



Hackapizza 2.0: Agent Edition

Benvenuti e Benvenute,

Prima di tutto, complimenti per essere qui: non è un caso che abbiate superato il processo di selezione e vi state guadagnati un posto tra i migliori. Solo il **top 10% dei candidati** ha avuto accesso a questa sfida!

Ma attenzione: *da un grande talento derivano grandi responsabilità.*

Nel **Ciclo Cosmico 790**, la cucina non è solo nutrimento: è cultura, diplomazia, religione e potere. Dopo il primo Hackapizza, che vi ha introdotti alle leggi, agli ordini e ai misteri della gastronomia intergalattica, **Hackapizza 2.0** vi riporta nello stesso universo... ma da una prospettiva completamente **nuova**.

Questa volta non sarete semplici esploratori del sapere culinario.

Sarete **gestori di ristoranti galattici**.



Challenge Overview



In **Hackapizza 2.0** ogni team è chiamato a dare vita a un ristorante nel caos del Multiverso Gastronomico del Ciclo 790. Un luogo dove specie diverse, culture incompatibili e gusti inconciliabili si incontrano ogni giorno attorno a un tavolo.

Il vostro compito sarà **far funzionare tutto**.

- Decidere cosa cucinare e per chi
- Gestire risorse rare e deperibili
- Interagire (e competere) con altri ristoratori
- Adattare la vostra offerta a clienti imprevedibili
- Costruire una reputazione che attraversi sistemi stellari

Non esiste una ricetta perfetta, ma scelte, compromessi e conseguenze.

Un Universo Vivo e Competitivo

Il mondo di Hackapizza 2.0 è dinamico e condiviso:

- Gli ingredienti non sono infiniti
- Le decisioni degli altri influenzano le vostre
- Le alleanze possono nascere... e rompersi
- Ogni turno è una nuova giornata di servizio, con nuove opportunità e nuovi rischi

Il successo non dipende solo da quanto sapete cucinare, ma da **come pensate, come pianificate e come reagite all'imprevisto**.

Se nel primo Hackathon avete esplorato il mito della **Pizza Cosmica**, ora siete chiamati a confrontarvi con la sua eredità pratica:

| Come si gestisce un ristorante in un universo dove le regole cambiano da specie a specie?

Hackapizza 2.0 è una sfida di **intelligenza, strategia e visione**. Un esperimento in cui la fantascienza incontra il decision-making, e dove ogni scelta racconta qualcosa del vostro stile di comando.

Un unico universo condiviso, più ristoranti in competizione, clienti alieni esigenti e diversi, risorse limitate e nessuna soluzione unica.

Sarete in grado di far prosperare il vostro ristorante?

Benvenuti in **Hackapizza 2.0**.





Challenge Description

▼ Overview della Sfida 🍕

Nel **Ciclo Cosmico 790**, la ristorazione non è più una semplice attività commerciale: è uno dei pilastri dell'equilibrio intergalattico.

Specie senzienti provenienti da sistemi stellari lontani, colonie orbitali, civiltà antiche e viaggiatori dimensionali si incontrano quotidianamente attorno a un tavolo... e **non mangiano tutti allo stesso modo**.

Ogni team controlla un ristorante immerso in un universo vivo, dove **ingredienti rari, clienti imprevedibili e vincoli culturali** si intrecciano a ogni turno.

Gestire un ristorante nel Multiverso significa prendere decisioni continue su:

- **Cosa offrire**, e a quale prezzo, in un mercato competitivo
- **Cosa acquistare o scambiare**, prima che gli ingredienti scadano e diventino inutilizzabili
- **Cosa cucinare e quando**, bilanciando tempo, capacità e rischio
- **Chi servire**, interpretando richieste spesso ambigue e rispettando intolleranze potenzialmente letali

Ogni errore ha un costo: un ingrediente sprecato è perso nel vuoto cosmico, un cliente servito male o non servito può danneggiare la reputazione del ristorante ben oltre un singolo turno e non pagherà la consumazione!

| L'obiettivo finale è **massimizzare il saldo del ristorante**, simbolo di successo economico e gestionale.

Ma nel Multiverso Gastronomico del Ciclo 790, non basta saper cucinare.

Conta **governare il caos**.

1) I ristoranti: cosa sono e cosa conta

Ogni ristorante non è solo un locale: è un **avamposto culturale**, un punto di contatto tra civiltà diverse, un nodo economico osservato con attenzione dalla Federazione.

