

«Talento Tech»

Testing QA

Clase 14



Clase N°14 | Introducción a Jira

Temario:

- Introducción básica a Jira: interfaz y navegación
- Estructura de Jira: Proyectos, Tableros, Backlog
- Fundamentos de Scrum en Jira: Sprints y su configuración
- Tipos de tickets: Epic, Story, Task, Bug, Sub-task
- Creación de tickets paso a paso:
 - Campos obligatorios y opcionales
 - Descripción efectiva de un bug
 - Adjuntar capturas de pantalla y evidencias
- Estimaciones en Jira: Story Points, Planning Poker
- Asignación de responsables y colaboración en equipo

Objetivos de la clase

En esta clase vamos a introducirnos en el uso de **Jira**, una de las herramientas más utilizadas por equipos de desarrollo y QA para la gestión de proyectos ágiles. Aprenderás a navegar por su interfaz, entender su estructura (proyectos, tableros, backlog), y crear distintos tipos de tickets como épicas, historias, tareas técnicas, bugs y subtareas. Exploraremos cómo configurar sprints, organizar el trabajo en un entorno Scrum y documentar correctamente cada caso. También haremos un ejercicio práctico guiado paso a paso.

Introducción básica a Jira: interfaz y navegación



Jira es una plataforma de gestión de proyectos desarrollada por Atlassian. Se utiliza para planificar tareas, seguir errores, colaborar entre equipos y aplicar metodologías ágiles como Scrum y Kanban.

¿Cómo registrarse en Jira y crear una cuenta gratuita?

Para comenzar a utilizar Jira, necesitás crear una cuenta en la plataforma de Atlassian. La versión gratuita es más que suficiente para realizar prácticas y proyectos pequeños de gestión ágil.

Pasos para registrarte en Jira (gratuitamente):

1. Ingresá a: <https://www.atlassian.com/software/jira>
2. Hacé clic en el botón **"Get it free"** o **"Empieza gratis"**.
3. Podés registrarte con tu cuenta de Google o crear un usuario con email y contraseña.
4. Elegí el nombre de tu sitio (será algo como tunombre.atlassian.net).
5. Seleccioná **Jira Software** como producto principal.
6. Elegí entre el modo de trabajo **Scrum** o **Kanban** (para esta clase recomendamos Scrum).
7. Completá la configuración inicial del proyecto y ya podrás comenzar a crear tus tickets.

💡 **Consejo:** No es necesario agregar otros usuarios en esta etapa. Podés usar el entorno tú solo para practicar.

Interfaz principal

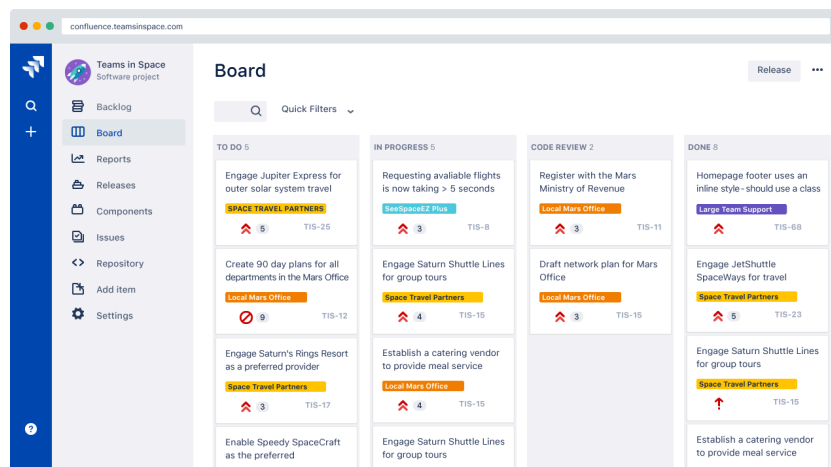
- **Menú lateral izquierdo:** Aquí podés acceder a tus proyectos, tableros, filtros guardados y configuraciones.
- **Buscador general (🔍):** Permite buscar tickets por código (ejemplo: JOB-123), palabra clave, etiquetas, etc.
- **Perfil de usuario (arriba a la derecha):** Cambiar idioma, notificaciones, o cerrar sesión.

Navegación por proyecto

1. Hacer clic en "Projects" > "View all projects" y seleccionar uno.
2. Dentro del proyecto, acceder a:
 - **Board (tablero):** Muestra visualmente las tareas divididas en columnas (To Do, In Progress, Done).
 - **Backlog:** Lista priorizada de todas las tareas aún no comenzadas.
 - **Reports:** Informes automáticos de seguimiento (burn down chart, velocity, etc.).

