

# Jesús Moral Aranda

✉️ jesusmoralaranda@outlook.es jesus.moral-aranda@valeo.com  
+34 675 94 54 75  
Location: Martos  
Age: 24

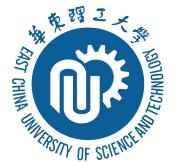


## Education

- University of Seville  
B.Sc. in Physics (expected 2025).  
B.Sc. in Materials Engineering (expected 2025).



- East China University of Science and Technology (Shanghai, China)  
Training in pattern recognition algorithms, artificial intelligence, and advanced materials.



- Gdańsk University of Technology (Gdańsk, Poland) (2024-2025)  
Erasmus exchange. Graduate coursework in nanotechnology: quantum chemical computing, quantum mechanics, microscopy techniques, and materials physics.



## Skills

- English (B2).
- Programming and tools: Python, MATLAB, Linux, LaTeX, basics on HTML, JS, CSS and DB. EPROC, PUMA, SRM, STR, PLM.
- Experience with 3D printing and design (FreeCAD and Fusion 360 Autodesk).

## Work Experience

- Referee, Spanish Football Federation (3 years).
- Research and physics laboratory assistant, ECUST (Shanghai).
  - Project Purchasing Trainee, Valeo Martos (5 months).

# Software development miniprojects at Valeo:



## Purchasing manager autofill tool

Datos Generales Oferta EPROC PO Packing List TAR Etiquetas Tracking

ID EPROC:  
MP98043409

OI EPROC:  
104200002718

Short Descriptions

V2513H-06A-NHN-FNC-V	V25090718VALEO_2513H-06A-NHN-FNC-V-conector VW ID1
2013TL-PHN1-V	V25090718VALEO_2513L-PHN1-V-conector VW ID1
2013C-06B-NHN-FNC-V	V25090718VALEO_2513C-06B-NHN-FNC-V-conector VW ID1
V2513H-06A-NHN-FNC-V	V25090718VALEO_2513H-06A-NHN-FNC-V-conector VW ID1
2013G-06B-NHN-FNC-V	V25090718VALEO_2513G-06B-NHN-FNC-V-conector VW ID1

Web interface (main view)

<https://drive.google.com/file/d/1vq88cMLhfT80mDa8-0zI298sVb57ovTB/view?usp=sharing>

## Analysis of non-fraudulent contracts

Art.	pathname	ponente	comis.	s.	e.	i.	n.	w.	Parágrafo (Contrato)	Parágrafo (Estantilla)	Palabras
1	1_350_258	1_351_1	2025	1_350_7	1_351_12	1_351_12	104_184	105_133	40_49	5	5 / 5C
1	2_374_514	1_372_1	5/5	2_374_10	1_374_9	1_374_9	271_77	271_78	24_29	50	38 / 34128
1	3_395_938	1_396_1	5/5	3_395_10	1_395_10	1_395_10	244_181	245_121	36_39	7	7 / 7F
4	4_757_75	1_758_1	5/5	4_757_12	1_757_12	1_757_12	207_28	207_29	36_36	7	7 / 7F
1	5_123_121	1_125_1	5/5	5_123_12	1_123_12	1_123_12	167_73	167_73	137_13	3	3 / 3C
4	6_461_58	1_461_1	5/5	6_461_10	1_461_10	1_461_10	207_28	207_29	36_36	4	4 / 4C
2	7_188_148	1_188_1	5/5	7_188_14	1_188_14	1_188_14	160_74	167_75	125_23	9	9 / 9N
8	8_150_118	1_151_1	5/5	8_150_11	1_150_11	1_150_11	160_74	167_75	125_23	2	2 / 2C
8	8_240_104	1_241_1	5/5	8_240_10	1_240_10	1_240_10	160_74	167_75	125_23	2	2 / 2C
18	18_367_18	1_368_1	5/5	18_367_18	1_367_18	1_367_18	160_74	167_75	125_23	2	2 / 2C
13	18_367_18	1_368_1	5/5	18_367_18	1_367_18	1_367_18	160_74	167_75	125_23	2	2 / 2C
12	18_367_183	1_368_1	5/5	18_367_18	1_367_18	1_367_18	160_74	167_75	125_23	90	90 / 96100
13	1_360_513	1_416_1	5/5	1_360_51	1_415_10	1_415_10	402_441	399_798	202_382	5	5 / 5C
14	1_363_560	1_364_1	5/5	1_363_56	1_363_56	1_363_56	402_441	402_441	24_23	5	5 / 5C
15	1_793_86	1_812_1	5/5	1_793_86	1_812_1	1_812_1	217_73	217_73	217_73	17	28 / 37208

