

MANUAL DE INSTALACIÓN DEL SISTEMA WEB PARA EL CÁLCULO DEL AGUA DE RIEGO

Facundo Paez. Andrés Vidaurre.



Universidad Nacional de la Patagonia San Juan Bosco
Desarrollo de Software



Contenido

1 Stack tecnológico.....	2
2 Instalación de dependencias	2
2.1 Instalación del OpenJDK 8 en Ubuntu	2
2.2 Instalación de Apache ANT 1.10.13 en Ubuntu	3
2.3 Instalación del OpenJDK 8 en Windows	3
2.4 Instalación de Apache ANT 1.10.13 en Windows.....	8
2.5 Instalación de GlassFish 5.0.1 Full Platform	9
2.6 Instalación de las bibliotecas gson 2.8.2, apache commons codec 1.4 y java jwt 4.2.1	10
3 Configuración de la aplicación	10
3.1 Configuración de las rutas de la aplicación y del servidor en el archivo common.properties	10
3.2 Configuración de la clave API de Visual Crossing Weather	10
3.3 Activación de métodos automáticos.....	11
3.3.1 Activación del método automático de obtención y persistencia de datos meteorológicos	11
3.3.2 Activación del método automático para establecer el estado finalizado en un registro de plantación que tiene un estado de desarrollo	11
3.3.3 Activación del método automático para establecer un estado de desarrollo en un registro de plantación en espera	12
3.3.4 Activación del método automático para el cálculo de la necesidad de agua de riego de un cultivo en desarrollo.....	12
3.4 Generación de una contraseña de aplicación en una cuenta de Google.....	13
3.5 Ajuste de GlassFish 5: Eliminación de la carpeta sun del archivo grizzly-npn-bootstrap.jar ..	13
3.6 (Opcional) Modificación de la clave secreta utilizada para la firma de un JWT (JSON Web Token).....	14
4 Despliegue de la aplicación	14
5 Lista de comandos ANT de la aplicación	15



1 Stack tecnológico

La aplicación del lado del navegador web está implementada con las siguientes tecnologías:

- Framework: AngularJS v1.6.4
- Biblioteca CSS: Bootstrap v3.3.7
- Biblioteca de JavaScript: jQuery v3.2.1

La aplicación del lado del servidor está implementada con las siguientes tecnologías:

- Kit de desarrollo de software: Oracle JDK (Java Development Kit) 8
- Motor de base de datos relacional: Apache Derby
- Framework de mapeo objeto-relacional: EclipseLink v2.1.0
- Biblioteca de autenticación y autorización: [Java JWT](#) v4.2.1
- Servidor Web: GlassFish 5.0.1 Full Platform

La aplicación fue desarrollada, probada y ejecutada utilizando las distribuciones GNU/Linux Manjaro y Ubuntu.

2 Instalación de dependencias

Para desplegar la aplicación se requieren los siguientes softwares:

1. Una implementación del JDK (Java Development Kit) versión 8. Esta implementación puede ser OracleJDK, OpenJDK, etc.
2. Apache ANT versión 1.10.13.
3. Servidor Web GlassFish 5.0.1 Full Platform.
4. Bibliotecas: gson v2.8.2, apache commons codec v1.4 y java jwt v4.2.1.

2.1 Instalación del OpenJDK 8 en Ubuntu

Para instalar el OpenJDK 8 en Ubuntu se debe abrir la línea de comandos y ejecutar los siguientes comandos en el orden en el que están escritos:

1. `sudo apt-get install openjdk-8-jdk`
2. `java -version`. Si resultado es un mensaje como el que se muestra a continuación, el JDK elegido se instaló satisfactoriamente.

openjdk versión "1.8.0_362"

OpenJDK Runtime Environment (build 1.8.0_362-8u362-ga-0ubuntu1~20.04.1-b09)

OpenJDK 64-Bit Server VM (build 25.362-b09, mixed mode)



2.2 Instalación de Apache ANT 1.10.13 en Ubuntu

1. Descargar Apache ANT 1.10.13 del siguiente enlace: <https://ant.apache.org/bindownload.cgi>
2. Descomprimir el archivo en una carpeta a elección.
3. Crear variable de entorno para Apache ANT. Para esto se debe abrir la línea de comandos y ejecutar el siguiente comando:

```
sudo nano /etc/profile.d/ant.sh.
```

