



ACERCA DEL AUTOR

ALEJANDRO JUAN CANOSA FERREIRO

Ingeniero técnico en informática de gestión por la Universidad Nacional de Educación a Distancia. Postgrado en Software Quality Assurance por la Universidad Politécnica de Cataluña. Certificado en Scrum Foundation Professional Certification (SFPC), DevOps Essentials Professional Certification (DEPC), Expert Level Certification en Katalon y a punto de certificarse en ISTQB , Professional Scrum Master e ITIL.

Consultor de QA con más de 10 años de experiencia en el mundo de calidad de software entre España y Colombia y experto en automatización de pruebas de software con herramientas como Selenium, TestComplete, Katalon o Rational Functional Tester de IBM.

1

HERRAMIENTAS DE QA

Software Tester



Una herramienta de prueba es un instrumento automatizado que soporta una o más actividades de prueba tales como planificación, control, especificación y ejecución.

Estas herramientas se pueden utilizar en todas las pruebas y ciclos de vida para ayudar a mejorar la eficiencia en los todos los procesos **QA**.

1.1 TIPOS DE HERRAMIENTAS DE PRUEBA

Hay 12 tipos de herramientas de pruebas.

- ✔ *Herramientas de análisis de cobertura de código.*
- ✔ *Herramientas de pruebas basadas en modelo.*
- ✔ *Herramientas de análisis de código fuente.*
- ✔ *Herramientas de pruebas unitarias.*
- ✔ *Herramientas de gestión de incidentes.*
- ✔ *Herramientas de ejecución de pruebas.*
- ✔ *Herramientas de pruebas de rendimiento.*
- ✔ *Herramientas de monitorización de pruebas.*
- ✔ *Herramientas de pruebas de seguridad.*
- ✔ *Herramientas de gestión de pruebas de datos.*
- ✔ *Herramientas de virtualización de servicios.*
- ✔ *Herramientas de gestión de pruebas.*

1.1.1 Herramientas de análisis de cobertura de código

Mide el % de código que se ha probado generalmente de un método, suele basarse en el número de caminos que se han ejecutado del método.

Hay varias herramientas para crear el informe de cobertura y suele depender del lenguaje de programación en que está escrito el código.

La cobertura puede ser de un método, de una clase de un módulo.

Estas herramientas crean automáticamente informes de cobertura.

Las herramientas más conocidas de este tipo son:



Clover es una herramienta de análisis de cobertura de código que soporta los lenguajes de **Java** y **Groovy**.

Es compatible con los marcos de prueba **JUnit** y **TestNG**.

Se integra con los **IDE** (entornos de desarrollo integrado) **IntelliJ**, **IDEA** y **Eclipse**.

Cobertura

Cobertura es una herramienta de cobertura de código gratuita y que se suele utilizar a través del gestor de **plugins** de **Maven**; analiza la cobertura del código para proyectos con lenguaje de programación **Java**.

JACOBO

Java Code Coverage

Es una herramienta de cobertura de código que es una biblioteca gratuita para el lenguaje **Java** y se integra con el **IDE Eclipse**

*Hay otras herramientas menos conocidas para lenguajes menos utilizados como **Ruby** y **Python***

Pytest-cov

Es un complemento gratuito de **Python** que se utiliza para generar informes de cobertura de código.

Se instala utilizando **pip** que es el instalador de paquetes de **Python** y que se maneja desde la línea de comandos.

Simplecov

Es una herramienta de cobertura de código para el lenguaje de programación.

Utiliza la propia biblioteca de cobertura integrada en **Ruby** para recuperar datos relevantes que se utilizara para determinar la cobertura de código después de la ejecución de pruebas.

1.1.2 Herramientas de pruebas basadas en modelo

Estas herramientas lo que hacen es generar casos de prueba automáticamente a partir de modelos.

Los modelos pueden ser representados en elementos como imágenes, gráficos, diagramas u otros tipos de elementos.

Los requisitos se tienen que representar de una manera que la herramienta los entienda.

