



Duale Taalentfy — Contrato del WebSocket

Cómo comunicarse con el servicio de descubrimiento de empresas: el mensaje que envía el cliente y **todos** los eventos que devuelve el servidor, con su significado.

Endpoint (producción): `wss://web-production-addac.up.railway.app/company-search`

Endpoint (local): `ws://localhost:8001/company-search`

El patrón es simple: el cliente abre la conexión, envía **un único mensaje JSON** y luego sólo **escucha**. El servidor responde con un *stream* de eventos hasta cerrar con `complete` (éxito) o `error` (fallo). Cada evento tiene siempre la misma envoltura:

```
{ "type": "<tipo>", "data": { ... } }
```

1. Autenticación

El WebSocket **exige** una API key. Se puede enviar de dos formas:

- **En la URL (recomendado):** `wss://.../company-search?api_key=TU_API_KEY`
- **En el primer mensaje JSON:** añadiendo el campo `"api_key": "TU_API_KEY"` (el servidor lo extrae y lo elimina antes de validar la petición).

El front servido en / inyecta la key automáticamente. Si la key falta o es incorrecta, el servidor responde con un evento `error` y cierra con código `1008` (ver §6).

2. Mensaje cliente → servidor (único)

Nada más conectar, el cliente envía **un solo** mensaje con la formación y la zona:

```
{
  "api_key": "TU_API_KEY",
  "formacion": {
    "url": "https://iesfernandoiii.es/.../c-f-g-s-desarrollo-de-aplicaciones-web/",
    "descripcion": "CFGs Desarrollo de Aplicaciones Web (DAW)"
  },
  "ubicacion": { "texto": "provincia de Jaén, España" },
  "max_empresas": 6,
  "excluir": ["Brooktec"]
}
```

Campo	Obligatorio	Significado
<code>formacion.descripcion</code>	Al menos uno de los dos	Texto libre de la formación (FP/ciclo).
<code>formacion.url</code>	Al menos uno de los dos	URL de la formación; el agente la lee para extraer el detalle.
<code>ubicacion.texto</code>	<code>texto</code> o <code>(lat + lng)</code>	Zona en lenguaje natural: «provincia de Jaén, España».
<code>ubicacion.lat / lng / radio_km</code>	Alternativa a <code>texto</code>	Zona por coordenadas: <code>{ "lat": 37.77, "lng": -3.79, "radio_km": 50 }</code> .
<code>max_empresas</code>	No	Nº objetivo de empresas (1–60). Si se omite, usa <code>MAX_COMPANIES</code> del servidor.
<code>excluir</code>	No	Lista de nombres que no deben aparecer (lo usa «Ampliar búsqueda»).

3. Eventos servidor → cliente (resumen)

type	Cuándo se emite	Veces	data
status	Progreso (arranque, inicio de cada lote, zona agotada)	1..N	{ stage, message }
discovery	Tras descubrir un lote de candidatas	1..N (una por lote)	{ total, empresas: [nombre, ...] }
company	Por cada empresa válida enriquecida	0..N	Ficha completa (ver §5.3)
discarded	Por cada empresa descartada por un guardrail	0..N	{ nombre, motivo }
complete	Al terminar (si no hubo error)	1 (final)	{ total, empresas, descartadas, coste }
error	Ante cualquier fallo	0..1 (final)	{ message }

El orden de company / discarded **no es determinista**: cada ficha se emite en cuanto su enrichment termina (hasta 30 en paralelo).
Discovery y enrichment se **solapan**: mientras se enriquece un lote, ya se descubre el siguiente.

4. Sesión completa de ejemplo (en orden de llegada)

```
// 1) Arranque de la búsqueda (siempre el primero)
{ "type": "status", "data": {
  "stage": "discovery",
  "message": "Analizando la formación y buscando empresas candidatas..." } }

// 2) Comienzo del primer lote de descubrimiento
{ "type": "status", "data": {
  "stage": "discovery",
  "message": "Lote 1: buscando 10 empresas (excluyendo 1 ya vistas) – 0/6 encontradas..." } }

// 3) Lote 1 descubierto: candidatas (sólo nombres; aún sin enriquecer)
{ "type": "discovery", "data": {
  "total": 5,
  "empresas": ["YusApi S.L.", "Quántico Soluciones Tecnológicas S.L.",
    "EBsoft S.L.", "Fornell Software", "ITSICAP"] } }

// 4) Se descubre el lote 2 EN PARALELO mientras se enriquece el 1 (solapamiento)
{ "type": "status", "data": {
  "stage": "discovery",
  "message": "Lote 2: buscando 10 empresas (excluyendo 6 ya vistas) – 0/6 encontradas..." } }
{ "type": "discovery", "data": {
  "total": 3, "empresas": ["INFORCLOUD", "TecnoSector", "NOK Soluciones Tecnológicas"] } }

// 5) Las fichas válidas van llegando según terminan (orden NO determinista)
{ "type": "company", "data": { "nombre": "Quántico Soluciones Tecnológicas S.L.", ...ficha... } }
{ "type": "company", "data": { "nombre": "EBsoft S.L.", ...ficha... } }

// 6) Una candidata descartada por guardrail (no cuenta hacia el objetivo)
{ "type": "discarded", "data": {
  "nombre": "ITSICAP", "motivo": "Ninguna de sus sedes está dentro de la provincia de Jaén." } }

{ "type": "company", "data": { "nombre": "YusApi S.L.", ...ficha... } }
{ "type": "company", "data": { "nombre": "INFORCLOUD", ...ficha... } }

// 7) La zona se agota (varios lotes seguidos sin novedad). Sólo aparece si se agota.
{ "type": "status", "data": {
  "stage": "discovery", "message": "No se encontraron más empresas nuevas en la zona." } }

// 8) Fin: listado completo + coste real (SIEMPRE el último en éxito)
{ "type": "complete", "data": {
  "total": 6, "empresas": [ ...6 fichas... ],
  "descartadas": [ { "nombre": "ITSICAP", "motivo": "Ninguna de sus sedes..." } ],
  "coste": { ...desglose, ver §5.5... } } }
```