4. Dentro del archivo ant.sh se deben escribir las siguientes líneas:

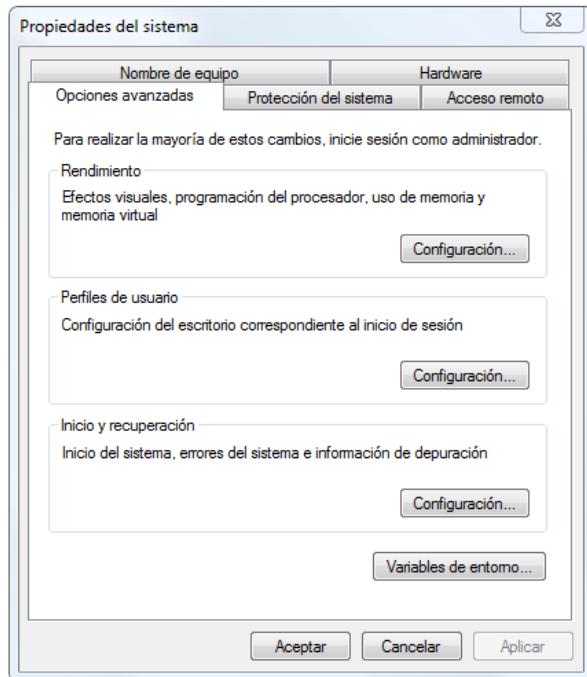
```
export ANT_HOME=<ruta en la que se descomprimió Apache ANT>  
export PATH=${ANT_HOME}/bin:$PATH
```

5. Guardar el archivo.
6. En la línea de comandos ejecutar el comando: `source /etc/profile.d/ant.sh`
7. En la línea de comandos ejecutar el comando: `ant -version`. Si el resultado es un mensaje como el que se muestra a continuación, Apache ANT 1.10.13 se instaló satisfactoriamente.

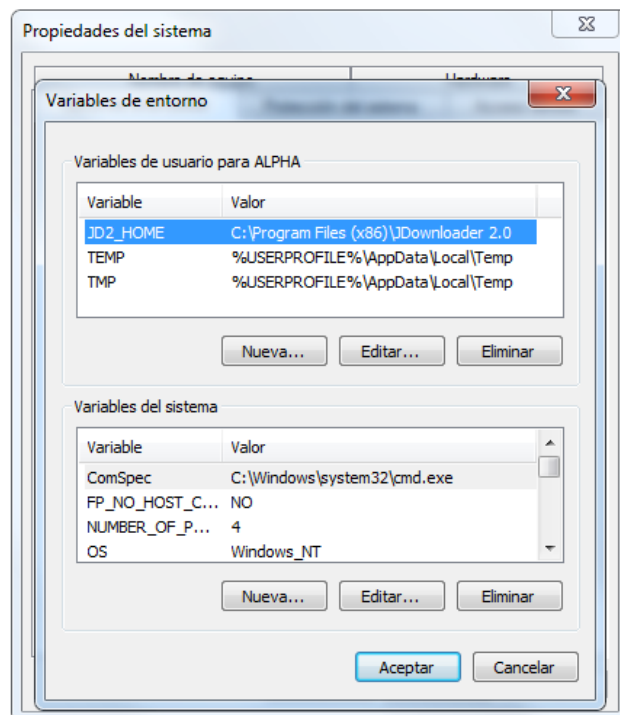
```
Apache Ant(TM) version 1.10.13 compiled on January 4 2023
```