Los casos de prueba se generan automáticamente a través de la representación abstracta de los requisitos hechos por la herramienta.

Algunas herramientas además de crear los casos de prueba automáticamente pueden ejecutar los casos de prueba.

Abajo tenemos 3 de las herramientas más utilizadas en el mercado para estas actividades.



Es una herramienta que permite automatizar las pruebas de un extremo a otro; permite diseñar casos de prueba, automatizarlos mediante **scripts**, generar y diseñar datos para las pruebas, analizar las pruebas y generar distintos informes de las pruebas.



Te permite crear casos de prueba ejecutables optimizados a través de casos de diagramas de flujo utilizando inteligencia artificial y se integra con herramientas como **Jira** o herramientas de **IBM** o **Microsoft**.



Genera casos de prueba y planes de prueba a partir de historias de usuario y criterios de aceptación y permite crear **dashboard** de QA personalizados.

Esta herramienta también permite generar documentación y métricas.

1.1.3 Herramientas de análisis de código fuente

Estas herramientas analizan el código fuente antes de ejecutarlo y lo comparan con reglas de codificación para generar informes y poder mejorar la calidad del código.

Estas herramientas se están volviendo imprescindibles ya que permiten encontrar defectos y vulnerabilidades en una fase temprana del código.

Estas herramientas analizan el código desde un repositorio y las reglas de análisis de código pueden personalizarse según proyecto u organización.

Estas herramientas crean informes automáticos y clasifican la calidad de nuestro código.

Las herramientas más importantes serían:



Es una herramienta que analiza la calidad de un código desde repositorio o desde **IDE** y tiene versión de pago y gratuita y versión en la nube que se llama **SonarCloud**.



Es una herramienta de análisis de código que se centra en la seguridad del código encontrado vulnerabilidades de manera temprana. Es de pago.

VERACODE

Es una herramienta que también se centra en la seguridad del código encontrando vulnerabilidades, es capaz de analizar lenguajes como **Java**, **JavaScript**, **VB.NET**.

Esta herramienta permite realizar análisis estáticos del código fuente, pero también análisis dinámicos donde se analiza la aplicación en vivo desde su URL.

1.1.4 Herramientas de pruebas unitarias

Se prueba que la unidad de componente más pequeña de un software funciona según lo diseñado; en general suele ser un método que forma parte de una clase.

A esta prueba o actividad se le llama prueba unitaria y se necesita un **framework** para poder realizarla.

Estas herramientas o **framework** están habitualmente integradas con el IDE o desarrollo de entorno integrado.

El desarrollador o analista de calidad codifica la prueba que se ejecuta contra el código de la aplicación.

Las herramientas más conocidas serían:



Es la última versión del framework para la automatización de pruebas unitarias en **Java**. Se apoya mucho en las últimas novedades de **Java 8**.



Es un marco de pruebas unitarios para todos los lenguajes **.NET**. La última versión soporta muchos de los lenguajes de **.NET**.



Es un **framework** de código abierto para **Java** que se utiliza para simular la llamada a un método externo en pruebas unitarias. Se utiliza en la metodología **TDD** (desarrollo basado en pruebas) y **BDD** (desarrollo basado en comportamiento).



Es un **framework** de pruebas unitarias que está basado en **JUnit** y **Nunit** y que al igual que **JUnit** trabaja con Java. Se puede utilizar con **Eclipse** y **Maven** utilizando un **plugin**.

1.1.5 Herramientas de gestión de incidentes

Los defectos o incidentes se suelen gestionar en una herramienta de gestión de pruebas.

Estas herramientas de gestión de pruebas suelen integrarse con herramientas de gestión de proyectos.

Estas herramientas suelen tener notificaciones por correo electrónico, notificaciones a chat como **Teams**.

Las herramientas más conocidas son:



Fue la primera herramienta de gestión de incidentes; es de código abierto, gratuita y está como aplicación web además se puede personalizar.

Permite hacer un seguimiento de las incidencias y defectos de un proyecto y tiene un control de visibilidad de errores.



