**UNIVERDAD TECNOLÓGICA DE NEZAHUALCÓYOTL**

**División de Tecnologías de la Información y Comunicación**

**Área Sistemas Informáticos**

**ASIGNATURA**

**GESTION DEL PROCESO**

**DE DESARROLLO DE SOFTWARE**

**DOCENTE**

**NORBERTO VERA GARCIA**

**ACTIVIDAD**

**PROYECTO SRS**

**ESTUDIANTES**

**DIEGO MARTIN GARRIDO FLORES**

**IVAN ROMERO MORA**

**JORGE MOISES NAVARRETE MORALES**

**MARTHA DIAZ ZUÑIGA**

**JOSÉ TRINIDAD MACIEL HERNÁNDEZ**

**TURNO**

**ITIC 1001-V**

Contenido

[INTRODUCCION 3](#_Toc109409128)

[DESCRIPCION DE LA EMPRESA 4](#_Toc109409129)

[EQUIPO DE TRABAJO 6](#_Toc109409130)

[EQUIPO DE TRABAJO DE LA EMPRESA 7](#_Toc109409131)

[PROBLEMATICA 8](#_Toc109409132)

[AFECTA A 8](#_Toc109409133)

[IMPACTA A LA ORGANIZACIÓN 8](#_Toc109409134)

[SOLUCIÓN TECNOLÓGICA 9](#_Toc109409135)

[OBJETIVO GENERAL 10](#_Toc109409136)

[OBJETIVOS ESPECIFICOS 10](#_Toc109409137)

[PROBLEM STATEMENT 11](#_Toc109409138)

[ARQUITECTURA 12](#_Toc109409139)

[PLATAFORMA DE EJECUCIÓN 13](#_Toc109409140)

[REQUERIMIENTOS FURPS 14](#_Toc109409141)

[FUNCIONALES 14](#_Toc109409142)

[NO FUNCIONALES 15](#_Toc109409143)

[REGLAS DEL NEGOCIO 17](#_Toc109409144)

[METODOLOGÍA DE DESARROLLO 21](#_Toc109409145)

[HERRAMIENTAS DE DESARROLLO 23](#_Toc109409146)

[DESCRIPCIÓN DE ROLES PARA EL SISTEMA 26](#_Toc109409147)

[DESCRIPCIÓN DE MENUS DEL SISTEMA 27](#_Toc109409148)

[PRINCIPALES NECESIDADES DE LOS USUARIOS INVOLUCRADOS 28](#_Toc109409149)

[PLAN DE TRABAJO 30](#_Toc109409150)

[DEVOPS 33](#_Toc109409151)

[INTRODUCCION 34](#_Toc109409152)

[DESARROLLO 35](#_Toc109409153)

[IDE GENEXUS 37](#_Toc109409154)

[DATABASE MYSQL 40](#_Toc109409155)

[GIT HUB 43](#_Toc109409156)

[JIRA 53](#_Toc109409157)

[PLAN DE PRUEBAS 65](#_Toc109409158)

[SELENIUM 67](#_Toc109409159)

[MAVEN 70](#_Toc109409160)

[MONITOREO CON GRAFANA 75](#_Toc109409161)

# INTRODUCCION

Se propone una solución a la empresa Admindata S.A de C.V, el desarrollo e implementación de un sistema web para la coordinación del área de Reclutamiento y Selección a través de un análisis de los procesos involucrados con el área de operaciones con el alta de la vacante, pasando por todo el procedimiento de reclutamiento de personal y finalizando con el envío de solicitud de contratación al área correspondiente, así mismo el desarrollo de una App para que los candidatos puedan llevar el seguimiento de sus postulaciones, ver y postularse en las vacantes que están vigentes.

El propósito de este sistema es agilizar el área de reclutamiento y selección, así como llevar un historial de todo los pasos establecidos en los procedimientos del área para las futura auditorias de ISO 9001:2015 que tendrá la empresa, tener un control total sobre las vacantes y los candidatos que han sido postulados, tener información actualizada para generar reportes al área de Dirección, y llevar una buena relación con los candidatos postulados con un seguimiento puntual.

Actualmente la empresa tiene diversos problemas como no tener la información concentrada de vacantes y candidatos, no saber que candidatos se han enviado al área de operaciones para que sean enviados con cliente, el área de reclutamiento no tiene respuesta rápida del área técnica para dar continuidad al procedimiento, pérdida de tiempo buscando datos e información de candidatos, no cuentan con una base de datos para buscar candidatos que ya hayan participado, no tienen como saber si algún candidato que ha participado cumple con requisitos para otras vacantes, candidatos duplicados, pérdida de tiempo por búsqueda de candidatos cuando la vacante ya fue cerrada, reclamos de clientes por la tardanza en la asignación de los candidatos seleccionados, reclamos de candidatos que se postulan y no tienen seguimiento, problemas con actualizaciones de perfiles que no son notificados a tiempo, los candidatos no saben si fueron postulados y no tienen seguimiento, estos problemas causan molestia a todas las áreas involucradas.

# DESCRIPCION DE LA EMPRESA

**Admindata**

Fundada en 2012, Admindata, es una Empresa 100% Mexicana proveedora de servicios y soluciones en TI, consultoría, desarrollo de sistemas y proyectos. Por más de 10 años se ha mantenido a la vanguardia, caracterizándose por aplicar tecnología de punta de acuerdo a las necesidades de cada cliente.

La experiencia del equipo de Ingenieros especializados en las diferentes herramientas, lenguajes, sistemas operativos y plataformas, ha tenido como resultado con sus clientes, el logro de una mejora competitiva que se ha visto reflejada en el cumplimiento puntual de los objetivos planteados.

**Misión**

Evolucionar productos y servicios de Tecnologías de la Información (TI), adelantándonos a las expectativas del cliente a través de personal capacitado, con valores humanos y espíritu de servicio, comprometidos día a día para lograr los objetivos establecidos por la organización.

**Visión**

Ser una empresa en continuo desarrollo, especialista en Tecnologías de la Información (TI), con presencia multinacional. Distinguida por la calidad del servicio a nuestros clientes, rentabilidad sostenida a los accionistas y crecimiento personal y profesional a colaboradores, con impacto positivo a la sociedad.

**Valores**

Todo el personal de ADMINDATA, ante sus partes interesadas declara y rubrica su compromiso de trabajar en el cumplimiento de su misión, y para el alcance de su visión, cumpliendo y viviendo los siguientes valores:

* **El RESPETO** a lo comprometido y a la libre expresión de las ideas y de las personas.
* **La HONESTIDAD** en el manejo de los recursos, que nuestro decir y hacer sea congruente, rechazando la mentira.
* **La LEALTAD** a la organización y al cambio.
* **La RESPONSABILIDAD** sobre las acciones realizadas e involucramiento con la satisfacción del cliente.
* **El TRABAJO EN EQUIPO** como parte de la integración de los colaboradores de la organización.
* **La ORIENTACIÓN AL CLIENTE** nos hace establecer vínculos con las necesidades y expectativas de nuestros clientes

Empresa con más de 10 años de experiencia en desarrollo de sistemas y asignación de personal, Cuenta con más de 60 consultores asignados en diferentes compañías, actualmente se vieron afectados por la pandemia, los clientes cerraron sus puertas por tal motivo no había entrevistas presenciales o requerimientos de personal externo, así mismo muchos consultores asignados se dieron de baja de la empresa por no ofrecer home office al 100%, con el cambio paulatino que se está dando, la empresa empieza a tener mayor contacto con clientes, generando nueva cartera y compromisos para la búsqueda de los mejores candidatos para ser asignados, es por ese motivo la necesidad de contar con un sistema que facilite el proceso de reclutamiento y selección.

# EQUIPO DE TRABAJO

El equipo de trabajo está compuesto por estudiantes de la carrera de Ing. en desarrollo y gestión de software.

|  |  |
| --- | --- |
|  | **Análisis y Calidad**  Martha Díaz Zúñiga  Se le asigna el rol de Análisis y calidad para documentar los procesos del sistema, así como realizar pruebas de funcionamiento y análisis de los procedimientos |
|  | **Project Manager**  Jorge Moisés Navarrete Morales  Se le asigna el rol de Project manager como intermediario entre la empresa y el equipo de trabajo, se encargara del avance y documentación del sistema, analizara los procedimientos para ofrecer la solución más factible al problema propuesto. |
|  | **Desarrollador**  José Trinidad Maciel Hernández  Se le asigna el rol de Desarrollador por la experiencia que ha demostrado en el análisis de sistemas y desarrollo en diferentes plataformas, se encargara del desarrollo del sistema con las herramientas seleccionadas |
|  | **Desarrollador**  Iván romero Mora  Se le asigna el rol de Desarrollador por la experiencia que ha demostrado en el análisis de sistemas y desarrollo en diferentes plataformas, se encargara del desarrollo del sistema con las herramientas seleccionadas |
|  | **Administrador de BD**  Diego Martin Garrido Flores  Se le asigna el rol de administrador de base de datos el cual se encargara de realizar el análisis y desarrollo y documentación de base de datos que se usara en el sistema. Así como apoyara en el análisis y documentación de los procesos de reclutamiento. |

# EQUIPO DE TRABAJO DE LA EMPRESA

Por parte de la empresa contaremos con la colaboración del representante legal y el equipo de las áreas de operaciones, reclutamiento y selección, entrevistadores técnicos y de sistemaas para el análisis e implementación del sistema a desarrollar.

|  |  |
| --- | --- |
|  | **Administrador General**  Mónica Garibay Martell  Es la encargada de todos los procesos internos de la empresa, lleva el control de la administración, finanzas y personal. |
| Alma Ruth Méndez - Business Transformation Practice Leader - Experis |  LinkedIn | **Gerente de Operaciones**  Alma Ruth Méndez Chávez  Es el enlace principal entre la empresa y los clientes, es quien da de alta las vacantes para iniciar la búsqueda de candidatos. |
|  | **Coordinadora de Reclutamiento y Selección**  Erika Zurema García Canedo  Controla y dirige todo el departamento de reclutamiento y selección, se encarga de revisar las vacantes, asignar el trabajo del equipo y entregar reportes a dirección del avance de las vacantes. |
|  | **Reclutadora**  Rosalina Solano Arellano  Se encarga de buscar candidatos viables y hacer los filtros necesarios para el procedimiento de reclutamiento y enviar candidatos viables. |
|  | . |

# PROBLEMATICA

Actualmente la empresa reporta problemáticas internas en todas las áreas debido a la mala organización que se tiene en todos los departamentos, los reclutadores no tienen seguimiento del área de operaciones, el área técnica no sabe la urgencia para la aplicación de exámenes lo cual retrasa los procedimientos, el área de operaciones no da seguimiento al área de reclutamiento con respecto a las entrevistas con cliente, los candidatos no tienen seguimiento a sus procesos.

Estas situaciones generan pérdidas a la empresa ya que es su giro principal y al no asignar personal calificado con los tiempos propuestos, mandar malos candidatos porque no pasan por todos los filtros lo cual hace que la empresa pierda su buena imagen que ha obtenido por su experiencia y genera una pérdida de clientes o recorte de vacantes a cubrir.

# AFECTA A

Este problema afecta a todas las áreas de la empresa, principalmente al área de Reclutamiento y Selección que es la encargada de proporcionar candidatos viables para cubrir las vacantes solicitadas, al área de operaciones ya que es la cara principal de la empresa y atiende al cliente directamente y al área de Dirección porque no se elaboran reportes de vacantes, candidatos, contratados, etc, que son muy necesarios para tomar acciones y generar cambios en la empresa.

Así mismo afecta a todos los candidatos que se postulan ya que al no tener un seguimiento adecuado y profesional, muchas veces ya no quieren participar en las vacantes que la empresa ofrece.

# IMPACTA A LA ORGANIZACIÓN

Finalmente el problema afecta a la empresa ya que no cumple con los contratos acordados con sus clientes, se mandan candidatos no seleccionados adecuadamente, no se cumplen los procedimientos establecidos y hay pérdidas económicas ocasionadas por las sustituciones de candidatos o perdidas de contratos con clientes.

# SOLUCIÓN TECNOLÓGICA

Desarrollo e implementación de un sistema web para almacenar toda la información del proceso de reclutamiento y selección, iniciando con la postulación del candidato y terminando cuando se tiene un status final (Aceptado/Descartado).

Implementación de una App que permitirá a los candidatos postularse a las vacantes vigentes, actualizar sus datos y Cv, así como saber el status en el que se encuentra en cada postulación en la que este participando.

Implementación de un sistema para la coordinación del área de Reclutamiento y Selección a través de un análisis de los procesos involucrados con el área de operaciones con el alta de la vacante, pasando por todo el procedimiento de reclutamiento de personal y finalizando con el envío de solicitud de contratación al área correspondiente, el mismo será implementado en un servidor con Windows Server 2016, será desarrollada para una plataforma web con un manejador de base de datos MySql, Apache Tomcat y Genexus 17 es una plataforma de desarrollo de software que simplifica y automatiza las tareas de crear y mantener aplicaciones de cualquier índole, se encuentra basado en el (Model- Driven Deveolpment o MDD), lo mejor de todo es que interactúa con lenguaje java

Se contara con un equipo con las mismas características que el servidor de producción para la implementación de los avances del sistema en el área de desarrollo, al cual se le implementaran las nuevas versiones, después de ser validadas con pruebas con el usuario final y herramientas de pruebas se documentara y se realizar un ticket de implementación en el servidor de producción del cliente el cual deberá tener varias aprobaciones para ser implementado, tomando en cuenta todas las políticas de las buenas practicas del área de sistemas del cliente.

El cliente actualmente ya cuenta con una infraestructura bien definida y procedimientos y políticas para la implementación de algún sistema en sus servidores las cuales debemos cumplir para la implementación de las mismas.

# OBJETIVO GENERAL

Implementar un sistema web y una App para garantizar que se cumpla el proceso completo de los candidatos postulados que participan en las vacantes mediante un flujo organizado para facilitar el seguimiento del área de Reclutamiento y Selección.

# OBJETIVOS ESPECIFICOS

* Centralizar la información del área de reclutamiento para que siempre este actualizada mediante una base de datos
* Facilitar la coordinación entre las áreas de Rys y operaciones para que se acorten los tiempos de espera en el proceso de asignación mediante alertas en el sistema.
* Facilitar el acceso a las evidencias para futuras auditorias de ISO9001:2015 mediante los reportes y almacenamiento de documentos
* Contar con el historial de las postulaciones de los candidatos para ser retomadas para futuras vacantes mediante las búsquedas en el sistema
* Facilitar el cumplimiento del procedimiento de reclutamiento y selección para que solo los candidatos viables sean enviados con el cliente mediante filtros realizados en el sistema.

# PROBLEM STATEMENT

La empresa Admindata SA de CV ha optado por el desarrollo de un sistema que permitirá llevar un control y seguimiento del área de Reclutamiento y Selección, el sistema contará con módulos específicos, iniciando con una pantalla de usuario y contraseña que tendrá acceso restringido según los roles establecidos por Dirección General.

Su funcionabilidad principal es que los candidatos postulados a una vacante cumplan con todo el proceso de reclutamiento y selección así mismo concentrar la información que es capturada por el área de reclutamiento y selección, operaciones y el área técnica, manteniendo actualizada e informada a cada una de las áreas sobre el ingreso y seguimientos de vacantes, de los candidatos postulados, así como controlar los procesos de la misma, lo cual facilitara que los candidatos sepan los status de sus postulaciones y que todas las áreas tengan la misma información en cualquier momento con el fin de evitar un retraso y doble trabajo por parte del personal, obteniendo un orden e identificación de cada postulado y ahorrando tiempo en los procesos ya implementados.

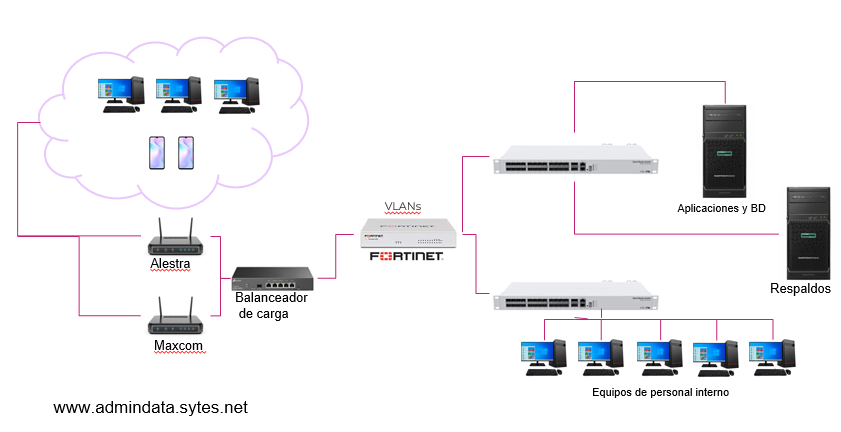
Las funcionalidades serán habilitadas y autorizados según el rol que sea asignado, se crea de acuerdo a las necesidades de la empresa, este sistema iniciara desde el establecimiento de roles de usuarios para acceso al mismo, módulos que inician desde la publicación y seguimiento de una vacante, seguimiento de candidatos que participaran en las vacantes en todo el procedimiento (entrevista de primer filtro, entrevista técnica, entrevista con cliente final) hasta la asignación del candidato viable y seleccionado por el cliente, permitirá a los candidatos ver el seguimiento que se le da a cada una de sus postulaciones.

La seguridad con la que cuenta la aplicación será un usuario y contraseña asignados de manera individual a todos los tipos de usuarios del sistema dependiendo del rol al que tengan acceso en un registro previo.

# ARQUITECTURA

Actualmente la empresa ADMINDATA ya cuenta con infraestructura implementada, a la cual nos darán acceso para la implementación del sistema, debemos cumplir con los procedimientos establecidos para la implementación en su servidor de producción, inicialmente se realizaran las pruebas en un servidor con las mismas características, después de validar el buen funcionamiento se solicitara la instalación por módulos hasta que se realizar una entrega total del sistema.

La infraestructura del cliente permitirá que el sistema siempre esté disponible las 24/7, con la mayor seguridad y protección ante cualquier fallo, realizando respaldos diarios de la base de datos y cada semana la última versión de la base de datos y el aplicativo son almacenados en un lugar alterno.



## PLATAFORMA DE EJECUCIÓN

El servidor en al cual se implementara corre bajo un sistema operativo Windows server 2016, base de datos (MySQL), Apache Tomcat y GENEXUS 17, con medidas de seguridad implementadas por el cliente, con respaldos diarios de la misma, los cuales son guardados de forma local en otro servidor y transportados a una zona de seguridad cada semana con el ultimo respaldo en un sitio alterno.

Características del servidor de producción:

* Servidor HPE ProLiant ML110 Gen10,
* Procesador Intel Xeon Bronze 3204 1.90GHz
* Memoria Ram 16GB DDR4
* DD 4TB SATA, 3.5"
* Tower (4,5U)
* Windows Server 2016
* Kaspersky Anti-Virus para Windows Server

Características del SITE:

* Ancho de banda mínimo de 100Mb/s con asistencia técnica las 24/7
* 2da línea de ancho de banda mínimo de 10Mb/s con asistencia técnica las 24/7 Dedicado
* Conexión simétrica dedicada con una disponibilidad del 99.9%,
* Sistema de almacenamiento SSD-SAN con tráfico ilimitado
* Certificado de SSL Wildcard
* Protección DDos, IPS/IDS, SIEM
* Backup diario
* Imágenes del servidor semanalmente
* Monitorización constante
* Acceso Remoto cifrado

# REQUERIMIENTOS FURPS

## FUNCIONALES

* El sistema controlará el acceso mediante un usuario y contraseña
* El sistema permitirá modificar nombre de usuario y contraseña al usuario de administrador
* El sistema permitirá solicitar modificación de usuario y contraseña al usuario administrador
* El sistema permitirá al administrador el registro de nuevos usuarios y asignara un rol específico.
* Cuando el usuario acceda a la aplicación se mostrara una pantalla inicial, donde se observara los datos de la persona que ingreso, así como sus notificaciones más relevantes.
* El sistema permita que el área de operaciones sea capaz de crear una vacante nueva
* El sistema permitirá al área de operaciones realizar cambios a las vacantes.
* El sistema enviará un aviso al área de Rys sobre cualquier modificación o creación de vacantes
* El sistema permita que el área de RyS sea capaz de dar de alta un nuevo candidato
* El sistema permitirá al área de Rys realizar cambios a la información de los candidatos
* El área de Rys podrá dar vincular en el sistema a los candidatos en las vacantes
* El área de Rys podrá modificar los status de candidatos
* El área de Rys podrá generar los exámenes para los candidatos
* El área de Rys podrá cargar documentos de los candidatos
* El área de Rys podrá enviar a los candidatos a entrevista técnica
* El área técnica podrá revisar las entrevistas realizadas por el área de Rys y realizar su entrevista en el sistema
* El área técnica podrá capturar el comentario final de su entrevista y cambiar el status de candidato
* El área de rys recibirá comentarios del área técnica y calificación del candidato
* Rys podrá generar en el sistema reporte de concentrado del candidato
* El área de operaciones podrá enviar a los candidatos seleccionados a entrevista con cliente
* El área de operaciones podrá cambiar status de candidatos con respuesta y capturar comentarios del cliente.
* El área de Rys podrá enviar al área de Administración del personal datos y documentos de candidatos aceptados por el cliente

## NO FUNCIONALES

Usabilidad (Usability)

* El sistema deberá ser intuitivo y de fácil uso para cualquier usuario
* El sistema deberá proporcionar mensajes de error informativos orientados al usuario y brinden una posible solución
* El sistema deberá mantener informado al usuario de lo que ocurre en la plataforma con tiempos rápidos de respuesta, barras de proceso, animaciones precisas y con unfeedback constante.
* El sistema deberá utilizar el lenguaje del usuario, con expresiones y palabras que le resulten familiares.
* El sistema proporcionará ayuda al usuario para evitar errores
* El sistema contará con estética y diseño minimalista, para evitar información innecesaria
* El sistema debe contar con manuales de usuario estructurados adecuadamente.

Confiabilidad (Reliability)

* El sistema deberá asegurar que los datos estén protegidos del acceso no autorizado
* El sistema deberá generar copias de seguridad cada N horas.
* Si se identifican ataques de seguridad o brecha del sistema, el mismo no continuará operando hasta ser desbloqueado por un administrador de seguridad.
* El sistema debe tener una disponibilidad del 99,99% de las veces en que un usuario intente accederlo.
* La tasa de errores cometidos por el usuario deberá ser menor del 1% de las transacciones totales ejecutadas en el sistema.
* La probabilidad de falla del sistema deberá ser la más mínima posible

Desempeño (Performance)

* Toda funcionalidad del sistema y transacción debe responder al usuario en menos de 3 segundos
* El sistema debe ser capaz de procesar N transacciones por segundo
* El sistema debe operar correctamente con N cantidad de usuarios concurrentes
* El ciclo de reinicio y encendido del sistema debe ejecutarse completamente en menos de 60 segundos.
* El sistema deberá estar disponible las 24 horas, los 7 días de la semana
* La aplicación debe ser entregada en un empaquetado (.war)
* Facilidad de Soporte (Supportability)
* El sistema debe ser mantenible y extensible.
* El sistema estará diseñado para que pueda migrarse a un hardware más actualizado a nuevas versiones de sistemas operativos.
* El sistema deberá poder migrar a otro tipo de red si es requerido
* El sistema debe poder establecer y mantener conexión estable con otros sistemas sin importar las diversas tecnologías utilizadas en su desarrollo.
* Los requisitos para su instalación deberán ser lo menos exigentes posible

# REGLAS DEL NEGOCIO

**Principal**

Todos los candidatos cumplan con el procedimiento de reclutamiento en cada postulación.

