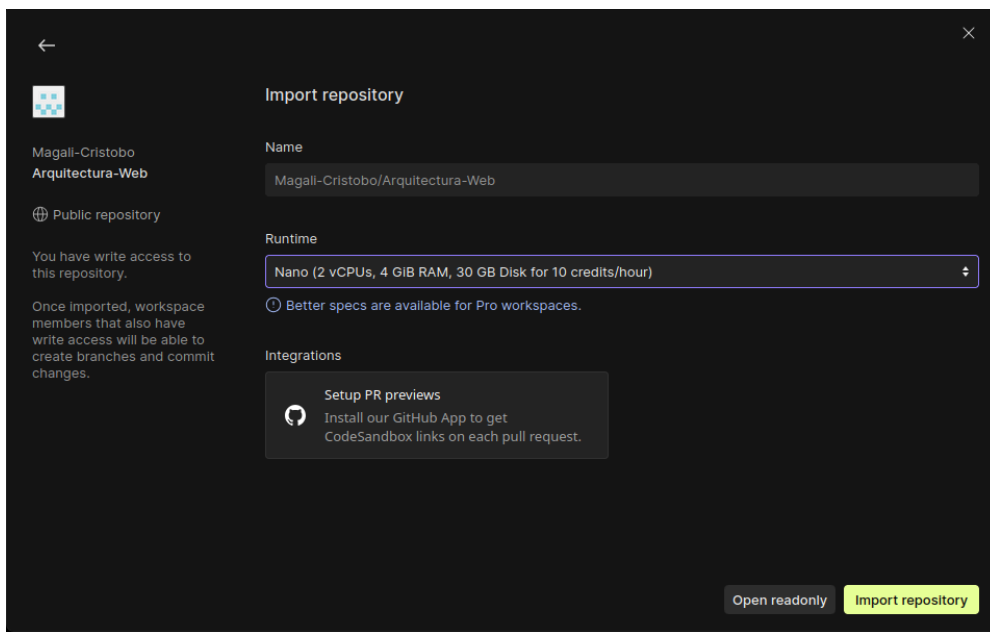


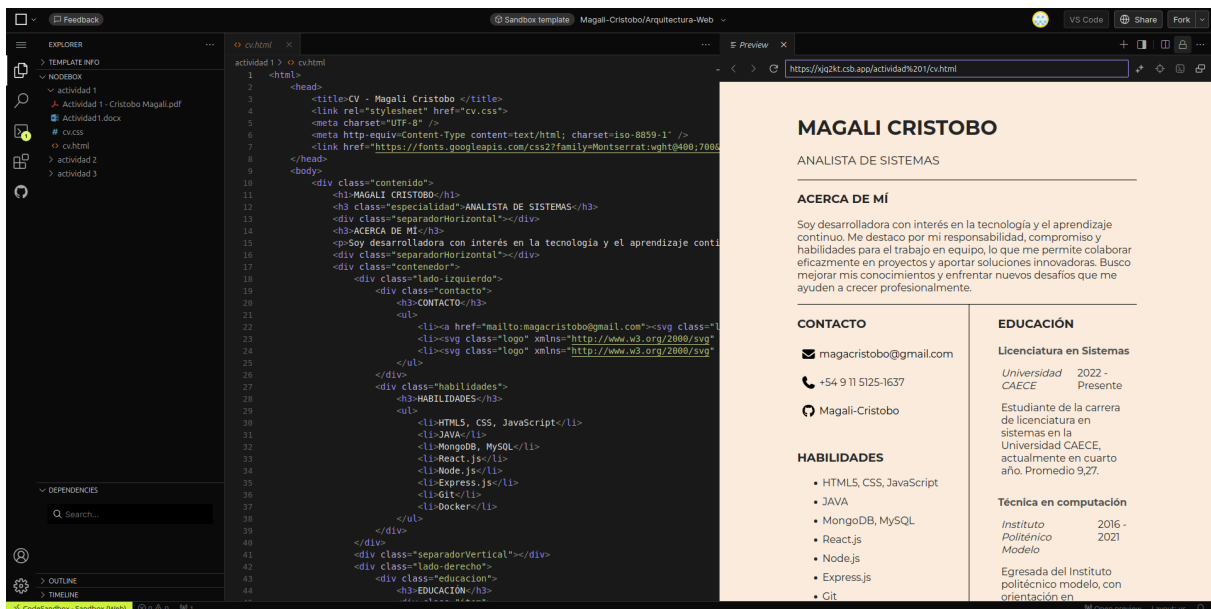
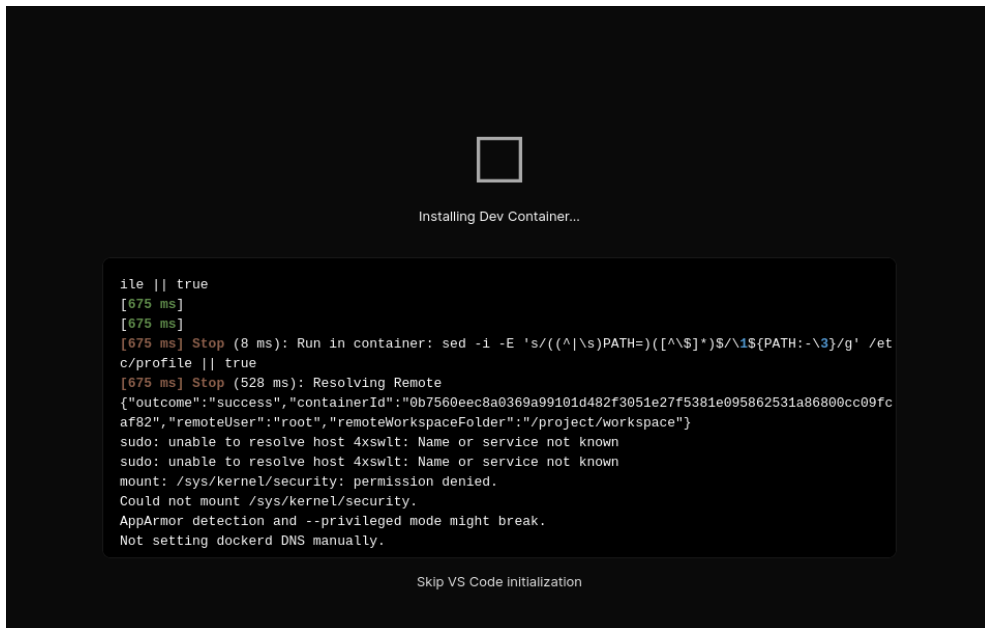
Actividad 4: Ambiente de desarrollo en la nube

