Product Homologation Pipeline Client Documentation

This document provides a detailed, client-ready explanation of how the product homologation system works including how sizes, categories, packaging, and brands are extracted and evaluated according to your strict business rules.

---

1. Overview

The system takes messy product descriptions from distributor files, cleans and processes them, and then attempts to match them to your product master using a combination of:

- Size and unit extraction
- Packaging and quantity recognition
- Clean name construction (excluding size/quantity/packaging)
- Category mapping via dictionary
- Fuzzy name matching
- Brand presence validation
- Size-first match rule enforcement

---

2. Extraction Logic

a. Sizes, Quantities, Packaging

We use a robust extraction engine to identify patterns such as:
- Sizes: `250 ml`, `2.5 L`, `5 kg`, `120 gr`, `0.5lt`
- Quantities: `x12`, `*8`, `12und`, `3uni`
- Packaging types: `bolsa`, `botella`, `pqt`, `fco`, etc.

All these elements are **removed** from the name to generate a clean version suitable for fuzzy matching.

The extracted elements are:
- `sizes`: list of standardized sizes
- `quantities`: quantities in formats like x12 or *4
- `packaging`: packaging type if detected
- `clean_name`: original name with above removed

---

b. Category Extraction

Category is extracted **using your provided dictionary**:
- File: `Dictionary.xlsx`
- Sheet: `"CATEGORIAS"`
- Columns used: `"ABREVIATURA (Abbreviation)"  "CATEGORIA_PROD"`

Logic:
- For each clean distributor name, we look for a match with any abbreviation in the dictionary.
- If found, the corresponding `CATEGORIA_PROD` is assigned as the normalized category.

Example:
- If the word "jabon" exists in the name and is mapped to `CUIDADO PERSONAL`, we assign `CUIDADO PERSONAL`.

---

c. Brand Validation

We do **not extract brand** from distributor names, but instead:
- Use the brand from the matched master row.
- Check if that brand keyword exists in the original distributor name.
- If it does `Brand Match = True`
- If not `Brand Match = False`

This ensures consistency with your existing master brand naming.

---

3. Matching Logic

- We clean both distributor and master names (remove sizes, quantities, special chars).
- We apply **fuzzy matching** (`fuzzywuzzy.token_sort_ratio`) between distributor `clean_name` and master `clean_name`.
- We extract:
  - Best fuzzy match
  - Fuzzy score

---

4. Business Rules Applied

1. **Size is mandatory**
- If **no size match**, we do **not** perform homologation even if:
  - Brand matches
  - Category matches
  - Fuzzy score is high

2. **Category match is preferred**
- If the category from the name (via dictionary) matches the masters category, it boosts confidence.

3. **Brand match required**
- We validate brand by checking if the master brand keyword is in the distributors original name.

4. **Existence in Master**
- We also validate if this specific combination (brand, category, size) **actually exists in your master**
- If not, we set:
  ```Exist in Product Master? = NO MATCH```

---

5. Confidence Labels

Based on fuzzy score and validation checks, we assign:
Fuzzy Score	Size Match	Brand Match	Confidence
90			HIGH
75		/	MEDIUM
  ```

| 50 | | | LOW |
| Any | | | NO MATCH |

---

6. Output

- Main output Excel saved in `output/matched_*.xlsx`
- Unmatchable or empty products saved in `output/invalid_products.xlsx`
- Each row contains:
  - Extracted Sizes
  - Clean Name
  - Fuzzy Match and Score
  - Category/Brand Match
  - Size Match
  - Final Match Confidence
  - "Exist in Product Master?" column

---

7. How to Run

1. Edit `config.yaml` with your files:
```yaml
distributor_file: "data/Base homologacin Clorox.xlsx"
master_file: "data/base_homologacion_ml.xlsx"
master_sheet: "Maestro General"
master_description_column: "DescripcionProducto"
dictionary_file: "data/Dictionary.xlsx"
output_file: "output/matched_clorox.xlsx"
```

2. Run:
```bash
python runner.py --config config.yaml
```

Summary

This pipeline is strictly aligned to your business logic:
- Size match comes first
- No homologation without real evidence in master
- Transparent logs and config-driven setup