenis
5. Reserva de cancha múltiple
6. Reserva de gimnasio
7. Cancelación de reservas de cubículos de estudio
8. Cancelación de reservas de cubículos de video
9. Cancelación de reservas de canchas de fútbol
10. Cancelación de reservas de canchas de tenis
11. Cancelación de reservas de cancha múltiple
12. Cancelación de reservas de gimnasio
13. Generación de bloques horarios de cubículos de estudio
14. Generación de bloques horarios de reservas de cubículos de video
15. Generación de bloques horarios de canchas de fútbol
16. Generación de bloques horarios reservas de canchas de tenis
17. Generación de bloques horarios de cancha múltiple
18. Generación de bloques horarios de gimnasio
19. Eliminación de bloques horarios de cubículos de estudio
20. Eliminación de bloques horarios de reservas de cubículos de video
21. Eliminación de bloques horarios de canchas de fútbol
22. Eliminación de bloques horarios reservas de canchas de tenis
23. Eliminación de bloques horarios de cancha múltiple
24. Eliminación de bloques horarios de gimnasio
25. Reportes de reservas vigentes
26. Reporte de reservas histórico

## Coherencia diseño (arquitectura y patrones)

Se ha verificado que todos los diseños se han implementado de acuerdo a lo especificado.

## Presentación

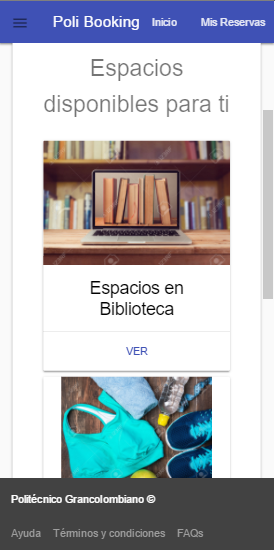
### Despliegue de la solución servidor público

La aplicación se ha desplegado en un servidor en Amazon Web Services (AWS):

La URL así como los datos de ingreso se encuentran en la wiki del proyecto: <https://github.com/ISWPOLI/poli-booking/wiki/Ambiente-QA>

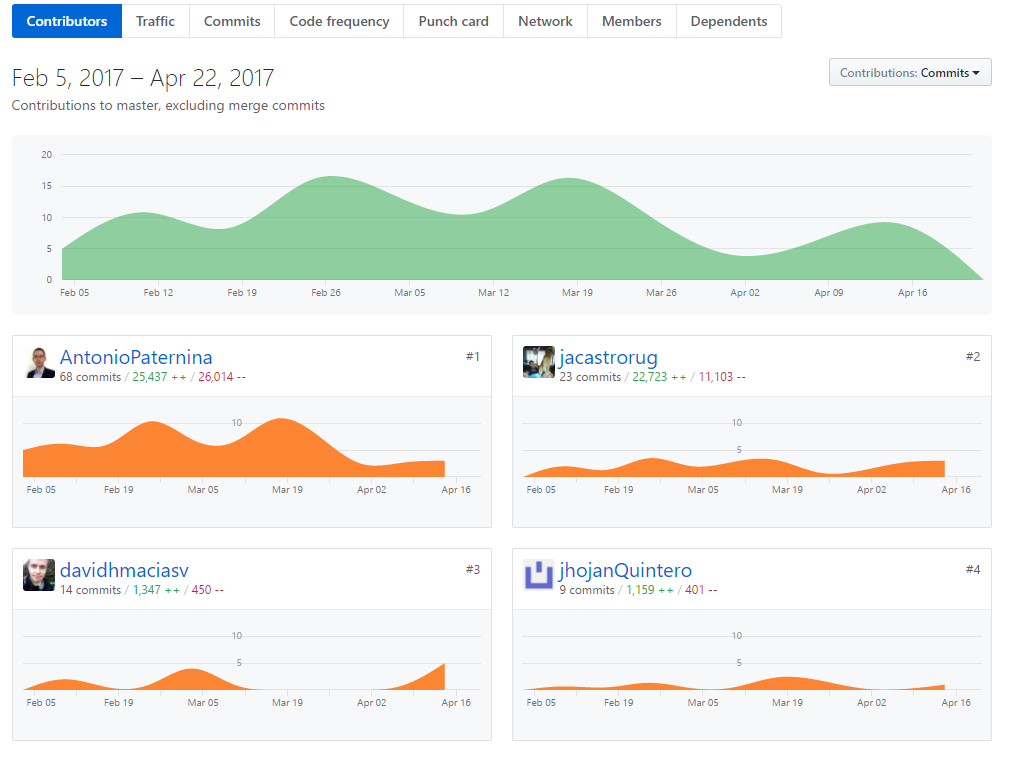
### Interface responsive (compatibilidad móviles)

