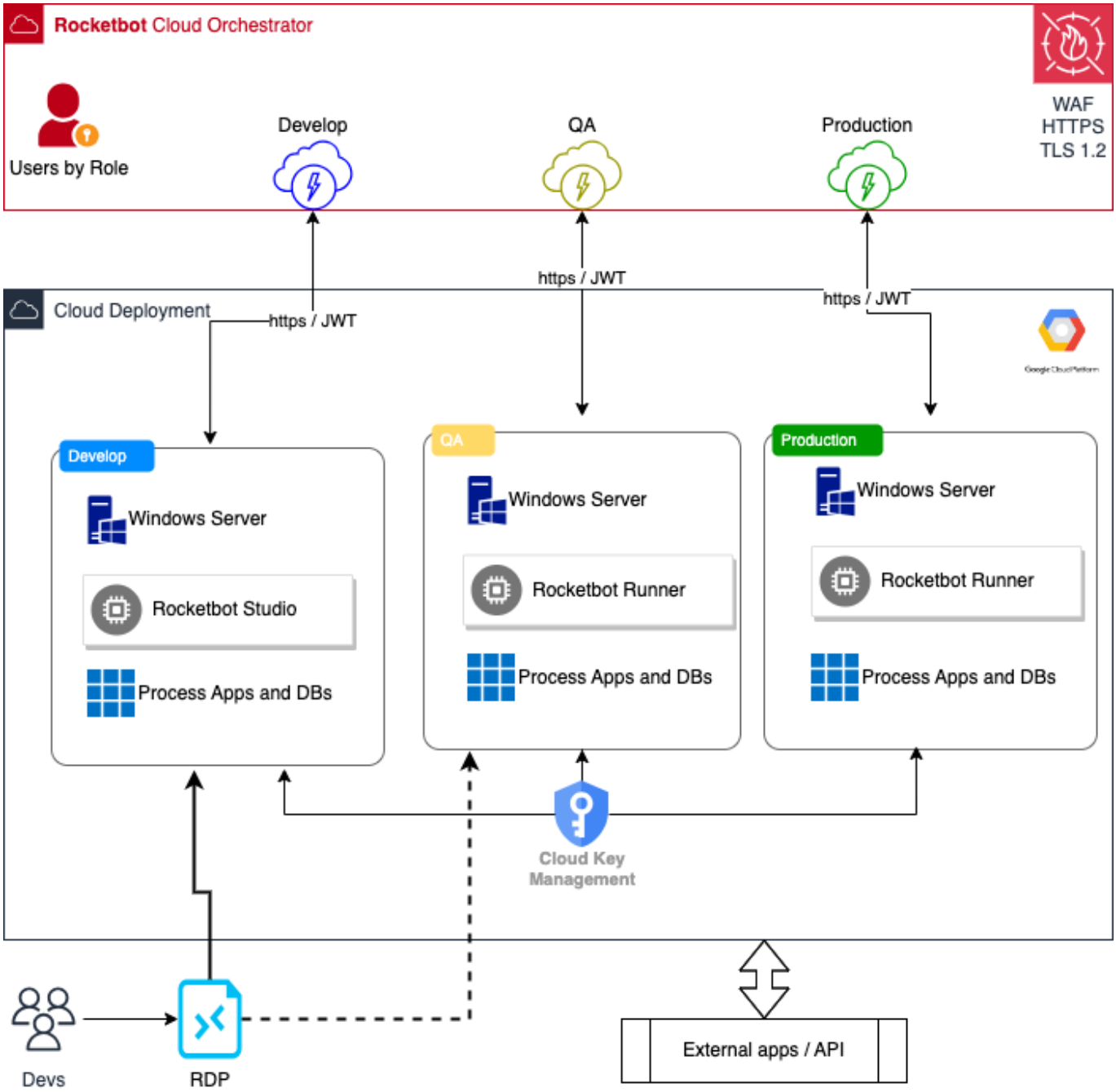
Despliegue de Rocketbot en GCP



En esta guía, te explicaremos los pasos para crear una máquina virtual (VM) en Google Cloud Platform (GCP) y montar Rocketbot Studio para ejecutar robots.

Arquitectura recomendada.



Crear y administrar VM de Windows Server

Compute Engine proporciona imágenes públicas con Windows Server que puedes usar para crear instancias.

Antes de comenzar

1. En la página del selector de proyectos de la consola de Google Cloud, selecciona o crea un proyecto de Google Cloud.

Nota: Si no planeas conservar los recursos que creaste durante este procedimiento, crea un proyecto en lugar de seleccionar uno existente. Cuando termines, puedes borrar el proyecto y quitar todos los recursos asociados con él.

