



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Sistemas y Tecnologías Web

FUTURA

Informe de prácticas

Daniel Álvarez Medina
(alu0101216126@ull.edu.es)

Dayana Armas Alonso
(alu0101228020@ull.edu.es)

Alberto Mendoza Rodríguez
(alu0101217741@ull.edu.es)

Jeff Pérez Frade
(alu0101038520@ull.edu.es)

Índice

1. Iteración #0	4
1.1 Introducción	4
1.2. Definiendo tareas en Pivotal Tracker	4
2. Iteración #1	6
2.1. Modelo de Datos	6
2.2. Desarrollo de los códigos	9
2.3. Pivotal Tracker.	12
2.3.1. Velocity and points accepted	13
2.3.2. Stories accepted	13
2.3.3. Story cycle time	13
2.3.4. Rejection Rate	14
2.3.5. Burnup	14
2.3.6. Cumulative Flow	15
3. Iteración #2	16
3.1. Introducción	16
3.2. Análisis y Diseño	16
3.3. Implementación.	18
3.4. Pivotal Tracker	25
3.4.1. Velocity and points accepted	25
3.4.2. Stories accepted	26
3.4.3. Story cycle time	26
3.4.4. Rejection Rate	26
3.4.5. Burnup	27
3.4.6. Cumulative Flow	27
4. Iteración #3	28
4.1. Introducción	28
4.2. Análisis y Diseño	28
4.3. Implementación de los test	29
4.4. Desarrollo de los códigos	29
4.5. Pivotal Tracker	31
4.5.1. Velocity and points accepted	31
4.5.2. Stories accepted	32
4.5.3. Story cycle time	33
4.5.4. Rejection Rate	33
4.5.5. Burnup	34
4.5.6. Cumulative Flow	35

5. Iteración #4	36
5.1. Introducción	36
5.2. Configuración .travis.yml	36
5.3. Build en Travis CI	39
5.4. Pivotal Tracker	41
5.4.1. Velocity and points accepted	41
5.4.2. Stories accepted	42
5.4.3. Story cycle time	43
5.4.4. Rejection Rate	43
5.4.5. Burnup	44
5.4.6. Cumulative Flow	45
6. Iteración #5	46
6.1. Introducción	46
6.2. Configuración de Codecov	46
6.3. Seguimiento en Pivotal Tracker	48
6.3.1. Velocity and points accepted	49
6.3.2. Stories accepted	50
6.3.3. Story cycle time	50
6.3.4. Rejection rate	51
6.3.5. Burnup	51
6.3.6. Cumulative Flow	52
7. Pivotal Tracker	52
7.1. Velocity and points accepted	53
7.2. Stories accepted	54
7.3. Story cycle time	55
7.4. Rejection rate	55
7.5. Burnup	56
7.6. Cumulative Flow	56
8. Conclusiones	56

1. Iteración #0

1.1 Introducción

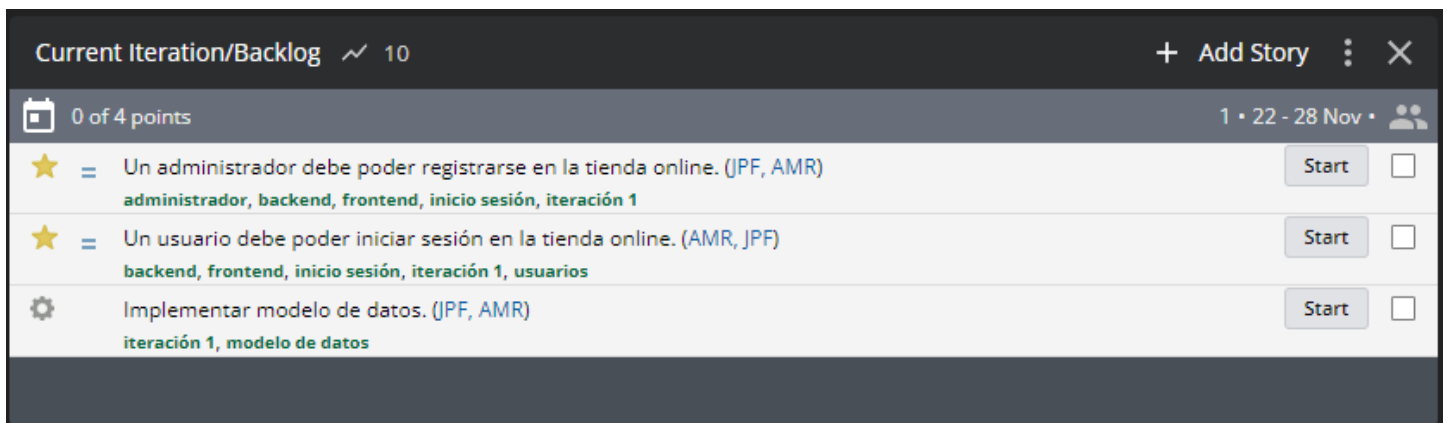
La **iteración #0** del proyecto consiste en empezar a planificar el desarrollo de la aplicación web **FUTURA**. Para ello es necesario identificar las primeras tareas a realizar y establecer cómo lograrlas. Esto se hará utilizando una metodología de desarrollo ágil mediante sprint semanales, junto con la herramienta **Pivotal Tracker** para hacer el seguimiento del proyecto.

Pivotal Tracker es una herramienta de gestión de proyectos ágiles que posibilita la colaboración en tiempo real en torno a un registro compartido priorizado. Los dos componentes del grupo se han registrado en esta herramienta con la finalidad de estar asociados al proyecto **2021-E10**.

En este informe se muestran las tareas añadidas en Pivotal Tracker de forma que sea posible llevar a cabo un seguimiento del proyecto.

1.2. Definiendo tareas en Pivotal Tracker

En primer lugar, hemos definido las historias de usuario que se deben completar en la primera iteración, para lo que se ha utilizado la pila **Backlog** que incluye las historias de usuario planificadas para sprints futuros. Las historias que se han añadido son las siguientes:



Como se puede observar los dos primeros elementos que aparecen en la lista incluyen una estrella ya que son **Feature**, esto indica que se son requisitos de usuario. Por su parte, *Implementar modelo de datos* tiene una rueda dentada debido a que es un **Chore**, que representan tareas que se deben hacer pero no aportan un valor real de negocio ni son errores.

Debido a que la iteración 1 comienza la misma semana en la que se realiza este informe se debe pulsar sobre **Start** en cada uno de los elementos de la lista que se

muestra en la imagen anterior. De esta forma se inicia el sprint correspondiente a la primera semana.

Current Iteration/Backlog ~ 10			+ Add Story	:	×
0 of 4 points			1 • 22 - 28 Nov •		
⚙	Implementar modelo de datos. (JPF, AMR) iteración 1, modelo de datos	Finish	<input type="checkbox"/>		
★ =	Un administrador debe poder iniciar sesión en la tienda online. (JPF, AMR) administrador, backend, frontend, inicio sesión, iteración 1	Finish	<input type="checkbox"/>		
★ =	Un usuario debe poder iniciar sesión en la tienda online. (AMR, JPF) backend, frontend, inicio sesión, iteración 1, usuarios	Finish	<input type="checkbox"/>		

Ahora las historias se encuentran en la pila **Current** ya que son en las que estamos trabajando actualmente. Cuando finalicemos de realizar las historias debemos pulsar sobre **Finish** para indicar que se han terminado.

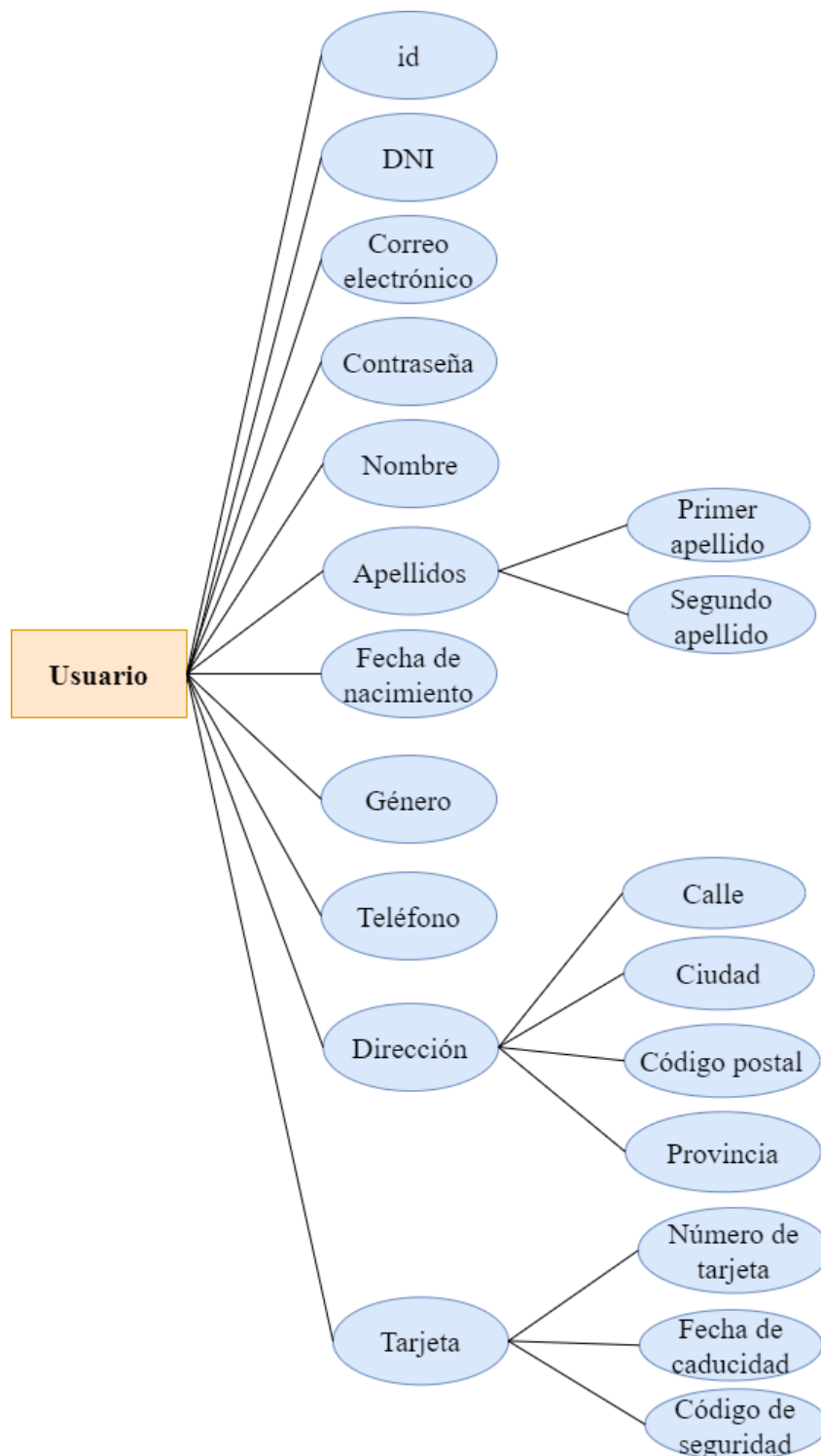
Además, también hemos definido historias de usuario en la pila **Icebox** para aquellas que debemos realizar pero todavía no están planificadas en el tiempo. A continuación, se muestra el contenido de esta pila:

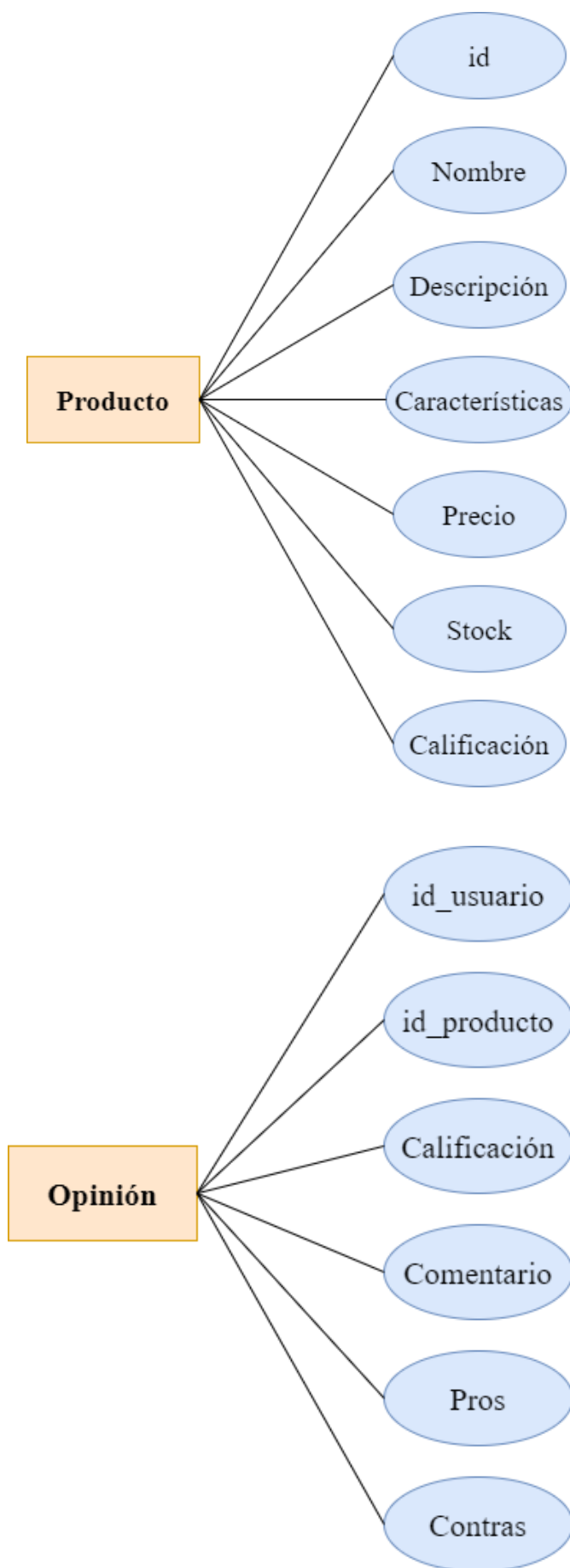
Icebox			+ Add Story	:	×
★ =	La tienda online debe mostrar a los usuarios el conjunto de productos y sus características. (AMR, JPF) backend, características productos, frontend, mostrar productos	Start	<input type="checkbox"/>		
★ =	Un usuario debe poder comprar los artículos que tiene en su carrito. (JPF) carrito, compra, gestión productos, productos	Start	<input type="checkbox"/>		
★ =	Un administrador tiene que poder modificar los productos de la tienda online. (AMR) administrador, backend, frontend, gestión productos	Start	<input type="checkbox"/>		
★ =	Un usuario debe poder ver los artículos de su carrito. (JPF) carrito, frontend, mostrar productos, productos, usuarios	Start	<input type="checkbox"/>		
★ =	Un usuario debe poder añadir un producto al carrito. (JPF, AMR) backend, carrito, compra, frontend	Start	<input type="checkbox"/>		
★ =	Un usuario debe poder registrarse. (AMR) backend, frontend, registro	Start	<input type="checkbox"/>		

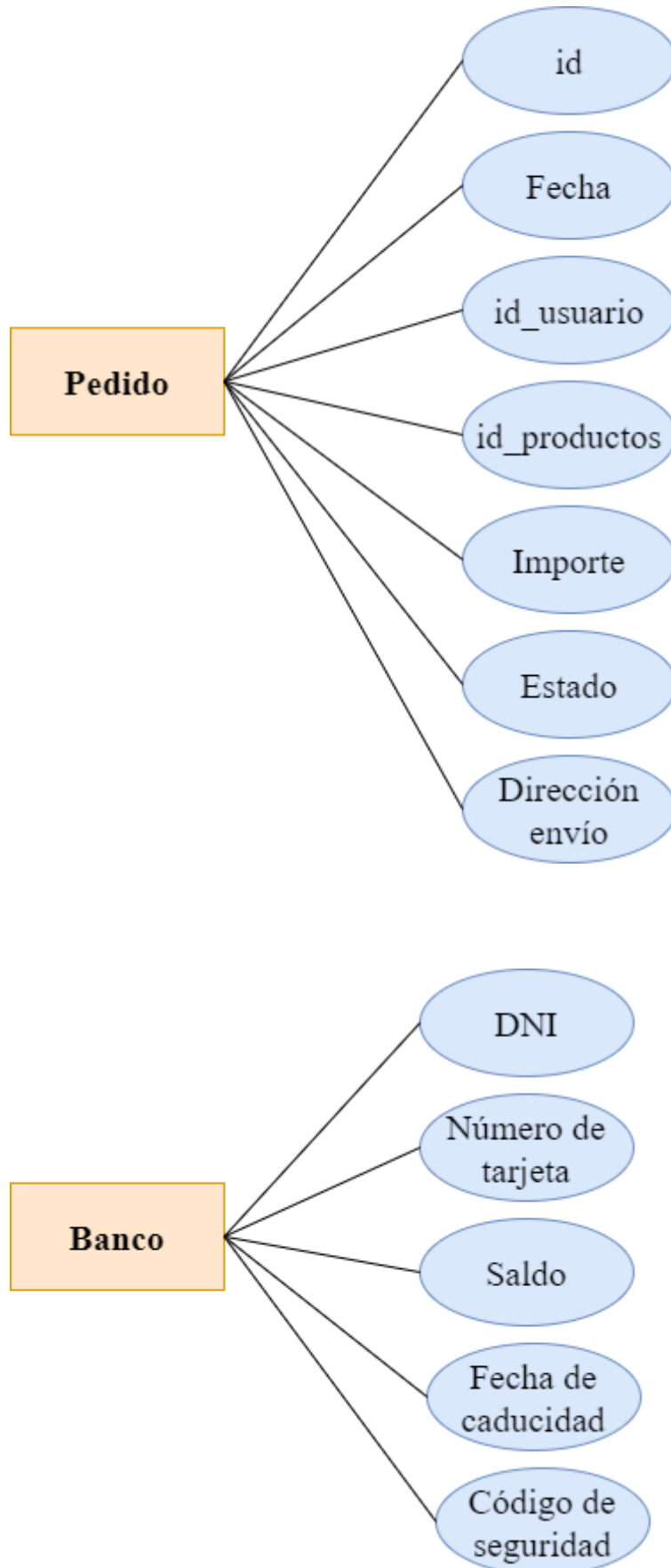
2. Iteración #1

2.1. Modelo de Datos

FUTURA necesitará varios modelos de datos, ya sea para los usuarios como para los productos que tendremos a la venta. Los modelos de datos iniciales que se han diseñado son:







2.2. Desarrollo de los códigos

El código que se ha desarrollado para implementar los modelos de datos anteriores se ha almacenado en el directorio **src/app/models** y se puede encontrar en el siguiente repositorio de GitHub:

<https://github.com/alu0101217741/SyTW-FUTURA-Iteracion-1.git>

El contenido de estos ficheros se muestra a continuación:

- **user.model.ts:**

```
type Surname = {  
  first_surname: string,  
  second_surname: string  
};  
  
type Gender = 'Masculino' | 'Femenino' | 'No especificado';  
  
type Address = {  
  street: string,  
  city: string,  
  postal_code: number,  
  province: string  
};  
  
type Card = {  
  card_number: number,  
  expiration_date: string,  
  security_code: number  
};  
  
export class User {  
  constructor(private id: number, private dni: number, private email:  
string, private password: string, private name: string,  
    private surname: Surname, private date_of_birth: string, private  
gender: Gender, private phone: number,  
    private address: Address, private credit_card: Card) {}  
}
```

