

# Estimacion y Planificacion Agil

Today .2022

Victor Portugal G.





# Historias de Usuarios

---

Los campos que se consideran  
**más necesarios** para describir  
de una manera adecuada las  
historias de  
usuario son:



---

**Descripción:** descripción sintetizada de la historia de usuario. El estilo puede ser libre, según mejor nos funcione, debe responder a tres preguntas: ¿Quién se beneficia? ¿Qué se quiere? y ¿Cuál es el beneficio?

**Estimación:** estimación del **esfuerzo necesario** en tiempo ideal de implementación de la historia de usuario. Según convenga al equipo también se puede utilizar unidades de desarrollo, conocidas como **puntos de historia** (estas unidades representan el tiempo teórico de desarrollo/persona que se estipule al comienzo del proyecto).

**Prioridad:** **sistema de priorización** que nos permite determinar el orden en el que las historias de usuario deben de ser implementadas.

---

---

**Dependiendo del tipo de proyecto, el funcionamiento del equipo y la organización, pueden ser aconsejables otros campos como:**



---

**ID:** identificador de la historia de usuario, **único** para la funcionalidad o trabajo.

■ **Título:** título descriptivo de la historia de usuario.

■ **Criterio de validación:** **pruebas de aceptación** consensuadas con el cliente o usuario. Estas son los

- criterios a veces transformados en pruebas que el código debe superar para dar como finalizada la
- implementación de la historia de usuario.

■ **Valor:** valor (normalmente numérico) que aporta la historia de usuario al cliente o usuario. El objetivo del equipo es **maximizar el valor y la satisfacción percibida por el cliente** en cada iteración. Este campo servirá junto con la estimación para determinar la prioridad con el que las historias de usuario deben de ser implementadas.

---

- 
- **Dependencias:** una historia de usuario no debería ser dependiente de otra historia, pero en ocasiones es necesario mantener la relación. En este campo se indicarían los identificadores de otras historias de las que depende.
  - **Persona asignada:** en casos en que queramos sugerir la persona que pueda implementar la historia de usuario. Recordar que en **scrum** es en último término el equipo autogestionado quién distribuye y por tanto asigna las tareas.
  - **Criterio de finalización:** la definición de **finalizada/hecho** incluye los criterios o actividades necesarias para dar por terminada una historia de usuario (desarrollada, probada, documentada...), que son las convenidas por el equipo y el propietario del producto.

- 
- **Sprint:** puede ser útil para organización del propietario del producto incluir el **número de sprint** en el que previsiblemente se vaya a realizar la historia.
  - **Riesgo:** riesgo **técnico o funcional** asociado a la implementación de la historia de usuario.
  - **Módulo:** módulo del **sistema o producto** al que pertenece.
  - **Observaciones:** para **enriquecer o aclarar** la información o cualquier uso que pueda ser útil.
-



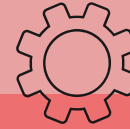
Historia de Usuario	
Número: 1	Usuario: Cliente
Nombre historia: Cambiar dirección de envío	
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: José Pérez	
<b>Descripción:</b> Como cliente quiero cambiar la dirección de envío de un pedido para que me pueda llegar a casa o a la oficina	
<b>Validación:</b> El cliente puede cambiar la dirección de entrega de cualquiera de los pedidos que tiene pendiente de envío	

Ilustración 5: Ejemplo una historia de usuario





**Si bien las historias de usuario son lo suficientemente flexibles como para describir la funcionalidad de la mayoría de los sistemas, no son apropiadas para todo. Si por cualquier razón, se necesita expresar alguna necesidad de una manera diferente a una historia de usuario, recomendamos que se haga.**



