Articoli e guide tecniche

Questo progetto raccoglie degli articoli in cui ho provato a mettere insieme alcune delle metodologie usate nello sviluppo del codice presente nell'organizzazione GitHub https://github.com/fugerit-org

Scarica in formato PDF

Indice dei contenuti

Guida all'Integrazione dei Modelli (Gemini, Claude, ChatGPT) in Antigravity IDE 1
1. Gestione delle Quote e dei Limiti 1
2. Flusso di Lavoro Ibrido (Hybrid Workflow) 2
3. Regole d'Oro per il Risparmio di Quota (Claude & ChatGPT) 3
4. Tecnica del "Prompt Handoff" (Meta-Prompting) 3
5. Risorse Utili e Riferimenti 6

Guida all'Integrazione dei Modelli (Gemini, Claude, ChatGPT) in Antigravity IDE

Questo documento illustra come combinare strategicamente l'uso di diversi modelli linguistici (Gemini, Anthropic Claude, OpenAI ChatGPT) all'interno dell'ambiente di lavoro con l'assistente agentico **Antigravity**, ottimizzando le quote di utilizzo e le finestre di contesto.

1. Gestione delle Quote e dei Limiti

A seconda degli abbonamenti attivi, la disponibilità e i limiti dei modelli variano notevolmente.

Modelli Gemini (Google)

- **Stato:** Coperti dall'abbonamento **Google AI Premium ($19.99/mese)**.
- **Quota:** Molto elevata / illimitata per l'uso quotidiano ordinario, con finestre di contesto eccezionalmente ampie (fino a 1M+ token nei modelli Gemini 1.5/2.0).
- **Utilizzo consigliato:** Modelli primari ("workhorse") per la maggior parte dei task: analisi di codebase, scansione del workspace, generazione di documentazione e test.

Modelli Anthropic (Claude Sonnet / Opus) e OpenAI (ChatGPT / GPT-4o)

- **Stato:** Gestiti tramite piano gratuito (free tier) dell'IDE oppure tramite chiavi API dedicate.
- **Quota: Molto limitata.** Nel piano gratuito / condiviso dell'IDE, sono soggetti a rigidi limiti di messaggi o token che si azzerano periodicamente (es. ogni 3–5 ore).

- **Utilizzo consigliato:** Modelli specialistici per compiti di precisione (logica algoritmica, refactoring, code review approfondita); da utilizzare in modo mirato per non esaurire rapidamente la quota.

La finestra di contesto di Claude Sonnet e GPT-4o è ampia (100k–200k token), ma **ogni token inviato nel contesto pesa sulla quota del piano gratuito**. Ridurre il contesto inutile è quindi la strategia più efficace per prolungare la disponibilità di questi modelli.

2. Flusso di Lavoro Ibrido (Hybrid Workflow)

Per massimizzare l'efficacia risparmiando le quote dei modelli più restrittivi (Claude / ChatGPT), si consiglia di suddividere il lavoro nelle seguenti fasi:

Fase di Lavoro	Modello Consigliato	Strategia e Motivazione
1. Analisi, Ricerca e Setup	Gemini Flash / Gemini 2.0	Lettura iniziale del codice, scansione del workspace ed elaborazione del piano d'azione. Sfrutta l'ampia finestra di contesto di Gemini a costo zero sulla quota Anthropic / OpenAI.
2. Implementazione della Logica Core	Claude Sonnet / GPT-4o	Scrittura del codice algoritmico principale o refactoring di classi complesse. Eccellente capacità di generazione sintattica corretta e ragionamento logico strutturato. Da invocare su **contesto minimo** preparato nella fase precedente (vedi Prompt Handoff).
3. Debugging Critico e Analisi Architetturale	Claude Opus / Gemini (Thinking)	Risoluzione di bug ostici o analisi di problemi di concurrency / architettura che richiedono ragionamento profondo (multi-step reasoning). Da usare con parsimonia per la quota elevata.
4. Scrittura Test e Documentazione	Gemini Flash	Generazione di unit test ripetitivi, JavaDoc, file README e documentazione AsciiDoc. Operazione rapida a consumo minimo della quota.
5. Code Review Finale e Validazione	Claude Sonnet / GPT-4o	Revisione critica del codice prodotto nelle fasi precedenti, verifica di edge case e controllo di sicurezza (OWASP, best practice). Contesto limitato al solo diff o alla classe modificata.

Warm-up contestuale: Quando si avvia Antigravity su un progetto nuovo, destinare sempre la prima sessione a Gemini Flash per indicizzare il workspace. Questo permette di estrarre in seguito snapshot di contesto precisi da passare a Claude / ChatGPT senza far scansionare l'intero progetto a questi ultimi.