- **product.model.ts:**

```
export class Product {
    constructor(private id: number, private name: string, private
description: string,
        private characteristics: string, private price: number, private
stock: number,
        private mark: number) {}
}
```

- **opinion.model.ts:**

```
export class Opinion {
    constructor(private id_usuario: number, private id_producto: number,
        private mark: number, private comment: string,
        private advantage: string, private disadvantage: string) {}
}
```

- **order.model.ts:**

```
type Status = 'Tramitando pedido' | 'En camino' | 'Entregado';

type Address = {
    street: string,
    city: string,
    postal_code: number,
    province: string
};

export class Order {
    constructor(private id: number, private date: string,
        private id_user: number, private id_product: number[],
        private price: number, private status: Status,
        private shipping_address: Address) {}
}
```

- **bank.model.ts:**

```
export class Bank {
    constructor(private dni: string, private card_number: number,
        private balance: number, private expiration_date: string,
        private security_code: number) {}
}
```

A continuación, mostramos los modelos finales que se han ido llevando a cabo durante el desarrollo de la aplicación:

- **user.ts:**

```
export class User {  
  
    constructor(  
        public id: string,  
        public email: string,  
        public password: string,  
        public firstName: string,  
        public lastName1: string,  
        public lastName2: string,  
        public admin: boolean,  
        public wallet: number,  
        public shoppingCart?: [{  
            id: string,  
            amount: string  
        }],  
        public token?: string  
    ) { }  
}
```

- **mobile.ts:**

```
export class Mobile {  
  
    constructor(  
        public _id: string,  
        public name: string,  
        public description: string,  
        public screenSize: number,  
        public camera: string,  
        public cpu: string,  
        public battery: number,  
        public ram: number,  
        public storage: number,  
        public category: string,  
        public brand: string,  
        public OS: string,  
    ) { }  
}
```

```
    public date: string,  
    public price: number,  
    public stock: number,  
    public mark: number,  
    public image: string  
  ) { }  
}
```

- **laptop.ts:**

```
export class Laptop {  
  
  constructor(  
    public id: string,  
    public name: string,  
    public description: string,  
    public screenSize: number,  
    public camera: string,  
    public cpu: string,  
    public ram: number,  
    public graphic_card: string,  
    public storage: number,  
    public category: string,  
    public brand: string,  
    public OS: string,  
    public date: string,  
    public price: number,  
    public stock: number,  
    public mark: number,  
    public image: string  
  ) { }  
}
```

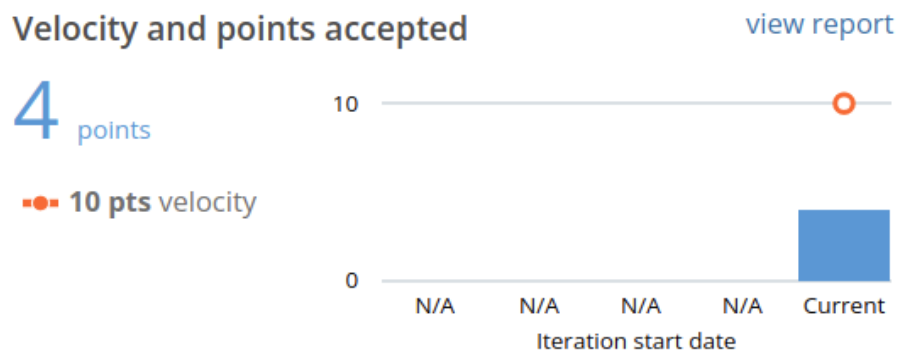
2.3. Pivotal Tracker.

Las analíticas están diseñadas para ayudarte a ver en qué se está trabajando, cómo están progresando las cosas y dónde puede necesitar ayuda tu equipo. No sólo muestran el progreso de alto nivel de tu trabajo, lo que te permite gestionar la fluidez del alcance e identificar los cuellos de botella, sino que también te permiten filtrar y profundizar en historias e iteraciones específicas.

A continuación, mostramos el desempeño del proyecto hasta la iteración #1.

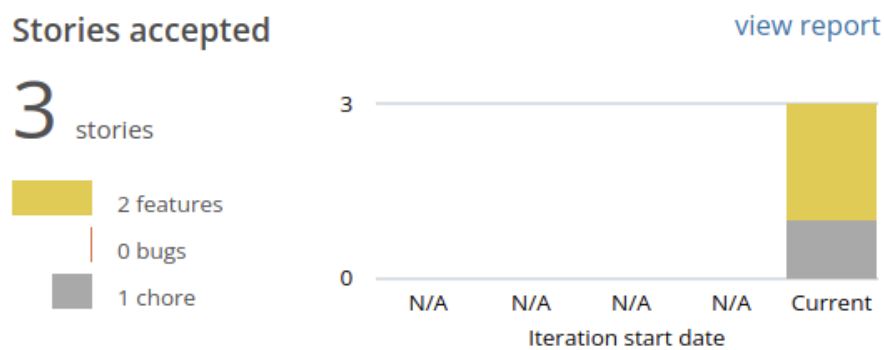
2.3.1. Velocity and points accepted

Traza el número de puntos aceptados en cada iteración, la velocidad de ejecución y la volatilidad del proyecto.



2.3.2. Stories accepted

Muestra la distribución de historias aceptadas y creadas de cada tipo a lo largo del tiempo.



2.3.3. Story cycle time

Demuestra el tiempo dedicado a las historias por estado, tipo de historia o puntos, y muestra todas las historias en una iteración determinada.

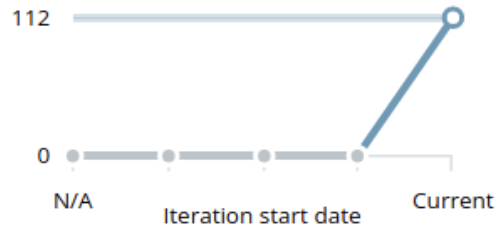
Story cycle time*

[view report](#)

Typical time between Start and Accept

112 hrs
per story

5-iteration avg: **112 hrs**



2.3.4. Rejection Rate

Muestra la tasa de rechazo de las historias y el tiempo que transcurre antes de que se reinicie (parte del informe de story cycle time.).

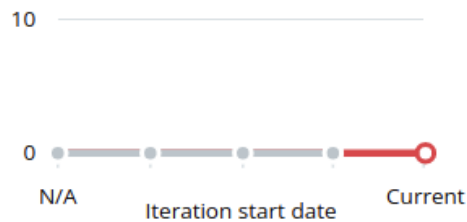
Rejection rate

[view report](#)

Percentage of rejections vs. acceptances and rejections

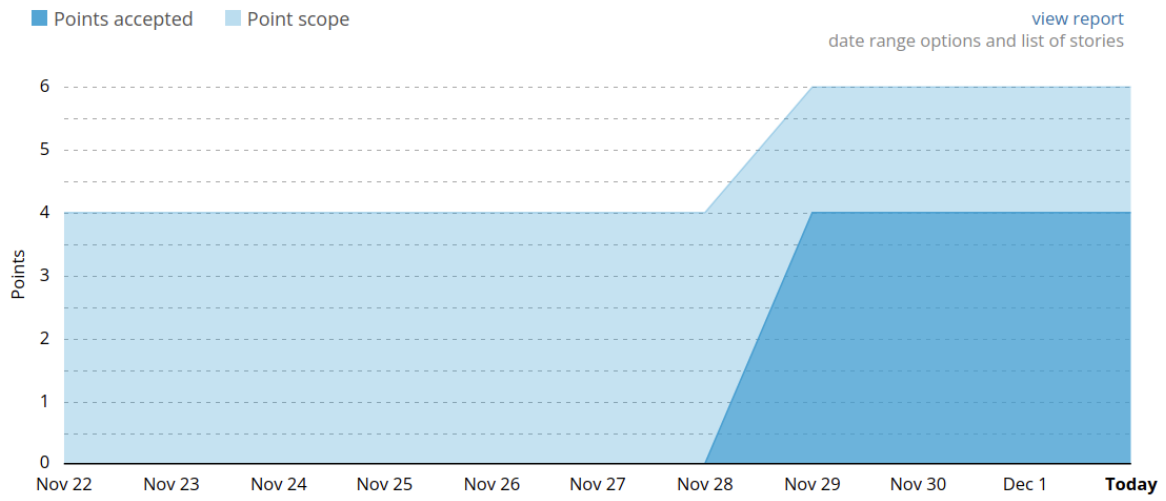
0%

5-iteration avg: **0%**



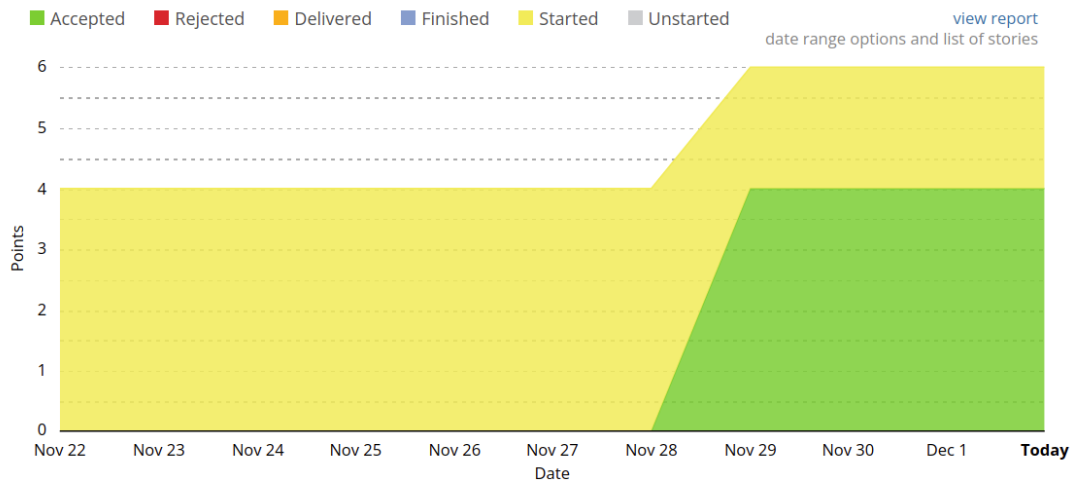
2.3.5. Burnup

Sigue el progreso hacia la finalización de un proyecto, mostrando cuántos puntos han sido aceptados del total de las iteraciones mostradas.



2.3.6. Cumulative Flow

Muestra el número de historias en cada estado (es decir, las que están hechas, las que están en proceso y las que aún no han comenzado) a lo largo del tiempo.



3. Iteración #2

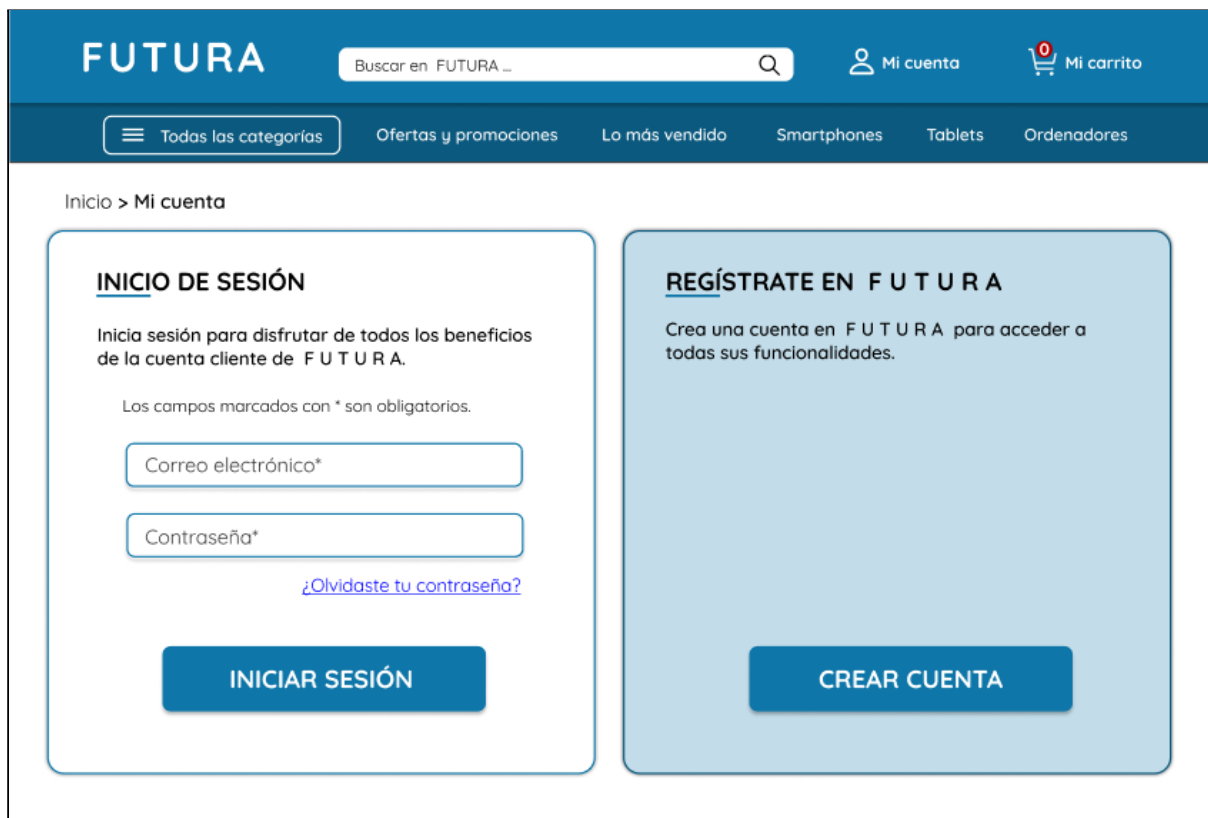
3.1. Introducción

En esta iteración se han implementado los módulos para el inicio de sesión y el registro empleando el framework **Angular**. Para ello, primero se ha llevado a cabo un análisis y diseño de estas pantallas con el objetivo de conocer mejor sus características y funcionalidades, de esta forma es posible determinar cuál es la mejor manera de desarrollarlas. Posteriormente con este conocimiento se han implementado los módulos para el registro y el inicio de sesión aplicando un diseño responsivo. Por último, se muestran las estadísticas de Pivotal tracker para la semana correspondiente a esta iteración.

3.2. Análisis y Diseño

En primer lugar, se han analizado los diseños realizados en la plataforma Figma para las pantallas que permiten iniciar sesión o registrar a un nuevo usuario. Los diseños y el análisis que se ha llevado a cabo se muestran a continuación:

- Pantalla **Mi Cuenta**:



FUTURA Buscar en FUTURA ... Mi cuenta Mi carrito

Todas las categorías Ofertas y promociones Lo más vendido Smartphones Tablets Ordenadores

Inicio > Mi cuenta

INICIO DE SESIÓN

Inicia sesión para disfrutar de todos los beneficios de la cuenta cliente de FUTURA.

Los campos marcados con * son obligatorios.

Correo electrónico*

Contraseña*

[¿Olvidaste tu contraseña?](#)

INICIAR SESIÓN

REGÍSTRATE EN FUTURA

Crea una cuenta en FUTURA para acceder a todas sus funcionalidades.

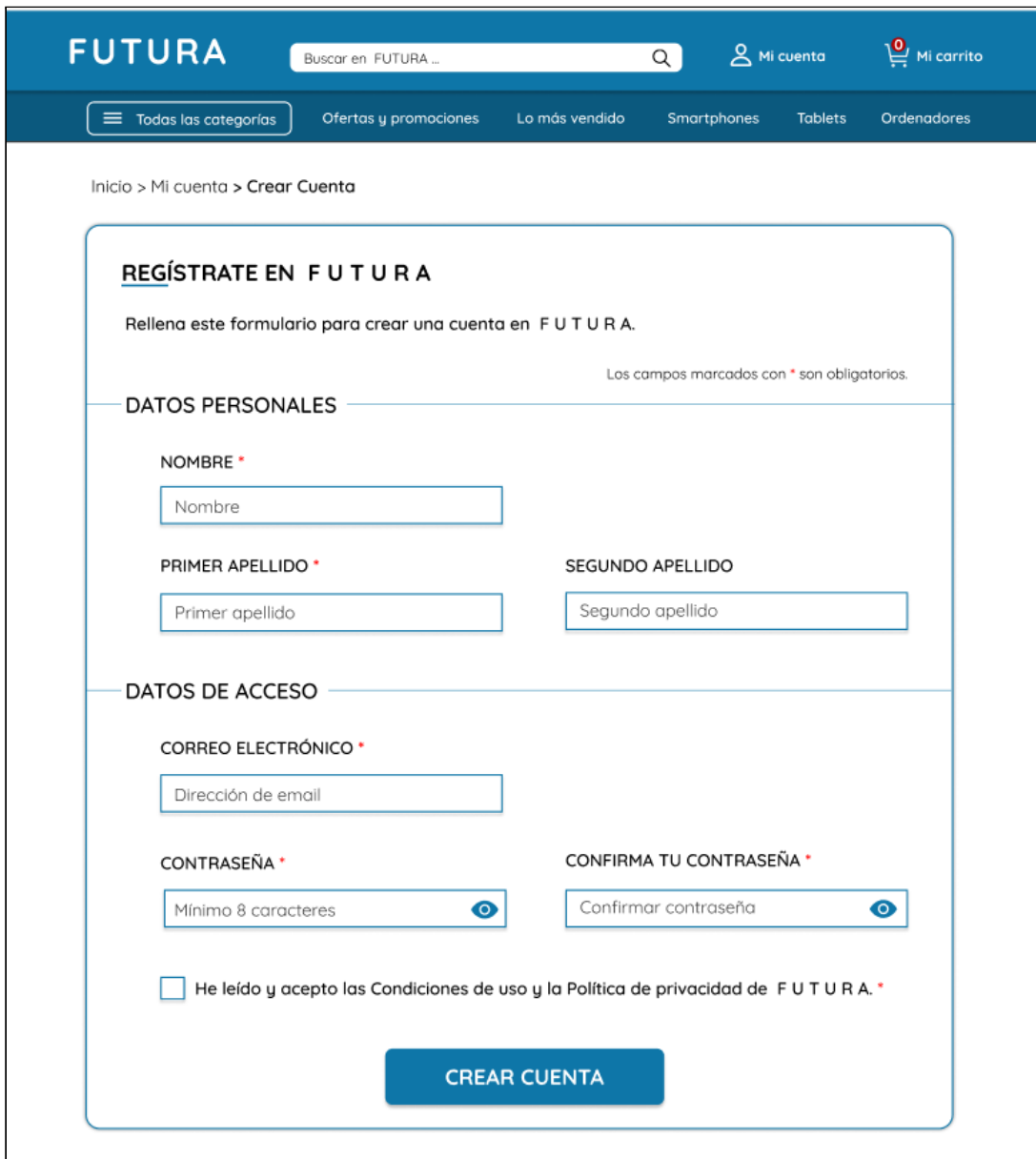
CREAR CUENTA



En esta pantalla es posible iniciar sesión introduciendo el correo electrónico y la contraseña. En caso de no disponer de una cuenta se debe pulsar sobre el botón

CREAR CUENTA para acceder al formulario de registro. También se puede ver la barra de navegación que permite acceder a las diferentes funcionalidades y categorías de la tienda online FUTURA.

