

# Guida Setup: VS Code + Gemini Code Assist (Google-Native Stack)

Questa guida sostituisce le indicazioni precedenti su Cursor. Configureremo VS Code per diventare una "stazione di controllo" completa per il Dental SaaS, integrando sviluppo AI e gestione Cloud.

## 1. Installazione Estensioni (Il "Core")

Non installare decine di plugin. Per questo stack servono solo questi essenziali:

1. **Google Cloud Code** (ID: googlecloudtools.cloudcode)
  - *Cos'è:* L'estensione ufficiale. Include **Gemini Code Assist**, Cloud Run emulators, e Kubernetes tools.
  - *Perché:* È il cervello che collega il tuo codice all'infrastruttura di Milano.
2. **ESLint & Prettier**
  - *Perché:* Per mantenere il codice Next.js/NestJS pulito automaticamente.
3. **Prisma** (ID: Prisma.prisma)
  - *Perché:* Evidenziazione sintassi per il file schema.prisma.
4. **Docker** (ID: ms-azuretools.vscode-docker)
  - *Perché:* Per gestire i container del database locale senza usare il terminale.

## 2. Configurazione Gemini Code Assist

Una volta installato "Google Cloud Code":

1. Clicca sull'icona **Cloud Code** nella barra di stato (in basso) o nella sidebar.
2. Seleziona "**Sign in to Google Cloud**".
  - *Importante:* Usa lo stesso account Google con cui hai creato il progetto GCP (quello dove abbiamo attivato le API Healthcare).
3. Seleziona il progetto GCP attivo (es. dental-saas-production).
4. Attiva **Gemini**: Clicca sull'icona (sparkle) nella barra laterale destra o usa Ctrl+Shift+I (o Cmd+Shift+I su Mac) per aprire la chat.

## 3. Workflow "Vibe Coding" con Gemini

Ecco come replicare la velocità di sviluppo di Cursor usando gli strumenti nativi Google:

### A. "Context Awareness" (Lavorare con i Documenti)

Gemini in VS Code eccelle quando gli dai contesto.

- **Scenario:** Devi creare il modulo per l'invio al Sistema Tessera Sanitaria.
- **Azione:**
  1. Apri la chat di Gemini.
  2. Non chiedere solo "fai il modulo".

3. Copia e incolla le specifiche tecniche del STS (o parti del PDF che hai caricato).
4. Prompt: "Basandoti su queste specifiche XML (incolla snippet), genera un servizio NestJS che serializza i dati della fattura. Usa fast-xml-parser."

## B. Refactoring Intelligente

- **Scenario:** Vuoi proteggere una route API.
- **Azione:**
  1. Seleziona il codice del Controller nel file.
  2. Clicca destro -> **Gemini Code Assist** -> **Edit** (o shortcut).
  3. Prompt: "Aggiungi il controllo RLS per verificare che il tenantId corrisponda all'utente loggato. Usa il TenantGuard."
  4. Gemini mostrerà un "Diff" (differenza) che puoi accettare con un click.

## C. Debugging dell'Infrastruttura

Questo è il vero vantaggio su Cursor.

- **Scenario:** Il deploy su Cloud Run fallisce.
- **Azione:** Invece di andare sulla console web di Google, apri il pannello **Cloud Run** dentro VS Code.
- Clicca sui log dell'errore.
- Tasto destro -> "**Explain with Gemini**".
- Lui analizzerà il log JSON di Google Cloud (che spesso è criptico) e ti dirà: "L'errore è dovuto alla mancanza del permesso storage.objects.create sulla Service Account. Aggiungilo nel file Terraform."

## 4. Prompt Strategici per il Progetto

Copia questi prompt nella chat di Gemini per velocizzare le fasi critiche:

### Per il Frontend (Shadcn/UI):

"Genera un componente React usando Shadcn UI e Tailwind. Deve essere una 'PatientCard' che mostra: Nome, Avatar (usa un placeholder), Ultima Visita, e un badge per lo stato 'Pagamenti'. Usa icone Lucide-React. Il design deve essere minimale e pulito."

### Per il Database (Prisma):

"Ho bisogno di aggiungere la gestione dei Lotti MDR. Modifica il mio modello Prisma (incolla modello esistente) aggiungendo l'entità 'MaterialBatch' collegata ai 'LabOrder'. Includi campi per data scadenza e codice UDI."

### Per i Test (Jest/Vitest):

"Scrivi un test unitario per questo servizio (seleziona codice). Verifica in particolare il caso limite in cui il codiceFiscale è invalido o l'importo è zero."

## 5. Pro Tip: .cursorrules -> SYSTEM PROMPT

Anche se non usi Cursor, puoi "istruire" Gemini.

Crea un file .md nella root (es. PROJECT\_CONTEXT.md) con le regole del progetto:

- "Usiamo Next.js 14 App Router."

- "Usiamo Tailwind per lo stile."
- "Ogni tabella DB deve avere tenantId."

Quando inizi una nuova task complessa, inizia la chat dicendo: "*Leggi il contesto seguente:*" e incolla il contenuto di quel file. Questo allinea l'AI ai tuoi standard.