**Administrador**

* Acceso total a todos los módulos.
* El administrador es el único que puede hacer cualquier modificación al usuario (cambio de nombre de usuario, contraseña o rol asignado)
* El administrador puede generar cualquier tipo de reporte en el sistema
* El administrador puede modificar información de vacantes
* El administrador puede modificar información de candidatos
* El administrador puede modificar información de postulaciones
* El administrador puede modificar información de exámenes técnicos
* El administrador es el único que podrá cambiar el reclutador dueño del candidato en caso extraordinario avalado por Dirección General.
* Se deberá contar con un log que guarde toda la información de los cambios realizados por los usuarios en el sistema
* El administrador podrá cambiar de status a los candidatos si es solicitado por Dirección General.

**Operaciones**

* Puede ver todas las vacantes
* Puede capturar vacantes nuevas
* Cuando se capture una vacante nueva el sistema deberá mandar un mensaje al área de rys con los datos de la nueva vacante.
* Puede modificar toda la información respecto a la vacante
* Cuando se de cualquier tipo de cambio a la vacante el sistema deberá mandar un mensaje al área de rys con los cambios realzados.
* El usuario de operaciones podrá ser capaz de generar el archivo de primer filtro para el área de reclutamiento que incluya todos los campos solicitados
* El usuario de operaciones podrá ser capaz de agregar campos para generar el archivo de primer filtro para el área de reclutamiento
* El sistema deberá validar que no se puede abrir una vacante si no se genera al mismo tiempo el archivo de primer filtro
* El usuario de operaciones únicamente podrá ver la información de candidatos mas no modificarla
* El usuario de operaciones únicamente podrá ver la información de postulaciones mas no modificarla
* El usuario de operaciones podrá generar el reporte de concentrado de candidatos
* El usuario de operaciones podrá hacer modificaciones en el reporte de concentrado de candidato
* El sistema permitirá enviar un correo al cliente anexando formato de envío de candidato viable, anexando cv recortado y concentrado de información de skills de candidato
* El usuario de operaciones podrá capturar la retroalimentación del cliente final y cambiar status general del candidato.
* Si el candidato es rechazado por el cliente, el sistema avisará al área de reclutamiento para que el candidato sea informado.
* El sistema permitirá cambiar al usuario de operaciones el status de seguimiento de las vacantes con envío de aviso al are de Rys de cualquier cambio.
* El sistema permitirá al usuario de operaciones poner en suspensión una vacante, para lo cual se deberá especificar el tiempo para que el área de reclutamiento pueda avisar a candidatos
* El sistema permitirá cambiar el status de standbye a activo únicamente si el área de operaciones tiene un formato autorizado con fecha de reinicio de búsqueda.
* Al momento de reactivar una vacante el sistema enviara un mensaje a Rys para sugerir llamar a los candidatos ya postulador para verificar disponibilidad
* El usuario de operaciones tendrá un botón de candidato seleccionado el cual emitirá un mensaje al área de reclutamiento iniciando el proceso de contratación.
* El sistema enviara de forma automática el formato de inicio de proceso de contratación con toda la información del candidato al área de reclutamiento y en rojo lo que haga falta para que sea registrado inmediatamente y enviado al área de Administración de personal
* El usuario de administrador podrá ver que reclutador es dueño del candidato asignado
* Puede generar reporte de status de vacantes por cliente
* Puede generar reporte de status de candidatos por vacante
* Puede generar reporte de status de candidatos por cliente
* Puede generar reporte de historial de vacantes por cliente
* Puede generar reporte de historial de candidatos por vacante por cliente

**Reclutamiento**

* El usuario de Rys podrá ver todas las vacantes sin importar status
* El usuario de Rys no podrá modificar datos de la vacante
* El usuario de Rys podrá dar de alta a candidatos nuevos en cualquier momento
* El sistema deberá validar el nombre del candidato al ser capturado por el usuario de Rys para evitar duplicados
* Si el usuario de Rys Captura un candidato que ya existe en el sistema, mostrara lo datos del reclutador que es dueño del candidato
* El sistema permitirá ceder candidatos entre cuentas de Rys
* El sistema debera guardar un historial de cambios de cuentas de Rys con fechas
* El usuario de Rys no puede modificar los datos de ingreso o dueño de candidato de los candidatos ya ingresados por otro reclutador.
* El usuario de Rys podrá ver la información de candidatos ingresados por otros usuarios de Rys
* El usuario de Rys podrá capturar todos los datos de contacto y seguimiento de los candidatos de cualquier usuario de Rys
* El sistema podrá desligar el reclutador de un candidato cada 6 meses.
* El sistema deberá validar que se llene por completo la pantalla de entrevista de primer filtro, permitiendo guardar los avances de la misma.
* El usuario de Rys podrá postular candidatos a las vacantes
* El usuario de Rys podrá cambiar el status del candidato en cualquier momento.
* El área de Rys podrá capturar la información del candidato en cualquier momento.
* El área de Rys podrá subir los documentos de los candidatos ya que hayan sido validados
* El usuario de Rys podrá postular candidatos a exámenes técnicos únicamente si esta validado que está completo el examen de primer filtro.
* El usuario de Rys puede ver status y comentarios de exámenes técnicos
* El usuario de Rys podrá coordinar las entrevistas con cliente y candidato cuando el área de operaciones lo haya solicitado
* El usuario de Rys podrá ver seguimiento de entrevista de candidatos con clientes
* El usuario de Rys podrá capturar seguimiento a candidatos en todas las 3 etapas (1r filtro, examen técnico, respuesta de cliente)
* Puede ver reporte de status de vacantes por cliente
* Puede ver reporte de status de candidatos por vacante
* Puede ver reporte de status de candidatos por cliente
* Puede ver reporte de historial de vacantes por cliente
* Puede ver reporte de historial de candidatos por vacante por cliente

**Técnicos**

* El usuario técnico podrá ver solo información de los candidatos primer filtro
* El usuario técnico podrá descargar solo el cv del candidato
* El usuario técnico podrá aplicar examen técnico a candidatos en cualquier momento
* El usuario técnico capturará observaciones de examen técnico
* El usuario técnico podrá cargar evidencias de los exámenes aplicados
* El usuario técnico calificar al candidato (viable o no viable)
* El sistema no permitirá calificar candidato sin que se llene el cuestionario de entrevista técnica
* El sistema no permitirá avanzar el proceso de un candidato si no está llena la entrevista técnica.
* El sistema no permitirá al área técnica ver quién es el dueño de un candidato

**Consultores**

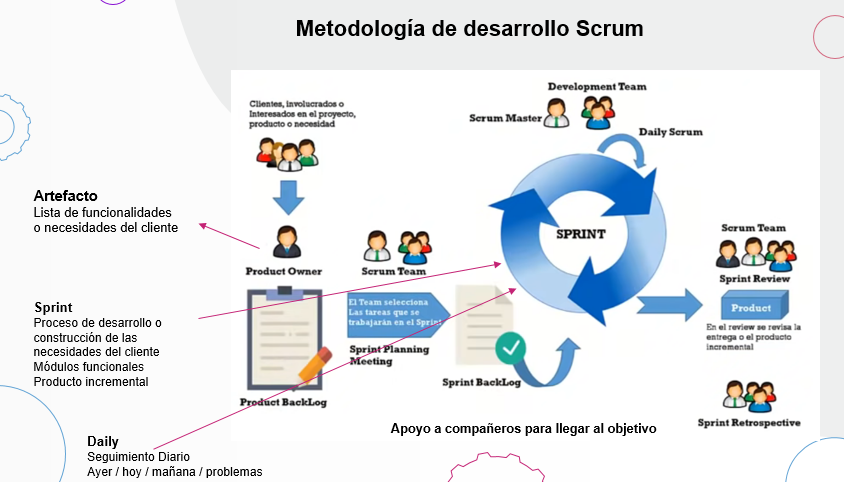
* Pueden registrarse en el sistema
* El candidato únicamente podrá ver status de sus postulaciones
* El candidato puede ver la observaciones de su proceso de selección
* Únicamente puede ver sus datos personales de contacto
* El candidato puede ver todas las vacantes activas en el sistema
* El candidato puede postularse a máximo 3 vacantes
* El candidato puede descartarse de cualquier postulación

# METODOLOGÍA DE DESARROLLO

**Metodología SCRUM**

Scrum es una metodología ágil y flexible para gestionar el desarrollo de software que permite el trabajo colaborativo entre equipos, scrum incluye un conjunto de reuniones, herramientas y funciones que, de forma coordinada, ayudan a los equipos a estructurar y gestionar su trabajo.

Reconoce que el equipo no lo sabe todo al inicio de un proyecto y evolucionará a través de la experiencia. Scrum está estructurado para ayudar a los equipos a adaptarse de forma natural a las condiciones cambiantes y a los requisitos de los usuarios, con el cambio de prioridades integrado en el proceso y ciclos de lanzamiento breves para que tu equipo pueda aprender y mejorar constantemente.



**Épica:** es un conjunto de trabajo grande que puede dividirse en tareas específicas (denominadas “historias de usuario”) en función de las necesidades o solicitudes de los clientes o usuarios finales.

**Una historia de usuario**: Una historia de usuario es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final o cliente

**Backlog del producto:** es la lista principal del trabajo, se trata de una lista dinámica de funciones, requisitos, mejoras y correcciones que actúa como la entrada para el backlog de sprint. Básicamente, se trata de la lista de "cosas por hacer" del equipo.

**Sprint:** Un sprint es el periodo real en que el equipo de scrum trabaja de forma conjunta para finalizar un incremento. La duración de un sprint suele ser de dos semanas, aunque algunos equipos manifiestan que les resulta más fácil una semana para el alcance o un mes para entregar un incremento valioso.

**Reunión rápida Daily**: Se trata de una reunión diaria de muy corta duración que tiene lugar siempre a la misma hora (normalmente, por las mañanas) y en el mismo sitio para facilitar las cosas. Muchos equipos tratan de finalizar la reunión en 15 minutos, La reunión rápida es el momento de expresar cualquier inquietud que se tenga con el cumplimiento del objetivo del sprint o de notificar los impedimentos existentes, Una forma habitual de realizar la reunión rápida es que cada miembro del equipo responda tres preguntas en el contexto de alcanzar el objetivo del sprint: ¿Qué hice ayer?, ¿Qué tengo planificado para hoy?, ¿Hay algún obstáculo?

# HERRAMIENTAS DE DESARROLLO

**GeneXus™**

* Es una plataforma de desarrollo de software que simplifica y automatiza las tareas de crear y mantener aplicaciones del tipo empresarial.
* GeneXus fue la opción más económica y con licencia amigable que encontramos entre más de 10 productos analizados que permitan desarrollar, implementar y administrar aplicaciones, se analizaron las siguientes herramientas Outsystems, Mendix, Appian, Kony, QuickBase, wavemaker.
* Permite el Desarrollo en varios lenguajes y trabajar con distintos manejadores de bases de datos, lo que te permite escoger entre los Sistemas operativos Windows o Linux.
* Permite realizar el sistema web y realizar una Aplicación Móvil
* Cuenta con su propio controlador de versiones.

**MySQL**

* Es una base de datos gratuita. Al ser de código abierto, no tiene costo.
* Es muy fácil de usar. Podemos empezar a usar la base de datos MySQL sabiendo unos pocos comandos.
* Es una base de datos muy rápida. Tiene muy bien rendimiento.
* Utiliza varias capas de seguridad. Contraseñas encriptadas, derechos de acceso y privilegios para los usuarios.
* Pocos requerimientos y eficiencia de memoria.
* Es compatible con Linux y Windows.

**Apache Tomcat**

* Apache es un servidor web HTTP de código abierto. Está desarrollado y mantenido por una comunidad de usuarios en torno a la Apache Software Foundation.
* Actualmente y desde el 1996 es el servidor web más usado en todo el mundo debido a su seguridad y estabilidad.
* Es gratuito y de fuente abierta.
* Instalación y configuración sencilla
* Altamente extensible y adaptable mediante módulos
* Funciones incorporadas para autentificación y validación de usuarios.
* Soporte para muchos lenguajes

**Java**

El generador Java es la solución GeneXus para desarrollar aplicaciones business to business (B2B). Además, con Java también se generan Web Panels lo que permite desarrollar aplicaciones bussiness to consumer (B2C), teniendo así una solución completa para el ambiente Internet.

Java es un lenguaje orientado a objetos y a Internet porque está diseñado para trabajar con el protocolo TCP/IP y habilita la carga dinámica de las páginas Web. Se trata de un lenguaje multiplataforma, es decir que corre en diferentes sistemas operativos, que no requiere de la instalación de un software para su ejecución, sino que alcanza con que el cliente tenga un navegador y la dirección Web correspondiente a la aplicación.

Para el desarrollo de aplicaciones B2C, GeneXus cuenta con el objeto Web Panel en sus generadores C/SQL, Visual Basic y Java. Mientras que C y VB usan la tecnología CGI-BIN, Java utiliza la tecnología Servlets. Comparativamente, el rendimiento de los Web Panels generados con Java, es mayor que los generados con Visual Basic, y es igual o menos rápido que C/SQL: aquí dependerá de la cantidad de webpanels y de la complejidad de la lógica de los mismos. Para generar Servlets, se requiere un JavaServerWeb Development Kit (JSWDK) para la compilación y un motor de Servlets en el servidor para la ejecución.

**Jira**

* JIRA es una herramienta para la gestión de proyectos, procesos, incidencias y actividades de cualquier tipo que puedan tener vinculados flujos de trabajo.
* Inicialmente fue diseñado para la gestión de proyectos de software, incluyendo la gestión de requerimientos, gestión de evolución del proyecto y notificación y resolución de incidencias y errores. Aún hoy en día se sigue utilizando para la gestión de proyectos de desarrollo de software basados en metodología ágil.
* Se utiliza en la actualidad para la gestión de proyectos de todo tipo ya que permite la definición, asignación y seguimiento de las tareas que conforman el proyecto.

# DESCRIPCIÓN DE ROLES PARA EL SISTEMA

Un rol es una colección de permisos definida para todo el sistema se pueden asignar a usuarios específicos en contextos específicos. La combinación de roles y contexto definen la habilidad de un usuario específico para hacer algo en el sistemas.

Los roles identificados para el sistema SRS de reclutamiento y se lección son los siguientes:

* **Administrador**

Este rol tendrá acceso a todos los módulos del sistema con todos los permisos existentes en el sistema.

* **Operaciones**

El usuario de operaciones podrá hacer cualquier cambio en el módulo de vacantes, podrá ver las postulaciones pero no podrá hacer ninguna modificación en reclutamiento o perfil de los candidatos

* **Coordinador de Reclutamiento y Selección**

Este usuario únicamente podrá ver los datos de las vacantes pero no modificarla, podrá hacer cambios en todos los módulos de candidatos, postulaciones y reclutamiento.

* **Reclutador**

Este usuario solo puede hacer modificaciones en sus candidatos, puede ver la información de los vacantes pero no modificarlas, podrá hacer cambios en los módulos de postulaciones, candidatos y reclutamiento

* **Consultor**

Este usuario solo podrá ver las postulaciones en las que está participando, podrá ver las vacantes vigentes para postularse y podrá cambiar su perfil.

* **Entrevistador Técnico**

Este usuario únicamente podrá ver el perfil del candidato no podrá hacer cambios, podrá descargar únicamente el examen de primer filtro y CV de los candidatos, podrá calificar a los candidatos para continuar con el proceso y debe documentar la entrevista de cada candidato antes de calificarlo.

# DESCRIPCIÓN DE MENUS DEL SISTEMA

La página web contara con los siguientes módulos:

**Home**

Página principal en la cual se mostrara un resumen de las actividades pendientes y vencidas del personal, avisos de dirección y de administración de personal.

**Vacantes**

Aquí se darán de alta las vacantes las cuales podrán ser administradas por el persona de reclutamiento y selección, manteniéndolas activas para darle seguimiento o cancelarlas.

**Candidatos**

En esta sección se capturará todos los datos referentes a los candidatos que participaran en las vacantes, como datos personales, estudios, experiencia, skills, certificaciones, contara con una sección para subir sus documentos.

**Reclutamiento**

En esta sección se enlazaran las vacantes y los candidatos que participaran en cada una de ellas, se llevara un histórico con notas de avance de cada proceso que se cumpla.

**Estadísticas**

Se contara con esta área para la elaboración de las principales estadísticas que la empresa necesita revisar constantemente y de manera gráfica con periodos de tiempo.

**Mi perfil**

En esta sección cada usuario podrá personalizar sus datos y contraseña.

**Control de Accesos**

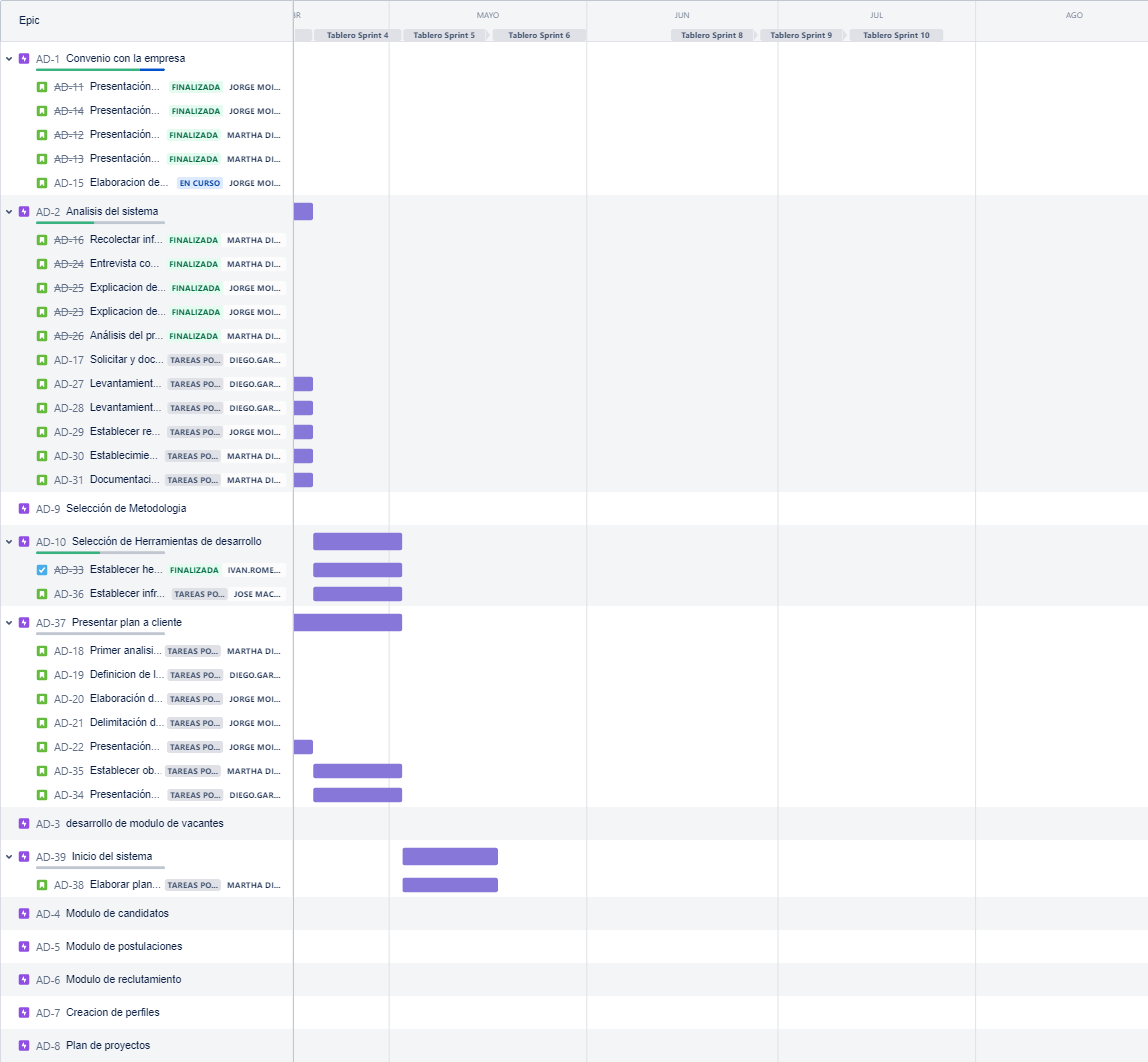
En esta sección solo tendrá acceso el administrador para dar de alta, baja, cambios a los usuarios.