3. Regole d’Oro per il Risparmio di Quota (Claude & ChatGPT)

Quando utilizzi i modelli Anthropic o OpenAI all’interno di Antigravity, adotta questi accorgimenti per non esaurire i messaggi disponibili:

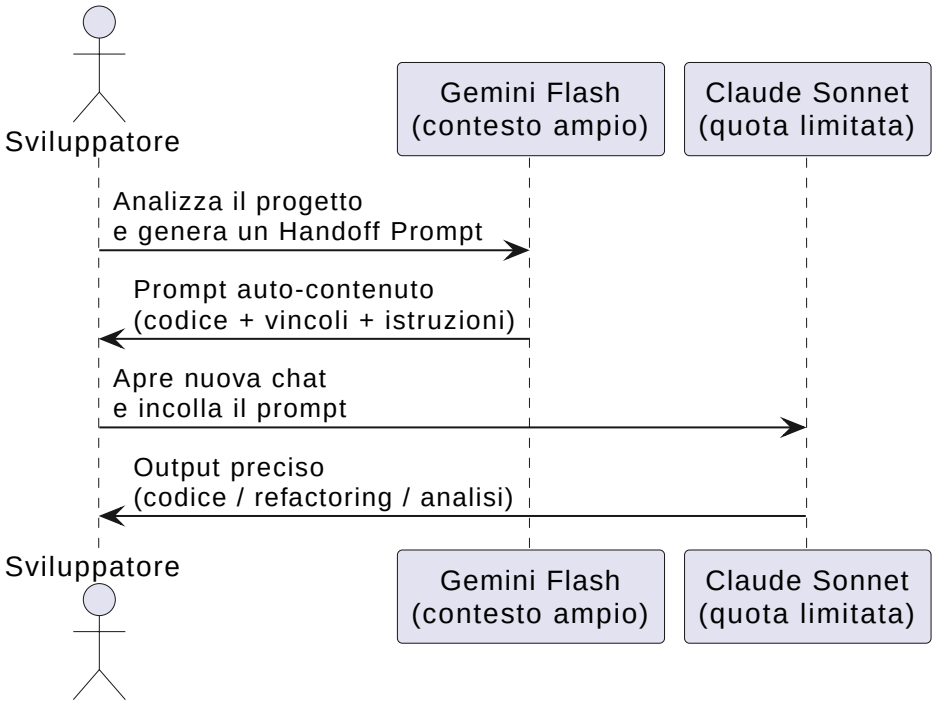
1. **Evita la cronologia inutile:** Quando passi da Gemini (usato per l’analisi preliminare) a Claude / ChatGPT per la scrittura del codice, apri una **nuova chat** o effettua un **Reset Context**. Se mantieni la stessa chat, l’intera cronologia dei messaggi precedenti verrà inviata al nuovo modello, consumando istantaneamente una gran quantità di token della quota limitata.
2. **Fornisci solo il contesto necessario:** Invece di far scansionare l’intero progetto a Claude, incolla nel prompt solo la porzione di codice o la firma del metodo su cui deve lavorare.
3. **Pianifica con Gemini, esegui con Claude:** Chiedi a Gemini Flash di generare un piano d’azione dettagliato, poi passa a Claude o GPT per implementare i singoli step del piano, uno alla volta.
4. **Tecnica del "Prompt Handoff" (Meta-Prompting):** Vedi la sezione dedicata al capitolo 4. Tecnica del "Prompt Handoff" (Meta-Prompting) per la descrizione completa e i template riutilizzabili.
5. **Usa modelli più leggeri per iterazioni intermedie:** Durante il ciclo di sviluppo, preferisci Claude Haiku o GPT-3.5 per le iterazioni rapide (es. piccoli fix, rinominare variabili, aggiungere log); riserva Sonnet / Opus solo per le trasformazioni significative.

4. Tecnica del "Prompt Handoff" (Meta-Prompting)

Il **Prompt Handoff** è una delle tecniche più potenti per l’ottimizzazione delle quote in un workflow ibrido multi-modello. Sfrutta la grande finestra di contesto di Gemini per generare istruzioni precise e auto-contenute da consegnare a Claude o ChatGPT, che così non hanno bisogno di accedere ad altri file del progetto.

4.1 Principio di Funzionamento

Il flusso si articola in tre passaggi:



Il diagramma PlantUML richiede un processore compatibile (es. `asciidoctor-diagram`). Se non disponibile, può essere rimosso senza perdita di contenuto.

