

Tavotsu / Backend-fibo-hackathon ×

Code



Issues



Pulls 1



Actions



Settings



Conversation 34

Commits 25

Checks 1

Files changed 18

PR requested by @Zenith-AB

Summary by CodeRabbit

Release Notes

- New Features

- Added user authentication and account security with secure token-based access
- Implemented user-specific data isolation across campaigns, products, and plans
- Added asynchronous background processing for image generation tasks
- New endpoints to check job status and retrieve user-scoped plans
- Added API health check endpoint

- Infrastructure

















































- Enabled cross-origin resource sharing (CORS) for broader API compatibility
- Enhanced configuration management and API resilience

Tip: You can customize this high-level summary in your review settings.

Tavotsu 1 hour ago • edited by coderabbitai bot ▼

Owner

Zenith-AB(Zenith) added 25 commits 10 hours agofeat: implement auth system and user history2ad37b9

  Auth	9667f8e
  añadido sistema de check para conexiones	4e21153
  Mongo DB y Supabase	c4095b2
  Permisos CORS	a1417e0
  Ngrok y Health	b45b854
  Correccion Ruta Faltante	77929ee
  Script para checkear	5a46b19
  Error 500 por permisos Fix	8fab534
  Correccion Ruta Bria	c002978
  debug	898f3f0
  debugII	4f58873
  Correccion, Ahora usa V2	8f79085
  Vercel para Backend	a8cbb17
  Añadiendo Agente Groq	b19b8f1
  Fix: Historial y DB	✓ 77df9fe
  Docs: Update README for Release v1.0	580393e
  Docs: Restore original README	✓ 318effa
  Fix: CoderRabbit Review (Security & Stability)	✓ db27463
  Fix: CoderRabbit Feedback (Cleanup & Requirements Pinning)	✓ e6544ef
  Fix: CoderRabbit Round 2 (Security & Code Quality)	✓ 720721c
  Fix: Add missing return type annotation (Ruff ANN204)	✓ 2b9c43c
  Fix: Pin urllib3>=2.3.0 to resolve Critical CVEs	✓ 3194140
  little fix	c11e33e
  Fix: Final Polish (Utilities & Deps Pinning)	✓ 2067e24


Walkthrough

This pull request introduces multi-tenant authentication via OAuth2 and Supabase, applies per-user access control across API routes, migrates job storage from in-memory to asynchronous database persistence, updates LLM and image generation services, and adds supporting configuration, dependencies, and infrastructure tooling.


Changes

Cohort / File(s)	Summary
Authentication & Security app/api/deps.py , app/core/security.py	Introduces FastAPI dependency for OAuth2 authentication using Supabase JWT validation. Adds <code>AuthUser</code> model, <code>oauth2_scheme</code> endpoint, and <code>get_current_user()</code> async dependency with token validation and error handling. Provides bcrypt-based password hashing and JWT token creation utilities via <code>security.py</code> .
Configuration & Credentials app/core/config.py	Adds Supabase credentials (SUPABASE_URL, SUPABASE_KEY, SUPABASE_JWT_SECRET), OpenAI/LLM settings (OPENAI_BASE_URL, LLM_MODEL_NAME), JWT authentication settings (SECRET_KEY, ALGORITHM, ACCESS_TOKEN_EXPIRE_MINUTES), and updates Bria API endpoint paths from v1 to v2.
API Routes & Endpoints app/api/routes.py	Adds per-user access control via <code>current_user</code> dependency injection across all endpoints. Introduces async background job processing with <code>generate_async()</code> and <code>get_job_status()</code> endpoints. Updates <code>create_campaign</code> , <code>upload_product</code> , <code>generate_plan</code> , <code>execute_plan</code> , <code>get_plan</code> , and <code>list_campaigns</code> with user ownership validation. Adds new <code>list_plans()</code> endpoint.
Job Persistence & Orchestration app/services/jobs.py , app/schemas/fibo.py	Migrates job storage from in-memory dictionary to async ORM-backed database persistence via Beanie. Adds new <code>Job</code> document model with fields for status tracking, results, and metadata. Updates all job CRUD operations to async with database integration and adds <code>user_id</code> field to Campaign, Product, and Plan models for multi-tenancy.
Service Layer Updates app/services/llm_planner.py , app/services/agent.py , app/services/bria.py , app/services/storage.py	Switches LLM integration from local Ollama to OpenAI API with fallback handling. Updates Bria service to support both V1 and V2 API response formats. Enhances agent system prompts with multi-phase workflow. Modifies <code>upload_image_to_supabase()</code> to accept <code>user_id</code> parameter for user-scoped storage paths.