Es un sistema de seguimiento de defectos, que es gratis, es una aplicación web y se puede personalizar; también tiene control de visibilidad de errores.



Es una aplicación web de pago que permite gestión de pruebas, requisitos y seguimiento de errores; también genera informes visuales y se integra con **Jira**.

XRAY

Es un **plugin** de **Jira** que permite gestionar las pruebas en un proyecto de **Jira** y crear informes para seguir el resultado de las pruebas y llevar una cobertura de los requerimientos de un proyecto.

1.1.6 Herramientas de ejecución de pruebas

Las herramientas de ejecución de pruebas permiten la creación de scripts de prueba automatizados, es decir, permiten automatizar la ejecución de los casos de pruebas.

Estas pruebas se utilizan para automatizar las pruebas funcionales.

Los scripts se pueden crear grabando mediante la codificación utilizando algún lenguaje de script.

Estas herramientas permiten reproducir los pasos de un caso de prueba de manera automatizada y se integran con otras herramientas como **Jira** o **Testlink**.

Las herramientas más populares de este tipo son:



Es el framework más utilizado para automatizar pruebas sobre aplicaciones web, aunque le está comiendo mucho terreno **Cypress**; es abierto, gratuito se puede integrar

con **Jenkins**, **GitLab**, **Bamboo** y con **Appium** para automatizar app's para móviles; soporta los navegadores más importantes como son Chrome, Firefox, Safari y Edge.



Es un framework que tiene versión gratis y versión de pago y que permite automatizar pruebas funcionales en aplicaciones web; se basa en **Javascript** soporta Chrome, Opera, Edge, pero no Firefox y Safari; es más rápido que **Selenium** y permite realizar una depuración profunda.

UTF

Es una herramienta de pago de **HP**, se puede utilizar en Windows y utiliza el lenguaje **VB** para escribir scripts de casos de prueba automatizados; te permite automatizar casos de prueba en aplicaciones web y aplicaciones de móviles; para integrarse con **ALM** requiere una licencia extra.

Test Complete

Es de pago y automatiza pruebas funcionales sobre aplicaciones web, de escritorio y aplicaciones móviles. Se integra con muchas herramientas populares como **Jenkins**, **Baboo**, **Jira**, **Azure**, **GitHub**, **Bitbucket** y muchas más. Puede escribir los scripts de prueba en distintos lenguajes de script y puede utilizarse en **Linux**, **Windows** y **MAC**.

Ranorex Studio

Utiliza la plataforma de **.NET** para crear sus scripts de prueba; permite automatizar las pruebas funcionales de aplicaciones web, de escritorio y móviles. Se integra con herramientas como **Jira** y **Jenkins**.

Rational Functional Tester

Fue la primera herramienta de este tipo y fue creada por **IBM**. Permite automatizar las pruebas manuales de aplicaciones web, de escritorio y móvil. Puede crear los scripts en **Java** o en algún lenguaje de **.NET** como **C#**. La herramienta se puede instalar en **Windows**, **Linux** o **MAC**.

Es una herramienta que es muy cara para las funciones que ofrece.

Katalon Studio

Es una herramienta que tiene versión gratuita y de pago, la de pago tiene muchas más características; permite automatizar pruebas funcionales en aplicaciones web, de

escritorio y aplicaciones móviles e incluso también automatizar pruebas sobre **API's**. El lenguaje para crear los scripts es **Groovy**. Se puede instalar en **Windows**, **Linux** y **MAC**.

Permite realizar integración continua con su solución **Katalon Runtime Engine**.

Tiene una plataforma para realizar pruebas sobre distintos navegadores en distintos sistemas operativos de distintos dispositivos.

También se integra con todo tipo de herramientas como **Jira**, **Bitbucket**, **Jenkins**, **Bamboo**, **Github**, **Slack**, etc.

También hay que comentar que permite realizar pruebas de rendimiento utilizando la plataforma **OctoPerf** y podría realizar pruebas de seguridad con herramientas como **OWASP ZAP**.

