

Bienvenidos al Curso De Capacitación Laboral

Tester de Aplicaciones



Soy el profesor Matias Ruiz

*Les doy la bienvenida al Centro de Formación Profesional
N°31*

¡¡¡Comencemos!!!

Unidad 1

¿Por qué hacer este curso?

Esta es una guía completa para ser un ingeniero de pruebas de software. Después de terminar este curso, tendrá un buen conocimiento de las pruebas de software y los procesos de software.

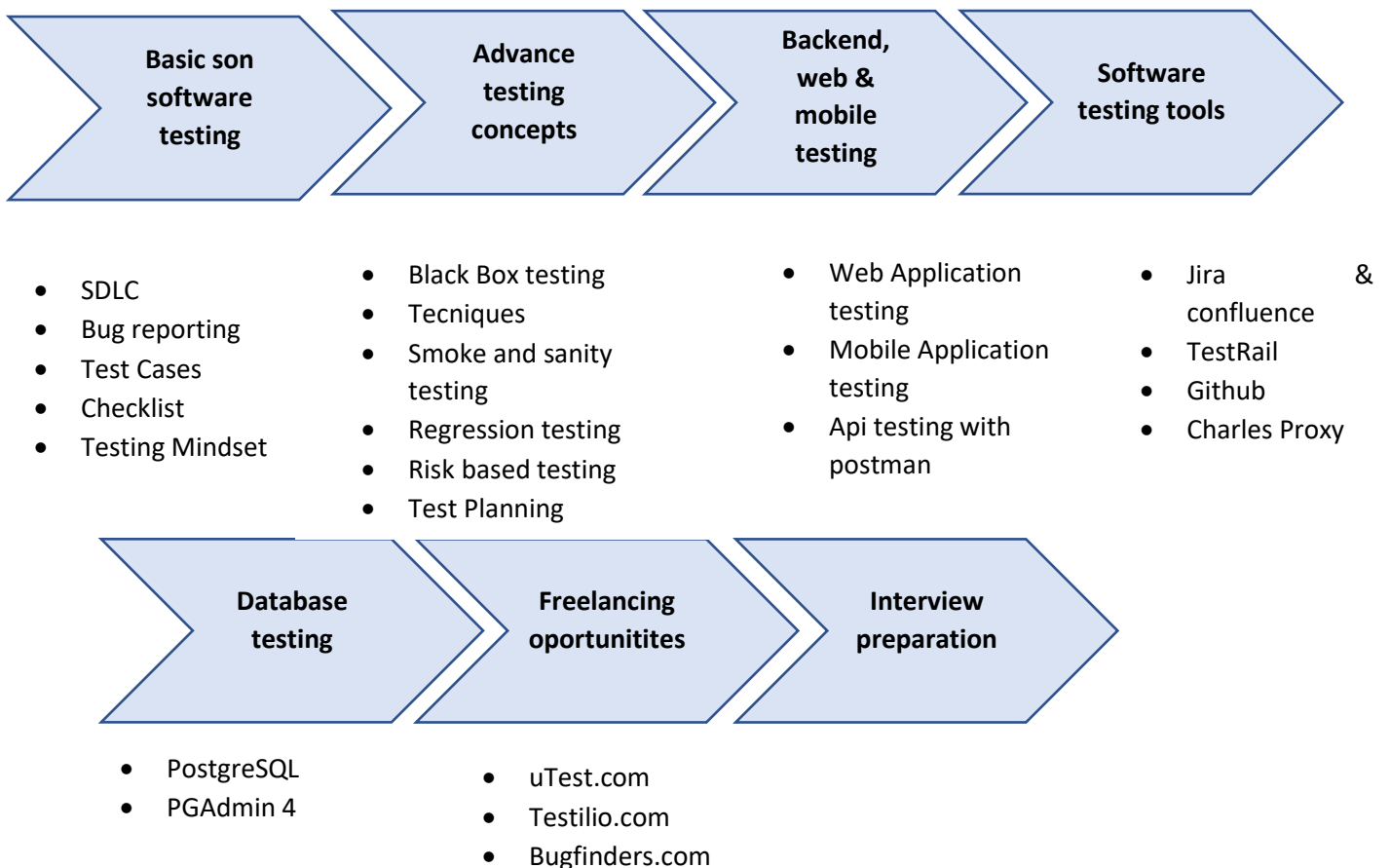
En este curso, compartiré contigo mis experiencias de la vida real. En otros cursos, hay mucha teoría innecesaria que no tiene valor en la vida real.