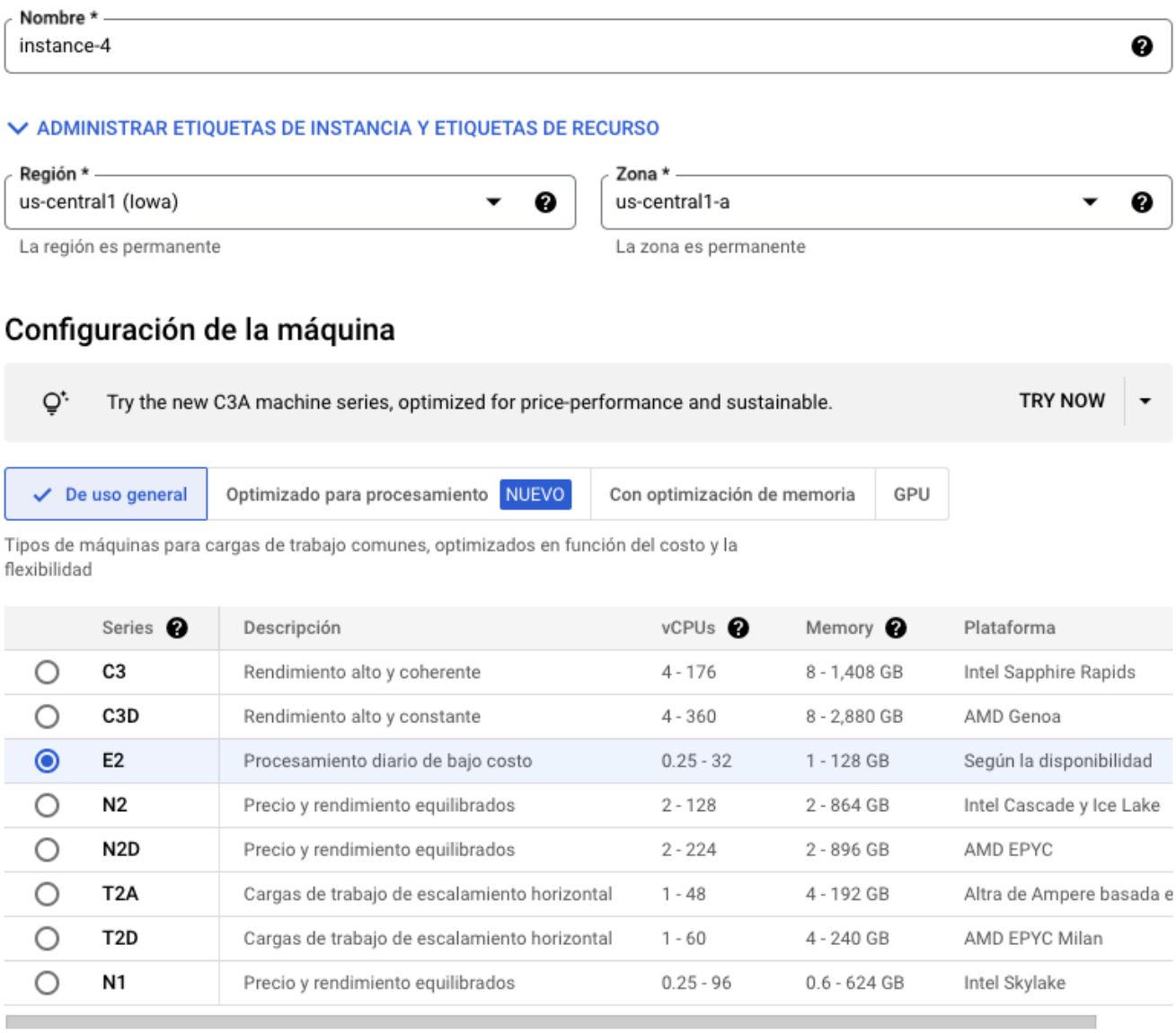
Ir al selector de proyectos

2. Asegúrate de que la facturación esté habilitada para tu proyecto de Google Cloud.
3. Habilita la API de Compute Engine. Habilitar la API de Compute Engine
4. Si no usas Windows en tu máquina local, instala un cliente de RDP, como el Escritorio remoto de Chrome.

Crea una instancia de VM de Windows Server

En la consola de Google Cloud, ve a la página **Crear una instancia**.

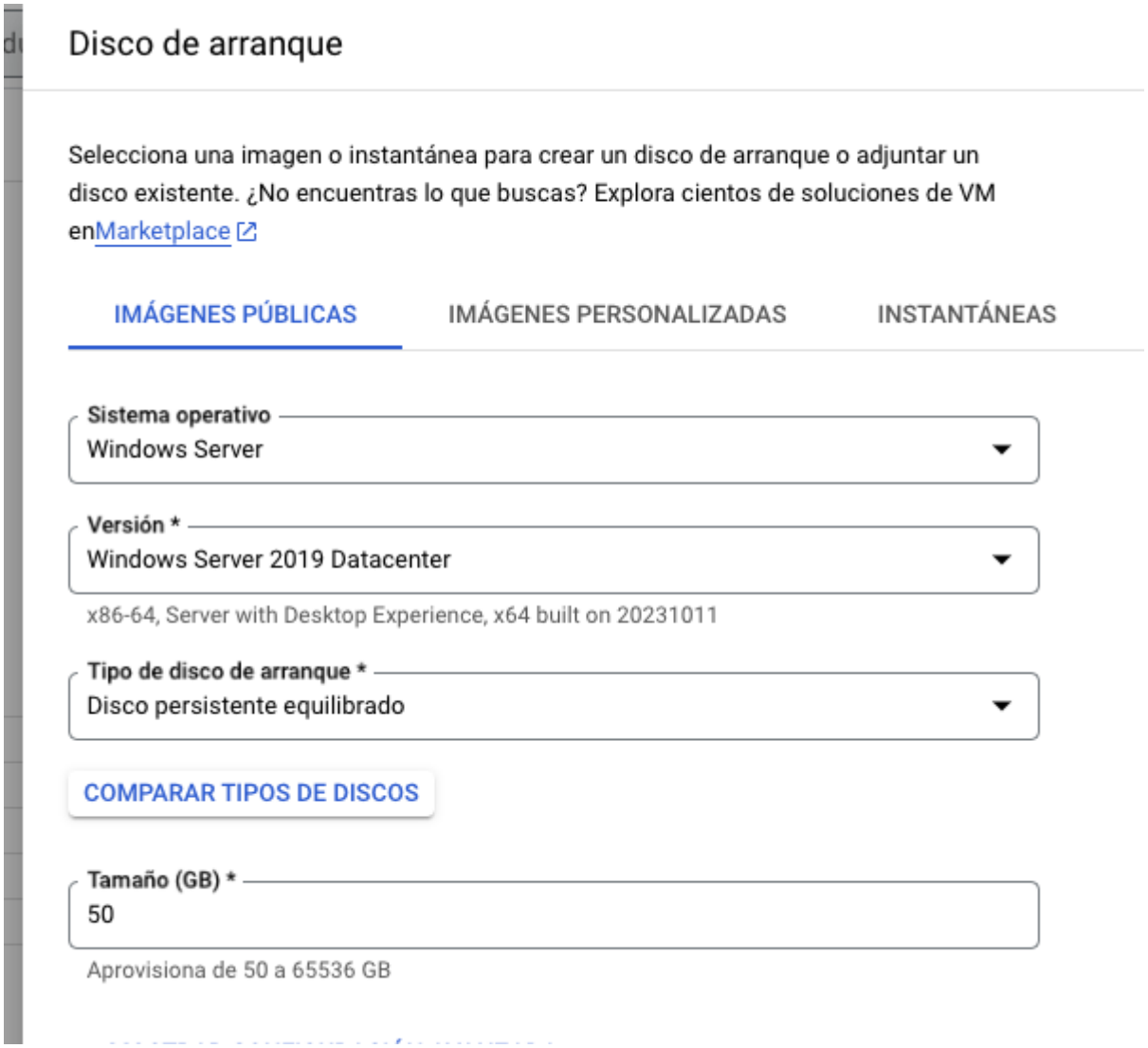
Ir a Crear una instancia



En la sección **Disco de arranque**, haz clic en **Cambiar** para comenzar a configurar el disco de arranque.

En la pestaña **Public images**, selecciona **Windows Server** en la lista **Sistema operativo**.

Elige **Windows Server 2019 Datacenter** en la lista **Versión**.



Haz clic en **Seleccionar**.

En la sección **Firewall**, selecciona **Permitir tráfico HTTP**.

Para crear la VM, haz clic en **Crear**.

Espera un momento hasta que se inicie la instancia de VM. Una vez que la instancia esté lista, se mostrará en la página **Instancias de VM** con un ícono de estado verde.

Conéctate a las VM de Windows mediante

RDP

WINDOWS

En este documento, se describe cómo conectarse a instancias de máquina virtual (VM) de Windows mediante RDP. Para conocer otras formas de conectarse a las VM de Windows, consulta Conéctate a las VM de Windows mediante PowerShell y Conéctate a las VM de Windows mediante SSH.