Analizando las características y la distribución de esta pantalla se ha decidido que la mejor forma de implementarla empleando Angular consiste en la creación de un módulo que represente todo el conjunto. Dentro de este módulo habrá que crear dos componentes: uno para la tarjeta en la que se inicia sesión y la otra para la tarjeta que hace posible acceder al formulario de registro. Por otra parte, la barra de navegación también se tiene que desarrollar como un componente pero este será externo al módulo, ya que se utilizará en todas las pantallas de la aplicación.

- Pantalla **Crear cuenta**:



FUTURA Buscar en FUTURA ...  Mi cuenta  Mi carrito

Todas las categorías Ofertas y promociones Lo más vendido Smartphones Tablets Ordenadores

Inicio > Mi cuenta > Crear Cuenta

REGÍSTRATE EN FUTURA

Rellena este formulario para crear una cuenta en FUTURA.

Los campos marcados con * son obligatorios.

DATOS PERSONALES

NOMBRE *

PRIMER APELLIDO *

SEGUNDO APELLIDO

DATOS DE ACCESO

CORREO ELECTRÓNICO *

CONTRASEÑA *

CONFIRMA TU CONTRASEÑA *

☐ He leído y acepto las Condiciones de uso y la Política de privacidad de FUTURA. *

CREAR CUENTA

El acceso a esta página se consigue pulsando sobre el botón CREAR CUENTA de la pantalla **Mi Cuenta**. Aquí se muestra el formulario de registro que un nuevo usuario debe completar para crear una cuenta en la tienda online. En este caso tras analizar su estructura se ha determinado que se tiene que generar un módulo para que represente toda la pantalla. Además, únicamente es necesario crear un nuevo componente, donde se implemente el formulario de registro. Como se ha comentado anteriormente la barra de navegación es otro componente que se utilizará en todas las páginas de la tienda.

3.3. Implementación.

El código implementado para desarrollar las pantallas **Mi Cuenta** y **Crear Cuenta** está disponible en el repositorio de GitHub empleado para el proyecto de la asignatura:

<https://github.com/SyTW-2122/E10.git>

En la raíz de este repositorio se encontrará todo el código desarrollado para el frontend y en la carpeta **server** el correspondiente al backend.

Es importante tener en cuenta que debido a que se van a realizar cambios en este repositorio a medida que se desarrollen más aspectos de la aplicación, para acceder al código implementado en esta iteración hay que observar el último commit correspondiente a la iteración 2.

El diseño final de las pantallas empleando el framework Angular se muestra a continuación:

- Pantalla **Mi Cuenta**:



- Pantalla **Crear cuenta:**

FUTURA[Mi cuenta](#)[Mi carrito](#)

Ofertas y promocionesLo más vendidoSmartphonesTabletsOrdenadoresGamingTelevisión

Inicio > Mi cuenta > **Crear cuenta**

REGÍSTRATE EN FUTURA

Rellena este formulario para crear una cuenta en FUTURA.

Los campos marcados con * son obligatorios.

DATOS PERSONALES

NOMBRE *

PRIMER APELLIDO *

SEGUNDO APELLIDO

DATOS DE ACCESO

CORREO ELECTRÓNICO *

CONTRASEÑA *

CONFIRMA TU CONTRASEÑA *

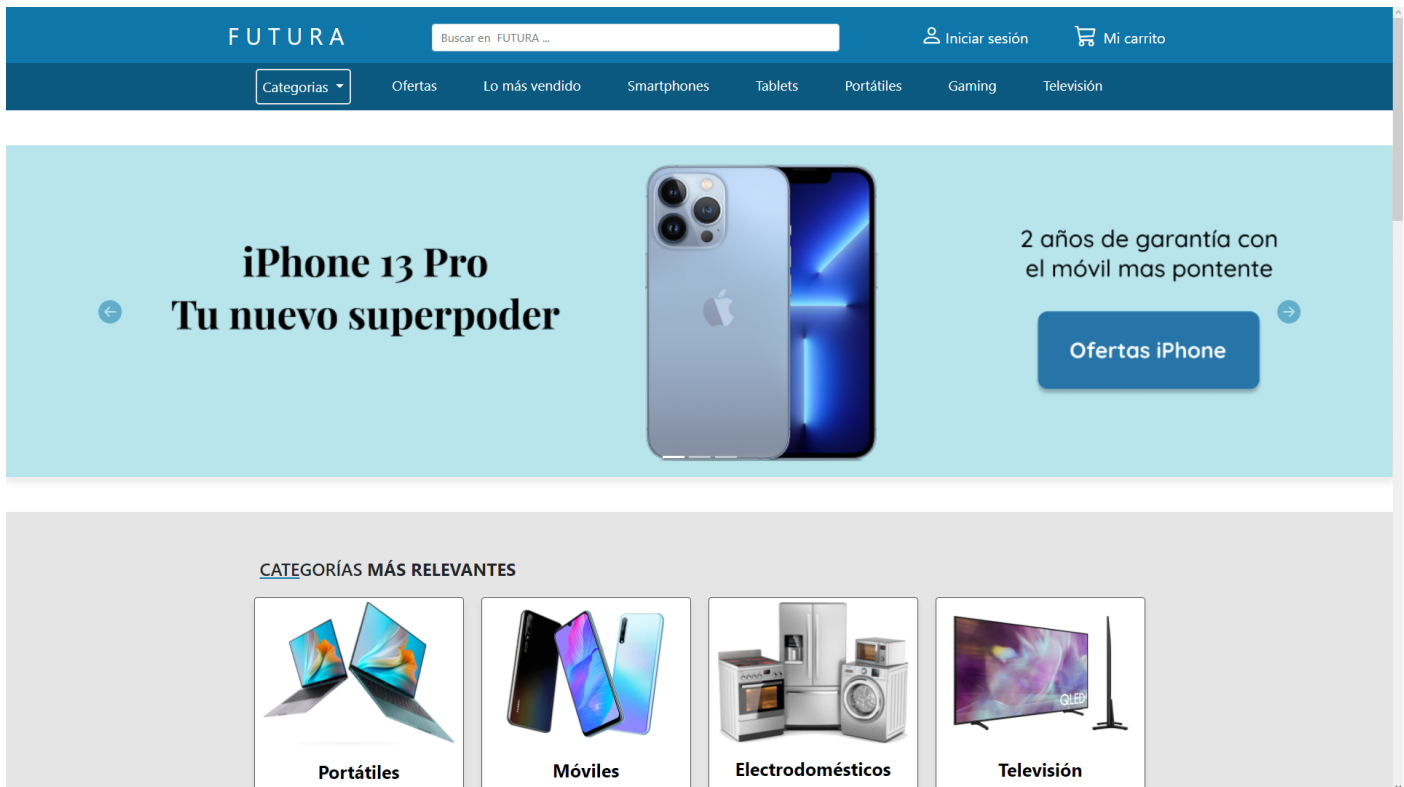
☐ He leído y acepto las Condiciones de uso y la Política de privacidad de FUTURA. *

CREAR CUENTA

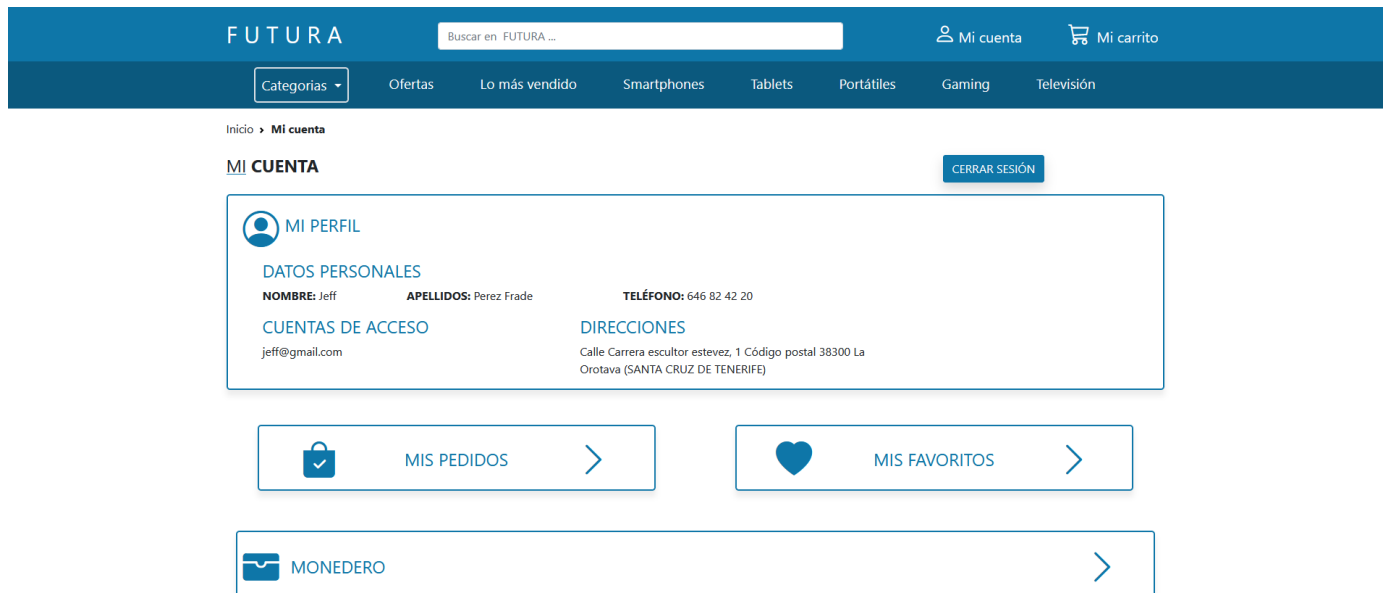
Cabe destacar que para la implementación de ambas pantallas se han aplicado funcionalidades de diseño responsivo.

A continuación, mostramos el diseño de las páginas del frontend que se han realizado mediante código:

- Pantalla **Principal**:



- Pantalla **Perfil**:



[Inicio](#) > [Mi cuenta](#) > **Cuenta Acceso**

[←VOLVER A FUTURA](#)

MI CUENTA

[PERFIL](#)

CUENTA DE ACCESO

[DATOS PERSONALES](#)

[DIRECCIONES](#)

[MIS PEDIDOS](#)

[MIS FAVORITOS](#)

[MONEDERO](#)

TU PERFIL

CUENTA DE ACCESO

Tu cuenta de acceso es **jeff@gmail.com** ¿Quieres cambiarla?

[CAMBIAR CUENTA >](#)

CONTRASEÑA ACTUAL

¿Quieres cambiar tu contraseña actual?

[CAMBIAR CONTRASEÑA >](#)

DARSE DE BAJA

Si tienes algún problema, antes de eliminar la cuenta puedes comunicarlo y trataremos de solucionarlo.

[ELIMINAR CUENTA >](#)

[Inicio](#) > [Mi cuenta](#) > **Datos Personales**

[←VOLVER A FUTURA](#)

MI CUENTA

[PERFIL](#)

[CUENTA DE ACCESO](#)

DATOS PERSONALES

[DIRECCIONES](#)

[MIS PEDIDOS](#)

[MIS FAVORITOS](#)

[MONEDERO](#)

TU PERFIL

DATOS PERSONALES

Gestiona tus datos.

Aquí puedes ver y editar los datos de tu perfil

NOMBRE

PRIMER APELLIDO **SEGUNDO APELLIDO**

FECHA DE NACIMIENTO

GÉNERO

TELÉFONO MÓVIL **TELÉFONO FIJO**

[GUARDAR CAMBIOS >](#)

Inicio > Mi cuenta > **Direcciones**

[←VOLVER A FUTURA](#)

MI CUENTA

[PERFIL](#)

[CUENTA DE ACCESO](#)

[DATOS PERSONALES](#)

DIRECCIONES

[MIS PEDIDOS](#)

[MIS FAVORITOS](#)

[MONEDERO](#)

TU PERFIL

TUS DIRECCIONES [AÑADIR NUEVA DIRECCION >](#)

DIRECCIÓN

Calle Carrera escultor estevez, 1 Código postal 38300 La Orotava (SANTA CRUZ DE TENERIFE)

[TELÉFONO](#) 646646464 [CONTACTO](#) Jeff Perez Frade

[MODIFICAR DIRECCIÓN >](#) [ELIMINAR DIRECCIÓN >](#)

Inicio > Mi cuenta > **Mis Pedidos**

[←VOLVER A FUTURA](#)

MI CUENTA

[PERFIL](#)

[CUENTA DE ACCESO](#)

[DATOS PERSONALES](#)

[DIRECCIONES](#)

MIS PEDIDOS

[MIS FAVORITOS](#)

[MONEDERO](#)

MIS PEDIDOS

Estos son los pedidos que has realizado en **FUTURA**.

Realizado:	jueves 4 de noviembre 2021
Número de pedido:	012457531178
Contiene:	5 artículos
Importe:	370,80 €
Estado:	Entregado
Ver Detalles >	

Realizado:	miércoles 10 noviembre 2021
Número de pedido:	01245753457
Contiene:	2 artículos
Importe:	185,32 €
Estado:	Entregado
Ver Detalles >	

Inicio > Mi cuenta > **Mis Favoritos**

[←VOLVER A FUTURA](#)

MI CUENTA

[PERFIL](#)

[CUENTA DE ACCESO](#)

[DATOS PERSONALES](#)

[DIRECCIONES](#)




[MIS PEDIDOS](#)

MIS FAVORITOS

[MONEDERO](#)

MIS FAVORITOS

Estos son los pedidos que has marcado como favoritos en **FUTURA**.

		
Apple iPhone 13 Pro Max	Monitor - Samsung 28" UHD 4K	ASUS TUF, Intel® Core™ i7
1259.30	250.20	1200.20

FUTURA

Buscar en FUTURA ...

Mi cuenta

Mi carrito

Categorías

Ofertas

Lo más vendido

Smartphones

Tablets

Portátiles

Gaming

Televisión

Inicio > Mi cuenta > Monedero

←VOLVER A FUTURA

MI CUENTA

PERFIL

CUENTA DE ACCESO

DATOS PERSONALES

DIRECCIONES

MIS PEDIDOS

MIS FAVORITOS

MONEDERO

MONEDERO

Hola Jeff, este es el saldo que tienes en FUTURA.

Saldo actual: 120 €

AÑADIR SALDO >

- Pantalla **Mi carrito:**

FUTURA

Buscar en FUTURA ...

Mi cuenta

Mi carrito

Categorías

Ofertas

Lo más vendido

Smartphones

Tablets

Portátiles

Gaming





Televisión

Inicio > Carrito

MI CARRITO

ARTICULOS EN EL CARRITO

Pulsa sobre "Tramitar Compra" para gestionar la compra de tus artículos.

ARTICULO	PRECIO	CANTIDAD	TOTAL	ELIMINAR
 iPhone 13	899.99€	1	899.99€	
 Huawei Nova 9	499€	1	499€	

Continuar Comprando

Subtotal: 1398.99€

Gastos de envío estimados: Gratis

TOTAL: 1398.99€

Tramitar Compra >

- Pantalla **Productos móviles:**

FUTURA

Buscar en FUTURA ...

Mi cuenta

Mi carrito

Categorías

Ofertas

Lo más vendido

Smartphones

Tablets

Portátiles

Gaming

Televisión

Inicio > Móviles

TODOS LOS SMARTPHONES

Ordenar por:

Filtros

PRECIO

Mínimo (€)

Máximo (€)

0

2000

MARCAS

☐ Todos

☒ Apple - 125

☒ Samsung - 732

☐ Huawei - 465

☒ Xiaomi - 361

☐ Sony - 212

ALMACENAMIENTO (GB)

☒ Todos

☐ 1 GB - 5

☒ 4 GB - 22



iPhone 13
Móvil - Apple iPhone 13, Blanco estrella, 256 GB, 5G, 6.1" OLED Super Retina XDR, Chip A15 Bionic, iOS
899.99 €
★★★★★



Samsung Galaxy A52
Móvil - Samsung Galaxy A52, Negro, 128 GB, 6 GB RAM, 6.5" Full HD+, SDM720G Octa-Core, 4500 mAh, Android 11
289.99 €
★★★



- Pantalla **Productos portátiles:**

FUTURA

Buscar en FUTURA ...

Mi cuenta

Mi carrito

Categorías

Ofertas

Lo más vendido

Smartphones

Tablets

Portátiles

Gaming

Televisión

Inicio > Ordenadores

TODOS LOS PORTÁTILES

Ordenar por:

Filtros

PRECIO

Mínimo (€)

Máximo (€)

0

2000

MARCAS

☒ Todos

☒ Apple - 125

☒ Samsung - 732

☒ Huawei - 465

☒ Xiaomi - 361

☒ Sony - 212

ALMACENAMIENTO (GB)

☐ Todos

☐ 1 GB - 5

☐ 4 GB - 22

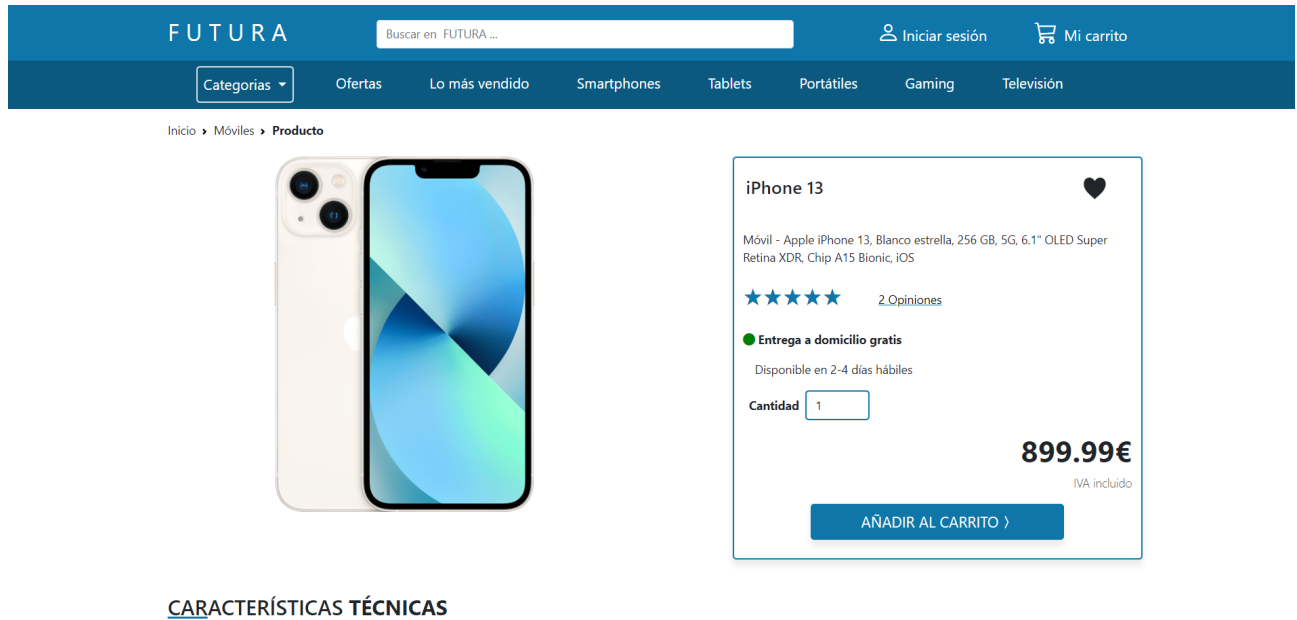


Lenovo Legion 5 15IMH05
Portátil gaming - Lenovo Legion 5 15IMH05, 15.6" FHD, Intel® Core™ i7-10750H, 16 GB, 1 TB, GTX 1650 Ti, W10H
1299 €
★★★★★