2.3 Instalación del OpenJDK 8 en Windows

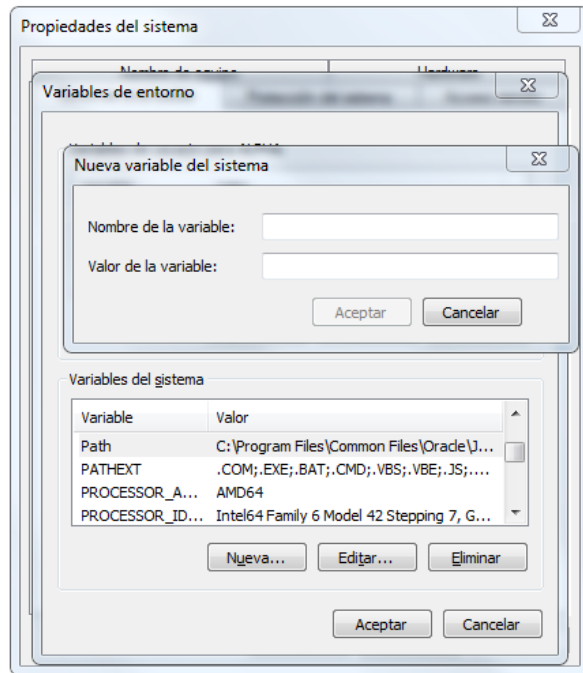
1. Descargar de este [enlace](#) el OpenJDK 8 para la versión utilizada de Windows.
2. Descomprimir el archivo en una carpeta a elección.
3. Abrir el cuadro de diálogo Propiedades del sistema. Haga clic en *Inicio > Panel de control > Sistema y seguridad > Sistema > Configuración avanzada del sistema* para mostrar el cuadro de dialogo *Propiedades del sistema*.



4. Abrir el cuadro de dialogo *Variables de entorno*. Presione el botón *Variables de entorno* de la parte inferior de la pestaña *Opciones avanzadas* del cuadro de dialogo *Propiedades del sistema*.

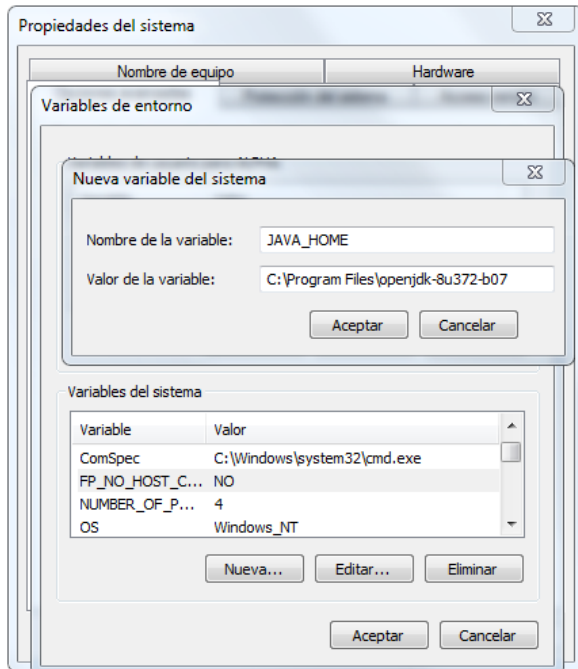


5. En el cuadro *Variables del sistema* presionar el botón *Nueva*. Esto hará que se despliegue el cuadro de dialogo *Nueva variable del sistema*.



6. Escribir JAVA_HOME como nombre de la nueva variable del sistema.

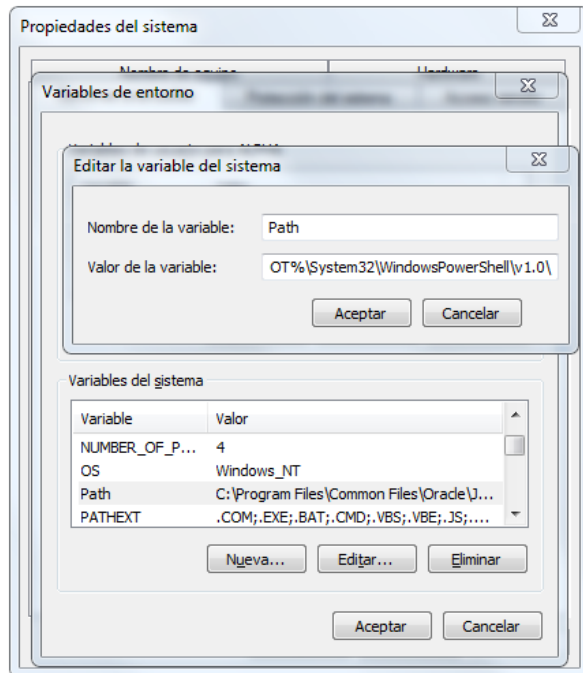
7. En el valor de la variable se debe escribir la ruta en la que se descomprimió el JDK.