Sobre total : se permite **pasarse** del objetivo (pedir 6 → devolver 6+), nunca quedarse corto, salvo que la zona se agote. Los descartes se reemplazan solos descubriendo más.

5. Detalle de cada evento

5.1 · **status** — progreso

Informativo (feedback para el usuario); no transporta resultados. Tres mensajes posibles, todos con `stage: "discovery"` :

message	Significado
«Analizando la formación y buscando empresas candidatas...»	Arranque. Se emite una vez , al principio.
«Lote N: buscando 10 empresas (excluyendo K ya vistas) — X/Y encontradas...»	Empieza un nuevo lote. N =nº de lote, K =nombres excluidos, X/Y =válidas/objetivo.
«No se encontraron más empresas nuevas en la zona.»	La zona se agotó (2 lotes seguidos sin candidatas nuevas). Tras esto llega complete .

5.2 · **discovery** — lote descubierto

```
{ "total": 5, "empresas": ["Nombre A", "Nombre B", ...] }
```

Nombres de las candidatas **recién descubiertas en ese lote** (aún sin ficha). Cada nombre acabará produciendo un **company** (válida) o un **discarded** (descartada). Se emite un **discovery por lote**.

5.3 · **company** — empresa válida (ficha completa)

```
{
  "nombre": "Quántico Soluciones Tecnológicas S.L.",
  "cif": "B23001234", // string | null
  "web": "https://quantico.es", // string | null
  "linkedin": "https://www.linkedin.com/company/...", // string | null
  "telefonos": ["+34 953 123 456"], // lista (vacía si no hay)
  "emails": ["info@quantico.es"], // lista (vacía si no hay)
  "num_empleados": "11-50", // rango de LinkedIn | null
  "sedes": [ // TODAS las sedes (coords geocodificadas)
    {
      "direccion": "C/ Hurtado 23, 1ºA", "ciudad": "Jaén",
      "provincia": "Jaén", "pais": "España", "codigo_postal": "23001",
      "coordenadas": { "lat": 37.766352, "lng": -3.787577 },
      "en_zona": true // ¿dentro de la zona pedida?
    }
  ],
  "descripcion": "Diseño y desarrollo web, software a medida...",
  "justificacion": "Sus actividades encajan con el ciclo DAW...",
  "match_porcentaje": 92, // entero 0-100 | null
  "fuentes": [ { "url": "https://quantico.es/contacto", "titulo": "Contacto" } ],
  "descartada": false, // en company siempre false
  "motivo_descarte": null
}
```

Campo	Significado
cif	CIF/NIF. null si no se localiza de forma fiable (no se inventa).
sedes[]	Todas las sedes, con dirección, provincia y coordenadas (Mapbox en código , nunca por el LLM).
sedes[].en_zona	true = sede dentro de la zona (seleccionada); false = otra sede disponible.
match_porcentaje	% de encaje formación ↔ empresa (0–100).
fuentes[]	Páginas realmente consultadas (citas), no auto-reportadas por el modelo.
descartada / motivo_descarte	En un company siempre false / null .

Los campos no localizables vienen a **null** / **[]** y no se inventan. CIF y código postal son los más propensos a quedar vacíos.

5.4 · **discarded** — empresa descartada por guardrail

```
{ "nombre": "ITSICAP", "motivo": "Ninguna de sus sedes está dentro de la provincia de Jaén." }
```

El enrichment descarta empresas fuera de la zona o no verificables (posibles alucinaciones). Las descartadas **no** cuentan hacia el objetivo, **sí** se excluyen en lotes posteriores, y aparecen resumidas en `complete.descartadas`. Sólo trae `nombre` y `motivo`.