Compartiré los temas y conocimientos más importantes contigo.

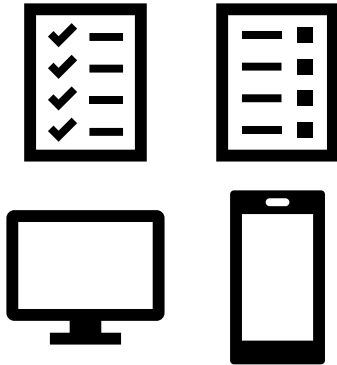
No olvides “empresas que buscan las habilidades adecuadas, el título universitario ya no es importante” Elon Musk

Naval Ravikant, cofundador de AngelList, ofreció tres consejos:

- Qué estudiar y cómo estudiarlo es más importante que dónde estudiar y por cuánto tiempo.
- Los mejores profesores están en Internet. Los mejores libros están en Internet. Los mejores compañeros están en Internet.
- Las herramientas para aprender son abundantes. Es el deseo de aprender lo que escasea.



Descripción general de las pruebas de software



- La prueba de software es una investigación realizada para proporcionar a las partes interesadas información sobre la calidad del producto o servicio de software que se está probando...
- En otras palabras, las pruebas de software son una actividad que implica controlar la calidad del software y asegurarse de que coincida con los requisitos del software y haga feliz a nuestro cliente.

¿Cuáles son los roles en la industria de TI?

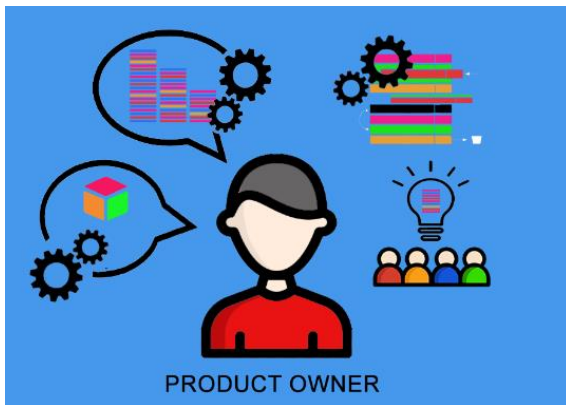
En las empresas de tecnología, existen diferentes roles que colaboran y se enfocan en entregar software de alta calidad. Cuando comienza su carrera, es importante comprender los diferentes roles y responsabilidades. Después de este curso, tendremos una sólida comprensión de las pruebas de software y su papel en el ecosistema de software.



Ingeniero de software: también conocido como programador de aplicaciones, el trabajo de un ingeniero de software generalmente incluye diseñar y programar software a nivel de sistema: sistemas operativos, sistemas de bases de datos, sistemas integrados, etc.



Analista de negocios: Son verdaderos mediocampistas, igualmente felices hablando con gente de tecnología, gerentes de negocios y usuarios finales. Identifican oportunidades de mejora en los procesos y operaciones comerciales utilizando tecnología de la información. El rol se basa en proyectos y comienza con el análisis de las necesidades de un cliente, recopilando y documentando los requisitos y creando un plan de proyecto para diseñar la solución tecnológica resultante. Los analistas de negocios necesitan conocimientos de tecnología, pero no necesariamente necesitan un título técnico.



Product Owner: El propietario del producto suele ser un usuario principal del sistema o alguien de marketing, gestión de productos o cualquier persona con un conocimiento sólido de los usuarios, el mercado, la competencia y las tendencias futuras para el dominio o tipo de sistema que se está desarrollando.



Project Manager: Organice a las personas, el tiempo y los recursos para asegurarse de que los proyectos de tecnología de la información cumplan con los requisitos establecidos y se completen dentro del tiempo y el presupuesto. Pueden administrar un proyecto completo de principio a fin o administrar parte de un "programa" más grande. No es un rol de nivel de entrada: los gerentes de proyecto deben estar bastante informados. Esto requiere experiencia y una buena base de tecnología y habilidades blandas, que son esenciales para trabajar con equipos de desarrollo tecnológico y gerentes comerciales de nivel superior.