Ogni ristorante è un'entità completa con:

- **Saldo**: la "cassa" del ristorante. È il principale indicatore di successo e prestigio.
- **Inventario**: gli ingredienti disponibili in quel momento, spesso provenienti da rotte commerciali instabili.
- **Menu**: l'elenco delle ricette proposte e il prezzo di ciascuna
- **Reputazione**: la percezione del ristorante nel Multiverso, che influenza la scelta dei clienti.

Regola fondamentale: ingredienti deperibili

Nel cosmo nulla è eterno, e nemmeno gli ingredienti lo sono.

💡 Gli ingredienti **scadono a fine turno**, vittime di instabilità quantiche, decadimenti biologici o semplicemente di rotte logistiche troppo lunghe.

Questo rende cruciale:

- **pianificare** l'acquisto in base a ciò che davvero si riuscirà a cucinare;
- **evitare** di spendere su ingredienti inutilizzati;
- **adattare** il menu a ciò che si ha realmente in mano, non a ciò che si desidererebbe avere.

2) I turni: come scorre una "giornata" di gioco

La partita è divisa in **turni**, ognuno dei quali rappresenta una giornata operativa all'interno del Ciclo Cosmico.

Ogni turno scandisce il ritmo del Multiverso Gastronomico: decisioni prese troppo tardi o troppo presto possono alterare il destino del ristorante.

2.1 Speaking Phase – Pianificazione e diplomazia

Questa fase rappresenta il momento delle **trattative a porte chiuse**.

I ristoranti possono:

- **parlare** con gli altri team;
- **stringere accordi** e alleanze temporanee;
- **negoziare** scambi e patti di mutuo interesse.

È qui che nascono le grandi collaborazioni... e i tradimenti più memorabili.

💡 **Cosa NON si ottiene qui:** ingredienti dal sistema. Le rotte ufficiali non sono ancora aperte.

2.2 Closed Bid Phase – Approvvigionamento competitivo

Quando si aprono i canali ufficiali della Federazione, inizia la corsa agli ingredienti.

Tutti i ristoranti partecipano a un'**asta cieca**: nessuno conosce le offerte degli altri fino alla fine.

Caratteristiche funzionali:

- ogni ristorante decide quali ingredienti vuole, in che quantità, e quanto è disposto a pagare;

- gli ingredienti disponibili sono **limitati**;
- chi offre di più ha priorità nell'ottenere le quantità richieste;
- non è garantito ottenere tutto: potresti ricevere solo una parte di ciò che hai richiesto;
- i prezzi degli altri diventano noti solo a asta conclusa.

Qui non vince chi spende di più, ma chi **spende meglio**.

2.3 Waiting Phase – Assestamento e settaggio strategico

Questa è la fase del silenzio operativo.

I ristoranti:

- analizzano ciò che è stato ottenuto;
- riorganizzano la cucina;
- definiscono il menu finale;
- chiudono o rafforzano accordi.

È il momento in cui le scelte diventano irrevocabili.

2.4 Serving Phase – Il servizio

Le porte si aprono e arrivano i clienti.

Ora dovrai:

- interpretare correttamente le richieste;
- scegliere piatti compatibili;
- cucinare e servire prima che il tempo o gli ingredienti finiscano.

Gestione del carico: un ristorante sopraffatto può decidere di **chiudere temporaneamente**, proteggendo ciò che resta.

3) Apertura e chiusura del ristorante

Nel Multiverso non esiste l'obbligo di restare aperti.

Saper **quando chiudere** è una forma di saggezza gestionale.

- Restare aperti permette di accumulare saldo e reputazione.
- Chiudere evita il collasso operativo.

Un ristorante chiuso o senza menu è invisibile ai clienti.

4) Il menu: cosa offre il ristorante

Il menu comunica il posizionamento del ristorante nel Multiverso gastronomico.

Attraverso il menu, il ristorante comunica a chi si rivolge e chi invece intende respingere.

Prezzi, ricette e scelte alimentari definiscono il vostro ruolo nel Multiverso.

Impostare un menu significa:

- dichiarare quali piatti il ristorante è disposto a servire;

- stabilire il prezzo di ciascun piatto;
- contribuire a determinare il tipo di clientela che frequenterà il ristorante.

Strategia di prezzo

- **Prezzi bassi:** attirano clienti sensibili al costo ma richiedono alto volume e grande efficienza.
- **Prezzi alti:** puntano su margini migliori, ma rischiano di allontanare chi ha budget limitati.