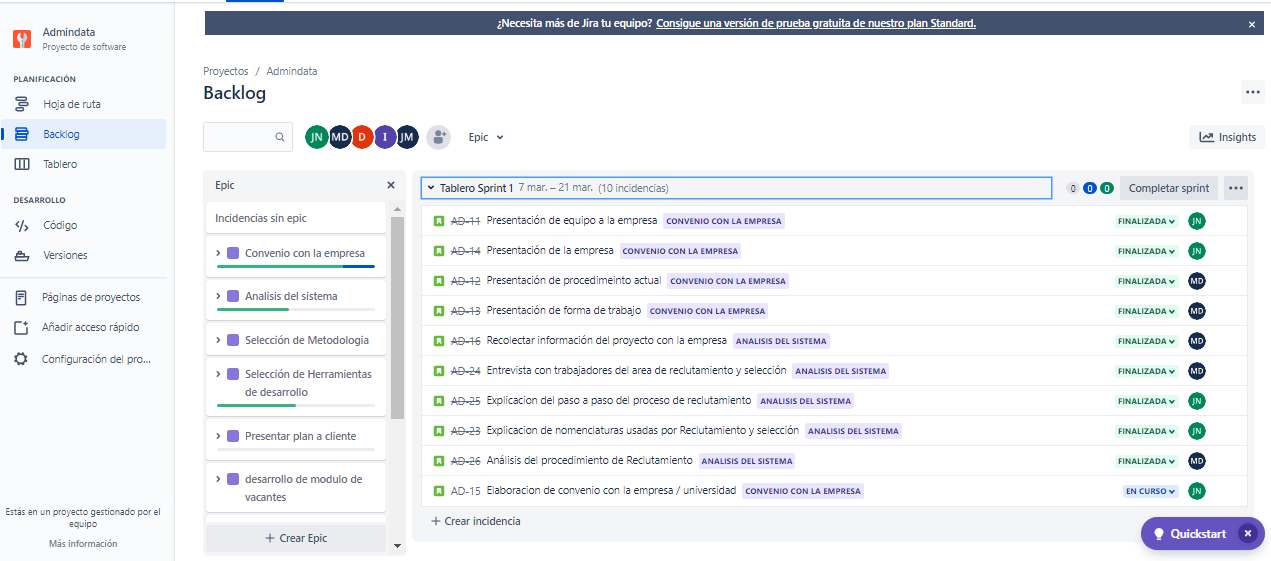
# PRINCIPALES NECESIDADES DE LOS USUARIOS INVOLUCRADOS

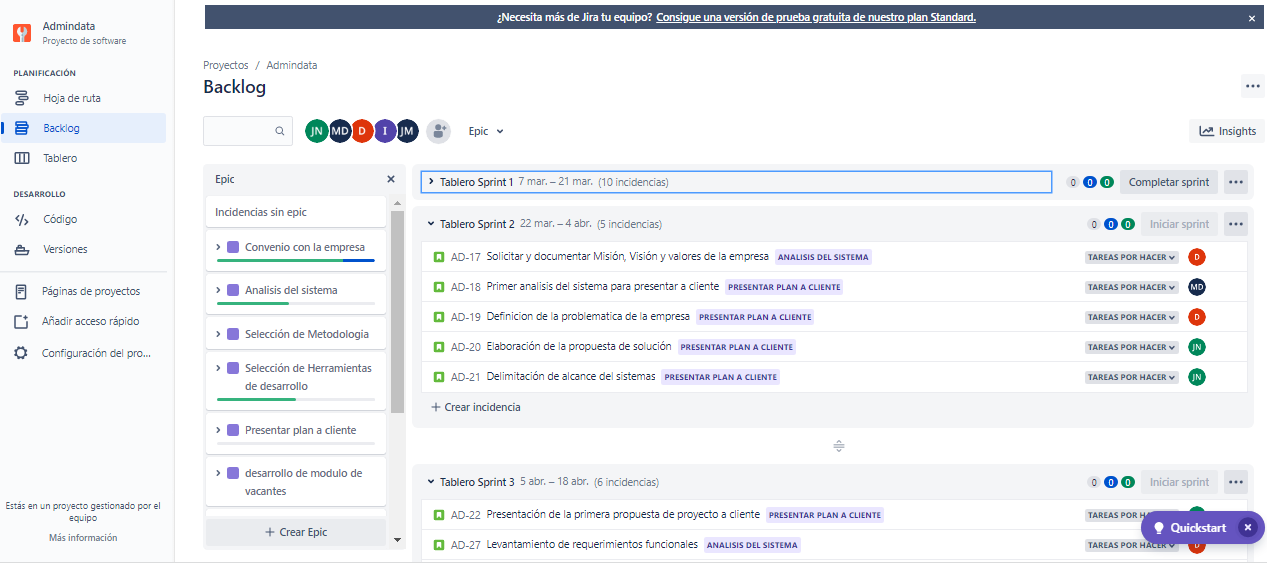
|  |  |  |
| --- | --- | --- |
| Tener la información concentrada de vacantes y candidatos | No se tiene un concentrado de información | Centralizar la información en una base de datos |
| Saber que candidatos se han enviado al área de operaciones | No se lleva un registro global de toda el área con estos pasos de sus procedimientos | El sistema mostrará los candidatos que participan en cada vacante y su seguimiento |
| Saber que candidatos son enviados con cliente | No se lleva un registro global de toda el área con estos pasos de sus procedimientos | El sistema mostrará los candidatos que participan en cada vacante y su seguimiento |
| Reclutamiento no tiene respuesta rápida del área técnica | Los técnicos no se enteran de que tienen entrevistas por elaborar | El sistema notificara al área para dar continuidad al proceso |
| Pérdida de tiempo buscando datos e información de candidatos | No se tiene información centralizada | Centralizar la información en una base de datos con acceso sencillo |
| Base de datos para buscar candidatos | No se tiene información centralizada | Centralizar la información en una base de datos con acceso sencillo |
| No tienen como saber si algún candidato que ha participado cumple con requisitos para otras vacantes | No hay comunicación entre reclutadores | Se podrá revisar cualquier perfil de cualquier candidato para ser re postulado |
| No tener candidatos duplicados | No hay comunicación entre reclutadores | Centralizar la información en una base de datos con acceso sencillo con validaciones |
| Pérdida de tiempo por búsqueda de candidatos | No se tiene información centralizada | Se podrá revisar cualquier perfil de cualquier candidato para ser re postulado |
| Reclamos de clientes por la tardanza | No se cumplen las expectativas del cliente | Con el proceso ya sistematizado se podrá enviar candidatos viables más rápido. |
| Reclamos de candidatos que se postulan y no tienen seguimiento | No se le da información al candidato | El sistema enviara un correo de cada paso por el que avance el candidato. |
| Problemas con actualizaciones de perfiles | No hay comunicación entre reclutadores | Ya no habrá perfiles duplicados con las validaciones que se implementaran |
| Los candidatos no saben si fueron postulados | No hay seguimiento a los candidatos | El sistema enviara un correo de cada paso por el que avance el candidato. |

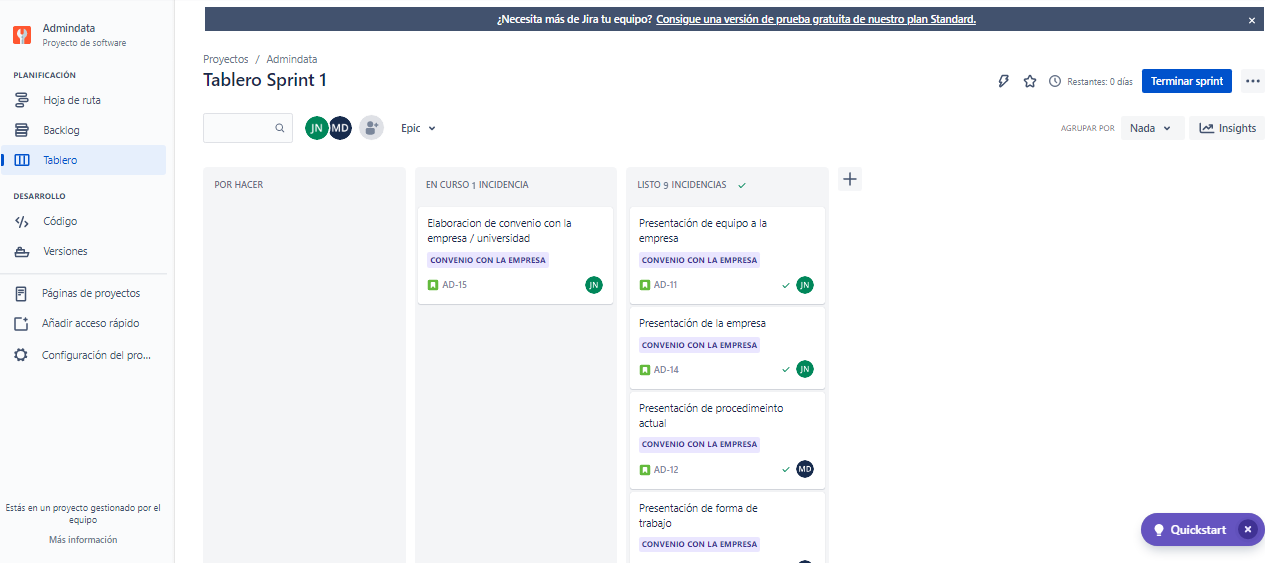
# PLAN DE TRABAJO

El sistema será llevado en la plataforma Jira, la cual nos ofrece el control completo del sistema con la metodología SCRUM



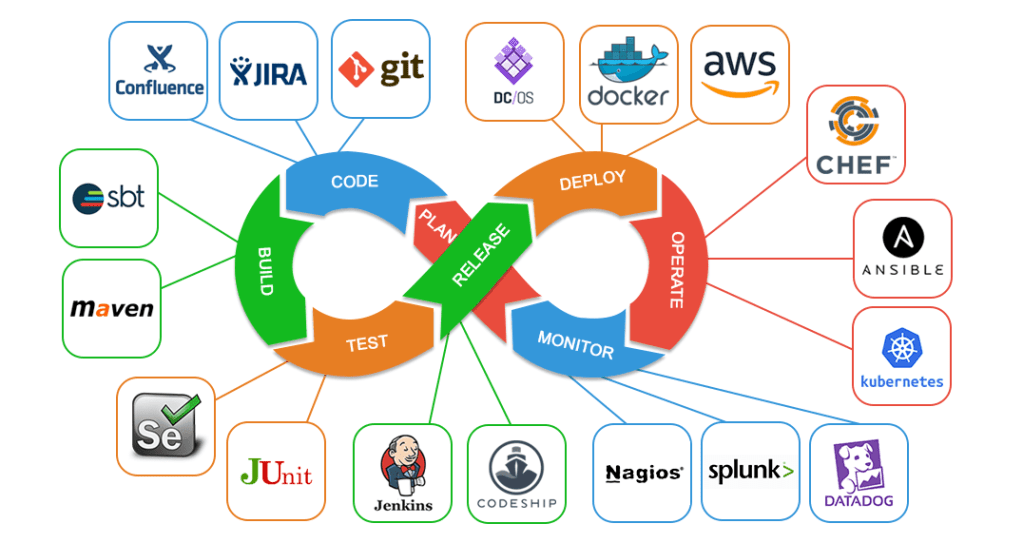






# DEVOPS

**DEVOPS**



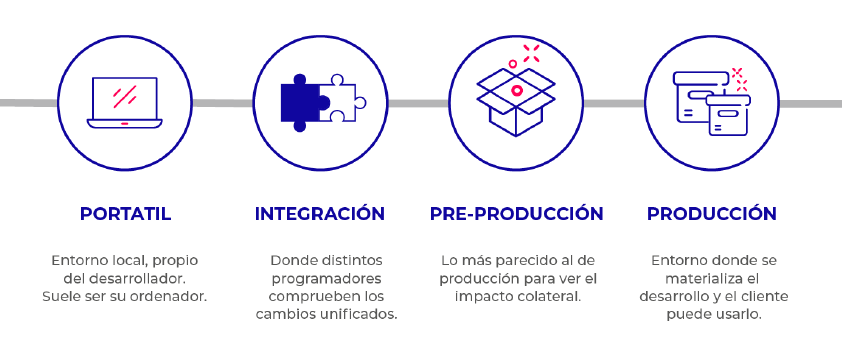
Una de las claves para garantizar la calidad en las entregas es reducir la intervención humana a la hora de instalar entornos, empaquetar, testar, entregar o incluso volver al estado anterior en caso de problemas.”

# INTRODUCCION

DevOps promueve un conjunto de prácticas que combina el desarrollo de software (dev) con operaciones y gestión de la infraestructura sobre la que va a ir instalado el software (ops). Todas estas partes participan juntas en el ciclo de vida de un producto o aplicación, y cuanto más involucradas estén, más rápida y productiva será la entrega de valor.

Mediante un proceso de descubrimiento, análisis, refinamiento y priorización, que puede ser ágil usando SCRUM, al desarrollador le llega una necesidad o requisito (requerimiento), que podría estar formulada como una historia de usuario. Como por ejemplo: “Como usuario de reclutamiento puedo ver cuantos candidatos están en primer filtro en las vacantes”. Entonces el desarrollador se encuentra con la responsabilidad de programar la funcionalidad necesaria para que los usuarios finales puedan llevar a cabo esa acción.

Una vez finalizada la programación de la funcionalidad y se hacen pruebas contra el entorno local del desarrollador, probarlas, subir los cambios a los distintos entornos, los cuales son como los siguientes:



Para estos cambios suelen requerir de un orden determinado y, al ser tan frecuente el proceso, lo mejor es automatizarlos. Así además evitamos errores por el factor humano. Por ello, como parte de la cultura DevOps, los cambios que se hagan en el desarrollo, al subir al repositorio de código, van a lanzar un proceso de integración contínua CI (continuous integration), que generalmente incluye la compilación del código, la ejecución de pruebas automáticas, la generación de artefactos y algunas otras fases como el análisis de calidad de código automático.

Si todo esto es correcto se podría ejecutar un proceso de despliegue continuo, (Continuous Deployment) de tal forma que la nueva versión de la aplicación con los nuevos cambios incluidos estaría disponible para ser probada en un entorno de integración. El entorno de integración es usado en su mayoría por los desarrolladores para verificar que la funcionalidad entregada cumple con los requisitos y no tiene errores. Es tan importante aportar nueva funcionalidad como no romper las que ya se habían entregado.

Tenemos que ser conscientes de que en las empresas suele haber muchos equipos de desarrollo que necesitan usar el entorno de integración. No dejar al libre albedrío el acceso y modificaciones de este entorno permite reducir errores humanos (como la ejecución de comandos incorrectos, la subida de versiones sin aprobación de las pruebas o, incluso, el olvido de la ejecución de alguno de los pasos del proceso) que, bien por inexperiencia o falta de conocimiento, pueden ocurrir. En este sentido, lo ideal es que los cambios sean automatizados.

# DESARROLLO

Los seguidores de las prácticas de DevOps a menudo incorporan a su «cadena de herramientas» de DevOps particular algunas herramientas que se adaptan perfectamente a estos métodos. El objetivo de estas herramientas es tratar de optimizar, acortar y automatizar las diversas etapas del flujo de trabajo de creación de software (o «canalización»). Muchas de estas herramientas también promueven los postulados principales de DevOps, como son la automatización, la colaboración y la integración entre los equipos de desarrollo y operaciones. A continuación se ofrece un ejemplo de herramientas que se emplean en las diversas etapas del ciclo de DevOps.

**Planificación.** En esta fase se definen los requisitos y valores empresariales. Algunas herramientas de muestra son Jira o Git, con las cuales se puede hacer un seguimiento de los problemas conocidos y llevar a cabo la gestión de los proyectos.

**Codificación.** Esta fase implica el diseño del software y la creación del código. Algunas herramientas de muestra son GitHub, GitLab, Bitbucket o Stash.

**Compilación.** En esta fase se gestionan las versiones y las compilaciones del software, y se utilizan herramientas automatizadas que ayudan a compilar y crear paquetes de código para publicarlos después para la producción. Se utilizan repositorios de código fuente o repositorios de paquetes que también «empaquetan» la infraestructura que se necesita para el lanzamiento del producto. Algunas herramientas de muestra son Docker, Ansible, Puppet, Chef, Gradle, Maven o JFrog Artifactory.

**Prueba.** Esta fase incluye la realización de pruebas continuas (manuales o automatizadas) para garantizar la calidad de la programación. Algunas herramientas de muestra son JUnit, Codeception, Selenium, Vagrant, TestNG o BlazeMeter.

**Puesta en marcha.** En esta fase se emplean herramientas que ayudan a gestionar, coordinar, programar y automatizar las tareas de producción de las versiones de productos. Algunas herramientas de muestra son Puppet, Chef, Ansible, Jenkins, Kubernetes, OpenShift, OpenStack, Docker o Jira.

**Funcionamiento.** En esta fase se gestiona el software durante su producción. Algunas herramientas de muestra son Ansible, Puppet, PowerShell, Chef, Salt o Otter.

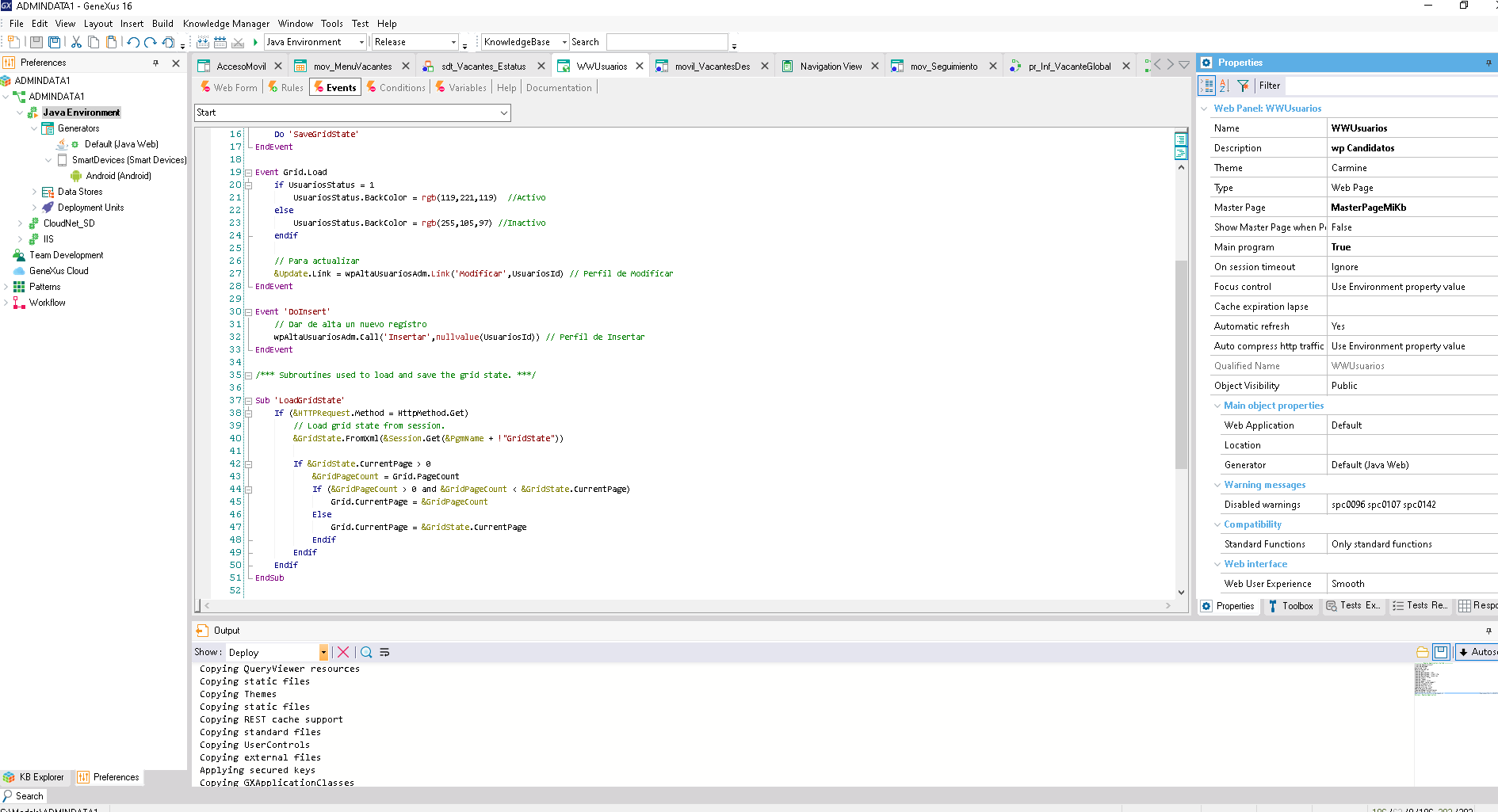
**Supervisión.** En esta fase se identifica y recopila información sobre problemas que surgen en una versión de software específica que se encuentra en producción. Algunas herramientas de muestra son New Relic, Datadog, Grafana, Wireshark, Splunk, Nagios o Slack.

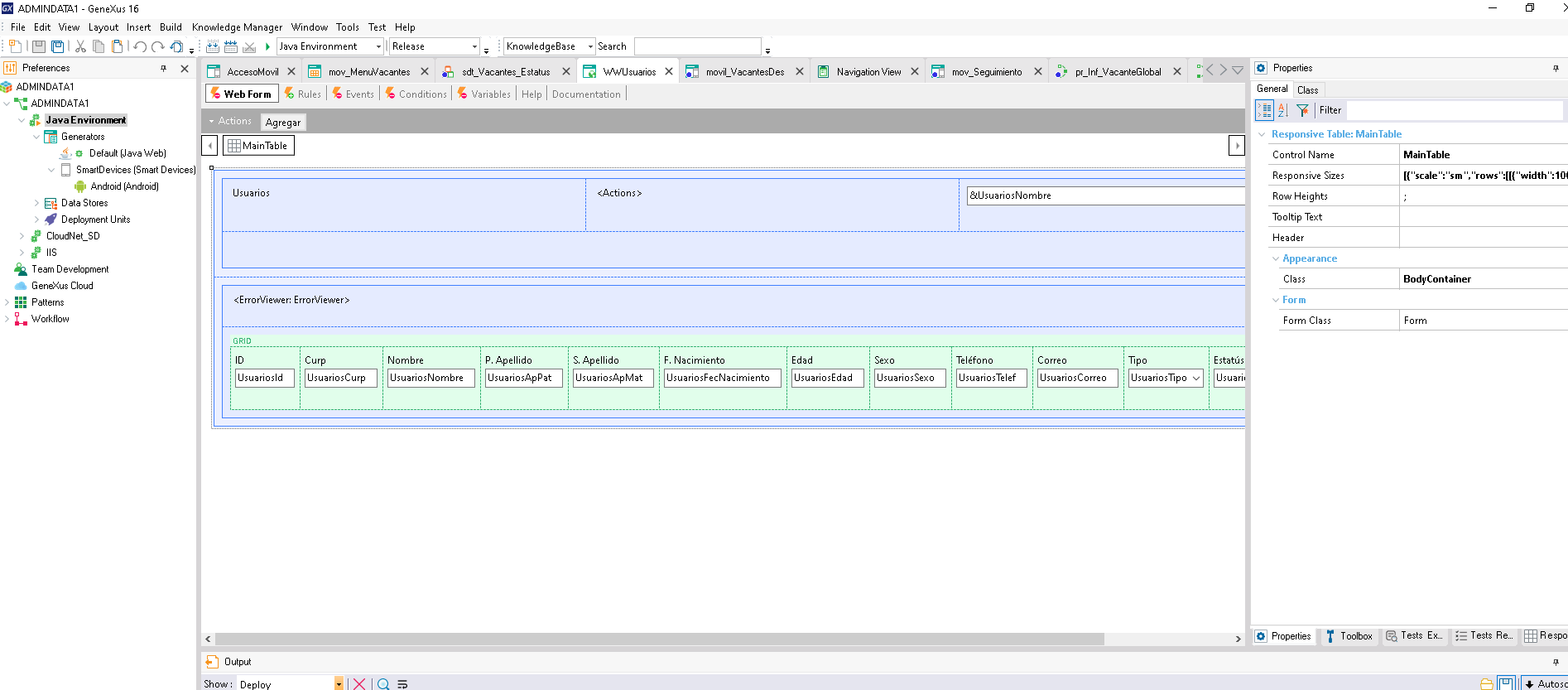
Una vez que tenemos decidido y definido lo que tenemos que hacer (análisis y requerimientos) es momento de que se seleccionen las herramientas de desarrollo que usaremos para este proyecto las cuales deben ser instaladas y configuradas de la manera más adecuada y eficaz para el proyecto a desarrollar para lo cual se instalara “el ambiente o entorno local del desarrollador”, en nuestro caso se usaran las siguientes herramientas:

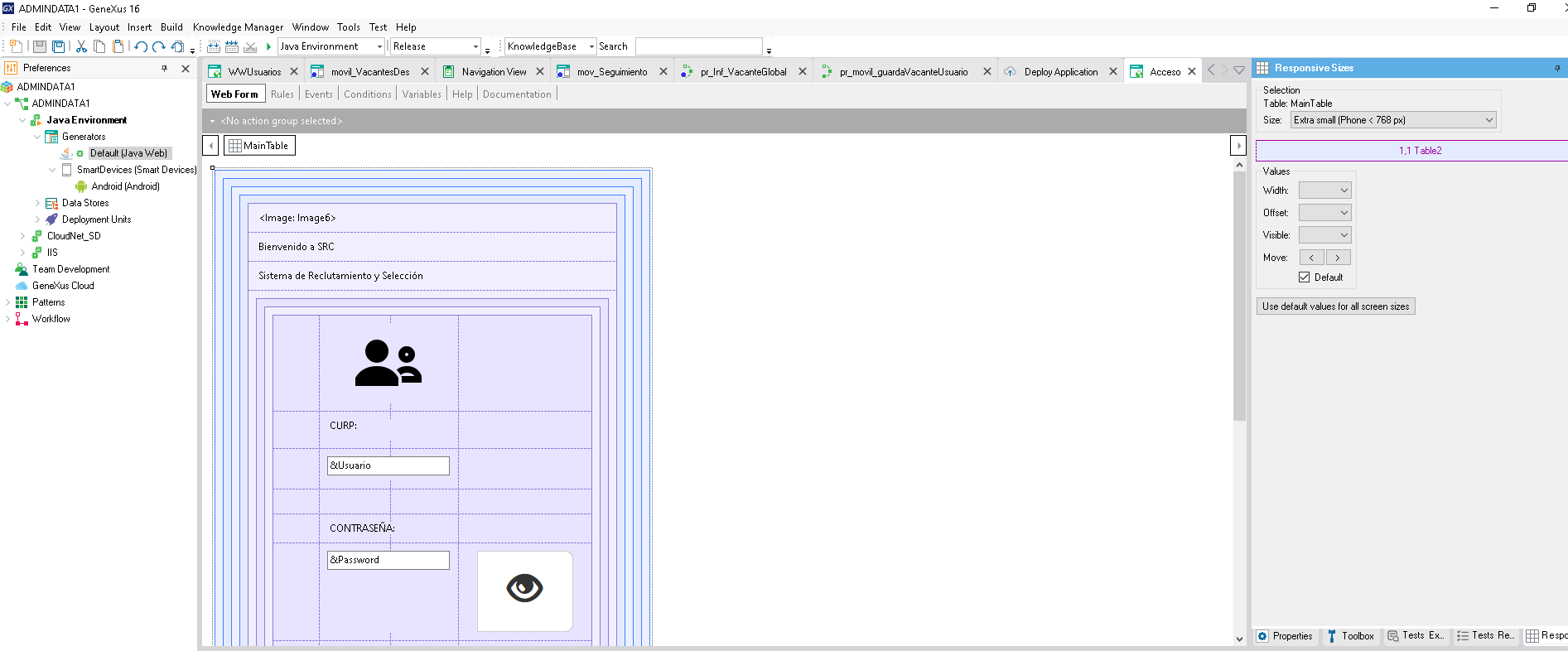
# IDE GENEXUS

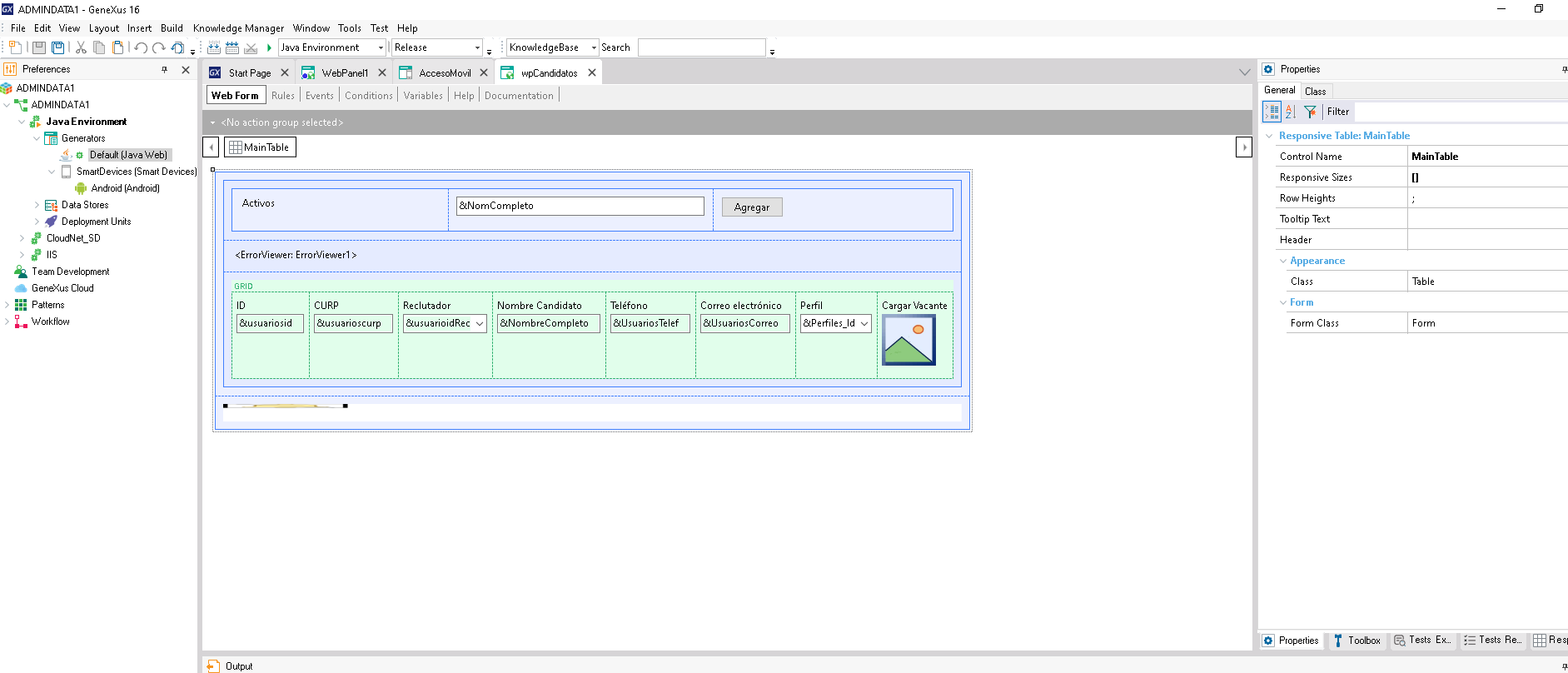
Un IDE (Integrated Development Environment) llamado **GENEXUS** es una interfaz de desarrollo, una herramienta que nos permite realizar desde las tareas más básicas, como codificar, hasta otras más avanzadas o accesorias, como hacer cambios en múltiples ficheros al mismo tiempo, nos permitirá el desarrollo de la aplicación Web, así como la aplicación Móvil.

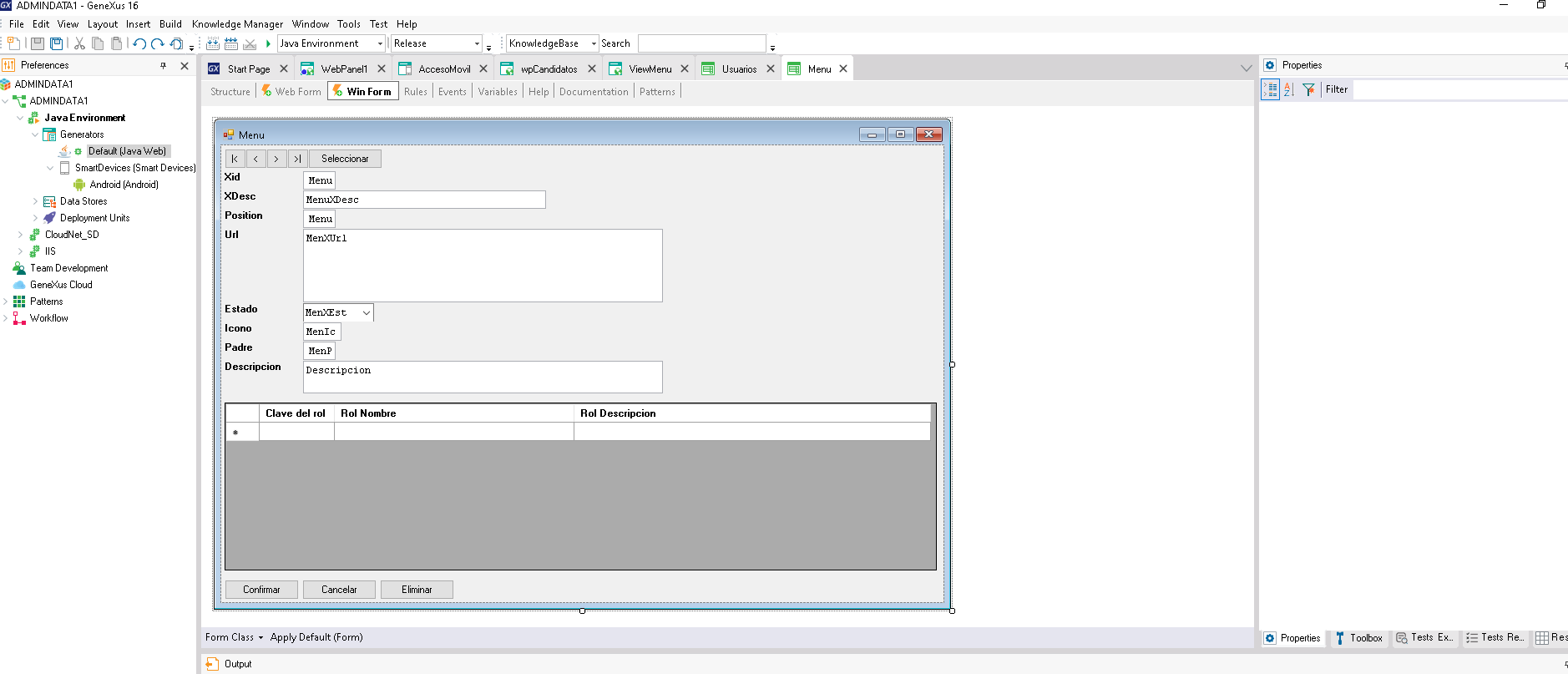
Para este proyecto se selecciona java por dos razones específicas, la primera de ellas y más importante es que el cliente otorgara el Framework Genexus para la elaboración del sistema, el cual trabaja con Java, se tomara un curso de inicio y comprensión de su uso, así mismo el personal técnico del equipo tiene conocimientos sólidos en java, lo cual permitirá un avance significativo en el proyecto mismo que será entregado y evaluado por el cliente, GeneXus es una plataforma de desarrollo de software que genera aplicaciones web, Android e iOS desde una única fuente, lo cual permite abarcar todo lo que necesitamos para la entrega de este proyecto, moderniza los sistemas heredados para reducir los costos de mantenimiento e incorporar cambios rápidamente.

**Desarrollo del sistema Admindata**



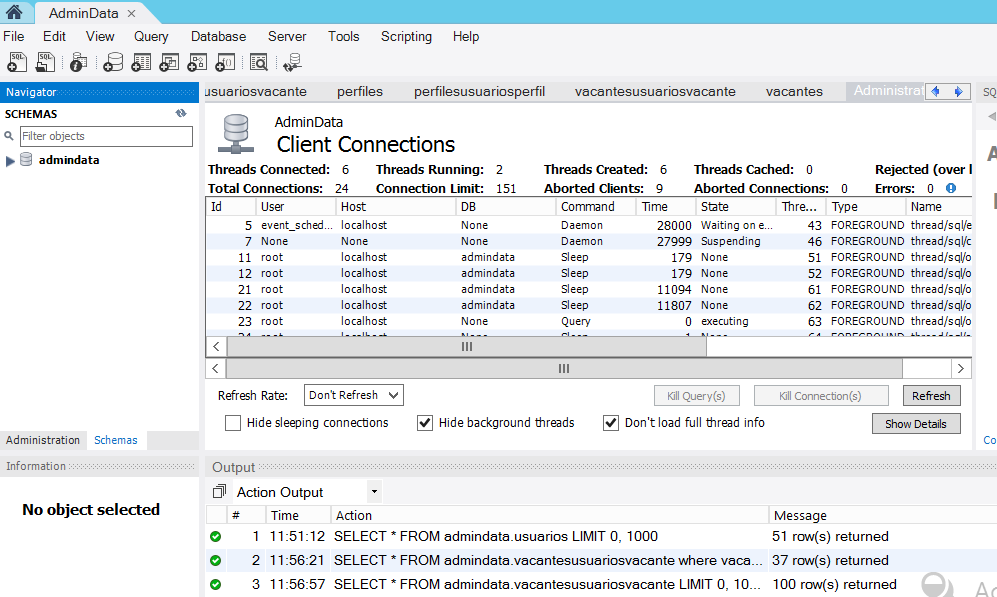


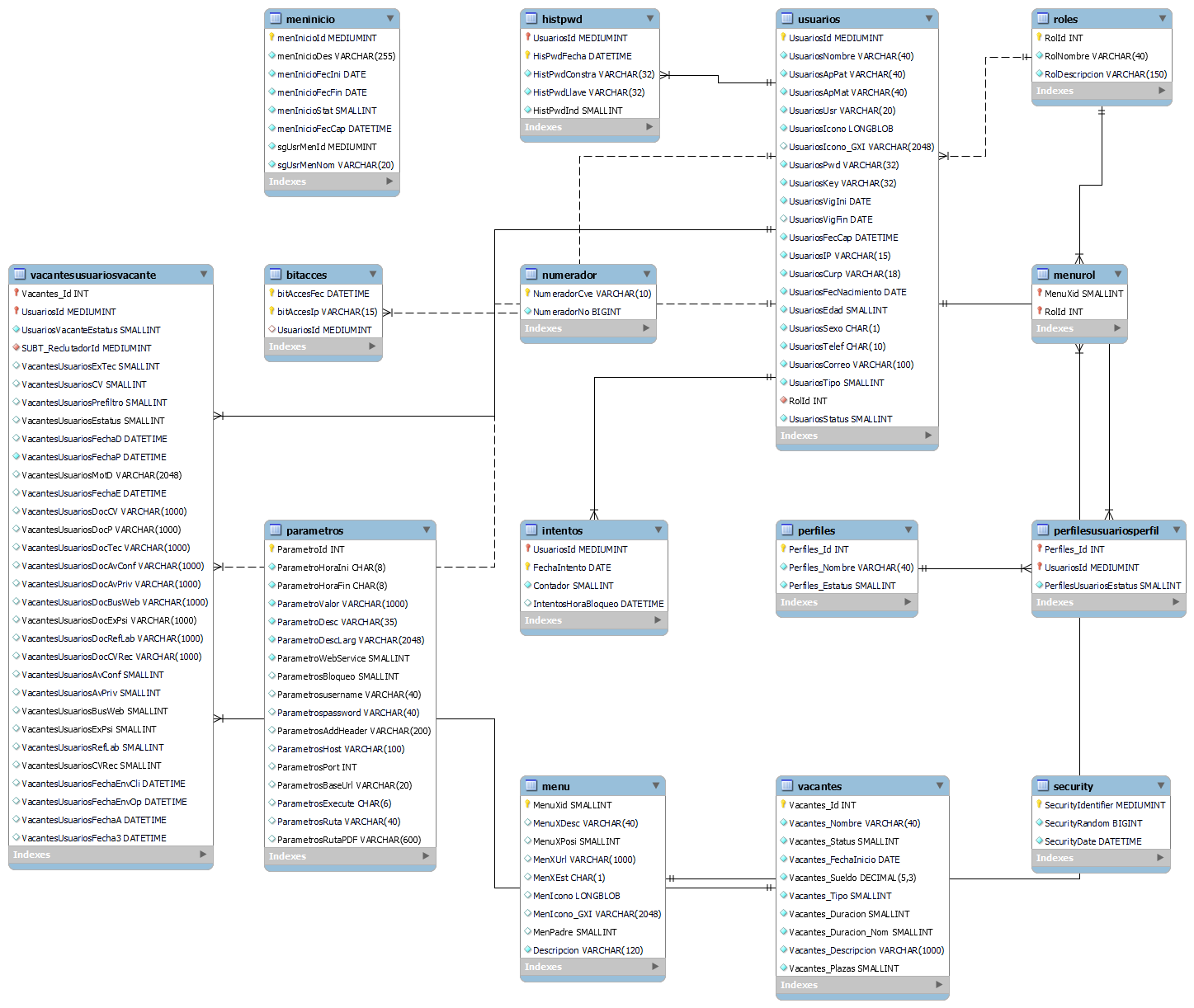


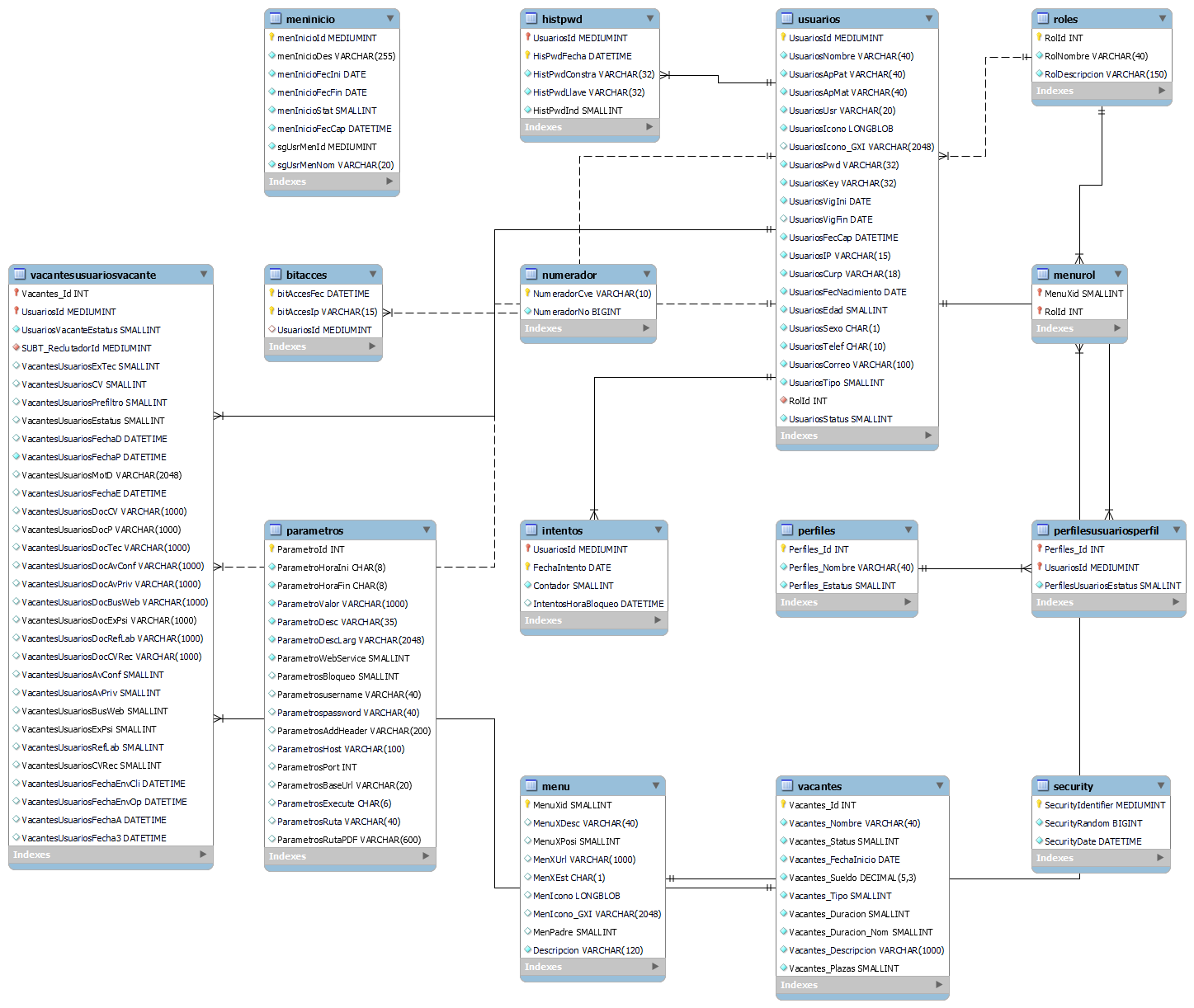


# DATABASE MYSQL

El gestor de base de datos que seleccionamos fue solicitado por el cliente ya que tiene varios desarrollos que ya lo usan, así mismo el equipo de trabajo acepto que era la mejor opción ya que es Multiplataforma: Para S.O. como Windows, Linux y Mac y ya se dispone de un servidor de producción donde está instalado, lo hemos usado en distintas prácticas, no se requiere ser un experto para su configuración ya que incluye unas características mínimas que nos sirven ampliamente para nuestras aplicaciones sin tener que recurrir a un DBA especializado para que administre la base datos.

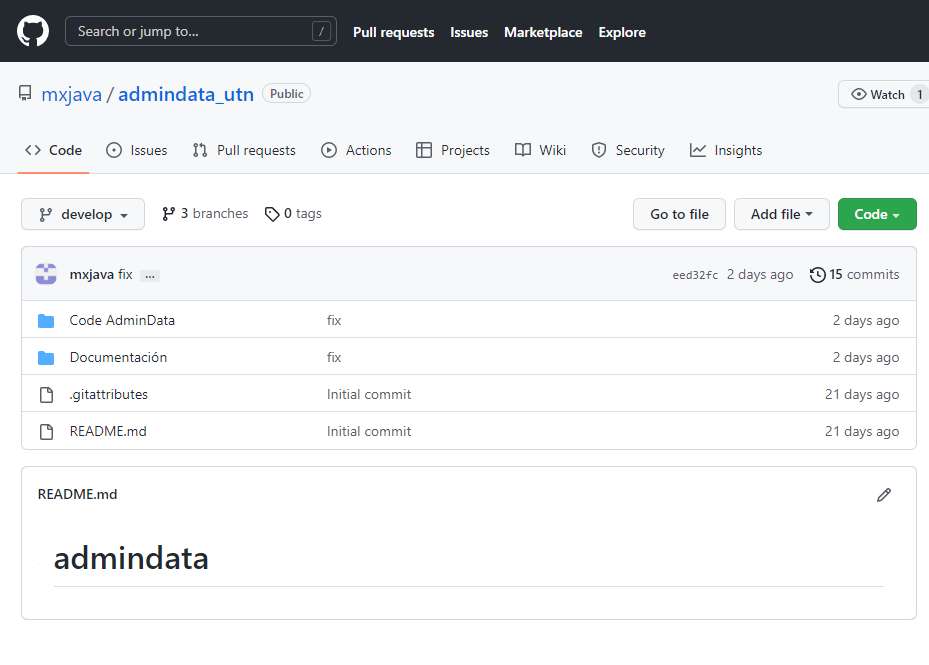






# GIT HUB

<https://github.com/mxjava/admindata_utn>



Git es un sistema de control de versiones distribuido, de código abierto y gratuito, es decir, donde vamos a guardar el código fuente de nuestras aplicaciones y todos los cambios que se hagan, cada participante tiene en su local el histórico, etiquetas y ramas. La gran ventaja de esto es que no necesitas estar conectado a la red para hacer cualquier operación contra el repositorio, por lo que el trabajo es mucho más rápido y tiene menos dependencias.