Software Tester: Los errores pueden tener un impacto enorme en la productividad y la reputación de una empresa de TI. Los evaluadores intentan anticipar todas las formas en que se puede usar una aplicación o sistema y cómo podría fallar. Los evaluadores preparan scripts de prueba y analizan los resultados, que se envían al líder del proyecto para que se puedan realizar las correcciones. Los probadores también pueden participar en las primeras etapas de los proyectos para anticipar las dificultades antes de que comience el trabajo. Potencialmente, puede obtener un alto nivel como tester.

¿Quién es responsable de las pruebas?

“Todo el equipo” responsable de la Calidad del software. El objetivo principal del té es entregar un producto de alta calidad, de manera eficiente y en el menor tiempo posible. Todos los miembros del equipo deben centrarse en la calidad con la guía del probador de software. El probador de software debe liderar al equipo en términos de calidad y asegurarse de que el producto satisfaga las necesidades del cliente.

¿Por qué las pruebas de software son importantes?

En la industria de TI, las pruebas de software se subestimaron y, como consecuencia, muchas empresas tuvieron muchos problemas importantes que les causaron millones de daños.

Ejemplo de live issues:

Error de \$440 millones de los Knights

- Uno de los creadores de mercado de acciones más grandes de Estados Unidos luchó por mantenerse a flote después de que un error de software provocó una pérdida de \$ 440 millones en solo 30 minutos. Las acciones de la empresa perdieron un 75 por ciento en dos días después de que el software defectuoso inundara el mercado con intercambios no deseados.

NASA – El orbitador climático de Marte

- La nave espacial de \$ 125 millones presumiblemente orbitó demasiado cerca de la superficie de Marte y se desintegró porque el equipo de ingeniería no pudo convertir sus medidas de unidades estadounidenses a métricas.

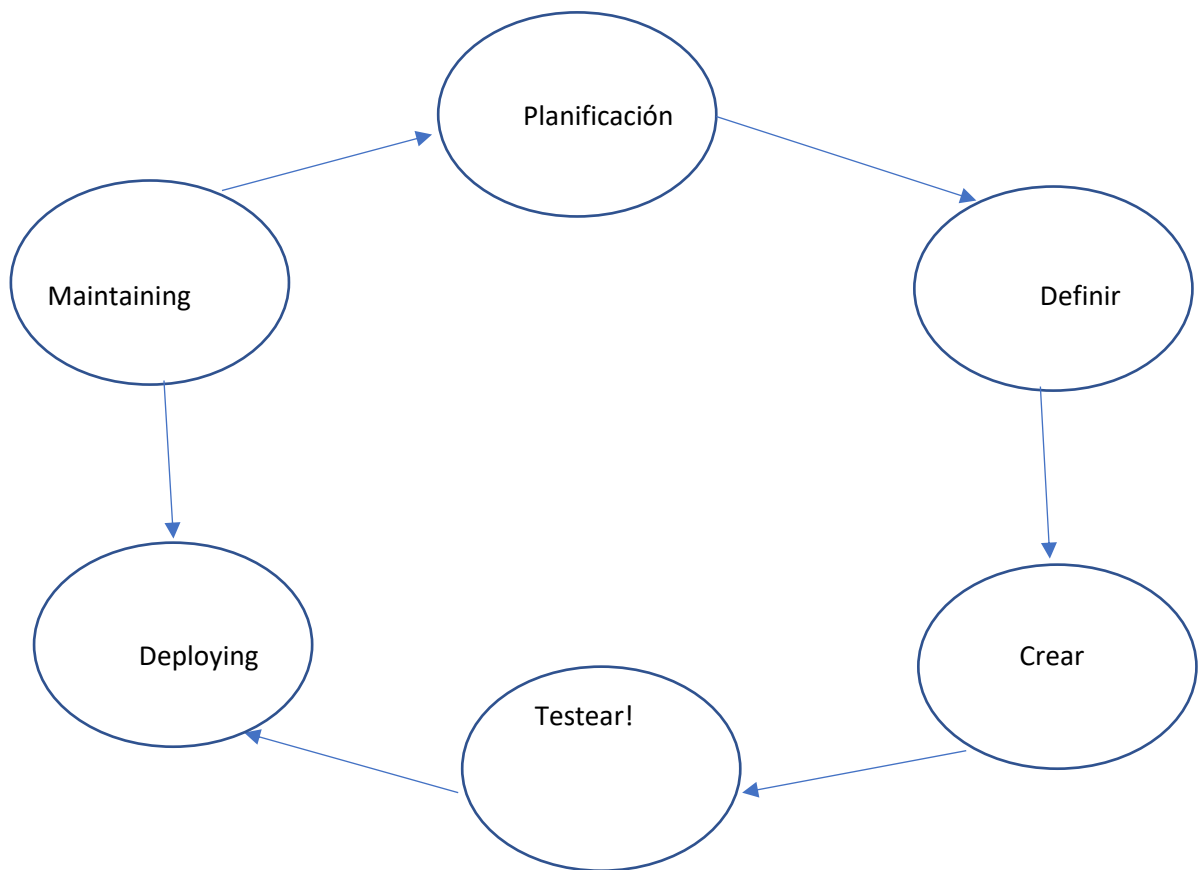
Una de las principales responsabilidades del probador de software es encontrar defectos lo antes posible (lo antes posible). Imagina que encuentras un error durante la fase de prueba:

Costo: el costo de los probadores que vuelven a probar las correcciones de errores/encontrar errores

Pero si encuentra un problema con el producto final, el costo será:

El costo de que los clientes encuentren errores, más la mesa de ayuda, más escaladas, más correcciones de errores, más nuevas pruebas, más reimplementación.

- ¿Qué es el ciclo de vida del desarrollo de software?
- Ciclo de vida de desarrollo de software (SDLC) es una terminología utilizada para explicar cómo se entrega el software a un cliente en una serie de pasos.
- Este proceso nos ayuda a producir software con la más alta calidad y el menor costo en el menor tiempo.



Plan: El equipo responde a la pregunta “¿Qué queremos?”

Definir/Análisis: El equipo responde a la pregunta "¿Cómo obtendremos lo que queremos?"

Las cuatro actividades principales involucradas en la fase de análisis son las siguientes:

1. Recopilación de requisitos del negocio (Cliente).
2. Creación de diagramas de proceso
3. Realizar un análisis detallado
4. Diseño de la infraestructura de TI

Construir: Es el momento del “Vamos a crear lo que queremos”.

Prueba/ Testing: Explica "¿Conseguimos lo que queríamos?"

Las dos actividades principales involucradas en el pase de desarrollo son las siguientes:

1. Escritura de casos de prueba.
2. Ejecución de casos de prueba

Despliegue: “Empecemos a usar lo que tenemos”.

Mantenimiento: “Acerquemos esto a lo que queremos”.



En el mundo de IT, el software entregado es igual al mundo de la "comida". Al final de las actividades del ciclo de vida del desarrollo de software, puede entregar el producto final a los clientes. Cada parte debe ser perfecta para recibir un gran producto final.

¿Cuáles son las actividades de prueba?

Las pruebas participan en cada parte del ciclo de vida del desarrollo de software, desde la planificación hasta la implementación.

- Planificación
 - Especificación
 - Ejecución
 - Finalización de la prueba
1. **1. Planificación de pruebas:** antes de iniciar los controles, el probador debe planificar qué y cómo probar el software. El plan de prueba es un documento que describe el alcance y las actividades de las pruebas de software.
 2. **2. Especificación de la prueba:** contiene dos componentes principales:
 3. °Diseñar casos de prueba: determinar “¿cómo! Se van a ejercer las condiciones de prueba identificadas.
 4. °Construir casos de prueba – implementación de los casos de prueba. E,G: Test scripts, test data’s...
 5. **Ejecución de la prueba:**
 - °Test Engineer ejecuta las pruebas y se asegura de que la aplicación coincida con los "resultados esperados".
 - °Los casos de prueba más importantes deben priorizarse y ejecutarse primero.
 6. **Finalización de la prueba:**

° Todas las actividades de prueba completadas con éxito

° Todos los errores de "alto" nivel resueltos

° Producto listo para entregar

- Informe de errores en las pruebas de software

- El error es un problema relacionado con el software.

- Los errores son resultados naturales de cualquier proyecto de software.