Por último comentar que tiene un montón de **plugins** con código hecho para acceder a una hoja **Excel** o para leer correos de **Gmail**.

Con respecto a su precio oscila entre 1600 y 3000 euros dependiendo si coges la versión Enterprise y acceso a la plataforma **TestCloud** que permite también guardar miles de informes de ejecución, un precio anual por las 2 licencias.

1.1.7 Herramientas de pruebas de rendimiento

Estas herramientas son capaces de crear y ejecutar escenarios con distintos nº de usuarios virtuales o procesos que actúan como usuarios para realizar pruebas de carga, de stress u otros tipos de pruebas de rendimiento y generar informes; no se puede realizar pruebas de rendimiento sin herramientas.

Se necesita grabar al menos un escenario para reproducirlo con distintos usuarios virtuales; cada usuario virtual podía tener un ancho de banda, localización, navegador y sistema operativo diferente; hay herramientas que permiten configurar esos ambientes.

Cada usuario virtual sería como un proceso en que se repite el escenario, pero simulando diferentes ambientes como los que hable anteriormente de localización, ancho de banda, navegadores, etc.

Las herramientas de rendimiento más populares son:



Es una herramienta ideal para realizar pruebas de rendimiento de aplicaciones web; permite el diseño y automatización de pruebas funcionales para crear escenarios que luego son repetidos por los **usuarios virtuales** (procesos). Es una herramienta gratuita y open source con mucha documentación y cursos online.



Load Runner es una herramienta para realizar pruebas de rendimiento, pertenece a **MicroFocus** y antes a **HP**; es multiprotocolo y los scripts se pueden crear en **C** y **Java**; es gratis hasta 50 usuarios a partir de ahí hay que pagar; esta herramienta está formada por varios componentes, es lenta de arrancar, pero es de las mejores en pruebas de rendimiento.

Si necesitas pruebas con miles de usuarios y el cliente tiene dinero para gastar es tu herramienta.

LoadUI

Es una herramienta para realizar pruebas de rendimiento que tiene versión gratuita **Open Source** y de pago; esta herramienta se integra con **SoapUI** que es una herramienta para realizar pruebas sobre servicios web que utilizan **SOAP** o para hacer pruebas de **API** que utilicen **REST**.

1.1.8 Herramientas de monitorización de pruebas

Las herramientas de monitoreo se utilizan para realizar un seguimiento continuo del estado del sistema para avisar de fallos o problemas y solucionarlos antes de que se caiga el sistema.

Existen herramientas de monitoreo de redes, servidores, base de datos, seguridad, rendimiento, sitios web, aplicaciones, básicamente pueden seguir cualquier dispositivo, aplicación o característica.

Estas herramientas se configuran con distintos **KPI's** y sus umbrales y las partes en las que se va a realizar la medición, una vez se hace esto el usuario puede seguir la medición en vivo.

Estas herramientas son esenciales en entornos DevOps y algunas de las más populares son las siguientes:



Es un software de monitorización de rendimiento que permite optimizar el rendimiento de las aplicaciones; detecta de manera automática las cargas de trabajo y los

microservicios que se ejecutan dentro de los contenedores **Kubernetes** y analiza cómo funcionan y se comunican entre sí para detectar microservicios que no estén funcionando bien.



Es una aplicación de supervisión y análisis de aplicaciones a gran escala; supervisa la infraestructura, el rendimiento de las aplicaciones, la administración de los registros y la experiencia del usuario en tiempo real; de las mejores herramientas.

1.1.9 Herramientas de pruebas de seguridad

Estas herramientas buscan vulnerabilidades, amenazas, riesgos que puedan causar grandes pérdidas o que impacten a la marca de la empresa. Estas herramientas buscan **vulnerabilidades** y también pueden poner a prueba el sistema o la aplicación simulando un ataque; se podría escribir un libro sobre seguridad informática quizás en la segunda parte de este libro si la hay se pueda tratar más temas como este o pruebas de **API**, pruebas exploratorias, priorización de requerimientos, automatización de pruebas, pruebas basadas en riesgos, simulación de servicios o **QA sourcing** o pruebas de usabilidad o **IA** en QA.