Git es un sistema de control de revisión distribuido, escalable y rápido con un conjunto de comandos inusualmente rico que proporciona operaciones de alto nivel y acceso completo a los elementos internos. Hoy en día, Git es, con diferencia, el sistema de control de versiones moderno más utilizado del mundo. Git es un proyecto de código abierto maduro y con un mantenimiento activo que desarrolló originalmente Linus Torvalds, el famoso creador del kernel del sistema operativo Linux, en 2005. Un asombroso número de proyectos de software dependen de Git para el control de versiones, incluidos proyectos comerciales y de código abierto. Los desarrolladores que han trabajado con Git cuentan con una buena representación en la base de talentos disponibles para el desarrollo de software, y este sistema funciona a la perfección en una amplia variedad de sistemas operativos e IDE (entornos de desarrollo integrados).

Git, que presenta una arquitectura distribuida, es un ejemplo de DVCS (sistema de control de versiones distribuido, por sus siglas en inglés). En lugar de tener un único espacio para todo el historial de versiones del software. Además de contar con una arquitectura distribuida, Git se ha diseñado teniendo en cuenta el rendimiento, la seguridad y la flexibilidad.

**Rendimiento**

Las características básicas de rendimiento de Git son muy sólidas en comparación con muchas otras alternativas. La confirmación de nuevos cambios, la ramificación, la fusión y la comparación de versiones anteriores se han optimizado en favor del rendimiento. Los algoritmos implementados en Git aprovechan el profundo conocimiento sobre los atributos comunes de los auténticos árboles de archivos de código fuente, cómo suelen modificarse con el paso del tiempo y cuáles son los patrones de acceso.

A diferencia de algunos progBranch de software de control de versiones, Git no se deja engañar por los nombres de los archivos a la hora de determinar cuál debería ser el almacenamiento y el historial de versiones del árbol de archivos; en lugar de ello, se centra en el contenido del propio archivo. Al fin y al cabo, los archivos de código fuente se cambian de nombre, se dividen y se reorganizan con frecuencia. El formato de objeto de los archivos del repositorio de Git emplea una combinación de codificación delta (que almacena las diferencias de contenido) y compresión, y guarda explícitamente el contenido de los directorios y los objetos de metadatos de las versiones.

Su arquitectura distribuida también permite disfrutar de importantes ventajas en términos de rendimiento.

**Seguridad**

Git se ha diseñado con la principal prioridad de conservar la integridad del código fuente gestionado. El contenido de los archivos y las verdaderas relaciones entre estos y los directorios, las versiones, las etiquetas y las confirmaciones, todos ellos objetos del repositorio de Git, están protegidos con un algoritmo de hash criptográficamente seguro llamado "SHA1". De este modo, se salvaguarda el código y el historial de cambios frente a las modificaciones accidentales y maliciosas, y se garantiza que el historial sea totalmente trazable.

Algunos otros sistemas de control de versiones carecen de protección contra las modificaciones ocultas realizadas con posterioridad, algo que puede suponer una grave vulnerabilidad de seguridad de la información para cualquier organización que se base en el desarrollo de software.

**Flexibilidad**

Uno de los objetivos clave de Git en cuanto al diseño es la flexibilidad. Git es flexible en varios aspectos: en la capacidad para varios tipos de flujos de trabajo de desarrollo no lineal, en su eficiencia en proyectos tanto grandes como pequeños y en su compatibilidad con numerosos sistemas y protocolos.

Git se ha ideado para posibilitar la ramificación y el etiquetado como procesos de primera importancia (a diferencia de SVN) y las operaciones que afectan a las Branch y las etiquetas (como la fusión o la reversión) también se almacenan en el historial de cambios. No todos los sistemas de control de versiones ofrecen este nivel de seguimiento.

**Implementando Git con GitHub**

GitHub es una plataforma de desarrollo colaborativo, también llamado forja. Es decir, una plataforma enfocada hacia la cooperación entre desarrolladores para la difusión y soporte de su software (aunque poco a poco se ha ido usando para otros proyectos más allá del software).

Como su propio nombre indica, se apoya sobre el sistema de control de versiones Git. Así, se puede operar sobre el código fuente de los progBranch y llevar un desarrollo ordenado. Además, esta plataforma está escrita en Ruby on Rails.

**Entorno**

Para el siguiente tutorial de configuración se está utilizando el siguiente entorno:

Hardware:

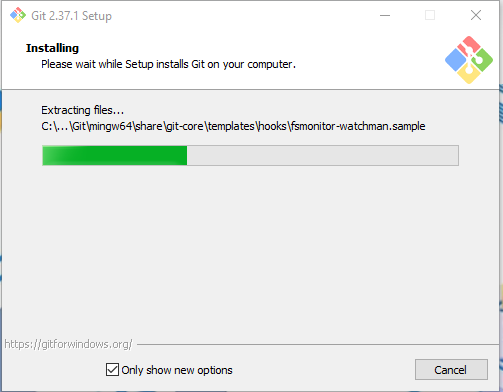
Acer Aspire 3 (AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx 2.00 GHz).

Sistema Operativo: Windows 10

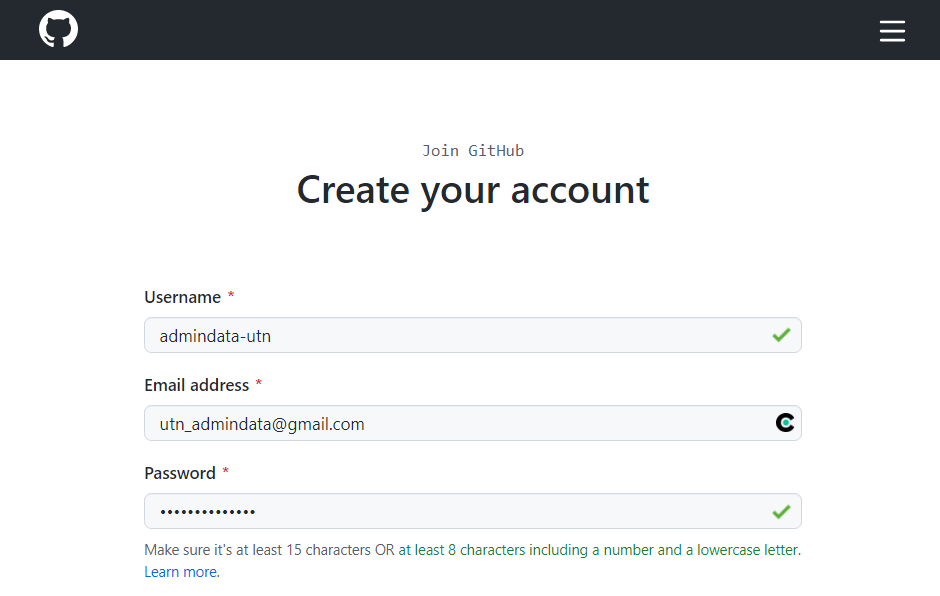
Git 2.37

**Paso 1: Instalar Git y crear una cuenta en GitHub**

Debido a que nuestro entorno de desarrollo se ejecuta sobre en Windows, se instalará la versión oficial de Git para este sistema operativo, que se encuentra en el enlace: https://git-scm.com/download/win

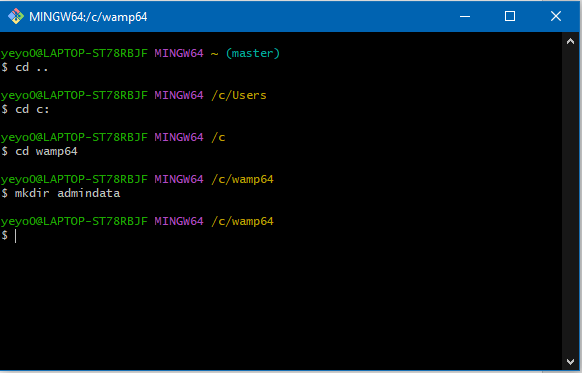


Una vez instalado Git, procedemos a crear una cuenta en GitHub



**Paso 2: Configuración de Git y creación de un repositorio local del Proyecto**

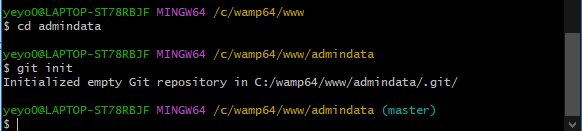
Primeramente, abrimos la terminal de Git y nos situamos en el directorio en el que se desarrollará el proyecto



Una vez hecho esto, tenemos que configurar Git agregando nuestro nombre y nuestro email para que Git pueda “firmar” nuestros commits. De esta forma estaremos identificando que ese commit.

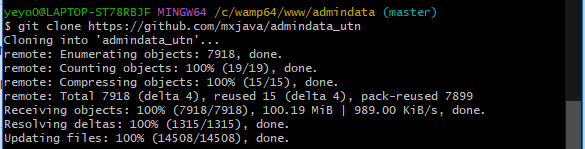


Una vez ubicados en el directorio raíz del proyecto, escribimos el comando “git init” que nos creará un nuevo repositorio

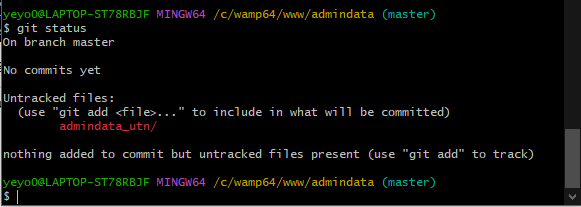


**Paso 3: Clonar un repositorio existente**

Debido a que ya llevamos un avance en el proyecto, ya existe otro repositorio en el cual se ha trabajado. Para poder trabajar en ese repositorio, podemos clonarlo en nuestro entorno local utilizando el comando “git clone”



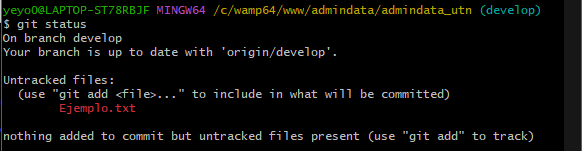
Esto nos descargará todos los archivos que contiene el proyecto para poder trabajar en él, esto desde luego nos genera un nuevo directorio llamado “admindata\_utn”, lo podemos comprobar con un “git status”.



Básicamente Git nos está avisando que encontró una nueva carpeta en el directorio, pero que no la a agregado al repositorio hasta que se lo indiquemos a Git a través de un commit.

**Paso 4: Crear un commit**

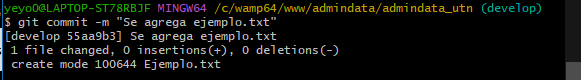
Para este paso, entraremos al directorio del repositorio clonado. Y una vez que hayamos trabajado sobre el proyecto, podemos observar los cambios a través de un git status.



Agregamos el archivo al repositorio, para así crear nuestro primer commit



Agregado el archivo, ya podemos ejecutar nuestro primer commit, el cuál estará identificado por el autor del cambio y una descripción de lo que se realizó

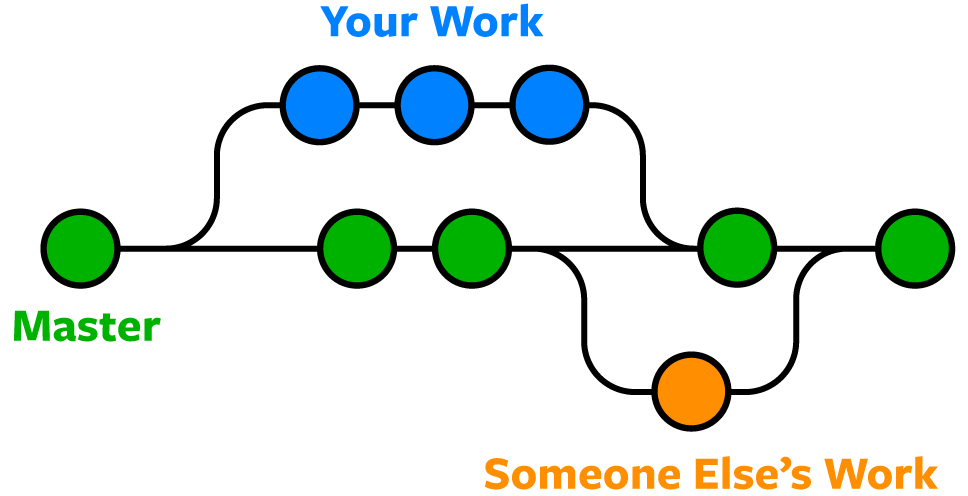


**Paso 5: Crear un Branch o Rama**

Ahora que ha realizado un commit, intentemos algo un poco más avanzado.

Digamos que deseamos crear una función nueva, pero preocupa realizar cambios en el proyecto principal mientras desarrolla la función. Aquí es donde entran las Branch de git.

Las Branch le permiten avanzar y retroceder entre los 'estados' de un proyecto. Los documentos oficiales de git describen las Branch de esta manera: 'Branch en Git es simplemente un puntero móvil ligero a una de estas confirmaciones'. Por ejemplo, si desea agregar una nueva página a su sitio web, puede crear una nueva Branch solo para esa página sin afectar la parte principal del proyecto. Una vez que haya terminado con la página, puede combinar los cambios de su Branch en la Branch principal. Cuando crea una nueva Branch, Git realiza un seguimiento de la confirmación de la que se 'ramificó' su Branch, por lo que conoce el historial detrás de todos los archivos.



Para crear un nuevo Branch, únicamente colocamos “git Branch <nombre rama>”

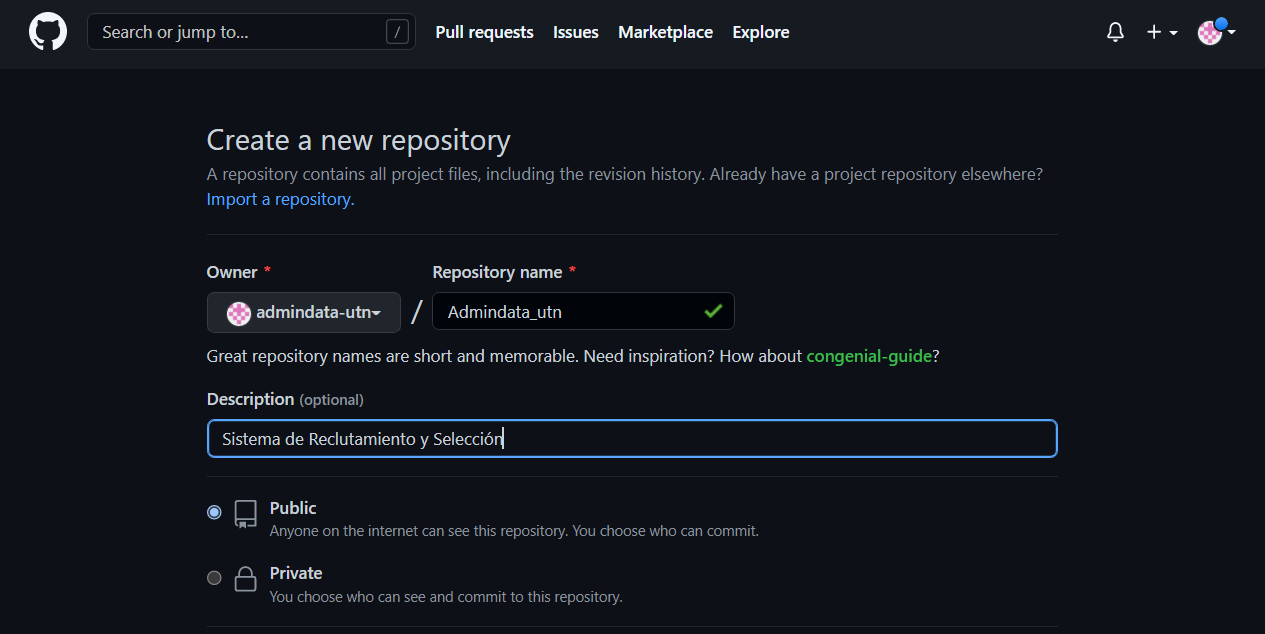


Podremos verificar que se ha creado, con el comando “git branch”



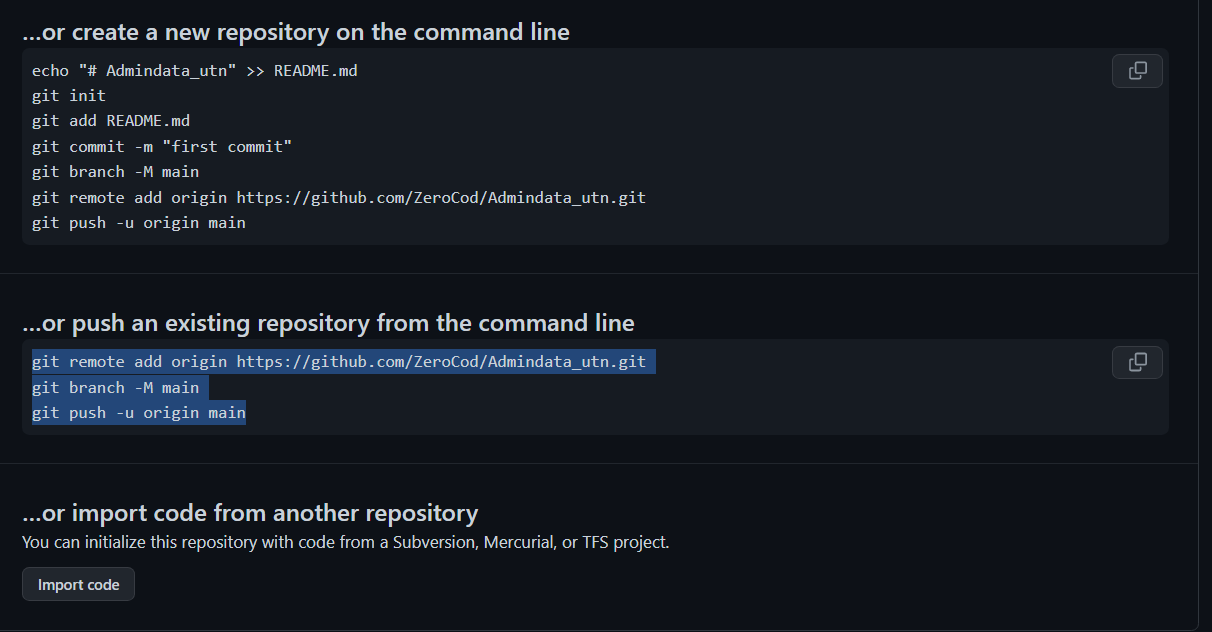
Esto nos puede servir mucho, cuando queremos hacer cambios experimentales, sin afectar la rama principal.

**Paso 6: Crear repositorio en GitHub y subir repositorio local**

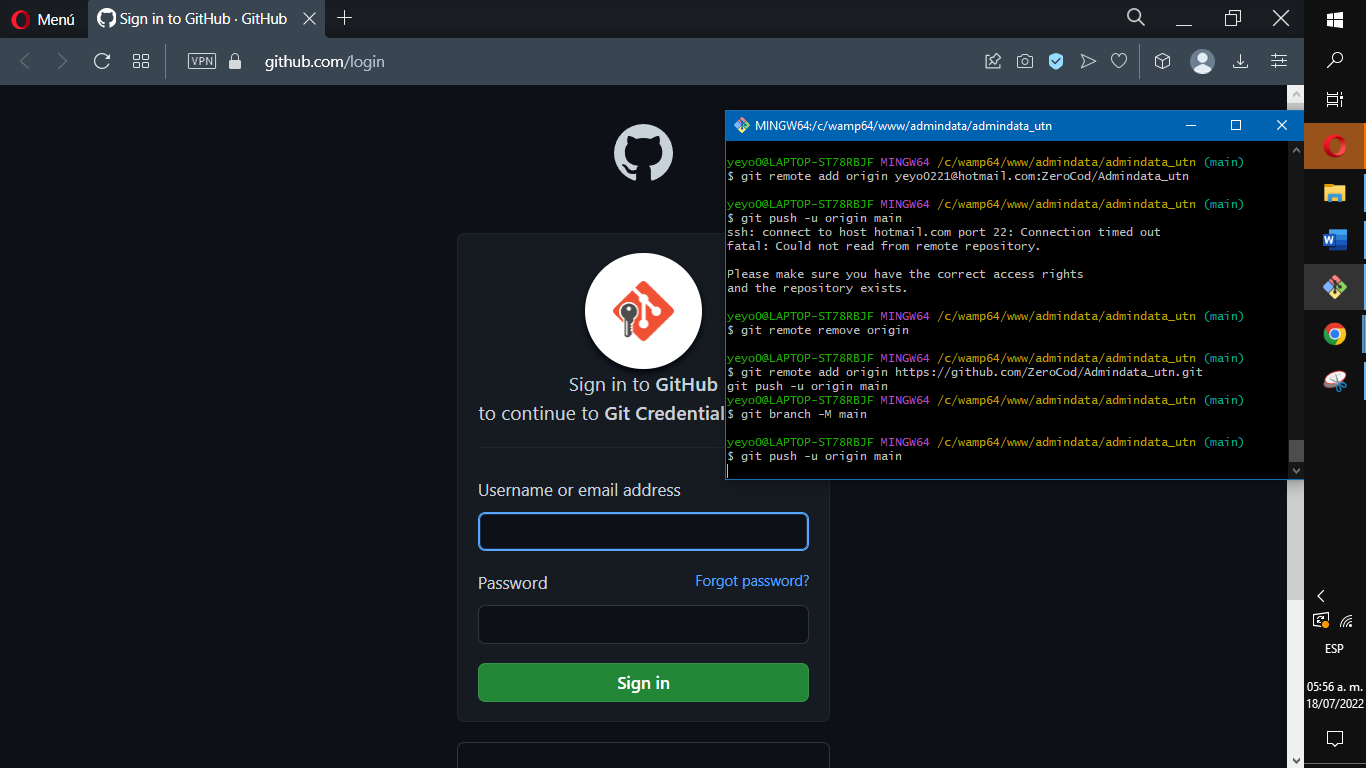


Terminado de completar la información, presionamos el botón 'Crear repositorio'.