Cohort / File(s)	Summary
Application Initialization &	Registers <code>Job</code> model in Beanie initialization. Adds CORS middleware configuration allowing all origins. Includes new

 Open

feat: implement auth system and user history #8

Tavotsu wants to merge 25 commits into `main` from `release/production...` 

coderabbitai  commented 1 hour ago • edited ▾

Verification & Debug Utilities check_bria.py , check_connections.py , check_s3.py , debug_mongo_plans.py	Adds standalone scripts to verify Bria API connectivity, MongoDB and Supabase connection health, S3/Supabase storage functionality, and to enumerate Plan documents with variations from MongoDB.
---	---

Estimated code review effort

 4 (Complex) |  ~45–75 minutes

- Heterogeneous changes across multiple layers** — authentication system, data model updates (adding `user_id`), service layer refactoring (Ollama→OpenAI, in-memory→persistent jobs), and infrastructure updates make this a multi-faceted review.
- Security-critical logic** — OAuth2 integration, Supabase JWT validation, access control enforcement, and password handling require careful verification.
- Architectural shift in job processing** — migration from synchronous in-memory storage to asynchronous ORM-backed persistence with new `async/await` patterns throughout.
- Multiple service integrations** — LLM provider switch, Bria API version handling, storage user-scoping, and new OpenAI client initialization.

Areas requiring extra attention:

- `app/api/routes.py` : Verify user ownership checks are consistently enforced across all endpoints (campaigns, products, plans, jobs).
- `app/services/jobs.py` : Confirm `async/await` patterns and ORM interactions are correctly implemented; check error handling for database failures.
- `app/api/deps.py` : Validate Supabase JWT verification logic, token extraction, and error response codes (401 vs. 403).
- `app/services/llm_planner.py` : Ensure OpenAI client initialization respects environment configuration and fallback paths work as intended.
- `app/services/storage.py` : Confirm user-scoped path construction and validation logic prevent path traversal or file access issues.
- `app/core/config.py` : Check for potential missing/empty `SECRET_KEY` initialization and verify all required Supabase credentials are documented.

Poem

🐇 A rabbit hops through auth's new gate,
Where users claim their plans and fate,
Jobs dance async, no longer bound,
To memory's cage—now DB-bound!
With Supabase keys and OpenAI's call,
Multi-tenant dreams rise tall! 🌟

Pre-merge checks and finishing touches

- ▶ ❌ Failed checks (1 warning)
- ▶ ✅ Passed checks (2 passed)
- ▶ ✨ Finishing touches

Comment `@coderabbitai help` to get the list of available commands and usage tips.



coderabbitai (bot) reviewed 1 hour ago

[View reviewed changes](#)



coderabbitai (bot) left a comment

Actionable comments posted: 32

⚠️ Caution

Some comments are outside the diff and can't be posted inline due to platform limitations.

- ▶ ⚠️ Outside diff range comments (5)

- ▶ 📄 Review details



app/api/deps.py

Comment on lines **+1** to **+14**

```
1 from fastapi import Depends, HTTPException, status
2 from fastapi.security import OAuth2PasswordBearer
3 from supabase import create_client, Client
4 from app.core.config import settings
5 from pydantic import BaseModel
6
7 # Dummy endpoint for Swagger (since we use Supabase external auth)
8 oauth2_scheme = OAuth2PasswordBearer(tokenUrl="/api/v1/auth/login")
9
10 import threading
11
12 # Lazy loading Supabase Client with atomic initialization
13 _supabase: Client | None = None
14 _supabase_lock = threading.Lock()
```



coderabbitai (bot) 1 hour ago

⚠️ Potential issue | 🔴 Critical

Reorganize imports and add missing `logger` definition.

The `threading` import on line 10 should be grouped with other standard library imports at the top. More critically, `logger` is referenced on line 49 but is never defined, which will cause a `NameError` at runtime.

Apply this diff to fix the import ordering and add the logger:

```
from fastapi import Depends, HTTPException, status
from fastapi.security import OAuth2PasswordBearer
from supabase import create_client, Client
+import threading
+import logging
from app.core.config import settings
from pydantic import BaseModel

# Dummy endpoint for Swagger (since we use Supabase external auth)
oauth2_scheme = OAuth2PasswordBearer(tokenUrl="/api/v1/auth/login")

- import threading
+ logger = logging.getLogger(__name__)
```



▶️ 🤖 Prompt for AI Agents



Resolve conversation

app/api/routes.py

Comment on lines 88 to 90

73	88	product = await Product.get(request.product_id)
74		if not product:
	89	if not product or product.user_id != current_user.id:
75	90	raise HTTPException(404, "Producto no encontrado")



coderabbitai (bot) 1 hour ago

⚠️ Potential issue | 🟡 Minor

Verify product belongs to the specified campaign.