# Consejos de buenas prácticas:

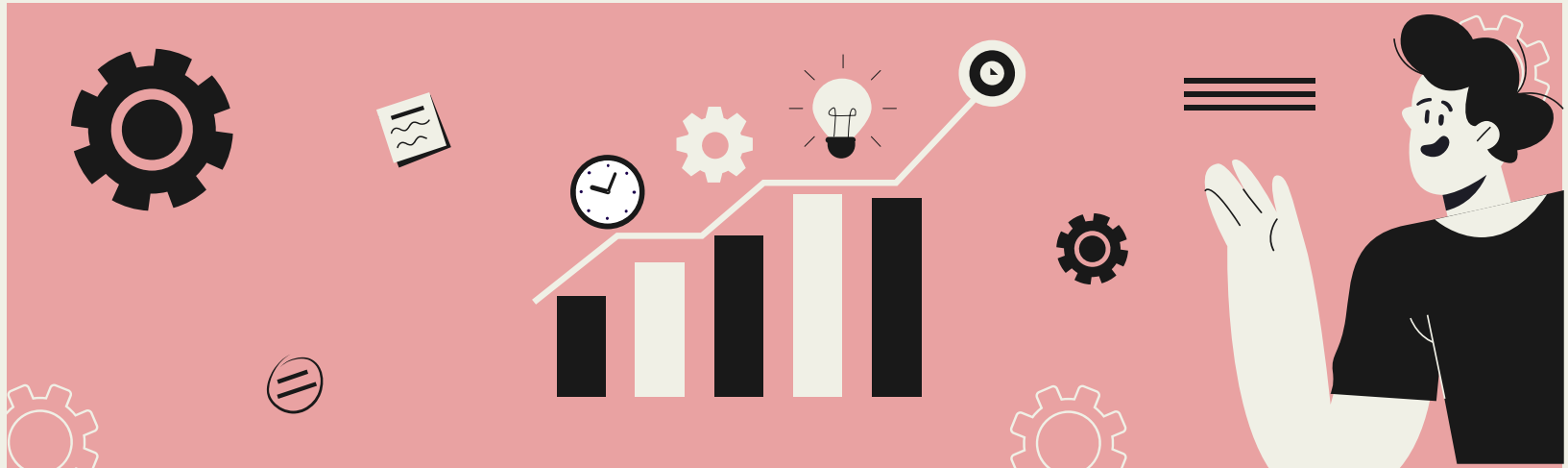
---

---

- 
- No escribir una descripción exhaustiva, solo lo justo.
  - Escribir el criterio de validación y ser suficientemente explícito.
  - Estimar todas las historias, no hacerlo puede crear falsas expectativas.
  - No fiar toda la información en las tarjetas, a veces es buena idea una documentación externa como una wiki por ejemplo.
  - Nunca dar una historia por finalizada cuando esta aun no ha sido testeada.
-

# Historias técnicas

---



- 
- La **pila de producto** contiene todos los "qués" a construir, elementos como historias de usuario, epics y temas. Pero en realidad, para poder realizar planificaciones tanto de sprint como de release, **debe de contener todo el trabajo a hacer**, y algunos de los elementos puede que no sean funcionales y sean **necesidades de carácter técnico** que no aporten valor de negocio, pero son consumidas por los elementos funcionales. También es interesante que negocio y propietario del producto tengan contacto con los aspectos técnicos, ya que así se forja un lenguaje común en ambas direcciones y se da consciencia de su importancia.
-

- 
- A estos elementos se les llama **historias técnicas** y son cosas como preparar un webserver, implementar un conjunto de tablas en una base de datos que va a ser consumida por varias funcionalidades, elementos de seguridad, escalabilidad, rendimiento, etc. Otros tipos de historias técnicas se centran en resolver deuda técnica y en refactorizaciones, otros en historias de exploración como un análisis técnico o uno funcional que sirve para despejar incertidumbre sobre alguna historia de usuario.
-

- 
- Las historias técnicas se escriben directamente en **texto técnico claro y preciso**, sin un patrón como ocurre con las historias de usuario. También tienen **criterios de validación** asociados que se comprueban en la revisión de sprint por la audiencia técnica correspondiente.
  - Aunque el propietario del producto sea el responsable de la pila y por tanto de todos sus elementos, en el caso de las historias técnicas los verdaderos "**propietarios**" son **perfiles de carácter técnico** como el equipo o un arquitecto. Estos no solo son responsables de la definición de la misma sino también de responder a dudas y preguntas y aclaraciones en la planificación y en la entrega de las historias.
-