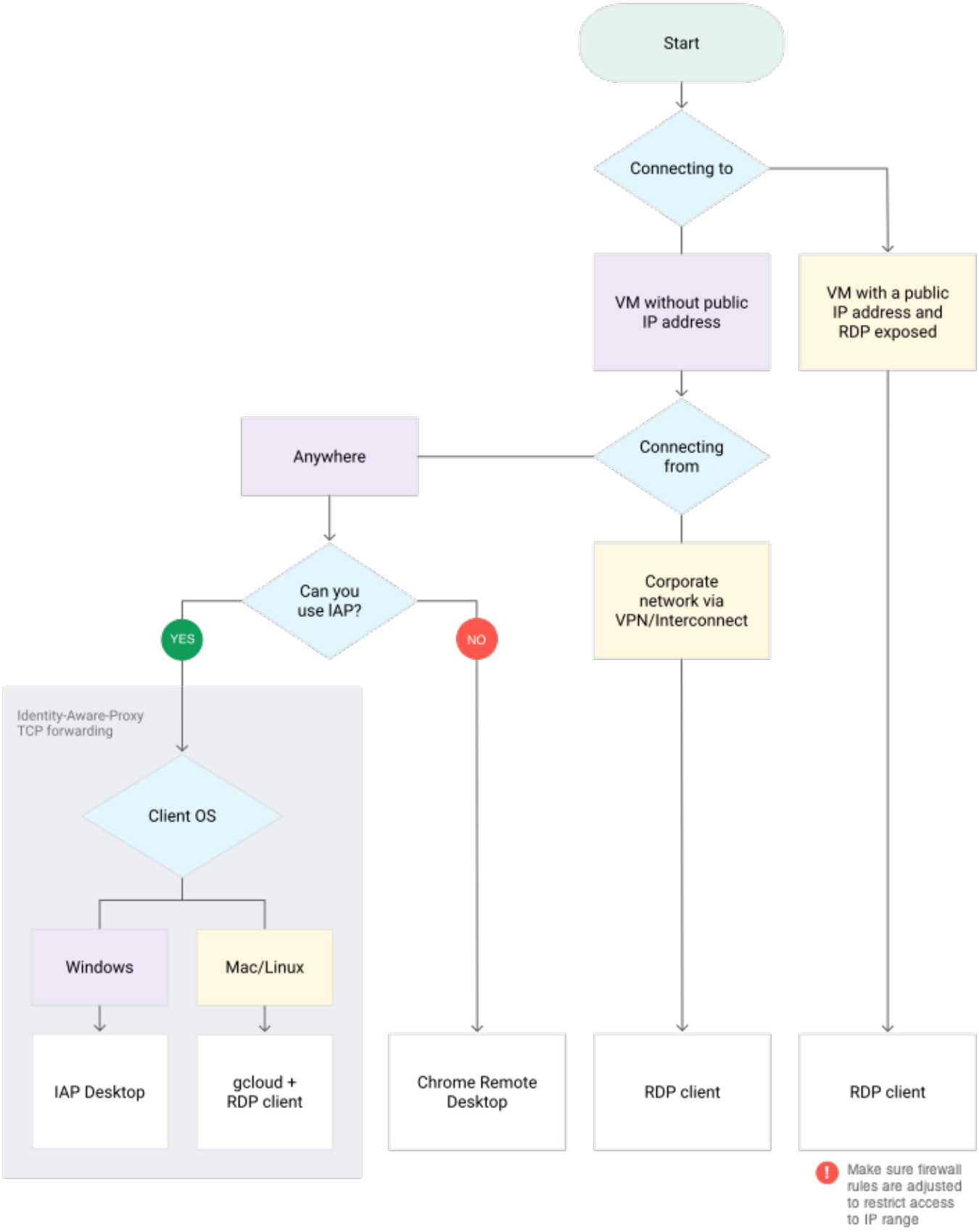
Antes de comenzar

- Si deseas usar los ejemplos de línea de comandos de esta guía, haz lo siguiente:
  1. Instala Google Cloud CLI o actualízala a la última versión.
  2. Configura una región y una zona predeterminadas.

- Asegúrate de que la VM permita el acceso a través del protocolo de escritorio remoto (RDP). De forma predeterminada, Compute Engine crea reglas de firewall que permiten el acceso RDP en el puerto TCP 3389. Para verificar que estas reglas de firewall existan, visita la página de reglas de firewall en la consola de Google Cloud y busca las reglas de firewall que permitan conexiones tcp:3389.

Conéctate a VM de Windows mediante RDP

Compute Engine admite varias formas de conectarte a tus instancias de Windows.



La mejor manera de conectarse al escritorio remoto de una instancia de Windows depende de varios factores:

- Si te conectas desde cualquier lugar de la Internet pública (Conectar desde > **Cualquier** de la ilustración anterior), es mejor habilitar la

redirección de TCP de Identity-Aware Proxy para tu proyecto. Luego, usa IAP Desktop (en Windows) o la CLI de Google Cloud junto con un cliente RDP. Si deseas obtener más información, consulta Clientes de escritorio remoto de Microsoft para conectarte a la instancia de Windows. Si no puedes usar el reenvío de TCP de Identity-Aware Proxy, usa el Escritorio remoto de Chrome.

- Si la instancia de VM tiene una dirección IP pública y las reglas de firewall permiten el acceso de RDP, usa un cliente de RDP. Si deseas obtener más información, consulta Clientes de escritorio remoto de Microsoft para conectarte a la instancia de Windows.
- Si la instancia de VM no tiene una IP pública y te conectas mediante el siguiente comando:Cloud VPN o Cloud Interconnect, puedes conectarte a la dirección IP privada de la VM mediante un cliente RDP. Para obtener más información, consulta Clientes de escritorio remoto de Microsoft.

Si tienes dificultades para conectarte mediante RDP, consulta Solución de problemas de RDP. Si no puedes conectarte a una instancia de Windows mediante el escritorio remoto, consulta Conéctate a la SAC de Windows en este documento.

Para conectarte al escritorio remoto de una instancia de Windows, usa uno de los siguientes procedimientos.

Puedes usar la app de Conexión a Escritorio remoto de Microsoft que forma parte de Windows para conectarte a instancias de Windows.

Antes de conectarte mediante la app de
Conexión a Escritorio remoto de Microsoft, asegúrate de que se cumpla uno de los siguientes requisitos previos:

- Tu instancia de VM tiene una dirección IP pública, y tus reglas de firewall permiten el tráfico de entrada de TCP desde la dirección IP pública de tu cliente hacia la instancia mediante el puerto 3389.
- Tu red local está conectada a tu VPC
mediante Cloud VPN o Cloud Interconnect, y las reglas de firewall permiten el tráfico de entrada de TCP de la dirección IP privada de tu cliente a la instancia mediante el puerto 3389.

Para conectarte con el Escritorio remoto de Microsoft Windows, sigue estos pasos:

1. Crea una cuenta de Windows y una contraseña, si aún no tienes una.

2. Para conectarte mediante Internet, usa la dirección IP *externa*. Para conectarte mediante Cloud VPN o Cloud Interconnect, usa la dirección IP *interna*.

   Para identificar las direcciones IP internas y externas de tu instancia de Windows, completa uno de los siguientes pasos:

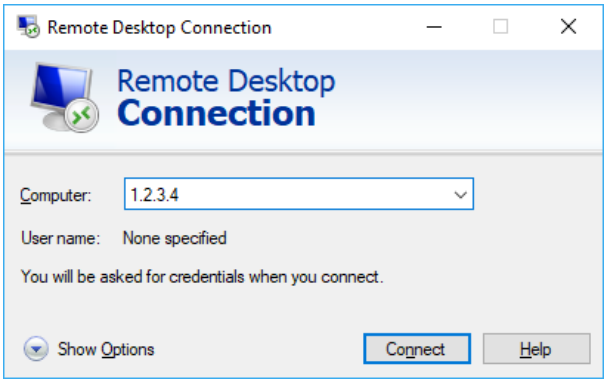
   - En la consola de Google Cloud, ve a la página **Instancias de VM**.

     Ir a la página Instancias de VM

   - Con la CLI de gcloud, ejecuta gcloud compute instances list:

     ```
     gcloud compute instances list
     ```

3. Abre la Conexión a Escritorio remoto de Microsoft Windows en tu máquina con Windows. Puedes encontrar el archivo ejecutable en %systemroot%\\system32\\mstsc.exe



4. En el cuadro **Computer**, ingresa la dirección IP.

   Si configuraste la instancia con el fin de que use un número de puerto diferente para RDP, agrega ese número después de la dirección IP, por ejemplo: 1.2.3.4:3389.

5. Haz clic en **Conectar** (Connect).

6. Ingresa tu nombre de usuario y contraseña, y haz clic en **Aceptar**.

   Si olvidaste la contraseña, puedes restablecerla.