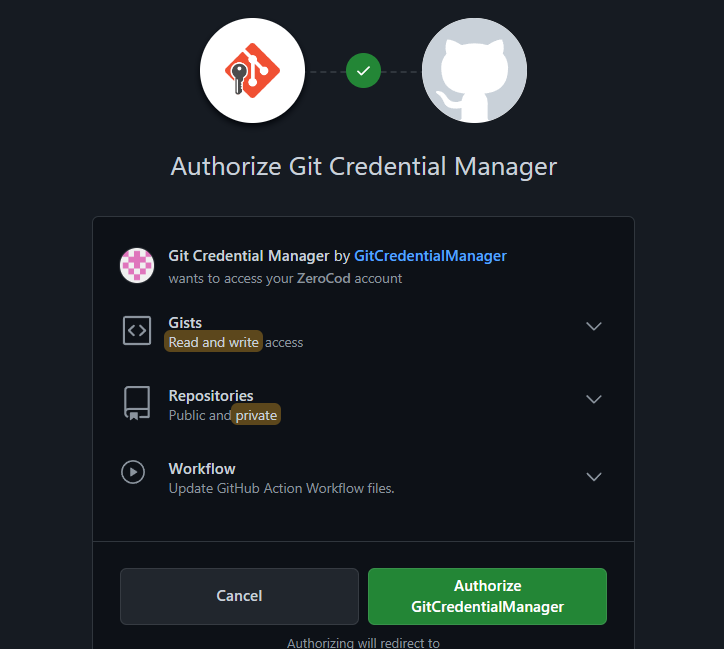
GitHub preguntará si deseamos crear un nuevo repositorio desde cero o si desea agregar un repositorio que haya creado localmente. En este caso, dado que ya hemos creado un nuevo repositorio localmente, queremos enviarlo a GitHub, así que siga la sección '... o envíe un repositorio existente desde la línea de comando':



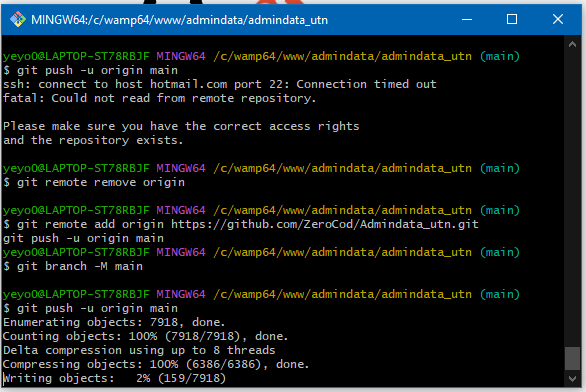
Copiamos, y pegamos en la terminal.



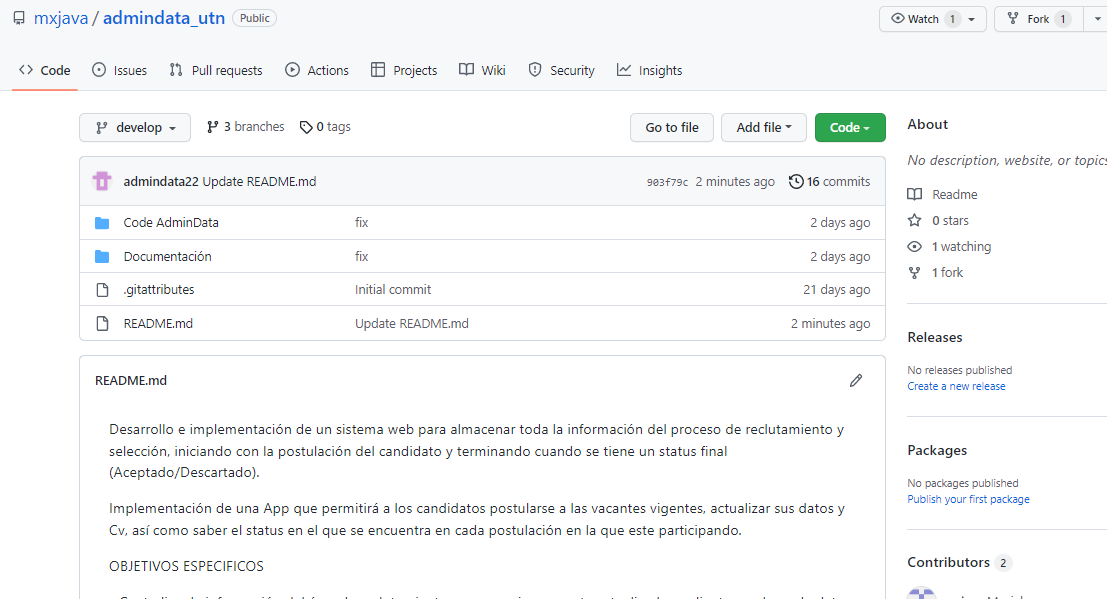
Git nos pedirá confirmar nuestra sesión en GitHub para poder subir nuestro repositorio local. Y autorizamos “Git Credential Manager”



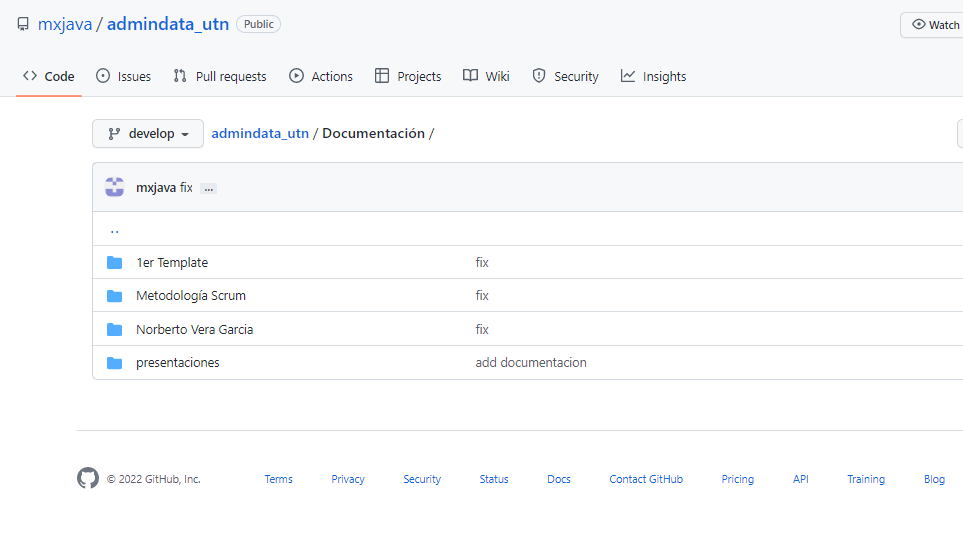
Hecho esto, se comenzará a subir nuestro repositorio local en el repositorio en línea de GitHub



En la página directamente se encontrara la información del proyecto:



En este repositorio tenemos tanto el sistema como la documentación correspondiente.



# JIRA

<https://admindata.atlassian.net/jira/software/projects/AD/boards/1>

Jira es la herramienta de desarrollo de software número 1 utilizada por los equipos ágiles. Millones eligen Jira para planificar y construir grandes productos. Jira ayuda a los equipos a entregar valor a los clientes al lanzarlos antes, con más frecuencia y de manera más iterativa.

Los equipos utilizan Jira para convertir las críticas en nuevas características y una mayor satisfacción del cliente. Con Jira, los equipos obtienen visibilidad de los objetivos a largo plazo, el estado del trabajo y la información de lanzamiento en tiempo real. A medida que los equipos se adaptan a los cambios del mercado, Jira ayuda a garantizar que los procesos evolucionen a la misma velocidad.

**Jira se puede instalar en:**

Cloud, SaaS, Web, Mac (desktop), Windows (desktop), Linux (desktop), Windows (local), Linux (local), Android (móvil), iPhone (móvil), iPad (móvil)



Jira es una herramienta para la administración de tareas de un proyecto, el seguimiento de errores e incidencias y para la gestión operativa de proyectos. La herramienta fue desarrollada por Atlassian. Inicialmente Jira se utilizó para el desarrollo de software, sirviendo de apoyo para la gestión de requisitos, seguimiento del estado de desarrollo y más tarde para la gestión de errores. Jira puede ser utilizado para la gestión y mejora de los procesos, gracias a sus funciones para la organización de flujos de trabajo.

Jira tiene diferentes áreas de responsabilidad y, por tanto, puede utilizarse de forma flexible. La herramienta web se utiliza con especial frecuencia en el ámbito del desarrollo de software, aunque su uso en áreas no técnicas también puede ser útil. Jira funciona como una herramienta de visión general y de planificación al mismo tiempo y, por lo tanto, ayuda a optimizar el flujo de trabajo dentro de un equipo. El software está dirigido tanto a los administradores como a los usuarios. Aquí se distribuyen las tareas, se comparten los estados intermedios y se identifican y remedian rápidamente las posibles fuentes de error. Para ello, se crean tickets para tareas individuales, que se asignan y se procesan.

Jira es especialmente adecuado para los métodos de gestión de proyectos Scrum y Kanban. Los proyectos pequeños, medianos e incluso muy extensos pueden llevarse a cabo de forma más eficaz con el software. Una función de información detallada también garantiza, en primer lugar, la documentación necesaria de un proyecto y, en segundo lugar, permite que los nuevos empleados se incorporen a un equipo existente de forma rápida y sencilla. Además del producto principal, existen numerosos plugins y otras soluciones de software de Atlassian para Jira, que pueden integrarse de forma rápida y sencilla para personalizar y optimizar el flujo de trabajo de uno o varios trabajos.

**Versiones de Jira**

**Jira Core es el todoterreno**. Con esta variante se pueden planificar proyectos y asignar tareas. Los gestores de proyectos y los jefes de equipo tienen una visión clara del conjunto del proyecto con todos los pasos intermedios, y los empleados tienen sus tareas actuales detalladas, incluidos los plazos de entrega de las mismas. La posibilidad de publicar comentarios también hace que la comunicación sea más eficiente y que el número de correos electrónicos, reuniones o llamadas telefónicas se reduzca al mínimo, incluso si los equipos no trabajan desde un mismo lugar.

**Jira Software** se dirige explícitamente a los equipos encargados de desarrollar nuevos programas informáticos. Además de las ventajas de la versión Core, esta variante también ofrece la posibilidad de crear tableros especiales de Scrum o Kanban. También se pueden establecer sprints de forma rápida y sencilla y planificar los backlogs. Especialmente para un enfoque ágil, donde los equipos pueden reaccionar directamente a los nuevos desarrollos y requisitos, el software Jira es un complemento que merece la pena.

La tercera opción es **Jira Service Desk**. Simplifica especialmente el contacto con los clientes o los empleados de una empresa más grande. Aquí se pueden crear y asignar tickets para consultas, quejas o mensajes. Esto no solo simplifica el flujo de trabajo, sino que las preguntas y críticas se responden mucho más rápido, lo que puede mejorar la satisfacción del cliente.

Para el proyecto de Admindata usaremos la versión Jira Software con una cuenta gratuita, la cual se configuro de la siguiente manera:

Entrar a la página de atlassian para crear una cuanta gratuita:



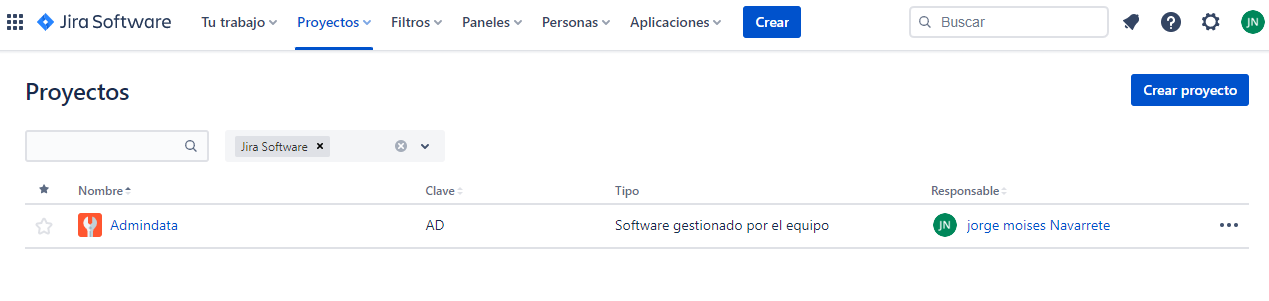
Características de la versión gratuita que usaremos:



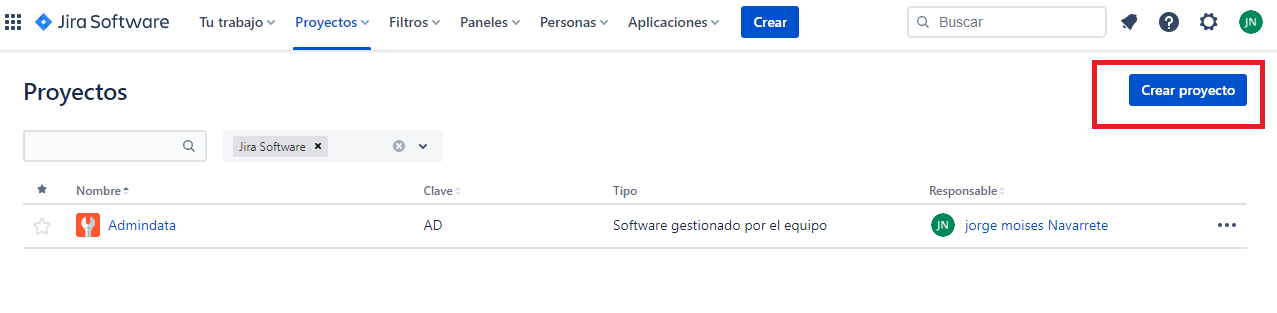
Lo único que se necesita es una cuenta de correo electrónico para hacer el registro, pedirá la validación del mismo para la activación de la cuenta:



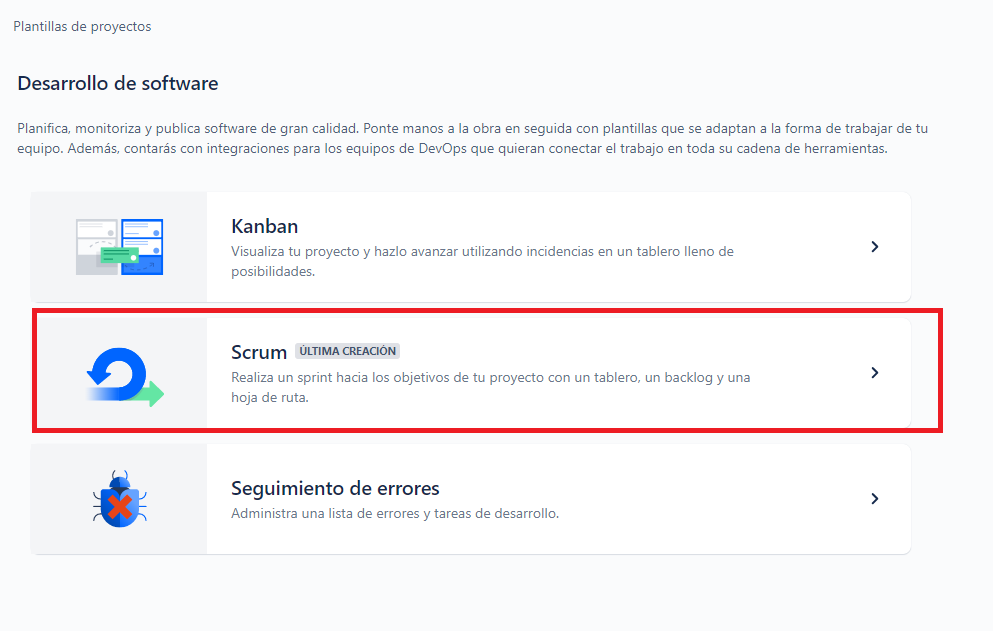
Al iniciar sesión aparecerán los proyectos en los que estamos participando



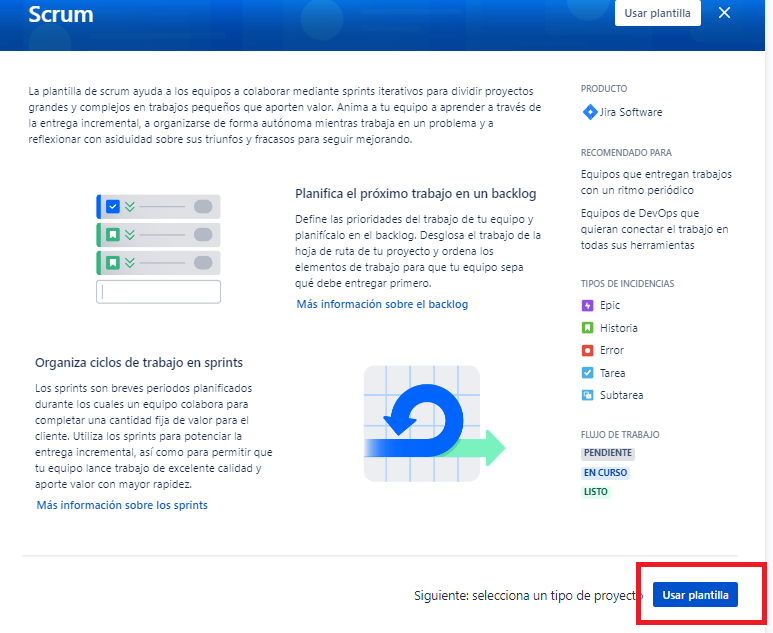
Para crear un proyecto nuevo dar clic en Crear proyecto:



Seleccionar la metodología que se usara, nosotros usamos SCRUM, se creara una plantilla que permitirá Planificar, monitorizar las actividades. El marco de scrum permite a los equipos de software gestionar proyectos complejos creando una cultura de colaboración. El tablero de Jira Scrum es la herramienta que une a los equipos en torno a objetivos únicos y promueve una entrega iterativa e incremental. Si scrum es el marco que guía a los equipos y sus funciones, eventos, artefactos y reglas, el tablero de scrum de Jira es la visualización de su progreso durante el ciclo de desarrollo.



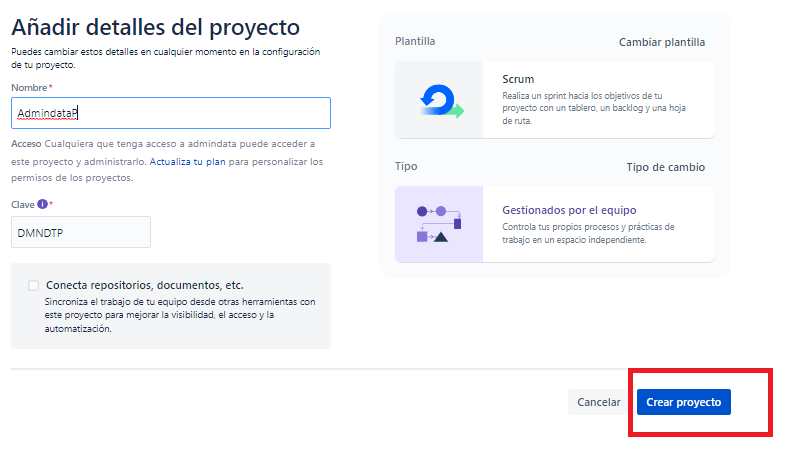
Especificaciones de la plantilla:



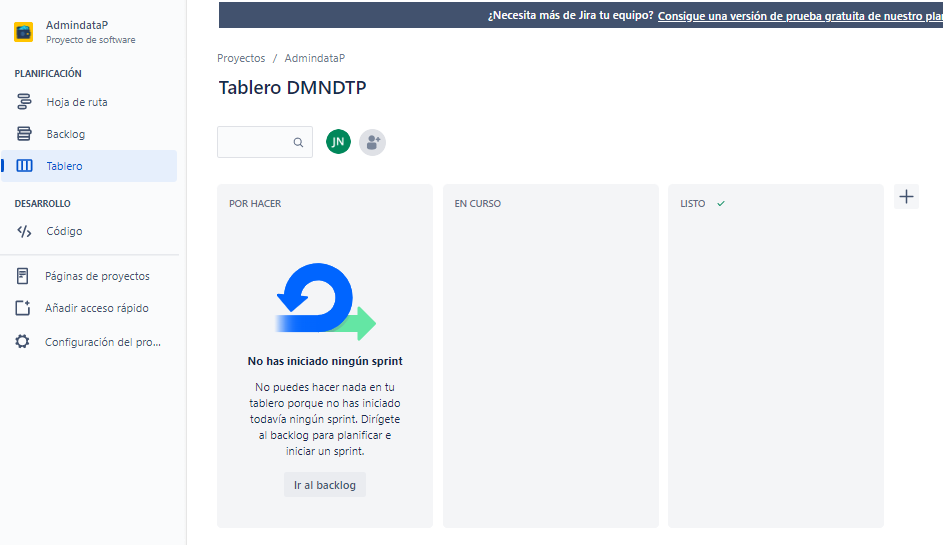
Se da clic en usar plantilla, se elige un tipo de proyecto si será gestionado por el equipo de trabajo o por algún grupo de la empresa, en nuestro caso seleccionamos gestionado por el equipo:



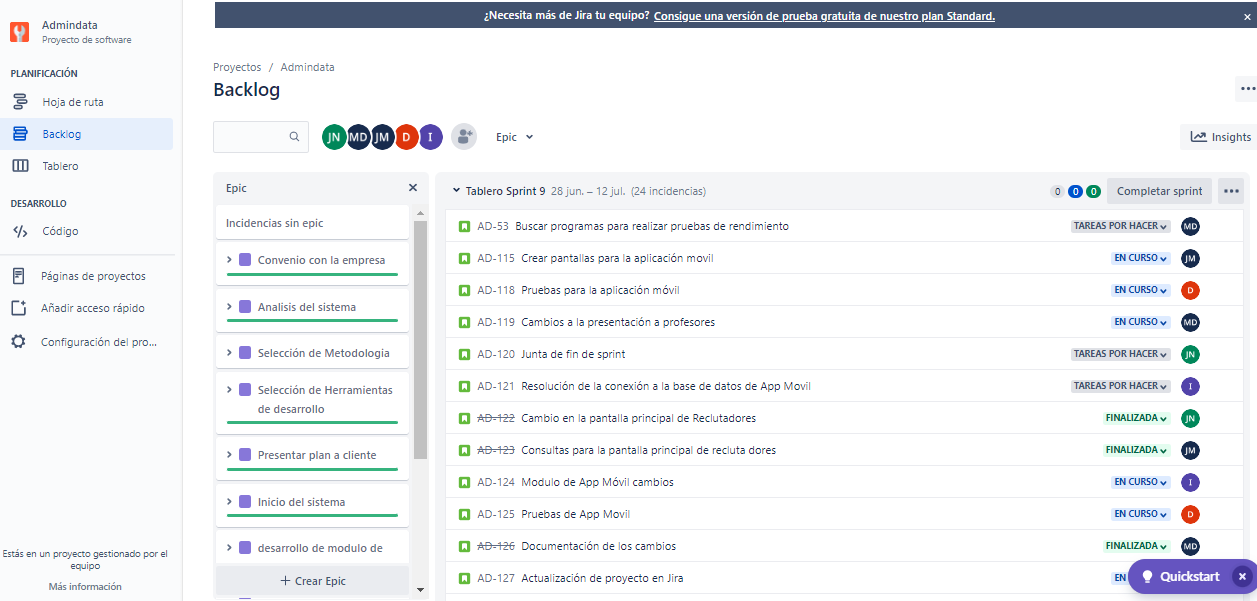
Añadir nombre del proyecto, y dar clic en crear proyecto:



Se crea el proyecto en blanco para empezar a configurar las tareas y los usuarios que las llevaran a cabo.