Como plataforma de desarrollo utilicé codesandbox.io porque tuve problemas para crearme una cuenta en codeanywhere, no me llegaba el sms de confirmación por más de que me registrara con gmail o github.

Configuración del entorno

La plataforma te permite crear un “sandbox” que sería un entorno de desarrollo aislado y preconfigurado para diversos lenguajes y frameworks. Otro modo que ofrece es el de importar un repositorio, que fue el que elegí. Por más de que permite crear un sandbox con un template para proyectos con HTML y CSS me pareció mejor idea tenerlo vinculado al repositorio para poder tener un control de los cambios y que se desplegara automáticamente luego de hacer un push en la rama.





De esta manera pude utilizarlo como editor de código y también como servidor para alojar el sitio. Me brindó la url <https://codesandbox.io/p/sandbox/github/Magali-Cristobo/Arquitectura-Web> para poder visualizarlo de forma pública.

Aclaración: como en el repositorio tengo varias carpetas y el html está dentro de la carpeta actividad 1, al ingresar al link en el preview se muestra una pantalla en blanco. Ahí se debe modificar la url del preview y agregar "/actividad 1/cv.html".

Comparación entre escenarios

Las ventajas de utilizar este proveedor en contraposición de un servidor o entorno local son:

- Facilidad de despliegue: solo tuve que importar el repositorio y darle permisos. El despliegue se hace automático cuando se hace push en una rama del repositorio. Localmente tardaría más tiempo configurando el servidor apache o nginx.
- Ahorro de recursos locales: nos saca la responsabilidad de pensar en un hardware o servidor donde alojarlo, al igual que un beneficio en relación a los recursos ya que no estamos utilizando los de nuestra máquina local.
- Colaboración con varios miembros.
- Costo inicial bajo: para proyectos de educación o personales es gratis.

Las desventajas que observé son:

- Dependencia en el proveedor: si tiene alguna falla nuestra aplicación no va a ser accesible a través de internet.
- No tenemos tanto control sobre la arquitectura: nos estamos desligando de ella al hostear la aplicación en el hardware del proveedor.
- Limitación de recursos en planes gratuitos: muchas veces los planes gratis tienen una limitación de recursos por lo que es muy probable que tengamos que pagar algún plan más elevado para aplicaciones con mayor número de clientes.

En mi opinión, para proyectos personales, educativos o que requieran una prueba rápida usaría un ambiente de desarrollo en la nube. Si ese proyecto comienza a crecer y contiene bases de datos con datos sensibles o mayor consumo de recursos, creo que lo mejor es un enfoque híbrido en el que se tenga un servidor dedicado para la aplicación en producción, y se complemente con herramientas en la nube como github para gestionar el versionado del código, cambios y para automatizar despliegues.