The endpoint validates that the product belongs to the current user, but does not verify that the product actually belongs to the specified `campaign_id`. A user could potentially generate a plan for a campaign using a product from a different campaign.

```
product = await Product.get(request.product_id)
- if not product or product.user_id != current_user.id:
+ if not product or product.user_id != current_user.id or product.campaign_id != campai
  raise HTTPException(404, "Producto no encontrado")
```



▶ 📄 Committable suggestion

▶ 🤖 Prompt for AI Agents



Reply...

Resolve conversation

app/api/routes.py

Comment on lines +186 to +190

186	# List User Plans (History)
187	@router.get("/plans", response_model=List[Plan])
188	async def list_plans(current_user: deps.AuthUser = Depends(deps.get_current_user)):
189	"""Lista todos los planes (historial) del usuario, ordenados por fecha"""
190	return await Plan.find(Plan.user_id == current_user.id).sort("-created_at").to_




coderabbitai (bot) 1 hour ago

 Nitpick |  Trivial



Consider adding pagination for scalability.

The endpoint returns all plans for a user without pagination. For users with extensive history, this could lead to large response payloads and slow queries.

```
+from typing import Optional
+
@router.get("/plans", response_model=List[Plan])
-async def list_plans(current_user: deps.AuthUser = Depends(deps.get_current_user)):
+async def list_plans(
+    current_user: deps.AuthUser = Depends(deps.get_current_user),
+    skip: int = 0,
+    limit: int = 50
+):
    """Lista todos los planes (historial) del usuario, ordenados por fecha"""
    - return await Plan.find(Plan.user_id == current_user.id).sort("-created_at").to_list()
    + return await Plan.find(Plan.user_id == current_user.id).sort("-created_at").skip(skip
```





- ▶  Tools
- ▶  Prompt for AI Agents



Reply...

Resolve conversation

app/api/routes.py

Comment on lines +203 to +205

203	from fastapi import Form
204	from app.services import jobs
205	from app.schemas.fibo import BriaParameters

 coderabbitai bot 1 hour ago

 Nitpick |  Trivial

Move imports to the top of the file.

These imports should be consolidated with the other imports at the top of the file for better organization and PEP 8 compliance.

- ▶  Prompt for AI Agents

Resolve conversation

app/api/routes.py

Comment on lines +262 to +270

```
262     async def process_generation_job(  
263         job_id: str,  
264         prompt: str,  
265         image_url: str = None,  
266         variations: int = 4,  
267         brand_guidelines: str = None,  
268         user_id: str = None, # Now accepts user_id  
269         aspect_ratio: str = "1:1" # New param  
270     ):
```

coderabbitai bot 1 hour ago

Refactor suggestion | Major

Use explicit `Optional` type hints.

Per PEP 484 and Ruff RUF013, implicit `Optional` via `= None` is discouraged. Use explicit type annotations.

```
+from typing import Optional  
+  
    async def process_generation_job(  
        job_id: str,  
        prompt: str,  
-        image_url: str = None,  
+        image_url: Optional[str] = None,  
        variations: int = 4,  
-        brand_guidelines: str = None,  
-        user_id: str = None,  
+        brand_guidelines: Optional[str] = None,  
+        user_id: Optional[str] = None,  
        aspect_ratio: str = "1:1"  
    ):
```



Committable suggestion skipped: line range outside the PR's diff.

► Tools

► Prompt for AI Agents



Reply...

Resolve conversation

app/api/routes.py

Comment on lines +277 to +278

277	from app.services import bria
278	from app.schemas.fibo import Plan, ProposedVariation, BriaParameters



coderabbitai bot 1 hour ago

Nitpick | Trivial

Move imports to the top of the file.

Imports inside functions should be avoided unless necessary to prevent circular imports. Consider moving these to the top-level imports.

▶ Prompt for AI Agents



Reply...

Resolve conversation

app/api/routes.py

Comment on lines +314 to +319

314	if isinstance(sp, str):
315	try:
316	import json
317	sp = json.loads(sp)
318	except json.JSONDecodeError:
319	sp = {}



coderabbitai bot 1 hour ago

Refactor suggestion | Major

Move json import outside the loop.

The `json` module is imported inside the loop on each iteration where the condition is met. This is inefficient and should be moved to the top of the file.

```
+import json # Add at top of file

# Handle SP format
sp = res.get("structured_prompt", {})
if isinstance(sp, str):
    try:
        import json
        sp = json.loads(sp)
    except json.JSONDecodeError:
        sp = {}
```



Committable suggestion skipped: line range outside the PR's diff.