Algunas herramientas para realizar pruebas de seguridad serían:

✔ **Wireshark**

Analiza los protocolos que hay en la red para ver si se puede capturar datos sensibles al utilizar protocolos no encriptados por ejemplo o con encriptamientos débiles.

✔ **Nmap**

Analiza los servicios que corren en un sistema.

✔ **Kali Linux**

Es un sistema operativo para realizar pruebas de penetración.

✔ **Nessus**

Analiza las vulnerabilidades de una aplicación, servidor, protocolo o sistema operativo.

✔ **Burp Suite**

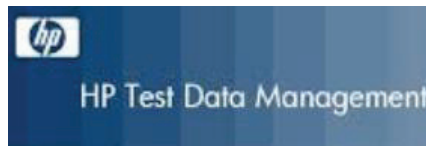
Analiza el tráfico de la aplicación.

1.1.10 Herramientas de gestión de datos

TDM es la herramienta de gestión de datos de prueba para la administración de los datos necesarios para realizar pruebas automatizadas y manuales y debería garantizar la calidad de sus datos además de su disponibilidad.

La ley **GDPR** ha obligado a tener una política de gestión de datos.

Dos herramientas populares para la gestión de datos son:



Hp Test Data Management es una herramienta que proporciona datos estructurados de calidad en cantidades adecuadas y de calidad.



Es una herramienta que crea datos de prueba realistas utilizando conexiones a **API's**.

1.1.11 Herramientas de virtualización de servicios

Son herramientas que emulan el comportamiento del software para eliminar el riesgo de entornos no disponibles o que aún no existen.

La virtualización de servicios es útil cuando necesitamos probar componentes que no están disponibles o que son caros de obtener como sería API's aún no desarrolladas.

Estas herramientas son capaces de crear una imagen del componente, que acepte las mismas entradas y devuelva las mismas salidas, pero sin la lógica del componente y conectarla al ambiente a probar y luego sustituirla por el componente ya terminado y operativo.

Algunas herramientas de este tipo serían:

Rational Test Virtualization Server

Es una plataforma de pruebas continuas basadas en la web que permite ejecutar distintas pruebas en la nube como pueden ser pruebas de API, funcionales y de rendimiento.

Realmente cogen servidores que están en la nube e instalan aplicaciones para poder realizar todo tipo de pruebas.

Es una herramienta del paquete **Rational** de **IBM** que es de pago.



Es una plataforma que es de pago y que permite hacer las siguientes actividades:

- ✔ *Automatizar pruebas API.*
- ✔ *Virtualizar servicios.*
- ✔ *Pruebas unitarias y cobertura de código.*
- ✔ *Análisis de código estático.*
- ✔ *Automatización de pruebas de rendimiento.*
- ✔ *Automatización de pruebas funcionales.*

Al parecer no tienes pruebas de usabilidad ni de seguridad.

1.1.12 Herramientas de gestión de pruebas

Son herramientas que permiten realizar un plan de pruebas, diseño de casos de pruebas, ejecución de casos de prueba, seguimientos de pruebas, proyectos y creación de informes de todo tipo.

Herramientas gratuitas serían.



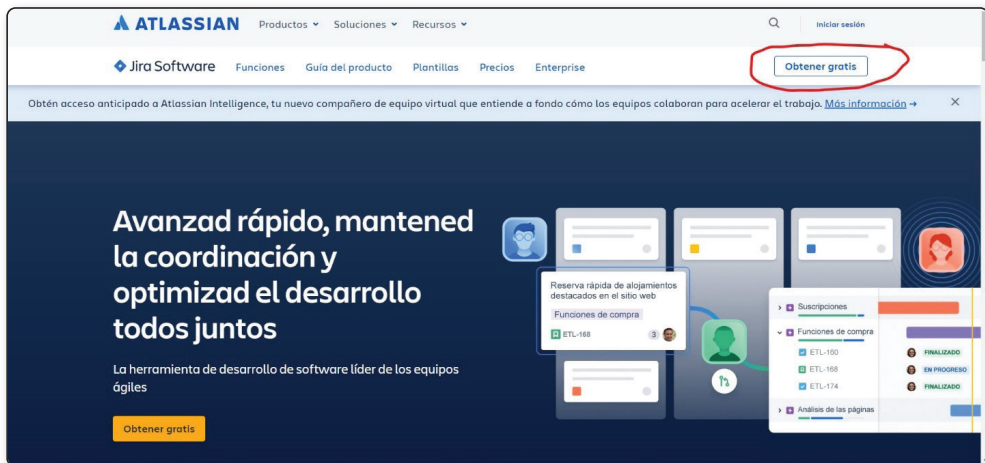
Herramientas de pago serían.