Estructura de Jira: Proyectos, Tableros, Backlog

- **Proyecto:** Agrupa todas las tareas y tableros relacionados a un objetivo común. Puede ser de desarrollo, soporte, QA, etc.
 - *Ejemplo:* Proyecto "Talento Lab Web" agrupa todas las tareas del equipo frontend.
- **Tablero (Board):** Representa visualmente el estado de las tareas mediante columnas. Por ejemplo:
 - To Do (por hacer)
 - In Progress (en proceso)
 - In Review (en revisión)
 - Done (finalizado)
- **Backlog:** Es la lista de tareas pendientes que todavía no están dentro de un sprint. Se utiliza para priorizar y planificar próximos sprints.



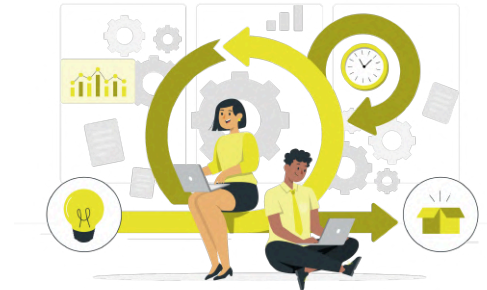
Fundamentos de Scrum en Jira: Sprints y su configuración

Jira permite implementar la metodología Scrum a través de su backlog interactivo y tableros.

- **Sprint:** Es un ciclo de trabajo (generalmente 1 o 2 semanas) donde se seleccionan tareas del backlog para ser completadas.

¿Cómo se configura un sprint?

1. Desde el backlog, hacer clic en "Create Sprint".
2. Arrastrar las tareas seleccionadas dentro del sprint.
3. Al iniciar el sprint, Jira comenzará a registrar el progreso (incluye gráficas como Burndown Chart).
4. Al finalizar, se cierra el sprint y se analiza qué tareas se completaron o quedaron pendientes.



Tipos de tickets en Jira (con ejemplos)

● Epic

- Es un conjunto de funcionalidades grandes que pueden dividirse en varias historias o tareas.
- *Ejemplo:* "Sistema de pagos con tarjeta de crédito y PayPal".

Proyectos / Proyecto Talento Tec... / SCRUM-7

Sistema de pagos con tarjeta de crédito y PayPal

[+ Añadir](#) [🔗 Aplicaciones](#)

Descripción

Esta epica tiene como objetivo lograr que se pueda pagar con tarjeta de credito y PayPal

Incidencias secundarias

0 % hecho

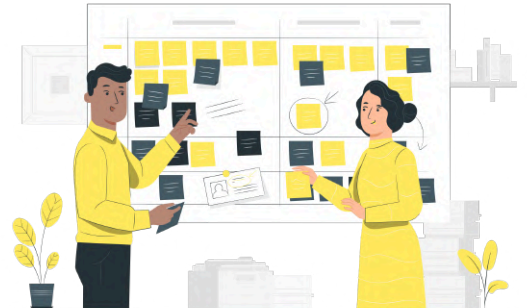
T...	Clave	Resumen	P...	P...	Estado	
<input checked="" type="checkbox"/>	SCRUM-2	Pago con tarjeta	=		TAREAS POR ...	

Como ves en la imagen, la épica tiene asociada las tareas relacionadas a sí misma

A continuación vamos a crear esa tarea

● User Story

- Describe una necesidad o funcionalidad desde la perspectiva del usuario.
- Tiene una estructura recomendada: **“Como [rol], quiero [acción] para [beneficio]”**.
- *Ejemplo:* "Como usuario registrado, quiero ingresar los datos de mi tarjeta para poder pagar en línea".



● Task

- Es una tarea técnica específica que no necesariamente está relacionada con una necesidad del usuario.
- *Ejemplo:* "Diseñar estructura de base de datos para almacenar pagos".

Proyectos / Proyecto Talento Tec... / SCRUM-7 / SCRUM-2

Diseñar estructura de base de datos para almacenar pagos

+ Añadir

Aplicaciones

Descripción

La base de datos debe ser realizada en MySql

● Bug

- Es un error o comportamiento inesperado del sistema.
- Se reporta con detalle, pasos para reproducir, resultado esperado y obtenido.
- *Ejemplo:* "Bug: Al ingresar una tarjeta vencida, el sistema permite continuar el pago sin mostrar error".

Añadir epic / SCRUM-3

Al ingresar una tarjeta vencida, el sistema permite continuar el pago sin mostrar error

+ Añadir

Aplicaciones

Descripción

T

Descripción del error

Pasos necesarios para reproducirlo

Grabar un video de Loom Usa el video para compartir, diagnosticar y corregir errores m... [PRUEBALO](#) ...