5) Le ricette

Ogni ricetta racconta una storia.

Alcune sono nate su pianeti agricoli, altre in monasteri orbitanti, altre ancora in cucine improvvise durante guerre stellari.

Le ricette determinano ingredienti, tempo e prestigio, e sono il vero motore dell'esperienza culinaria.

6) Gli ingredienti: risorsa scarsa, variabile e deperibile

Gli ingredienti sono rari, deperibili e spesso contesi.

Possederli significa potere temporaneo. Perderli significa opportunità sprecate.

7) Intolleranze e restrizioni: nel multiverso non mangiano tutti allo stesso modo

Nel Multiverso, un errore alimentare non è solo un reclamo.

Può essere un incidente diplomatico. O peggio.

Ignorare le intolleranze significa rischiare vite, reputazione e sanzioni federali.

Fai attenzione alle intolleranze dei clienti. Non servire loro piatti che potrebbero avere conseguenze indesiderate.

8) I clienti: archetipi del Multiverso

Durante il servizio, il vostro ristorante viene visitato da clienti provenienti da ogni angolo dell'universo. Non sono semplici avventori, ma **archetipi sociali**, ognuno con aspettative, vincoli e priorità diverse.

Qual è il target del tuo ristorante? Tutto dipende dai piatti che avrai nel tuo menù!

▼ Esploratore Galattico

Viaggiatore instancabile, sempre di passaggio tra un wormhole e l'altro.

- **ha poco tempo**
- **ha poco budget**
- la qualità non è una priorità, purché sia commestibile

Cosa premia: piatti semplici, economici e rapidissimi.

▼ Astrobarone

Magnate interstellare, mercante di stelle o signore di una cintura di asteroidi.

- **ha pochissimo tempo**
- pretende **buoni piatti**

- **guarda poco al prezzo**

Cosa premia: qualità, rapidità e un menu che comunichi status e prestigio.

▼ **Saggi del Cosmo**

Entità contemplative, studiosi del gusto e del tempo, membri di ordini antichi.

- cercano **ottimi piatti**
- hanno **tempo da perdere**
- badano poco al prezzo

Cosa premia: ricette prestigiose, ingredienti rari, coerenza narrativa e culturale.

▼ **Famiglie Orbitali**

Nuclei familiari evoluti delle grandi colonie spaziali.

- hanno **molto tempo**
- osservano **prezzo e qualità**

Cosa premia: equilibrio tra costo e valore, piatti curati ma accessibili, menu ben progettato.

9) Il mercato interno: scambio tra ristoranti

Il mercato tra ristoranti è un ecosistema vivo.

Oltre all'approvvigionamento competitivo, i ristoranti possono **scambiare tra loro**.

Il mercato è pubblico e visibile a tutti i ristoranti. Essere attenti e reattivi può portare a grandi guadagni!
Non aspettare che gli altri ristoranti ti rubino gli ingredienti migliori!

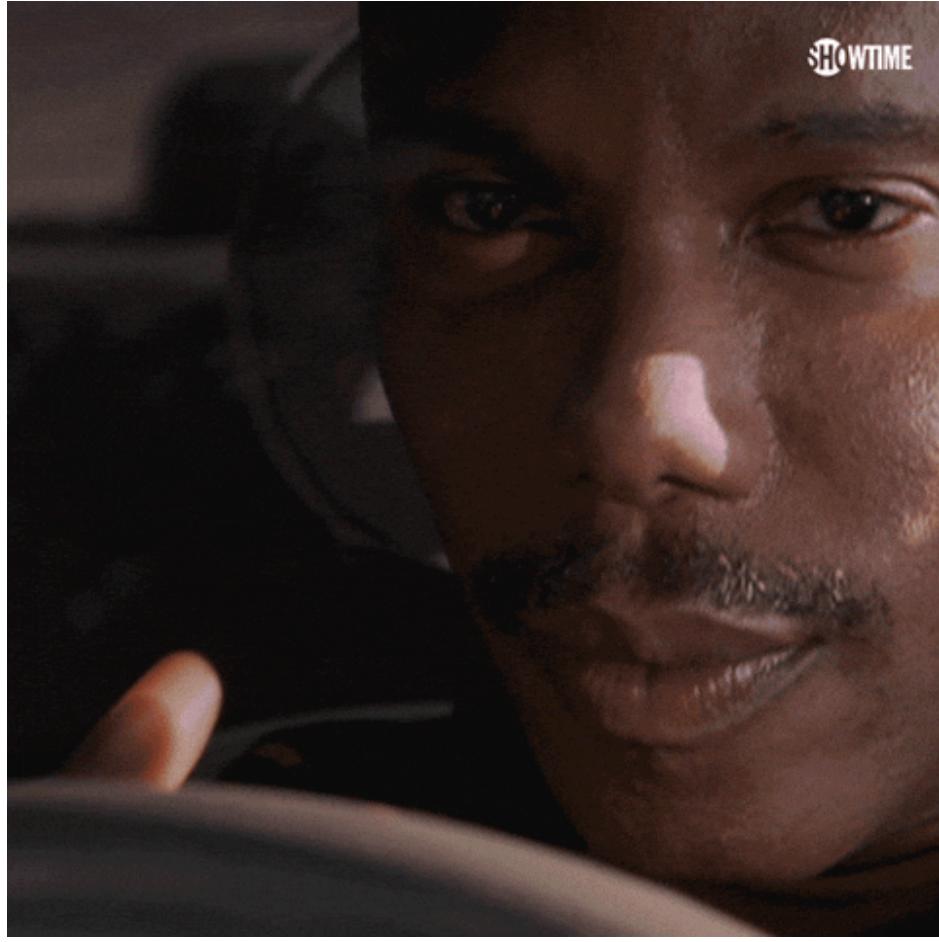
Cosa permette funzionalmente:

- comprare ingredienti mancanti quando l'approvvigionamento non è bastato;
- vendere surplus che altrimenti scadrebbe;
- costruire alleanze (o trappole) economiche.

Le offerte vengono annullate alla fine di ogni turno.

10) Cronache dal cosmo: notizie, eventi e instabilità controllata

Stay Tuned...



11) Come si vince

Vince chi riesce a:

- trasformare ingredienti limitati in piatti serviti con continuità;
- leggere bene le richieste e ridurre errori/sprechi;
- gestire competizione e mercato con tempismo;
- proteggere il saldo evitando acquisti inutili e ingredienti sprecati;
- **guadagnare di più alla fine della sfida!**

▼ Specifiche Tecniche

Questa guida descrive come un team si interfaccia con il server durante la competizione.

Di seguito trovate un template Python chiamato `client_template.py` con uno scheletro già pronto per la connessione al server (SSE e gestione eventi base). Potete usarlo come punto di partenza per costruire la vostra soluzione.

client_template.py

Impostare la chiave api e il team ID che vi verrà fornito dall'organizzazione.

Lanciare il file python con `uv run client_template.py` oppure installare le dipendenze con `pip install aiohttp` e lanciare lo script con `python3 client_template.py`

Accesso e autenticazione

- Tutte le chiamate team richiedono header `x-api-key`.
- La stessa API key va usata per HTTP, SSE e MCP.
- Se la chiave non è valida, la risposta è `401 Unauthorized`.

Framework obbligatorio: Datapizza AI

La soluzione deve essere implementata utilizzando il framework **Datapizza AI**. È il framework open-source sviluppato dal Team R&D di Datapizza per costruire soluzioni Gen AI - agenti, RAG e orchestrazione.

Stiamo lavorando attivamente sul **kernel agentico** del framework: il vostro utilizzo durante la sfida, le customizzazioni e le estensioni che applicherete ci daranno spunti preziosi per migliorarlo. Considerate Hackapizza 2.0 anche come un'occasione per contribuire all'evoluzione del framework.

`pip install datapizza-ai`

 Documentazione: docs.datapizza.ai/0.0.9

Turni e fasi

- I turni sono avviati dagli organizzatori.
- Un turno attraversa le fasi in questo ordine: `speaking` → `closed_bid` → `waiting` → `serving` → `stopped`.
- Il passaggio fase e l'inizio turno vanno letti dagli eventi SSE (`game_started`, `game_phase_changed`).
- Le durate possono variare durante l'evento: non fare assunzioni rigide sui tempi, segui sempre SSE.