---

Se pueden identificar  
diferentes tipos de historias  
técnicas:





- 
- ■ **Arquitectura:** construyen elementos como las API's que crean la estructura, funcionamiento e interacción entre distintas las partes del software. Ejemplo: "Implementar un sistema de login seguro".
  - ■ **Infraestructura de producto:** historias que son consumidas directamente por historias de usuario.
  - Esto podría incluir infraestructura nueva y/o modificada, y oportunidades de refactorización originada por alguna necesidad funcional. Ejemplo: "Preparar los servidores de base de datos y web".
  - ■ **Infraestructura del equipo:** historias que respaldan al equipo en su capacidad para entregar software. Suelen ser historias para herramientas y marcos de pruebas, métricas, diseño, planificación... y también pueden implicar que el equipo "desarrolle" o "compre e instale" algo. Ejemplo: "Preparar un sistema de integración continua"
-

- 
- **Refactorización:** estas son historias que representan candidatos para refactorizar, como por ejemplo lo es la deuda técnica. Pero no solo el código necesita de refactorización, también puede incluir diseños, automatización, herramientas y cualquier documentación de proceso.

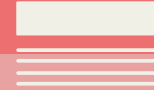
Ejemplo: "Homogeneizar el código de la función de cálculo de préstamos".

- ■ **Spikes:** historias de exploración limitadas en tiempo que dan respuesta a una cuestión, o reúnen información para una toma de decisión posterior o el diseño de una solución.

Ejemplo: "Evaluar Oracle versus SQL/Server".

---

- 
- ○ **Spike técnico:** si no estamos seguros de cómo desarrollar algo desde un punto de vista técnico creamos este tipo de spike, una breve actividad que se centra en encontrar un enfoque de desarrollo, en determinar la factibilidad y el impacto de las estrategias de diseño.
  - ○ **Spike funcional:** sirven a los equipos para descubrir los detalles de las funcionalidades y los diseños a través de la creación de prototipos y llegar a entender exactamente lo necesita el cliente.
-



# Estimación



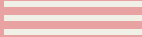

---



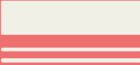


Como hemos visto la "**estimación**" es una información necesaria en las historias de usuario:



- 
- ○ Ayuda en la priorización de las mismas por parte del propietario del producto.
  - ■ Permite al equipo hacer el corte en la pila de producto de qué historias caben en el sprint
  - La estimación ágil se basa en la **estimación relativa**, y una buena serie para la estimación de historias de usuario es la **serie de Fibonacci**. Es un hecho que para historias de usuario pequeñas la incertidumbre en la estimación es mucho menor que para las historias grandes, y mientras **el tamaño de las historias crece linealmente la incertidumbre crece exponencialmente**.
-



La distancia entre los números de la serie refleja el hecho de que la incertidumbre inherente a la estimación crece proporcionalmente con el tamaño de la historia de usuario. Así, las diferencias entre 1, 2 y 3 puntos de historia son probablemente mejor comprendidas que las diferencias entre 13, 21 y 42.