Tipos de bugs

- **Errores visuales:** imágenes truncadas, problemas de alineación
- **Errores funcionales:** el usuario no puede pasar a la página siguiente
- **Problemas de rendimiento:** se tarda 10 segundos en abrir la página web

Cómo escribir un informe de error

- • Cuando encuentre un problema con el software, ¡es hora de corregir un error!
- Un informe de errores debe ser "auto explicativo" y lo más simple posible.

Marker / MAR-131

[Pricing] - Update the price to \$29 ← **Title / Summary**

Edit Comment Assign To Do In Progress Workflow Admin

Details

Type: Bug Status: **TO DO** (View workflow)
Priority: High Resolution: Unresolved
Labels: None
Environment: — Browser Chrome 54.0.2840.71 Screen Size 1920 x 1200 Viewport Size 1607 x 920 Zoom L...

Description ←

Summary:
The price mentioned on the pricing page is not correct

Steps to Reproduce: ←
Go to the pricing page

Expected Results
The price for the basic plan should ne \$29 ←

Actual Results:
The price for the basic plan is currently \$25

Source URL: <https://www.shopify.com/pricing>

Attachments ←

Drop files to attach, or browse.

People

Assignee: gary
Assign to me

Reporter: Christophe Han

Votes: 0

Watchers: 1 Stop watching th

Dates

Created: 1 minute ago
Updated: 1 minute ago

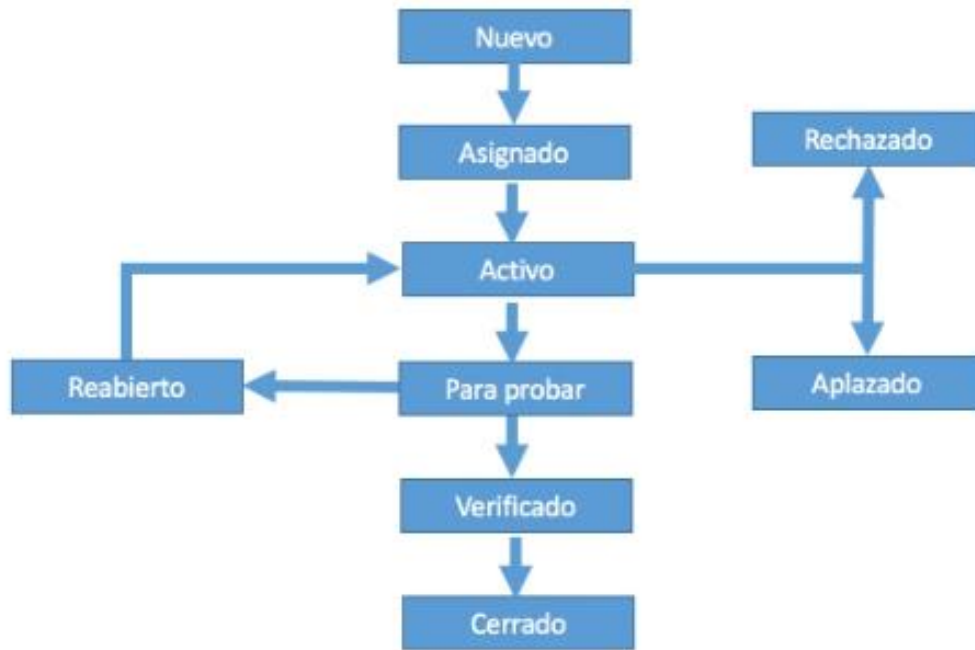
Agile

View on Board

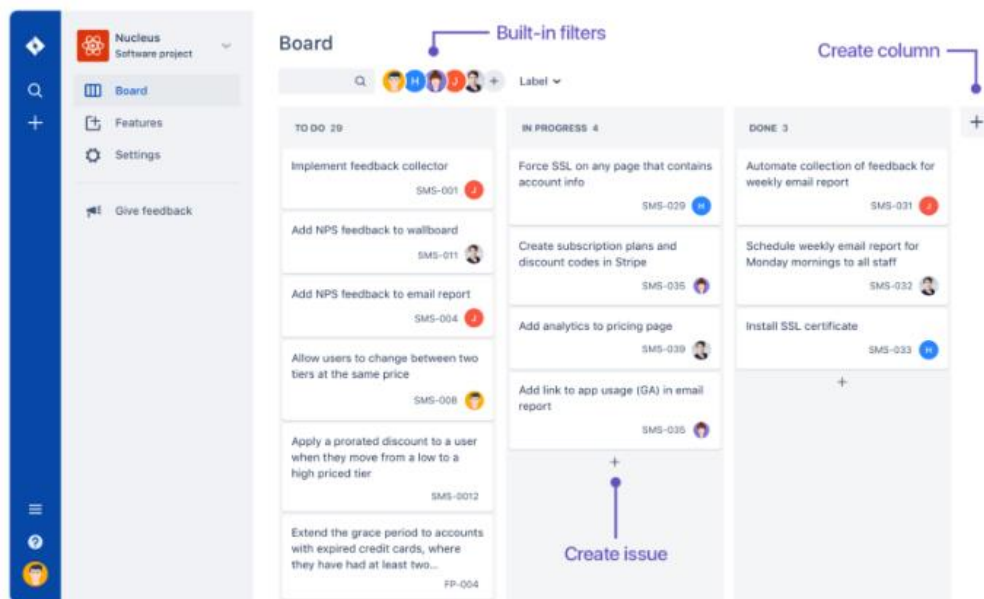
HipChat discussions

Do you want to discuss this issue? Connect

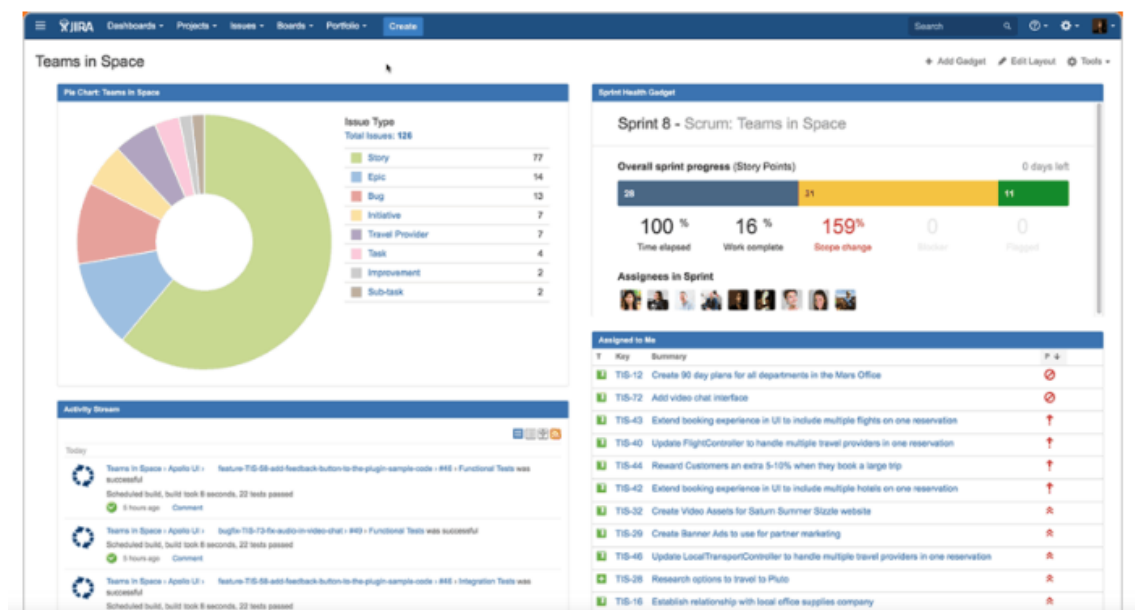
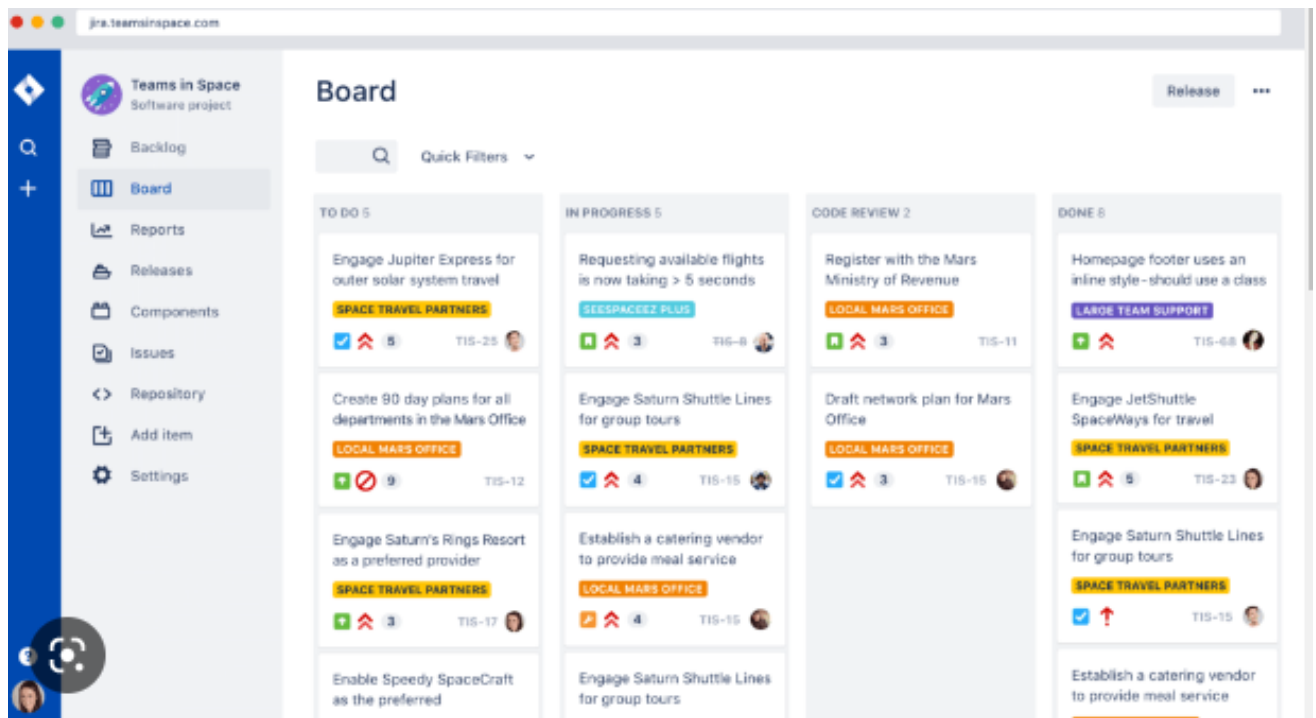
Connect Dismiss