5.5 · **complete** — fin de la búsqueda

Es el **último** evento en una ejecución correcta. El `coste` es real (medido del uso de tokens), desglosado en dos fases: `busqueda` (discovery) y `enrichment`.

```
{
  "total": 6,
  "empresas": [ /* fichas completas, igual que el evento company */ ],
  "descartadas": [ { "nombre": "ITSICAP", "motivo": "Ninguna de sus sedes..." } ],
  "coste": {
    "coste_usd": 0.5123, // total = busqueda + enrichment
    "busqueda": {
      "modelo": "gpt-5.4-mini", "coste_usd": 0.0712,
      "coste_input_usd": 0.0123, "coste_output_usd": 0.0456,
      "coste_web_search_usd": 0.0, "coste_serper_usd": 0.0015, "coste_fetch_usd": 0.0,
      "input_tokens": 16400, "cached_input_tokens": 2048, "output_tokens": 10130,
      "web_searches": 0, "serper_calls": 2, "fetch_calls": 4, "llamadas_api": 3,
      "precios": { "input_per_m": 0.75, "cached_input_per_m": 0.075, "output_per_m": 4.50,
        "web_search_por_llamada": 0.01, "serper_por_llamada": 0.00075, "fetch_por_llamada": 0.0 }
    },
    "enrichment": {
      "modelo": "gpt-5.4-mini", "coste_usd": 0.4411,
      "coste_input_usd": 0.1200, "coste_output_usd": 0.2800,
      "coste_web_search_usd": 0.0, "coste_serper_usd": 0.00375, "coste_fetch_usd": 0.0,
      "input_tokens": 160000, "cached_input_tokens": 40000, "output_tokens": 62000,
      "web_searches": 0, "serper_calls": 5, "fetch_calls": 12, "llamadas_api": 7,
      "precios": { /* mismas claves que en busqueda */ }
    }
  }
}
```

Clave (por fase)	Significado
<code>coste_usd</code> (raíz)	Total = <code>busqueda.coste_usd</code> + <code>enrichment.coste_usd</code> .
<code>coste_input_usd</code> / <code>coste_output_usd</code>	Coste de tokens de entrada / salida (salida incluye <i>reasoning</i>).
<code>coste_web_search_usd</code>	<code>web_search</code> nativo de OpenAI (\$0.01/llamada). 0 si se usa Serper/fetch.
<code>coste_serper_usd</code>	Llamadas a <code>buscar_web</code> (Serper, \$0.00075/llamada).
<code>coste_fetch_usd</code>	Llamadas a <code>leer_url</code> (fetch de FQsource, gratis por defecto).
<code>web_searches</code> / <code>serper_calls</code> / <code>fetch_calls</code>	Nº de llamadas a cada herramienta.
<code>llamadas_api</code>	Nº de llamadas a la API del modelo en esa fase.
<code>precios</code>	Tarifas aplicadas (por 1M tokens y por llamada de tool).

Referencia empírica con `gpt-5.4-mini` + Serper/fetch (sin `web_search` nativo): **~\$0.07–0.10 por empresa**.

5.6 · **error** — fallo

Si llega un `error`, es el **último** evento y el servidor cierra la conexión.

message	Causa	Cierre
«El servidor no tiene API_KEY configurada; el WebSocket está deshabilitado.»	El servidor arrancó sin API_KEY .	1008
«API key inválida o ausente.»	Falta la key o no coincide.	1008
«Petición inválida: ...»	JSON mal formado, falta formacion / ubicacion , o valores fuera de rango.	normal
«<mensaje de la excepción>»	Fallo inesperado durante la búsqueda.	normal

6. Comportamiento clave (cliente)

- **Bucle de escucha:** envía 1 mensaje y luego itera leyendo eventos hasta complete o error (ambos terminales).
- **Streaming, orden no garantizado:** los company / discarded llegan según terminan los enrichments, intercalados con status / discovery de lotes posteriores. Usa complete.empresas como verdad final.
- **Solapamiento de lotes:** verás varios status / discovery antes de que terminen todas las fichas del lote anterior. Es normal.
- **Overshoot:** total puede superar max_empresas (nunca quedarse corto, salvo zona agotada).
- **Deduplicación:** el servidor no repite empresas ya emitidas ni las de excluir .
- **Coordenadas y descarte por zona son deterministas** (Mapbox en código), no dependen del LLM.

7. Cliente mínimo de referencia (Python)

```
import asyncio, json, websockets

async def main():
    url = "wss://web-production-addac.up.railway.app/company-search?api_key=TU_API_KEY"
    req = {
        "formacion": {"url": "https://iesfernandoiii.es/.../daw/"},
        "ubicacion": {"texto": "provincia de Jaén, España"},
        "max_empresas": 6,
    }
    async with websockets.connect(url, max_size=None, ping_interval=20) as ws:
        await ws.send(json.dumps(req))
        async for raw in ws:
            ev = json.loads(raw); tipo, data = ev["type"], ev.get("data", {})
            if tipo == "status": print(".", data["message"])
            elif tipo == "discovery": print("descubiertas:", data["empresas"])
            elif tipo == "company": print("✓", data["nombre"], data["match_porcentaje"], "%")
            elif tipo == "discarded": print("x", data["nombre"], "-", data["motivo"])
            elif tipo == "complete": print("FIN:", data["total"], ". $", data["coste"]["coste_usd"]); break
            elif tipo == "error": print("ERROR:", data["message"]); break

    asyncio.run(main())
```