Portátil - HP 15S-EQ1040NS
15.6" HD, AMD 3020e, 4 GB, 128 GB SSD, Gráficos AMD Radeon™
300 €
★★★★


- Pantalla **Producto móvil específico:**



FUTURA Buscar en FUTURA ... [Iniciar sesión](#) [Mi carrito](#)

[Categorías](#) [Ofertas](#) [Lo más vendido](#) [Smartphones](#) [Tablets](#) [Portátiles](#) [Gaming](#) [Televisión](#)

[Inicio](#) > [Móviles](#) > **Producto**

iPhone 13 

Móvil - Apple iPhone 13, Blanco estrella, 256 GB, 5G, 6.1" OLED Super Retina XDR, Chip A15 Bionic, iOS

★★★★★ 2 Opiniones

Entrega a domicilio gratis

Disponible en 2-4 días hábiles

Cantidad

899.99€ IVA incluido

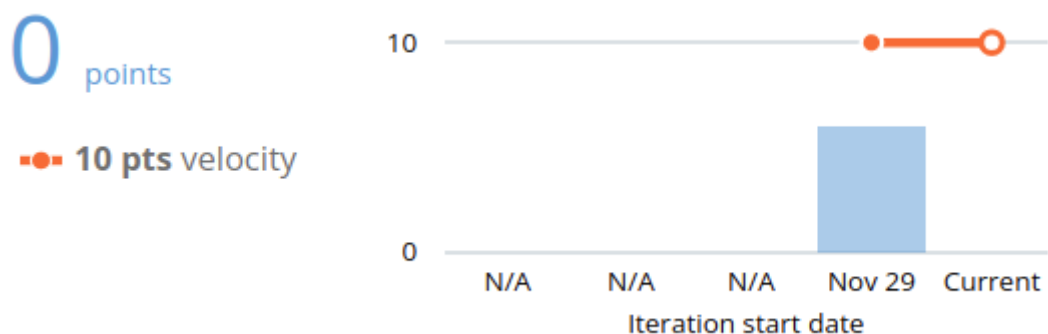
[AÑADIR AL CARRITO >](#)

CARACTERÍSTICAS TÉCNICAS

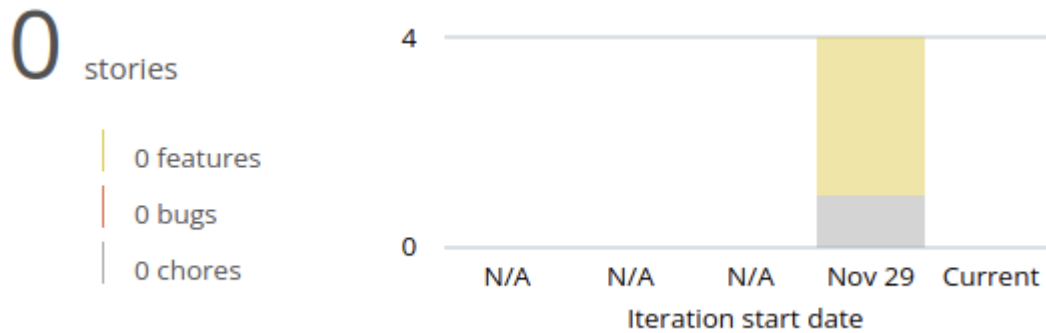
3.4. Pivotal Tracker

A continuación, se muestran las estadísticas generadas por Pivotal Tracker para la iteración #2 correspondiente a la semana 29 Nov - 5 Dec.

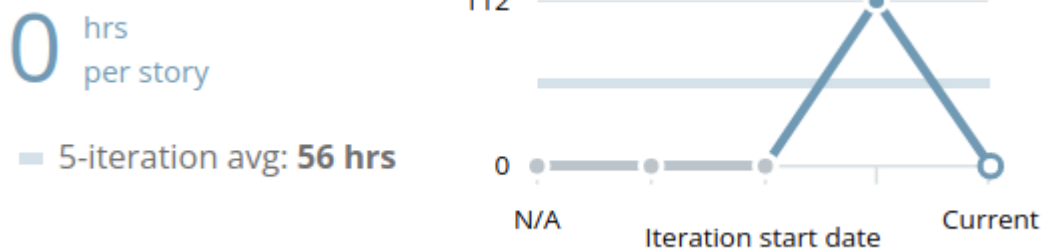
3.4.1. Velocity and points accepted



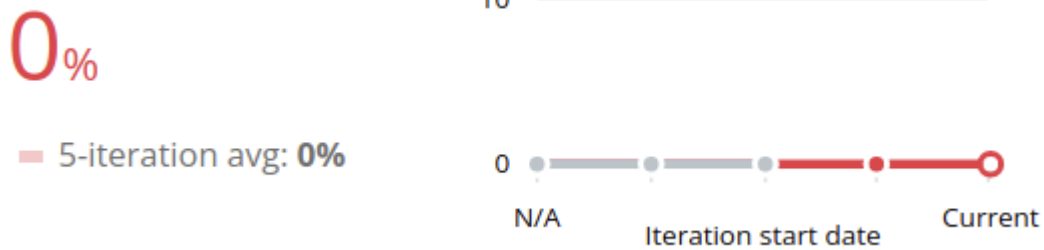
3.4.2. Stories accepted



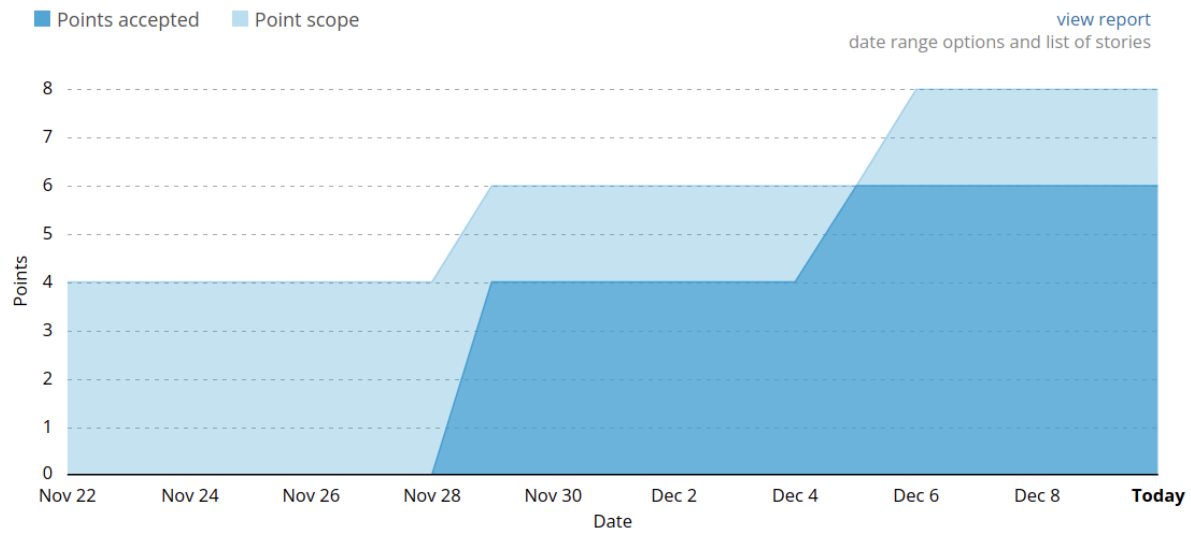
3.4.3. Story cycle time



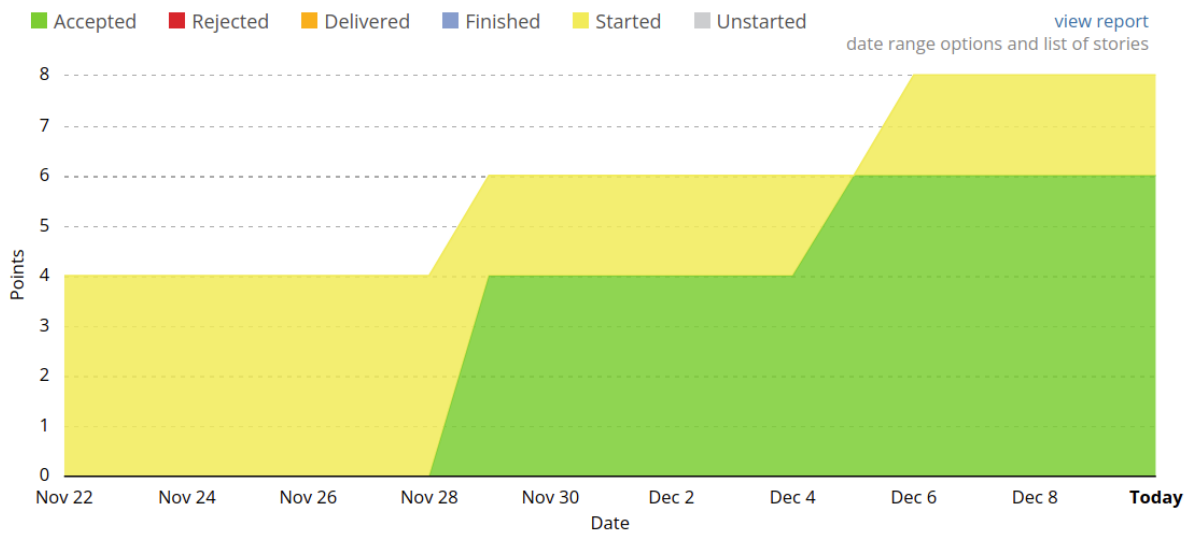
3.4.4. Rejection Rate



3.4.5. Burnup



3.4.6. Cumulative Flow



4. Iteración #3

4.1. Introducción

En anteriores iteraciones se ha creado el modelo de usuarios para la aplicación FUTURA, así como los métodos para el acceso y registro de usuarios. En esta iteración se llevará a cabo una nueva funcionalidad que consiste en el control de sesión, esto se realizará empleando el esquema de tokens **JWT**.

Para lo anterior, primero se ha llevado a cabo un análisis y diseño de las funcionalidades necesarias para esta fase del modelado de usuarios. Luego a partir de este planteamiento, se ha comenzado la fase de implementación empleando metodologías TDD. Para ello primero se han implementado los tests y comprobado que fallan, y posteriormente se han desarrollado los códigos que han permitido pasar esos tests. Por último, se muestran las estadísticas de Pivotal tracker para la semana correspondiente a esta iteración.

Todo el código desarrollado en esta iteración está disponible en el repositorio de GitHub empleado para el proyecto de la asignatura:

<https://github.com/SyTW-2122/E10.git>

Es importante tener en cuenta que debido a que se van a realizar cambios en este repositorio a medida que se desarrollen más aspectos de la aplicación, para acceder al código implementado en esta iteración hay que observar el último commit correspondiente a la iteración 3 realizado el 17 de diciembre de 2021.

4.2. Análisis y Diseño

A continuación, se muestran las funcionalidades que se requieren para esta tercera iteración del problema de modelado de usuarios:

- Redirigir con la sesión iniciada a la página de inicio de la tienda online, si el usuario inicia sesión con las credenciales correctas.
- Informar de la obligatoriedad de introducir tanto el correo electrónico como la contraseña, en caso de que el usuario pulse el botón de iniciar sesión sin indicar alguna de ellas.
- Informar al usuario cuando introduzca un correo electrónico o contraseña incorrecta.
- Proporcionar una página de perfil donde sea posible cerrar sesión, esta página será accesible para los usuarios que tengan la sesión iniciada y pulsen sobre “Mi cuenta” en la barra de navegación.

4.3. Implementación de los test

Una vez se conocen las funcionalidades que se deben cubrir en esta iteración, se implementan los tests que comprobarán su correcto funcionamiento. Para ello, se utiliza una metodología TDD por tanto el proceso de desarrollo consistirá en implementar pruebas que deben fallar, para posteriormente desarrollar el código necesario para conseguir que esas pruebas pasen correctamente.

Al utilizar **Angular** se trabaja con dos herramientas, por un lado se emplea el framework **Jasmine** para crear los test unitarios. A su vez también es necesario **Karma** que se trata de un ejecutor de tareas para nuestros tests.

En este caso las pruebas desarrolladas han sido para el control de sesión y se encuentran en el fichero **signin.component.spec.ts** dentro del directorio **E10/src/app/pages/my-account/components/signin**.

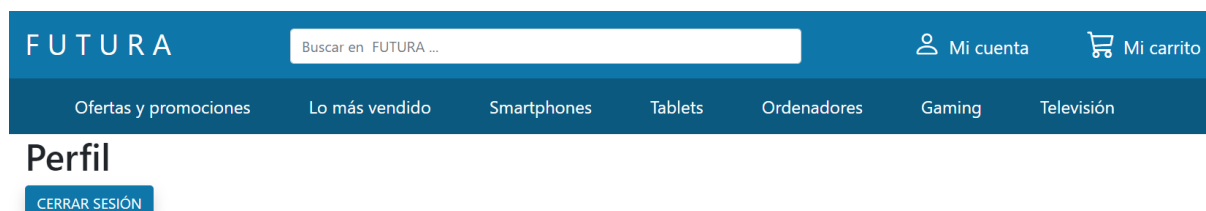
4.4. Desarrollo de los códigos

Tras realizar los tests y observar que fallan, se desarrollan los aspectos necesarios de la aplicación para lograr que todos pasen correctamente. Para implementar el código se ha seguido el tutorial [Angular 8 - Tutorial y ejemplo de autenticación JWT](#).

Cabe destacar que se utiliza un backend falso simulado de forma predeterminada para que pueda ejecutarse en el navegador sin una API real, por ello para iniciar sesión es necesario introducir las credenciales siguientes:

usuario: test **contraseña:** test

Una vez se hayan indicado este usuario y contraseña, se redirigirá a la página principal de la tienda online FUTURA. Posteriormente, para cerrar sesión se debe pulsar sobre **Mi Cuenta** en la barra de navegación lo que da acceso a la página de perfil, desde aquí se proporciona un botón que hace posible cerrar la sesión del usuario.




Además, en caso de que el usuario pulse el botón de iniciar sesión sin indicar el correo electrónico o la contraseña, se informa de la obligatoriedad de introducir estos campos.

INICIO DE SESIÓN



Inicia sesión para disfrutar de los beneficios de la cuenta cliente de FUTURA.

Correo electrónico



El correo electrónico es obligatorio.

Contraseña

La contraseña es obligatoria.

[¿Olvidaste tu contraseña?](#)

INICIAR SESIÓN


Por último, se informa al usuario en el caso de que introduzca un correo electrónico o contraseña incorrecta.

INICIO DE SESIÓN

Inicia sesión para disfrutar de los beneficios de la cuenta cliente de FUTURA.

Correo electrónico

Contraseña



[¿Olvidaste tu contraseña?](#)

INICIAR SESIÓN

El correo electrónico o la contraseña son incorrectos.

4.5. Pivotal Tracker

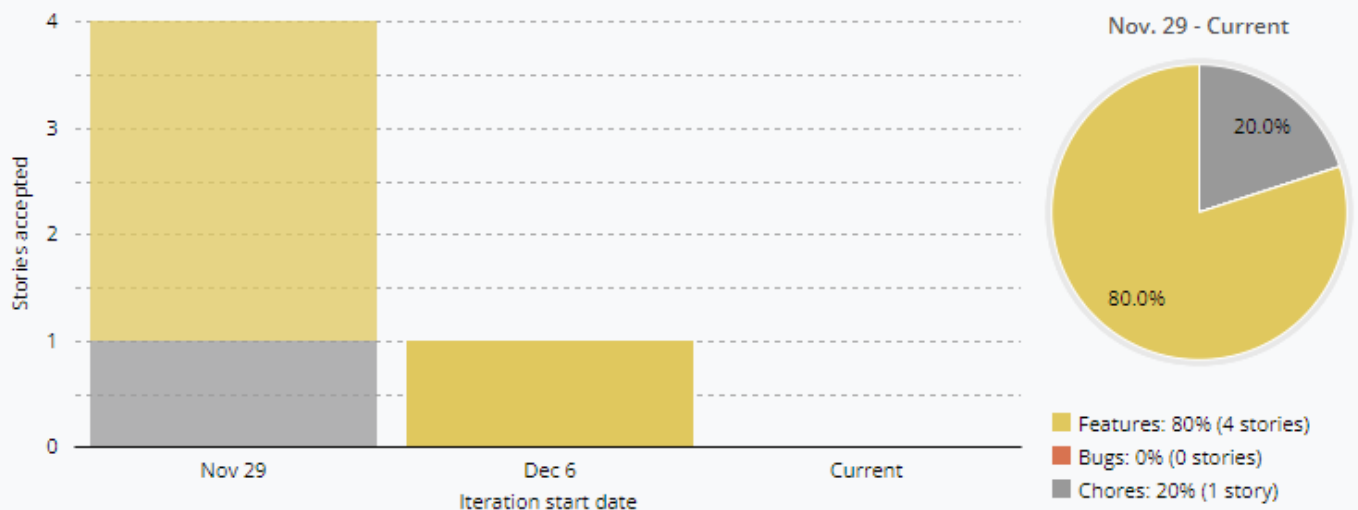
A continuación, se muestran las estadísticas generadas por Pivotal Tracker para la iteración #3 correspondiente a la semana del 6 al 12 de diciembre.