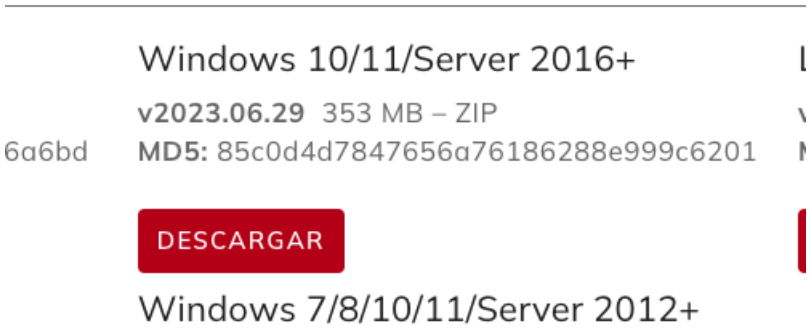
Fuente:
https://cloud.google.com/compute/docs/instances/connecting-to-windows?hl=es-419#remote-desktop-connection-app

Instalar Rocketbot en Windows

1. Para instalar Rocketbot dirijase a https://rocketbot.com/es/rocketbot-studio-rpa/

2. Elija descargar y seleccione la version de Rocketbot para Windows.

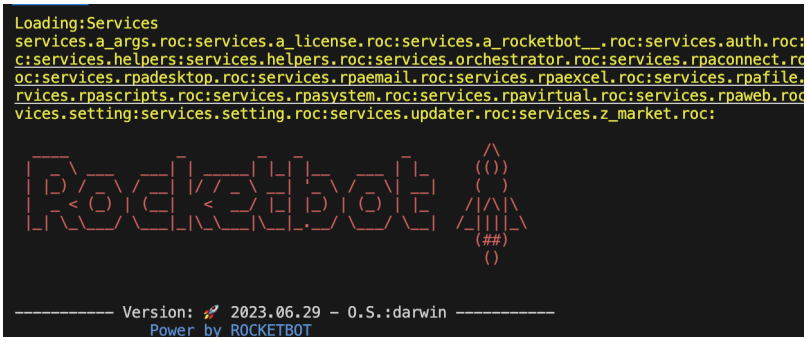
3. Descargue y descomprima el ZIP en un carpeta dentro de la maquina donde ejecutará Rocketbot. por ejemplo: C:\\rocketbot



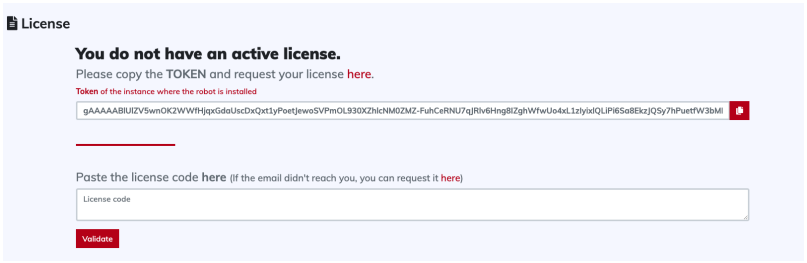
4. Ingrese a la carpeta de Rocketbot y ejecute la aplicacion **rocketbot.exe**



5. Se abrira el navegador por defecto en el puerto 5000 (configurable) y le pedirá una licencia.



6. Copie el Token y complete los datos en la pagina de licencias



7. Recibirá en su correo la licencia. Pegue el código en Rocketbot Studio y comience a programar robots.

Consideraciones y TIPs

Rocketbot Studio/Runner requiere una sesion activa y desktop activo.

Mantener interfaz al desconectarse del servidor

Para evitar estos problemas, los servidores windows cuentan con un comando llamado **tscon.exe** que permite tomar la sesión que tienes abierta en el servidor y moverla a la consola, de esta manera tu robot tendrá una interfaz donde trabajar luego de desconectarte del servidor.

En las pestañas *Desconectarme y ejecutar una tarea* o *Desconectarme y ejecutar ROC* de la aplicación **RDP-Rocketbot** puedes crear un archivo bat con el comando tscon. Este archivo debes ejecutarlo como administrador cada vez que quieras desconectarte del servidor (o ejecutar tu robot, según la opción elegida) y te desconectará del servidor y mantendrá la interfaz gráfica.

Cuidado! Al no abrir el escritorio remoto desde una pantalla física, la resolución puede cambiar y ser menor. Revisa la sección tres para ver como trabajar con esto.

El bat puedes editarlo y revisar que es lo que contiene, pero resumimos lo importante acá. El bat tiene un script de powershell que obtiene el id de la sesión que estas usando al entrar al rdp y la utiliza en el comando tscon para que este la abra en la sesión de la consola.

```
@powershell -NoProfile -ExecutionPolicy unrestricted -Command
"$sessionid=((quser $env:USERNAME | select -Skip 1) -split '\\s+')[2]; tscon
$sessionid /dest:console" 2> UnlockErrors.log
```

Si necesitas que el bat ejecute tu robot al desconectarte, verás algo así


```
@powershell -NoProfile -ExecutionPolicy unrestricted -Command
"$sessionid=((quser $env:USERNAME | select -Skip 1) -split '\\s+')[2]; tscon
$sessionid /dest:console" 2> UnlockErrors.log
```


```
timeout 10
cd c:\\rocketbot
rocketbot.exe -start=nombre_robot
```

Mantener la interfaz al minimizar el escritorio remoto

Para poder minimizar el escritorio remoto y no afectar el funcionamiento del robot, debes modificar algunos registros en la máquina que abre el rdp, estos registros son:

- Para el usuario actual:
  - HKEY_CURRENT_USER\\Software\\Microsoft\\Terminal Server Client (32 bit)
  - HKEY_CURRENT_USER\\Software\\Wow6432Node\\Microsoft\\Terminal Server Client (64 bit)

- Para todos los usuarios:
  - HKEY_LOCAL_MACHINE\\Software\\Microsoft\\Terminal Server Client (32 bit)
  - HKEY_LOCAL_MACHINE\\Software\\Wow6432Node\\Microsoft\\Terminal Server Client (64 bit)

En la pestaña *Conectarme a Escritorio Remoto* de la aplicación RDP-Rocketbot, puedes cambiar los registros con un solo botón. Solo recuerda ejecutar la aplicación como administrador.

Cambio de resolución al desconectarte del servidor

Si ya solucionaste los problemas anteriores, pero tu **robot** falla y/o aún no reconoce las imagenes, es probable que tengas un problema con la **resolución**. Ya mencionamos en la primera parte de este post que al desconectarte el servidor pierde la interfaz gráfica y es necesario utilizar **tscon**. Esto

permite que el robot pueda sacar capturas, pero como el servidor ya no cuenta con tu pantalla para ajustar una resolución, toma sus propias resoluciones por defecto, que en la mayoría de los casos es de 1024×768.

Si tu robot trabaja con **virtualización**, este cambio de resolución puede afectarlo, por lo cual es importante que, antes de comenzar a desarrollar tu robot, revises las resoluciones que permite tu servidor al desconectarte.

En la pestaña *Ver las resoluciones de pantalla* de la aplicación RDP-Rocektbot, puedes ver las resoluciones que permite el servidor en la tabla *Resolucoones de pantalla desde la consola de TSCON*.