Se ha verificado que el sitio web es responsive:



### Manejo de repositorios Informes de actividad

Disponible en: <https://github.com/ISWPOLI/poli-booking/graphs/contributors>



# Documentación

## Requerimientos

### Modelado del negocio (Canvas)

Ver **1. Modelado del negocio (Canvas).pdf**

### Objetivos SMART (general y especifico)

Ver **2. Objetivos Smart.pdf**

### Documento de levantamiento de requerimientos (deseable IEE 830)

Ver **3. Documento de levantamiento de requerimientos (IEE830).pdf**

### Modelado de casos de Uso + fichas

Ver **4. Modelado de casos de uso + fichas.png**

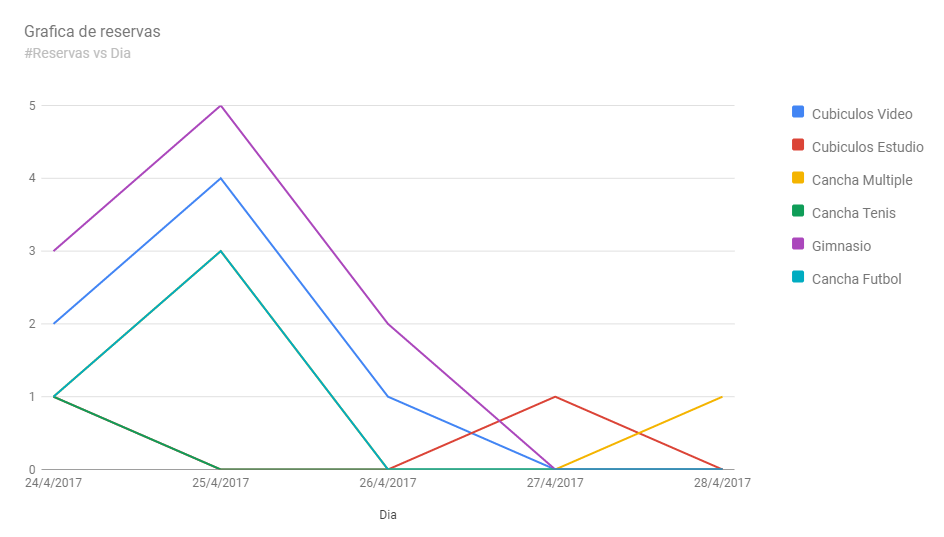
### Diagramas de clases

Ver diagramas en la carpeta **5. Diagramas de clases**