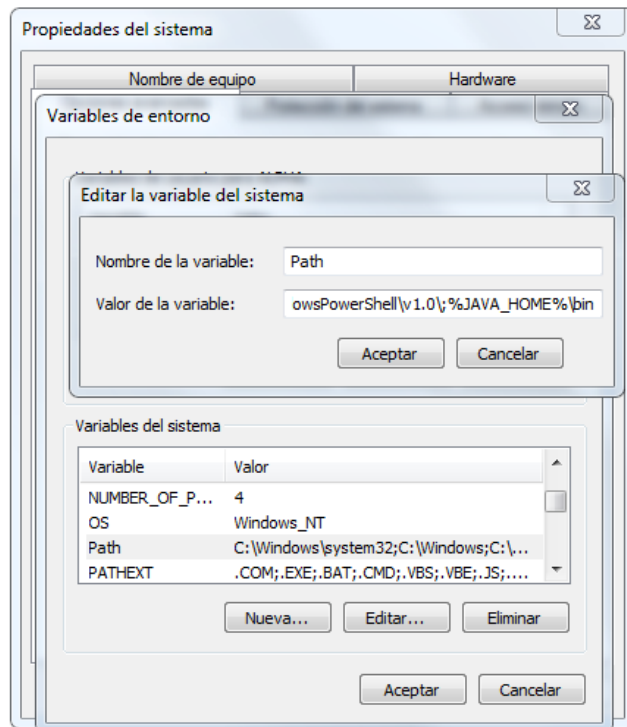
8. Clic en el botón *Aceptar*.



9. Editar la variable de entorno Path. Desplácese por el cuadro *Variables del sistema* para seleccionar la variable Path. Haga clic en el botón *Modificar*. Esto hará que se despliegue el cuadro de dialogo *Editar la variable del sistema*.



10. Modificar la variable Path. Coloque el cursor dentro del campo *Valor de la variable*. Use la flecha derecha para desplazar el cursor hasta el final de la lista. Al final de la lista, escriba un punto y coma, seguido de %JAVA_HOME%, seguido de "\bin;" (sin las comillas). No coloque espacios antes o después de lo que escriba. No se permiten espacios antes o después de cada valor en una variable de entorno. Haga clic en el botón *Aceptar* para aplicar sus cambios a la variable Path.



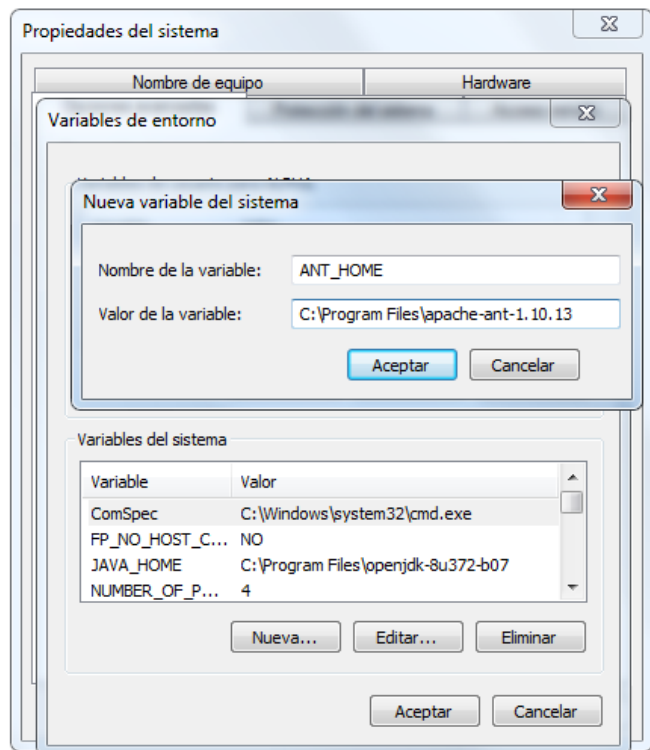
11. Abrir la línea de comandos y ejecutar el comando *java -version*. Si el resultado es un mensaje como el que se muestra a continuación, el JDK elegido se instaló satisfactoriamnte.

```
openjdk version "1.8.0_372-372"  
OpenJDK Runtime Environment (build 1.8.0_372-372-b07)  
OpenJDK 64-Bit Server VM (build 25.372-b07, mixed mode)
```