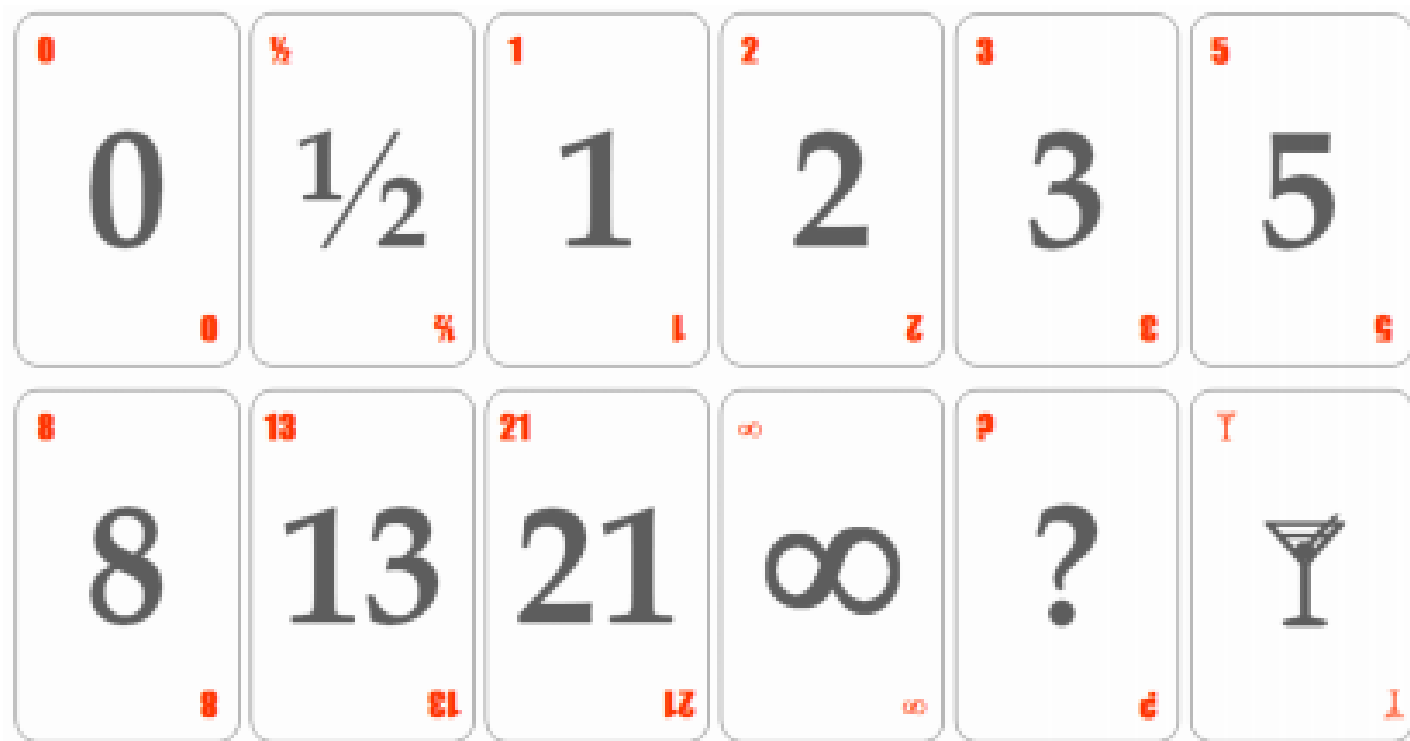


Ilustración 8: Cartas planning poker con la serie de Fibonacci



# User Story Mapping

---

**User Story Mapping** es una técnica muy útil para construir una pila de producto que vaya más allá de una lista unidimensional de historias de usuario y épicas. Esta técnica nos permite obtener los siguientes beneficios:



- 
- ■ **Visión compartida:** a través de la participación en la elaboración del User Story Map todos los involucrados construirán una visión compartida del producto a obtener, lo que facilitará mantener el foco en la solución a lo largo de su construcción.
  - ■ **Alineación:** los miembros del equipo y negocio (propietario del producto e interesados o clientes) podrán estar alineados sobre lo que se va a construir sabiendo el por qué o las razones subyacentes.
  - ■ **Mejor entendimiento de lo que quieren los clientes:** a través del modelado de los clientes y de lo que van a hacer con el producto, se puede alcanzar un alto nivel de entendimiento de las verdaderas necesidades de los clientes.
-