▶  Prompt for AI Agents



Resolve conversation

app/api/routes.py

Comment on lines +356 to +358

356	<code>except Exception as e:</code>
357	<code>logger.error(f"Job failed: {e}")</code>
358	<code>await jobs.fail_job(job_id, str(e))</code>



coderabbitai bot 1 hour ago

 Nitpick |  Trivial




Capture traceback when failing jobs.

The `fail_job` function accepts a `trace` parameter, but it's not being populated here. Capturing the traceback would aid debugging.

```
+ import traceback
+
except Exception as e:
-     logger.error(f"Job failed: {e}")
-     await jobs.fail_job(job_id, str(e))
+     logger.exception(f"Job {job_id} failed")
+     await jobs.fail_job(job_id, str(e), trace=traceback.format_exc())
```



Committable suggestion skipped: line range outside the PR's diff.

- ▶  Tools
- ▶  Prompt for AI Agents
- 

Reply...

Resolve conversation

app/api/routes.py

Comment on lines +367 to +369

367	# Security: Enforce Ownership
368	if status.get("user_id") and status.get("user_id") != current_user.id:
369	raise HTTPException(status_code=404, detail="Job not found")





coderabbitai bot 1 hour ago

 Potential issue |  Minor

- ▶  Analysis chain

Confirm: `user_id` is optional and creates a security gap in job ownership validation.

The `create_job` function signature allows `user_id` to be `None` (line 36 in `app/services/jobs.py`), and the `Job` model defines it as optional (`app/schemas/fibo.py:136`). The ownership check at lines 368-369 only validates if `user_id` is present; jobs without a `user_id` would be accessible to any authenticated user. While the current single caller at line 245 always provides `user_id=current_user.id`, the function design allows this gap to exist. Either make `user_id` required in the `Job` model and function signature, or explicitly deny access when `user_id` is missing.

- ▶  Prompt for AI Agents
- 

Reply...

Resolve conversation

app/core/config.py

Comment on lines +41 to +44

41	# Auth Settings
42	SECRET_KEY: str = os.getenv("")
43	ALGORITHM: str = "HS256"
44	ACCESS_TOKEN_EXPIRE_MINUTES: int = 60 * 24 * 7 # 1 week

 coderabbitai (bot) 1 hour ago




 Potential issue |  Critical

Critical: SECRET_KEY configuration is broken.

os.getenv("") passes an empty string as the environment variable name, which will always return None . This breaks JWT signing and should be:

```
# Auth Settings
- SECRET_KEY: str = os.getenv("")
+ SECRET_KEY: str = os.getenv("SECRET_KEY", "")
ALGORITHM: str = "HS256"
ACCESS_TOKEN_EXPIRE_MINUTES: int = 60 * 24 * 7 # 1 week
```



- ▶  Committable suggestion
 - ▶  Prompt for AI Agents
- 

Reply...

Resolve conversation

app/core/security.py

Comment on lines +1 to +5

1	from datetime import datetime, timedelta
2	from typing import Optional, Union, Any
3	from jose import jwt
4	from passlib.context import CryptContext
5	from app.core.config import settings

 coderabbitai (bot) 1 hour ago

 Refactor suggestion |  Major

Consolidate duplicate imports.

datetime module components are imported twice. Consolidate at the top:

```
-from datetime import datetime, timedelta
+from datetime import datetime, timedelta, timezone
from typing import Optional, Union, Any
from jose import jwt
from passlib.context import CryptContext
from app.core.config import settings

pwd_context = CryptContext(schemes=["bcrypt"], deprecated="auto")
...
-from datetime import timezone
-
def create_access_token(subject: Union[str, Any], expires_delta: Optional[timedelta] = None):
```

Also applies to: 15-15

▶  Prompt for AI Agents



Resolve conversation

app/core/security.py

Comment on lines +17 to +25

```
17     def create_access_token(subject: Union[str, Any], expires_delta: Optional[timedelta] = None):
18         if expires_delta:
19             expire = datetime.now(timezone.utc) + expires_delta
20         else:
21             expire = datetime.now(timezone.utc) + timedelta(minutes=settings.ACCESS_TOKEN_EXPIRE_SECONDS)
22
23         to_encode = {"exp": expire, "sub": str(subject)}
24         encoded_jwt = jwt.encode(to_encode, settings.SECRET_KEY, algorithm=settings.ALGORITHM)
25         return encoded_jwt
```






coderabbitai bot 1 hour ago

⚠ Potential issue | 🔴 Critical

▶  Analysis chain

Fix critical configuration error in SECRET_KEY.