### Diagramas de navegación Mockups

Ver **7. Diagramas de navegación mockups.pdf**

### Diseño BI (Reporte gráfico su solución)



## Planeación

### Perfilado del equipo de trabajo

Ver **1. Perfilado del equipo de trabajo.pdf**

### Selección de metodología

Ver **2. Selección de metodología.pdf**

### History Map- Cronograma.

Ver **3. User Story Map- Cronograma.png**

### Release planning-

Ver **4. Release plan.png**

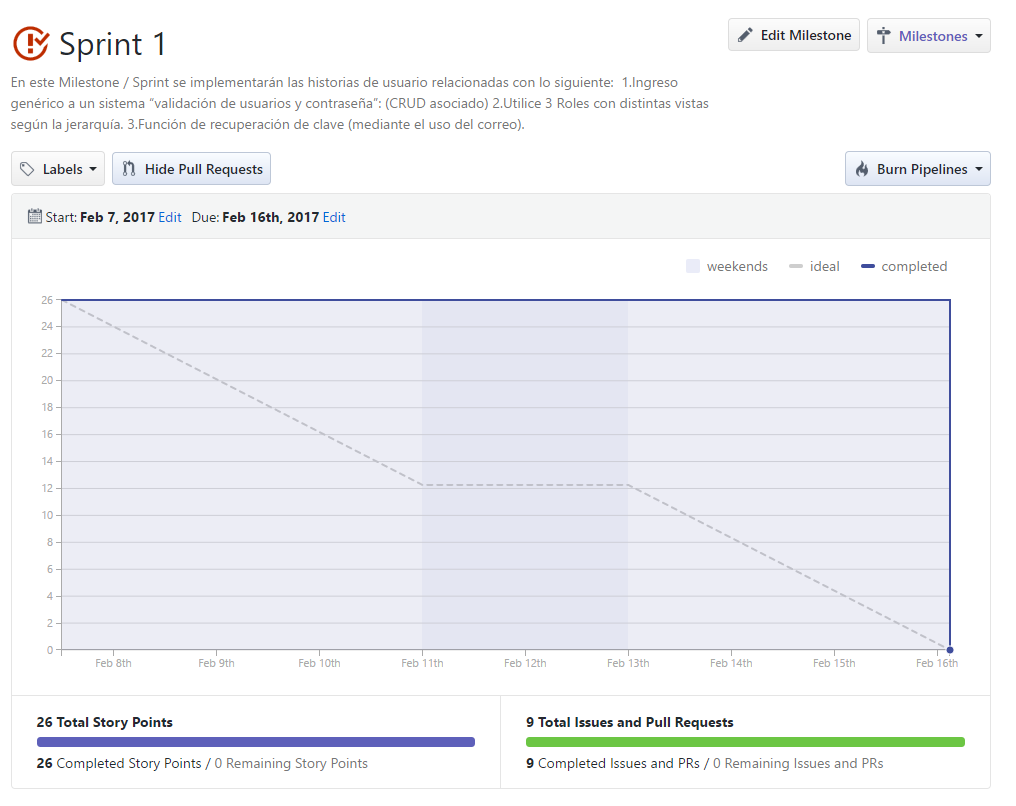
### Estimación por puntos de función.

Ver **5. Estimación por puntos de función.xlsx**

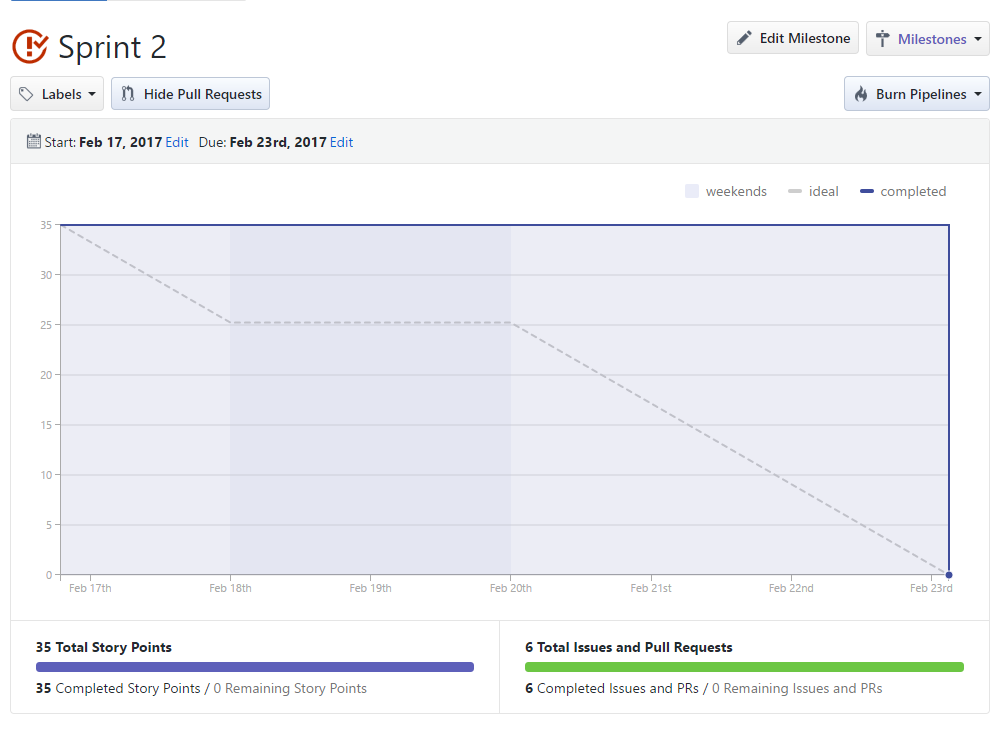
### BurnDown chart Seguimiento de actividades (Reportes de velocidad de código).

A continuación se muestran los gráficos de BurnDown a partir de la gestión de proyecto realizada en Github.