4.5.1. Velocity and points accepted

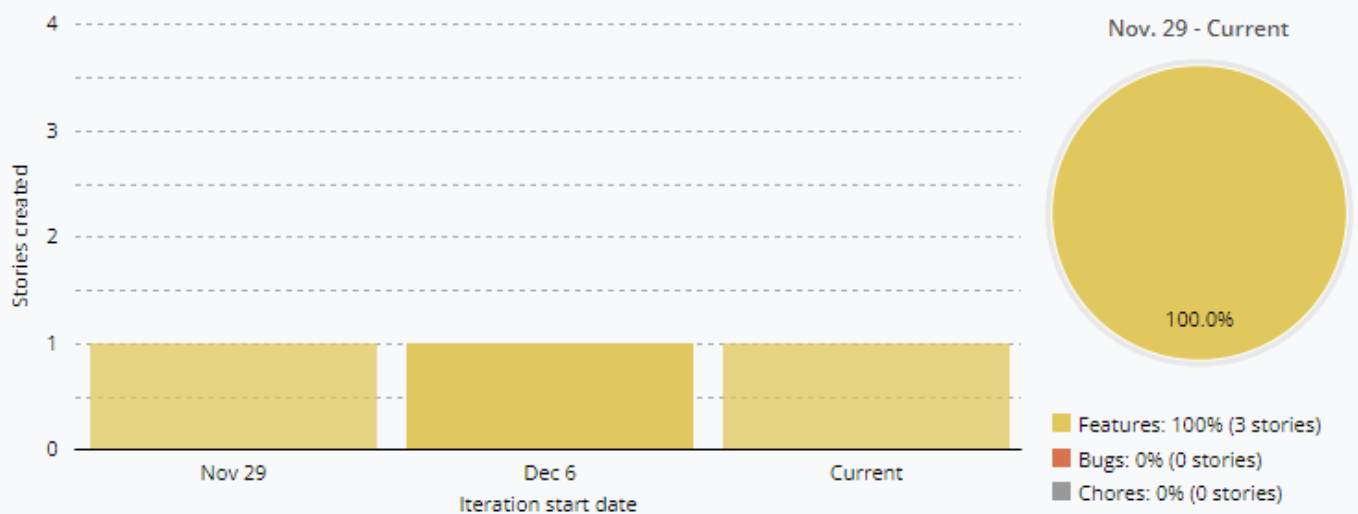


4.5.2. Stories accepted

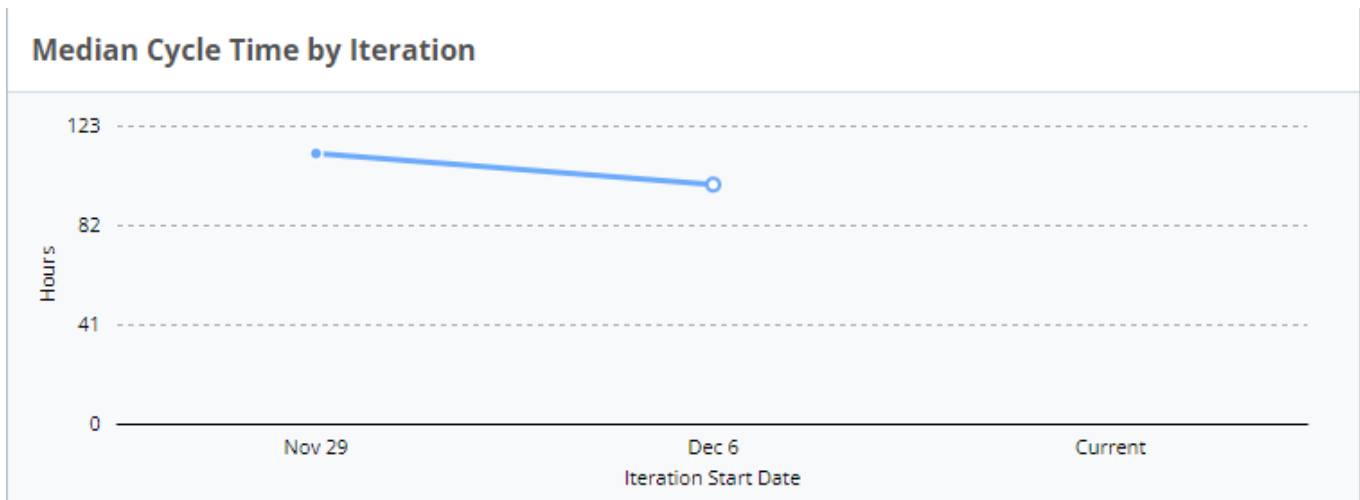
Accepted Story Types



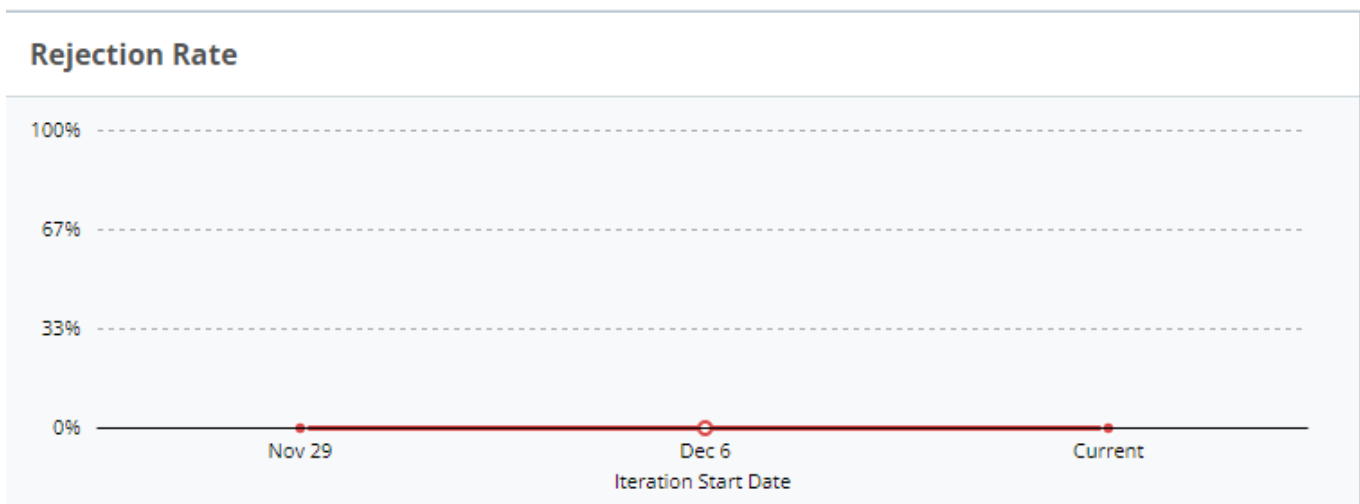
Created Story Types



4.5.3. Story cycle time

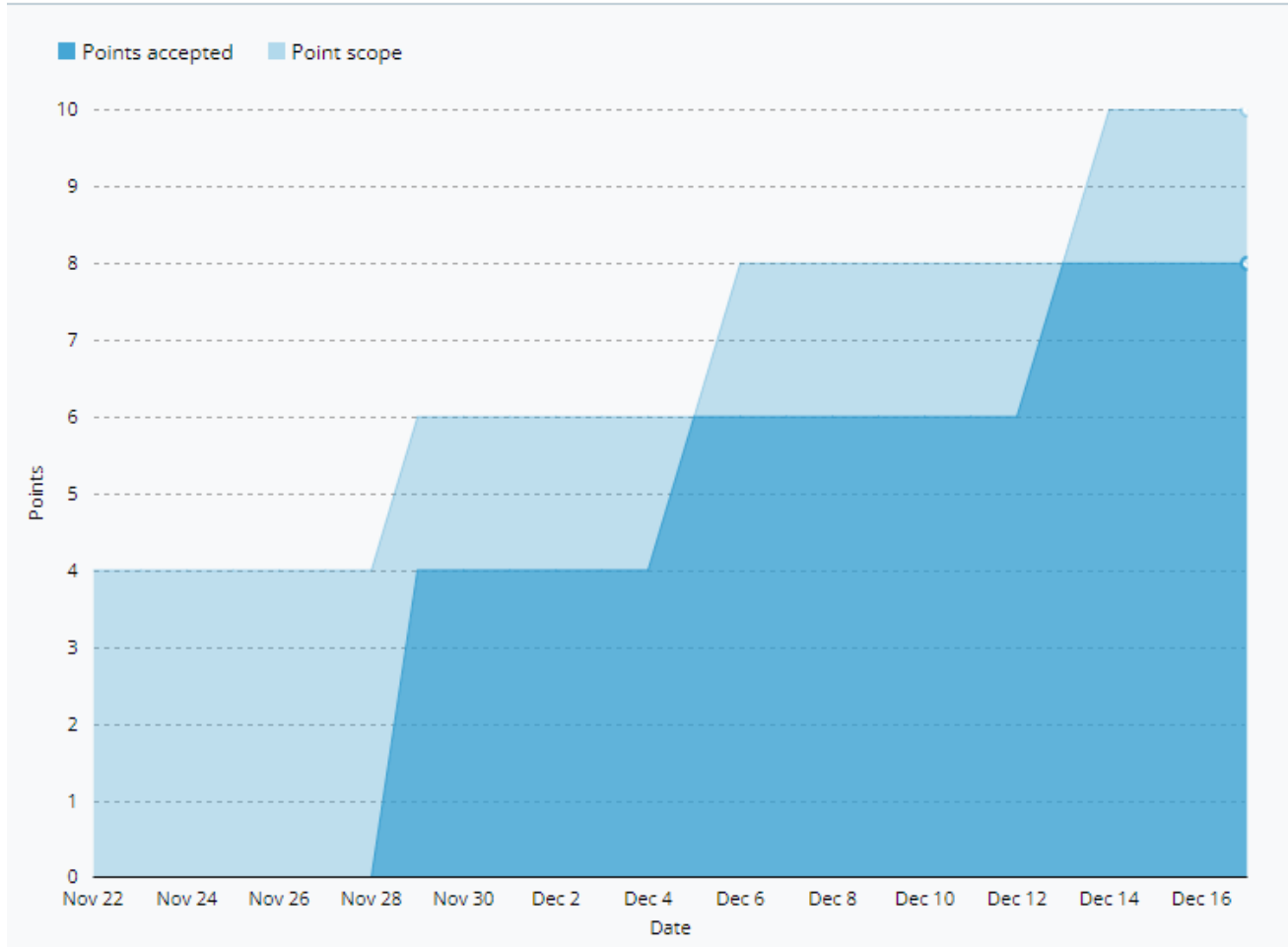


4.5.4. Rejection Rate

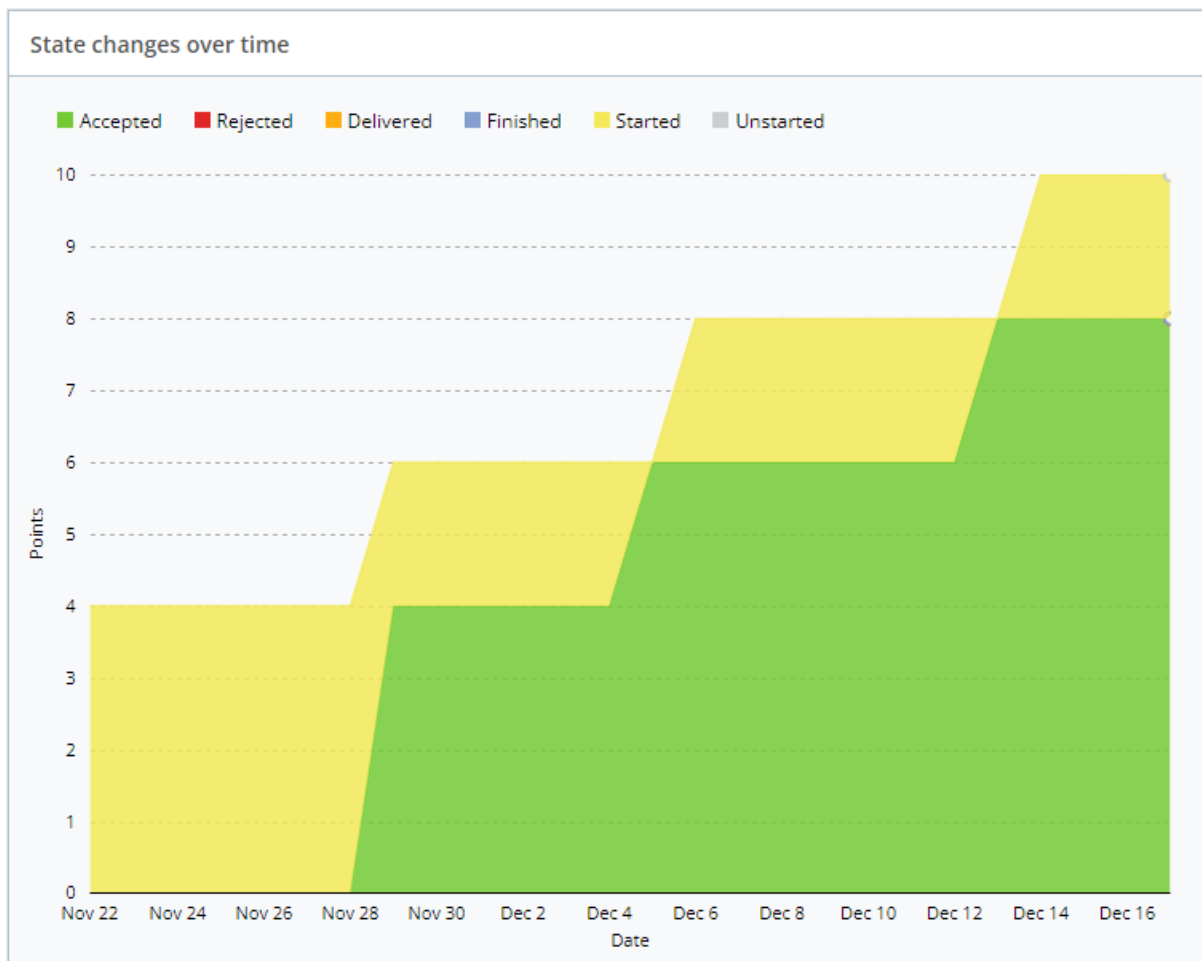


4.5.5. Burnup

Points accepted vs. points prioritized



4.5.6. Cumulative Flow



5. Iteración #4

5.1. Introducción

Tras realizar el control de sesión empleando el esquema de tokens **JWT**, en esta iteración se integra en el proyecto **FUTURA** un sistema de integración continua / despliegue continuo. Existen varias soluciones online disponibles, en este caso se ha elegido [Travis CI](#).

También será necesario trabajar con [Slack](#) que se trata de una herramienta de comunicación grupal que se integra con otras herramientas online, teniendo soporte para algunos entornos de Integración Continua como Travis-CI. Por lo que se va a configurar un sistema de notificaciones en Slack, de forma que se envíen los resultados de las pruebas e implementaciones del proyecto al espacio de trabajo correspondiente a este equipo.

Todo el código desarrollado en esta iteración está disponible en el repositorio de GitHub empleado para el proyecto de la asignatura:

<https://github.com/SyTW-2122/E10.git>

Es importante tener en cuenta que debido a que se van a realizar cambios en este repositorio a medida que se desarrollen más aspectos de la aplicación, para acceder al código implementado en esta iteración hay que observar el último commit correspondiente a la iteración 4 denominado *"Iteration 4 successful tests with Travis CI"* que fue realizado el 30 de diciembre de 2021.

5.2. Configuración .travis.yml

Las compilaciones en Travis CI se configuran principalmente mediante el archivo **.travis.yml** que debe estar alojado en la raíz del repositorio. Esto permite que su configuración esté controlada por versiones y sea flexible.

El contenido del fichero .travis.yml para este proyecto se muestra a continuación:

```
dist: focal

language: node_js
node_js:
  - "16"

addons:
  chrome: stable
```

```
cache:
  directories:
    - ./node_modules

install:
  - npm install

script:
  - npm run lint
  - npm run test-headless
  - npm run build
```

En este fichero se han especificado las siguientes configuraciones:

- **dist:** se corresponde con la distribución con la que se va a lanzar nuestra build.
- **language:** se corresponde con el lenguaje de programación utilizado. En este caso JavaScript.
- **node_js:** especifica la versión de Node.js que se está empleando.
- **addons:** permite indicar el Chrome que se utilizará.
- **cache:** indica la ubicación donde Travis CI puede almacenar en caché contenido que no cambia con frecuencia, para acelerar su proceso de compilación.
- **install:** se indican la instalación de dependencias necesarias para el funcionamiento de los scripts que se indican posteriormente.
- **script:** ejecuta una serie de órdenes durante la compilación.

Además, también se han modificado los scripts del fichero **package.json** teniendo finalmente lo siguiente:

```
"scripts": {
  "ng": "ng",
  "start": "ng serve",
  "build": "ng build",
  "watch": "ng build --watch --configuration development",
  "test": "ng test",
  "lint": "ng lint",
  "test-headless": "ng test -- --no-watch --no-progress
--browsers=ChromeHeadlessCI"
},
```

El último fichero que se ha editado ha sido el **karma.conf.js** con el objetivo de ajustar la configuración para ejecutar las pruebas del navegador Chrome. Lo que se ha modificado en este fichero se muestra a continuación:

```
browsers: ['ChromeHeadlessCI'],
customLaunchers: {
  ChromeHeadlessCI: {
    base: 'ChromeHeadless',
    flags: ['--no-sandbox']
  }
},
```

De esta forma se ha agregado un lanzador personalizado llamado **ChromeHeadlessCI**. La base de este es **ChromeHeadless**, que trae todas las características de la plataforma web moderna proporcionadas por Chromium y el motor de renderizado Blink a la línea de comandos. De esta forma es una gran herramienta para pruebas automatizadas y entornos de servidor en los que no necesita un shell de interfaz de usuario visible.

Una vez realizado lo anterior, se deben configurar las notificaciones en **Slack** de forma que sea posible recibir los resultados de probar e implementar el proyecto en Travis CI. Para ello está disponible un espacio de trabajo configurado en Slack para la monitorización de los proyectos de la asignatura.

El espacio de trabajo se denomina **SyTW**, tras acceder a él es necesario especificar el canal al que se desea unir, en este caso el correspondiente al equipo **E10**. Cuando ya se ha accedido es posible disponer de los tokens del canal.

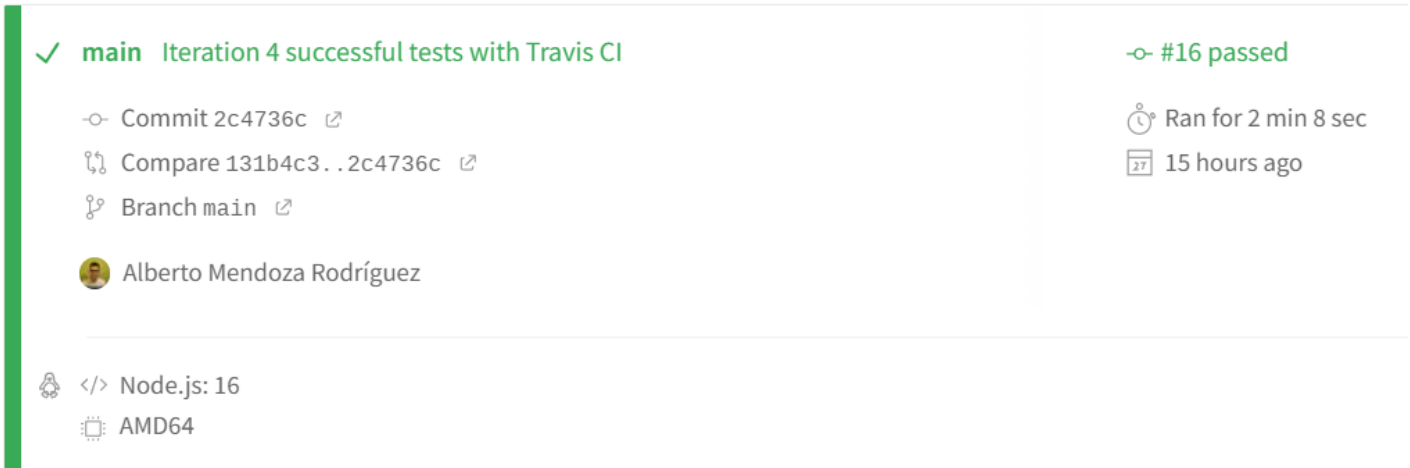
Con los tokens del canal se puede configurar las notificaciones en Slack a través de Travis CI. Esto se consigue añadiendo lo siguiente en el fichero `.travis.yml`:

```
notifications:
  slack:
    secure: XXXXXXXXXXXX
```

En **secure** se debe indicar el token cifrado correspondiente al canal, esto se logra empleando el comando **travis encrypt** con el token adecuado.