1.2 CASO PRÁCTICO. CREACIÓN DE UN PROYECTO DE SCRUM EN JIRA

En esta práctica vamos a seguir todos los pasos para tener un proyecto de Scrum en **Jira**.

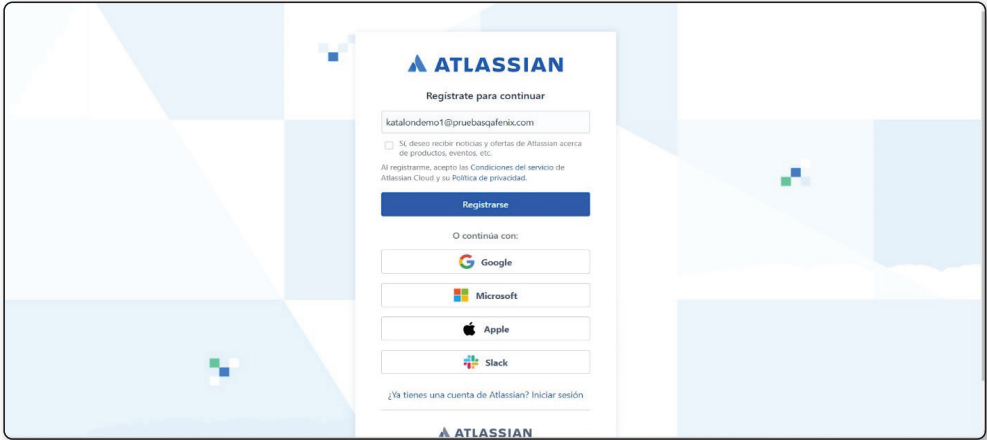
Buscamos en Google “Jira” y pulsamos en el primer resultado, accedemos a la página principal de Jira donde en la parte superior pulsamos Obtener gratis para crearnos una cuenta de Jira.



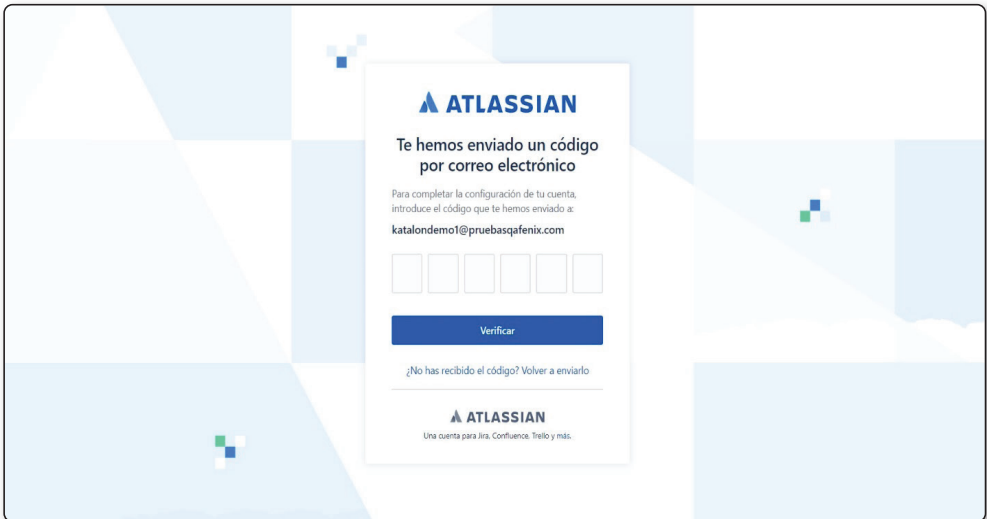
Tenemos que introducir en el campo Dirección de correo electrónico del trabajo un email que sea de una empresa o tener nuestro propio dominio, en Hostinger por 75 euros al año puedes tener una plataforma para crear páginas web con **WordPress** y por 7 euros al año comprar un dominio y crear un montón de correos.