2.4 Instalación de Apache ANT 1.10.13 en Windows

1. Descargar Apache ANT 1.10.13 del siguiente enlace: <https://ant.apache.org/bindownload.cgi>
2. Descomprimir el archivo en una carpeta a elección.
3. Abrir el cuadro de diálogo *Propiedades del sistema*. Haga clic en *Inicio > Panel de control > Sistema y seguridad > Sistema > Configuración avanzada del sistema* para mostrar el cuadro de dialogo *Propiedades del sistema*.
4. Abrir el cuadro de dialogo *Variables de entorno*. Presione el botón *Variables de entorno* de la parte inferior de la pestaña *Opciones avanzadas* del cuadro de dialogo *Propiedades del sistema*.
5. En el cuadro *Variables del sistema* presionar el botón *Nueva*. Esto hará que se despliegue el cuadro de dialogo *Nueva variable del sistema*.
6. Escribir ANT_HOME como nombre de la nueva variable del sistema.
7. En el valor de la variable se debe escribir la ruta en la que se descomprimió Apache ANT.

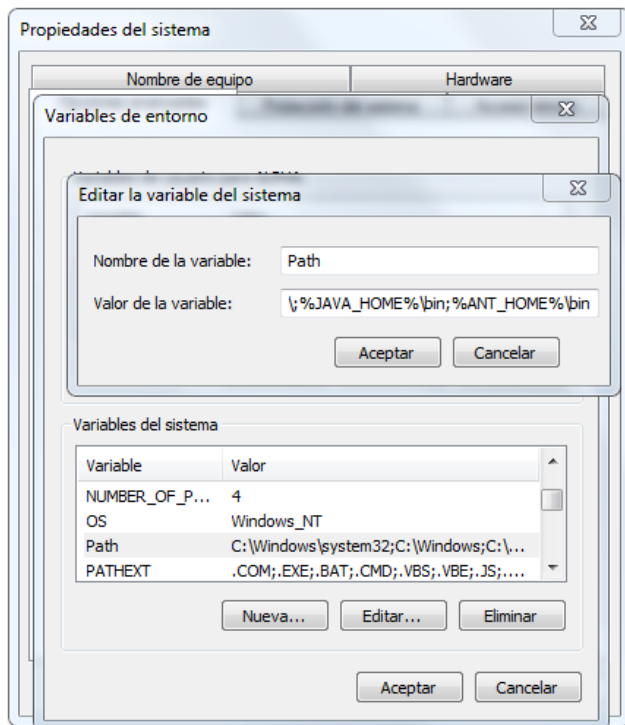


8. Clic en el botón *Aceptar*.



9. Editar la variable de entorno Path. Desplácese por el cuadro *Variables del sistema* para seleccionar la variable Path. Haga clic en el botón *Modificar*. Esto hará que se despliegue el cuadro de dialogo *Editar la variable del sistema*.

10. Modificar la variable Path. Coloque el cursor dentro del campo *Valor de la variable*. Use la flecha derecha para desplazar el cursor hasta el final de la lista. Al final de la lista, escriba un punto y coma, seguido de %ANT_HOME%, seguido de “\bin;” (sin las comillas). No coloque espacios antes o después de lo que escriba. No se permiten espacios antes o después de cada valor en una variable de entorno. Haga clic en el botón *Aceptar* para aplicar sus cambios a la variable Path.



11. Abrir la línea de comandos y ejecutar el comando *ant -version*. Si el resultado es un mensaje como el que se muestra a continuación, Apache ANT se instaló satisfactoriamente.

```
ANT_OPTS is set to -Djava.security.manager=allow
Apache Ant(TM) version 1.10.13 compiled on January 4 2023
```

2.5 Instalación de GlassFish 5.0.1 Full Platform

1. Descargar GlassFish 5.0.1 Full Platform del siguiente enlace:

<https://javaee.github.io/glassfish/download>

2. Descomprimir el archivo en una carpeta a elección.



2.6 Instalación de las bibliotecas gson 2.8.2, apache commons codec 1.4 y java jwt 4.2.1

1. Descargar la biblioteca gson v2.8.2 del siguiente enlace: [gson-2.8.2.jar](#)
2. Descargar la biblioteca apache commons codec v1.4 del siguiente enlace: [apache-commons-codec-1.4.jar](#)
3. Descargar la biblioteca java jwt v4.2.1 del siguiente enlace: [java-jwt-4.2.1.jar](#)
4. Descomprimir los tres archivos descargados.
5. Mover únicamente los archivos gson-2.8.2.jar, apache-commons-codec-1.4.jar y java-jwt-4.2.1.jar a la carpeta *lib* de GlassFish 5 FullPlatform, la cual se encuentra en la ruta glassfish5/glassfish.