Line 42 in app/core/config.py contains SECRET_KEY: str = os.getenv("") , which passes an empty string to os.getenv() . This will always return None , making JWT token creation fail. Change to SECRET_KEY: str = os.getenv("SECRET_KEY") with an appropriate default or ensure the environment variable is set as required for production.

- ▶  Tools
- ▶  Prompt for AI Agents
- 

Reply...

Resolve conversation

app/main.py

Comment on lines +39 to +45

```
39 app.add_middleware(  
40     CORSMiddleware,  
41     allow_origins=["*"], # Permitir todo para el Hackathon (Frontend local, etc)  
42     allow_credentials=True,  
43     allow_methods=["*"],  
44     allow_headers=["*"],  
45 )
```

 coderabbitai bot 1 hour ago

 Potential issue |  Major

Security concern: Overly permissive CORS configuration.

Using allow_origins=["*"] with allow_credentials=True is a security risk. Per the CORS specification, when credentials are allowed, the Access-Control-Allow-Origin header cannot be * and browsers will reject such responses. This configuration is contradictory and may not behave as expected.

While the comment indicates this is for the hackathon, consider restricting origins before production:

```
app.add_middleware(  
    CORSMiddleware,  
-   allow_origins=["*"], # Permitir todo para el Hackathon (Frontend local, etc)  
+   allow_origins=["http://localhost:3000", "http://localhost:5173"], # Add specific frc  
    allow_credentials=True,  
    allow_methods=["*"],
```



```
allow_headers=["*"],  
)
```

▶  Prompt for AI Agents



Resolve conversation

app/schemas/fibo.py

Comment on lines +141 to +142

141	created_at: float = Field(default_factory=lambda: datetime.now().timestamp())
142	updated_at: float = Field(default_factory=lambda: datetime.now().timestamp())



coderabbitai bot 1 hour ago

⚠️ Potential issue | 🟡 Minor

Use timezone-aware timestamps.

`datetime.now()` without a `timezone` argument returns a naive datetime, which can cause inconsistencies in distributed systems or across time zones.

```
- created_at: float = Field(default_factory=lambda: datetime.now().timestamp())  
- updated_at: float = Field(default_factory=lambda: datetime.now().timestamp())  
+ created_at: float = Field(default_factory=lambda: datetime.now(tz=datetime.timezone.utc).timestamp())  
+ updated_at: float = Field(default_factory=lambda: datetime.now(tz=datetime.timezone.utc).timestamp())
```

Or import and use `datetime.timezone.utc`:

```
from datetime import datetime, timezone  
# ...  
created_at: float = Field(default_factory=lambda: datetime.now(tz=timezone.utc).timestamp())
```

▶  Tools

▶  Prompt for AI Agents



Reply...

Resolve conversation

app/services/agent.py

Comment on lines 27 to +30


24


27

28


29

30

 coderabbitai bot 1 hour ago

 Nitpick

 |

 Trivial

Remove extraneous blank lines.

Multiple consecutive blank lines reduce readability.


```
        return _openai_client
```


```
-
```


```
-
```

```
-
```

```
    async def brand_guidelines_to_variations(
```

▶  Committable suggestion

▶  Prompt for AI Agents



Reply...

Resolve conversation

app/services/bria.py

Comment on lines 65 to 74

58

65

59

60

61

```
        return {
```

```
            "image_url": result.get("result_url"),
```

```
            "structured_prompt": result.get("structured_prompt"),
```

```
            "status": result.get("status")
```

https://github.com/Tavotsu/Backend-fibo-hackathon/pull/8

17/31

```

66         "image_url": img_url,
67         "structured_prompt": result_data.get("structured_prompt"),
68         "status": result_data.get("status", "complete")
62     69     }
63     else:
64         70         error_msg = f"Error FIBO API: {response.status_code} - {response.text}"
65         71         logger.error(error_msg)
66         72         with open("backend_error.log", "a") as f:
67             73             f.write(f"API Error: {error_msg}\n")
68         74         raise BriaAPIError(error_msg)

```



coderabbitai (bot) 1 hour ago



Potential issue



Critical

Critical: Unreachable error handling code.

The `return` statement on line 69 makes lines 70-74 unreachable. The error handling for non-200 responses will never execute. This appears to be a merge/edit issue where the success block and error block got combined incorrectly.

