



Gitpod

Evaluación de entornos integrados de desarrollo en la nube
(Cloud IDE) para la generación de ejecutables y gestión de
código.

RA2a, RA2b, RA2c, RA2d, RA2g

I. Instalación y Requisitos

Entendiendo el concepto de Cloud Development Environment (CDE)

Un IDE que no se instala

Naturaleza Efímera

A diferencia de los IDEs locales, Gitpod es un **CDE (Cloud IDE)**.
No se descarga un ejecutable pesado al PC.

- **Zero Config:** El entorno está listo en segundos.
- **Consistencia:** "Funciona en mi nube" significa que funciona para todo el equipo.
- **Potencia:** Delegas el procesamiento a servidores de alto rendimiento.



Cómo Comenzar



1. Registro

Acceso vía SSO con **GitHub**, **GitLab** o **Bitbucket**. Sincronización inmediata de repositorios.



2. Extensión

Extensión para Chrome/Firefox que añade el botón "**Open in Gitpod**" directamente en Git.



3. Lanzamiento

Prefijo `gitpod.io/#` en cualquier URL para instanciar el contenedor de desarrollo.

II. Configuración y Customización

Infraestructura como código para el desarrollador

Configuración como Código

El archivo `.gitpod.yml`

Todo el entorno se define en la raíz del proyecto. Permite automatizar:

- Instalación de extensiones de **Open VSX**.
- Tareas de inicio (npm install, compilación).
- Puertos de red y prebuilds.

Gitpod utiliza la interfaz de VS Code, permitiendo temas y atajos familiares.

Project.js — revel-api

```
... JS Project.js ×  
src > pages > JS Project.js > ...  
1 import React, { useState, useEffect } from "react";  
2 import { useParams } from "react-router";  
3 import Loading from "../components/common>Loading";  
4 import PageHeader from "../components/layout/PageHeader";  
5 import Card from "../components/project/Card";  
6 import RemindersCard from "../components/project/RemindersCard";  
7 import TeamCard from "../components/project/TeamCard";  
8 import LinksCard from "../components/project/LinksCard";  
9  
10 export default function Project() {  
11   const { id } = useParams();  
12   const [project, setProject] = useState();  
13   const [loading, setLoading] = useState(true);  
14  
15   useEffect(() => {  
16     fetch(`/api/project/${id}`)  
17       .then((response) => response.json())  
18       .then((data) => {  
19         setProject(data);  
20         setLoading(false);  
21       });  
22   }, [ ]);  
23  
24   return (  
25     <>  
26       <div className="h-full bg-gray-100">  
27         {loading} ? (  
28           <Loading />  
29         ) : (  
30           <>  
31             <PageHeader title={`#${project.channelName}`} />  
32             <div className="m-3 mb-0 flex flex-col lg:flex-row">  
33               <div className="w-full lg:w-1/2 p-3">  
34                 <Card title="Project Brief">{project.brief}</Card>  
35                 <RemindersCard reminders={project.Reminders} />  
36               </div>  
37               <div className="w-full lg:w-1/2 p-3">  
38                 <TeamCard members={project.Teams} />  
39                 <LinksCard links={project.Links} />  
40               </div>  
41             </div>  
42         </div>  
43       </div>  
44     </>  
45   );  
46 }  
47  
48 Ln 48, Col 1 Spaces: 2 UTF-8 LF JavaScript
```

IV. Funcionalidades Críticas

Build, Debugging y Control de Versiones

Herramientas de Ingeniería



Generación y Build

Entorno **Linux nativo** (Docker). Soporte para make, gradle, npm.

Gestión automática de dependencias mediante detección de pom.xml o package.json.



Depuración Avanzada

Breakpoints condicionales y puntos de ruptura en línea.

Control total: **Step Into, Over, Out**. Inspección de variables en tiempo real en el panel lateral.

Eficiencia: Tiempo de Configuración



Gitpod elimina el "drift" de configuración entre desarrolladores.

Modelo de Licencia y Coste

Plan	Horas / Uso	Precio (Est.)	Ideal para...
Free	50h / mes	0€	Estudiantes y Open Source
Personal	Uso extendido	~9€ / mes	Freelancers
Professional	Horas ilimitadas	~25€ / mes	Equipos y Empresas

Mantenimiento Efímero



Lanzamiento

Se crea un contenedor limpio desde la imagen Docker.



Actualización

El IDE (VS Code) se actualiza solo en el SaaS.



Desarrollo

Cambios se guardan en Git.
El estado es temporal.



Cierre

El workspace se destruye tras inactividad. Higiene total.

Preguntas de Evaluación

1. ¿Dónde se define la configuración del entorno para que sea idéntica para todo el equipo?
2. Si cierras un workspace sin hacer "Push", ¿qué sucede con tus cambios?

¡Gracias por vuestra atención!

Image Sources



<https://gitpod.gallerycdn.vsassets.io/extensions/gitpod/gitpod-desktop/0.1.2025061709/1750152947158/Microsoft.VisualStudio.Services.Icons.Default>

Source: [marketplace.visualstudio.com](https://marketplace.visualstudio.com/items?itemName=gitpod.gitpod)



https://static.vecteezy.com/system/resources/previews/038/048/999/non_2x/server-rack-icon-in-flat-color-circle-style-computer-data-file-center-hosting-cloud-transfer-vector.jpg

Source: [www.vecteezy.com](https://www.vecteezy.com/free-vector/server-rack)



<https://assets.browserlondon.com/wp-content/uploads/2020/10/Screenshot-2020-10-27-at-11.26.59.png>

Source: www.browserlondon.com