3 Configuración de la aplicación

Además de instalar el software necesario para la ejecución de la aplicación, también es necesario realizar ciertas configuraciones para poder ejecutarla. A continuación, se presentan las configuraciones que se deben realizar para ejecutar la aplicación.

3.1 Configuración de las rutas de la aplicación y del servidor en el archivo `common.properties`

1. Abrir el archivo **common.properties** de la ruta sistema-web-riego/config del directorio de la aplicación.
2. Modificar el contenido de la variable `EXAMPLES_HOME` con la ruta absoluta del directorio de la aplicación. El resultado debe tener la siguiente forma: `EXAMPLES_HOME=<ruta absoluta>/sistema-web-riego`. Por ejemplo, `EXAMPLES_HOME=/home/alpha/Escritorio/wd/sistema-web-riego`
3. Modificar el contenido de la variable `SERVER_HOME` con la ruta absoluta del directorio del servidor web GlassFish 5. El resultado debe tener la forma: `SERVER_HOME=<ruta absoluta>/glassfish5`. Por ejemplo, `SERVER_HOME=/home/alpha/Escritorio/wd/glassfish5`

3.2 Configuración de la clave API de Visual Crossing Weather

La aplicación utiliza el servicio meteorológico [Visual Crossing Weather](#) para obtener los datos meteorológicos de la ubicación geográfica de una parcela, los cuales son necesarios para calcular la necesidad de agua de riego de un cultivo en la fecha actual (es decir, hoy) [mm/día]. Para obtener dichos datos, la aplicación realiza peticiones HTTP al servicio Visual Crossing Weather haciendo uso de una clave API provista por el mismo. A continuación, se describen los pasos a seguir para generar y utilizar una clave API de Visual Crossing Weather.

1. Crear una cuenta en [Visual Crossing Weather](#).
2. Iniciar sesión en la cuenta e ingresar en el menú de la cuenta mediante el botón que tiene la etiqueta *Account*, el cual se encuentra en la parte superior derecha de la página web de Visual Crossing Weather.
3. Copiar la clave API (*API key* en inglés).
4. Abrir el archivo **ClimateClient.java** de la ruta sistema-web-riego/app/src/climate del directorio de la aplicación y reemplazar la palabra `YOUR_VISUAL_CROSSING_WEATHER_API_KEY` de la constante `API_KEY` por la clave API copiada en el paso 3.



3.3 Activación de métodos automáticos

La aplicación tiene una serie de métodos automáticos, los cuales realizan una determinada tarea. Por defecto, estos métodos están desactivados.

3.3.1 Activación del método automático de obtención y persistencia de datos meteorológicos

Luego de configurar la clave API de la cuenta de Visual Crossing Weather, lo que sigue es activar el método automático encargado de obtener y persistir los datos meteorológicos de la ubicación geográfica de cada una de las parcelas activas de la base de datos subyacente. A continuación, se describen los pasos para activar dicho método.

1. Abrir el archivo **ClimateRecordManager.java** de la ruta `sistema-web-riego/app/src/climate` del directorio de la aplicación.
2. Eliminar las dos barras diagonales de la línea de código `@Schedule(second = "*", minute = "*", hour = "0/2", persistent = false)` del método `getCurrentWeatherDataset`.

3.3.2 Activación del método automático para establecer el estado finalizado en un registro de plantación que tiene un estado de desarrollo

Cuando la fecha de cosecha de un registro de plantación que tiene un estado de desarrollo (en desarrollo, desarrollo óptimo, desarrollo en riesgo de marchitez, desarrollo en marchitez) es estrictamente menor (es decir, anterior) a la fecha actual (es decir, hoy), dicho registro debe adquirir el estado "Finalizado". El método automático `modifyToFinishedStatus()` de la clase Java `PlantingRecordManager` obtiene de la base de datos subyacente los registros de plantación que tienen un estado de desarrollo y verifica si un registro de plantación que presuntamente tiene un estado de desarrollo, está finalizado (es decir, si su fecha de cosecha es estrictamente menor a la fecha actual). Si lo está, modifica el estado de un registro de plantación al estado "Finalizado". En caso contrario, no realiza dicha modificación. A continuación, se describen los pasos para activar dicho método.