Apply this diff to fix the control flow:

```

return {
    "image_url": img_url,
    "structured_prompt": result_data.get("structured_prompt"),
    "status": result_data.get("status", "complete")
}
+ else:
    error_msg = f"Error FIBO API: {response.status_code} - {response.text}"
    logger.error(error_msg)
    with open("backend_error.log", "a") as f:
        f.write(f"API Error: {error_msg}\n")
    raise BriaAPIError(error_msg)

```



Committable suggestion skipped: line range outside the PR's diff.

► Tools

► Prompt for AI Agents



Reply...

Resolve conversation

app/services/bria.py

Comment on lines +72 to +73

72	with open("backend_error.log", "a") as f:
73	f.write(f"API Error: {error_msg}\n")



coderabbitai (bot) 1 hour ago

Refactor suggestion | Major

Avoid blocking file I/O in async functions.

Using `open()` in an async function blocks the event loop. Since `logger.error()` is already called, the file write is redundant. If persistent error logging is needed, configure a file handler on the logger instead.

```
error_msg = f"Error FIBO API: {response.status_code} - {response.text}"
logger.error(error_msg)
- with open("backend_error.log", "a") as f:
-     f.write(f"API Error: {error_msg}\n")
raise BriaAPIError(error_msg)
```



▶ Committable suggestion

▶ Tools

▶ Prompt for AI Agents



Reply...

Resolve conversation

app/services/bria.py

Comment on lines 76 to 81

68	76	except httpx.TimeoutException:
69	77	raise BriaAPIError("Timeout al conectar con Bria API")
70	78	except httpx.RequestError as e:
	79	with open("backend_error.log", "a") as f:
	80	f.write(f"Request Error: {str(e)}\n")
71	81	raise BriaAPIError(f"Error de conexión: {str(e)}")



coderabbitai (bot) 1 hour ago





 Refactor suggestion |  Major

Chain exceptions and remove blocking file I/O.

Use `raise ... from` to preserve the exception chain for better debugging. Also, the blocking file write should be removed (same issue as above).

```
except httpx.TimeoutException:
-     raise BriaAPIError("Timeout al conectar con Bria API")
+     raise BriaAPIError("Timeout al conectar con Bria API") from None
except httpx.RequestError as e:
-     with open("backend_error.log", "a") as f:
-         f.write(f"Request Error: {str(e)}\n")
-     raise BriaAPIError(f"Error de conexión: {str(e)}")
+     raise BriaAPIError(f"Error de conexión: {e}") from e
```



- ▶  Committable suggestion
 - ▶  Tools
 - ▶  Prompt for AI Agents
- 

Reply...

Resolve conversation

app/services/jobs.py

Comment on lines +63 to +72

```
63     async def update_job(job_id: str, **kwargs):
64         job = await get_job(job_id)
65         if not job:
66             return
67
68         for k, v in kwargs.items():
69             if hasattr(job, k):
70                 setattr(job, k, v)
71         job.updated_at = time.time()
72         await job.save()
```

 coderabbitai bot 1 hour ago

 Nitpick |  Trivial

Consider adding type annotation for `**kwargs` .

For better IDE support and type checking, consider typing the kwargs:

```
-async def update_job(job_id: str, **kwargs):  
+async def update_job(job_id: str, **kwargs: Any) -> None:
```



Committable suggestion skipped: line range outside the PR's diff.

► Tools

► Prompt for AI Agents



Resolve conversation

app/services/jobs.py

Comment on lines +74 to +87

```
74     async def add_event(job_id: str, message: str):  
75         job = await get_job(job_id)  
76         if not job:  
77             return  
78  
79         job.events.append(  
80             "t": time.time(),  
81             "msg": message  
82         })  
83         # Keep only last 250 events  
84         if len(job.events) > 250:  
85             job.events = job.events[-250:]  
86         job.updated_at = time.time()  
87         await job.save()
```



coderabbitai (bot) 1 hour ago

Nitpick | Trivial

Potential race condition with concurrent updates.

The pattern of `await get_job()` followed by mutation and `await job.save()` is not atomic. Under high concurrency, two simultaneous `add_event` calls could result in lost events. For a hackathon this is acceptable, but for production consider using MongoDB's `$push` operator via Beanie's update operations:



```
async def add_event(job_id: str, message: str):
    event = {"t": time.time(), "msg": message}
    await Job.find_one(Job.job_id == job_id).update(
        {"$push": {"events": event}, "$set": {"updated_at": time.time()}}
    )
```

This applies similarly to `add_result` and `add_partial_result`.