[Guardar](#) Cancelar

● Sub-task

- Son tareas más pequeñas que dependen de una Task o User Story.
- *Ejemplo:* "Subir componente de validación de fecha a repositorio" dentro de la Task de frontend.

Creación de tickets paso a paso

Campos más comunes:

- **Summary (Resumen):** Título corto. Ej: "Validar fecha de expiración en formulario de tarjeta".
- **Description (Descripción):** Debe detallar claramente el propósito, contexto, pasos, errores, etc.
- **Assignee:** Responsable que trabajará en la tarea.
- **Priority:** Determina urgencia (High, Medium, Low).
- **Labels:** Ayudan a agrupar por temas o módulos (ej: **frontend**, **urgente**, **QA**).
- **Attachments:** Se pueden subir imágenes, archivos, capturas, videos, logs, etc.

¿Cómo describir un Bug correctamente?

1. **Resumen claro:** "Error al validar número de tarjeta inválida".
2. **Entorno:** Navegador Chrome v111, entorno de testing.
3. **Pasos para reproducir:**
 - Ir al formulario de pago.
 - Ingresar número de tarjeta inválido: 1234 5678 9012 3456.
 - Hacer clic en "Pagar".
4. **Resultado esperado:** Debería aparecer mensaje "Tarjeta inválida".
5. **Resultado obtenido:** No muestra ningún mensaje y permite continuar.
6. **Adjuntar captura del error.**

Añadir epic / SCRUM-3

Error al validar número de tarjeta inválida

+ Añadir Aplicaciones

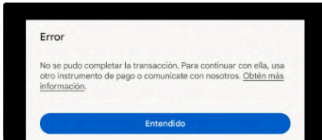
Descripción

T B I ... A

Error al validar número de tarjeta inválida

1. **Entorno:** Navegador Chrome v111, entorno de testing.
2. **Pasos para reproducir:**
 - Ir al formulario de pago.
 - Ingresar número de tarjeta inválido: 1234 5678 9012 3456.
 - Hacer clic en "Pagar".
3. **Resultado esperado:** Debería aparecer mensaje "Tarjeta inválida".
4. **Resultado obtenido:** No muestra ningún mensaje y permite continuar.

Adjuntar captura del error.



Estimaciones en Jira: Story Points, Planning Poker y más

Al trabajar en metodologías ágiles como Scrum, es importante **estimar las tareas** antes de iniciar un sprint. Esto permite al equipo planificar su capacidad de trabajo de forma realista.

¿Qué son los Story Points?

Los **Story Points** son una unidad de medida relativa utilizada para estimar el **esfuerzo total requerido** para completar una historia de usuario o tarea. No miden tiempo exacto, sino el nivel de dificultad, complejidad técnica, cantidad de trabajo y riesgos asociados.

- 1 punto: tarea muy simple, sin dependencias ni riesgos.
- 3 puntos: tarea con complejidad media, requiere validaciones o pequeños desarrollos.
- 8 puntos: tarea compleja, múltiples pasos, posible integración con otros sistemas.

💡 **Ejemplo:** Crear un botón con diseño y comportamiento básico = 1 punto. Implementar autenticación OAuth con validación externa = 8 puntos.

Planning Poker

Es una técnica colaborativa de estimación:

1. Cada miembro del equipo tiene cartas con puntos (1, 2, 3, 5, 8...).
2. Se lee la historia o tarea.
3. Cada persona muestra su carta al mismo tiempo.
4. Si hay mucha diferencia, se discuten las razones.
5. Se vuelve a votar hasta lograr consenso.

Also available as Jira plugin. [Check it out](#)

Scrum Poker for agile teams

Easy-to-use and fun estimations.

[Start new game](#)

TRUSTED BY TEAMS AT

Microsoft Google hp Deloitte. Figma

Issues

Module UI	Vote	3
Automation Test	Voting now	-
Refactor	Vote	-
Add CTA	Vote	-

0 1 2 3 5 8 13 ?