Esas resoluciones son las resoluciones que puedes utilizar, pero no quiere decir que puedes comenzar a construir el robot. Debes revisar la resolución que tiene el servidor al desconectarte y si es posible cambia a la resolución que necesitas. Para cambiar la **resolución de pantalla**, puedes utilizar el módulo Windows y también puedes utilizar el comando *Sacar captura de pantall*a de la sección Desktop para válidar si la resolución es la correcta o no.

Ésto lo puedes realizar con **RDP-Helper**, acá encontrarás los pasos necesarios Pasos Resolución

Si al salir de un escritorio remoto no es posible modificar la resolución de forma manual y/o usando el módulo de Rocketbot, puedes modificar realizar la siguiente modificación:

Buscar en el regedit todos los registros que se llamen ***DefaultSettings_XResolution*** y ***DefaultSettings_YResolution*** y cambiar la resolución a la que necesitas

Prevenir el bloqueo de pantalla

Para evitar que el servidor bloquee la pantalla debes, en primer lugar, consultar con el equipo de TI si es posible modificar las configuraciones necesarias para que esto no ocurra, ya que dependiendo de las politicas de seguridad de la empresa, podría no ser posible modificar esta configuración.

En la sección *Configuraciones* de la aplicación RDP-Rocketbot, puedes modificar los registros necesarios para deshabilitar el bloqueo de pantalla. El registro que se modificará es *HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows*

Cambiando Policies Link

For Windows 7/Windows 10/Windows Server 2012 and above:

- Press + and type ***gpedit.msc***

  Windows+R

- Navigate the tree view on the left to ***Computer***

Configuration/Administrative Templates/Windows Components/Remote Desktop Services/Remote Desktop Session Host/Session Time Limits

- Adjust the four settings as desired (I believe you want to adjust ***Set time limit for disconnected sessions*** and ***Terminate session when time limits are reached***)

Habilitar Pantallas Virtuales

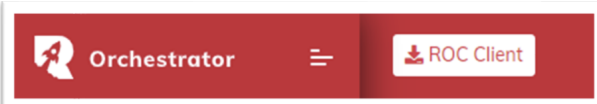
 Si experimentas problemas con la resolución de la VM, consulta la siguiente documentación al respecto.

En algunos casos, la VM solo permitía una resolución muy baja por defecto. Siguiendo la documentación indicada y asegurándonos de **apagar la VM antes de activar la casilla de configuración**, logramos establecer una resolución adecuada para el proceso requerido.

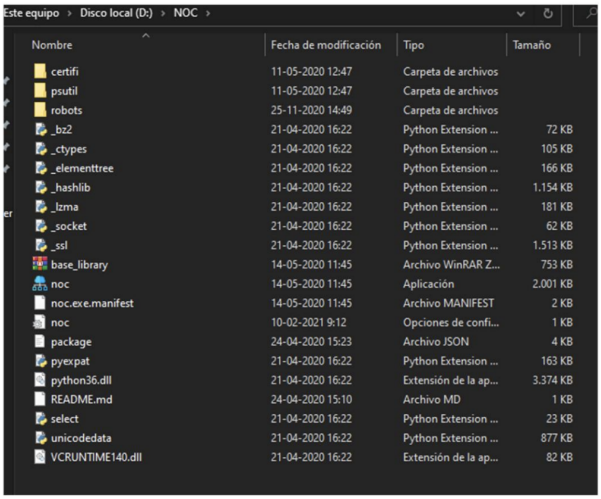
Conectar con Orquestador

Para conectar con el Orquestador de Rocketbot debe contar con una cuenta habilitada de Rocketbot Orchestrator Center. En la VM de GCP tiene que tener habilitado el acceso a las web ***.myrb.io** y ***.rocketbot.com** esto permitirá al cliente de orquestador NOC conectarse con las url de api de orquestador.

INSTALACIÓN Y CONFIGURACIÓN CLIENTE ORQUESTADOR

1. Descargar cliente desde el orquestador: Haga click en el boton del menu superior del orquestod “ROC Client” para descargar el ZIP con el cliente.


2. Descomprima noc.zip en una carpeta donde tenga permisos de escritura, no se recomienda que sea en la carpeta personal o desktop. Recomendacion : C:\\rocketbot\\noc



3. Configurar archivo noc.ini con:
   a. Credenciales de usuario con un ROL con permisos limitados



   b. Key de instancia: Ingrese al proceso dentro del proyecto, una vez adicionado una instancia le mostrará una Key



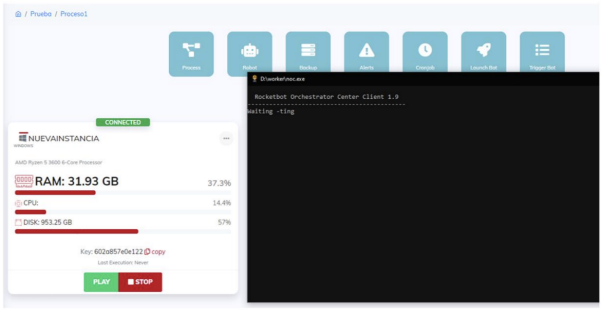
   c. Url de orquestador, ej: https://roc.myrb.io/s4 (sin / final) d. Ruta de instalación de Rocketbot, incluyendo ejecutable. por ejemplo c:\\rocketbot

```
[USER]
# user access
user =
password =
#or API Key
apikey = eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.xxx.u3PJ2eed_pDOGfrGpVe_tjJJDIhM1YSlBm
# Instance ID
key = xxxxxx


[NOC]
# you Url Orchestrator
# https://devnoc.myrb.io
# https://roc.myrb.io/s1
# https://roc.myrb.io/s2
# https://roc.myrb.io/s4
# https://roc.myrb.io/dev
server = https://roc.myrb.io/dev      You, 6 months ago • 🔧 chore(noc.ini): update
proxy =
#Send logs to Orchestrator?
logs = true
[ROCKETBOT]
# Rocketbot bin
path =/Users/lucianodavidcuello/projects/rocketbot/dist/rocketbot/rocketbot
# No debug on Rocketbot?
nodebug = false

#Screenshot available ?
screenshot = true
#Command line options
comandline = --update-drivers
```

4. Ejecute noc.exe y el cliente de orquestador mostrara que la instancia está conectada.



Server con Múltiples Usuarios

La duda recurrente es “¿Cómo es el licenciamiento en una máquina donde acceden múltiples usuarios al mismo tiempo?”.

Para este caso tenemos 3 opciones.

Opción 1:

Primero debemos revisar si dejando ***Rocketbot Studio*** en la raíz, pueden acceder todos los usuarios y utilizarlo, si es así, entonces sería 1 Licencia con X cantidad de ejecuciones en paralelo (la cantidad según el número de usuarios que se conectarán)

Opción 2:

Si no permite acceder a cada usuario a ***Rocketbot Studio*** instalado en la raíz (Opción 1), entonces debe ser una licencia por cada ambiente de usuario.

Opción 3:

Con la licencia Online se puede conectar cada usuario en cualquier máquina con ***Rocketbot Studio instalado***, siempre y cuando tenga disponible la cantidad de conexiones necesarias.