▶ Prompt for AI Agents



Resolve conversation

app/services/llm_planner.py

Comment on lines 82 to +84

115	82	<code>except Exception as e:</code>
116		<code>logger.warning(f"Error llamando a LLM: {e}. Usando fallback.")</code>
117		
	83	<code>logger.exception(f"Error LLM ({self.model}): {e}")</code>
	84	



coderabbitai (bot) 1 hour ago

Nitpick | Trivial

Fix redundant exception in logging.

`logger.exception()` already includes the exception information; passing `e` in the f-string is redundant.

```
except Exception as e:
-     logger.exception(f"Error LLM ({self.model}): {e}")
+     logger.exception("Error LLM (%s)", self.model)
```



▶ Committable suggestion

▶ Tools

▶ Prompt for AI Agents



[Resolve conversation](#)

app/services/llm_planner.py

Comment on lines +87 to +108

```
87         @staticmethod
88         def _safe_json_extract(text: str) -> Optional[str]:
89             """Extrae el primer bloque JSON válido de un string (hacky robustez)."""
90             if not text:
91                 return None
92             text = text.strip()
93             # Remove markdown fences
94             if text.startswith("```"):
95                 text = text.strip("`")
96             # drop lang
97             if "\n" in text:
98                 text = text.split("\n", 1)[1]
99
100             text = text.strip()
101
102             # Find start
103             start_candidates = [i for i in (text.find("{"), text.find "[")) if i != -1]
104             if not start_candidates:
105                 return None
106
107             start = min(start_candidates)
108             return text[start:]
```



coderabbitai (bot) 1 hour ago



Potential issue |



Minor

JSON extraction may return incomplete JSON.

The `_safe_json_extract` method finds the start of JSON but doesn't validate or find the matching closing bracket. This could return truncated JSON if there's trailing text.

Consider adding validation or finding the balanced closing bracket:


```
start = min(start_candidates)
- return text[start:]
+ candidate = text[start:]
+ # Attempt to validate by parsing
+ try:
+     json.loads(candidate)
```




```
+         return candidate
+     except json.JSONDecodeError:
+         # Try to find balanced brackets
+         pass
+     return candidate # Return as-is, let caller handle parse errors
```

Alternatively, since `json.loads()` is called immediately after in line 74, the current approach is acceptable if parse errors are handled there—which they are via the `except` block falling through to `_fallback()` .

Committable suggestion skipped: line range outside the PR's diff.

▶  Prompt for AI Agents



Reply...

Resolve conversation

check_bria.py

Comment on lines +12 to +14

12	# Based on search results: V2 is more standard now
13	base_url = "https://engine.prod.bria-api.com/v2"
14	endpoint = "/image/generate"

 coderabbitai bot 1 hour ago


 Nitpick |  Trivial


Consider using centralized configuration instead of hardcoded values.

The base URL and endpoint are hardcoded here but also defined in `app/core/config.py` . For consistency and to avoid drift, consider importing from the shared config:

```
- # Based on search results: V2 is more standard now
- base_url = "https://engine.prod.bria-api.com/v2"
- endpoint = "/image/generate"
+ from app.core.config import settings
+ base_url = settings.BRIA_API_URL
+ endpoint = settings.BRIA_IMAGE_GENERATE_ENDPOINT
```



▶  Committable suggestion

▶  Prompt for AI Agents

Resolve conversation

check_bria.py

Comment on lines +49 to +54

```
49         if "result_url" in data:
50             print(f"🖼️ Image URL: {data['result_url']}")
51         elif isinstance(data, list) and len(data) > 0 and "urls" in data[0]:
52             print(f"🖼️ Image URL: {data[0]['urls'][0]}")
53         else:
54             print("⚠️ Warning: Unexpected JSON structure.")
```



coderabbitai (bot) 1 hour ago

🔪 Nitpick | 🟦 Trivial

Response parsing differs from production service.

The production code in `app/services/bria.py` handles `result.image_url` (nested) but this script checks for `result_url` (top-level) and array formats. Consider aligning the parsing logic:

```
if "result_url" in data:
    print(f"🖼️ Image URL: {data['result_url']}")
+ elif "result" in data and "image_url" in data["result"]:
+     print(f"🖼️ Image URL: {data['result']['image_url']}")
+ elif "image_url" in data:
+     print(f"🖼️ Image URL: {data['image_url']}")
elif isinstance(data, list) and len(data) > 0 and "urls" in data[0]:
    print(f"🖼️ Image URL: {data[0]['urls'][0]}")
```



▶️ 📄 Committable suggestion

▶️ 🤖 Prompt for AI Agents

Resolve conversation

check_connections.py

Comment on lines +33 to +35

33	if not url or "supabase.co" not in url:
34	print("❌ Supabase URL is missing or invalid.")
35	return False

 coderabbitai bot 1 hour ago

 Potential issue |  Minor

Overly restrictive Supabase URL validation.

