



INGENIERÍA DEL SOFTWARE LABORATORIO

Especificación de requisitos y planificación de un
proyecto

Realizar la especificación de requisitos de un proyecto realizado
bajo el marco metodológico Scrum

Profesor: Luis Pedraza Gomara

Ernesto González Pradas
ernesto.gonzalez023@comunidadunir.net

Índice

Contenido

Índice.....1

Introducción2

 Descripción del Sprint Backlog3

 Primer panel y Burndown Chart5

 Cuarto panel y Burndown Chart7

 Enfoques de reutilización de software9

Conclusión.....11

Bibliografía12

Introducción

Para esta actividad, vamos a realizar la especificación de requisitos de un proyecto utilizando el marco de trabajo Scrum. La aplicación de la cual tomaremos los requisitos y realizaremos las correspondientes historias, tareas, asignaciones de tiempos a dichas tareas y burndown chart, estará basada en una plataforma de servicios de series, películas y documentales online, como por ejemplo Netflix, HBO...

Para llevar a cabo estas tareas, contaremos con un equipo de cuatro personas (Un desarrollador de back, dos de front y un QA), que trabajan a tiempo completo de lunes a viernes, es decir, por cada desarrollador contaremos con ocho horas diarias de trabajo, o lo que es lo mismo, cuarenta horas semanales.

Seleccionaremos tres historias de prioridad alta, del Product Backlog suministrado por nuestro Product Owner. Elaboraremos un sprint de cinco días en el cual descompondremos cada una de las historias en las tareas necesarias para llevarlas a cabo junto con su burndown chart correspondiente a los días primero y cuarto.

Descripción del Sprint Backlog

A continuación, pasamos a detallar los PBI (Product Backlog Items) seleccionados para su implementación en este primer sprint y posteriormente los descompondremos en sus correspondientes tareas individuales:

- Historia 1 (Hist-001): Como usuario no registrado necesito poder registrarme en la plataforma como usuario productor para poder enviar mis contenidos y monetizar mis creaciones.
- Historia 2 (Hist-002): Como usuario productor necesito enviar mis contenidos a la plataforma para que estén disponibles para los usuarios registrado y puedan verlos.
- Historia 3 (Hist-003): Como usuario no registrado necesito poder registrarme en la plataforma como usuario registrado para poder acceder a los contenidos.

Ya que estamos seleccionando las tareas de más alta prioridad para una plataforma online de series, películas y documentales, lo más básico sería empezar por las historias que permiten a los usuarios que suben contenido a la plataforma (en este caso usuario productor) darles la capacidad de poder registrarse en la aplicación y posteriormente poder subir el contenido.

Finalmente, para este primer sprint, la última tarea seleccionada es implementar un registro para que los usuarios registrados puedan acceder a los contenidos. Esta última tarea la hemos metido en este sprint ya que, aunque las otras dos llevaran más tiempo de desarrollo, se puede reutilizar el código implementado en la tarea Hist-001 en la Hist-003 con pequeñas modificaciones y así tener una primera versión del producto donde los usuarios registrados ya pueden visualizar contenidos.

Ahora vamos a descomponer cada una de las tareas vistas anteriormente en tareas individuales para poder llevar a cabo sus respectivos desarrollos:

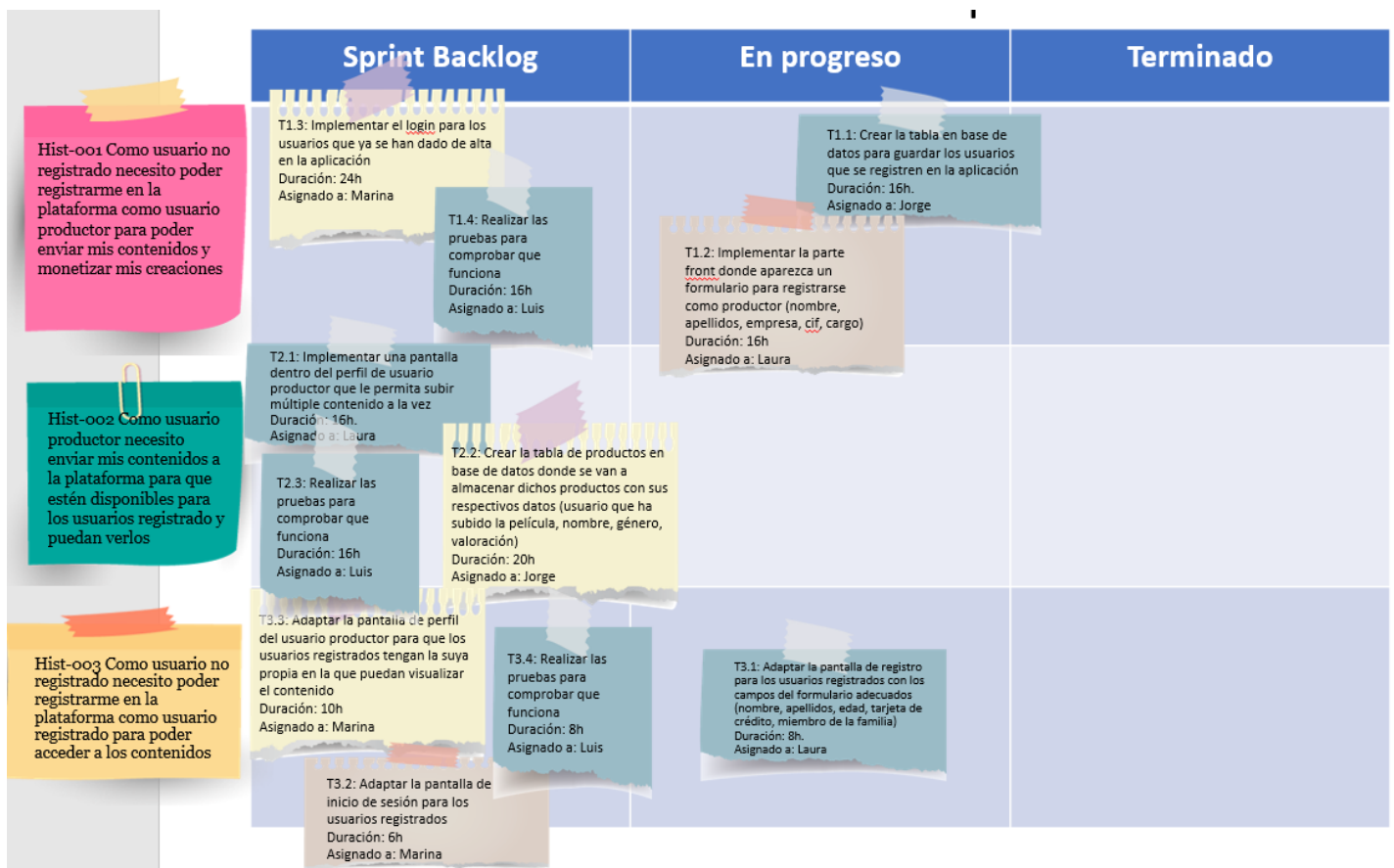
- Hist-001:
 - o T1.1: Crear la tabla en base de datos para guardar los usuarios que se registren en la aplicación.
 - o T1.2: Implementar la parte front donde aparezca un formulario para registrarse como productor (nombre, apellidos, empresa, cif, cargo).
 - o T1.3: Implementar el login para los usuarios que ya se han dado de alta en la aplicación.
 - o T1.4: Realizar las pruebas para comprobar que funciona.

- Hist-002:
 - T2.1: Implementar una pantalla dentro del perfil de usuario productor que le permita subir múltiple contenido a la vez.
 - T2.2: Crear la tabla de productos en base de datos donde se van a almacenar dichos productos con sus respectivos datos (usuario que ha subido la película, nombre, género, valoración).
 - T2.3: Realizar las pruebas para comprobar que funciona.
- Hist-003
 - T3.1: Adaptar la pantalla de registro para los usuarios registrados con los campos del formulario adecuados (nombre, apellidos, edad, tarjeta de crédito, miembro de la familia).
 - T3.2: Adaptar la pantalla de inicio de sesión para los usuarios registrados.
 - T3.3: Adaptar la pantalla de perfil del usuario productor para que los usuarios registrados tengan la suya propia en la que puedan visualizar el contenido.
 - T3.4: Realizar las pruebas para comprobar que funciona.