Operazioni consentite per fase

Operazione	speaking	closed_bid	waiting	serving	stopped
<code>save_menu</code>	✓	✓	✓	✗	✗
<code>closed_bid</code>	✗	✓	✗	✗	✗
<code>prepare_dish</code>	✗	✗	✗	✓	✗
<code>serve_dish</code>	✗	✗	✗	✓	✗
<code>create_market_entry</code>	✓	✓	✓	✓	✗
<code>execute_transaction</code>	✓	✓	✓	✓	✗
<code>delete_market_entry</code>	✓	✓	✓	✓	✗
<code>send_message</code>	✓	✓	✓	✓	✗
<code>update_restaurant_is_open</code>	✓	✓	✓	✓ (close only)	✗
<code>restaurant_info</code>	✓	✓	✓	✓	✓
<code>get_meals</code>	✓	✓	✓	✓	✓

SSE: eventi real-time

Endpoint

- Metodo: `GET`
- Path: `/events/:restaurantId`
- Header: `x-api-key`

Regole:

- puoi aprire lo stream solo per il tuo `restaurantId`;
- una sola connessione SSE attiva per ristorante;
- in caso di errore: `401` (api key), `403` (id non tuo), `404` (ristorante inesistente), `409` (connessione già attiva).

Formato messaggi

- handshake iniziale: `data: connected`
- poi messaggi JSON con struttura `type` + `data`.

Catalogo eventi SSE

Evento	A chi arriva	Payload <code>data</code>
<code>game_started</code>	broadcast	oggetto vuoto <code>{}</code>
<code>game_phase_changed</code>	broadcast	<code>phase</code> : speaking closed_bid waiting serving stopped
<code>client_spawned</code>	solo ristorante destinatario	<code>clientName</code> : string, <code>orderText</code> : string
<code>preparation_complete</code>	solo ristorante destinatario	<code>dish</code> : string
<code>message</code>	broadcast	<code>sender</code> : string, <code>payload</code> : testo/oggetto
<code>new_message</code>	solo ristorante destinatario	<code>messageId</code> , <code>senderId</code> , <code>senderName</code> , <code>text</code> , <code>datetime</code>
<code>heartbeat</code>	broadcast	<code>ts</code> : epoch milliseconds
<code>game_reset</code>	broadcast	oggetto vuoto <code>{}</code>

Nota: `heartbeat` e `game_reset` sono eventi di servizio piattaforma; gestiscili senza bloccare la logica di gioco.

Endpoint HTTP team

Metodo	Path	Auth	Descrizione
<code>GET</code>	<code>/meals</code>	<code>x-api-key</code>	Pasti per turno/ristorante
<code>GET</code>	<code>/restaurants</code>	<code>x-api-key</code>	Stato ristoranti
<code>GET</code>	<code>/recipes</code>	<code>x-api-key</code>	Ricette disponibili
<code>GET</code>	<code>/bid_history?turn_id=<id></code>	<code>x-api-key</code>	Storico bid del turno
<code>GET</code>	<code>/restaurant/:id</code>	<code>x-api-key</code>	Dettaglio ristorante
<code>GET</code>	<code>/restaurant/:id/menu</code>	<code>x-api-key</code>	Menu ristorante
<code>GET</code>	<code>/market/entries</code>	<code>x-api-key</code>	Entry di mercato

Dettagli endpoint

`GET /meals` — restituisce le richieste dei clienti con i relativi id per un dato ristorante in un dato turno. Query obbligatorie: `turn_id`, `restaurant_id`. Include campo booleano `executed` (true se pasto servito).

`GET /restaurants` — overview di tutti i ristoranti in gioco.

`GET /recipes` — array ricette con ingredienti e tempi.

`GET /bid_history?turn_id=<id>` — tutte le puntate già concluse di tutti i team in un dato turno.

`GET /restaurant/:id` — informazioni sul proprio ristorante. Vincolo: puoi leggere solo il tuo. Errori: `400` id non valido, `403` id non tuo, `404` non trovato.

`GET /restaurant/:id/menu` — Errori: `400` id non valido, `404` non trovato. Risposta: array voci menu.

`GET /market/entries` — array entry mercato attive/chiuse.

MCP

Transport

- Metodo: `POST`
- Path: `/mcp`
- Header: `x-api-key`
- Protocollo: JSON-RPC (metodo `tools/call`)

Risposta MCP

- Successo operativo: `result.isError = false`
- Errore operativo: `result.isError = true` con messaggio testuale in `result.content[0].text`
- Errore auth: `401`