Como actualizar a Rocketbot 2023

Las versiones anteriores a Rocketbot 2023.03.20 no cuentan con actualizaciones automáticas, por lo que se necesita actualizar manualmente a la última versión de rocketbot. En este post explicaremos los pasos que debes realizar para actualizar Rocketbot

Ver updates

Migración de archivos

Para mantener la licencia y los robots, tienes que copiar desde tu versión anterior a la versión 2023 los siguientes archivos

- license.json: Este archivo contiene tu licencia
- robot.db: Este archivo contiene la base de datos de robots que viene por defecto en Rocketbot

Actualización de módulos

La versión 2023.03.20 realiza una actualización de la versión de python que utiliza. Esto puede generar conflictos con algunos módulos de las versiones anteriores, por tanto, no recomendamos copiar la carpeta modules de versiones anteriores, sino instalar las últimas versiones disponibles en el marketplace

Para actualizar o instalar los módulos que necesitan tus robots, puedes ir a la pestaña Robot de tu robot y al final de la página, instalar o actualizar los módulos correspondientes

Sección de módulos en la pestaña Robot

También puedes buscarlo desde el addon Install Mods o desde nuestro Marketplace

Migración de Robots

La última versión, además de actualizar el Python interno de 3.6.8 de 32bits a Python 3.10.4 de 64 bits también corrige el funcionamiento de alguno de los comandos nativos. Esto puede ocacionar que robots desarrollados en versiones anteriores, no funcionen correctamente en la última versión. Ya sea por correciones para bugs de comandos o por librerías externas de Python instaladas en Rocketbot.

Para migrar los robots recomendamos lo siguiente:

- Para robots en producción, no actualizar a la nueva versión a menos que sea necesario. Puedes mantener las dos versiones de Rocketbot funcionando en la misma máquina
- Actualiza el Python de tu máquina a Python 3.10 de 64 bits e instala las

librerías que utilizan tus robots en la carpeta “modules/libs” de la nueva versión
- Actualiza a la última versión los archivos que indica el updater
- Si quieres migrar un robot productivo, realiza pruebas con la nueva versión antes de actualizar

Url utilizadas por el Updater

Las actualizaciones automáticas son revisadas y descargadas directamente desde el repositorio de Github de Rocketbot. Si tienes restricciones a URLs públicas, te dejamos un listado de las urls que necesita rocketbot para actualizar

- Consulta de nuevas versiones: https://raw.githubusercontent.com/rocketbot-cl/rocketbot/main/package.json
- Descarga de archivos: https://github.com/rocketbot-cl/rocketbot/raw/main/updates/*
- Actualización de drivers de Chrome: https://chromedriver.storage.googleapis.com/
- Actualización de drivers de Firefox: https://github.com/mozilla/geckodriver/releases

---

RPA on Virtual Server

Depending on the process you wish to automate and how you intend to run the **robots**, your **server** will need to meet specific requirements for the robot to operate well.

Processes can be divided into two types: those that require a **graphical interface** (robots that take screenshots or use the keyboard and mouse) and those that do not (robots that emulate a terminal, automate xlsx files, use an API, etc.). It is critical to consider this when developing a robot because most servers **lose the graphical interface** when disconnected, causing robots to fail.

Screen locking and **minimizing the remote desktop** can also have an impact on your robot’s performance. The operating system should never lock the screen on a machine running an automated process (**RPA**), as the robot will be **unable** to take **screenshots** or move the mouse to the indicated places. When you minimize the rdp, Windows may pause some processes or remove the interface, so you must configure your machine to prevent this from happening.

In **Rocketbot**, we have developed an application that allows you to easily configure your server, creating the necessary.bat files to run your robot and/or modifying the **registers** to prevent screen problems. You can download this application here: RDP-Rocketbot. Remember to save it to the server

and/or machine as needed.

Maintain interface when server disconnects.

To avoid these issues, Windows servers have a command called **tscon.exe** that allows you to take an open session on the server and move it to the console, giving your robot an interface to work with after disconnecting from the server.

The tscon command can be used to create a bat file in the RDP-Rocketbot application's *Disconnect and run a task* or Disconnect and run ROC tabs. This file must be run as administrator each time you want to disconnect from the server (or run your robot, depending on the option selected), and it will disconnect you from the server while maintaining the graphical interface.

Beware! If the remote desktop is not opened from a physical screen, the resolution may change and become lower. Refer to section three for alternatives.

You can edit the bat and examine its contents, but we have summarized the crucial information here. The bat contains a powershell script that retrieves the session id of the session you are using when logging into the rdp and uses it in the tscon command to open the console session.

```
@powershell -NoProfile -ExecutionPolicy unrestricted -Command
"$sessionid=((quser $env:USERNAME | select -Skip 1) -split '\s+')[2]; tscon
$sessionid /dest:console" 2> UnlockErrors.log
```

If you need the bat to run your robot after disconnecting, you will see something like this:

```
@powershell -NoProfile -ExecutionPolicy unrestricted -Command
"$sessionid=((quser $env:USERNAME | select -Skip 1) -split '\s+')[2]; tscon
$sessionid /dest:console" 2> UnlockErrors.log

timeout 10
cd c:\rocketbot
rocketbot.exe -start=nombre_robot
```

Maintain interface when minimizing remote desktop

To minimize the remote desktop without interfering with the robot's operation, modify the following registers in the machine that opens the rdp:

- For the current user:
  - HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client (32 bit)

    - HKEY_CURRENT_USER\Software\Wow6432Node\Microsoft\Terminal Server Client (64 bit)
- For all users:
    - HKEY_LOCAL_MACHINE\Software\Microsoft\Terminal Server Client (32 bit)
    - HKEY_LOCAL_MACHINE\Software\Wow6432Node\Microsoft\Terminal Server Client (64 bit)

You can change the logs with a single click in the RDP-Rocketbot application's Connect to Remote Desktop tab. Just keep in mind that you must run the application as an administrator.

Change in resolution after disconnecting from the server

If you have already resolved the previous issues, but your **robot** still fails and/or does not recognize the images, you most likely have a **resolution** issue. We already mentioned in the first section of this post that when you disconnect the server, the graphical interface is lost and you must use **tscon**. This enables the robot to take screenshots, but because the server no longer has access to your screen to set a resolution, it uses its own default resolution, which is 1024×768 in most cases.

If your robot uses **virtualization**, this resolution change may have an impact, so it is critical that you check the resolutions that your server allows when you disconnect before beginning to develop your robot.

The resolutions allowed by the server in the Screen resolutions table from the TSCON console can be seen in the *View screen resolutions* tab of the RDP-Rocketbot application.

Those are the resolutions you can use, but that does not mean you can begin building the robot. When you disconnect, you should check the resolution of the server and, if possible, change it to the resolution you require. You can use the Windows module to change the **screen resolution**, and you can also use the Take screenshot command in the Desktop section to see if the resolution is correct.

A sample robot can be downloaded here

test_resolucionDownload

If when exiting the remote desktop it is not possible to change the resolution manually and/or using the Rocketbot module, you can try the following:

Change the resolution in regedit by searching for all registries named **DefaultSettings XResolution** and **DefaultSettings YResolution.**

Prevent screen lock

To prevent the server from blocking the screen, you should first consult with the IT team to see if it is possible to modify the necessary settings to prevent this from happening, as this may not be possible depending on the company's security policies.

In the Settings section of the RDP-Rocketbot application, you can modify the registers needed to disable the screen lock. The registry to be modified is *HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows*

Changing Policies Link

Instalacioń Rocketbot en macOS

1. Descargar Rocketbot Studio

Ingresar a la página de Rocketbot, clickear el botón **Descargue Rocketbot** y elegir el sistema operativo macOS.

2. Descomprimir

Una vez descargado, descomprimimos el archivo .zip, obtendremos una carpeta llamada Rocketbot.

3. Ejecutar Rocketbot

Luego de descomprimir podemos dejar la carpeta donde estimemos conveniente, en este ejemplo quedará en Desktop.