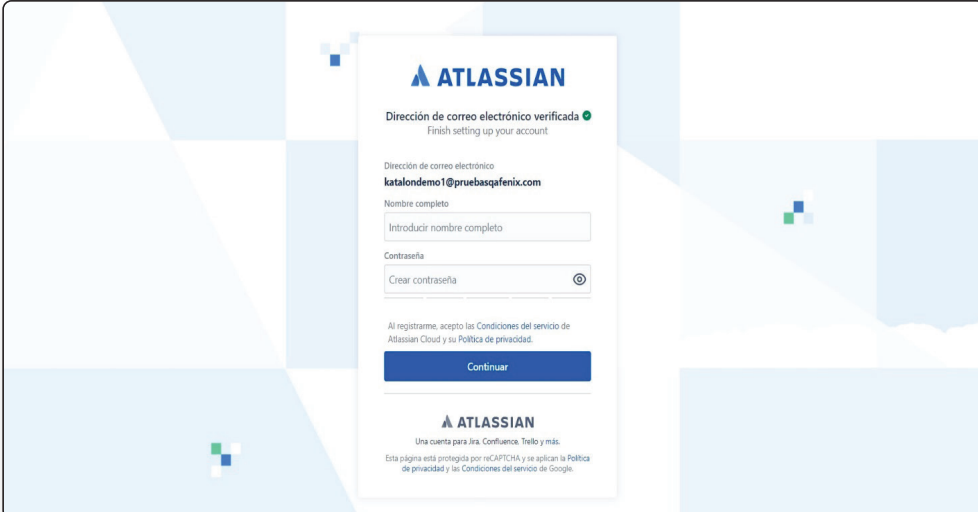
Una vez introducimos un correo de empresa pulsamos en Registrarse.



Jira nos envía un código al correo que hemos puesto, son 6 dígitos y debemos introducirlo en cada campo.



Introducimos un nombre que podía ser el nombre de la empresa en mi caso puse mi nombre completo y una contraseña segura, que podía ser 8 caracteres, con alguna letra en mayúsculas, minúsculas, número, letras y un carácter especial como _ y pulsamos en continuar.



Le ponemos un nombre al proyecto que crearas en Jira que en mi caso es tienda virtual y pulsamos en el botón empezar a trabajar con Jira.



Jira nos hará una encuesta para saber un poco de nosotros y personalizar nuestra experiencia, en mi caso pondré Desarrollo de software porque vamos a crear una tienda virtual.

Háblanos un poco de ti

Esto nos ayudará a personalizar tu experiencia.

¿En qué tipo de equipo trabajas?

Operaciones Legal Ventas

Recursos humanos

Servicio de atención al cliente

Desarrollo de software Soporte de TI

Marketing Finanzas

Otro

Saltar pregunta

Mantenemos protegida tu información de acuerdo con lo establecido en nuestra política de privacidad. [🔗](#)

Nuestra función será Ingeniero de control de calidad así que la marcamos y pulsamos en siguiente.

Háblanos un poco de ti

Esto nos ayudará a personalizar tu experiencia.

¿Cuál es tu función?

Desarrollador web Gestor de producto

Gestor de control de calidad

Gestor de proyecto

Ingeniero de control de calidad

Ingeniero de software Director de desarrollo

DevOps Diseñador

Otro

Saltar pregunta

Mantenemos protegida tu información de acuerdo con lo establecido en nuestra política de privacidad. [🔗](#)

Vamos a indicar que vamos a realizar todas las funciones que hay abajo porque realmente un **Team Leader** o gestor de proyecto haría todo eso y más y pulsamos en siguiente.