5.3. Build en Travis CI

Gracias a la configuración realizada en el apartado anterior, ahora cada vez que se haga un push al repositorio de GitHub se iniciará una build en Travis CI. En la siguiente imagen se muestra el resultado exitoso de la última build que se ha completado para esta iteración:





✓ **main** Iteration 4 successful tests with Travis CI → #16 passed


Commit 2c4736c [↗](#) ⌚ Ran for 2 min 8 sec

Compare 131b4c3...2c4736c [↗](#) 📅 15 hours ago

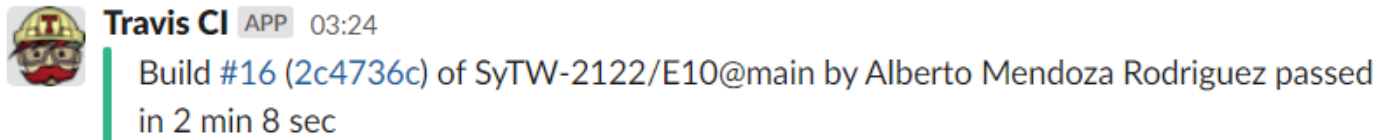
Branch main [↗](#)


 Alberto Mendoza Rodríguez

 </> Node.js: 16

 AMD64

Esta build se ha notificado en el canal **e10** de Slack con el siguiente mensaje:



 **Travis CI** APP 03:24

Build #16 (2c4736c) of SyTW-2122/E10@main by Alberto Mendoza Rodriguez passed in 2 min 8 sec

Debido a las políticas de cambio de **Travis-CI**, se ha tenido que cambiar dicha herramienta por **Github Actions**. Esta herramienta es una plataforma de integración continua y entrega continua (CI/CD) que permite automatizar su canalización de compilación, prueba e implementación. Puede crear flujos de trabajo que construyan y prueben cada solicitud de extracción en su repositorio, o implementar solicitudes de extracción combinadas en producción. Para ello, se ha llevado a cabo la configuración de Github Actions en el fichero **tests.yml**, este también contiene la configuración del Codecov que se explicará en la siguiente iteración.

```
name: Tests

on:
  push:
    branches: [ main, develop, storeProducts ]
  pull_request:
    branches: [ main ]
```



```
jobs:
  tests:

    runs-on: ubuntu-latest

    strategy:
      matrix:
        node-version: [16.x]
        # See supported Node.js release schedule at
https://nodejs.org/en/about/releases/

    steps:
      - uses: actions/checkout@v2
      - name: Use Node.js ${{ matrix.node-version }}
        uses: actions/setup-node@v2
        with:
          node-version: ${{ matrix.node-version }}

      - run: npm install
      - run: npm run test-headless
      - run: npm run build
      - run: npm run codecov

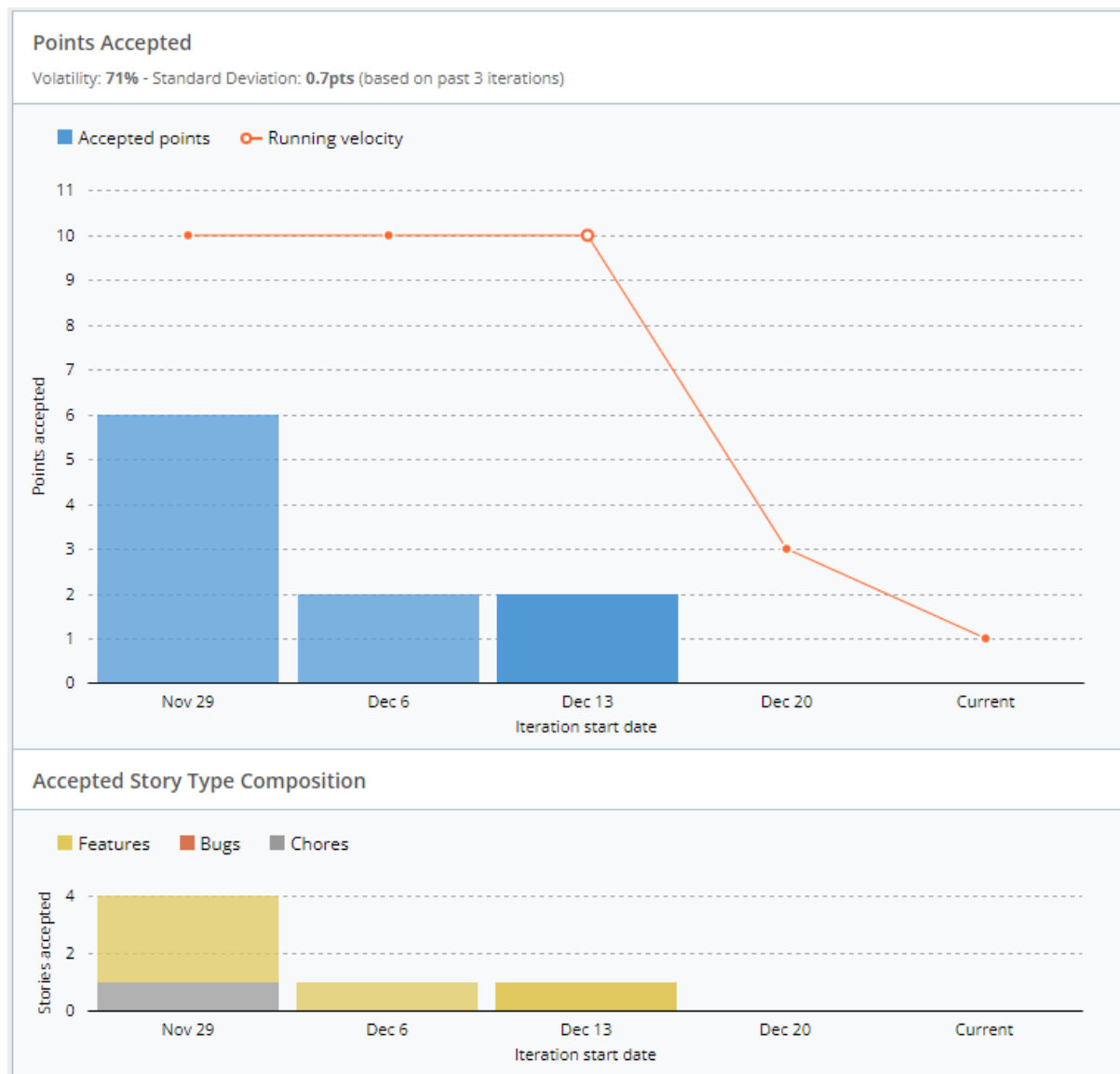
      - name: Slack Notification
        uses: rtCamp/action-slack-notify@v2
        env:
          SLACK_WEBHOOK: ${{ secrets.SLACK_WEBHOOK }}

      - name: Upload coverage to Codecov
        uses: codecov/codecov-action@v2
        with:
          token: ${{ secrets.CODECOV_TOKEN }}
```

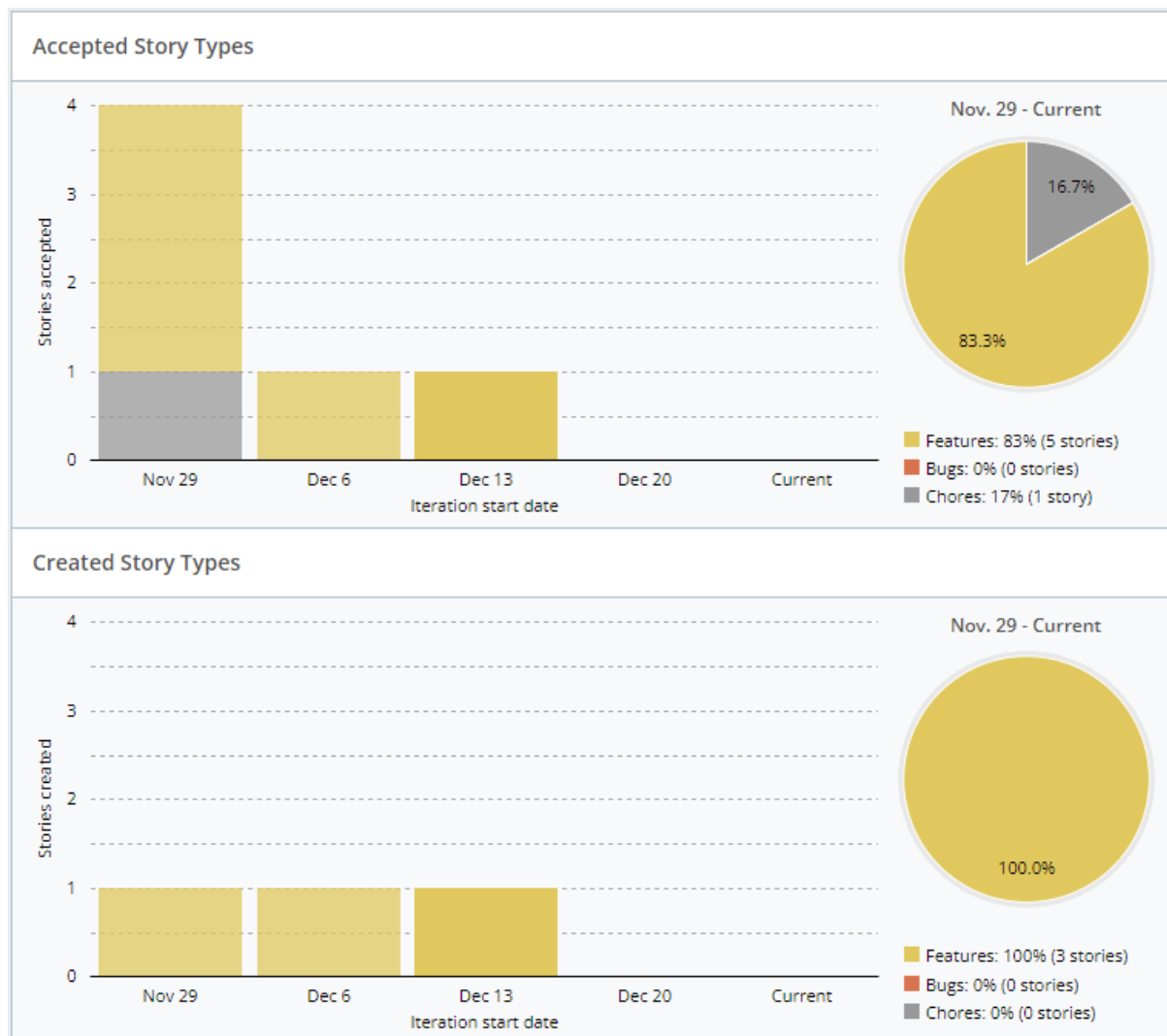
5.4. Pivotal Tracker

A continuación, se muestran las estadísticas generadas por Pivotal Tracker para la iteración #4 correspondiente a la semana del 13 al 19 de diciembre.

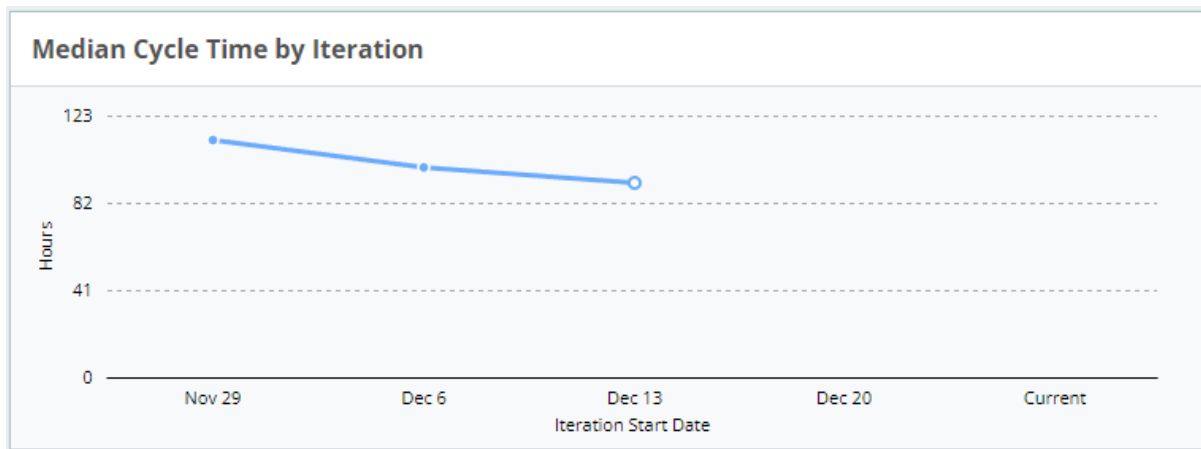
5.4.1. Velocity and points accepted



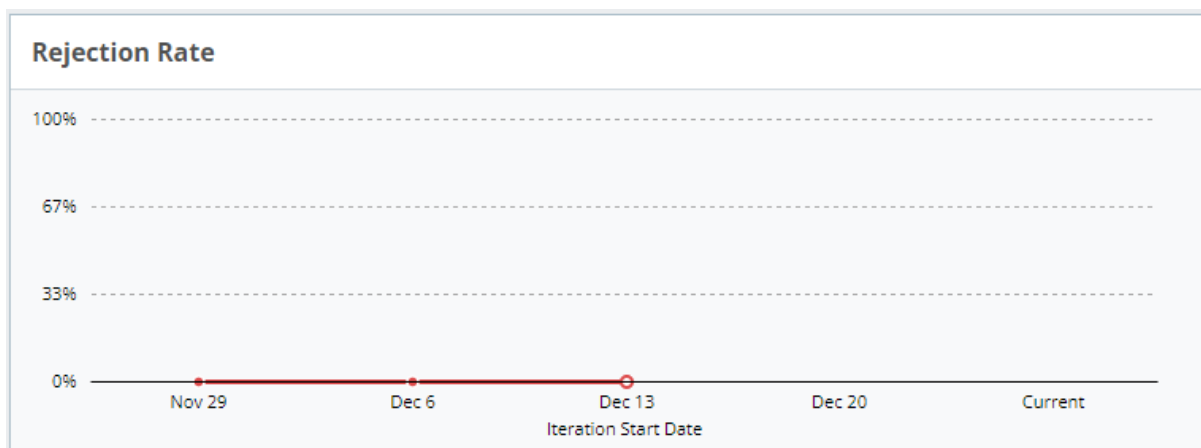
5.4.2. Stories accepted



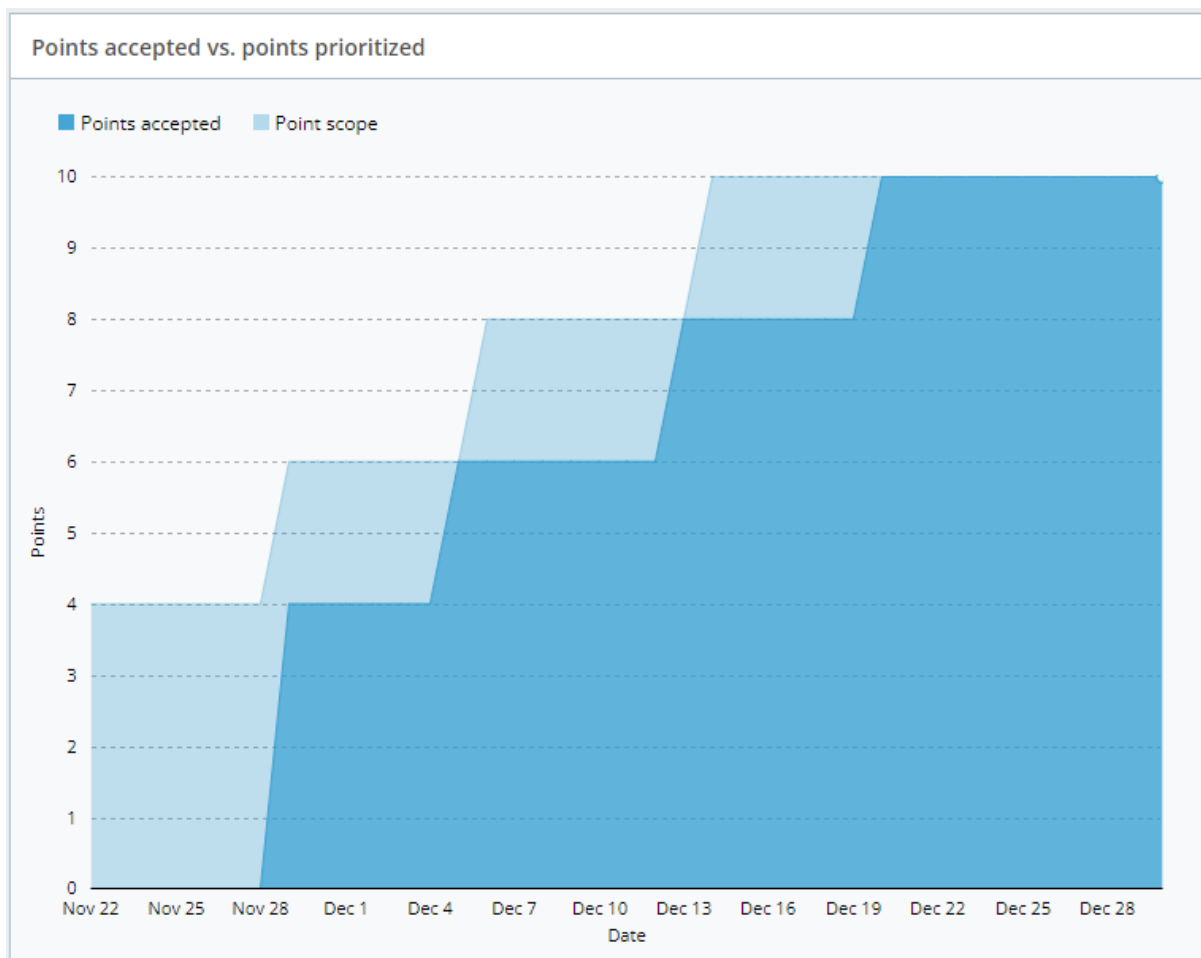
5.4.3. Story cycle time



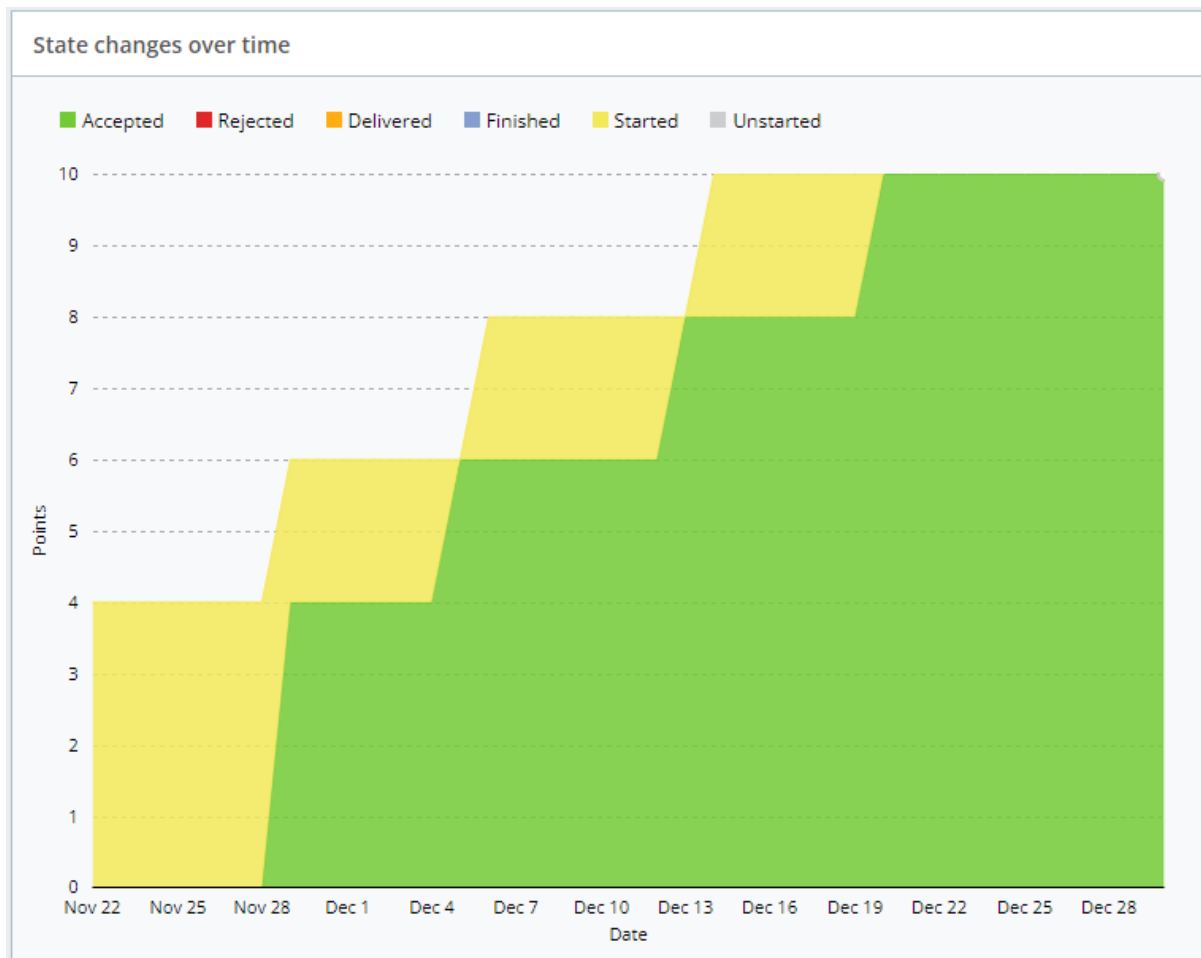
5.4.4. Rejection Rate



5.4.5. Burnup



5.4.6. Cumulative Flow



6. Iteración #5

6.1. Introducción

En esta iteración hemos llevado a cabo la configuración de un sistema de análisis de Calidad de Código. Para ello, hemos utilizado la opción (b), que permite instalar codecov.io como herramienta para el análisis de cobertura de código. Esta herramienta es una de las métricas más utilizadas para medir la calidad de un código.