1. Abrir el archivo **PlantingRecordManager.java** de la ruta `sistema-web-riego/app/src/plantingRecord` del directorio de la aplicación.
2. Eliminar las dos barras diagonales de la línea de código `@Schedule(second = "*", minute = "*", hour = "0/23", persistent = false)` del método `modifyToFinishedStatus`.



3.3.3 Activación del método automático para establecer un estado de desarrollo en un registro de plantación en espera

Cuando la fecha de siembra es menor o igual a la fecha actual (es decir, hoy) y la fecha de cosecha de un registro de plantación en espera es mayor o igual a la fecha actual, dicho registro debe adquirir un estado de desarrollo (en desarrollo, desarrollo óptimo, desarrollo en riesgo de marchitez, desarrollo en marchitez). El método automático `modifyToInDevelopmentStatus()` de la clase Java `PlantingRecordManager` obtiene de la base de datos subyacente los registros de plantación que tienen el estado “En espera” y verifica si un registro de plantación presuntamente en espera, tiene un estado de desarrollo (es decir, si la fecha actual es mayor o igual a la fecha de siembra y menor o igual a la fecha de cosecha). Si lo tiene, modifica el estado de un registro de plantación a un estado de desarrollo. En caso contrario, no realiza dicha modificación. A continuación, se describen los pasos para activar dicho método.

1. Abrir el archivo **PlantingRecordManager.java** de la ruta `sistema-web-riego/app/src/plantingRecord` del directorio de la aplicación.
2. Eliminar las dos barras diagonales de la línea de código `@Schedule(second = "*", minute = "*", hour = "1/23", persistent = false)` del método `modifyToInDevelopmentStatus`.

3.3.4 Activación del método automático para el cálculo de la necesidad de agua de riego de un cultivo en desarrollo

La aplicación calcula automáticamente en la fecha actual (es decir, hoy) la necesidad de agua de riego [mm/día] de un cultivo en desarrollo. Esto lo realiza mediante el método automático `setCropIrrigationWaterNeed()` de la clase Java `PlantingRecordManager`. En cada día del año este método obtiene de la base de datos subyacente los registros de plantación que tienen un estado de desarrollo (en desarrollo, desarrollo óptimo, desarrollo en riesgo de marchitez, desarrollo en marchitez) y para cada uno de ellos calcula la necesidad de agua de riego del cultivo que contienen, el cual está en proceso de desarrollo. A continuación, se describen los pasos para activar dicho método.

1. Abrir el archivo **PlantingRecordManager.java** de la ruta `sistema-web-riego/app/src/plantingRecord` del directorio de la aplicación.
2. Eliminar las dos barras diagonales de la línea de código `@Schedule(second = "*", minute = "*", hour = "1/2", persistent = false)` del método `setCropIrrigationWaterNeed`.



3.4 Generación de una contraseña de aplicación en una cuenta de Google

La aplicación utiliza el servidor de correo electrónico de Google (Gmail) para el envío de correos electrónicos de confirmación de registro y restablecimiento de la contraseña de la cuenta de un usuario. Para que esto funcione es necesario tener una cuenta de Google y generar en ella una contraseña de aplicación. A continuación, se describen los pasos para generar una contraseña de aplicación con una cuenta de Google.

1. Iniciar sesión en una cuenta de Google.
2. Ingresar en *Administrar tu cuenta de Google*.
3. Ingresar en *Seguridad*.
4. Ingresar en la opción *Verificación en 2 pasos* del apartado *Cómo acceder a Google* y activar la verificación en dos pasos.
5. Dirigirse al apartado *Contraseña de aplicaciones*, el cual se encuentra al final de la página web que se muestra al ingresar en la opción *Verificación en 2 pasos* del apartado *Cómo acceder a Google*.
6. Ingresar en la opción *Contraseñas de aplicaciones*.
7. En la lista desplegable que tiene la etiqueta “Seleccionar app”, seleccionar la opción “Otra (nombre personalizado)”.
8. Ingresar como nombre la sigla SWCAR (Sistema Web para el Cálculo del Agua de Riego).
9. Copiar la contraseña de aplicación.
10. Abrir el archivo **Email.java** de la ruta `sistema-web-riego/app/src/util` del directorio de la aplicación.
11. Reemplazar el valor de la constante `SENDER_EMAIL` por la dirección de correo electrónico en la que se generó la contraseña de aplicación.
12. Reemplazar el valor de la constante `EMAIL_PASSWORD` pegando la contraseña de aplicación copiada en el paso 9.