- 
- ■ **Mejor entendimiento de los problemas que enfrentan los clientes:** aplicando esta técnica se puede lograr un alto nivel de empatía con los clientes, tanto por participar en la dinámica, como también por tener la oportunidad de ponerse en el lugar del cliente a la hora de modelar lo que hará con el producto y cómo lo hará, pudiendo pensar en cuáles son sus problemas y cómo solucionarlos mediante el producto que se va a construir.
  - ■ **Un mapa de historias de usuario ordenado por versión:** al modelar un backbone del flujo del cliente a través del producto, podemos definir el conjunto de historias que conformarán el Mínimo Producto Viable (MVP) a sacar al mercado, que le permita a los usuarios llevar a cabo todo el flujo de una manera básica (con por lo menos una funcionalidad por cada actividad obligatoria en el flujo), así como una idea inicial de las siguientes versiones que se prevé sacar posteriormente, que puede cambiar en función del feedback y otros elementos del mercado obtenidos cuando salga a producción el MVP y cada una de las versiones siguientes.
-

El objetivo de esta técnica es que de manera conjunta se definan, descubran, prioricen y estimen las historias de usuarios y/o épicas que se prevén como parte del producto a construir.

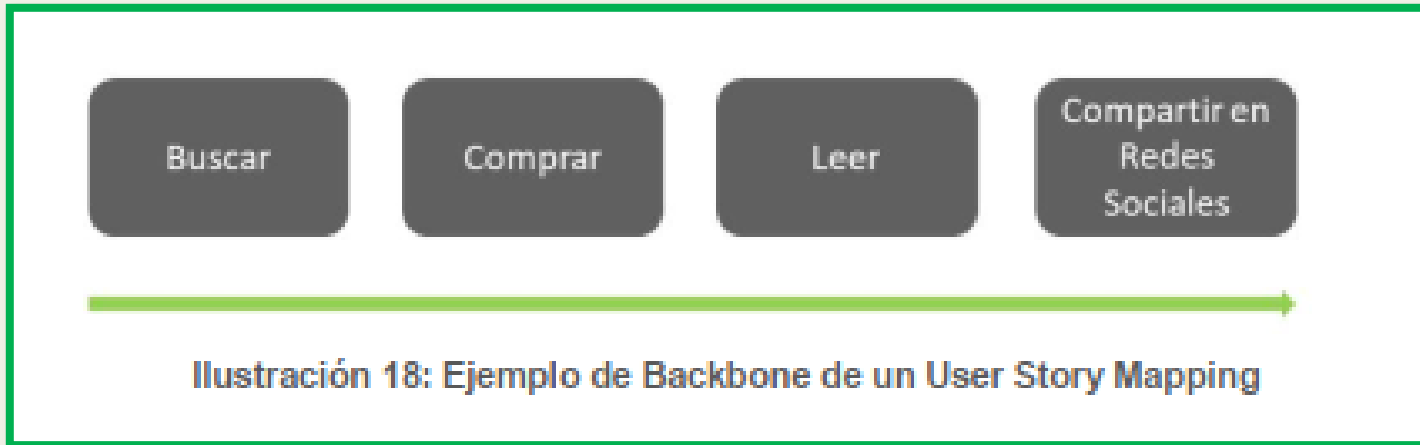


---