Todo el código desarrollado en esta iteración está disponible en el repositorio de GitHub empleado para el proyecto de la asignatura:

<https://github.com/SyTW-2122/E10.git>

Es importante tener en cuenta que debido a que se van a realizar cambios en este repositorio a medida que se desarrollen más aspectos de la aplicación, para acceder al código implementado en esta iteración hay que observar el último commit correspondiente a la iteración 5 denominado *"Iteration 5 CCQ with Codecov"* que fue realizado el 27 de enero de 2022.

Cabe destacar que en las anteriores iteraciones, habíamos implementado Travis CI, pero debido a las nuevas políticas, no se puede seguir trabajando en dicha herramienta. Es por ello, que la solución que hemos llevado a cabo consiste en configurar la integración continua en GitHub Actions.

6.2. Configuración de Codecov

Inicialmente, hemos generado el **token en Codecov**, añadiendo como clave segura en GitHub Actions dicho token, y una vez hecho esto, hemos modificado el fichero `tests.yml` para añadir la configuración del token.

```
- name: Upload coverage to Codecov
  uses: codecov/codecov-action@v2
  with:
    token: ${ secrets.CODECOV_TOKEN }
```

Para generar los informes a partir de Karma, hemos modificado el fichero *karma.conf.js* de la siguiente manera:

```
preprocessor: {  
  'src/**/*.ts': ['coverage']  
},  
reporters: ['progress', 'coverage', 'kjhtml'],  
coverageReporter: {  
  dir: require('path').join(__dirname, './coverage/futura'),  
  subdir: '.',  
  reporters: [  
    { type: 'html' },  
    { type: 'text-summary' },  
    { type: 'lcov' }  
  ]  
},
```

Como se puede observar, se ha añadido el *preprocessor* que indica la ruta de ficheros en los que está escrito el código fuente. Luego, en *reporters* hemos incluido el propio 'coverage' y finalmente, el tipo de reporte realizado *{type: 'lcov'}* para generar el informe de cobertura y cargarlo en Codecov.

De esta manera, se muestran las capturas de pantalla del informe generado en Codecov:

Informe de cobertura		
CABEZA 5116167	Parche	Cambio
73,68 %	–	23,68 %
La cobertura promedio de cambios para esta confirmación es – (parche). Fuente de datos de la comparación entre e9ce8d4 y 5116167		

Como se puede observar el código está cubierto en un **73,68 %**. En parche no se muestra ninguna estadística ya que no se ha añadido ninguna línea respecto al commit anterior. Por último, el **23,68 %** se debe a que se incorporaron más pruebas que provoca que el código tenga ese porcentaje más de cobertura.

Una vez que se han desarrollado todas las pruebas de la aplicación, se obtiene un informe de cobertura como el siguiente:

Coverage report

HEAD a0edec0	Patch	Change
81.25 %	–	–

The average coverage of changes for this commit is – (patch). Data source from comparing between [Commits](#) and a0edec0

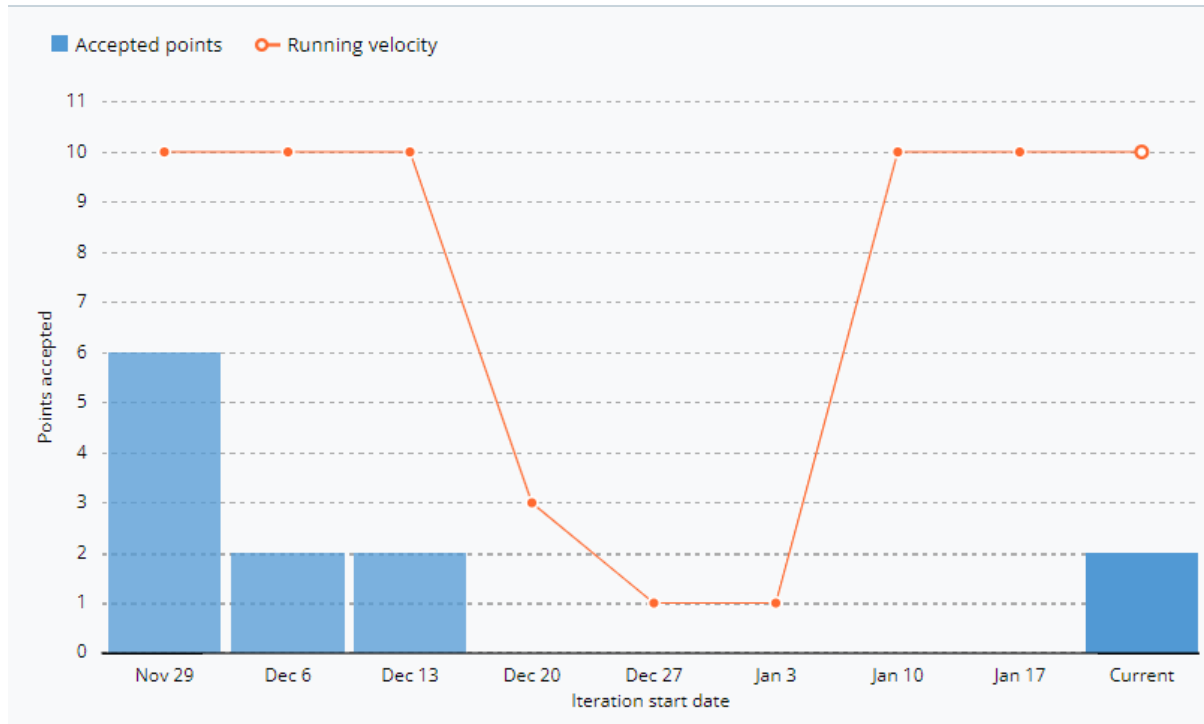
6.3. Seguimiento en Pivotal Tracker

A continuación, se muestran las estadísticas generadas por Pivotal Tracker para la iteración #5.

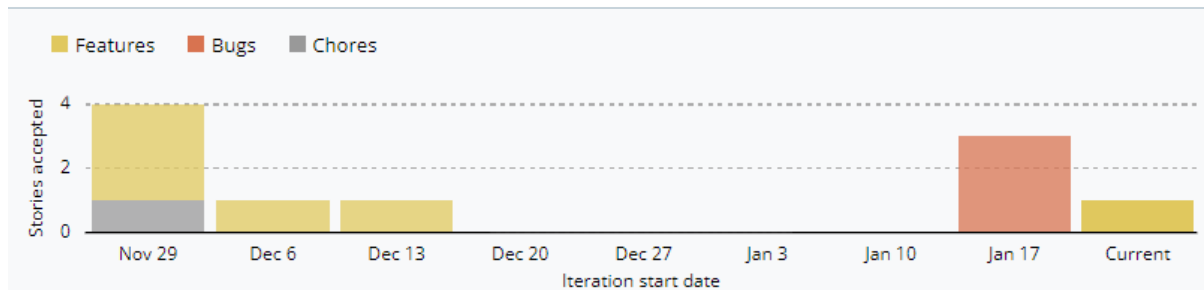
6.3.1. Velocity and points accepted

Points Accepted

Volatility: 0% - Standard Deviation: 0pts (based on past 3 iterations)