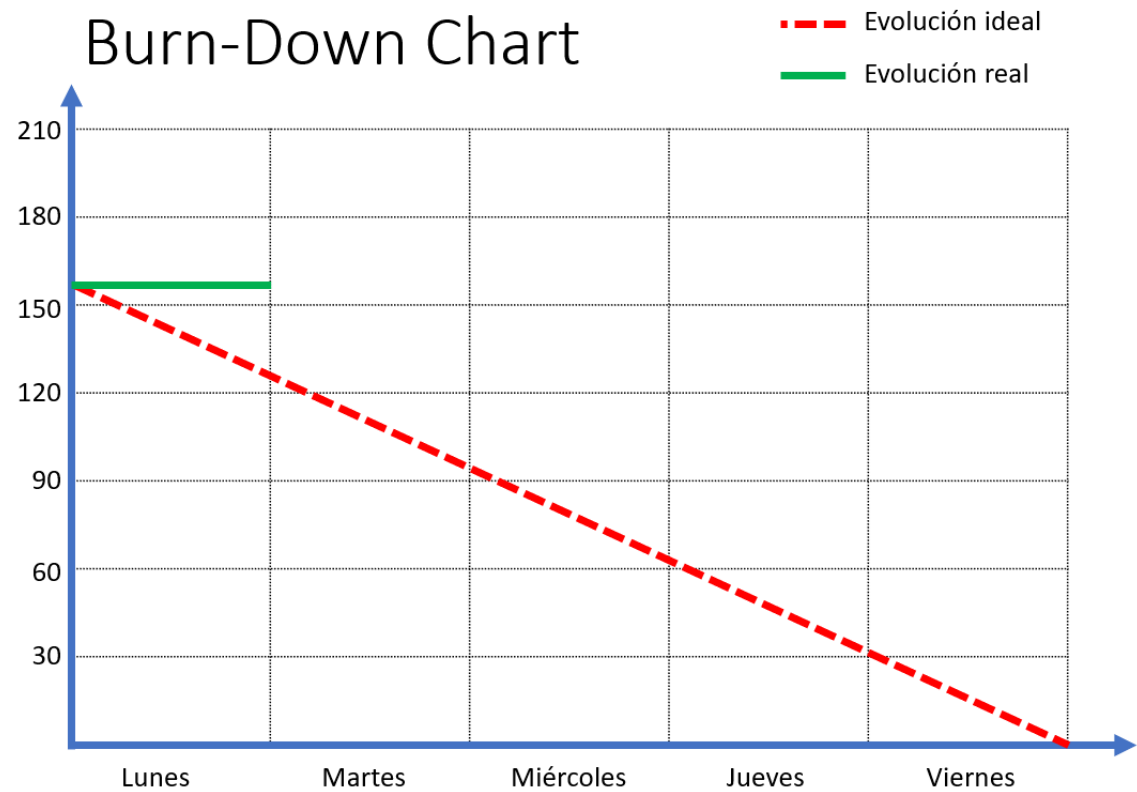
Primer panel y Burndown Chart

A continuación, vamos a mostrar el primer panel o tablero y como quedaría el burndown chart después del primer día de desarrollo, es decir, al finalizar la primera reunión diaria del sprint.

El tablero está formado por las historias que hemos descrito anteriormente y por una tabla donde iremos poniendo las tareas desgranadas de cada una de las historias en función del estado en el que estén, ya sea *Sprint Backlog* (todavía no se ha comenzado el desarrollo de una tarea), *En progreso* (la tarea correspondiente se está desarrollando) y *Terminado* (la tarea ha sido finalizada). En estas tareas especificamos lo que se tiene que hacer, quien lo tiene que hacer y la duración de esta. Por otro lado, colocamos a la izquierda de la tabla, las historias que hemos definido previamente con su código de identificación (Hist-001, Hist-002 e Hist-003) y su descripción completa.