In assenza di un renderer PlantUML, il flusso può essere riassunto così:

```
[Sviluppatore] → Gemini Flash (analisi progetto)
                        ↓
              genera Handoff Prompt
                        ↓
[Sviluppatore] → Copia prompt → Nuova chat Claude Sonnet
                                        ↓
                                Output preciso, a contesto minimo
```

4.2 Variante Base: Context Snapshot

È la forma più semplice del Prompt Handoff. Al termine di un'analisi svolta con Gemini, chiedi esplicitamente:

> Generami un prompt autonomo e dettagliato da dare in input a Claude Sonnet per eseguire [descrivi l'azione, es. "il refactoring della classe `OrderService`"].
>
> Il prompt deve includere:
>
> 1. Il codice sorgente rilevante estratto direttamente dal file (non fare riferimento a percorsi).
> 2. I vincoli tecnici da rispettare (es. compatibilità Java 17, framework

> Quarkus 3.x, stile immutabile per i DTO).
> 3. Le istruzioni passo-passo necessarie all'esecuzione del task.
> 4. Il formato atteso dell'output (es. "riscrivi solo il metodo X", "produce solo il diff").
>
> Il prompt deve essere auto-contenuto: Claude Sonnet non deve aver bisogno di aprire altri file del progetto.
>
> — Prompt da inviare a Gemini Flash

Dopo aver ricevuto il prompt da Gemini, apri una **nuova chat pulita** con Claude Sonnet, incolla il prompt e ottieni il massimo della precisione con il minimo consumo di token.

4.3 Variante Avanzata: Chain Handoff (Gemini → Claude → ChatGPT)

In task complessi che richiedono competenze specialistiche di più modelli, è possibile concatenare più handoff in sequenza:

Step	Modello	Ruolo nel Chain
1	Gemini Flash	Analisi del codebase e generazione del piano d'azione strutturato. Output: documento Markdown con sezioni per ogni sub-task.
2	Claude Sonnet	Implementazione del sub-task più critico (logica algoritmica, sicurezza, refactoring). Input: prompt auto-contenuto generato da Gemini al passo 1. Output: codice prodotto + breve spiegazione delle scelte.
3	GPT-4o	Validazione e code review del codice prodotto al passo 2, con focus su edge case e compatibilità API. Input: solo il codice prodotto (senza la storia dei passi precedenti).

In ogni transizione del Chain Handoff, aprire **sempre una nuova chat** con il modello successivo. Non trascinare la cronologia tra step: il contesto accumulato annullerebbe il vantaggio di risparmio della quota.

4.4 Template Riutilizzabile per il Prompt Handoff

Il seguente template può essere adattato rapidamente per qualunque task:

```
## Contesto del Task
**Progetto:** [nome progetto]
**Framework / Tecnologie:** [es. Quarkus 3.x, Java 17, Oracle 19c]
**Obiettivo:** [descrizione sintetica dell'azione da compiere]
```

```
## Codice Sorgente Rilevante
```java
// Incolla qui il codice estratto da Gemini
```

Vincoli e Requisiti
- [Vincolo 1: es. "non modificare la firma dei metodi pubblici"]
- [Vincolo 2: es. "mantenere la compatibilità con Java 11 se usi stream"]
- [Vincolo 3: es. "aggiungere Javadoc su tutti i metodi modificati"]

Istruzioni Passo-Passo
1. [Passo 1]
2. [Passo 2]
3. [Passo N]

Formato Atteso dell'Output
[es. "Riscrivi solo il corpo del metodo X, senza modificare il resto della classe."]
[es. "Produce un diff unificato (formato `git diff`) applicabile direttamente."]
```

4.5 Quando NON Usare il Prompt Handoff

Il Prompt Handoff non è sempre la scelta ottimale. Evitarlo quando:

- Il task è banale e risolvibile direttamente con Gemini (es. generare un README).
- Il codice da includere nel prompt supera i 50.000 token: in tal caso conviene mantenere Gemini come unico modello per non rischiare di saturare anche la finestra di Claude.
- Il contesto da trasmettere è fortemente dipendente da file di configurazione multipli e frammentati: in quel caso Gemini può analizzare meglio l'intero workspace in autonomia.

5. Risorse Utili e Riferimenti

Strumenti e Piattaforme

- **Google Gemini:** Interfaccia Web Gemini | Google AI Studio (chiavi API)
- **Anthropic Claude:** Interfaccia Web Claude | Anthropic Console
- **OpenAI ChatGPT:** Interfaccia Web ChatGPT | OpenAI Developer Platform

Metodologie e Concetti Chiave

- **Meta-Prompting (Prompt Handoff):** Tecnica in cui un LLM genera le istruzioni ottimali per un altro LLM, isolando e sintetizzando il contesto necessario al task.
- **Context Snapshot:** Variante base del Prompt Handoff — estrazione di un frammento di contesto auto-contenuto da passare a un modello con quota limitata.
- **Chain Handoff:** Variante avanzata — concatenazione di più modelli specializzati in sequenza, con nuova chat ad ogni transizione.

- **In-Context Learning:** Ottimizzazione del contesto fornito al modello per ridurre al minimo allucinazioni e consumo inutile di token.
- **Hybrid AI Development Workflow:** Approccio di sviluppo software che sfrutta in sequenza i punti di forza dei diversi provider (Google, Anthropic, OpenAI) in base alla fase del ciclo di vita del software (SDLC).