Háblanos un poco de ti

Esto nos ayudará a personalizar tu experiencia.

¿Cuáles son tus tareas principales? (Opción múltiple)

- Diseño y/o diagramas
- Documentación y gestión de archivos
- Seguimiento de productividad y generación de informes
- Proporcionar soporte y asistencia técnica
- Planificación y coordinación de proyectos
- Gestión de datos y seguridad
- Redacción de código y revisiones
- Pruebas y control de calidad
- Mejorar los flujos de trabajo y los procesos

Omitir pregunta **Siguiente**

Mantenemos protegida tu información de acuerdo con lo establecido en nuestra [política de privacidad](#).

Elegimos la plantilla de **Scrum** para nuestro proyecto y pulsamos en siguiente.

Selecciona una plantilla para tu primer proyecto

Si no sabes bien qué elegir, no te preocupes. Puedes crear rápidamente un nuevo proyecto si este no es el adecuado para ti.

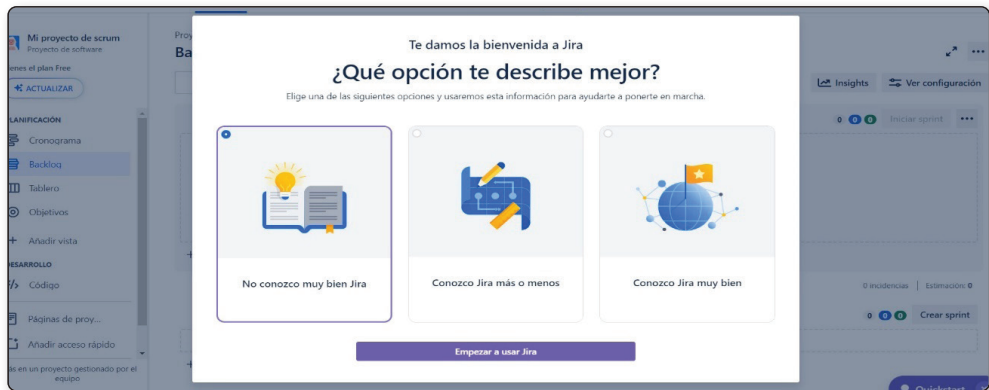
KANBAN
Un sencillo tablero para visualizar tu flujo de trabajo
Una forma flexible y eficiente de gestionar el trabajo ágil, con un tablero y un cronograma.
POPULAR

SCRUM
Entrega trabajos en bloques de tiempo cortos y repetibles
Planifica, prioriza y programa el trabajo mediante sprints ágiles con un backlog, un tablero y un cronograma.

NEGOCIO
Colaboración para equipos empresariales
Diseñada para ayudar a tu equipo empresarial a colaborar y comunicarse, con vistas de listas y calendarios, formularios y mucho más.

Siguiente

Seleccionamos no conozco muy bien **Jira** y pulsamos en Empezar a utilizar **Jira**, nos hará un mini tour.



Abajo he marcado en rojo las partes más importantes:

- Crear incidencia que puede ser historias de usuario, tareas, subtarear, para los bugs habría que instalar **XR**Ray para gestionar las pruebas.
- Crear épica es para historias de usuario muy grandes de varios Sprints.
- Backlog donde tendremos todas las historias de usuario que formaran parte de las distintas épicas por ejemplo los distintos módulos de una tienda virtual.
- Cuando tengamos las historias de este sprint agrupadas por épicas, que serán nuestros módulos de la tienda, se añaden al tablero del Sprint 1.
- Indicamos las fechas para la duración de cada Sprint, que pueden ser por ejemplo cada 2 semanas.
- Una vez hecho esto ya podemos crear el Sprint 1.
- Y después Iniciar Sprint que esta deshabilitado porque aún no creamos el Sprint.