Como podemos ver en la captura, al finalizar el primer día ya tenemos tres tareas en progreso (T1.1, T1.2 y T3.1). Continuación vamos a ver como quedaría el burndown chart del primer día:

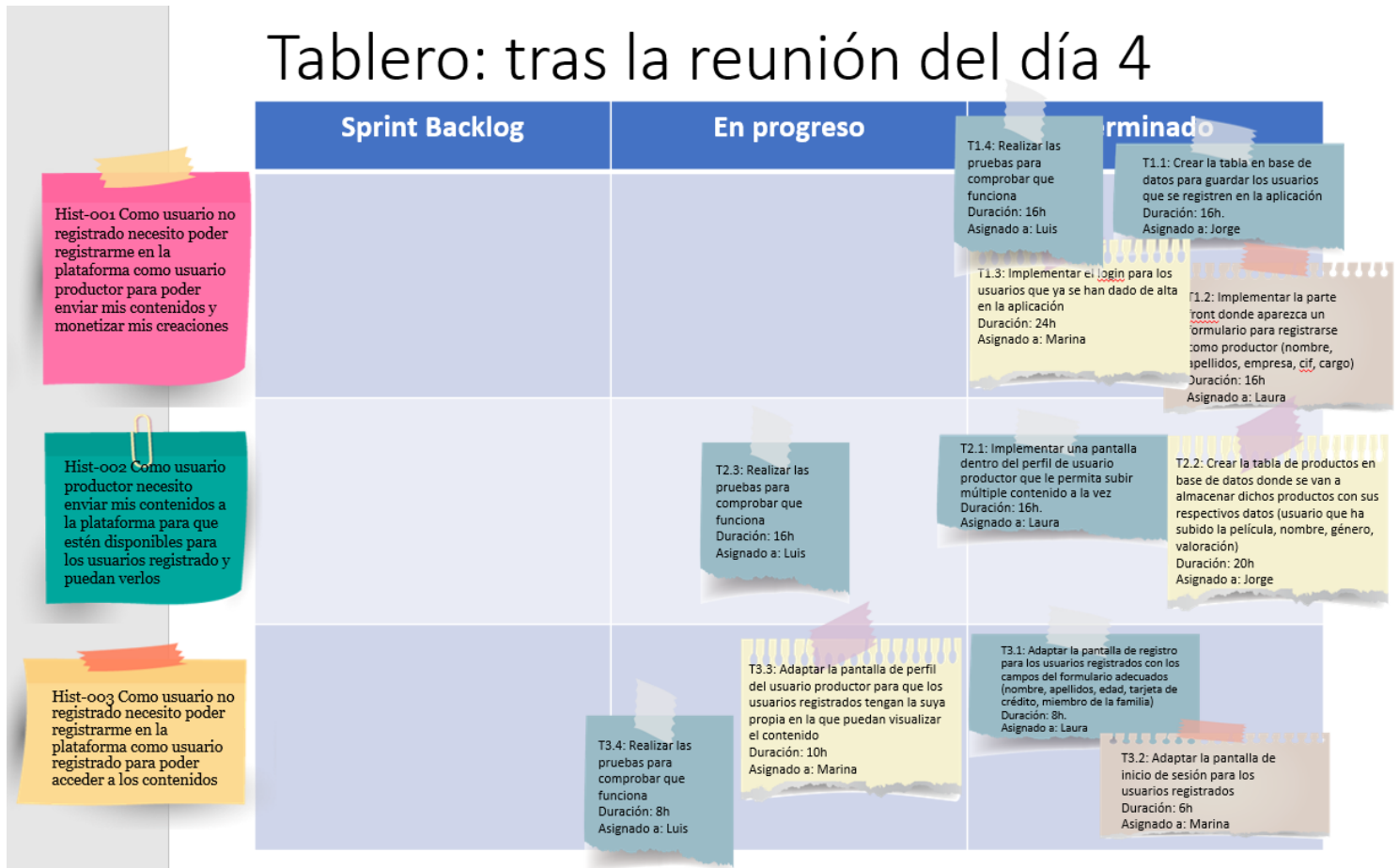


Como al finalizar el primer día, solo tenemos tres tareas en progreso y ninguna terminada, la evolución real quedaría tal y como se muestra en la captura.