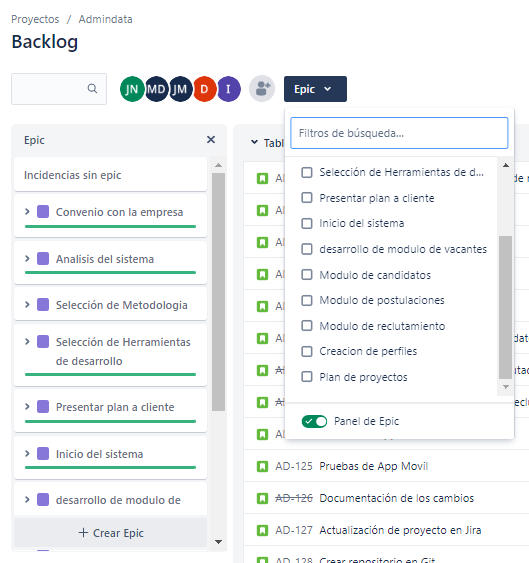
Pantalla principal del proyecto Admindata:



En la pestaña de Backlog podemos encontrar las actividades que se están llevando a cabo, está dividida en las siguientes secciones:

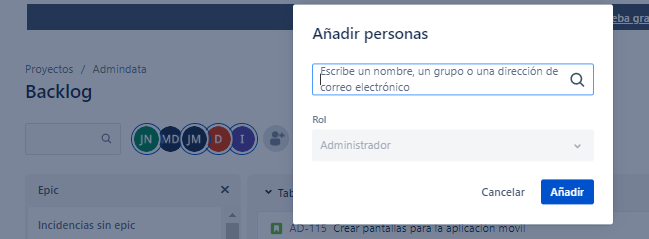
**Épicas (Epic]):**

Un epic es una gran cantidad de trabajo que se puede desglosar en varias historias de menor tamaño, a veces denominadas "incidencias" en Jira. Los epics suelen abarcar a varios equipos en varios proyectos, y puede incluso hacerse un seguimiento de ellos en varios tableros



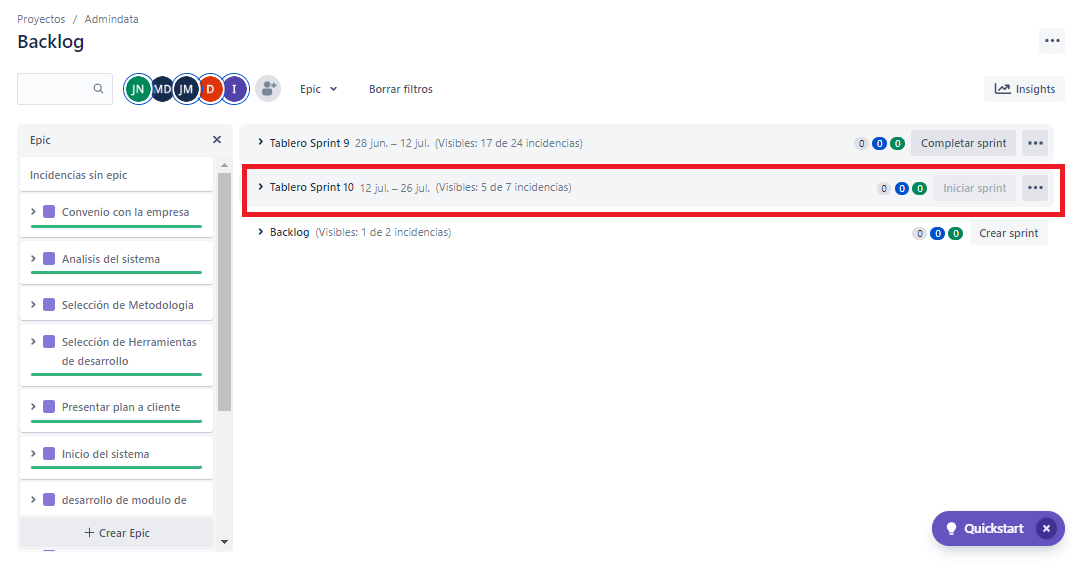
**Usuarios:**

Un usuario de Jira Software es cualquier usuario que pueda iniciar sesión en Jira Software y que exista en Gestión de usuarios. Podemos añadir y eliminar usuarios a medida que cambie nuestro equipo, se necesita el correo del usuario al cual se le asignara un perfil dentro de la herramienta.



**Sprints:**

Un sprint es un periodo de tiempo fijado en un ciclo de desarrollo continuo durante el cual los equipos completan tareas procedentes del backlog del producto. Normalmente, al final del sprint el equipo habrá compilado e implementado un incremento de producto funcional.

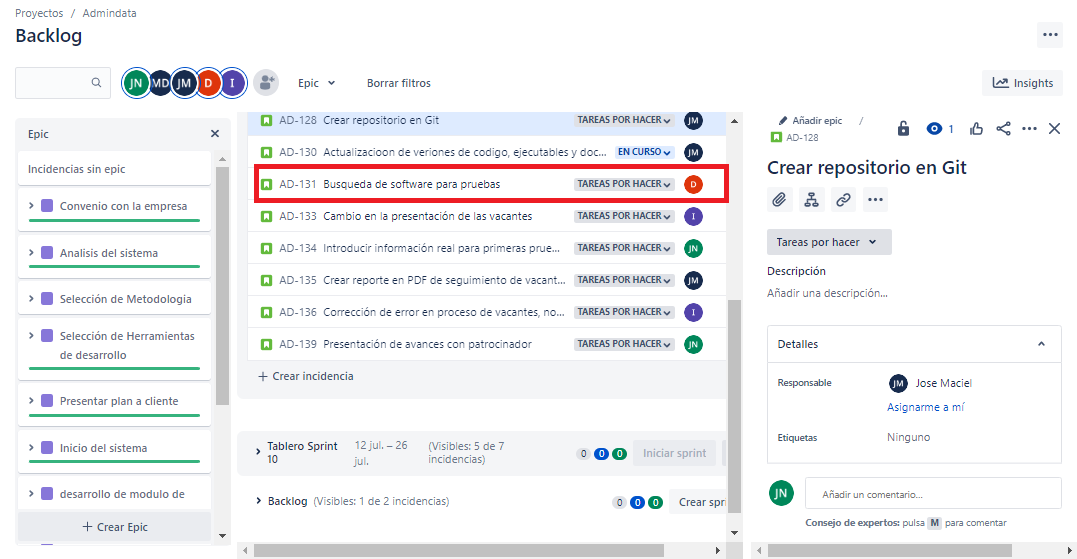


**Incedencias:**

Las incidencias son las piezas fundamentales de los proyectos en Jira Software. Una incidencia puede representar una historia, un epic, un error, una función pendiente de creación o cualquier otra tarea del proyecto.

Como se muestra en la imagen se crea una nueva incidencia en la cual debe indicarse la tarea a realizar, la cual puede tener subtareas y en la vista global aparecerá el status de la tarea, así como el responsable de atenderla.

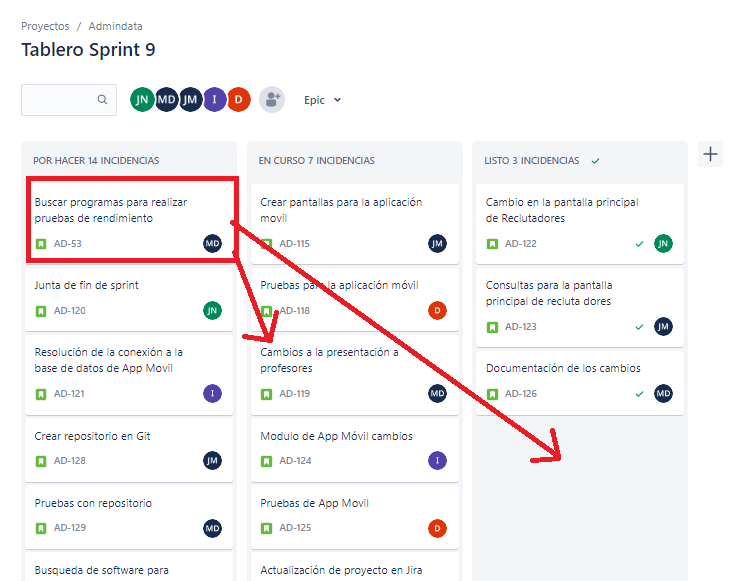
Así mismo del dalo derecho aparecen las especificaciones que se le han dado a cada una de las tareas, cada tarea puede moverse a voluntad para establecer prioridades y se puede cerrar el spring y dejar tareas no realizadas como parte del back log y se mueven en automático al siguiente sprint.



**Tableros:**

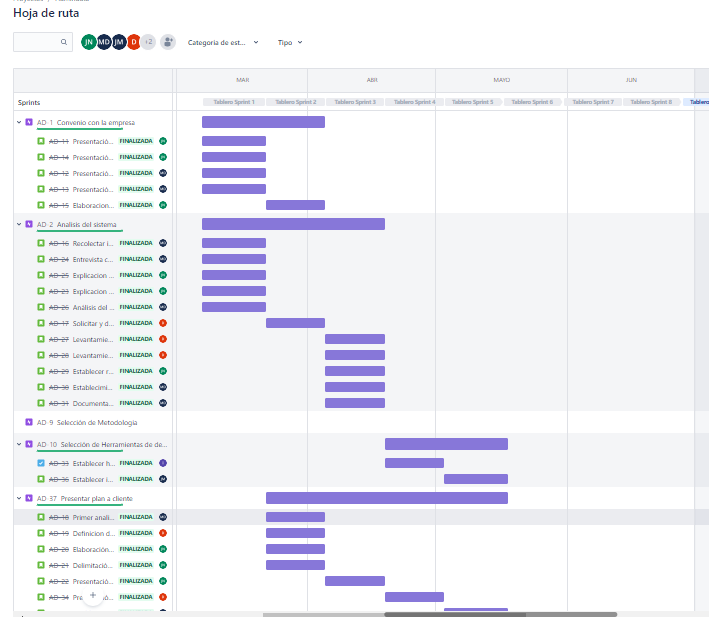
Los tableros de scrum de Jira son una excelente herramienta para supervisar el estado de cada componente de un proyecto; por ejemplo, a la hora de actualizar la información profesional, al momento de realizar una tarea podrá moverse a cada espacio como por hacer, en curso, o finalizado.

Para los equipos distribuidos remotos, un panel puede proporcionar información importante sobre los desarrollos de proyectos tanto esperados como inesperados. También nos ayuda a recordar a los miembros del equipo que tienden a olvidarse de actualizar el progreso de las tareas.



**Hoja de ruta:**

Las hojas de ruta corresponden a un equipo, y son muy prácticas para planificar con varios meses de antelación grandes cantidades de trabajo (a nivel de epics) dentro de un proyecto. La hoja de ruta del proyecto es una descripción general gráfica de alto nivel de los objetivos y entregables del proyecto presentada en un cronograma. A diferencia del plan de proyecto, donde se concretan los detalles, la hoja de ruta debe ser sencilla y sin minucias.



# PLAN DE PRUEBAS

Un caso de prueba responde la pregunta "¿qué voy a probar?". Los casos de prueba se desarrollan para definir las cosas que es necesario validar a fin de asegurar que el sistema funciona correctamente y está construido con un alto nivel de calidad.

Las suites de pruebas también resultan útiles para los siguientes tipos de pruebas:

Pruebas de verificación de compilación: una recopilación de casos de prueba que realizan una validación básica de la mayoría de las áreas funcionales del producto. Las pruebas se ejecutan tras cada compilación de producto y antes de ascender dicha compilación para que la utilice un público mayor.

Pruebas aleatorias: una recopilación de casos de prueba que aseguran la funcionalidad básica del producto. Generalmente, se trata del primer nivel de pruebas que se realiza tras efectuar cambios en el sistema en pruebas.

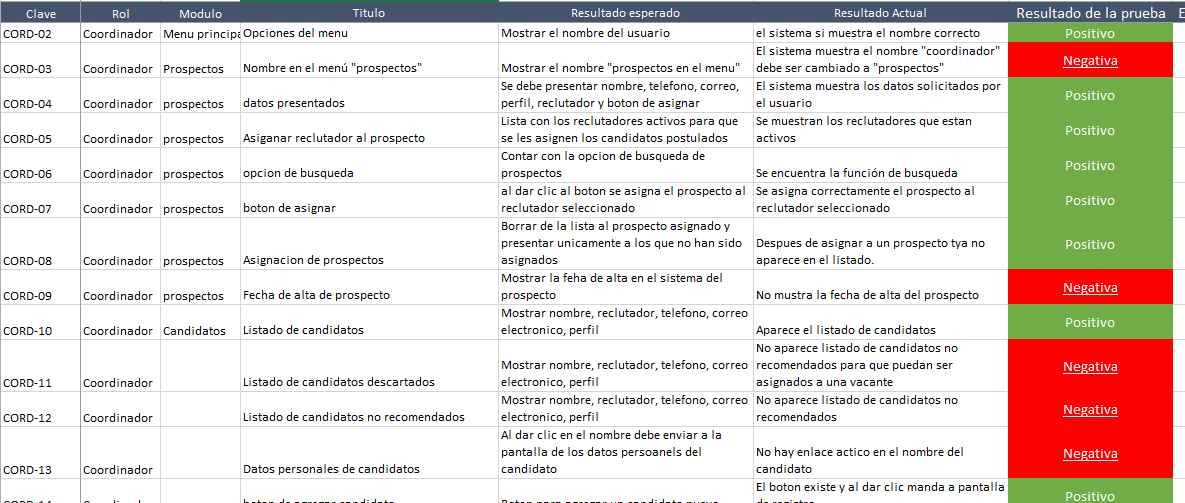
Pruebas de integración integrales: una recopilación de casos de prueba que cruzan los límites entre productos y aseguran que los puntos de integración entre productos se ejercen y se validan.

Pruebas de verificación funcionales: una recopilación de casos de prueba que se centran en una función específica de un producto. La ejecución de este tipo de prueba de un conjunto de pruebas asegura que se han probado varios aspectos de una característica específica.

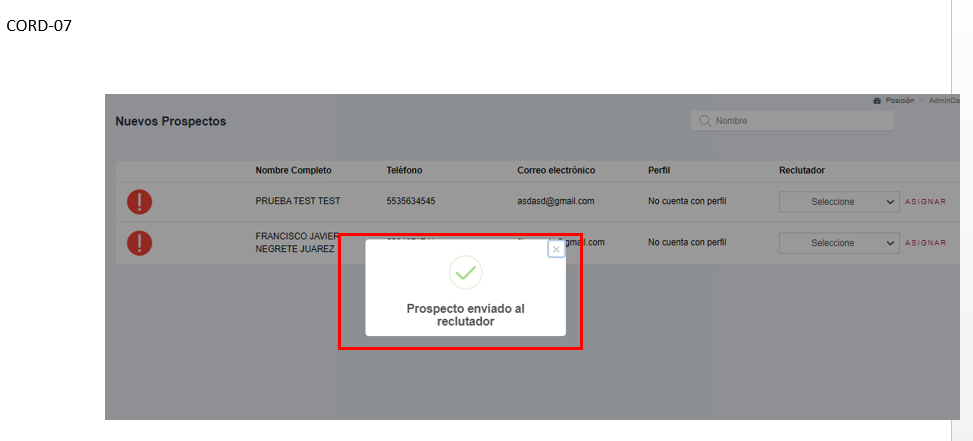
Pruebas de regresión: una recopilación de casos de prueba que se utilizan para realizar una regresión sobre áreas de producto funcionales.

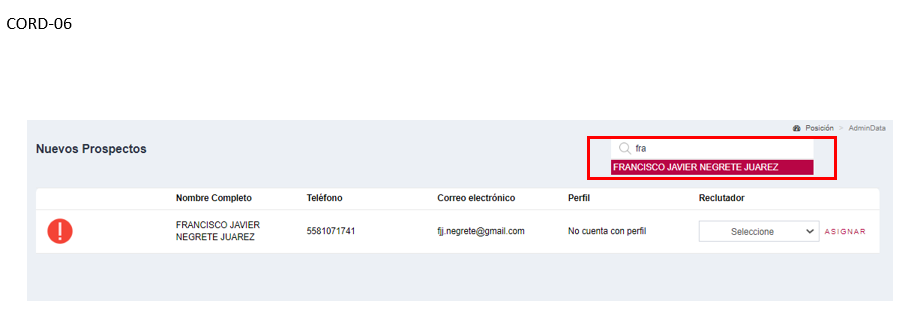
Una suite de pruebas es una colección de casos de prueba que se agrupan para ejecutar pruebas, en admindata realizaremos las pruebas del sistema de forma manual, por tiempo de proyecto y facilidad para realizarlas y comprobar que el sistema funciona correctamente, las pruebas realizadas se hicieron de la siguiente manera:

Pruebas manuales, se elaboró un set de pruebas que se uso durante las diferentes etapas de desarrollo e implementación del sistema, en cada liberación del producto se realizaron pruebas de funcionamiento, se creó un Excel para documentar cada prueba realizada:



De este plan de pruebas se generó evidencia de cada prueba realizada:





Se intentó la configuración de herramientas como selenium y junit sin éxito por falta de experiencia y tiempo del equipo, se realizaron instalaciónes de Maven para hacer pruebas de compilación.

# SELENIUM

Selenium es un conjunto de utilidades que facilita la labor de obtener juegos de pruebas para aplicaciones web. Para ello nos permite grabar, editar y depurar casos de prueba, que podrán ser ejecutados de forma automática e iterativa posteriormente. Además de ser una herramienta para registrar acciones, permite editarlas manualmente o crearlas desde cero. Las acciones se basan en el uso de diferentes API's en diferentes lenguajes (PHP, Ruby, JAVA, Javascript, etc). Entre su principales características podemos nombrar:

* Facilidad de registro y ejecución de los test.
* Referencia a objetos DOM en base al ID, nombre o a través de XPath.
* Auto-completado para todos los comandos.
* Las acciones pueden ser ejecutadas paso a paso.
* Herramientas de depuración y puntos de ruptura (breakpoints).
* Los test pueden ser almacenados en diferentes formatos.
* Se descarga e instala la Versión de Java

La versión actual de Selenium está basada exclusivamente en HTML y JavaScript y permite a los desarrolladores probar y registrar las interacciones con una aplicación web y luego repetirlas las veces que se desee, de forma completamente automática. Los componentes más importantes para realizar este proceso de pruebas son los siguientes:

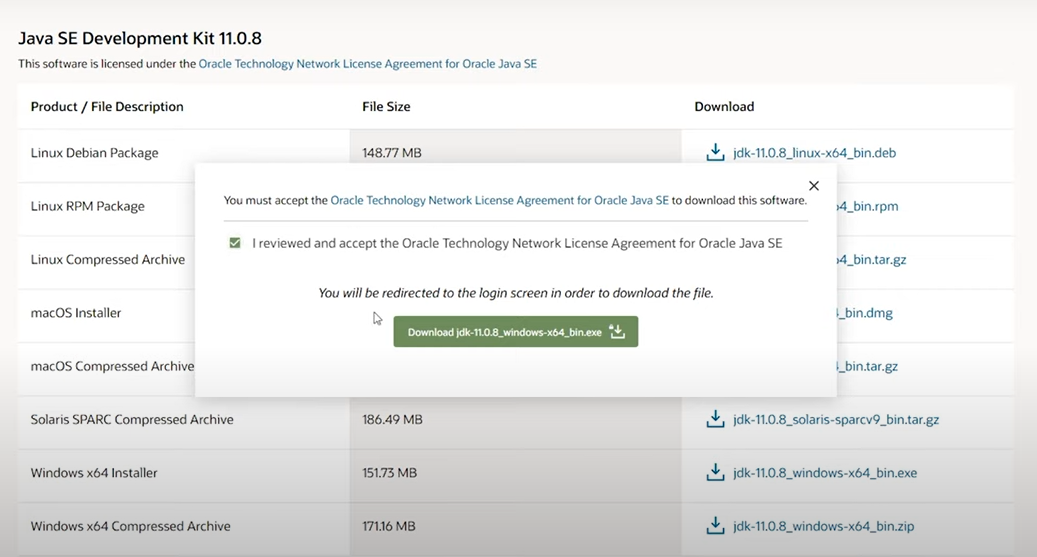
Selenium Core: el módulo del núcleo contiene la funcionalidad básica del marco de trabajo, incluyendo el JavaScriptTestRunner y la API de comandos de prueba básicos.

Selenium IDE: Selenium IDE es el entorno de desarrollo del marco de pruebas, sobre el que se basa la extensión IDE para Chrome y Firefox. Se utiliza para grabar y reproducir pruebas.

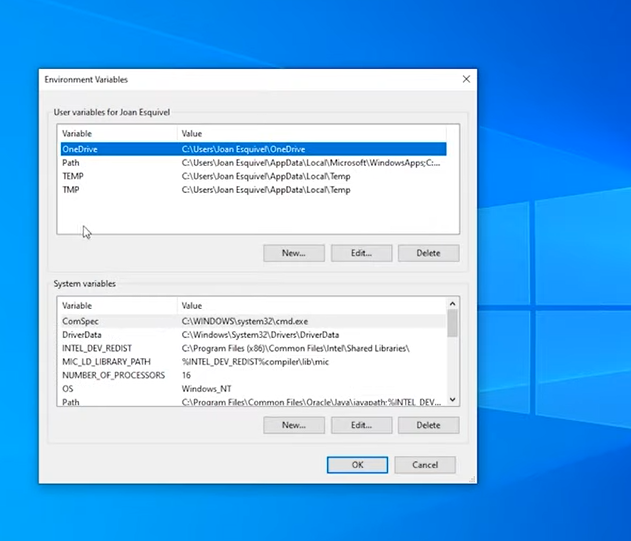
Selenium WebDriver: WebDriver es la interfaz básica para simular las interacciones del usuario con cualquier navegador, ya sea Firefox, Chrome, Edge, Safari o Internet Explorer. Desde 2018, la API es un estándar W3C oficial.

Selenium Grid: Selenium Grid es una extensión de WebDriver y de su predecesor Selenium Remote Control (RC), que permite la ejecución de pruebas en paralelo en múltiples servidores. De este modo, el tiempo de pruebas se reduce considerablemente.