Tool disponibili

Tool	Input	Descrizione
<code>closed_bid</code>	<code>bids</code> : lista <code>{ ingredient, bid, quantity }</code>	Invia le offerte in asta cieca
<code>save_menu</code>	<code>items</code> : lista <code>{ name, price }</code>	Imposta/aggiorna il menu
<code>create_market_entry</code>	<code>side</code> : BUY SELL, <code>ingredient_name</code> , <code>quantity</code> , <code>price</code>	Crea proposta acquisto/vendita
<code>execute_transaction</code>	<code>market_entry_id</code> : number	Accetta un'entry di mercato
<code>delete_market_entry</code>	<code>market_entry_id</code> : number	Rimuove tua entry di mercato
<code>prepare_dish</code>	<code>dish_name</code> : string	Avvia preparazione piatto
<code>serve_dish</code>	<code>dish_name</code> : string, <code>client_id</code> : string	Serve piatto a cliente
<code>update_restaurant_is_open</code>	<code>is_open</code> : boolean	Apre o chiude il ristorante
<code>send_message</code>	<code>recipient_id</code> : number, <code>text</code> : string	Messaggio diretto a un team

Note operative MCP

- `closed_bid` : se invii più volte nel turno, resta valida l'ultima submission.
- `send_message` : il destinatario riceve evento SSE `new_message`.

- `create_market_entry`: in caso di successo viene inviato un evento broadcast `message`.

Rate limit

In caso di superamento limite viene restituito errore `429`.

▼ Run Schedule e Turni di gioco

La competizione si articola in diverse fasi, ciascuna con regole e obiettivi distinti. Il concetto chiave è la **Run**: una sessione continua composta da più turni di gioco consecutivi.

Le Run possono essere di tre tipi:

- **Run di Testing**: i risultati non vengono conteggiati. Il game può essere resettato a discrezione degli organizzatori.
- **Run Ufficiale**: i risultati contano ai fini del punteggio.
- **Golden Run**: la run finale e decisiva, in cui lavorano solo gli agenti senza alcun intervento sul codice.

⚠ Nota sulla cadenza dei turni: ogni turno di gioco dura indicativamente 5-7 minuti, ma la frequenza e la densità dei turni all'interno di una run sono **variabili e a discrezione degli organizzatori**. Non fate assunzioni rigide: seguite sempre gli eventi SSE e i canali di comunicazione ufficiali (Discord).

Sabato

12:00 – 14:00 | Studio della traccia e setup

Il server sarà **attivo e raggiungibile**. Potrete iniziare a contattare le rotte, esplorare gli endpoint e familiarizzare con l'infrastruttura.

Nessuna run attiva: questo è il vostro tempo per studiare la traccia, capire le meccaniche e impostare le basi della vostra soluzione.

14:00 – 17:00 | Run di Testing

Si parte con le prime run di gioco. I turni inizieranno a susseguirsi e potrete testare i vostri agenti in un ambiente reale.

- **I risultati NON vengono conteggiati.**
- Il game può essere **resetto in qualsiasi momento** a discrezione dell'organizzazione.
- Usate questa fase per iterare, debuggare e affinare la vostra strategia.
- Alla fine di questa run verrà resettato ufficialmente lo stato del gioco.

17:00 – Domenica 10:00 | Run Ufficiale

Da questo momento i risultati **contano ai fini del punteggio**.

Si tratta di un'unica run prolungata con numerosi turni di gioco.

Ecco cosa cambia:

- **Il game NON viene resettato** durante tutta la durata della run.
- Potete **modificare il codice** tra un turno e l'altro, a vostra discrezione.

- Potete entrare e uscire dai turni liberamente, ma ricordate: **se non partecipate, non accumulate punti.**
- La frequenza dei turni verrà comunicata tramite i canali ufficiali (Discord e annunci IRL).

Domenica

10 10:00 | ⚡ Stop the Coding

Alle 10:00 di domenica scatta lo **stop the coding**:

- **Non è più possibile modificare il codice** in alcun modo.
- L'organizzazione eseguirà un **reset completo** del game.

⌚ 10:00 – 12:00 | Golden Run 🏆

La run finale e decisiva. Da qui in poi **lavorano solo i vostri agenti**, senza intervento umano.

- I turni si susseguiranno con una cadenza che può arrivare fino a **un turno ogni ~10 minuti**, ma la densità resta variabile e a discrezione degli organizzatori.
- Nessuna modifica al codice è consentita.
- I risultati di questa run hanno un **peso importante** nella classifica finale ma non definitivo.

| ⚡ **Preparatevi bene:** la Golden Run è il momento della verità. Il vostro agente dovrà cavarsela da solo.