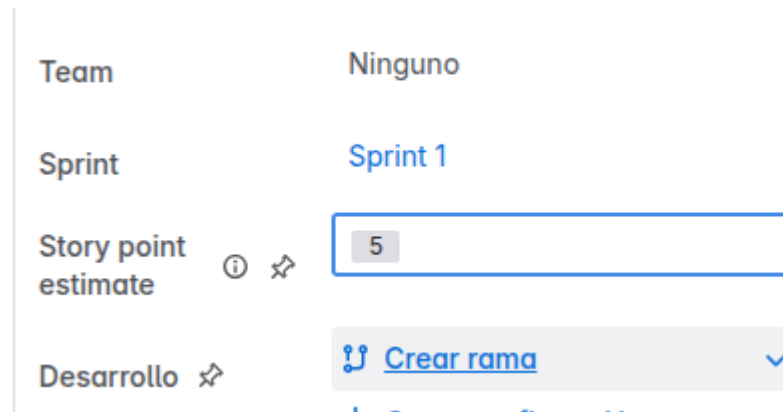
✅ Esto fomenta el diálogo, detecta puntos de vista diferentes y alinea expectativas.

¿Se puede estimar en horas?

Sí. Jira también permite estimar en **horas o días**, aunque lo más común en equipos Scrum es usar Story Points para estimaciones relativas. Las horas pueden ser útiles para tareas muy técnicas o mantenimiento.

¿Dónde se coloca la estimación en Jira?

- En el campo **"Story Points"**.



The screenshot shows a portion of the Jira ticket creation interface. On the left, there is a vertical sidebar with labels: 'Team', 'Sprint', 'Story point estimate', and 'Desarrollo'. To the right of these labels are their respective values: 'Ninguno' for Team, 'Sprint 1' for Sprint, a text input field containing '5' for Story point estimate, and a dropdown menu showing 'Crear rama' for Desarrollo. The 'Story point estimate' field is highlighted with a blue border.

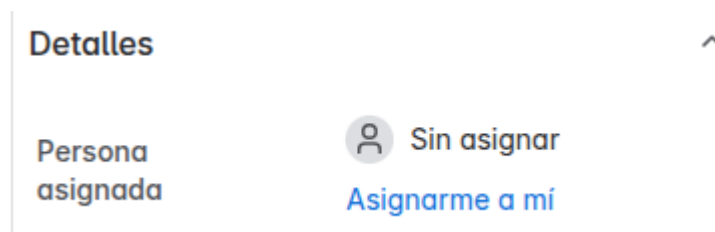
💡 Recordá que estimar bien ayuda a evitar sprints sobrecargados o mal planificados.

Asignación de responsables en Jira

Cada ticket creado en Jira debe tener un **responsable asignado**, es decir, la persona encargada de ejecutar la tarea o resolver el bug. Esto permite distribuir el trabajo, dar seguimiento y evitar tareas abandonadas.

¿Cómo se asigna un ticket?

1. En el campo **"Assignee"** (Responsable), seleccionar un miembro del equipo.
2. Si nadie está asignado, Jira lo mostrará como "Unassigned" o "Sin asignar".



The screenshot shows the 'Detalles' section of a Jira ticket. Under the heading 'Person assigned', there is a person icon followed by the text 'Sin asignar'. Below this, there is a blue link that says 'Asignarme a mí'.

3. También se puede reasignar más adelante si la tarea cambia de manos.

Detalles ^ 

Persona asignada  Emiliano Spinoso


Buenas prácticas al asignar:


- Asegurarse de que cada ticket tenga una persona asignada antes de comenzar el sprint.
- Asignar según especialidad (ej. bugs de frontend a desarrolladores de UI).
- Evitar la sobreasignación: una persona con muchas tareas pierde el foco.











Tip:

Si no sabés quién debería tomar una tarea, dejala como “Unassigned” pero agregá un comentario etiquetando a alguien (ej. [@María](#), ¿podés tomar esto?).

Actividad

Todo Comentarios Historial Registro de actividad 



T ▾ | B | I | ... | A ▾ |   |        + ▾ | 

[@JoseFernandez](#) crees que puedes tomar esta tarea?

Buenas prácticas al usar Jira

- Usar títulos descriptivos y claros.
- Priorizar bien según impacto y urgencia.
- Agregar capturas o grabaciones al reportar bugs.
- No dejar tareas sin responsable asignado.
- Actualizar el estado de los tickets a diario.
- Documentar bien historias y tareas para facilitar colaboración.

Material Complementario

- [Guía oficial de Jira para principiantes \(Atlassian\)](#)
- [Guía de buenas prácticas para escribir bugs](#)
- [Scrum en Jira explicado por Atlassian](#)
- [Curso gratuito de Jira en YouTube \(Atlassian\)](#)
- [Guía de Story Points y Planning Poker](#)

¡Trabajando en Talento Lab!

Silvia y Matías te llaman a una reunión rápida de planificación. El equipo de Talento Lab acaba de confirmar el inicio del desarrollo de un nuevo módulo: el sistema de pagos online para los postulantes premium.