3.5 Ajuste de GlassFish 5: Eliminación de la carpeta `sun` del archivo `grizzly-npn-bootstrap.jar`

Es necesario eliminar la carpeta `sun` del archivo `grizzly-npn-bootstrap.jar` de GlassFish 5 porque de no hacerlo la aplicación tendrá la excepción `exception java.lang.NoClassDefFoundError: sun/security/ssl/HelloExtension` en tiempo de ejecución, lo cual hará que el módulo de envío de correos electrónicos no funcione.

1. Ingresar en la carpeta `endorsed` de GlassFish 5, la cual se encuentra en la ruta `glassfish5/glassfish/modules`.
2. Abrir el archivo `grizzly-npn bootstrap.jar` con `winrar` o su descompresor preferido.
3. Eliminar la carpeta `sun` del archivo `grizzly-npn bootstrap.jar`.



3.6 (Opcional) Modificación de la clave secreta utilizada para la firma de un JWT (JSON Web Token)

La aplicación utiliza JWT (JSON Web Token) para la autenticación y la autorización. Un JWT se firma con una clave secreta, la cual es una cadena de caracteres que debe conocer únicamente la aplicación del lado servidor, ya que de lo contrario la misma será vulnerable.

La aplicación tiene una clave secreta en el archivo **secretKeyInserts.sql** de la ruta sistema-web-riego/app/etc/sql con la que firma JWTs. Usted puede cambiar este clave secreta por otra o incluso añadir otras con las que la aplicación firme JWTs.

4 Despliegue de la aplicación

Una vez instaladas las dependencias y realizadas las configuraciones se está en condiciones para desplegar la aplicación. Para ello se debe abrir la línea de comandos y dirigirse a la ruta sistema-web-riego/app del directorio de la aplicación. Luego, se deben ejecutar los siguientes comandos en el orden en el que están escritos:

1. ant start
2. ant createdb
3. ant deploy
4. ant basicdata

Abrir un navegador web y acceder a la página web del URL: <http://localhost:8080/swcar/>. A modo de ejemplo, se puede utilizar el nombre de usuario liam y la contraseña liam para iniciar sesión y utilizar la aplicación.

El comando ant basicdata carga la base de datos subyacente con los datos mínimos que se necesitan para que un usuario pueda usar la aplicación. Estos datos son los tipos de cultivos, las regiones, los cultivos, los suelos, los estados de un registro de plantación (finalizado, en espera, en desarrollo, desarrollo óptimo, desarrollo en riesgo de marchitez, desarrollo en marchitez y muerto), las radiaciones solares y las insolaciones máximas para diferentes latitudes. Además de cargar estos datos, también carga una clave secreta, con la cual la aplicación firma JWTs (JSON Web Token), el nombre de usuario liam con la contraseña liam y el nombre de usuario admin con la contraseña admin.



5 Lista de comandos ANT de la aplicación

La aplicación tiene los siguientes comandos, los cuales se ejecutan mediante la herramienta Apache ANT:

- ant start. Inicia la ejecución de GlassFish 5 y Apache Derby.
- ant stop. Detiene la ejecución de GlassFish 5 y Apache Derby.
- ant createdb. Crea un pool de conexiones a la base de datos.
- ant deletedb. Elimina un pool de conexiones a la base de datos.
- ant deploy. Crea las tablas de la aplicación en la base de datos, empaqueta la aplicación en un archivo WAR y coloca este paquete en GlassFish 5.
- ant undeploy. Elimina las tablas de la aplicación en la base de datos y elimina la aplicación (contenida en un archivo WAR) de GlassFish 5.
- ant redeploy. Realiza las acciones de los comandos “ant undeploy” y “ant deploy” en este orden.
- ant all. Carga las tablas de la aplicación en la base de datos con un conjunto de datos de muestra.
- ant eraseall. Elimina el contenido de todas las tablas de la aplicación en la base de datos.
- ant basicdata. Carga las tablas de la aplicación en la base de datos con el conjunto de datos mínimos que se necesitan para que un usuario pueda utilizar la aplicación.

Para ejecutar cada uno de estos comandos se debe abrir la línea de comandos y dirigirse a la ruta sistema-web-riego/app del directorio de la aplicación.