## VLS AI Working Group: Metaprompts for LEA

<https://drive.google.com/file/d/11MqypvAvkJLrfX2JnZEsxuEvFIKmlL1M/view?usp=sharing>

**Meta Prompting for AI Systems**

Yifan Zhang<sup>1</sup> Yang Yuan<sup>1,2</sup> Andrew Chi-Chih Yao<sup>1,2</sup>  
<sup>1</sup>IIS, Tsinghua University <sup>2</sup>Shanghai Q1 Zhi Institute  
yifanzhangresearch@gmail.com, yangyang@tsinghua.edu.cn,  
andrewccyao@tsinghua.edu.cn

**Abstract**

We introduce Meta Prompting (MP), a framework that elevates the reasoning capabilities of large language models (LLMs) by focusing on the formal structure of a task rather than content-specific examples. We establish a theoretical foundation for this paradigm, formalizing MP as a process of generating and refining prompts based on the structure of a task, thereby guaranteeing that compositional problem-solving strategies can be systematically decomposed into modular prompt structures. We extend this concept to Recursive Meta Prompting (RMP), an automated process where an LLM can generate and refine its own prompts. We model this as an iterative loop following as a module, providing a principled framework for automated prompt engineering. Our claims are validated through experiments demonstrating that a Queen-T2B base model, guided by a single, example-agnostic meta-prompt, achieves state-of-the-art results on MATH, GSM8k, and Game of Thrones. These results are achieved with substantial token efficiency gains over traditional few-shot methods.

Project Page: <https://github.com/meta-prompting/meta-prompting>

**Task\_I:**

Task\_I - 1st PROMPT UNDER TESTING @Ever

Task\_I is a category C<sub>3</sub>

**Ob(Task\_I):**

Task\_1:= cost analysis  
Task\_2:= comparison of supplier's offer for a part  
Task\_3:= target offer

Canonical adapters:  $\alpha_{2-1} : b_2 \rightarrow l_1, \quad \alpha_{3-2} : l_3 \rightarrow b_2, \quad \alpha_{3-1} : l_3 \rightarrow l_1$ .

Homs:  $\text{Hom}_{\text{Task}}(T_1, T_2) = \{ \beta_{12} : b_1 \times O_1 \rightarrow O_2 \},$   
 $\text{Hom}_{\text{Task}}(T_2, T_3) = \{ \beta_{23} : b_2 \times O_2 \rightarrow O_3 \},$   
 $\text{Hom}_{\text{Task}}(T_1, T_3) = \{ \beta_{13} : b_1 \times O_1 \rightarrow O_3 \}$

Composition:  $(\beta_{23} \circ \alpha_{12})(x_1, o_1) = \beta_{23}(x_1, \beta_{12}(\alpha_{3-2}(x_1), o_1))$ .

**Diagram:**

```

graph LR
    subgraph Task_I [Task_I]
        direction TB
        A[Task_I - 1st PROMPT UNDER TESTING @Ever] --> B[Task_I is a category C3]
        B --> C[Ob(Task_I)]
        C --> D[Task_1:= cost analysis]
        C --> E[Task_2:= comparison of supplier's offer for a part]
        C --> F[Task_3:= target offer]
        D --> G[Canonical adapters: alpha_21:b2->l1, alpha_32:l3->b2, alpha_31:l3->l1]
        G --> H[Homs: Hom_Task(T1,T2)={beta_12:b1xO1->O2}, Hom_Task(T2,T3)={beta_23:b2xO2->O3}, Hom_Task(T1,T3)={beta_13:b1xO1->O3}]
        H --> I[Composition: (beta_23 o alpha_12)(x1,o1)=beta_23(x1,beta_12(alpha_32(x1),o1))]
    end

```

**Flowchart:**

```

graph LR
    TASK[TASK  
• input_i_info  
• output_i  
• meta_prompt] --> INCREMENT[INCREMENT  
• increment_i  
• increment_o]
    INCREMENT --> LEA[LEA]
    LEA --> OUTPUT[output_i]
    OUTPUT --> MEMORY[save to memory]
    LEA --> SELF_I[SELF_I  
• self_i_info  
• self_i]
    SELF_I --> INCREMENT

```