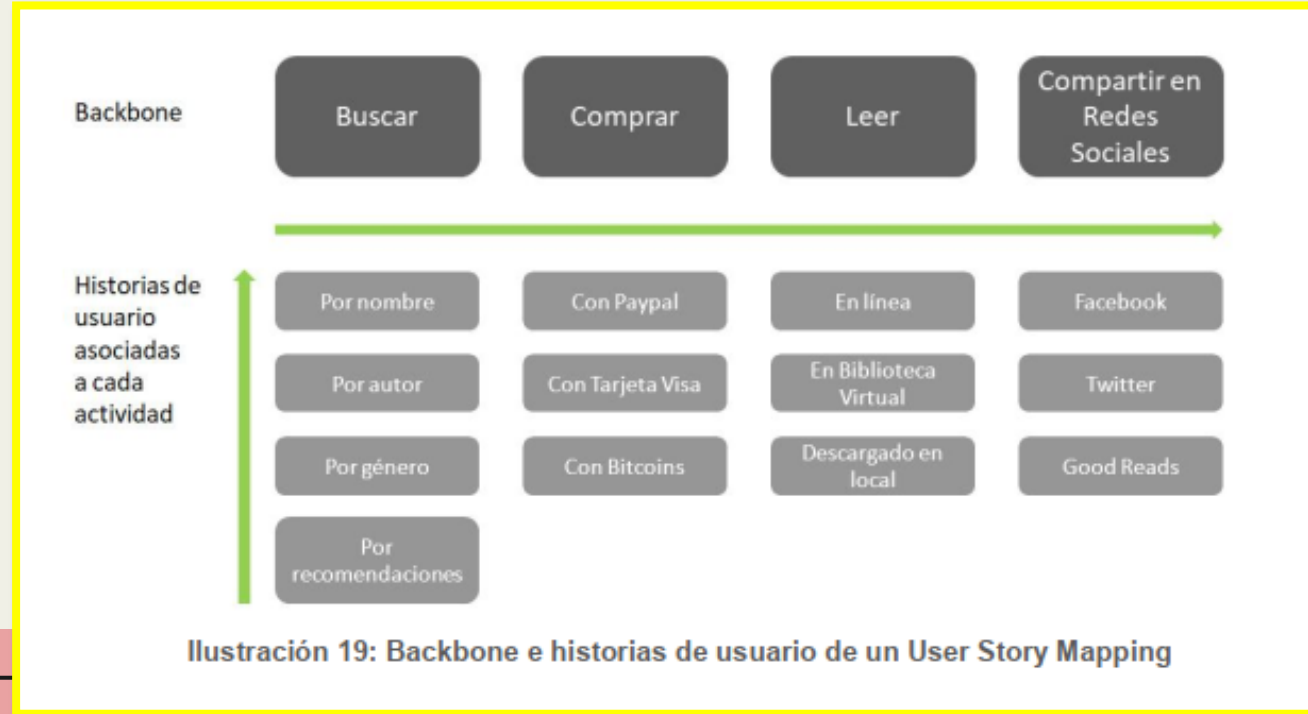
**Para emplear la técnica del User Story Mapping se van definiendo los siguientes elementos:**



- 
- 1. **Backbone del User Story Map:** el backbone o espina dorsal del User Story Map captura las actividades de alto nivel que un usuario va a realizar cuando use el producto que se quiere construir. Por ejemplo, el backbone de un proceso de compra de un libro en formato digital en una web de venta de libros on-line sería:



- 2. **Historias de usuario asociadas con cada actividad del proceso ordenadas por valor (las más valiosas en la parte superior):** para cada actividad que va a realizar el usuario con el producto se definen las historias de usuario que le van a permitir realizar la actividad. Luego ordenarlas de arriba abajo colocando las más valiosas o prioritarias más arriba. Para el ejemplo de la compra de un libro en formato digital en una web de venta de libros, un conjunto de historias de usuario asociado a cada actividad y priorizadas por valor de arriba hacia abajo sería:



- 3. **Mínimo producto viable (v 1.0) y las siguientes versiones que se prevé liberar del producto:** se determinan cuáles son las historias de usuario que compondrán el mínimo producto viable (v 1.0) y las siguientes versiones reflejando esto en el User Story Map, como por ejemplo:



# Plan de Entregas (Release Plan)

---

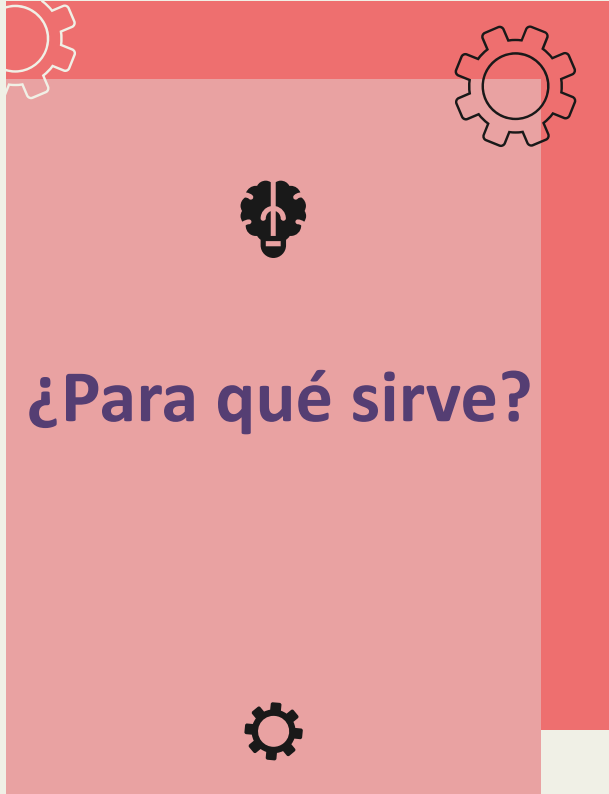


Un «release plan» o plan de proyecto es un conjunto de historias de usuario (normalmente épicas) agrupadas por «releases» o versiones del producto que se ponen a disposición de los usuarios incrementando el valor para estos respecto de la anterior.

También nos referimos a él como «plan de versiones» o «plan de entregas».

El concepto de «release» ayuda al equipo a enfocarse, les da un contexto y evita moverse sin rumbo de una iteración a otra



A slide with a light pink background and a darker pink header and footer. The header contains a white clock icon. The text is in a dark grey font and explains the importance of project planning. A black gear icon is located in the bottom left corner.

Elaborar un plan de proyecto es necesario porque: ayuda, al equipo, a decidir cuánto se debe desarrollar y cuánto tiempo se tardará antes de tener un producto entregable comunica las expectativas sobre lo que se puede desarrollar y en qué tiempo (para que el resto de la organización pueda hacerse una idea) sirve para tener una idea del progreso.

Release 1 - Comercializar Eventos				
Prioridad	Como ...	Necesito ...	Para ...	Estimación
Sprint 1 - Velocidad: 15 puntos				
1	Comercial	Crear un evento confirmado	Hacer el seguimiento del mismo	3
2	Comercial	Ver listado de eventos confirmados	No superponer eventos	2
3	Comercial	Modificar evento confirmado	Corregir cualquier error o re programarlo	2
4	Comercial	Cancelar evento confirmado	Dejar de seguirlo	1
5	Comercial	Listar los eventos en un sitio web	Que los interesados puedan verlos	2
6	Comercial	Publicar los detalles de cada evento	Que los interesados puedan verlos	5
Sprint 2 - Velocidad: 15 puntos				
7	Comercial	Generar un texto con fechas y valores	Pegarlo en los e-mail de respuesta	5
9	Interesado	Pre-Inscribirme	Iniciar la reserva de mi vacante	2
8	Comercial	Dashboard de inscripciones a cursos	Conocer el estado de completitud de cada curso	8
Sprint 3 - Velocidad: 14 puntos				
10	Comercial	Ser notificado de cada inscripción	Poder reaccionar en tiempo real frente a cada una	2
11	Comercial	Confirmar la inscripción sin pago (pago a cuenta)	Financiar ciertas vacantes	2
12	Comercial	Conocer los Pagos Pendientes por evento	Realizar el seguimiento de los pagos	3
13	Interesado	Pagar en efectivo	Confirmar mi vacante	1
14	Interesado	Pagar con Cheque	Confirmar mi vacante	1





## USER MANAGEMENT

## BOOK ROOM

## CHANGE RESERVATION

Register

Login

Update preferences

Search for room

View room info

Back room

View reservation

Change reservation

Cancel reservation

Create new user

Create login form

By room type

Show pictures

Credit card form

By reservation #

Change date

### RELEASE 1

By availability

View room details

Change Room

By location

### RELEASE 2

Update name and email

Integrate with 3rd party

PayPal

By username

Cancel online

Add payment information

### RELEASE 3

Embed room registration form on homepage

Make room details available via API

Google Checkout

Change reservation API call

### BACKLOG

Google Account Login

Upload profile image

Book room through Expedia.com

Facebook Login

Book room through Kayak.com