Cuarto panel y Burndown Chart

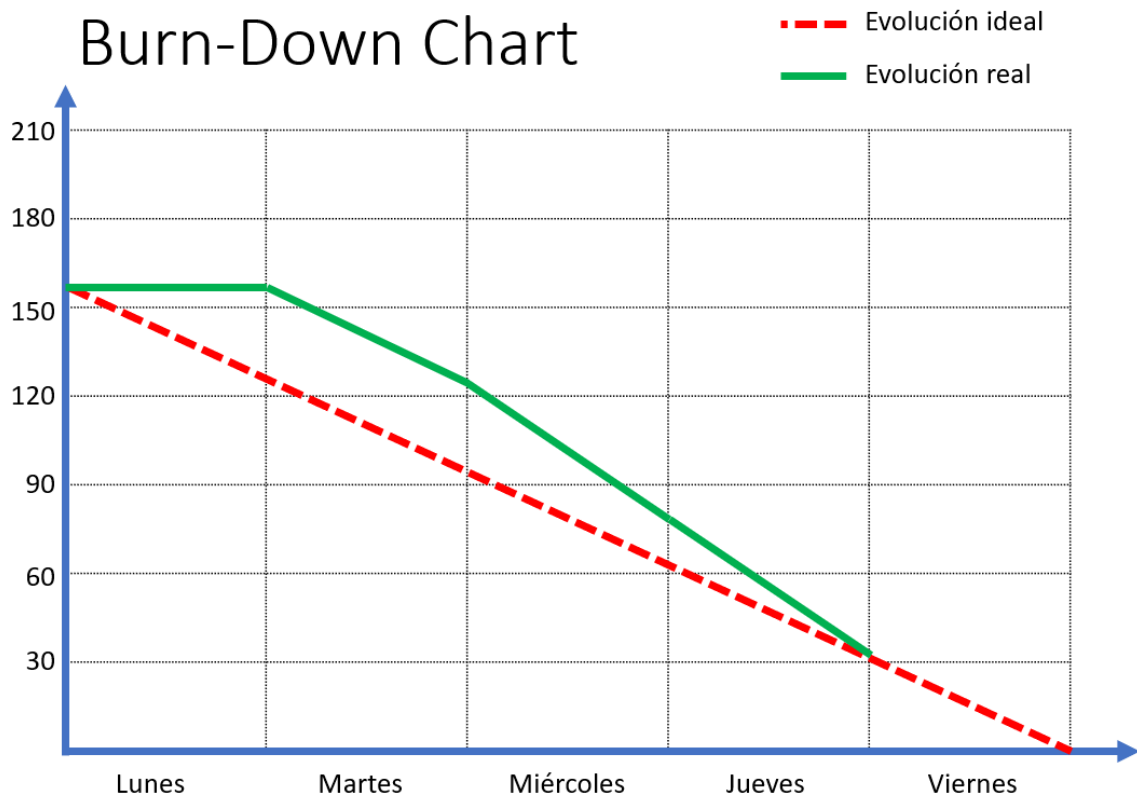
Ahora vamos a representar el tablero y el gráfico de burndown chart al finalizar el cuarto día, tras la reunión diaria del sprint.

Tablero: tras la reunión del día 4



Como podemos ver en el tablero del 4 día, ya se han terminado varias tareas (T1.1, T1.2, T1.3, T1.4, T2.1, T2.2, T3.1 y T3.2) mientras que siguen en desarrollo las tareas (T2.3, T3.3 y T3.4). Hay algunas tareas como hemos visto con anterioridad a la hora de definir cada una de ellas, que se podían reutilizar el código de una en otra.

El burndown chart del cuarto día quedaría de la siguiente forma:



Como podemos observar, los primeros días del desarrollo, el equipo empezó por tareas que llevaban más tiempo, pero posteriormente como hemos comentado con anterioridad al reutilizar desarrollos realizados para las primeras tareas en las últimas, la evolución real a podido llegar a la evolución ideal planificada en este gráfico desde el inicio.

Enfoques de reutilización de software

Para finalizar, vamos a proponer tres estrategias de reutilización de software para aplicar en nuestro proyecto de aplicación de servicio de series, películas y documentales online.