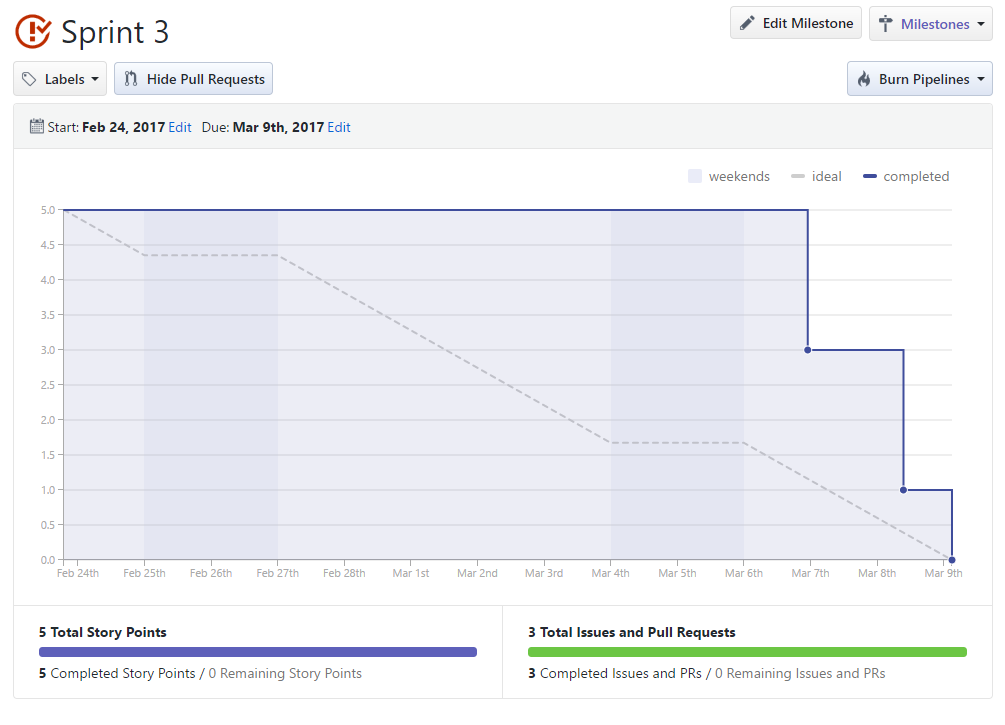
1. Sprint1



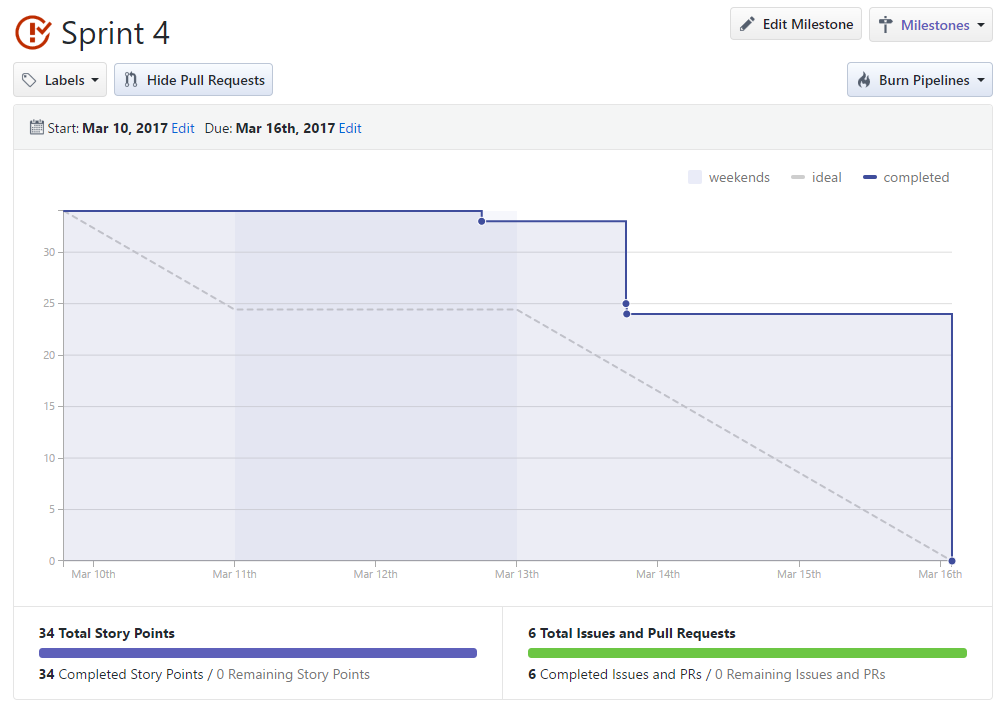
1. Sprint2



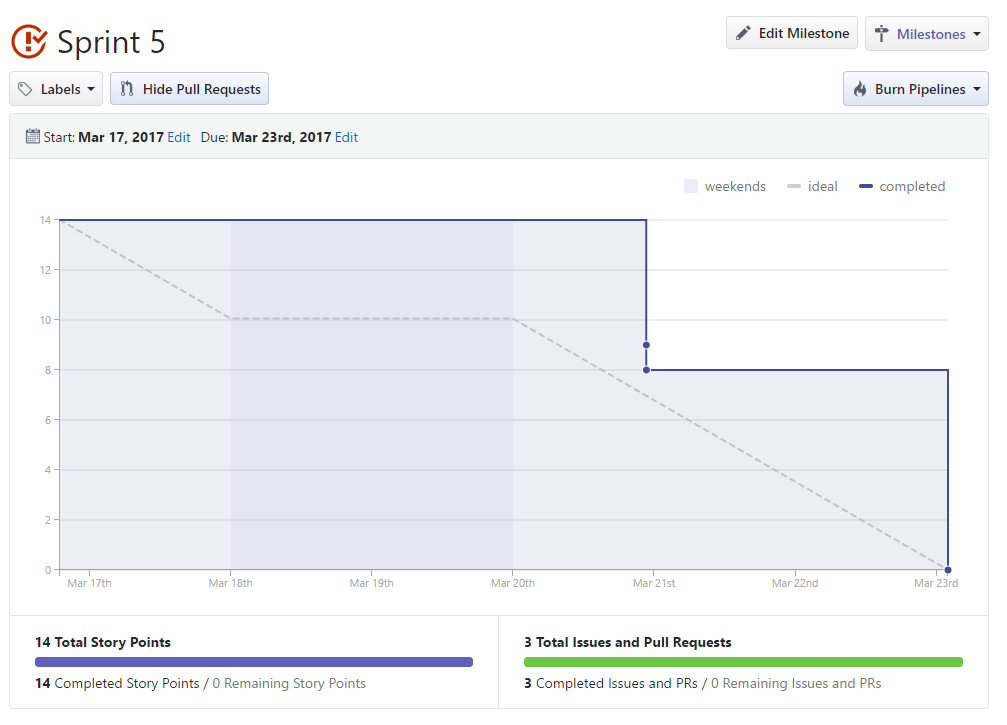
1. Sprint3



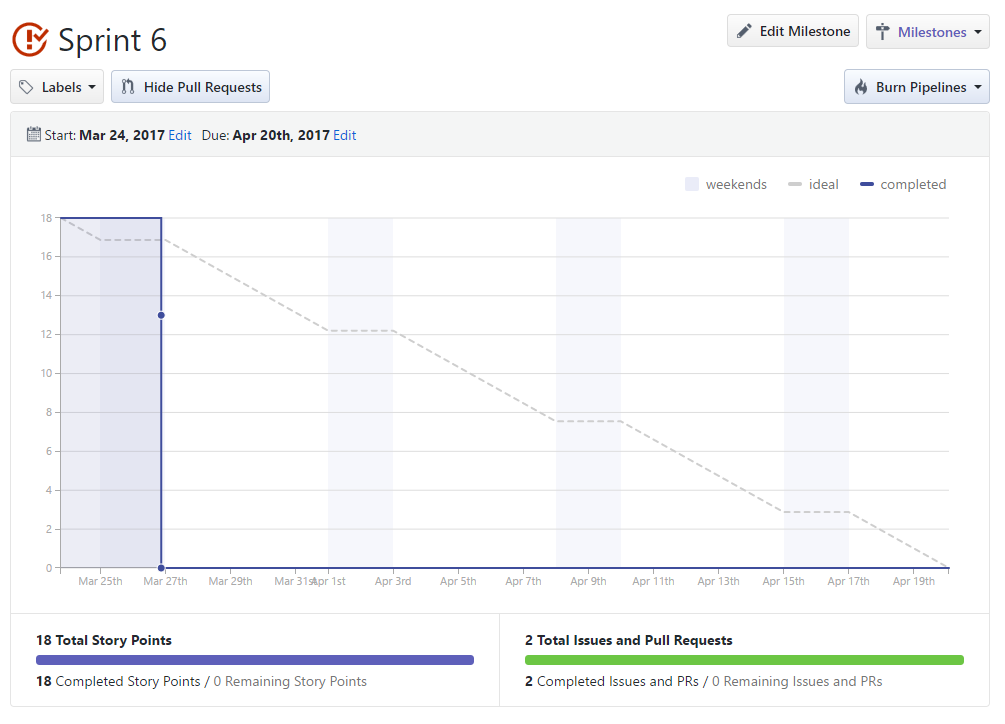
1. Sprint4



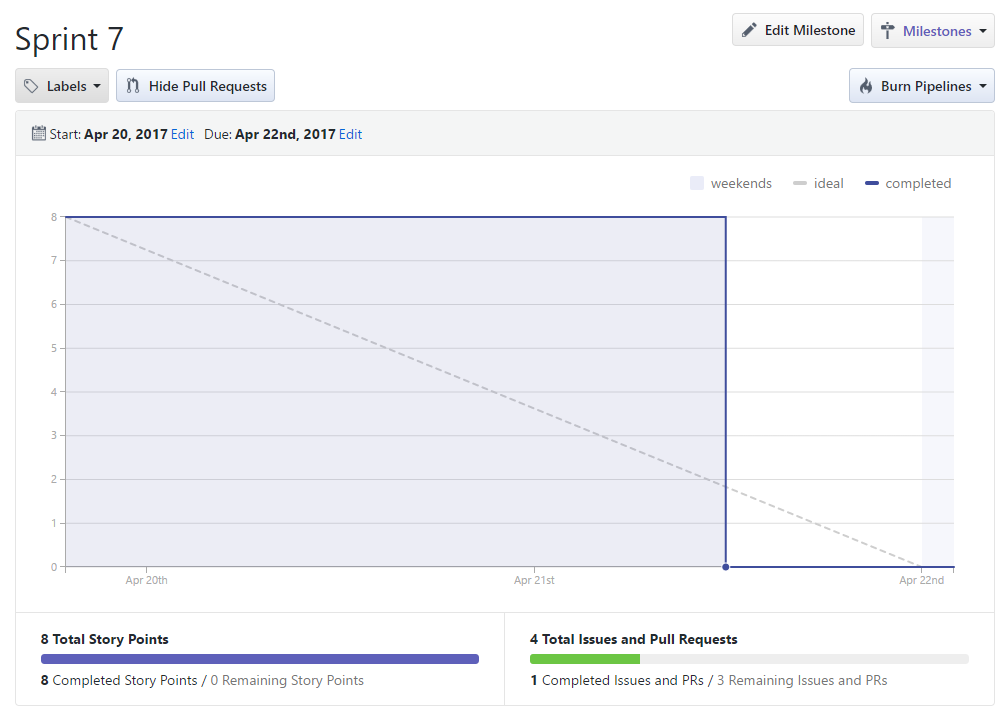
1. Sprint5



1. Sprint6



1. Sprint7



## Diseño

### Patrones de diseño (50%)

Se usaron frameworks con una opinión marcada en cuanto a la organización fundamental de la aplicación lo cual facilitó la adopción de buenas prácticas y patrones.

Por nombrar algunos:

* Observer para el manejo de eventos en Backbone JS
* Singleton para el manejo de componentes de Spring

### Diagrama UML de patrones

Esto se evidencia en los diagramas de UML de la aplicación.

### Arquitectura del sistema (50%)

#### Diagrama de infraestructura

Ver **3.3.1 Diagrama de infraestructura.png**

#### Estilos: Diagrama de capas –diagrama componentes

Ver **3.3.2.1 Diagrama de capas.png y 3.3.2.2 Diagrama de componentes.png**

#### De Conectores, Interfaces (protocolosTivadis)

Ver **3.3.3 Protocolos.png**

#### De las Herramientas y frameworks

Ver **3.3.4 - Arquitectura de las herramientas y frameworks.pdf**

## Implementación

### Código fuente.

Anexo a esta entrega

## Pruebas

### Pruebas unitarias

Ver documentos en carpeta 4. Pruebas- unitarias

### Logs de pruebas

Ver documentos en carpeta 5. Pruebas- funcionales - log de pruebas