Abrir la carpeta y dar doble click en el archivo ejecutable llamado "**rocketbot**"

3. Conceder permisos

Al dar doble click en el ejecutable, nos mostrará un mensaje de aviso donde indica que no se puede abrir la aplicación ya que no es un desarrollador identificado.

Realizaremos los siguientes pasos:

- Damos click en **OK**



- Ir a **Preferencias del Sistema – Seguridad y Privacidad**

Nos aparecerá que el uso de rocketbot se bloqueó, debemos dar click en "**Abrir de todos modos**".



Nos aparecerá el siguiente mensaje, damos click en "**Abrir**".



- Esto nos abrirá nuestro navegador predeterminado con la dirección **localhost:5000**



Ya podemos solicitar la licencia en: http://license.rocketbot.co/

5. Consideraciones Extra

Si además de permitir que abra Rocketbot, nos arroja alertas de Python o de alguna librería como estos ejemplos:





Debemos cancelar la alerta y dar los permisos de seguridad correspondientes.

- En la ventana de "**Seguridad y Privacidad**", ir a la pestaña "**Privacidad**", en el menú izquierdo hacer scroll y seleccionar "**Herramientas para desarrolladores**".



Si no tenemos la opción de **Herramientas para desarrolladores** (Developer Tools), debemos abrir una ventana de Terminal y escribir el siguiente comando spctl developer-mode enable-terminal con eso ya nos debe aparecer

- También debemos instalar xcode con la siguiente instrucción desde el Terminal xcode-select --install

Nos aparecerá una ventana indicando que debemos instalar las tools, clickeamos **Install** y esperamos que termine



- Volvemos a Seguridad y Privacidad y clickeamos el candado de la esquina inferior izquierda para poder realizar cambios.



Esto nos pedirá la contraseña de nuestro usuario, la ingresamos y seleccionamos “**Desbloquear**”.



- Ahora podemos dar permisos al **Terminal**, en el caso que tengamos además otro terminal (ej: **iTerm**) y lo utilicemos para ejecutar rocketbot, también se debe seleccionar, luego seleccionamos el candado para guardar los cambios.



- Finalmente podemos ejecutar rocketbot nuevamente y ya nos abrirá **localhost:5000** en nuestro navegador predeterminado.

  Si al abrir localhost:5000 nos aparece un error de acceso denegado a localhost, debemos cambiar el puerto de Rocketbot



Para cambiar el puerto debemos ir a la carpeta de Rocketbot y abrir el archivo rocketbot.ini



y cambiar donde dice port, elegimos otro puerto, por ejemplo el 5003



quedando de esta forma



Volvemos a ejecutar rocketbot y ahora ya nos cargará en el nuevo puerto pidiéndonos activar la licencia



Automatización de Aplicaciones de Escritorio

Las aplicaciones de escritorio no son tan estandarizadas como los son las aplicaciones web, mientras que para aplicaciones web utilizamos en su mayoría comandos que **automatizan** el navegador, para aplicaciones de escritorio tenemos diferentes métodos según el tipo de aplicación. En esta entrada te mostraremos las **consideraciones** y **métodos** que debes conocer antes de comenzar a desarrollar un robot para automatizar **aplicaciones de escritorio**.

Procesos que pueden presindir de una aplicación

Existen algunos procesos que si bien utilizamos una aplicación para realizarlos, existen métodos que permiten a los robots hacerlo de forma más eficiente. Recuerda que automatizar no es solo hacer robots que repliquen el comportamiento humano frente a la computadora.

En este tipo de aplicaciones se encuentran aquellas actividades que se pueden realizar por terminal o scripts bash, aplicaciones que tienen su versión web y que cuentan con una API o que trabajan bajo algún protocolo. Estos procesos pueden ser:

- Automatización de correos electrónicos por conexión imap, smtp, pop.
- Manipulación de archivos como mover, renombrar, leer, crear, etc.
- Consultas a base de datos
- Aplicaciones con versión web como Slack, Google Drive, Telegram, One Drive
- Conexiones ssh o telnet
- Aplicaciones simuladas en una terminal

No necesitas tener instalada la aplicación para automatizar estos procesos, solo debes descargar el módulo correspondiente en nuestro Marketplace. Estos procesos en su mayoría permiten ejecuciones en paralelo.

Aplicaciones con su propio sistema de scripting

Algunas empresas ya conocen el poder de la automatización y nos permiten interactuar con sus aplicaciones a través de su propio sistema de scripting. De esta manera, podemos controlar la aplicación sin necesidad de utilizar el mouse o teclado y utilizar los métodos internos que tiene cada aplicación

para automatizar nuestro proceso.

Para automatizar estas aplicaciones es necesario tener instalada la aplicación en la misma máquina donde se ejecutará tu robot. Aplicaciones de este tipo son: SAP, SAPB1, Excel, Word, Outlook y puedes encontrar módulos para automatizarlos en Marketplace. Si conoces alguna aplicación que necesitas automatizar y esta tiene sistema de scripting, comunicate con nosotros y desarrollamos el componente.

Dependiendo de la aplicación, puedes ejecutar más robots al mismo tiempo. Revisa si la aplicación permite abrir múltiples instancias.

Aplicaciones Java o Windows Forms

Aunque algunas aplicaciones no tengan su propio sistema de scripting, si pueden ser automatizados de esta forma a través de las dll de Java o Windows. Automatizar una aplicación por scripting es similar a automatizar una página web, solo necesitas obtener información del objeto que quieres controlar, como el título, la clase, el id, el tipo de objecto, etc.

Esto no puede obtenerse directamente de la aplicación, pero puedes usar alguno de nuestros grabadores como el DesktopRecorder o JavaRecorder. Con esta herramienta puedes grabar las acciones que quieres realizar en la aplicación para luego replicar lo grabado en un robot. No es sencillo determinar que aplicaciones permiten automatizarlo por scripting. La mejor recomendación es descargar los grabadores y revisar si detectan tu aplicación.

La aplicación debe estar instalada en la misma máquina donde se ejecutará el robot y al hacer uso del mouse y/o teclado, no se puede trabajar con robots en paralelo que automaticen el mismo proceso

Otras aplicaciones

Si tu aplicación no se puede automatizar de las formas mencionadas anteriormente, puedes automatizar la aplicación por virtualización y/o comandos de teclado. Para esto, es necesario tener algunas consideraciones:

- Si el robot se ejecutará en un servidor, revisa la entrada RPA en servidores virtuales
- Si la aplicación permite utilizar combinaciones de tecla, utiliza el teclado por sobre virtualización. Puedes conocer los shortcut de la aplicación al ver una letra subrayada bajo la palabra donde quieres hacer click. Para replicar en Rocketbot, revisa la siguiente entrada Enviar Tecla
- Si se debe extraer información, revisa si puedes usar el portapapeles para copiar el contenido.

- Como se sacarán capturas de pantalla, se utilizará el mouse o el teclado. No se puede automatizar la misma aplicación en paralelo

---

Ejecutar en Paralelo

Rocketbot permite ejecuciones en paralelo, lo que quiere decir que podemos lanzar más de 1 robot al mismo tiempo.

Uno de los requisitos es contar con una licencia de Producción que permita este tipo de ejecuciones.

Al iniciar nuestro Rocketbot Studio, nos informará en la consola cuántos robots en paralelo nos permite:



> Tener en cuenta que al estar Rocketbot Studio abierto es considerado un proceso, por lo tanto si tengo un máximo de 2 procesos en paralelo, y requiero ejecutar 2 robots, debo cerrar el Studio.