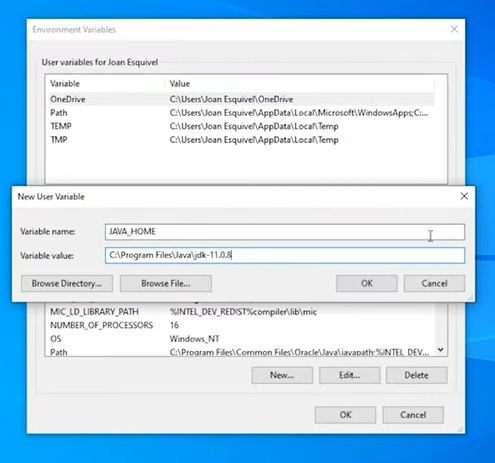
**Instalación de Selenium.**



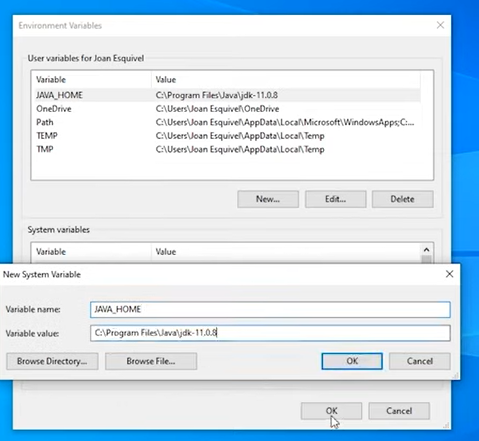
Se dan de alta las variables de ambiente, se escribe en inicio, variables de inicio y aparece editar las variables de entorno del sistema:



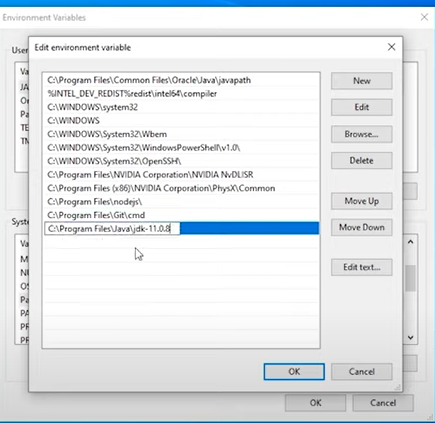
Se crea una variable de usuario apuntando a donde instalamos el JDK



Después de crea una variable de sistema para la misma ruta:

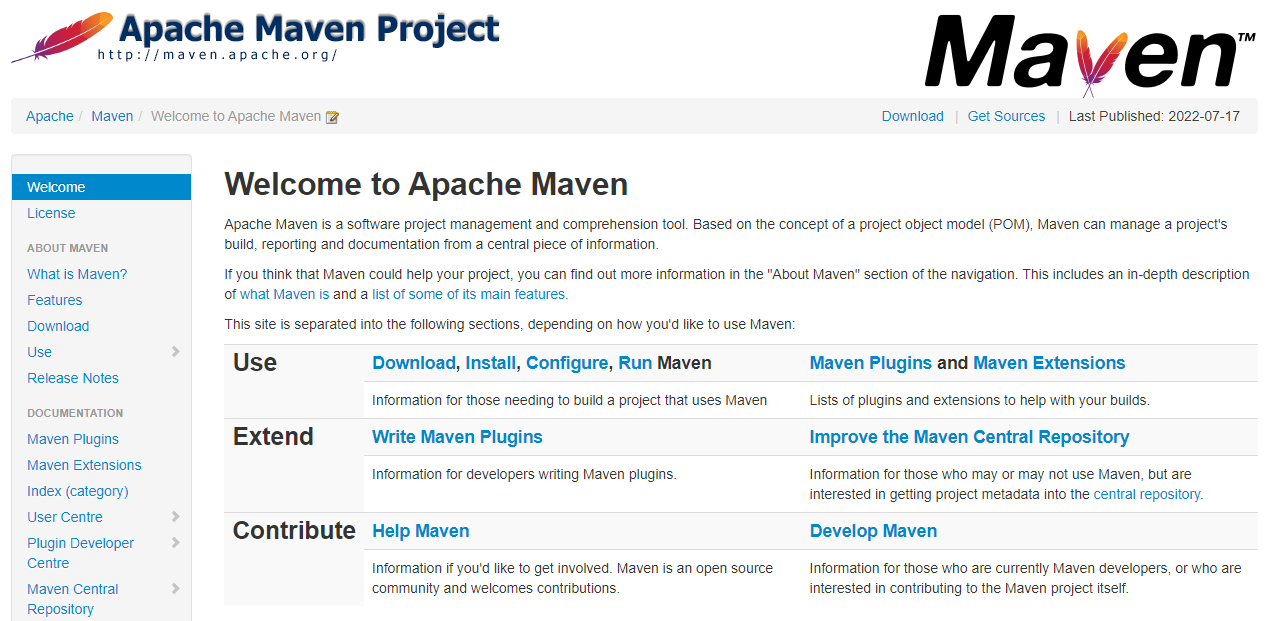


Buscar la línea de path y crear el valor con la misma ruta:

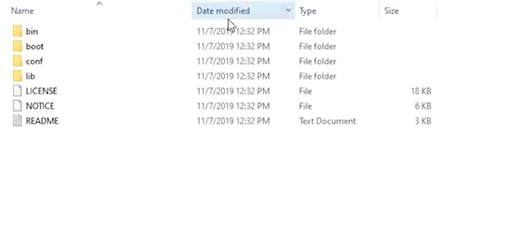


Con esto ya tenemos el JAVA configurado para poder usarlo, el siguiente paso es la instalación de Maven.

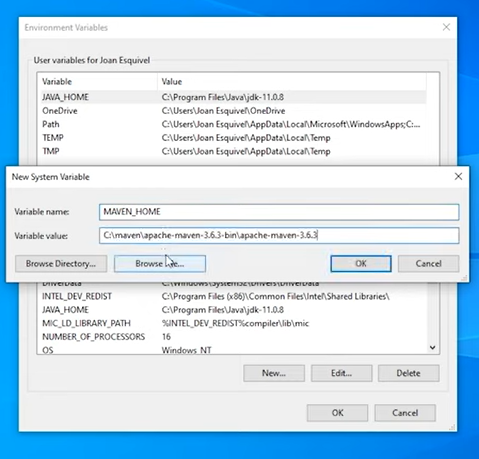
# MAVEN



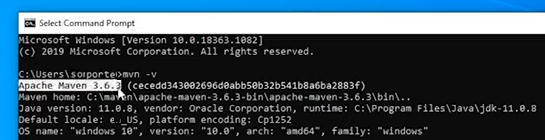
Se descarga el paquete y se crea un directorio en raíz para poner el contenido de la descarga



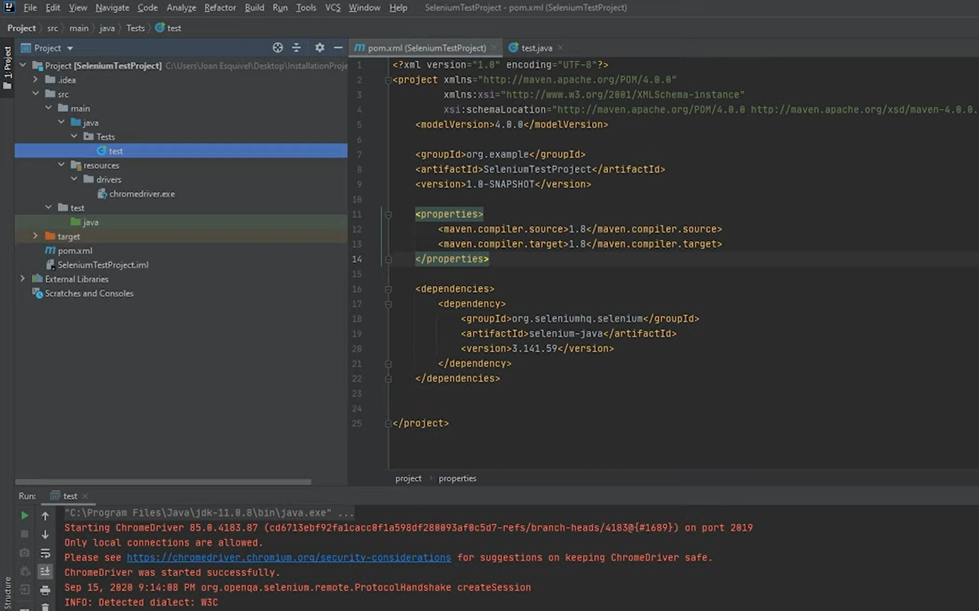
Se crea una nueva variable de entorno apuntando al directorio de maven



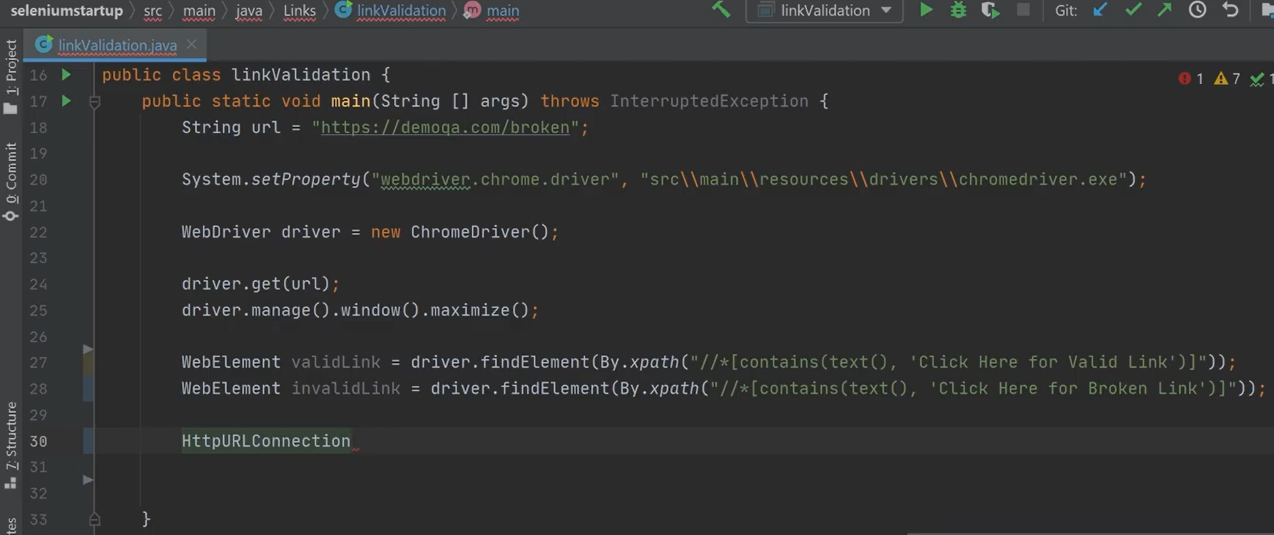
Se realiza la prueba por comando para ver que se haya instalado correctamente

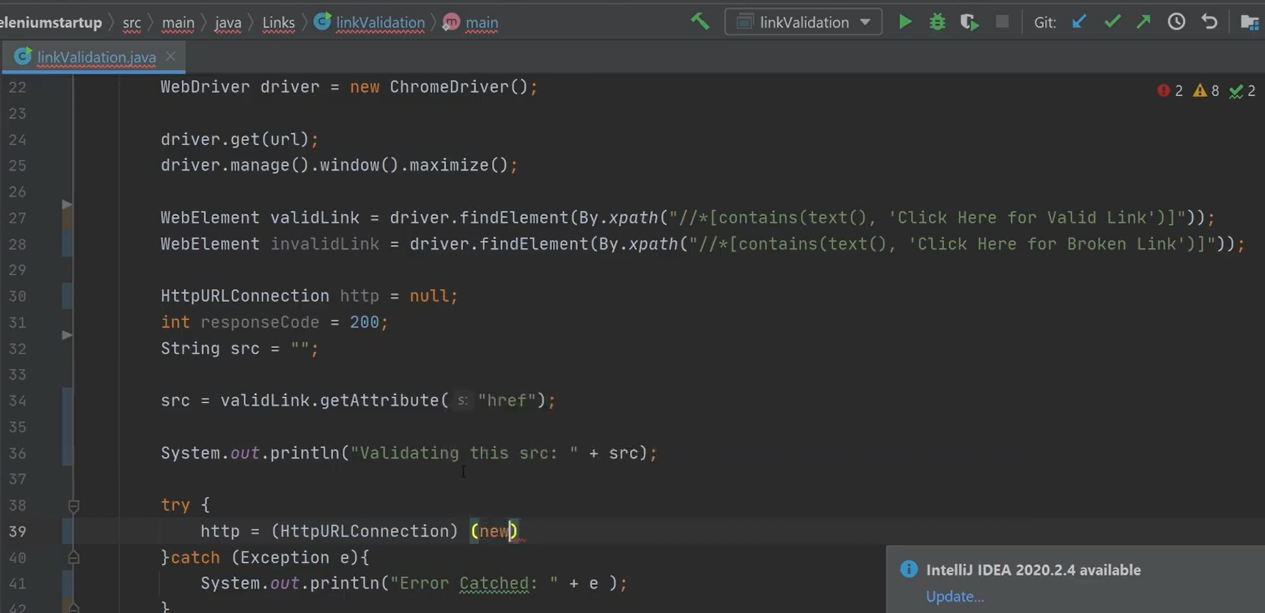


Se abre Maven y se pone el proyecto, se hace una prueba de ejecución de selenium, el cual ya es integrado a Maven copiando los directorios y creando la dependencia.



Prueba de código de enlaces rotos en la página web:





Los scripts de prueba en varios lenguajes, incluyendo JavaScript, Java, C#, Ruby y Python. Existen ya varios ejemplos que pueden ser adaptados a nuestros proyectos como el siguiente código:

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

# Crear una sesión de Firefox

driver = webdriver.Firefox()

driver.implicitly\_wait(30)

driver.maximize\_window()

# Acceder a la aplicación web

driver.get("http://www.google.es")

# Localizar cuadro de texto

search\_field = driver.find\_element\_by\_id("lst-ib")

search\_field.clear()

# Indicar y confirmar término de búsqueda

search\_field.send\_keys("Término de búsqueda")

search\_field.submit()

# Obtener la lista de resultados de la búsqueda y mostrarla

# mediante el método find\_elements\_by\_class\_name

lists= driver.find\_elements\_by\_class\_name("\_Rm")

# Pasar por todos los elementos y reproducir el texto individual

i=0

for listitem in lists:

print (listitem.get\_attribute("innerHTML"))

i=i+1

if(i>10):

break

# Cerrar la ventana del navegador

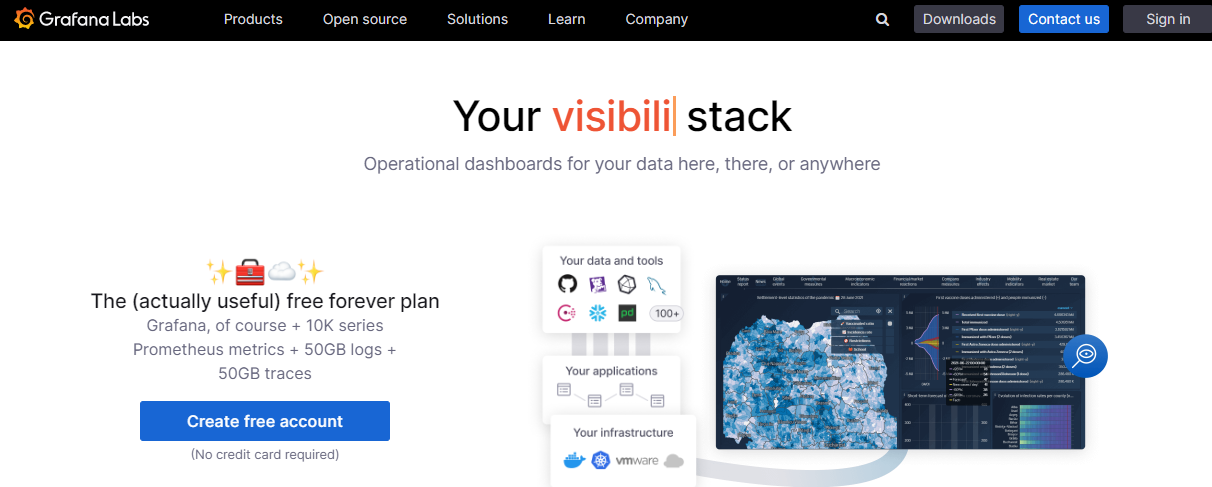
driver.quit()

Este script llama al motor de búsqueda Google automáticamente en Mozilla, introduce un término de búsqueda y luego evalúa o registra los resultados.

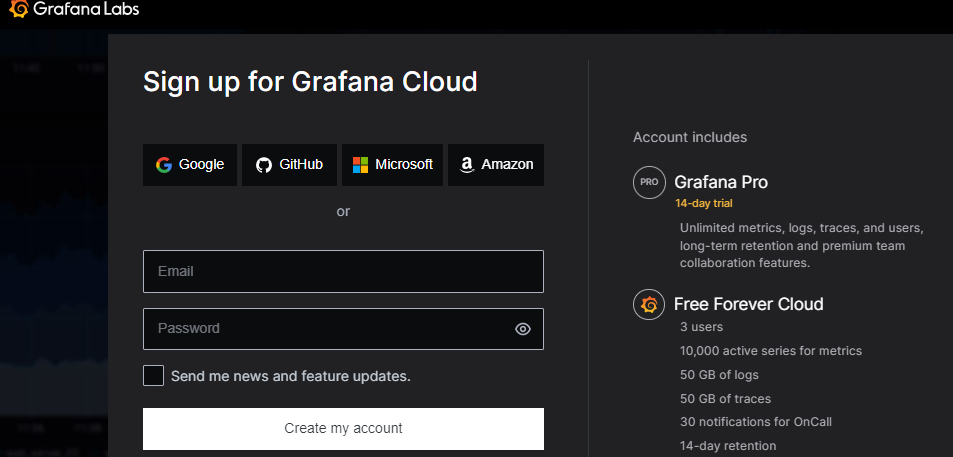
# MONITOREO CON GRAFANA

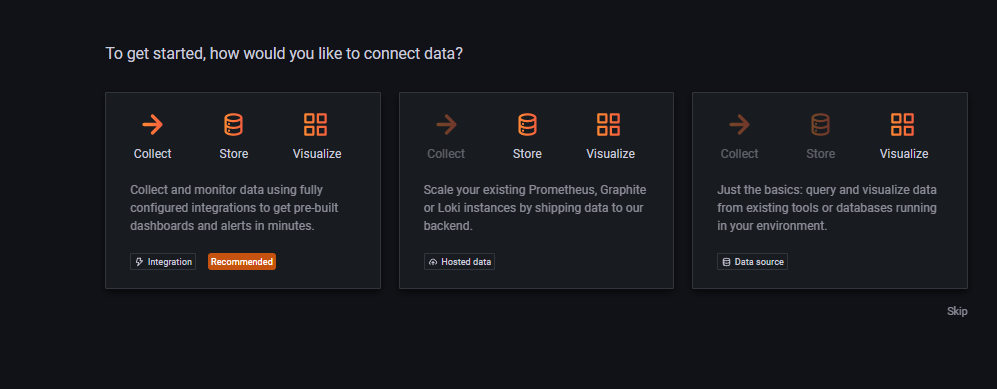
Grafana es una herramienta de código abierto para el análisis y visualización de métricas. Se utiliza frecuentemente para visualizar de una forma elegante series de datos en el análisis de infraestructuras y aplicaciones. Ackstorm ha diseñado e implementado un nuevo sistema de monitorización gráfico basado en Grafana.

Es una solución de código abierto que sirve para ejecutar análisis de datos, extraer métricas que dan sentido ante enormes cantidades de datos y monitorear aplicaciones y recursos hardware con la ayuda de atractivos paneles de control personalizables. Grafana es una herramienta de interfaz de usuario centralizada en la obtención de datos a partir de consultas, como también del almacenamiento de estos y su visualización. Es completamente de código abierto, y está respaldada por una gran comunidad entusiasta y dedicada.



Para este proyecto nos registramos con una cuenta gratuita

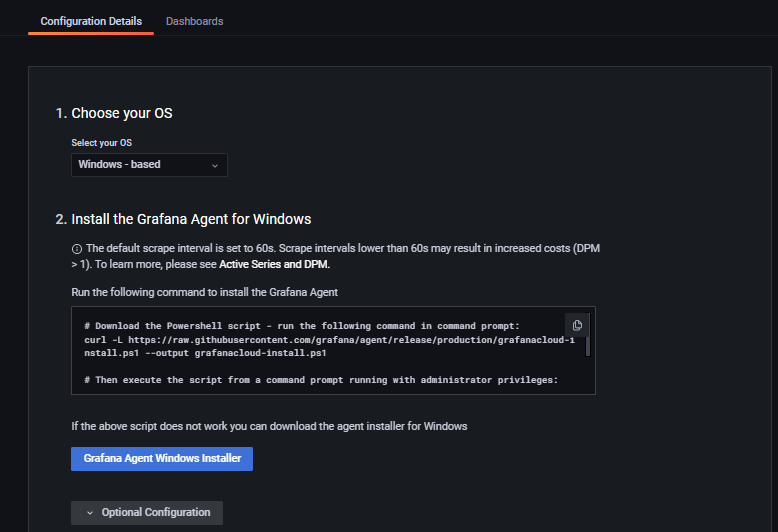




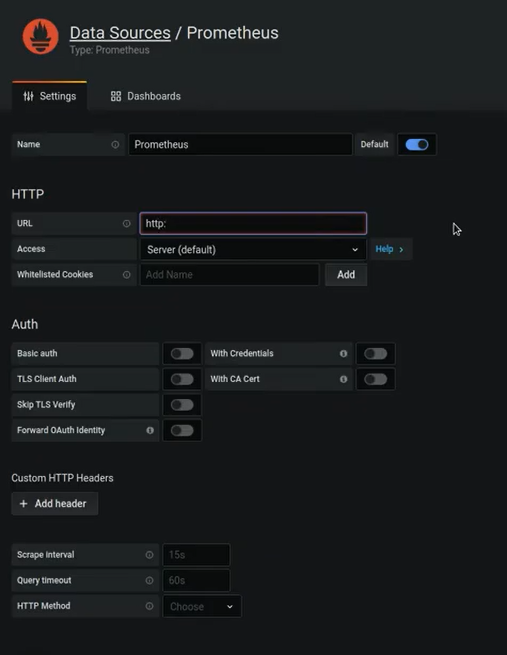
Listado de herramientas que podemos usar para monitorear:

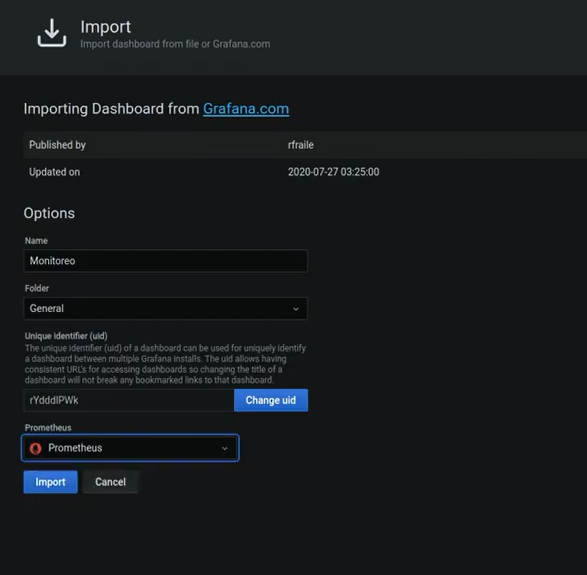


Seleccionamos MySql



Descargamos el instalador e iniciamos el proceso, después de la instalación hacemos un nuevo Dashboard





Se configura con la información de acceso de nuestro servidor,



Con esta herramienta se pueden monitorear diversos recursos en tiempo real, solamente que se necesita tener un plan con licenciamiento que es costoso para poder hacer un ejercicio de nuestra aplicación con este sistema.