Ciclo de un bug
Ejemplo de Jira



<https://www.atlassian.com/blog/jira-software/preview-new-agility-boards-jira-software>



Github

<https://github.com/kaishuu0123/rebacklogs>

master 39 branches 19 tags

Go to file Code

About

kaishuu0123 Merge pull request #82 from kaishuu0123/dependabot/bundler/pu...	✓ b6fccb9 on Jul 6, 2020	138 commits
app	refs RE-87 RE-88: Add pager for User/Group Management page	3 years ago
bin	first commit (v0.1)	4 years ago
config	Fix routing for projects.	4 years ago
db	refs RE-2: User registration and user authentication are possible wit...	4 years ago
docker-volumes	first commit (v0.1)	4 years ago
images_for_README	add images for README	4 years ago
lib	first commit (v0.1)	4 years ago

Re:Backlogs is an Open Source Project Management Tool.

rebacklogs.saino.me/

project-planning agile bug-tracker
issue-tracker self-hosted
project-management kanban boards
task-management rebacklogs

Readme
MIT license
166 stars
11 watching

Re:Backlogs Settings admin

R

Backlogs

Kanban

Closed Sprints

Settings

>

Sprint-1

No period specified

0.0

REEN-10 Can Create Public Project in Create Project Page 0.0

REEN-11 Deploy to Heroku 0.0

Sprint-2

No period specified

0.0

REEN-12 Add Link to Docs pages from Re:Backlogs 0.0

REEN-13 Can Add Tags to Ticket 0.0

Backlogs

Add Story Add Sprint

REEN-1 Can View Ticket History 0.0

REEN-2 Can File upload for Tickets 0.0

REEN-3 Can direct access to ticket URL 0.0

REEN-4 Can copy Ticket Title to clipboard 0.0

REEN-5 Can OAuth Login & Registration 0.0

REEN-6 Can Restrict User Registration 0.0

REEN-7 [NEED SPEC] Can Ticket multiselect 0.0

REEN-8 [NEED SPEC] Bulk Operation for Ticket 0.0

REEN-9 Can support pager for Project index 0.0