Para iniciar más de un robot, lo podemos realizar a través de un .bat si estamos en Windows o desde un .sh si estamos en macOS/Linux.

Windows

El archivo .bat debe tener lo siguiente:

```
cd C:/ruta/a/carpeta/Rocketbot
START rocketbot.exe -start=robot1
START rocketbot.exe -start=robot2
```

macOS/Linux

El archivo .sh debe tener lo siguiente:

```
cd /ruta/a/carpeta/Rocketbot
./rocketbot -start=robot1 &
./rocketbot -start=robot2
```

Luego podemos ejecutar el .bat/.sh mediante un doble click o agregarlo a una tarea programada.

Para saber que procesos se pueden ejecutar en paralelo revise este link ¿que considerar para ejecucion en paralelo?

Filtros Email

ALL

All messages in the mailbox.

ANSWERED

Messages with the \Answered flag set.

BCC

Messages that contain the specified string in the envelope structure’s BCC field.

BEFORE

Messages whose internal date (disregarding time and timezone) is earlier than the specified date.

BODY

Messages that contain the specified string in the body of the message.

CC

Messages that contain the specified string in the envelope structure’s CC field.

DELETED

Messages with the \Deleted flag set.

DRAFT

Messages with the \Draft flag set.

FLAGGED

Messages with the \Flagged flag set.

FROM

Messages that contain the specified string in the envelope structure’s FROM field.

HEADER

Messages that have a header with the specified field-name (as

defined in [RFC-2822]) and that contains the specified string in the text of the header (what comes after the colon). If the string to search is zero-length, this matches all messages that have a header line with the specified field-name regardless of the contents.

KEYWORD

Messages with the specified keyword flag set.

LARGER

Messages with an [RFC-2822] size larger than the specified number of octets.

NEW

Messages that have the \Recent flag set but not the \Seen flag. This is functionally equivalent to “(RECENT UNSEEN)”.

NOT

Messages that do not match the specified search key.

OLD

Messages that do not have the \Recent flag set. This is functionally equivalent to “NOT RECENT” (as opposed to “NOT NEW”).

ON

Messages whose internal date (disregarding time and timezone) is within the specified date.

OR

Messages that match either search key.

RECENT

Messages that have the \Recent flag set.

SEEN

Messages that have the \Seen flag set.

SENTBEFORE

Messages whose [RFC-2822] Date: header (disregarding time and timezone) is earlier than the specified date.

SENTON

Messages whose [RFC-2822] Date: header (disregarding time and timezone) is within the specified date.

SENTSINCE

Messages whose [RFC-2822] Date: header (disregarding time and timezone) is within or later than the specified date.

SINCE

Messages whose internal date (disregarding time and timezone) is within or later than the specified date.

SMALLER

Messages with an [RFC-2822] size smaller than the specified number of octets.

SUBJECT

Messages that contain the specified string in the envelope structure’s SUBJECT field.

TEXT

Messages that contain the specified string in the header or body of the message.

TO

Messages that contain the specified string in the envelope structure’s TO field.

UID

Messages with unique identifiers corresponding to the specified unique identifier set. Sequence set ranges are permitted.

UNANSWERED

Messages that do not have the \Answered flag set.

UNDELETED

Messages that do not have the \Deleted flag set.

UNDRAFT

Messages that do not have the \Draft flag set.

UNFLAGGED

Messages that do not have the \Flagged flag set.

UNKEYWORD

Messages that do not have the specified keyword flag set.

UNSEEN

Messages that do not have the \Seen flag set.

Problema Diploma Academy

Aveces existe problemas de caché cuando se quiere ver o descargar el diploma de Academy, para solventarlo deben realizar lo siguiente:

1.- Inspeccionar la página

2.- Ir a la pestaña Network

3.- Desactivar el caché si aún no se tiene marcado

4.- Realizar ctrl +r (o command + r en mac) un par de veces

5.- Esperar que la página recargue y volver a clickear el botón del Diploma

dejamos un video de los pasos a seguir:

http://docs.rocketbot.com/wp-content/uploads/2021/07/Cert_Academy.mp4

Manejo de DatePicker

La mayoría de los datapickers no permiten modificar el input de la fecha de forma sencilla, ya sea enviando con el comando de *enviar texto web* o modificando el *value* con Javascript.

Dependiendo de la librería o framework que utilice la página que quieres automatizar, será el método que debes seguir.

Angular

Para saber si la web está desarrollada en angular, el input que laza el datepicker debe tener algún atributo que comience con *ng-*, por ejemplo:

<input id="datepicker_id" type="date" ng-model="mov.calendar.fromDate" ng-change="mov.update"/>

Para modificar datepicker, debes seguir los siguientes pasos:

1. Copiar el JS Path (igual como copias el XPath) y almacenarlo en una variable, en el ejemplo, la variable se llama element .
2. Obtener el nombre del modelo. En el input de ejemplo, el modelo (*ng-model*) es *mov.calendar.fromDate.* Esta es la variable que debe modificarse con la fecha, lo escribimos después de scope() (ver ejemplo)
3. Si el input tiene el atributo *ng-change,* se debe repetir lo mismo que en el paso anterior, invocando la función.
4. Escribir el script de ejemplo en el comando *ejecutar javascript* reemplazando con los valores correspondientes.

Ejemplo:

```
element = document.querySelector("#datepicker_id") // Este es el JS Path
 angular.element(element).scope().mov.calendar.fromDate = "{fecha}" // Acá
modificamos la fecha
 angular.element(element).scope().mov.update() // Acá aplicamos el cambio
```

Ember

Para saber si una aplicación está desarrollada con ember, debemos buscar si al inicio del body de la página, tiene algún id o clase con la palabra ember.

Para modificar el datepicker, se deben seguir estos pasos:

1. Obtener el nombre que aparece en la ruta de la página, luego del #. Ejemplo, si la ruta es *tupagina.com/#/reporte-control,* el nombre del control será *reporte-control*.
2. Obtener el id del input que se quiere modificar.
3. Escribir el script de ejemplo en el comando *ejecutar javascript* reemplazando con los valores correspondientes.

Input de ejemplo: <input id="fechaInicial" type="date"/>

Script de ejemplo:

```
function getApplication() {
 var namespaces = Ember.Namespace.NAMESPACES, application;
 namespaces.forEach(function(namespace) {
  if(namespace instanceof Ember.Application) {
   application = namespace;
   return false;
  }
 });
 return application;
}
```

```
app = getApplication() // obtenemos la aplicación de ember
controller = app.__container__.lookup('controller:reporte-control')
//obtenemos el controlador
controller.set('fechaInicial', new Date("{fechaInicial}")) // modificamos la
fecha
```

Nuxtjs

Para saber si una aplicación está desarrollada con **nuxt** , debemos buscar si al inicio del body de la página, tiene algún id con la palabra nuxt, por ejempo <Div id="__nuxt">.

Para modificar el datepicker, se deben seguir estos pasos:

1. buscar el elemento con atributo __vue__
2. Si no está, entonces buscar el ancestro que lo tenga y desde el atributo __vue__ buscar el Children(hijo)
3. luego buscar el modelo y ejecutar el callback con la fecha

Ejemplo:

```
element = document.getElementById('id').__vue__// buscar el elemento con
atributo __vue__
element =
document.getElementById('id_ancestro')._vue__.children[pos_element]// Si no
está, entonces buscar el ancestro que lo tenga y desde el atributo __vue__
buscar el Children(hijo)
element.data.model.callback(new Date(mm-dd-yyyy))// luego buscar el modelo y
ejecutar el callback con la fecha
```