La primera estrategia de reutilización sería a nivel de aplicación, es decir, grano grueso, aplicar un framework de trabajo para el desarrollo de las tareas. Concretamente podríamos usar groovy on rails para realizar la parte back y angular 12 para la parte front. Con esta disposición de back y front estaríamos utilizando también un patrón de arquitectura, concretamente MVC (Modelo, Vista, Controlador), en el que definiríamos el **modelo** en los dominios (o entidades) de groovy, las **vistas** en los componentes de angular (html, css y ts) y el **controlador** sería una mezcla entre los service.ts que utilizamos dentro de angular (los cuales llaman a un controller del back) y los propios controllers de groovy (controllers del back) con sus implementaciones propias de las interfaces que vienen definidas por los dominios. Con esta disposición de tecnología tenemos otra ventaja y es que podemos separar totalmente lo que sería la parte back del front al más puro estilo de microservicios, es decir, el back publicaría sus controllers como un API REST y serían los service.ts de angular los que los consumirían (mediante peticiones get, post, put delete...), o en su defecto, dichos servicios los podría consumir en un futuro otra parte de la aplicación hecha en otra tecnología que no fuera angular, como por ejemplo Python para explotar datos y mostrar informes a los usuarios proveedores. Con esto estaríamos dando no solo un enfoque oportunista (aprovechamos recursos disponibles cuando son necesarios) si no que también le damos un enfoque proactivo, ya que estamos pensando a largo plazo como podemos implantar nuevas partes de la aplicación de una manera más sencilla y aislada (alto nivel de acoplamiento, bajo nivel de dependencia) mediante componentes externos (grano medio).

La segunda estrategia de reutilización sería utilizar gitflow. Gitflow es una estrategia o forma de trabajar con ramas dentro de un proyecto de desarrollo. Esta forma de trabajar nos permite tener una rama principal o “master” y una rama “develop” de la que surgirán ramas por cada tarea de nuestro proyecto, es decir, para la tarea “T2.1: Implementar una pantalla dentro del perfil de usuario productor que le permita subir múltiple contenido a la vez.” Se creará una rama a partir de develop (con el nombre por ejemplo de Tarea2.1) y se realizará el desarrollo pertinente, integrándolo posteriormente con dicha rama develop y haciendo finalmente un merge a la rama master. Esto tiene varias ventajas, mejora y facilita el control de versiones de nuestra

aplicación, permite una integridad rápida en el equipo de trabajo si entrara a trabajar un nuevo desarrollador y sobre todo permite hacer ramas “hotfix” para arreglar posibles errores en producción de una manera más sencilla de cara a integrarlo en el código que ya teníamos sin perder nada.

La tercera y última estrategia de reutilización, sería utilizar clases de groovy que ya están hechas por terceros para integrar el framework con una base de datos weblogic por ejemplo. Este nivel de granularidad sería, Grano fino ya que estaríamos reutilizando funciones para conectarnos a una base de datos mediante cadenas de conexión jdbc, por ejemplo.

Conclusión

La actividad me ha parecido muy interesante para poner en práctica los conocimientos teóricos adquiridos en clase. Es complicado ver teóricamente una toma de requisitos y como transformar eso en una pila de producto con las historias de usuario bien hechas (**como usuario** no registrado **necesito ... para poder ...**).

También el poder utilizar un panel Kanban con los post-it con sus respectivas historias y posteriormente las tareas ver como se mueven visualmente de una columna a otra. En los archivos adjuntos de la actividad, pongo todos los días del primer sprint tanto del Kanban como del burndown chart. Me parece que queda más completa la actividad si se ven todos los días trabajados, pero no los quise meter dentro de la memoria para poder sintetizar y resumir los conceptos al máximo.

Finalmente, la parte de reutilización he utilizado algunos frameworks y sistemas de trabajo que utilizo en el día a día en mi propio trabajo y que creo que podían encajar muy bien con los conceptos vistos en el tema 4 de la asignatura a un nivel global.

Bibliografía

Gomara, L. P. (1 de Marzo de 2022). Tema 02: Scrum. Madrid, Madrid, España.

Gomara, L. P. (1 de Abril de 2022). Tema 04: Reutilización del software. Madrid, Madrid, España.