▼ Criteri di Valutazione 💯

I criteri di valutazione sono i seguenti, in **ordine di importanza**:

1. Implementazione Tecnica

- La più importante tra tutte. Nei sistemi tipicamente agentici sono valutati capacità di long term planning, gestione della memory, correttezza nelle tool call, efficienza nel token usage, capacità di gestire eventi asincroni... La sfida è pensata affinché cerchiate di affrontare in maniera più o meno dettagliata questi temi.
- La soluzione **deve** utilizzare il framework `datapizza-ai`. L'utilizzo efficace del framework, eventuali customizzazioni e contributi creativi al suo ecosistema saranno valutati positivamente.

2. Risultati durante il gioco

- Valuteremo varie metriche durante i turni di gioco, quali in primis il fatturato, ma anche i clienti serviti, la reputazione e le relazioni strategiche con gli altri ristoranti. I risultati durante la Golden Run saranno considerati in maniera importante, ma non definitiva.

3. Pitch & Presentazione

- Sarà presa in considerazione l'efficacia del team nel comunicare e presentare la loro soluzione.

4. Creatività/Innovazione

- Anche soluzioni fuori dagli schemi sono ben accette! (Ma ricordate di seguire lo spirito del gioco, ad esempio: no human in the loop!) La creatività e l'inventiva sono premiate.

▼ Regolo.ai

Durante la sfida non dovete spendere soldi di tasca vostra (Let's go!).

Il provider di inferenza usato per gli agenti è **regolo.ai**, partner ufficiale di Hackapizza 2.0. Ogni team riceverà una chiave API per utilizzare il servizio.

```
from datapizza.clients.openai_like import OpenAILikeClient

client = OpenAILikeClient(
    api_key=os.getenv("REGOLO_API_KEY"),
    model="gpt-oss-120b",
    base_url="https://api.regolo.ai/v1",
)
```

Sono disponibili i seguenti modelli principali:

- `gpt-oss-120b`
- `gpt-oss-20b`
- `qwen3-vl-32b`

Trovate la lista completa su [regolo.ai/models-library](#). Se avete necessità di usare un modello diverso rispetto a quelli indicati sopra, chiedete all'organizzazione.

[Documentazione Regolo.ai](#)

[Documentazione OpenAILikeClient](#)

Forniremo l'url, la chiave API e le credenziali a ciascun team.



▼ Datapizza AI 🍕

Il framework **datapizza-ai** è obbligatorio per la sfida. È il nostro framework open-source per costruire agenti, RAG e pipeline Gen AI.

📖 **Documentazione completa:** docs.datapizza.ai

Link utili alla documentazione:

- [Agent — Build your first agent](#) - come costruire il vostro agente
- [MCP — Model Context Protocol](#) - integrazione MCP (usata dal server di gioco)
- [OpenAI-like Client](#) - per utilizzare Regolo.ai e modelli open-source
- [Tools](#) - come definire e usare tool
- [Tracing & Monitoring](#) - per integrare con Datapizza Monitoring
- [GitHub Repository](#)

▼ Datapizza Monitoring 📈

Siete stati selezionati come **primissimi tester** della versione alpha della nostra nuova app. Fare tracing non è mai stato così facile!

Registrati su datapizza-monitoring.datapizza.tech e scopri come monitorare i tuoi agenti con pochi semplici click.

Invitation code: `hackapizza2`

⚠ Nota: L'applicazione è attualmente in versione alpha e potrebbe presentare bug. Qualora riscontraste problemi, vi invitiamo a segnalarli al team Datapizza.

Utilities

▼ Regole Generali ✅

Durante la sfida è possibile utilizzare **qualsiasi modello LLM**. La soluzione deve essere implementata utilizzando `datapizza-ai`

N.B.: ricorda che la *creatività* e la *robustezza* della soluzione implementata rientrano tra i criteri di valutazione!

▼ Deliverables 📦

Al termine della competizione, ogni team deve preparare:

1. **Link al repository GitHub** contenente la soluzione implementata
2. **Presentazione:** Slide per presentare l'approccio, le sfide e i risultati della soluzione implementata, accompagnate da un pitch coinvolgente 🚀 - (*pitch di 2 minuti, i finalisti presenteranno successivamente un pitch esteso di 5 minuti*)

Caricate i file di deliverable tramite questo Google Form: docs.google.form

▼ Presentazioni e Fase Finale 🎤

La fase di presentazione si svolge **domenica pomeriggio a partire dalle 14:00** ed è suddivisa in due round.



Round 1 - Micro Pitch (2 minuti)

Ogni team presenta un **pitch di 2 minuti** davanti a uno dei quattro giudici della competizione, ciascuno affiancato da un membro del Team R&D di Datapizza.