Silvia te dice:



“Vamos a comenzar desde cero este módulo, así que necesitamos dejar todo bien organizado en Jira. Quiero que prepares una estructura de tickets clara, siguiendo la metodología Scrum que venimos aplicando.”

Matías agrega: “Empecemos con una épica que agrupe toda la funcionalidad. Luego vamos desglosando en historias, tareas técnicas y hasta bugs si ya detectamos alguno. No te olvides de estimar los tickets, asignarlos y etiquetarlos correctamente. Esto nos va a ayudar a tener visibilidad y trazabilidad desde el día uno.”



Te entregan un resumen funcional del nuevo módulo y te piden que cargues todo en Jira de forma estructurada para el próximo sprint.

Ejercicio práctico.

Crear y organizar tickets en Jira

Llevar tu trabajo de las Clases 1–8 a Jira

Objetivo: Replicar en Jira todo el backlog relacionado a <https://talentolab-test.netlify.app/>, los test cases y los defectos que diseñaste en las primeras ocho clases, organizándolos en un proyecto Scrum real.

1. Crea un proyecto

- Nombre: **Talento Lab Web – QA**
- Tipo: Scrum

2. Configura tu Backlog

- Importa o crea dos **Epics** basadas en tus fases iniciales:
 - **Storytelling**
 - **Requerimientos & User Stories**
- Bajo cada Epic, crea las **User Stories** que redactaste (mínimo 2 por Epic), usando el formato:
Como [tipo de usuario], quiero [acción] para [beneficio].

3. Agrega las tareas técnicas

- Para cada User Story, crea una **Task** que represente:
 - Diseño de casos de prueba
 - Ejecución manual
 - Automatización inicial (si aplicó)
- Dentro de cada Task, crea al menos una **Sub-task** (por ejemplo: maquetar formulario, escribir Test Case en Excel, subir capturas a Confluence).

4. Registra los defectos

- Crea **Bug tickets** para los 3 defectos que documentaste en la Clase 6.
 - Asegúrate de completar: Summary, Description, Pasos para reproducir, Resultado esperado/obtenido, Severity, Priority, Attachments (capturas).

5. Vincula pruebas y requisitos

- En cada Story o Task, añade un link (“relates to” o “tests”) hacia las Sub-tasks de Test Cases y Bugs correspondientes.

6. Estimaciones y Planning Poker

- Asigna **Story Points** a cada User Story:
 - Historias muy simples → 1 SP
 - Complejas → 3–5 SP
 - De integración o API → 8 SP
- Simula una sesión de Planning Poker (puede ser ficticia): deja un comentario en cada Story con la votación del equipo (1, 2, 3, 5, 8).

7. Crea un Sprint

- Desde el Backlog, haz **“Create sprint”**, arrastra:
 - Al menos 4 User Stories (con sus Tasks y Bugs)
- **Inicia el sprint** y muévelos entre columnas (To Do → In Progress → Done) a medida que “ejecutas” los Test Cases y cierras los Bugs.

8. Colaboración y asignaciones

- Asigna cada ticket (Story, Task, Bug, Sub-task) a un miembro ficticio del equipo (puedes usar nombres genéricos).
- Etiqueta (“labels”) cada ticket con el módulo: [storytelling](#), [requerimientos](#), [test-cases](#), [bug](#).

Al finalizar, tendrás en Jira un **backlog completo**, **sprints configurados**, **historias de usuario**, **tareas de prueba**, **defectos** y **estimaciones**, todo vinculado y listo para el seguimiento ágil de tu proyecto QA.

Próximos Pasos

A partir de la próxima clase vamos a **profundizar en la ejecución del ciclo completo de QA dentro de Jira**, aplicando los conceptos que hoy aprendiste.

¿Qué vas a hacer en la Clase 15?

- Conocer el flujo de trabajo completo de un ticket en un tablero Scrum (To Do, In Progress, Code Review, Testing, PM Review, Done).
- Aprender a mover y actualizar tareas en cada estado.
- Crear **test cases formales dentro de Jira** con el plugin Zephyr.
- Ejecutar esos test cases y registrar los resultados de forma estructurada.
- Gestionar el **ciclo de vida completo de un bug**, desde que se detecta hasta que se cierra.
- Generar reportes de pruebas y evaluar si se cumplió con la Definition of Done.

Todo esto lo vas a aplicar en un entorno simulado de proyecto real con Talento Lab, utilizando los tickets que creaste esta semana. A partir de ahora, no solo vas a planificar, sino también **testear activamente**, **validar**, **reportar** y **cerrar historias funcionales** como lo harías en un equipo de QA profesional.



Buenos Aires
aprende
Agencia de Habilidades para el Futuro

BA Buenos
Aires
Ciudad