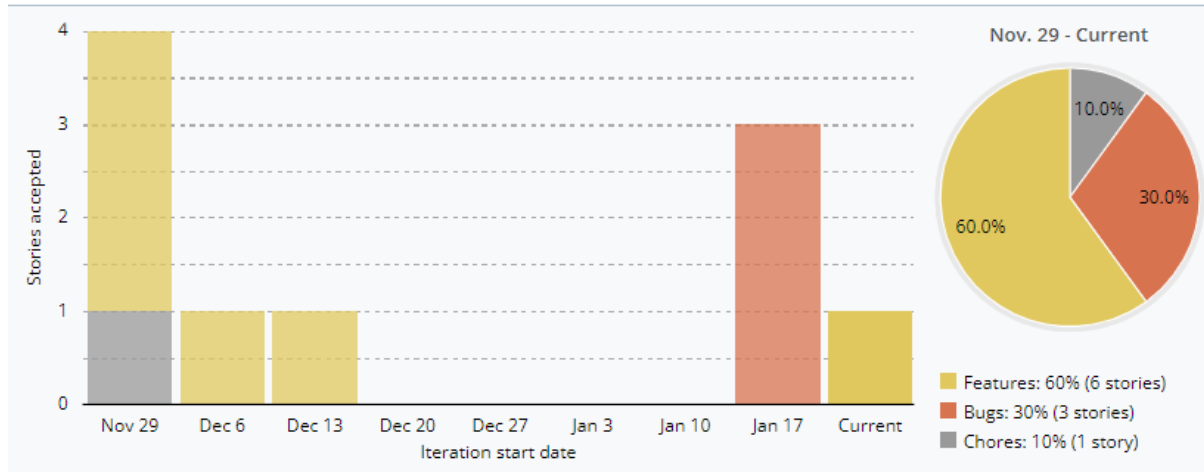


Accepted Story Type Composition

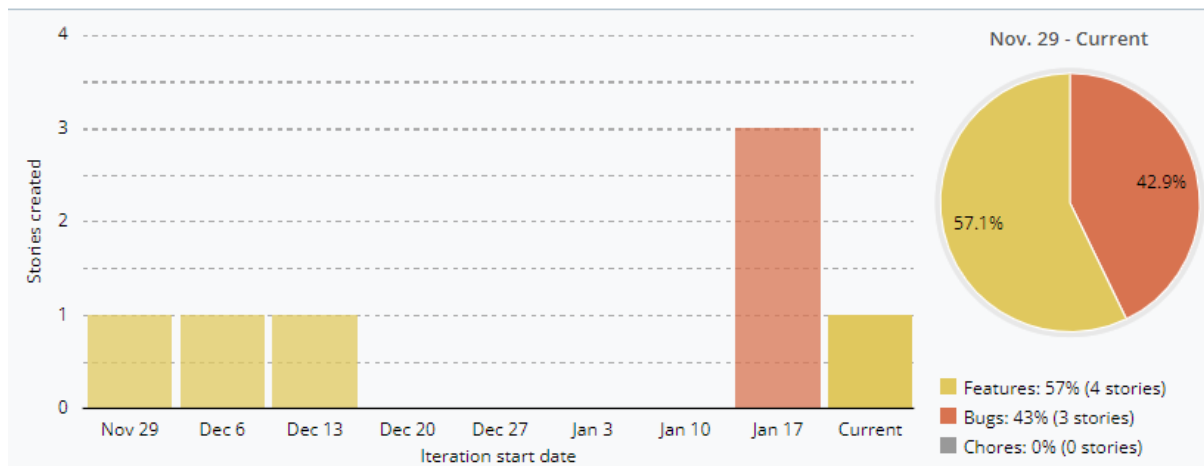


6.3.2. Stories accepted

Accepted Story Types

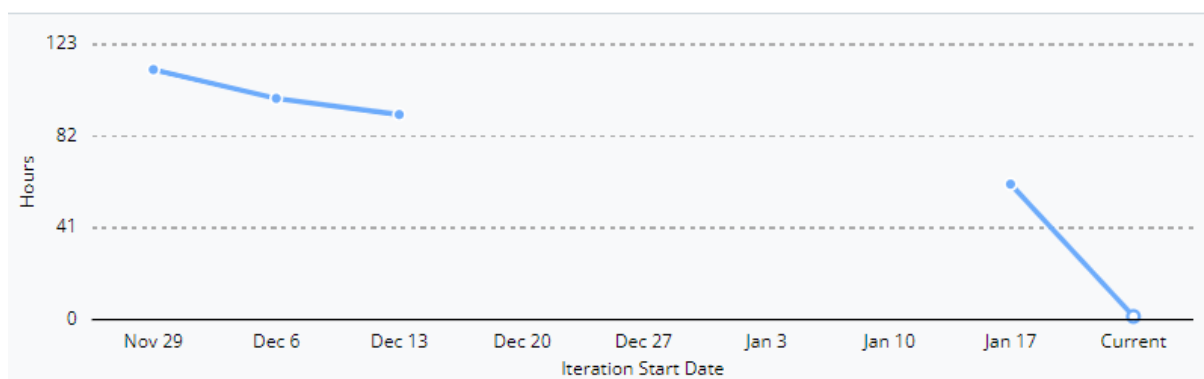


Created Story Types



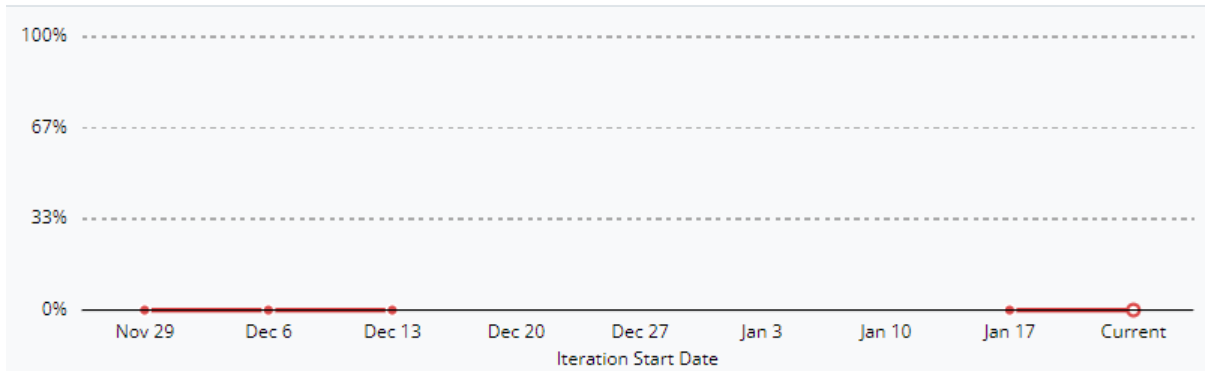
6.3.3. Story cycle time

Median Cycle Time by Iteration

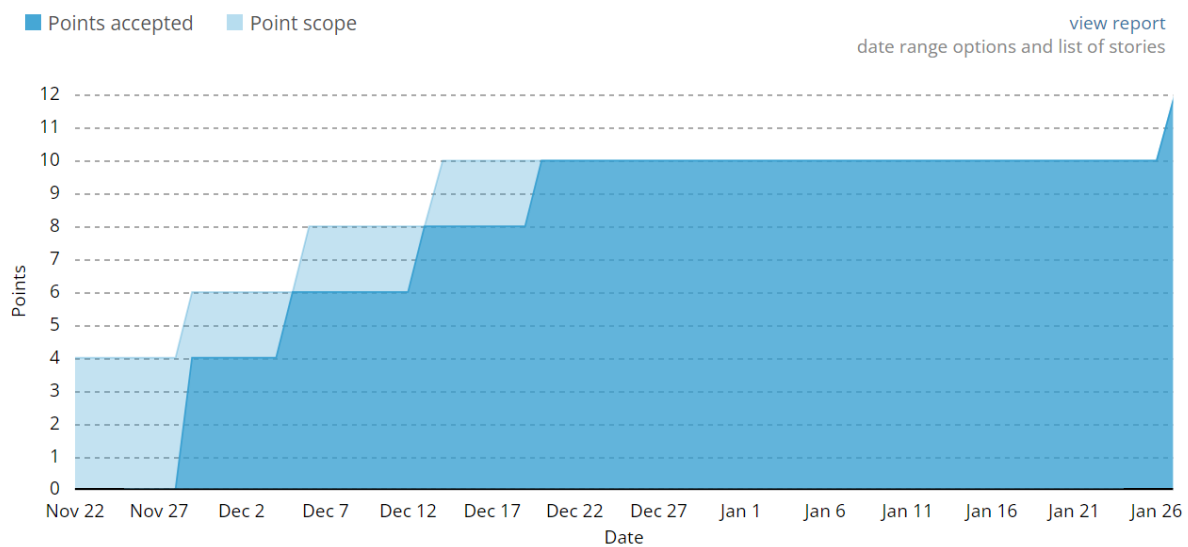


6.3.4. Rejection rate

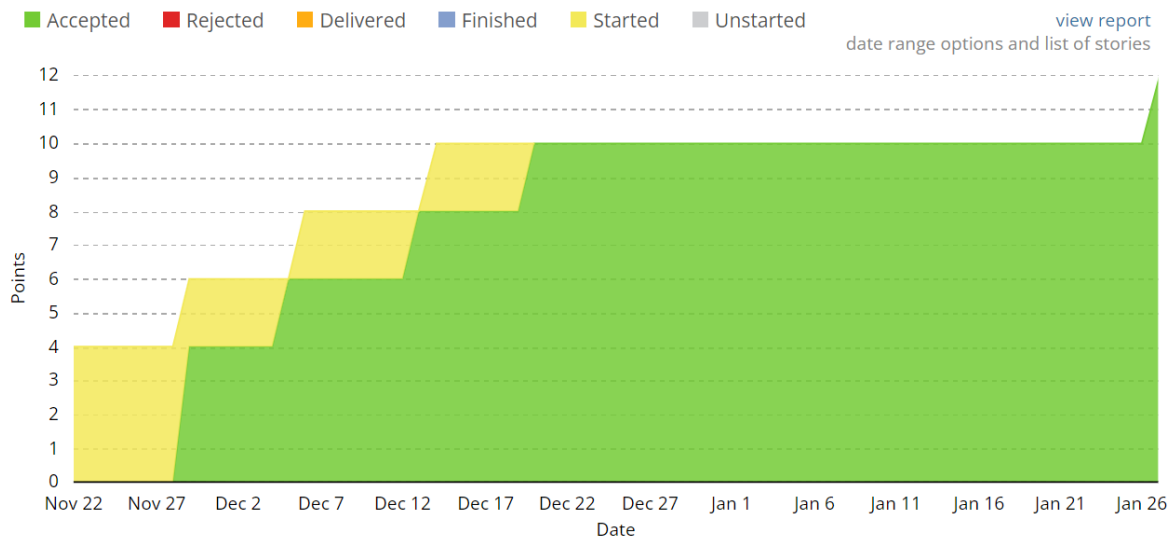
Rejection Rate



6.3.5. Burnup



6.3.6. Cumulative Flow



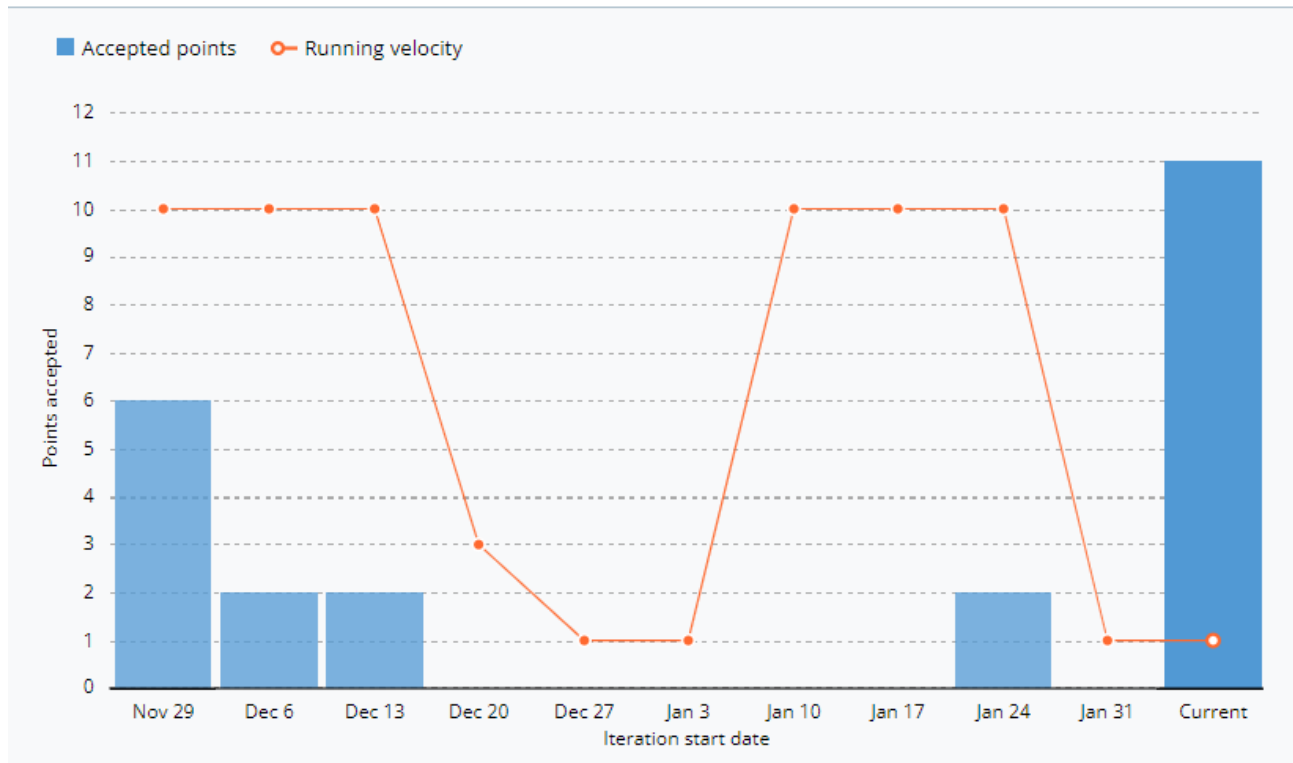
7. Pivotal Tracker

Finalmente, se quiere destacar las gráficas del trabajo realizado en Pivotal Tracker durante la última semana del proyecto que marca desde el 7 de febrero al 13 de febrero.

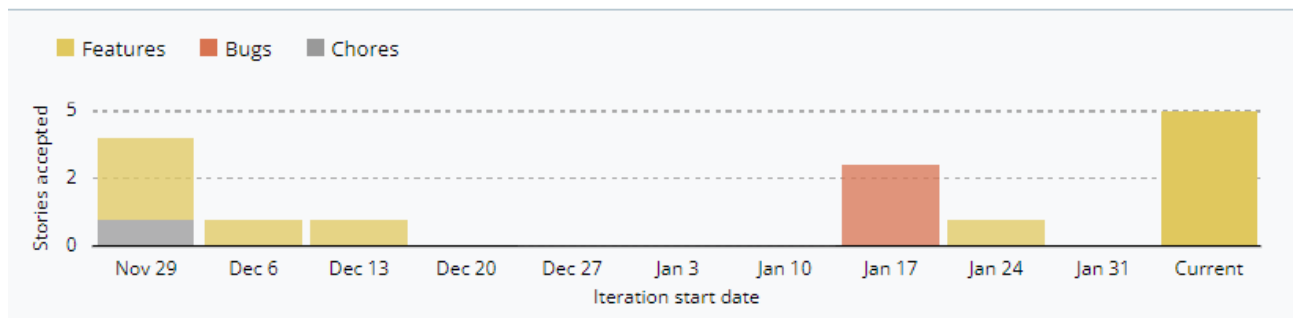
7.1. Velocity and points accepted

Points Accepted

Volatility: **141%** - Standard Deviation: **1.4pts** (based on past 3 iterations)

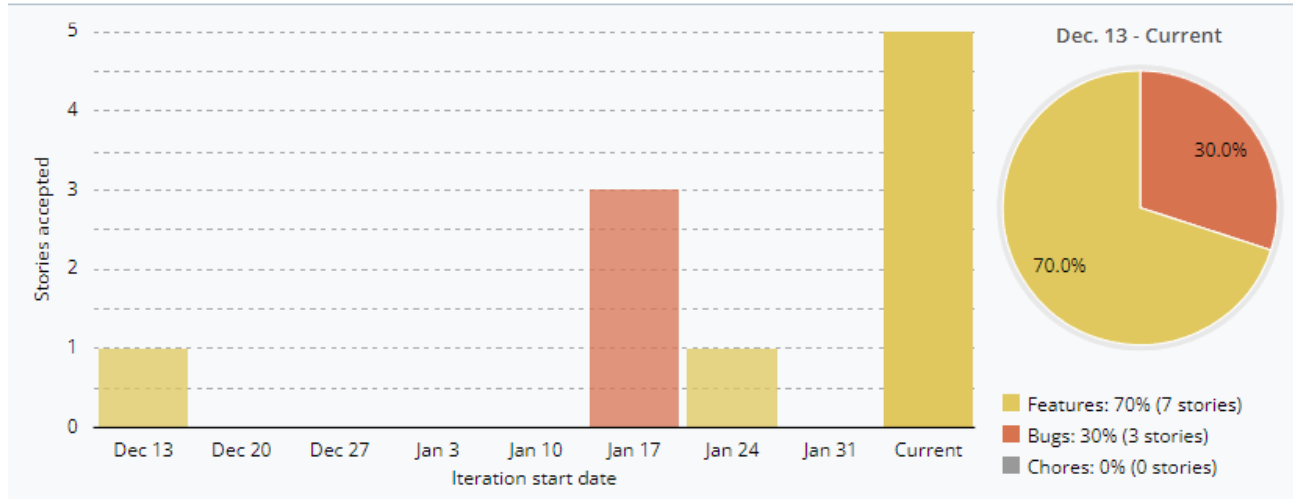


Accepted Story Type Composition

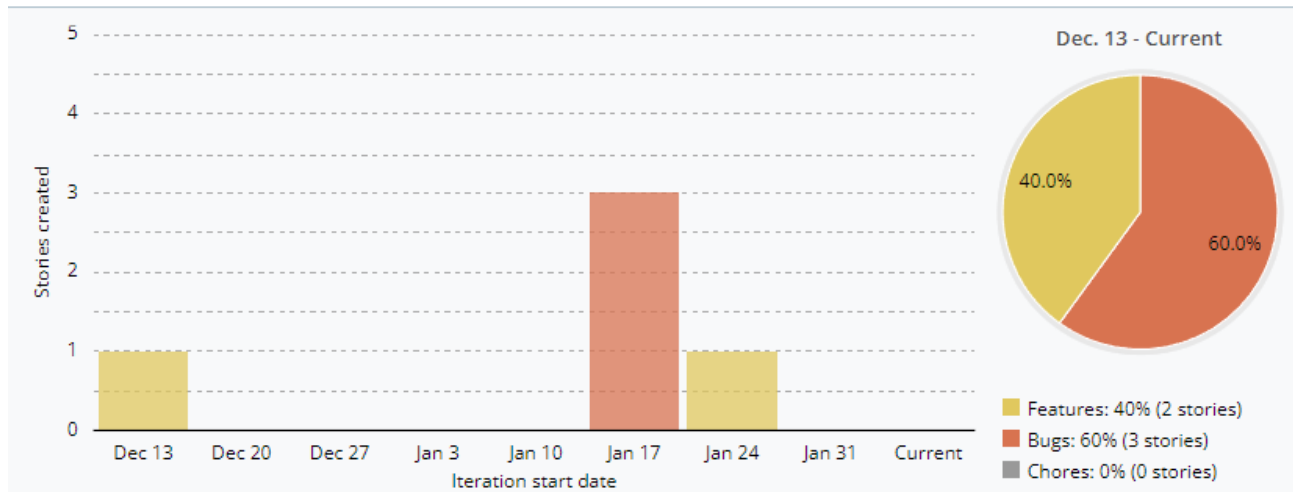


7.2. Stories accepted

Accepted Story Types

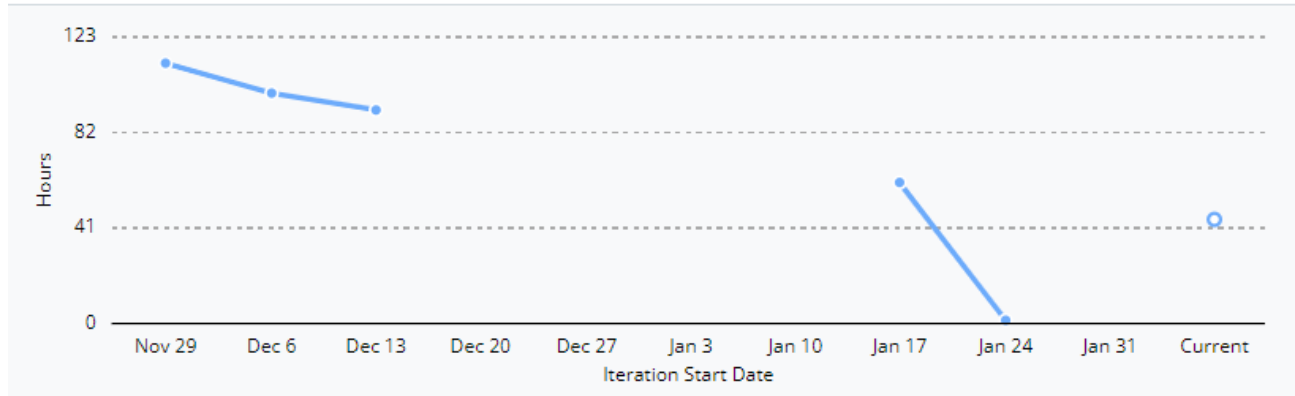


Created Story Types



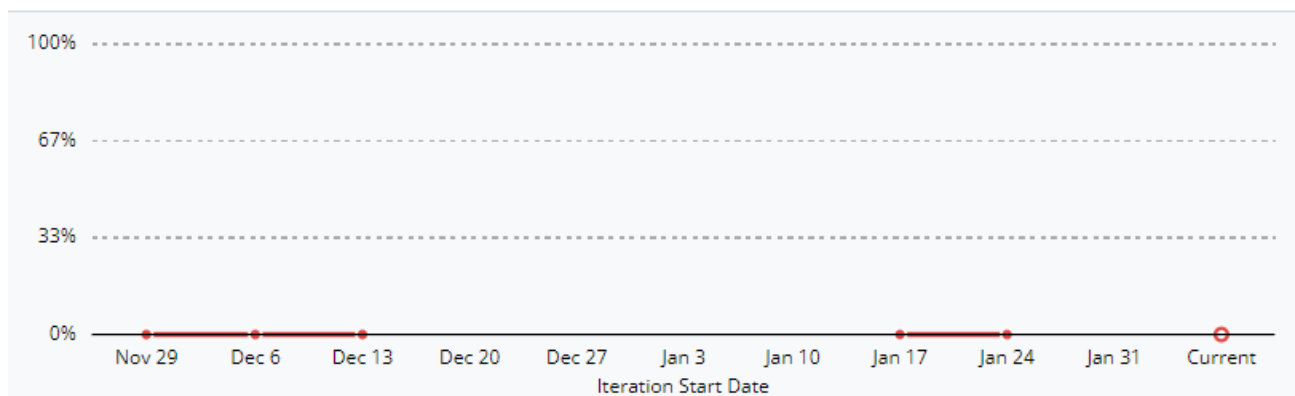
7.3. Story cycle time

Median Cycle Time by Iteration

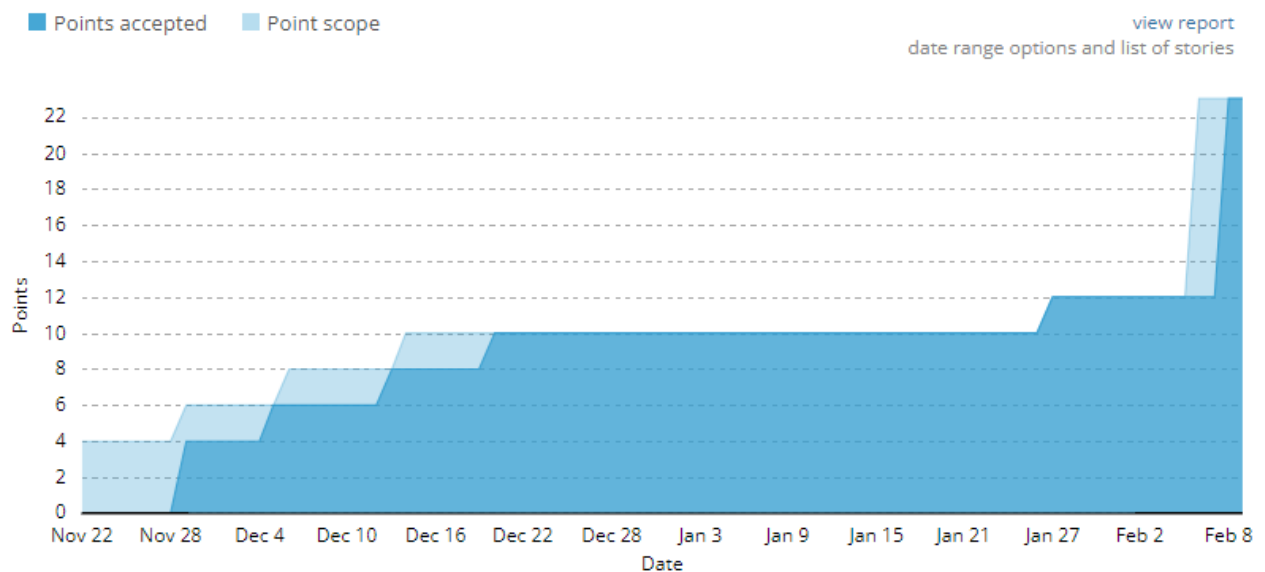


7.4. Rejection rate

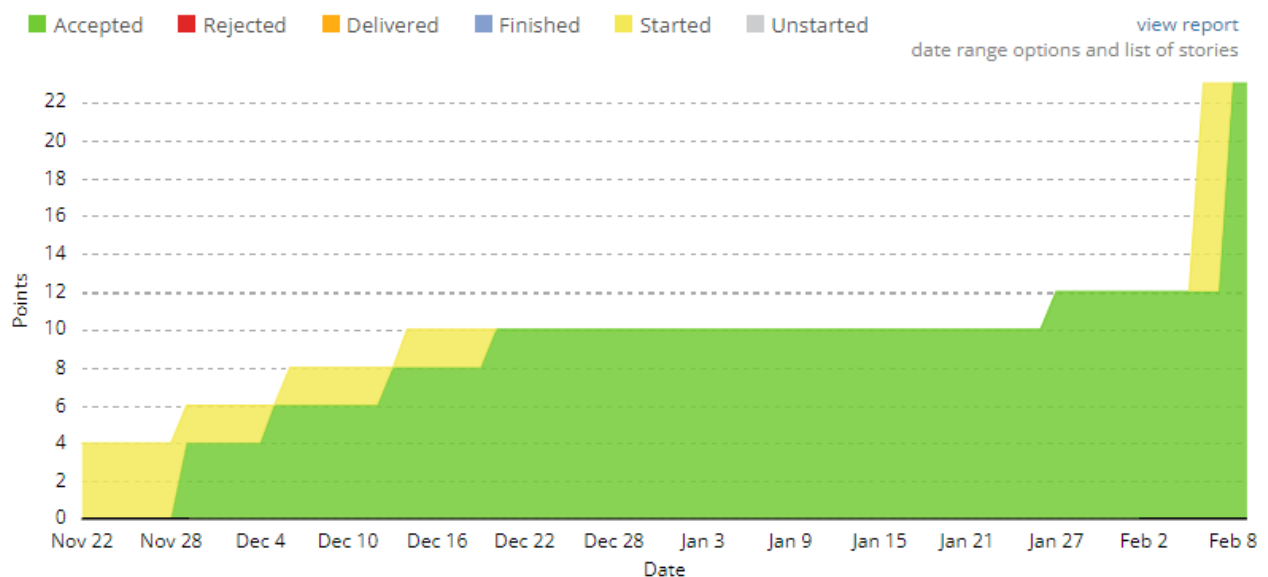
Rejection Rate



7.5. Burnup



7.6. Cumulative Flow



8. Conclusiones

En conclusión, el desarrollo de estas iteraciones nos ha permitido organizarnos y planificarnos de la mejor manera a la hora de llevar a cabo la aplicación. Es por ello que herramientas como Pivotal Tracker aportan gran ayuda a la gestión de las tareas a realizar en un proyecto donde se tiene distintos miembros.

Además, con esta aplicación se ha conseguido que tuviésemos mejores conocimientos acerca del desarrollo de las tecnologías webs. A pesar de la serie de

dificultades que han ido surgiendo a medida que se ha ido realizando el código, se ha logrado superar cada uno de estos obstáculos y avanzar en el proceso de aprendizaje.

Finalmente, podemos destacar la autonomía que se ha tenido para realizar la aplicación dado que las herramientas y las tecnologías utilizadas han sido escogidas a gusto del alumnado permitiendo una mayor comodidad e interés por parte del mismo.