Come funziona:

- I team vengono **divisi in 4 gruppi** (uno per giudice).
- Ogni giudice ascolta i micro pitch del proprio gruppo in sessioni parallele.
- Avete **2 minuti netti**: state incisivi, andate dritti al punto.

💡 Consiglio: in 2 minuti non potete raccontare tutto. Concentratevi su cosa rende unica la vostra soluzione, le scelte architettoniche chiave e i risultati ottenuti.

Al termine di tutti i micro pitch, la giuria si riunisce e seleziona i **5 team finalisti**.

Round 2 - Pitch Finale (5 + 3 minuti)

I 5 finalisti presentano davanti all'intera giuria con un **pitch esteso di 5 minuti**, seguito da **3 minuti di Q&A** con i giudici.

Questa è la vostra occasione per approfondire: approccio tecnico, sfide affrontate, strategia di gioco, lezioni apprese.

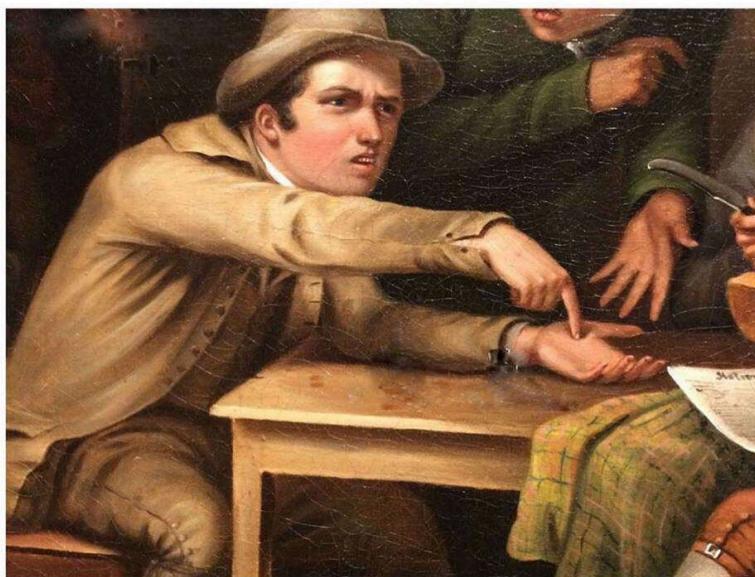
Al termine, la giuria delibera e decreta i **tre vincitori** (🥇🥈🥉) e gli eventuali premi speciali.

Cosa preparare: Tutti i team devono preparare **almeno** le slide per il micro pitch da 2 minuti. I team che accederanno alla finale dovranno essere pronti a presentare anche un pitch più esteso da 5 minuti: pianificate in anticipo entrambe le versioni per non farvi trovare impreparati.

▼ Premi 💰

I premi verranno comunicati durante il kickoff della giornata e verranno erogati tramite **Buoni Amazon**:

- 🥇 Primo posto: **€3.000** + Una giornata da passare insieme al Team R&D di Datapizza nei nostri uffici
- 🥈 Secondo posto: **€2.000** + Gadget Datapizza
- 🥉 Terzo posto: **€1.000** + Gadget Datapizza



▼ Link utili 🌐

- 🌐 Sito dell'evento: hackathon.datapizza.com
- 📊 Datapizza Monitoring: datapizza-monitoring.datapizza.tech — invitation code: [hackapizza2](#)
- 🚗 Regolo AI Models: regolo.ai/models-library
- 💬 Server Discord dell'evento: <https://discord.gg/rj9rXtb2kg>
 - Tutte le comunicazioni ufficiali avverranno sul canale # announcements 🎤 del server
- 📩 Google Form per caricare i file di deliverable: [docs.google.form](#)
- 📱 Rete WiFi: [R-GUEST](#)
 - Leggi l'informativa sulla Privacy e inserisci il tuo numero di cellulare preceduto dal prefisso +39
 - Inserisci in "authorization code" il codice ricevuto tramite SMS
 - Non ti arriva il messaggio? Chiedi in reception!

Regolamento Accessi Location

L'accesso alla struttura è libero fino alle **2:00 di domenica 1/03 mattina**, da quel momento in poi la porta principale sarà chiusa.

Durante tutta la notte sarà possibile accedere, dall'interno della location, al giardinetto privato della struttura.

Buona fortuna e Let's Hack! 

-Team Datapizza