The check "supabase.co" not in url will fail for self-hosted Supabase instances or alternative domains. Consider validating URL structure instead of domain content.

- if not url or "supabase.co" not in url:


- print("❌ Supabase URL is missing or invalid.")

- return False

+ if not url or not url.startswith(("http://", "https://")):

+ print("❌ Supabase URL is missing or invalid (must be a valid HTTP(S) URL).")

+ return False



▶  Prompt for AI Agents



Reply...

Resolve conversation

2 hidden conversations

[Load more...](#)

debug_mongo_plans.py

Comment on lines +1 to +7

1	import asyncio
2	import os
3	from dotenv import load_dotenv
4	load_dotenv()
5	from motor.motor_asyncio import AsyncIOMotorClient
6	from beanie import init_beanie
7	from app.schemas.fibo import Campaign, Product, Plan



coderabbitai bot 1 hour ago

Nitpick | Trivial

Consider import ordering.

`load_dotenv()` is called before importing app modules. This is intentional if app modules depend on environment variables at import time, but the unconventional placement could confuse readers. Consider grouping standard imports first:

```
import asyncio
import os
+
from dotenv import load_dotenv
load_dotenv()
+
from motor.motor_asyncio import AsyncIOMotorClient
```



▶ Prompt for AI Agents



Resolve conversation

debug_mongo_plans.py

Comment on lines +22 to +25

22	<code>await init_beanie(</code>
23	<code> database=client[db_name],</code>
24	<code> document_models=[Campaign, Product, Plan]</code>
25	<code>)</code>



coderabbitai bot 1 hour ago

Nitpick | Trivial

Document models list is missing `Job` .

The main application in `app/main.py` registers `Job` alongside `Campaign` , `Product` , and `Plan` . For consistency:

```
-from app.schemas.fibo import Campaign, Product, Plan
+from app.schemas.fibo import Campaign, Product, Plan, Job
...
    await init_beanie(
```



```
database=client[db_name],
- document_models=[Campaign, Product, Plan]
+ document_models=[Campaign, Product, Plan, Job]
)
```

Committable suggestion skipped: line range outside the PR's diff.

▶ Prompt for AI Agents



Reply...

Resolve conversation

requirements.txt

```
12 beanie
13 boto3
14 pydantic-settings
1 fastapi>=0.109.0
```



coderabbitai bot 1 hour ago

⚠️ Potential issue | 🔴 Critical

Update fastapi to 0.109.1+ to address HIGH severity vulnerability.

FastAPI 0.109.0 has a HIGH severity vulnerability (PYSEC-2024-38) related to python-multipart dependencies. FastAPI 0.109.1 released a patch for this issue. Update to fastapi>=0.109.1 to mitigate the vulnerability.

```
-fastapi>=0.109.0
+fastapi>=0.109.1
```



▶ Committable suggestion

▶ Tools

▶ Prompt for AI Agents



Reply...

Resolve conversation

requirements.txt

```
4 pydantic-settings>=2.0,<3
5 python-dotenv>=1.0.0
6 pandas>=2.0.0
7 requests>=2.31.0
```



coderabbitai (bot) 1 hour ago



Potential issue | Critical

Update requests to 2.32.4+ to address .netrc credentials leak vulnerability.

Requests releases prior to 2.32.4 may leak .netrc credentials to third parties for specific maliciously-crafted URLs ([GHSA-9h9g-9r4m-mvj7](#), MEDIUM severity). Users should upgrade to version 2.32.4 to receive a fix.

```
-requests>=2.31.0
+requests>=2.32.4
```



► Committable suggestion

► Tools

► Prompt for AI Agents



Reply...

Resolve conversation

requirements.txt

```
15 bcrypt<4.0.0
16 python-jose[cryptography]
17 supabase
18 urllib3>=2.5.0
```



coderabbitai (bot) 1 hour ago



Potential issue | Critical

Update urllib3 to 2.6.0+ to address HIGH severity DoS vulnerabilities.

The number of links in the decompression chain was unbounded allowing a malicious server to insert a virtually unlimited number of compression steps leading to high CPU usage and massive memory allocation for the decompressed data ([GHSA-gm62-xv2j-4w53](#), HIGH severity). Upgrade to at least urllib3 v2.6.0 in which the library limits the number of links to 5.

```
-urllib3>=2.5.0  
+urllib3>=2.6.0
```



▶  Committable suggestion

▶  Tools

▶  Prompt for AI Agents



Resolve conversation

